College Football: Predicting Success

Gloria, Tyler, and Ethan

Overview

Question

“How can different factors of a teams performance be used to predict the overall success in a given season?”

Summary of Findings

We found the question interesting for us because our goal is discovering various possibilities that makes team winning and losing and use the knowledge to determine possible predictions outcome of it.

The data set is a college football data set from 2018 and 2019. Our data set came from Kaggle and contains data for 130 division 1 football programs and has data about each teams specific offensive, defensive and special team performance and rank.

We are going to use 2018 as our train or main data and the 2019 data as our test data. and looking at how different aspects of a teams performance (total yard, yards given up per game, rush/pass yards, etc.) contribute to a teams overall wins and losses.

Prior Work

Summary and critique of prior work

Author Name: Samuel Tan

Project Website: https://www.kaggle.com/stmy456/college-football-analysis (https://www.kaggle.com/stmy456/college-football-analysis)

Data Website: https://www.kaggle.com/jeffgallini/college-football-team-stats-2019?select=cfb19.csv (https://www.kaggle.com/jeffgallini/college-football-team-stats-2019?select=cfb19.csv)

We found the data on Kaggle while searching for data that encompasses college football. The data was pulled from the NCAA website and used by multiple people to do their own analysis of the data. The author of the project that we are looking at wanted to look at how offensive and defensive metrics correlated to on field success, something that we want to look at as well. The author found some interesting information regarding how some defensive metrics predicted decently well if a team will win or loss. Interceptions on the opposing quarterback were the most “critical individual play in determining win percentage” according to the author.

The author used some different kind tools that we have not used and did not do some things that we did to make it a little easier to comprehend. He did not have a conclussion stating what actually was a good predictor and what where not.

How this project extends prior work

When working on this project we want to use similar offensive and defensive metrics to see what will help us predict win percentage of a team. We want to actually understand what are good indicators and what are not good indicators of this. It will hopefully be clear to us by the end of the project which kind of metrics are good predictors of winning percentage in college football. We are going to use two years of data to make sure that our data is models work not only on the year of the data but it works on other years of data as well.

Approach

Problem Description

Which factors of a teams performance correlate can predict wins and losses the best?

We are going to approach this first by looking at some simple graphs looking at one factor being compared for all teams. We can look for patterns in this graphs to give us a good idea which stats may be more or less helpful in actual predictions. We got the data from Kaggle, but the data itself came from the NCAA. Sports are a great way to look at and make comparisons. As these games were happening, the stats after each week were most likely compiled into a datasheet and then posted on the NCAA website after the conclusion of each season.

Data

```
cfb18_data <- read.csv("cfb18-1.csv")
cfb19_data <- read.csv("cfb19-1.csv")
```

Provenance

The data originally came from NCAA stats but it does not have all the team information and uses many acronyms that are obscure. With the data available, the author, Jeff Gallini scraped the team statistics for each college football season from 2013 to the present and published it on Kaggle. We found and downloaded the data from Kaggle.

Structure

There is a total of 130 rows each representing a different individual team in alphabetical order. It also specifies which conference the team comes from. For example, the first row Air Force (Mountain West) telling us the team and conference. For each individual team, there are 152 different columns containing different stats about that team. It starts with games played, wins and losses and then goes into specific performance part of a team. There are rankings (defensive, offensive, etc), average yards per game, touchdowns for the whole year, total plays, and many more that all tell a part of a teams overall success. Its obvious to us at this point that we won't be needing each columns data, because something like average kickoff return yards won't be something that has a huge effect on a teams chances to win/lose.

Appropriateness for task

Overall, our data set contains a very large amount of data which is a good thing. We will have plenty of variables to look at and consider as we try and make predictions. One thing that would be nice is if the team and conference were separate columns so we could compare smaller groups of teams that are on the same scale. For example, comparing an SEC team to a MAC team isn't the best way to decide which team is better. We may try and find a way to separate teams by conference.

Exploratory Data Analysis

```
cfb18_sep <- cfb18_data  %>%
  separate(col = Team, into = c("Team","Conference"), sep = "\\(") %>%
  separate(col = Conference, into = c("Conference"), sep = "\\)")
```

