STAT 471: Homework 1

James Kuang

Due: September 15, 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: Alex Chen

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

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
```

```
## Warning: package 'ggrepel' was built under R version 4.1.1
```

```
library(kableExtra) # for printing tables
```

```
## Warning: package 'kableExtra' was built under R version 4.1.1
```

```
library(cowplot)    # for side by side plots
```

```
## Warning: package 'cowplot' was built under R version 4.1.1
```

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`.]

```
mlb_raw <- read_csv(file = "../../data/MLPayData_Total.csv") # Importing the csv
mlb_raw
```

```
## # 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 for 54 columns, which does match up with the data description. There are 30 teams and 54 columns described (3 + 3 x 17).

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.]

```
mlb_aggregate <- mlb_raw %>% # Creating the mlb_aggregate
  select("Team.name.2014", "payroll", "avgwin") %>% #Selecting the relevant columns
  rename("team" = "Team.name.2014", "payroll_aggregate" = "payroll",
         "pct_wins_aggregate" = "avgwin") #Renaming the columns
mlb_yearly <- mlb_raw %>% # Creating mlb_yearly
  select(!c("payroll", "avgwin")) %>% # Selecting the columns in the new tibble
  rename("team" = "Team.name.2014") %>% # Renaming the columns
  pivot_longer(!"team", names_to = "col_name", # Pivoting longer the non-team columns
               values_to = "values") %>%
  separate("col_name", c("prefix", "year", "suffix"), c(1, 5)) %>% #Separating the name
  mutate("tidy_col_name" =  # Recoding the prefix/suffix combinations to be meaningful
          recode(paste0(prefix,suffix), p = "payroll", X = "num_wins",
                 X.pct = "pct_wins"),
         .keep = "unused") %>%  #Keep only the unused columns
  pivot_wider(names_from = "tidy_col_name", # Pivoting wider with the new column names
              values_from = "values")
```

```
mlb_aggregate
```

```
## # 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
```

```
mlb_yearly
```

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

mlb_aggregate contains 30 rows, because it is aggregating for each team over the entire seventeen year period. However, mlb_yearly has 510 rows because it is providing information for all 30 teams over each of the 17 years.

1.3 Quality control (10 points)

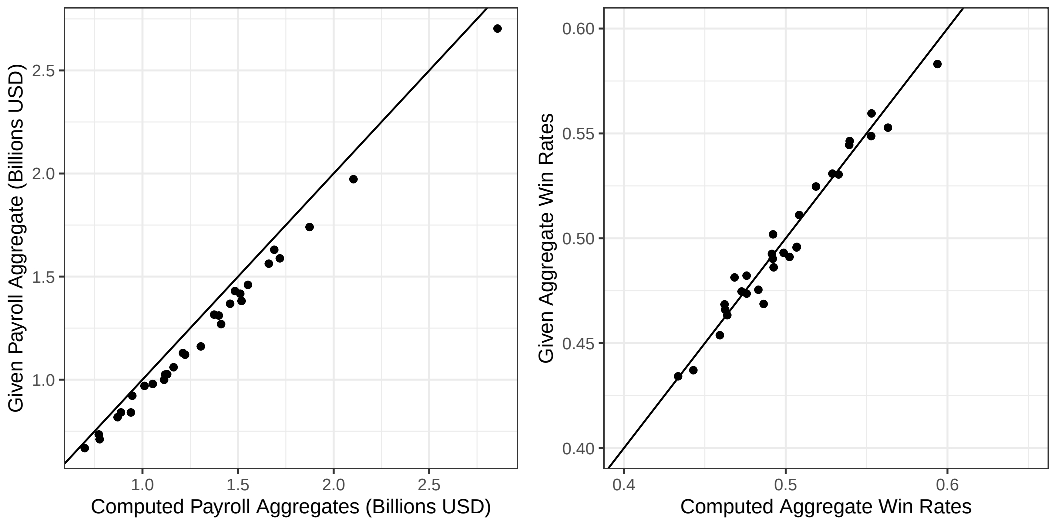
It's always a good idea to check whether a dataset is internally consistent. In this case, we are given both aggregated and yearly data, so we can check whether these match. To this end, carry out the following steps:

