STAT 471: Homework 1

Nico Melton

Due: September 19, 2021 at 11:59pm

Contents

Instructions

Setup

Pull the latest version of this assignment from Github and set your working directory to `stat-471-fall-2021/homework/homework-1`. Consult the getting started guide if you need to brush up on `R` or `Git`.

Collaboration

The collaboration policy is as stated on the Syllabus:

> "Students are permitted to work together on homework assignments, but solutions must be written up and submitted individually. Students must disclose any sources of assistance they received; furthermore, they are prohibited from verbatim copying from any source and from consulting solutions to problems that may be available online and/or from past iterations of the course."

In accordance with this policy,

Please list anyone you discussed this homework with: Kennedy Manley

Please list what external references you consulted (e.g. articles, books, or websites):

Writeup

Use this document as a starting point for your writeup, adding your solutions after "**Solution**". Add your R code using code chunks and add your text answers using **bold text**. Consult the preparing reports guide for guidance on compilation, creation of figures and tables, and presentation quality.

Programming

The `tidyverse` paradigm for data wrangling, manipulation, and visualization is strongly encouraged, but points will not be deducted for using base `R`.

Grading

The point value for each problem sub-part is indicated. Additionally, the presentation quality of the solution for each problem (as exemplified by the guidelines in Section 3 of the preparing reports guide will be evaluated on a per-problem basis (e.g. in this homework, there are three problems). There are 100 points possible on this homework, 85 of which are for correctness and 15 of which are for presentation.

Submission

Compile your writeup to PDF and submit to Gradescope.

Case study: Major League Baseball

What is the relationship between payroll and wins among Major League Baseball (MLB) teams? In this homework, we'll find out by wrangling, exploring, and modeling the dataset in `data/MLPayData_Total.csv`, which contains the winning records and the payroll data of all 30 MLB teams from 1998 to 2014.

The dataset has the following variables:

- `payroll`: total team payroll (in billions of dollars) over the 17-year period
- `avgwin`: the aggregated win percentage over the 17-year period
- `Team.name.2014`: the name of the team
- `p1998`, ..., `p2014`: payroll for each year (in millions of dollars)
- `X1998`, ..., `X2014`: number of wins for each year
- `X1998.pct`, ..., `X2014.pct`: win percentage for each year

We'll need to use the following `R` packages:

```
library(tidyverse)  # tidyverse
library(ggrepel)    # for scatter plot point labels
library(kableExtra) # for printing tables
library(cowplot)    # for side by side plots
library(gridExtra)  # for plotting side by side
library(kableExtra) # for printing better tables
library(stargazer)  # for displaying regression results
```

1 Wrangle (30 points for correctness; 5 points for presentation)

1.1 Import (5 points)

- Import the data into a `tibble` called `mlb_raw` and print it.
- How many rows and columns does the data have?
- Does this match up with the data description given above?

[Hint: If your working directory is `stat-471-fall-2021/homework/homework-1`, then you can use a *relative path* to access the data at `../../data/MLPayData_Total.csv`.]

Solution.

```
# read data using 'read_csv'
mlb_raw <- read_csv("../../data/MLPayData_Total.csv")
mlb_raw # print
```

```
## # A tibble: 30 x 54
##    payroll avgwin Team.name.2014 p1998 p1999 p2000 p2001 p2002 p2003 p2004 p2005
##      <dbl>  <dbl> <chr>          <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1    1.12  0.490 Arizona Diamo~  31.6  70.5  81.0  81.2 103.   80.6  70.2  63.0
##  2    1.38  0.553 Atlanta Braves  61.7  74.9  84.5  91.9  93.5 106.   88.5  85.1
##  3    1.16  0.454 Baltimore Ori~  71.9  72.2  81.4  72.4  60.5  73.9  51.2  74.6
##  4    1.97  0.549 Boston Red Sox  59.5  71.7  77.9 110.  108.   99.9 125.  121.
##  5    1.46  0.474 Chicago Cubs    49.8  42.1  60.5  64.0  75.7  79.9  91.1  87.2
##  6    1.32  0.511 Chicago White~  35.2  24.5  31.1  62.4  57.1  51.0  65.2  75.2
```

```
## 7   1.02   0.486 Cincinnati Re~  20.7  73.3  46.9  45.2  45.1  59.4  43.1  59.7
## 8   0.999  0.496 Cleveland Ind~  59.5  54.4  75.9  92.0  78.9  48.6  34.6  41.8
## 9   1.03   0.463 Colorado Rock~  47.7  55.4  61.1  71.1  56.9  67.2  64.6  47.8
## 10  1.43   0.482 Detroit Tigers  19.2  35.0  58.3  49.8  55.0  49.2  46.4  69.0
## # ... with 20 more rows, and 43 more variables: p2006 <dbl>, p2007 <dbl>,
## #   p2008 <dbl>, p2009 <dbl>, p2010 <dbl>, p2011 <dbl>, p2012 <dbl>,
## #   p2013 <dbl>, p2014 <dbl>, X2014 <dbl>, X2013 <dbl>, X2012 <dbl>,
## #   X2011 <dbl>, X2010 <dbl>, X2009 <dbl>, X2008 <dbl>, X2007 <dbl>,
## #   X2006 <dbl>, X2005 <dbl>, X2004 <dbl>, X2003 <dbl>, X2002 <dbl>,
## #   X2001 <dbl>, X2000 <dbl>, X1999 <dbl>, X1998 <dbl>, X2014.pct <dbl>,
## #   X2013.pct <dbl>, X2012.pct <dbl>, X2011.pct <dbl>, X2010.pct <dbl>, ...
```

