NHL Playoff Wins Analysis

Ethan Greig

3/4/2018

NHL Playoff Wins Analysis

First, determine which variables are most strongly correlated with playoff success (measured in # of playoff wins). This will be done on a 5 year and 10 year timeframe. Each metric will be represented as a difference wrt the league average of that metric during that season.

Since the correlation coefficient is location-scale invariant, we can directly compare how much these statistics correlate with playoff success, even when the statistics are percentages, counting totals, points-based, etc.

Independent Variables:

1. GF% (Corsica2)
2. AdjGF% (morehockeystats)
3. 5v5 GF% (Corsica1)
4. 5v5 xGF% (Corsica1)
5. 5v5 CF% (Corsica1)
6. Time_Leading (morehockeystats)
7. 5v5 Shooting% (Corsica1)
8. 5v5 Save% (Corsica1)
9. PP% (NHL.com)
10. PK% (NHL.com)
11. Regular Season Wins (NHL.com)
12. Regular Season Regulation+Overtime Wins (NHL.com)
    Note that 1-3 are measurements of how many goals were scored by the team vs how many goals were allowed by the team. 4-6 constitute a variety of indicators of how many scoring opportunities a team generates compared to other teams during the season. 7 measures how effective a team is at gaining and maintaining leads in hockey games. 8-9 are primarily measurements of luck, but can be skewed by relatively strong shooters or goalies. 10-11 are special teams indicators, which also have a luck component, and tend to be less important in the playoffs. 12-13 are overall indicators of regular season performance.

Dependent Variables:

1. Playoff Wins (NHL.com)

Import Data

```r
corsica1 <- read.csv("input_csvs/corsica_5v5_08-17.csv", check.names=FALSE)
corsica1$Season <- strtoi(substr(corsica1$Season, 6, 9))
corsica1 <- corsica1[, c('Team', 'Season', 'CF%', 'GF%', 'xGF%', 'Sh%', 'Sv%')]

corsica2 <- read.csv("input_csvs/corsica_all_08-17.csv", check.names=FALSE)
corsica2$Season <- strtoi(substr(corsica2$Season, 6, 9))
corsica2 <- corsica2[, c('Team', 'Season', 'allGF%', 'allGF', 'allGA')]
```

```r
morestats <- read.csv("input_csvs/misc_stats_playoffwins_08-17.csv", check.names=FALSE)
morestats <- morestats[, c('Team', 'Season', 'HDCF', 'Time Led', 'Playoff Wins', 'GF_EN', 'GA_EN', 'PP%

raw_data <- merge(merge(corsica1, corsica2, by=c("Team", "Season")), morestats, by=c("Team", "Season"))
raw_data$`AdjGF%` <- with(raw_data, round(100*(allGF-GF_EN)/(allGF-GF_EN+allGA-GA_EN),2))
raw_data <- raw_data[, c('Team', 'Season', 'allGF%', 'AdjGF%', 'GF%', 'xGF%', 'CF%', 'HDCF', 'Time Led'
colnames(raw_data) <- c('Team', 'Season', 'GF%', 'AdjGF%', '5v5_GF%', '5v5_xGF%', '5v5_CF%', '5v5_HDCF%

head(raw_data)
```

```
##   Team Season   GF% AdjGF% 5v5_GF% 5v5_xGF% 5v5_CF% 5v5_HDCF% Time_Led
## 1  ANA   2008 51.71  51.76   52.97    48.73   50.71     49.45    18.70
## 2  ANA   2009 50.32  50.87   52.73    51.10   50.96     53.54    19.13
## 3  ANA   2010 48.95  49.13   50.34    45.83   47.35     45.10    20.77
## 4  ANA   2011 50.21  50.66   46.53    44.08   44.35     44.88    20.58
## 5  ANA   2012 47.29  47.56   47.26    47.13   48.54     47.70    24.53
## 6  ANA   2013 53.82  52.97   55.28    50.07   47.93     48.77    19.98
##   5v5_Sh% 5v5_Sv%  PP%  PK% Wins ROWins Playoff_Wins
## 1    7.30   93.62 16.6 83.1   47     39            2
## 2    8.40   92.42 23.6 79.7   42     35            7
## 3    8.23   92.75 21.0 79.3   39     34           -1
## 4    7.87   92.31 23.5 81.3   47     43            2
## 5    7.99   91.66 16.6 82.0   34     31           -1
## 6    8.59   93.01 21.5 81.5   30     24            3
```

Finalize Data

```r
adj_data <- raw_data
adj_data_10 <- adj_data[adj_data$`Playoff_Wins`>=0,]
head(adj_data_10)
```

```
##   Team Season   GF% AdjGF% 5v5_GF% 5v5_xGF% 5v5_CF% 5v5_HDCF% Time_Led
## 1  ANA   2008 51.71  51.76   52.97    48.73   50.71     49.45    18.70
## 2  ANA   2009 50.32  50.87   52.73    51.10   50.96     53.54    19.13
## 4  ANA   2011 50.21  50.66   46.53    44.08   44.35     44.88    20.58
## 6  ANA   2013 53.82  52.97   55.28    50.07   47.93     48.77    19.98
## 7  ANA   2014 56.44  56.17   58.44    51.45   49.80     52.62    23.92
## 8  ANA   2015 50.78  50.35   51.47    52.13   50.96     52.19    21.32
##   5v5_Sh% 5v5_Sv%  PP%  PK% Wins ROWins Playoff_Wins
## 1    7.30   93.62 16.6 83.1   47     39            2
## 2    8.40   92.42 23.6 79.7   42     35            7
## 4    7.87   92.31 23.5 81.3   47     43            2
## 6    8.59   93.01 21.5 81.5   30     24            3
## 7    9.83   92.62 16.0 82.2   54     51            7
## 8    8.48   91.82 15.7 81.0   51     43           11
```

```r
adj_data_5 <- adj_data_10[adj_data_10$Season>2012,]
head(adj_data_5)
```

```
##    Team Season   GF% AdjGF% 5v5_GF% 5v5_xGF% 5v5_CF% 5v5_HDCF% Time_Led
## 6   ANA   2013 53.82  52.97   55.28    50.07   47.93     48.77    19.98
## 7   ANA   2014 56.44  56.17   58.44    51.45   49.80     52.62    23.92
## 8   ANA   2015 50.78  50.35   51.47    52.13   50.96     52.19    21.32
```

```
## 9   ANA   2016 53.35  52.59   49.42    53.01   52.42     52.22    23.55
## 10  ANA   2017 52.76  52.48   53.11    52.02   49.67     52.26    22.53
## 30  BOS   2013 54.51  54.30   55.42    53.24   54.39     54.22    24.35
##    5v5_Sh% 5v5_Sv%  PP%  PK% Wins ROWins Playoff_Wins
## 6     8.59   93.01 21.5 81.5   30     24            3
## 7     9.83   92.62 16.0 82.2   54     51            7
## 8     8.48   91.82 15.7 81.0   51     43           11
## 9     6.69   92.39 23.1 87.2   46     43            3
## 10    7.77   92.98 18.7 84.7   46     43           11
## 30    7.31   93.18 14.8 87.1   28     24           14
```

Correlations With Playoff Wins

```
stats <- colnames(adj_data_10)[4:ncol(adj_data_10)-1]
correlations_5_10 <- data.frame(matrix(ncol=13, nrow=2))
colnames(correlations_5_10) <- stats
rownames(correlations_5_10) <- c("5 years", "10 years")

correlations_5_10[1,] <- cor(adj_data_5[,4:ncol(adj_data_5)-1], adj_data_5$`Playoff_Wins`)
correlations_5_10[2,] <- cor(adj_data_10[,4:ncol(adj_data_10)-1], adj_data_10$`Playoff_Wins`)

write.csv(correlations_5_10, 'output_csvs/correlations.csv')
correlations_5_10
```

```
##                GF%    AdjGF%   5v5_GF%  5v5_xGF%   5v5_CF% 5v5_HDCF%
## 5 years  0.3458189 0.3358988 0.2745515 0.2436887 0.2310239 0.2050112
## 10 years 0.3416835 0.3317128 0.2426668 0.2060818 0.1818608 0.1710590
##            Time_Led     5v5_Sh%   5v5_Sv%         PP%       PK%      Wins
## 5 years  0.06812042 -0.07465011 0.1704371 -0.10292430 0.2153303 0.1373872
## 10 years 0.20747087 -0.03171249 0.1122820  0.06001718 0.2038975 0.1417614
##             ROWins
## 5 years  0.1154794
## 10 years 0.1488450
```

Correlations With Each Other

```
# correlations_5 <- cor(adj_data_5[,3:ncol(adj_data_5)])
# correlations_10 <- cor(adj_data_10[,4:ncol(adj_data_10)-1])
#
# eigenspace <- eigen(correlations_10) ## 6 or 9 factors
# n <- 6
# C <- as.matrix(eigenspace$vectors[,1:n])
# D <- matrix(0, dim(C)[2], dim(C)[2])
# diag(D) <- eigenspace$values[1:n]
# loadings <- C %*% sqrt(D)
# loadings
#
# S.h2 <- rowSums(loadings^2)
# S.h2
#
# S.u2 <- diag(correlations_10) - S.h2
```

```r
# S.u2
```

