Midterm: Simple Linear Regression

MATH 530-630

- Logistics
- Overview
- HLO Lahman
- Data wrangling in `dplyr`
- Univariate EDA (+ more wrangling)
- Bivariate EDA (+ even more wrangling)
- Regression model
- Wrap-up

Logistics

Due anytime on Monday November 9. There are two sections to complete: the simple linear regression lab portion (below), and exercises (see link for PDF). You'll submit your work for the lab portion as a PDF file knit from your `.Rmd` file. Please set `echo = TRUE` and `include = TRUE` as a global option for all of your R code chunks.

Don't forget about the exercises as well: (http://cslu.ohsu.edu/~presmane/courses/MATH630/midterm%20exercises.pdf)http://cslu.ohsu.edu/~presmane/courses/MATH630/midterm%20exercises.pdf (http://cslu.ohsu.edu/~presmane/courses/MATH630/midterm%20exercises.pdf)

Overview

You will need to install/load the following packages:

```
library(plyr) #this must be loaded before dplyr
library(dplyr)
library(ggplot2)
library(broom)
```

In 2003, Michael Lewis published "Moneyball: The Art of Winning an Unfair Game." It featured the now famous true story about the Oakland Athletics baseball team and its unique general manager Billy Beane, who helped the A's become a successful basebal franchise despite carrying one of the smallest payrolls in baseball. That is, the A's were consistently able to exceed expectations about their ability to win games given their relatively low budget. We'll use a large dataset in R to explore the association between wins and payrolls. Beane became general manager of the A's in 1997, so we'll focus on data since then. To get this data, install and load the `Lahman` package.

```
install.packages("Lahman")
library(Lahman)
?Lahman
```

The data stored in the `Lahman` package is structured slightly differently than others we have worked with before from CRAN. The main form of this database is a relational database in Microsoft Access format. The data we need to access is stored in two separate tables.

1. `Salaries` : this contains each player's salary for each season and an identifier for the team that paid them. You can calculate each team's yearly payroll using this table.
2. `Teams` : this table contains statistics for each team and season, including wins.

```
Teams <- Lahman::Teams
Salaries <- Lahman::Salaries
```

Create these into objects in your global environment **exactly** like the above- I am trying to save you pain here!!

HLO Lahman

1. Conduct a high-level overview (HLO) of the Teams and Salaries datasets (see here for example command menu using `babynames` dataset (http://cslu.ohsu.edu/~presmane/courses/MATH630/Math_530-630_Class_0.html)).

- Are they data.frames, matrices, vectors, lists?
- What is the unit of analysis in the dataset?
- How many variables/columns?
- How many rows/observations?

Find the variables for games won, team, year, and salary.

- Which variables are continuous?
- Which variables are discrete?
- Which variables are categorical? How many levels do they have?
- What about missing data for any variables?

Data wrangling in `dplyr`

2. Use `dplyr` to create a new dataset that includes yearly payroll for each team in the

`Salaries` dataframe. (**hint:** you should have a dataset with 828 observations of 3 variables at the end that looks like this)

```
Source: local data frame [6 x 3]
Groups: yearID

  yearID teamID  payroll
1   1985    ATL 14807000
2   1985    BAL 11560712
3   1985    BOS 10897560
4   1985    CAL 14427894
5   1985    CHA  9846178
6   1985    CHN 12702917
```

3. Add this `payroll` column to the `Teams` dataframe. (**1st hint:** you can do a `merge()` in base R or a `inner_join()` in dplyr by two variables using the `by = c("var1", "var2")` command; **2nd hint:** you should still have a dataset with 828 observations of 49 variables at the end of this)

4. We'll focus on the years 2000 - 2013. Use `dplyr` to `filter()` the dataset you created with the `Teams` data plus the `payroll` column for just those years. (**hint:** you should have a dataset with 420 observations of 49 variables at the end of this)

