Lab 05 - MLB Wins

Due: Thursday, Feb 27 at 11:59pm

Team gitData - Nathan Kim, Avanti Shah, Blossom Mojekwu

Packages

```
library(tidyverse)
library(ggpol)
library(broom)
```

Data

```
teams_default <- read_csv("data/teams.csv")
```

Tasks

Task 1

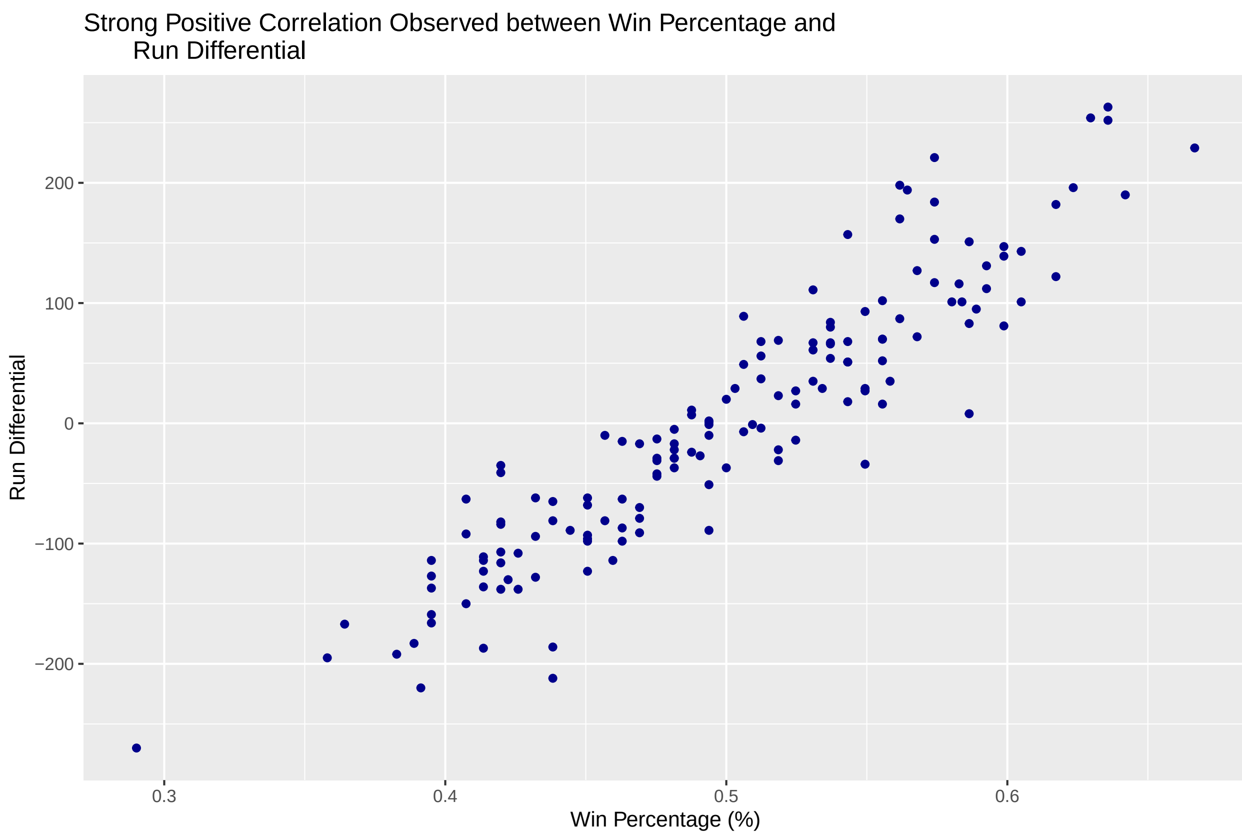
```
teams <- teams_default %>%
  mutate(win_pct = w / g) %>%
  mutate(rd = r - ra) %>%
  mutate(hd = h - ha) %>%
  mutate(bbd = bb - bba) %>%
  mutate(sod = so - soa)
teams
```

```
# A tibble: 150 x 45
   name  franch_id year_id lg_id div_id  rank     g     w     l div_win wc_win
   <chr> <chr>       <dbl> <chr> <chr>  <dbl> <dbl> <dbl> <dbl> <chr>   <chr>
 1 Ariz~ ARI          2014 NL    W          5   162    64    98 N       N
 2 Atla~ ATL          2014 NL    E          2   162    79    83 N       N
 3 Balt~ BAL          2014 AL    E          1   162    96    66 Y       N
 4 Bost~ BOS          2014 AL    E          5   162    71    91 N       N
 5 Chic~ CHW          2014 AL    C          4   162    73    89 N       N
 6 Chic~ CHC          2014 NL    C          5   162    73    89 N       N
 7 Cinc~ CIN          2014 NL    C          4   162    76    86 N       N
 8 Clev~ CLE          2014 AL    C          3   162    85    77 N       N
 9 Colo~ COL          2014 NL    W          4   162    66    96 N       N
10 Detr~ DET          2014 AL    C          1   162    90    72 Y       N
# ... with 140 more rows, and 34 more variables: lg_win <chr>, ws_win <chr>,
#   r <dbl>, ab <dbl>, h <dbl>, x2b <dbl>, x3b <dbl>, hr <dbl>, bb <dbl>,
#   so <dbl>, sb <dbl>, cs <dbl>, hbp <dbl>, sf <dbl>, ra <dbl>, er <dbl>,
#   era <dbl>, cg <dbl>, sho <dbl>, sv <dbl>, i_pouts <dbl>, ha <dbl>,
#   hra <dbl>, bba <dbl>, soa <dbl>, e <dbl>, dp <dbl>, fp <dbl>,
```