Calculate Differences in Each Stat for Every Playoff Matchup

```r
series <- read.csv("input_csvs/all_playoff_series.csv")
series$Home_Won <- round(series$Home_W., 0)

stats <- c('GF%', '5v5_xGF%', 'PK%', 'Time_Led', 'ROWins')
dStats <- stats
for (i in 1:length(stats)) {
  stat <- stats[i]
  dStat <- paste0('d', stat)
  dStats[i] <- dStat
  series[dStat] <- NA
}

regular_season_data <- adj_data_10[, c('Team', 'Season', stats)]

getDifferences <- function(row, df) {
  df_year <- df[df$Season==row$Year,]
  a1 <- df_year[as.character(df_year$Team)==as.character(row$Home),]
  a2 <- df_year[as.character(df_year$Team)==as.character(row$Away),]
  for (i in 1:length(stats)) {
    stat <- stats[i]
    dStat <- dStats[i]
    row[dStat] <- a1[[stat]] - a2[[stat]]
  }
  return(row)
}

for (row in 1:nrow(series)) {
  series[row,] <- getDifferences(series[row,], regular_season_data)
}

head(series)
```

```
##   Year Home Away Home_W. Home_Won  dGF% d5v5_xGF% dPK% dTime_Led dROWins
## 1 2008  ANA  DAL    0.33        0 -2.03     -0.89 -2.5     -3.52      -1
## 2 2009  DET  ANA    0.57        1  4.28      1.46 -1.4      5.17      10
## 3 2010  ARI  DET    0.43        0 -0.02     -0.78  0.6      0.70      -2
## 4 2011  ANA  NSH    0.33        0 -2.64     -6.48 -3.6     -2.45       5
## 5 2012  ARI  CHI    0.67        1  0.92     -2.27  7.4     -2.30      -2
## 6 2013  ANA  DET    0.43        0  1.23     -2.32 -0.2     -0.64       2
```

```r
colMeans(series[,dStats])
```

```
##      dGF% d5v5_xGF%      dPK% dTime_Led   dROWins
## 2.3687333 0.5541333 0.4493333 1.8223333 4.6400000
```

Logistic Regression

The binary model runs regression only on whether the higher-seeded team won or lost. The weighted model is an abuse of glm because it uses non-integer success variables (win% in the series)

```
regression_binary <- function(dataset) {
  return(suppressWarnings(glm(`Home_Won` ~ `dGF%` + `d5v5_xGF%` + `dPK%` + `dTime_Led` + `dROWins`,data=
}

regression_weighted <- function(dataset) {
  return(suppressWarnings(glm(`Home_W.` ~ `dGF%` + `d5v5_xGF%` + `dPK%` + `dTime_Led` + `dROWins`,data=d
}

binary_model <- regression_binary(series)
weighted_model <- regression_weighted(series)
summary(binary_model)
```

```
##
## Call:
## glm(formula = Home_Won ~ `dGF%` + `d5v5_xGF%` + `dPK%` + dTime_Led +
##     dROWins, family = binomial(link = "logit"), data = dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1695  -1.0734   0.5740   0.9754   2.0140
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.21550    0.24496  -0.880  0.37901
## `dGF%`       0.29415    0.09988   2.945  0.00323 **
## `d5v5_xGF%`  0.08469    0.05674   1.493  0.13555
## `dPK%`       0.08976    0.05551   1.617  0.10586
## dTime_Led   -0.02325    0.08080  -0.288  0.77357
## dROWins     -0.05652    0.05089  -1.110  0.26679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 206.23  on 149  degrees of freedom
## Residual deviance: 181.49  on 144  degrees of freedom
## AIC: 193.49
##
## Number of Fisher Scoring iterations: 3
```

```
write.csv(binary_model$coefficients, 'output_csvs/initial_model.csv')
```

Cross Validation

```
logLoss <- function(pred, actual){
  -1*mean(log(pred[model.matrix(~ actual + 0) - pred > 0]))
}
```

```r
mse <- function(pred, actual){
  mean((pred-actual)^2)
}

ten_fold_cross_validate <- function(dataset){
  series_shuffled <- dataset[sample(nrow(dataset)),]
  folds <- cut(seq(1,nrow(series_shuffled)),breaks=10,labels=FALSE)

  log_loss_binary <- NA
  log_loss_weighted <- NA
  mse_binary <- NA
  mse_weighted <- NA

  for (i in 1:10) {
    testRows <- which(folds==i,arr.ind=TRUE)
    test_data <- series_shuffled[testRows,]
    train_data <- series_shuffled[-testRows,]

    fold_binary_model <- regression_binary(train_data)
    fold_weighted_model <- regression_weighted(train_data)

    test_data$bin_pred <- predict(fold_binary_model, test_data[,dStats], type='response')
    test_data$wgt_pred <- predict(fold_weighted_model, test_data[,dStats], type='response')

    log_loss_binary[i] <- logLoss(test_data$bin_pred, test_data$Home_Won)
    log_loss_weighted[i] <- logLoss(test_data$wgt_pred, test_data$Home_Won)
    mse_binary[i] <- mse(test_data$bin_pred, test_data$Home_Won)
    mse_weighted[i] <- mse(test_data$wgt_pred, test_data$Home_Won)
  }
  return(c(mean(log_loss_binary), mean(log_loss_weighted), mean(mse_binary), mean(mse_weighted)))
}

ten_fold_metrics <- matrix(NA, nrow=200, ncol=4)
colnames(ten_fold_metrics) <- c("Log Loss Binary", "Log Loss weighted", "Mean Squared Error Binary", "Me
for(j in 1:200) {
  ten_fold_metrics[j,] <- ten_fold_cross_validate(series)
}

colMeans(ten_fold_metrics)
```

```
##             Log Loss Binary           Log Loss weighted
##                   0.5596033                   0.5941415
##   Mean Squared Error Binary Mean Squared Error weighted
##                   0.2262093                   0.2310608
```

Binary-trained model always has lower log-loss, but that metric rewards conservative (40% to 60%) predictions.
2016 Comparison

```r
seasons_for_2016 <- series[series$Year<2016,]
binary_model_2016 <- glm(`Home_Won` ~ `dGF%` + `d5v5_xGF%` + `dPK%` + `dROWins`, data=seasons_for_2016,

playoffs_2016 <- series[series$Year==2016,]
playoffs_2016$bin_pred <- predict(binary_model_2016, playoffs_2016, type='response')

summary(binary_model_2016)
```

```
##
## Call:
## glm(formula = Home_Won ~ `dGF%` + `d5v5_xGF%` + `dPK%` + dROWins,
##     family = binomial(link = "logit"), data = seasons_for_2016)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1770  -1.0267   0.4940   0.9986   1.7497
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.45420    0.28561  -1.590  0.11176
## `dGF%`       0.31137    0.10725   2.903  0.00369 **
## `d5v5_xGF%`  0.05215    0.06022   0.866  0.38650
## `dPK%`       0.11176    0.06390   1.749  0.08030 .
## dROWins     -0.03327    0.05468  -0.608  0.54293
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 165.52  on 119  degrees of freedom
## Residual deviance: 140.87  on 115  degrees of freedom
## AIC: 150.87
##
## Number of Fisher Scoring iterations: 4
```

```r
playoffs_2016[c('Home', 'Away', 'Home_Won', 'bin_pred')]
```

```
##     Home Away Home_Won  bin_pred
## 9    ANA  NSH        0 0.6518886
## 19   STL  CHI        1 0.5548607
## 29   DAL  STL        0 0.3689273
## 39   DAL  MIN        1 0.6356171
## 49   T.B  DET        1 0.7555869
## 59   FLA  NYI        0 0.3930856
## 69   L.A  S.J        0 0.4015320
## 79   S.J  NSH        1 0.5072879
## 89   T.B  NYI        1 0.4749745
## 99   PIT  NYR        1 0.8140909
## 109  WSH  PHI        1 0.8478197
## 119  WSH  PIT        0 0.4244600
## 129  PIT  T.B        1 0.5772182
## 139  PIT  S.J        1 0.5948153
## 149  STL  S.J        0 0.4036199
```

```r
logLoss(playoffs_2016$bin_pred, playoffs_2016$Home_Won)
```

```
## [1] 0.4650487
```

```r
mse(playoffs_2016$bin_pred, playoffs_2016$Home_Won)
```

```
## [1] 0.1686403
```

FACTOR ANALYSIS

Calculate Differences in Each Stat for Every Playoff Matchup

```
series <- read.csv("input_csvs/all_playoff_series.csv")
series$Home_Won <- round(series$Home_W., 0)

stats <- c('GF%', 'AdjGF%', '5v5_xGF%', '5v5_CF%', 'Time_Led', '5v5_Sh%', '5v5_Sv%', 'PP%', 'PK%', 'ROW
dStats <- stats
for (i in 1:length(stats)) {
  stat <- stats[i]
  dStat <- paste0('d', stat)
  dStats[i] <- dStat
  series[dStat] <- NA
}

regular_season_data <- adj_data_10[, c('Team', 'Season', stats)]

getDifferences <- function(row, df) {
  df_year <- df[df$Season==row$Year,]
  a1 <- df_year[as.character(df_year$Team)==as.character(row$Home),]
  a2 <- df_year[as.character(df_year$Team)==as.character(row$Away),]
  for (i in 1:length(stats)) {
    stat <- stats[i]
    dStat <- dStats[i]
    row[dStat] <- a1[[stat]] - a2[[stat]]
  }
  return(row)
}

for (row in 1:nrow(series)) {
  series[row,] <- getDifferences(series[row,], regular_season_data)
}

head(series)
```

```
##   Year Home Away Home_W. Home_Won  dGF% dAdjGF% d5v5_xGF% d5v5_CF%
## 1 2008  ANA  DAL    0.33        0 -2.03   -2.25     -0.89    -1.00
## 2 2009  DET  ANA    0.57        1  4.28    3.46      1.46     6.35
## 3 2010  ARI  DET    0.43        0 -0.02    0.95     -0.78    -2.05
## 4 2011  ANA  NSH    0.33        0 -2.64   -1.28     -6.48    -4.79
## 5 2012  ARI  CHI    0.67        1  0.92    1.90     -2.27    -3.16
## 6 2013  ANA  DET    0.43        0  1.23    0.76     -2.32    -5.64
##   dTime_Led d5v5_Sh% d5v5_Sv% dPP% dPK% dROWins
## 1     -3.52    -1.59     1.94 -1.5 -2.5      -1
## 2      5.17    -0.37    -0.94  1.9 -1.4      10
## 3      0.70     1.15     1.00 -4.6  0.6      -2
## 4     -2.45    -0.30    -1.08  8.3 -3.6       5
## 5     -2.30    -0.08     2.08 -1.7  7.4      -2
## 6     -0.64     1.94    -0.45  3.1 -0.2       2
```

