DATA MINING AND WRANGLING

DSC1107

Formative Assessment 2

BILLONES, Cristel Kaye P.
2021016541

Case study: Major League Baseball

#1. WRANGLE

library(tidyverse)
library (dplyr)

load("C:/Users/Cipher/Desktop/CRISTEL_DMW/ml_pay.rdata")
# Check the objects in the environment
ls()
Ls

```
ml_pay              30 obs. of 54 variables
  $ payroll        : num  1.12 1.38 1.16 1.97 1.46 ...
  $ avgwin         : num  0.49 0.553 0.454 0.549 0.474 ...
  $ Team.name.2014: Factor w/ 30 levels "Arizona Diamondbacks",..: 1 2 3 4 5…
  $ p1998          : num  31.6 61.7 71.9 59.5 49.8 ...
  $ p1999          : num  70.5 74.9 72.2 71.7 42.1 ...
  $ p2000          : num  81 84.5 81.4 77.9 60.5 ...
  $ p2001          : num  81.2 91.9 72.4 109.6 64 ...
  $ p2002          : num  102.8 93.5 60.5 108.4 75.7 ...
  $ p2003          : num  80.6 106.2 73.9 99.9 79.9 ...
  $ p2004          : num  70.2 88.5 51.2 125.2 91.1 ...
  $ p2005          : num  63 85.1 74.6 121.3 87.2 ...
  $ p2006          : num  59.7 90.2 72.6 120.1 94.4
```

#1.1 : Import
mlb_raw <- as_tibble(ml_pay)
print(mlb_raw)
colnames(mlb_raw)
View(ml_pay)

	payroll	avgwin	Team.name.2014	p1998	p1999	p2000	p2001	p2002	p2003	p2004	p2005	p2006
1	1.1208736	0.4902585	Arizona Diamondbacks	31.61450	70.49600	81.02783	81.20651	102.82000	80.64033	70.20498	63.01583	59.68
2	1.3817118	0.5527605	Atlanta Braves	61.70800	74.89000	84.53784	91.85169	93.47037	106.24367	88.50779	85.14858	90.15
3	1.1612117	0.4538250	Baltimore Orioles	71.86092	72.19836	81.44743	72.42633	60.49349	73.87750	51.21265	74.57054	72.58
4	1.9723587	0.5487172	Boston Red Sox	59.49700	71.72500	77.94033	109.55891	108.36606	99.94650	125.20854	121.31194	120.09
5	1.4597668	0.4736557	Chicago Cubs	49.81600	42.14276	60.53933	64.01583	75.69083	79.86833	91.10167	87.21093	94.42
6	1.3153909	0.5111170	Chicago White Sox	35.18000	24.53500	31.13350	62.36300	57.05283	51.01000	65.21250	75.22800	102.75
7	1.0247816	0.4861602	Cincinnati Reds	20.70733	73.27846	46.86720	45.22788	45.05039	59.35567	43.06786	59.65828	60.90
8	0.9991810	0.4959225	Cleveland Indians	59.54317	54.44250	75.88087	91.97498	78.90945	48.58483	34.56930	41.83040	56.03
9	1.0261536	0.4633760	Colorado Rockies	47.71465	55.44350	61.11119	71.06800	56.85104	67.17967	64.59040	47.78900	41.23
10	1.4297408	0.4822029	Detroit Tigers	19.23750	34.95967	58.26517	49.83117	55.04800	49.16800	46.35355	68.99818	82.61
11	1.0601501	0.4687202	Houston Astros	48.30400	54.33900	51.28911	60.38267	63.44842	71.04000	74.66630	76.77902	92.55