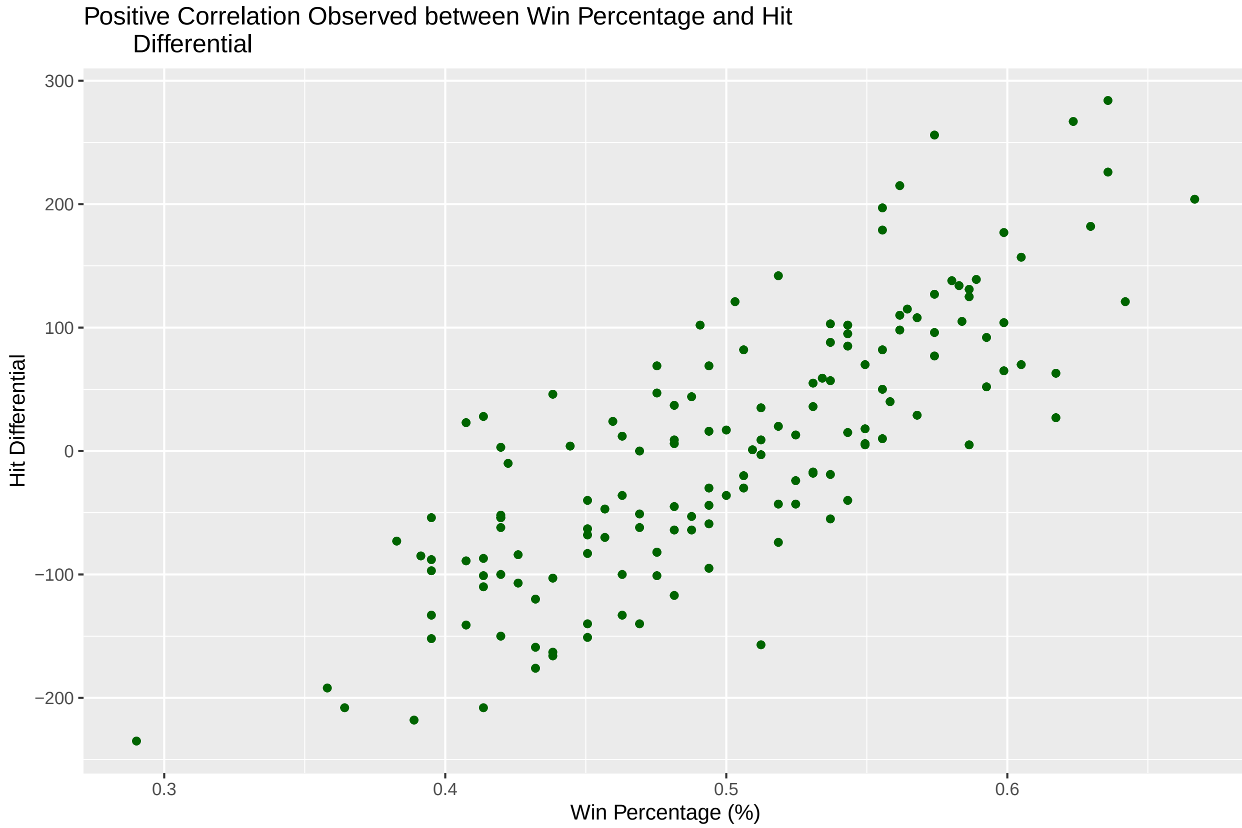
```
#   attendance <dbl>, win_pct <dbl>, rd <dbl>, hd <dbl>, bbd <dbl>, sod <dbl>
```

