Homework Assignment 5

Max H. Farrell – Chicago Booth
BUS41100 Applied Regression Analysis

Due at the beginning of class of week 8

1 Pricing Experiment

Return to the pricing experiment we studied in class.

(a) Create the same 2×2 table that we made in class. Use this information to compute the intercept and slope of a **linear regression** of `buy` on `price`. Show your steps. Compare these steps to what we did for logistic regression in class.

(b) Run the logistic regression and obtain the intercept and slope. Compare the slope estimates, linear versus logistic, in sign and magnitude.

The data includes a variable `customerSize` that gives the size of the customer firm (remember, the customers of this business are themselves firms). The sizes are coded 0, 1, 2, for small, medium, and large firms.

(c) Create the 2×2 table of `buy` and `price` for medium sized firms. Compute the slope coefficient of a logistic regression for only this subgroup.

(d) Compare your answer above to the slope coefficient we found in class, which was for all firms.

(e) Use `glm` to run a logistic regression for only medium sized firms. What do the measures of statistical uncertainty tell you for this subgroup? Do the same for large firms.

(f) Include `price` and `customerSize` into one logistic regression, using all firms. That is, fit the model

$$\mathbb{P}[\texttt{buy} \mid \texttt{price}, \texttt{customerSize}] = \frac{e^{\beta_0+\beta_1\texttt{price}+\beta_2\texttt{customerSize}}}{1+e^{\beta_0+\beta_1\texttt{price}+\beta_2\texttt{customerSize}}}.$$

Interpret the coefficient row for `customerSize`, both the estimates and the measures of uncertainty. Compare to what you found in the previous part, running subgroup-specific models.

(g) Now include the interaction of `price` and `customerSize`. Describe what you find and how it compares to the results from part **(f)**.

(h) Now treat `customerSize` as a factor variable. Run the main effects model, akin to **(f)** and the interactions model, akin to **(g)**. Comment on what you find here and how it compares to those prior parts.

2 Flight Delays

We actually get to to a little box 1 work in this problem.

Our goal in this problem is to understand flight delays using regression techniques. We may wish to optimize the flights schedule or flight paths, or we could be a planner in charge of a single airport or airline. The file `flights.rda` has data from 2013 on 308,879 flights departing from the three major New York City area airports: `JFK`, `LGA`, and `EWR`. Further, the file `weather.rda` has weather information. We will use both of these files. (This data comes from `https://blog.rstudio.com/2014/07/23/new-data-packages/`.)

Your first job is data preparation. You need to load the files in and merge them. These are *not* csv files. You can load an R data set using `load("data file goes here")`, with all the usual disclaimers about paths and file names.

(a) First, load both data sets and look at a `summary` of each. Examine the character variables and convert to factor variables as needed (no need to convert to factors variables you won't use).

(b) To merge the data sets we first need to figure out what we are merging by. What variables index each row of the two data sets? That is, what variables uniquely identify each row of each data set? Of these identifiers, what set does it make sense to merge with? You need a set that is present in both data sets and uniquely identifies the rows in at least one data set.

Coding help: you may find the commands `unique`, `duplicated`, and `anyDuplicated` useful. For example, try `anyDuplicated(weather$date)`. Hmm, lots of duplicates. What is the most times we see each date? Answer:

```
> max(table(weather$date))
[1] 3
```

Why do we see each date at most three times? Because we have for three airports: `JFK`, `LGA`, and `EWR`. Now try `anyDuplicated(weather[,c("origin", "date")])`.

(c) Use the unique identifiers you just found to merge the data sets. Use the `merge` function:

```
full.data <- merge(flights, weather, by=c("origin", "another_variable", "a_third_variable", ...)
```

Before you merged, did you even look at the variables you merged by? I bet you didn't. Whoops. Check `table(flights$hour)`. What's the problem? Justify your fix for the problem and re-merge the data.

Now that our data is merged, we will proceed to the analysis.

(d) Use `dep_delay` and `arr_delay` to create a new variable `any.delay` that is zero if the flight departs on time (or early) and arrives on time (or early), like this:

```
full.data$any.delay <- (full.data$dep_delay>0) | (full.data$arr_delay>0)
```

Run a logistic regression to predict `any.delay` using the categorical `origin` and excluding a constant, like this:

```
glm(any.delay ~ origin-1, family="binomial", data=full.data)
```

(e) In the summary output from that model there's a line that doesn't belong, something like this:

```
(XXXX observations deleted due to missingness)
```

Did you even notice the missing values when you ran `summary` back in part **(a)**? I bet you didn't. Whoops. Let's investigate. Use the `is.na` function to find the missing data. Where did the problem originate? Track it back as far as you can. Try `table(is.na(full.data$arr_delay))`. Does that give you the same number of missing observations that were reported in your logistic regression? Why or why not? What are you going to do about the missingness?

