Field Goal Model

Step 1: Expected (Field) Goals

```r
install.packages('RCurl')
```

```
## Installing package into '/opt/r'
## (as 'lib' is unspecified)
```

```r
#loading required libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(RCurl)
```

```
##
## Attaching package: 'RCurl'
##
## The following object is masked from 'package:tidyr':
##
##     complete
```

```r
library(ggplot2)
```

```r
# Loading the Data
url <- getURL("https://raw.githubusercontent.com/statsbylopez/StatsSports/master/Data/nfl_fg.csv")
nfl.kick <- read.csv(text = url)
head(nfl.kick)
```

```
##   Team Year GameMinute Kicker Distance ScoreDiff Grass Temp Success
## 1  PHI 2005          3  Akers       49         0 FALSE   72       0
## 2  PHI 2005         29  Akers       49        -7 FALSE   72       0
## 3  PHI 2005         51  Akers       44        -7 FALSE   72       1
## 4  PHI 2005         14  Akers       43        14  TRUE   82       0
## 5  PHI 2005         60  Akers       23         0  TRUE   75       1
## 6  PHI 2005         39  Akers       34        -3  TRUE   68       1
```

```r
#characterisitics of the data - columns
names(nfl.kick)
```

```r
#summary stats for variables in data
summary(nfl.kick)
```

```
##      Team                 Year       GameMinute       Kicker
##  Length:11187       Min.   :2005   Min.   : 1.00   Length:11187
##  Class :character   1st Qu.:2007   1st Qu.:19.00   Class :character
##  Mode  :character   Median :2010   Median :30.00   Mode  :character
##                     Mean   :2010   Mean   :32.74
##                     3rd Qu.:2013   3rd Qu.:46.00
##                     Max.   :2015   Max.   :77.00
##
##     Distance      ScoreDiff            Grass              Temp
##  Min.   :18.0   Min.   :-45.0000   Mode :logical   Min.   :-6.00
##  1st Qu.:28.0   1st Qu.: -4.0000   FALSE:5053      1st Qu.:49.00
##  Median :37.0   Median :  0.0000   TRUE :6134      Median :61.00
##  Mean   :36.9   Mean   :  0.5843                   Mean   :59.07
##  3rd Qu.:45.0   3rd Qu.:  6.0000                   3rd Qu.:70.00
##  Max.   :76.0   Max.   : 48.0000                   Max.   :99.00
##                                                    NA's   :2059
##     Success
##  Min.   :0.0000
##  1st Qu.:1.0000
##  Median :1.0000
##  Mean   :0.8327
##  3rd Qu.:1.0000
##  Max.   :1.0000
##
```

```
#visualization of variables
ggplot(data = nfl.kick, aes(GameMinute), ) +
         geom_histogram()
ggplot(data = nfl.kick, aes(ScoreDiff)) +
         geom_histogram()
ggplot(data = nfl.kick, aes(Distance)) +
         geom_histogram()
```

The summary statistics above and the visualizations help understand how the variable is distributed in the data, and help select the characteristics for our model.

The distance variable is uniformly distributed till about 48 yards, and then we see a right tail implying that further that the number of kicks reduce after that point as the number of kicks drastically reduce.

The Game Minute variable is rather uniformly distributed, with a clear mode just before the second half. The data also looks symmetric. For the purposes of this model, we will change the variable to 'GameQuarter' through grouping of time intervals by 15 minutes.

```
nfl.kick <- nfl.kick %>%
  mutate(GameQuarter = if_else(GameMinute<=15, 1, if_else(GameMinute<=30, 2, if_else(GameMinute<=45, 3,
```

For the ScoreDiff variable, it has a bell shaped distribution and is symmetric with a close mean and median. This variable is transformed into another variable for Win, Loss or Draw depending on the difference in the score at that point.

```
nfl.kick <- nfl.kick %>%
  mutate(WLD = if_else(ScoreDiff==0, 0, if_else(ScoreDiff<0, -1, 1)))
```

```
#summary stats for new variables
summary(nfl.kick$GameQuarter)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   2.531   4.000   4.000
```

```r
summary(nfl.kick$WLD)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.000000 -1.000000  0.000000  0.006525  1.000000  1.000000
```

The variables we select are - the distance from which a shot is taken, what game situation the team was in, the quarter the game was being played in and whether it was played on Grass or Turf

The logistic regression model is below:

$$\log(\frac{P(success = 1)}{1 - P(success = 1)}) = \beta_0 + \beta_1 * Distance + \beta_2 * WLD + \beta_3 * GameQuarter + \beta_4 * Grass + e$$

Success is defined as whether a kick was 'made', whether it was a goal or not.

```r
#fitting the model using logistic regression
fit.1 <- glm(Success ~ Distance + WLD + GameQuarter + Grass, data = nfl.kick, family = "binomial")
summary(fit.1)
```

```
##
## Call:
## glm(formula = Success ~ Distance + WLD + GameQuarter + Grass,
##     family = "binomial", data = nfl.kick)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7593   0.2510   0.4060   0.6432   1.5842
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.784639   0.155419  37.220  < 2e-16 ***
## Distance    -0.102842   0.003144 -32.706  < 2e-16 ***
## WLD         -0.012010   0.030766  -0.390  0.69625
## GameQuarter  0.017131   0.025290   0.677  0.49816
## GrassTRUE   -0.168180   0.054725  -3.073  0.00212 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10105.0  on 11186  degrees of freedom
## Residual deviance:  8738.3  on 11182  degrees of freedom
## AIC: 8748.3
##
## Number of Fisher Scoring iterations: 5
```

The following model is produced:

$$\log(\frac{P(success = 1)}{1 - P(success = 1)}) = 5.78 - 0.103 * Distance - 0.012 * WLD + 0.017 * GameQuarter - 0.168 * Grass + e$$

From the coefficients, we can understand the effect of a variable of the log-odds of a goal. All variables except GameQuarter seem to negatively affect the probability of a goal as they increase. This makes sense, as it is difficult to take a kick from further out, it can be argued that a losing position is positively associated with a kick, and lastly, playing on grass seems to negatively affect the probability.

```r
exp(fit.1$coeff)
```

```
## (Intercept)    Distance         WLD GameQuarter   GrassTRUE
## 325.2646864   0.9022697   0.9880614   1.0172787   0.8452019
```

```r
exp(confint(fit.1))
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %      97.5 %
## (Intercept) 240.5504271 442.4104954
## Distance      0.8966790   0.9078015
## WLD           0.9302238   1.0494690
## GameQuarter   0.9681037   1.0690087
## GrassTRUE     0.7590959   0.9407458
```

Since the model is expressed in log-odds, we exponentiate the coefficients to better understand the model.