- 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.

```
mlb_aggregate_computed <- mlb_yearly %>% #Creating the computed values' tibble
  group_by(team) %>%
  summarise(payroll_aggregate_computed = sum(payroll)/1000, #Adding up the payrolls
            pct_wins_aggregate_computed =
              sum(num_wins)/sum(num_wins/pct_wins)) #Computing win percentages
mlb_aggregate_joined <- mlb_aggregate %>% # Creating a merged tibble
  merge(mlb_aggregate_computed)
```

```
# Plotting the computed vs actual payroll values
payroll_comparison <- ggplot(mlb_aggregate_joined,
                             aes(x= payroll_aggregate_computed, y = payroll_aggregate)) +
  geom_point() +
  geom_abline(slope=1, intercept= 0) + #Adding the 45 degree line
  theme_bw() + # Setting a nice theme
  labs(x = "Computed Payroll Aggregates (Billions USD)", # Adding informative axis labels
```

```
       y = "Given Payroll Aggregate (Billions USD)")
# Plotting the computed vs actual pct_win values
pct_wins_comparison <- ggplot(mlb_aggregate_joined,
                             aes(x= pct_wins_aggregate_computed,
                                 y = pct_wins_aggregate)) +
  geom_point() +
  geom_abline(slope=1, intercept= 0) + #Adding the 45 degree line
  theme_bw() + # Setting a nice theme
  labs(x = "Computed Aggregate Win Rates", # Adding informative axis labels
       y = "Given Aggregate Win Rates") +
  scale_x_continuous(breaks = c(0.4, 0.5, 0.6), # Setting my own custom scales
                     limits = c(0.4, 0.65)) +
  scale_y_continuous(breaks = c(0.4, 0.45, 0.5, 0.55, 0.6),
                     limits = c(0.4, 0.6))
plot_grid(payroll_comparison, pct_wins_comparison, # Plotting the two graphs side by side
          align = "h")
```



These values tend to generally line up with one another. However, the computed payroll values slightly dip below those of the given values, which is likely due to rounding differnces.

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

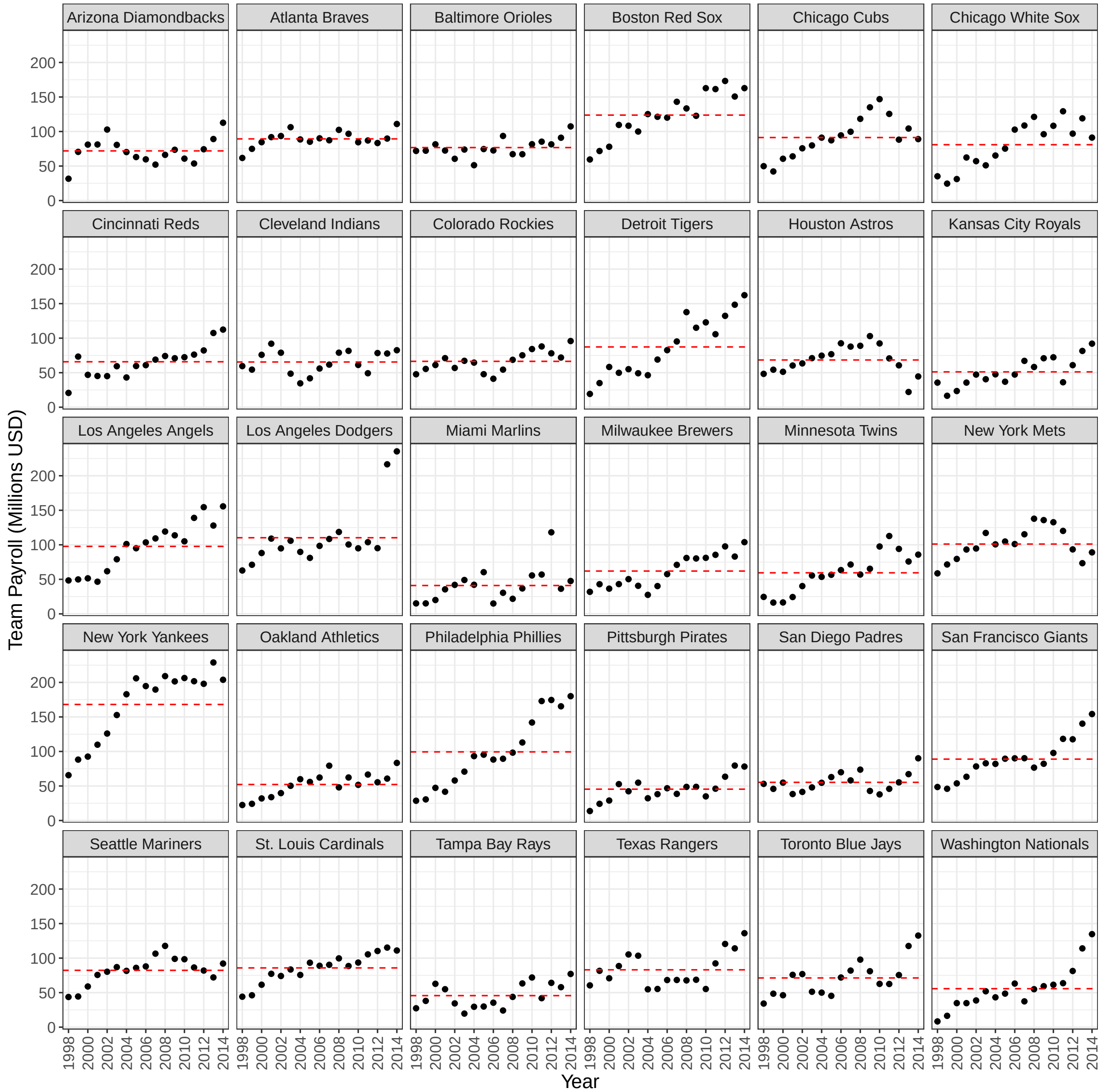
Now that the data are in tidy format, we can explore them by producing visualizations and summary statistics.

