ALIGNED DATA MODELS FOR PREDICTIVE ANALYTICS

by

Xin Ji

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Financial Services Analytics

Summer 2018

ALIGNED DATA MODELS FOR PREDICTIVE ANALYTICS

by

Xin Ji

Approved: _____

Bintong Chen, Ph.D.
Chair of the Institute of Financial Services Analytics

Approved: _____

Bruce Weber, Ph.D.
Dean of the College of Business and Economics

Approved: _____

Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Adam Fleischhacker, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Yi-Lin Tsai, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Xiao Fang, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Errol Lloyd, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Pak-Wing Fok, Ph.D.
Member of dissertation committee

ACKNOWLEDGMENTS

I owe my deepest appreciation to my advisor, Dr. Adam Fleischhacker, who supports me all the way from the very beginning till the end of my journey with the Institute of Financial Services Analytics. He is the most encouraging person I've ever met. He always shines like a light, guiding me to walk through the mist. It has been such a wonderful four years of studying and working with Adam. It is my great honor and luck to be his student.

To my dissertation committee, I express great thanks to Dr. Yi-Lin Tsai for his generous time and help on my first publication, to Dr. Xiao Fang for his supervision on the health care project and his continuing support to improve my work, to Dr. Errol Lloyd for his encouragement and advices on editing my thesis, to Dr. Pak-Wing Fok for his comments and suggestions. Besides my dissertation committee members, I am very grateful to Dr. Yukai Lin's generous sharing of the data he collected.

I would like to extend my thanks to friends and colleagues who genuinely care and help, especially Xiaohang Zhao. His generous help and tutoring in programming and insightful suggestions on model building mean a lot to me.

I also want to express special thanks to my family: to Dr. Yijie Jiang, my darling who unconditionally supports me through my education and shared many household work and baby sitting to spare me precious time; to Nova, my beloved little one, who comforts me when I am frustrated and lowers my anxiety and stress level before I drive myself nuts; to my dearest mom for supporting me through an extended period of time with family growing, her generous sacrifice of time and life convenience is second to none.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This dissertation is a study of three application areas where the underlying data is misaligned with the question that is being asked of it. In these three problems, novel transformations and/or structuring of the captured data are deployed prior to any modeling efforts. Subsequent modeling steps are customized to take advantage of the transformed data and to reveal insights desirable to a firm querying their captured data sources. We refer to the manipulated data and the associated modeling techniques as *aligned data models*.

The first aligned data model developed in this dissertation seeks to model the concept of uniqueness and explore the predictive capabilities of representing uniqueness within a modeling process. Two application areas are explored: 1) *predicting competition intensity* where the more unique offerings of a firm are used to differentiate competitors and 2) *predicting purchase behavior* where the unique attributes of customers are presumed to make them more likely to exhibit similar purchase behavior. This study develops a uniqueness-motivated probabilistic similarity measure, extending the existing literature on similarity measures, to capture the notion of uniqueness and show its value in applications. The developed method can capture uniqueness for various kinds of data including measures that are numeric, categorical, hierarchical, and distance-based.

The second aligned data model focuses on identifying areas where the intensity of bivariate relationships becomes interesting to a decision maker. Since linear relationship assumptions do not vary over the domain of the explanatory variable, this study examines how to model a bivariate non-linear relationship so that areas of relationship intensity that are of interest to the decision maker can be uncovered. These areas might

be regions of no relationship, regions of an intense positive/negative relationship, or regions within a defined interval of intensity.

To facilitate this mining for relationship intensity, we developed the R package *gpviz* and its underlying methodology. The package learns non-linear relationships using Gaussian process and numerically approximates a distribution of relationship intensity by using the first derivatives of a Gaussian process prediction model. By visualizing the distribution of first derivatives, users immediately have access to subtleties of any non-linear bivariate relationship. Through specifying the bounds of relationship intensity, users can identify regions of interest (i.e. values of the explanatory variable) that are particularly pertinent to their decision making. Example applications are shown including one where data from failed commercial banks is used to identify ranges of a bank's deposit-asset ratio where FDIC losses, associated with a bank's failure, tend to increase dramatically.

The third aligned data model extracts insight from problems where particular sequences of hierarchically-structured activities, not just the presence of activities, provides predictive power for future events. In this study, we highlight an application of great value to medical practitioners and their patients; namely, surgical adverse event prevention. Here the model used seeks to predict the probability of hospitalized patients experiencing surgical adverse events. Given that activities in this setting have an underlying hierarchical relationship, we reshape the data then leverage a dynamic recurrent neural network to incorporate this data structure and show that it is capable of improving prediction performance.

In summary, this dissertation focuses on proposing data models which align to the needs and insights of the problems they seek to address. As opposed to workflows that mine associations from captured data, this study motivates all modeling decisions with real needs prior to performing any inference or algorithmic processes. The three parts of this study all differ slightly in focus and hence, contribute to a well-rounded study of making aligned data models. The uniqueness work is methodologically focused, the non-linear work is computationally focused, and the activity-sequence work

is more application driven. Taken together, this work proposes several useful machine learning models and a sophisticated software package all designed to solve real problems by better aligning a data model with the insights that are to be extracted.

Chapter 1

INTRODUCTION

Decision maker's often believe (falsely) that insights and results derived from data are simply a matter of supplying data to a computer, running an algorithm, and then interpreting the results. This practice, often referred to as data dredging, can provide very limited interpretability and/or insights. Instead of mining gold, data dredging leads to a lot of mud [2]. To improve upon data-dredging, aligning data models with real-world needs and causal mechanisms becomes critical for data analytics success.

This dissertation explores methods of aligning data models with objects or interpretations in the real-world. This work includes proposing new data structures and methods that echo real-world interpretability, developing toolkits to visualize and understand relationships that are known to have non-linear relationships in the real-world, and converting data representations of real-world events to make better sense of structural information for predicting adverse outcomes.

Each chapter of this dissertation has its own introduction and relevant background associated with a problem or class of problem that a decision maker might face. The modelling of each decision scenario is done in such a way so that novel alignment between data/methods and the real-world interpretatability or usefulness is made clear - we term this an *aligned model*. For each aligned model, interpetability and performance comparisons are used for validation.

In Chapter 2, a uniqueness-motivated similarity measure is proposed which compares instances using feature value uniqueness as a point of comparison. Using this measure, two instances (e.g. people, items, observations) are considered more similar

when the features they share are more unique (e.g. two opera singers are more related than two cashiers). In Chapter 3, the proposed uniqueness-motivated similarity measure is applied to two business problems, one being competitor identification in a local market, and the other being customer segmentation in airline traveler's ancillary purchase prediction. The first application is an unsupervised learning problem while the second application has a true label, i.e. airline traveler's purchase indicator. Good performance is demonstrated through comparison with traditional marketing solicitation methods in the unsupervised case, and through comparison with other popular machine learning algorithms in the supervised case.

Chapter 4 is motivated by the need to understand and find notable relationship regions - areas where the relationship intensity between explanatory variable and dependent variable satsify some criteria. The relationships explored in this chapter are assumed non-linear and Gaussian process regression (GPR) is expanded to enable the discovery of remarkable regions of relationship intensity. The chapter starts with deriving relevant distribution properties of a GPR prediction's first derivative; the first derivative becomes a proxy for the instensity of the relationship at a specified value of the independent variable(s). To computationally implement and operationalize region identification, an R package, **gpviz**, is developed for Gaussian process modelling and visualizing plausible predictions of relationship intensity. The package is subsequently tested on data regarding failed banks.

Chapter 5 aligns hybrid neural network model consisting of multi-layer perceptrons and dynamic recurrent neural network with hospitalized patient's admission data and surgical procedure data to predict the probability of hospitalized patients going through surgical adverse events. The hybrid neural network model handles both static data and sequential data, we also reshape each surgical procedure code to incorporate its hierarchical strucuture. Empirical experiements have shown its superior predictive accuracy and performance.

Chapter 2

UNIQUENESS-MOTIVATED SIMILARITY MEASURE

2.1 Introduction

Uniqueness is a key source of competitive advantage and an important attribute to customers' decision making. Idiosyncratic resources and unique capabilities of companies help firms attain competitive advantage in their marketplaces [3]. In fact, most contemporary strategic management approaches highlight the role of uniqueness in determining a firm's competitive position [4, 5, 6, 7].

Given that uniqueness is instrumental in determining competitive position, we propose a uniqueness-motivated probabilistic similarity measure to quantify the intensity of competition among associated firms. The intuition is that business entities who share unique attributes are considered closer competitors than businesses that share more common features. In a seminal piece, Keller's conceptual model of brand equity asserts that it is the *unique* associations that consumers make with brands that create value for a brand [8]. For measuring uniqueness, Keller proposes two methods: 1) a direct method asking consumers what they consider to be unique about a brand, and 2) an indirect method of comparing characteristics among competitors. In this thesis, we seek to automate the indirect method using readily available social media data and then validate that this method's results are consistent with those of the direct method of asking consumers.

Automation of competitor identification based on uniqueness has the promise of replacing, or at least supplementing, the more laborious efforts of marketers who use surveys and focus groups [9, 10]. Social tags contain valuable information about the value of a firm and its competitiveness and brand strength [11]; as such, social tags

have become a popular data source to automate competitor identification. For example, Nam et al. [12] propose a method to analyze user-generated social tags associated with brands and help marketing managers derive distinctive insights from the information contained in social tags.

Our proposed metric enables the automation of competitor identification in two ways: 1) data collection: the proliferation of social media data enables automation in data collection (e.g. web scraping) for traditional marketing research methods (see [13, 14] for example); 2) application: our uniqueness-motivated similarity measure can be combined with existing clustering algorithms and readily accessible social media tags, ratings, etc. to view firms through the lens of uniqueness and produce a local competition landscape.

In terms of methods to derive the market structure from similarity data, most of the relevant studies utilize some variant of cluster analysis, such as k-means [15], k-medoids [16], or hierarchical clustering techniques [17]. This thesis leverages hierarchical clustering for its ability to visualize the data via dendrogram output. Other studies leveraging visualization for market structure include Ringel and Skiera [18] which develops an approach to generating insights into competition asymmetries; Lee and Bradlow [13] which creates visualizations to describe market structure derived from online customer reviews; and France and Ghose [19] which visualizes brand competition in a restaurant context that leverages Yelp review instances.

Our contributions to the literature include creating a uniqueness-motivated similarity measure that accommodates the multitude of data types found in readily available data, leveraging this similarity measure within a clustering algorithm to automate the process of competitor identification, and validating the utility of our proposed method in assessing local competition by comparing the calculated landscape against the perceptions of surveyed consumers. For this validation step, we analyze competition in the restaurant industry, an industry which contains an abundance of readily-available data (e.g. Yelp, Trip Advisor, Google Reviews, etc.) and which has shown that uniqueness of both resources and capability prove critical to sustaining competitive advantage [20].

2.2 Related Background

2.2.1 Similarity Measure

Similarity measures are typically constructed for either numeric or categorical data. The most popular numerical measure of similarity is Euclidean distance, but numerous other measures can be used to capture different relationships [21]. In terms of categorical data, since no inherent order is available, a contingency table is usually produced based on all the categories. Subsequently, a similarity measure is constructed using either probabilistic or information/entropy-based calculations to fill the entries of the contingency table. Performance of these similarity measures are found to be directly related to the characteristics of the data set and no similarity measure can claim superiority all of the time [22].

A third type of data, mixed data, contains both numerical and categorical features. When comparing mixed data, the literature typically measures similarity as a weighted sum of the numeric and categorical portions of the data set. Alternatively, categorical variables can be transformed into binary vectors and a binary similarity function such as simple matching (Hamming distance), Rogers and Tanimoto measure, Gower and Legendre measure, Jaccard coefficient, or the Sokal and Sneath measure may be used [23]. Among all the matching similarity measures, the Jaccard coefficient is perhaps the most popular, but this measure is known to perform poorly in cases where data points are related and not well separated as is the case in studying the attributes of close competitors [24, 25].

The notion of uniqueness is addressed in many similarity measures that operate only on categorical measures. Smirnov [26] assigns similarity scores based on the observed attribute value distribution. Lin and Brown [27] proposes a similarity measure based on occurrence probability of each word in a way that less frequent words have higher information gains. Markines et al. [28] review information-theoretic similarity measures applied in social bookmarking applications. Han et al. [29] uses the inverse logarithm of N (the number of users who share a particular interest) as weights to formulate a similarity measure that evaluates the similarity between two individuals'

social interests. Han's work is in the same spirit of our work, but our purposes require more than just using categorical data to measure similarity.

Methods of integrating mixed data measures remain understudied. One exception is the probabilistic similarity index proposed by Goodall [30] for use in an ecology context and which we extend in this thesis. In Goodall's approach, two "individuals" (or species) are deemed "similar" by sharing features with unique values. For example, if we look at the presence of pouches in various animal species (e.g. as in kangaroos and koalas), this feature is quite unique and hence, implies a closer relationship among species that share this feature. Goodall further develops this notion to measure similarities between individuals with respect to the simultaneous consideration of multiple features (e.g. eye color, height, and gender). Our proposed measure extends Goodall's work to new data types not found in an ecological context.

Meanwhile, it is worth mentioning that many outlier-detection algorithms will use uniqueness or distinctiveness [31, 32], especially those works seeking to identify outliers. Data items in small groups considered as outliers are often removed to make clustering outcomes more reliable and robust against data perturbations [33]. Our thesis values uniqueness due to the nature of competitor identification and competitive advantage, not for outlier detection. We don't view the competitors with unique characteristics as outliers or noise, on the contrary, we use the uniqueness of competitors to construct the competition landscape.

2.2.2 The SBAC Algorithm

Li and Biswas [34] adopt Goodall's similarity measure within a clustering algorithm called the Similarity-Based Agglomerative Clustering (SBAC) algorithm. The SBAC algorithm allows for clustering based on both numeric and categorical data. It has demonstrated superior performance in unsupervised learning tasks and shown no noticeable bias toward either numeric or categorical features.

This thesis extends the work in Li and Biswas [34] by accommodating two-types of features with relational structures that are not included in their work or the

original work by Goodall. In SBAC, each feature has one value for each data object; for example, hair color for an individual. In this thesis, multi-valued features are accommodated; for example, a Yelp description of McDonald's may consist of two tags: "burgers" and "fast food". To conceptualize this idea, we refer to this multi-valued feature structure as a *bag-of-tags*. Then, our goal is to measure the similarity between two bags-of-tags. We follow the established folksonomy literature (see [35] for more details) and define similarity between two data objects' bag-of-tags based on relatedness indicated by co-occurrence and uniqueness of values. For example, restaurant tag "Italian" is more similar to the restaurant tag "pizza" rather than the tag "sushi" because the former pair "Italian" and "pizza" co-occurs in the population of restaurants while the latter pair "Italian" and "sushi" does not co-occur.

The main literature stream to handle the similarity between two *bags-of-tags* is co-occurrence, which is one commonly used criterion to measure similarity between words and tags [36]. The validity of measuring relatedness using co-occurrence dates back to the literature of association rule mining [37]. Most previous measures of tag relatedness are drawn from the folksonomy literature and consist of various co-occurrence statistics [38]. For example, the CACTUS algorithm deems values to be strongly connected if their co-occurrence is significantly higher than the expected value under attribute-independence assumption [39]. Recent literature has adopted this idea for competition identification, see [40] for example. Nam et al. [12] uses co-occurrence patterns of social tags across brands in the tagging networks as the key metric to capture strength between a brand and a tag. And Ringel and Skiera [18] use co-occurrence data of all products to identify the number of existing submarkets and identify the products belonging to each submarket. Our technique also relies on co-occurrence and is similar to the work by Netzer et al. [41] who measures similarity between products based on the frequency of their co-occurrence in forum messages. While there are more complex techniques in the literature that look at marginal frequencies in relation to a set of other features [42] or co-occurrence relationships that exhibit transitive properties [43], the simplicity of using co-occurrence frequency proves sufficient in our context

(e.g. using Yelp tags to assess restaurant competition).

The second type of data we incorporate in our proposed algorithm is a *distance/dissimilarity matrix*, which is a symmetric matrix that describes the relationship between any two data objects. For example, the distance/dissimilarity can be captured by the geographic distance between two restaurants. The shorter the distance, the more likely the restaurants are direct competitors; therefore, we can perceive that the two restaurants may compete for the same clientele. For handling *distance/dissimilarity matrix*, we extend the concepts of Li and Biswas [34] to measure the uniqueness of distance relationships among observed data objects. Related literature includes converting relation matrices to fuzzy sets [44, 45], rough sets [46], and soft spatial weight matrix [47].

2.3 Uniqueness-motivated Probabilistic Similarity Measure

In this section, we extend Goodall's similarity measure to accommodate the *bag-of-tags* and *distance matrix* data types. We start with a quick overview of the Goodall's similarity measure and the SBAC algorithm and then highlight the modifications we introduce. For this section, we will consider two data objects, i and j, which may or may not have the same value for feature k. The domain set consisting all feature values that feature k takes in the sample is denoted as $\mathcal{D}_k = \{k_1, k_2, ...k_m\}$.

2.3.1 Independent Categorical Features

Two data objects i and j are considered maximally dissimilar if they have different values for categorical feature k; when they have the same value for feature k, their similarity measure is a function of the uniqueness of their feature value within the sample: the more unique the shared feature value is, the more similar the two data objects are. Operationally, Li and Biswas [34] defined the *more similar feature value set* (MSFVS) to capture such uniqueness:

$$MSFVS(k_l) = \{k_s \in \mathcal{D}_k : f(k_s) \leq f(k_l)\} \tag{2.1}$$

where $f(k_l)$ denotes the frequency counts of feature value k_l. The *MSFVS* for feature value k_l is the set consisting all feature k's values that occur less or equally frequently as feature value k_l.

To mathematically represent uniqueness, probability is used. The probability of picking two data objects i and j whose values of feature k happen to be identical is $P(k_l)^2 = \frac{f(k_l)(f(k_l)-1)}{n(n-1)}$, where n is the sample size, and the dissimilarity measure becomes the summation of all the probabilities associated with feature values included in their *MSFVS*; if data objects i and j have different values for feature k, their dissimilarity is one:

$$\begin{aligned} V_k^i \neq V_k^j \implies D_k^{ij} &= 1 \\ V_k^i = V_k^j = k_l \implies D_k^{ij} &= \sum_{k_s \in MSFVS(k_l)} P(k_s)^2 \end{aligned} \tag{2.2}$$

2.3.2 Numeric Features

The similarity measure for numeric features is defined in a similar style as for categorical features. The only difference is when numeric values differ; instead of being considered maximally dissimilar, their dissimilarity measure is a function of their segment length, i.e. the difference in their values.

Given a comparison between two pairs of data objects, (i, j) versus (l, m), if they have identical segment lengths for feature k, $|(V^i)_k - (V^j)_k| = |(V^l)_k - (V^m)_k|$, then the segment with lower cumulative frequency is considered more unique, thus more similar. Following the logic of uniqueness, *more similar feature segment set* (MSFSS) is defined to include all segments of smaller or equal gap but less or equal cumulative frequency [34]:

$$\begin{aligned} MSFSS(k_l, k_m) = \{(k_s, k_t) : (k_t - k_s < k_m - k_l) \quad \vee \\ (k_t - k_s = k_m - k_l \quad \wedge \\ \sum_{x=k_s}^{k_t} f(x) \leq \sum_{x=k_l}^{k_m} f(x)^1, \\ k_l, k_m, k_s, k_t \in \mathcal{D}_k, k_l \leq k_m, k_s \leq k_t\} \end{aligned} \tag{2.3}$$

The probability of picking any two data objects with either identical or different values for feature k is calculated as:

$$\begin{aligned} V_k^i = V_k^j = k_l \implies P(k_l)^2 = \frac{f(k_l)(f(k_l)-1)}{n(n-1)} \\ k_l = V_k^i \neq V_k^j = k_m \implies P(k_l)P(k_m) = \frac{2f(k_l)f(k_m)}{n(n-1)} \end{aligned} \tag{2.4}$$

Under this measure, the probabilities of all possible segments from the sample sum up to 1.

The dissimilarity measure is the summation of the probabilities associated with feature value segments in their *MSFSS*:

$$D_k^{ij} = \sum_{(k_s,k_t)\in MSFSS(V_k^i,V_k^j)} P(k_s)P(k_t) \tag{2.5}$$

2.3.3 Accommodating Features with Relational Structures

Real life data often contains features with relational structures that Goodall's similarity measure is not equipped for and the SBAC algorithm is unable to handle. We extend Goodall's similarity measure to calculate both diagonal entries and off-diagonal entries, taking different but correlated values into consideration as well. One type is *bag-of-tags* as observed in social tagging (e.g. a restaurant tagged with {Italian, pizza}, another type is a distance/dissimilarity matrix as is required to measure distances between locations in a set. The *generalized more similar feature value set* is proposed to accommodate these two relational structures.

2.3.3.1 Accommodating Bag-of-tags

Our uniqueness-motivated probabilistic similarity measure takes tag-tag associations into account and handles the relatedness between two bags by cross comparing

[1] When the probability density function of x is available, $\int_{k_s}^{k_t} p(x)dx \leq \int_{k_l}^{k_m} p(x)dx$ should be used.

tags and aggregating all the comparisons. Any pair of data objects with identical feature value is considered to be more similar than a pair of data objects with different feature values.

$$(V_k^i = V_k^j) \wedge (V_k^l \neq V_k^m) \implies S_k^{ij} > S_k^{lm} \tag{2.6}$$

The similarity measure for two data objects whose feature values are different, but somehow associated, is calculated using co-occurence. Given any two values k_s and k_t, if their co-occurrence frequency is positive, $f(k_s, k_t) > 0$, the two values are considered related, and are no longer considered to have zero similarity. We define the *generalized more similar feature value set* (GMSFVS) as:

$$\begin{aligned} GMSFVS(k_l, k_m) = &\{(k_s, k_t) : (k_s = k_t) \vee \\ &(k_s \neq k_t \wedge (0 < f(k_s, k_t) \leq f(k_l, k_m))) \\ &k_l, k_m, k_s, k_t \in \mathcal{D}_k\} \end{aligned} \tag{2.7}$$

The probability of picking any two data objects with either identical or different values for feature k is calculated as:

$$\begin{aligned} V_k^i = V_k^j = k_l &\implies P(k_l)^2 = \frac{f(k_l)(f(k_l)-1)}{m(m-1)} \\ k_l = V_k^i \neq V_k^j = k_m &\implies P(k_l)P(k_m) = \frac{2f(k_l)f(k_m)}{m(m-1)} \end{aligned} \tag{2.8}$$

where m is different from the sample size n; when the categorical feature is multi-valued such as a bag-of-tags, n refers to the number of bags in the sample while m refers to the total number of tags.

The dissimilarity measure for any pair of tags is the sum of the probabilities of all values in its *GMSFVS*:

$$D_k^{ij} = \sum_{(k_s,k_t) \in GMSFVS(V_k^i, V_k^j)} P(k_s)P(k_t) \tag{2.9}$$

Ultimately, we need to compare the dissimilarity between the *bags-of-tags*. Since each bag may be multi-valued, we compute dissimilarity for each combination of tag

pairs between the two data objects and then, aggregate their dissimilarity measure. In this thesis, we use the softmax function to generate a weighting scheme based on tag pairwise similarities. In probability theory, the softmax function is often used to represent a categorical distribution, also regarded as a generalization of sigmoid function for multiple value attributes [48]. The purpose of using the softmax function on tag pairwise similarities is to amplify the impact of tag pairs embedded with uniqueness. The reason for using weighted average instead of simply using average is because, as the number of tags within *bags-of-tags* grows, the uniqueness embedded in a few tag pairs can be averaged out and hardly be distinguishable.

The dissimilarity is calculated as the weighted summation of the probabilities associated with attribute value segments in their *MSFSS*. Mathematically speaking, consider data object i with bag-of-tags $\{V^i(Tag_1), V^i(Tag_2), ...V^i(Tag_p)\}$ and data object j with bag-of-tags $\{V^j(Tag_1), V^j(Tag_2), ...V^j(Tag_q)\}$, then their dissimilarity is calculated as follows:

$$\begin{aligned} w_{(l,m)} &= \frac{exp(1 - D^{ij}(Tag_l, Tag_m))}{\sum_{s\in\{1,2,...p\},t\in\{1,2,...q\}} exp(1 - D^{ij}(Tag_s, Tag_t))} \\ D^{ij}_{Bag-of-tags} &= \frac{1}{pq} \sum_{l\in\{1,2,...p\},m\in\{1,2,...q\}} w_{(l,m)} D^{ij}(Tag_l, Tag_m) \end{aligned} \tag{2.10}$$

2.3.3.2 Accommodating Distance/Dissimilarity Matrix

In the domain of market competitor identification, where customers come from is often used as a point from which to measure competitiveness. However, since customers may come from multiple sources, defining a single customer origination point becomes impossible when assessing an entire local market with multiple competitors. Therefore, we compute a distance vector between any one data object and all other data objects. Treating each vector as a row in a distance matrix yields a feature that can be represented using a symmetric distance matrix. The diagonal entries of the distance matrix are zeros since they indicate self-distance. Creation of this matrix is automated by crawling restaurant addresses from Yelp.com and calculating distances using Google Maps Distance Matrix API.

Thus, distance becomes a relative quantity between data objects, with its domain set consisting of every value included in the distance matrix $\mathcal{D}_k = \{k_1, k_2, ...k_m\}$. The SBAC algorithm does not have a measure designed for distance/dissimilarity matrix. We propose a *generalized more similar feature value set* (GMSFVS) for this numeric-matrix structure consisting of all the relative quantities that are smaller:

$$GMSFVS(V^{ij}) = \{V \in \mathcal{D}_k : V \leq V^{ij}\} \tag{2.11}$$

Since any value k_l in the distance matrix relates to geographic distance betwen data objects i and j, which is already a pair of objects to be compared with, we compute the probability for picking each value within the distance matrix:

$$V_k^{ij} = k_l \implies P(k_l) = \frac{f(k_l)}{n^2} \tag{2.12}$$

where $f(k_l)$ is the frequency count of feature value k_l in the entire distance matrix, and n is the sample size.

And their dissimilarity measure is to sum up all the probabilities associated with feature values that indicate a closer distance:

$$D_k^{ij} = \sum_{k_s \in GMSFVS(k_l)} P(k_s) = \sum_{k_s \in GMSFVS(k_l)} \frac{f(k_s)}{n^2} \tag{2.13}$$

This distance similarity measure takes closer restaurants as more similar competitors and the dissimilarity of two data objects with the longest distance will reach the maximum of 1.

2.3.4 Aggregating Uniqueness-motivated Similarity Across Multiple Features

For assessing similarity across multiple features, Fisher's χ^2 transformation $\chi^2 = -2 \ln P$ has been shown to work well using data from both continuous and discrete populations. As long as all features have a large variance of distinct observations, Fisher's χ^2 transformation aligns well with our uniqueness-driven mindset [49]. When comparing data objects i and j of multiple features, as opposed to simple averaging

which aggregates dissimilarities across multiple features linearly, Fisher's χ^2 transformation leads to small aggregated dissimilarity when each of the feature dissimilarities between data objects i and j is small. In other words, only when multiple features of data objects i and j are deemed unique and similar will the two data objects be considered unique and similar. Figure 2.1 presents two methods of dissimilarity aggregation over two numeric features where $d1$ represents the dissimilarity measure between two data objects for the first feature and $d2$ represents the dissimilarity measure between two data objects for the second feature. The first aggregation method uses simple averaging and iso-contours of the aggregated dissimilarity measures appear as lines. The second aggregation method is done using a χ^2 transformation; the D^{ij} iso-contours are now pulled towards the origin. The effect of the transformation is to label two data objects sharing a unique value as much more similar than would be done under simple averaging.

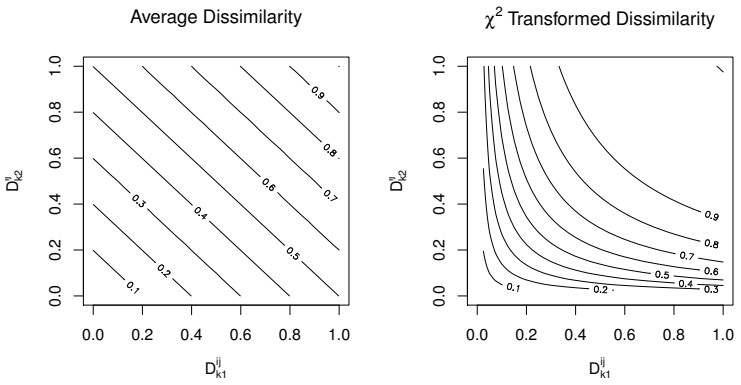


Figure 2.1: Comparison of dissimilarity calculations.

For categorical features, explicit calculations for Fisher's χ^2 transformation along with Lancaster's proposed correction are used to reduce bias [50] (see details in the next section).

2.3.5 Algorithm for Computing Uniqueness-motivated Similarities between Data Objects

Input: Data set $\mathcal{D}$ over a set of mixed attributes $\mathcal{A}$

Output: Dissimilarities between data objects

Begin

Step 1: For each feature $\mathcal{A}_k \in \mathcal{A}$, generate all possible value pairs (k_l, k_m).

- if the value pair contains missing value, $D(k_l, k_m) = 1$;
- otherwise, estimate probabilities of the value pair, $P(k_l)^2$ or $P(k_l)P(K_m)$.

Step 2:

- If feature k is numeric, order all theoretical probabilities by absolute difference (primary key, ascending), frequency (secondary key, ascending);
- If feature k is categorical, order all theoretical probabilities by frequency (primary key, ascending), co-occurrence if feature k is *bag-of-tags* (secondary key, descending).

Step 3: Cumulatively sum up the probabilities following the order in Step 2, for every unique combination of primary key and secondary key. Use the maximum cumulative sum of probabilities as dissimilarity score for value pairs having such primary and secondary key combination.

Step 4: For any data object pairs (i, j),

- for numeric features, $(\chi_c^2)^{ij} = -2\sum_{k=1}^{t_c} \ln D_k^{ij}$, where t_c is the number of numeric features. χ_c^2 follows χ^2 distribution with t_c degrees of freedom;
- for categorical features, especially the ones with a small number of possible observations, Lancaster proposes a modified transformation to reduce bias, *mean value* χ^2 *transformation*:

$$(\chi_d^2)^{ij} = 2\sum_{k=1}^{t_d} \Big[1 - \frac{D_k^{ij} \ln D_k^{ij} - (D_k^{ij})' \ln (D_k^{ij})'}{D_k^{ij} - (D_k^{ij})'}\Big] \tag{2.14}$$

where t_d is the number of categorical features, D_k^{ij} is the dissimilarity between data objects i and j for categorical feature k, $(D_k^{ij})'$ is the next smaller dissimilarity. χ_d^2 follows χ^2 distribution with t_d degrees of freedom;

- for *bag-of-tags*, the set consisting of multiple tags is treated as one categorical feature and use the categorical feature χ_d^2 transformation formula. For *distance/dissimilarity matrix*, we use the numeric feature χ_c^2 transformation formula;

- the combined dissimilarity index can be approximated by:

$$D^{ij} = e^{-\frac{(\chi^2)^{ij}}{2}} \sum_{k=0}^{t_c+t_d+t_r-1} \frac{\left(\frac{(\chi^2)^{ij}}{2}\right)^k}{k!} \tag{2.15}$$

where t_r is the number of features with related structure. The probability distribution after aggregation is χ^2 distributed with $(t_c+t_d+t_r)$ degree of freedom. In 2.3.6, we present an example where the degree of freedom is (t_c+t_d+2) reflecting additional *bag-of-tags* and *distance/dissimilarity matrix* features.

End

2.3.6 Illustrative Example

In the following section, we illustrate how the similarity measures are calculated among 5 different restaurants in Table 2.1. Each restaurant (data object) consists of both categorical and numeric features. The categorical feature contains a bag-of-tags, specifically, a bag of Yelp tags. The numeric features comprise average amounts spent in the restaurant and average ratings of the restaurants. We also observe a distance matrix as in Table 2.2 among the restaurants.

Table 2.1: Sample dataset

Restaurant	Amount	Yelp Tags	Rating
Applebees	15	{ Sports Bars, Burgers, American (Traditional) }	2.5
Buffalo Wild Wings	14	{ Chicken Wings, American (Traditional), Sports Bars }	3
Burger King	7	{ Fast Food, Burgers, Hot Dogs }	2
McDonald's	6	{ Burgers, Fast Food }	3.5
Popeyes	8	{ Fast Food, Chicken Wings }	2

Table 2.2: Distance Matrix (meter)

	Applebees	Buffalo Wild Wings	Burger King	McDonald's	Popeyes
Applebees	0	1973	3286	4060	7600
Buffalo Wild Wings	1973	0	1322	2087	5627
Burger King	3286	1322	0	914	4454
McDonald's	4060	2087	914	0	3540
Popeyes	7600	5627	4454	3540	0

The dissimilarity measure $D(i,j)_k$ between two data objects i and j for feature k is computed respectively as follows:

2.3.6.1 Bag-of-tags

To measure similarity between two bags-of-tags, we cross compare each pair of tags from each vector and average the dissimilarities. Regards to the comparisons between tags, we show two calculation examples, one for comparing identical tags, one for comparing different tags.

First example shows the dissimilarity measure for a pair of identical tags (American (Traditional), American (Traditional)):

$$\begin{aligned} GMSFVS(At,At) =& \{(Hd,Hd),(At,At),(Cw,Cw),\\ & (Sb,Sb)\}\\ D(At,At) =& P(Hd)^2 + P(At)^2 + P(Cw)^2 + P(Sb)^2\\ =& 0 + \frac{2\times 1}{13\times 12}\times 3\\ =& 0.038\\ S(At,At) =& 1 - D_{(At,At)} = 0.962 \end{aligned}$$

Second example shows the dissimilarity measure for a pair of different tags (Burgers, Fast Food):

$$
\begin{aligned}
GMSFVS(Bg, Ff) = \{&(Hd, Hd), (At, At), (Cw, Cw),\\
&(Sb, Sb), (Bg, Bg), (Ff, Ff), (At, Bg),\\
&(At, Cw), (Bg, Sb), (Bg, Hd), (Cw, Ff),\\
&(Cw, Sb), (Ff, Hd), (At, Sb), (Bg, Ff)\}\\
D(Bg, Ff) = &P(Hd)^2 + P(At)^2 + P(Cw)^2\\
&+ P(Sb)^2 + P(Bg)^2 + P(Ff)^2\\
&+ P(At, Bg) + P(At, Cw) + P(Bg, Sb)\\
&+ P(Bg, Hd) + P(Cw, Ff) + P(Cw, Sb)\\
&+ P(Ff, Hd) + P(At, Sb) + P(Bg, Ff)\\
D(Bg, Ff) = &\ 0.667\\
S(Bg, Ff) = &\ 1 - D(Bg, Ff) = 0.333
\end{aligned}
$$

2.3.6.2 Numeric Feature

An example of comparing average spending amount for Burger King and Popeyes:

$$
\begin{aligned}
MSFVS(7, 8) = &\{(6, 7), (7, 8), (14, 15)\}\\
D(7, 8) = &\ 2P(6)P(7) + 2P(7)P(8) + 2P(14)P(15)\\
= &\ \frac{2 \times 1 \times 1}{5 \times 4} \times 3 = 0.3
\end{aligned}
$$

2.3.6.3 Distance/Dissimilarity Matrix

Given the distance feature as a matrix, we compute similarity for each value within the distance matrix. For example, calculating the dissimilarity between Applebees and Buffalo Wild Wings:

$$
\begin{aligned}
GMSFVS(1973) = &\ 0, 914, 1322, 1973\\
D(1973) = &\ \frac{f(0)}{n^2} + \frac{f(914)}{n^2} + \frac{f(1322)}{n^2} + \frac{f(1973)}{n^2}\\
= &\ \frac{5}{5^2} + \frac{2}{5^2} \times 3 = 0.44
\end{aligned}
$$

2.4 Summary

Competitor identification is an important precursor to establish and maintain competitive advantage. This chapter extends a uniqueness-motivated probabilistic similarity measure to accommodate multivariate categorical (*bag-of-tags*) and numerical (*distance/dissimilarity matrix*) data structures. In Chapter 3, we will empirically show that our measure and corrsponding algorithm can produce competitor identification results that closely mirror the results of a more laborious marketing survey process.

Chapter 3

APPLICATIONS OF UNIQUENESS-MOTIVATED PROBABILISTIC SIMILARITY MEASURE

3.1 Application in Local Competitor Identification

In this section, we validate the results derived from our uniqueness-motivated probabilistic similarity measure using the context of local restaurant competition in the Main Street area of Newark, Delaware (i.e. the college town associated with the University of Delaware in the United States). Downtown Newark is a vibrant dining destination with over 50 restaurants and the Main Street area was awarded the 2011 Great American Main Street Award [51]. Given the area's proximity to the University of Delaware campus, typical customers for these establishments are students from the University of Delaware. Hence, we compare the results of our algorithm using data scraped from Yelp and Google with a more traditional elicitation-based approach, namely a consumer survey that serves as a benchmark for comparison. Data and accompanying code can be found online at `http://bit.ly/uniqAlg`.

The data feed for our clustering algorithm contains both discrete and continuous features (see Appendix Table A.2). The discrete features come from social tags, i.e. Yelp tags, for restaurants in downtown Newark. The continuous features consist of the average transaction amount in the restaurants, the customer rating of the restaurants listed on Yelp and a distance matrix that captures pair-wise physical distance between two restaurants. Based on discussions with a merchant bank interested in our uniqueness-based algorithm and who can easily automate the collection of transaction data, we include transaction amount as a feature in our dataset. However, due to privacy concerns, the transaction amounts were not gathered through automated means, rather they were gathered through survey using a convenience sample, which

is a typical practice in relevant studies [52]. The distance matrix was generated using Google Maps Distance Matrix API [53]. The intuition of using these data is simple: two restaurants are more likely to be perceived as direct competitors if they share similar social tag(s), social ratings, are located close to each other, and charge similar prices.

In terms of implementation, Table 3.1 summaries data inputs, calculation methods, and outputs of both the SBAC algorithm and our proposed extensions.

Table 3.1: Summary of Algorithm's Inputs and Outputs

<table>
<tr><th rowspan="2">Input Features</th><th rowspan="2">Data Source</th><th colspan="2">Algorithm</th><th rowspan="2">Output</th></tr>
<tr><th>SBAC</th><th>Our proposal</th></tr>
<tr><td>Transaction amount</td><td>Collected from survey</td><td colspan="2" rowspan="2">Numeric feature similarity 2.3.2</td><td rowspan="4">Similarity matrix & Dendrogram</td></tr>
<tr><td>Yelp rating</td><td>Crawled online</td></tr>
<tr><td>Yelp tags</td><td>Crawled online</td><td>Independent categorical feature similarity 2.3.1</td><td>Bag-of-tags similarity 2.3.3.1</td></tr>
<tr><td>Distance matrix</td><td>Addresses crawled online, distance calculated by Google Map API</td><td>NA</td><td>Distance matrix similarity 2.3.3.2</td></tr>
</table>

3.1.1 Empirical Test for Algorithm Validation

As the idea of competitive identification is a subjective concept, validation of the algorithm becomes a challenge. Common practices use the concepts of customer conception [10] or managerial congruence [19]. We conducted an aided awareness consumer survey [54, 55, 56] with over 200 participants to serve as a benchmark for validating the results of our automated algorithm. From the survey results, we construct a substitution matrix which represents the degree of substitutability among a set of restaurants. We investigate how algorithms using uniqueness, i.e. the orignal SBAC algorithm and our proposed similarity measure extensions, can recover the market structure derived via the aided awareness consumer surveys.

As additional validation between our algorithm results and consumer perceptions, we performed a similarity-based perceptual mapping survey, also known as multidimensional perceptual scaling [57]. While this method directly collects opinions on pairwise similarities, it is very laborious to do this type of survey because the number of questions needed increases quadratically with number of restaurants. Therefore, we randomly selected a subset of the studied restaurants and created the questions from that restaurant subset. Mantel's test will be adopted for direct comparison on the algorithm's similarity matrix and the matrix obtained by survey.

3.1.2 Data Collection

We selected the 25 most popular and reviewed restaurants in the Main Street area in Newark, Delaware, to serve as *seed restaurants*; restaurants for which we will assess their competitiveness against all others. The seeds were chosen based on the number of Yelp reviews and their popularity in a completed pilot study of 19 participants.

The basic assumption typical of studies that collect brand substitution data is that if an item in a submarket becomes unavailable, then consumers are more likely to switch to items within the submarket rather than to items outside of the submarket [58]. Therefore, to assess competitiveness and obtain substitution patterns among the restaurants, we asked survey participants questions assessing the ability of competitors to the seed restaurant in satisfying their needs by asking: "Assuming your intention was to go to Restaurant A at Address A, but you find out it is closed. What is the likelihood that you would visit each of these other restaurants?". Respondents used a scale from 1 to 7 (extremely unlikely to extremely likely) to compare the seed restaurant with up to eight randomly picked restaurants previously visited by the participant.

All survey participants were University of Delaware students who signed up for the survey voluntarily and came in-person to fill the questionnaire at a designated time on a lab computer. To ensure the validity of responses, the first section provided the full list of 50 restaurants and asked the survey respondent to indicate if he/she had

visited each of the restaurants. All questions in later sections are generated from the list of seed restaurants that the respondent confirmed that he/she had been to before. Intra-survey validity checks resulted in 35 out of 234 respondents failing at least one validity check and their responses were eliminated.

To ascertain survey participant's perception of competition, we collected perceptions of similarities among 10 designated restaurants by asking for judgments about all possible pairs. "On a scale from 1 to 7 (mostly disagree to mostly agree), how much do you agree with this statement: Restaurant A and Restaurant B compete for customers." Note that this required 45 assessments from each participant in the survey; this similarity-based perceptual mapping question was added only during the second round of surveys.

The survey, consisting of either the first type or both types of questions, was completed by 234 students from University of Delaware [1]. After eliminating the 35 responses that failed at least one of the included validity questions, we have 199 responses. Demographics of the respondents are shown in Appendix Table A.1. More than 65% of the respondents visit restaurants in Newark more than once per week, which makes them a representative source of customers for the sampled restaurants.

3.1.3 Results

3.1.3.1 Correlation Tests

For the aided awareness questions asked about the 25 *seed restaurants*, we collected willingness to switch scores on a Likert scale and then averaged the Likert scores for each combination of seed restaurant and potential competitor. Hence, for each seed restaurant, survey responses are summarized in a vector of length 49 as each seed restaurant is compared with up to 49 other restaurants in the Main Street area.

[1] We ran two rounds of surveys in total. First round was run in September 2016 and collected 126 valid responses. The second round was run in February 2017 and collected 73 valid responses.

Survey responses were compared to corresponding similarity scores as calculated using the original SBAC and this thesis's proposed method, respectively. Using Pearson's correlation coefficient, we find statistically significant results suggesting that both the SBAC algorithm and this thesis's extensions provide results that correlate with survey responses (see Table 3.2).

Table 3.2: Pearson's Correlation Comparison

	Df	P-value	Coefficient
SBAC & Survey	1220	<2.2e-16	0.193
The proposed method & Survey	1220	<2.2e-16	0.412

Afterwards, we compare the magnitude of the correlation coefficients between algorithms and the survey data. As shown in Table 3.3, this thesis's proposed method provides superior results. In fact, in 24 out of the 25 effect size comparisons (i.e. one comparison for each seed restaurant), the correlation coefficient for this thesis's proposed method is higher than the correlation coefficient for the SBAC algorithm. Using a two-tailed binomial test, (p-value = 1.55e-06), we find supportive evidence that our proposed extensions can yield better correlated results with consumer opinions.

To ascertain the source of our algorithm's performance advantage, we investigate whether the bag-of-tags feature and/or the distance matrix lead(s) to the superior performance of this thesis's proposed method. To observe changes of correlation coefficients with the introduction of each feature type, we shut off the proposed related feature structure one at a time. The results are shown in Table 3.3. In 22 out of 25 pairwise comparisons (binomial test p-value = 0.00016), the incorporation of geographic distance matrix feature yields higher correlation coefficients than the original SBAC algorithm does. In 24 out of 25 pairwise comparisons (binomial test p-value = 1.55e-06), the accommodation of *bag-of-tags* feature yields higher correlation coefficients than the original SBAC algorithm does as compared with the survey ratings (see Table 3.3). Hence, both the accommodation of geographic distance matrix and

Table 3.3: Pearson's Correlation Coefficient Comparison

Seed Restaurant	Correlation Coefficients				Df
	SBAC & Survey Rating	Generalized with Distance Matrix & Survey Rating	Generalized with Tag Relatedness & Survey Rating	This thesis's algorithm & Survey Rating	
All Seed Restaurants	0.193***	0.351***	0.392***	0.412***	1207
Ali Baba	0.014	0.299*	0.305*	0.315*	47
Bamboo House	0.144	0.174	0.383**	0.391**	46
Buffalo Wild Wings	0.104	0.506***	0.370**	0.433**	47
Caffe Gelato	0.161	0.503***	0.637***	0.633***	47
California Tortilla	0.432**	0.440**	0.439**	0.466***	46
Catherine Rooney's Irish Pub	-0.092	0.539***	0.476***	0.521***	47
Chick-fil-A	0.149	0.301*	0.420**	0.422**	46
Chipotle Mexican Grill	0.205	0.356*	0.390**	0.405**	47
Cosi	0.114	0.289*	0.410**	0.400**	46
Deer Park Tavern	0.288*	0.496***	0.436**	0.505***	47
El Diablo Burritos	0.201	0.440**	0.481***	0.501***	47
Grain Craft Bar + Kitchen	0.309*	0.307*	0.400**	0.391**	47
Grotto Pizza	0.319*	0.563***	0.508***	0.562***	46
Home Grown Cafe	0.147	0.543***	0.565***	0.579***	46
Iron Hill Brewery & Restaurant	0.252	0.330*	0.432**	0.427**	46
Jimmy John's	0.387**	0.439**	0.465***	0.493***	46
Klondike Kate's	0.326*	0.508***	0.564***	0.568***	47
McDonald's	0.427**	0.472***	0.474***	0.518***	46
Newark Deli & Bagels	0.172	0.519***	0.485***	0.526***	47
Panera Bread	0.295*	0.519***	0.493***	0.525***	47
Ramen Kumamoto	0.046	0.222	0.252	0.260	45
Subway	0.295*	0.270	0.173	0.220	45
Taverna	0.030	0.394**	0.496***	0.482***	47
The Red Bowl	0.288*	0.366**	0.420**	0.438**	46
Vita Nova	0.227	0.216	0.232	0.263	45

Note: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$

the accommodation of *bag-of-tags* both contribute to this task with their respective advantages likely to be context-specific.

3.1.3.2 Visualization of Willingness to Switch

In addition to statistical tests, we confirm our proposed method's ability to mimic consumer assessments of competitor structure through visualization of the results. The purpose of this visualization is to examine the distribution of rating responses in addition to comparing the average ratings as done in previous section.

In Figure 3.1, the vertical axis is sorted according to output of this thesis's

algorithm. Restaurants ranked by the uniqueness-driven clustering algorithm are presented in ascending order of the computed similarity with the most similar competitor restaurants lying at the bottom of the axis and least similar restaurants at the top. Due to the limitation of space, we have only included competitor restaurants receiving top 20% of responses for the listed seed restaurant. We have 50 restaurants in the sample, therefore, for each seed restaurant, there is roughly 10 restaurants included in its visualization of willingness to switch. Competitor restaurants with less than 10 survey responses are eliminated. The horizontal axis represents the percentage of Likert scores receiving a rating from 1 to 7, and these percentages are shown as horizontally stacked bars using color to denote the given Likert scores; darker red represents more survey responses indicating close competitors. Given this visualization structure, if the algorithm is performing well, then we expect to observe a pattern of darker shades to the lower-right of the plot and lighter shades to the upper-left. This pattern is clear in the example of Figure 3.1.

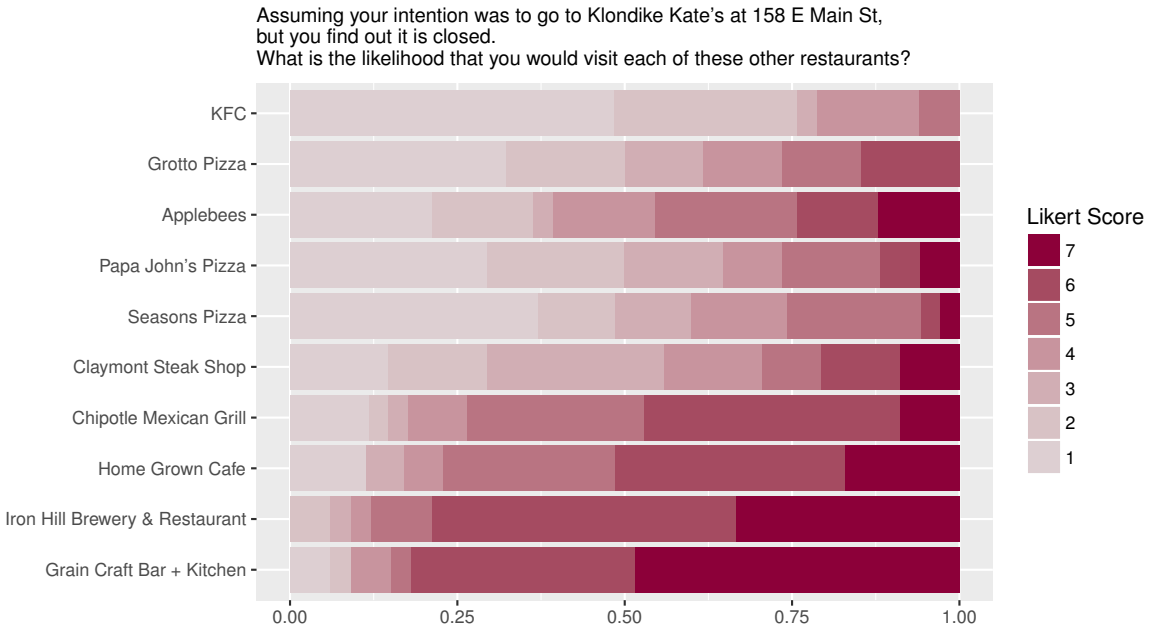


Figure 3.1: Survey Response Composition Ranked by Ascending Similarity

In addition to the one seed restaurant shown in Figure 3.1, visualization of the results for all other seed restaurants are enclosed in the Appendix (see Figure A.1).

The color pattern of darker red in the lower right is clear throughout the graphs. There are a few exceptions to this pattern worthy of note. Specifically, Panera bread is often underestimated as a competitor. Its unique tags of both "soup" and "salads" combined with a popularity among survey respondents that is not seen in the underlying automated dataset lead to this underestimation. Looking at some other restaurants, like "Grain Craft Bar + Kitchen" and "El Diablo Burritos", we suspect baseline popularity plays a large role in where people will choose to go no matter what *seed restaurant* they are given as a competitor.

3.1.3.3 Mantel Test on Perceptual Map and Similarity Matrix

By asking the similarity-based perceptual mapping questions, we collect perceptions of the similarities among ten designated restaurants. We took the average of the responses from 1 to 7 scale points, and formed a matrix of similarity perceptions for comparison to automated methods. Similarity matrices are converted to dissimilarity matrices by simply taking the additive inverse ($S = 1 - D$). The diagonal entries of all these matrices are assigned as NA values since it is not necessary to measure the similarity between any restaurant with itself.

Mantel's test is a statistical regression test that takes in distance or dissimilarity matrices which summarize pairwise similarities [59]. The test has been widely applied in population genetics, ecology, anthropology and biology [60]. It is one of several permutational tests for associations between distance matrices [61]. As a formal hypothesis test, we leverage the Mantel's test ability to compare an observed similarity matrix with another similarity matrix derived from a conceptual or numerical model.