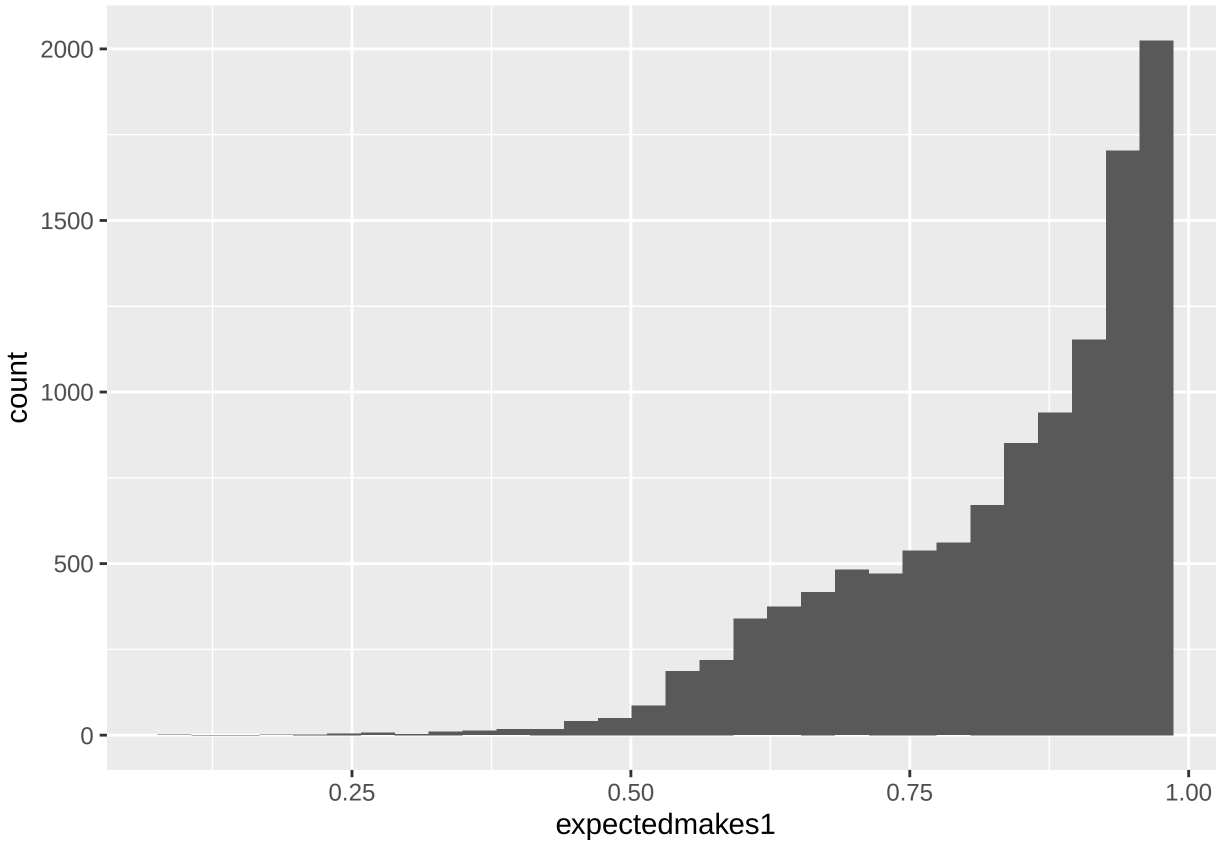
$$\frac{P(success = 1)}{1 - P(success = 1)} = 325.26 + 0.9 * Distance + 0.99 * WLD + 1.02 * GameQuarter + 0.85 * Grass + e$$

A one-unit increase in the explanatory variables will multiple the odds ratio by the exponential value of the coefficients. Therefore, as WLD and Game Quarter are not statistically significant results, and their coefficients are close 1, we can say that a losing position and as the game goes on, a kick is slightly more likely to go in, on average. On the other hand, a one-unit increase in distance will reduce the odds ratio by 0.9*(probability of sucess)*(probabilty of failure), where the probability is estimated for the data point in question. Similarly, Grass has a negative effect in the odds ratio as well.

```r
nfl.kick <- nfl.kick %>%
  mutate(expectedmakes1 = fitted(fit.1))
nfl.kick <- nfl.kick %>%
  mutate(extramakes1 = Success-expectedmakes1)
```

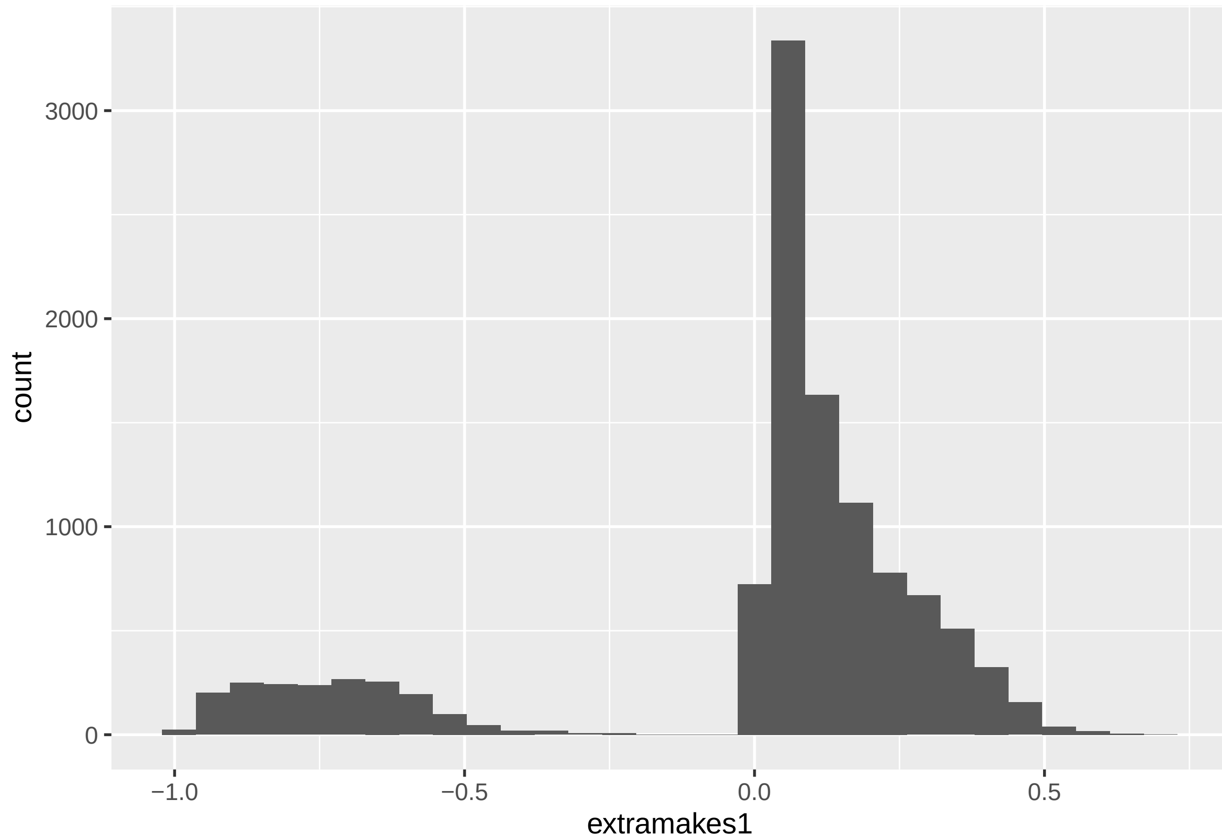
```r
ggplot(nfl.kick, aes(expectedmakes1)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

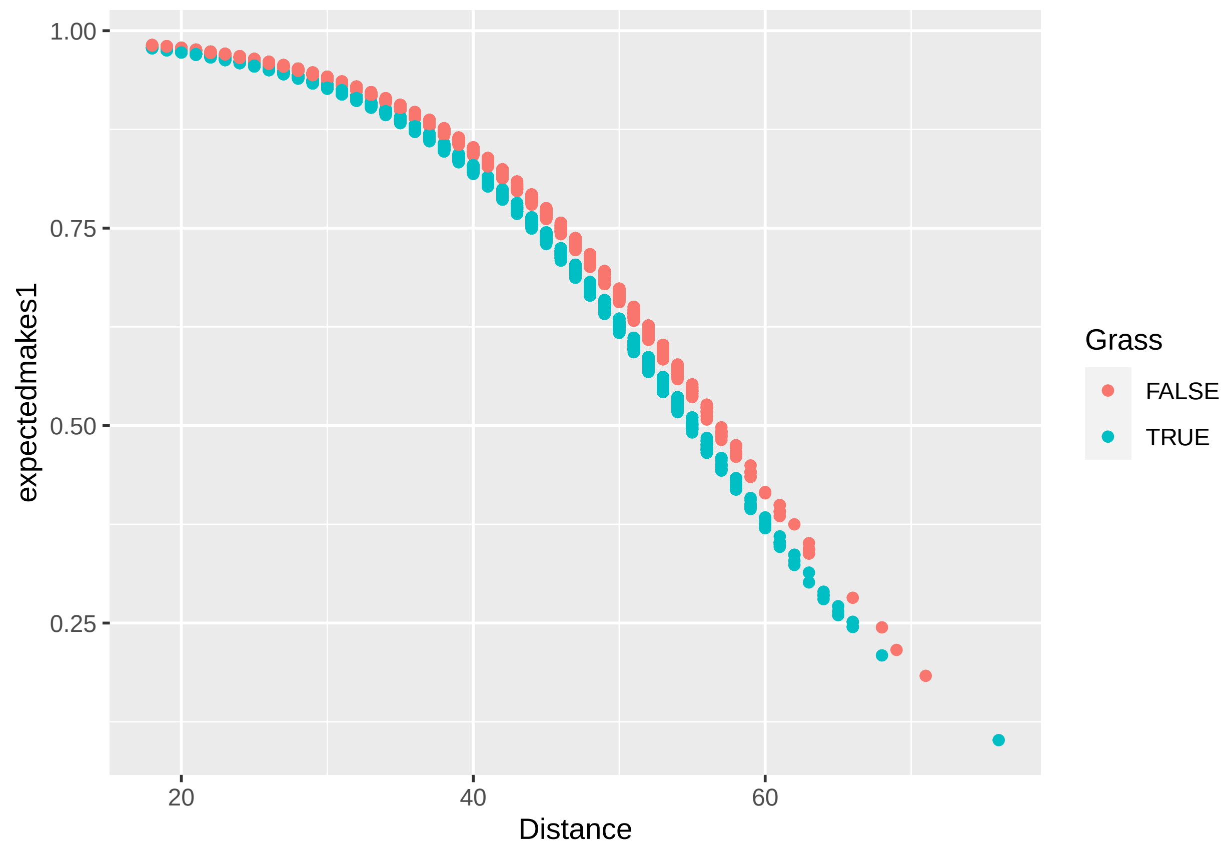



