DSC_1107_FA2

Code

Frances Aneth Rosales

2024-02-26

Applying the Packages First

Code

```
## Warning: package 'tidyverse' was built under R version 4.2.3
```

```
## Warning: package 'tibble' was built under R version 4.2.3
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```
## Warning: package 'readr' was built under R version 4.2.3
```

```
## Warning: package 'purrr' was built under R version 4.2.3
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
## Warning: package 'stringr' was built under R version 4.2.3
```

```
## Warning: package 'forcats' was built under R version 4.2.3
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.1     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflict
s to become errors
```

Code

```
## Warning: package 'ggrepel' was built under R version 4.2.3
```

Code

```
## Warning: package 'kableExtra' was built under R version 4.2.3
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

Code

```
## Warning: package 'cowplot' was built under R version 4.2.3
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

1 Wrangle (35 points for correctness; 5 points for presentation)

1.1 Import (5 points)

- Import the data into a tibble called mlb_raw and print it.

Code

```
## # A tibble: 30 × 54
##    payroll avgwin Team.n…¹ p1998 p1999 p2000 p2001 p2002 p2003 p2004 p2005 p2006
##      <dbl>  <dbl> <fct>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1   1.12   0.490 Arizona…  31.6  70.5  81.0  81.2 103.   80.6  70.2  63.0  59.7
##  2   1.38   0.553 Atlanta…  61.7  74.9  84.5  91.9  93.5 106.   88.5  85.1  90.2
##  3   1.16   0.454 Baltimo…  71.9  72.2  81.4  72.4  60.5  73.9  51.2  74.6  72.6
##  4   1.97   0.549 Boston …  59.5  71.7  77.9 110.  108.   99.9 125.  121.  120.
##  5   1.46   0.474 Chicago…  49.8  42.1  60.5  64.0  75.7  79.9  91.1  87.2  94.4
##  6   1.32   0.511 Chicago…  35.2  24.5  31.1  62.4  57.1  51.0  65.2  75.2 103.
##  7   1.02   0.486 Cincinn…  20.7  73.3  46.9  45.2  45.1  59.4  43.1  59.7  60.9
##  8   0.999  0.496 Clevela…  59.5  54.4  75.9  92.0  78.9  48.6  34.6  41.8  56.0
##  9   1.03   0.463 Colorad…  47.7  55.4  61.1  71.1  56.9  67.2  64.6  47.8  41.2
## 10   1.43   0.482 Detroit…  19.2  35.0  58.3  49.8  55.0  49.2  46.4  69.0  82.6
## # … with 20 more rows, 42 more variables: p2007 <dbl>, p2008 <dbl>,
## #   p2009 <dbl>, p2010 <dbl>, p2011 <dbl>, p2012 <dbl>, p2013 <dbl>,
## #   p2014 <dbl>, X2014 <int>, X2013 <int>, X2012 <int>, X2011 <int>,
## #   X2010 <int>, X2009 <int>, X2008 <int>, X2007 <int>, X2006 <int>,
## #   X2005 <int>, X2004 <int>, X2003 <int>, X2002 <int>, X2001 <int>,
## #   X2000 <int>, X1999 <int>, X1998 <int>, X2014.pct <dbl>, X2013.pct <dbl>,
## #   X2012.pct <dbl>, X2011.pct <dbl>, X2010.pct <dbl>, X2009.pct <dbl>, …
```

Code

- How many rows and columns does the data have?

Code

```
## Number of rows of ML Pay: 30
```

Code

```
## Number of columns of ML Pay: 54
```

- Does this match up with the data description given above?

In accordance of the data that I imported into the table with the given data description, it is indeed similar data as assumed.

Therefore, we can now analyze the data of Major League Baseball (MLB) teams' payroll.

