Hierarchical Models for Crowdsourced Bicycle Route Ratings

A Thesis
Presented to
The Division of Mathematics and Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Will Jones

May 2016

Approved for the Division
(Mathematics)

Andrew Bray

Acknowledgements

I want to thank a few people.

Preface

This is an example of a thesis setup to use the reed thesis document class.

Table of Contents

List of Tables

List of Figures

Abstract

The preface pretty much says it all.

Dedication

I want to thank a few people.

Introduction

Knock Software's *Ride Report* app combines a simple thumbs-up/thumbs-down rating system with GPS traces of bicycle rides to compile a crowdsourced data set of which routes are and are not stressful for urban bicyclists.

The app that collects the data is simple: *Ride Report* automatically detects when a user start riding their bike, records the GPS trace of the route, and then prompts the user at the end of the ride to give either a thumbs-up or thumbs-down rating. From this, they were able to create a simple "stress map" of Portland, OR, which displays the average ride rating of rides going through each discretized ride segment.

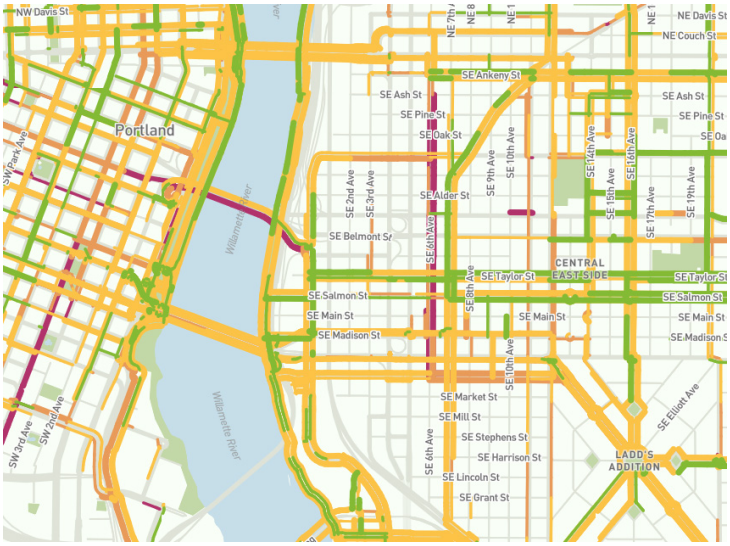

Figure 1: Ride Report's Stress Map for Portland. Greener road segments incidate less while more red segments indicate more stressful streets. "Stress" computed by taking the average rating for each segment.

The app privileges reducing barriers to response to increase sample size over ensuring quality and consistent responses. This presents the first problem: how can we analyze ratings when riders are likely rating rides inconsistently?

At the same time, we have another challenge. We have ratings associated with routes, but we would like to know the effect of particular road segments, for both inference (what effect does this road segment have on the rating?) and prediction (given a route, what do we expect the rating to be?) purposes.

Finally, there are interesting issues with missing data. First, the sample of rides

we have are biased towards routes that riders percieve as better. Second, a significant portion of the rides are missing a response, and non response is unlikely to be independent of the response. The good news is that we have all the predictors for every ride, enabling us to build a model that is able to leverage the data with missing responses rather than listing the missing data as a liability.

0.1 Accounting for Rider Rating Variance

For ratings we are interested in modeling variance between riders (as we might expect riders to have different rating patterns than their peers) and within riders (as riders may not rate the same route and conditions the same every time). To model this, we propose using multilevel regression, with random effects from each rider. This approach has been used in similar situations, in one case to model sexual attraction[1].

In a multilevel model, we fit a regression where the intercept term varies by group but comes from a common distribution. For example, if we let r_i be the rating of the ith ride, X_i be the ride-level variables, then we can fit a regression:

$$\mathbb{P}(r_i = 1) = \text{logit}^{-1}\left(\alpha_{j[i]} + X_i\beta\right),$$

where α_j is an intercept specific to rider j. In addition, the rider intercepts come from a common distribution,

$$\alpha_j \sim N(\mu_\alpha, \sigma_\alpha^2),$$

where μ_α is the mean of all the α_js.

Using varying intercepts will allow our model to exhibit the property of varying rates of riders giving stressful ratings. This can be extended to model differences in how riders respond to different kinds of road conditions using varying slope models. We explore multilevel model further in Section Section 2.2 and multilevel models for riders in Section **??**.

0.2 Addressing Road Segments as a Level

We examine multiple approaches to modeling road segments. In the first, we regard add regression term for the route, that is a weighted sum of terms for each road segment in the route, weighted by the lengths of the segments. The terms for the

the contribution of the route to be a weighted sum of the contributions of road segments in the route, weighted by the lengths of the segments.

0.3 Approaching Missing Data at Multiple Levels

This data set contains missing data at two levels.

First, there are many routes without any ratings. The routes taken by riders are not a random sample of routes, but are often already chosen by the rider as the least

[1] Mackaronis, Strassberg, Cundiff, & Cann (2013)

stressful ride. As we might expect because of this bias, only a small proportion of rides are rated as "stressful."

One solution we explore to this problem is adding a segment popularity parameter as a segment-level predictor.

Second, not every ride is given a rating. We do have the route they chose, and all associated covariates, but the response variable, rating is missing. As we will discuss in chapter **??**, the pattern of non-response is unlikely to be unbiased.

To address the second problem, we first build a separate probability model that predicts non response, and then integrate that model into our main model.

Chapter 1

Data Sources

We combine several data sources to do our analysis. Information about individual rides, including the GPS trace, the rider, and timestamp were provided by Ride Report. Weather data were collected from Weather Underground's archive of the KPDX weather station and a Portland Fire Bureau station.

Our goal in this chapter is to discuss this data and what considerations we should have in mind before exploring it in depth. This includes how and by whom the data was collected, who and what is this data actually representative of, and what samples were taken of the data.

Some of these considerations, such as the limited demographics represented in the Ride Report data, pose serious limitations to how our inferences can be generalized. Others, such the large number of missing responses in the Ride Report data, motivate the analysis we are doing in this thesis. Finally, there are other considerations which we will acknowledge here, but addressing them is out of the scope of this thesis. This data set contains an abunance of potential research questions, only a fraction of which could be reasonably addressed in one thesis.

1.1 Ride Report

Ride Report's data is the focus of this paper. Knock Software created the app to collect large amounts of information about urban cyclists' routes and experiences on those routes. The hope is that this information will be valuable to city planners.[1]

Ride Report's approach to crowdsourcing this data is particularly important to understand. The app automates every piece of the data collection process except for the rating given by the rider. Thus, the app casts aside nuanced and (somewhat) reliable human input in favor of increasing sample size (i.e. one could imagine a similar app where users have more control over how the route is recorded, have the ability to rate on a more fine-grained scale, and are given more direction in what they are rating for.) This leads to some potential issues that we need to address in our models.

Before we get into the potential issues in the data collection, let's examine the

[1] Knock's other project is making a cheaper bicycle counter for cities to monitor traffic flow, again intended to be sold to cities wishing to improve bike infrastructure.

data collection process itself. When installed on a persons phone, the Ride Report app attempts to automatically detect when the user starts riding their bicycle, based acceleromoter data, when a user leaves a familiar Wi-Fi network, and some other pieces of information. When the app detects the start of a ride, it starts recording a GPS trace. At the end of the users ride, the app detects them getting of their bike (in a similar process to how it detected the start of a ride) and prompts them to give a rating of the ride. The ride data is saved then, even if the user does not provide a rating.

This automatic detection of when a ride starts and stops leads to two related and common errors in the dataset: first, one ride is often split into two or more rides at points, such as at a stoplight or a train crossing, where a cyclist stops for an extended period of time; second, car rides are sometimes misclassified as bicycle rides and vice-versa (car rides are not rated.) The app does allow riders to correct the misclassification, but there is currently no way to join split rides back together (Knock is working on changing that, though.)

The app only recently became publicly availible and has undergone significant changes in the course of it's life. In particular, while the ratings have always been binary, the labels have changed at various points in time. For a while the rating labels were "Stressful" and "Chill", while now they are labelled "Recommend" and "Avoid" (see Figure 1.1.) Other fundamentals of the data collection process, have remained constant, however.