```
ggplot(nfl.kick, aes(extramakes1)) +
  geom_histogram()
```

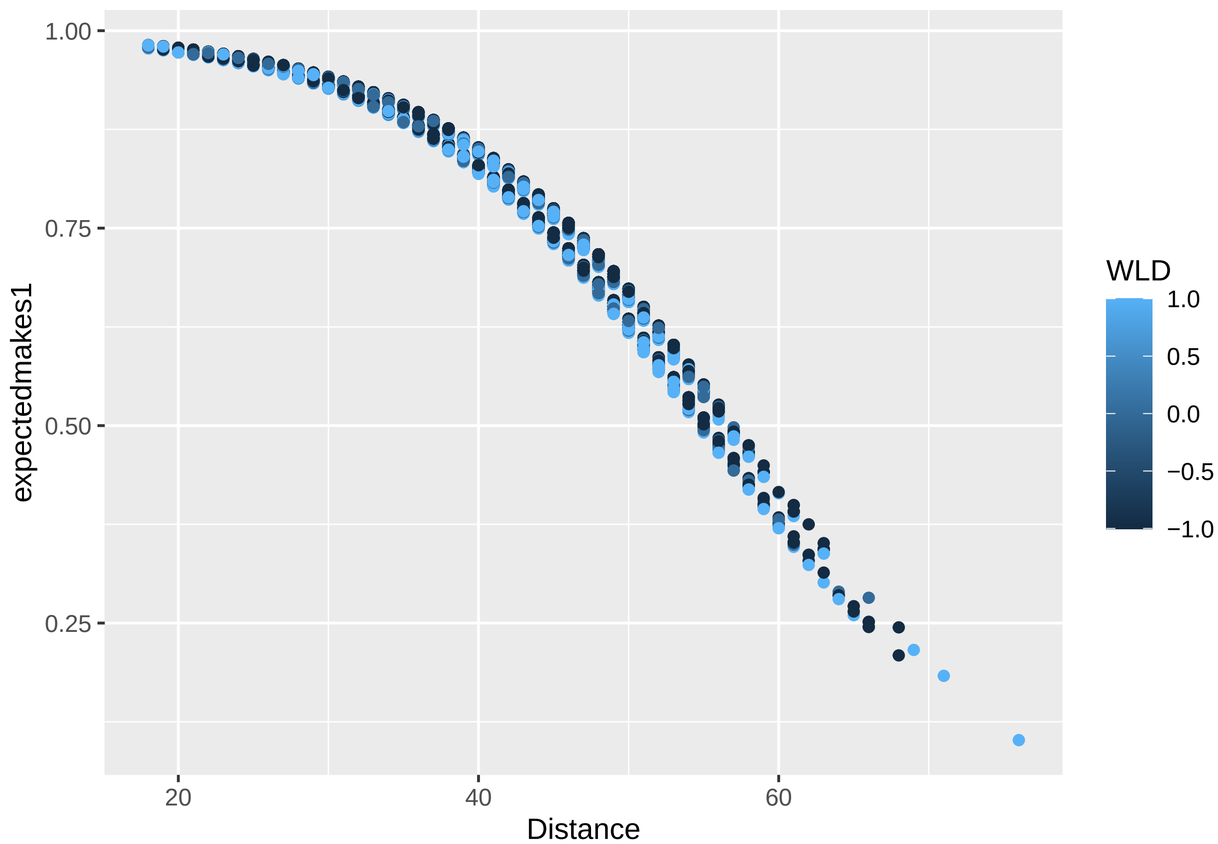
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



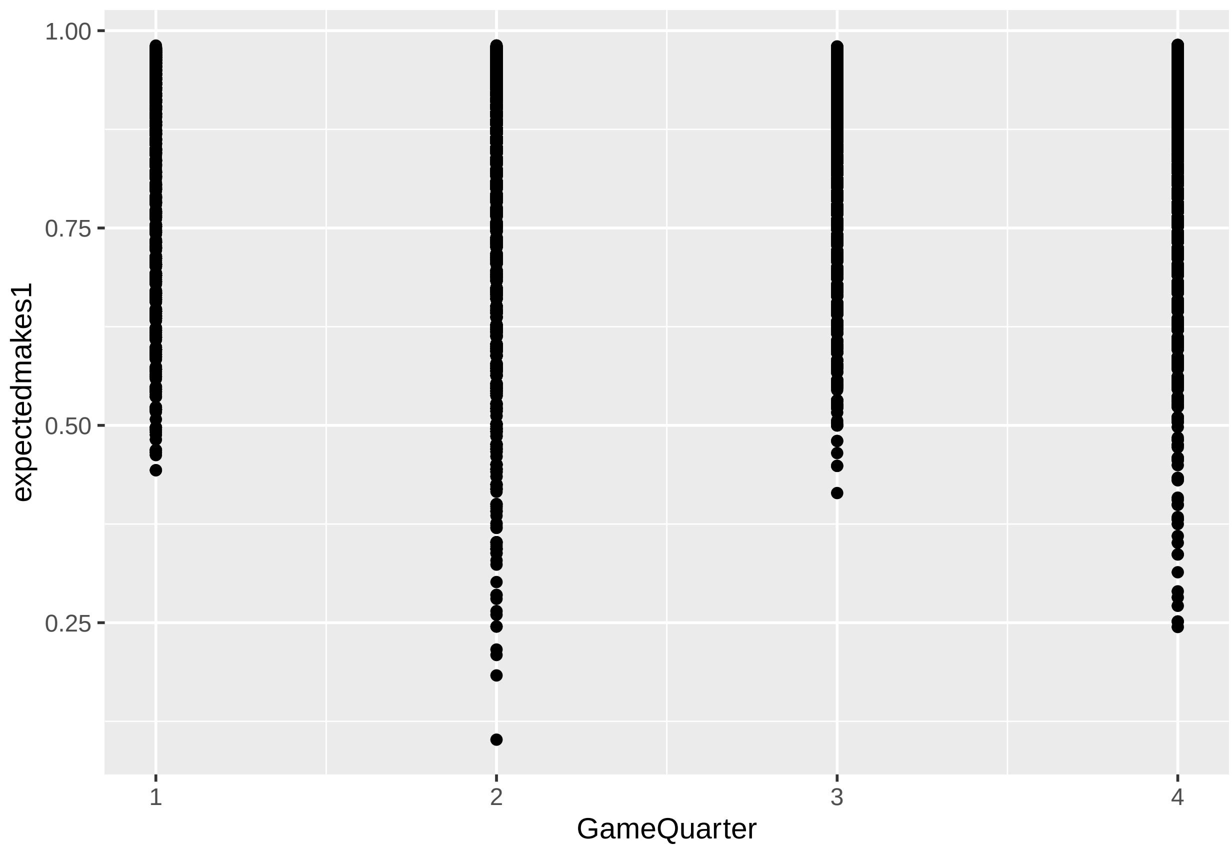