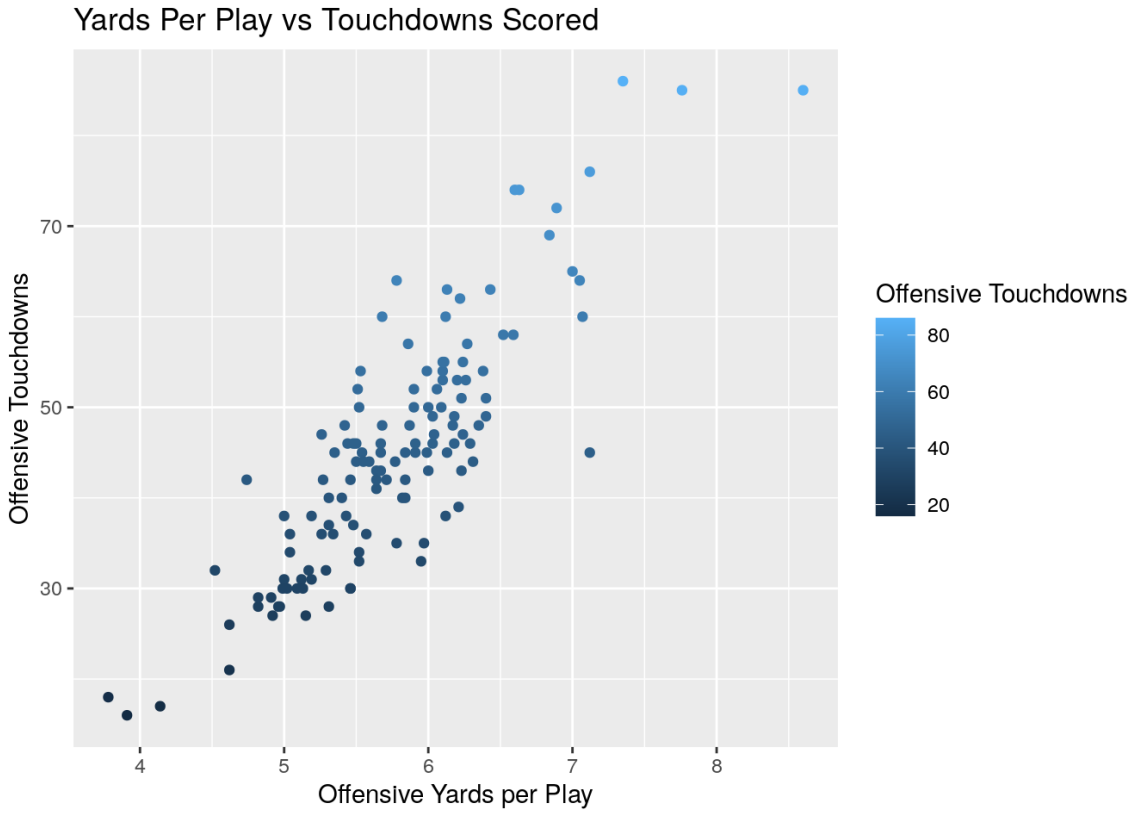
```
Warning: Expected 1 pieces. Additional pieces discarded in 129 rows [1, 2, 3, 4,
5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
cfb19_sep <- cfb19_data  %>%
  separate(col = Team, into = c("Team","Conference"), sep = "\\(") %>%
  separate(col = Conference, into = c("Conference"), sep = "\\)")
```

```
Warning: Expected 1 pieces. Additional pieces discarded in 130 rows [1, 2, 3, 4,
5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

Exploratory Plot 1

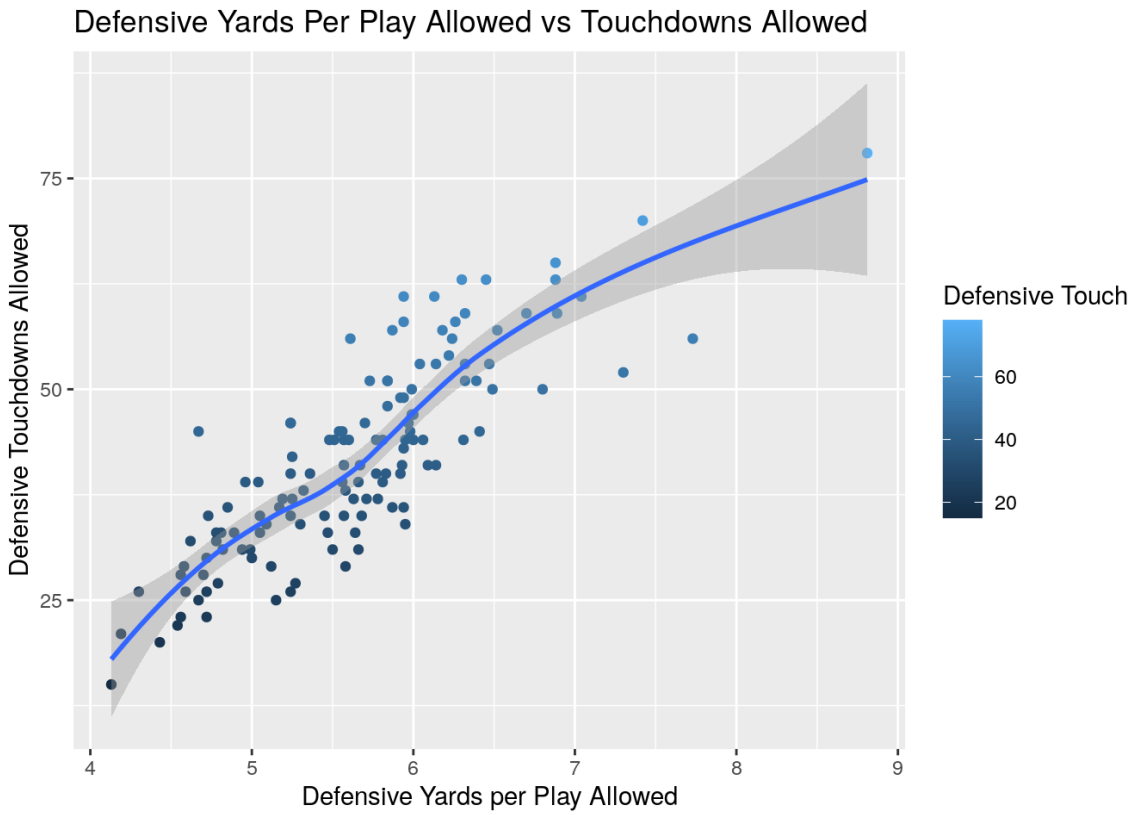
```
cfb18_sep %>%
  ggplot(aes(x = Off.Yards.Play, y = Off.TDs, color = Off.TDs)) +
  labs(y = "Offensive Touchdowns",
       x = "Offensive Yards per Play",
       color = "Offensive Touchdowns",
       title = "Yards Per Play vs Touchdowns Scored") +
  geom_point()
```



Exploratory Plot 2

```
cfb18_sep %>%
  ggplot(aes(x = Yards.Play.Allowed, y = Off.TDs.Allowed, color = Off.TDs.Allowed)) +
  labs(y = "Defensive Touchdowns Allowed",
       x = "Defensive Yards per Play Allowed",
       color = "Defensive Touch",
       title = "Defensive Yards Per Play Allowed vs Touchdowns Allowed") +
  geom_point() +
  geom_smooth()
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

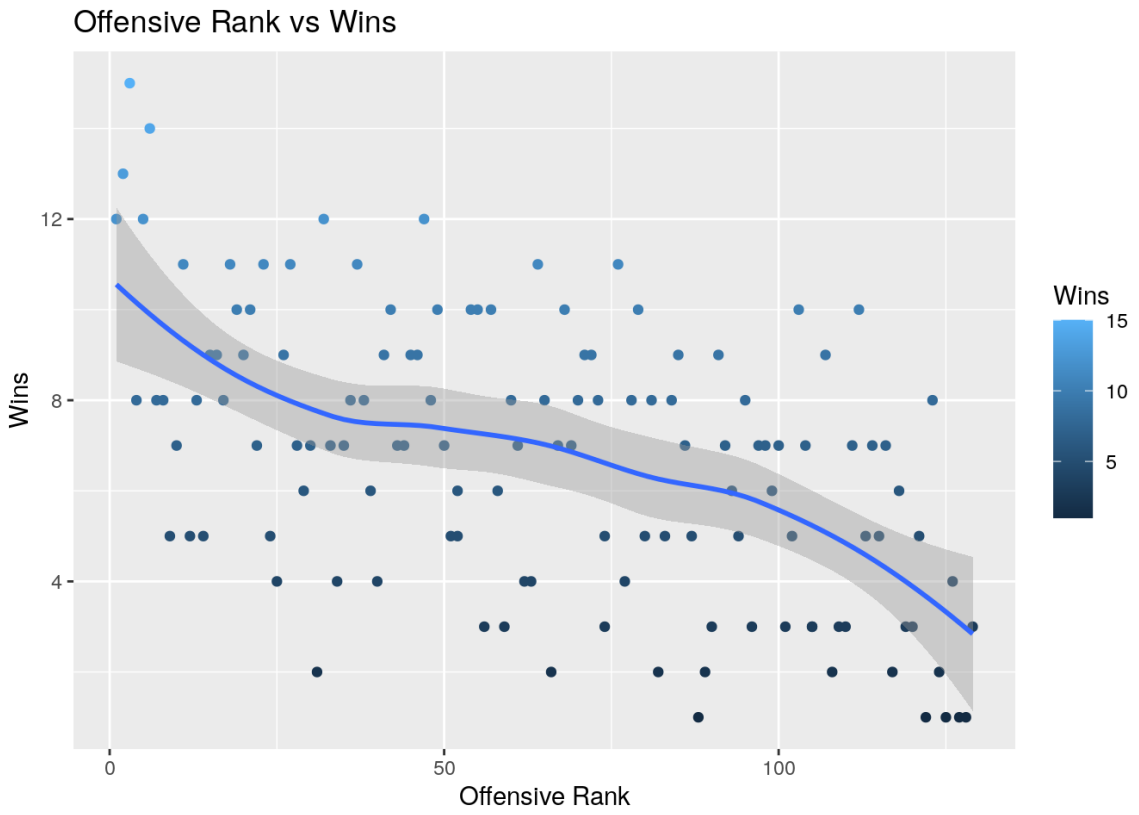


When looking at yards per play allowed and touchdowns allowed it makes sense that defenses that allow shorter plays are also likely to allow fewer touchdowns.