We implement Mantel's test (see Table 3.4) through R "vegan" package [62]. P-values are determined by 999 permutations. The Mantel statistic r is the Pearson's product-moment correlation coefficient, falling in the range $[-1, 1]$, closer to 1 indicates a strong positive correlation, but the magnitude of correlation is often comparatively small even when proved statistical significant [63]. The Mantel statistic r shows that both the SBAC algorithm and this thesis's algorithm produces positively correlated

results compared with the similarity-based perceptual mapping matrix. And our algorithm produces a higher correlation with perceptual mapping result than the SBAC algorithm does. Thus, automated methods can produce correlated results to a well established marketing method such as perceptual mapping.

Table 3.4: Mantel Test Results

	Mantel Statistic r
SBAC & perceptual mapping	0.462***
This thesis's algorithm & perceptual mapping	0.515***

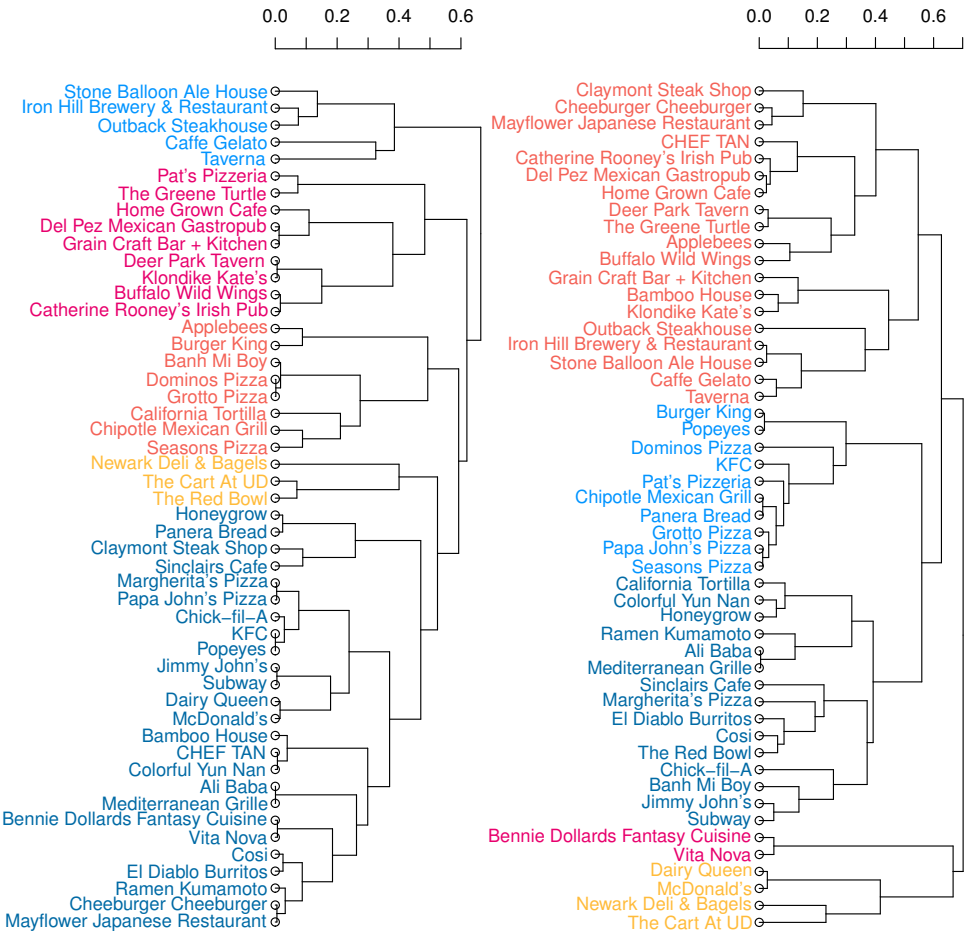
Note: *** $p < 0.001$

3.1.4 Discussion of Results

Based on the scrape-able data feed powering the dissimilarity calculations (see Appendix Table A.2) and the calculated distance matrix of pair-wise physical distances between any two restaurants, we use this thesis's algorithm to measure dissimilarity and subsequently derive the dendrogram in Figure 3.2(b) using the average linkage method. For comparison, the original SBAC algorithm can also be used to generate a dendrogram as presented in Figure 3.2(a). Since the original SBAC formulation does not accommodate the bag-of-tags feature, the algorithm parses the vector of tags into different categorical features, e.g. *Yelp tag1*, and assumes independence between the parsed tag features. The distance matrix is not included as input for the original SBAC.

In discerning differences between Figures 3.2(a) and 3.2(b), we can highlight examples where both distance and tag relatedness play roles in creating the more logically clustered groups of Figure 3.2(b). For example, look at the position of Papa John's pizza in each dendrogram. The original SBAC model places its closest competitor as another pizza place, namely Margherita's pizza. In reality, however, Papa John's is across the street from Seasons Pizza, and hence they are much more likely to be close competitors. The dendrogram shown in Figure 3.2(b) captures this. Additionally, restaurants Applebee's and Burger King are clustered together as close competitors

by the orignal SBAC algorithm; they both share "burgers" as their second Yelp tag. However, when looking at the Yelp tags through the bag-of-tags feature of this thesis's algorithm, Burger King is more appropriately placed with fellow fast-food chain Popeye's while the "American (Traditional)" tag is used to discern a close association of Applebee's to Buffalo Wild Wings. This is a much more appropriate grouping.



(a) Original SBAC (b) This Thesis's Algorithm

Figure 3.2: Dendrograms

3.2 Application in Traveler Ancillary Purchase Prediction

3.2.1 Data Collection

The second application originates from a business project. Data is granted by an airline solution technology company, who shows a great interest in our proposed uniqueness-motivated similarity measure and its application in airline revenue management.

The data consists of 21,418 anonymous airline passenger records and their ancillary purchases. It is a typical supervised learning scenario with some static information collected from airline passengers and a binary label. The static information collected from airline travelers include indicators of booking such as group booking, round trip, and infants on board, cabin, dominant departure and destination, trip start day of week, trip duration days, advance purchase days, group size etc. And the binary label indicates if the passenger purchase any ancillary service other than booking the air ticket, such as checked bag, ticket change fee, upgrade, reserved seat assignment, airport early sale, booking fee or miscellaneous charge.

3.2.2 Results

Some airline travelers own unique features that may trigger certain ancillary purchase, for example, parents traveling with infants tend to purchase reserved seats to have extra room on the flight. By acknowledging the value of feature uniqueness, we've combined our uniqueness-motivated similarity measure with the K-Nearest Neighbor method. By computing the uniqueness-motivated similarity score between every pair of traveler records in the data, we implement the K-Nearest Neighbor based on an implicit assumption that similar airline travelers tend to have similar ancillary purchases.

The K-NN method naturally goes hand in hand with similarity measure and thus is appropriate to be combined with our similarity proposal. However, it is difficult to select the optimum value of k due to the variation of probability of error with the number of nearest neighbors k. Therefore, we implemented the distance-weighted

k-nearest-neighbor proposed by Dudani [64], which is believed to yield smaller probabilities of error.

We compared the results yielded by our proposed similarity measure together with K-NN with other popular machine learning methods as benchmark. The area under the ROC (receiver operating characteristics) curve (AUC) performance is a reliable and popular measure to evaluate predictive ability of learning algorithms [65] and we used AUC for evaluation. The results are reported in Table 3.5. Our uniqueness-motivated similarity measure combined with k-Nearest Neighbors algorithm is superior compared with other benchmark classifiers such as K-mean clustering, decision tree, support vector machine, etc.

Table 3.5: AUC Performance Comparison

Classifier	AUC
Uniqueness-motivated Similarity Measure + KNN	0.702
K-means Clustering (numeric features only)	0.582
Decision Tree	0.514
Logistic Regression	0.675
Naive Bayes	0.674
SVM (Linear kernel)	0.516
SVM (RBF kernel)	0.614

We also visualize the performance of our proposed uniqueness-motivated similarity measure with K-NN as in Figure 3.3, with horizontal axis indicating the predicted probability of ancillary services purchase while the vertical axis of the upper plot is the counts of actual purchasers versus non-purchasers and the vertical axis of the lower plot is the percentile of actual purchasers versus non-purchasers. It shows a clear pattern that as the predicted purchase probability goes up, the actual percentile of ancillary services purchase goes up.

The ancillary data provides very detailed information of the ancillary service purchased. We have categorized 32 different kinds of ancillary services into seven major categories for further prediction purpose (see categorization criteria in appendix B.1).

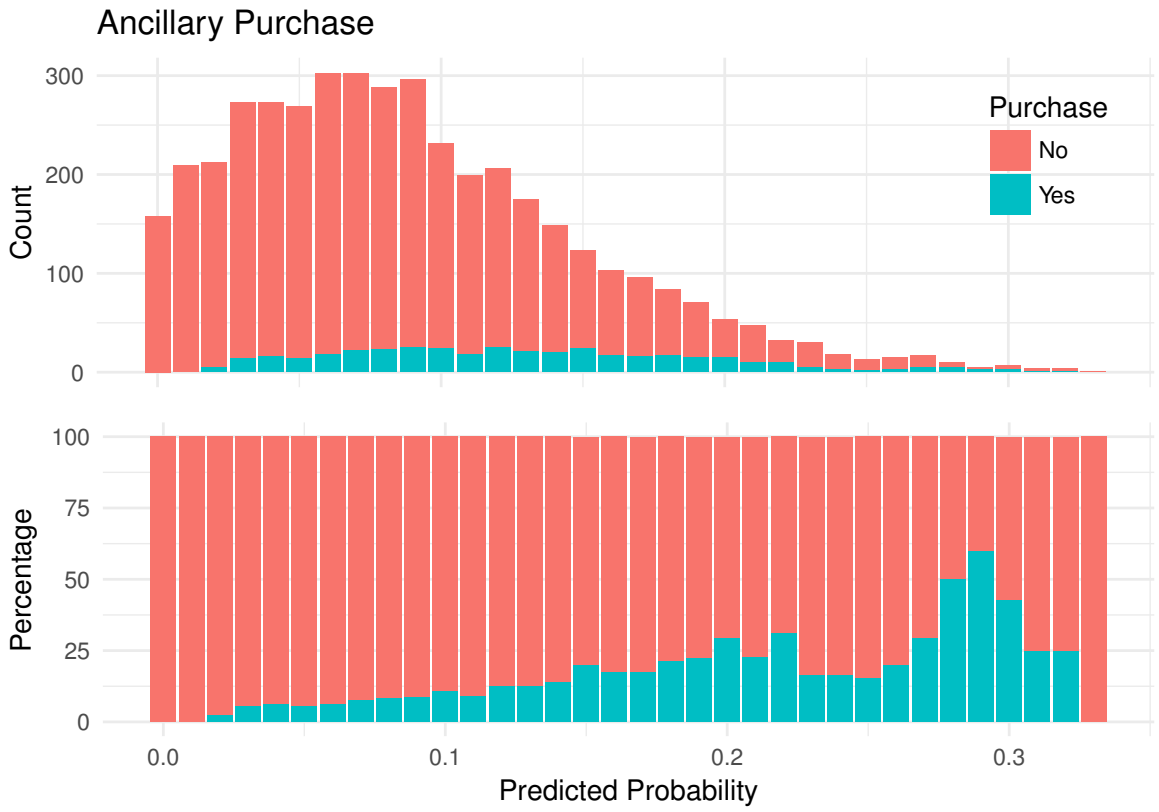


Figure 3.3: K-Nearest Neighbor using Uniqueness-motivated Similarity Performance of Traveler Ancillary Purchase Probability Prediction

The visualization of all seven ancillary services categories are presented in the Appendix B. All of the ancillary services subcategories supports that our method is performing well in predicting the probability of purchase, as the predicted probability goes up, the actual purchase percentile goes up.

This applications shows the predictive power of the uniqueness-motivated similarity measure in airline ancillary purchase prediction. Practically this method can be applied to airline revenue management, e.g. by better predicting the probability of ancillary purchases, travel agencies may target market bundle services to passengers with higher purchase probabilities to increase the total revenue.

3.3 Summary

In this chapter, we have shown two applications with our proposed uniqueness-motivated similarity. In the unsupervised learning task, i.e. to identify competitors, we compare the algorithm learning results with traditional marketing solicitation methods for validation. In the supervised learning task, i.e. to predict airline ancillary purchase, we use AUC performance to measure the prediction accuracy. In both applications, our proposed uniqueness-motivaed similarity has demonstrated its good predictive capability and potential for practical use. The automation of the competitor identification process may save market researches time/money and accelrate the rate at which these studies can be conducted. And accurate prediction of airline ancillary purchases may bring opportunities to earn more total revenue for airline companies and/or travel agencies.

Chapter 4

GPVIZ: AN R PACKAGE FOR VISUALIZING DISTRIBUTION OF GAUSSIAN PROCESS PREDICTIONS' SLOPES

4.1 Introduction

The Bayesian method is well-known for its capability of discerning probabilities on events measuring uncertainty. In this chapter, we have leveraged a Bayesian approach to conduct a predictive analysis via the Gaussian process. And we also derived the distribution of the Gaussian process predictions' first derivative for business insight extraction.

The Gaussian process is a generalization of the Gaussian probability distribution, where a stochastic process governs the properties of functions. One can view a function as a continuous but discrete mapping at each data point. When exploring the properties of the function given a finite number of points, the Gaussian process gives you inferences that closely fit with the observations [66].

In the realm of supervised learning, we typically care about not only the prediction for the new unseen data, but we are also curious about the level of confidence we have in our prediction. When exploring a bivariate nonlinear relationship between an independent variable and a dependent variable, it is of interest in many applications to visualize the nonlinear relationship rather than describing it with a few parameters. In fact, many applications tend to discover certain regions where independent variables are related to the dependent variable in a particular fashion. For example, pharmaceutical scientists studying dose-response curves want to know the range of dosages that guarantees a cure of disease and avoids the side-effects brought by over-dosage; economists studying interest rates want to find the range of interest rates that boost economic growth while controlling the inflation rate; banking regulators concerned about risk

control are interested in finding a safe region of commercial banks' debt-to-asset ratio such that they would be able to recover most of their assets when they fail.

In order to describe the change rates between an independent variable and a dependent variable, we can use the slope of their relationship curve. In order to determine our confidence in such relevancy, we need a distribution of slopes instead of simple point estimations. As the Gaussian process is a very useful tool to help us extract nonlinear relationships, we can use the Gaussian process for predictive analyses and take the first derivative of each Gaussian process realization to collect a distribution of slopes.

In order to obtain the distribution of the Gaussian process predictions' slopes, we developed the *gpviz* package. It visualizes the Gaussian process predictions and describes the distribution of the Gaussian process predictions' slopes as the dependent variable changes along with the independent variable.

The *gpviz* package models the Gaussian process in R using the probabilistic programming language Stan. As compared with frequentist approaches, the alternative Bayesian methods have several advantages such as explicitly incorporating prior knowledge about parameters into the model, but this practice was limited for a long time because the posterior distributions of complex models are not available analytically. Markov Chain Monte Carlo (MCMC) sampling could be one way to describe the posterior distributions. Over the past decade, both sampling algorithms and computing power have improved rapidly. Thus sampling algorithms are nowadays more efficient. Several softwares implement sampling techniques: Metropolis-Hastings Algorithm [67], Gibbs Sampling [68], Slice Sampling [69] (JAGS [70, 71], BUGS [72], WinBUGS [73, 74], OpenBUGS [75, 76]); Reversible-jump [77, 78]; Sequential Monte Carlo [79, 80]; Hamiltonian Monte Carlo (Stan); some Phython packages such as the MCMC Hammer (emcee), Bayesian Statistical Modeling in Python (pymc), the Python Interface to Stan (pystan); some R packages such as adaptMCMC, atmcmc, BRugs, mcmc, MCMCpack, ramcmc, rjags, etc.

In contrast to the Metropolis-Hastings algorithm and the Gibbs sampler, Stan

implements Hamiltonian Monte Carlo and its extension No-U-Turn Sampler (NUTS). These algorithms converge more quickly than the aforementioned methods, especially for high-dimensional models. And Stan does not require conjugate priors as opposed to the Gibbs sampler [81]. Users specify log density functions in Stans probabilistic programming language and get: 1) a full Bayesian statistical inference with Markov chain Monte Carlo (MCMC) sampling; 2) an approximate Bayesian inference with variational inferences; 3) a penalized maximum likelihood estimation with optimization [82]. Stan has gained thousands of users, from both industry and academia. They rely on Stan for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business [83]. Therefore, we choose to model the Gaussian process and the distribution of its predictions' slopes using Stan.

4.1.1 Gaussian Process

A Gaussian process ($\mathcal{GP}$) is a stochastic process, i.e. a collection of random variables, of which any finite collection has a multivariate normal distribution.

The Gaussian process involved in machine learning is a lazy learning method. It leverages kernel functions to measure similarities between points for predictive analysis. The Gaussian process predictions are not single point estimations, but a distribution of predictions over the unseen data point.

The advantages of the Gaussian process are: its flexibility, its nonlinearity, and it is non-parametric (does not assume a particular function form between the independent and dependent variables). The Gaussian process can conveniently be used to specify very flexible nonlinear relationships. Any positive definite function can be used as a covariance function, which gives $\mathcal{GP}$ much flexibility to fit the data [84]. The prediction interpolates the observations (at least for regular kernels). The prediction is probabilistic (Gaussian) so that one can compute empirical confidence intervals and decide based on those if one should refit (online fitting, adaptive fitting) the prediction in some region of interest. The generality of the supported functions makes Gaussian

priors popular choices for priors in general multivariate nonlinear regression problems [85, 86, 87].

The disadvantages of the Gaussian process include: 1) it is not sparse, i.e., it uses the whole sample/features information to perform the prediction; 2) it loses efficiency in high dimensional spaces, namely when the number of features exceeds a few dozen.

There are other alternative models to interpolate data and fit nonlinear relationships, such as splines. Kimeldorf and Wahba [88] have proven that splines are a special case of Gaussian process regression. Once we take a certain kernel type in the Gaussian process regression, we can obtain exactly the same result as spline fitting.

4.1.2 Region of Interest

In a nonlinear relationship, the dependent variable changes at different rates from the independent variable. As many applications may be interested, we would like to explore the region where the bivariate relationship intensity is of interest to decision makers. The informativeness of relationship intensity should be relevant to business needs, problem context based, and needs to be defined by the user/application.

As the Gaussian process gets useful in describing nonlinear relationships, we can model the Gaussian process via MCMC algorithms for regression tasks. However, as for many machine learning algorithms, outputting merely predictions makes the algorithm a black box. Even though Gaussian process regression may output a distribution of predictions of unseen data rather than point estimations, it still does not help very much with the interpretability of the model. Interpretability is a highly valuable element in modeling, especially in industrial applications. For example, a commercial bank would prefer using a generalized linear regression over a sophisticated machine learning algorithm for the sake of interpretability, especially when facing customers and regulators.

The way we extract the region of interest relies on the Gaussian process regression realizations from Stan. By deriving the distribution of the Gaussian process

predictions' slopes, we can visualize the nonlinear relationship between the independent variable and the dependent variable. Theses slopes can be very useful in telling people about the region where the bivariate relationship is intense versus where the bivariate relationship is rather uncertain. This region of interest correlates to the actual demand from the user who wants to interpret the relationship, the user gets to define informativeness. By exploring the distribution of the Gaussian process predictions' slopes, we can have a vivid understanding of how certain or uncertain the predictions are along the change of the independent variable. Thus a user-defined region of interest could be extracted based on the distribution of slopes.

With the Gaussian process's probabilistic predictions or empirical realizations, one can extract a region of interest based on the context of the application. It could possibly be a region where the dependent variable changes quickly as the independent variable changes, such as the range at which the net-loss of failed banks soars up against the deposit-asset ratio. When combined with domain knowledge, this might be a financial application in discovering a safe zone of debt-to-asset ratio for commercial banks to control their risk.

4.1.3 Comparison with Relevant R Packages

There are many other R packages related to Bayesian plots: *bayesplot* provides plots for Bayesian model diagnostics, particularly for models interfacing with Stan [89, 90]; *rstanarm* models applied Bayesian regression, emulates other R model-fitting functions but uses Stan (via the rstan package) for the backend estimation [91]; *tidybayes* combines Bayesian analysis with tidy data and ggplot workflow. The package facilitates composing data for, and extracting samples from, Bayesian models in a tidy data format, and provides visualization tools for analysing tidy samples from Bayesian models, including estimates and intervals, eye plots (intervals with densities), and fits curves with uncertainty bands [92]. Our workflow is very similar to the *tidybayes* package but we provide visualization for the distribution of prediction's first derivatives.

Compared with theses general packages for Bayesian modeling and visualization, our package *gpviz* is more focused on visualizing the Gaussian process prediction's first derivatives to describe the intensity of nonlinear bivariate relationships.

In terms of the Gaussian process modeling, most R packages focus on covariance kernel function or computation speed: *brms* uses Stan to conduct Bayesian regression modeling and provides a function for the Gaussian process modeling [81]; *GPfit* fits a Gaussian process model to continuous input variables and simulator outputs obtained from a scalar valued deterministic computer simulator [93]; *FastGP* includes functionality for Toeplitz matrices whose inverse needs $\mathcal{O}(n^2)$ time [94]; *mlegp* finds maximum likelihood estimates of the Gaussian process for univariate and multidimensional responses [95]; *kergp* provides Gaussian process interpolation, regression and simulation, emphasizing user-defined covariance kernels [96]. Many of these packages also provide plotting functions for diagnosis. To our best knowledge, we are the first to focus on the Gaussian process prediction's first derivatives, and the first to visualize the distribution of the slopes to extract regions of interest. But we also need to admit that our package cannot yet offer a variety of kernel functions for users to choose from. It is a possible future expansion of our package.

4.2 Derivation of Gaussian Process First Derivative's Distribution

4.2.1 Gaussian Process Regression

Ebden [97] made a very good introduction to Gaussian process for regression. For the readability of this chaper, we also reiterate the mathematical formulation of the problem and set up the foundation for our later derivation in Section 4.2.3. We summarize the Gaussian process regression model in Figure 4.1.

A Gaussian process can be viewed as a generalization of Gaussian distribution. Contrary to having a mean vector and a covariance matrix, a Gaussian process is specified by its mean function $m(x)$ and covariance matrix K consisting of kernel

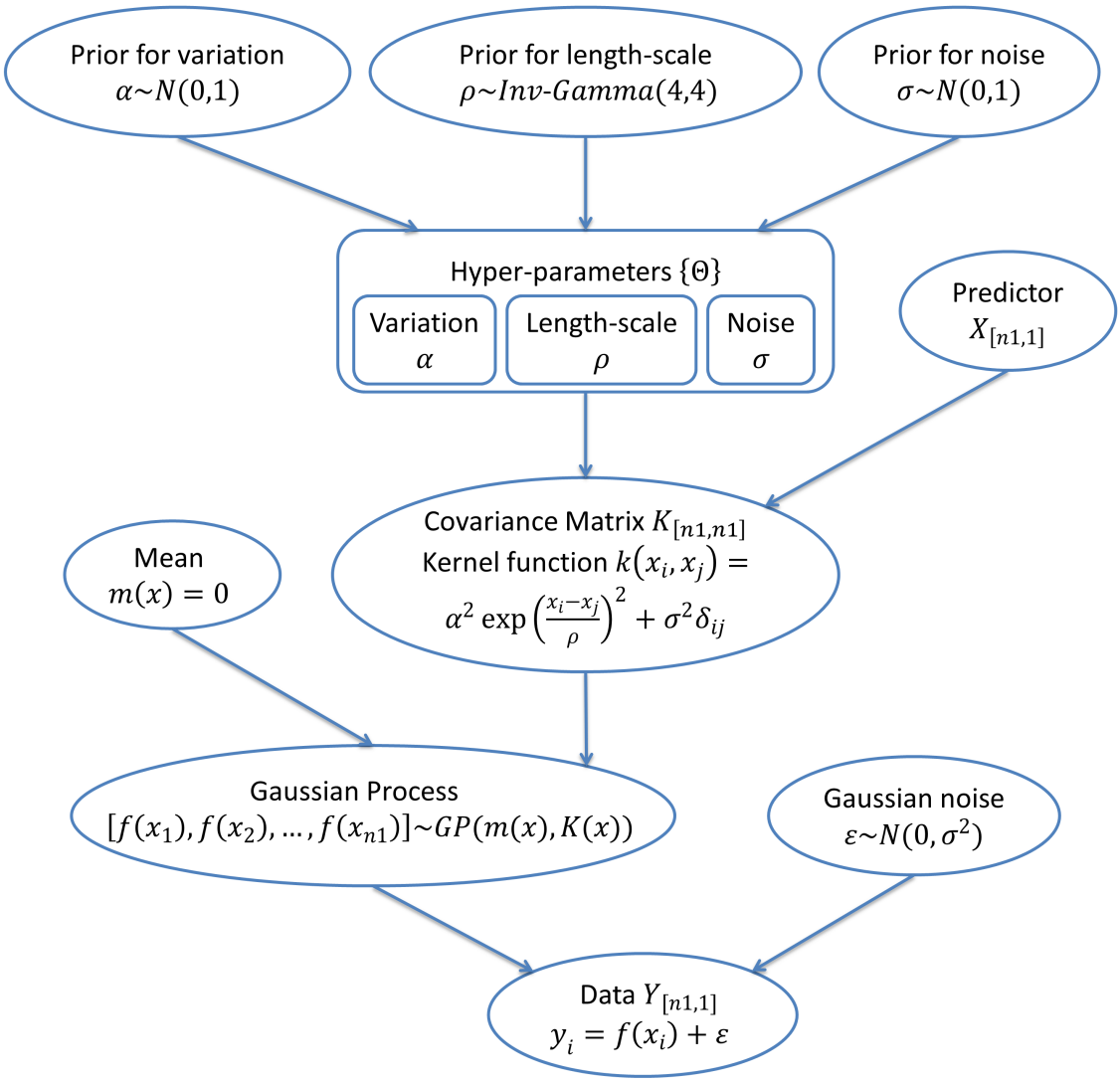


Figure 4.1: Gaussian process modelling workflow

functions $k(x_i, x_j)$.

$$f(x) \sim \mathcal{GP}(m(x), K) \tag{4.1}$$

To address noise in the training outputs, the most common assumption is that the Gaussian noises in the outputs are i.i.d. [84]. With that assumption, every $f(x)$ has an extra covariance with itself and only itself since the noises are assumed to be uncorrelated:

$$\begin{aligned} y(x) &= f(x) + \epsilon,\ \epsilon \sim \mathcal{N}(0, \sigma^2) \\ y(x) &\sim \mathcal{GP}(m(x), K + \sigma^2 I) \end{aligned} \tag{4.2}$$

In many applications, the mean function of the Gaussian process is assumed to be zero, $m(x) = 0$ [98]. This is a reasonable assumption when the data is centralized, or when we don't have much prior information about where the median distribution lies.

The key of Gaussian process learning is to choose a proper covariance function, also known as kernel function. The intuition of the kernel function is that if x_i and x_j are similar, then we expect the outputs of these two points to be similar as well. The parameters used in the kernel function are referred to as hyperparameters, denoted as Θ.

In this R package, we use the most popular kernel function: the exponential quadratic function [99]. Many alternative kernel functions are available for pattern discovery and extrapolation [100], some of which we plan to incorporate into our package. The exponential quadratic kernel function is:

$$k(x_i, x_j) = \alpha^2 exp\Big[-\frac{(x_i - x_j)^2}{2\rho^2} \Big] + \sigma^2 \delta_{i,j} \tag{4.3}$$

where $\delta_{i,j}$ is the Kronecker delta function:

$$\delta_{i,j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{4.4}$$

Here, our hyperparameters $\Theta = \{\alpha, \rho, \sigma\}$. Notice here the i and j are indices of input. We may have identical inputs, but their noises are still assumed to be independent. Besides taking care of noise, the addition of σ^2 ensures the positive definiteness of K.

The only term in the exponential quadratic kernel function involving inputs x_i and x_j is their difference $x_i - x_j$, which produces a stationary process since the covariance matrix remains unchanged at any pair of outputs $K(X|\Theta) = K(X + \epsilon\ |\Theta)$.

We assign prior distributions to the hyperparameters and then use the Stan sampler to update the priors while observing the training examples by maximizing the log marginal likelihood $log(p(y|X, \Theta))$. The choices for prior hyperparameter distributions are explained and justified in Section 4.3.3.

4.2.2 Gaussian Process Regression Prediction

Given training data points $X = [x_1, x_2, ..., x_{n1}]^T, Y = [y_1, y_2, ..., y_{n1}]^T$ and test data inputs $X^* = [x_1^*, x_2^*, ..., x_{n2}^*]^T$, we would like to estimate the test data outputs $Y^* = [y_1^*, y_2^*, ..., y_{n2}^*]^T$. The Gaussian process assumes that the function values $f(x)$ and $f(x^*)$ to be random variables jointly following a multivariate Gaussian distribution:

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f^*} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K} & \boldsymbol{K_*}^T \\ \boldsymbol{K_*} & \boldsymbol{K_{**}} \end{bmatrix}\right) \tag{4.5}$$

$$\boldsymbol{f} = [f(x_1), f(x_2), ..., f(x_{n1})]^T$$

$$\boldsymbol{f^*} = [f(x_1^*), f(x_2^*), ..., f(x_{n1}^*)]^T$$

$\boldsymbol{K}$ is the covariance matrix:

$$\boldsymbol{K} = \begin{bmatrix} k(x_1, x_1), & k(x_1, x_2), \dots & k(x_1, x_{n1}) \\ k(x_2, x_1), & k(x_2, x_2), \dots & k(x_2, x_{n1}) \\ \vdots & & \\ k(x_n, x_1), & k(x_n, x_2), \dots & k(x_{n1}, x_{n1}) \end{bmatrix}$$

$$\boldsymbol{K_*} = \begin{bmatrix} k(x_1, x_1^*), & k(x_1, x_2^*), \dots & k(x_1, x_{n2}^*) \\ k(x_2, x_1^*), & k(x_2, x_2^*), \dots & k(x_2, x_{n2}^*) \\ \vdots & & \\ k(x_{n1}, x_1^*), & k(x_{n1}, x_2^*), \dots & k(x_{n1}, x_{n2}^*) \end{bmatrix}$$

$$\boldsymbol{K_{**}} = \begin{bmatrix} k(x_1^*, x_1^*), & k(x_1^*, x_2^*), \dots & k(x_1^*, x_{n2}^*) \\ k(x_2^*, x_1^*), & k(x_2^*, x_2^*), \dots & k(x_2^*, x_{n2}^*) \\ \vdots & & \\ k(x_{n2}^*, x_1^*), & k(x_{n2}^*, x_2^*), \dots & k(x_{n2}^*, x_{n2}^*) \end{bmatrix}$$

It is a well-known fact that the Gaussian process with its hyperparameter priors has a very nice posterior distribution [84]:

$$\boldsymbol{f^*}|\boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{K_*}^T \boldsymbol{K}^{-1} Y,\ \boldsymbol{K_{**}} - \boldsymbol{K_*}^T \boldsymbol{K}^{-1} \boldsymbol{K_*}) \tag{4.6}$$

$$y_i^* \sim \mathcal{N}(f_i^*, \sigma) \quad \forall\ i \in \{1, 2, ..., n2\} \tag{4.7}$$

We would like to use $\boldsymbol{f^*}$ to describe the posterior distribution of Y^* for three reasons: 1) the actual realizations of y^* vary due to the noise term σ; 2) for each post warm-up Stan iteration, we sample a set of hyperparameters Θ and $\boldsymbol{f^*}$, then generate Y^* with the sampled $\boldsymbol{f^*}$ and σ. Therefore, the posterior distribution of y^* is a Gaussian mixture for which we don't have an analytical form. Nor can we assert that y^* follows a Gaussian distribution; 3) with that said, for each iteration $j \in \{1, 2, ..., m\}$, as y^{*j} follows a Gaussian distribution, $E(y_i^{*j}) = f_i^{*j},\ \forall\ i \in \{1, 2, ..., n2\}$. Therefore, once we have obtained converged posterior results and collected a sufficient amount of effective

posterior iterations, $\boldsymbol{K_*}^T\boldsymbol{K}^{-1}Y$ should be an unbiased estimation of Y^*, and we can get the posterior distribution of $\boldsymbol{f^*}$ via numeric sampling.

For our Gaussian process learning, let's say we have run m rounds of post warm-up iterations. We will end up with m pairs of $\Theta = \{\alpha, \rho, \sigma\}$ hyperparameters. And $E(\boldsymbol{K_*}^T\boldsymbol{K}^{-1}Y)$ can be written as:

$$E(\boldsymbol{K_*}^T\boldsymbol{K}^{-1}Y) = \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{K_{*i}}^T\boldsymbol{K_i}^{-1}Y, \quad i \in \{1, 2, \ldots, m\} \tag{4.8}$$

Since we have all m pairs of $\Theta = \{\alpha, \rho\}$ hyperparameters, we can get the full posterior distribution of $\boldsymbol{K_*}^T\boldsymbol{K}^{-1}Y$. Numerically, we can calculate any confidence interval given a level of confidence (i.e. posterior quantile that centers at the posterior distribution median). That's to say, we can get an expectation of the y^* predictions as well as the uncertainty band associated with these predictions.

In order to describe the trend and changes of y^*, we can take the first derivative of $\boldsymbol{K_*}^T\boldsymbol{K}^{-1}Y$. For any iteration $i \in 1, 2, \ldots, m$, the first derivative of $\boldsymbol{K_*}^T\boldsymbol{K}^{-1}Y$ is written as:

$$\begin{aligned}
\frac{\partial k(x, x_*)}{\partial x_*} &= \frac{\alpha_i^2(x - x_*)}{\rho_i^2} exp\Big[-\frac{(x - x_*)^2}{2\rho_i^2}\Big] \\
&= \frac{1}{\rho_i^2}(x - x_*)k(x, x_*) \\
\frac{\partial \boldsymbol{K_*}^T\boldsymbol{K}^{-1}Y}{\partial x_*} &= \Big[\frac{\partial k(x_1, x_*)}{\partial x_*}, \frac{\partial k(x_2, x_*)}{\partial x_*}, \ldots, \frac{\partial k(x_{n1}, x_*)}{\partial x_*}\Big]\boldsymbol{K}^{-1}Y \\
&= \frac{1}{\rho_i^2}[X - x_*]^T \odot [k(X, x_*)]^T\boldsymbol{K}^{-1}Y
\end{aligned} \tag{4.9}$$

$\odot$ here means element-wise product.

Therefore, with m pairs of $\Theta = \{\alpha, \rho, \sigma\}$ hyperparameters, we can also get the posterior distribution of the first derivative for the mean of f^*, which converges to the mean of y^*. And we can plot an uncertainty interval band of the first derivative in regards to a certain level of confidence. Again, this uncertainty interval is represented by a quantile specified by the confidence level centered at the median from the posterior distribution of the first derivatives. We will from now on refer to this interval as the uncertainty interval [101].

4.2.3 Distributional Properties of the First Derivative

Inspired by the work of McHutchon [102], we would like to derive the distribution of $\frac{\partial \boldsymbol{f^*}}{\partial x^*}|\Theta$ in detail for bivariate cases. Even though the posterior distribution of $\frac{\partial \boldsymbol{f^*}}{\partial x^*}$ does not have an analytical closed-form solution, as will be illustrated later in this section, we can sample from the distribution of $\frac{\partial \boldsymbol{f^*}}{\partial x^*}|\Theta$ and estimate the posterior distribution of the slopes numerically.

Firstly, as described in Equation 4.6, we denote $\bar{\boldsymbol{f}}^* = \boldsymbol{K_*}\boldsymbol{K}^{-1}Y$; as described in Equation 4.9, we have

$$\begin{aligned}
\frac{\partial \bar{\boldsymbol{f}}^*}{\partial x^*}|\Theta &= \frac{\partial \boldsymbol{K_*}^T \boldsymbol{K}^{-1}Y}{\partial x_*} \\
&= \Big[\frac{\partial k(X, x_*)}{\partial x_*}\Big]^T \boldsymbol{K}^{-1}Y \\
&= \frac{1}{\rho^2}[X - x_*]^T \odot [k(X, x_*)]^T \boldsymbol{K}^{-1}Y \\
&= \frac{1}{\rho^2}\big[[X - x_*] \odot [k(X, x_*)]\big]^T \boldsymbol{K}^{-1}Y
\end{aligned} \tag{4.10}$$

For the convenience of notation, we denote $K_p = \frac{1}{\rho^2}[X - x_*] \odot [k(X, x_*)]$.

Computationally, Cholesky decomposition [103] is adopted to effectively decompose the covariance matrix $\boldsymbol{K}$ into the product of a lower triangular matrix $\boldsymbol{L}$ and its conjugate transpose $\boldsymbol{L^*}$.

$$\boldsymbol{K} = \boldsymbol{L}\boldsymbol{L^*} \tag{4.11}$$

Then, Equation 4.10 can be written as:

$$\begin{aligned}
\frac{\partial \bar{\boldsymbol{f}}^*}{\partial x^*}|\Theta &= K_p^T \boldsymbol{K^{-1}} Y \\
&= K_p^T [\boldsymbol{L}\boldsymbol{L^*}]^{-1} Y \\
&= \big(\boldsymbol{L}^{-1}K_p\big)^T \boldsymbol{L}^{-1}Y
\end{aligned} \tag{4.12}$$

Secondly, we will use the definition of the first derivative to derive the properties of $\frac{\partial \boldsymbol{f^*}}{\partial x^*}|\Theta$.

As addressed in Equation 4.6, $\boldsymbol{f^*}$ follows a Gaussian Process such that we can rewrite any $f^*(x^*) = \bar{f}^*(x^*) + z_*$. By definition:

$$\begin{aligned}\frac{\partial f^*}{\partial x^*} &= \lim_{\delta \to 0} \frac{f^*(x^* + \delta) - f^*(x^*)}{\delta} \\ &= \lim_{\delta \to 0} \frac{\bar{f}^*(x^* + \delta) + z_\delta - \bar{f}^*(x^*) - z_*}{\delta} \\ &= \lim_{\delta \to 0} \frac{\bar{f}^*(x^* + \delta) - \bar{f}^*(x^*)}{\delta} + \lim_{\delta \to 0} \frac{z_\delta - z_*}{\delta} \\ &= \frac{\partial \bar{f}^*}{\partial x^*} + \lim_{\delta \to 0} \frac{z_\delta - z_*}{\delta}\end{aligned} \tag{4.13}$$

Now we have decomposed the first derivative of f^* into two parts, the first part is known by Equation 4.12 while the second part needs further derivation. To understand the second term, we can write up the relationship between z_δ and z_* first:

$$\begin{cases} f^*(x^* + \delta) = \bar{f}^*(x^* + \delta) + z_\delta \\ f^*(x^*) = \bar{f}^*(x^*) + z_* \end{cases} \tag{4.14}$$

$$\begin{bmatrix} z_* \\ z_\delta \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} k_{**} - K_*^T \boldsymbol{K}^{-1} K_* & k_{*\delta} - K_*^T \boldsymbol{K}^{-1} K_\delta \\ k_{\delta *} - K_\delta^T \boldsymbol{K}^{-1} K_* & k_{\delta\delta} - K_\delta^T \boldsymbol{K}^{-1} K_\delta \end{bmatrix} \right) \tag{4.15}$$

where $k_{**} = k(z_*, z_*), k_{*\delta} = k(z_*, z_\delta), k_{\delta *} = k(z_\delta, z_*), k_{\delta\delta} = k(z_\delta, z_\delta)$,
$K_* = [k(X, z_*)]^T, K_\delta = [k(X, z_\delta)]^T$.

Therefore,

$$E(\lim_{\delta \to 0} \frac{z_\delta - z_*}{\delta}) = \lim_{\delta \to 0} E(\frac{z_\delta - z_*}{\delta}) = 0 \tag{4.16}$$

Since z_* and z_δ are Gaussian distributed, the term $\lim_{\delta \to 0} \frac{z_\delta - z_*}{\delta}$ is also Gaussian distributed. Now we know the expectation of the term and would like to derive the

variance of it as well, in order to specify its distribution.

$$
\begin{aligned}
& var(\lim_{\delta\to 0}\frac{z_\delta - z_*}{\delta}) \\
=& \lim_{\delta\to 0}\frac{1}{\delta^2}[var(z_\delta)+var(z_*)-cov(z_\delta,z_*)-cov(z_*,z_\delta)] \\
=& \lim_{\delta\to 0}\frac{1}{\delta^2}[k_{\delta\delta}-K_\delta^T\boldsymbol{K}^{-1}K_\delta+k_{**}-K_*^T\boldsymbol{K}^{-1}K_*]-k_{\delta *}+K_\delta^T\boldsymbol{K}^{-1}K_*-k_{*\delta}+K_*^T\boldsymbol{K}^{-1}K_\delta] \\
=& \lim_{\delta\to 0}\frac{1}{\delta^2}[k_{\delta\delta}+k_{**}-k_{*\delta}-k_{\delta *}] \\
=& \lim_{\delta\to 0}\frac{1}{\delta^2}[k_{\delta\delta}+k_{**}-k_{*\delta}-k_{\delta *}-(K_\delta-K_*)^T\boldsymbol{K}^{-1}(K_\delta-K_*)] \\
=& \lim_{\delta\to 0}\frac{1}{\delta}\big[(\frac{k_{\delta\delta}-k_{*\delta}}{\delta})-(\frac{k_{\delta *}-k_{**}}{\delta})\big]-\lim_{\delta\to 0}\frac{(K_\delta-K_*)^T}{\delta}\boldsymbol{K}^{-1}\frac{(K_\delta-K_*)}{\delta}
\end{aligned}
\tag{4.17}
$$

Regarding to the first term in Equation 4.17,

$$
\begin{aligned}
& \lim_{\delta\to 0}\frac{1}{\delta}\big[(\frac{k_{\delta\delta}-k_{*\delta}}{\delta})-(\frac{k_{\delta *}-k_{**}}{\delta})\big] \\
=& \lim_{\delta\to 0}\frac{1}{\delta}\big[\frac{k(x^*+\delta,x^*+\delta)-k(x^*,x^*+\delta)}{\delta}\big]-\lim_{\delta\to 0}\frac{1}{\delta}\big[\frac{k(x^*+\delta,x^*)-k(x^*,x^*)}{\delta}\big] \\
=& \lim_{\delta\to 0}\frac{1}{\delta}\frac{\partial k(x_1,x_2)}{\partial x_1}|_{x_1=x^*,x_2=x^*+\delta}-\lim_{\delta\to 0}\frac{1}{\delta}\frac{\partial k(x_1,x_2)}{\partial x_1}|_{x_1=x^*,x_2=x^*} \\
=& \lim_{\delta\to 0}\frac{1}{\delta}\Big[\frac{\partial k(x_1,x_2)}{\partial x_1}|_{x_1=x^*,x_2=x^*+\delta}-\frac{\partial k(x_1,x_2)}{\partial x_1}|_{x_1=x^*,x_2=x^*}\Big] \\
=& \frac{\partial^2 k(x_1,x_2)}{\partial x_1\partial x_2}|_{x_1=x^*,x_2=x^*} \\
=& \frac{\partial^2\alpha^2 exp(-\frac{1}{2\rho^2}(x_1-x_2)^2)}{\partial x_1\partial x_2}|_{x_1=x^*,x_2=x^*} \\
=& \frac{\alpha^2}{\rho^2}exp\big(-\frac{1}{2\rho^2}(x_1-x_2)^2\big)[1-\frac{1}{\rho^2}(x_1-x_2)^2]|_{x_1=x^*,x_2=x^*} \\
=& \frac{\alpha^2}{\rho^2}
\end{aligned}
\tag{4.18}
$$

Regarding to the second term in Equation 4.17,

$$
\begin{aligned}
&\lim_{\delta\to 0}\frac{(K_\delta - K_*)^T}{\delta}\boldsymbol{K}^{-1}\frac{(K_\delta - K_*)}{\delta}\\
=&\lim_{\delta\to 0}\frac{(K_\delta - K_*)^T}{\delta}\boldsymbol{K}^{-1}\lim_{\delta\to 0}\frac{(K_\delta - K_*)}{\delta}\\
=&\lim_{\delta\to 0}\frac{[k(X,x^*+\delta)-k(X,x^*)]^T}{\delta}\boldsymbol{K}^{-1}\lim_{\delta\to 0}\frac{[k(X,x^*+\delta)-k(X,x^*)]}{\delta}\\
=&\Big[\frac{\partial k(X,x^*)}{\partial x^*}\Big]^T\boldsymbol{K}^{-1}\Big[\frac{\partial k(X,x^*)}{\partial x^*}\Big]\\
=&K_p^T\boldsymbol{K}^{-1}K_p\\
=&K_p^T\big(\boldsymbol{L}\boldsymbol{L^*}\big)^{-1}K_p\\
=&\big(\boldsymbol{L}^{-1}K_p\big)^T\boldsymbol{L}^{-1}K_p
\end{aligned}
\tag{4.19}
$$

Therefore, we get the distributional properties of $\frac{\partial f^*}{\partial x^*}$:

$$
\begin{aligned}
\frac{\partial f^*}{\partial x^*}|\Theta \ &\sim \mathcal{N}\Big(\frac{\partial \bar{f}^*}{\partial x^*},\ \sqrt{\frac{\alpha^2}{\rho^2}-K_p^T\boldsymbol{K}^{-1}K_p}\Big)\\
&= \mathcal{N}\Big(\big(\boldsymbol{L}^{-1}K_p\big)^T\boldsymbol{L}^{-1}Y,\ \sqrt{\frac{\alpha^2}{\rho^2}-\big(\boldsymbol{L}^{-1}K_p\big)^T\boldsymbol{L}^{-1}K_p}\Big)
\end{aligned}
\tag{4.20}
$$

4.3 Implementation and Visualization in R

The R package *gpviz* is available on GitHub. User can install the package with simply two lines:

```
install.packages("devtools")
devtools::install_github("PPJane/gpviz")
```

After the installation is completed, users need to call the package into memory to use it by executing:

```
library(gpviz)
```

The R package *gpviz* utilizes the rStan package to model the Gaussian process with Stan and implements the Gaussian process with the function *gpslope*.

The usage of the function is simply:

```
resultlst <- gpslope(data = hsbReport_cleaned,
                     x_ = "deposit_asset_ratio",
                     y_ = "loss_asset_ratio",
                     iter = 1000,
                     condition1 = ">0")
```

The function *gpslope* takes in several arguments: 1) a data frame consisting of preprocessed variables (such as missing values and outliers), 2) a user designated x variable and y variable, 3) a user specified number of Stan iterations, and 4) user defined conditions of slopes where the corresponding region of x might be of interest.

The full specification of the function is:

```
gpslope (data, x_, y_,
         trainingpect = 0.1, seed = 123,
         iter = 100, chains = 4,
         conf = 0.5, condition1 = NULL, condition2 = NULL,
         dn = 64,
         rho_alpha = 4, rho_beta = 4,
         alpha_mean = 0, alpha_sd = 1,
         sigma_mean = 0, sigma_sd = 1)
```

The function *gpslope* implements the Gaussian process for regression by taking "trainingpect" percent of the sample as training data and the whole sample as test data. It runs the user designated number of iterations and number of chains via rStan. And the user needs to specify the conditions under which the slopes will be considered of interest; the user can also specify the distribution properties for hyperparameters, which we will discuss in more detail in Chapter 4.3.3. When rStan draws samples from the Gaussian process, it also samples the predictions' slopes such that we can empirically estimate the distribution of the slopes. The output of function *gpslope* is

a list consisting of the Stan fit object, three ggplot objects, and the argument values that the user passed to the function.

Users can make Stan fit checks against the Stan fit object for model diagnostics. Similar to many other methods, Stan requires the convergence diagnostic for best practices. Because the number of iterations, the number of chains, and the prior distribution properties for the hyperparameters are all tunable. Users need to check some rule-of-thumb diagnostics to validate the fit, such as R-hat < 1 and the number of effective samples per iteration [104]. Some visual diagnostics packages are also very useful, such as the bayesplot package [89, 105].

Users can arrange the ggplot objects to visualize the distribution of predictions' slopes either by uncertainty interval estimation or by density estimation. We will show examples separately in Section 4.3.1.2 and 4.3.1.3. Users can also tailor-make their graphs by adding layers to the ggplot objects generated by *gpslope*.

Another function *gpapply* is conveniently designed when the user has a handful of features and wants to examine the bivariate relationships one at a time. The function *gpapply* allows the user to specify a list of independent variables and loop over each independent variable to explore its relationship with a dependent variable. The function arranges the outputs as pdf files consisting of two ggplots stacked on top of each other. An example is illustrated in Section 4.3.2.

4.3.1 Example 1: Exploring the Relationship Between Debt-to-asset Ratio and Net-loss on Asset When Banks Failed

We collected 2,961 cases of federal deposit insurance corporation failure and assistance transaction in United States (50 states and DC) between 1986 and 2017 [1]. We preprocessed the data set by: 1) eliminating missing values of estimated loss gathered by the end of 2017 and ending up with 2,638 complete cases, 2) mutating two ratios of interest, and 3) removing outliers if any:

[1] Data: FDIC/Failures and Assistance Transactions/Historical Statistics on Banking, URL: https://www5.fdic.gov/hsob/SelectRpt.asp?EntryTyp=30&Header=1

- Compute Deposit-asset Ratio

$$Deposit\text{-}asset\ Ratio = \frac{Deposit}{Asset}$$

- Compute Net-loss on Asset [2]

$$Net\text{-}loss\ on\ Asset = \frac{Estimated\ Net\text{-}loss}{Asset}$$

- Remove an outlier case (Meridian Saving Association based in Arlington, TX failed on 4/6/1989 with net-loss on asset as high as 10.87), as this outlier point is extremely far away from the rest of the data points (see the left side of Figure 4.2). Then we have 2,637 complete cases as plotted in the right side of Figure 4.2.

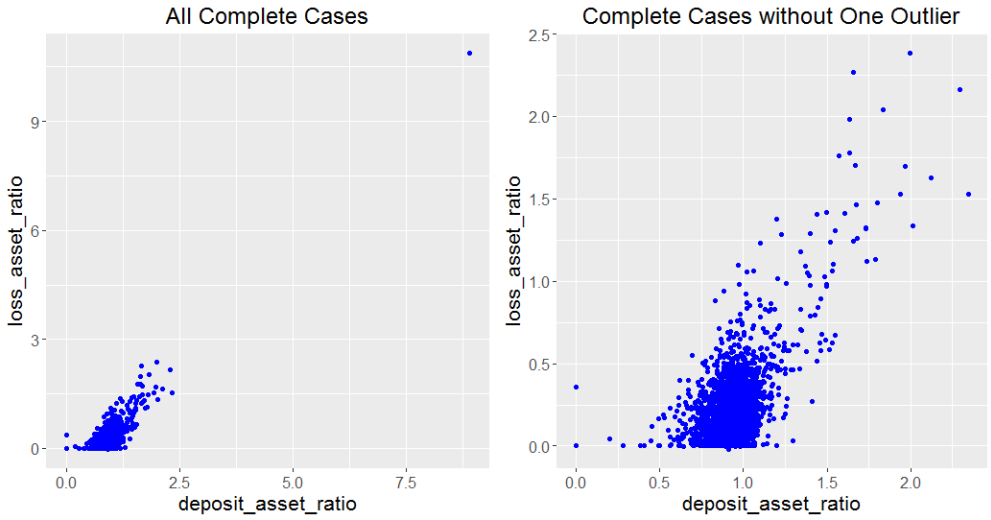


Figure 4.2: Data Pre-processing Outlier Removal

[2] The net-loss on asset could be negative when the estimated loss is negative. This scenario is very rare, but is possible when institutions were operated under government control between the date of failure and the final resolution date in a bridge bank operated by a conservatorship operated by the Resolution Trust Corporation or the FDIC. Seven institutions like this operated by the Resolution Trust Corporation in the year 1990 and 1991 have negative net-loss outcomes; Net-loss on asset could be greater than one. This scenario is also infrequent, but possible for the same reason as stated in the case when net-loss is negative. 42 institutions operated by the Resolution Trust Corporation between 1989 and 1991 have more net-loss than asset book value recored by the date of failure.