1.2 Tidy (15 points)

We need to change the variables containing the following
mlb_aggregate c: aggregate data
mlb_yearly: year-by-year data
mlb_total: columns named team,payroll_aggregate, pct_wins_aggregate
mlb_yearly: contain columns named team, year, payroll, pct_wins, num_wins

mlb_aggregate tibble

Code

```
## # A tibble: 30 × 3
##    team                 payroll_aggregate pct_wins_aggregate
##    <fct>                            <dbl>              <dbl>
##  1 Arizona Diamondbacks             1.12               0.492
##  2 Atlanta Braves                   1.38               0.563
##  3 Baltimore Orioles                1.16               0.457
##  4 Boston Red Sox                   1.97               0.551
##  5 Chicago Cubs                     1.46               0.475
##  6 Chicago White Sox                1.32               0.507
##  7 Cincinnati Reds                  1.02               0.491
##  8 Cleveland Indians                0.999              0.505
##  9 Colorado Rockies                 1.03               0.463
## 10 Detroit Tigers                   1.43               0.474
## # … with 20 more rows
```

mlb_yearly tibble

Code

```
## # A tibble: 30 × 53
##    team  the_p…¹ payro…² payro…³ payro…⁴ payro…⁵ payro…⁶ payro…⁷ payro…⁸ payro…⁹
##    <fct>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 Ariz…   1.12     31.6    70.5    81.0    81.2   103.     80.6    70.2    63.0
##  2 Atla…   1.38     61.7    74.9    84.5    91.9    93.5   106.     88.5    85.1
##  3 Balt…   1.16     71.9    72.2    81.4    72.4    60.5    73.9    51.2    74.6
##  4 Bost…   1.97     59.5    71.7    77.9   110.    108.     99.9   125.    121.
##  5 Chic…   1.46     49.8    42.1    60.5    64.0    75.7    79.9    91.1    87.2
##  6 Chic…   1.32     35.2    24.5    31.1    62.4    57.1    51.0    65.2    75.2
##  7 Cinc…   1.02     20.7    73.3    46.9    45.2    45.1    59.4    43.1    59.7
##  8 Clev…   0.999    59.5    54.4    75.9    92.0    78.9    48.6    34.6    41.8
##  9 Colo…   1.03     47.7    55.4    61.1    71.1    56.9    67.2    64.6    47.8
## 10 Detr…   1.43     19.2    35.0    58.3    49.8    55.0    49.2    46.4    69.0
## # … with 20 more rows, 43 more variables: payroll_2006 <dbl>,
## #   payroll_2007 <dbl>, payroll_2008 <dbl>, payroll_2009 <dbl>,
## #   payroll_2010 <dbl>, payroll_2011 <dbl>, payroll_2012 <dbl>,
## #   payroll_2013 <dbl>, payroll_2014 <dbl>, pct_wins_x2014.pct <dbl>,
## #   pct_wins_x2013.pct <dbl>, pct_wins_x2012.pct <dbl>,
## #   pct_wins_x2011.pct <dbl>, pct_wins_x2010.pct <dbl>,
## #   pct_wins_x2009.pct <dbl>, pct_wins_x2008.pct <dbl>, …
```

1.3 Quality control (15 points)

mlb_aggregate_computed tibble

Code

```
## # A tibble: 30 × 5
##    team                 payroll_aggregate pct_wins_aggregate payroll_a…¹ pct_w…²
##    <fct>                            <dbl>              <dbl>       <dbl>   <dbl>
##  1 Arizona Diamondbacks             1.12               0.492       1223.   0.492
##  2 Atlanta Braves                   1.38               0.563       1518.   0.563
##  3 Baltimore Orioles                1.16               0.457       1305.   0.457
##  4 Boston Red Sox                   1.97               0.551       2104.   0.551
##  5 Chicago Cubs                     1.46               0.475       1552.   0.475
##  6 Chicago White Sox                1.32               0.507       1375.   0.507
##  7 Cincinnati Reds                  1.02               0.491       1119.   0.491
##  8 Cleveland Indians                0.999              0.505       1113.   0.505
##  9 Colorado Rockies                 1.03               0.463       1129.   0.463
## 10 Detroit Tigers                   1.43               0.474       1484.   0.474
## # … with 20 more rows, and abbreviated variable names
## #   ¹payroll_aggregate_computed, ²pct_wins_aggregate_computed
```