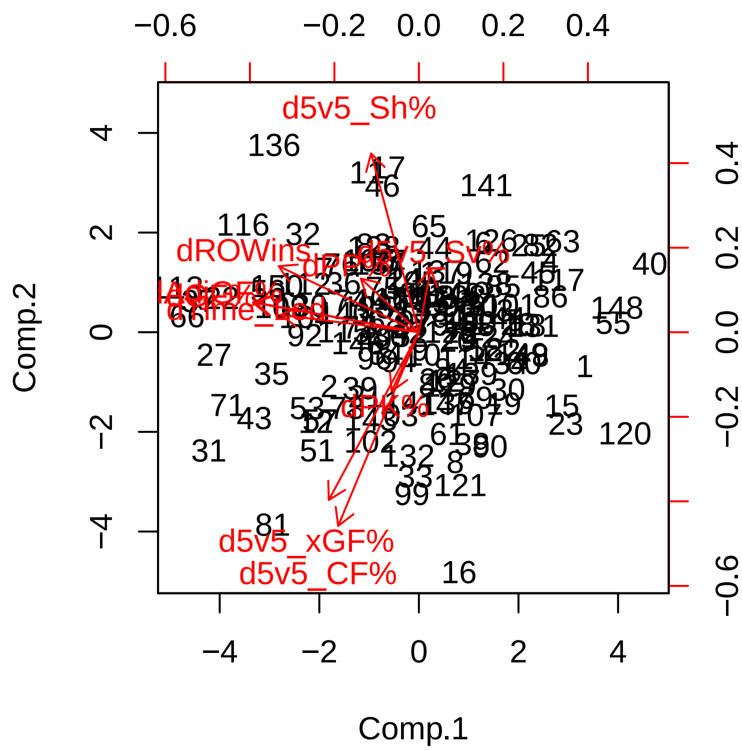
```
colMeans(series[,dStats])
```

```
##      dGF%   dAdjGF% d5v5_xGF%  d5v5_CF% dTime_Led  d5v5_Sh%  d5v5_Sv%
```

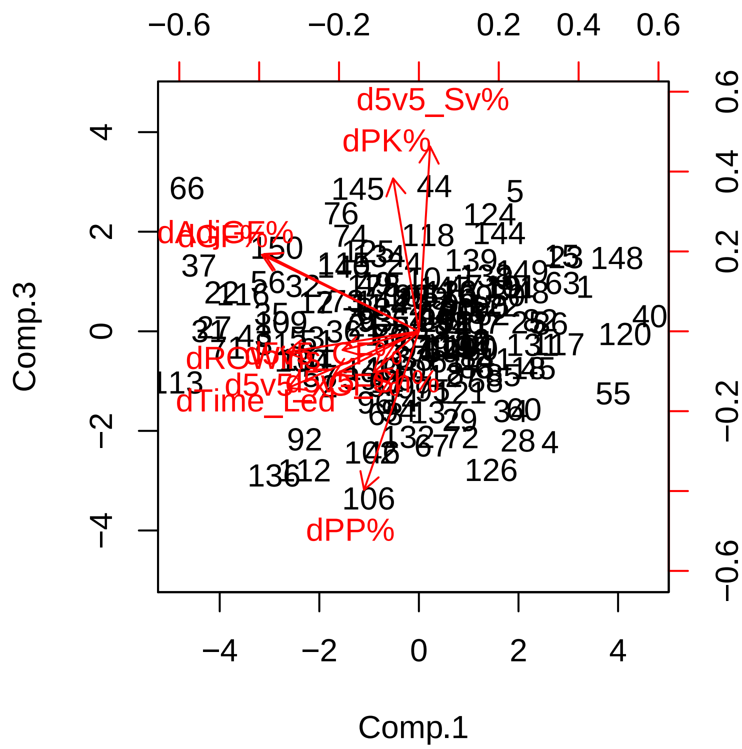
```
## 2.3687333 2.2268000 0.5541333 0.4273333 1.8223333 0.4026667 0.1891333
##      dPP%      dPK%   dROWins
## 0.9120000 0.4493333 4.6400000
```