After data preprocessing, we get a clean data frame. Then we input this data frame into the function *gpslope* for further analysis.

Example usage:

```
resultlst <- gpslope(data = hsbReport_cleaned,
                     x_ = "deposit_asset_ratio",
                     y_ = "loss_asset_ratio",
                     seed = 123456,
                     iter = 2000, condition1 = ">0")
```

The full specification for this example is:

```
resultlst <- gpslope(data = hsbReport_cleaned,
                     x_ = "deposit_asset_ratio",
                     trainingpect = 0.1, seed = 123456,
                     iter = 2000, chains = 4,
                     conf = 0.5,
                     condition1 = ">0", condition2 = NULL,
                     dn = 64,
                     rho_alpha = 4, rho_beta = 4,
                     alpha_mean = 0, alpha_sd = 1,
                     sigma_mean = 0, sigma_sd = 1)
```

The function *gpslope* takes the preprocessed data frame "hsbReport_cleaned" as input and trains 10% of the data with the Gaussian process via the rStan package. The Stan model is specified in a separate Stan file listed in Section 4.3.3, and the Stan file specifications are explained in 4.2.1. Notice that the training percentage is set low for the merit of computing speed. Users can increase the training percentage if the data set is rather small. But keep in mind that when the data set grows, sampling from many training instances can take a long time.

```
pred_fit <- resultlst[["stanfit"]]
gpPlot <- resultlst[["gppred"]]
gpSlopeCI <- resultlst[["gpCI"]]
gpSlopeDen <- resultlst[["gpDensity"]]
argVal <- resultlst[["argVal"]]
```

The function *gpslope* outputs a list consisting of the Stan fit object, three ggplot objects, and the argument values the user passed to the function. The three ggplot objects are: 1) Gaussian process regression on the training and test data points, with ten randomly picked Gaussian process realizations; 2) Gaussian process predictions' first derivative interval at the user-specified confidence level; 3) Gaussian process predictions' first derivative distribution density estimation, with color gradient shading from grey (when the probability density gets low) to red (when the probability density of the first derivative gets higher than 97.5% quantile of probability density point estimations).

4.3.1.1 Stan Fit Diagnostic

```
pred_fit <- resultlst[["stanfit"]]
print(pred_fit, pars = c("alpha", "rho", "sigma"))

Inference for Stan model: predict_gauss.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

      mean   se_mean sd 2.5%  25%  50%  75%  97.5% n_eff Rhat
alpha 1.02    0.01 0.37 0.51 0.75 0.95 1.21  1.93  2511    1
rho   0.94    0.00 0.24 0.59 0.77 0.90 1.06  1.50  2604    1
sigma 0.15    0.00 0.01 0.14 0.15 0.15 0.16  0.17  3199    1
```

Users can check some rule-of-thumb diagnostics for validation, such as 1) R-hat < 1, and 2) the number of effective samples being sufficiently large when compared with the number of total post-warmup draws [104].

```
library(rstan)
traceplot(pred_fit, pars = c("alpha", "rho", "sigma"))
```

Users can also spot the trace plots of the hyperparameters by simply looking at the shape of trace plots. Trace plot provides a visual way to inspect sampling behavior and assess mixing across chains and convergence. A caterpillar shaped trace plot often suggests a fairly good exploration of the parameter space. With 4,000 post-warmup draws, we get the posterior distribution of the hyperparameters. The trace plots of the hyperparameters in this example are depicted in Figure 4.3. All hyperparameters α, ρ and σ show nice caterpillar like traces, which means that the sampler has explored the parameter space fairly well and estimations from the four Monte Carlo Marchov chains converge. Some visual diagnostics packages are also very useful, such as the bayesplot package [89, 105].

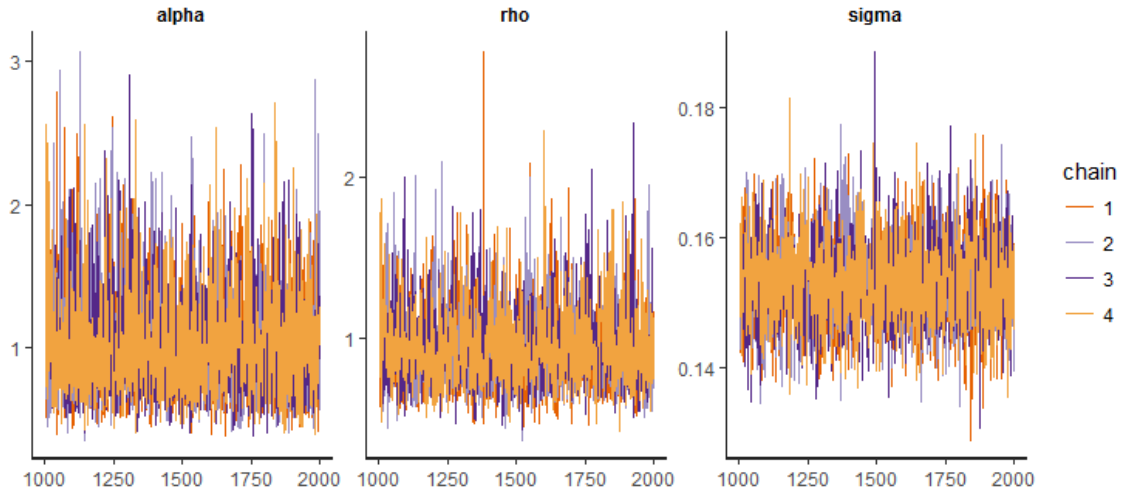


Figure 4.3: Trace Plots for Hyper-parameters

4.3.1.2 Uncertainty Interval Estimation

```
library(grid)
library(ggplot2)
pdf(file = paste0("./gp_CI_", argVal$data, "_",
                  ifelse(is.numeric(argVal$conf),
                         argVal$conf,
                         formals(gpslope)$conf), "_",
                  argVal$x_, "_", argVal$y_, ".pdf"),
    height = 20, width = 18, paper = "letter")
grid.newpage()
grid.draw(rbind(ggplotGrob(gpPlot),
                ggplotGrob(gpSlopeCI), size = "last"))
dev.off()
```

After validating the Stan fit, users can check the uncertainty interval estimation plots using the example code above. Users can also tailor-make some plots by adding layers to the ggplot objects. In this example, we highlight the regions that correspond to positive relationships between the deposit-asset ratio and the loss-asset ratio at the confidence level of 50%, as shown in Figure 4.4. When the deposit-asset ratio is between 0.7 and 2.2, the relationship between the deposit-asset ratio and the loss-asset ratio is believed to be positive at the confidence level of 50%. And the segment between 1.25 and 1.85 corresponds to a higher slope magnitude, which sends out a signal to bank regulators that such a segment needs to be monitored closely.

It is rather intuitive to observe a positive relationship between the deposit-asset ratio and the loss-asset ratio, as the bank gets deeper into debt. When the bank fails, it will be hard to recover its debt. Of course this relationship shall be impacted by many other factors, such as the lending and economic environment, e.g. it is much harder to collect debt during an economic depression.

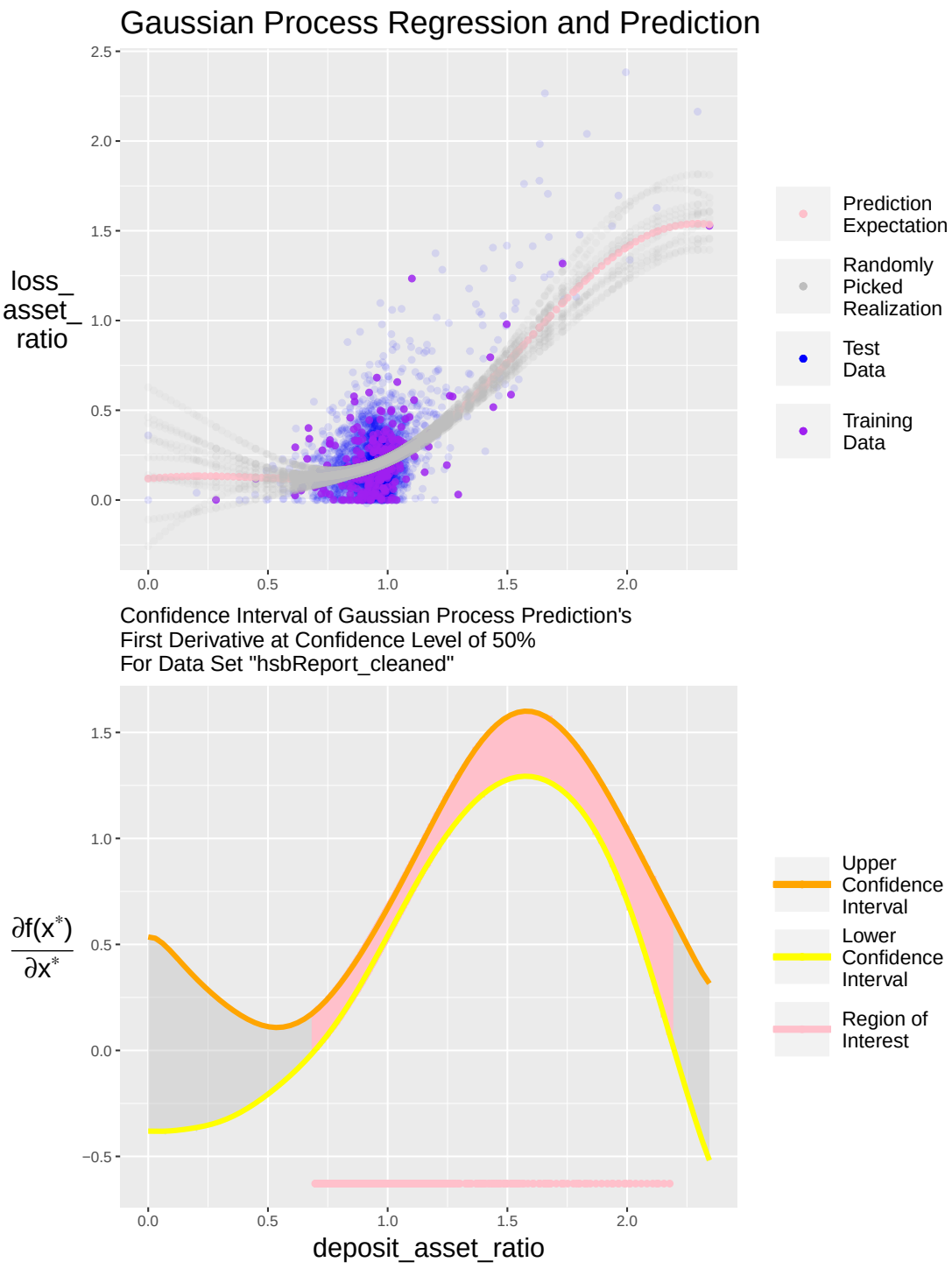


Figure 4.4: Uncertainty Interval Estimation at 50% Confidence Level

At the confidence level of 50%, we can see that a region of the deposit-asset ratio $\in [0.7, 2.2]$ has positive slopes of predictions, which indicates that the expected net-loss on asset ratio increases as the deposit-asset ratio increases. And the magnitude of the slopes keeps increasing up until the deposit-asset ratio around 1.6, and then starts decreasing again.

Users can also specify a different confidence level. Notice that as the confidence level increases, the region of interest naturally will shrink.

```
resultlst <- gpslope(data = hsbReport_cleaned,
                     x_ = "deposit_asset_ratio",
                     y_ = "loss_asset_ratio",
                     seed = 123456, iter = 2000,
                     conf = 0.95,
                     condition1 = ">0")
```

We hereby increase the confidence level from 50% to 95%. And the result of the code block above is demonstrated in Figure 4.5. We find out that when we increase the confidence level to 95%, the region of interest shrinks to $[0.8, 2.05]$.

In this example, we used the function *gpslope* to extract insights from an FDIC failed banks' dataset to explore any region of interest that could aid in setting regulatory alerts for FDIC insured banks and institutes. This could be a useful tool for regulators, especially when used to assess financial risks with a full picture of risk uncertainty, as the Bayesian method produces not only a single point estimation, but also a distribution of point estimations.

Figure 4.5: Uncertainty Interval Estimation at 95% Confidence Level

4.3.1.3 Density Estimation

```
library(grid)
library(ggplot2)
pdf(file = paste0("./gp_density_", argVal$data, "_",
                  argVal$x_, "_", argVal$y_, ".pdf"),
    height = 20, width = 18, paper = "letter")
grid.newpage()
grid.draw(rbind(ggplotGrob(gpPlot),
                ggplotGrob(gpSlopeDen), size = "last"))
dev.off()
```

The ggplot object "gpSlopeDen" is a density estimation of the Gaussian process predictions' first derivatives. Contrary to the confidence interval estimation, users do not need to specify a confidence level or conditions to extract regions of interest. Notice that there is a "dn" argument in the function *gpslope*, it is the "n" argument in the R function *density*, which refers to the number of equally spaced points at which the density is to be estimated. The R function *density* suggests users to use the "n" argument by the power of 2. We set the default value to 64 since it gives a fairly good visual result. User can increase the default value when the density shade seems to break up.

Users can visually check the density estimation and easily spot the region where an intense relationship between the independent variable and the dependent variable is predicted. The intense relationship usually corresponds to an area where an abundance of training data points cluster such that the Gaussian process has a high confidence in its predictions.

After validating the Stan fit, users can check the confidence interval estimation plots by the example code above. Users can also tailor-make the plots by adding layers to the gpSlopeDen object.

The output of the example code is shown in Figure 4.6. There is an intense relationship between the debt-asset ratio and the loss-asset ratio when the debt-asset ratio is between 0.7 and 1.2. It appears to be close to zero around 0.75, but keeps increasing from 0.8 to 1.2. It seems that after the deposit-asset ratio goes over 1.0, the risk of losing asset value is believed to soar, increasing at a faster rate than the deposit-asset ratio. This gives regulators a clue about how to balance their risk control policy while protecting commercial banks' profitability to some extent.

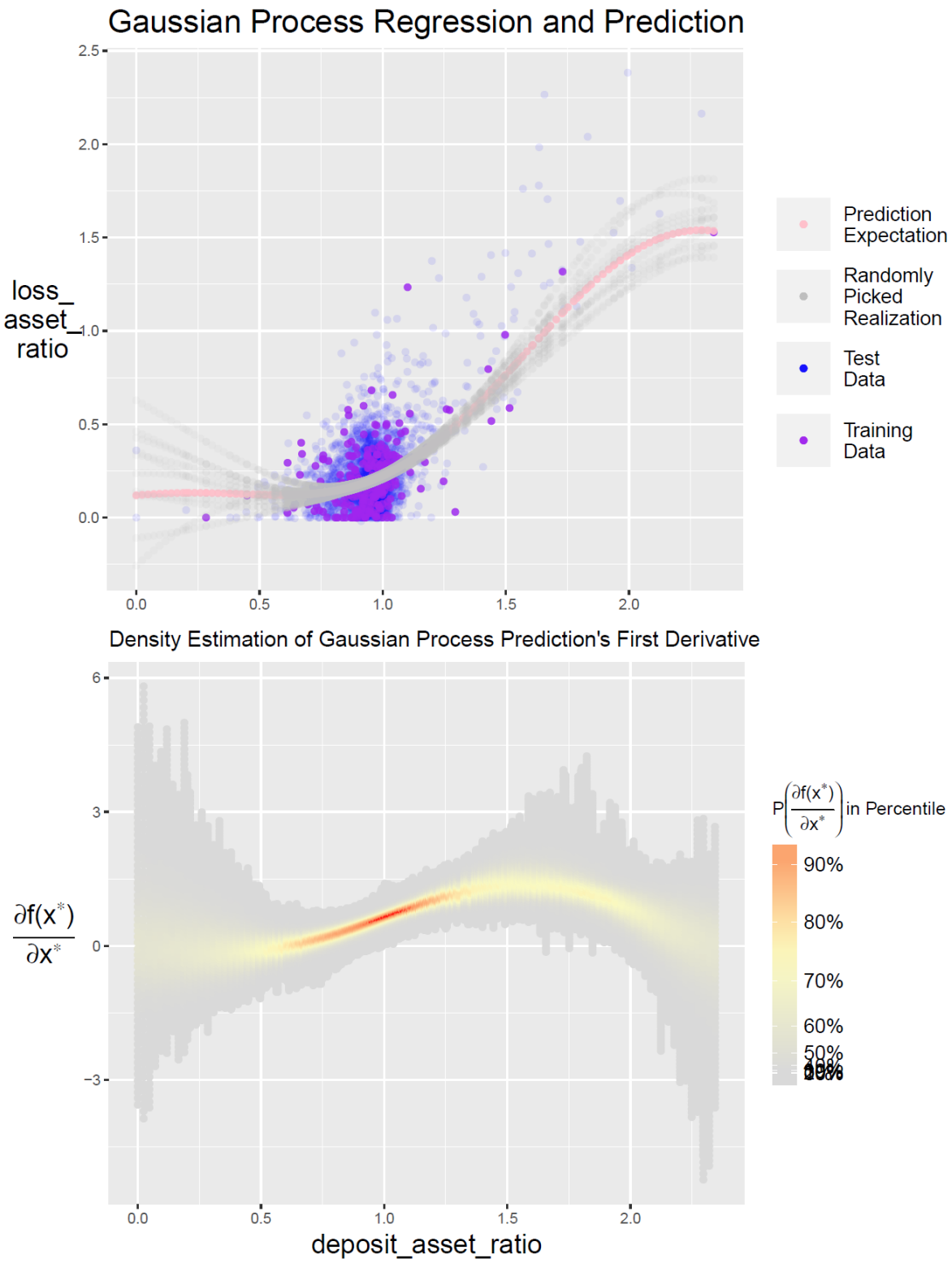


Figure 4.6: Density Estimation

4.3.2 Example 2: Exploring Relationships Between Wine Physiochemical Properties and Wine Quality

In this example, we used the wine quality data set available on the UCI Machine Learning Repository [106]. There are two data sets related to red and white variants of the Portuguese "Vinho Verde" wine, consisting of physiochemical properties (numeric independent variables) and sensory scores (depdent variable).

Since we have a handful of independent variables to be examined, we can use the function *gpapply* to loop the function *gpslope* over and generate reports in plots.

```
wine_red_x_lst <- c("sulphates", "pH", "total.sulfur.dioxide",
                    "alcohol", "volatile.acidity",
                    "free.sulfur.dioxide", "fixed.acidity",
                    "residual.sugar", "chlorides",
                    "density", "citric.acid")
gpapply(data = winequality_red,
        x_lst = wine_red_x_lst, y_ = "quality",
        iter = 1000, condition1 = '>0')
```

Users just need to assign a list of variable names to be examined and feed the list of variables through the function *gpapply*'s argument "x_lst". The rest of the arguments stay the same as in the function *gpslope*.

The outputs of the function *gpapply* are pdf documents containing the plots of Gaussian process predictions, the prediction's slope's confidence interval, and its density estimations. After studying the output results, we summarized our findings in Table 4.1. Notice that this package is for bivariate relationship examination and does not perform multivariate analyses. We do acknowledge that if variables have dependencies, the actual impact of the variables could be different.

The most important features that have shown a region of interest to potentially improve red wine quality are "sulphates", "citric acidity" and "alcohol". Meanwhile, "chlorides" and "volatile acidity" have an intense negative relationship with red wine

Table 4.1: Summary for Red Wine Quality Data Set

		Estimates at Confidence Level of 0.5		Estimates from Density	
Feature	Relationship with wine quality	Region of Interest	Maximum Slope	Tense Relationship with wine quality	Corresponding Region
sulphates	concave	[0.1, 0.875]	5.000	positive	[0.625, 0.875]
citric.acidity	concave	[0, 0.38]	2.500	positive	[0.18, 0.38]
alcohol	concave	[8.3, 12.1]	1.100	positive	[9.5, 12]
fixed.acidity	quasi concave	[4.5, 6.5] & [7.5, 11.1]	0.500	slightly positive	[7.5, 10]
pH	concave	[2.72, 3.125]	1.500	uncorrelated	[3.13, 3.45]
residual.sugar	mostly negative	[0.8, 1.9]	0.380	uncorrelated	[2, 6]
free.sulfur.dioxide	mostly negative	[1, 9]	0.075	uncorrelated	[10, 30]
total.sulfur.dioxide	mostly negative	[5, 25]	0.035	uncorrelated	[25, 80]
chlorides	negative			negative	[0, 0.15]
volatile.acidity	negative	NA		negative	[0.45, 0.68]
density	uncorrelated			uncorrelated	whole sample range

quality, which also gives wine manufacturers a good guideline to strictly control the amount of chlorides and volatile acidity during wine making, especially in the region where the negative relationship is pretty dense. All the plots are enclosed in Appendix D.

We hereby demonstrate the bivariate relationship between one of the most important features "sulphates" (identified by Cortez et al. [1]) and red wine quality in Figures 4.7 and 4.8.

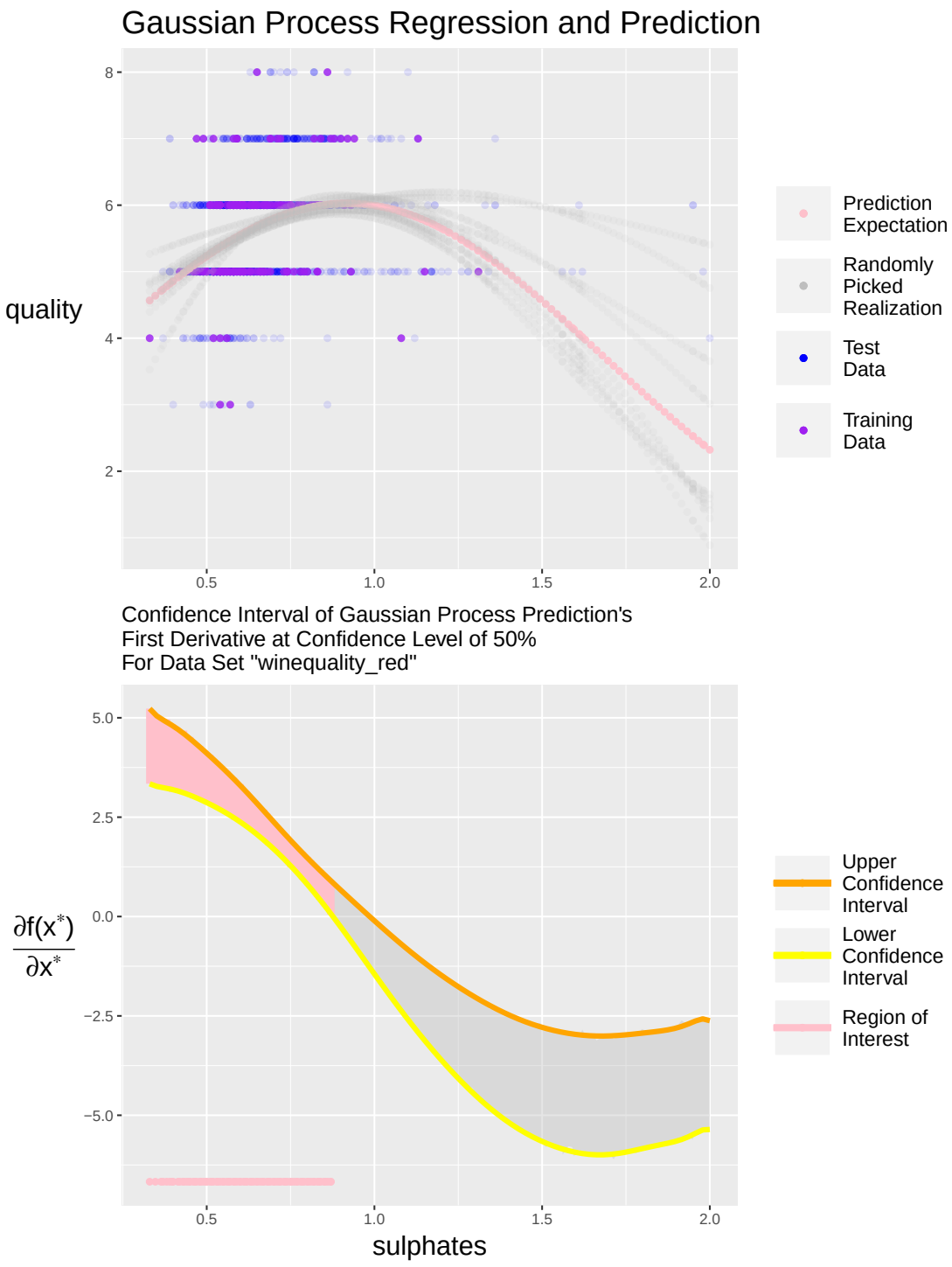


Figure 4.7: Confidence Interval Estimation between Sulphates and Red Wine Quality

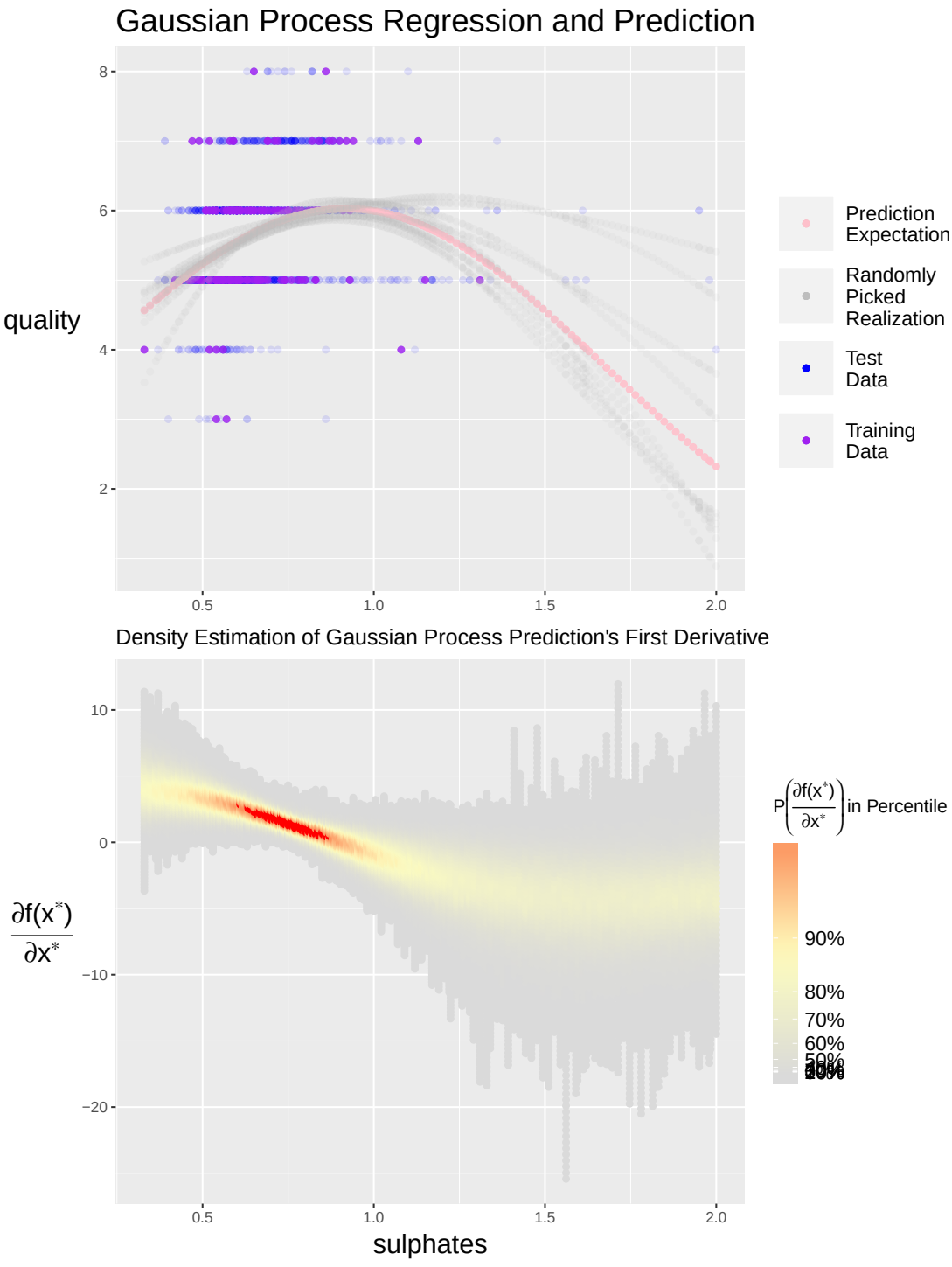


Figure 4.8: Density Estimation between Sulphates and Red Wine Quality

Other researchers who studies the wine quality data set [1] have also ranked the importance of each physiochemical feature in red and white wine quality prediction, as reported in Figure 4.9. In their finding, sulphates are also the most important feature when it comes to red wine. They speculate that an increase in sulphates might be related to the fermenting nutrition, which is very important to improve the wine's aroma [1]. Note that sulphates here are not the same as sulfites, sulfur-containing chemicals often used as food preservatives [107]. Sulphates are anti-inflammatory and anti-depressant. They are needed for making stomach acid and digestive enzymes, so that we can break down the food we eat into useful components [108].

We also applied the function *gpapply* to the white wine data set and summarized our findings in Table 4.2. All the figures are enclosed in Appendix D.

Table 4.2: Summary for White Wine Quality Data Set

		Estimates at Confidence Level of 0.5		Estimates from Density	
Feature	Relationship with wine quality	Region of Interest	Maximum Slope	Tense Relationship with wine quality	Corresponding Region
alcohol	quasi linear	[8, 13.7]	0.400	positive	[10, 11.5]
sulphates	quasi linear	[0.22, 0.89]	1.100	positive	[0.45, 0.73]
pH	quasi linear	[2.72, 3.42]	1.350	positive	[3.13, 3.4]
citric.acidity	quasi concave	[0, 0.3] & [1.2, 1.25]	7.500	positive negative	[0.25, 0.3] [0.3, 0.4]
free.sulfur.dioxide	non-linear	[0, 35] & [65, 90] & [245, 270]	0.100	positive negative	[15, 35] [35, 50]
residual.sugar	non-linear	[0, 1]	0.250	uncorrelated	[2, 18]
fixed.acidity	quasi concave	[3.75, 5.4]	0.500	slightly negative	[6, 7.8]
total.sulfur.dioxide	concave	[5, 80]	0.018	negative	[90, 190]
volatile.acidity	quasi linear			negative	[0.27, 0.48]
chlorides	negative	NA		negative	[0.05, 0.2]
density	negative			negative	[0.9875, 1.005]

White wine features "alcohol", "sulphates" and "pH" are found to have almost linear positive relationships with white wine quality. The feature "free sulfur dioxide" has a highly nonlinear relationship with white wine quality. And features "total sulfur dioxide", "volatile acidity", "chlorides" and "density" show intense negative relationships with white wine quality.

Other researchers [109] have found that the accuracy rate for the classifiers built on the white wine date set is influenced by a higher number of physiochemical features including "alcohol", "density", "free sulfur dioxide", "chlorides", "citric acid",

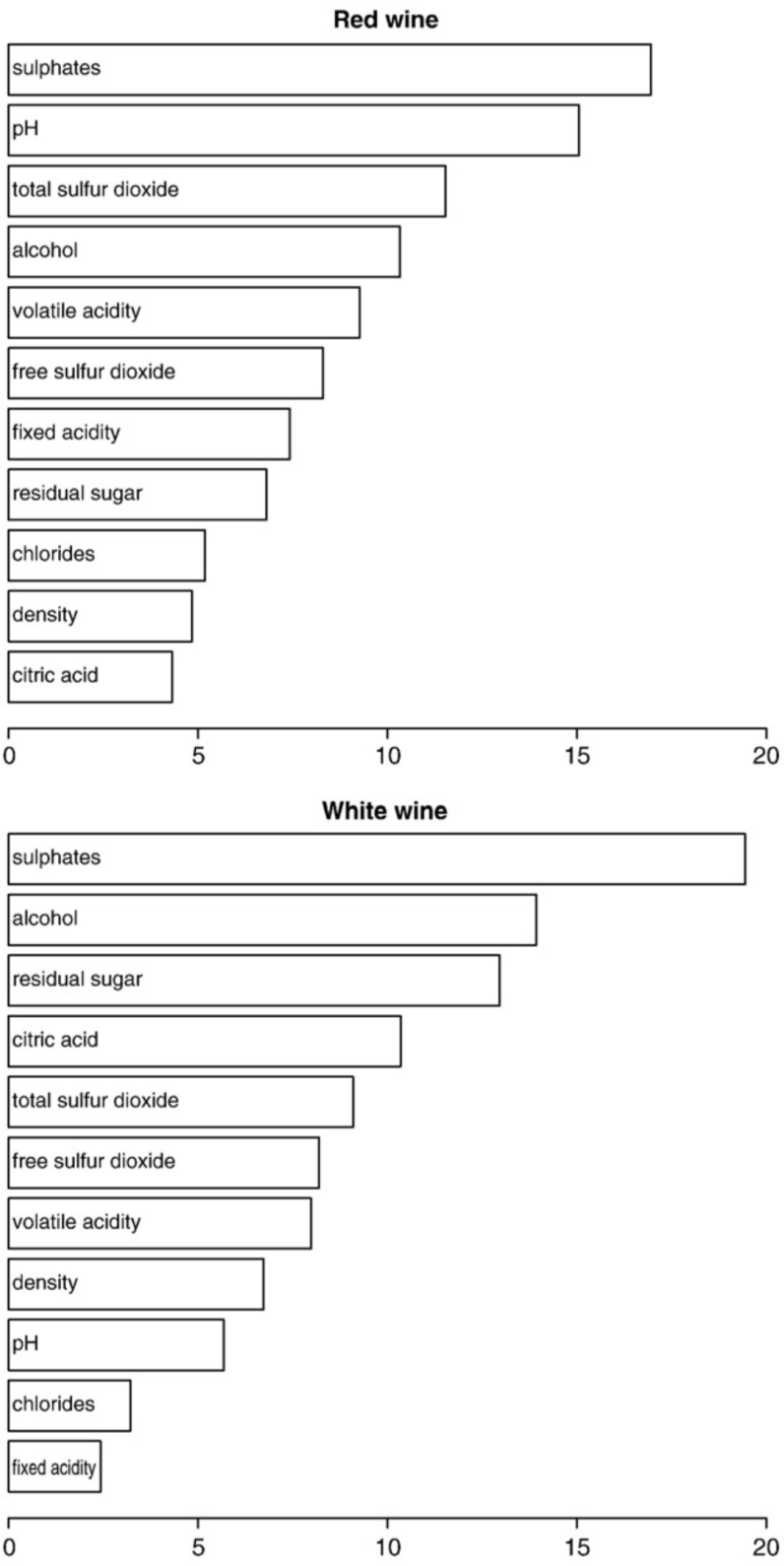


Figure 4.9: The red and white wine input importances [1]

and "volatile acidity". Alternatively red wine quality is highly correlated to only four attributes, "alcohol", "sulphates", "total sulfur dioxide", and "volatile acidity". Our findings are consistent with current literature, showing more sensitivity among white wine physiochemical features when it comes to white wine quality.

4.3.3 Explanation of the Function *gpslope* in Detail

We use Stan to model the Gaussian process. In terms of choosing kernel function, we adopted the most commonly used exponential quadratic kernel function. We get the posterior distribution of the hyperparameters by updating a user-specified percentage of the data points as the training data, and predict on the whole data set as the test data. For visualization purposes, we deliberately extend the test data to include a sequence of points along the "x" axis, such that the results could be plotted continuously.

The Stan file specifies the Gaussian Process modeling as explained in Section 4.2.1. Full Stan model specifications are available in Appendix C. The hyperparameters $\Theta = \{\alpha, \rho, \sigma\}$ control the kernel function and therefore the covariance matrix of a Gaussian process. Firstly, we need to assign priors to the hyperparameters, which should be defined based on prior knowledge of the scale of the output values (α), the scale of the output noise (σ), and the scale at which distances are measured among inputs (ρ). The hyperparameters α and ρ are weakly identified [110]. As suggested by the Stan manual, we put soft constraints on the length-scale parameter ρ that allow the prior to represent both high-frequency and low-frequency functions. An inverse gamma prior works well since it has a sharp left tail that puts negligible mass on infinitesimal length-scales, but a generous right tail for large length-scale values. Additionally, an inverse gamma distribution is a boundary-avoiding distribution. It has zero density at zero and therefore pushes away the posterior of the length-scale parameter ρ from zero. We do penalize infinitesimal ρ values because they result in high-frequency realizations and easily step into the problem of over-fitting.

The parameter α explains the variation and acts similarly to the prior for the variance of linear models. Therefore we can use the same prior for α as in linear models, such as a half-t or half-Gaussian prior. The benefit of using a half-t or half-Gaussian prior on α is that we put nontrivial prior mass around zero, which allows the Gaussian process to support zero functions and allows the possibility that the Gaussian process won't contribute to the conditional mean of the total outputs [111].

The prior distributions for the hyperparameters are set by suggestions from the Stan manual [111]. User can specify the prior distribution parameters for the hyperparameters to reflect their domain knowledge, such as how wide or concentrated they believe the hyperparameters are. The default values are set to:

$$\begin{aligned} \alpha &\sim \mathcal{N}(0,1) \\ \rho &\sim Inv\text{-}Gamma(4,4) \\ \sigma &\sim \mathcal{N}(0,1) \end{aligned} \tag{4.21}$$

The posterior distribution of $\frac{\partial f^*}{\partial x^*}$ does not have an analytical solution, but we have proved that $\frac{\partial f^*}{\partial x^*}|\Theta$ has distributional properties as shown in Equation 4.20. Therefore, we use Stan to draw samples of $\frac{\partial f^*}{\partial x^*}$ in the warm-up period. The empirical samples represent the posterior distribution of $\frac{\partial f^*}{\partial x^*}$. Then we plot the confidence interval derived from the posterior distribution of $\frac{\partial f^*}{\partial x^*}$.

Based on Equation 4.8 and 4.9, we can get the posterior distribution of $K_* K^{-1} Y$ and $\frac{\partial K_* K^{-1} Y}{\partial x_*}$. We plot the mean value of $K_* K^{-1} Y$ and $\frac{\partial K_* K^{-1} Y}{\partial x_*}$ and their user-defined uncertainty interval from the posterior distributions.

In regards to the uncertainty interval, there are a lot of options to choose from [112]: such as, a 95% interval; a half-width of the 68% interval as a quantile-based standard error; a central 50% median-based interval, etc. Many people will be inclined to read the 95% confidence intervals, because the p-value is conventionally expected at 0.05. But it is rather unstable under the Bayesian setting because each end only relies on 2.5% of the draws. Therefore, we suggest using a confidence level smaller than 95% for more computational stability and trying to convey an unrealistic near-certainty

estimate. Any single interval might not be descriptive enough and we decide to give users the option to specify the confidence level for interval extraction.

4.4 Summary

The R package *gpviz* provides a visualization tool to display the distributional properties of the Gaussian process prediction's slopes, including uncertainty interval estimation at a user specified confidence level and distribution probability density estimations.

We describe the *gpslope* and *gpapply* functions of the package and implement the numeric estimation of the Gaussian process prediction's slope distribution via rStan. The *gpslope* function visualizes the distribution of the Gaussian process predictions and the distribution of the predictions' slopes. The region of interest defined by the user specified conditions are highlighted if there is any. And the function *gpapply* loops the function *gpslope* over a list of independent variables to examine the bivariate relationships one by one.

We take the first derivative of the mean of the prediction $K_* K^{-1} Y$. Then we can get a changing ratio of the mean and the uncertainty band of the changing ratio. Therefore, we can infer a relatively certain region of interest where the dependent variable changes as expected.

One of the main barriers to the further uptake of the Gaussian process lies in the computational cost arising from matrix operations. Many researchers have made valuable contributions to improve the speed of Gaussian process regression [113], but this still remains a hurdle for the Gaussian process to cross, especially in applications with large datasets. Due to this constraint, the package *gpviz* currently works better with small data sets and could only take data up to the limit that the Gaussian process can handle.

We would like to expand the independent variable from one dimension to multiple dimensions. The challenges mainly consist of: 1) defining the region of interest in high-dimensional space, and 2) visualizing high-dimensional space. The first challenge

requires a rigorous definition of the region of interest in order to choose the calculation and derivation of the derivative(s) We can possibly use partial derivatives in regards to each dimension or use the gradient of the Gaussian process realization surfaces. The second challenge remains hard to solve. One alternative might be plotting some cross-sections.

Chapter 5

PREDICTING SURGICAL ADVERSE EVENTS WITH A HYBRID NEURAL NETWORK MODEL

5.1 Introduction

Adverse events are defined as unintended injuries caused by medical management that prolong hospitalization and/or produce a disability at the time of discharge [114]. Adverse events during hospitalization affect nearly one out of ten patients [115], leading to injuries, deaths, and massive amounts of extra medical expenditures each year. Empirical studies have examined the feasibility of detecting adverse events through record reviews and have shown that 36.9% of hospitalized patients' adverse events are judged to be preventable [116]. Additionally, if one expands the definition of adverse events to include both minor and moderate severity, about half of the adverse events are preventable under current standards of care [117]. Therefore, accurately predicting adverse events is of great value to both practitioners and patients, as a substantial amount of adverse events may be prevented proactively.

Among the existing researches on the prediction of adverse events, most studies focus on adverse drug event detection and prediction. Some researchers generalize the prediction to all categories of adverse events (see 5.2 for more details). Meanwhile, there are systematic review studies showing that in-hospital adverse events were most frequently operation-related and that a substantial portion of these events are preventable [115, 118]. It is a well-acknowledged fact that interventions aimed at preventing these operation-related adverse events have the potential to make a substantial difference. However, literature focusing on surgical adverse event prediction is very limited.

Surgical adverse events are defined as harmful events related to preoperative care, surgery, or postoperative care [119]. Studies have confirmed that surgical admissions have high risks for adverse events [120]. However, there are two main challenges imbedded in surgical events' data structures: 1) mixed types of data, consisting of both sequential surgical procedure codes and static patient admission information (such as patients' demographics and commorbidities); 2) the procedure codes come with an inherent hierarchical structure. To address these challenges, we need technical innovations to predict surgical adverse events accurately, rather than using the existing methods that are typically applied in adverse drug events and/or other types of adverse events.

We propose a novel method that utilizes both hospitalized patients' admission data and their surgical procedures to predict their probabilities of having surgical adverse events. The novelty of our method lies in three aspects: 1) we design a representation of medical code to address its hierarchical structure; 2) in order to handle both sequential and static data, we design a hybrid neural network of the dynamic recurrent neural network and the multilayer perceptron neural network; 3) we are among the first to apply a recurrent neural network to predict surgical adverse events and make significant achievements in accurately estimating the probability of surgical adverse events.

In order to prevent surgical adverse events from happening in practice, we only need patients' admission data, which is readily accessible thanks to the Electronic Health Record (EHR) system, as well as a full surgery procedure plan specified under the International Classification of Diseases, 9th Revision, Clinical Modification (ICD-9-CM) coding guidelines. Highly risky cases predicted by our model could be examined more throughly. Physicians can rearrange their surgical plans if necessary, and then check if the prediction of their patient's surgical adverse event risk could be lowered by rearranging their surgery plan. This would be a very useful tool for practitioners to evaluate patients' risk of experiencing surgical adverse events and can potentially save lives as well as millions of dollars in medical expenses.

5.2 Related Background

5.2.1 Surgical Adverse Events

Current literature about adverse events and surgical adverse events are mostly retrospective empirical researches. Many of them focus on identifying relevant or causal factors for surgical adverse events via examining differences among different patient demographic groups, different types of surgical operations, and subgroups of surgeons. Brennan et al. [114] found that the rate of adverse events increased strongly with the increasing of age. Persons 65 or older had more than double the risk of persons 16 to 44 years of age, which may due to the fact that older people are likely to have more complicated illnesses and often require more complicated interventions. Meanwhile, they found no significant difference between sexes in rates of adverse events. Their observations concerning rates of adverse events and negligence among specialties have implications relevant to the system of malpractice insurance. Specialists in neurosurgery, cardiac and thoracic surgery, and vascular surgery had higher rates of adverse events, but not higher rates of negligence. Their data suggests that the variation among specialists in rates of litigation do not reflect different levels of competence, but rather differences in the kinds of patients and diseases for which the specialist treats. Physicians' estimates of disability were another potential source of error. Gawande et al. [121] conducted a retrospective chart review of 15,000 randomly selected admissions to Colorado and Utah hospitals during 1992 to identify and analyze surgical adverse events. Based on a two-stage record-review process, they estimated the incidence, morbidity, and preventability of surgical adverse events. They characterized their distribution by types of injury, physician specialties and determined incidence rates by procedures. They identified 12 common operations with significantly elevated adverse event incidence rates (ranging from 4.4% to 18.9%); 8 operations that also carried a significantly higher risk of a preventable adverse event including lower extremity bypass graft, abdominal aortic aneurysm repair, colon the prostate or a bladder tumor, cholecystectomy, hysterectomy, and appendectomy; technique-related complications,

wound infections, and postoperative bleeding produced nearly half of all surgical adverse events; Anderson et al. [118] wrote a systematic review by reviewing articles retrieved from systematic searches in major medical databases and estimated that approximately 1 out of 20 surgical patients experience preventable adverse events, most of which were related to periprocedural care and not errors of surgical technique.

Efforts have been made on improving patients' medical records to reveal more information for physician's diagnosis and procedure plans. The patient safety indicator (PSI) [122] is a set of measures used in hospital patients' discharge data to provide perspectives on patients' safety. In particular, the patient safety indicator screens for adverse events that patients may experience as a result of exposure to the health care system. These events are likely amendable to prevention by changes at the system or provider level. Romano et al. [123] examined the criterion validity of the Agency for Health Care Research and Quality (AHRQ) Patient Safety Indicators using clinical data from the Veterans Health Administration (VA) National Surgical Quality Improvement Program (NSQIP). Sensitivities were 19%-56% for original PSI definitions; and 37%-63% using alternative PSI definitions. Positive likelihood ratios were 65-524 using original definitions, and 64-744 using alternative definitions. "Postoperative respiratory failure" and "postoperative wound dehiscence" exhibited significant increases insensitivity after modifying few classes and individual ICD-9 codes performed well enough to be used to flag surgery AEs.

Some qualitative studies have identified human error factors that cause adverse events. Two types of human errors are classified: one occurs when the plan is adequate but the associated actions do not go as intended, such as slips and lapses; one occurs when actions may go entirely as planned but the plan is inadequate to achieve its intended outcome, which is termed as a mistake [124].

5.2.2 Predictive Analysis of Surgical Adverse Events

Existing predictive analyses of surgical adverse events are very limited. Some researches are relevant but their methods are rather rudimentary and/or heavily labor

intensive, such as frequency comparison, sensitivity and preventability estimation, and two-stage systematic reviews. Liu and Leung [125] compared the relative importance of intraoperative versus preoperative factors for predicting adverse postoperative outcomes in elderly geriatric patients. They found that the most prevalent preoperative risk factors were a history of hypertension and coronary artery, pulmonary, and neurologic diseases. Zhan [126] used ICD-9-CM codes and patient medical record abstraction to identify postoperative deep vein thrombosis and pulmonary embolism (DVT/PE) events, which are common complications after surgery and associated with substantial excess mortality and length of stay. The result of sensitivity estimates were 67% (72/108 cases) for DVT, 74% (23/31) for PE, and 68% (90/133) for DVT/PE combined. Bhise et al. [127] used electronic health record data repository of a health system and refined the methods of the Institute of Healthcare Improvement's Global Trigger Tool review process by two physicians independently reviewing eligible "e-trigger" positive records to identify preventable diagnostic and care management adverse events. We can see a gap between existing methods and ideally an automatable, accurate and efficient predictive model. Current predictive analyses of surgical adverse event involve much human intelligence rather than artificial intelligence, and are mostly reactive rather than proactive.

Comparatively, there are much more predictive models established for predicting and/or detecting adverse drug events: 1) the main driving force behind most computerized pharmacovigilance methods is "dis-proportionality analysis" (DPA) and its extensions. DPA methods use frequency analysis of contingency tables to quantify statistical associations between specific drug-event combinations mentioned in spontaneous reports [128]; 2) another category is multivariate logistic regression based approaches, which is appropriate to handle confounding by controlling or adjusting for the presence of other covariates when examining a drug-event association [129, 130]; 3) a rising popular category of methods are unsupervised machine-learning approaches, such as association rule mining [131, 132], probabilistic topic modeling [133], clustering

[134, 135, 136] and network analysis [137, 138]. These methods are not directly applicable to predictions of surgical adverse events due to the differences in data structures. Adverse drug event detection methods heavily rely on data in the form of drug-reaction reports, including reports from the Federal Drug Administration, spontaneous reports and social media data. On the contrary, surgical adverse events are reported by hospitals and the data available to us only contains patients' electronic health record information such as admission data and multiple procedure codes with time stamps.

5.2.3 Sequential Pattern Mining

One of the major differences between the surgical adverse events data and other adverse events data is that surgical operations are usually presented in a sequence of procedure codes.

Sequential pattern mining, originally proposed by Rakesh Agrawal in 1995 [139], often goes hand in hand with sequential data, so as in the adverse event studies: Reps et al. applied sequential rule mining on patients age, gender and medical history to predict illness based on UK Health Improvement Network general practice database. Their key result is that sequential rules present the possibility of determining the likelihood of re-infections [140]; Malhotra et al. examined the use of electronic healthcare reimbursement claims for analyzing healthcare delivery and practice patterns across the US. The discovered clinical procedure sequences reveal significant differences in the overall costs incurred across different parts of the US, indicating a lack of consensus amongst practitioners in treating ASD patients [141]; Wright et. al. [142] focused on patients who prescribed diabetes medication and used sequential pattern mining to predict the patient's next pharmacological therapy, including drug class and generic drug level. They achieved an accuracy of 90.0% in drug class prediction and 64.1% in predicting a more detailed drug level. They found that using one or two items in the patients medication history led to more accurate predictions than not using any history or using the entire history; Lin et. al. [132] proposed unexpected temporal association rules to describe unanticipated episodes where certain event patterns are

infrequent but may lead to adverse drug events since the traditional sequential pattern mining methods cannot detect the nuances of infrequent patterns; Graphical statistical approach is used to summarizing and visualizing the temporal association between the prescription of a drug and the occurrence of a medical event [143].

In this chapter, we propose a novel method to make full use of the sequential surgical procedures instead of using sequential pattern mining. The main reason why we don't use sequential patter mining is that we got mixed types of data to deal with, both static data and sequential data, using sequential pattern mining to handle the sequential data leaves us no way but to analyze static data independently and to combine the static data analysis with the sequential data analysis somehow. Therefore, the combination of the two pieces is an ad-hoc practice rather than a theoretically justified method.

5.2.4 Neural network and deep learning

In 1943, McCulloch and Pitts [144] proposed the concept of neural networks as mathematical programming models. They formed mathematical description of the neuron and the network structure built on neurons. The technology has witnessed a few waves of popularity and criticism until recent decade when Hinton [145, 146] proposed to increase the number of layers and form deep artificial neural network. The deep neural network structure has obtained many achievements in image recognition and natural language processing, which aroused a great interest in deep learning from both academia and industry.

Recurrent neural network has shone lights on sequential data processing such as text and speech [147]. Bate et al. [148] used Bayesian confidence propagation neural network for adverse reaction signal detection by using Bayesian statistics implemented in a neural network architecture to analyze drug-reaction relationships. Recurrent neural network is used to find the complex dependencies, e.g. isolating patterns for rofecoxib and celecoxib [149]. Huynh et al. [150] investigated different neural network architectures for adverse drug reactions classification and proposed two neural network

models: 1) convolutional recurrent neural network by concatenating convolutiona neural networks with recurrent neural networks, and 2) convolutional neural network with attention by adding attention weights to convolutional neural networks. Their experiments, which were evaluated on a Twitter dataset containing informal language and an adverse drug effects dataset constructured by medical case reports, showed that all neural network architectures outperform traditional maximum entropy classifiers.