The data has 30 rows and 54 columns where each row is a team and each column is a variable. This data matches the data description above because it contains 30 MLB teams and has all variables listed above including `payroll`, `avgwin`, `Team.name.2014`, `p1998-p2014`, `X1998-X2014`, and `X1998.pct-X2014.pct`.

1.2 Tidy (15 points)

The raw data are in a messy format: Some of the column names are hard to interpret, we have data from different years in the same row, and both year-by-year and aggregate data are present.

- Tidy the data into two separate `tibbles`: one called `mlb_aggregate` containing the aggregate data and another called `mlb_yearly` containing the year-by-year data. `mlb_aggregate` should contain columns named `team`, `payroll_aggregate`, `pct_wins_aggregate` and `mlb_yearly` should contain columns named `team`, `year`, `payroll`, `pct_wins`, `num_wins`. Comment your code to explain each step.
- Print these two `tibbles`. How many rows do `mlb_aggregate` and `mlb_yearly` contain, and why?

[Hint: For `mlb_yearly`, the main challenge is to extract the information from the column names. To do so, you can `pivot_longer` all these column names into one column called `column_name`, `separate` this column into three called `prefix`, `year`, `suffix`, `mutate` `prefix` and `suffix` into a a new column called `tidy_col_name` that takes values `payroll`, `num_wins`, or `pct_wins`, and then `pivot_wider` to make the entries of `tidy_col_name` into column names.]

Solution.

```
# use 'select' to select and rename columns `team`, `payroll_aggregate`,
#`pct_wins_aggregate` from 'mlb_raw' and assign returned tibble to 'mlb_aggregate'
mlb_aggregate = mlb_raw %>%
  select(team = Team.name.2014,
         payroll_aggregate = payroll,
         pct_wins_aggregate = avgwin)
mlb_aggregate # print
```

```
## # A tibble: 30 x 3
##    team                  payroll_aggregate pct_wins_aggregate
##    <chr>                             <dbl>              <dbl>
##  1 Arizona Diamondbacks               1.12              0.490
##  2 Atlanta Braves                     1.38              0.553
##  3 Baltimore Orioles                  1.16              0.454
##  4 Boston Red Sox                     1.97              0.549
```

```
##  5 Chicago Cubs                    1.46              0.474
##  6 Chicago White Sox               1.32              0.511
##  7 Cincinnati Reds                 1.02              0.486
##  8 Cleveland Indians               0.999             0.496
##  9 Colorado Rockies                1.03              0.463
## 10 Detroit Tigers                  1.43              0.482
## # ... with 20 more rows
```

```
# use 'select' to select and rename team and all yearly payroll, percent wins,
#and number of wins variables from 'mlb_raw' and assign to 'mlb_yearly'
mlb_yearly = mlb_raw %>%
  select(team = Team.name.2014,
         !c("payroll", "avgwin")) # all vars except `payroll` and `avgwin`
# rename all win percentage variable to have prefix pctX
#and no suffix (to prep for 'pivot_longer')
mlb_yearly = mlb_yearly %>%
  rename_with(~ str_sub(paste0("pct", .x), start = 1, end = 8),
              ends_with(".pct"))
# use 'pivot_longer' to pivot the yearly data in each row to long
#format for `payroll`, `pct_wins`, and `num_wins` separately
payroll_yearly = mlb_yearly %>%
  select(team, matches("p\\d")) %>%
  pivot_longer(!team, names_to = "year", names_prefix = "p", values_to = "payroll")
pct_wins_yearly = mlb_yearly %>%
  select(team, starts_with("pctX")) %>%
  pivot_longer(!team, names_to = "year", names_prefix = "pctX", values_to = "pct_wins")
num_wins_yearly = mlb_yearly %>%
  select(team, starts_with("X")) %>%
  pivot_longer(!team, names_to = "year", names_prefix = "X", values_to = "num_wins")
# use 'full_join' to combine each long format data set, override 'mlb_yearly'
mlb_yearly = payroll_yearly %>%
  full_join(pct_wins_yearly, by = c("team", "year")) %>%
  full_join(num_wins_yearly, by = c("team", "year"))
mlb_yearly # print
```

```
## # A tibble: 510 x 5
##    team                 year  payroll pct_wins num_wins
##    <chr>                <chr>   <dbl>    <dbl>    <dbl>
##  1 Arizona Diamondbacks 1998     31.6    0.401       65
##  2 Arizona Diamondbacks 1999     70.5    0.617      100
##  3 Arizona Diamondbacks 2000     81.0    0.525       85
##  4 Arizona Diamondbacks 2001     81.2    0.568       92
##  5 Arizona Diamondbacks 2002    103.     0.605       98
##  6 Arizona Diamondbacks 2003     80.6    0.519       84
##  7 Arizona Diamondbacks 2004     70.2    0.315       51
##  8 Arizona Diamondbacks 2005     63.0    0.475       77
##  9 Arizona Diamondbacks 2006     59.7    0.469       76
## 10 Arizona Diamondbacks 2007     52.1    0.556       90
## # ... with 500 more rows
```