Exploratory Plot 3

```
cfb18_sep %>%
  ggplot(aes(x = Off.Rank, y = Win, color = Win)) +
  labs(y = "Wins",
       x = "Offensive Rank",
       color = "Wins",
       title = "Offensive Rank vs Wins") +
  geom_point() +
  geom_smooth()
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

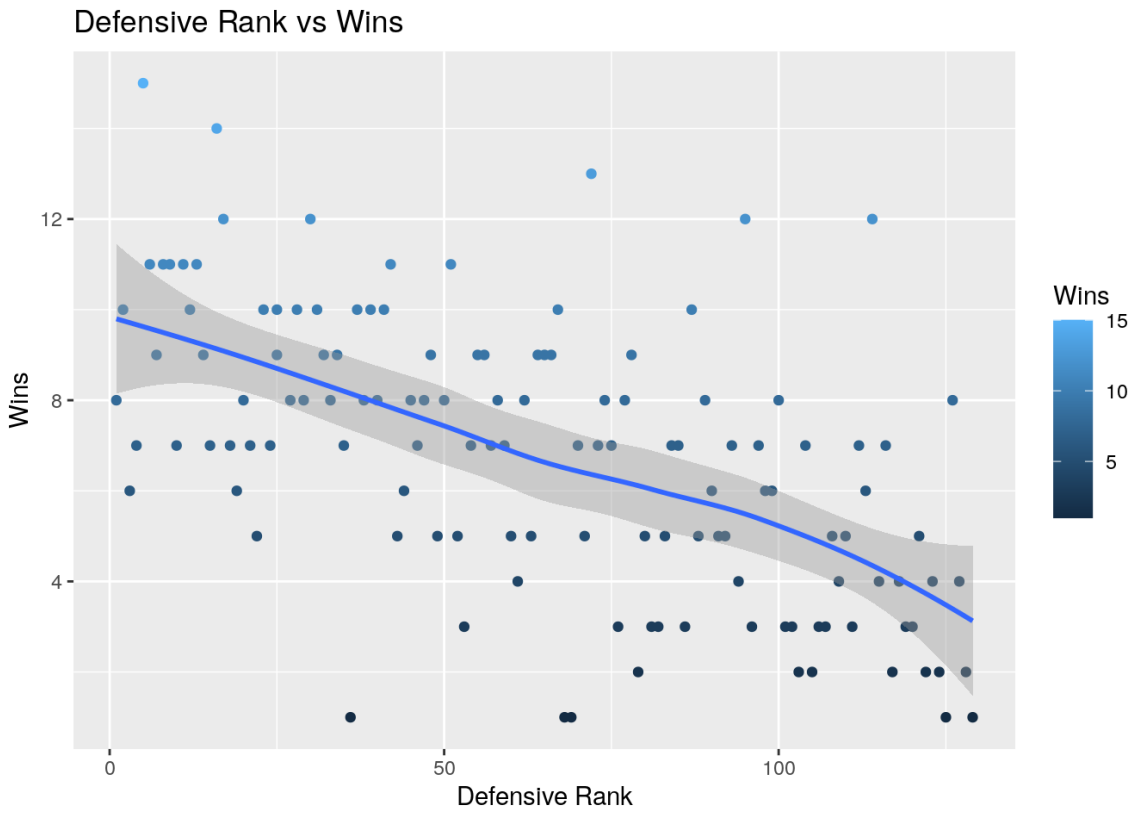


If a team has a high offensive rank, they have more wins than the ones on the bottom rank. High offensive rank means that they are strong in scoring and offense part which means they are more likely to win.

Exploratory Plot 4

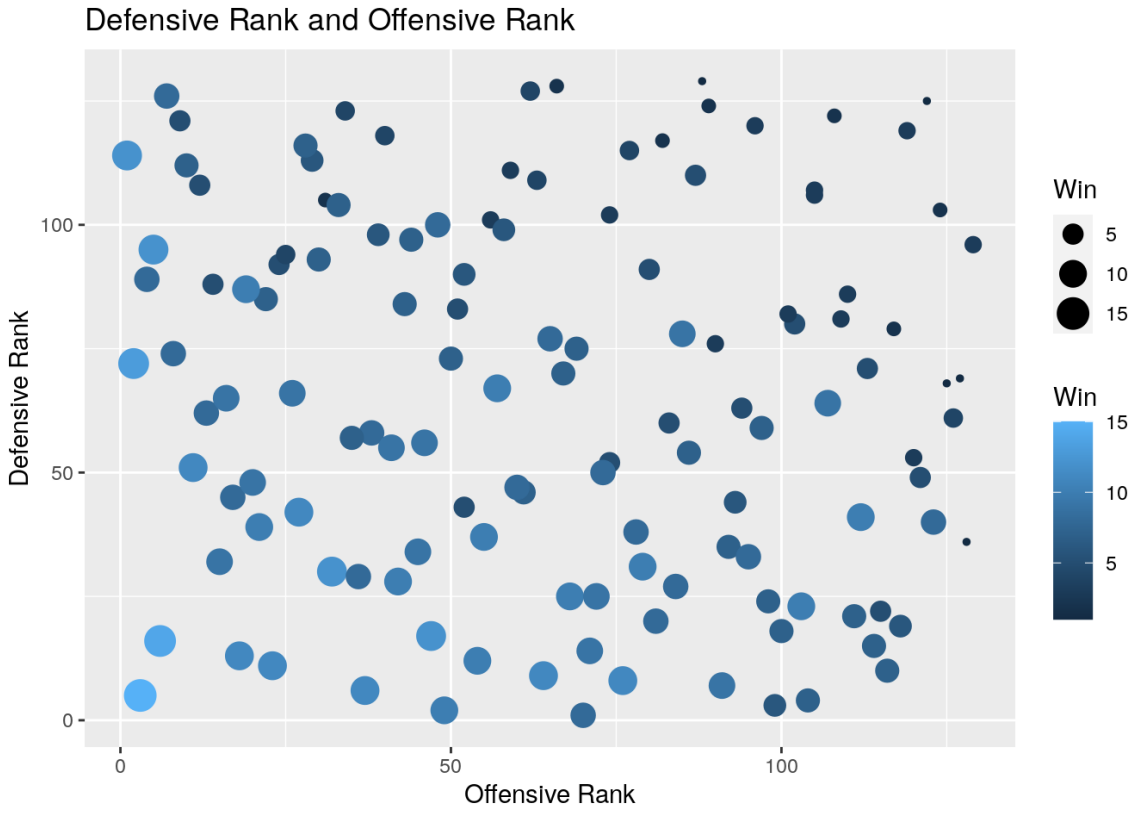
```
cfb18_sep %>%
  ggplot(aes(x = Def.Rank, y = Win, color = Win)) +
  labs(y = "Wins",
       x = "Defensive Rank",
       color = "Wins",
       title = "Defensive Rank vs Wins") +
  geom_point() +
  geom_smooth()
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Exploratory Plot 5

```
cfb18_sep %>%
  ggplot(aes(x = Off.Rank, y = Def.Rank, size = Win, color = Win)) +
  labs(title = "Defensive Rank and Offensive Rank", y = "Defensive Rank", x = "Offensive Rank") +
  geom_point()
```



Individual Variables

We are trying to predict the end of season outcomes in terms of wins with our modeling. Some of the variables we want to use to predict this are split into an offensive and defensive prediction. For offense, we will use offensive rank, touchdowns scored and yards per game. Similarly, for defense we will use defensive rank, touchdowns given up and yards allowed per game. We think that out of the different variables we have available in our data set, these will be the best indicator of how many wins a team will get.

Measuring Accuracy

We will measure the accuracy by comparing the predicted wins of a given team to their actual wins that we have. We will also look at the MAE and see how good or bad out prediction is.

