Chapter 2

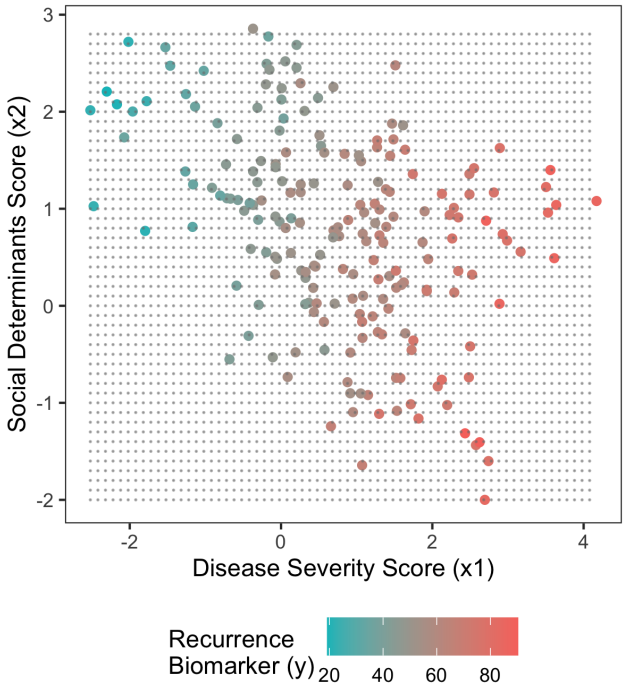
The Basics of Regression

Classification is a form of supervised learning in which the outcome is a category. **Regression** is another form of supervised learning in which the outcome is a numeric value. For example, it may be a lab value, physical characteristic (height, weight, etc.), or numeric measurement (e.g. oxygen saturation).

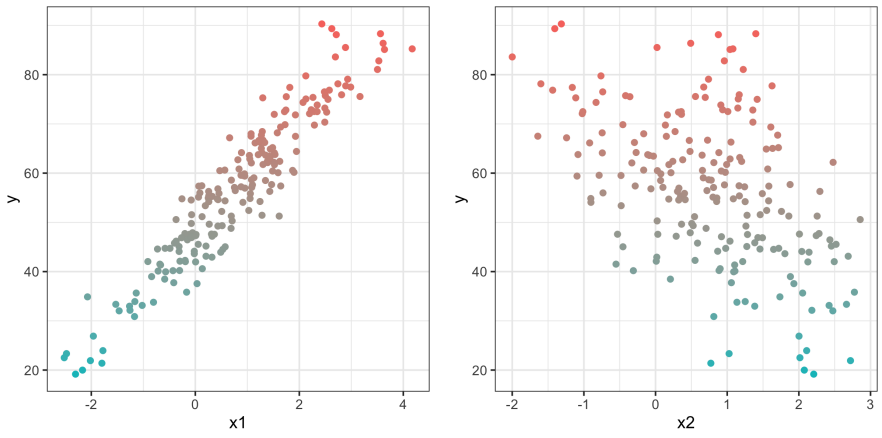
2.1 Visualizing the Regression Problem

Let's consider the same setup from Section 1.2 but this time with a quantitative outcome: a "recurrence biomarker" that indicates the likelihood of recurrence of disease.

Again, we have data on two predictors: a disease severity score (x_1), which characterizes the severity of the illness for which the patient was originally treated, and a social determinants score (x_2), which characterizes a patient's socioeconomic status. We have measurements of x_1 and x_2 on the same 200 patients as in Section 1.2.



This is a plot of the data in a single plane. The color represents the value of the recurrence biomarker – the height of the point above the plane. We want to design a model that will predict the value of the biomarker (y) based on the values of the two predictors, x_1 and x_2. These plots show the **univariate** relationship of each predictor with the outcome.