`mlb_aggregate` has 30 (one for each team) and 3 columns for `team`, `payroll_aggregate`, and `pct_wins_aggregate`. `mlb_yearly` has 510 rows and 5 columns. There are 510 rows because each of the 30 teams has 17 rows of data (one for each year between 1998 and 2014). The 5

columns include the 3 variables of interest (`payroll`, `pct_wins`, and `num_wins`) and `year` and `team` variables.

1.3 Quality control (10 points)

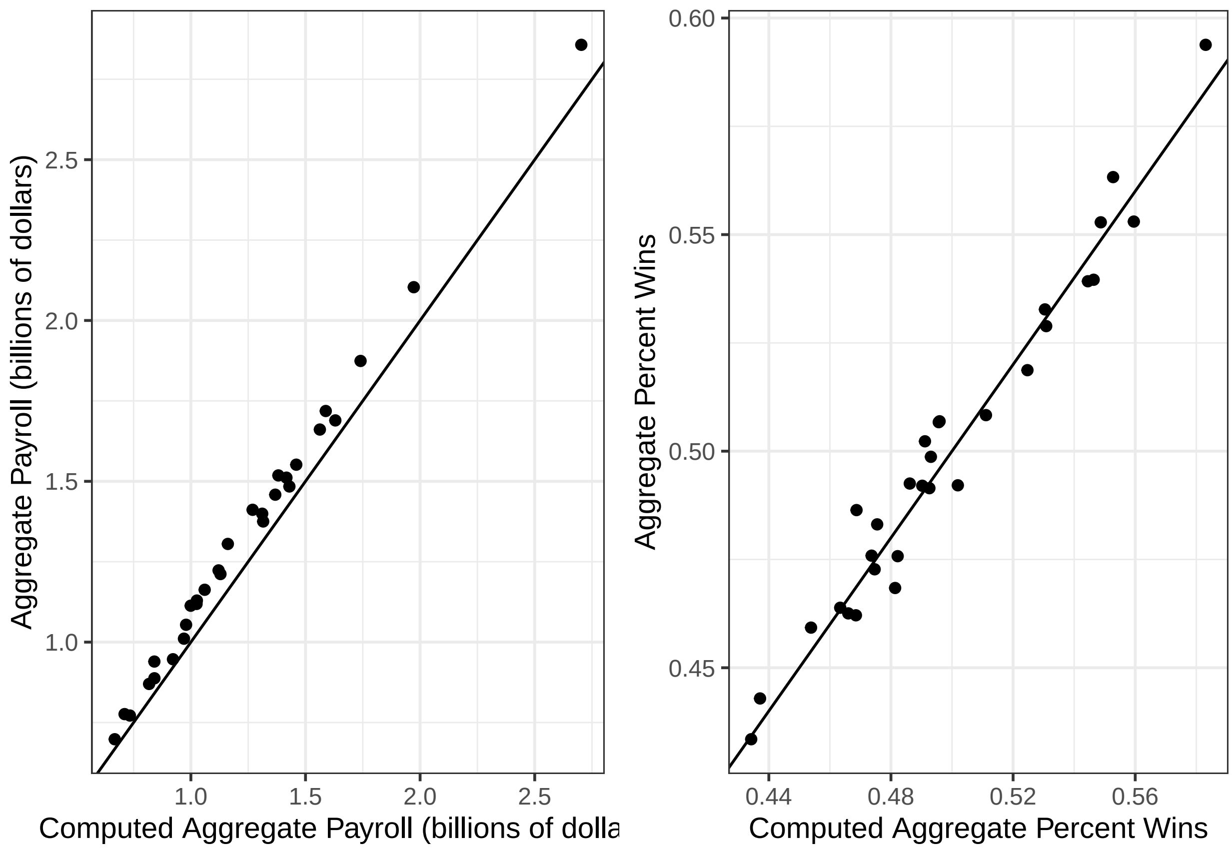
It's always a good idea to check whether a dataset is internally consistent. In this case, we are given both aggregated and yearly data, so we can check whether these match. To this end, carry out the following steps:

- Create a new `tibble` called `mlb_aggregate_computed` based on aggregating the data in `mlb_yearly`, containing columns named `team`, `payroll_aggregate_computed`, and `pct_wins_aggregate_computed`.
- Ideally, `mlb_aggregate_computed` would match `mlb_aggregate`. To check whether this is the case, join these two `tibbles` into `mlb_aggregate_joined` (which should have five columns: `team`, `payroll_aggregate`, `pct_wins_aggregate`, `payroll_aggregate_computed`, and `pct_wins_aggregate_computed`.)
- Create scatter plots of `payroll_aggregate_computed` versus `payroll_aggregate` and `pct_wins_aggregate_computed` versus `pct_wins_aggregate`, including a 45° line in each. Display these scatter plots side by side, and comment on the relationship between the computed and provided aggregate statistics.