In summary, our literature review reveals that most existing methods on adverse events prediction focus on predicting adverse drug events but neglect surgical adverse events which also influence a substantial amount of hospitalized patients. Due to the unique data structure embedded in surgical procedures, existing methods in detecting and/or predicting adverse drug events cannot be directly applied to predict surgical adverse events. Therefore, we have proposed a novel method to accommodate the unique data structure that surgical procedures impose. We designed a hybrid neural network, consisting of the multilayer perceptrons to handle static admission data and the dynamic recurrent neural network to handle sequential surgical procedures. Not only the mixed data is handled, but also the hierarchical structure of medical procedure codes is taken care of to enhance the learning.

5.3 Problem Formulation

5.3.1 Multilayer Perceptron (MLP)

One of the best known neural networks is the multilayer feed-forward neural network, also called multilayer perceptron model with perceptrons referring to the artificial neurons nested in the network. We use a multilayer perceptron to model the static information collected from patients when admitted. The first layer is an input layer, consisting of a set of numerical inputs (x_i) with categorical variables one-hot encoded. The inputs are multiplied by weight matrices and processed by the hidden units via activation function through each hidden layer. The output layer contains results from the last hidden layer. The perceptron structure is described as in Figure 5.1, a multilayer perceptron (MLP) model stacks multiple layers of perceptrons

and is described in Figure 5.5. In regards to activation function choices, we use Leaky ReLU function for each neuron in our MLP model. As compared to the rectified linear unit (ReLU) function, which becomes very popular in the last few years due to its quick convergence of stochastic gradient descent, Leaky ReLU allows a small positive gradient when the unit is not active and thus prevent neurons from "dying" (never being activated again once hitting zero point) in the network. In our MLP model, we use $\alpha = 0.2$, which is the default setting in TensorFlow.

$$ReLU : f(x) = max(0, x) \tag{5.1}$$

$$Leaky\ ReLU : f(x) = \begin{cases} x, & x > 0 \\ \alpha x, & x <= 0 \end{cases} \tag{5.2}$$

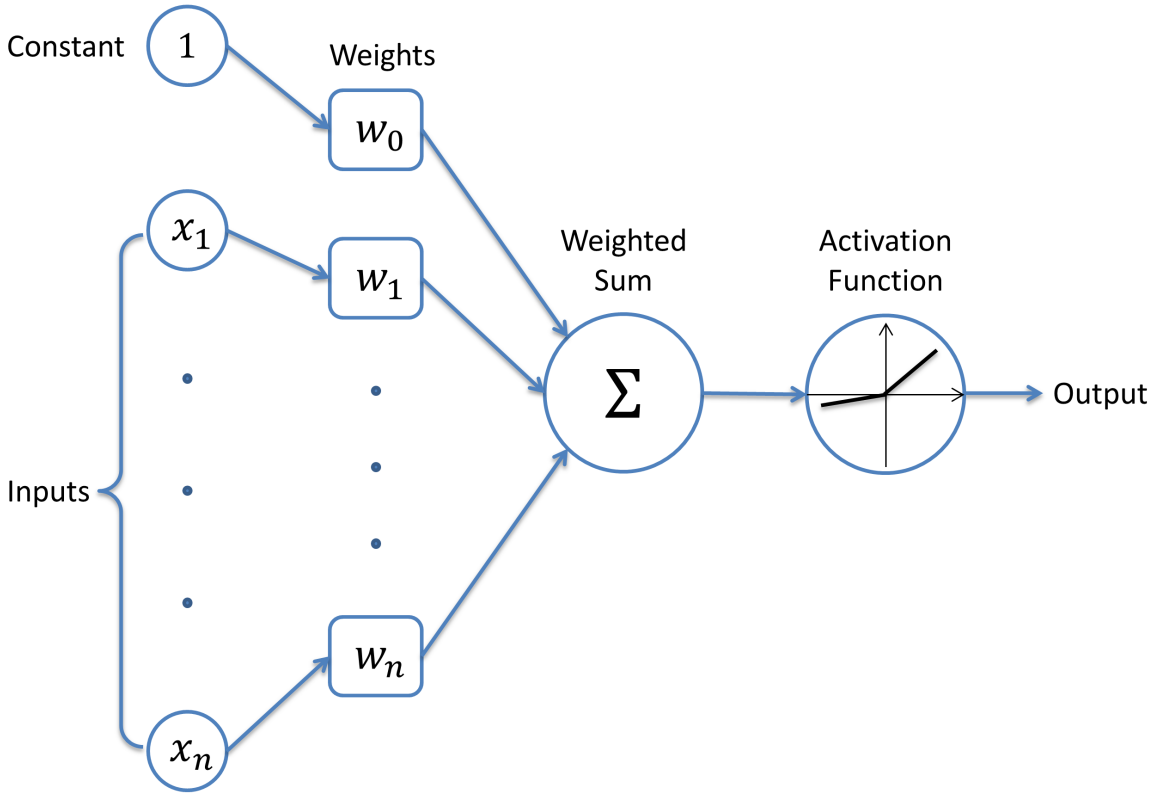


Figure 5.1: Perceptron structure

5.3.2 A Recurrent Neural Network: Long Short-Term Memory (LSTM)

The dynamic recurrent neural network model we use is long short-term memory (LSTM) model [151], which is one of the most famous and popular models in recurrent neural networks. We describe a single LSTM unit as in Figure 5.2. The long-term memory unit c changes slowly, and c^t is c^{t-1} added by something; while the short-term memory unit h changes faster and h^t can be very different from h^{t-1}.

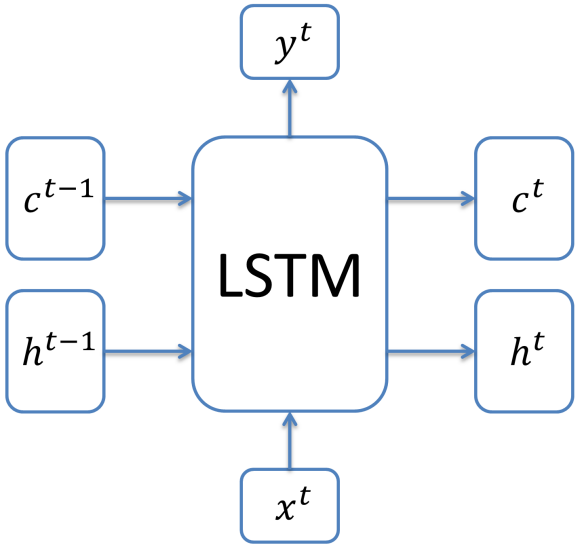


Figure 5.2: Long-short memory (LSTM) unit structure - summary

The input layer includes a sequence of x ordered by time stamp t. In our context, x is the surgical ICD-9 code. Since the ICD-9 code has an inherent hierarchy when coded. We address its hierarchical structure by expanding an ICD-9 code to a vector. Each element of such a vector contains a number that represents the category/level of the ICD-9 code in the coding hierarchy structure. For example, an ICD-9 code "01.02" (incision and excision of skull, brain, and cerebral meninges; cisternal pun) is reshaped to [1, 01, 01.0, 01.02]. The first element "1" corresponds to the highest level of the coding structure (operations on the nervous system); the second element "01" corresponds to the second level of the coding structure (incision and excision of skull, brain, and cerebral meninges); "01.0" corresponds to the third level of the coding

structure (cranial puncture); and "01.02" corresponds to the fourth level of the coding structure (ventriculopuncture through previously implanted catheter).

For a LSTM unit corresponding to a time stamp t, the rapidly changed hidden state from last time stamp h^{t-1} is concatenated with input x^t to compute z after activation function (often use hyperbolic tangent function) and z^i, z^f, z^o after sigmoid function, these four computations are processed in parallel via TensorFlow framework:

$$z = tanh\Big(W \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\Big) \tag{5.3}$$

$$z^i = tanh\Big(W^i \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\Big) \tag{5.4}$$

$$z^f = tanh\Big(W^f \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\Big) \tag{5.5}$$

$$z^o = tanh\Big(W^o \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}\Big) \tag{5.6}$$

After the four vectors (z, z^i, z^f, z^o) are computed, c^t, h^t, y^t are computed sequentially:

$$c^t = z^f \odot c^{t-1} + z^i \odot z \tag{5.7}$$

$$h^t = z^o \odot tanh(c^t) \tag{5.8}$$

$$y^t = \sigma(W' h^t) \tag{5.9}$$

The structure is described as in Figure 5.3, with dash arrows indicating operation with weight matrices and regular arrows connecting basic math operations such as element-wise multiplication $\odot$.

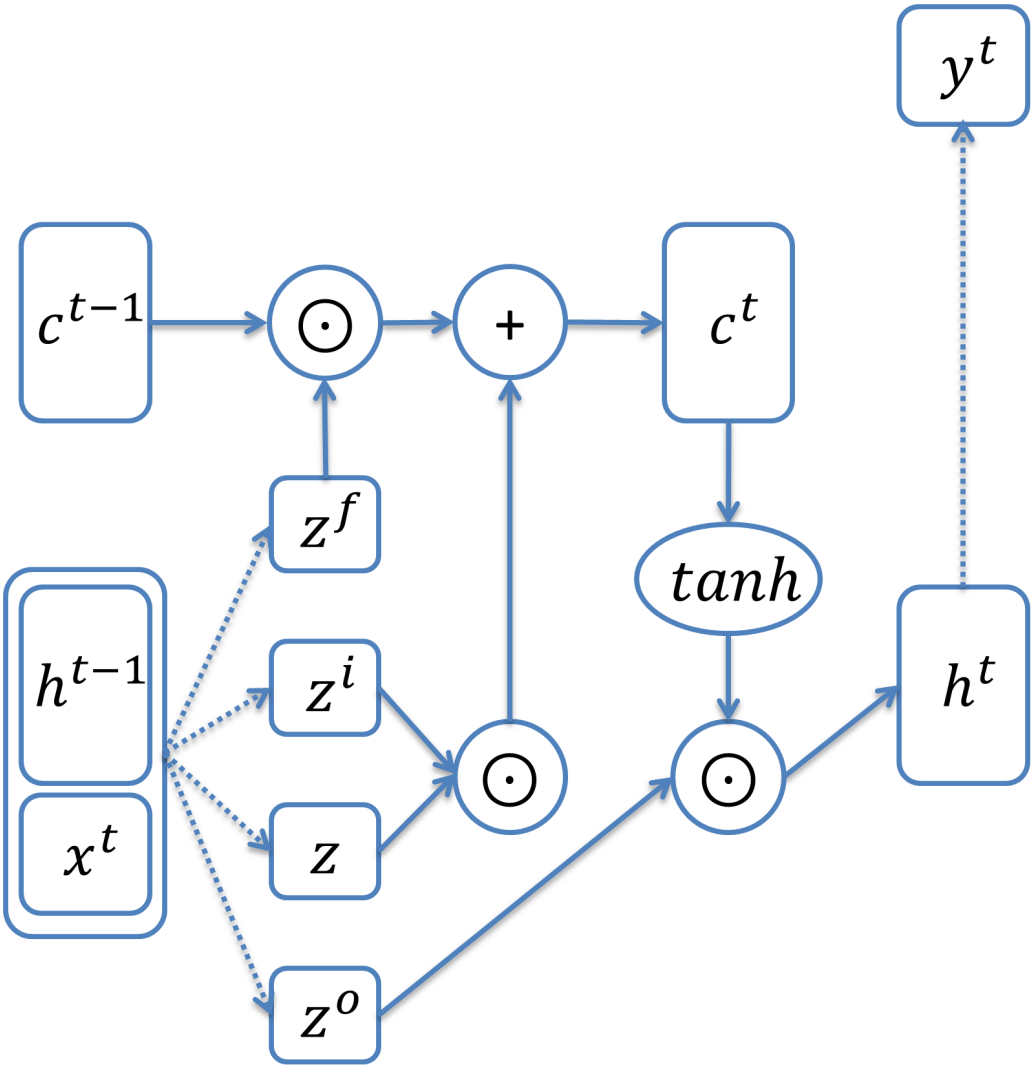


Figure 5.3: Long-short memory (LSTM) unit structure - details

When one unit LSTM cell finishes computing, c^t, h^t are passed as a tuple to the next LSTM cell together with sequantial input x^{t+1} as training inputs for the next LSTM cell. The sequential structure is described in Figure 5.4.

5.3.3 A Hybrid Model of Multilayer Perceptron and Dynamic Recurrent Neural Network

Appropriate neural network architecture can be trained to predict dependent variables with great predictive power [152]. We propose a hybrid model of multi-layer perceptron and LSTM as described in Figure 5.5. We use three hidden layers in MLP and three stacked layers of LSTM. We decide to utilize a dynamic LSTM because the length of surgical procedures each patient receives is different. Therefore, we structure different surgical sequences to the same length $batch\_max\_len$ by padding zeros so that LSTM can handle the batch of inputs, but we also keep track of the effective surgical length n_e to extract the outputs corresponding to the effective length of surgery.

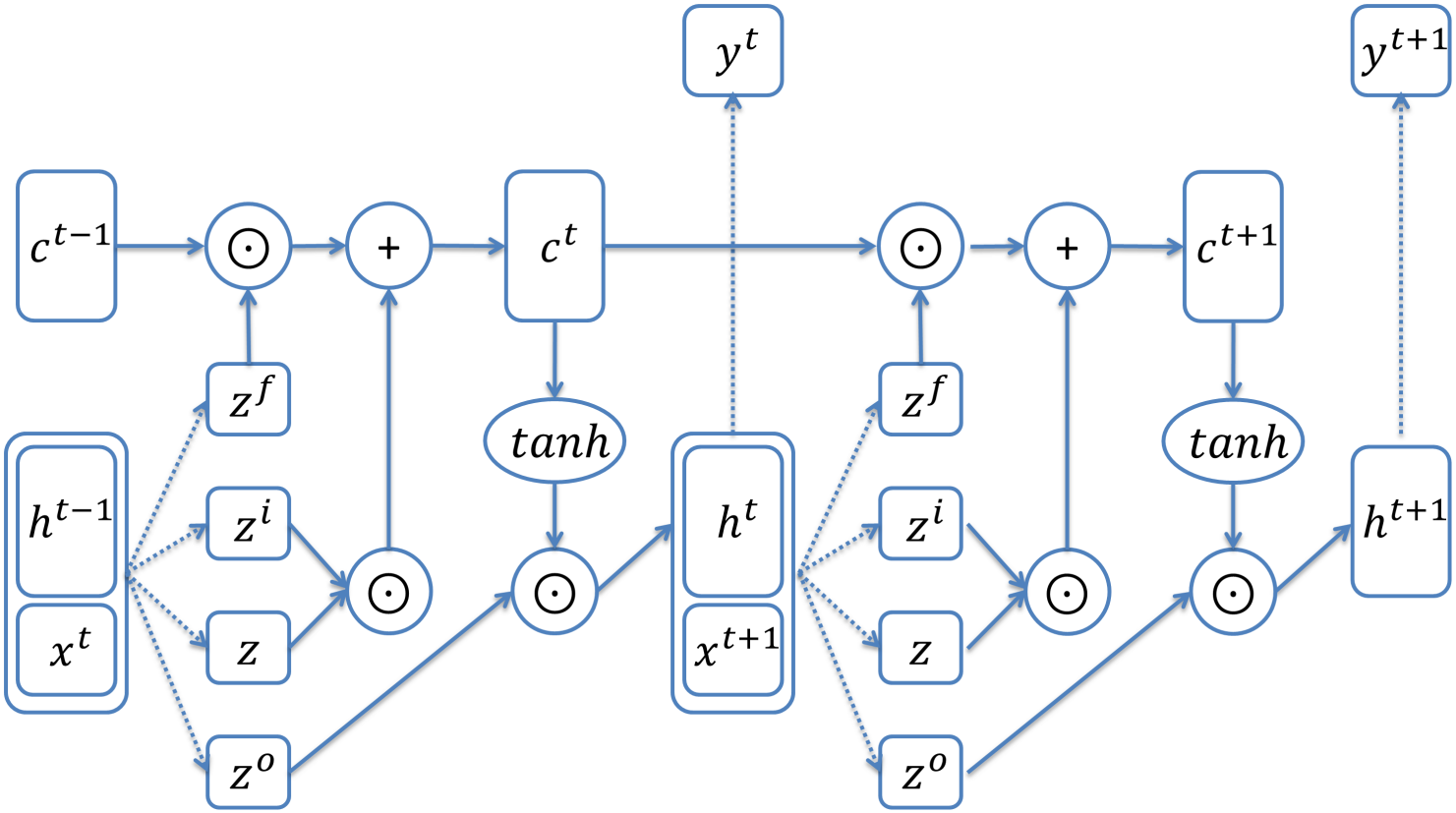


Figure 5.4: Long short-term memory (LSTM) sequential structure

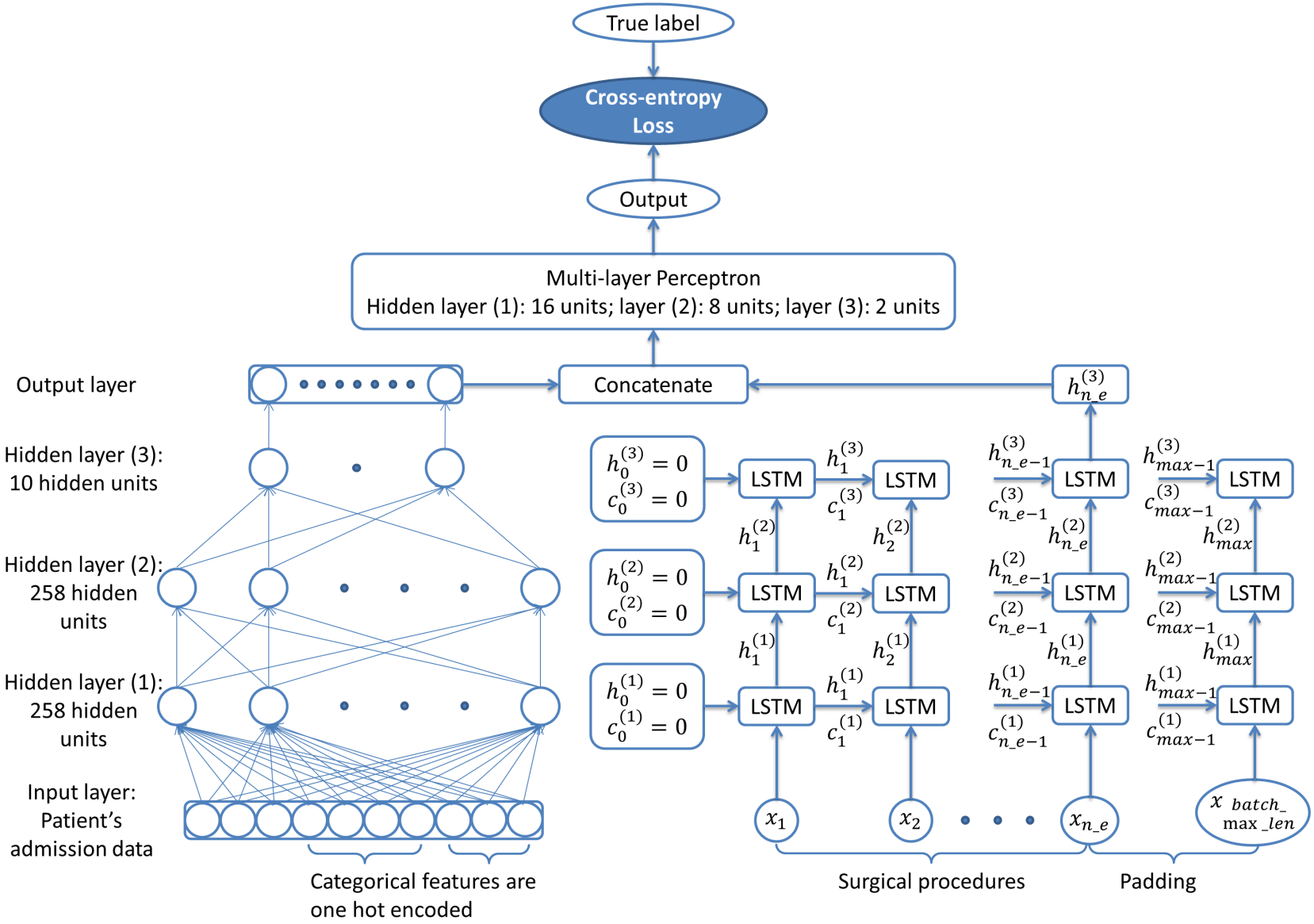


Figure 5.5: Hybrid model of MLP and dynamic LSTM

5.4 Empirical Evaluation

5.4.1 Data Collection

The data used in our experiment is collected by Dr. Yukai Lin and shared with us in 2016. The data was originally obtained from the Florida Agency for Health Care Administration (AHCA), to which hospitals in Florida are required to submit patient discharge information for accountability and transparency in health care services. The dataset contains information about the patient, attending physician, and health care facilities in Florida between 2010 and 2014. The dataset also contains patient-level medical diagnoses and procedures for each hospital stay, which are all encoded using the ICD-9-CM standard. In order to protect patient's privacy, all the records are anonymous, therefore, each record stands for an individual hospital stay and we don't have any information about patient's past history of in-hospitalization experiences.

In order to evaluate the predictive power of our hybrid model, we firstly filtered 6,085,794 hospitalized patients who went through surgical procedures rather than receiving only medicine treatment from a total of 10,531,828 patient records. Then we made a few more necessary data preprocessing as described in Section 5.4.2 and 5.4.3.

5.4.2 Admission Data

As discussed in Section 5.2, we include many patient demographics from admission data to feed as the input layer of multi-layer perceptrons, such as age, gender and race. We also include time, quarter and year of the admission, indicator of admitted on a weekday or weekend, indicator for rural residents or not, estimated distance in miles for the patient to hospital, medicare severity diagnosis related group type, and payer. We also include some risk measures, including seventeen indicators for different types of Charlson Comorbidity [153], Elixhauser Comorbidity [154] and vw score [155] (a weighted summary score based on the 30 comorbidities from the Elixhauser Comorbidity Index).

Since the data is collected as discharge data, we need to remove the information obtained from and after the surgery for our model to learn and predict the surgical

adverse event before occurrence. The removed information includes discharge status, length of days stayed in the hospital, and charges. Outcomes are kept as true labels, with only surgical adverse events marked as positive while everything else marked as negative.

5.4.3 Surgical Procedure Data

Our implicit assumption is that the sequence of surgical procedures is related to the surgical outcome. Since we want to use the sequential information to predict the probability of surgical adverse events, we will not have any revision procedures available before surgeries or scheduled in the surgical procedure plan. Hence we would like to eliminate the surgical procedures done after surgical adverse events happen.

We analyzed the co-occurrence of all the procedure codes with surgical adverse event outcomes. 31 out of 3594 unique ICD-9 procedure codes in our sample have 100% co-occurrence with surgical adverse event outcome and 0 standard deviation. Such codes include other revisions of vascular procedure, reopening of recent laparotomy site, control of hemorrhage (not otherwise specified), reopening of recent thoracotomy site, suture of laceration of bladder, suture of laceration of large intestine, reopening of laminectomy site and so on. These procedures apparently are used to fix issues brought by surgical adverse events. Therefore, these codes are not supposed to exist in the surgical plan.

Noticeably, there are 15 procedure codes having 100% co-occurrence with surgical AE but only occurred once in the entire sample. To avoid coincidence, we have examined each one of the 15 procedure codes and keep 4 out of 15 codes in the data set due to the fact that they do not indicate a direct link with surgical adverse events. They are electrocochleography, percutaneous hysterogram, pelvic gas contrast radiology, and lymphangiogram of lower limb.

What's more, our data contains six patient procedure ICD-9 codes that are deleted, replaced, or expanded. Our data set contains three of the six codes: code 48.5 "ABDPERNEAL RES RECTM NOS" (194 cases), code 85.7 "Total reconstruction

of breast" (24 cases), and code 39.8 "Operations on carotid body and other vascular bodies" (20 cases).

1. Code 48.5 is expanded to 48.50, 48.51, 48.52, and 48.59. The 194 (310) cases with code 48.5 are mapped to 48.50, exact matches to "Abdominoperineal resection of the rectum, NOS".

2. Code 85.7 is expanded to 85.70 85.79. The (161) cases with code 85.7 are mapped to 85.70, exact matches to "Total reconstruction of breast, NOS".

3. Code 39.8 is expanded to 39.81 39.89. The 20 (142) cases with code 39.8 are mapped to 39.89, exact matches to "Other operations on carotid body, carotid sinus and other vascular bodies". [1]

After removing the post surgical AE procedures, we end up with 6,083,475 unique hospital records. Then we split 90% as training data and 10% as test data. Since our data is heavily unbalanced, with less than 4% positive labels across the whole sample, we use stratification based on positive labels when splitting the training and test records to make sure the label distributions in training and test sets are similar and comparable.

5.4.4 Results

Since the data we dealt with is very unbalanced, purely measuring prediction accuracy is no longer a useful indication of the classifier's predictive power, e.g. simply predicting all negatives can result in accuracy over 96%. To measure the predictive accuracy of our model, we used the area under the receiver operating characteristic curve (AUC) as many prior predictive analytics studies did [156] to measure the predictive power of our hybrid neural network model. ROC is a graphical plot to illustrate the diagnostic ability of a binary classifier as its discrimination threshold changes. The

[1] All these number of cases are reported based on their principle procedure code. The numbers in bracket are reported based on all procedure codes.

horizontal axis of an ROC curve is the false positive rate while the vertical axis of an ROC curve is the true positive rate. An ideal ROC curve bends towards the left-upper corner as the classifier gets very precise when predicting positives with high probability estimations. Therefore, the area under the ROC curve (AUC) represents the performance of the classifier. Since the AUC value is proportion of area, it ranges from 0 to 1. A higher AUC value means that the model can better separate positive and negative cases, and an AUC score lower than 0.5 indicates that the classifier does even worse than random guesses.

In our context, a higher AUC means that our model can better distinguish among hospitalized patients to predict the highly risky ones who may experience surgical adverse events based on merely their admission data and surgical plans. The ROC curve of our hybrid neural network model performed on test set is reported in Figure 5.6. Our AUC score obtained on test set is 0.83. The data generating process and training-test splits are randomly shuffled and trained a handful of times. We have seen that the AUC scores obtained on the test sets are quite stable at about $0.83-0.84$ each time.

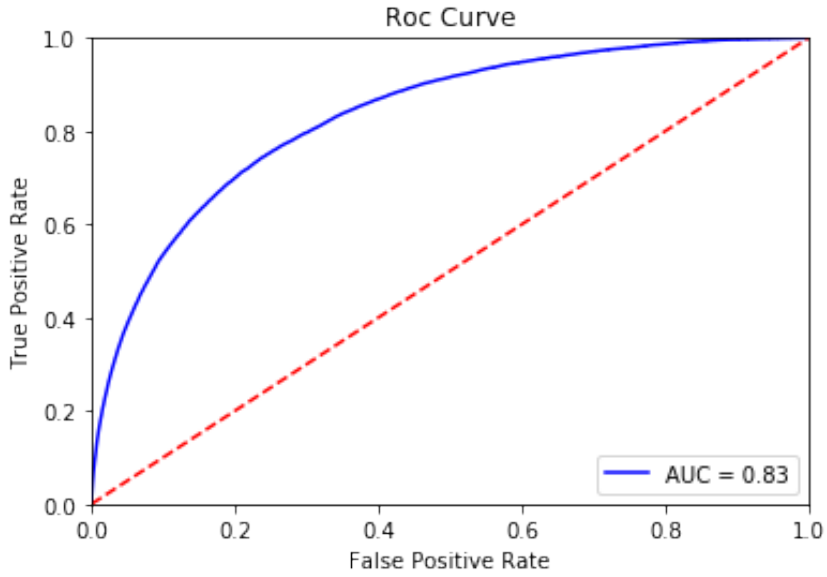


Figure 5.6: ROC curve and AUC Performance on test data

Our model has a very good discriminatory power when compared with relevant literature results, see Table 5.1. Our AUC result 0.83 stands among the best when compared with existing methods' results.

Table 5.1: Comparison of AUC Performance on Surgical Adverse Event Prediction

Study	Main Data Source	Data Collection	Methods	Number of Features	Evaluation Design	AUC
Genovese et al. (2017) [157]	Clinical registry	52,562 patients	GLMM	10	15% holdout	0.818
Mortazavi et al. (2017) [158]	Patient's clinical records	5,214 patients	GBM, GLM, RF	9828	5-fold CV	0.81-0.83
Dodson et al. (2014) [159]	Clinical registry	240,632 procedures	GLM	21	30% holdout	0.72
Kusy et al. (2013) [160]	Patients' genomic expressions	107 patients	ANN, GEP	10	30% holdout	0.82
Krone et al. (2000) [161]	Clinical registry	41,071 patients	GLM	7	Not applied	0.69
Geraci et al. (1993) [162]	Patients' clinical records	2,213 patients	GLM	11	50% holdout	0.64
Rosen et al. (1992) [163]	Patients' clinical records	8,126 patients	GLM	5-10	50% holdout	0.64-0.75

[2]Note: ANN = artificial neural network, AUC = area under the curve, CV = cross-validation, GBM = generalized boosted model, GEP = gene expression programming, GLM = general linear model, GLMM = generalized linear mixed model, RF = random forest

Beyond AUC evaluations, we also conduct top k precision evaluation as reported in Table 5.2. The results is very promising: the top 1% patients from the test set predicted as highly risky ones turned out to have more than ten times higher chance to experience surgical adverse events than random guesses.

Table 5.2: Top k Precision on Test Data

Threshold of probability	k in percentage	top-k-precision
0.364	1%	0.432
0.176	5%	0.268
0.110	10%	0.196
0.021	50%	0.072
0.001	99%	0.040

5.5 Conclusion

Smart health has been a rising star in data science fields. Many endeavors have been made and many more are on the way. In 2009, Google published a flu prediction

paper in *Nature*. Ginsberg et al. [164] presented a method to analyze massive amounts of Google search queries to track influenza-like disease and estimate the current level of weekly influenza activity in each region of the U.S. Their reporting lag is approximately one day as compared to the US Centers for Disease Control and Prevention (CDC)'s typically one to two weeks. This practice has attracted much attention from health-care professionals. However, there is definitely a long way for us to go before predictive models can reliably work as health service supports. Even with Google Flu Trends (GFT), the exemplary model we value, problems exist: *Nature* reported that GFT was predicting more than double the proportion of doctor visits for influenza-like illness than the CDC [165]; GFT also missed the 2011-2012 flu season by a very large margin for 100 out of 108 weeks [166]. Lazer et al. [166] shows that GFT overlooks considerable information that can be extracted from traditional statistical methods.

We firmly believe that big data is not just about the size of the data but about using data to understand the unknown. Some of the methods such as neural networks naturally need massive amounts of labeled data to be trained into a powerful predictive model. We have the luxury of a massive labeled medical record data set, and we choose the neural network for its merits with big data sets. And we trained the neural network framework to accommodate the unique data structure of surgical procedures and the underlying problem that needs to be solved.

Our contributions are: (1) we develop a novel method for the prediction of surgical adverse events, which is an important but largely unexplored area; (2) due to the uniqueness of surgical data, including the structure (hierarchical structure of ICD-9 codes), and the mixed sequence and static information, it is more challenging to predict surgical adverse events. We propose a method to accommodate the data specialties involved in surgical data and make full use of all the information to make better predictions; (3) we are among the first to adopt a dynamic recurrent neural network to process sequential information embedded in procedure sequences and to combine the network with multi-layer perceptrons to handle static admission data. The model is proved with good performance on unseen test data; (4) this study brings

a huge potential for practical use, physicians can use the predictions to evaluate the risk of surgeries when the surgery plan is worked out, since the model is good at precisely predicting highly risky cases. This model will be very useful and much less intimidating since it won't give out too many false alarms.

We leveraged a hybrid neural network model of the multilayer perceptron (MLP) and the dynamic recurrent neural network to learn the probability of patients experiencing surgical adverse events. Our model can accommodate the mixed type of static data (such as patient's demographics and commorbidities information) and sequential data of surgical procedure codes that the patient went through. We have shown that our model generates superior predictive performance over other popular machine learning algorithms in the prediction of surgical adverse events.

5.6 Future Work

Future work involves further exploration of the neural network design, including representing the hierarchical information of surgical procedure codes from the recurrent neural network, and incorporating the time stamp of each surgical procedure into the recurrent neural network.

One piece of critical information is not yet utilized in current network design: diagnoses. We would like to include this information in our model in the near future, including present on admission (POA) indicator and primary physicians' diagnoses reflected in ICD-9 diagnosis code form. Houchens et al. [167] pointed out that POA data, which captures whether diagnoses are present on admission, distinguishes comorbidities from potential in-hospital complications. After incorporating POA information, most cases of decubitus ulcer (86%-89%), postoperative hip fracture (74%-79%), and postoperative pulmonary embolism/deep vein thrombosis (54%-58%) were no longer considered in hospital patient safety events.

Chapter 6

CONCLUDING REMARKS

In the introductory chapter of this dissertation, we have discussed the need to align data models with real-world needs. This dissertation explores such methods and proposes new data structures and methods to echo real-world interpretability, in Chapters 2 and 3 respectively. In Chapter 4, we develop toolkits to visualize and understand non-linear relationships with Gaussian process regression. In Chapter 5, we convert data representations of hierarchical codes from real-world events to make better sense of structural information for predicting adverse outcomes.

In particular, Chapter 2 proposes extensions to a uniqueness-motivated similarity, including the *bag-of-tags* similarity measure and the distance matrix similarity measure. Chapter 3 applies this uniqueness-motivated similarity to two applications: one aims to automate local competitor identification, another aims to predict airline traveler's purchase probability of ancillary services and products. In terms of validation, we use survey results from well established marketing solicitation methods as the benchmark to which we compare our algorithm results in the unsupervised learning task; and we use AUC performance to measure the accuracy of our classifier in supervised classifications. Chapter 4 derives the distributional properties of the Gaussian process predicitons' first derivatives, and we develop an R package "gpviz" as a visualization tool to display the distribution of these derivatives (slopes). Our R package provides two different versions of visualization: one is an uncertainty interval estimated at a user-specified confidence interval, while the other is a probability density estimation of the distribution. Chapter 5 proposes a novel method for surgical adverse event prediction that accommodates data specialties involved in surgical data structures. We utilize the dynamic recurrent neural network to process the sequential information of

surgical procedures, and we propose a hybrid neural network of multi-layer perceptron (MLP) and long short-term memory (LSTM) to handle mixed types of surgical data.

These topics are broadly connected under the umbrella of aligning data models with real-world needs. Going forward beyond the scope of this thesis, aligned data models are both fruitful to practitioners due to the need for tailor-made models and of interest to academics because of the novelty embedded in aligned data methods. While there is ample work to be done, we have illustrated a few specific research directions at the end of each chapter. And I will keep moving forward with the goal of aligning data models with real-world needs in my future career pursuits.

BIBLIOGRAPHY

[1] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4):547–553, 2009.

[2] Geoff Norman. Data dredging, salami-slicing, and other successful strategies to ensure rejection: Twelve tips on how to not get your paper published. *Advances in Health Sciences Education*, 19(1):1–5, Mar 2014.

[3] Jay B Barney. Resource-based theories of competitive advantage: A ten-year retrospective on the resource-based view. *Journal of Management*, 27(6):643–650, 2001.

[4] Khalid Hafeez, YanBing Zhang, and Naila Malak. Core competence for sustainable competitive advantage: A structured methodology for identifying core competence. *IEEE Transactions on Engineering Management*, 49(1):28–35, 2002.

[5] Nicolai J Foss and Thorbjørn Knudsen. The resource-based tangle: Towards a sustainable explanation of competitive advantage. *Managerial and Decision Economics*, 24(4):291–307, 2003.

[6] Ron Sanchez. Understanding competence-based management: Identifying and managing five modes of competence. *Journal of Business Research*, 57(5):518–532, 2004.

[7] David P Lepak, Ken G Smith, and M Susan Taylor. Value creation and value capture: A multilevel perspective. *Academy of Management Review*, 32(1):180–194, 2007.

[8] Kevin Lane Keller. Conceptualizing, measuring, and managing customer-based brand equity. *Journal of Marketing*, 57(1):1–22, 1993.

[9] Robert M Groves. *Survey errors and survey costs*. Wiley-Interscience, Hoboken, N.J, 2004.

[10] Bruce H Clark. Managerial identification of competitors: Accuracy and performance consequences. *Journal of Strategic Marketing*, 19(3):209–227, 2011.

[11] Hyoryung Nam and Pallassana Krishnan Kannan. The informational value of social tagging networks. *Journal of Marketing*, 78(4):21–40, 2014.

[12] Hyoryung Nam, Yogesh V Joshi, and PK Kannan. Harvesting brand information from social tags. *Journal of Marketing*, 81(4):88–108, 2017.

[13] Thomas Y Lee and Eric T Bradlow. Automated marketing research using online customer reviews. *Journal of Marketing Research*, 48(5):881–894, 2011.

[14] Sangkil Moon and Wagner A Kamakura. A picture is worth a thousand words: Translating product reviews into a product positioning map. *International Journal of Research in Marketing*, 34(1):265–285, 2017.

[15] RJ Kuo, LM Ho, and Clark M Hu. Integration of self-organizing feature map and k-means algorithm for market segmentation. *Computers & Operations Research*, 29(11):1475–1493, 2002.

[16] Jeffrey S Larson, Eric T Bradlow, and Peter S Fader. An exploratory look at supermarket shopping paths. *International Journal of Research in Marketing*, 22(4):395–414, 2005.

[17] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Clustering product features for opinion mining. In *Proceedings of the fourth ACM International Conference on Web Search and Data Mining*, pages 347–354. ACM, 2011.

[18] Daniel M Ringel and Bernd Skiera. Visualizing asymmetric competition among more than 1,000 products using big search data. *Marketing Science*, 35(3):511–534, 2016.

[19] Stephen L France and Sanjoy Ghose. An analysis and visualization methodology for identifying and testing market structure. *Marketing Science*, 35(1):182–197, 2016.

[20] Cathy A Enz. Creating a competitive advantage by building resource capability the case of outback steakhouse korea. *Cornell Hospitality Quarterly*, 49(1):73–78, 2008.

[21] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[22] Shyam Boriah, Varun Chandola, and Vipin Kumar. Similarity measures for categorical data: A comparative evaluation. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 243–254. SIAM, 2008.

[23] Amir Ahmad and Lipika Dey. A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets. *Pattern Recognition Letters*, 32(7):1062–1069, 2011.

[24] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25(5):345–366, 2000.

[25] Matteo Golfarelli and Elisa Turricchia. A characterization of hierarchical computable distance functions for data warehouse systems. *Decision Support Systems*, 62:144–157, 2014.

[26] ES Smirnov. On exact methods in systematics. *Systematic Biology*, 17(1):1–13, 1968.

[27] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[28] Benjamin Markines, Ciro Cattuto, Filippo Menczer, Dominik Benz, Andreas Hotho, and Gerd Stumme. Evaluating similarity measures for emergent semantics of social tagging. In *Proceedings of the 18th International Conference on World Wide Web*, pages 641–650. ACM, 2009.

[29] Xiao Han, Leye Wang, Noel Crespi, Soochang Park, and Ángel Cuevas. Alike people, alike interests? Inferring interest similarity in online social networks. *Decision Support Systems*, 69:92–106, 2015.

[30] David W Goodall. A new similarity index based on probability. *Biometrics*, 22(4):882–907, 1966.

[31] Song Lin and Donald E Brown. An outlier-based data association method for linking criminal incidents. *Decision Support Systems*, 41(3):604–615, 2006.

[32] Xiaochun Wang, Xia Li Wang, and D Mitch Wilkes. A minimum spanning tree-inspired clustering-based outlier detection technique. In *Industrial Conference on Data Mining*, volume 7377, pages 209–223. Springer, 2012.

[33] Lionel Ott, Linsey Pang, Fabio T Ramos, and Sanjay Chawla. On integrated clustering and outlier detection. In *Advances in Neural Information Processing Systems 27*, pages 1359–1367. Curran Associates, Inc., 2014.

[34] Cen Li and Gautam Biswas. Unsupervised learning with mixed numeric and nominal data. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):673–690, 2002.

[35] Manish Gupta, Rui Li, Zhijun Yin, and Jiawei Han. An overview of social tagging and applications. In *Social Network Data Analytics*, pages 447–497. Springer, 2011.

[36] John A Bullinaria and Joseph P Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526, 2007.

[37] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993.

[38] Ciro Cattuto, Dominik Benz, Andreas Hotho, and Gerd Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In *International Semantic Web Conference*, volume 5318, pages 615–631. Springer, 2008.

[39] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. Cactus-clustering categorical data using summaries. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 73–83. ACM, 1999.

[40] Praveen Aggarwal, Rajiv Vaidyanathan, and Alladi Venkatesh. Using lexical semantic analysis to derive online brand positions: An application to retail marketing research. *Journal of Retailing*, 85(2):145–158, 2009.

[41] Oded Netzer, Ronen Feldman, Jacob Goldenberg, and Moshe Fresko. Mine your own business: Market-structure surveillance through text mining. *Marketing Science*, 31(3):521–543, 2012.

[42] Gautam Das, Heikki Mannila, and Pirjo Ronkainen. Similarity of attributes by external probes. In *KDD*, volume 98, page 23, 1998.

[43] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Clustering categorical data: An approach based on dynamical systems. *The VLDB Journal*, 8(3-4):222–236, 2000.

[44] Lotfi A Zadeh. Similarity relations and fuzzy orderings. *Information sciences*, 3(2):177–200, 1971.

[45] Ashraf Al-Quran and Nasruddin Hassan. Fuzzy parameterised single valued neutrosophic soft expert set theory and its application in decision making. *International Journal of Applied Decision Sciences*, 9(2):212–227, 2016.

[46] Didier Dubois and Henri Prade. Rough fuzzy sets and fuzzy rough sets. *International Journal of General System*, 17(2-3):191–209, 1990.

[47] Xianning Wang and Zhi Xiao. A novel soft spatial weights matrix method based on soft sets. *International Journal of Applied Decision Sciences*, 9(1):39–69, 2016.

[48] J. Milgram, Mohamed Cheriet, and R. Sabourin. Estimating accurate multiclass probabilities with support vector machines. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 3, pages 1906–1911 vol. 3, July 2005.

[49] Ronald Aylmer Fisher, Frank Yates, et al. *Statistical tables for biological, agricultural and medical research.* Number Ed. 3. Oliver and Boyd, Edinburgh, 1949.

[50] HO Lancaster. The combination of probabilities arising from data in discrete distributions. *Biometrika*, 36(3/4):370–382, 1949.

[51] National Main Street Center. *Great American main street award winners*, 2016 (accessed February 28, 2017). http://www.mainstreet.org/main-street/awards/gamsa/past-winners.html.

[52] Wayne S DeSarbo, Rajdeep Grewal, and Jerry Wind. Who competes with whom? A demand-based perspective for identifying and representing asymmetric competition. *Strategic Management Journal*, 27(2):101–129, 2006.

[53] Google. *Google maps distance matrix API*, 2017 (accessed February 28, 2017). https://developers.google.com/maps/documentation/distance-matrix/.

[54] Edward Blair and Scot Burton. Cognitive processes used by survey respondents to answer behavioral frequency questions. *Journal of Consumer Research*, 14(2):280–288, 1987.

[55] Jenni Romaniuk, Byron Sharp, Samantha Paech, and Carl Driesener. Brand and advertising awareness: A replication and extension of a known empirical generalisation. *Australasian Marketing Journal (AMJ)*, 12(3):70–80, 2004.

[56] Vivek Srinivasan, Chan Su Park, and Dae Ryun Chang. An approach to the measurement, analysis, and prediction of brand equity and its sources. *Management Science*, 51(9):1433–1448, 2005.

[57] Paul E Green and Frank J Carmone. Multidimensional scaling: An introduction and comparison of nonmetric unfolding techniques. *Journal of Marketing Research*, 6(3):330–341, 1969.

[58] Glen L Urban, Philip L Johnson, and John R Hauser. Testing competitive market structures. *Marketing Science*, 3(2):83–112, 1984.

[59] Nathan Mantel. The detection of disease clustering and a generalized regression approach. *Cancer Research*, 27(2):209–220, 1967.

[60] Eric Bonnet and Yves Van de Peer. zt: A software tool for simple and partial mantel tests. *Journal of Statistical Software*, 7(10):1–12, 2002.

[61] E Jacquelin Dietz. Permutation tests for association between two distance matrices. *Systematic Biology*, 32(1):21–26, 1983.

[62] Jari Oksanen, F. Guillaume Blanchet, Michael Friendly, Roeland Kindt, Pierre Legendre, Dan McGlinn, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos, M. Henry H. Stevens, Eduard Szoecs, and Helene Wagner. vegan: Community ecology package, 2017. R package version 2.4-2.

[63] Pierre Dutilleul, Jason D Stockwell, Dominic Frigon, and Pierre Legendre. The mantel test versus pearson's correlation analysis: Assessment of the differences for biological and environmental studies. *Journal of Agricultural, Biological, and Environmental Statistics*, 5(2):131–150, 2000.

[64] Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):325–327, 1976.

[65] Jin Huang and Charles X Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310, 2005.

[66] Tom M Mitchell. *Machine Learning*. McGraw-Hill Boston, MA, 1997.

[67] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[68] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[69] Radford M Neal. Slice sampling. *Annals of Statistics*, pages 705–741, 2003.

[70] Kurt Hornik, Friedrich Leisch, and Achim Zeileis. Jags: A program for analysis of bayesian graphical models using gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, 2003.

[71] Martyn Plummer. Jags: Just another gibbs sampler, 2004. Version 0.8.

[72] David Spiegelhalter, Andrew Thomas, Nicky Best, and Wally Gilks. Bugs 0.5: Bayesian inference using gibbs sampling manual (version ii). *MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK*, pages 1–59, 1996.

[73] David J Lunn, Andrew Thomas, Nicky Best, and David Spiegelhalter. Winbugs-a bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10(4):325–337, 2000.

[74] David Spiegelhalter, Andrew Thomas, Nicky Best, and Dave Lunn. Winbugs user manual, 2003.

[75] David Spiegelhalter, Andrew Thomas, Nicky Best, and Dave Lunn. Openbugs user manual, version 3.0. 2. *MRC Biostatistics Unit, Cambridge*, 2007.

[76] Sibylle Sturtz, Uwe Ligges, and Andrew Gelman. R2openbugs: a package for running openbugs from r, 2010.

[77] Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[78] Mark Pagel and Andrew Meade. Bayesian analysis of correlated evolution of discrete characters by reversible-jump Markov chain Monte Carlo. *The American Naturalist*, 167(6):808–825, 2006.

[79] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[80] N. de Freitas. Rao-blackwellised particle filtering for fault diagnosis. In *Proceedings, IEEE Aerospace Conference*, volume 4, pages 4–1767–4–1772 vol.4, 2002.

[81] Paul-Christian Bürkner et al. brms: An r package for bayesian multilevel models using stan. *Journal of Statistical Software*, 80(1):1–28, 2016.

[82] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1), 2017.

[83] Tracy M Sweet. *A Review of Statistical Rethinking: A Bayesian Course With Examples in R and Stan*, volume 42. SAGE Publications Sage CA: Los Angeles, CA, 2017.

[84] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer, 2004.

[85] Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

[86] Oliver Stegle, Sebastian V Fallert, David JC MacKay, and Søren Brage. Gaussian process robust regression for noisy heart rate data. *IEEE Transactions on Biomedical Engineering*, 55(9):2143–2151, 2008.

[87] Datong Liu, Jingyue Pang, Jianbao Zhou, Yu Peng, and Michael Pecht. Prognostics for state of health estimation of lithium-ion batteries based on combination gaussian process functional regression. *Microelectronics Reliability*, 53(6):832–839, 2013.

[88] George S Kimeldorf and Grace Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970.

[89] Jonah Gabry and Tristan Mahr. *Bayesplot plotting bayesian models*, 2018 (accessed Jun 2, 2018). http://mc-stan.org/bayesplot/.

[90] Jonah Gabry, Daniel Simpson, Aki Vehtari, Michael Betancourt, and Andrew Gelman. Visualization in Bayesian workflow. *arXiv preprint arXiv:1709.01449*, 2017.

[91] Stan Development Team. rstanarm: Bayesian applied regression modeling via Stan., 2016. R package version 2.13.1.

[92] Matthew Kay. *Using tidy data with Bayesian samplers*, 2018 (accessed Jun 13, 2018. https://mjskay.github.io/tidybayes/articles/tidybayes.html.

[93] Blake MacDonald, Pritam Ranjan, and Hugh Chipman. GPfit: An R package for Gaussian process model fitting using a new optimization algorithm. *arXiv preprint arXiv:1305.0759*, 2013.

[94] Giri Gopalan and Luke Bornn. Fastgp: An R package for Gaussian processes. *arXiv preprint arXiv:1507.06055*, 2015.

[95] Garrett M Dancik. *mlegp: Maximum likelihood estimates of Gaussian processes*, 2013 (accessed Jun 13, 2018). http://cran.r-project.org/web/packages/mlegp/index.html.

[96] Yves Deville, David Ginsbourger, Olivier Roustant, and Nicolas Durrande. *kergp: Gaussian Process Laboratory*, 2015 (accessed Jun 13, 2018). https://CRAN.R-project.org/package=kergp.

[97] Mark Ebden et al. Gaussian processes for regression: A quick introduction. *The Website of Robotics Research Group in Department on Engineering Science, University of Oxford*, 2008.

[98] J Bernardo, J Berger, A Dawid, A Smith, et al. Regression and classification using gaussian process priors. *Bayesian Statistics*, 6:475, 1998.

[99] Brian D Ferris, Dieter Fox, and Neil Lawrence. Wifi-slam using gaussian process latent variable models. In *the 20th International Joint Conference on Artificial Intelligence*, pages 2480–2485, 2007.

[100] Andrew Wilson and Ryan Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning*, pages 1067–1075, 2013.

[101] Andrew. *Instead of "confidence interval", lets say "uncertainty interval"*, 2010 (accessed March 27, 2018). http://andrewgelman.com/2010/12/21/lets_say_uncert/.

[102] Andrew McHutchon. Differentiating gaussian processes. *Cambridge (ed.)*, 2013.

[103] Kunio Tanabe and Masahiko Sagae. An exact cholesky decomposition and the generalized inverse of the variance-covariance matrix of the multinomial distribution, with applications. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(1):211–219, 1992.

[104] John Howell. *Stan Best Practices*, 2017 (accessed Jun 2, 2018). https://github.com/stan-dev/stan/wiki/Stan-Best-Practices.

[105] Jonah Gabry. *Visual MCMC diagnostics using the bayesplot package*, 2018 (accessed Jun 2, 2018). https://cran.r-project.org/web/packages/bayesplot/vignettes/visual-mcmc-diagnostics.html.

[106] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[107] Wisconsin Department of Health Services. *Sulfates*, 2018 (accessed Jun 4, 2018). https://www.dhs.wisconsin.gov/chemical/sulfates.htm.

[108] Margaret Moss. *Sulphates and Sulphites - the Good, the Moderately Bad and the Ugly*, 2009 (accessed Jun 4, 2018). https://www.foodsmatter.com/allergy_intolerance/sulphites/articles/sulphates_sulphites.html.

[109] P Appalasamy, A Mustapha, ND Rizal, F Johari, and AF Mansor. Classification-based data mining approach for quality control in wine production. *Journal of Applied Sciences*, 12(6):598–601, 2012.

[110] Hao Zhang. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99(465):250–261, 2004.

[111] Stan Development Team. *Stan*, 2018 (accessed March 26, 2018). http://mc-stan.org/.