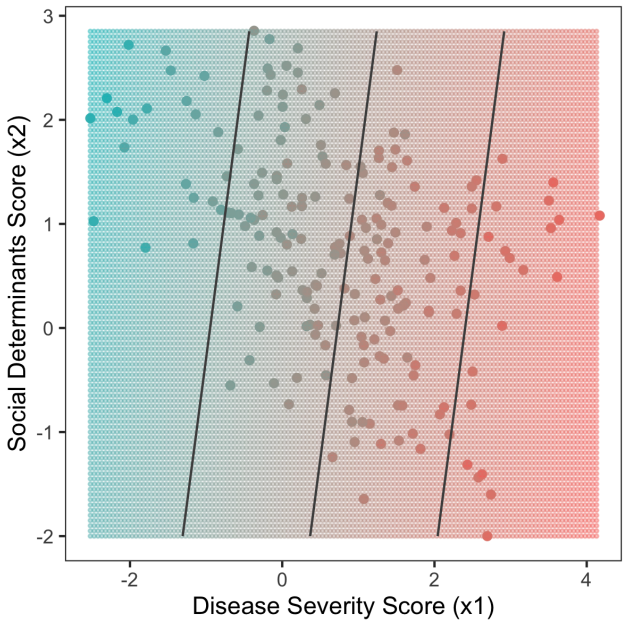
Question 2.1

Which of the two predictors, x_1 or x_2, appears to more strongly influence the value of the recurrence biomarker? Explain your reasoning using evidence from the preceding three plots.

2.2 Three Regression Algorithms

2.2.1 Linear Regression

The regression analogue of logistic regression is **linear regression**[1]. Linear regression creates a hyperplane that slices through the cloud of training datapoints such that it passes as close as possible, on average, to the data. This is, of course, easiest to see when the feature space is two-dimensional, as it is here:



[1]The terminology here is confusing. When we learn about generalized linear models in Chapter 12, you'll see why logistic regression has the word "regression" in its name even though it's a classification algorithm.

The three lines shown here sit on the hyperplane learned by the linear regression model. They are located at heights corresponding to the 25th, 50th, and 75th percentiles of the outcome, y (the biomarker value). The plane tilts downward toward the upper left corner of the $x_1 \times x_2$ grid and upward toward the bottom right corner. It may be helpful to visualize grabbing the $x_1 \times x_2$ plane and rotating/translating it so that it passes through the middle of the training data. Here is a summary of the trained linear regression model:

```
Call:
lm(formula = y ~ x1 + x2, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-11.9218  -3.1032   0.2891   2.8316  12.5813

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  49.8600     0.5370  92.844  < 2e-16 ***
x1           10.4372     0.2855  36.555  < 2e-16 ***
x2           -1.8824     0.3609  -5.215 4.63e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.769 on 197 degrees of freedom
Multiple R-squared:  0.9026,    Adjusted R-squared:  0.9016
F-statistic: 912.4 on 2 and 197 DF,  p-value: < 2.2e-16
```

At each point (x_1, x_2) in the feature space, the model's predicted value of the recurrence biomarker, $\hat{y}$, is

$$\hat{y} = 49.8600 + 10.4372x_1 - 1.8824x_2$$

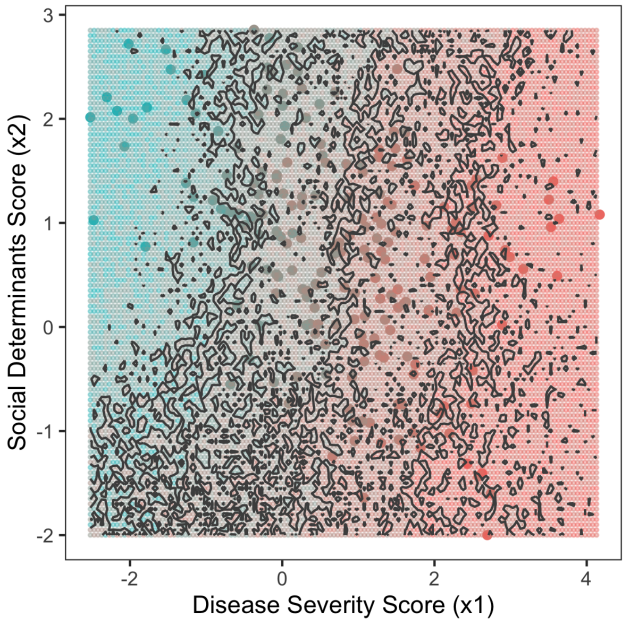
Question 2.2

Compare and contrast the output from the linear regression model with the output from the logistic regression model in Chapter 1. What looks the same? What looks different? What is being predicted in each case?