Solution.

```r
# create new tibble 'mlb_aggregate_computed'
mlb_aggregate_computed = mlb_yearly %>%
  group_by(team) %>%
  summarise(payroll_aggregate_computed = sum(payroll) / 1000,
            pct_wins_aggregate_computed = mean(pct_wins))
# join 'mlb_aggregate' and 'mlb_aggregate_computed'
mlb_aggregate_joined = full_join(mlb_aggregate, mlb_aggregate_computed, by = "team")

# scatter plot of 'payroll_aggregate_computed' vs 'payroll_aggregate'
plot_payroll = mlb_aggregate_joined %>%
  ggplot(aes(x = payroll_aggregate, y = payroll_aggregate_computed)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  labs(x = "Computed Aggregate Payroll (billions of dollars)",
       y = "Aggregate Payroll (billions of dollars)") +
  theme_bw()
# scatter plot of 'pct_wins_aggregate' vs 'pct_wins_aggregate_computed'
plot_pct_wins = mlb_aggregate_joined %>%
  ggplot(aes(x = pct_wins_aggregate, y = pct_wins_aggregate_computed)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  labs(x = "Computed Aggregate Percent Wins", y = "Aggregate Percent Wins") +
  theme_bw()
# plot aggregate payroll and percent wins side by side
grid.arrange(plot_payroll, plot_pct_wins, ncol=2)
```



The computed aggregate payroll for all teams is slightly less than the provided aggregate payroll. The computed aggregate percent wins are similar to the provided aggregate percent wins but many teams have either slightly higher or slightly lower computed values - there is variation from the provided values.

2 Explore (40 points for correctness; 7 points for presentation)

Now that the data are in tidy format, we can explore them by producing visualizations and summary statistics.

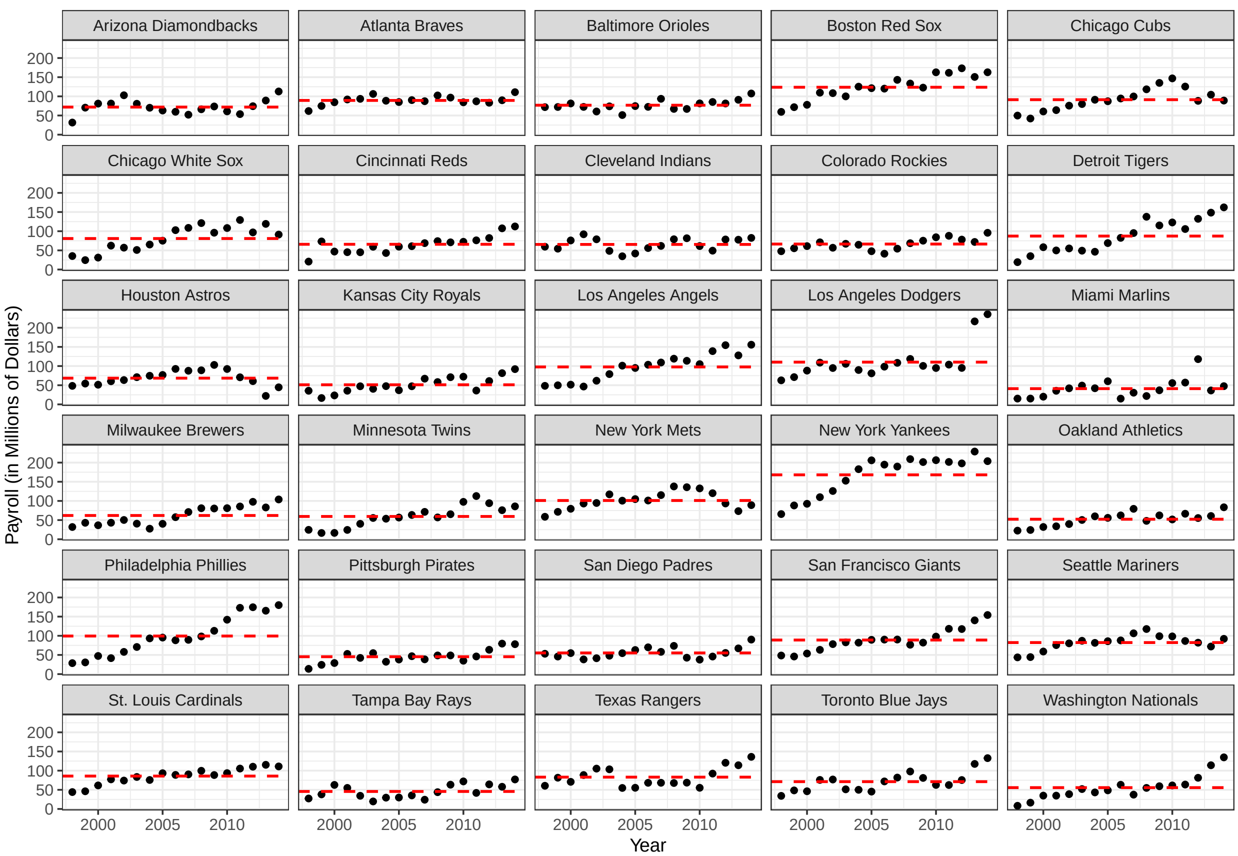
2.1 Payroll across years (15 points)