The data colletion method itself has some problems, but there also may be some biases in the population of riders using the app. The app is only availible on iOS, so only iPhone owners could use this application which may imply a bias toward riders of higher socioeconomic status. At the time of the start of the thesis, the app was in private beta, meaning only people who actively sought out using the app were able to use it. Now the application is public and on the Apple App Store, making it more widely availible. So many of the earlier rides may be people within the developer's personal network. Unfortunately, it's hard to make any solid conclusions about the users of the app because Ride Report doesn't collect any demographic data about their riders.

One other issue with the Ride Report data guided our analysis: privacy. Because the data involves timestamps and GPS locations of people's commutes, the data is very sensitive: one could easily infer someone's home and workplace based on their most commons routes. In fact, this data is protected by an end-user license agreement (EULA) which prevents sharing of data, without the explicit permission of those involved. This presented a logistic challenge: how were we to do inference and data exploration without access to the data?

By agreement with Knock Software, identifying data must be kept private. With permission from five riders, Knock was able to give us a small subset consisting of all the rides from those five riders, to be kept confidential. That is the data set we used for prototyping models and some basic exploratory analysis. Knock also agreed to allow us to run models fitting scripts on larger samples of their dataset, as long as they were performed on their computers, with no identifying data leaving their system.

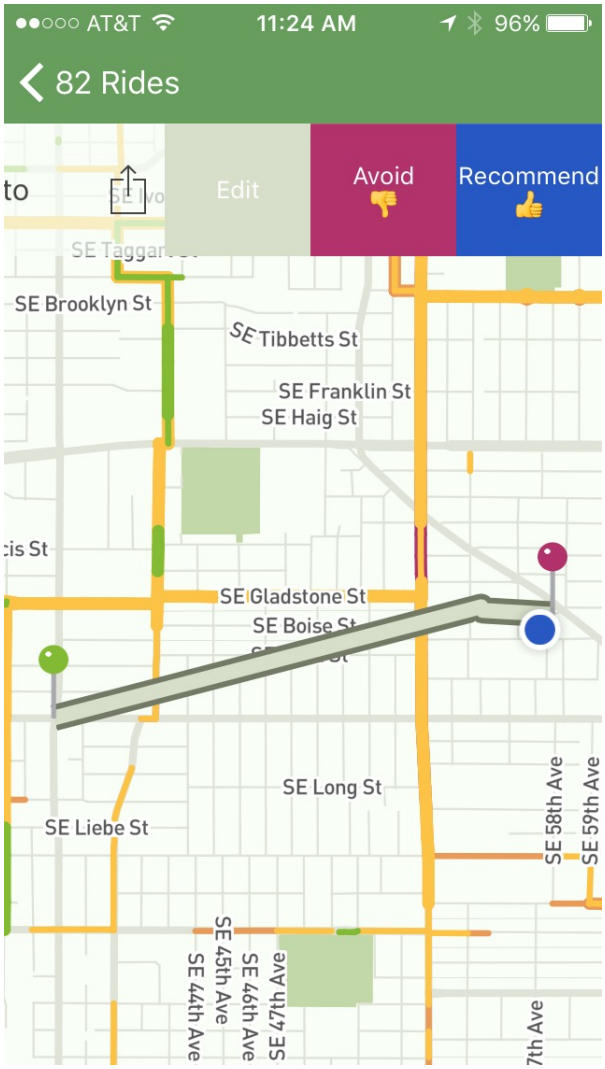

Figure 1.1: The Ride Report app's interface has changed significantly between versions, including the rating text displayed after a ride. This is the current version as of Februrary 2015.

While at first this set up seems like an inconvenience, it actually has some advantages. One of the pitfalls of having an entire data set, especially a high dimensional one, is that in performing exploratory analyses it is often too easy to find spurious "statistically significant" results. Instead, we must come up with our models before running them, greatly limiting the choices we can make in the garden of forking paths.

1.2 Weather Data

Slippery roads and formidable winds are no fun for anyone balancing on a two-wheeled vehicle. Weather is, then, one of the most obvious family of predictors for ride rating, at least intuitively. We use the time of a ride to join in data about the weather

conditions during the ride, including

- the temperature,
- whether and how much it is raining,
- whether the roads are wet or have puddles,
- wind and gust speed.

We include the first two, temperature and percipitation, to account for rider comfort. A swealtering, frigid, or stormy day could make an unpleasant experience for a bicyclist and thus could lead to more negative ratings.

On the other hand, we include the last two, wet road and gust speed, as factors that impact safety. During and after storms, puddles often accumulate in bike lanes before the center of the road, pushing cyclists into lanes shared by cars, which are often more dangerous.

Gust speeds impact the aerodynamics of a ride, which are particular important for bicyclists. It's one of the main reasons cyclists care about getting into lower (and more aerodynamic) rider positions. Thus, high wind or gust speeds may affect rider rating.

We are limiting our study to rides in Portland, Oregon. Given this, we can first assume that it may be reasonable to expect that riders are used to the same climate, and thus have somewhat similar responses to weather. This also makes it reasonable to use data from one nearby weather station, rather than attempting to collect from several stations and creating a spatial model for weather.

For daily summaries of weather conditions, we used weather history from the KPDX weather station at Portland International Airport downloaded from Weather Underground. From this we were able to get daily weather data, including

- Average, minimum, maximum temperature for the day.
- Total percipitation.
- Mean wind speed, as well as gust speed (speed of brief, strong winds.)

We also got hourly rainfall data from a data stream at the Portland Fire Bureau Rain Gage at 55 SW Ash St., which is just about the geographic center of Portland. This just gives raw uncorrected rain guage data, but gives us a fine grain look at how much rain there has been recently.

For daily weather data, such as temperature highs and average wind speed, we use information from the KPDX weather station. This weather station is the best calibrated in the area. It is further from the geographic center of the rides we are examining, but because the weather is daily summary statistics, we don't expect closer weather stations to be much more informative. Figure 1.2 shows the geographic positions of these two stations.

1.3 Road Data

To get road feature data, such as bike lanes and intersection types, we had to bring in other data. More on this later. (We may use Open Street Map data and/or data from civicapps.org/datasets)

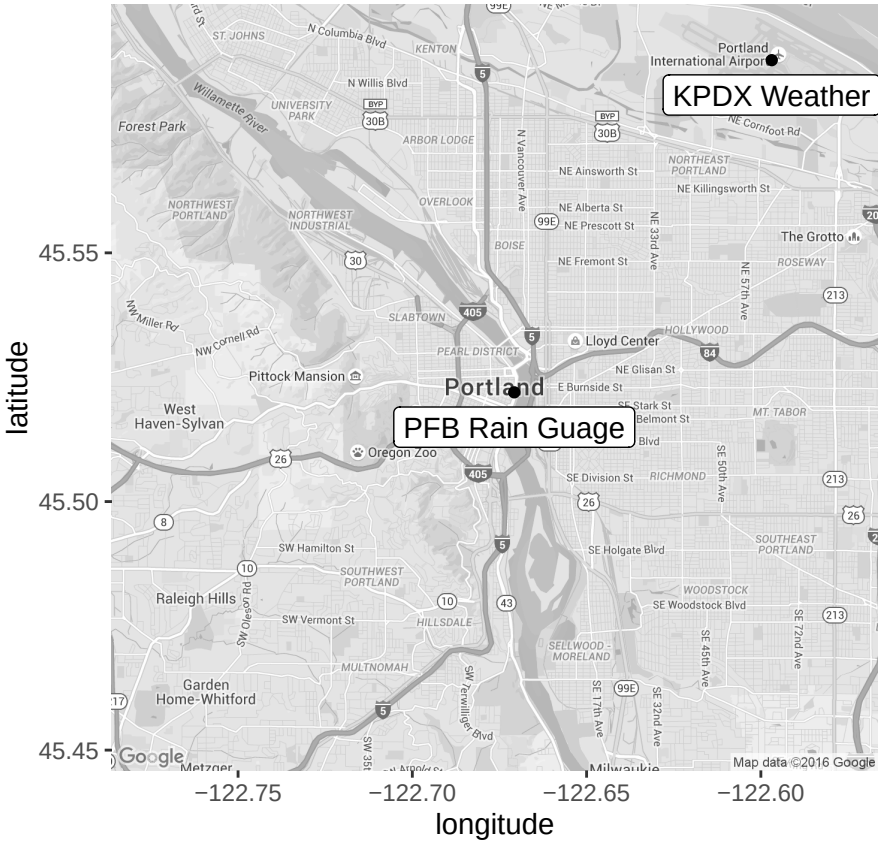

Figure 1.2: Positions of weather data collection sites. Daily weather information was collected at the KPDX weather station at Portland International Airport. Hourly percipitation data was collected at the Portland Fire Bureau's rain guage in downtown Portland.