```
ggplot(nfl.kick,aes(Distance, expectedmakes1 ,color=Grass)) + geom_point()
```




```
ggplot(nfl.kick,aes(Distance, expectedmakes1 ,color=WLD)) + geom_point()
```




```
ggplot(nfl.kick,aes(GameQuarter, expectedmakes1)) + geom_point()
```



Step 2: Points above average

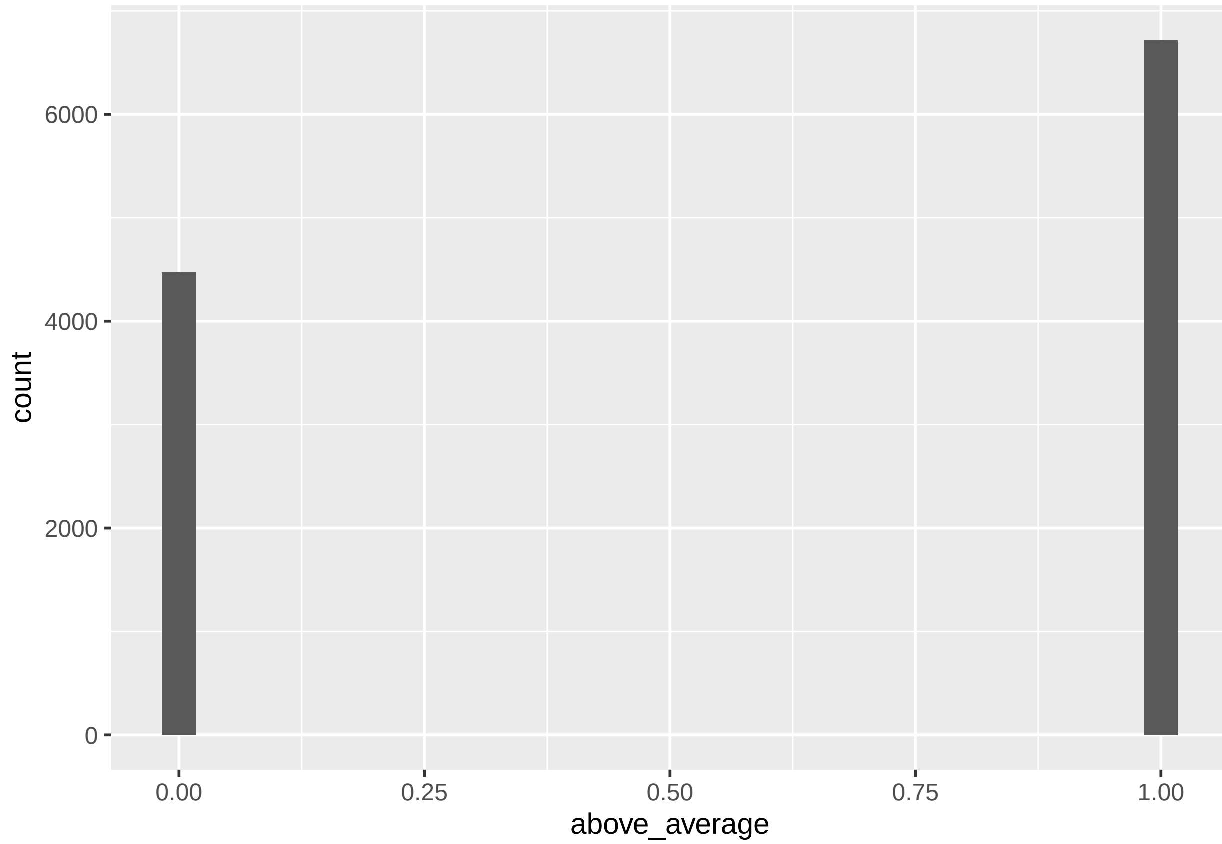
```
summary(nfl.kick$expectedmakes1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1018  0.7471  0.8736  0.8327  0.9438  0.9820
```

```
#adding a variable to show that a shot was above average
kick <- nfl.kick %>%
  select(Kicker, expectedmakes1, Success, Distance) %>%
  mutate(above_average = if_else(expectedmakes1>mean(expectedmakes1), 1, 0))
```

```
ggplot(kick,aes(above_average)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



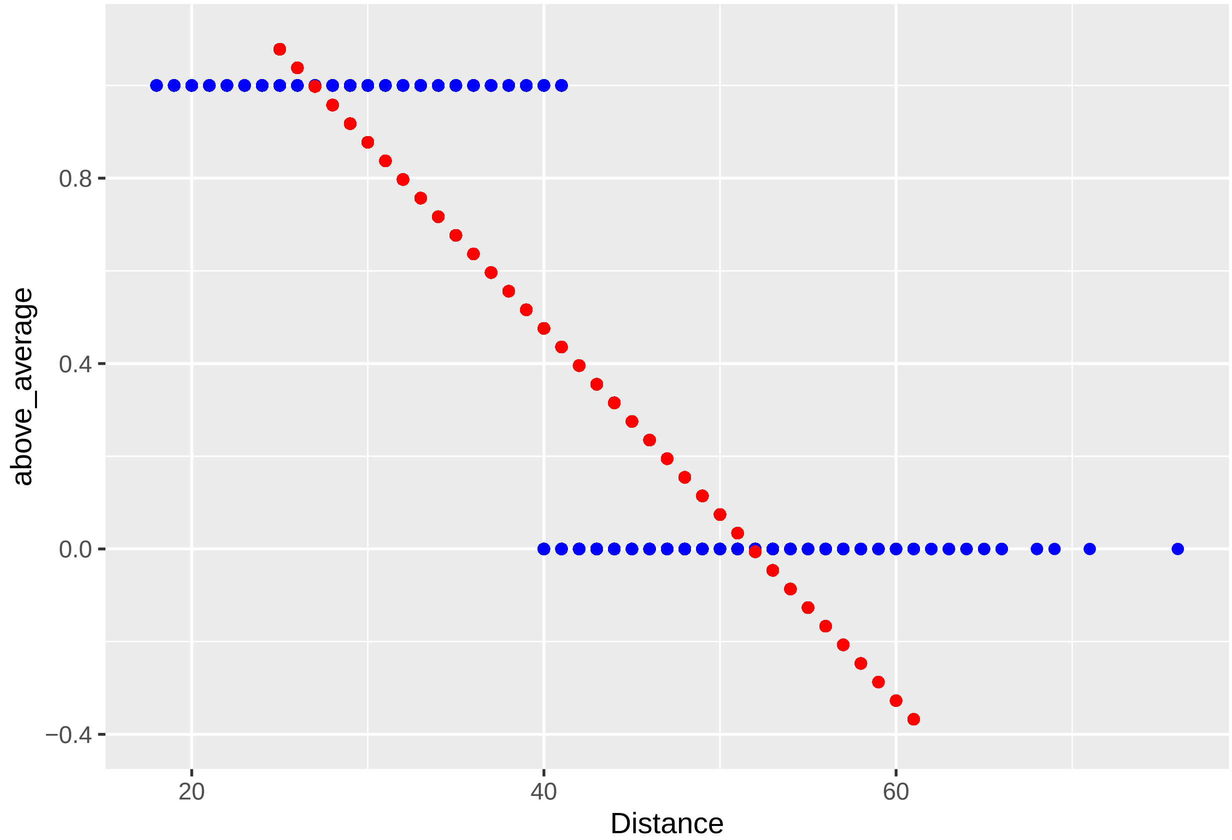

```r
linearDist <- lm(above_average ~ Distance, data = kick)
summary(linearDist)
```

```
##
## Call:
## lm(formula = above_average ~ Distance, data = kick)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4758 -0.1987 -0.0341  0.2029  0.9699
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.0821560  0.0096132   216.6   <2e-16 ***
## Distance    -0.0401581  0.0002512  -159.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2703 on 11185 degrees of freedom
## Multiple R-squared:  0.6956, Adjusted R-squared:  0.6956
## F-statistic: 2.556e+04 on 1 and 11185 DF,  p-value: < 2.2e-16
```

```r
kick <- kick %>%
  mutate(LinearPrediction = fitted(linearDist))
```

```
ggplot() + geom_point(data=kick,aes(x=Distance, y=above_average),color="blue") + geom_point(data=kick,ae
```

```
## Warning: Removed 1668 rows containing missing values (`geom_point()`).
```