Task 2 - Elaborate Upon Narrative

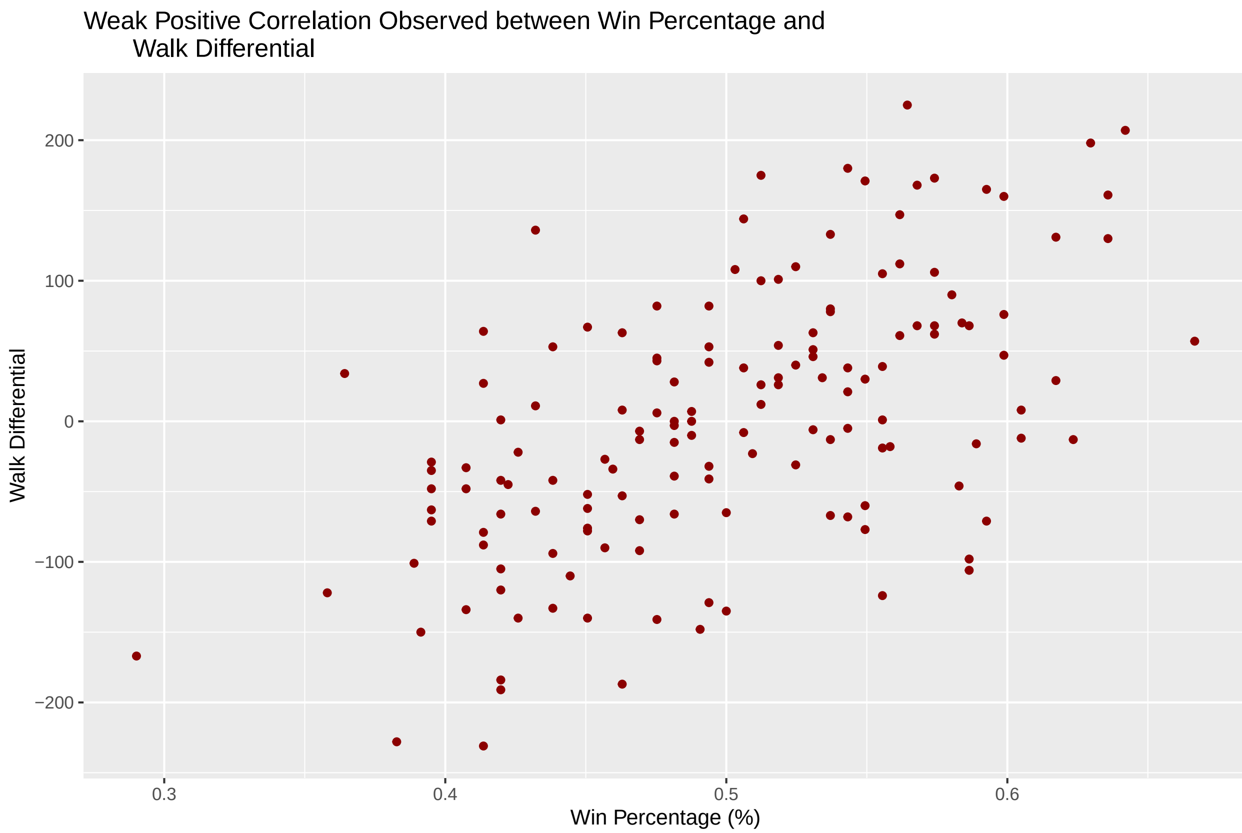
```r
ggplot(data = teams, mapping = aes(x = win_pct, y = rd)) +
  geom_point(color = "dark blue") +
  labs(title = "Strong Positive Correlation Observed between Win Percentage and
       Run Differential", x = "Win Percentage (%)", y = "Run Differential")
```