Create Scatter Plots

Code

```
## Warning: package 'gridExtra' was built under R version 4.2.3
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

Code

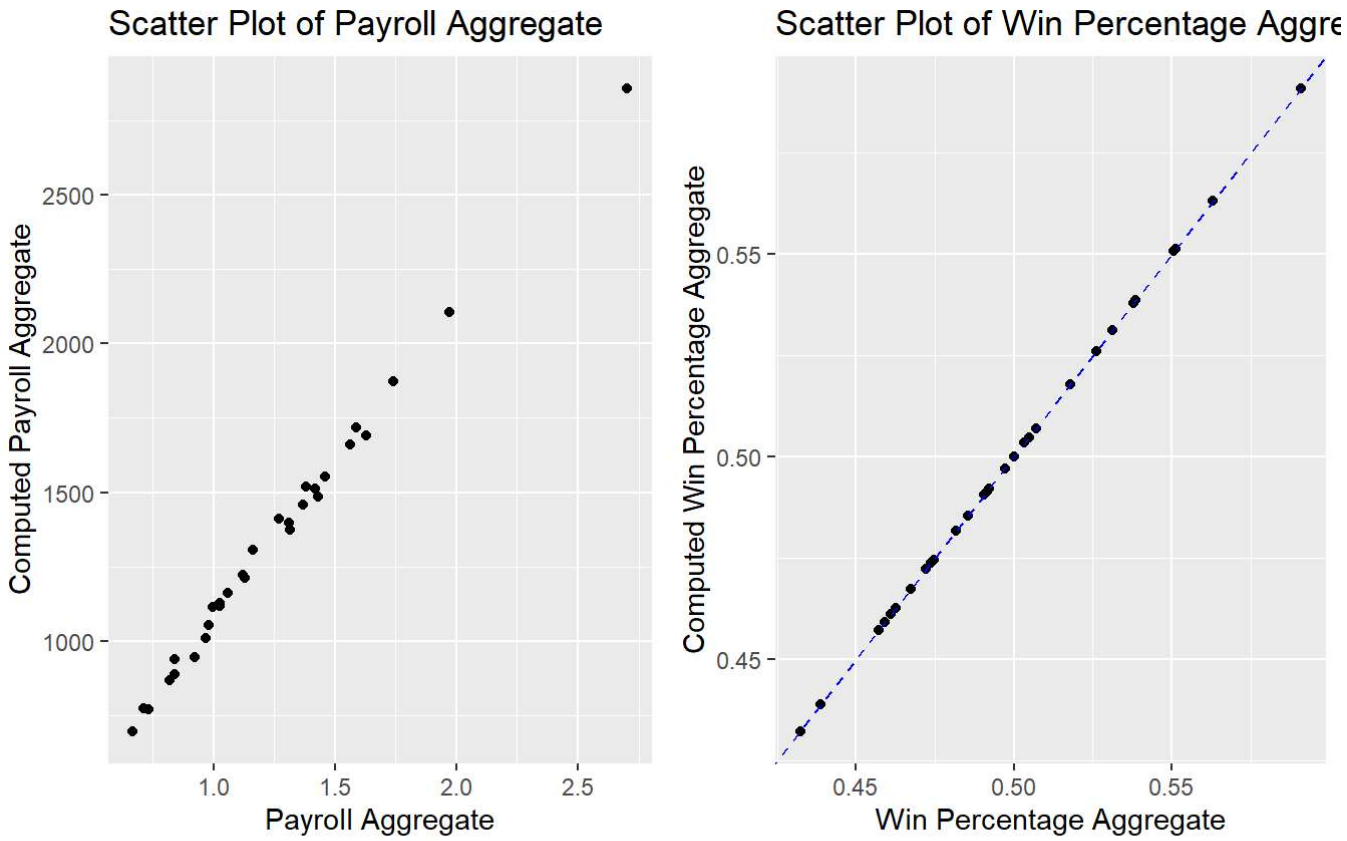


Figure 1.1

Using function ggplot, getting the plot of **payroll_aggregate_computed versus payroll_aggregate** and **pct_wins_aggregate_computed versus pct_wins_aggregate** , we have concluded the plotting shows a linear model which pertains into a proportion increment of data. This will also that the data is close to similar to each other as desired.