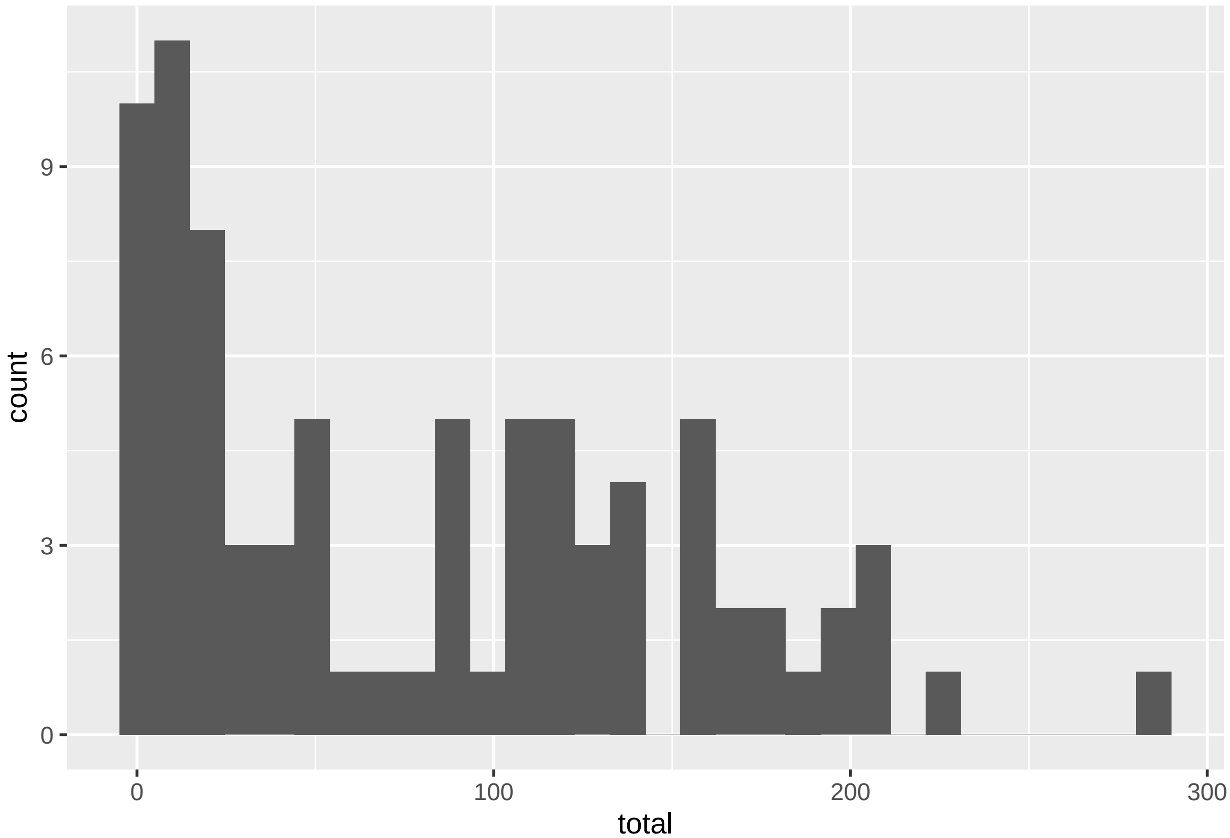
Step 3: Top 5's

```
Kicker1 <- kick %>%
    group_by(Kicker) %>%
    summarize(total=sum(above_average),numkicks=n(),pointsperkick=total/numkicks)
head(Kicker1)
```

```
## # A tibble: 6 x 4
##   Kicker   total numkicks pointsperkick
##   <chr>    <dbl>    <int>         <dbl>
## 1 Akers      211      336         0.628
## 2 Andersen    40       51         0.784
## 3 Andrus       3        5         0.6
## 4 Bailey      91      162         0.562
## 5 Barth       87      166         0.524
## 6 Bironas    161      283         0.569
```

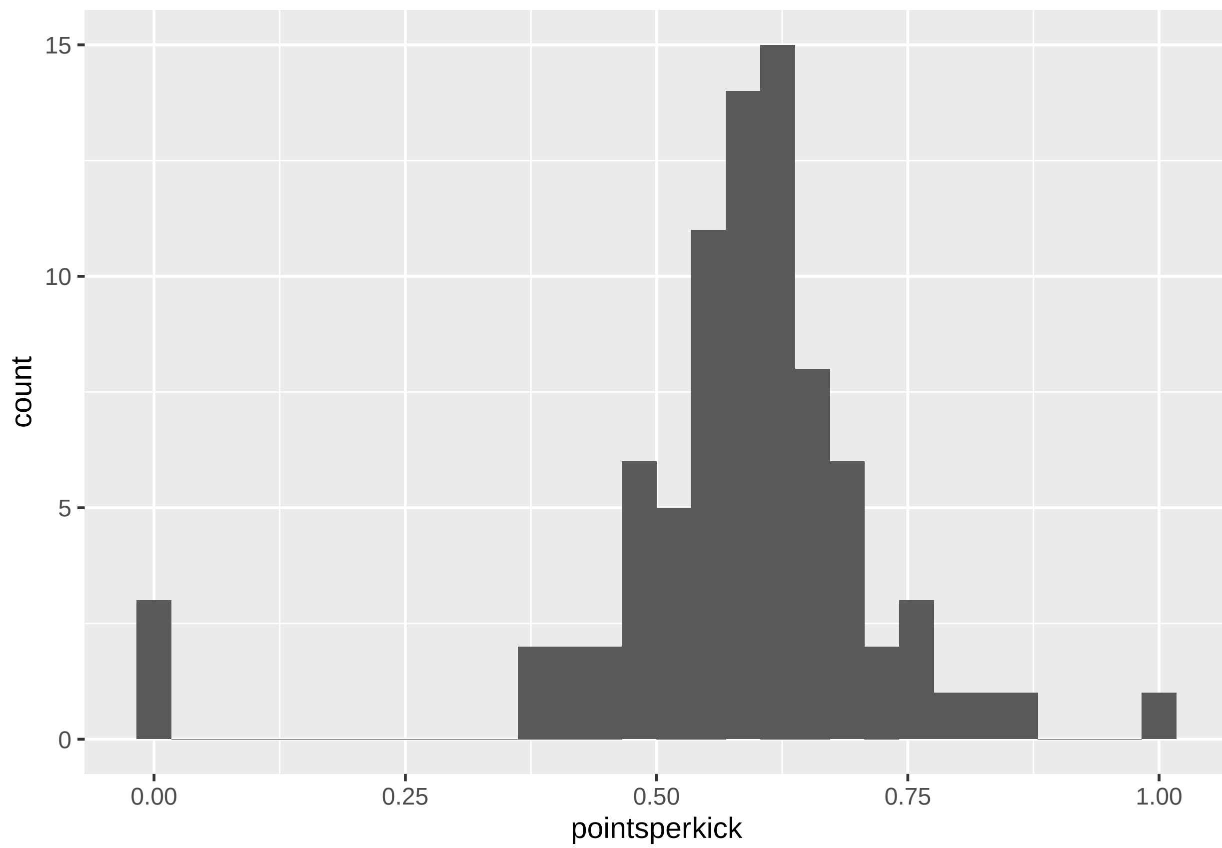
```
ggplot(Kicker1,aes(total)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(Kicker1,aes(pointsperkick)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The first distribution does not have any clear shape, but a few clear outliers are seen. The second distribution shows that the model always predicted an above average shot for a player and never predicted the same for a few. The rest of the distribution looks rather symmetric around 0.62 and ranges from 0.36 to 0.88. The most amount of values are centred around the data.