Factor Analysis

```r
dStat_table <- series[,dStats]
pca <- princomp(dStat_table, cor=TRUE)

biplot(pca,choices=c(1,2),scale=0)
```




```r
biplot(pca,choices=c(1,3),scale=0)
```




```r
prin_data <- series[,c('Year', 'Home', 'Away', 'Home_W.', 'Home_Won')]
prin_data <- cbind(prin_data, pca$scores)
prin_data
```

```
##      Year Home Away Home_W. Home_Won      Comp.1       Comp.2       Comp.3
## 1    2008  ANA  DAL    0.33        0  3.33137969 -0.675487442  0.899279981
## 2    2009  DET  ANA    0.57        1 -1.80309521 -1.078059899 -1.100219655
## 3    2010  ARI  DET    0.43        0  1.72804151  0.437142489  1.059109264
## 4    2011  ANA  NSH    0.33        0  2.63410295  1.423305261 -2.221396030
## 5    2012  ARI  CHI    0.67        1  1.93141678  0.235265070  2.810133113
## 6    2013  ANA  DET    0.43        0  1.29309473  1.790010634 -0.724809223
## 7    2014  ANA  DAL    0.67        1 -1.89700832  1.276124736  0.577418751
## 8    2015  ANA  CGY    0.80        1  0.73014833 -2.604865345 -0.850901268
## 9    2016  ANA  NSH    0.43        0  0.02224916 -0.350206755  0.524543996
## 10   2017  ANA  CGY    1.00        1  0.09088942 -0.451716518  1.056582535
## 11   2008  MTL  BOS    0.57        1 -1.04194874  3.211576021  0.529538924
## 12   2009  S.J  ANA    0.33        0 -2.06793540 -1.795221340  0.588061673
## 13   2010  BUF  BOS    0.33        0 -0.73632575  1.447217394 -0.068390753
## 14   2011  DET  ARI    1.00        1 -0.35282168 -1.406359516 -1.387160969
## 15   2012  ARI  NSH    0.80        1  2.86955128 -1.474580022  1.514993341
## 16   2013  BOS  TOR    0.57        1  0.80568749 -4.816701206  0.797446770
## 17   2014  ANA  L.A    0.43        0 -0.62157972  3.311121482 -1.011292804
## 18   2015  ANA  WPG    1.00        1  2.19697093  0.137168579 -0.741146660
## 19   2016  STL  CHI    0.57        1  1.70351822 -1.434716084  0.882828547
## 20   2017  ANA  NSH    0.33        0  0.76330389 -0.171992878  0.438139235
## 21   2008  S.J  CGY    0.57        1 -1.07264095 -1.281756851  0.121001609
```

```
## 22  2009  BOS  MTL    1.00          1 -3.96086507  0.691133621  0.784871409
## 23  2010  BOS  PHI    0.43          0  2.94993629 -1.837191573  1.487977119
## 24  2011  BOS  MTL    0.57          1 -0.29333016  1.068748299  1.360630812
## 25  2012  ARI  L.A    0.20          0  2.20210509  1.743394051  0.177314093
## 26  2013  BOS  NYR    0.80          1  0.35979015 -0.950240340  0.206395830
## 27  2014  BOS  MTL    0.43          0 -4.11453363 -0.448554804  0.083547609
## 28  2015  ANA  CHI    0.43          0  1.98960753  0.161113369 -2.195823588
## 29  2016  DAL  STL    0.43          0  0.82166355 -0.120314351 -1.783346737
## 30  2017  ANA  EDM    0.57          1  1.78134190 -1.153885311  0.725336856
## 31  2008  DET  COL    1.00          1 -4.21842406 -2.375645365  0.003540189
## 32  2009  BOS  CAR    0.43          0 -2.32962036  1.969881083  0.908936731
## 33  2010  CHI  VAN    0.67          1 -0.07171977 -2.894831041 -0.570748841
## 34  2011  PHI  BOS    0.00          0  1.84291134 -0.620802438 -1.603421817
## 35  2012  BOS  WSH    0.43          0 -2.95852083 -0.833465144  0.349855614
## 36  2013  CHI  BOS    0.67          1 -1.51480523  0.981839137 -0.004652587
## 37  2014  BOS  DET    0.80          1 -4.41889840  0.625096796  1.319623460
## 38  2015  VAN  CGY    0.33          0  1.06959724 -2.244179702 -0.252301286
## 39  2016  DAL  MIN    0.67          1 -1.18414478 -1.124931522 -0.954981313
## 40  2017  OTT  BOS    0.67          1  4.63820533  1.374343901  0.302015838
## 41  2008  MIN  COL    0.33          0  1.05583496  0.168727727  0.073139006
## 42  2009  CHI  CGY    0.67          1 -0.21495987  0.002764835  0.546007566
## 43  2010  CHI  NSH    0.67          1 -3.30510825 -1.725578995 -0.091935463
## 44  2011  BOS  T.B    0.57          1  0.31041277  1.696060095  2.920709683
## 45  2012  NSH  DET    0.80          1  2.39763684  1.197578014 -0.742306406
## 46  2013  PIT  BOS    0.00          0 -0.73200083  2.940545401 -2.432091553
## 47  2014  STL  CHI    0.33          0  0.92496000  0.282358131  0.918403087
## 48  2015  CHI  MIN    1.00          1  2.26185948 -0.469475781  0.783252080
## 49  2016  T.B  DET    0.80          1 -0.88932483 -0.134860827  0.965418850
## 50  2017  CHI  NSH    0.00          0  1.08184992  0.790864362 -0.174757759
## 51  2008  DET  DAL    0.67          1 -2.03965639 -2.372810903 -0.197403394
## 52  2009  N.J  CAR    0.43          0 -0.16668466 -0.074839248  0.071818637
## 53  2010  CHI  PHI    0.67          1 -2.24306445 -1.523838919 -0.128171277
## 54  2011  VAN  BOS    0.43          0 -0.66365278 -0.431486059 -1.096477242
## 55  2012  FLA  N.J    0.43          0  3.90073079  0.181905713 -1.252955449
## 56  2013  CHI  DET    0.57          1 -3.03038562  0.825916635  0.987748449
## 57  2014  CHI  MIN    0.67          1 -1.98616164 -1.774140803 -0.902281658
## 58  2015  NSH  CHI    0.33          0  1.34076284 -0.340656699 -1.001708148
## 59  2016  FLA  NYI    0.33          0  0.46324187  0.550969917 -0.400275207
## 60  2017  PIT  CBJ    0.80          1  2.10780534 -0.621721952 -1.570195454
## 61  2008  S.J  DAL    0.33          0  0.56988620 -2.042810886 -0.604474690
## 62  2009  PIT  CAR    1.00          1  1.47296516  1.414977763  0.341241249
## 63  2010  S.J  CHI    0.00          0  2.88518521  1.824356405  0.967937316
## 64  2011  PHI  BUF    0.57          1 -0.19288844  0.681900462  0.224938827
## 65  2012  VAN  L.A    0.20          0  0.20516424  2.129736302  0.061192810
## 66  2013  CHI  MIN    0.80          1 -4.65659138  0.320222629  2.875375392
## 67  2014  CHI  L.A    0.43          0  0.26135917  0.701347495 -2.295093867
## 68  2015  T.B  CHI    0.33          0 -0.66902280  0.214148280 -1.662622333
## 69  2016  L.A  S.J    0.20          0  1.13372853 -1.316441765  0.699605950
## 70  2017  EDM  S.J    0.67          1  1.23592785  0.523725391 -0.361417393
## 71  2008  DET  NSH    0.67          1 -3.84326521 -1.456033460 -0.324581623
## 72  2009  DET  CHI    0.80          1  0.85183297 -0.007820049 -2.121254155
## 73  2010  S.J  COL    0.67          1 -1.44924073 -1.519151251  0.605213081
## 74  2011  VAN  CHI    0.57          1 -1.37367634  0.639408334  1.922824575
## 75  2012  STL  L.A    0.00          0 -0.71757136  0.884720520  0.909734274
```

```
## 76  2013  CHI  L.A    0.80        1 -1.56244776  1.359823807  2.368431159
## 77  2014  COL  MIN    0.43        0 -0.60787296  1.434659760 -0.071169885
## 78  2015  T.B  DET    0.57        1 -1.19055685  0.092110677  0.245011310
## 79  2016  S.J  NSH    0.57        1  0.25869907  0.153449256 -0.293048372
## 80  2017  MIN  STL    0.20        0 -0.58513665 -0.073874982 -0.094752612
## 81  2008  DET  PIT    0.67        1 -2.93581780 -3.863735815 -0.259422834
## 82  2009  VAN  CHI    0.33        0  2.43911127  1.750421995  0.218855016
## 83  2010  S.J  DET    0.80        1 -0.90902853  1.757415144  0.446882630
## 84  2011  S.J  DET    0.57        1  0.65718519 -0.749033556  0.139816743
## 85  2012  N.J  L.A    0.33        0  1.69455209  0.918832748 -0.882122703
## 86  2013  STL  L.A    0.33        0  2.64343476  0.717078283  0.154886522
## 87  2014  PIT  CBJ    0.67        1 -0.15833882  0.144124100 -0.509241409
## 88  2015  STL  MIN    0.33        0  1.16563323  0.144704261 -0.724087762
## 89  2016  T.B  NYI    0.80        1  0.34739766 -1.054013903  0.239494405
## 90  2017  MTL  NYR    0.33        0  1.42949296 -2.292230068  0.473691949
## 91  2008  MTL  PHI    0.20        0 -0.03684089  0.772866327 -0.335921531
## 92  2009  DET  CBJ    1.00        1 -2.29367793 -0.061575614 -2.172207759
## 93  2010  VAN  L.A    0.67        1 -0.88046306  0.501100624  0.211162035
## 94  2011  S.J  L.A    0.67        1 -0.40446268 -0.580621745 -1.598834564
## 95  2012  PHI  N.J    0.20        0  0.27288484  0.602344827 -1.185630133
## 96  2013  L.A  S.J    0.57        1 -0.88632994 -0.526605777 -1.431554686
## 97  2014  S.J  L.A    0.43        0 -0.08150886  0.216153690 -1.125484098
## 98  2015  MTL  OTT    0.67        1  0.80633248  0.528917186  0.531414889
## 99  2016  PIT  NYR    0.80        1 -0.14038893 -3.248090019  0.715128734
## 100 2017  STL  NSH    0.33        0  1.03200562  0.196819377 -0.298484103
## 101 2008  PIT  OTT    1.00        1  1.83669983  0.543930543  0.897210590
## 102 2009  DET  PIT    0.43        0 -0.97068933 -2.196865697 -2.436678885
## 103 2010  PHI  MTL    0.80        1 -0.54321229 -1.665687774 -0.742268068
## 104 2011  VAN  NSH    0.67        1 -2.27730552  0.303871215 -0.575475782
## 105 2012  NYR  N.J    0.33        0 -0.13481290  0.998016628  0.716008352
## 106 2013  MTL  OTT    0.20        0 -1.00879980  1.440418106 -3.359077640
## 107 2014  L.A  NYR    0.80        1  1.13558260 -1.661077557  0.170889520
## 108 2015  WSH  NYI    0.57        1  0.81835863  0.718788847  0.372880770
## 109 2016  WSH  PHI    0.67        1 -2.76839631  0.503967032  0.178546307
## 110 2017  PIT  NSH    0.67        1 -0.38574949  0.667462793 -0.791702583
## 111 2008  PIT  PHI    0.80        1  0.89494771  0.373533432  0.180070937
## 112 2009  WSH  NYR    0.57        1 -2.29757522  0.975642469 -2.788549518
## 113 2010  WSH  MTL    0.43        0 -4.85896947  0.894899806 -1.029785451
## 114 2011  WSH  NYR    0.80        1  0.92086223 -0.514687559 -0.278408698
## 115 2012  NYR  WSH    0.57        1 -1.51147056  0.015249735  1.359303336
## 116 2013  PIT  NYI    0.67        1 -3.53171512  2.158487803  0.741324883
## 117 2014  MTL  NYR    0.33        0  2.79797094  1.054231713 -0.255702821
## 118 2015  NYR  WSH    0.57        1  0.18465752  0.713757117  1.930901366
## 119 2016  WSH  PIT    0.33        0  0.55145314  1.191616038 -0.306593735
## 120 2017  OTT  NYR    0.67        1  4.13792145 -2.031660605 -0.057334198
## 121 2008  WSH  PHI    0.43        0  0.83112115 -3.062960515 -1.225477920
## 122 2009  PIT  PHI    0.67        1  1.51104071 -0.355724018  0.522936653
## 123 2010  N.J  PHI    0.20        0  0.63578663  0.007321301  0.787857487
## 124 2011  PIT  T.B    0.43        0  1.42074932 -0.084509562  2.353025144
## 125 2012  NYR  OTT    0.57        1 -1.02280715  0.498528497  1.603315123
## 126 2013  WSH  NYR    0.43        0  1.45181464  1.839759738 -2.786071958
## 127 2014  T.B  MTL    0.00        0  0.70218300 -0.969974931 -0.893639240
## 128 2015  NYR  PIT    0.80        1 -0.90630829  1.702819232  0.915544416
## 129 2016  PIT  T.B    0.57        1  0.62086075 -1.072768548 -0.338559269
```