- Plot `payroll` as a function of `year` for each of the 30 teams, faceting the plot by `team` and adding a red dashed horizontal line for the mean payroll across years of each team.
- Using `dplyr`, identify the three teams with the greatest `payroll_aggregate_computed`, and print a table of these teams and their `payroll_aggregate_computed`.
- Using `dplyr`, identify the three teams with the greatest percentage increase in payroll from 1998 to 2014 (call it `pct_increase`), and print a table of these teams along with `pct_increase` as well as their payroll figures from 1998 and 2014.
- How are the metrics `payroll_aggregate_computed` and `pct_increase` reflected in the plot above, and how can we see that the two sets of teams identified above are the top three in terms of these metrics?

[Hint: To compute payroll increase, it's useful to `pivot_wider` the data back to a format where different years are in different columns. Use `names_prefix = "payroll_` inside `pivot_wider` to deal with the fact column names cannot be numbers. To add different horizontal lines to different facets, see this webpage.]

Solution.

```
# create dataset with team name and mean payroll for plotting mean line in facets
mean_payroll = mlb_yearly %>%
  group_by(team) %>%
  summarise(mean_payroll = mean(payroll)) %>%
  ungroup()
# plot payroll as a function of year, facet wrap by team
mlb_yearly %>%
  ggplot(aes(x = as.integer(year), y = payroll)) +
  geom_point(size = 0.7) +
  facet_wrap(~ team, nrow = 6) +
  geom_hline(aes(yintercept = mean_payroll),
             color = "red",
             linetype = "dashed",
             data = mean_payroll) +
  theme_bw(base_size = 7) +
  labs(x = "Year", y = "Payroll (in Millions of Dollars)")
```




```
# get teams with the highest computed aggregate payroll
mlb_aggregate_computed %>%
  arrange(desc(payroll_aggregate_computed)) %>%
```

```
  select(Team = team,
         "Aggregate Computed Payroll (in Billions of Dollars)" =
           payroll_aggregate_computed) %>%
  slice_head(n = 3) %>%
  kbl()
```

Team	Aggregate Computed Payroll (in Billions of Dollars)
New York Yankees	2.86
Boston Red Sox	2.10
Los Angeles Dodgers	1.87

```
# get the teams with the greatest percentage increase in payroll from 1998 to 2014
mlb_yearly %>%
  select(team, year, payroll) %>%
  pivot_wider(names_from = "year",
              values_from = "payroll",
              names_prefix = "payroll_") %>%
  mutate(pct_increase = (payroll_2014 - payroll_1998) / payroll_1998 * 100) %>%
  arrange(desc(pct_increase)) %>%
  select(Team = team,
         "Percentage Increase (1998-2014)" = pct_increase,
         starts_with("payroll")) %>%
  slice_head(n = 3) %>%
  kbl()
```

Team	Percentage Increase (1998-2014)	payroll_1998	payroll_1999	payroll_2000	payroll_2001
Washington Nationals	1520	8.32	16.4	34.8	34.8
Detroit Tigers	743	19.24	35.0	58.3	49.8
Philadelphia Phillies	529	28.62	30.6	47.3	41.7