```
Kicker2 <- Kicker1 %>% arrange(desc(total))
Kicker3 <- Kicker1 %>% arrange(desc(pointsperkick))
head(Kicker2, 5)
```

```
## # A tibble: 5 x 4
##   Kicker     total numkicks pointsperkick
##   <chr>      <dbl>    <int>         <dbl>
## 1 Brown        285      488         0.584
## 2 Gostkowski   227      342         0.664
## 3 Akers        211      336         0.628
## 4 Dawson       208      332         0.627
## 5 Vinatieri    206      339         0.608
```

```
head(Kicker3, 5)
```

```
## # A tibble: 5 x 4
##   Kicker   total numkicks pointsperkick
##   <chr>    <dbl>    <int>         <dbl>
## 1 Schmitt      3        3         1
## 2 Stitser      7        8         0.875
## 3 Peterson    21       25         0.84
## 4 Andersen    40       51         0.784
## 5 Carney     107      139         0.770
```

```r
#add a variable to assess whether a shot is a long or not (45 yards+)
kick <- nfl.kick %>%
  select(Kicker, expectedmakes1, Success, Distance) %>%
  mutate(longshot=if_else(Distance>45, 1, 0)) %>%
  mutate(above_average = if_else(expectedmakes1>mean(expectedmakes1), 1, 0))

#fliter for only longshots
Kicker4 <- kick %>%
    select(longshot, Kicker, expectedmakes1, Success, above_average) %>%
    filter(longshot==1) %>%
    group_by(Kicker) %>%
    summarize(total=sum(above_average),numkicks=n(),pointspergame=total/numkicks)
Kicker4 %>% arrange(desc(total))
```

```
## # A tibble: 77 x 4
##    Kicker   total numkicks pointspergame
##    <chr>    <dbl>    <int>         <dbl>
##  1 Akers        0       66             0
##  2 Andersen     0        6             0
##  3 Andrus       0        1             0
##  4 Bailey       0       52             0
##  5 Barth        0       49             0
##  6 Bironas      0       77             0
##  7 Boswell      0        9             0
##  8 Brien        0        2             0
##  9 Brindza      0        4             0
## 10 Brown        0      129             0
## # i 67 more rows
```

```r
Kicker5 <- Kicker4 %>% arrange(desc(pointspergame))
head(Kicker4, 5)
```

```
## # A tibble: 5 x 4
##   Kicker   total numkicks pointspergame
##   <chr>    <dbl>    <int>         <dbl>
## 1 Akers        0       66             0
## 2 Andersen     0        6             0
## 3 Andrus       0        1             0
## 4 Bailey       0       52             0
## 5 Barth        0       49             0
```

```r
head(Kicker5, 5)
```

```
## # A tibble: 5 x 4
##   Kicker   total numkicks pointspergame
##   <chr>    <dbl>    <int>         <dbl>
## 1 Akers        0       66             0
## 2 Andersen     0        6             0
## 3 Andrus       0        1             0
## 4 Bailey       0       52             0
## 5 Barth        0       49             0
```

Looking at 35 yard+ kicks:

```r
kick <- nfl.kick %>%
  select(Kicker, expectedmakes1, Success, Distance) %>%
```

```
  mutate(longshot=if_else(Distance>35, 1, 0)) %>%
  mutate(above_average = if_else(expectedmakes1>mean(expectedmakes1), 1, 0))

#looking at player stats
Kicker6 <- kick %>%
    select(longshot, Kicker, expectedmakes1, Success, above_average) %>%
    filter(longshot==1) %>%
    group_by(Kicker) %>%
    summarize(total=sum(above_average),numkicks=n(),pointspergame=total/numkicks)
Kicker7 <- Kicker6 %>% arrange(desc(total))
Kicker8 <- Kicker6 %>% arrange(desc(pointspergame))
head(Kicker7, 5)
```

```
## # A tibble: 5 x 4
##   Kicker    total numkicks pointspergame
##   <chr>     <dbl>    <int>         <dbl>
## 1 Brown        71      274         0.259
## 2 Akers        55      180         0.306
## 3 Gould        54      188         0.287
## 4 Vinatieri    53      186         0.285
## 5 Feely        49      154         0.318
```

```
head(Kicker8, 6)
```

```
## # A tibble: 6 x 4
##   Kicker   total numkicks pointspergame
##   <chr>    <dbl>    <int>         <dbl>
## 1 Schmitt      1        1           1
## 2 Peterson     6       10           0.6
## 3 Andersen    11       22           0.5
## 4 Andrus       2        4           0.5
## 5 Brien        2        4           0.5
## 6 Stitser      1        2           0.5
```

Step 4: Measuring Kicker Effectiveness

```
kick <- nfl.kick %>%
  select(Kicker, expectedmakes1, Success, Distance) %>%
  mutate(above_average = if_else(expectedmakes1>mean(expectedmakes1), 1, 0))
head(kick)
```

```
##   Kicker expectedmakes1 Success Distance above_average
## 1  Akers      0.6819010       0       49             0
## 2  Akers      0.6881884       0       49             0
## 3  Akers      0.7925121       1       44             0
## 4  Akers      0.7684149       0       43             0
## 5  Akers      0.9650959       1       23             1
## 6  Akers      0.8987345       1       34             1
```

```
Kicker1 <- kick %>%
    group_by(Kicker, Distance) %>%
    summarize(total=sum(above_average),numkicks=n(),pointsperkick=total/numkicks)
```