```
## 130 2017  PIT  OTT    0.57        1 -2.36758043  0.530587065 -0.590407894
## 131 2008  N.J  NYR    0.20        0  2.28498609  0.125247302 -0.273298255
## 132 2009  WSH  PIT    0.43        0 -0.21434865 -2.480984174 -2.122385067
## 133 2010  PIT  OTT    0.67        1  0.05613531  0.427129112  0.655766961
## 134 2011  VAN  S.J    0.80        1 -0.80326874  1.374944102  1.509920055
## 135 2012  PIT  PHI    0.33        0  0.56703495 -1.422554496  0.346411640
## 136 2013  PIT  OTT    0.80        1 -2.90921558  3.757251673 -2.896961279
## 137 2014  PIT  NYR    0.43        0  0.36011914  1.218383009 -1.629296740
## 138 2015  NYR  T.B    0.43        0  1.34484058  0.969261849  1.116444069
## 139 2016  PIT  S.J    0.67        1  1.04918795 -0.837021356  1.426606402
## 140 2017  WSH  PIT    0.43        0 -1.51269288  0.384289711  1.296221989
## 141 2008  PIT  NYR    0.80        1  1.35326458  2.956291384 -0.557274221
## 142 2009  VAN  STL    1.00        1 -0.74442531  0.656626811  0.600682350
## 143 2010  PIT  MTL    0.43        0 -0.96841717 -1.753294475 -0.760019105
## 144 2011  WSH  T.B    0.00        0  1.61296221  0.288192726  1.975801282
## 145 2012  STL  S.J    0.80        1 -1.22655677 -0.222327378  2.860598663
## 146 2013  VAN  S.J    0.00        0  0.71591190  0.765387258 -0.369289329
## 147 2014  NYR  PHI    0.57        1  0.51642727 -1.438388743  0.880245937
## 148 2015  MTL  T.B    0.33        0  3.97702386  0.491787016  1.470717256
## 149 2016  STL  S.J    0.33        0  2.07028338 -0.442303738  1.202996010
## 150 2017  WSH  TOR    0.67        1 -2.85491131  0.924726192  1.676129690
##             Comp.4      Comp.5       Comp.6       Comp.7        Comp.8
## 1   -1.8916454436 -0.24904837  0.404419556  0.207760688  0.4628860895
## 2   -0.2020516932 -0.03567764 -0.142807370  1.089038897 -0.7589920727
## 3    0.5452701547 -1.14296517 -0.362378564 -0.354824253  0.1062542435
## 4   -0.7714358006  1.82740537  0.578836091  0.355108796 -0.6447817974
## 5    0.0520624891  0.99460742 -0.317495139 -0.844021226  0.3873847848
## 6    0.7793908532  0.34474720  0.352000748 -0.850174719 -0.0949799002
## 7    0.1922285156 -0.02144567  1.155401021  0.651421831  0.1691365483
## 8    0.6041616209 -1.10341940 -0.703928983  0.879009070  0.7866094676
## 9    0.2210451838  1.78490918 -0.495750468  0.294779030  0.3252831033
## 10   0.2454827752 -0.29772219 -0.443929153 -0.354345492  0.9248955517
## 11   0.4823715793  1.43439995 -0.572399584 -0.796994760 -0.1077182446
## 12  -0.2785168532  0.79782865  0.431742467  0.805438947  0.7141519452
## 13   0.3377952256  0.20955553  0.604504790  0.753547743  0.1079863702
## 14   0.8385928214  1.75106534  0.598294179 -0.747366745 -0.1793811249
## 15   0.5872958237 -1.02981078 -0.630325142  0.038486080 -0.9966238484
## 16  -1.6561342680 -0.93242385  0.535246583 -0.193607806  0.2035836822
## 17   1.6792726896 -0.18222939  0.481273028  0.794958243  0.0837094959
## 18   0.8610050754  0.13616584  1.574155212  0.841214111 -0.1489061688
## 19   0.7108491151  0.34940216  0.719790225 -1.268981732  0.3874913935
## 20   0.5491435041  0.54028935 -0.538613765  0.405552505  0.7742693822
## 21   0.4993716325  0.77409845 -0.616063600 -0.245629024  0.3829124527
## 22  -1.3874084828 -0.19283577  0.055645127  0.162601571  0.6282392512
## 23   0.3553727076 -0.14898572 -0.075041096 -0.908451685 -0.5526141364
## 24  -0.2797516289 -1.45050349  0.385416052 -0.506486582 -0.5040755322
## 25   0.9072991664 -0.81069799  0.678462058 -0.020120849 -0.3371419241
## 26   1.2951096813  0.60766459 -1.746196327  0.849924157  0.1293821434
## 27  -1.2113139359 -0.72031545  0.283408779 -0.629082356 -0.1383449288
## 28   1.8271917000 -0.92301048  0.352083982  0.692852955  0.6387355619
## 29   0.7794420994 -0.42604965  0.343372077  0.335171285 -0.4786300199
## 30   0.9210768809 -0.09004316 -0.741560975  0.554524480  0.6888458289
## 31  -1.4603534999  0.68857405 -0.927081992  0.537061165  0.2107753469
## 32  -0.4486025930 -0.03483784 -0.880224980 -0.865966061  0.4988722660
```

```
## 33   1.7772764682 -0.35221831 -0.988590334 -0.102741663 -0.1561326943
## 34   1.3104175179 -0.02476257 -0.235131356 -0.014804465  0.3441232595
## 35   0.4611548964 -0.92134075 -1.381810109 -0.838961551 -0.3776036553
## 36   0.4257597154 -0.18092368  0.271899886 -0.690121045 -0.6592239235
## 37  -1.5862030420  0.02851493  0.727832016  0.025269817  0.6186768654
## 38   0.6862457153  0.97068665 -0.944160156  0.575160719  0.0747627268
## 39   0.9444401967  1.54868389  0.807099261  0.909240880 -0.0844073064
## 40  -0.8691364999 -1.34980721  0.070287272  0.430171821 -0.2317684874
## 41  -0.3718340640  1.70722815 -1.378084097  0.928273725 -0.2546464334
## 42  -1.7637538184 -0.53989355 -1.065736922 -0.681264306 -0.1327978015
## 43   2.3938435493  0.60793417 -1.184985854 -0.773717414 -0.2670833131
## 44  -1.5883777348 -0.84515495  0.105394516  0.261732190 -1.0592525152
## 45   0.0383611491  1.80792879  0.052867321  0.033519203  1.0787110624
## 46  -1.1606576514 -0.25834076  0.169431248 -0.502493820  0.4193227550
## 47   0.2711624355  0.69128844 -0.608652151  0.132684083  0.5759264625
## 48  -2.4288690926  0.19598853 -0.506404080 -0.068355020 -1.1467629243
## 49   0.8356866574 -0.66603731 -0.146537152 -0.231861350  0.0354790677
## 50  -0.6412098481 -0.51494753  0.238550818  1.118007880 -0.0612609743
## 51  -1.6583947623 -0.37537272  0.470065607  0.150615476  0.7262264863
## 52  -0.2439317549 -0.88763529 -0.679648858 -0.112845733  0.8024562310
## 53   1.7533414205 -0.95028286  0.009262853  0.088502167 -0.6205497594
## 54  -0.0076806325  1.77053026 -0.195182162 -0.421223399  0.0196658210
## 55  -2.7383159399 -1.15736965 -1.048501525  0.888132242 -0.3997950536
## 56   2.0278287701 -0.21664634 -0.434637734 -0.329907121 -0.5202940036
## 57   1.2843257459  0.12164432  0.179175958 -0.534915994 -0.7843782045
## 58   0.6475946431 -1.16105438  0.916721235 -0.488331522  0.5965921508
## 59  -0.5324048122 -1.79970680 -0.978374188  0.041268764 -0.0195873773
## 60  -0.0618100091 -0.16414739  0.492449213 -1.005220921  0.1005710982
## 61  -0.1644869046 -0.16937737  0.344439487 -0.156034522  1.1173007373
## 62   1.6489286194 -0.71347721 -0.399869577 -1.255528232  0.0743504015
## 63  -1.4790911216  0.66750912  0.179718337 -0.439334767  0.5172883170
## 64   0.5754343274 -0.78335616 -0.404763455  0.768685634 -0.0379431795
## 65  -0.0483511746  0.53850988  1.670197368 -0.341007689 -1.0780601193
## 66   0.2181316533 -0.29236799 -1.137541223 -0.538270077 -0.7185462257
## 67   1.3670799465  0.05821787  0.025427552 -0.782673262 -0.6155577583
## 68   1.8219773665 -0.36030759  0.667657263 -0.122065423  0.4750979020
## 69  -0.5026298973 -0.03774497  0.192164620  0.873053888 -0.4032897082
## 70  -0.6851903793  0.94384764  0.108701959 -1.383036768 -0.4442444250
## 71  -1.7572610737  0.06787137 -0.121305379 -0.012613764 -0.2887754577
## 72  -0.1599336725  0.50187707 -0.312847482 -0.016435184 -0.4911985247
## 73  -0.0919408253  0.98721383  0.756299532 -0.393049850 -0.0550146962
## 74  -0.0740060805  1.40650221  0.310408703  0.522880720  0.3182214492
## 75  -0.6020286347 -0.24577851  2.042430846 -0.248417257  0.1385267014
## 76   0.4945436793 -0.46737368 -0.634502024 -0.273343567 -0.1858565138
## 77   0.4509285531  0.86780938 -0.011244592  1.207350917 -0.1789010793
## 78   2.0775480303 -0.98686329 -0.028407262  0.590237470  0.4311031168
## 79  -0.3503737179  0.27489110  0.801246637 -0.518438076  0.0606939129
## 80  -0.0124511405 -1.13277094  0.012831809 -0.804853988  0.3407101654
## 81   0.0322002423  0.15864884  0.258321145  0.502590289 -0.5697890612
## 82   0.7142798377 -0.13347344 -0.251430786 -0.281909712  0.3438009163
## 83   0.2775862042 -0.33209569 -0.488543428 -0.385738904  0.4709604929
## 84  -1.4017927475 -0.79684323 -0.134377324 -0.671492984  0.7365041032
## 85   2.1612369128  0.42893401  0.458225835 -0.469175718 -0.0006492602
## 86   0.5324719379  0.51957022 -0.355107867  0.015237061 -0.2171830016
```

```
## 87   0.6047887635  1.35466468  0.281049329 -0.237838950 -0.2040136701
## 88  -1.5297413733  0.66145635 -0.193627981 -0.699578219 -0.6400538881
## 89  -0.0932724664 -0.75887305  0.055415839  0.251861940  0.0019273158
## 90  -0.7807681259 -0.11399090 -0.083934011 -0.271046906  0.4792375917
## 91  -0.0970636044 -0.07745571 -0.782152301 -0.052610890 -0.2621422371
## 92  -2.1849047557  1.23497351 -0.629680080  0.132874282 -0.6156802367
## 93   0.3419084990 -0.17324282  0.013626692  0.538339161  0.1791848352
## 94  -2.2593212248  0.02681831  0.498272721 -0.163447814  0.5051403021
## 95  -1.5026813360 -1.00977154  1.333743606 -0.032548092 -0.4185825154
## 96   0.7868459370 -0.58875179  0.055956531  0.938708408 -0.9103577704
## 97   1.4582144720  0.18460390 -0.557778515 -0.377516927  0.5664350890
## 98  -0.2394162355  0.29428105 -0.013630148  0.681443173  0.1283631352
## 99   0.5237958970  0.83579229  0.666356483 -0.982601692  0.3991462674
## 100  0.8178817489  1.13696032 -0.153971241  0.289510906  0.3505265239
## 101 -1.3342218907  0.45933333 -0.193230423 -0.545568003  0.8464771082
## 102 -1.8183695741  0.56472424 -0.350020855  0.446954371 -0.7557979845
## 103  0.4491853028 -0.44852892  0.419127255 -0.017641252  0.2444234316
## 104 -1.1262633319  1.65613302  0.607769485 -0.219702374 -0.6414476945
## 105 -1.0772475045 -0.62015828  1.493972352  0.597071622  0.2730431546
## 106  0.4520435761 -1.84094596 -0.596657330 -0.266904471  0.0350015909
## 107 -0.4163011251 -1.44610784 -0.029406451 -0.537937386 -0.2694436038
## 108 -0.6755742595  