2 Explore (50 points for correctness; 10 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.

Code

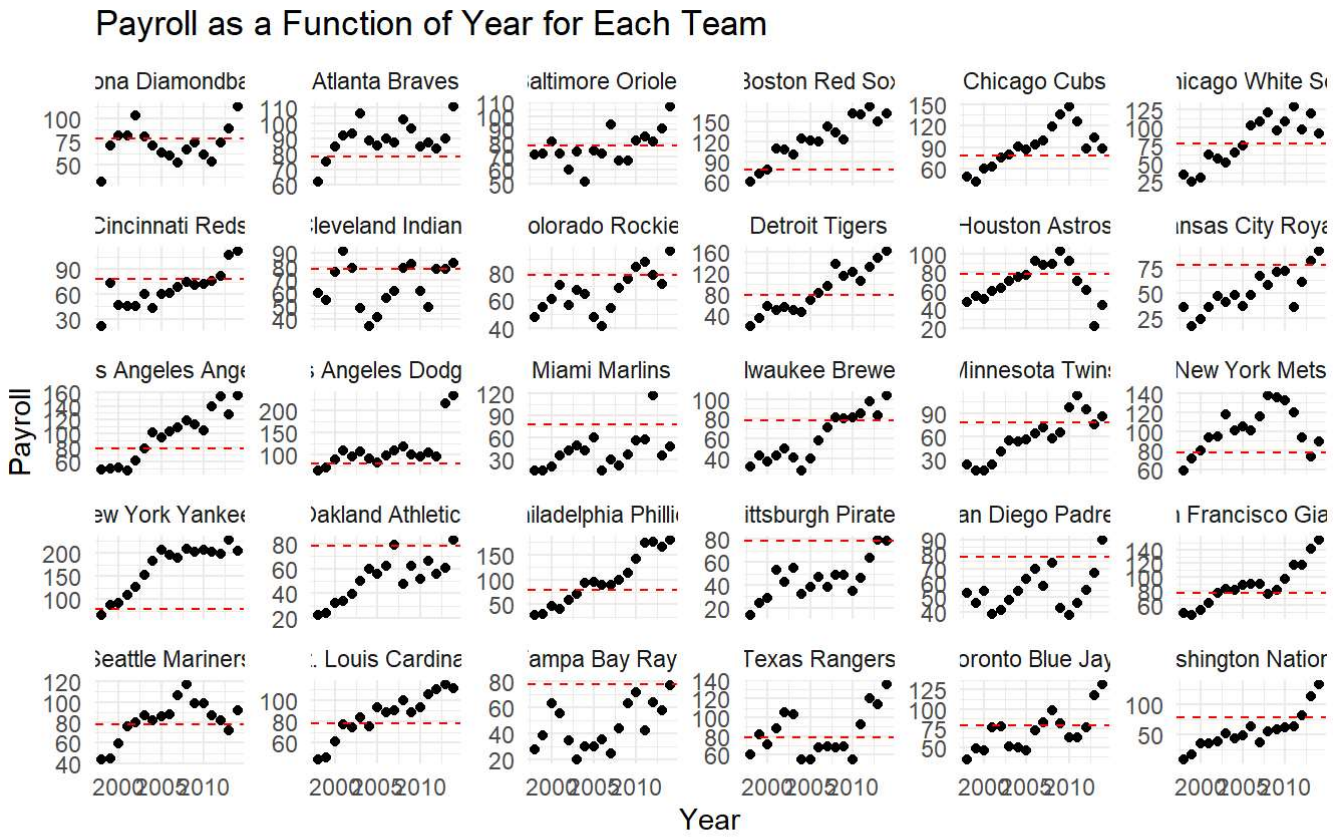


Figure 1.2

Using function ggplot and gather, getting the plot of payroll as a function of year for each of the 30 teams. We can now analyze together the plot of payroll of each team over the years of 1998 to 2014. We can also say that the mean of the payroll of each team are not equal together as the red horizontal line varied for 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.

Code