5. A major part of the Oakland A's strategy was to exploit underused statistics, such as a player's ability to get on base and slugging percentage, to obtain excellent players who may have been underestimated by typical statistics like home runs, RBIs (runs batted in), and batting average. Obtaining players who excelled in these underused statistics turned out to be much more affordable for the team. The `battingStats()` function in the `Lahman` function calculates for you all of these statistics based on the `Batting` dataframe. Use my code below to create a new dataframe called `bat_stats` that gives you the following variables (do not get side-tracked by trying to understand this function- just treat this code as a gift and move on; note that this function will give you a warning if you did not load `plyr` before `dplyr`):

    - on-base percentage `OBP`
    - slugging percentage `SlugPct`
    - on-base percentage + slugging `OPS`

```
bat_stats <- battingStats(data = Lahman::Batting,
             idvars = c("playerID", "yearID", "stint", "teamID", "lgID"),

             cbind = TRUE)
```

6. Write a `dplyr` expression to create a new dataframe that contains means for each of these three new variables for each team and year from 2000 - 2013 (rather than for each player). (**hint:** use `mean(variable, na.rm = TRUE)` to remove missing values, otherwise any 1 missing value will cause `mean()` to return `NA` ; **2nd hint:** order matters here- consider carefully when to `filter` , `group_by` , and `summarise` (you should use all 3, and in fact in that order); **3rd hint:** you should have a dataset with 420 observations of 5 variables at the end that looks like this).

```
Source: local data frame [6 x 5]
Groups: yearID

  yearID teamID   ob_perc slug_perc       ops
1   2000    ANA 0.3385926 0.3756667 0.7142593
2   2000    ARI 0.3016216 0.3618378 0.6634595
3   2000    ATL 0.2430000 0.3009167 0.5439167
4   2000    BAL 0.2474000 0.3088571 0.5562571
5   2000    BOS 0.2644571 0.3063429 0.5708000
6   2000    CHA 0.2937600 0.3396800 0.6334400
```

7. Adds these new batting statistic columns to your current dataframe (**hint:** you can do a `merge()` in base R or a `inner_join()` in dplyr by two variables using the `c("var1", "var2")` command; **2nd hint:** you should still have a dataset with 420 observations at the end of this).

Univariate EDA (+ more wrangling)

8. **Number of seasons:** We'll do some counting detective work together (at this point, my dataframe is called `teams_bat`): how many teams are there? Which teams have data for the least number of seasons? Which have the most seasons?

```
teams_bat %>%
  group_by(teamID) %>%
  tally() %>%
  print(n = 33)
```

It looks like most teams have all 14 seasons worth of data, which is great. But some have less. Some have much less. Based on this, let us all exclude 3 teams: ANA (5 seasons), MIA (2 seasons), and MON (5 seasons). All the remaining teams will then have data for > 8 seasons. You should now have a dataframe with 408 observations.

```
teams_bat <- teams_bat %>%
  filter(!(teamID %in% c("ANA", "MIA", "MON"))) # you should understand w
hat this does
```

9. **Number of games played per season:** So now we know that all teams have data for > 8 seasons. Another concern is that some teams played in more or less games per season than others. Let's figure out the range of games played in each season. Take the dataframe you created in #8 and do the following in this sequence using the pipe operator `%>%` : `group_by(yearID)` , `select(G)` , and use `summarise_each(funs(?, ?, ?, ?))` to examine the minimum, maximum, median, and mean number of games per season (respectively, you fill in the `?` s). Is there a lot of variability in number of games played per season across teams? What is the range of games played by teams per season?

10. **Center, spread, and shape:** Explore the following variables using your tools for numerical detective work in R.
    - Number of games won
    - Mean on-base percentage
    - Mean slugging percentage
    - Mean on-base percentage + slugging

Bivariate EDA (+ even more wrangling)

11. Use `ggplot2` to create a scatterplot showing mean payroll (x-axis) and mean number of wins (y-axis) across *all* time periods and teams (**1st hint:** add a regression line using `geom_smooth(method = "lm")` ; **2nd hint:** it may be little tough to read the x-axis because of the large scale of the payroll variable: try changing the scale by using `x = payroll/1000` when you set the aesthetics).