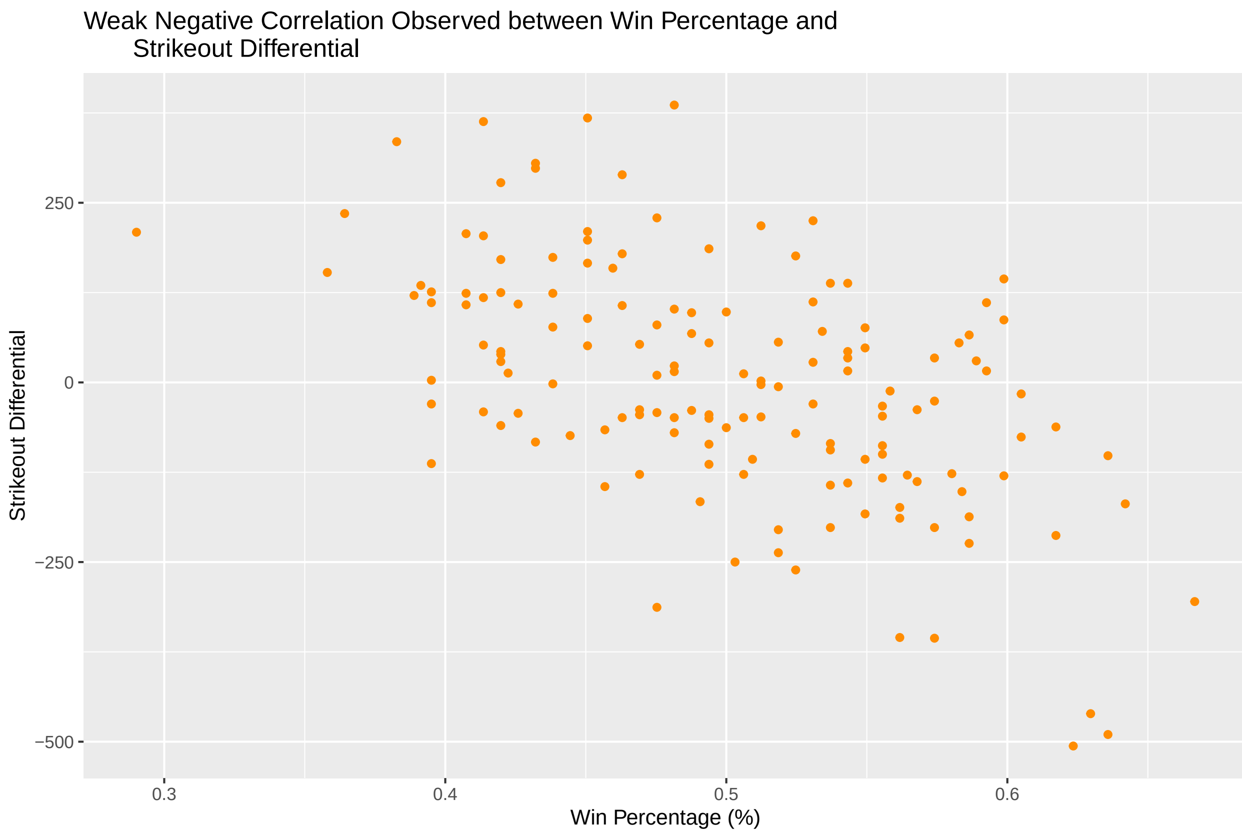

```r
ggplot(data = teams, mapping = aes(x = win_pct, y = hd)) +
  geom_point(color = "dark green") +
  labs(title = "Positive Correlation Observed between Win Percentage and Hit
       Differential", x = "Win Percentage (%)", y = "Hit Differential")
```




```
ggplot(data = teams, mapping = aes(x = win_pct, y = bbd)) +
  geom_point(color = "dark red") +
  labs(title = "Weak Positive Correlation Observed between Win Percentage and
      Walk Differential", x = "Win Percentage (%)", y = "Walk Differential")
```




```r
ggplot(data = teams, mapping = aes(x = win_pct, y = sod)) +
  geom_point(color = "dark orange") +
  labs(title = "Weak Negative Correlation Observed between Win Percentage and
       Strikeout Differential", x = "Win Percentage (%)",
       y = "Strikeout Differential")
```