# How many rows and columns does the data have?
dim(mlb_raw)

```
> dim(mlb_raw)
[1] 30 54
```

# Does this match up with the data description given above?
# Check the column names of the dataset
column_names <- names(mlb_raw)
print(column_names)

```
 [1] "payroll"     "avgwin"      "Team.name.2014" "p1998"      "p1999"
 [6] "p2000"       "p2001"       "p2002"       "p2003"       "p2004"
[11] "p2005"       "p2006"       "p2007"       "p2008"       "p2009"
[16] "p2010"       "p2011"       "p2012"       "p2013"       "p2014"
[21] "X2014"       "X2013"       "X2012"       "X2011"       "X2010"
[26] "X2009"       "X2008"       "X2007"       "X2006"       "X2005"
[31] "X2004"       "X2003"       "X2002"       "X2001"       "X2000"
[36] "X1999"       "X1998"       "X2014.pct"   "X2013.pct"   "X2012.pct"
[41] "X2011.pct"   "X2010.pct"   "X2009.pct"   "X2008.pct"   "X2007.pct"
[46] "X2006.pct"   "X2005.pct"   "X2004.pct"   "X2003.pct"   "X2002.pct"
[51] "X2001.pct"   "X2000.pct"   "X1999.pct"   "X1998.pct"
```

#1.2 : Tidy
Create mlb_aggregate tibble

```
mlb_aggregate <- mlb_raw %>%
  select(`Team.name.2014`, payroll, matches("^X\\d{4}\\.pct$")) %>%
  mutate(payroll_aggregate = payroll,
       pct_wins_aggregate = rowSums(across(matches("^X\\d{4}\\.pct$")))) %>%
  select(-payroll, -matches("^X\\d{4}\\.pct$")) %>%
  rename(team = `Team.name.2014`)
```

```
mlb_aggregate          30 obs. of 3 variables
    $ team              : Factor w/ 30 levels "Arizona Diamondbacks",..: 1 2 3…
    $ payroll_aggregate : num [1:30] 1.12 1.38 1.16 1.97 1.46 ...
    $ pct_wins_aggregate: num [1:30] 8.37 9.57 7.77 9.37 8.07 ...
mlb_raw                30 obs. of 54 variables
```

Create mlb_total with columns team, payroll_aggregate, pct_wins_aggregate

```
mlb_total <- mlb_aggregate %>%
  select(team, payroll_aggregate, pct_wins_aggregate)
```

print(mlb_total)

```
mlb_total              30 obs. of 3 variables
    $ team              : Factor w/ 30 levels "Arizona Diamondbacks",..: 1 2 3…
    $ payroll_aggregate : num [1:30] 1.12 1.38 1.16 1.97 1.46 ...
    $ pct_wins_aggregate: num [1:30] 8.37 9.57 7.77 9.37 8.07 ...
```

```
mlb_yearly <- mlb_raw %>%
  pivot_longer(cols = starts_with("p"), names_to = "year", values_to = "pct_wins") %>%
  mutate(year = as.integer(gsub("\\D", "", year))) %>%
  select(`Team.name.2014`, year, pct_wins) %>%
  left_join(select(mlb_raw, `Team.name.2014`, payroll, avgwin), by = "Team.name.2014") %>%
  arrange(`Team.name.2014`, year)
```

```
mlb_yearly          540 obs. of 5 variables
    $ Team.name.2014: Factor w/ 30 levels "Arizona Diamondbacks",..: 1 1 1 1 1...
    $ year          : int [1:540] 1998 1999 2000 2001 2002 2003 2004 2005 2006...
    $ pct_wins      : num [1:540] 31.6 70.5 81 81.2 102.8 ...
    $ payroll       : num [1:540] 1.12 1.12 1.12 1.12 1.12 ...
    $ avgwin        : num [1:540] 0.49 0.49 0.49 0.49 0.49 ...
```

```
#Check number of rows
nrow(mlb_aggregate)
nrow(mlb_yearly)
```

```
> nrow(mlb_aggregate)
[1] 30
> nrow(mlb_yearly)
[1] 540
```

```
#1.3
library(dplyr)

mlb_aggregate_computed <- mlb_total %>%
  group_by(team) %>%
  summarise(
    payroll_aggregate_computed = sum(payroll_aggregate),
    pct_wins_aggregate_computed = sum(pct_wins_aggregate)
  )
```

```
mlb_aggregate_compu...   30 obs. of 3 variables
    $ team                       : Factor w/ 30 levels "Arizona Diamondbacks",...
    $ payroll_aggregate_computed : num [1:30] 1.12 1.38 1.16 1.97 1.46 ...
    $ pct_wins_aggregate_computed: num [1:30] 8.37 9.57 7.77 9.37 8.07 ...
```

```
# Print mlb_aggregate_computed to check the result
print(mlb_aggregate_computed)
```

```
# A tibble: 30 × 3
   team                 payroll_aggregate_computed pct_wins_aggregate_computed
   <fct>                                     <dbl>                       <dbl>
 1 Arizona Diamondbacks                      1.12                         8.37
 2 Atlanta Braves                            1.38                         9.57
 3 Baltimore Orioles                         1.16                         7.77
 4 Boston Red Sox                            1.97                         9.37
 5 Chicago Cubs                              1.46                         8.07
 6 Chicago White Sox                         1.32                         8.62
 7 Cincinnati Reds                           1.02                         8.35
 8 Cleveland Indians                         0.999                        8.58
 9 Colorado Rockies                          1.03                         7.86
10 Detroit Tigers                            1.43                         8.06
```