12. One variable we are not accounting for in this scatterplot is year. It is possible that payrolls increase from season to season. Check this out using the same `ggplot` code you just used above, but make this plot with year on the x-axis and payroll/1000 on the y-axis.
    - A scatterplot may not be the best way to look at this pattern, since year is a discrete variable. So also try making boxplots stratified by `yearID` . At this point, you may want to see what `str(teams_bat$yearID)` . You can tell `ggplot` that we'd like to instead treat `yearID` as a factor variable by stating: `x = factor(yearID)` .

13. It seems like we should account for year in some way, since payrolls do seem to increase over time. One idea is to calculate each team's payroll per year *relative* to other teams in the same year/season, instead of using raw payroll values (or really `payroll/1000` values). One way to do this is to transform the payroll variable into a z-score, but do this separately for each year. This way, a team whose payroll is at the mean in the years 2000 and 2013 would still

have a payroll z-score of 0 both years even if the actual payroll values doubled (or tripled, or halved!) over the same time period. We'll do this step-by-step:

- Create new variables for the average payroll and the standard deviation of payrolls each year across teams and add them to your dataframe. (**hint:** you want the dataframe to have the same number of rows as it had before, so do not use `summarise` here; this is a job for `mutate`. **2nd hint:** if you do it right, the values for both of these columns should be the same for all teams within each year, as you can see below- use `select(variable names)` then `arrange(yearID) %>% head()` or `tail()` to check yourself)
- Add another variable to your dataset that is the z-score for each team for each year. My first and last six (arranged by `yearID`) are below to check your math.
- Make a scatterplot in `ggplot` with year on the x-axis and payroll z-scores on the y-axis and two geoms: `geom_point()` and `geom_smooth(method = "lm")`. What do you see? How is this plot different from the previous one with payroll/1000 on the y-axis (from #12)?
- Make a new scatterplot (minus `geom_smooth`) in `ggplot` with year on the x-axis and payroll z-scores on the y-axis. This time, add an additional aesthetic to `colour` the points in the scatterplot with a different color for each `teamID`, and an additional geom called `geom_line()`. What do you see? What is *not* surprising here? (**hint:** this is not a trick question, follow the lines left to right by color)
- **Save this dataset!!** I called mine `teams_bat` - you will need this later!!

```
teams_bat %>%
  select(teamID, yearID, payroll, mean_paypy, sd_paypy) %>%
  arrange(yearID) %>%
  head()
```

```
Source: local data frame [6 x 5]
Groups: yearID

  teamID yearID  payroll mean_paypy sd_paypy
1    ARI   2000 81027833   56488450 21731419
2    ATL   2000 84537836   56488450 21731419
3    BAL   2000 81447435   56488450 21731419
4    BOS   2000 77940333   56488450 21731419
5    CHA   2000 31133500   56488450 21731419
6    CHN   2000 60539333   56488450 21731419
```

```
teams_bat %>%
  select(teamID, yearID, payroll, mean_paypy, sd_paypy) %>%
  arrange(yearID) %>%
  tail()
```

```
Source: local data frame [6 x 5]
Groups: yearID

  teamID yearID   payroll mean_paypy sd_paypy
1    SFN   2013 140180334  103480129 47968483
2    SLN   2013  92260110  103480129 47968483
3    TBA   2013  52955272  103480129 47968483
4    TEX   2013 112522600  103480129 47968483
5    TOR   2013 126288100  103480129 47968483
6    WAS   2013 113703270  103480129 47968483
```

```
teams_bat %>%
  select(teamID, yearID, payroll, z_paypy) %>%
  arrange(yearID) %>%
  head()
```

```
Source: local data frame [6 x 4]
Groups: yearID

  teamID yearID  payroll    z_paypy
1    ARI   2000 81027833  1.1292122
2    ATL   2000 84537836  1.2907296
3    BAL   2000 81447435  1.1485207
4    BOS   2000 77940333  0.9871368
5    CHA   2000 31133500 -1.1667416
6    CHN   2000 60539333  0.1864067
```

```
teams_bat %>%
  select(teamID, yearID, payroll, z_paypy) %>%
  arrange(yearID) %>%
  tail()
```

```
Source: local data frame [6 x 4]
Groups: yearID

  teamID yearID   payroll     z_paypy
1    SFN   2013 140180334  0.7650900
2    SLN   2013  92260110 -0.2339040
3    TBA   2013  52955272 -1.0532928
4    TEX   2013 112522600  0.1885086
5    TOR   2013 126288100  0.4754783
6    WAS   2013 113703270  0.2131220
```