Modeling

```
add_predictions <- function(data, model, variable_name = ".pred", model_name = deparse(substitute(model))) {
  model %>%
    predict(data) %>%
    rename(!!enquo(variable_name) := .pred) %>%
    mutate(model = model_name) %>%
    bind_cols(data)
}
```

```
train <- cfb18_sep
test <- cfb19_sep
```

###model 1 offensive linear predictions

```
model1 <- linear_reg() %>%
  fit(Win ~ Off.Rank+ Off.Yards.Play + Off.TDs, data = cfb18_sep)
```

```
model1 %>%
  tidy() %>%
  select(term, estimate)
```

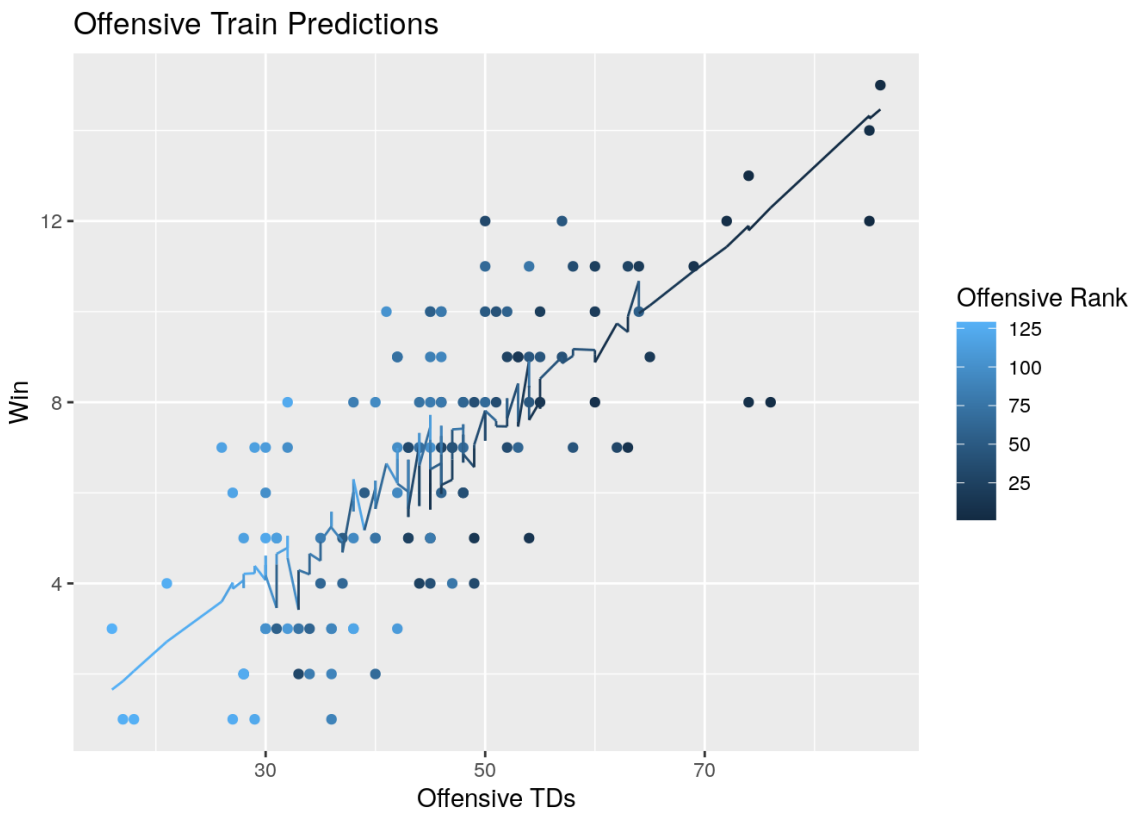
```
# A tibble: 4 × 2
  term           estimate
  <chr>             <dbl>
1 (Intercept)     -4.69
2 Off.Rank         0.0208
3 Off.Yards.Play   0.0435
4 Off.TDs          0.218
```

```
train_predictions <- train %>%
  add_predictions(model1) %>%
  ggplot(aes(x = Off.TDs, y = Win, color = Off.Rank)) +
  labs(x = "Offensive TDs", y = "Win", color = "Offensive Rank", title = "Offensive Train Predictions") +
  geom_point() +
  geom_line(aes(y = .pred))
```

```
Warning: Unknown columns: `(Intercept)`
```

```
New names:
* ...40 -> ...44
```

```
train_predictions
```

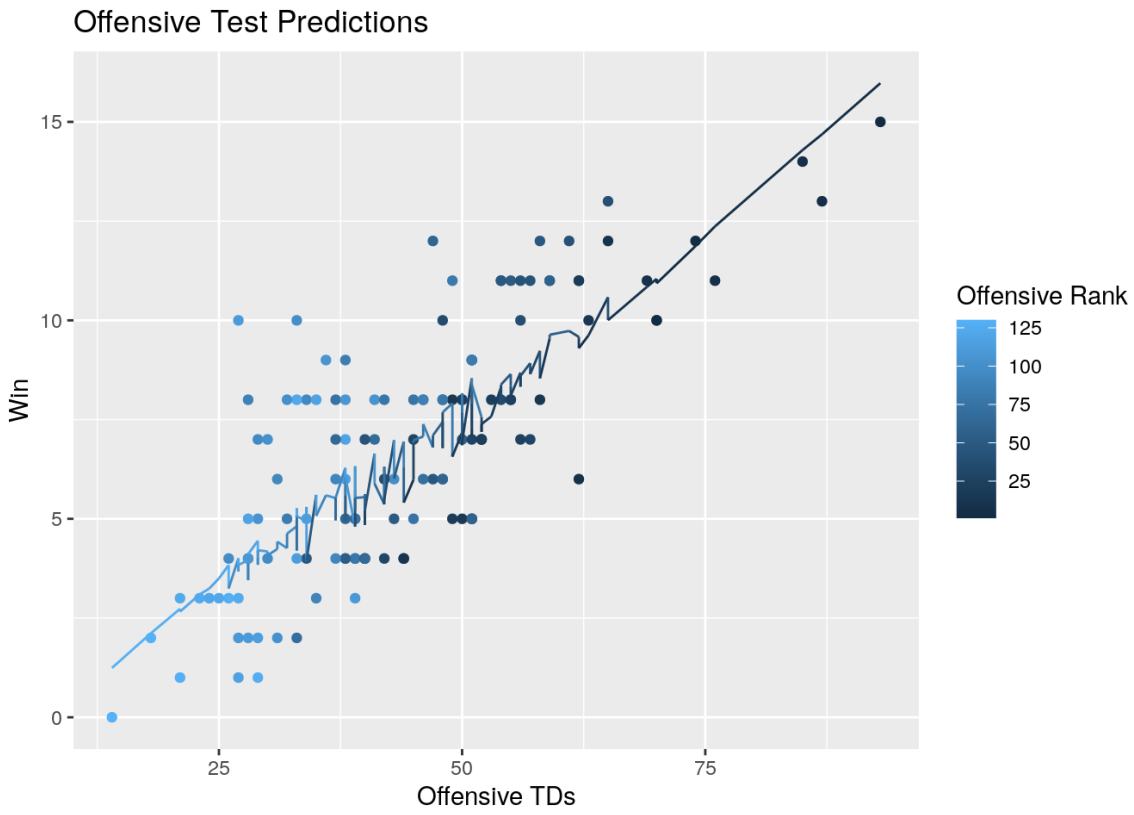



```
test_predictions <- test %>%
  add_predictions(model1) %>%
  ggplot(aes(x = Off.TDs, y = Win, color = Off.Rank)) +
  labs(x = "Offensive TDs", y = "Win", color = "Offensive Rank", title = "Offensive Test Predictions") +
  geom_point() +
  geom_line(aes(y = .pred))
```

```
Warning: Unknown columns: `(Intercept)`
```

```
New names:
* ...40 -> ...44
```