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.]

```
mlb_aggregate_computed <- mlb_aggregate_computed %>% # Calculate the average payroll
  mutate(average_payroll = payroll_aggregate_computed/17 * 1000)
ggplot(mlb_yearly, aes(x = year, y = payroll)) + # Plot the payroll by year
  geom_point() +
  scale_x_discrete(breaks=seq(1998, 2014, 2)) +
  facet_wrap(team ~ .) + # Facet this graph by teams
  # Add the dashed, red lines of the mean payroll to each graph
  geom_hline(mlb_aggregate_computed,
             mapping = aes(yintercept = average_payroll),
             linetype='dashed', col = 'red') +
  theme_bw() + # Use the black and white theme
  theme(axis.text.x = element_text(# A little theme working to make it better
    angle = 90, vjust = 0.5, hjust=.9),
    text = element_text(size = 14)) +
  labs(y = "Team Payroll (Millions USD)", x = "Year")
```



The three teams with the greatest payroll:

```
mlb_aggregate_computed %>% #Getting the three highest payroll teams
  # Arrange them from highest to lowest aggregate payroll
  arrange(desc(payroll_aggregate_computed)) %>%
  select(team, payroll_aggregate_computed) %>%
  rename(Team = team, "Computed Aggregate Payroll" = payroll_aggregate_computed) %>%
  slice(1:3) %>%  # Get the top 3
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "Teams with the highest aggregate payroll") %>%
  kable_styling( position = "center", latex_options = "HOLD_position")
```

Table 1: Teams with the highest aggregate payroll

Team	Computed Aggregate Payroll
New York Yankees	2.86
Boston Red Sox	2.10
Los Angeles Dodgers	1.87

The three teams with the greatest percentage increases in payroll:

```
mlb_yearly %>%
  # Get only the columns needed to find the greatest increase
  select(c(team, year, payroll)) %>%
  # Pivoting it wider to make it easier to calculate the increase
  pivot_wider(names_from = year, values_from = c(payroll), names_prefix = "payroll_") %>%
  mutate(pct_increase = #Calculating the increase
           (payroll_2004 - payroll_1998)/payroll_1998 * 100) %>%
  # Arranging the data from highest to lowest payroll increase
  arrange(desc(pct_increase)) %>%
  select(team, payroll_1998, payroll_2014, pct_increase) %>%
  rename(Team = team, "% Increase in Payroll" = pct_increase,
         "Payroll 1998 (Millions $)" = payroll_1998,
         "Payroll 2014 (Millions $)" = payroll_2014) %>%
  slice(1:3) %>% # Getting the top 3
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "Teams and their payroll increase") %>%
  kable_styling( position = "center", latex_options = "HOLD_position")
```