[112] Ying Liu, Andrew Gelman, and Tian Zheng. Simulation-efficient shortest probability intervals. *Statistics and Computing*, 25(4):809–819, 2015.

[113] C. J. Moore, A. J. K. Chua, C. P. L. Berry, and J. R. Gair. Fast methods for training gaussian processes on large datasets. *Royal Society Open Science*, 3(5), 2016.

[114] Troyen A Brennan, Lucian L Leape, Nan M Laird, Liesi Hebert, A Russell Localio, Ann G Lawthers, Joseph P Newhouse, Paul C Weiler, and Howard H Hiatt. Incidence of adverse events and negligence in hospitalized patients: Results of the Harvard Medical Practice Study I. *New England Journal of Medicine*, 324(6):370–376, 1991.

[115] Eefje N de Vries, Maya A Ramrattan, Susanne M Smorenburg, Dirk J Gouma, and Marja A Boermeester. The incidence and nature of in-hospital adverse events: A systematic review. *BMJ Quality & Safety*, 17(3):216–223, 2008.

[116] G Ross Baker, Peter G Norton, Virginia Flintoft, Régis Blais, Adalsteinn Brown, Jafna Cox, Ed Etchells, William A Ghali, Philip Hébert, Sumit R Majumdar, et al. The canadian adverse events study: The incidence of adverse events among hospital patients in canada. *Canadian Medical Association Journal*, 170(11):1678–1686, 2004.

[117] Charles Vincent, Graham Neale, and Maria Woloshynowych. Adverse events in british hospitals: Preliminary retrospective record review. *BMJ*, 322(7285):517–519, 2001.

[118] Oliver Anderson, Rachel Davis, George B Hanna, and Charles A Vincent. Surgical adverse events: A systematic review. *The American Journal of Surgery*, 206(2):253–262, 2013.

[119] Mark Van Tuinen, Susan Elder, Carolyn Link, Susan Li, John H Song, and Tracey Pritchett. *Surveillance of surgery-related adverse events in Missouri using ICD-9-CM codes*. Agency for Healthcare Research and Quality (US), 2005.

[120] Ashley Kable, Robert Gibberd, and Allan Spigelman. Adverse events in five surgical procedures. *Clinical Governance: An International Journal*, 14(2):145–155, 2009.

[121] Atul A Gawande, Eric J Thomas, Michael J Zinner, and Troyen A Brennan. The incidence and nature of surgical adverse events in colorado and utah in 1992. *Surgery*, 126(1):66–75, 1999.

[122] S Kuske, C Lessing, R Lux, A Schmitz, and M Schrappe. Patient safety indicators for medication safety (amts-psi): International status, transferability and validation. *Gesundheitswesen (Bundesverband der Arzte des Offentlichen Gesundheitsdienstes (Germany))*, 74(2):79–86, 2012.

[123] Patrick S Romano, Hillary J Mull, Peter E Rivard, Shibei Zhao, William G Henderson, Susan Loveland, Dennis Tsilimingras, Cindy L Christiansen, and Amy K Rosen. Validity of selected ahrq patient safety indicators based on va national surgical quality improvement program data. *Health Services Research*, 44(1):182–204, 2009.

[124] James Reason. Human error: Models and management. *BMJ: British Medical Journal*, 320(7237):768, 2000.

[125] Linda L Liu and Jacqueline M Leung. Predicting adverse postoperative outcomes in patients aged 80 years or older. *Journal of the American Geriatrics Society*, 48(4):405–412, 2000.

[126] Chunliu Zhan, James Battles, Yen-pin Chiang, and David Hunt. The validity of icd-9-cm codes in identifying postoperative deep vein thrombosis and pulmonary embolism. *Joint Commission Journal on Quality and Patient Safety*, 33(6):326–331, 2007.

[127] Viraj Bhise, Dean F Sittig, Viralkumar Vaghani, Li Wei, Jessica Baldwin, and Hardeep Singh. An electronic trigger based on care escalation to identify preventable adverse events in hospitalised patients. *BMJ Qual Saf*, 27(3):241–246, 2018.

[128] A Bate and SJW Evans. Quantitative signal detection using spontaneous adr reporting. *Pharmacoepidemiology and Drug Safety*, 18(6):427–436, 2009.

[129] Alexander Genkin, David D Lewis, and David Madigan. Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304, 2007.

[130] Ola Caster, G Niklas Norén, David Madigan, and Andrew Bate. Large-scale regression-based pattern discovery: The example of screening the who global drug safety database. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 3(4):197–208, 2010.

[131] Yanqing Ji, Hao Ying, Peter Dews, Ayman Mansour, John Tran, Richard E Miller, and R Michael Massanari. A potential causal association mining algorithm for screening adverse drug reactions in postmarketing surveillance. *IEEE Transactions on Information Technology in Biomedicine*, 15(3):428–437, 2011.

[132] Huidong Jin, Jie Chen, Hongxing He, Graham J Williams, Chris Kelman, and Christine M O'Keefe. Mining unexpected temporal associations: Applications in detecting adverse drug reactions. *IEEE Transactions on Information Technology in Biomedicine*, 12(4):488–500, 2008.

[133] Jonathan H Chen, Mary K Goldstein, Steven M Asch, Lester Mackey, and Russ B Altman. Predicting inpatient clinical order patterns with probabilistic topic models vs conventional order sets. *Journal of the American Medical Informatics Association*, 24(3):472–480, 2017.

[134] Azadeh Nikfarjam, Abeed Sarker, Karen OConnor, Rachel Ginn, and Graciela Gonzalez. Pharmacovigilance from social media: Mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681, 2015.

[135] Rave Harpaz, Hector Perez, Herbert S Chase, Raul Rabadan, George Hripcsak, and Carol Friedman. Biclustering of adverse drug events in the fda's spontaneous reporting system. *Clinical Pharmacology & Therapeutics*, 89(2):243–250, 2011.

[136] Jeffrey L Schnipper, Claus Hamann, Chima D Ndumele, Catherine L Liang, Marcy G Carty, Andrew S Karson, Ishir Bhan, Christopher M Coley, Eric Poon, Alexander Turchin, et al. Effect of an electronic medication reconciliation application and process redesign on potential adverse drug events: A cluster-randomized trial. *Archives of Internal Medicine*, 169(8):771–780, 2009.

[137] Andrew Bate, Marie Lindquist, IR Edwards, Sten Olsson, Roland Orre, Anders Lansner, and R Melhado De Freitas. A bayesian neural network method for adverse drug reaction signal generation. *European Journal of Clinical Pharmacology*, 54(4):315–321, 1998.

[138] Jeffrey S Brown, Martin Kulldorff, K Arnold Chan, Robert L Davis, David Graham, Parker T Pettus, Susan E Andrade, Marsha A Raebel, Lisa Herrinton, Douglas Roblin, et al. Early detection of adverse drug events within population-based health networks: Application of sequential testing methods. *Pharmacoepidemiology and Drug Safety*, 16(12):1275–1284, 2007.

[139] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14, Mar 1995.

[140] Jenna Reps, Jonathan M Garibaldi, Uwe Aickelin, Daniele Soria, Jack E Gibson, and Richard B Hubbard. Discovering sequential patterns in a UK general practice database. In *Biomedical and Health Informatics (BHI), 2012 IEEE-EMBS International Conference on*, pages 960–963. IEEE, 2012.

[141] K. Malhotra, T. C. Hobson, S. Valkova, L. L. Pullum, and A. Ramanathan. Sequential pattern mining of electronic healthcare reimbursement claims: Experiences and challenges in uncovering how patients are treated by physicians. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2670–2679, Oct 2015.

[142] Aileen P Wright, Adam T Wright, Allison B McCoy, and Dean F Sittig. The use of sequential pattern mining to predict next prescribed medications. *Journal of Biomedical Informatics*, 53:73–80, 2015.

[143] G Niklas Norén, Johan Hopstadius, Andrew Bate, Kristina Star, and I Ralph Edwards. Temporal pattern discovery in longitudinal electronic patient records. *Data Mining and Knowledge Discovery*, 20(3):361–387, 2010.

[144] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

[145] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[146] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[147] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[148] Andrew Bate, Marie Lindquist, I Ralph Edwards, and Roland Orre. A data mining approach for signal detection and analysis. *Drug Safety*, 25(6):393–397, 2002.

[149] Andrew Bate. Bayesian confidence propagation neural network. *Drug Safety*, 30(7):623–625, 2007.

[150] Trung Huynh, Yulan He, Alistair Willis, and Stefan Rüger. Adverse drug reaction classification with deep neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 877–887, 2016.

[151] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 2000.

[152] Qing Cao, Bradley T Ewing, and Mark A Thompson. Forecasting wind speed with recurrent neural networks. *European Journal of Operational Research*, 221(1):148–154, 2012.

[153] Hude Quan, Vijaya Sundararajan, Patricia Halfon, Andrew Fong, Bernard Burnand, Jean-Christophe Luthi, L Duncan Saunders, Cynthia A Beck, Thomas E Feasby, and William A Ghali. Coding algorithms for defining comorbidities in icd-9-cm and icd-10 administrative data. *Medical Care*, pages 1130–1139, 2005.

[154] Anne Elixhauser, Claudia Steiner, D Robert Harris, and Rosanna M Coffey. Comorbidity measures for use with administrative data. *Medical Care*, 36(1):8–27, 1998.

[155] Nicolas R Thompson, Youran Fan, Jarrod E Dalton, Lara Jehi, Benjamin P Rosenbaum, Sumeet Vadera, and Sandra D Griffith. A new elixhauser-based comorbidity summary measure to predict in-hospital mortality. *Medical Care*, 53(4):374, 2015.

[156] Jacques Wicki, Arnaud Perrier, Thomas V Perneger, Henri Bounameaux, and Alain François Junod. Predicting adverse outcome in patients with acute pulmonary embolism: A risk score. *Thromb and Haemost*, 84(4):548–552, 2000.

[157] Elizabeth A Genovese, Larry Fish, Rabih A Chaer, Michel S Makaroun, and Donald T Baril. Risk stratification for the development of respiratory adverse events following vascular surgery using the society of vascular surgery's vascular quality initiative. *Journal of Vascular Surgery*, 65(2):459–470, 2017.

[158] Bobak J Mortazavi, Nihar Desai, Jing Zhang, Andreas Coppi, Fred Warner, Harlan M Krumholz, and Sahand Negahban. Prediction of adverse events in patients undergoing major cardiovascular procedures. *IEEE Journal of Biomedical and Health Informatics*, 21(6):1719–1729, 2017.

[159] John A Dodson, Matthew R Reynolds, Haikun Bao, Sana M Al-Khatib, Eric D Peterson, Mark S Kremers, Michael J Mirro, Jeptha P Curtis, et al. Developing a risk model for in-hospital adverse events following implantable cardioverter-defibrillator implantation: A report from the ncdr (national cardiovascular data registry). *Journal of the American College of Cardiology*, 63(8):788–796, 2014.

[160] Maciej Kusy, Bogdan Obrzut, and Jacek Kluska. Application of gene expression programming and neural networks to predict adverse events of radical hysterectomy in cervical cancer patients. *Medical & Biological Engineering & Computing*, 51(12):1357–1365, 2013.

[161] Ronald J Krone, Warren K Laskey, Craig Johnson, Stephen E Kimmel, Lloyd W Klein, Bonnie H Weiner, JJ Adolfo Cosentino, Sarah A Johnson, and Joseph D Babb. A simplified lesion classification for predicting success and complications of coronary angioplasty. *American Journal of Cardiology*, 85(10):1179–1184, 2000.

[162] Jane M Geraci, Amy K Rosen, Arlene S Ash, Kathleen J McNiff, and Mark A Moskowitz. Predicting the occurrence of adverse events after coronary artery bypass surgery. *Annals of Internal Medicine*, 118(1):18–24, 1993.

[163] Amy K Rosen, Jane M Geraci, Arlene S Ash, Kathleen J McNiff, and Mark A Moskowitz. Postoperative adverse events of common surgical procedures in the medicare population. *Medical Care*, pages 753–765, 1992.

[164] Jeremy Ginsberg, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski, and Larry Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2009.

[165] Declan Butler. When google got flu wrong. *Nature*, 494(7436):155, 2013.

[166] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The parable of Google flu: Traps in big data analysis. *Science*, 343(6176):1203–1205, 2014.

[167] Robert L Houchens, Anne Elixhauser, and Patrick S Romano. How often are potential patient safety events present on admission? *Joint Commission Journal on Quality and Patient Safety*, 34(3):154–163, 2008.

Appendix A

APPLICATION IN LOCAL COMPETITOR IDENTIFICATION

Corresponding to Chapter § 3 Section 3.1

Average frequency of visiting restaurants in Newark, DE for lunch	More than five times a week	8	Average frequency of visiting restaurants in Newark, DE for dinner	More than five times a week	7
	Three to five times a week	28		Three to five times a week	31
	Once to twice a week	92		Once to twice a week	95
	Two to three times a month	34		Two to three times a month	38
	About once a month	19		About once a month	17
	Less than once a month	15		Less than once a month	11
	Never	3		Never	0
Average spending in restaurants every month	Less than \$50	49	School Year	Freshman	2
	\$50 - \$99	73		Sophomore	90
	\$100 - \$199	45		Junior	86
	\$200 - \$299	16		Senior	16
	\$300 - \$399	5	Using student dinning plan?	Yes	76
	\$400 - \$499	6		No	123
	More than \$500	5	Living in dorm or off campus?	Dorm	58
Sex	Male	66		Off Campus	141
	Female	133	Ethnicity	White	165
Age	18 - 20	161		Black or African American	6
	21 - 23	31		Asian	18
	More than 24	7		Other	8

Table A.1: Demographic Description of Survey Respondents (N=199)

[1]We have ran two rounds of survey sessions in total. The first round only contained the first type of questions asking them to rank their willingness to switch if a *seed restaurant* is not available, the sessions were run in September 2016 and collected 125 valid responses. We added the second type of questions (similarity-based perceptual mapping) as an alternative validation method to compare our algorithm result and survey results. Then the second round of survey sessions were run in February 2017 and collected 73 valid responses. There is no overlap or any interaction in the student subjects between the two rounds of surveys. The restaurant competition environment is also stable during the survey periods. And there is no significant difference between the demographics of survey respondents in the first round sessions and the second round sessions.

Restaurant	Amount	YelpTags	Rating
Ali Baba	13	{Mediterranean, Hookah Bars, Lebanese}	4.5
Applebees	15	{Sports Bars, Burgers, American (Traditional)}	2.5
Bamboo House	18	{Chinese, Sushi Bars}	3.5
Banh Mi Boy	9	{Vietnamese, Coffee & Tea, Sandwiches}	4
Bennie Dollards Fantasy Cuisine	7	{Food Trucks}	5
Buffalo Wild Wings	14	{Chicken Wings, American (Traditional), Sports Bars}	3
Burger King	7	{Fast Food, Burgers, Hot Dogs}	2
Caffe Gelato	22	{Ice Cream & Frozen Yogurt, Italian, Gelato}	3.5
California Tortilla	9	{Mexican, Tex-Mex, Vegetarian}	4
Catherine Rooney's Irish Pub	16	{Breakfast & Brunch, American (Traditional), Irish Pub}	3
Cheeburger Cheeburger	13	{Burgers}	3
CHEF TAN	15	{Chinese}	4
Chick-fil-A	8	{Fast Food}	4
Chipotle Mexican Grill	10	{Mexican, Fast Food}	2.5
Claymont Steak Shop	12	{Cheesesteaks, Sandwiches}	3
Colorful Yun Nan	14	{Chinese, Noodles}	4
Cosi	11	{American (New)}	3.5
Dairy Queen	6	{Fast Food, Ice Cream & Frozen Yogurt}	3.5
Deer Park Tavern	17	{Bars, American (Traditional)}	3.5
Del Pez Mexican Gastropub	16	{Mexican, Latin American, Gastropubs}	3.5
Dominos Pizza	10	{Pizza, Chicken Wings, Sandwiches}	1.5
El Diablo Burritos	11	{Mexican}	4
Grain Craft Bar + Kitchen	19	{American (New), Bars, Gastropubs}	3.5
Grotto Pizza	9	{Pizza, Burgers, Sandwiches}	2.5
Home Grown Cafe	16	{American (New), Vegetarian, Music Venues}	3.5
Honeygrow	12	{American (New), Salad, Honey}	4
Iron Hill Brewery & Restaurant	21	{American (New), Breweries}	4
Jimmy John's	7	{Sandwiches}	4
KFC	8	{Fast Food, Chicken Wings}	2.5
Klondike Kate's	18	{Bars, American (New)}	3
Margherita's Pizza	8	{Pizza}	3.5
Mayflower Japanese Restaurant	13	{Japanese}	3
McDonald's	6	{Burgers, Fast Food}	3.5
Mediterranean Grille	14	{Mediterranean}	4.5
Newark Deli & Bagels	7	{Delis, Bagels, Breakfast & Brunch}	3.5
Outback Steakhouse	21	{Steakhouses}	3
Panera Bread	12	{Sandwiches, Salad, Soup}	2.5
Papa John's Pizza	8	{Pizza}	2.5
Pat's Pizzeria	11	{Pizza, Sports Bars, Karaoke}	2.5
Popeyes	8	{Fast Food, Chicken Wings}	2
Ramen Kumamoto	13	{Ramen}	4
Seasons Pizza	9	{Italian, Pizza, Chicken Wings}	2.5
Sinclairs Cafe	12	{Breakfast & Brunch, Cafes}	3.5
Stone Balloon Ale House	21	{Gastropubs, Pubs, Comfort Food}	4
Subway	7	{Sandwiches, Fast Food}	4
Taverna	23	{Italian, Bars, Pizza}	4
The Cart At UD	7	{Asian Fusion, Food Trucks, Street Vendors}	3
The Greene Turtle	17	{American (Traditional), Sports Bars, Burgers}	3
The Red Bowl	11	{Asian Fusion, Chinese}	3.5
Vita Nova	28	{American (New)}	5

Table A.2: Data Feed

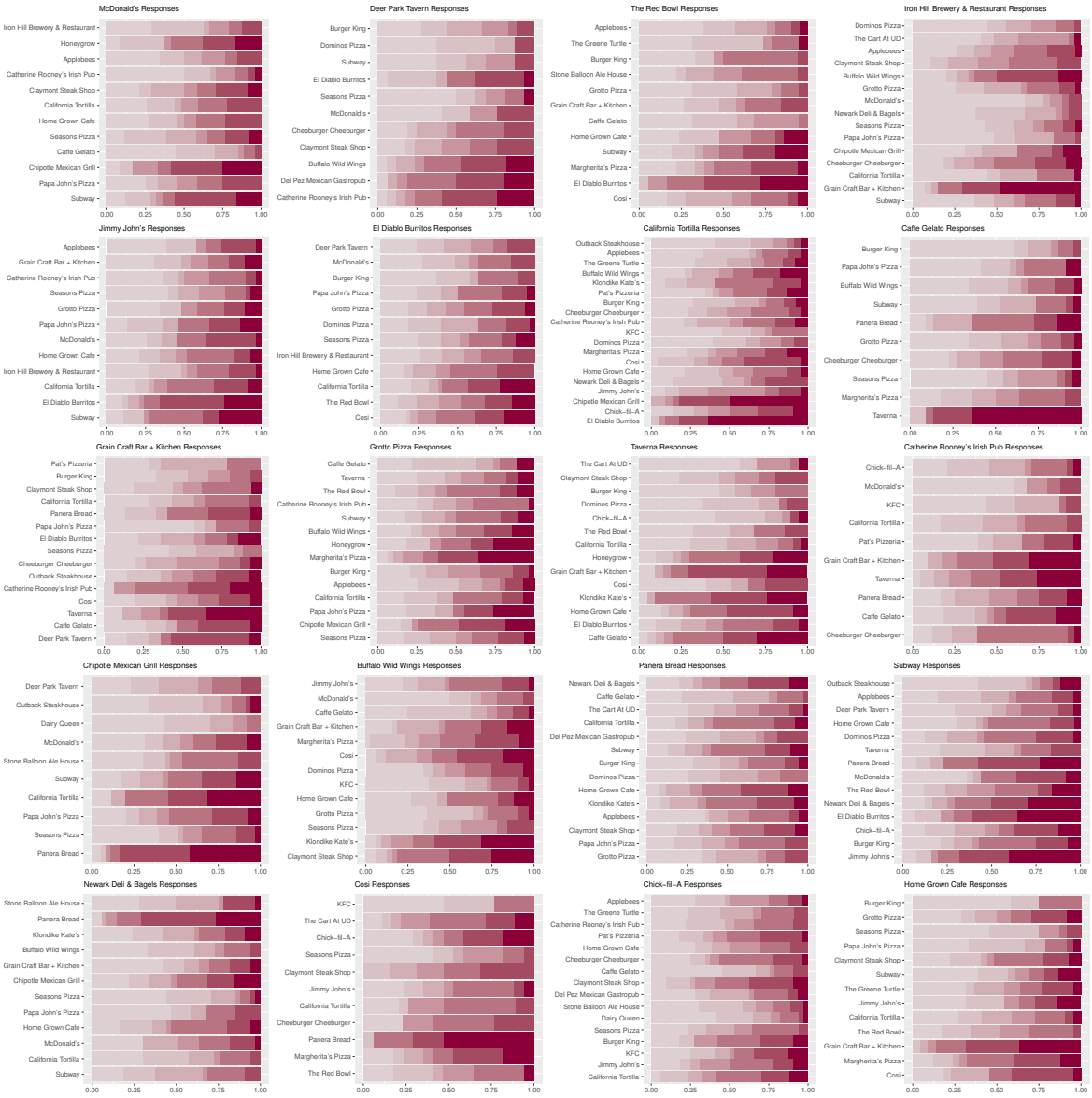


Figure A.1: Survey Response Composition Ranked by Similarity Measure

Appendix B

APPLICATION IN TRAVELER ANCILLARY PURCHASE PREDICTION

Corresponding to Chapter § 3 Section 3.2

Table B.1: Categorization of Detailed Ancillary Purchases

Airport Early Sale (N=280)	AIRPORT EARLY SALE
Booking Fee (N=184)	BOOKING FEE MANUALY PRICED GSF BOOKING FEE GSF BOOKING FEE MANUALY PRICED
Checked Bag (N=1012)	CHECKED BAG FIFTH CHECKED BAG FIRST CHECKED BAG FOURTH CHECKED BAG SECOND CHECKED BAG SIXTH CHECKED BAG THIRD EXTRA BAG UPTO 32KG FIFTH CHECKED BAG UPTO 23KG FIRST CHECKED BAG UPTO 23KG FOURTH CHECKED BAG UPTO 23KG THIRD CHECKED BAG UPTO 23KG UPGRADE TO 32KG UPTO70LB 32KG BAGGAGE XBAG ON OAL MKTD EY OPERATED
Ticket Change Fee (N=610)	GSF TICKET CHANGE FEE GSF TKT CHANGE FEE FOR WEB GSF TKT CHG FEE MANUALY PRICED TKT CHG FEE MANUALLY PRICED TKT CHANGE FEE FOR WEB
Upgrade (N=307)	INSTANT AIRPORT UPGRADE C TO F INSTANT AIRPORT UPGRADE Y TO C MILES UPGRADE Y TO C PUSH UPGRADE C TO F PUSH UPGRADE Y TO C UPGRADE MILES FROM J TO F PLUS GRADE
Miscellaneous Charge (N=57)	NAME CORRECTION FEE
Pre Reserved Seat Assignment (N=282)	PRE RESERVED SEAT ASSIGNMENT

Figure B.1: K Nearest Neighbor using Uniqueness-motivated Similarity Performance of Traveler Ancillary Purchase Probability Prediction on Airport Early Sale

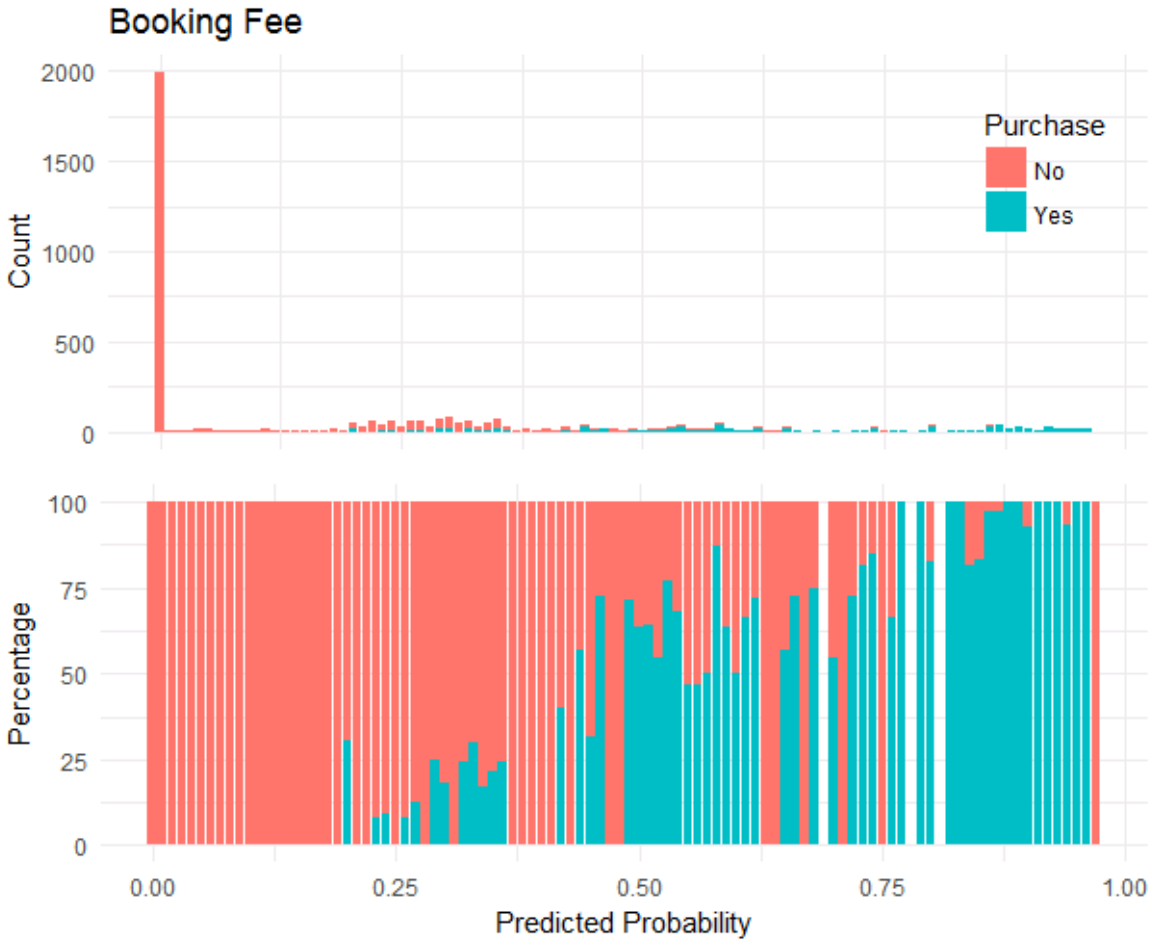


Figure B.2: K Nearest Neighbor using Uniqueness-motivated Similarity Performance of Traveler Ancillary Purchase Probability Prediction on Booking Fee

Figure B.3: K Nearest Neighbor using Uniqueness-motivated Similarity Performance of Traveler Ancillary Purchase Probability Prediction on Checked Bag

Figure B.4: K Nearest Neighbor using Uniqueness-motivated Similarity Performance of Traveler Ancillary Purchase Probability Prediction on Ticket Change Fee

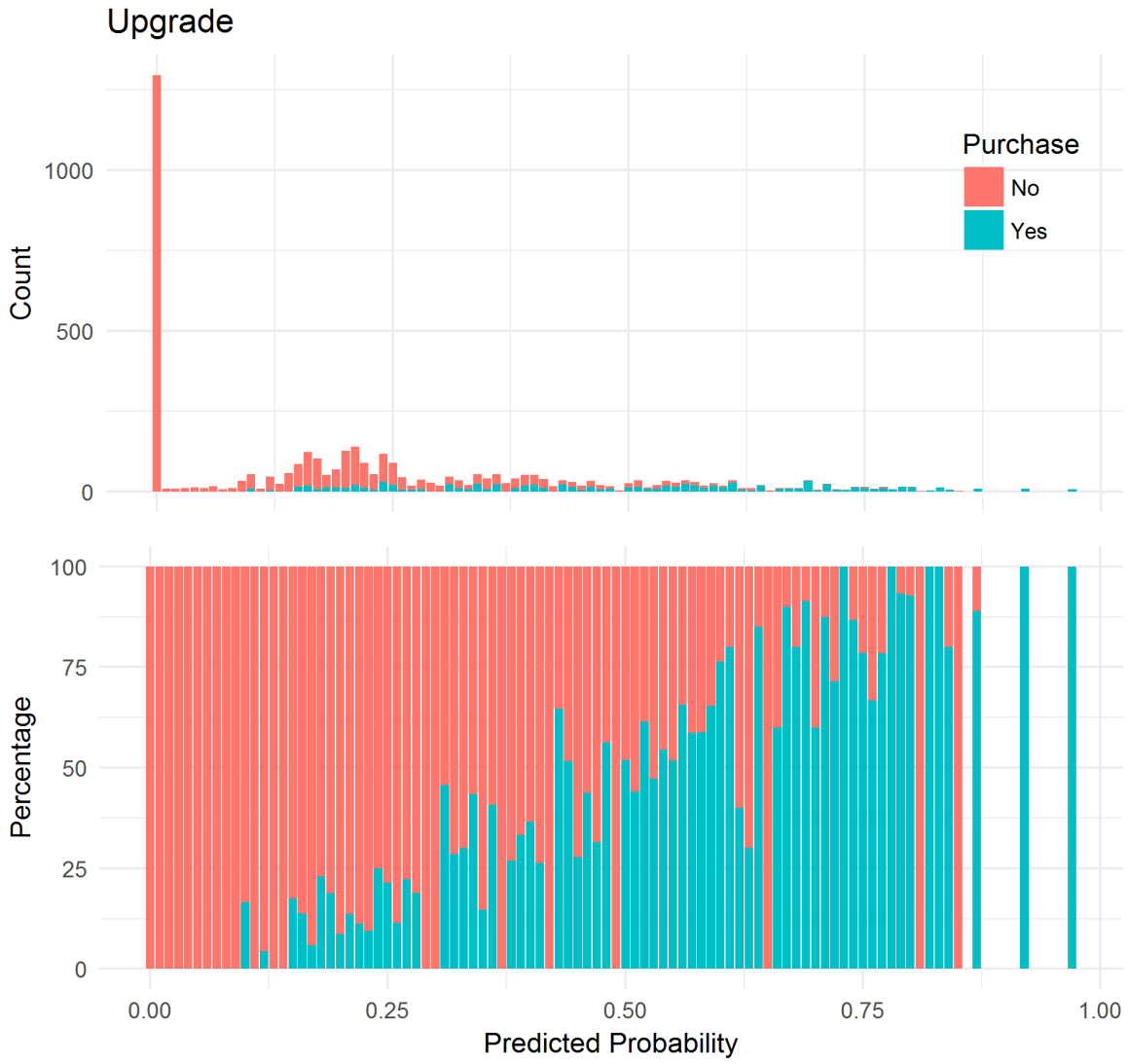


Figure B.5: K Nearest Neighbor using Uniqueness-motivated Similarity Performance of Traveler Ancillary Purchase Probability Prediction on Upgrade

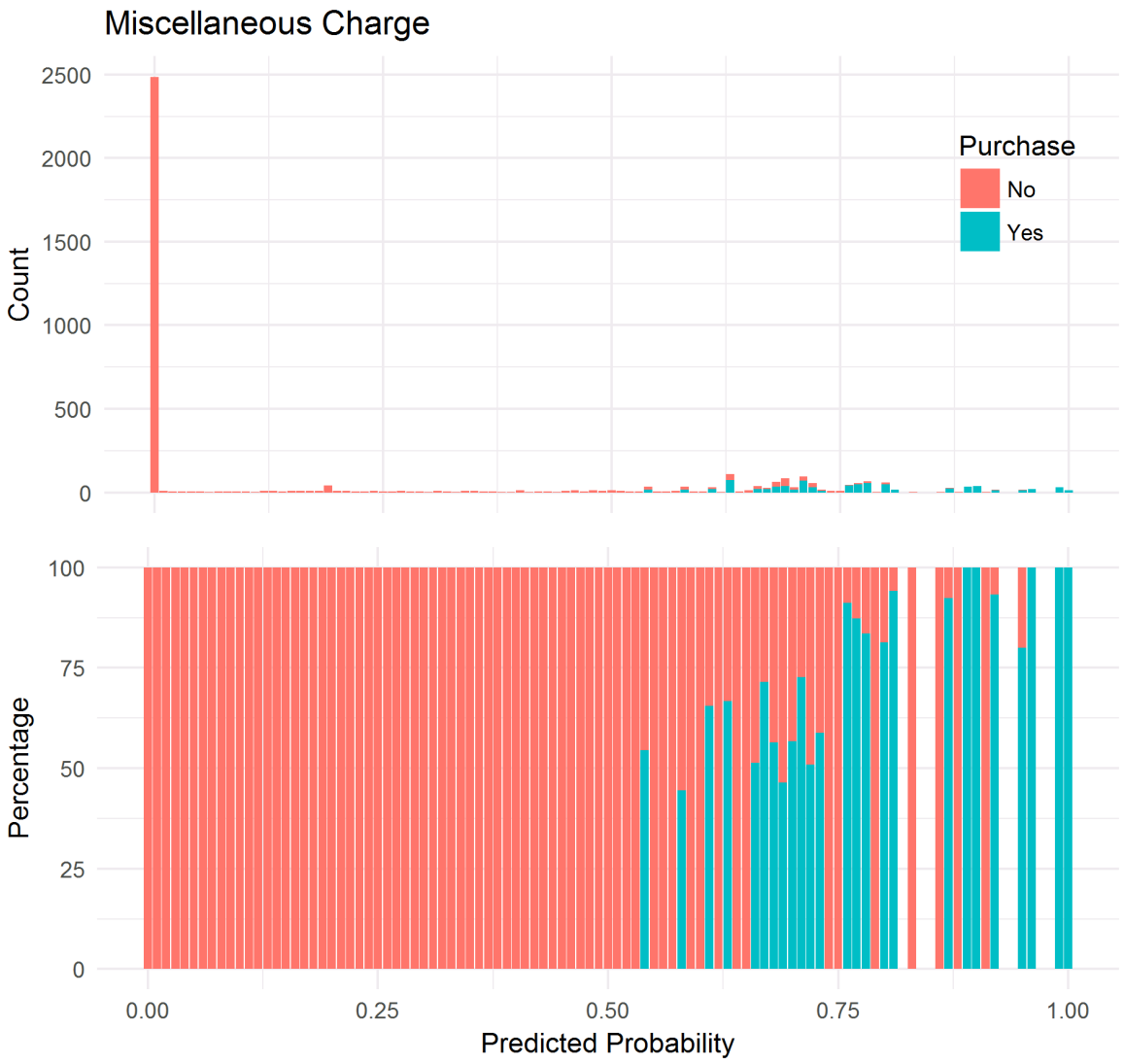


Figure B.6: K Nearest Neighbor using Uniqueness-motivated Similarity Performance of Traveler Ancillary Purchase Probability Prediction on Miscellaneous Charge

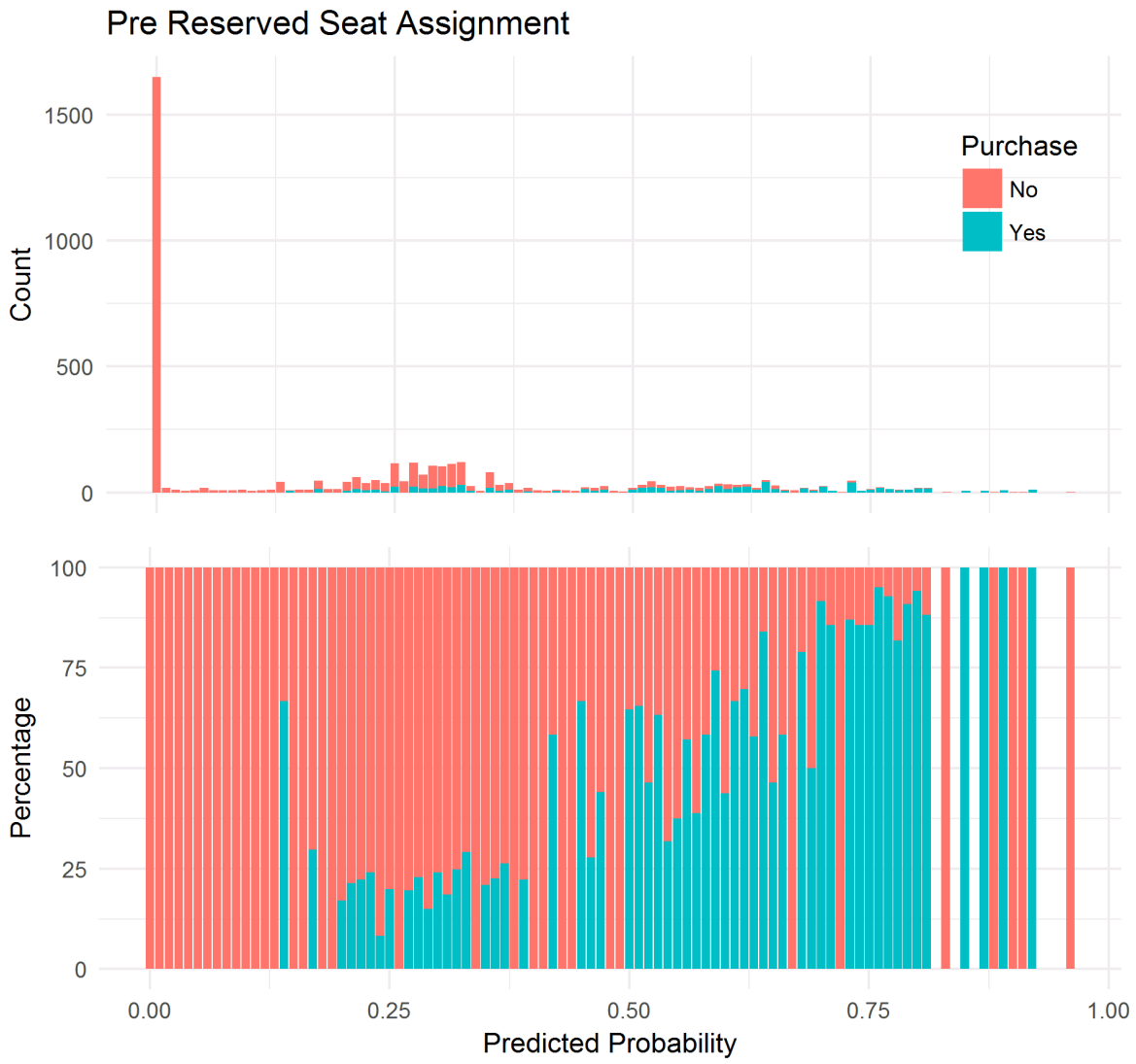


Figure B.7: K Nearest Neighbor using Uniqueness-motivated Similarity Performance of Traveler Ancillary Purchase Probability Prediction on Pre Reserved Seat Assignment

Appendix C

STAN CODE USED FOR GAUSSIAN PROCESS REGRESSION MODELING

Corresponding to Chapter § 4 Section 4.3

```
functions{
 matrix gp_pred_rng(real[] x1,
 vector y1,
 real[] x2,
 real alpha,
 real rho,
 real sigma,
 real delta) {
  int N1 = size(x1);
  int N2 = size(x2);
  vector[N2] f2_mu;
  vector[N2] f2;
  vector[N2] f2_prime_mu;
  vector[N2] f2_prime_var;
  matrix[3, N2] f2_stat;
  {
   matrix[N1, N1] L_K;
   vector[N1] L_div_y1;
   matrix[N1, N2] k_x1_x2;
   matrix[N1, N2] x_diff;
```

```
matrix[N1, N2] k_x1_x2_p;
matrix[N1, N2] L_div_k_x1_x2;
vector[N1] L_div_k_x1_x2_p;
matrix[N2, N2] f2_cov;
matrix[N2, N2] diag_delta;
matrix[N1, N1] K;

K = cov_exp_quad(x1, alpha, rho)
    + diag_matrix(rep_vector(1, N1))*square(sigma);
L_K = cholesky_decompose(K);
k_x1_x2 = cov_exp_quad(x1, x2, alpha, rho);
L_div_k_x1_x2 = mdivide_left_tri_low(L_K, k_x1_x2);
L_div_y1 = mdivide_left_tri_low(L_K, y1);

f2_mu = L_div_k_x1_x2` * L_div_y1;
f2_cov = cov_exp_quad(x2, alpha, rho)
       - L_div_k_x1_x2` * L_div_k_x1_x2;

diag_delta = diag_matrix(rep_vector(delta, N2));
f2 = multi_normal_rng(f2_mu, f2_cov + diag_delta);

for (i in 1:N1){
 for (j in 1:N2){
  x_diff[i, j] = x1[i] - x2[j];
 }
}

k_x1_x2_p = ( 1 / square(rho)) * x_diff .* k_x1_x2;
```

```
   for (i in 1:N2){
    L_div_k_x1_x2_p = mdivide_left_tri_low(L_K, k_x1_x2_p[,i]);
    f2_prime_mu[i] = L_div_k_x1_x2_p` * L_div_y1;
    f2_prime_var[i] = sqrt( square(alpha/rho)
                      - L_div_k_x1_x2_p` * L_div_k_x1_x2_p );
   }

   f2_stat[1] = to_row_vector(f2);
   f2_stat[2] = to_row_vector(f2_prime_mu);
   f2_stat[3] = to_row_vector(f2_prime_var);
  }
 return f2_stat;
 }
}

data{
 int<lower=1> N1;
 int<lower=1> N2;
 real x1[N1];
 vector[N1] y1;
 real x2[N2];
 real<lower=0> rho_alpha;
 real<lower=0> rho_beta;
 real<lower=0> alpha_mean;
 real<lower=0> alpha_sd;
 real<lower=0> sigma_mean;
 real<lower=0> sigma_sd;
}
```

```
transformed data{
 real delta = 1e-9;
 vector[N1] mu;
 for(n in 1:N1) mu[n] = 0;
}

parameters{
 real<lower=0> alpha;
 real<lower=0> rho;
 real<lower=0> sigma;
}

model{
 matrix[N1,N1] Sigma;
 matrix[N1,N1] L_S;

 Sigma = cov_exp_quad(x1, alpha, rho)
       + diag_matrix(rep_vector(1, N1))*square(sigma);

 L_S = cholesky_decompose(Sigma);
 y1 ~ multi_normal_cholesky(mu, L_S);

 rho ~ inv_gamma(rho_alpha, rho_beta);
 alpha ~ normal(alpha_mean, alpha_sd);
 sigma ~ normal(sigma_mean, sigma_sd);
}

generated quantities{
```

```
 matrix[3, N2] f2_stat;
 vector[N2] f2;
 vector[N2] fp2;
 vector[N2] f2_prime_mu;
 vector[N2] f2_prime_var;
 vector[N2] f2_prime;
 f2_stat = gp_pred_rng(x1, y1, x2, alpha, rho, sigma, delta);
 f2 = f2_stat[1]`;
 f2_prime_mu = f2_stat[2]`;
 f2_prime_var = f2_stat[3]`;

 for (n2 in 1:N2){
  fp2[n2] = normal_rng(f2[n2], sigma);
  f2_prime[n2] = normal_rng(f2_prime_mu[n2], f2_prime_var[n2]);
 }
}
```

Appendix D

EXPLORING THE RELATIONSHIP BETWEEN WINE PHYSIOCHEMICAL PROPERTIES AND WINE QUALITY

Corresponding to Chapter § 4 Section 4.3.2

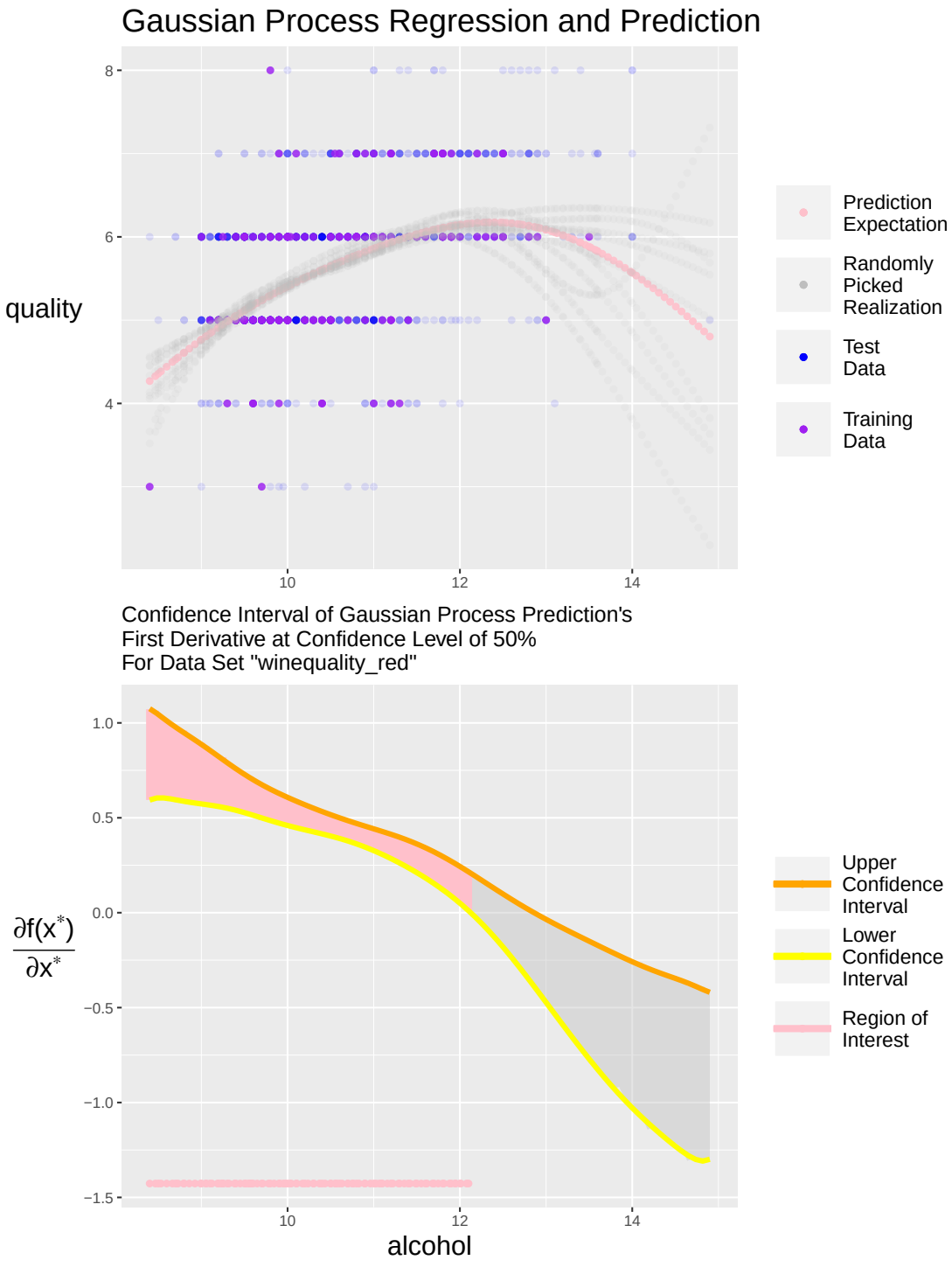


Figure D.1: Example 2: Estimate confidence interval of the relationship between alcohol and red wine quality

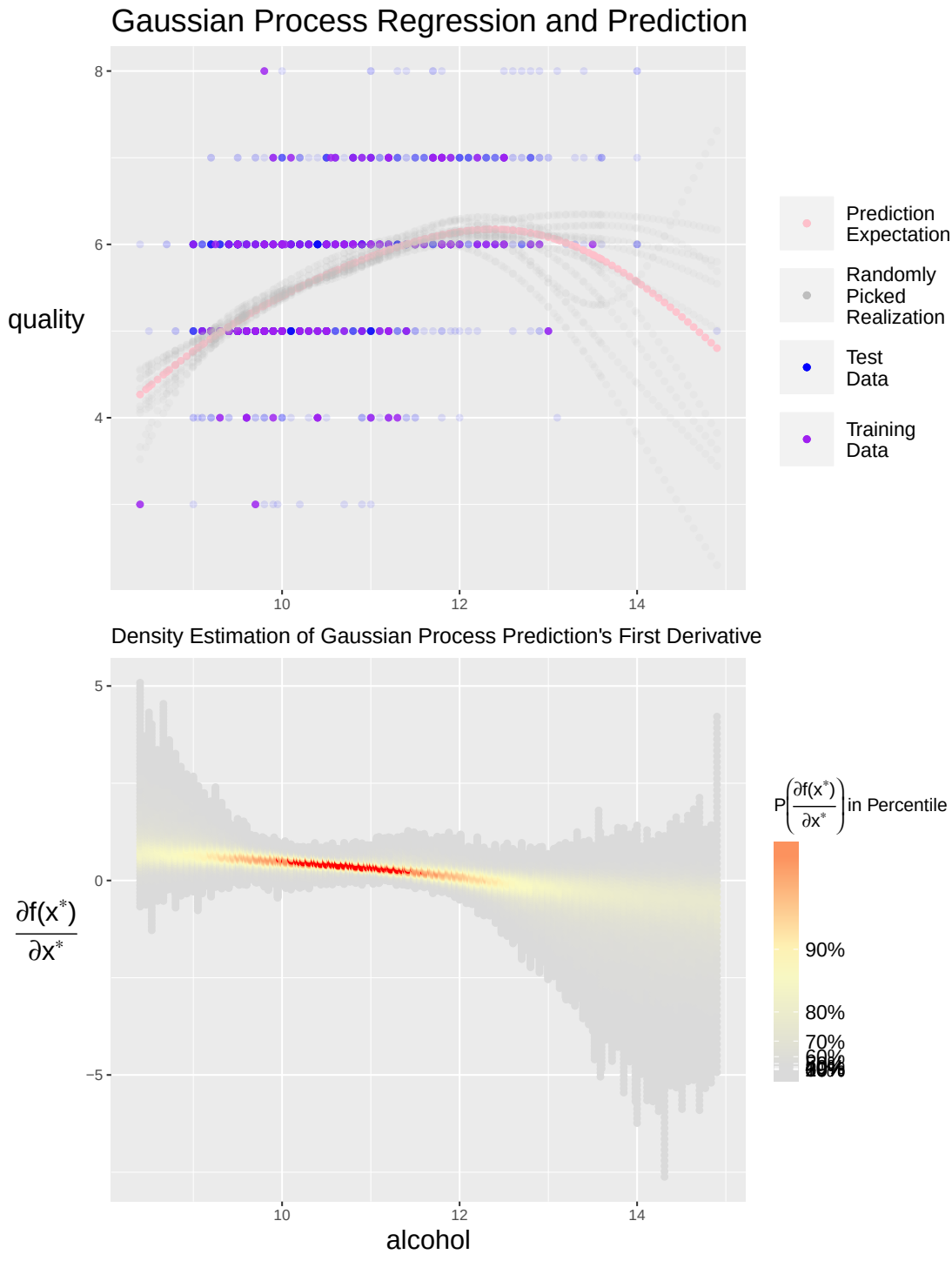


Figure D.2: Example 2: Estimate the relationship intensity between alcohol and red wine quality