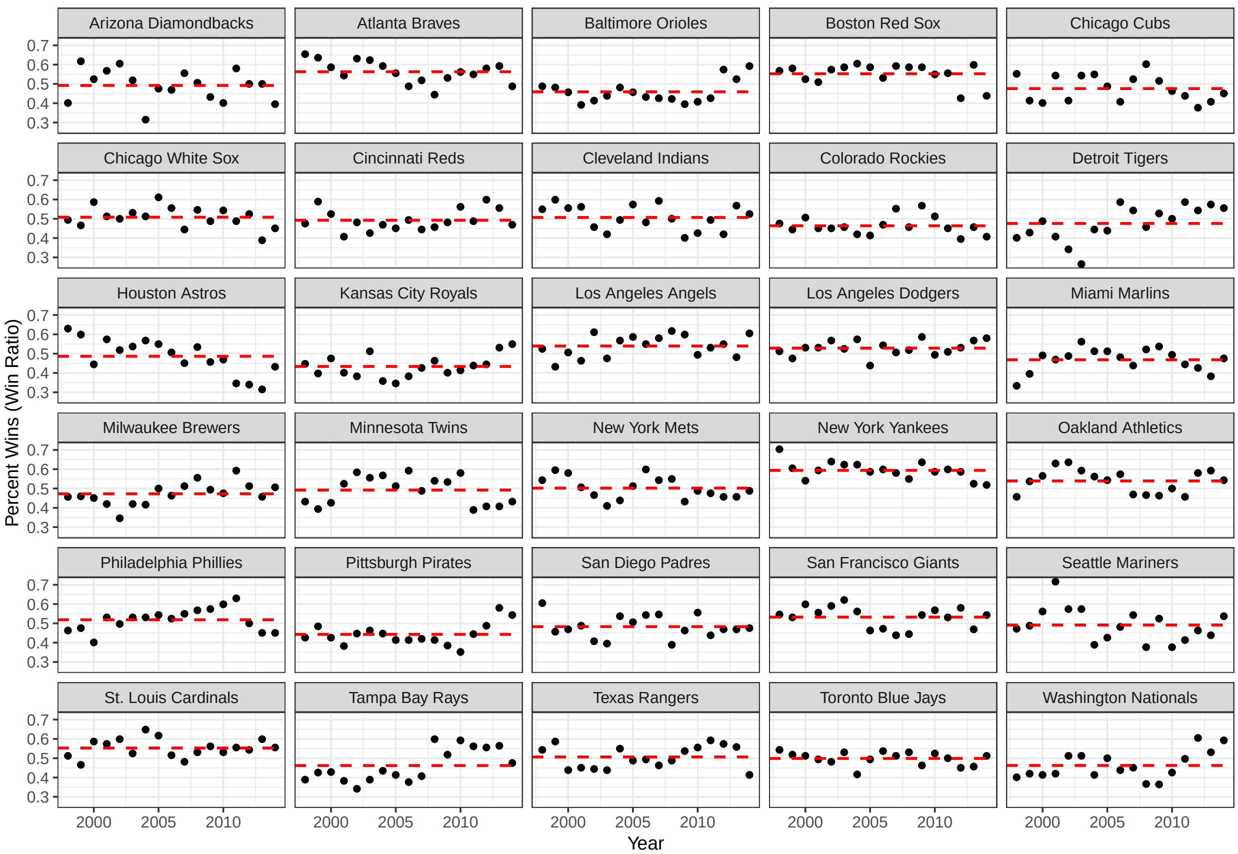
Payroll aggregate are reflected implicitly in the plot above through the mean payroll line for each team. Because mean payroll is just aggregate payroll divided by the number of years and all teams have the same years, we can see which teams have the highest `payroll_aggregate_computed` by looking at the mean payroll lines. We can see that the top 3 teams with the highest `payroll_aggregate_computed` have the highest mean payroll lines (red dotted line in the plot). We can see the `pct_increase` measure reflected in the plot by looking at the vertical distance between the first and last plotted payroll point for each team. Teams with the largest distance had the largest percentage increase payroll over the 17 years period.

2.2 Win percentage across years (10 points)

- Plot `pct_wins` as a function of `year` for each of the 30 teams, faceting the plot by `team` and adding a red dashed horizontal line for the average `pct_wins` across years of each team.
- Using `dplyr`, identify the three teams with the greatest `pct_wins_aggregate` and print a table of these teams along with `pct_wins_aggregate`.
- Using `dplyr`, identify the three teams with the most erratic `pct_wins` across years (as measured by the standard deviation, call it `pct_wins_sd`) and print a table of these teams along with `pct_wins_sd`.
- How are the metrics `pct_wins_aggregate` and `pct_wins_sd` reflected in the plot above, and how can we see that the two sets of teams identified above are the top three in terms of these metrics?

Solution.

```r
# create dataset with 'team' and 'mean_pct_wins' for plotting mean line in facets
mean_pct_wins = mlb_yearly %>%
  group_by(team) %>%
  summarise(mean_pct_wins = mean(pct_wins)) %>%
  ungroup()
# plot 'pct_wins' as a function of 'year', facet wrap by team
mlb_yearly %>%
  ggplot(aes(x = as.integer(year), y = pct_wins)) +
  geom_point(size = 0.7) +
  facet_wrap(~ team, nrow = 6) +
  geom_hline(aes(yintercept = mean_pct_wins),
             color = "red",
             linetype = "dashed",
             data = mean_pct_wins) +
  theme_bw(base_size = 7) +
  labs(x = "Year", y = "Percent Wins (Win Ratio)")
```




```r
# get the teams with the highest aggregated percent wins
mlb_aggregate %>%
  arrange(desc(pct_wins_aggregate)) %>%
  select(Team = team,
         "Aggregate Percent Wins (Win Ratio)" = pct_wins_aggregate) %>%
  slice_head(n = 3) %>%
  kbl()
```

Team	Aggregate Percent Wins (Win Ratio)
New York Yankees	0.583
St. Louis Cardinals	0.560
Atlanta Braves	0.553

```r
# get the teams with the most erratic 'pct_wins' across all years
mlb_yearly %>%
  select(team, year, pct_wins) %>%
  group_by(team) %>%
  mutate(pct_wins_sd  = sd(pct_wins)) %>%
  select(Team = team,
         "Standard Deviation of Percent Wins" = pct_wins_sd) %>%
  ungroup() %>%
  distinct(Team, .keep_all = TRUE) %>%
  arrange(desc("Standard Deviation of Percent Wins")) %>%
  slice_head(n = 3) %>%
  kbl()
```

Team	Standard Deviation of Percent Wins
Arizona Diamondbacks	0.083
Atlanta Braves	0.058
Baltimore Orioles	0.059

Percent wins aggregate is reflected implicitly in the plot above through the mean percent wins line for each team. Because mean percent wins is just aggregate percent wins divided by the number of years and all teams have the same years, we can see which teams have the highest `pct_wins_aggregate` by looking at the mean percent wins lines. Percent win standard deviation can be seen in the plots as the spread/distance of the points around the mean line for each team. Teams with the highest standard deviation have their points, on average, further from the mean percentage win line.