14. Our solution to the above (obvious) problem will be to use each team's average payroll z-score across seasons as the predictor variable. Similarly, if we turn to the variable we wish to predict- number of wins- we also want to calculate the mean number of wins across seasons. Use `dplyr` to create a new dataset (**save the current dataset as something before doing this step!** I called mine `teams_bat` - you will need this later!!), which will be the analytic dataset for our regression model, that includes two new variables: average payroll z-score and average number of wins. Both averages should be calculated for each team across all seasons. (**hint:** use `mean(variable, na.rm = TRUE)` to remove missing values, otherwise any 1 missing value will cause `mean()` to return `NA` ; recall that we left some teams in our dataset that had < 14 seasons' worth of data!; **2nd hint:** you should now have a dataset with only 30 observations- one for each team)

15. We are going to take a quick aside to reflect on the nature of z-scores. In the above code, you should have calculated the average of several z-scores to get the average payroll z-score for each team across all observed seasons. What will be the mean and standard deviation of this new variable across the 30 teams? That is, is your new average payroll z-score *also* a z-score? Are you surprised? Why or why not?

16. Now create a scatterplot to see the association between average payroll z-scores (x-axis) and average number of wins (y-axis). (**hint:** again, use the dataset with 30 observations- you should see 30 points in this scatterplot). Calculate the correlation between these two variables. Use both the plot and the correlation statistics to evaluate (in words) the form (does the relationship look linear?) and strength of the association between these two variables. Would you be comfortable using a linear model to predict the mean number of wins in a given season given their average relative payroll for that season?

Regression model

17. Build a simple linear regression model predicting mean wins from mean payroll z-scores across seasons (**hint:** the dataset you use should be the one with 30 observations!).

18. What are the total, model, and residual sums of squares for this simple linear regression? What percent of the variation in mean wins is "explained" by variation in mean payroll z-scores?

19. Write up a summary of your findings. Here is an example:

> The results indicate that years of education significantly predicted prestige (b_1 = 5.36), with 72.3% of the variance in prestige accounted for by education levels. Each year increase in education was associated with a 5.36 unit increase in prestige. The OLS regression equation for predicting prestige is of the form:
>
> $$prestige_i = -10.723 + 5.361 education_i + \epsilon_i$$

20. What is the average of all $\hat{y}_i$ values (in any simple linear regression model) equal to? (**hint:** it is a sample statistic)

21. What is the variance of the residuals in your regression model? The standard error? Compare the variance of the residuals to sample variance of mean wins overall, and to your model R^2. How are these three statistics related (in any simple linear regression model)?

22. Obviously, the book and movie about the Oakland A's suggests that this team may be an outlier in terms of the predicting wins from payroll. Look specifically at this team: what is the observed number of mean wins? What is the predicted? What is the residual? How many standard deviations above/below the residual mean is the Oakland A's residual value? Are there any other teams with a residual value as extreme or more extreme than the Oakland A's?