Chapter 2

Methods

In building our models, we use a multitude of statistical methodologies. We outline here the central methods used, both to familiarize the reader and to establish the notation we use throughout this paper.

Statistics is often split up into prediction and inference. Both are important to us here: we use inference to answer our research questions and prediction—through cross validation—to evaluate the validity of our models.

Our models combine logistic regression and multilevel regression. To evaluate our models, we use graphical tools, such as the separation plot, to examine the performance.

2.1 Logistic Regression

Statistical models are often split into regression models—models with a quantitative response—and classification models—models with a categorical response. Thus, it may seem odd that we are using a regression model when our response variable, ride rating, is a binary outcome.

But, when modeling a binary variable Y, we consider it a Bernoulli random variable,

$$Y = \text{Bernoulli}(p),$$

where p is the probability the outcome is 1 and $1 - p$ is the probability the outcome is 0. So our outcome variable Y may be binary, but the primary quantity of interest behind that outcome is p, a quantitative variable. This is why we consider logistic regression a regression method.

Logistic regression is one form of generalized linear regression. Recall that in linear regression we use data with response variable y_i and j predictors $x_{i1}, \ldots, x_{ij}$ to fit the best-fitting linear function

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_j x_{ij} + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2)$, by estimating $\beta_0, \ldots, \beta_j$. We can equivalently write,

$$Y \sim N(\beta_0 + \beta_1 x_1 + \ldots + \beta_j x_j, \sigma^2).$$

We could, in fact, try to predict p with a linear regression, though such a model will always have the problem of predicting probabilities outside of the range of $[0, 1]$. That's not a recipe for simple interpretation or reliable predictions. Generalized linear regression uses a "link function," g, to modify the regression so the range of the response more accurately reflects the practical range of the variable:

$$g(y_i) = \beta_0 + \beta_1 x_1 + \ldots + \beta_j x_j + \epsilon.$$

In logistic regression, the "link" function is the logit function, $\text{logit} : [0, 1] \to \mathbb{R}$,

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right).$$

This function is also known as the log-odds, because odds are defined as $p/1 - p$ for any probability p.

So in logistic regression, we model the probability of $y_i = 1$ as,

$$\mathbb{P}(y_i = 1) = \text{logit}^{-1}(\beta_0 + \beta_1 x_1 + \ldots + \beta_j x_j).$$

Notice that the inverse logit function[1] maps values from $\mathbb{R}$ to $[0, 1]$. Thus, the function provides a convenient way to map linear combinations of other variables onto values that are valid probabilities. Other such functions exist and are also used for regressions with binary responses, such as the probit function. Logistic regression, however, is easier to interpret and more efficient to compute.

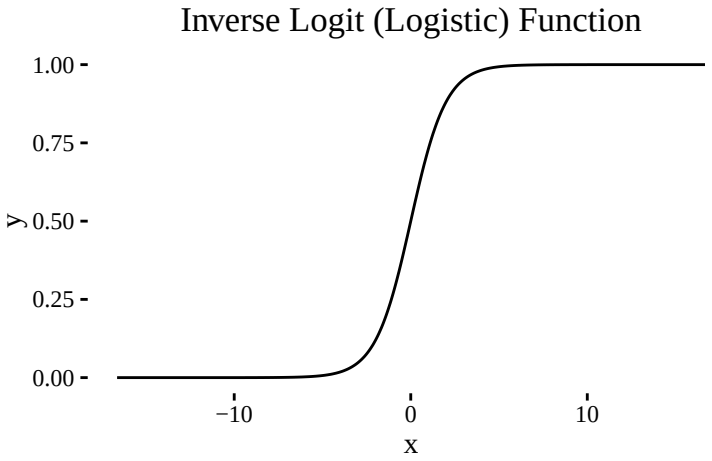

Figure 2.1: The inverse logit function gives a convenient way to map linear combinations of real numbers to valid probability values.

Though coefficients from a logistic regression can't be interpreted the same way as a linear model, they do have a convenient interpretation. Because the linear part of the model represents log odds, the coefficients are log odds ratios; That is, the exponentiated coefficients $e^{\beta_1}, \ldots, e^{\beta_j}$ represent the multiplicative effect a one-unit

[1] sometimes called the logistic function

increase in the corresponding predictor has on the odds. For example, if we fit a simple logistic regression with $\mathbb{P}(Y = 1) = \text{logit}^{-1}(\alpha + \beta + X)$, we could interpret $e^{\beta} = 1.1$ as meaning that a one unit increase in X gives a 10% increase in the odds that $Y = 1$.

We started out talking about the ways logistic regression is both a classification and regression model. We will mostly care about logistic regression as a regression model. One of the main ways we will assess our models is through the accuracy of predictions. But, in the classification framework, in order to predict the outcomes $\hat{Y}$, we need to choose a probability threshold to decide which outcome we should predict for a given observation. When we care mainly about prediction, this approach can make sense. Our interest, however, lies in prediction. So rather than choose an arbitrary threshold, we will evaluate the accuracy of our predicted probabilities themselves. (How exactly do we do this? See Section 2.3.)

2.2 Hierarchical Models and Mixed Effects Models

Data often contain hierarchies. For example, a set of student test scores may contain the hierarchy of districts and schools those students attend. Or a set of soil samples may have been taken at several distinct sites, thus having a hierarchy of sites and samples. In the bike ride data we examine, there is the hierarchy of riders and rides.

We will talk about different "levels" of variables corresponding to places in this hierarchy. When we refer to "ride-level variables," we refer to variables that are specific to a ride, whereas we refer to "rider-level variables" as those specific to the rider, and thus also all the rides that rider takes. For example, we consider length a ride-level variable and total number of rides taken a rider-level variable.

We will also discuss road segment-level variables, which are variables that are specific to the road segments in the route of a ride (e.g. length, presence of bike lanes, etc). But there isn't a clear road segment-ride hierarchy: each ride contains multiple road segments and each road segment is contained by multiple rides. Thus, this isn't a case where multilevel modeling is applicable. (The ideas behind it, though, may be fruitfully adapted.)

Gelman describes two traditional ways of dealing with hierarchical data that multilevel models contrasts with: "complete pooling" and "no pooling."[2] In "complete pooling," we ignore the group-level variables, and give identical estimates for parameters for every group. In "no pooling," we do entirely separate regressions for each group. Multilevel models are a comprimise between these extremes ("partial pooling", as Gelman calls it) where all the data is considered in a single regression with some parameters shared between groups and some different between groups. (This will become clearer when we introduce examples of models.)

These multilevel models work for other forms of regression, but we will focus on logistic regression, as it is the method we use in this paper. We will be using notation adapted from Gelman and Hill's description of multilevel models.[3] Consider a data set composed of

[2] Gelman & Hill (2006), p. 7

[3] Gelman & Hill (2006), p. 251–252

- i observations of a binary response variable y_i,
- m observation level predictors $X_i = x_i^1, \ldots, x_i^m$,
- j groups in which the observations are split into,
- l group-level predictors $U_{j[i]} = u_{j[i]}^1, \ldots, u_{j[i]}^l$, where $j[i]$ is the group of the ith observation.