2.3 Win percentage versus payroll (10 points)

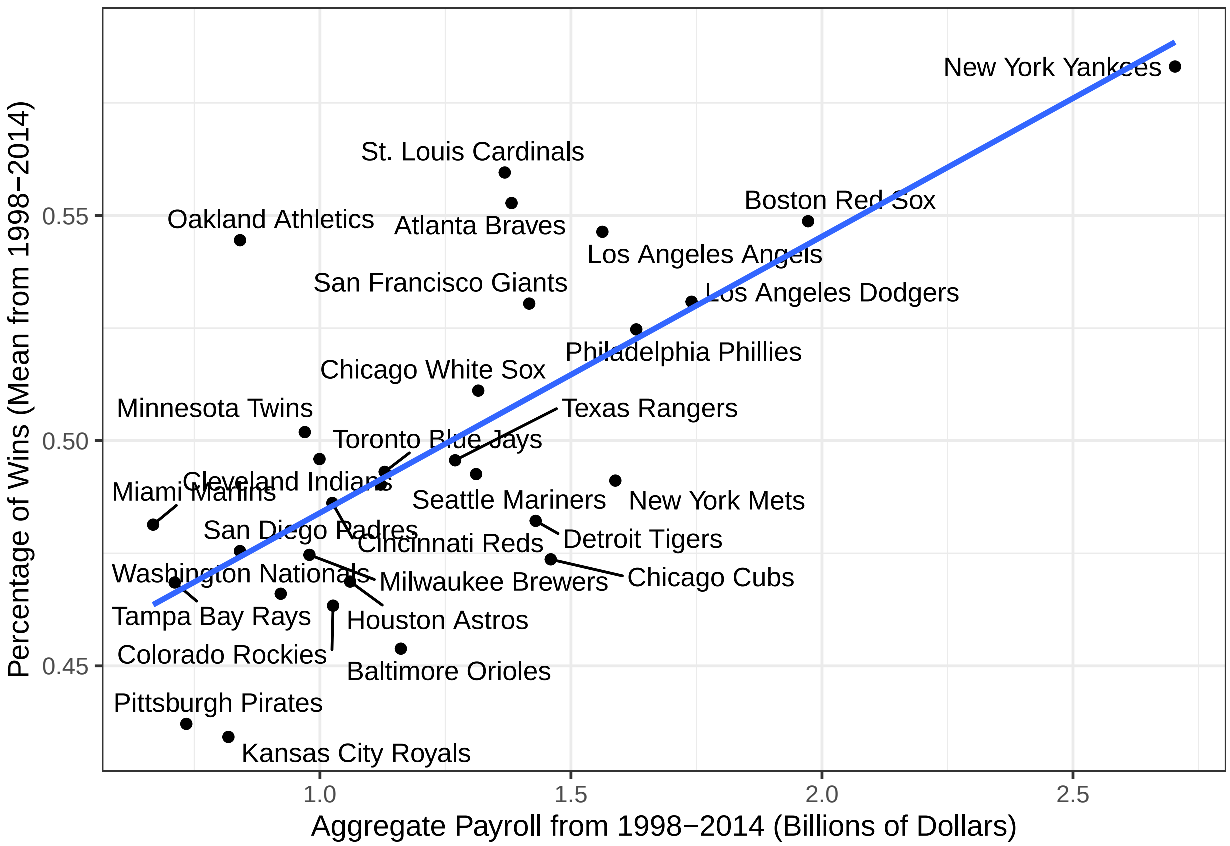
The analysis goal is to study the relationship between win percentage and payroll.

- Create a scatter plot of `pct_wins` versus `payroll` based on the aggregated data, labeling each point with the team name using `geom_text_repel` from the `ggrepel` package and adding the least squares line.
- Is the relationship between `payroll` and `pct_wins` positive or negative? Is this what you would expect, and why?