```
## # A tibble: 3 × 2
##   team                payroll_aggregate_computed
##   <chr>                                    <dbl>
## 1 Boston Red Sox                           2104.
## 2 Los Angeles Dodgers                      1874.
## 3 New York Yankees                         2857.
```

- 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.

Code

```
## # A tibble: 3 × 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.
```

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?

As mentioned in our initial plotting of payroll of each teams are distinct to each other as its horizontal line varied to each other, which implicates a different mean of payrolls.
The **Boston Red Sox, Los Angeles Dodgers, and New York Yankees** are identified using dplyr as the teams with the **highest payroll_aggregate_computed values**, characterized by high and varying payroll figures over the years.
While, The pct_increase metric shows the percentage increase in payroll from 1998 to 2014, with the **Washington Nationals, Detroit Tigers, and Philadelphia Phillies** showing **the greatest percentage increases** , indicating substantial growth in payroll over the analyzed period.

The plot shows how team payroll fluctuates over time, with teams with higher payroll aggregate values or significant pct_increase values easily identified. This visual representation complements quantitative analysis using dplyr, providing a more comprehensive understanding of payroll dynamics in Major League Baseball, as teams with wider spreads identified.
Additionally, we have shown that the top 3 of high payroll over the years (highes payroll_aggregate_computed) is different with top 3 of the greatest percentage increases of payroll, thus as assumed the an implication of different mean of payrolls.

2.2 Win percentage across years (15 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.

Code

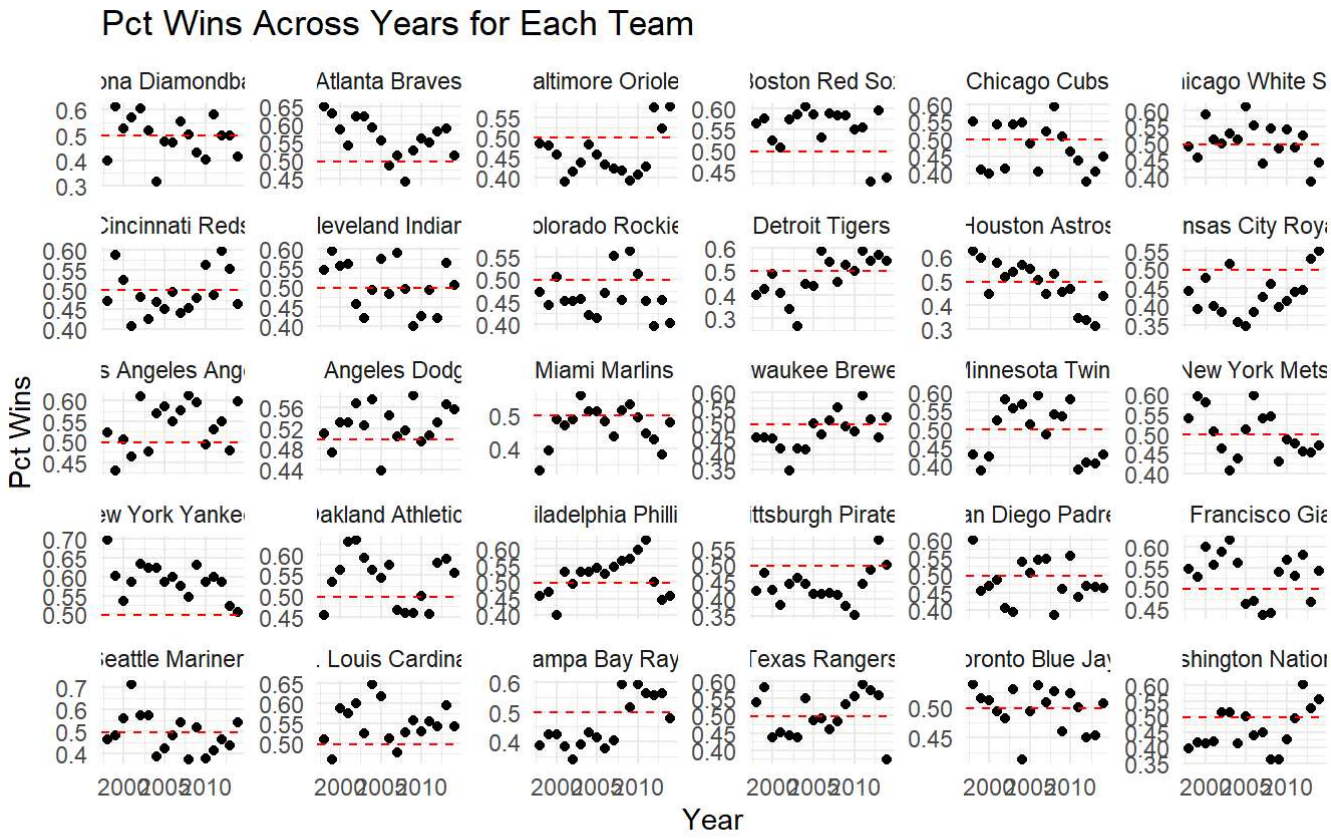


Figure 1.3

- Using dplyr, identify the three teams with the greatest pct_wins_aggregate_computed and print a table of these teams along with pct_wins_aggregate_computed.

Code