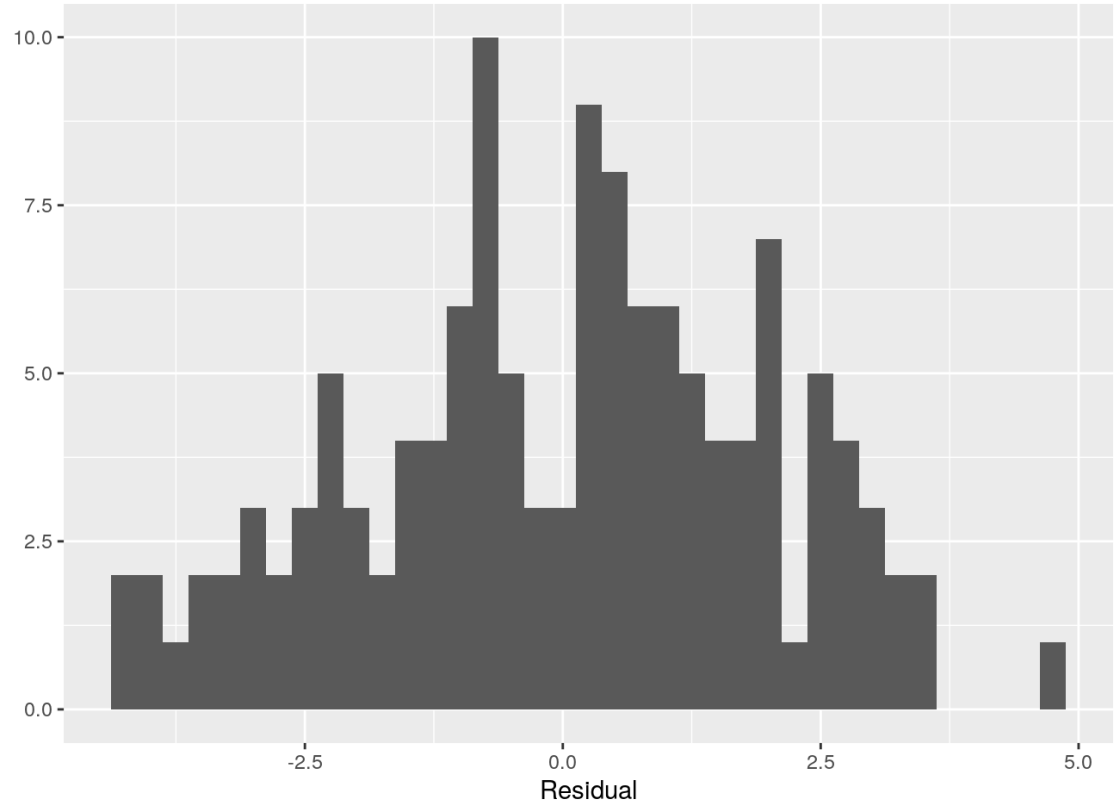
```
test_predictions
```




```
train %>%
  add_predictions(model1) %>%
  mutate(resid = Win - .pred) %>%
  ggplot(aes(x = resid)) +
  geom_histogram(binwidth = .25) +
  labs(x = "Residual", y = "")
```

```
Warning: Unknown columns: `(Intercept)`
```

```
New names:
* ...40 -> ...44
```



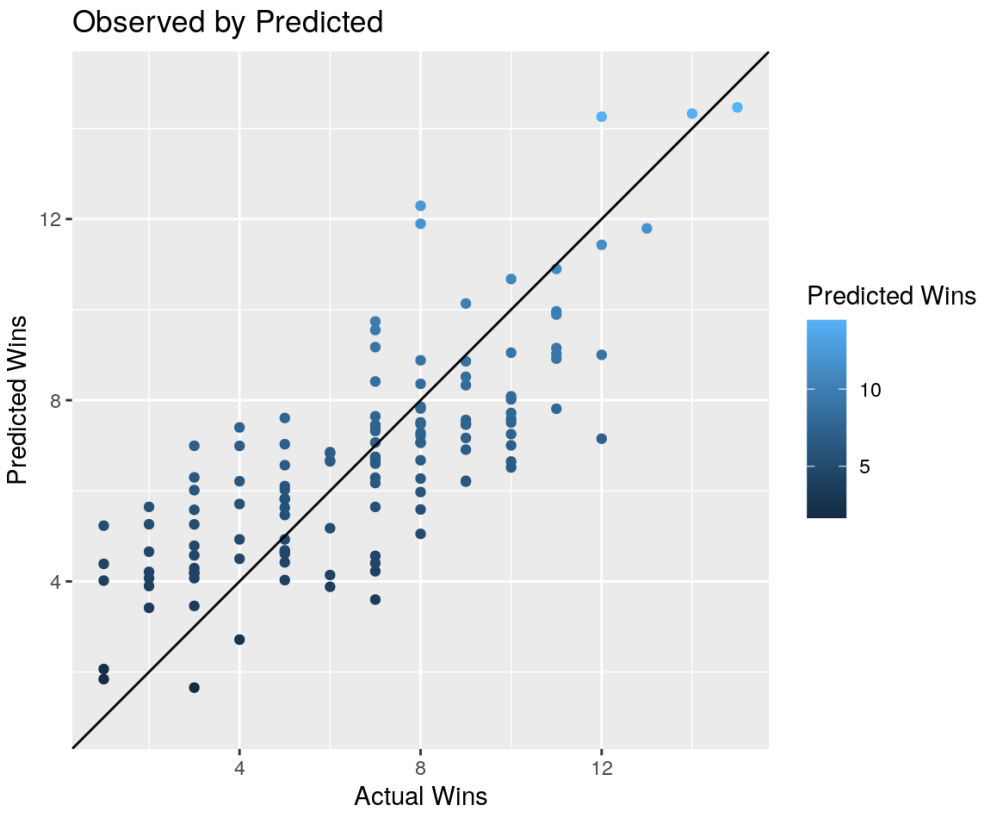

```
  geom_vline(xintercept = 0)
```

```
mapping: xintercept = ~xintercept
geom_vline: na.rm = FALSE
stat_identity: na.rm = FALSE
position_identity
```

```
train %>%
  add_predictions(model1) %>%
  ggplot(aes(x = Win, y = .pred, color = .pred)) +
  geom_point(alpha = 1) +
  coord_obs_pred() +
  geom_abline() +
  labs(x = "Actual Wins", y = "Predicted Wins", title = "Observed by Predicted", color = "Predicted Wins")
```

```
Warning: Unknown columns: `(Intercept)`
```

```
New names:
* ...40 -> ...44
```




```
train %>%
  add_predictions(model1) %>%
  group_by(Team) %>%
  mae(truth = Win, estimate = .pred)
```

```
Warning: Unknown columns: `(Intercept)`
```

```
New names:
* ...40 -> ...44
```

```
# A tibble: 129 × 4
   Team               .metric .estimator .estimate
   <chr>              <chr>   <chr>          <dbl>
 1 "Air Force "       mae     standard       1.02
 2 "Akron "           mae     standard       1.28
 3 "Alabama "         mae     standard       0.327
 4 "Appalachian St. " mae     standard       1.98
 5 "Arizona "         mae     standard       0.466
 6 "Arizona St. "     mae     standard       0.347
 7 "Arkansas "        mae     standard       2.07
 8 "Arkansas St. "    mae     standard       2.03
 9 "Army West Point " mae     standard       2.08
10 "Auburn "          mae     standard       0.780
# … with 119 more rows
```