1.48116575 -0.849870989 -0.932019073 -0.1277781672
## 109  0.9910585281  0.96765234  0.613798117  0.131326515  0.2665105412
## 110 -0.2532858557  0.26082241  0.219910788 -0.131349451  0.2457629228
## 111 -0.2300189857 -1.28805857  0.007882045 -0.454389492  0.5417685989
## 112 -1.8143462370 -0.01007988  0.273028122  0.068717787  0.2091538491
## 113 -0.2812329101 -1.53770157  0.691127546  0.173429983 -0.1912486950
## 114  0.1018615619  0.82825108  0.055341252  1.353087356  0.0183126650
## 115  0.5718587639  0.53007670  0.491092686 -0.008784592  0.3092858541
## 116 -1.3636589456 -0.52737305 -1.267126195  0.382842819  0.9428193134
## 117  0.6283555096 -0.10817071 -0.501270429  0.123358931  0.2986777942
## 118  1.2116963473 -0.68459797  0.387314619  1.458186732 -0.0869725726
## 119  0.0003680733  1.10920437  0.461135183  0.418332352 -0.4889098844
## 120 -0.2546360366 -0.26831188 -0.734478409  0.260063977  0.3401733505
## 121  0.2845567584 -1.49916229  0.672223045 -0.103083711 -0.2633106060
## 122  0.7580906322 -1.50676232  0.142962617 -0.285439927  0.0688433040
## 123 -0.1494590207 -0.43330675  0.580782328  0.369710131 -0.1786323776
## 124 -0.4107630481 -0.13852656 -0.049863799 -0.075830354 -1.1287267969
## 125  0.7683046701  0.44688793  1.433785312  0.102141485  1.1544206087
## 126 -0.4991386055  1.19332056 -1.166103903 -0.188065899 -0.2261618749
## 127 -0.6836812092 -1.05627890  0.352722593 -1.052232023 -0.8804162860
## 128  0.2962648421 -0.56448219  0.698738138  0.472777088 -0.2493483089
## 129 -0.3304333055  0.16931418  0.281882831 -0.863875159  0.6040481151
## 130 -0.1610008837  0.15903497  0.253861960 -1.020161096  0.0010434599
## 131 -0.1882078037  0.10683738  0.914064497  0.539192468 -0.2396965762
## 132 -1.6869754573  1.01594301 -0.199168498  0.654695518  0.4965664845
## 133 -0.3942181062 -0.62464686 -0.775984755 -0.608789882 -0.2199118816
## 134  0.6803385385  1.24834858 -0.645747059  0.338352414 -0.7923592120
## 135  1.4107322636  0.90949129  0.013113195 -0.865568288 -0.7571506461
## 136 -0.4110933502 -1.38384139 -1.296242394  0.205594848  0.4571311196
## 137  0.8839573442  0.47543140 -0.370250538 -0.574056387  0.0316932501
## 138 -0.3471528565 -0.05322443 -0.901226536  0.661830458 -0.0670926442
## 139  0.4993829190 -0.05763321  0.088849782  0.170855412  0.1269966694
## 140  0.0838439293  0.16766030 -1.071751568  0.477471962 -0.0131629492
```

```
## 141 -0.4992784769  0.07269797  0.501693935 -0.196474397  0.3916468846
## 142 -0.6162674364 -1.32799092  0.080194644  0.351316523  0.0027947689
## 143  2.0563665575 -0.98137637  1.228162816  0.164048828 -0.3504494436
## 144 -0.5989294703  0.35443096  0.825046864  0.128952271 -0.7096980951
## 145  1.0107882120  0.94843155  0.885811973  0.529737782 -0.0233881470
## 146  1.1356398521 -1.14100096 -1.099809507  1.421073333 -0.5523801732
## 147 -0.8009071721 -0.12828152  0.173547559 -0.011042907 -0.0189111147
## 148 -1.0696557264  0.24701933 -0.325945878  0.103700850 -0.0895201882
## 149  0.0264591085  1.05816241 -0.131038477  0.472150258 -0.0663888390
## 150 -0.5863871558 -0.26433990  0.593617468  0.531565527 -0.1031547941
##           Comp.9      Comp.10
## 1   -0.035201590 -0.040647430
## 2    0.247254154  0.057257020
## 3   -0.259564445 -0.210404699
## 4    0.616158703 -0.376233102
## 5   -0.367732715 -0.216664954
## 6    0.086563179  0.202333293
## 7   -0.148900225 -0.010853277
## 8   -0.773825393 -0.036726607
## 9    0.025948369  0.012555487
## 10   0.023493251  0.052904641
## 11   0.128865096 -0.076546447
## 12   0.812798841  0.136433328
## 13   0.040766565 -0.232948509
## 14   0.048313828 -0.220235538
## 15  -0.181355793 -0.121809308
## 16   0.235352370  0.095402724
## 17   0.098054833 -0.129347936
## 18  -0.113569730  0.174620771
## 19  -0.661045130 -0.138788909
## 20   0.032915113  0.118708654
## 21   0.794227218 -0.223422848
## 22  -0.257190520 -0.096283008
## 23   0.079004672  0.069331470
## 24   0.045621071  0.273744275
## 25  -0.095726325 -0.121423696
## 26  -0.327863673  0.146682770
## 27  -0.524828980  0.136462715
## 28  -0.114574625  0.018813598
## 29   0.176442414  0.516755075
## 30   0.004923706 -0.026818486
## 31  -0.255432167 -0.039196565
## 32  -0.651455580  0.186386023
## 33   0.795600465 -0.089206574
## 34   0.286764008  0.003855882
## 35   0.275663499  0.069333549
## 36   0.273192758  0.115355290
## 37  -0.019355485  0.216579669
## 38  -0.430655987  0.041180722
## 39  -0.544888506  0.249607076
## 40   0.324807115  0.152446880
## 41  -0.383430493 -0.065799799
## 42   0.974142926  0.099858675
## 43   0.483668631  0.001762523
```

```
## 44   0.086329481 -0.015860215
## 45   0.592927168  0.212404817
## 46   0.039848168  0.138754400
## 47  -0.133563448 -0.085456273
## 48   0.078335123 -0.093150062
## 49   0.203490396 -0.045156295
## 50  -0.081592977  0.285044098
## 51   0.071771025  0.042342011
## 52  -0.333288415 -0.031065136
## 53   0.389843654 -0.070818678
## 54  -0.051735070  0.013293539
## 55   0.336226202 -0.047768128
## 56   0.213503737  0.063563267
## 57   0.006325543  0.171365419
## 58  -0.192666143 -0.285902335
## 59   0.342770033  0.002315869
## 60  -0.160418723  0.143531419
## 61   0.166384924 -0.090691477
## 62  -0.727026914  0.242167335
## 63  -0.773734967  0.072438269
## 64   0.322009328  0.131438681
## 65  -0.420727503 -0.112157437
## 66  -0.020383160  0.012927389
## 67   0.244182681 -0.022339141
## 68  -0.200780706  0.059642388
## 69  -0.493536428 -0.068356186
## 70  -0.095769868  0.081723664
## 71   0.249681590 -0.079098361
## 72  -0.424476465 -0.202395222
## 73  -0.822573871  0.112356902
## 74  -0.212015610  0.083145716
## 75   0.165296973 -0.172911929
## 76   0.330425324  0.023730253
## 77  -0.035847768  0.047811584
## 78   0.166257996  0.072994846
## 79   0.156686645 -0.104552400
## 80   0.575611098  0.184520513
## 81  -0.075493806 -0.110535601
## 82  -0.445228727 -0.173840786
## 83  -0.377299836 -0.140080707
## 84   0.194899678  0.081583104
## 85  -0.069374412 -0.127189154
## 86   0.234353460 -0.099418205
## 87   0.466471828  0.144448371
## 88   0.017368157  0.017010204
## 89   0.072537807 -0.217777636
## 90  -0.505462966 -0.276663720
## 91   0.227065006 -0.119135914
## 92   0.027329044 -0.055648273
## 93  -0.491914216 -0.103602116
## 94   0.304699905 -0.020519664
## 95   0.074914724  0.005413760
## 96  -0.154482489  0.103756836
## 97   0.602041604 -0.107238256
```

```
## 98   0.126246474 -0.165028139
## 99   0.158751088  0.126274465
## 100 -0.450366686  0.117822146
## 101  0.675145368  0.057821222
## 102  0.061955384 -0.099072315
## 103  0.190994964 -0.038363021
## 104 -0.010562393 -0.116948933
## 105  0.519993190 -0.055575381
## 106 -0.020869249 -0.239452433
## 107 -0.191507468  0.117521252
## 108 -0.395034937  0.088354796
## 109  0.031199144  0.050399979
## 110  0.105517052  0.113532804
## 111 -0.295567042  0.074086462
## 112 -0.307056013  0.021818372
## 113 -0.700384634  0.027924451
## 114 -0.423905898  0.156047768
## 115  0.700296984  0.038503117
## 116 -0.381795492  0.011689257
## 117  0.236282605  0.017234040
## 118  0.253526400 -0.114066698
## 119 -0.104414358 -0.001767086
## 120  0.132273881  0.094838187
## 121 -0.563273227 -0.032669247
## 122 -0.498541162 -0.140298983
## 123  0.091666970  0.221225333
## 124  0.119818491  0.022372035
## 125  0.725504514 -0.255352070
## 126 -0.327039760  0.177813004
## 127 -0.145138166 -0.062637124
## 128  0.543285569  0.132006337
## 129  0.116697085  0.243395310
## 130 -0.236514322 -0.317942578
## 131  0.089328529 -0.097112691
## 132 -0.089938639  0.018742269
## 133 -0.067978488 -0.194261468
## 134  0.017037924 -0.026238155
## 135 -0.162360478  0.073916343
## 136  0.061503068 -0.164664072
## 137  0.257746968  0.078331890
## 138  0.528515016 -0.088965177
## 139 -0.209739156  0.100925494
## 140 -0.280311603 -0.121761080
## 141  0.146040798  0.141997533
## 142 -0.470309649 -0.006587069
## 143  0.110872051 -0.114805544
## 144  0.027180928  0.207052877
## 145 -0.262927302 -0.059028399
## 146 -0.148987781  0.107364047
## 147  0.440899721 -0.061057946
## 148  0.287532908 -0.034141165
## 149 -0.447163334 -0.096493108
## 150  0.011734927 -0.080130672
```

```r
pca$loadings
```

```
##
## Loadings:
##           Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
## dGF%      -0.491         0.240                      -0.170 -0.234  0.239
## dAdjGF%   -0.484         0.242                0.101 -0.299 -0.254  0.291
## d5v5_xGF% -0.266 -0.496 -0.137        -0.265  0.255 -0.350  0.632
## d5v5_CF%  -0.239 -0.573        -0.142                0.133 -0.496 -0.564
## dTime_Led -0.409        -0.177  0.104 -0.212 -0.772  0.303  0.213
## d5v5_Sh%  -0.141  0.529 -0.128  0.433 -0.313        -0.279        -0.555
## d5v5_Sv%          0.187  0.578 -0.583 -0.153 -0.137         0.324 -0.375
## dPP%      -0.172  0.159 -0.497 -0.393  0.592 -0.140 -0.367        -0.178
## dPK%             -0.174  0.478  0.519  0.602 -0.178         0.160 -0.201
## dROWins   -0.415  0.195                0.209  0.496  0.657  0.221 -0.108
##           Comp.10
## dGF%       0.736
## dAdjGF%   -0.669
## d5v5_xGF%
## d5v5_CF%
## dTime_Led
## d5v5_Sh%
## d5v5_Sv%
## dPP%
## dPK%
## dROWins
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings       1.0    1.0    1.0    1.0    1.0    1.0    1.0    1.0
## Proportion Var    0.1    0.1    0.1    0.1    0.1    0.1    0.1    0.1
## Cumulative Var    0.1    0.2    0.3    0.4    0.5    0.6    0.7    0.8
##                Comp.9 Comp.10
## SS loadings       1.0     1.0
## Proportion Var    0.1     0.1
## Cumulative Var    0.9     1.0
```

```r
pca$center
```

```
##      dGF%   dAdjGF% d5v5_xGF%  d5v5_CF% dTime_Led  d5v5_Sh%  d5v5_Sv%
## 2.3687333 2.2268000 0.5541333 0.4273333 1.8223333 0.4026667 0.1891333
##      dPP%      dPK%   dROWins
## 0.9120000 0.4493333 4.6400000
```

```r
pca$scale
```

```
##      dGF%   dAdjGF% d5v5_xGF%  d5v5_CF% dTime_Led  d5v5_Sh%  d5v5_Sv%
## 3.0132028 3.0151295 3.4389055 3.8715881 2.8865829 1.1646033 0.9983246
##      dPP%      dPK%   dROWins
## 3.9198116 3.4375911 5.0731713
```