Join mlb_aggregate and mlb_aggregate_computed

mlb_aggregate_joined <- inner_join(mlb_aggregate, mlb_aggregate_computed, by = "team")

```
mlb_aggregate_joined   30 obs. of 5 variables
    $ team                     : Factor w/ 30 levels "Arizona Diamondbacks",..
    $ payroll_aggregate        : num [1:30] 1.12 1.38 1.16 1.97 1.46 ...
    $ pct_wins_aggregate       : num [1:30] 8.37 9.57 7.77 9.37 8.07 ...
    $ payroll_aggregate_computed : num [1:30] 1.12 1.38 1.16 1.97 1.46 ...
    $ pct_wins_aggregate_computed: num [1:30] 8.37 9.57 7.77 9.37 8.07 ...
```

Print mlb_aggregate_joined to check the result

print(mlb_aggregate_joined)

```
# A tibble: 30 × 5
   team         payroll_aggregate pct_wins_aggregate payroll_aggregate_co…¹ pct_wins_aggregate_c…²
   <fct>                    <dbl>              <dbl>                  <dbl>                  <dbl>
 1 Arizona Dia…             1.12                8.37                  1.12                    8.37
 2 Atlanta Bra…             1.38                9.57                  1.38                    9.57
 3 Baltimore O…             1.16                7.77                  1.16                    7.77
 4 Boston Red …             1.97                9.37                  1.97                    9.37
 5 Chicago Cubs             1.46                8.07                  1.46                    8.07
 6 Chicago Whi…             1.32                8.62                  1.32                    8.62
 7 Cincinnati …             1.02                8.35                  1.02                    8.35
 8 Cleveland I…             0.999               8.58                  0.999                   8.58
 9 Colorado Ro…             1.03                7.86                  1.03                    7.86
10 Detroit Tig…             1.43                8.06                  1.43                    8.06
# ℹ 20 more rows
```

#Scatter Plots

install.packages("gridExtra")

library(gridExtra)

install.packages("ggplot2")

library(ggplot2)

payroll_scatter_plot <- ggplot(mlb_aggregate_joined, aes(x = payroll_aggregate, y = payroll_aggregate_computed)) +

geom_point() +

geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +

labs(x = "Payroll Aggregate", y = "Computed Payroll Aggregate", title = "Payroll Aggregate vs. Computed Payroll Aggregate")

Scatter plot for pct_wins_aggregate

pct_wins_plot <- ggplot(mlb_aggregate_joined, aes(x = pct_wins_aggregate, y = pct_wins_aggregate_computed)) +