We could fit a model where the intercept varies by group:

$$\mathbb{P}(y_i = 1) = \text{logit}^{-1}(\alpha_{j[i]} + X_i\beta),$$
$$\alpha_{j[i]} \sim N(\gamma_0 + U_{j[i]}\gamma, \sigma_\alpha^2),$$

where $\alpha_{j[i]}$ is the intercept for the jth group, β is a vector coefficients for the observation-level predictors, γ_0 are the group-level intercepts, and γ is a vector of coefficients for the group-level predictors. We could also specify a similar model where there are no group-level predictors, such that we simply have different intercepts for each group,

$$\alpha_{j[i]} \sim N(\gamma_0, \sigma_\alpha^2),$$

We can also consider a model that has slopes varying by group. For simplicity, let's consider just one observation-level predictor, x_i, that will have varying slopes $\beta_{j[i]}$ as well as one group-level predictor. We could specify the model as,

$$\mathbb{P}(y_i = 1) = \text{logit}^{-1}(\alpha_{j[i]} + \beta_{j[i]}x_i),$$

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} = N\left(\begin{pmatrix} \gamma_0^\alpha + \gamma_1^\alpha u_j \\ \gamma_0^\beta + \gamma_1^\beta u_j \end{pmatrix}, \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix}\right).$$

2.3 Tools for evaluating models

After fitting our models, we will want to know, how do each of our models compare? Did adding a particular term enhance or diminish the accuracy of our model? While we are fitting our models for inference, we will be evaluating them by the accuracy of their predictions. That is, how accurately and confidently can the model dicern which rides were rated as good and which as bad?

The predictions we will evaluate will be done in the context of cross validation. Usually, we will just use a simpy 2-fold cross validation, where we split the data into two random samples, fitting the model on one set (the "training set"), and using the model to predict on the other set (the "testing set") to check accuracy. We can also do k-fold cross validation, where we split the data into k samples, and for each of the samples we train the model with the other $k-1$ samples and then test with the left out sample.

When splitting the data into testing and training sets, we can not do a simple random sample. If we do, we will end up with riders in the testing set that aren't in the training set, which our models that use rider information cannot predict on. Thus, we will usually do stratified sampling of rides with riders as strata.

When evaluating the predictions from a model, statistics such as classifications rate, false-positive rate, and true-negative rate can be calculated for each validation. For a more comprehensive look at predictive accuracy, we use the separation plot as well as homemade plot to assess our models.

2.3.1 The Separation Plot

The separation plot, created by Greenhill et. al.[4], is designed to show how well a logistic regression model can distinguish between high and low probability events.

Creating a separation plot first requires a model fit to training data and testing data to evaluate predictive accuracy on. From the testing data, we need a vector Y of observed binary response data and a vector $\hat{Y}$ of predicted probabilities of a 1 for each observation, predicted using our model fitted to training data.

We plot the data $(Y, \hat{Y})$ as a sequence of vertical strips, colored according to observed outcome, Y, and ordered from low to high probability based on $\hat{Y}$. A curve is superimposed upon the stripes showing the $\hat{Y}$ as a line graph. And finally, a small triangle is placed indicated the point at which the two colors of lines would meet if all observations $Y = 0$ were placed to the left of all the $Y = 1$ observations; in other words, showing where the boundary would be if the two classes were perfectly separated by the model.

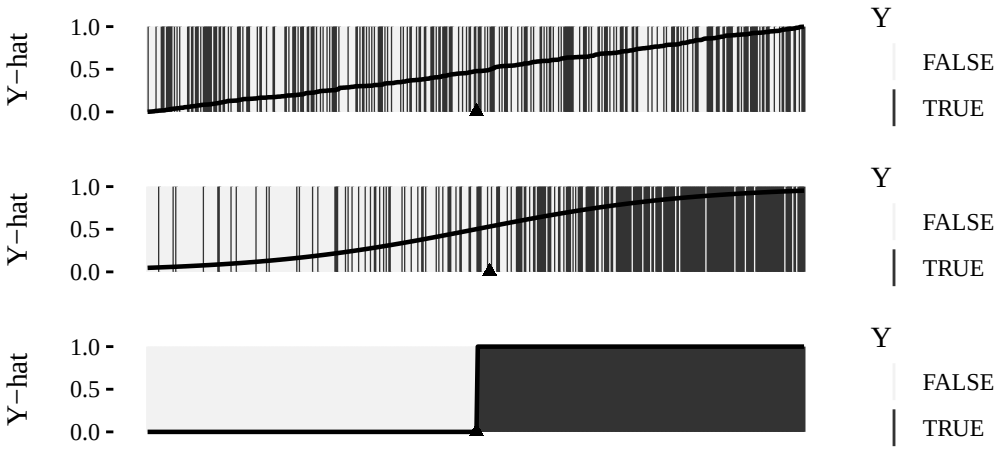

Figure 2.2: Above we have examples of three separation plots. The first plot shows what it looks like when Y and $\hat{Y}$ are uncorrelated. The second plot shows a fairly good model, where the Y are generated as Bernoulli($\hat{Y}$). The third plot shows a model where the responses are fully separated.

Separation plots don't do well with larger sample sizes: if there are two many observations, it becomes difficult to read. This is because the resolution of the medium the plot is displayed on may not be fine-grained enough to show a single observation's

[4]Greenhill, Ward, & Sacks (2011)

line, thus obscuring the pattern. In these cases, we can plot a random sample or modify the graph to be more suitable.

2.3.2 Our Probability Plots

One graphical tool we adapted for this project maps probabilities of the outcome against continuous variables. We use this for exploration to understand the relationship between predictors and our binary response, but it's also useful for evaluating our models.

The goal of the plot is to show a measure of an emperical $\hat{p}$, the probability that the response is 1, on the y-axis, and a continuous variable X on the x-axis. A simple approach would be to bin the observations by the values of X, and within each bin, model the data as a bernoulli random variable and compute a local value for p.

The problem with the binning approach is that we get some error due to how we happen to bin things. An alternative is to do the same computation for a sliding window rather than a bin. This reduces the error due to our binning, but we still have to choose a proper window width. We will choose the window width that minimizes the leave-one-out error (the sum of the squares of the error in predicting a single observations based on a regression fitted to the rest of the observations.)

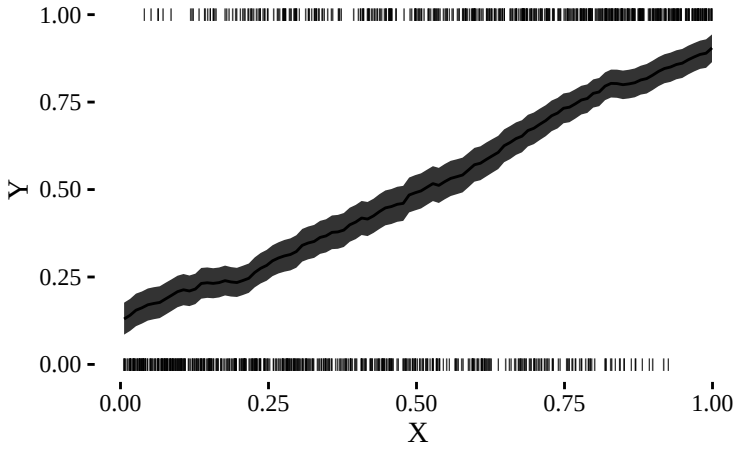

Figure 2.3: Example of probability plot. 1000 Xs were sampled from the distribution Uniform$(0, 1)$ and each Y_i was generated from the distribution $Y_i \sim$ Bernoulli(X_i). The smoothed curve shows an approximation of p for each value of X_i, with 90% confidence intervals in grey.

We also make use of this plot to compare the predicted probabilities for a testing set to the empirical probablilities in the data. J. Esarey and A. Pierce advocate using this in their heatmap plot to assess model fit, showing that such a plot will show model mispecification problems better than other measures commonly used, like AIC.[5]

[5]Esarey & Pierce (2012)

Chapter 3

Modeling Rides and Riders

Complex statistical models can accurately model intricate processes. But they also run the risk of overfitting to data. To avoid this, we build up our models from simple to complex, comparing the models with cross validation to make sure the complexities introduced add real value.