```
## `summarise()` has grouped output by 'Kicker'. You can override using the
## `.groups` argument.
```

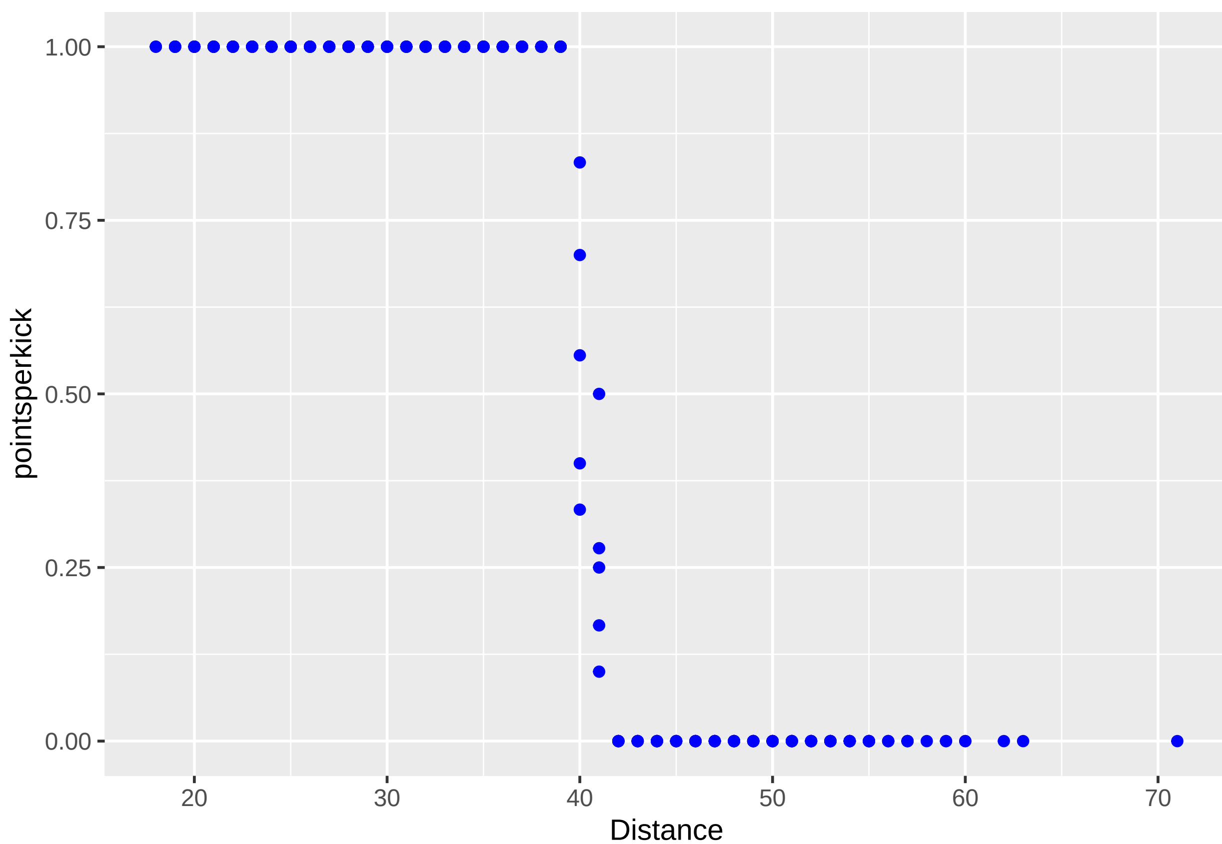
```r
head(Kicker1)
```

```
## # A tibble: 6 x 5
## # Groups:   Kicker [1]
##   Kicker Distance total numkicks pointsperkick
##   <chr>     <int> <dbl>    <int>         <dbl>
## 1 Akers        18     3        3             1
## 2 Akers        19     5        5             1
## 3 Akers        20     8        8             1
## 4 Akers        21     5        5             1
## 5 Akers        22    13       13             1
## 6 Akers        23     9        9             1
```

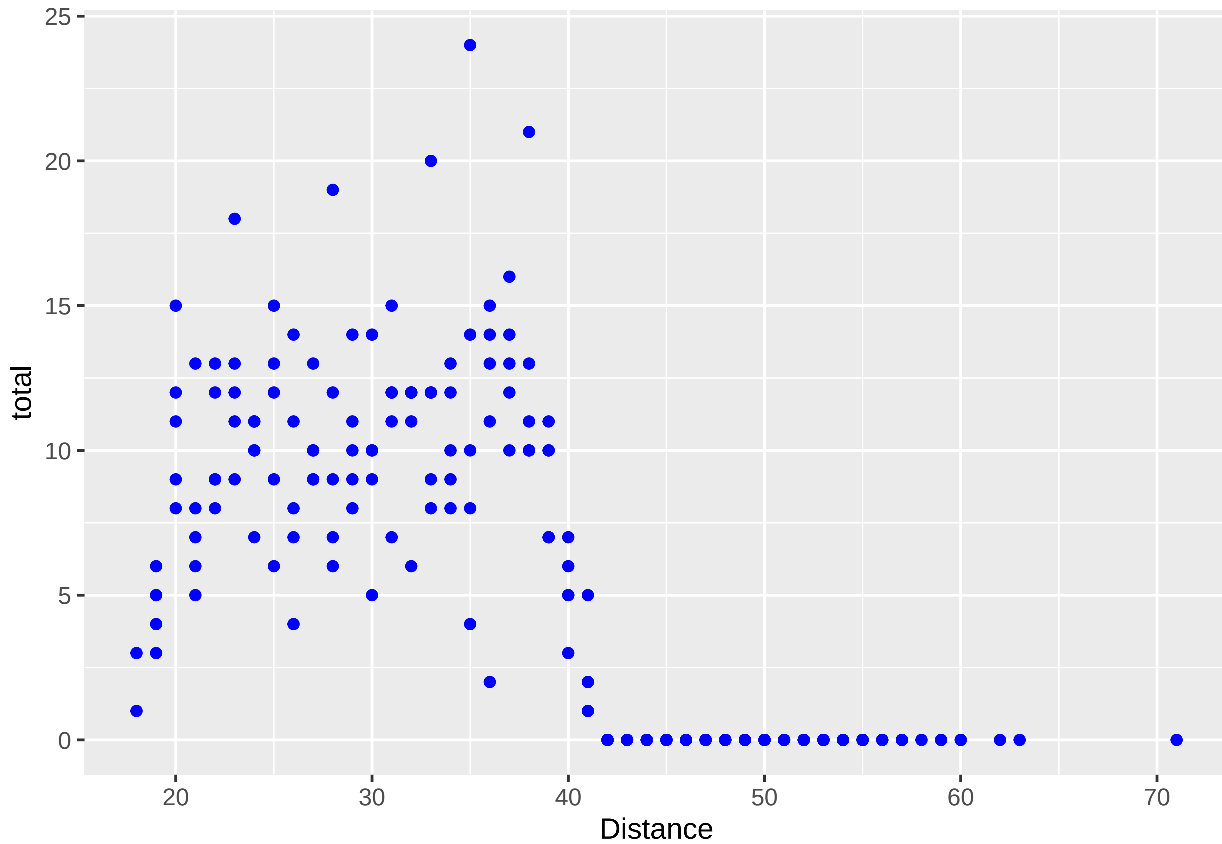
```r
#filtering out for only the top 5 kickers based on total above average shots
Kicker1 <- Kicker1 %>%
  filter(Kicker == "Akers" | Kicker== "Brown" | Kicker == "Gostkowski" |Kicker ==
           "Vinatieri" | Kicker == "Dawson")
```

```r
Akers <- Kicker1 %>%
  filter(Kicker == "Akers")
```

```r
ggplot() + geom_point(data=Kicker1,aes(x=Distance, y=pointsperkick),color="blue")
```




```r
ggplot() + geom_point(data=Kicker1,aes(x=Distance, y=total),color="blue")
```



We can see that, from both distributions that till around the 40 yard mark, the model always predicts an above average kick, however it starts falling after every yard, after that mark.