geom_point() +

geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +

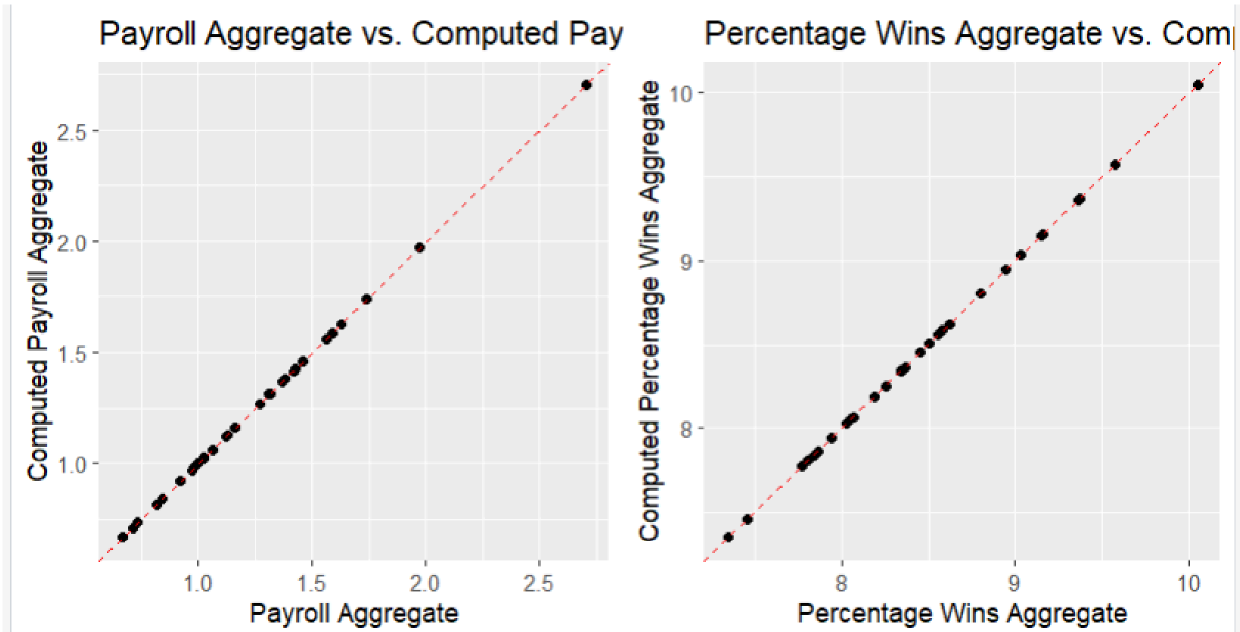
labs(x = "Percentage Wins Aggregate", y = "Computed Percentage Wins Aggregate", title = "Percentage Wins Aggregate vs. Computed Percentage Wins Aggregate")

Arrange plots side by side

combined_plots <- grid.arrange(payroll_scatter_plot, pct_wins_plot, ncol = 2)

Display the combined plots

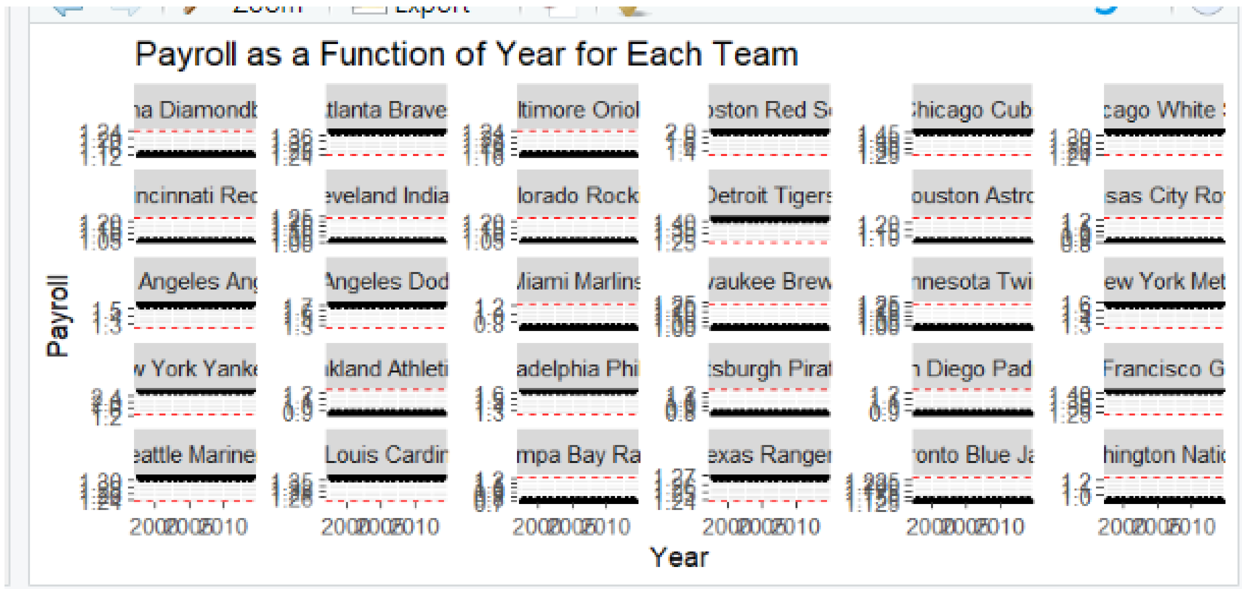
print(combined_plots)