Solution.

```r
# plot `pct_wins` by `payroll`
mlb_aggregate %>%
  ggplot(aes(x = payroll_aggregate, y = pct_wins_aggregate)) +
  geom_point() +
  geom_text_repel(aes(label = team), size = 3.5, box.padding = unit(0.25, "lines")) +
  geom_smooth(method = "lm", se = FALSE) +
  theme_bw() +
  labs(x = "Aggregate Payroll from 1998-2014 (Billions of Dollars)",
       y = "Percentage of Wins (Mean from 1998-2014)")
```

```
## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



The relationship between payroll and percent wins is positive. This is what I would expect because teams with more money can presumably pay better players that in turn contribute to more wins.

2.4 Team efficiency (5 points)

Define a team's *efficiency* as the ratio of the aggregate win percentage to the aggregate payroll—more efficient teams are those that win more with less money.

- Using `dplyr`, identify the three teams with the greatest efficiency, and print a table of these teams along with their efficiency, as well as their `pct_wins_aggregate` and `payroll_aggregate`.
- In what sense do these three teams appear efficient in the previous plot?

Side note: The movie "Moneyball" portrays "Oakland A's general manager Billy Beane's successful attempt to assemble a baseball team on a lean budget by employing computer-generated analysis to acquire new players."

Solution.

```r
# find the most efficient teams and display in a table
mlb_aggregate %>%
  mutate(efficiency = pct_wins_aggregate / payroll_aggregate) %>%
  select(team, efficiency, pct_wins_aggregate, payroll_aggregate) %>%
  arrange(desc(efficiency)) %>%
  slice_head(n = 3) %>%
  kbl()
```

team	efficiency	pct_wins_aggregate	payroll_aggregate
Miami Marlins	0.721	0.481	0.668
Tampa Bay Rays	0.659	0.469	0.711
Oakland Athletics	0.648	0.545	0.841

All three of these teams lie above the fitted line for payroll vs percentage wins. This means these teams yield a higher than expected percentage wins based on their payroll.

3 Model (15 points for correctness; 3 points for presentation)

Finally, we build a predictive model for `pct_wins_aggregate` in terms of `payroll_aggregate` using the aggregate data `mlb_aggregate`.

3.1 Running a linear regression (5 points)

- Run a linear regression of `pct_wins_aggregate` on `payroll_aggregate` and print the regression summary.
- What is the coefficient of `payroll_aggregate`, and what is its interpretation?
- What fraction of the variation in `pct_wins_aggregate` is explained by `payroll_aggregate`?

Solution.

```r
# create model for `pct_wins_aggregate` in terms of `payroll_aggregate`
#and display results
model1 = lm(pct_wins_aggregate ~ payroll_aggregate, data = mlb_aggregate)
stargazer(model1, type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                         pct_wins_aggregate
## -----------------------------------------------
## payroll_aggregate            0.061***
##                               (0.012)
##
## Constant                     0.423***
##                               (0.015)
##
## -----------------------------------------------
## Observations                    30
## R2                             0.494
```

```
## Adjusted R2                    0.476
## Residual Std. Error       0.027 (df = 28)
## F Statistic           27.400*** (df = 1; 28)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

The coefficient of `payroll_aggregate` is 0.061. This means that for for every 1 billion dollar increase in payroll, the expected increase in win percentage is 0.061 or 6.1%. R^2 represents the fraction of the variation in `pct_wins_aggregate` explained by `payroll_aggregate`. This is 0.494 or 49.4%.

3.2 Comparing Oakland Athletics to the linear trend (10 points)

- Given their payroll, what is the linear regression prediction for the winning percentage of the Oakland Athletics? What was their actual winning percentage?
- Now run a linear regression of `payroll_aggregate` on `pct_wins_aggregate`. What is the linear regression prediction for the `payroll_aggregate` of the Oakland Athletics? What was their actual payroll?

Solution.

```r
# get the Oakland Athletics' aggregate payroll and aggregate percent wins
oakland_payroll_aggregate = mlb_aggregate %>%
  filter(team == "Oakland Athletics") %>%
  pull(payroll_aggregate)
oakland_pct_wins_aggregate = mlb_aggregate %>%
  filter(team == "Oakland Athletics") %>%
  pull(pct_wins_aggregate)
# prediction for winning percentage
0.423 + (0.061 * oakland_payroll_aggregate)
```

```
## [1] 0.474
```

```r
# actual winning percentage
oakland_pct_wins_aggregate
```

```
## [1] 0.545
```

```r
# create model for `payroll_aggregate` in terms of `pct_wins_aggregate`
#and display results
model2 = lm(payroll_aggregate ~ pct_wins_aggregate, data = mlb_aggregate)
stargazer(model2, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                          payroll_aggregate
## ------------------------------------------------
## pct_wins_aggregate          8.060***
```

```
##                                (1.540)
##
## Constant                       -2.780***
##                                (0.770)
##
## -----------------------------------------------
## Observations                      30
## R2                               0.494
## Adjusted R2                      0.476
## Residual Std. Error       0.309 (df = 28)
## F Statistic           27.400*** (df = 1; 28)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```
# prediction for payroll
-2.780 + (8.060 * oakland_pct_wins_aggregate)
```

```
## [1] 1.61
```

```
# actual payroll
oakland_payroll_aggregate
```

```
## [1] 0.841
```

The linear regression prediction for Oakland Athletics' winning percentage is 0.474. Their actual winning percentage is 0.545

The linear regression prediction for Oakland Athletics' aggregate payroll is 1.61 billion dollars. Their actual payroll is 0.841 billion dollars.