23. Create a bootstrap distribution for the correlation and the regression coefficients. Copy and paste the following code into your file, and annotate each line with a `#` to (briefly) explain what each line of code is doing. Figure out how many bootstrap samples had a higher correlation than the one you observed as your original sample correlation, and how many bootstrap samples had a higher slope coefficient than the one you observed.

```
N <- 10^4
cor.boot <- numeric(N)
int.boot <- numeric(N)
slope.boot <- numeric(N)
n <- # number of observations here
for (i in 1:N){
    index <- sample(n, replace = TRUE)
    team.boot <- dataframe[index, ] # resampled data
    cor.boot[i] <- cor(team.boot$x, team.boot$y) # what is x and y?
    # recalculate linear model estimates
    team.boot.lm <- lm(y ~ x, data = team.boot) # what is x and y?
    int.boot <- coef(team.boot.lm)[1] # new intercept
    slope.boot <- coef(team.boot.lm)[2] # new slope
  }

mean(cor.boot)
sd(cor.boot)
quantile(cor.boot, c(.025, .975))

hist(cor.boot)
observed <- cor(originaldataframe$x, originaldataframe$y) # what is x and
 y?
abline(b = observed, col = "red") # add line at original sample correlati
on
# do the same as above for slope.boot (don't worry about int.boot)
```

24. One possible hiccup in our model is that one could argue that wins may have been relatively "cheaper" earlier (when the A's strategy was less well known). Go back to question 13 (I know, sorry: it had 408 observations of 55 variables), and work with **that** dataset. Mine was balled `teams_bat`. Use a new function, `cut()`, to create a categorical variable that splits our `yearID` variable into two time intervals: 2000 - 2006 and 2007 - 2013. Then look at your work for question 14 and update to re-calculate average wins and average payroll z-scores separately for each team and time interval (**hint:** that means two variables in a `dplyr::group_by()` statement).

```
newdataframe <- dataframe %>% # rename these dataframes as appropriate
  mutate(new_interval_var = cut(old_cont_var, breaks = c(2000, 2006, 2013
), right = TRUE, include.lowest = TRUE)) # rename variables as appropriat
e
table(newdataframe$new_interval_var, newdataframe$old_var) # trust but ve
rify
```

25. Using `ggplot2`, create *one* plot, with side-by-side scatterplots for each time interval, showing mean payroll (x-axis) and mean number of wins (y-axis) across *all* teams (**1st hint:** add a regression line using `geom_smooth(method = "lm")`; **2nd hint:** it may be little tough to read the x-axis because of the large scale of the payroll variable: try changing the scale by using `x = payroll/1000` when you set the aesthetics; **3rd (BIG) hint:** this is sounding like a job for `facet_wrap()`; **4th hint:** adding a `geom_text()` layer is a neat trick here, try: `geom_text(aes(label = teamID))`). Comment on differences you see between these two plots, and compare to your previous scatterplot across all seasons.
26. Now, run two linear regression analyses (as shown in class), one for each time interval, using `dplyr::group_by() %>% do()` and `broom::tidy()/glance()/augment()`. Compare the coefficient estimates to each other, and to your original model. Take the Oakland A's team as a specific case: which of your three model/time interval regression models (model 1: across all seasons; model 2: 2000 - 2006; model 3: 2007 - 2013) was better at predicting mean wins for them specifically? Which model overall accounted for the most variability in mean wins overall across all teams? How is the R^2 estimate related to the plain old correlation between average wins and average payroll z-scores for each time interval (and in general in any simple linear regression model)?

Wrap-up

You made it! This exercise had a lot of data wrangling in it! In your research careers, whether they are academic or not, you will typically spend much more time prepping data prior to doing an actual analysis, and doing model diagnostics after you do the actual analysis. The actual analysis bit usually is fairly straightforward in terms of using R. More on "data janitor" work:

- NY Times: For big data scientists, hurdle to insights is janitor work (http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html?_r=0)
- Data Wrangling: Making data useful again (http://www.sciencedirect.com/science/article/pii/S2405896315001986)
- Research directions in data wrangling: Visualizations and transformations for usable and credible data (http://idl.cs.washington.edu/files/2011-DataWrangling-IVJ.pdf)

Homework 3 will revisit this dataset and analysis to do some model diagnostics, so stay tuned! You may have also noticed that we didn't use those "underused" statistics, but we will in a multiple regression, also in a later homework.