```
## # A tibble: 3 × 2
##   team             pct_wins_aggregate_computed
##   <chr>                                  <dbl>
## 1 Atlanta Braves                         0.563
## 2 Boston Red Sox                         0.551
## 3 New York Yankees                       0.591
```

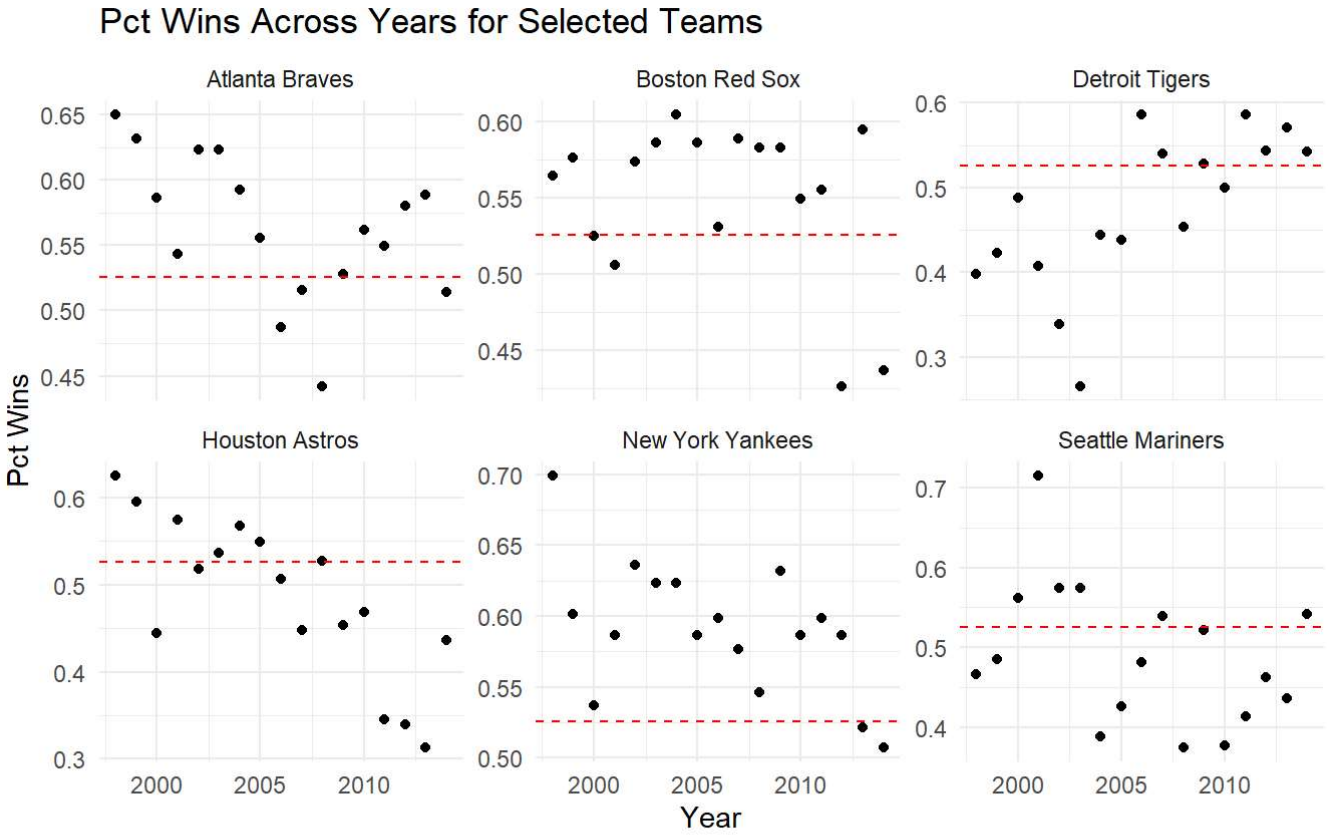
- Using dplyr, identify the three teams with the most erratic pct_wins across years(as measured by the standard deviation, call it pct_wins_sd) and print a table of these teams along with pct_wins_sd.

Code