This model is over predicting wins in every single conference by at least .9 wins.

```
model2 <- linear_reg() %>%
  fit(Win ~ Def.Rank + Yards.Play.Allowed + Off.TDs.Allowed, data = cfb18_sep)
model2
```

```
parsnip model object

Fit time:  3ms

Call:
stats::lm(formula = Win ~ Def.Rank + Yards.Play.Allowed + Off.TDs.Allowed,
    data = data)

Coefficients:
       (Intercept)            Def.Rank  Yards.Play.Allowed     Off.TDs.Allowed
          15.51605            -0.02328            -1.08909            -0.02655
```

###Model 2 defensive linear predictions

```
model2 %>%
  tidy() %>%
  select(term, estimate)
```

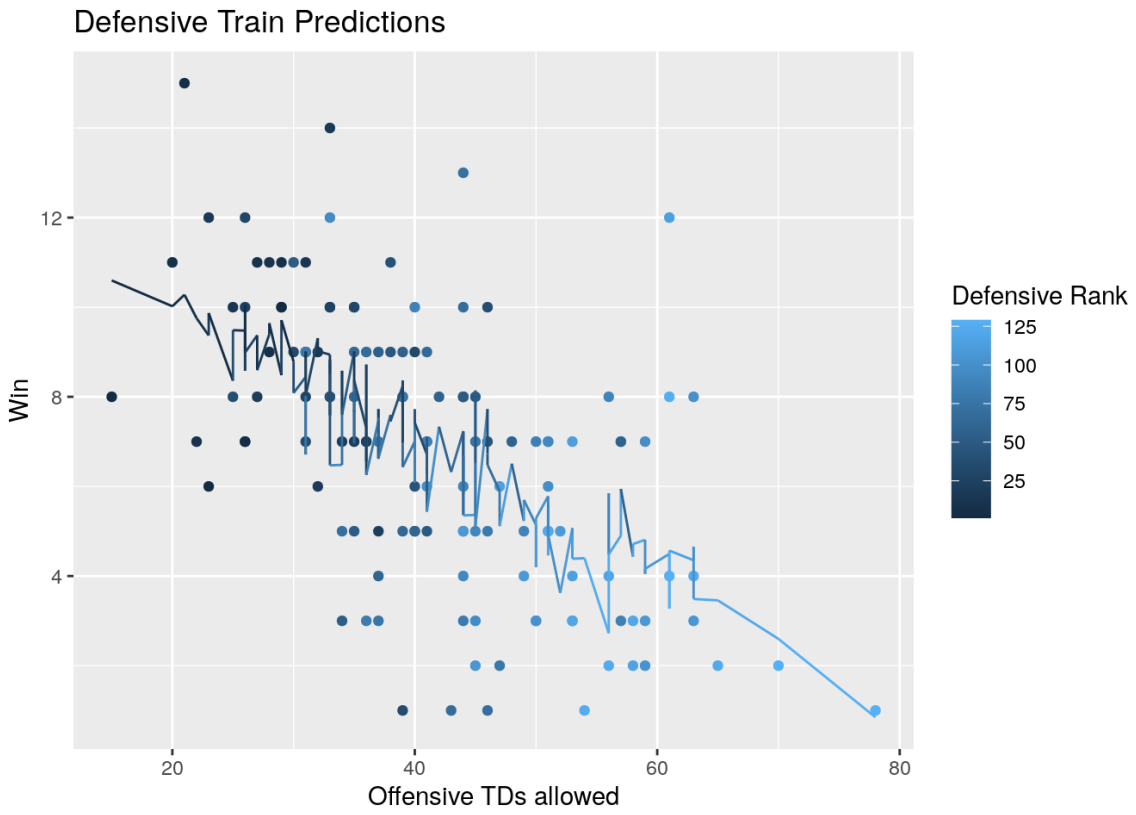
```
# A tibble: 4 × 2
  term               estimate
  <chr>                 <dbl>
1 (Intercept)         15.5
2 Def.Rank            -0.0233
3 Yards.Play.Allowed  -1.09
4 Off.TDs.Allowed     -0.0266
```

```
train_predictions <- train %>%
  add_predictions(model2) %>%
  ggplot(aes(x = Off.TDs.Allowed, y = Win, color = Def.Rank)) +
  labs(x = "Offensive TDs allowed", y = "Win", color = "Defensive Rank", title = "Defensive Train Predictions") +
  geom_point() +
  geom_line(aes(y = .pred))
```

```
Warning: Unknown columns: `(Intercept)`
```

```
New names:
* ...40 -> ...44
```

```
train_predictions
```

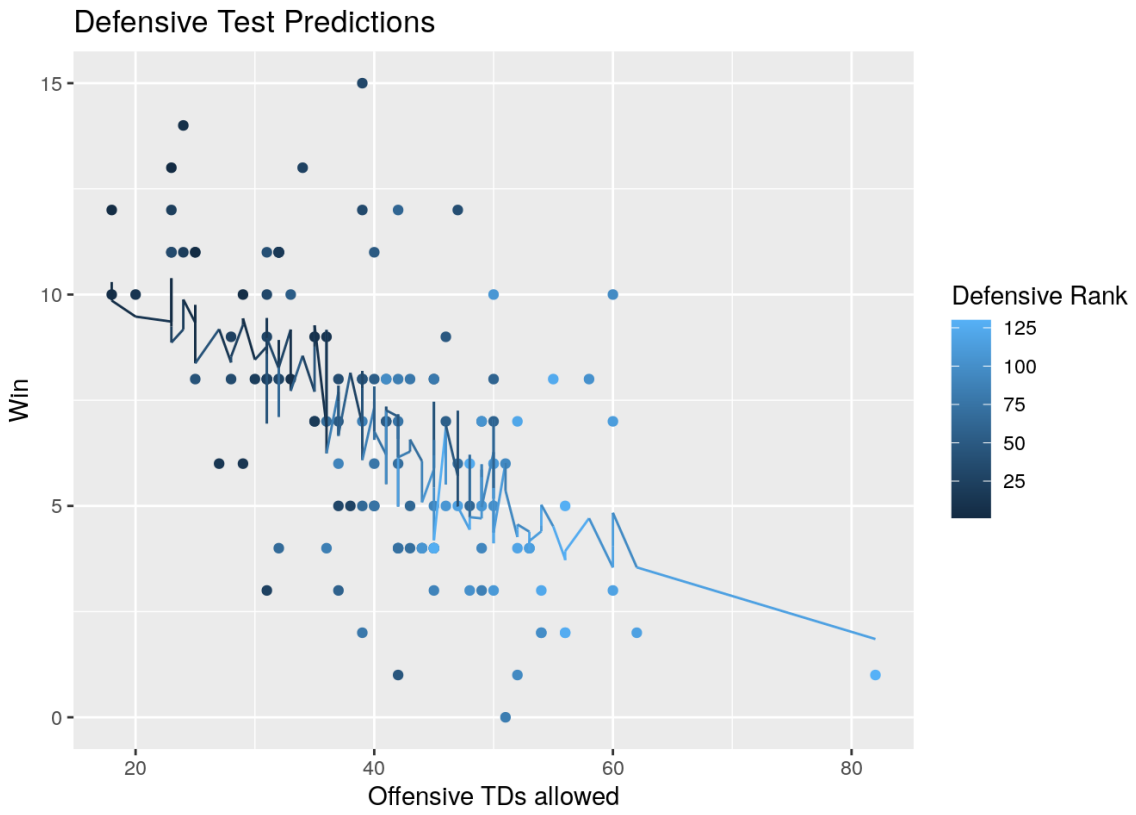



```
test_predictions <- test %>%
  add_predictions(model2) %>%
  ggplot(aes(x = Off.TDs.Allowed, y = Win, color = Def.Rank)) +
  labs(x = "Offensive TDs allowed", y = "Win", color = "Defensive Rank", title = "Defensive Test Predictions") +
  geom_point() +
  geom_line(aes(y = .pred))
```

```
Warning: Unknown columns: `(Intercept)`
```

```
New names:
* ...40 -> ...44
```