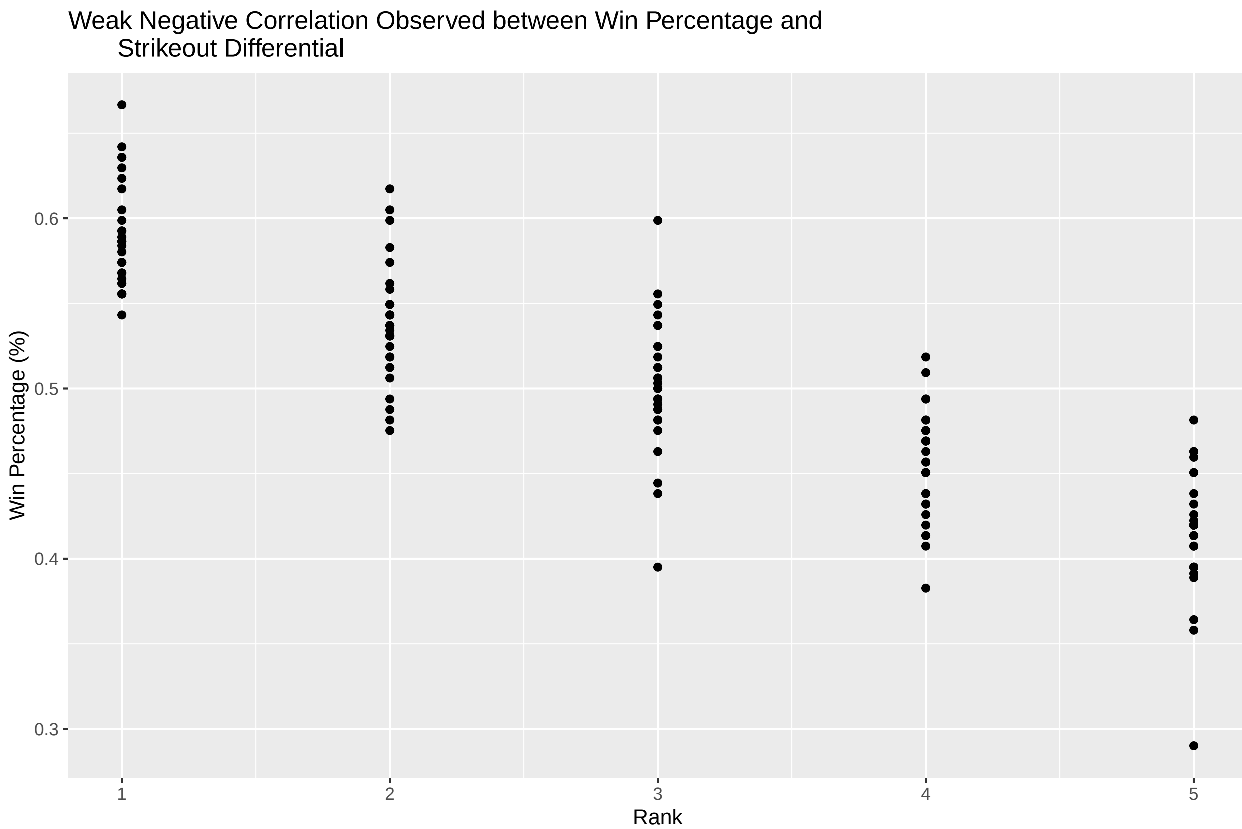

```
teams %>%
  select(win_pct, rd, hd, bbd, sod) %>%
  cor()
```

```
           win_pct         rd         hd        bbd        sod
win_pct  1.0000000  0.9268104  0.8031713  0.5752652 -0.5534366
rd       0.9268104  1.0000000  0.8449338  0.6649954 -0.5693114
hd       0.8031713  0.8449338  1.0000000  0.3616847 -0.6223871
bbd      0.5752652  0.6649954  0.3616847  1.0000000 -0.3546139
sod     -0.5534366 -0.5693114 -0.6223871 -0.3546139  1.0000000
```

Here, it seems as though rd (run differential) has the strongest correlation with win percentage. Sod, or strike out differential, has the weakest correlation with win percentage. Finally, walk differential and hit differential are both positively correlated with win percentage.

Task 3

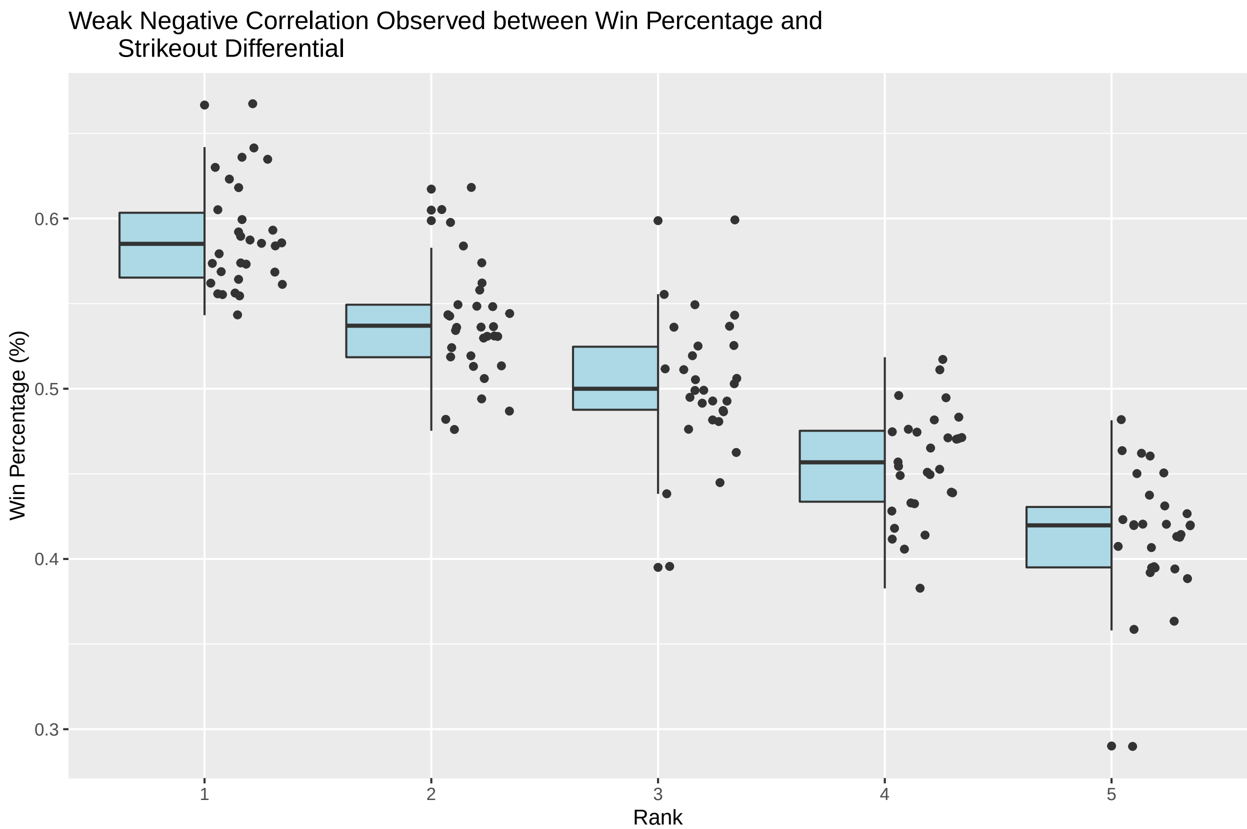
```
ggplot(data = teams, mapping = aes(x = rank, y = win_pct)) +
  geom_point() +
  labs(title = "Weak Negative Correlation Observed between Win Percentage and
       Strikeout Differential", x = "Rank", y = "Win Percentage (%)")
```