In this chapter we focus on building models that incorporate information about rider, weather conditions, time of day, and ride length. In brief, our models start with a logistic regression model considering only ride-level variables, and formulate more complex models by adding various terms. Table 3.1 describes each model briefly along with the models label.

Table 3.1: List of models we evaluate in this chapter.

Model	**Description**
Model 1	(Baseline) Classical logistic regression
Model 2	Add rider intercepts
Model 3	Add trigonometric terms for time of day
Model 4	Additive model with cubic cyclic spline for time of day
Model 5	Additive model with spline for ride length
Model 6	Remove random rider intercepts from Model 4

3.1 The Models

Model 1, which we will use as the baseline for comparing further models, is a multivariate logistic regression model. Our set of predictors are

- X^{length}, standardized ride length
- X^{rain}, rainfall during hour of ride, in tenths of inches
- X^{rain4h}, rainfall during past four hours before ride, in tenths of inches
- X^{gust}, gust speed for the day, in miles per hour
- X^{temp}, average temperature, in degrees Fahrenheit.

These $p = 5$ predictors will be our standard ride-level predictors in this chapter.

We will denote the predictor matrix X, with each predictor a column of the matrix:

$$X = (X^{\text{length}}\ X^{\text{rain}}\ X^{\text{rain4h}}\ X^{\text{gust}}\ X^{\text{temp}}).$$

Let $Y_i = 1$ if the ith ride received a negative rating, and $Y_i = 0$ if it received a positive rating. Then, our first model will be,

$$\mathbb{P}(Y_i = 1) = \text{logit}^{-1}(\alpha + X_i\beta),$$

where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^p$ are parameters to be estimated.

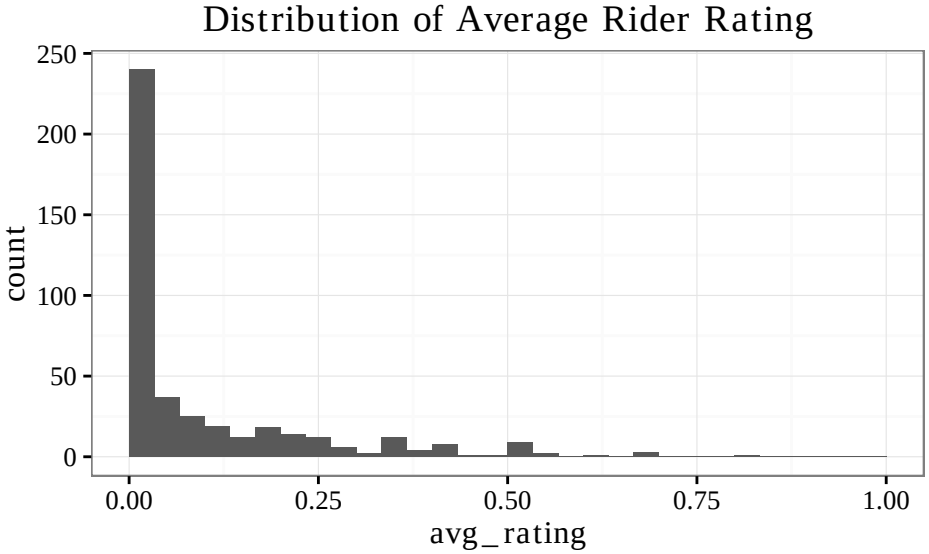

Figure 3.1: The overall rates at which each rider gives a negative rating for a ride varies greatly. This is our primary motivation for including rider intercepts and predictors.

Riders appear to have different tendencies to rate rides negatively more often, as we note in Figure 3.1. In fact, many riders give zero or nearly zero negative ratings. For **Model 2**, we account for this by adding intercepts that vary by rider:

$$\mathbb{P}(Y_i = 1) = \text{logit}^{-1}(\alpha + \alpha_{j[i]} + X_i\beta).$$

Rider intercepts themselves aren't as interesting than how they deviate from the mean, so we actually keep a fixed intercept α and constrain the rider intercepts, α_j, by specifying

$$\alpha_j \sim N(0, \sigma_\alpha^2).$$

Starting with **Model 3**, we address time of day, $t \in [0, 24)$ as a predictor. (We measure time of day in hours since midnight.) We use time of day to account for the various daily trends that may affect ratings, including as a simple way to model the overall traffic level, which is difficult to model on it's own. These patterns are cyclic and very non-linear, which means we have to be a little more creative in how

we incorporate them into our model. One approach is to add sinusoidal terms with a period of one day. We would be interested in fitting a term,

$$\beta \sin(Tx^{\text{time}} + \phi),$$

where β and ϕ are coefficients estimated and $T = 2\pi/d$, where d is the period of one day (24 hours.) This form isn't easy to estimate, but we can transform this expression into the sum of two trigonometric functions:

$$\begin{aligned}\beta \sin(Tx + \phi) &= \beta \left(\sin(Tx)\cos(\phi) + \cos(Tx)\sin(\phi)\right) \\ &= (\beta + \cos(\phi))\sin(Tx) + \sin(\phi)\cos(Tx) \\ &= \beta_1 \sin(Tx) + \beta_2 \cos(Tx),\end{aligned}$$

where $\beta_1 = \beta + \cos(\phi)$ and $\beta_2 = \sin(\phi)$. To add a little more flexibility, we can also add another sinusoidal term with half the period.

We also want to take into account that weekday patterns may be different than weekend hourly patterns. We also have a variable X^{weekend} that serves as a weekend indicator. For Model 3, we add two sets of sinusoidal terms: one set for weekdays and one for weekends. More explicitly, we define the model,

$$\begin{aligned}\mathbb{P}(Y_i = 1) = \text{logit}^{-1}(&\alpha + \alpha_{j[i]} + X_i\beta \\ &+ X^{\text{weekend}} \cdot [\beta^{t1}\sin(T \cdot t) + \beta^{t2}\cos(T \cdot t) \\ &+ \beta^{t3}\sin(T/2 \cdot t) + \beta^{t4}\cos(T/2 \cdot t)] \\ &+ (1 - X^{\text{weekend}}) \cdot [\beta^{t1}\sin(T \cdot t) + \beta^{t2}\cos(T \cdot t) \\ &+ \beta^{t3}\sin(T/2 \cdot t) + \beta^{t4}\cos(T/2 \cdot t))].\end{aligned} \tag{3.1}$$

For **Model 4** and **Model 5**, we abandon parametric methods and use a cyclic non-parametric smoother to model time of day. The only problem is that we need a way to combine our parametric and multilevel parts of the model with a new non-parametric part. This is where additive models come in.

3.2 Additive Models and Smoothing Splines

We want to explore using non-parametric methods to model the relationship with time and length, but we wish to keep the other parts of our model. We can do this with an additive model. Additive models assume that the response is the sum of functions of each of the predictors:

$$\text{logit}(\mathbb{P}(y_i = 1)) = \alpha + \sum_{j=1}^{p} f_j(x_{ij}).$$

These functions can be linear, so generalized linear regression is a subset of additive

models. But more interestingly, these functions can be non-parametric.[1] One of the most common types of functions fit are smoothing splines.

Smoothing splines are essentially cubic functions stitched together at points called "knots" such that the full piece-wise function is continuous and has continuous first and second derivative. One can further define cyclic cubic splines, which simply have the constraint that the last knot and first knot are treated as the same, thus allowing a continuous cyclic function to be fit.[2]

Computation of multilevel additive models with splines is available in the `gamm4` package, which we use to fit the two following models.

Model 4 will introduce a cyclic cubic splines for time of day.

3.3 Model Evaluation

To fit the data, we got all of the rides in Portland, OR from [insert data here] to [insert date here]for riders that had over 20 rides. There were 35,370 rides, 14,032 of which were rated. Overall, 10.88 percent of these rides were given a negative rating. There were 518 riders in the data set.

All six models were fit twice: first to the full data set to get good estimates of the parameters, and second to a sample of 80 percent of the rides. We used the latter fit to make predictions for the remaining 20 percent of rides, to determine the predictive accuracy for out-of-sample observations. (The sample had to be stratified by rider, to guarantee that every rider had at least on ride in the training set; otherwise models with random intercepts would be unable to make predictions for some rides.)