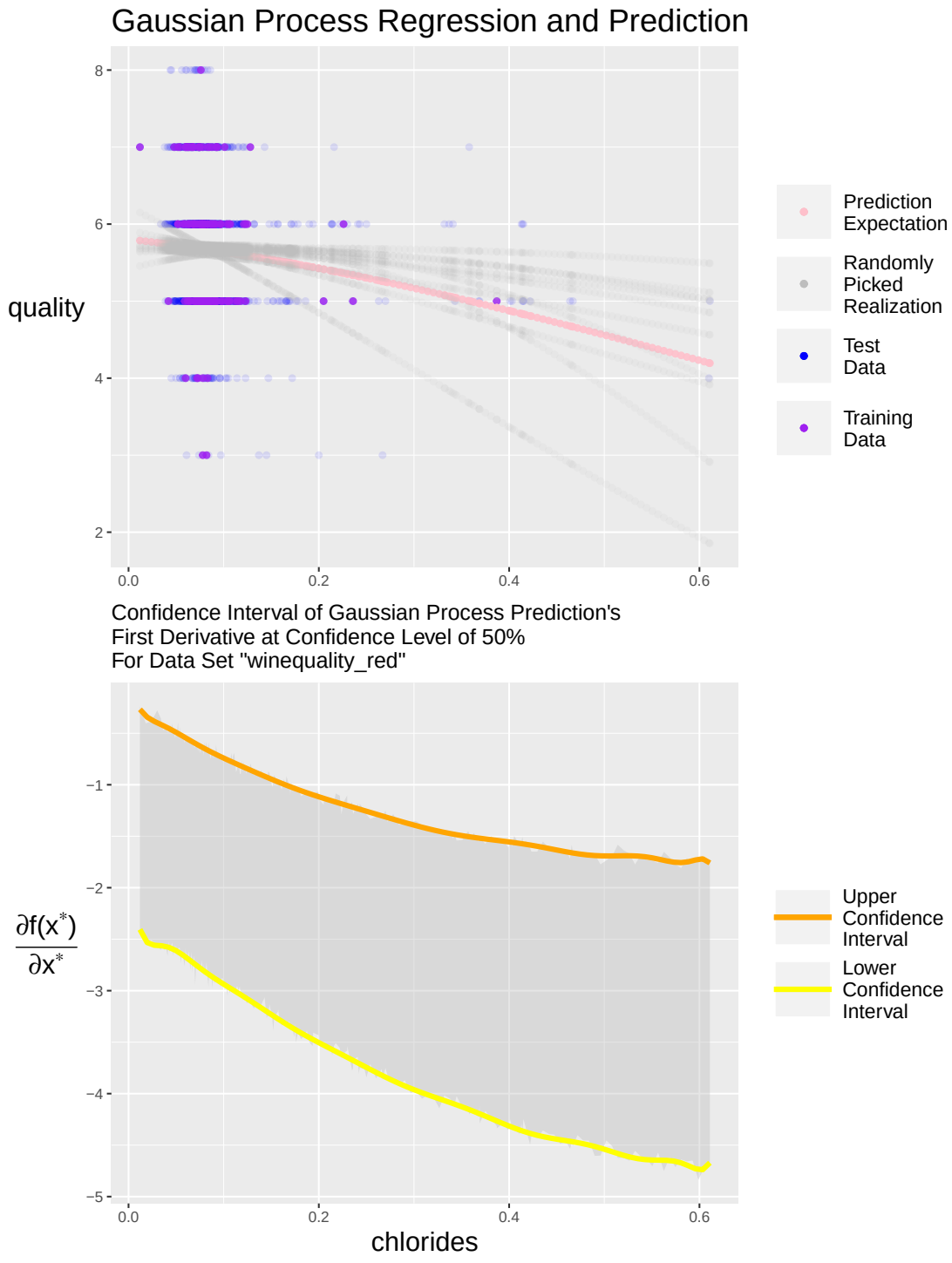


Figure D.3: Example 2: Estimate confidence interval of the relationship between chlorides and red wine quality

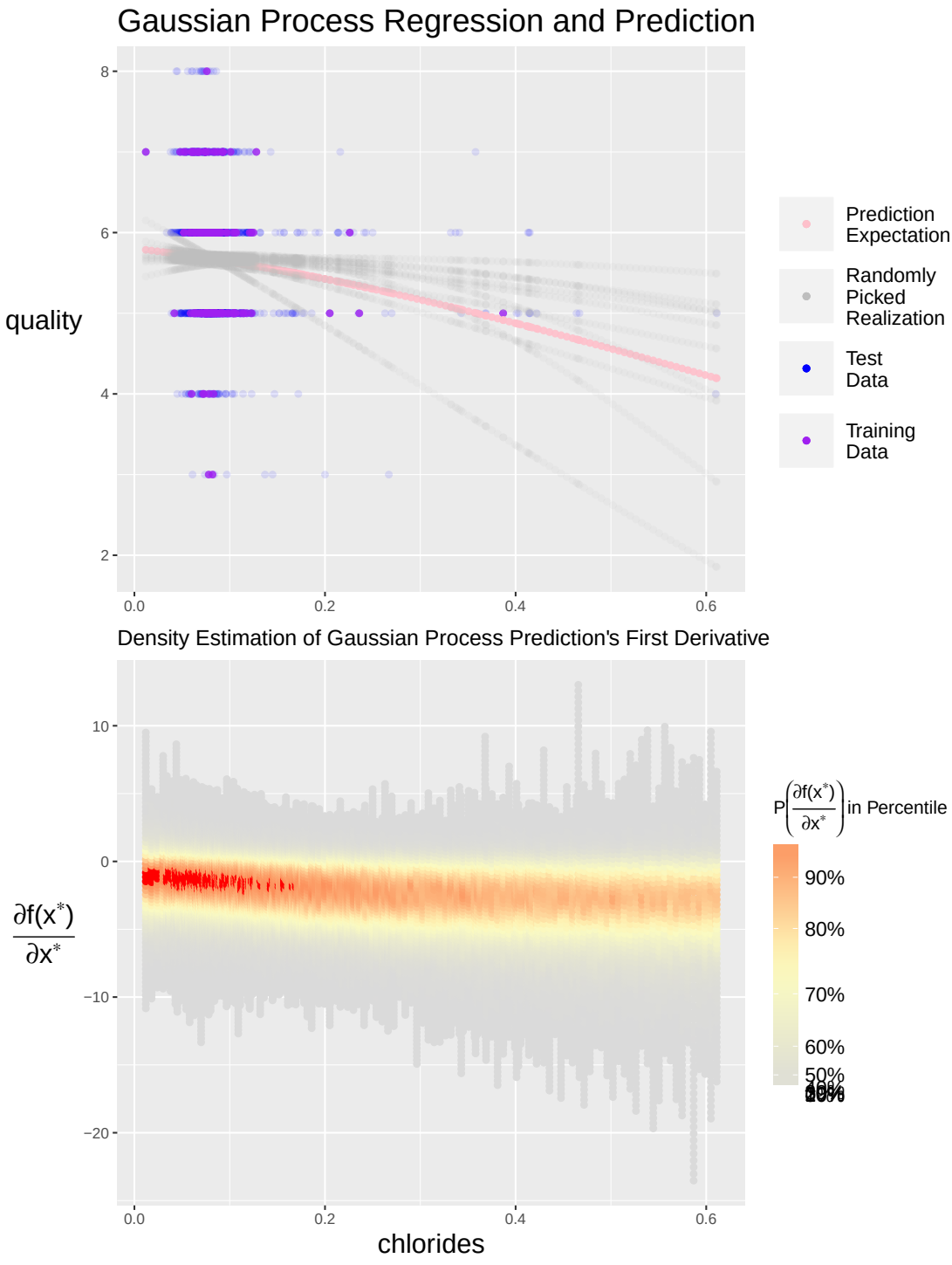


Figure D.4: Example 2: Estimate the relationship intensity between chlorides and red wine quality

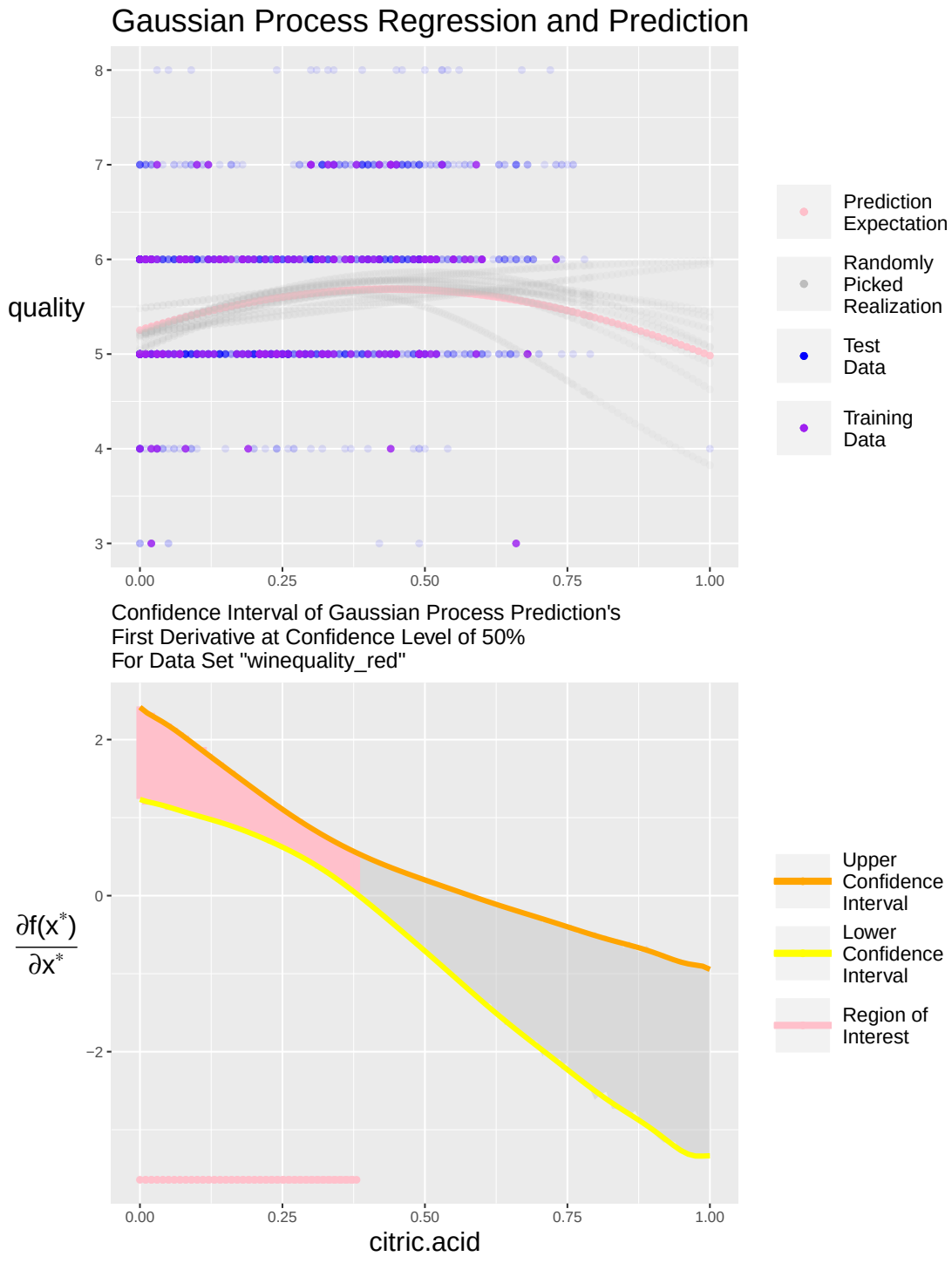


Figure D.5: Example 2: Estimate confidence interval of the relationship between citric and red wine quality

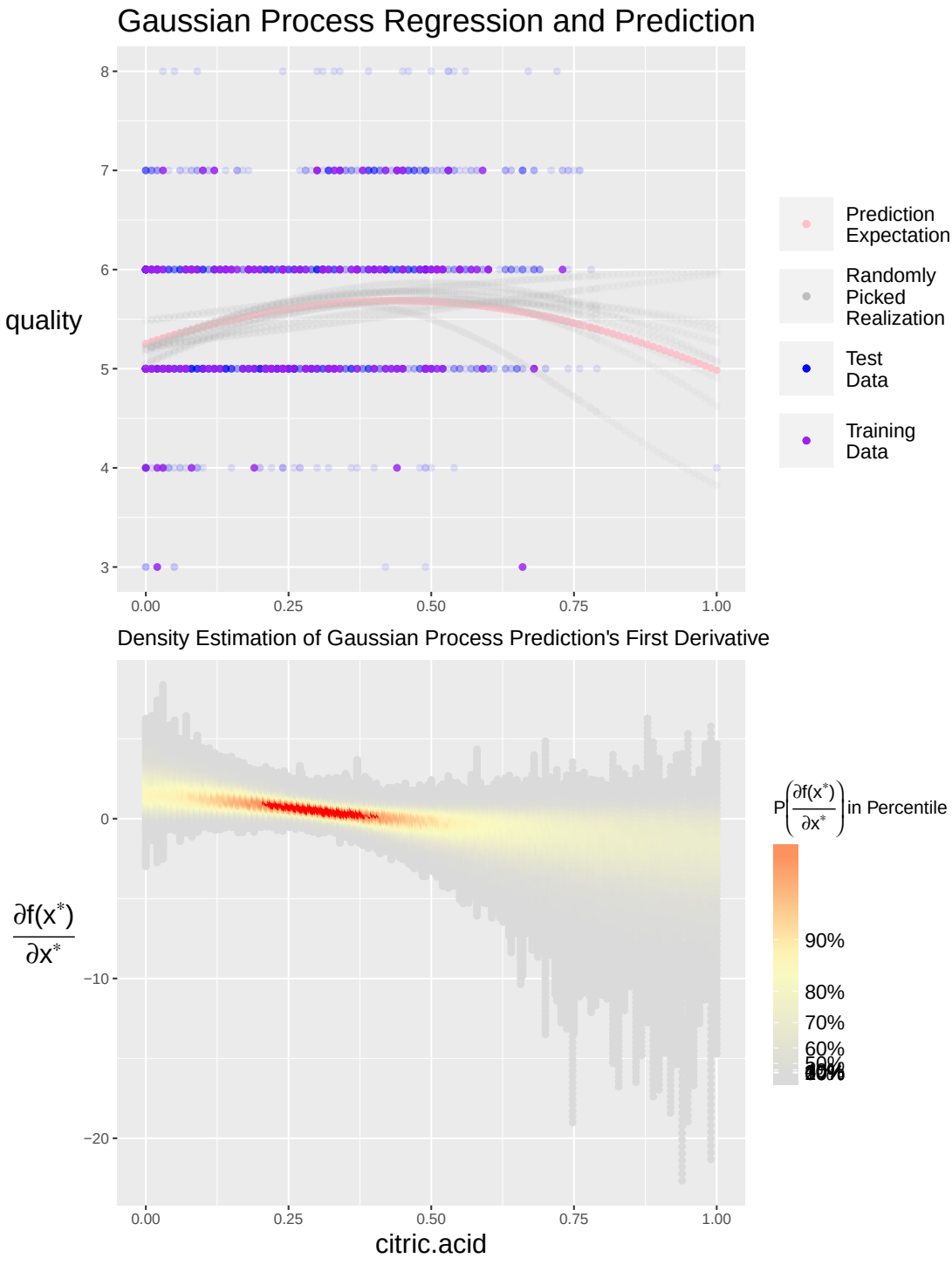


Figure D.6: Example 2: Estimate the relationship intensity between citric acid and red wine quality

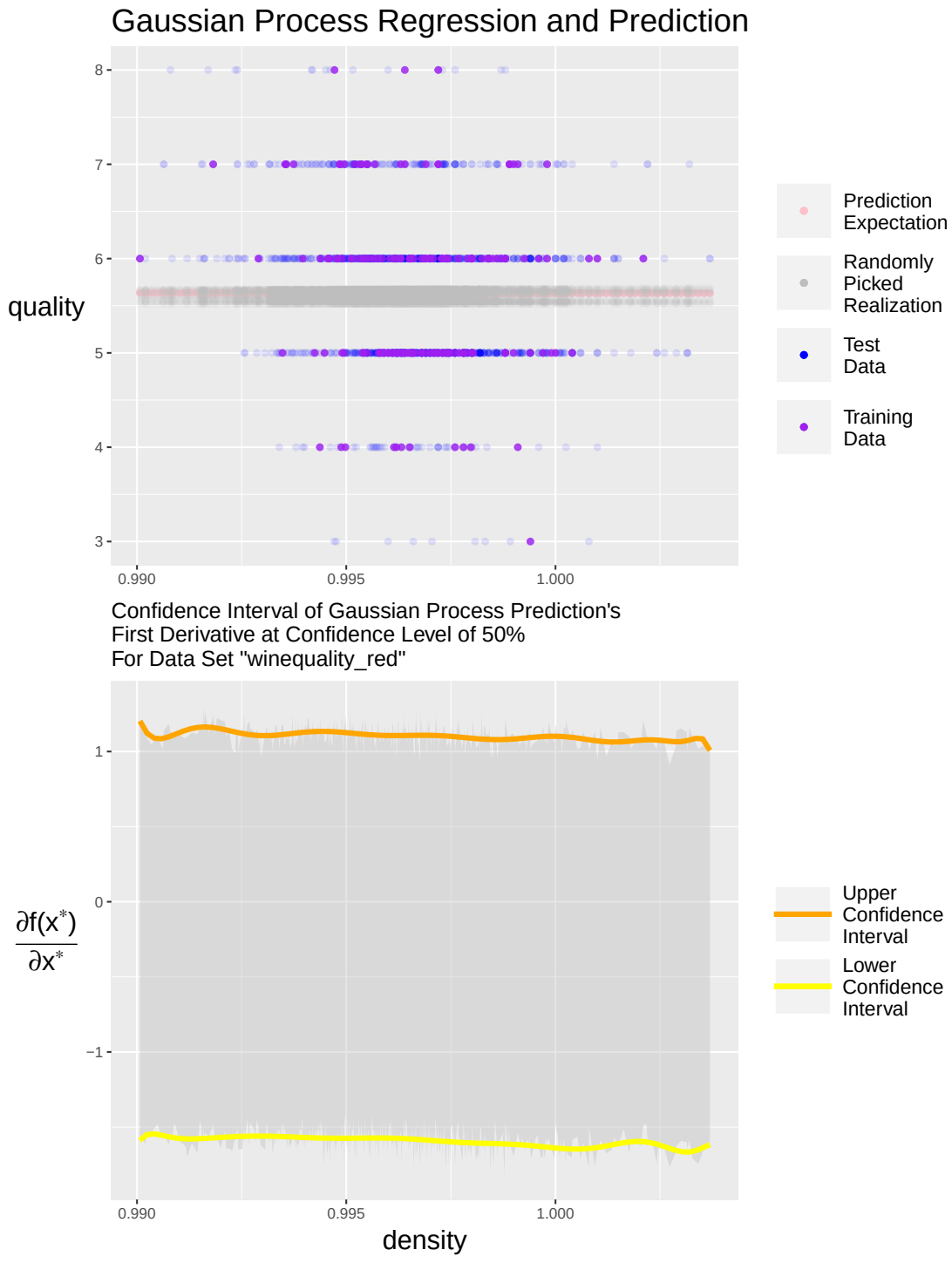


Figure D.7: Example 2: Estimate confidence interval of the relationship between density and red wine quality

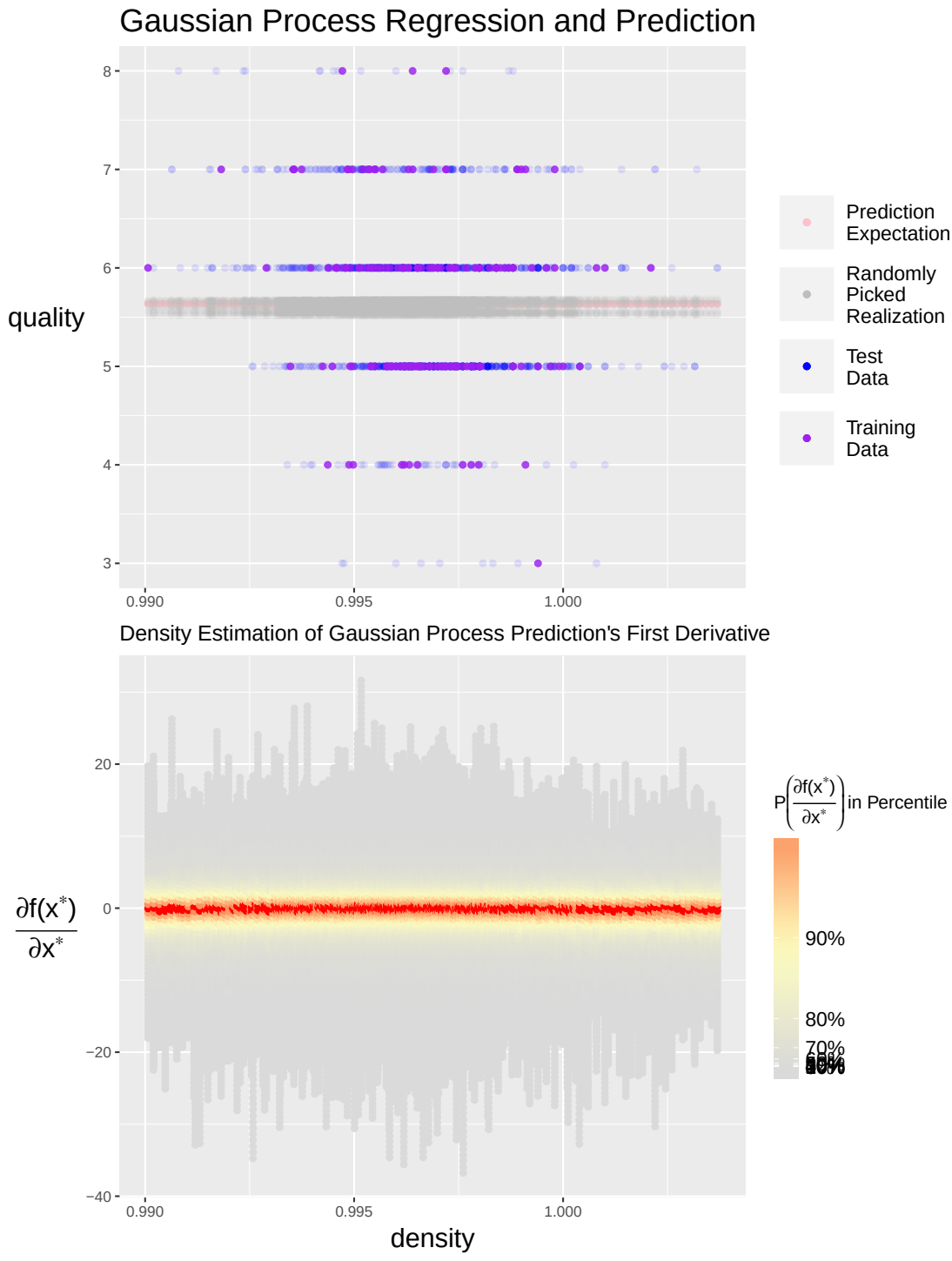


Figure D.8: Example 2: Estimate the relationship intensity between density and red wine quality

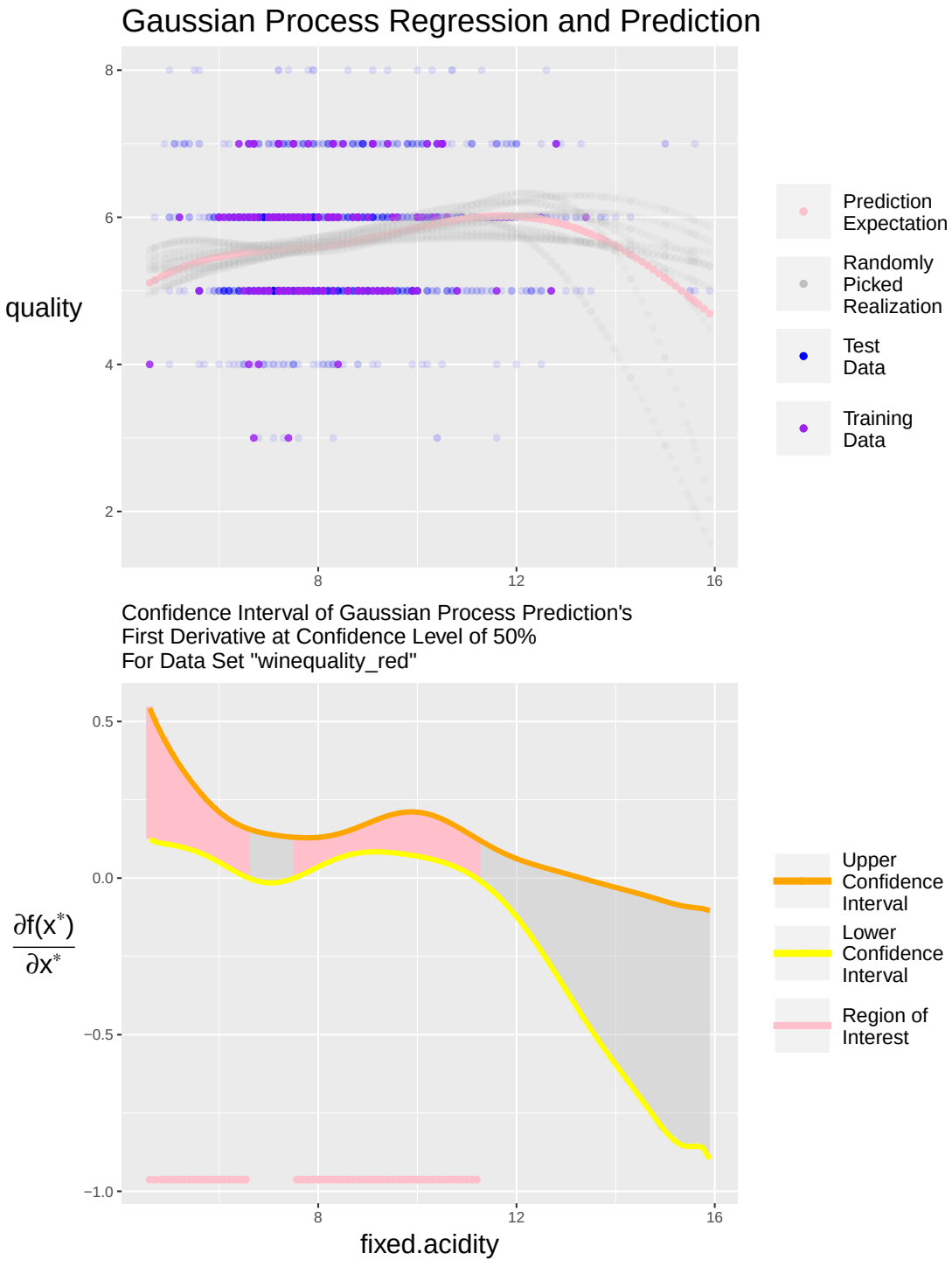


Figure D.9: Example 2: Estimate confidence interval of the relationship between fixed acidity and red wine quality

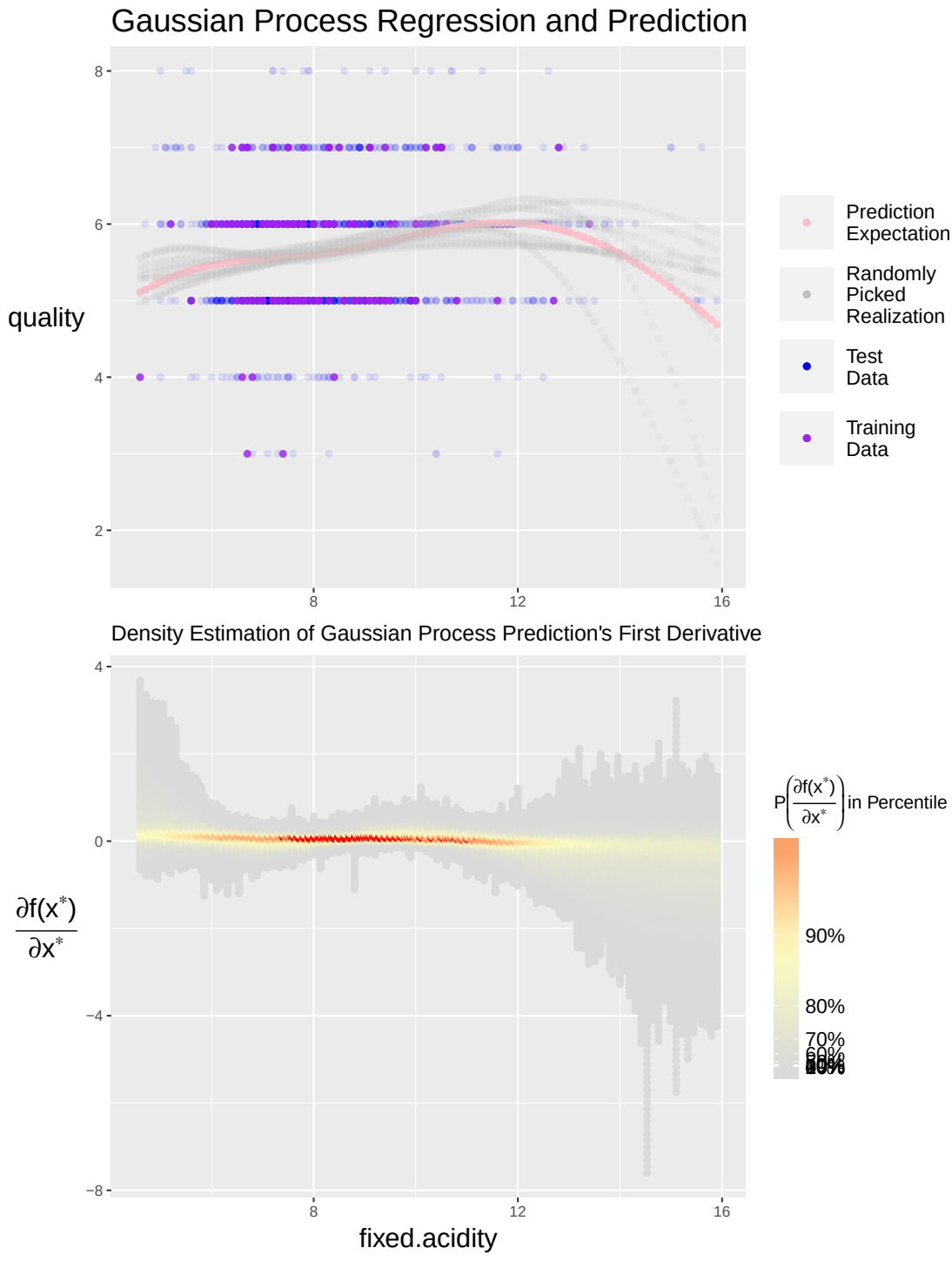


Figure D.10: Example 2: Estimate the relationship intensity between fixed acidity and red wine quality

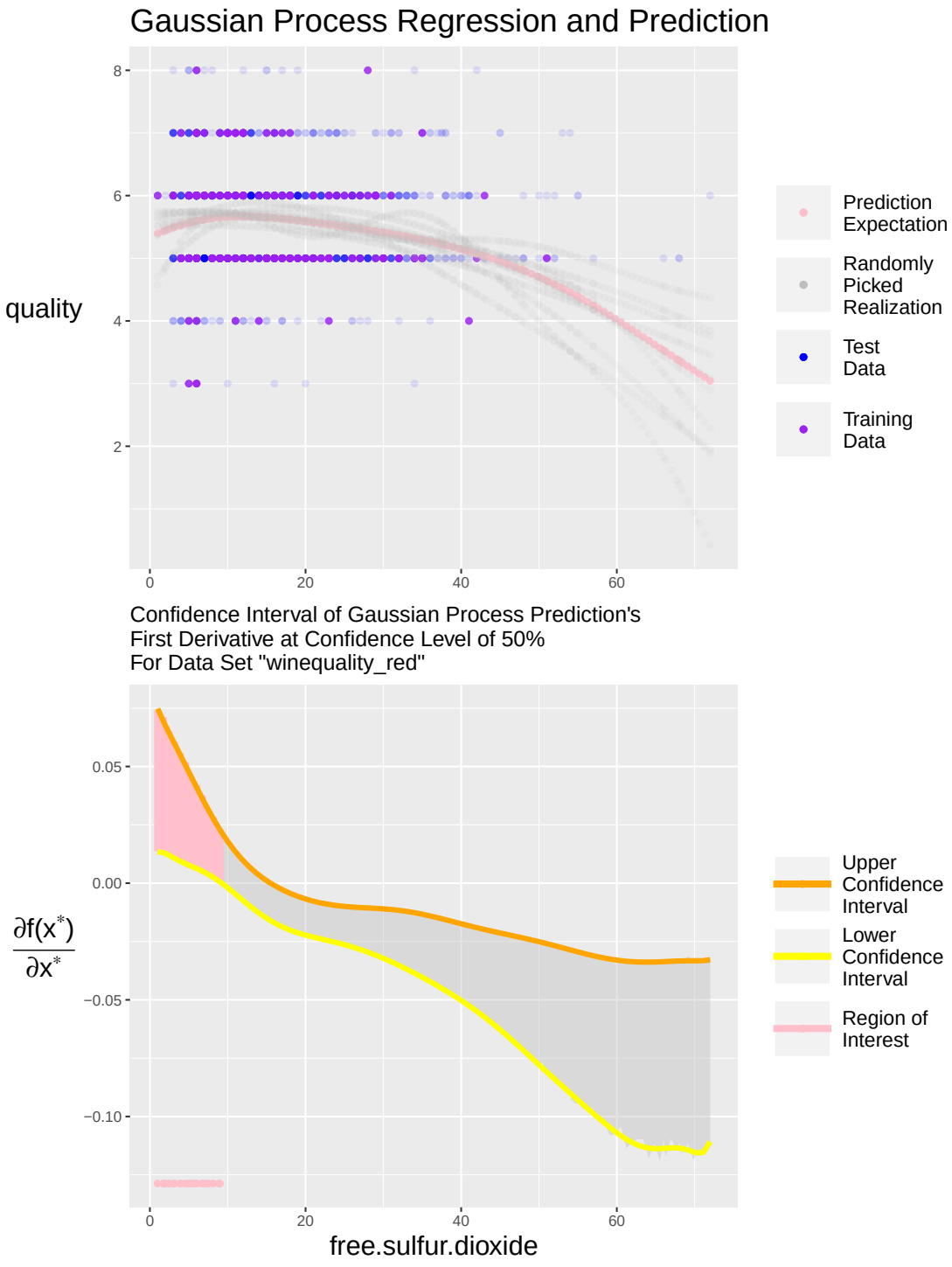


Figure D.11: Example 2: Estimate confidence interval of the relationship between free sulfur dioxide and red wine quality

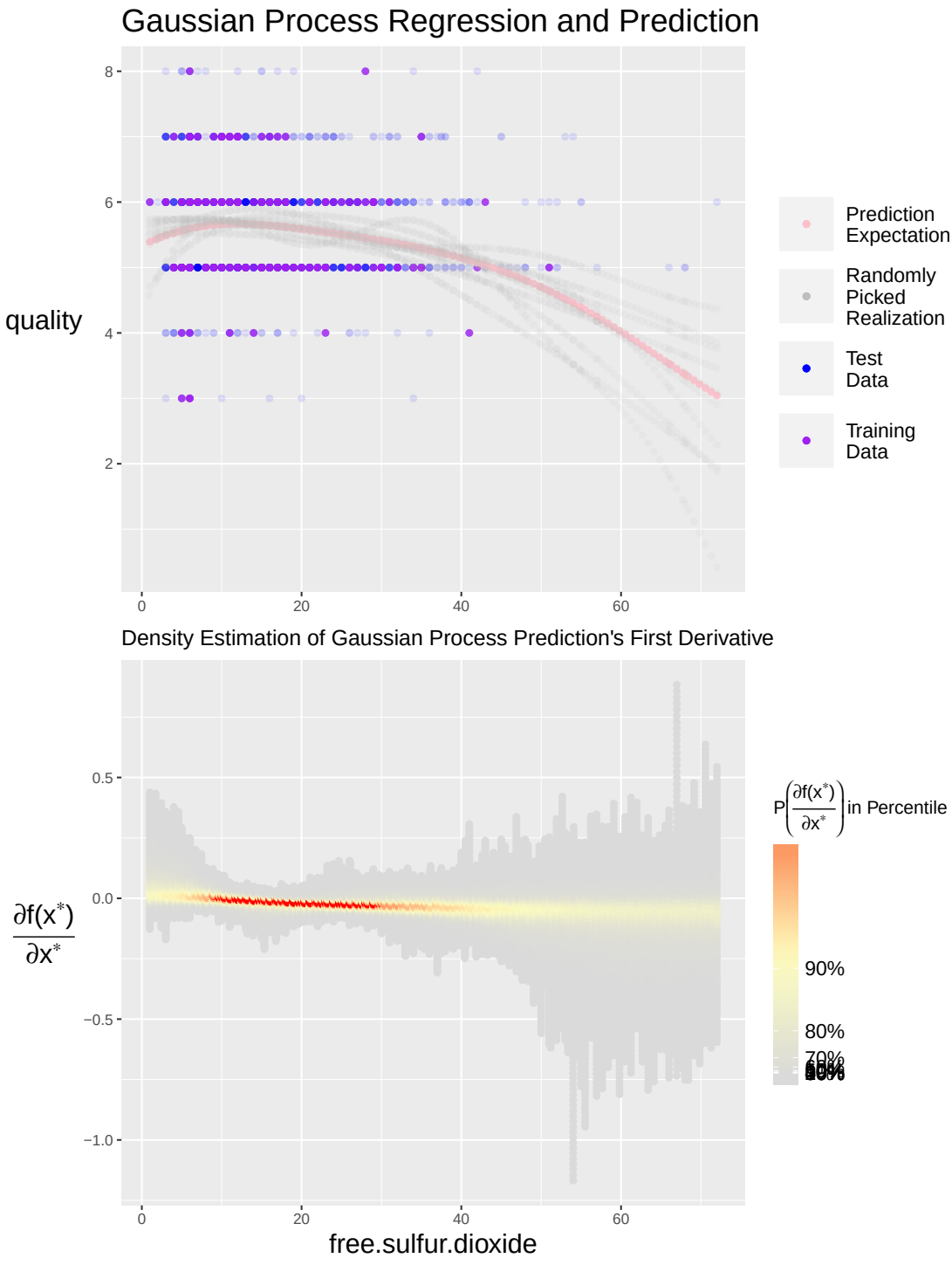


Figure D.12: Example 2: Estimate the relationship intensity between free sulfur dioxide and red wine quality

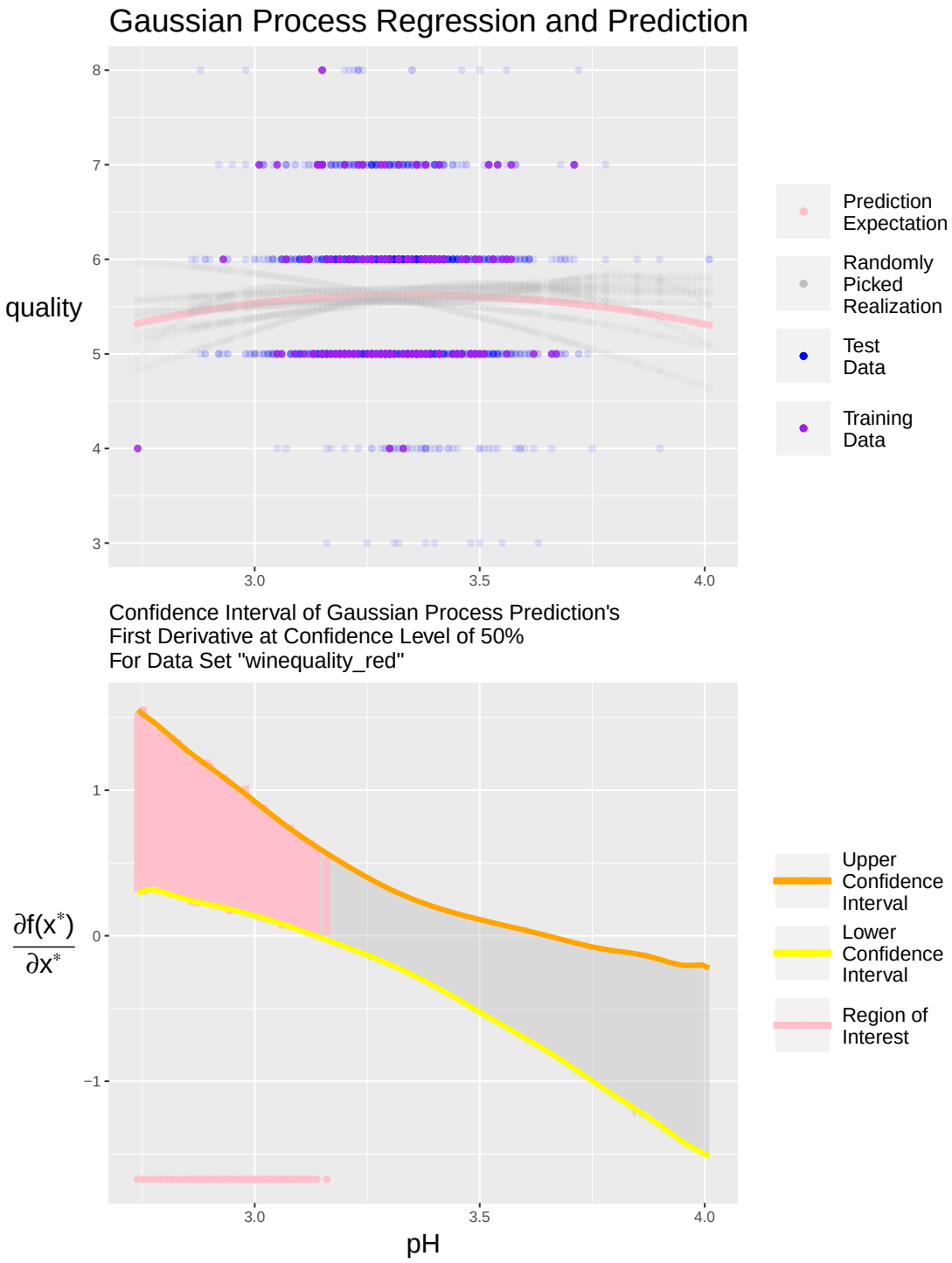


Figure D.13: Example 2: Estimate confidence interval of the relationship between pH and red wine quality

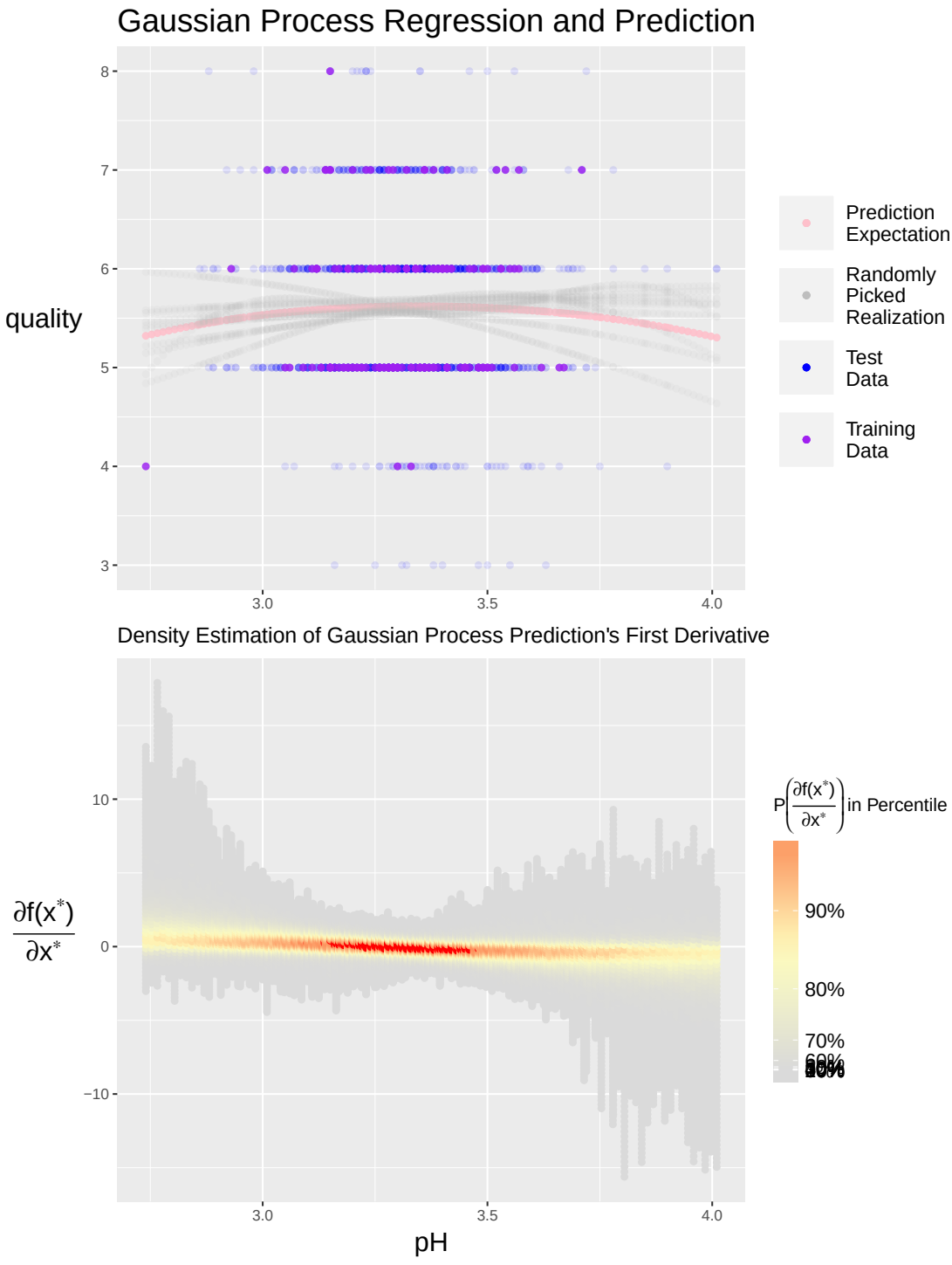


Figure D.14: Example 2: Estimate the relationship intensity between pH and red wine quality

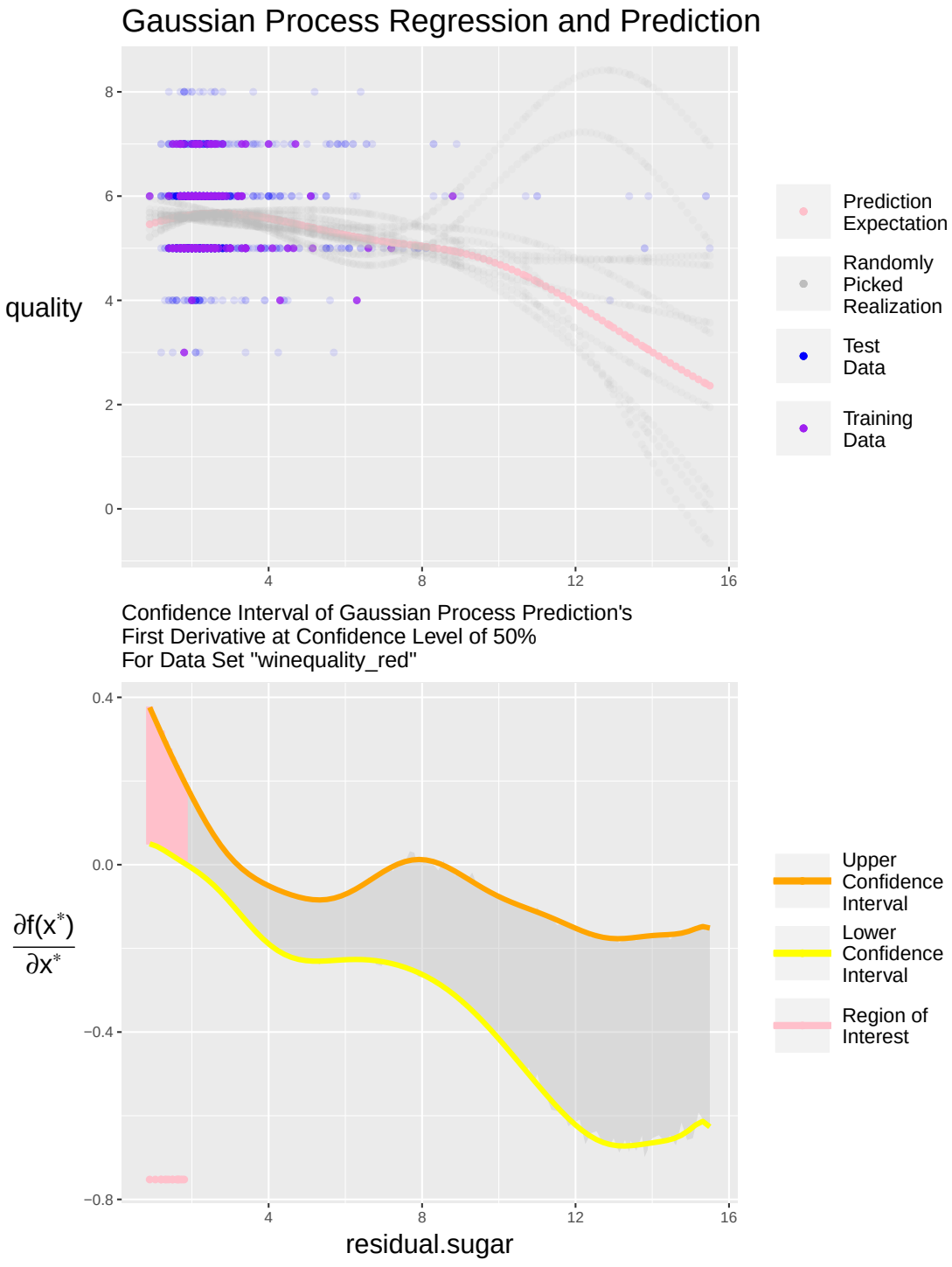


Figure D.15: Example 2: Estimate confidence interval of the relationship between residual sugar and red wine quality