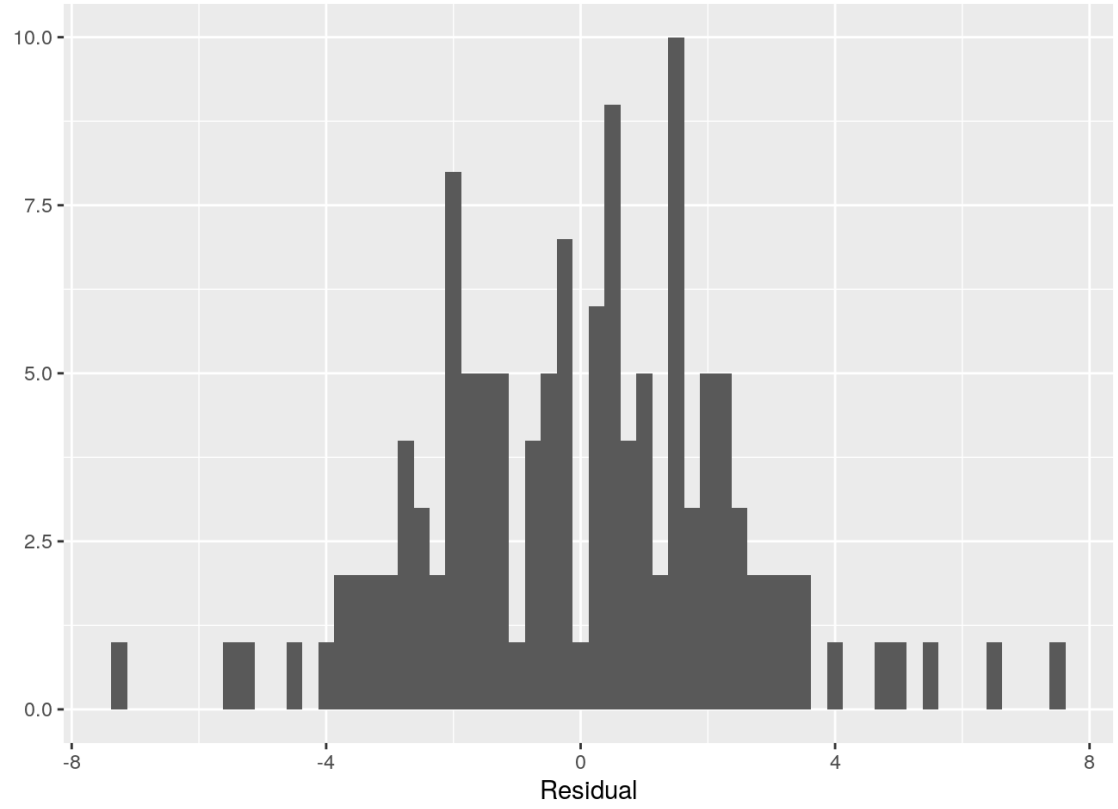
```
test_predictions
```




```
train %>%
  add_predictions(model2) %>%
  mutate(resid = Win - .pred) %>%
  ggplot(aes(x = resid)) +
  geom_histogram(binwidth = .25) +
  labs(x = "Residual", y = "")
```

```
Warning: Unknown columns: `(Intercept)`
```

```
New names:
* ...40 -> ...44
```



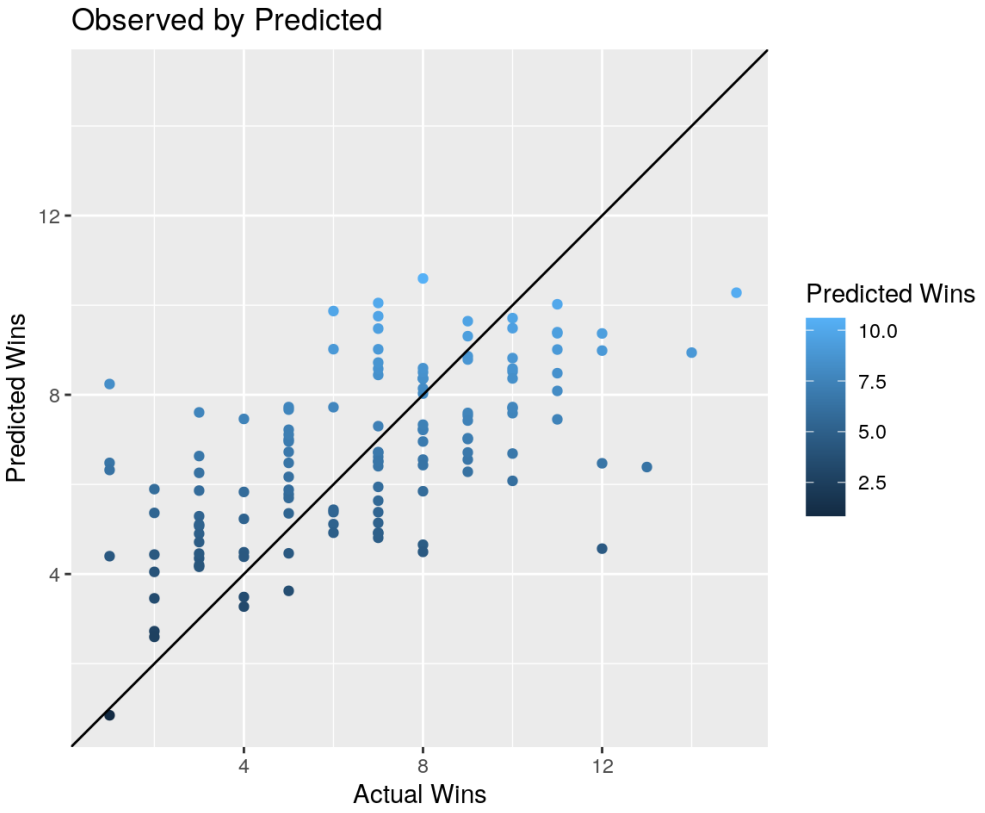

```
  geom_vline(xintercept = 0)
```

```
mapping: xintercept = ~xintercept
geom_vline: na.rm = FALSE
stat_identity: na.rm = FALSE
position_identity
```

```
train %>%
  add_predictions(model2) %>%
  ggplot(aes(x = Win, y = .pred, color = .pred)) +
  geom_point(alpha = 1) +
  coord_obs_pred() +
  geom_abline() +
  labs(x = "Actual Wins", y = "Predicted Wins", title = "Observed by Predicted", color = "Predicted Wins")
```

```
Warning: Unknown columns: `(Intercept)`
```