Naming of Principal Components

Comp.1: "Creating Wins"

This metric represents your team's (negative) ability the win games. Value comes from GF%, AdjGF%, Time_Led, and ROWins. ### Comp.2: "Quantity vs Quality" The higher this metric, the more your team relies on skill and high percentage plays to score. The lower this metric, the more your team relies on controlling the possession battle and generating a higher volume of shot attempts. Value comes from 5v5_Sh%, and negatively from xGF% and CF% ### Comp.3: "Defensive vs Offensive" This metric is higher for teams which tend to play tight, low-scoring games, and lower for teams who play a shootout-style high-scoring game. Value comes from Sv% and PK%, and negatively from PP%

Logistic Regression

The binary model runs regression only on whether the higher-seeded team won or lost. The weighted model is an abuse of glm because it uses non-integer success variables (win% in the series)

```
num_factors <- 3
Comps <- NULL
for (i in 1:num_factors) {
  Comps[i] <- paste0('Comp.', i)
}

regression_binary <- function(dataset) {
  return(suppressWarnings(glm(`Home_Won` ~ `Comp.1` + `Comp.2` + `Comp.3`,data=dataset, family=binomial
}

regression_weighted <- function(dataset) {
  return(suppressWarnings(glm(`Home_W.` ~ `Comp.1` + `Comp.2` + `Comp.3`,data=dataset, family=binomial(
}

binary_model <- regression_binary(prin_data)
weighted_model <- regression_weighted(prin_data)
summary(binary_model)
```

```
##
## Call:
## glm(formula = Home_Won ~ Comp.1 + Comp.2 + Comp.3, family = binomial(link = "logit"),
##     data = dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0246  -1.0713   0.5498   0.9913   1.8899
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2600     0.1795   1.448 0.147505
## Comp.1       -0.3704     0.1080  -3.429 0.000605 ***
## Comp.2       -0.2019     0.1286  -1.571 0.116285
## Comp.3        0.4412     0.1549   2.848 0.004402 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 206.23  on 149  degrees of freedom
## Residual deviance: 181.99  on 146  degrees of freedom
## AIC: 189.99
##
## Number of Fisher Scoring iterations: 3
```

```
summary(weighted_model)
```

```
##
## Call:
## glm(formula = Home_W. ~ Comp.1 + Comp.2 + Comp.3, family = binomial(link = "logit"),
##     data = dataset)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.29427  -0.29962  -0.02179   0.28636   1.28898
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.14774    0.16589   0.891    0.373
## Comp.1      -0.13863    0.09044  -1.533    0.125
## Comp.2      -0.09632    0.11757  -0.819    0.413
## Comp.3       0.11938    0.13690   0.872    0.383
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 149  degrees of freedom
## Residual deviance: 39.393  on 146  degrees of freedom
## AIC: 199.85
##
## Number of Fisher Scoring iterations: 3
```

Cross Validation

```
logLoss <- function(pred, actual){
  -1*mean(log(pred[model.matrix(~ actual + 0) - pred > 0]))
}

mse <- function(pred, actual){
  mean((pred-actual)^2)
}

ten_fold_cross_validate <- function(dataset){
  series_shuffled <- dataset[sample(nrow(dataset)),]
  folds <- cut(seq(1,nrow(series_shuffled)),breaks=10,labels=FALSE)

  log_loss_binary <- NA
  log_loss_weighted <- NA
  mse_binary <- NA
  mse_weighted <- NA

  for (i in 1:10) {
```

```
    testRows <- which(folds==i,arr.ind=TRUE)
    test_data <- series_shuffled[testRows,]
    train_data <- series_shuffled[-testRows,]

    fold_binary_model <- regression_binary(train_data)
    fold_weighted_model <- regression_weighted(train_data)

    test_data$bin_pred <- predict(fold_binary_model, test_data[,Comps], type='response')
    test_data$wgt_pred <- predict(fold_weighted_model, test_data[,Comps], type='response')

    log_loss_binary[i] <- logLoss(test_data$bin_pred, test_data$Home_Won)
    log_loss_weighted[i] <- logLoss(test_data$wgt_pred, test_data$Home_Won)
    mse_binary[i] <- mse(test_data$bin_pred, test_data$Home_Won)
    mse_weighted[i] <- mse(test_data$wgt_pred, test_data$Home_Won)
  }
  return(c(mean(log_loss_binary), mean(log_loss_weighted), mean(mse_binary), mean(mse_weighted)))
}

ten_fold_metrics <- matrix(NA, nrow=200, ncol=4)
colnames(ten_fold_metrics) <- c("Log Loss Binary", "Log Loss weighted", "Mean Squared Error Binary", "Me
for(j in 1:200) {
  ten_fold_metrics[j,] <- ten_fold_cross_validate(prin_data)
}

colMeans(ten_fold_metrics)
```

```
##             Log Loss Binary           Log Loss weighted
##                   0.5521464                   0.5914924
##   Mean Squared Error Binary Mean Squared Error weighted
##                   0.2215097                   0.2284080
```

```
# 1-13 -> 0.238
# 1-8 -> 0.232
# 1-6 -> 0.232
# 1-5 -> 0.229
# 1-4 -> 0.226 0.225
# 1,2,3 -> 0.223 0.222 0.221
# 1,2 -> 0.232 0.230
# 1,3 -> 0.222 0.220
```