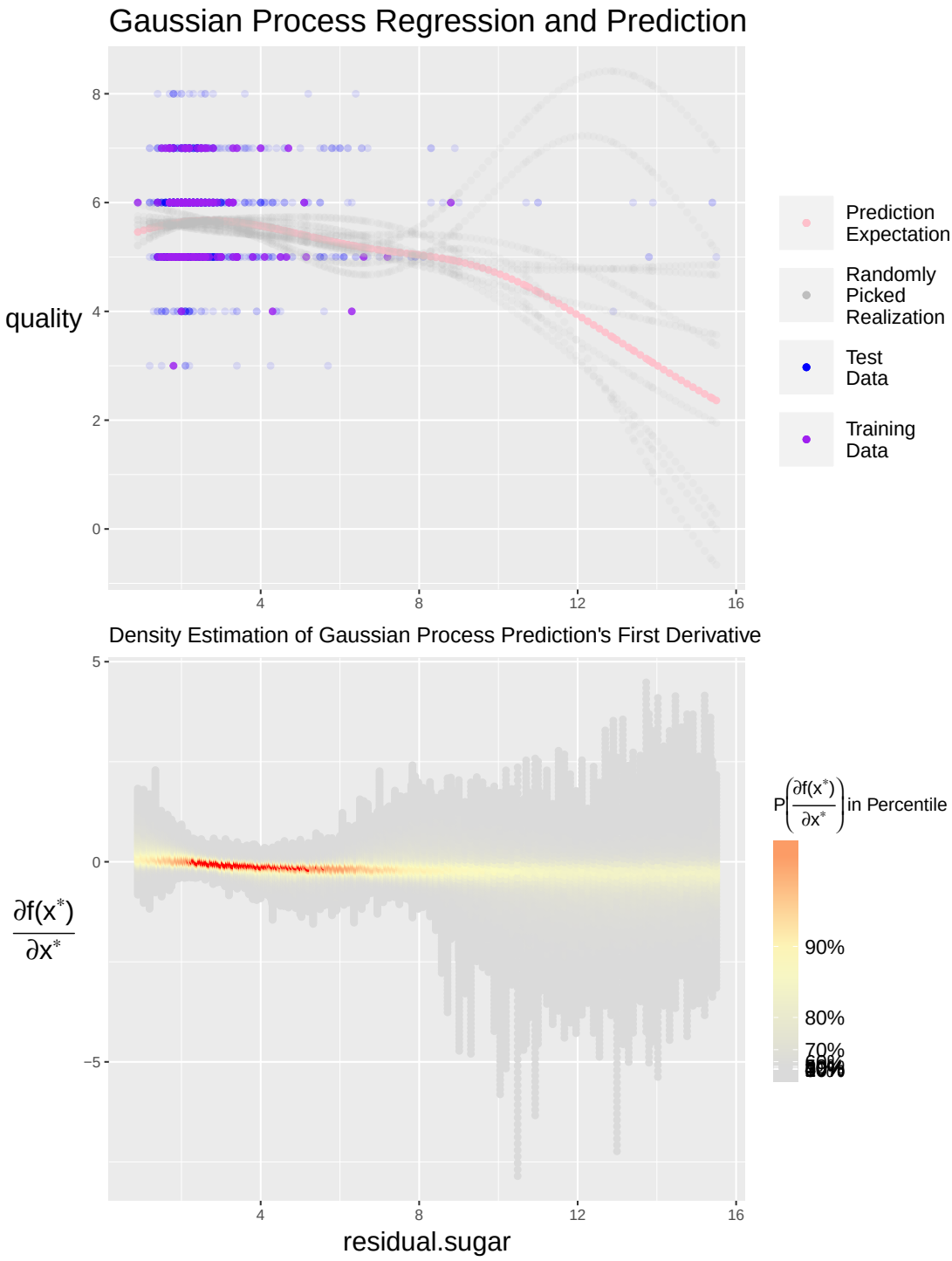


Figure D.16: Example 2: Estimate the relationship intensity between residual sugar and red wine quality

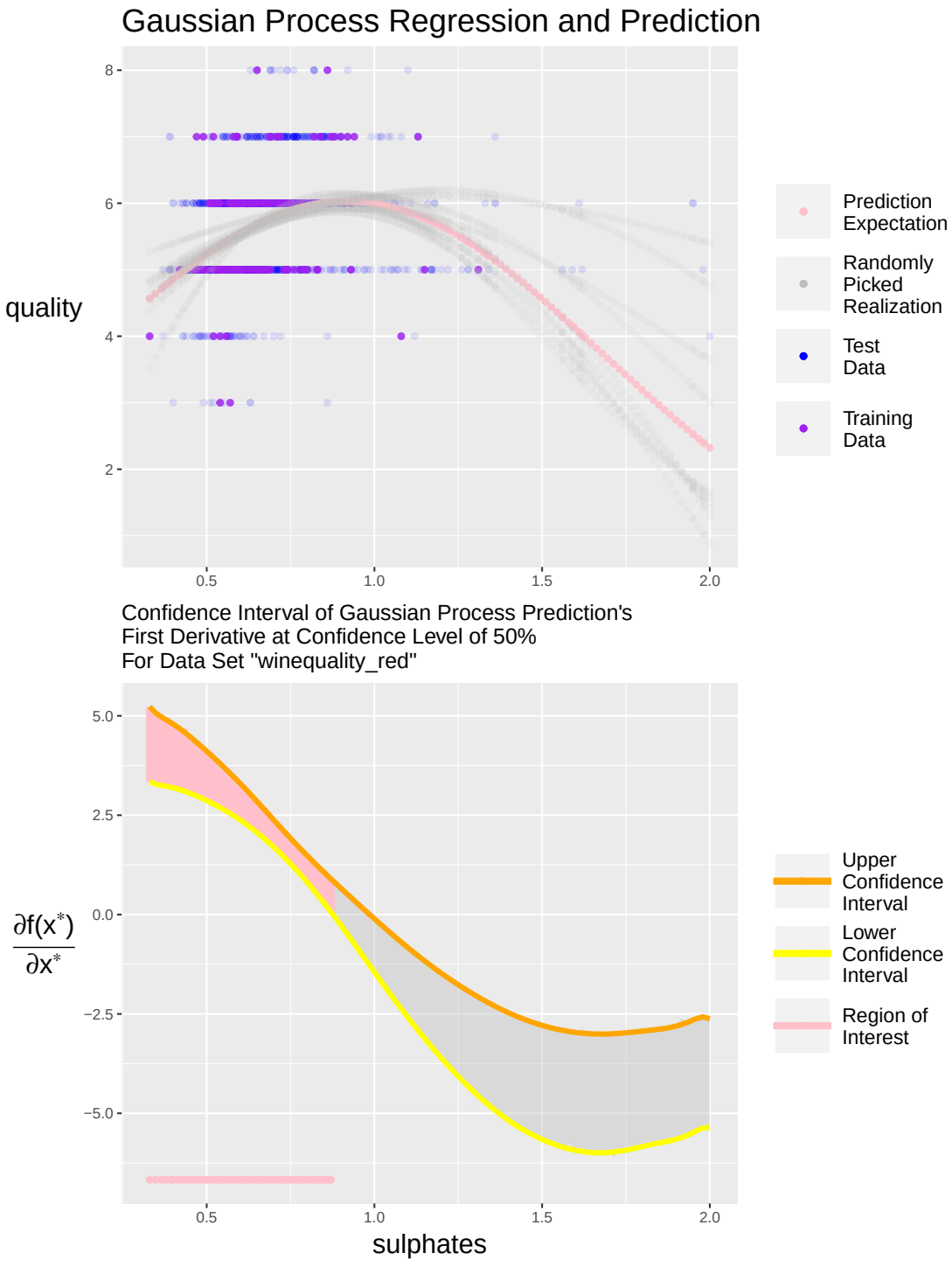


Figure D.17: Example 2: Estimate confidence interval of the relationship between sulphates and red wine quality

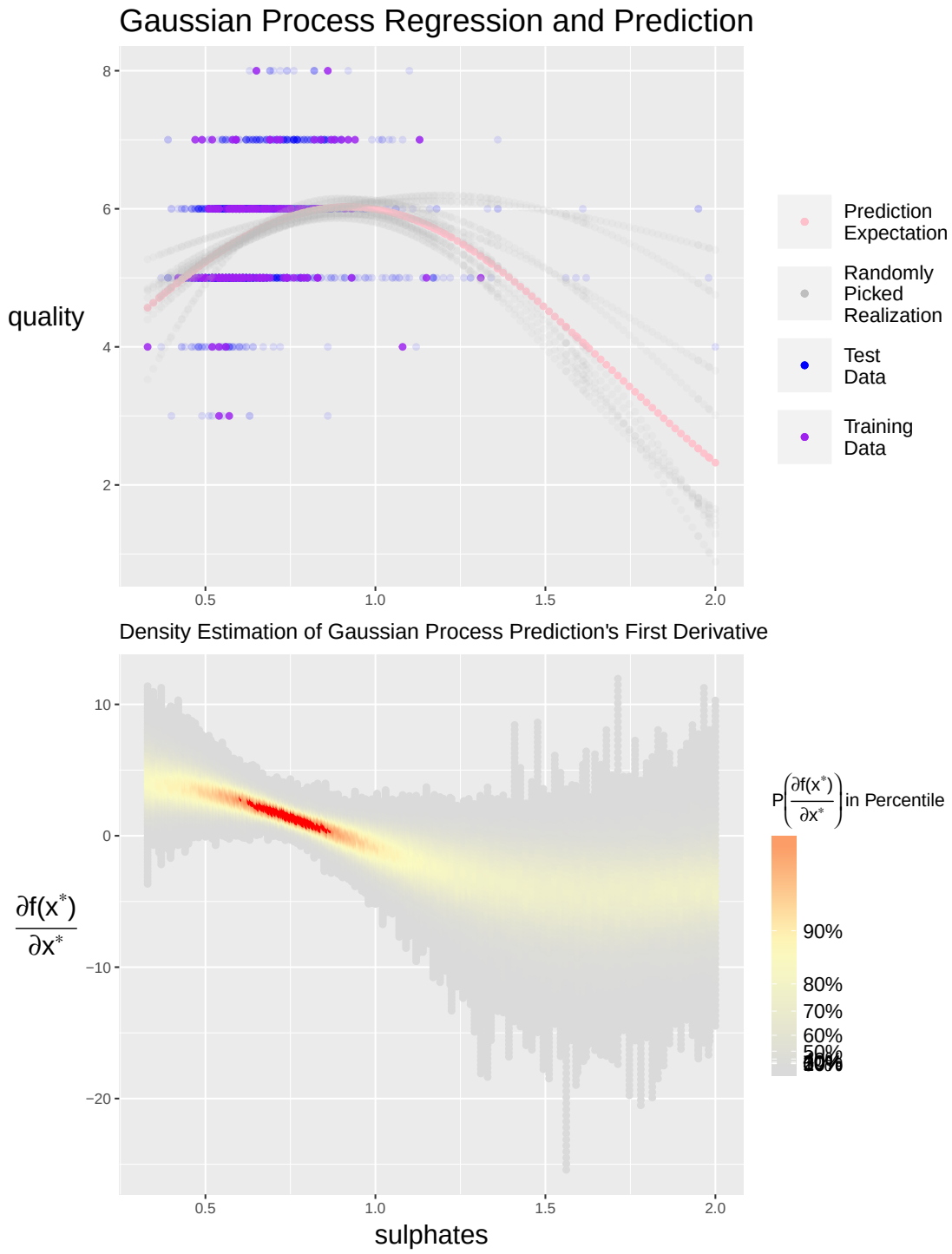


Figure D.18: Example 2: Estimate the relationship intensity between sulphates and red wine quality

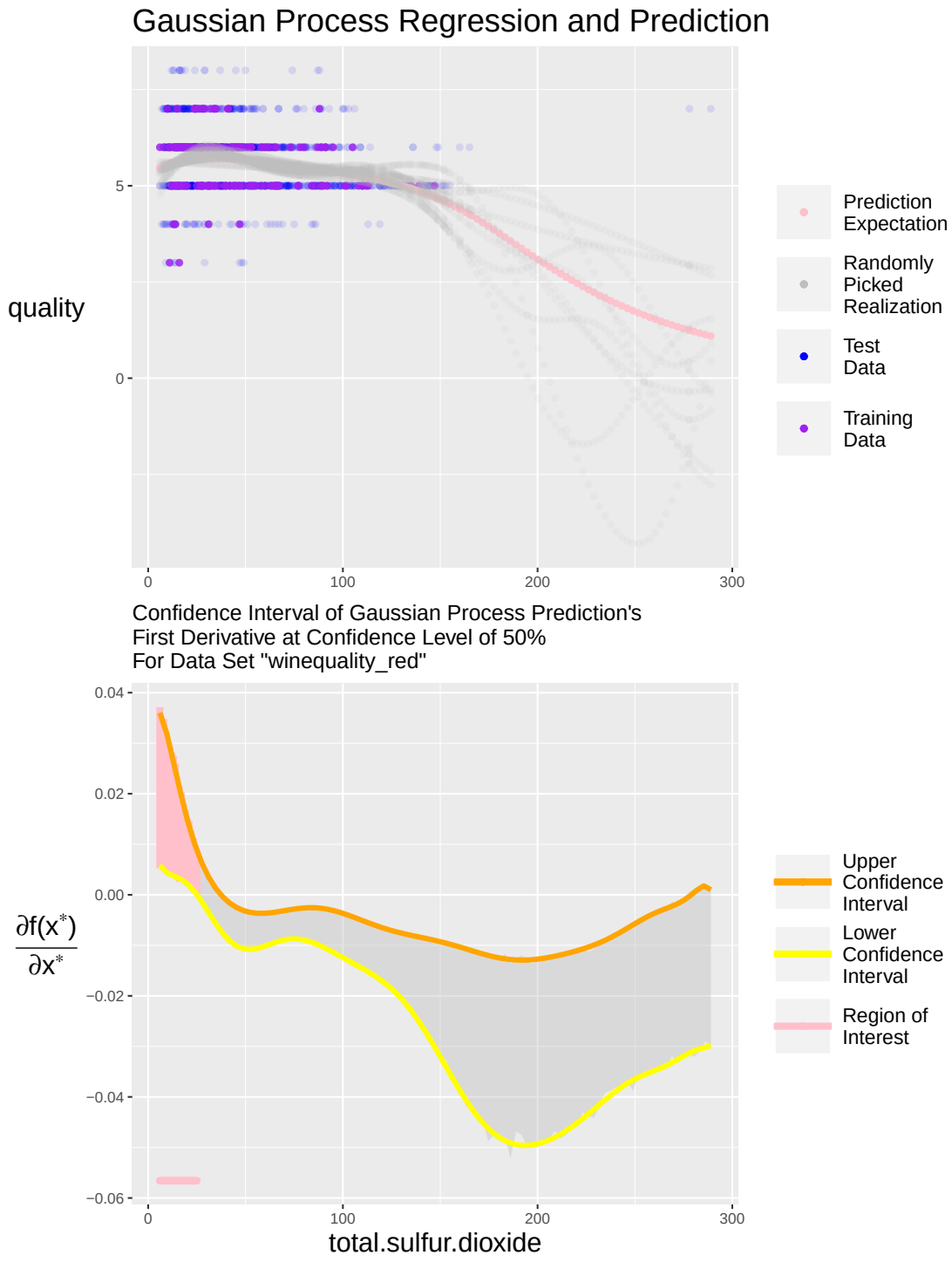


Figure D.19: Example 2: Estimate confidence interval of the relationship between total sulfur dioxide and red wine quality

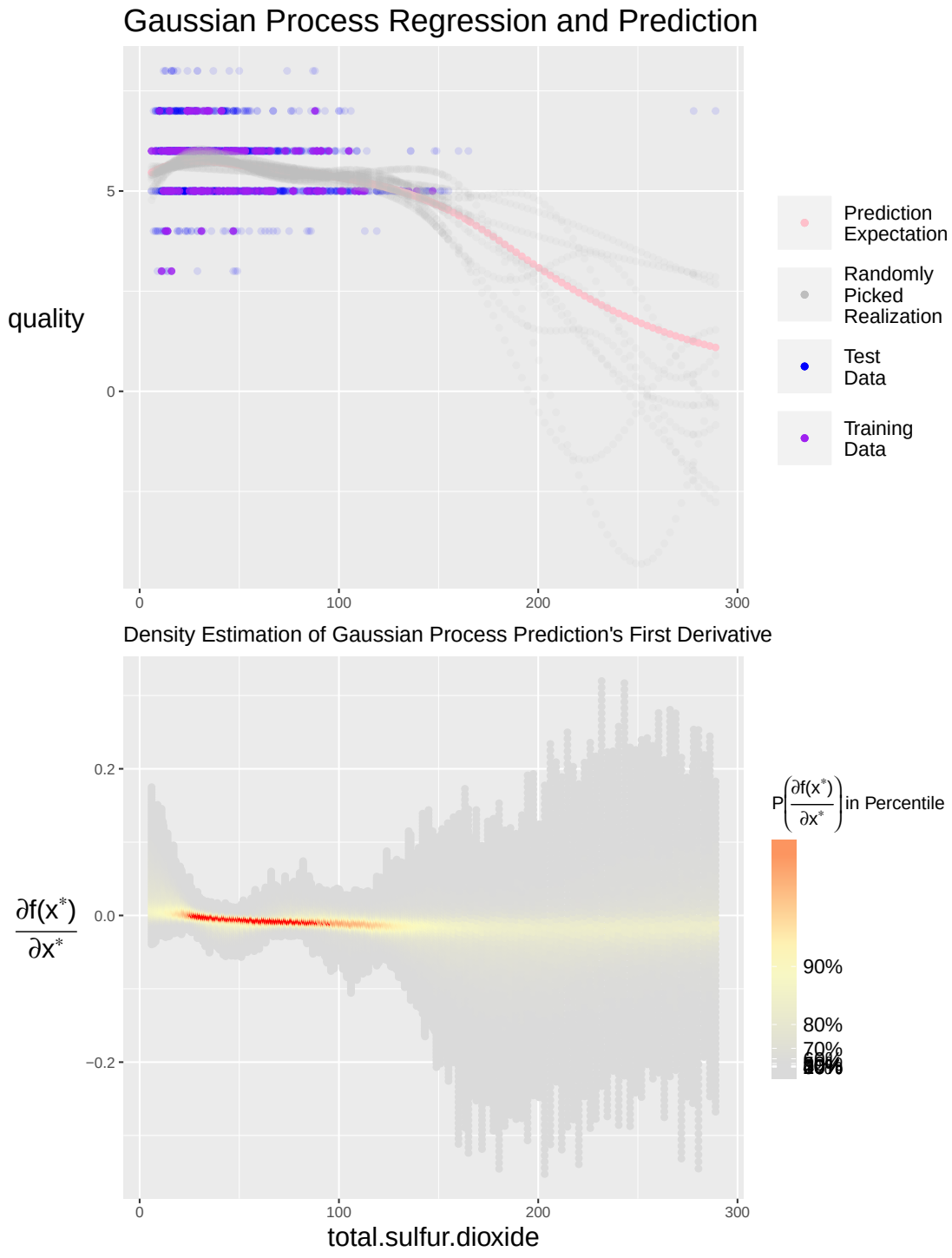


Figure D.20: Example 2: Estimate the relationship intensity between total sulfur dioxide and red wine quality

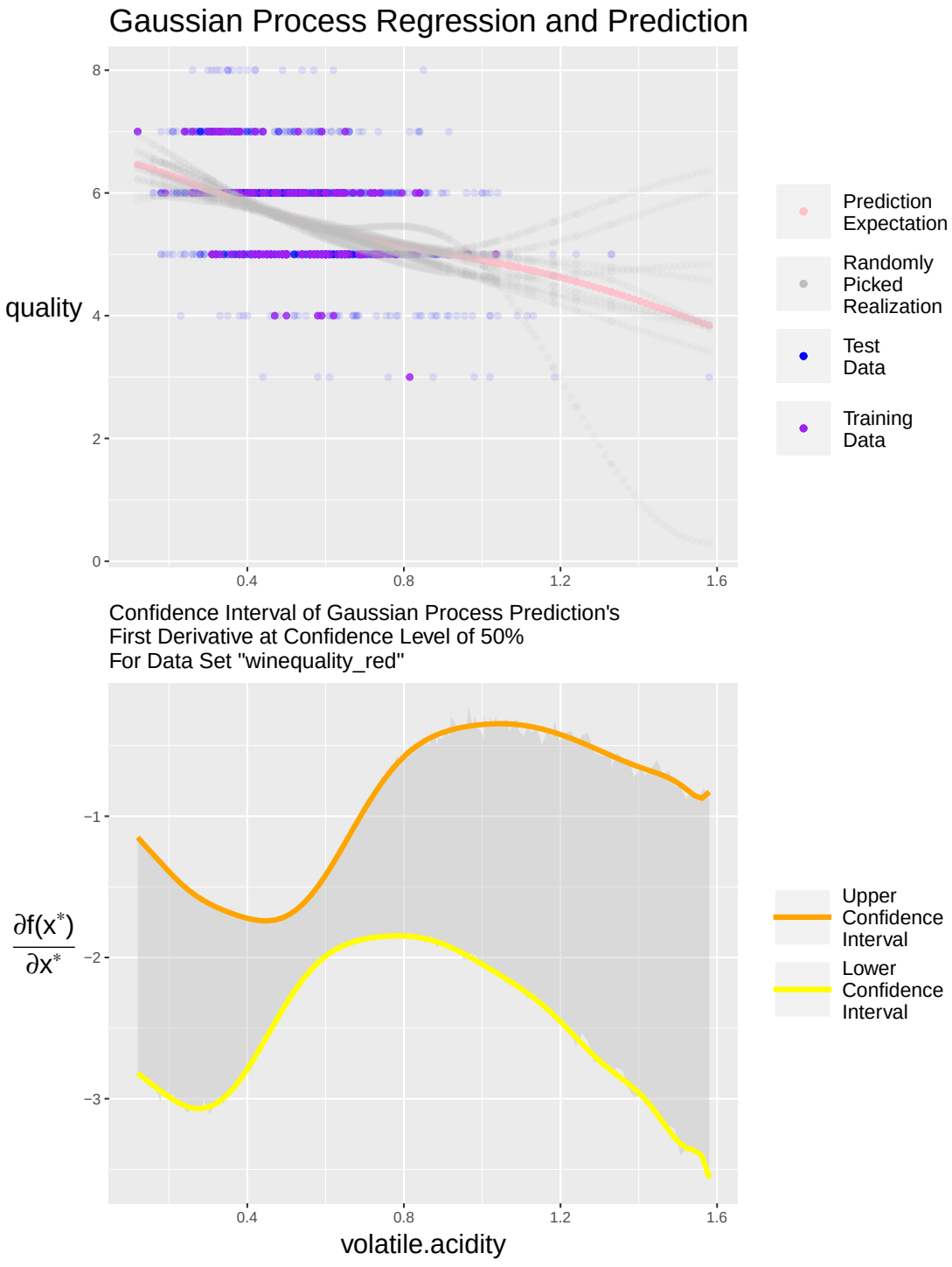


Figure D.21: Example 2: Estimate confidence interval of the relationship between volatile acidity and red wine quality

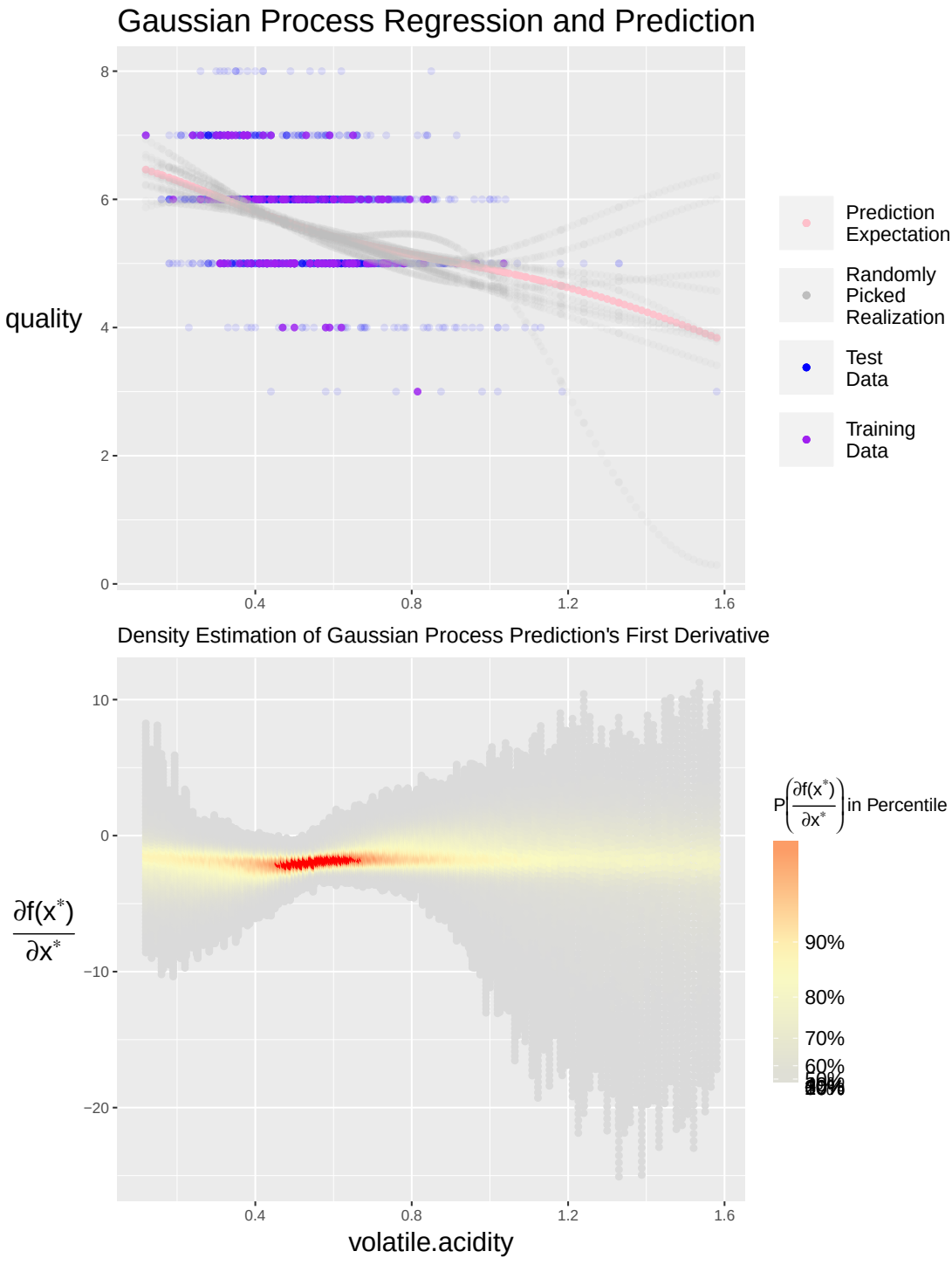


Figure D.22: Example 2: Estimate the relationship intensity between volatile acidity and red wine quality

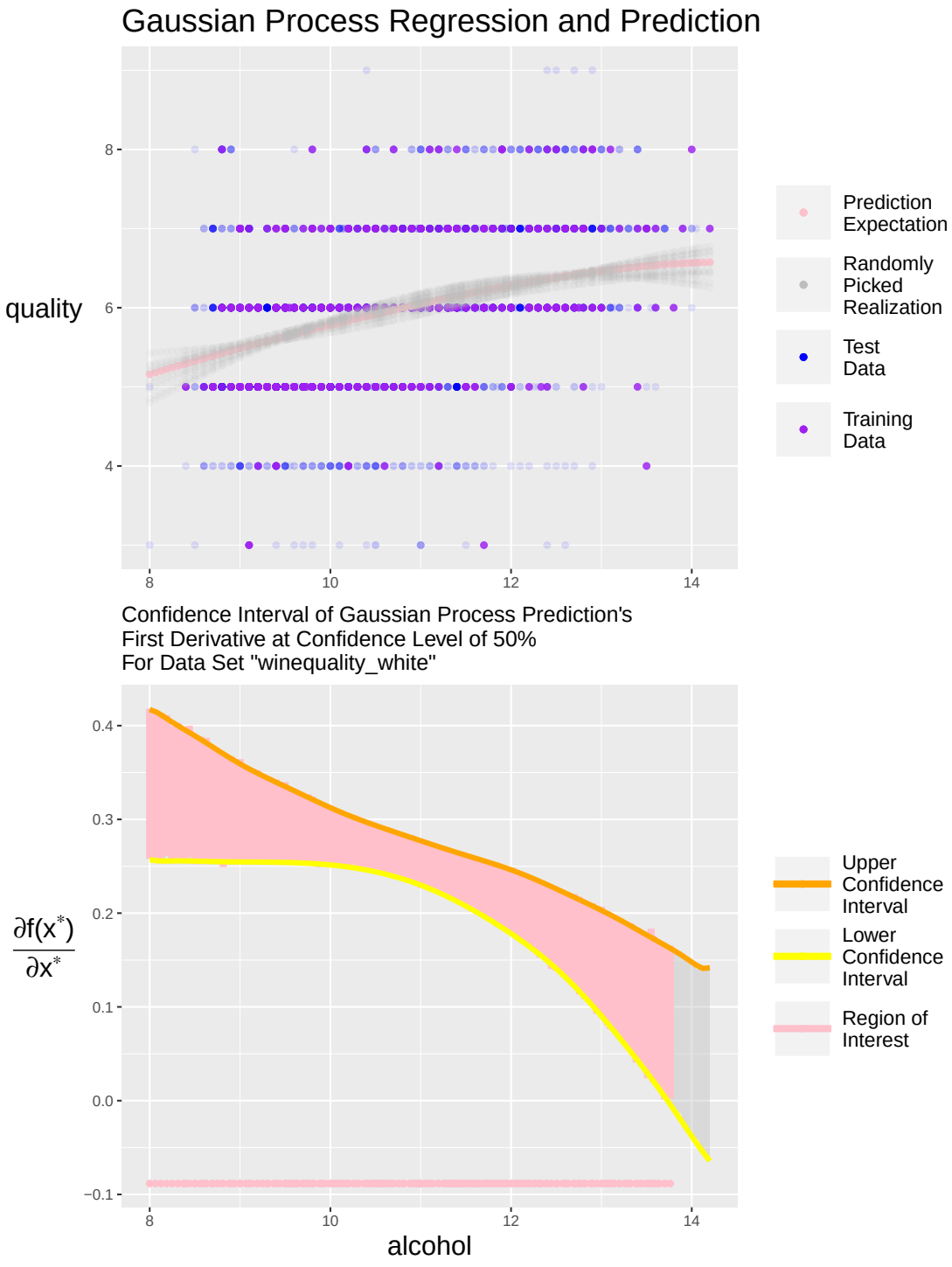


Figure D.23: Example 2: Estimate confidence interval of the relationship between alcohol and white wine quality

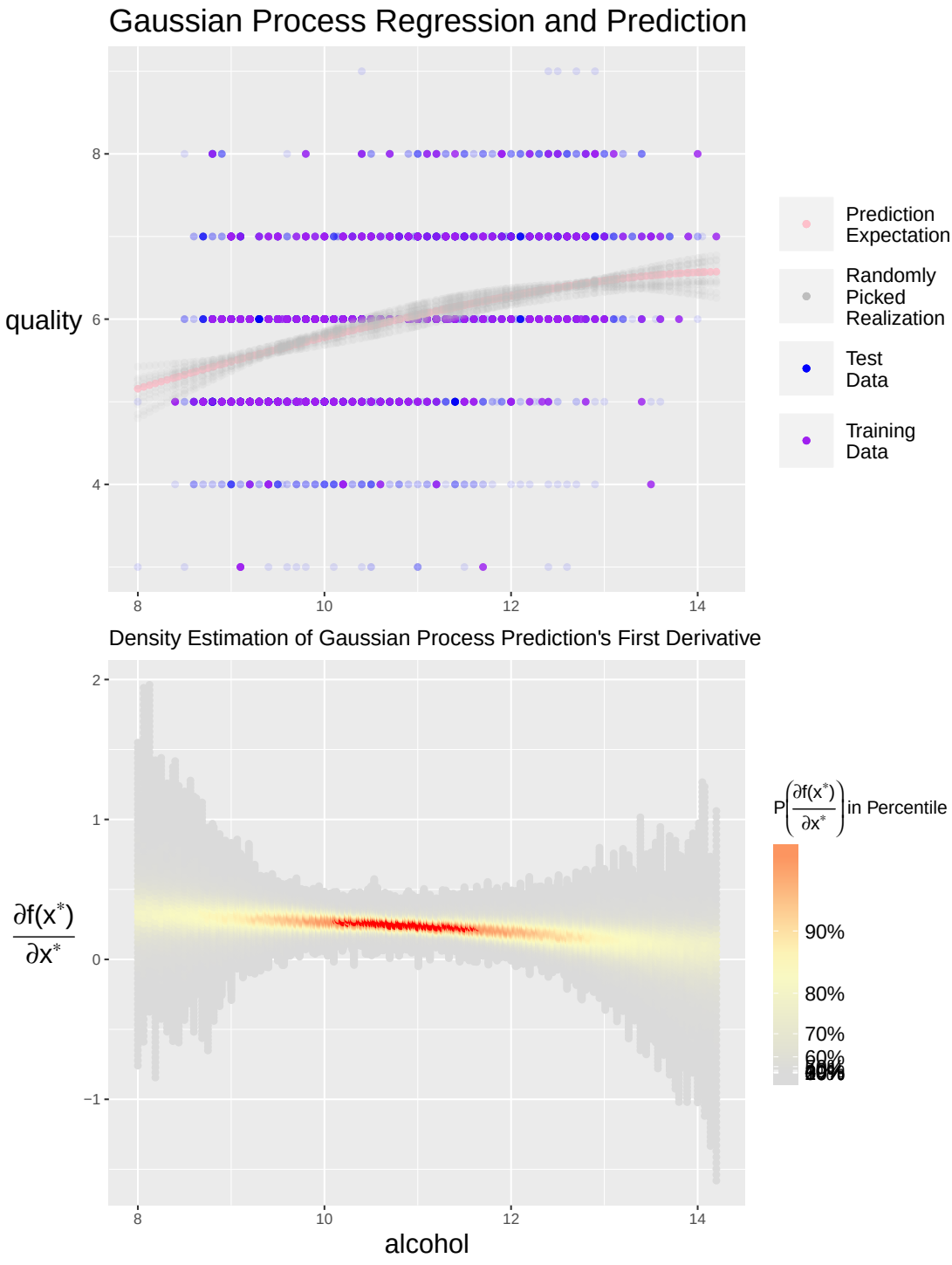


Figure D.24: Example 2: Estimate the relationship intensity between alcohol and white wine quality

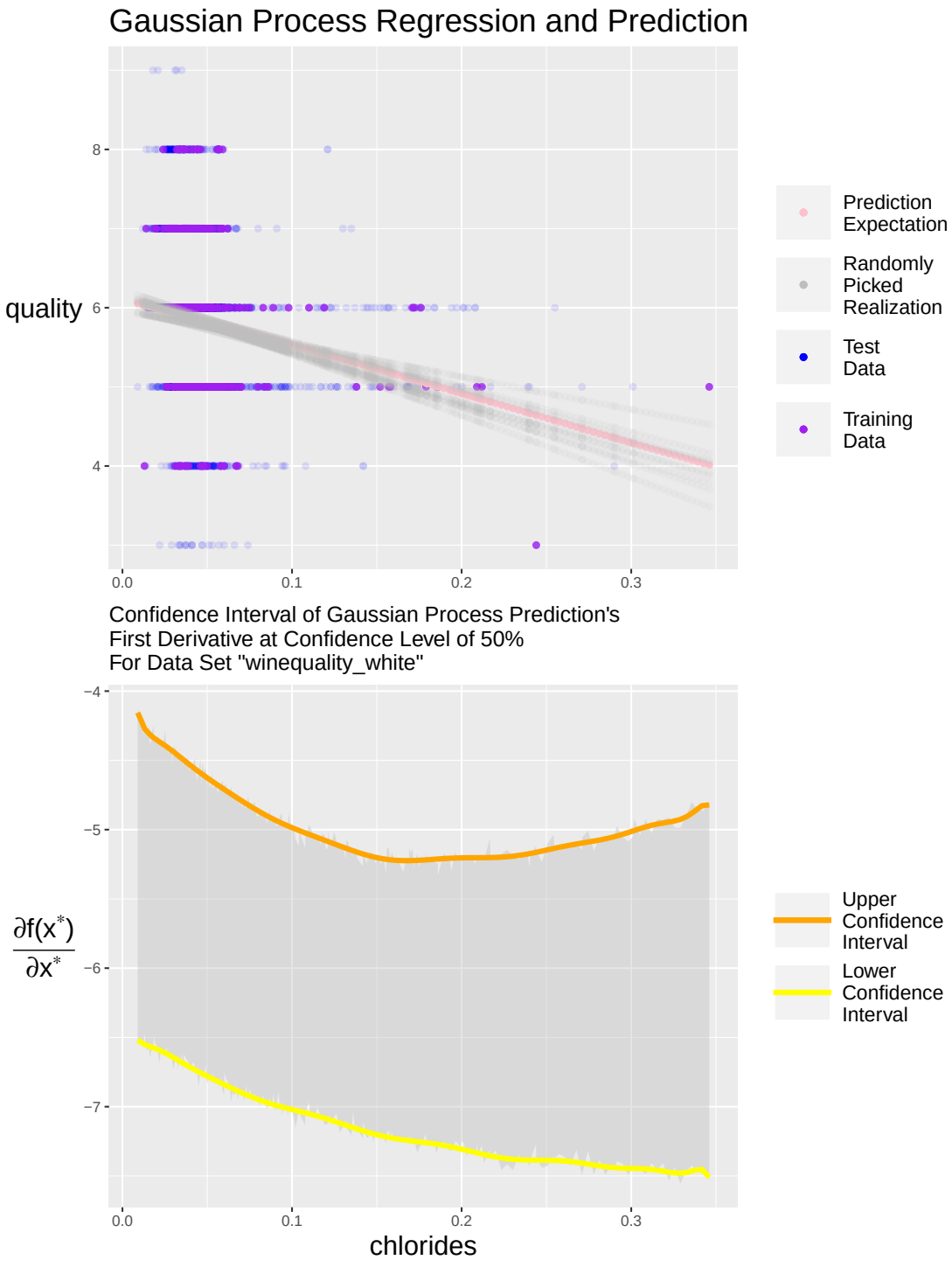


Figure D.25: Example 2: Estimate confidence interval of the relationship between chlorides and white wine quality

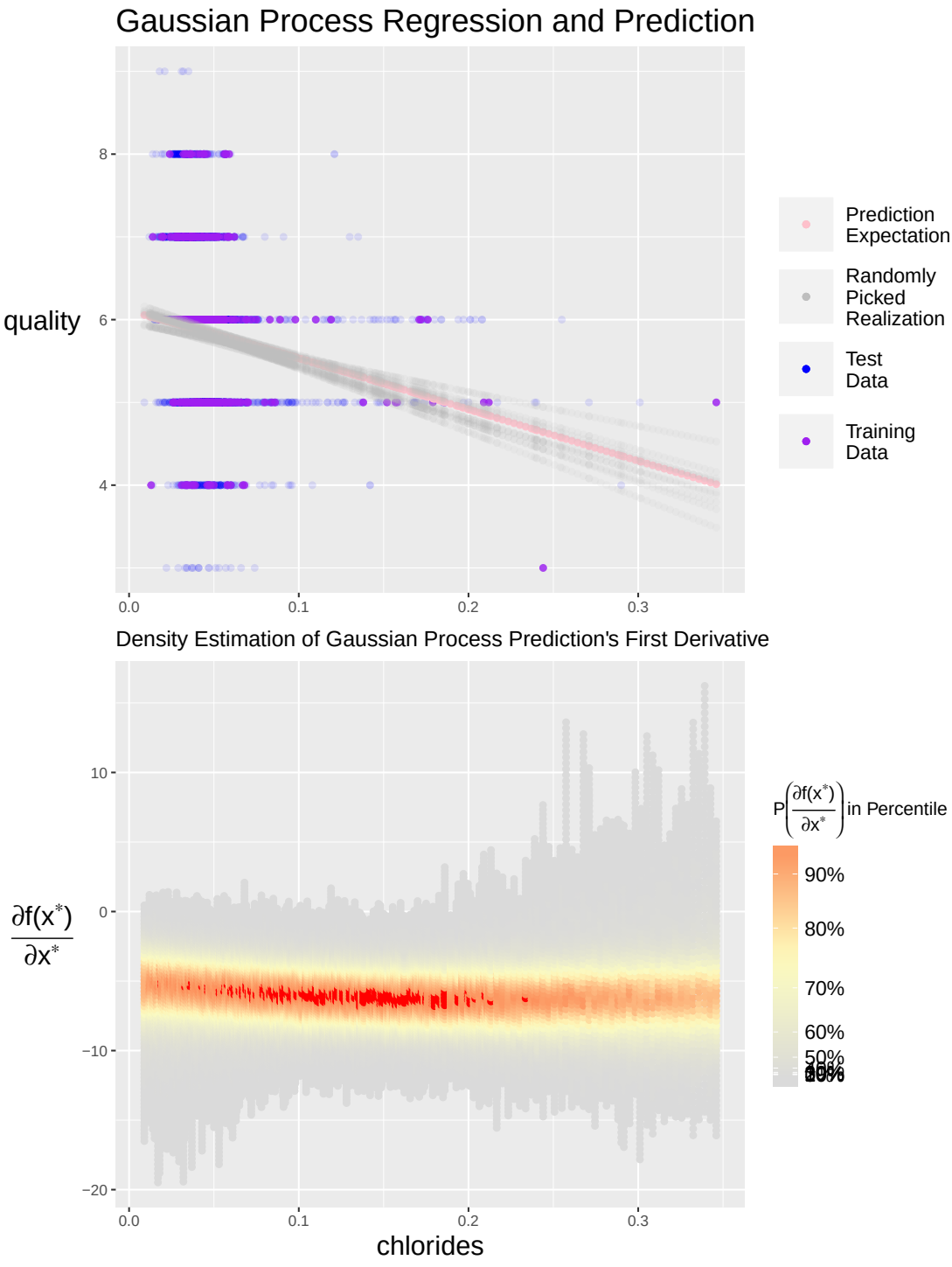


Figure D.26: Example 2: Estimate the relationship intensity between chlorides and white wine quality

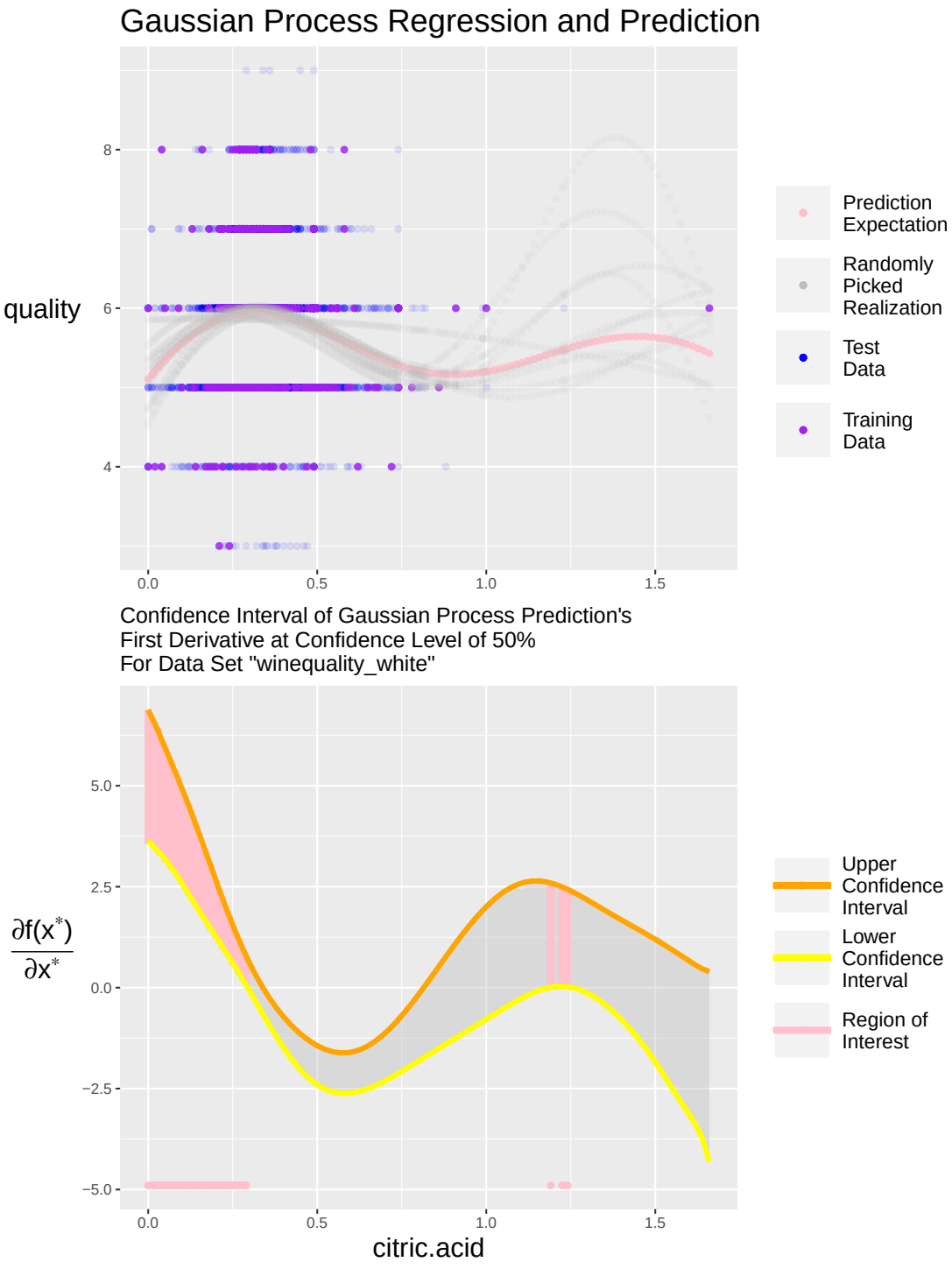


Figure D.27: Example 2: Estimate confidence interval of the relationship between citric and white wine quality

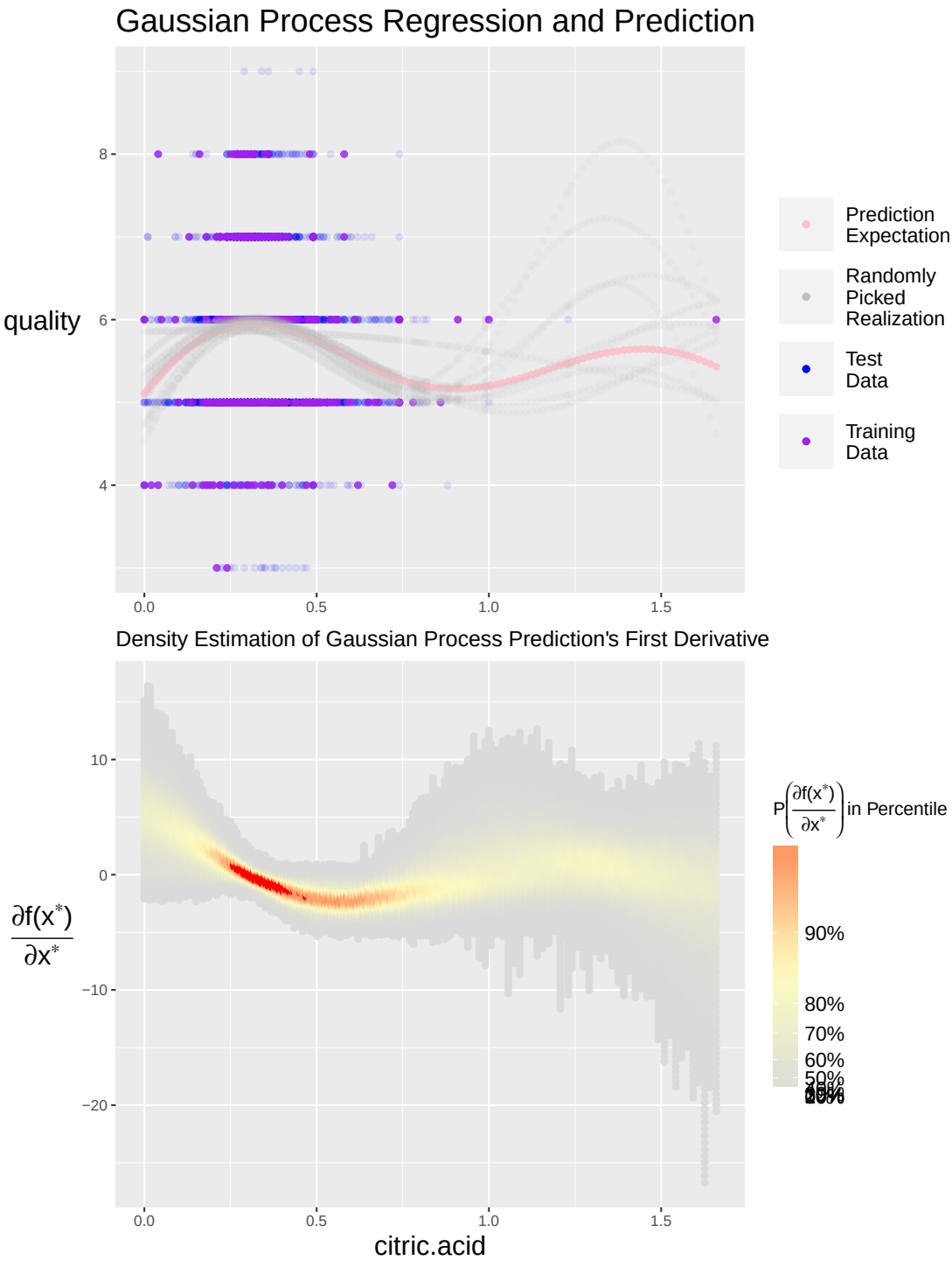


Figure D.28: Example 2: Estimate the relationship intensity between citric acid and white wine quality

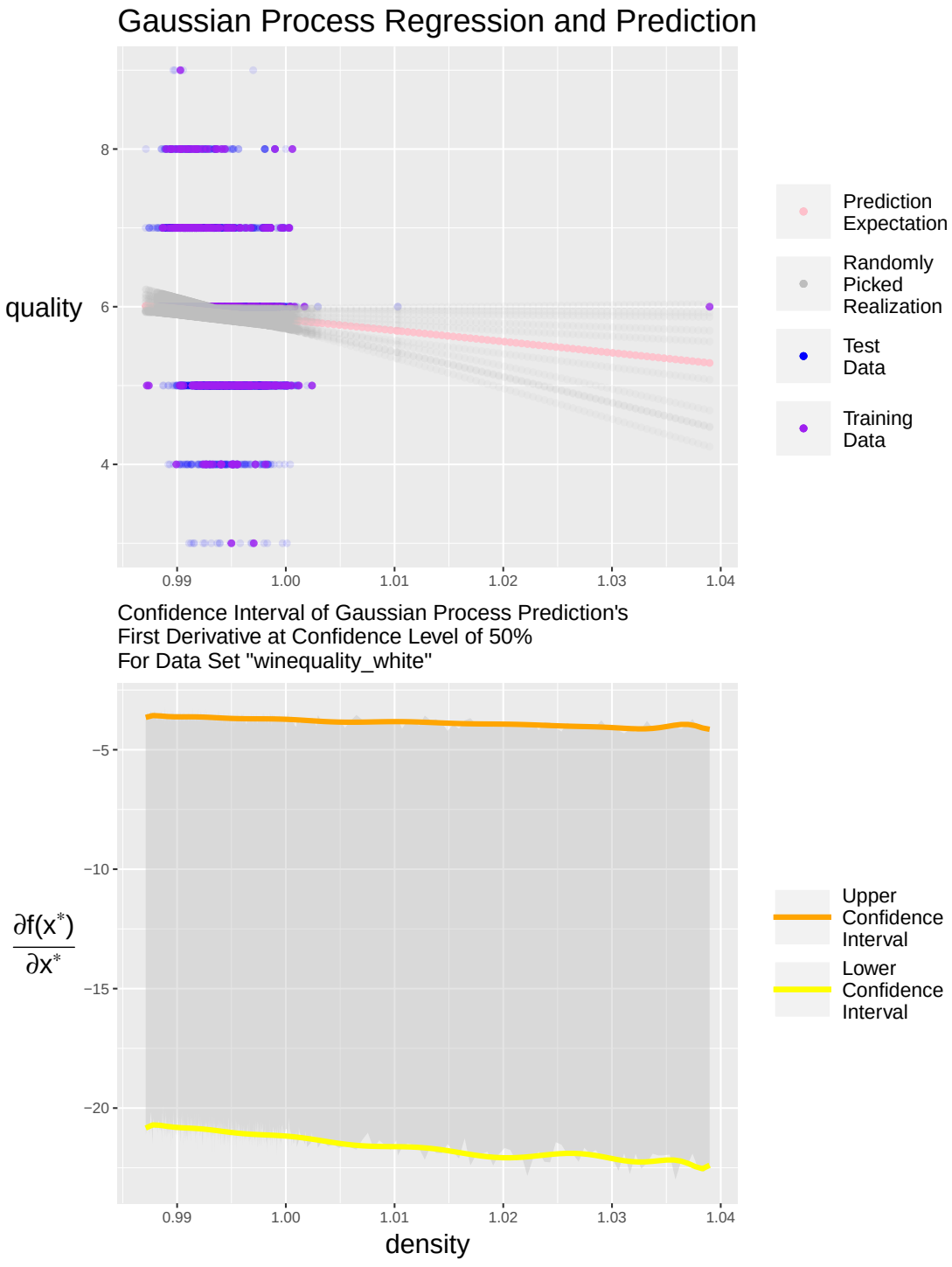


Figure D.29: Example 2: Estimate confidence interval of the relationship between density and white wine quality