The separation plots in Table 3.2 give a clear initial picture of how these model fits compare. Model 1 performs very poorly compared to those that include rider intercepts, assigning the same probability to most observations. Models that include the rider intercept perform similarly. The log likelihoods and AIC scores, shown in Table 3.2, corroborate this. Adding time dependency doesn't seem to impact predictive ability. We will see later, however, that it gives a fascinating result to interpret.

The gains from the rider intercepts are great, but we are compelled to ask: how much of that gain could have been achieved with randomly chosen groups? *i.e.* if riders were randomly assigned to rides, would the flexibility in the model created by allowing intercepts to vary increase predictive performance to the same degree? To test this, we ran a Model 4 after we randomly assign the rides a rider. This quick test nullified this skepticism, as you can see in the resulting separation plots in Figure 3.2.

[1] How are these models fit? Using what's known as the Backfitting Algorithm. We define the kth partial residuals $Y^{(k)} = Y - \left(\alpha \sum_{j \neq k} f_j(x_j)\right)$. (That is, define the portion of Y leftover for $f_k(x_k)$ to fit to after the other f_j's have had their share.) Then, iteratively fit each of the functions f_j on the partial residuals $Y^{(j)}$ until each of the functions converge. For a further quick look at additive models, check out Cosma Shalizi's lecture notes (Shalizi (2013a))

[2] For a brief and entertaining introduction to smoothing splines, see Shalizi (2013b). For a more in-depth look at splines, check out Wood (2006)

Table 3.2: Model fit summaries for full-data fittings.

Model	Separation Plot	$\mathcal{L}$	AIC
Model 1		-4,788	9,589
Model 2		-3,971	7,956
Model 3		-3,923	7,876
Model 4		-3,931	7,879
Model 5		-3,928	7,877
Model 6		-4,716	9,458

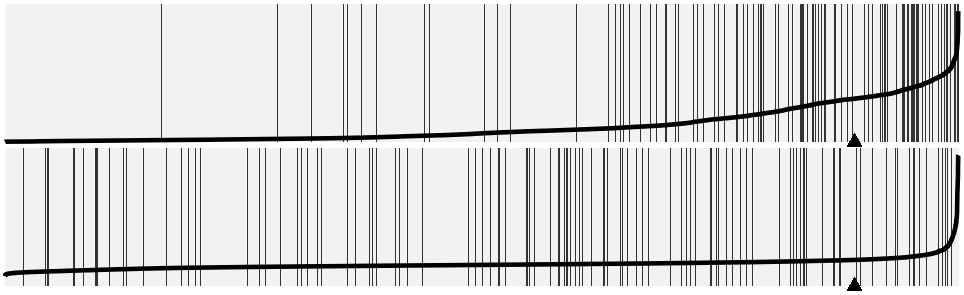

Figure 3.2: Separation plots for models 2 compared to a similar model where riders are randomly assigned to rides.

3.4 Model Results

Table 3.3 presents the fixed effect estimates for our models.

The marginal fits for time of day, shown in Figure 3.3, are predictable. On weekdays, the probability of a negative rating peaks in the morning and afternoon, around when we expect rush hour traffic, and on weekends it stays steady throughout the day. While Model 4 and Model 5 give similar fits for time of day, Model 3's predictions peak at different times on weekdays and exhibit much more variability on weekends. There are two probable reasons for these differences: first, the sinusoidal terms are less flexibile than the splines; second, the splines, because they are a non-parametric functions, penalize complexity of the fit while the parametric sinusoidal form does not, making the splines more conservative in their "curviness." The former explains the discrepancies in the weekday fits while the latter explains the discrepancies in the weekend fits. Given these differences, fitting time of day with splines is preferable; there is no motivation to constrain the functional form to any strict parametric form.

But these marginal time-of-day fits don't just tell a story about our time terms; they also reveal part of why the random rider intercepts are such powerful predictors. Notice that in Figure 3.3, the scale at which the Model 6 time fitted probabilities vary is much larger than the scale at which the other models' predictions vary. (The

Table 3.3: Regression coefficients for Model 1, Model 2, Model 4, and Model 6. 95% confidence intervals are given in parentheses.

Regression Term	Model 1	Model 2	Model 4	Model 6
Log(length)	-0.122	-0.100	-0.092	-0.114
	(-0.180, -0.063)	(-0.168, -0.033)	(-0.163, -0.022)	(-0.174, -0.054)
Mean Temp.	0.042	0.071	0.071	0.059
	(-0.0014, 0.098)	(0.001, 0.141)	(0.000, 0.142)	(0.002, 0.115)
Gust speed	0.007	0.006	0.005	0.007
	(-0.0003, 0.014)	(-0.002, 0.013)	(-0.003, 0.013)	(-0.019, 0.027)
Rainfall	0.008	0.012	0.008	0.004
	(-0.015, 0.031)	(-0.015, 0.038)	(-0.019, 0.035)	(-0.019, 0.027)
Rainfall 4-Hour	0.014	0.016	0.017	0.015
	(0.006, 0.022)	(0.007, 0.025)	(0.008, 0.027)	(0.007, 0.023)
Intercept	-2.284	-3.081	-3.133	-2.328
	(-2.443, -2.125)	(-3.392, -2.770)	(-3.441, -2.824)	(-2.489, -2.168)

differences is so large we had to show them in separate plots!) Without allowing for varying rider intercepts, the time terms take on a significant role. Yet, interestingly, the time term has nowhere near the amount of information that the rider intercepts seem to encode, according to the separation plots in Table 3.2.

A clearer picture of what is going on is painted in Figure 3.4. Model 1's predictions, which has a fixed intercept and no time dependence, don't vary by time of day. The models with random intercepts (2–5) show strong temporal patterns, with concentrated spikes in the morning and evening. Model 6, which had the time of day spline but a fixed intercept, has a temporal pattern, but does not represent the same degree time dependence that the random intercept models do.

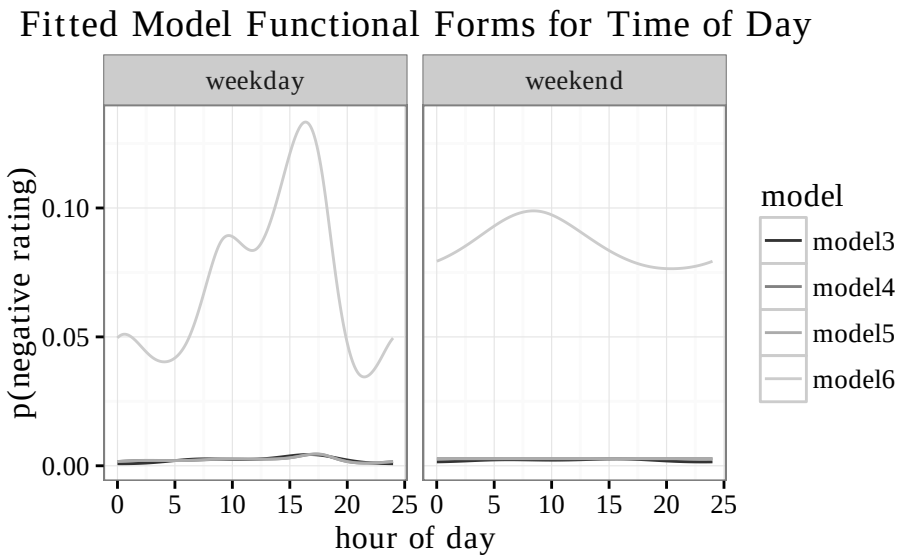

Figure 3.3: Predicted probabilities of a negative rating by time for a typical ride. The rider was chosen so the intercept was closest to the mean intercept for model 6. The median length and average mean temperature were used, and all other predictors were set to zero.