(f) Ignoring or fixing the missingness problem, use the output of the logistic regression to compute how much more or less likely to be delayed is a flight leaving from `EWR` compared to one leaving from `JFK`.

(g) Use a series of logistic regressions similar to the one in part **(d)** to study how the different airports do, relative to each other, for delays of > 0 minutes, ≥ 10 minutes, $\geq 20 minutes$, and so forth, up to ≥ 60 minutes. What do you find about the different airports?

(h) Now examine only departure delays. Is there any difference between what you find now versus looking at any delay above?

We will stick with departure delays from now on

(i) Make a scatter plot of the departure delay in minutes against the different weather measurements. Use these plots to determine which weather variables you want to put into a *linear* model to predict/explain departure delay in minutes.

Coding help: making a scatter plot with this many data points is often not helpful. First, it takes forever. Second, you wind up with a black cloud. Let's try two other things.

(1) Make boxplots, which at least shows the conditional median. Make a continuous variable into a rounded version of itself using `round` or `floor`. For example:

```
rounded.temp <- 5*floor(full.data$temp/5)
```

Now make a boxblot with `rounded.temp` on the x-axis. Try limiting the range of the y axis to show more detail.

(2) Make scatter plots with only a sample of the data. Use the command `sample.int` to get a random sample of 5% or 10% of the full data, and use this for plotting.

(j) Build a multiple linear regression using the variables above. Discuss the output.

(k) Use the same independent variables to now predict the binary outcome of any departure delay and the binary outcome of a large departure delay (set your own threshold in minutes for what "large" means). Comment on any differences or similarities that are notable between the linear and logistic regressions, in either sign or magnitude of the coefficients.

Use this as an opportunity to practice your understanding of coefficients in logistic regressions and odds ratios!

3 Hourly Bike Sharing

From Capital Bikeshare (D.C.'s Divvy) we have *hourly* bike rentals & weather info (the data used in week 5 was a collapsed version of this data). See `bikeSharing.csv`. The data are from 2011 & 2012, and we have the first 19 days of every month.[1] We have the number of rentals from registered and non-registered users (`registered` and `casual`, respectively). The available information is

- `datetime` – the date and hour, formatted as "2011-04-12 23:00:00" for 11PM, April 12, 2011
- `season` – Spring, summer, fall, winter
- `holiday` – Is the day a holiday?
- `workingday` – Is it a work day (not holiday, not weekend)?
- `weather` – coded 1=nice, 2=OK, 3=bad, 4=terrible
- `temp` – degrees Celsius
- `atemp` – "feels like" in Celsius
- `humidity` – relative humidity
- `windspeed`

(a) Examine the data for outlier patterns in the weather and remove them. Give clear justification for each data point removed.

(b) Split the data into training and testing samples (size and sampling scheme is up to you). We won't use the test set until the very end.

(c) Treat the hour of the day as a factor variable (which takes 24 different levels). Using linear regression, study which hours of the data are the most important for predicting bike rental demand. Does the pattern vary between registered and non-registered users? Do the important hours change with the season? *(You may find the commands* `substr` *and* `model.matrix` *useful.)*

Plot the coefficients to answer these questions.

(d) Compare these coefficients to box plots of hourly rentals. What do you learn from each plot that you don't learn from the other?

(e) Create a small set of dummies for logical groups of hours, such as commuting times, nighttime, etc. For both user types, fit a model for user types that uses only your created set of dummy variables.

(f) Use the partial-F test (all still within the training data) to compare, for each user type, the two different ways of measuring the hour of the day. Be precise about the null hypothesis, including a description of how one model is nested in the other. Discuss your findings.

(g) Use the test data to compare out-of-sample MSE performance of all four models you've considered (both user types and both ways of measuring the hour of the day). Discuss your findings and how these findings compare to the partial-F findings.

(h) Add to each of the four models above the main effects for season + holiday + workingday + weather + temp + atemp + humidity + windspeed. Repeat the partial-F and out-of-sample MSE comparisons. How do your findings change relative to when you compared simpler models in part **(f)**? What do you learn from this? Does the comparison of partial-F vs. MSE change with these models?

(i) Rerun the models with different testing/training splits (different seeds, different sizes). Are your conclusions stable?

[1] This data comes from a data-mining competition: the data for this problem is in fact the training data set, and the rest of the days were the test data, withheld by the company sponsoring the competition.