Table 2: Teams and their payroll increase

Team	Payroll 1998 (Millions $)	Payroll 2014 (Millions $)	% Increase in Payroll
Washington Nationals	8.32	135	419
Philadelphia Phillies	28.62	180	226
New York Yankees	65.66	204	178

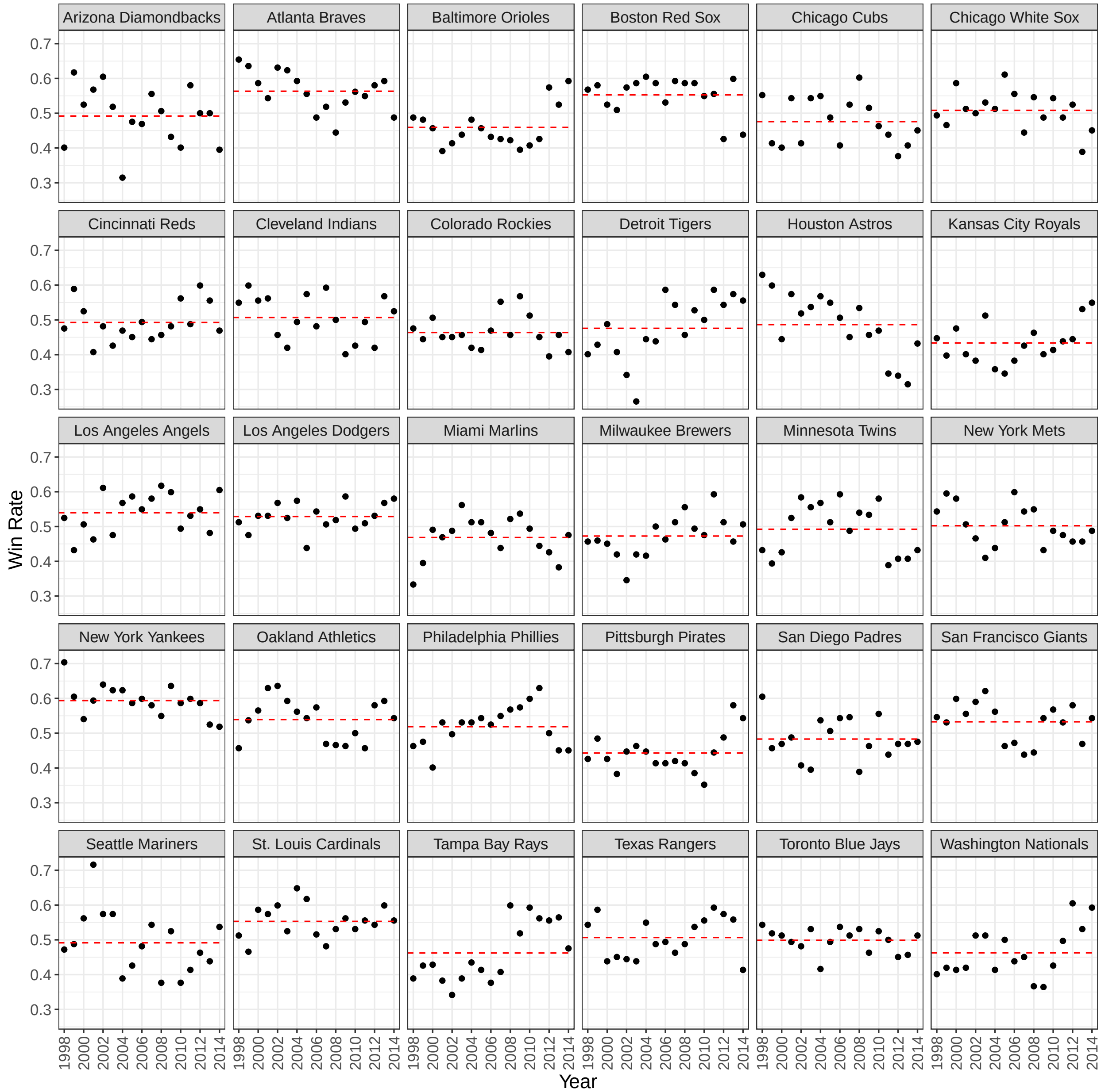
The payroll_aggregate_computed and pct_increase are reflected in the above plot by the red line (i.e mean payroll) and the trend of the dots, respectively. Mean payroll is just the aggregate payroll divided by seventeen. The trend of the dots is the literal increase in payroll value. We can see that the top three teams in payroll_aggregate_computed have the highest red line and the top three teams in pct_increase have the dots that increase the most.