#2 EXPLORE

#2.1
library(ggplot2)
#2.1.1
Convert Team.name.2014 to factor for correct ordering in facet_wrap
mlb_yearly$Team.name.2014 <- factor(mlb_yearly$Team.name.2014, levels = unique(mlb_yearly$Team.name.2014))

```
# Plot payroll as a function of year, faceting by team
payroll_plot <- ggplot(mlb_yearly, aes(x = year, y = payroll)) +
  geom_point() +
  geom_line() +
  facet_wrap(~Team.name.2014, scales = "free_y") +
  labs(x = "Year", y = "Payroll", title = "Payroll as a Function of Year for Each Team")

# Add a red dashed horizontal line for the mean payroll across years for each team
payroll_plot <- payroll_plot +
  geom_hline(aes(yintercept = mean(payroll), color = "Mean Payroll"), linetype = "dashed") +
  scale_color_manual(values = "red", guide = FALSE)

# Display the plot
print(payroll_plot)
```



#2.1.2
library(dplyr)

# Identify the three teams with the greatest payroll_aggregate_computed
top_three_teams <- mlb_aggregate_computed %>%
 arrange(desc(payroll_aggregate_computed)) %>%
 head(3)

# Print a table of the top three teams and their payroll_aggregate_computed
print(top_three_teams)

```
  team                 payroll_aggregate_computed pct_wins_aggregate_computed
  <fct>                                     <dbl>                       <dbl>
1 New York Yankees                           2.70                       10.0
2 Boston Red Sox                             1.97                        9.37
3 Los Angeles Dodgers                        1.74                        8.94
```

#2.1.3
library(dplyr)

# Calculate payroll figures for 1998 and 2014
mlb_payroll_1998_2014 <- mlb_raw %>%
 select(Team.name.2014, p1998, p2014) %>%
 rename(team = Team.name.2014, payroll_1998 = p1998, payroll_2014 = p2014)

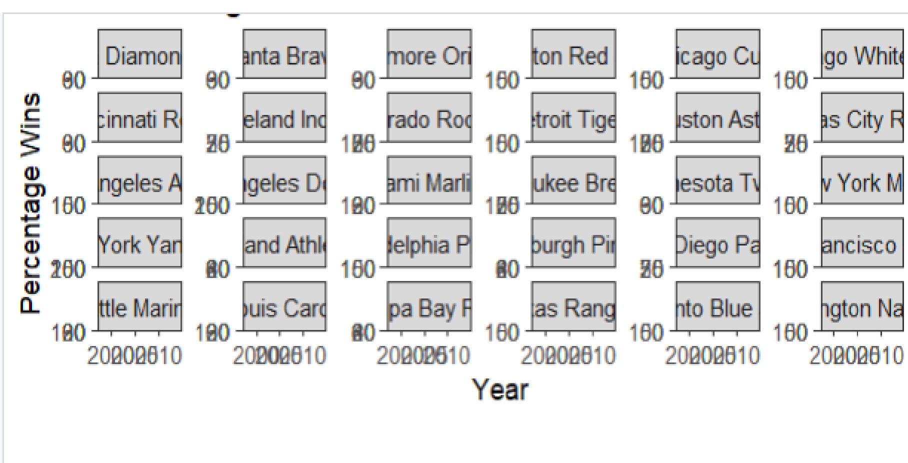
# Calculate pct_increase
mlb_payroll_1998_2014 <- mlb_payroll_1998_2014 %>%
 mutate(pct_increase = ((payroll_2014 - payroll_1998) / payroll_1998) * 100)

```
# Identify the three teams with the greatest percentage increase in payroll
top_three_increase <- mlb_payroll_1998_2014 %>%
  arrange(desc(pct_increase)) %>%
  head(3)

# Print a table of the top three teams with their payroll figures from 1998 and 2014, and
pct_increase
print(top_three_increase)
```

```
# A tibble: 3 x 4
  team                  payroll_1998 payroll_2014 pct_increase
  <fct>                        <dbl>        <dbl>        <dbl>
1 Washington Nationals          8.32         135.        1520.
2 Detroit Tigers               19.2          162.         743.
3 Philadelphia Phillies        28.6          180.         529.
```