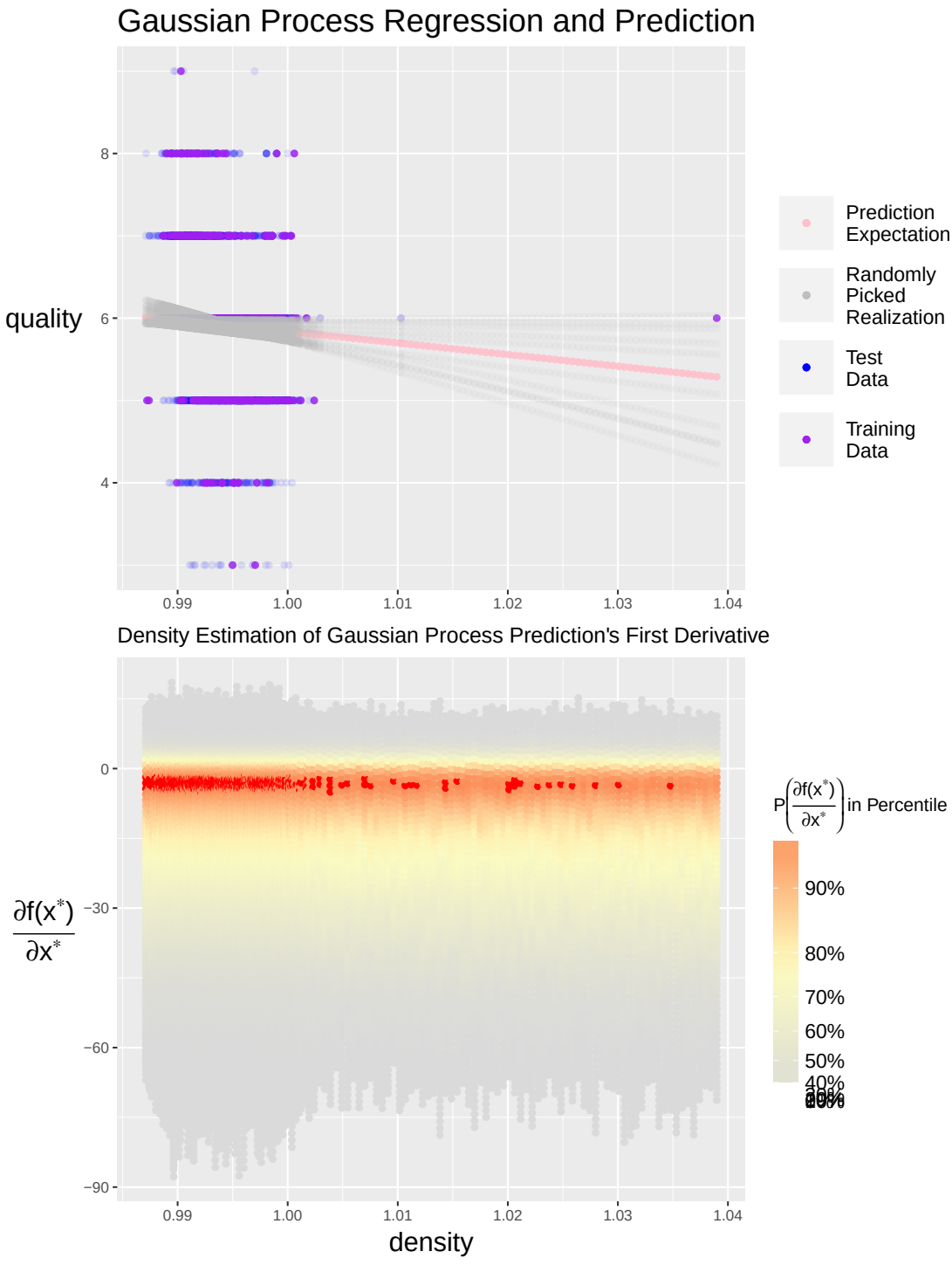


Figure D.30: Example 2: Estimate the relationship intensity between density and white wine quality

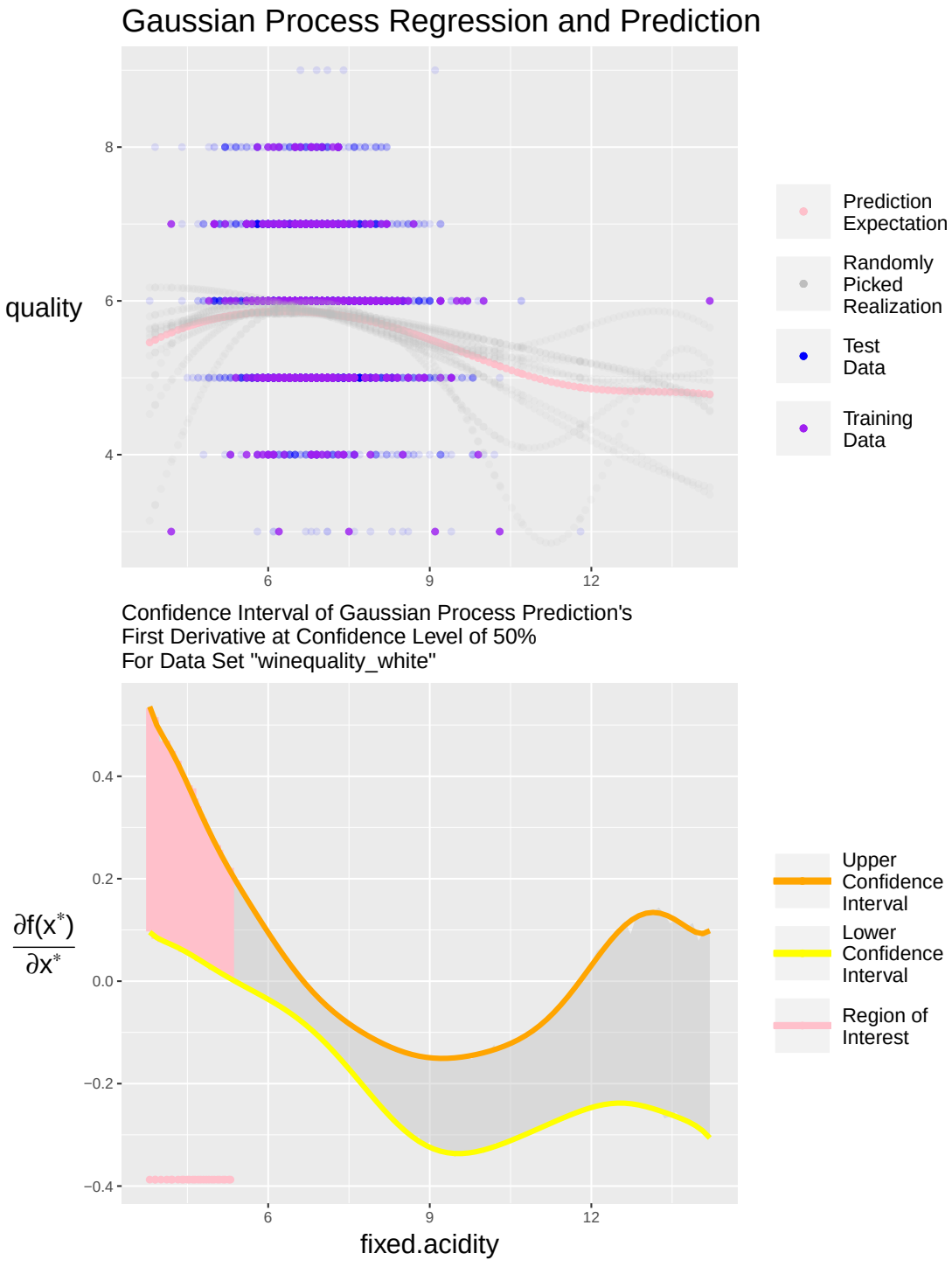


Figure D.31: Example 2: Estimate confidence interval of the relationship between fixed acidity and white wine quality

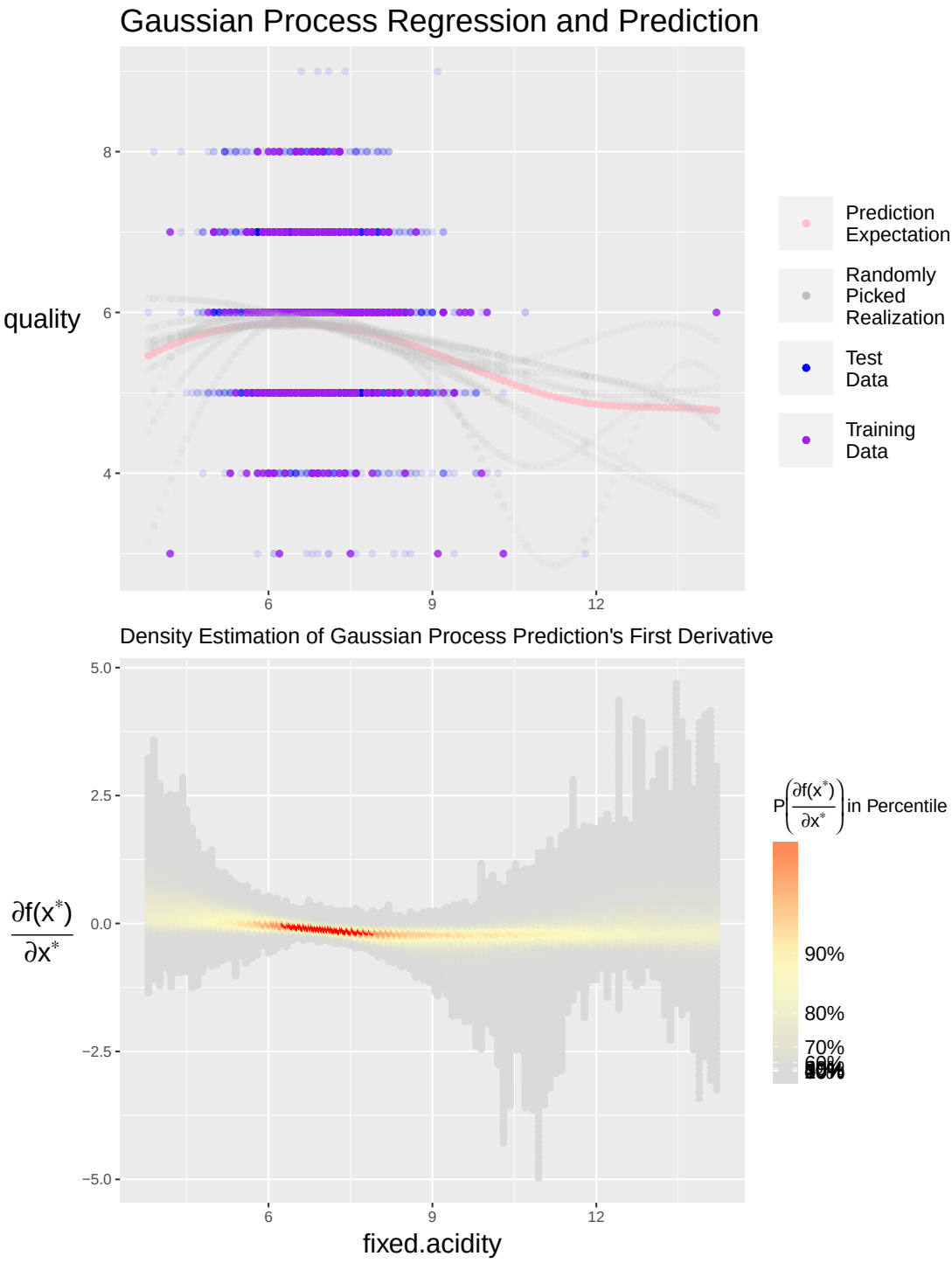


Figure D.32: Example 2: Estimate the relationship intensity between fixed acidity and white wine quality

Figure D.33: Example 2: Estimate confidence interval of the relationship between free sulfur dioxide and white wine quality

Figure D.34: Example 2: Estimate the relationship intensity between free sulfur dioxide and white wine quality

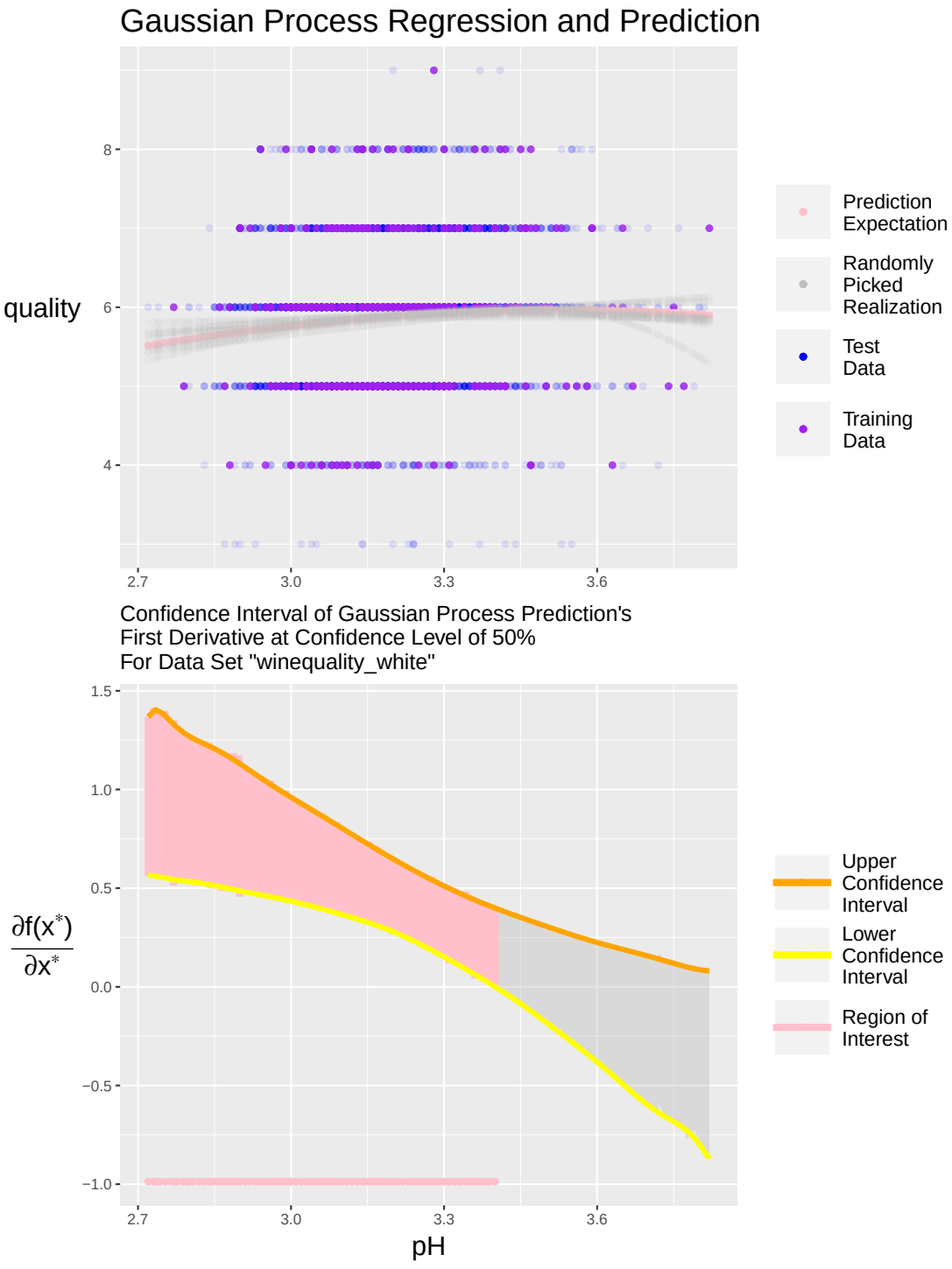


Figure D.35: Example 2: Estimate confidence interval of the relationship between pH and white wine quality

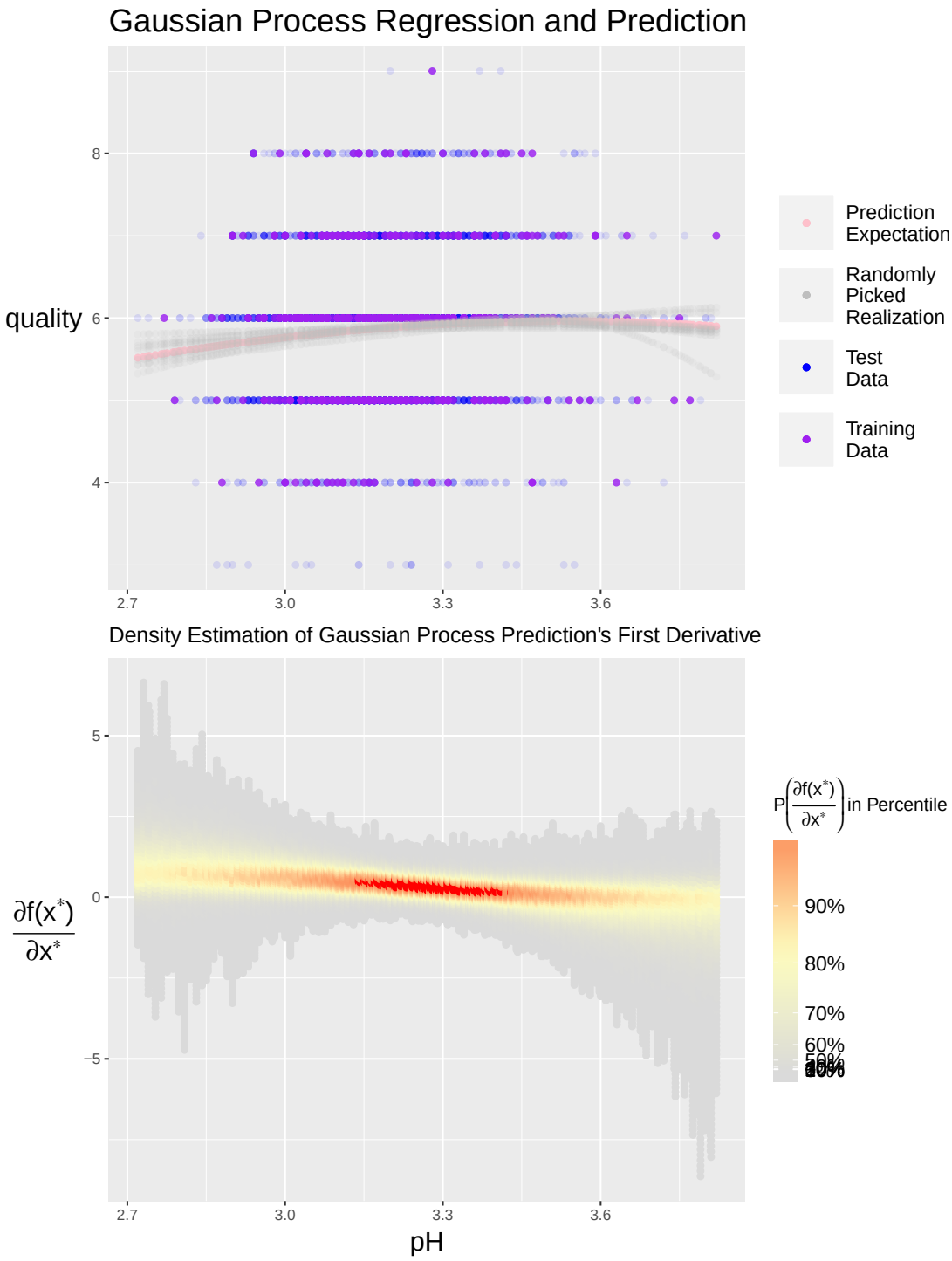


Figure D.36: Example 2: Estimate the relationship intensity between pH and white wine quality

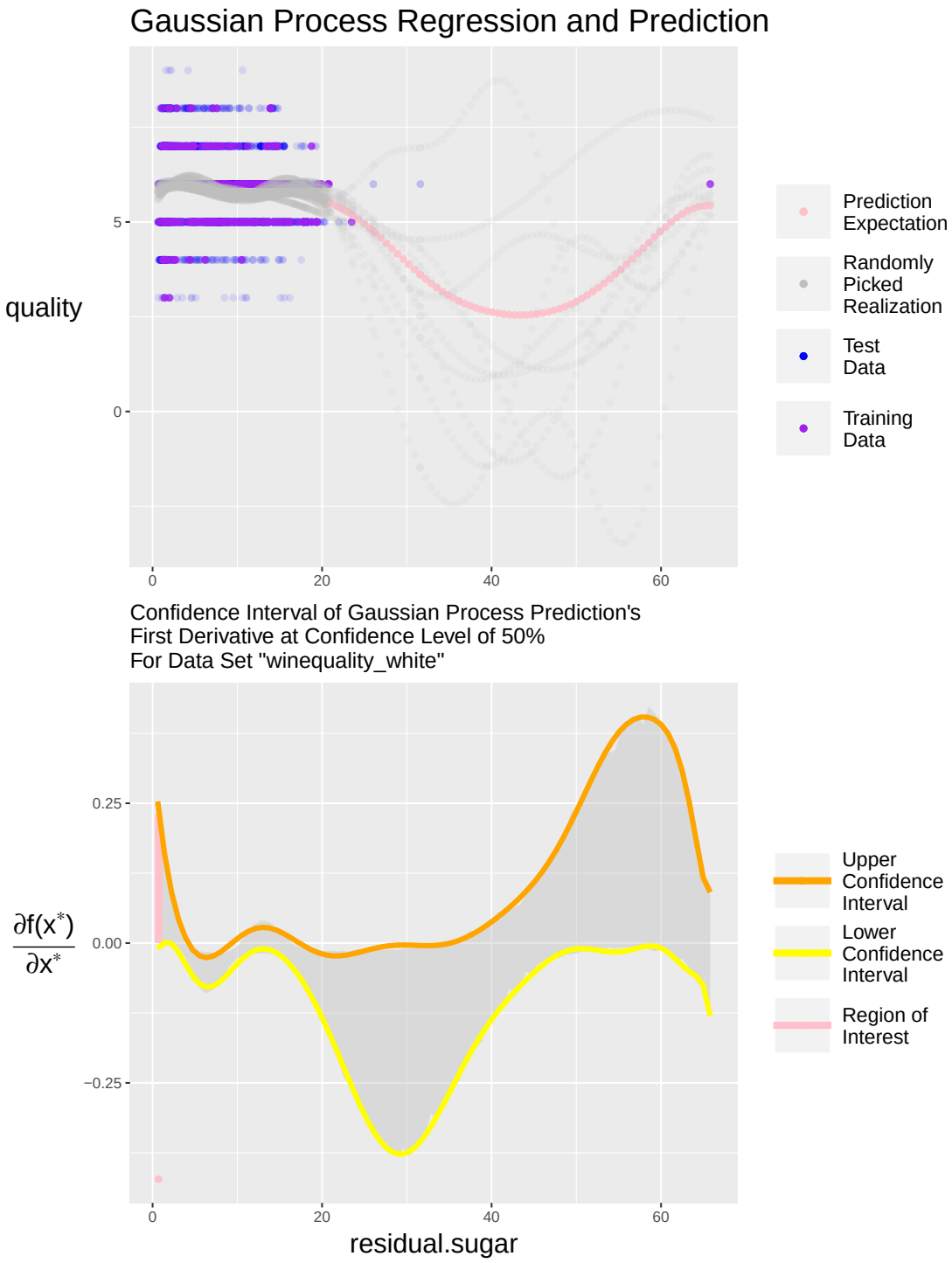


Figure D.37: Example 2: Estimate confidence interval of the relationship between residual sugar and white wine quality

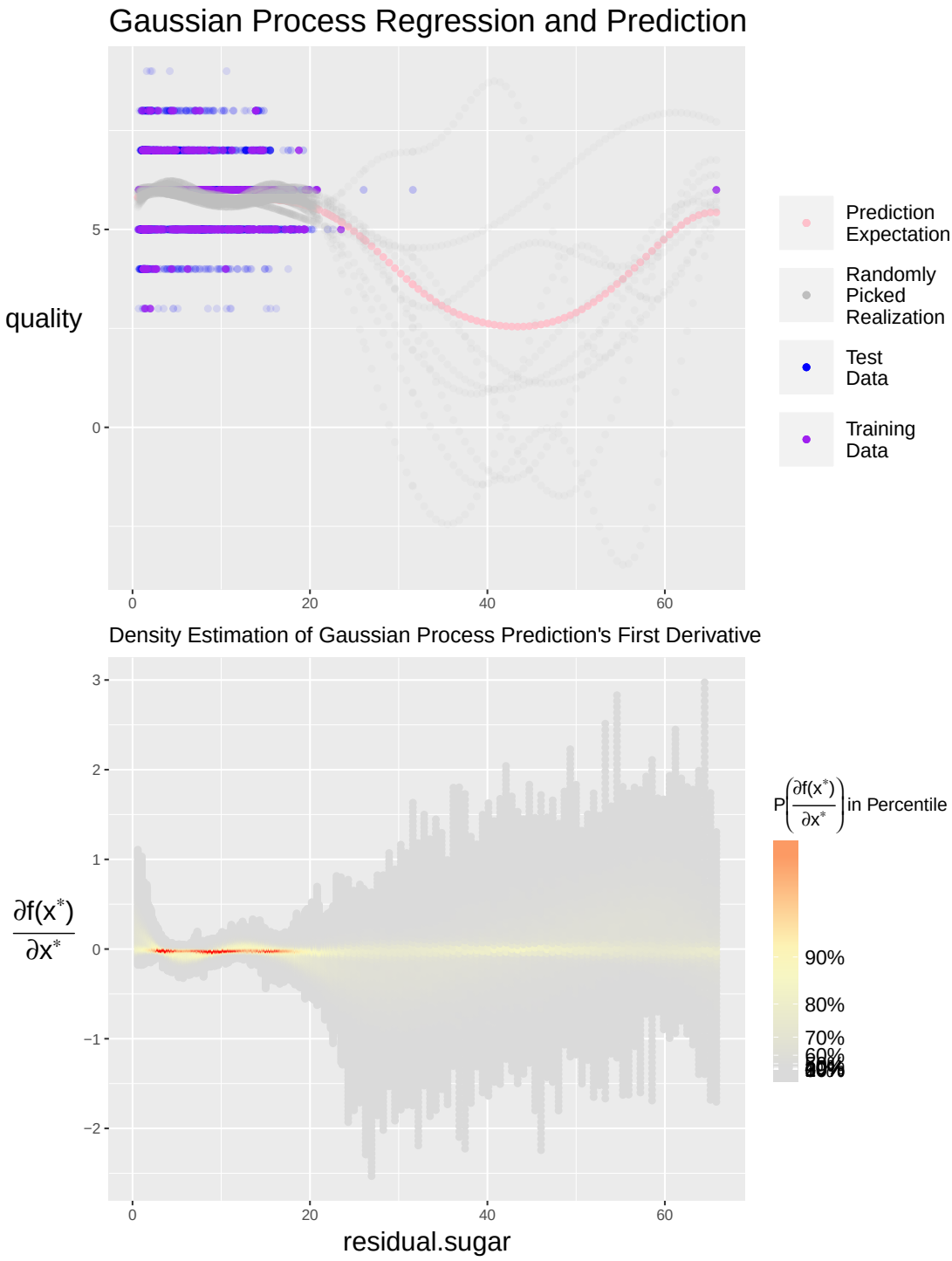


Figure D.38: Example 2: Estimate the relationship intensity between residual sugar and white wine quality

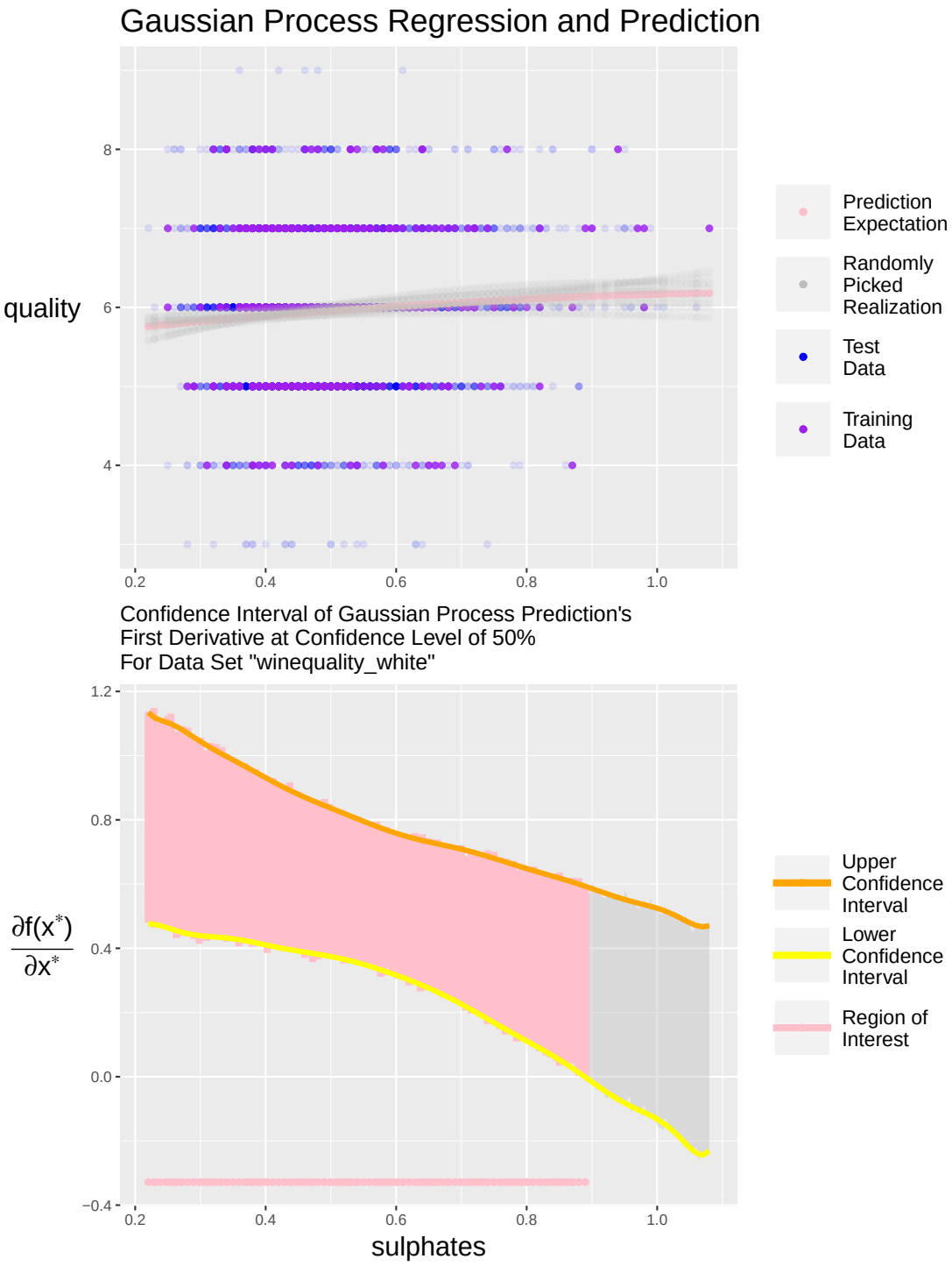


Figure D.39: Example 2: Estimate confidence interval of the relationship between sulphates and white wine quality

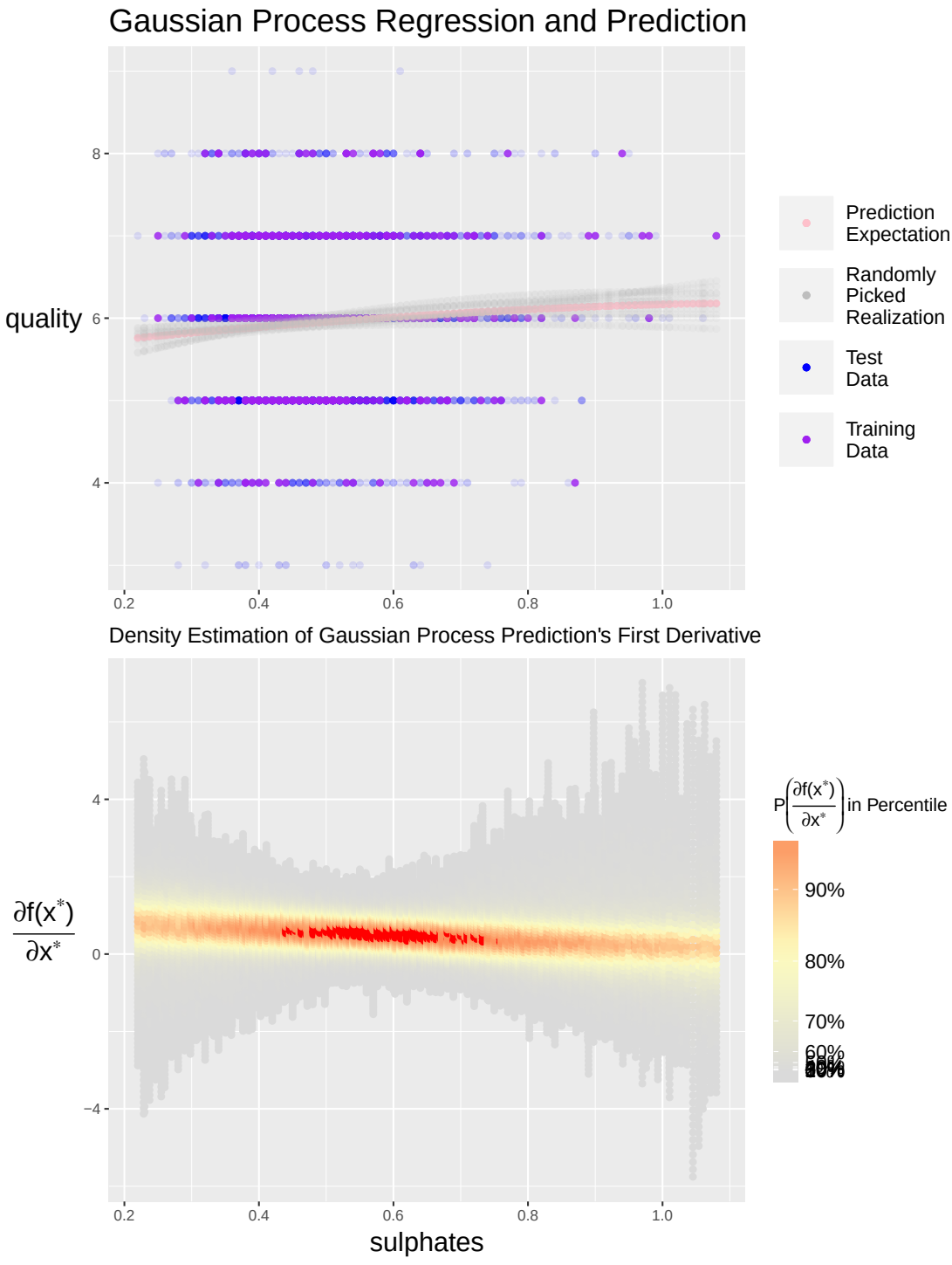


Figure D.40: Example 2: Estimate the relationship intensity between sulphates and white wine quality

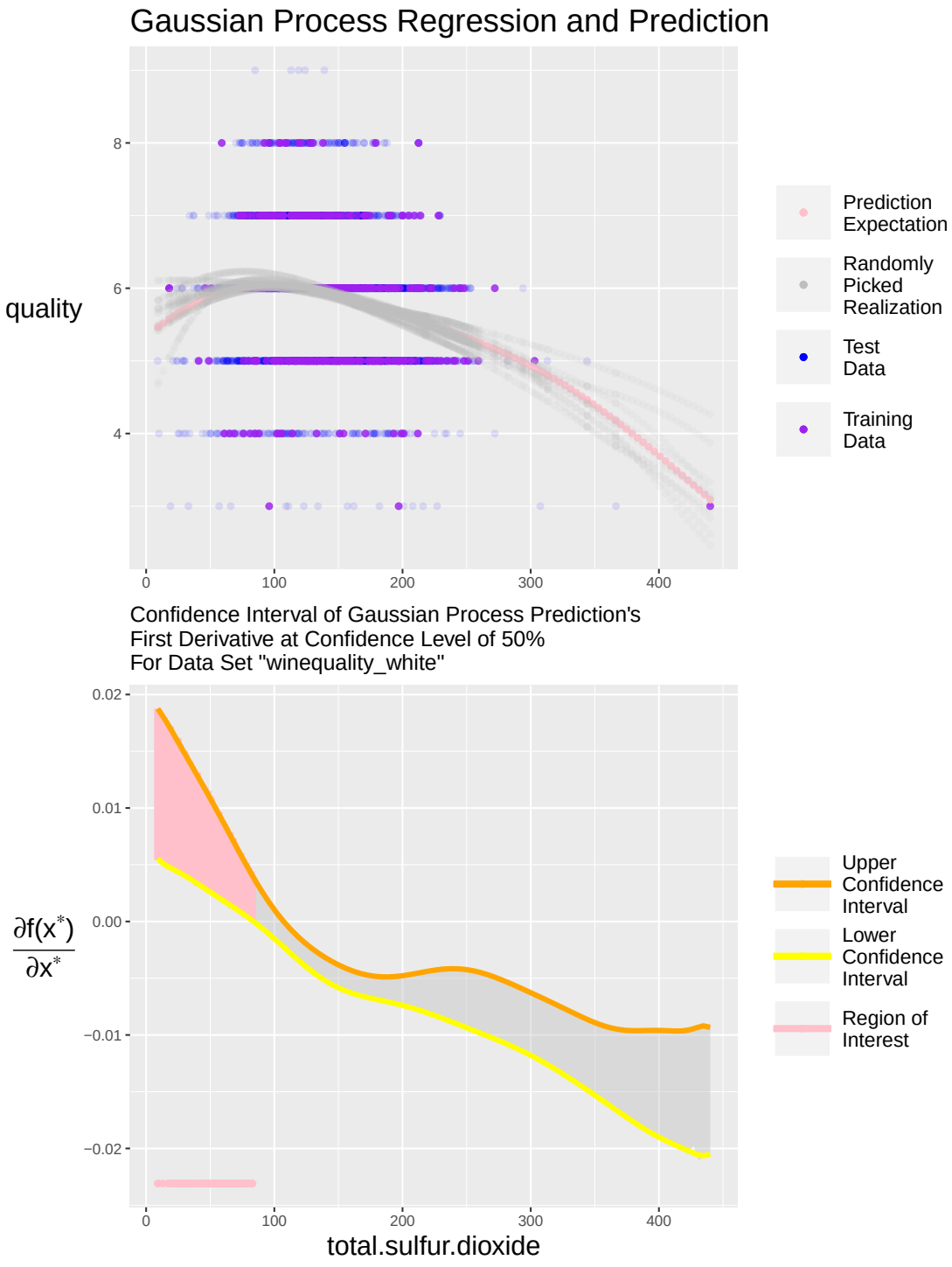


Figure D.41: Example 2: Estimate confidence interval of the relationship between total sufur dioxide and white wine quality

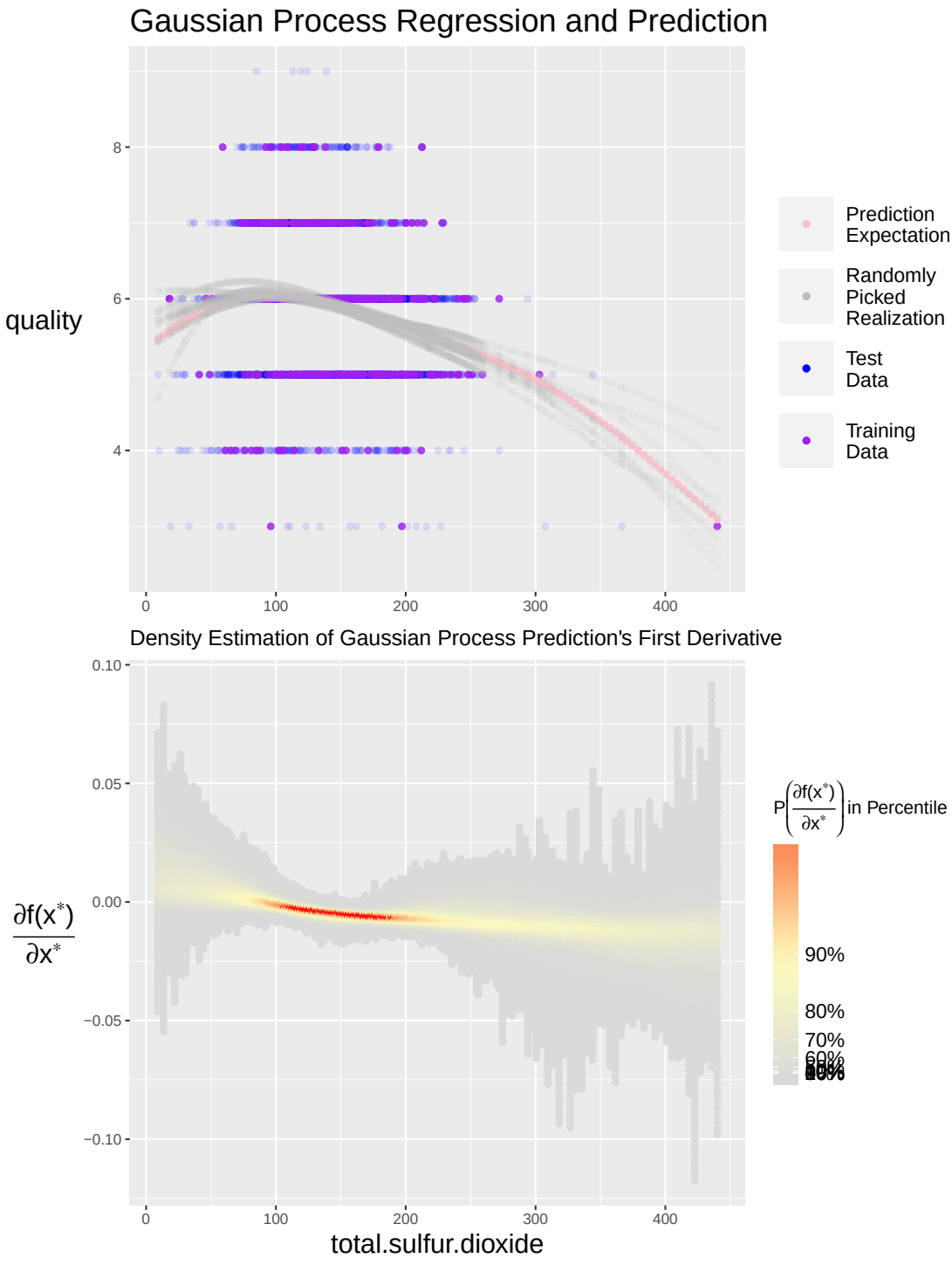


Figure D.42: Example 2: Estimate the relationship intensity between total sulfur dioxide and white wine quality

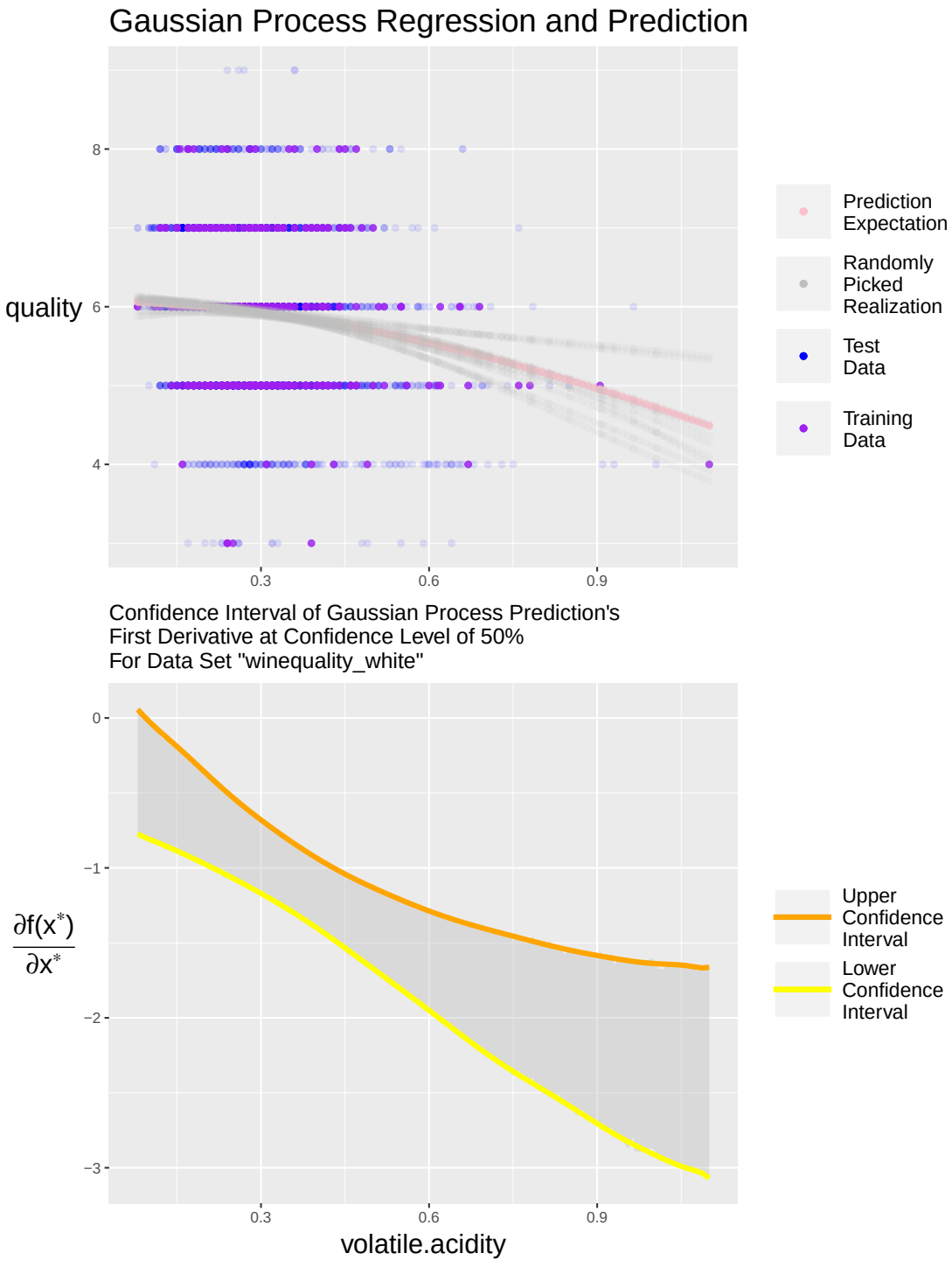


Figure D.43: Example 2: Estimate confidence interval of the relationship between volatile acidity and white wine quality

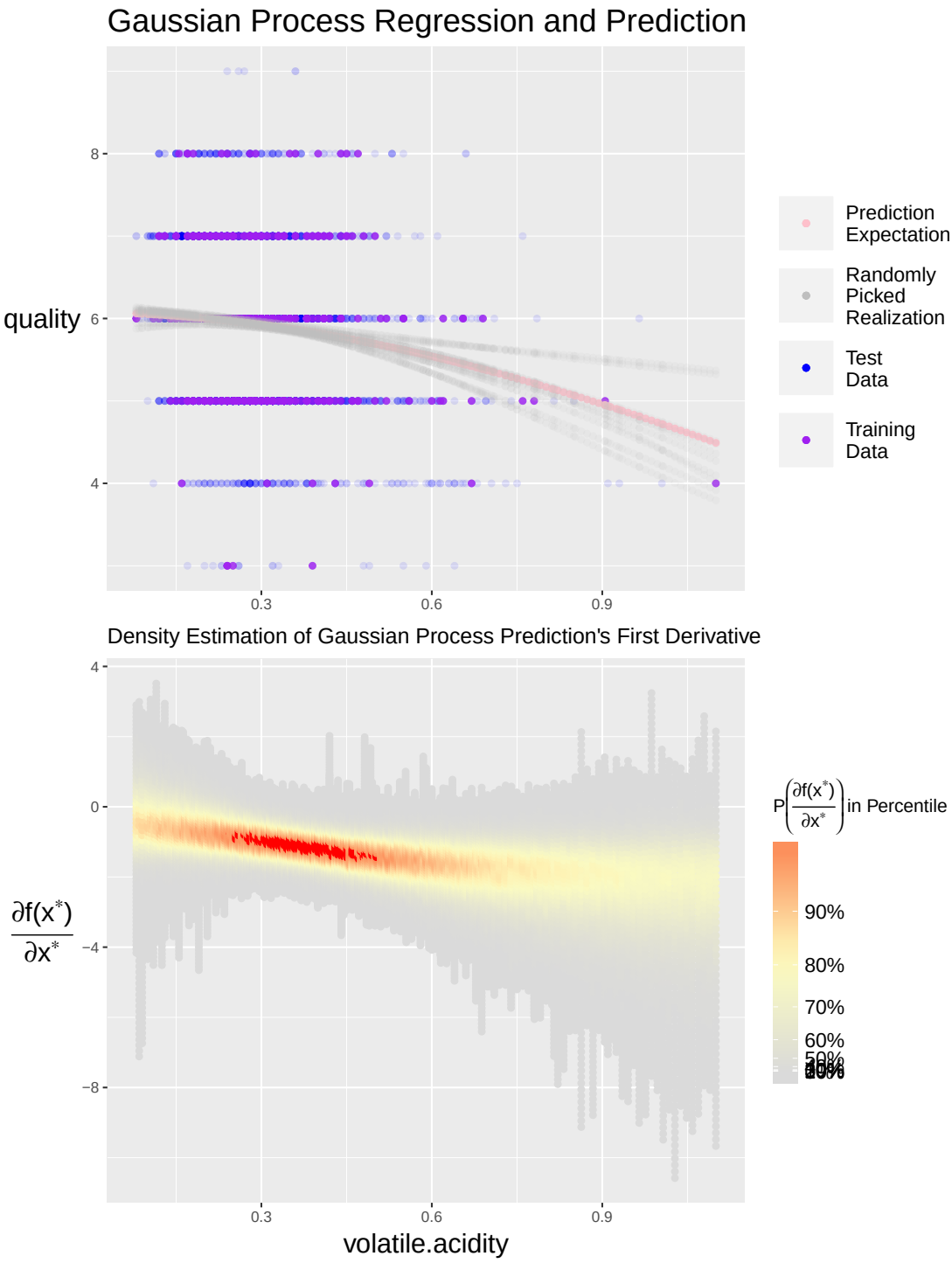


Figure D.44: Example 2: Estimate the relationship intensity between volatile acidity and white wine quality

Appendix E

IRB/HUMAN SUBJECTS APPROVAL

Corresponding to Chapter § 3 Section 3.1

RESEARCH OFFICE

210 Hullihen Hall
University of Delaware
Newark, Delaware 19716-1551
Ph: 302/831-2136
Fax: 302/831-2828

DATE: August 11, 2016

TO: Xin Ji, PhD candidate
FROM: University of Delaware IRB

STUDY TITLE: [943329-1] Newark Restaurant Competitor Identification Study

SUBMISSION TYPE: New Project

ACTION: DETERMINATION OF EXEMPT STATUS
DECISION DATE: August 11, 2016

REVIEW CATEGORY: Exemption category # *(2)*

Thank you for your submission of New Project materials for this research study. The University of Delaware IRB has determined this project is EXEMPT FROM IRB REVIEW according to federal regulations.

We will put a copy of this correspondence on file in our office. Please remember to notify us if you make any substantial changes to the project.

If you have any questions, please contact Nicole Farnese-McFarlane at (302) 831-1119 or nicolefm@udel.edu. Please include your study title and reference number in all correspondence with this office.

cc:

Figure E.1: IRB/Human Subjects Approval Notification