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 `payroll_aggregate_computed` 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?

```
mlb_average_winrate <- mlb_yearly %>% # Calculating the average win rate for each team
  group_by(team) %>%
  summarise(avg_winrate = mean(pct_wins), # The average win rate
            pct_wins_sd = sd(pct_wins)) # The sd of the win rates
ggplot(mlb_yearly, aes(x = year, y = pct_wins)) + # Plot the win rate by year
  geom_point() +
  scale_x_discrete(breaks=seq(1998, 2014, 2)) +
  facet_wrap(team ~ .) + # Facet this graph by teams
  # Add the dashed, red lines of the mean win rate to each graph
  geom_hline(mlb_average_winrate,
             mapping = aes(yintercept = avg_winrate), linetype='dashed', col = 'red') +
  theme_bw() +
  theme(axis.text.x = # A little theme working to make it better
    element_text(angle = 90, vjust = 0.5, hjust=.9),
    text = element_text(size = 14)) +
  labs(y = "Win Rate", x = "Year")
```



The three teams with the greatest win rate:

```
mlb_aggregate %>% #Getting the three highest teams
 # Arrange them from highest to lowest aggregate payroll
  arrange(desc(pct_wins_aggregate)) %>%
  select(c("team", "pct_wins_aggregate")) %>%
  rename(Team = team, "Win Rate" = pct_wins_aggregate) %>%
  slice(1:3) %>%  # Get the top 3
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "Teams and their win rate") %>%
  kable_styling( position = "center", latex_options = "HOLD_position")
```

Table 3: Teams and their win rate

Team	Win Rate
New York Yankees	0.58
St. Louis Cardinals	0.56
Atlanta Braves	0.55

The three teams with the most erratic win rate:

```
mlb_average_winrate %>% #Getting the three highest teams
  # Arrange them from highest to lowest aggregate payroll
  arrange(desc(pct_wins_sd)) %>%
  select(c("team", "pct_wins_sd")) %>%
  rename(Team = team, "SD of Win Rate" = pct_wins_sd) %>%
  slice(1:3) %>%  # Get the top 3
  kable(format = "latex", row.names = NA,
        booktabs = TRUE, digits = 2,
        caption = "Teams and the SD of their win rate") %>%
  kable_styling( position = "center", latex_options = "HOLD_position")
```

Table 4: Teams and the SD of their win rate

Team	SD of Win Rate
Houston Astros	0.09
Detroit Tigers	0.09
Seattle Mariners	0.09

The pct__wins__aggregate and pct__wins__sd are reflected in the plot above by the red line (i.e. mean win rate) and the movement of the dots, respectively. Mean win rate is just the aggregate payroll divided by seventeen. The movement of the dots is the literal variation in win rate We can see that the top three teams in pct__wins__aggregate have the highest red line and the top three teams in pct__wins__sd have the dots that vary the most.

2.3 Win percentage versus payroll (10 points)

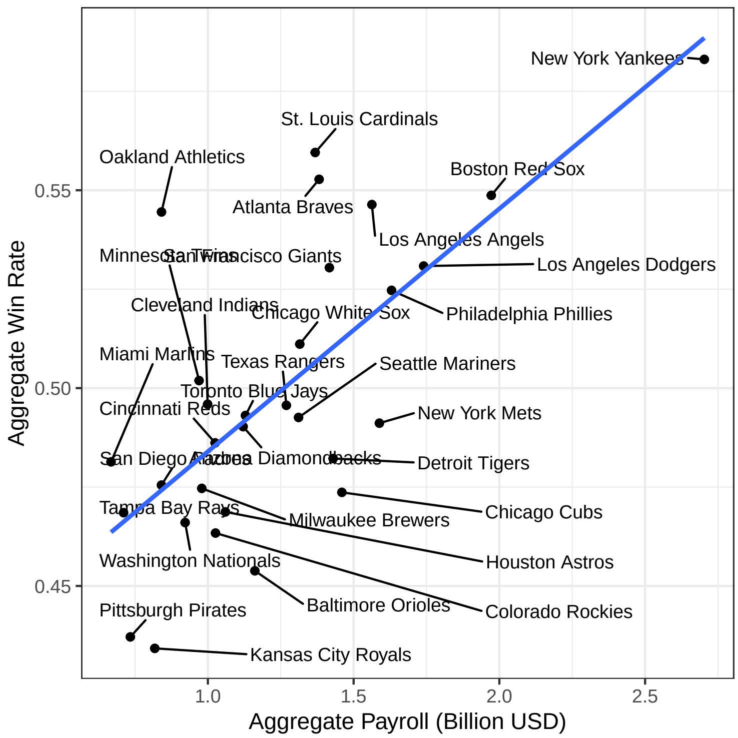
The analysis goal is to study the relationship between win percentage and payroll.