```
linearDist <- lm(pointsperkick ~ Distance, data = Akers)
summary(linearDist)
```

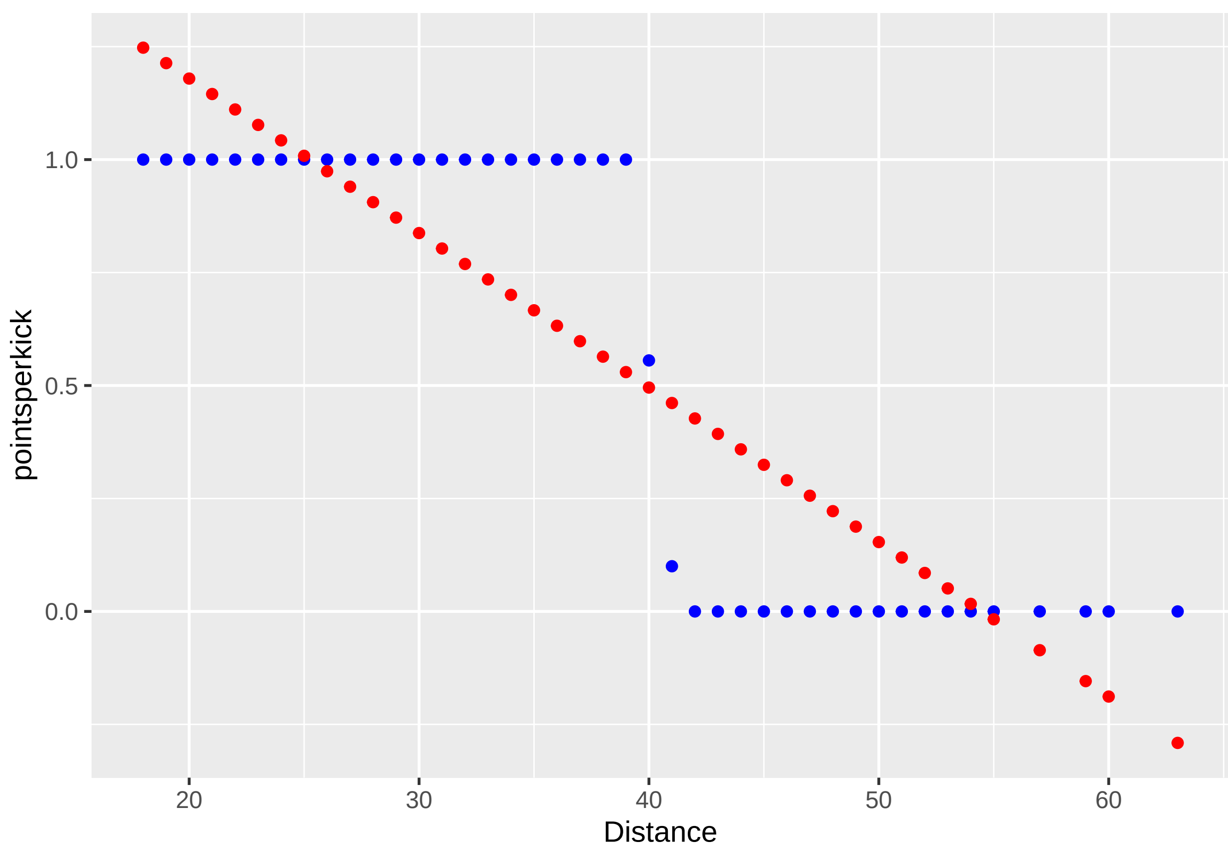
```
##
## Call:
## lm(formula = pointsperkick ~ Distance, data = Akers)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42707 -0.18562 -0.01255  0.18194  0.47035
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.863134   0.124355   14.98  < 2e-16 ***
## Distance    -0.034192   0.003057  -11.18 6.95e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2473 on 40 degrees of freedom
## Multiple R-squared:  0.7577, Adjusted R-squared:  0.7516
## F-statistic: 125.1 on 1 and 40 DF,  p-value: 6.952e-14
```

```
Akers <- Akers %>%
  mutate(LinearPrediction = fitted(linearDist))
linearDista <- lm(total ~ Distance, data = Akers)
```

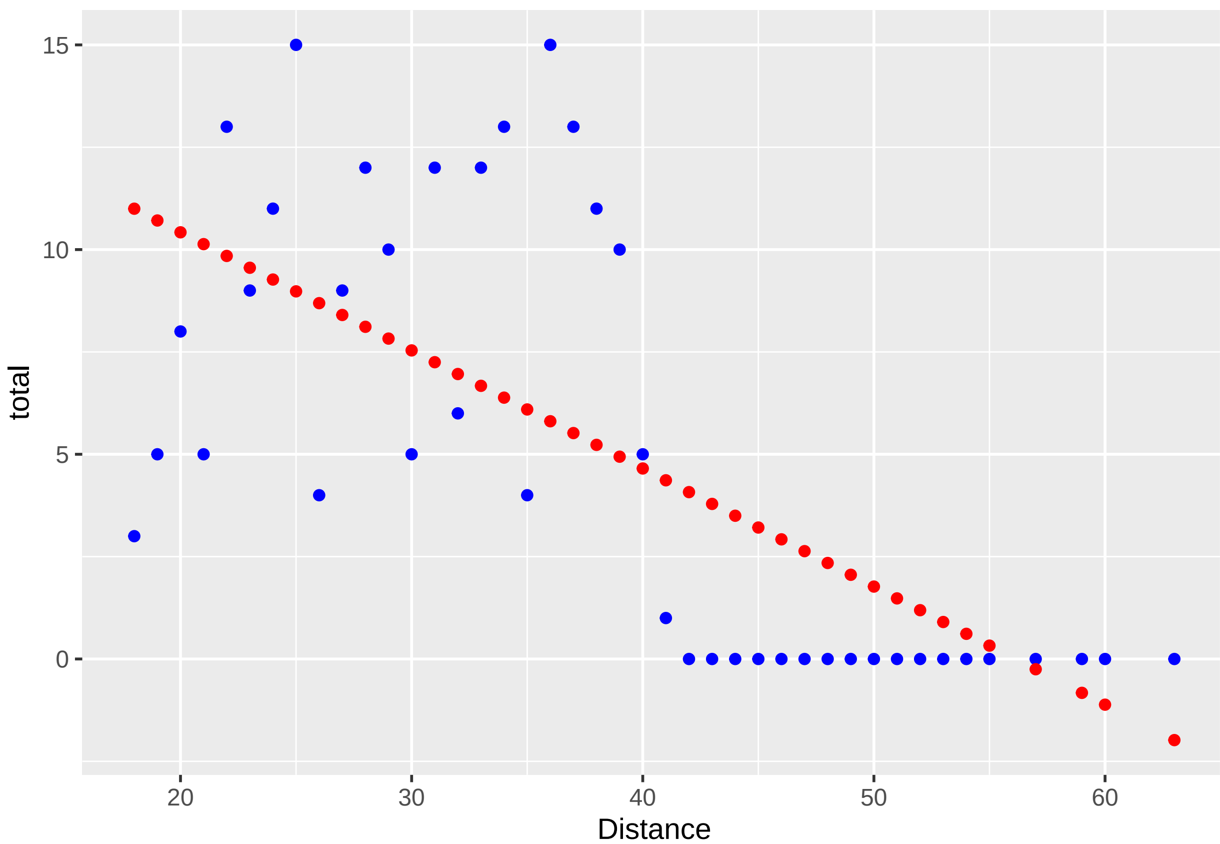
```
summary(linearDista)
```

```
##
## Call:
## lm(formula = total ~ Distance, data = Akers)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.999 -2.610 -0.759  2.126  9.193
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.19055    1.99625   8.110 5.60e-10 ***
## Distance    -0.28844    0.04908  -5.877 7.03e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.97 on 40 degrees of freedom
## Multiple R-squared:  0.4634, Adjusted R-squared:   0.45
## F-statistic: 34.54 on 1 and 40 DF,  p-value: 7.028e-07
```

```
Akers <- Akers %>%
  mutate(LinearPredictiona = fitted(linearDista))
ggplot() + geom_point(data=Akers,aes(x=Distance, y=pointsperkick),color="blue") + geom_point(data=Akers
```




```
ggplot() + geom_point(data=Akers,aes(x=Distance, y=total),color="blue") + geom_point(data=Akers,aes(x=Di
```



Similar visulations are seen with only Akers' kicks.