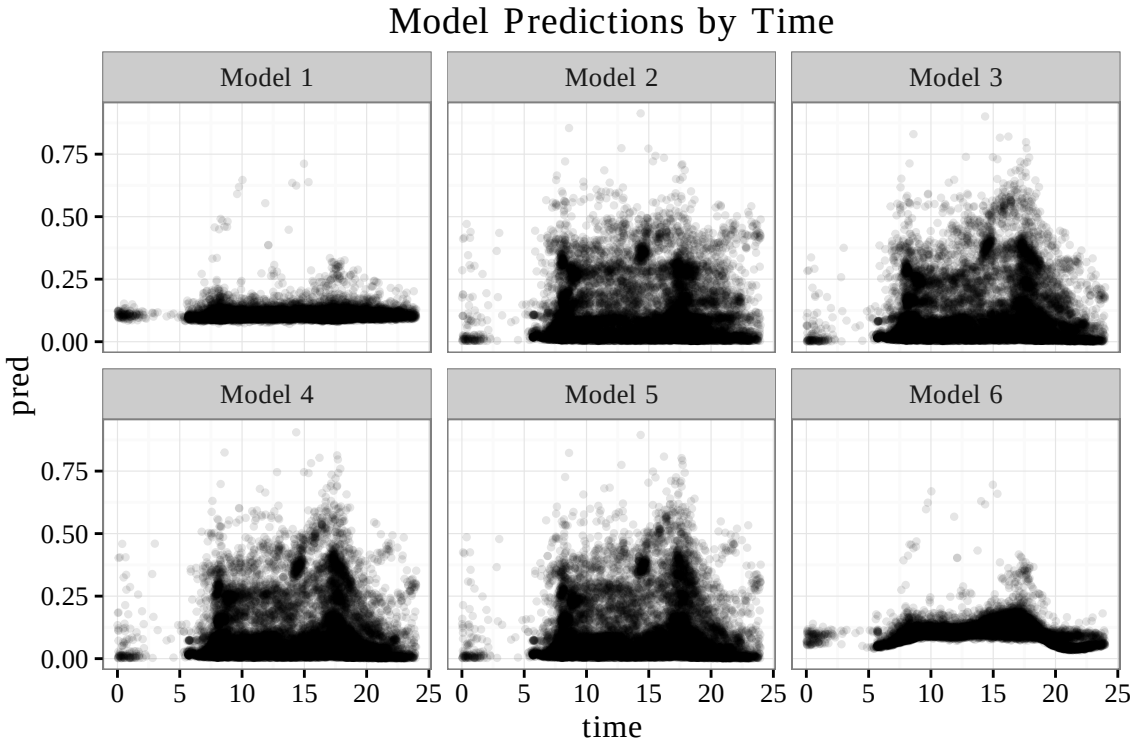

Figure 3.4: Each of the models predictions for all rides with time of day on the x-axis. Notice how starting with model 2, daily trends start to emerge. This indicates that the rider intercepts are picking up on time of day trends, which must be reflectedd in riders typical ride.

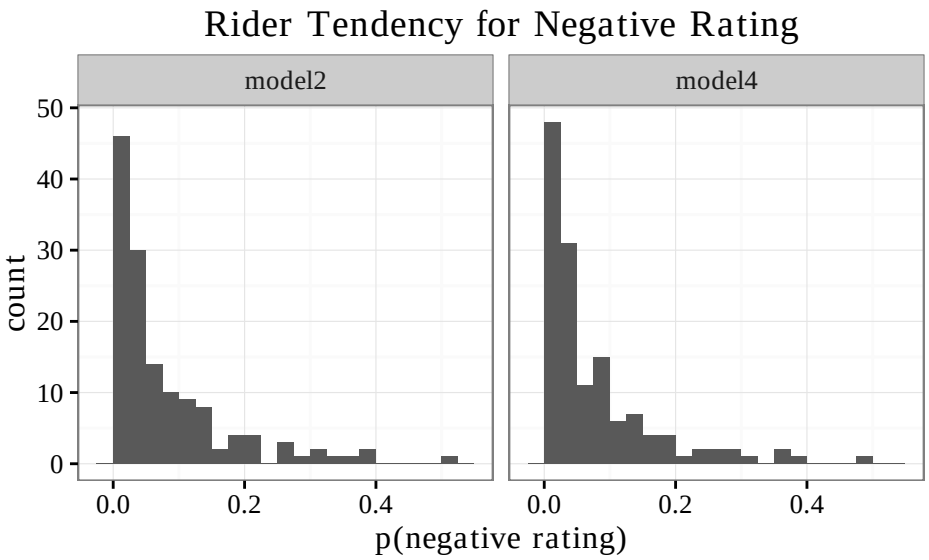

Figure 3.5: Predicted probability of a negative rating for a typical ride for each rider, for Model 2 and Model 4. The typical ride is a ride at noon on a weekday with median length, mean temperature, and other variables at zero.

Chapter 4

Classifying Riders

We don't expect riders to be the same. In fact, we saw this with our models in Chapter 3: the rider intercepts encoded a lot of information about the typical time of day a rider tends to ride. In this chapter we try to get a picture of how these patterns in rider behavior vary between riders, and then determine if this gives useful information to our model as group-level predictors.

4.1 Characterizing Rider Length and Time Patterns

Sometime we would like to give a summary of patterns we see in data. Sometimes statistical summaries like the mean or median are useful. But take the example of the times of day a rider goes on bicycle rides. Is the mean or median time useful at all? Not really.

A good description of a riders time of day patterns might be the proportion of their rides fall within each our of the day. This gives us a sort of distribution for their riders time of day. The question we can answer from there is: Can these patterns be described with good accuracy with a lower dimensional vector?

Define the number of rides in hour $h \in \{0, \dots, 23\}$ for rider j as

$$n(h, j) = \left| \left\{ i \mid X_i^{\text{rider}} = j, \lfloor X_i^{\text{time}} \rfloor = h \right\} \right|$$

(Recall that $X^{\text{time}} \in [0, 24)^n$, i.e it is measured in hours since midnight.)

We can then define the rider time of day matrix as P, where

$$P_{ij} = \frac{n(i, j)}{\sum_{h=0}^{23} n(h, j)}.$$

Now we can ask, what are the principal components of these column vectors?

4.2 Models with Rider-level predictors

Chapter 5

Modeling Missing Response

We have a significant portion of observations with no response. But we do have all the predictors for every observation. So just as we built a model to predict the rating, can we build a model to predict whether a rider will give a rating? Going further, could we combine our rating model and nonresponse model into one model, such that our predictions for ratings will take into account biases in nonresponse?

We attempt to address these two questions in this chapter. The reader should be warned, however, that this exploration is cursory. While modeling nonresponse is important in understanding the data it is not our main goal here. That being said, a more in-depth exploration of nonreponse in the Ride Report app would be worthwhile.

5.1 What could possibly go wrong?

Let look at a toy example. Let,

$$X \sim \text{Normal}(0, 1),$$

$$Y \sim \text{Binomial}(\text{logit}^{-1}(4X)),$$

$$R \sim \text{Binomial}(0.3 + 0.8Y).$$

Now, let Y_i be NA (a missing value) if and only if $R_i = 1$.

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

5.2 Using Expectation Maximization

We can try to perform the expectation maximization algorithm here, using the weighting method proposed by Ibrahim and Lipsitz[1]. Let y_i be our binary response and x_i be our predictors. With these we have our complete data logistic regression model $f(y_i \mid x_i, \beta)$, where β is a vector of parameters in the complete data model. Now define

[1] Ibrahim & Lipsitz (1996)