- 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?

```
# Creating the aesthetic for the payroll vs pct wins graph
ggplot(data = mlb_aggregate, aes(x=payroll_aggregate, y = pct_wins_aggregate)) +
  geom_point() + # Adding the points
  geom_text_repel(aes(label = team), # Adding the labels
                  # Adjusting some settings to make it look better
                  box.padding = 0.6, size = 3, max.overlaps = 20,
                  min.segment.length = 0.2,
                  force = 4) +
```

```r
  geom_smooth(method='lm', se= FALSE) + # Adding the least squares line
  theme_bw() +
  labs(x = "Aggregate Payroll (Billion USD)", y = "Aggregate Win Rate")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The relationship between payroll and win_rate is positive, which makes sense as more money means better players and therefore 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."

```r
mlb_aggregate <- mlb_aggregate %>%
  mutate(efficiency = pct_wins_aggregate/payroll_aggregate)
#Getting the three highest efficiency teams
mlb_aggregate %>%
  # Arrange them from highest to lowest aggregate payroll
  arrange(desc(efficiency)) %>%
  select(c("team", "efficiency")) %>%
  rename(Team = team, Efficiency = efficiency) %>%
  slice(1:3) %>%  # Get the top 3
```

```
kable(format = "latex", row.names = NA,
      booktabs = TRUE, digits = 2,
      caption = "Highest efficiency teams") %>%
kable_styling( position = "center", latex_options = "HOLD_position")
```

Table 5: Highest efficiency teams

Team	Efficiency
Miami Marlins	0.72
Tampa Bay Rays	0.66
Oakland Athletics	0.65

These are the teams with the highest win rate given their payrolls, so they will be in the top left of the scatter plot.

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`?

```
lm_pctwins <- lm(mlb_aggregate, formula = pct_wins_aggregate~payroll_aggregate) #
summary(lm_pctwins)
```

```
##
## Call:
## lm(formula = pct_wins_aggregate ~ payroll_aggregate, data = mlb_aggregate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.04003 -0.01749  0.00094  0.01095  0.07030
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.4226     0.0153   27.56  < 2e-16 ***
## payroll_aggregate   0.0614     0.0117    5.23  1.5e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.027 on 28 degrees of freedom
## Multiple R-squared:  0.494,  Adjusted R-squared:  0.476
## F-statistic: 27.4 on 1 and 28 DF,  p-value: 1.47e-05
```

The coefficient is 0.0614. This means that for every billion dollars spent on the team, the winrate increased by 0.0614. Only 0.494 of the variation is explained.

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?

```
oakland_athletics <- mlb_aggregate %>%
  filter(team == "Oakland Athletics")
predict(lm_pctwins, oakland_athletics)
```

```
##     1
## 0.474
```

The predicted win rate is 0.474, while their actual win rate was 0.545.

```
lm_payroll <- lm(formula = payroll_aggregate~pct_wins_aggregate, data =mlb_aggregate)
summary(lm_payroll)
```

```
##
## Call:
## lm(formula = payroll_aggregate ~ pct_wins_aggregate, data = mlb_aggregate)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7674 -0.1871 -0.0371  0.1663  0.7843
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -2.78       0.77   -3.61   0.0012 **
## pct_wins_aggregate     8.06       1.54    5.23  1.5e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.309 on 28 degrees of freedom
## Multiple R-squared:  0.494,  Adjusted R-squared:  0.476
## F-statistic: 27.4 on 1 and 28 DF,  p-value: 1.47e-05
```

```
predict(lm_payroll, oakland_athletics)
```

```
##    1
## 1.61
```

The predicted payroll of the Oakland A's is $1.61 billion, while the actual is $0.841 million.