From this visualization, we can tell that teams with higher ranks tend to have higher win percentages, which can be explained by the declining relationship between the center-points vertical set of points. However, we cannot observe the actual spread of the data points, the median, or the quartiles through this plot.

Task 4

```
ggplot(data = teams, mapping = aes(x = factor(rank), y = win_pct)) +
  geom_boxjitter(fill = "light blue") +
  labs(title = "Weak Negative Correlation Observed between Win Percentage and
      Strikeout Differential", x = "Rank", y = "Win Percentage (%)")
```



The jittered points in the boxjitter plot spread out the individual data points in order to make them more easily viewable by the reader, hence gauging the spread of the points more intuitively. The box portion of the boxjitter plot shows the median and the two quartiles (upper and lower quartiles) of the win percentages for each rank. The points outside of the vertical lines (for each rank) are outliers. Overall, this shows much more detailed information compared to the previous graph and hence is preferred.

Task 5

```
lm_rd <- lm(win_pct ~ rd, data = teams)

lm_rd %>%
  tidy() %>%
  mutate(estimate = round(estimate, 5)) %>%
  select(term, estimate)
```

```
# A tibble: 2 x 2
  term        estimate
  <chr>          <dbl>
1 (Intercept)  0.500
2 rd           0.00059
```

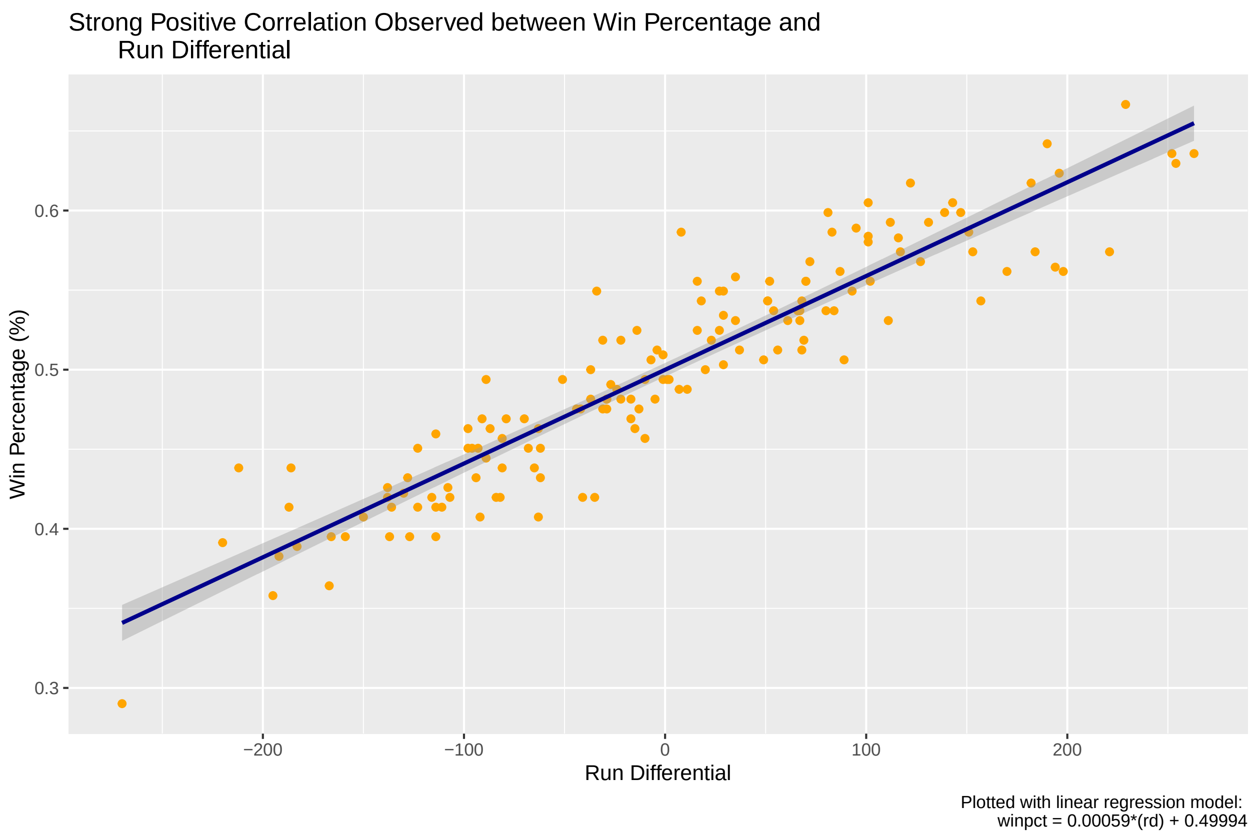
where `b0 = 0.49994`; `b1 = 0.00059`

The linear model can be written out as:

$win\widehat{Percentage} = 0.49994 + 0.00059*(\text{rd})$

Task 6

```r
ggplot(data = teams, mapping = aes(x = rd, y = win_pct)) +
  geom_point(color = "orange") +
  geom_smooth(method = "lm", color = "dark blue") +
  labs(title = "Strong Positive Correlation Observed between Win Percentage and
       Run Differential",
       caption = "Plotted with linear regression model:
       winpct = 0.00059*(rd) + 0.49994",
       x = "Run Differential", y = "Win Percentage (%)")
```



From this visualization, we can see that run differential and win percentage have a positive relationship. The regression line has a medium-positive slope and a positive intercept, with a few outliers, and there seem to be a balanced number of points on either side of the regression line. Hence, we can see that the regression line fits the data quite well - it provides a good visualization of the trend between the variables and allows for future prediction and extrapolation.

Task 7