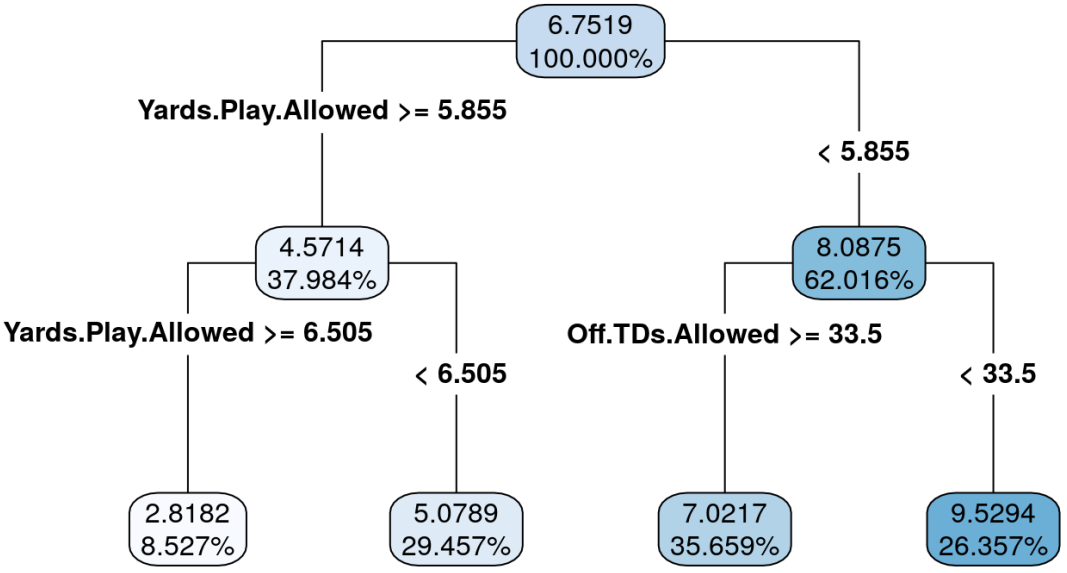
```
New names:
* ...40 -> ...44
```



Model 3

```
model3 <-
  decision_tree(mode = "regression", tree_depth = 2) %>%
  fit(Win ~ Yards.Play.Allowed + Off.TDs.Allowed, data = train)

model3 %>%
  extract_fit_engine() %>%
  rpart.plot(roundint = FALSE, digits = 5, type = 4)
```



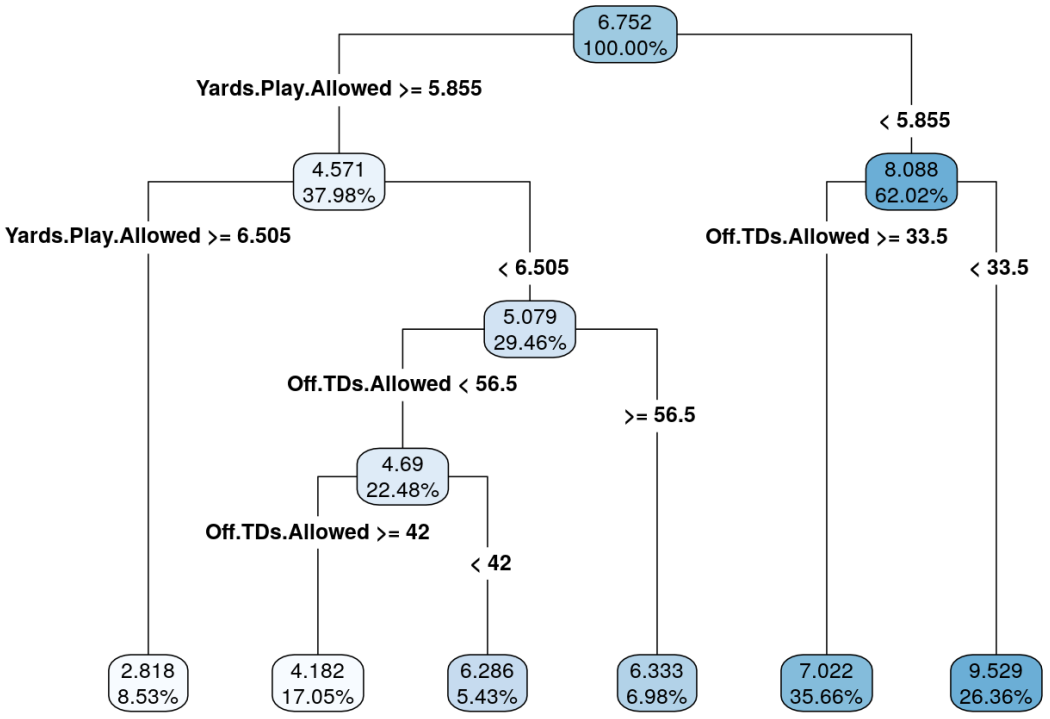
```
train %>%
  add_predictions(model3)
```

```
New names:
* ...40 -> ...44
```

```
# A tibble: 129 × 155
   .pred model  Team   Conference Games   Win  Loss Off.Rank Off.Plays Off.Yards
   <dbl> <chr>  <chr>  <chr>      <int> <int> <int>    <int>     <int>     <int>
 1  5.08 model3 "Air … MWC           12     5     7       52       882      4978
 2  7.02 model3 "Akro… MAC           12     4     8      126       764      3533
 3  9.53 model3 "Alab… SEC           15    14     1        6      1009      7830
 4  9.53 model3 "Appa… Sun Belt      13    11     2       37       859      5603
 5  7.02 model3 "Ariz… Pac-12        12     5     7       24       881      5492
 6  7.02 model3 "Ariz… Pac-12        13     7     6       50       899      5417
 7  5.08 model3 "Arka… SEC           12     2    10      117       810      4028
 8  7.02 model3 "Arka… Sun Belt      13     8     5       17       980      6061
 9  9.53 model3 "Army… FBS Indep…    13    11     2       76       922      5103
10  9.53 model3 "Aubu… SEC           13     8     5       78       894      5069
# … with 119 more rows, and 145 more variables: Off.Yards.Play <dbl>,
#   Off.TDs <int>, Off.Yards.per.Game <dbl>, Def.Rank <int>, Def.Plays <int>,
#   Yards.Allowed <int>, Yards.Play.Allowed <dbl>, Off.TDs.Allowed <int>,
#   Total.TDs.Allowed <int>, Yards.Per.Game.Allowed <dbl>,
#   First.Down.Rank <int>, First.Down.Runs <int>, First.Down.Passes <int>,
#   First.Down.Penalties <int>, First.Downs <int>, First.Down.Def.Rank <int>,
#   Opp.First.Down.Runs <int>, Opp.First.Down.Passes <int>, …
```

```
model4<-
  decision_tree(mode = "regression", tree_depth = 30) %>%
  fit(Win ~ Yards.Play.Allowed + Off.TDs.Allowed, data = train)

model4 %>%
  extract_fit_engine() %>%
  rpart.plot(roundint = FALSE, digits = 4, type = 4)
```



Modeling reasoning

We first chose to create linear models based on offensive and defensive

Summary

Discussion of Findings

After looking at our linear models, we could see a difference in the accuracy of the offensive predictions compared to defensive. The offensive model was quite linear meaning it was accurate while the defensive model was more spread out with barely any linear relationship.

Limitations and Social / Ethical Considerations

One variable that would have been helpful to have is strength of schedule. Strength of schedule would have given us a better indicator of a good team versus a team from a smaller conference that just played teams they were able to score a lot on. This is something that may have made our predictions slightly more accurate.

Future Directions

While exploring more of the data with the train and test data set. A question that came to mind is whether if we could predict win or loss specifically a game with 1v1 teams individually. If we look into another data that has the information of both teams, I wonder how can it help predict the outcome and winner of the game.