```
## # A tibble: 3 × 2
##   team             pct_wins_sd
##   <fct>                  <dbl>
## 1 Detroit Tigers        0.0898
## 2 Houston Astros        0.0914
## 3 Seattle Mariners      0.0892
```

Re-PLOT of TOP TEAMS

Code



Plot of Top (pct_wins_sd) and (pct_wins_aggregate_computed)

How are the metrics pct_wins_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?

The **"Atlanta Braves", "Boston Red Sox", "New York Yankees"** are identified using dplyr as the teams with the **highest pct_wins_aggregate_computed values**, characterized by high and varying wins figures over the years.

While, The pct_wins_sd metric shows the standard deviation in percentage win from 1998 to 2014, with the **“Detroit Tigers”,“Houston Astros”, “Seattle Mariners”** showing **the greatest standard deviation in percentage win** .

As we plot the Top 3 teams of 2 different characteristic, we can see as shown that the plotting implicates a plotting **higher than 0.50**

. Thus, indeed we can see why the team top the rank.

2.3 Win percentage versus payroll (15 points)

- 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.

Code

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

```
## Warning: ggrepel: 7 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

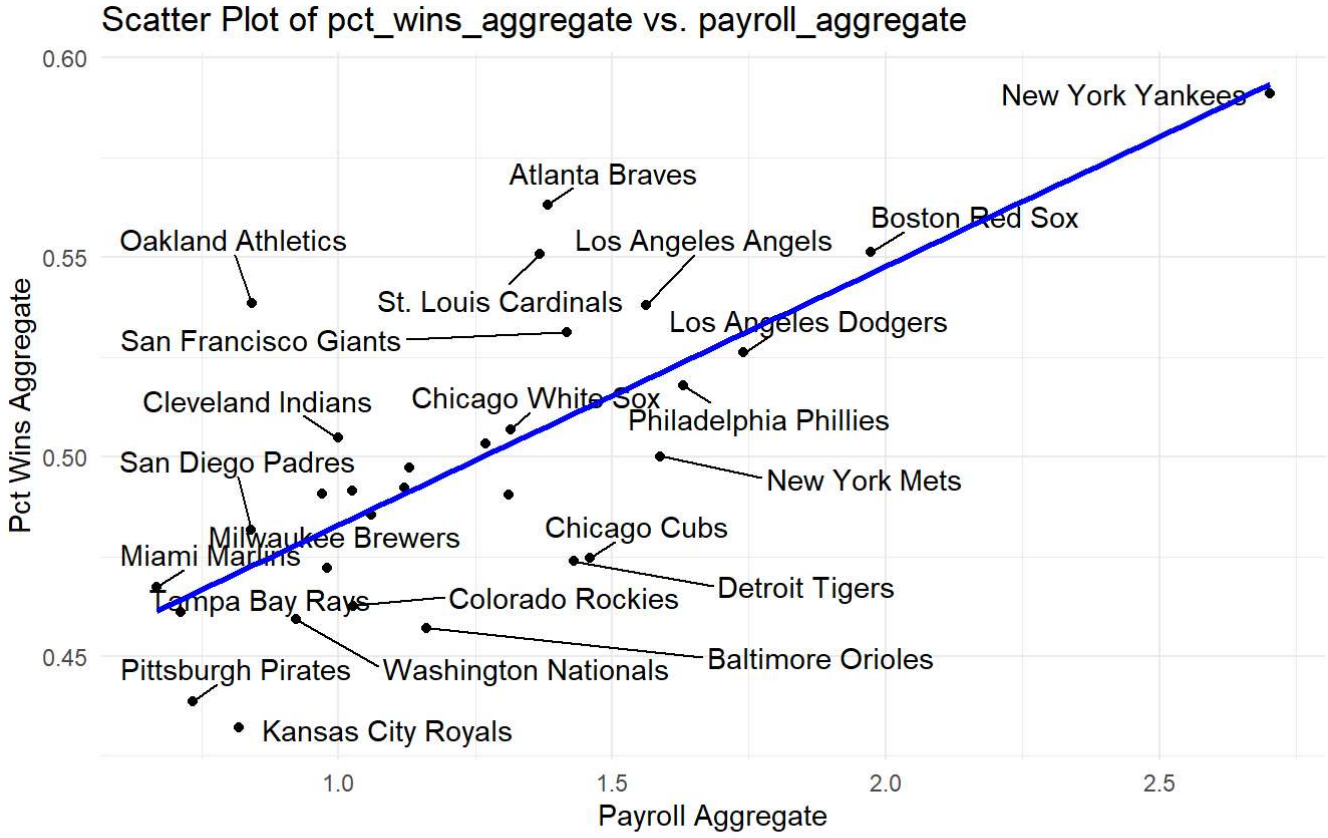


Figure 1.4

Isthe relationship between payroll and pct_wins positive or negative? Isthis what you would expect, and why?

The relationship between payroll and pct_wins is posssitive as shown in the plot.
We can see that the plotting in Figure 1.4 that the New York Yankees, Boston Red Sox, and Los Angeles Dodgers continuously increases as Payroll and Pct_Win proportionally increases. Just like what we have on the Top 3 in our **payroll_aggregate_computed** which are also New York Yankees, Boston Red Sox, and Los Angeles Dodgers.