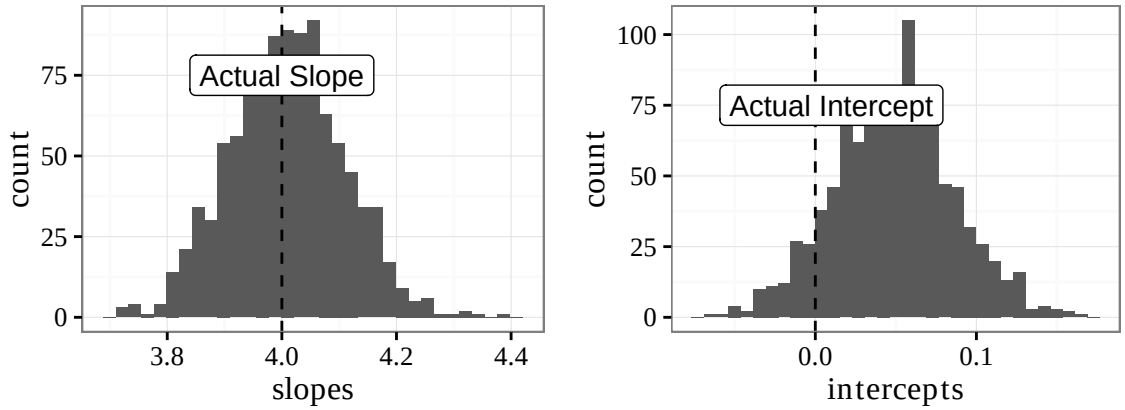

Figure 5.1: Missing response when missingness is correlated with the response leads to biased estimates in the complete case model of the intercept, but not the slopes. Here, we bootstrapped estimates of the slope and intercept with each bootstrapped dataset a different pattern of missing data for the same original full data.

$$r_i = \begin{cases} 1, & \text{if } y_i\text{is missing;} \\ 0, & \text{if } y_i\text{is observed;} \end{cases} \tag{5.1}$$

for $i = 1, \ldots, n$. We then specify a logistic regression model for missingness (the r_i's): $f(r_i \mid x_i, y_i, \alpha)$, where α is the vector of parameters in the missingness model.

We begin the algorithm by getting our first estimates of α and β. We obtain $\beta^{(1)}$ by estimating β with only the non-missing data. We can then estimate the y_i for the missing data using $\beta^{(1)}$, and then use those estimates to compute $\alpha^{(1)}$.

For the E-step, we compute weights for each observation with missing response

$$w_{i\,y_i,(t)} = f(y_i \mid r_i, x_i, \alpha^{(t)}, \beta^{(t)}) = \frac{f(y_i \mid x_i, \beta^{(t)}) f(r_i \mid x_i, y_i, \alpha^{(t)})}{\sum_{y_i=0}^{1} f(y_i \mid x_i, \beta^{(t)}) f(r_i \mid x_i, y_i, \alpha^{(t)})}. \tag{5.2}$$

(For observed responses, $w_{i\,y_i,(t)} = 1$.) We can compute $f(y_i \mid x_i, \beta^{(t)})$ and $f(r_i \mid x_i, y_i, \alpha^{(t)})$ by making use of predictions from regression models. So in *R*, we can fit models and use the `predict()` function to get our probabilities from each of these models.

For the M-step, we find our next estimates of the parameters, $\alpha^{(t+1)}$ and $\beta^{(t+1)}$, by maximizing

$$Q(\alpha, \beta \mid \alpha^{(t)}, \beta^{(t)}) = \sum_{i=1}^{n} \sum_{y_i=0}^{m_i} w_{iy_i,(t)} l(\alpha, \beta; x_i, y_i, r_i). \tag{5.3}$$

We do this by first by estimating $\beta^{(t+1)}$ using weighted maximum likelihood for the complete data model, and then estimating $\alpha^{(t+1)}$ using the same method. The nice things here is we can simply make use of `glm` in *R* to do these computations. In

order to create the data to fit these models, we create an augmented data set where for each observation missing the response, we record as two rows for the two possible values and add a column for weight.

Figure 5.2: Creation of augmented data set for the weighted method of the EM algorithm for missing response data.

Original Data				**Augmented Data**			
y_i	x_i	r_i		y_i	x_i	r_i	w_i
1	2.4	0		1	2.4	0	1
0	1.3	0	$\rightarrow$	0	1.3	0	1
NA	-0.4	0		1	-0.4	0	0.2
				0	-0.4	0	0.8

We repeat the E and M step until the values of α and β converge.

As an example, we simulated a dataset from the same model we presented earlier of size 10^5. As shown in Table 5.1, the estimate for the intercept in the model that only considers the complete data is way off, but the model resulting from the EM algorithm is just as accurate as the model computed fit to the full data (with missing values filled in from the original data model.)

Table 5.1: Coefficients for models fit to simulated data set ($\pm$ twice the standard error.)

Model	$\hat{\beta}_0$	$2 \cdot SE_{\hat{\beta}_0}$	$\hat{\beta}_X$	$2 \cdot SE_{\hat{\beta}_X}$
Actual	0	—	4	—
Full Data Model	0.030	0.021	4.009	0.052
Complete Data Model	-0.543	0.038	4.047	0.093
EM Final Model	-0.017	0.021	4.059	0.053

Let's extend this model a little bit. Let

$$X_1 \sim \text{Normal}(0, 1), \quad X_2 \sim \text{Normal}1, 2,$$

$$Y \sim \text{Binomial}(\text{logit}^{-1}(0.4X_1 + 0.3X_2)),$$

$$\psi \sim \text{Binomial}(0.3 + 0.1Y + 0.2X_2).$$

Okay, so the intercepts are biased. But for such a subjective measure like rider rating, why should we care if this is biased? Well what about the rider intercepts? Are their estimates biased in a wierd way? Let's try a random intercept model.

Chapter 6

Unfinished Work: Incorporating Routes

These models so far do not incorporate route. Unfortunately, I was not able to get to models that actually do. The data transformations necessary were more than I could reasonably do in addition to the other parts of developing the models. So I leave the models as they are, but here I explain some of my work toward the goal of incorporating routes and describe some potential modeling approaches.

6.1 Regression Terms for Road Segments

Now we have the task of incorporating our knowledge of riders' routes into our regression. Our approach here will be to consider routes as sequences of discrete road segments, each of which have known properties. This is convenient because we have such data about roads that give us bike lanes, road size, etc. It is even possible for us to calculate popularity of particular segments easily.

Assume we have K total road segments in our road network and for each ride we have $\Omega_i \subseteq \{1, \ldots, K\}$, the set of road segments that are in the route of ride i. Let l_k be the length of the kth segment and define the length of ride i to be:

$$L_i = \sum_{k \in \Omega_i} l_k.$$

For the kth road segment, define the m-dimensional vector $W_k = W_k^1, W_k^2, \ldots, W_k^m$ road segment-level predictors. Then we shall define the term in our regression for the route of ride i as

$$R_i = \frac{1}{L_i} \sum_{k \in \Omega_i} l_k W_k \beta^{\text{road}},$$

Where β^{road} is a vector of coefficients for the road segment level predictors. When actually computing this value, it may be convenient to factor out the β^{road}

Conclusion

If we don't want Conclusion to have a chapter number next to it, we can add the `{.unnumbered}` attribute. This has an unintended consequence of the sections being labeled as 3.6 for example though instead of 4.1. The LaTeX commands immediately following the Conclusion declaration get things back on track.

More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

References

Cressie, N., & Wikle, C. K. (2011). *Statistics for spatio-temporal data*. John Wiley & Sons.

Esarey, J., & Pierce, A. (2012). Assessing fit quality and testing for misspecification in binary-dependent variable models. *Political Analysis*, *20*(4), 480–500.

Gelman, A., & Hill, J. (2006). *Data analysis using regression and multilevel/Hierarchical models*. The Edinburgh Building, Cambridge CB2 8RU, UK: Cambridge University Press, New York.

Greenhill, B., Ward, M. D., & Sacks, A. (2011). The separation plot: A new visual method for evaluating the fit of binary models. *American Journal of Political Science*, *55*(4), 991–1002.

Ibrahim, J. G., & Lipsitz, S. R. (1996). Parameter estimation from incomplete data in binomial regression when the missing mechanism is nonignorable. *Biometrics*, *52*(3), 1071–1078.

Mackaronis, J. E., Strassberg, D. S., Cundiff, J. M., & Cann, D. J. (2013). Beholder and beheld: A multilevel model of perceived sexual appeal. *Archives of Sexual Behavior*.

Shalizi, C. (2013a, February). Additive models. Retrieved from `http://www.stat.cmu.edu/~cshalizi/uADA/13/lectures/ch09.pdf`

Shalizi, C. (2013b, February). Splines. Retrieved from `http://www.stat.cmu.edu/~cshalizi/uADA/13/lectures/ch08.pdf`

Wood, S. (2006). *Generalized additive models: An introduction with r*. CRC press.