2.2.2 K Nearest Neighbors (KNN)

Regression using KNN works very similarly to KNN for classification. In classification, we allow the nearest K points to vote on the label of a new test

point. In regression, we **interpolate** between the values of the surrounding points to come up with the value of y for a test point. Typically this is done just by averaging the y values of the nearest K points, but you can also do something more sophisticated, like weight their contributions by distance to the test point. Here is a contour plot of the regression surface produced by KNN ($K = 15$) for our example:

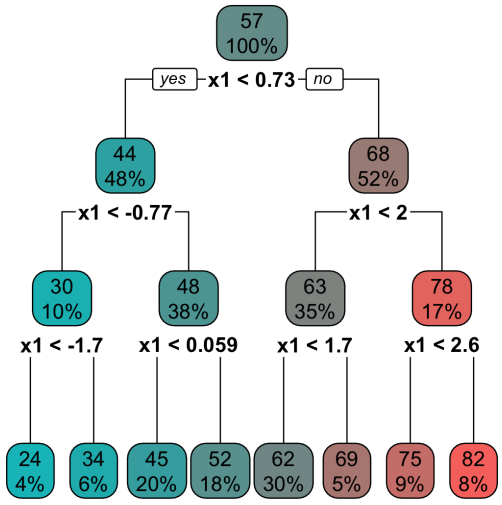


The contours are again drawn at the 25th, 50th, and 75th percentiles of the outcome, y. This looks like a bit of a mess compared to the linear regression plot, but at the same time, the KNN algorithm is able to capture arbitrarily complex relationships between x_1, x_2, and y that can be missed by other regression algorithms.

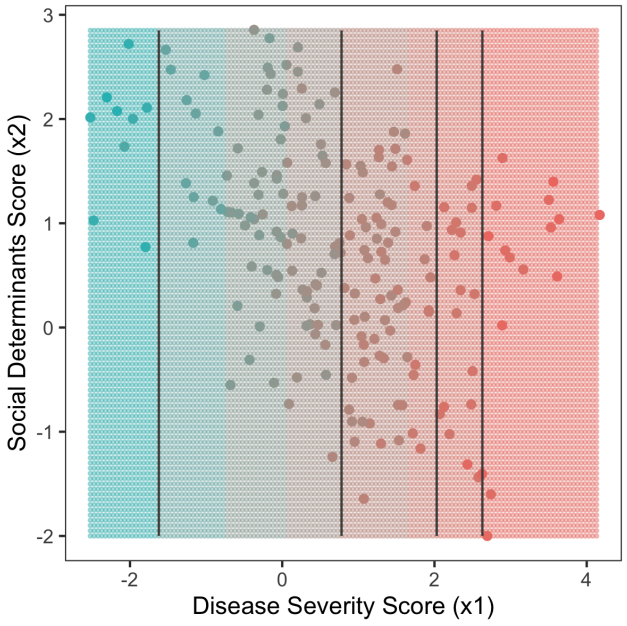
2.2.3 Decision Tree

Decision tree regression is similar to decision tree classification except that the output at each leaf is not a class label or the probability of membership in the positive training class (both of which are shown on the tree in Section 1.3.3),

but a numeric value. That value corresponds to the mean outcome value for the points in that leaf.



The predicted biomarker values for a decision tree trained on this dataset (created using the `rpart` package in R with default parameters) are shown here:



You can see that the decision tree always chooses to split on x_1, the disease severity score, rather than x_2. Revisit Question 3.1 to remind yourself of why this is. The regression surface produced by the decision tree looks like a set of stairs climbing higher and higher as one moves from left to right across the $x_1 \times x_2$ plane. The predicted value of y, the recurrence biomarker, is constant within each stair.

Question 2.3

Compare this decision tree with the decision tree for the classification problem in Chapter 1. What is the same? What is different?

Question 2.4

This **regression tree** has eight leaves. What region of the feature space does each leaf correspond to?

Question 2.5

What are the advantages and disadvantages of each of these three regression algorithms (linear regression, KNN, regression tree)?