However, we cannot really say that there's a relationship between with the payroll and pct_wins only if we use more tool like a Simple Linear Model, etc., to see if there is a relationship that would make the data indeed proportionally into each other.

2.4 Team efficiency (5 points)

- 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_computed and payroll_aggregate_computed.

Code

```
## # A tibble: 3 × 4
##   team              payroll_aggregate_computed pct_wins_aggregate_comp…¹ effic…²
##   <chr>                                  <dbl>                     <dbl>   <dbl>
## 1 Miami Marlins                           698.                    0.0275 3.94e-5
## 2 Oakland Athletics                       888.                    0.0317 3.57e-5
## 3 Tampa Bay Rays                          776.                    0.0271 3.49e-5
## # … with abbreviated variable names ¹pct_wins_aggregate_computed, ²efficiency
```

In what sense do these three teams appear efficient in the previous plot?

In accordance with our previous plot Figure 1.4, to say that a team is efficient, the quotient of pct_wins_aggregate_computed divided by payroll_aggregate_computed would be big. In other words, their payroll might not be big however, the percentage of them winning is great.
As seen in Figure 1.4 again, the most obvious team in the plot is **Oakland Athletics** as the x-axis (Payroll) of team may be low, however the y-axis (Percentafe Wins) is high. In which can appear as well into our Top Three Teams which are Miami Marlins, Oakland Athletics, and lastly, Tampa Bay Rays.