#2.1.4

```
#2.2.1
pct_wins_plot <- ggplot(mlb_yearly, aes(x = year, y = pct_wins)) +
  geom_point() +                    # Add points
  geom_hline(aes(yintercept = mean(pct_wins), color = "Average"), linetype = "dashed") +  #
Add average line
  facet_wrap(~ Team.name.2014, scales = "free_y") +  # Facet by team
  labs(x = "Year", y = "Percentage Wins", title = "Percentage Wins vs. Year") +  # Labels
  theme_bw() +                      # White background theme
  theme(legend.position = "bottom") # Legend position

# Display the plot
print(pct_wins_plot)
```

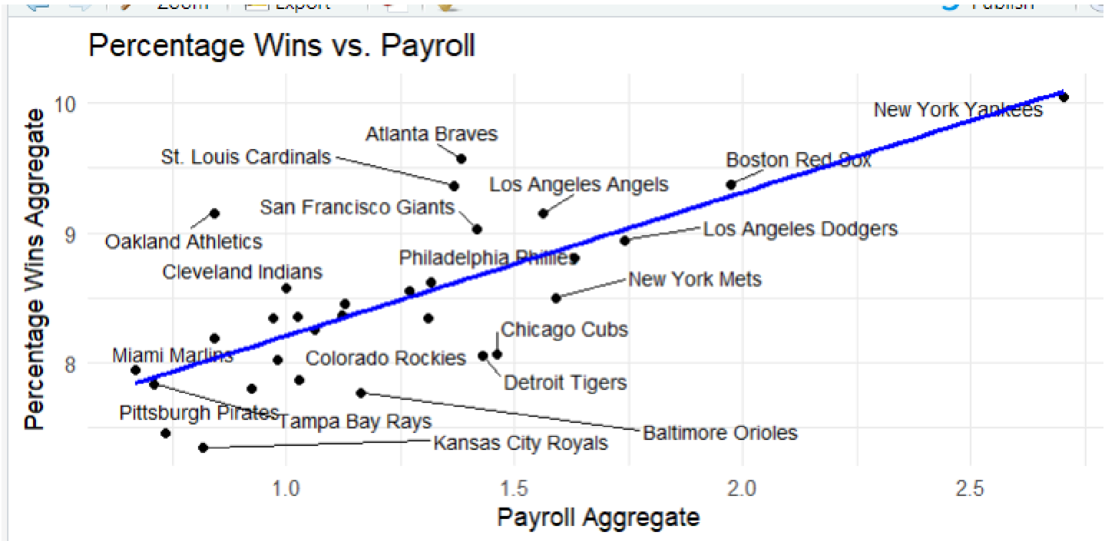


```
#2.2.3
library(dplyr)
# Calculate the standard deviation of pct_wins for each team
team_pct_wins_sd <- mlb_yearly %>%
  group_by(Team.name.2014) %>%
  summarise(pct_wins_sd = sd(pct_wins, na.rm = TRUE)) %>%
  ungroup()

# Identify the three teams with the most erratic pct_wins across years
top_three_erratic_teams <- team_pct_wins_sd %>%
  arrange(desc(pct_wins_sd)) %>%
  slice_head(n = 3)

# Print a table of these teams along with pct_wins_sd
print(top_three_erratic_teams)
```