The slope of this linear model is `0.00059`, which indicates a weak positive relationship between the variables, which makes sense with regards to the data considering the scale of the axes (win percentage has small increments but run differential has larger increments). As a result, although it appears that the relationship is strong, it follows a weak positive linear trend.

The intercept is `0.49994`, which means that when run differential = 0 (i.e. the team allows as many runs as it scores), their win percentage is roughly half (50%). This means that a team with a run differential of 0 won half of all past games played and lost half of all past games played, meaning they had an equally likely chance of winning and losing a given game. This makes sense with regards to the data as we can assume that there was an equal amount of other teams in the league with run differentials higher than 0 and lower than 0, so every team with rd = `0` were equally likely to win and lose.

Task 8

```
glance(lm_rd) %>%
  select(r.squared)
```

```
# A tibble: 1 x 1
  r.squared
      <dbl>
1     0.859
```

The strength of the fit of a linear model is commonly evaluated using `R`-squared. This result shows us that roughly **85.9%** of the variability in win percentages of included teams can be explained by their run differentials. This tells us that the remainder of the variability (approximately **14.1%**) is explained by variables not included in the model. This is plausible as we are observing the effect of run differentials on win percentages, so the higher the run differential (more runs scored and fewer runs allowed), the more likely a team is to have a higher win percentage (with other factors assumed to be constant). As a result, we can say that the run differential of a team strongly impacts its win percentage.

Task 9

```
lm_sod <- lm(win_pct ~ sod, data = teams)

lm_sod %>%
  tidy() %>%
  mutate(estimate = round(estimate, 5)) %>%
  select(term, estimate)
```

```
# A tibble: 2 x 2
  term        estimate
  <chr>          <dbl>
1 (Intercept)  0.500
2 sod         -0.00024
```

where `b0 = 0.49994`; `b1 = -0.00024`

The linear model can be written out as:

$\widehat{winPercentage}$ = 0.49994 + -0.00024*(sod)

Task 10

```
newyorkMets <- tibble(Team = "New York Mets", R = 791, Ra = 737, SO = 1384,
                      SOa = 1520, W = 86, L = 76)

newyorkMets %>%
  mutate(rd = R - Ra) %>%
  mutate(sod = SO - SOa) %>%
  mutate(win_pct = W/(W+L))
```

```
# A tibble: 1 x 10
  Team               R    Ra    SO   SOa     W     L    rd   sod win_pct
  <chr>          <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
1 New York Mets    791   737  1384  1520    86    76    54  -136   0.531
```