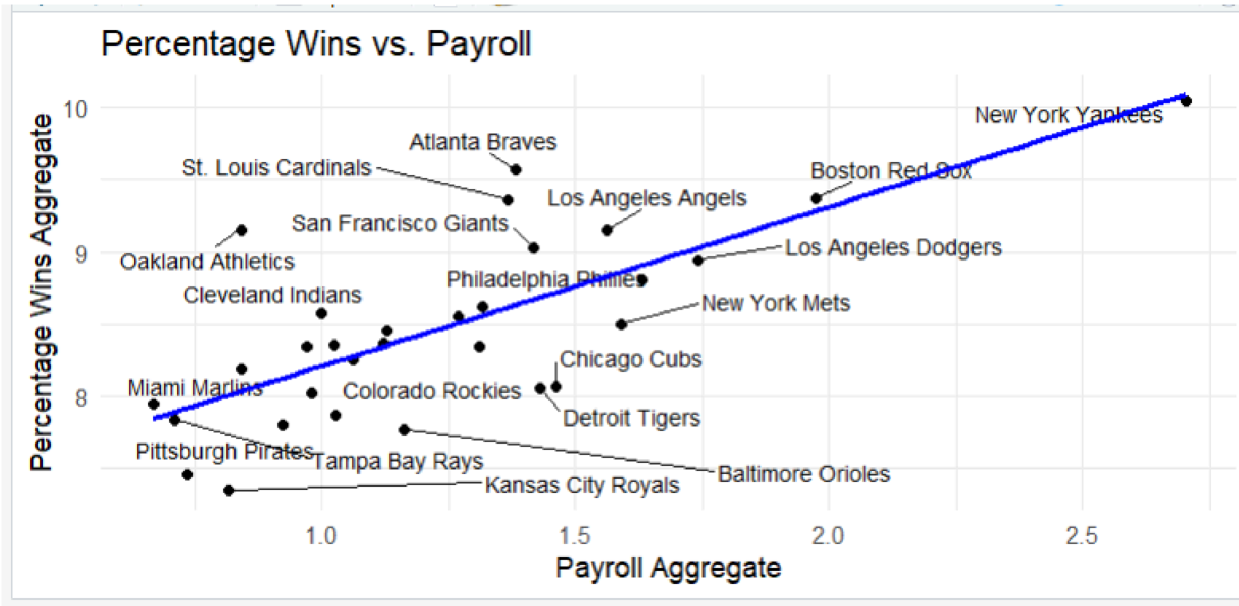
```
  Team.name.2014        pct_wins_sd
  <fct>                       <dbl>
1 New York Yankees             63.2
2 Philadelphia Phillies        55.1
3 Los Angeles Dodgers          51.3
```

```
#2.3
library(ggplot2)
install.packages("ggrepel")
library(ggrepel)

# Create scatter plot with labels
scatter_plot <- ggplot(mlb_aggregate, aes(x = payroll_aggregate, y = pct_wins_aggregate, label = team)) +
  geom_point() +  # Scatter plot
  geom_text_repel(size = 3, box.padding = unit(0.5, "lines")) +  # Add labels with ggrepel
  geom_smooth(method = "lm", se = FALSE, color = "blue") +  # Add least squares line
  labs(x = "Payroll Aggregate", y = "Percentage Wins Aggregate", title = "Percentage Wins vs. Payroll") +
  theme_minimal()  # Optional: Customize the theme if needed

# Print the scatter plot
```

```
print(scatter_plot)
```




```
#2.4
library(dplyr)

# Assuming you have a tibble named mlb_aggregate_computed containing columns:
# team, pct_wins_aggregate_computed, and payroll_aggregate_computed

# Calculate efficiency (wins per unit of payroll)
mlb_efficiency <- mlb_aggregate_computed %>%
  mutate(efficiency = pct_wins_aggregate_computed / payroll_aggregate_computed)

# Identify the top three teams with the greatest efficiency
top_three_efficiency <- mlb_efficiency %>%
  top_n(3, efficiency) %>%
  arrange(desc(efficiency))  # Arrange in descending order of efficiency

# Print a table of the top three teams along with their efficiency,
# pct_wins_aggregate_computed, and payroll_aggregate_computed
print(top_three_efficiency)
```



Percentage Wins vs. Payroll
Percentage Wins Aggregate
Payroll Aggregate
10
9
8
1.0
1.5
2.0
2.5
New York Yankees
Atlanta Braves
St. Louis Cardinals
Boston Red Sox
Los Angeles Angels
San Francisco Giants
Oakland Athletics
Los Angeles Dodgers
Philadelphia Phillies
Cleveland Indians
New York Mets
Chicago Cubs
Miami Marlins
Colorado Rockies
Detroit Tigers
Pittsburgh Pirates
Tampa Bay Rays
Baltimore Orioles
Kansas City Royals