Chosen MLB Team: **New York Mets**

2019 Run Differential: `54`

(`lm_rd`) 2019 Predicted winPercentage = `0.49994 + 0.00059*(54) = 0.5318`

Actual 2019 winPercentage = `0.5309`

2019 Strikeout Differential: `-136`

(`lm_sod`) Predicted 2019 winPercentage = `0.49994 + -0.00024*(-136) = 0.53258`

Actual 2019 winPercentage = `0.5309`

The run differential model, lm_rd, is better at predicting the New York Mets actual 2019 win percentage. The win percentage predicted by the lm_rd model was 0.5318 while the win percentage predicted by the lm-sod model was 0.53258. Thus, the win percentage produced by the run differential model was closer to the actual 2019 win percentage at 0.5309.

lm_rd model - actual = 0.5318 - 0.5309 = 0.0009

lm_sod model - actual = 0.53258 - 0.5309 = 0.00168

Task 11

```r
lm_rank <- lm(win_pct ~ factor(rank), data = teams)

lm_rank %>%
  tidy() %>%
  mutate(estimate = round(estimate, 5)) %>%
  select(term, estimate)
```

```
# A tibble: 5 x 2
  term           estimate
  <chr>             <dbl>
1 (Intercept)      0.589
2 factor(rank)2   -0.0509
3 factor(rank)3   -0.0869
4 factor(rank)4   -0.133
5 factor(rank)5   -0.174
```

where `b0 = 0.58877`; `b1 = -0.05086`; `b2 = -0.08686`; `b3 = -0.13332`; `b4 = -0.17431`

The linear model can be written out as:

$\widehat{winPercentage} =$ `0.58877 - 0.05086*(factor(rank)2) - 0.08686*(factor(rank)3) - 0.13332*(factor(rank)4) - 0.17431*(factor(rank)5)`

Task 12

The intercept is `0.58877`. The intercept means that when the rank = 1, the win percentage will be roughly 58.88%. The better ranked a team is, the closer the win percentage gets to the intercept.

-0.05086 is the coefficient for factor(rank)2 which means that when rank 2 is compared to the baseline, rank 1, the win percentage is expected to be lower, on average, by 5.09 percent.

-0.08686 represents the decrease in win percentage by about 8.67% when rank 3 is compared to rank 1.

-0.13332 represents the decrease in win percentage by about 13.33% when rank 4 is compared to rank 1.

-0.17431 means that the win percentage decreases by an average of 17.43% when rank 5 is compared to rank 1.

Task 13

```r
lm_rank_base5 <- lm(win_pct ~ fct_relevel(factor(rank), "5"), data = teams)

lm_rank_base5 %>%
  tidy() %>%
  mutate(estimate = round(estimate, 5)) %>%
  select(term, estimate)
```

```
# A tibble: 5 x 2
  term                                 estimate
  <chr>                                   <dbl>
1 "(Intercept)"                          0.414
2 "fct_relevel(factor(rank), \"5\")1"    0.174
3 "fct_relevel(factor(rank), \"5\")2"    0.123
4 "fct_relevel(factor(rank), \"5\")3"    0.0874
5 "fct_relevel(factor(rank), \"5\")4"    0.041
```

The coefficients of this model are all positive rather than negative as in Task 11. The absolute value of the coefficients would be similar if the estimates in the Task 11 summary table was in reverse order. The coefficients in lm_rank has the baseline set as 1 and compares all of the ranks greater than 1 to rank 1. The coefficients in lm_rank_base5 sets the baseline as 5 and compares all of the ranks less than 5 to rank 5. This is why when rank 1 is compared to rank 5, there is an average increase of 17.43% to the win percentage. As the ranks get higher and closer to 5, the estimates decrease because a bigger rank will not increase the win percentage as much.

Task 14

```r
glance(lm_rank) %>%
  select(r.squared)
```

```
# A tibble: 1 x 1
  r.squared
      <dbl>
1     0.765
```

```r
glance(lm_rank_base5) %>%
  select(r.squared)
```

```
# A tibble: 1 x 1
  r.squared
      <dbl>
1     0.765
```

I would expect the R^2 for the lm_rank and lm_rank_base5 model to be similar and as high as 0.7651. The R^2 for models lm_rank and lm_rank_base5 means that roughly 76.5% of the variability in win percentage can be explained by rank, specifically rank comparisons where the baseline is set to rank 1 or rank 5. Rank is an important factor in determining win percentage because to formulate rank, a team's performance is considered.