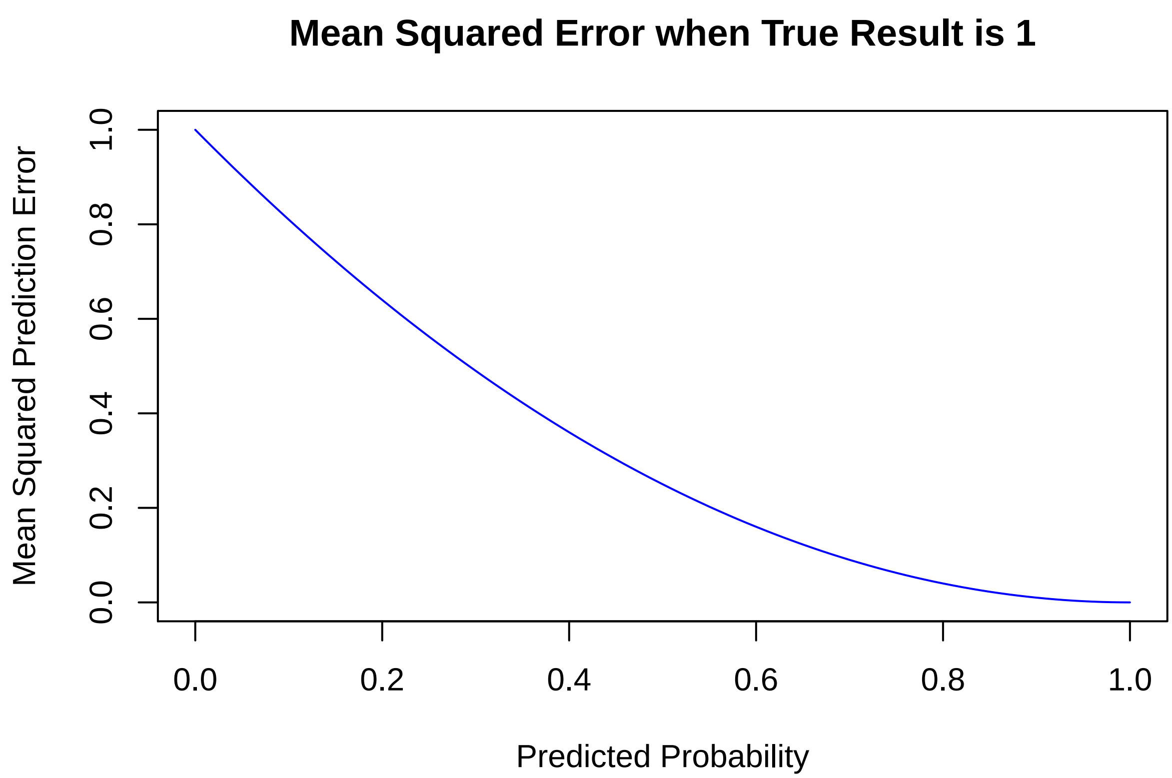
Apply Current Model With Exact Specifications

```
playoff_teams_2018 <- read.csv('input_csvs/playoff_teams_2018.csv', check.names=FALSE)

predict_matchup <- function(home_team, away_team) {
  home_stats <- playoff_teams_2018[playoff_teams_2018$Team==home_team,stats]
  away_stats <- playoff_teams_2018[playoff_teams_2018$Team==away_team,stats]
  raw_diffs <- home_stats - away_stats
  colnames(raw_diffs) <- dStats
  nrm_diffs <- data.frame(0)
  for (i in 1:length(dStats)) {
    dStat <- dStats[i]
    center <- pca$center[dStat]
```

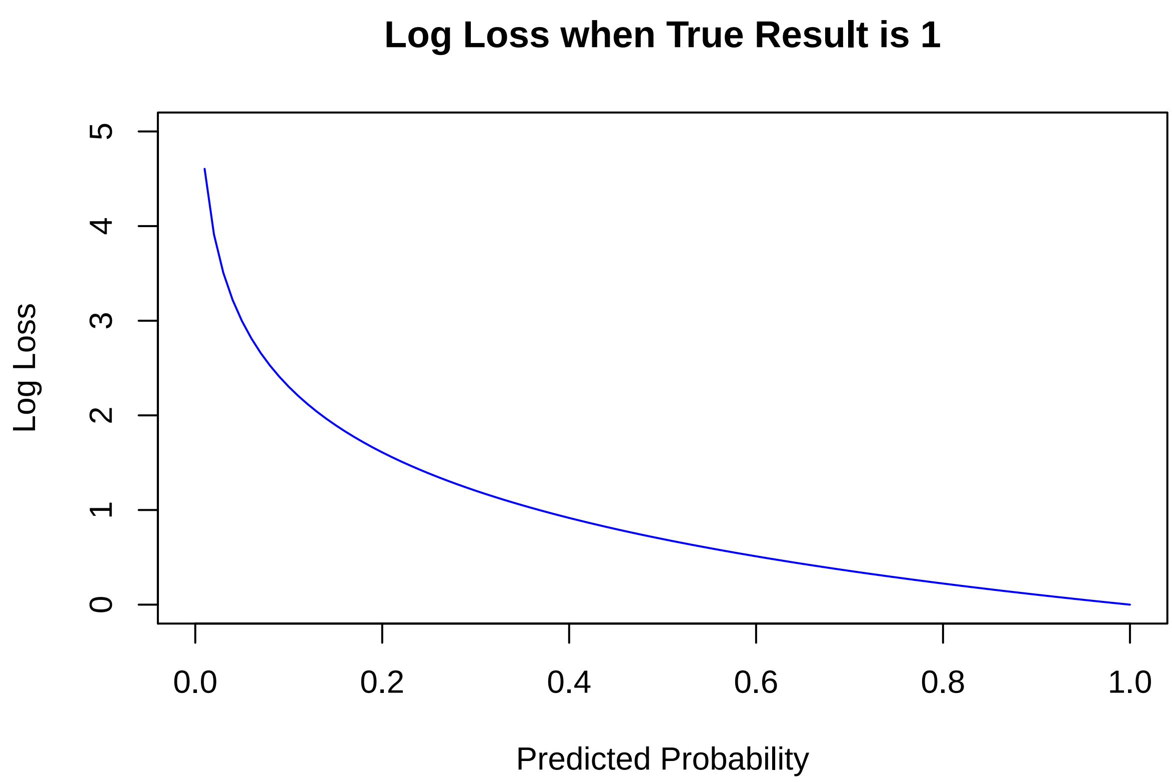
```
    scale <- pca$scale[dStat]
    nrm_diffs[,dStat] <- (raw_diffs[dStat]-center)/scale
  }
  component_values <- data.frame(0)
  for (j in 1:length(Comps)) {
    component <- Comps[j]
    loadings <- pca$loadings[,component]
    component_values[,component] <- 0
    for (k in 1:length(dStats)) {
      stat <- dStats[k]
      contribution <- loadings[stat] * nrm_diffs[stat]
      component_values[,component] <- component_values[,component] + contribution
    }
  }
  predict(binary_model, component_values[,Comps], type='response')
}

predict_matchup('BOS', 'ANA')
```

```
##         1
## 0.5759154
```

```
squareerror <- function(x) {(x-1)^2}
logloss <- function(x) {-log(x)}
plot(squareerror, xlim=c(0,1), ylim=c(0,1), xlab='Predicted Probability', ylab='Mean Squared Prediction
```



```
plot(logloss, xlim=c(0,1), ylim=c(0,5), xlab='Predicted Probability', ylab='Log Loss', main='Log Loss wh
```



IMPORT FOR PREDICTION

```
corsica1_pred <- read.csv("input_csvs/2018/corsica_5v5_18.csv", check.names=FALSE)
corsica1_pred$Season <- 2018
corsica1_pred <- corsica1_pred[, c('Team', 'Season', 'CF%', 'GF%', 'xGF%', 'Sh%', 'Sv%')]

corsica2_pred <- read.csv("input_csvs/2018/corsica_all_18.csv", check.names=FALSE)
corsica2$Season <- 2018
corsica2_pred <- corsica2_pred[, c('Team', 'Season', 'allGF%', 'allGF', 'allGA')]

morestats_pred <- read.csv("input_csvs/2018/misc_stats_playoffwins_18.csv", check.names=FALSE)
morestats_pred <- morestats_pred[, c('Team', 'Season', 'Time Led', 'GF_EN', 'GA_EN', 'PP%', 'PK%', 'ROW

raw_data_pred <- merge(merge(corsica1_pred, corsica2_pred, by="Team"), morestats_pred, by="Team")
raw_data_pred$`AdjGF%` <- with(raw_data_pred, round(100*(allGF-GF_EN)/(allGF-GF_EN+allGA-GA_EN),2))
raw_data_pred <- raw_data_pred[, c('Team', 'Season', 'allGF%', 'AdjGF%', 'xGF%', 'CF%', 'Time Led', 'Sh%
colnames(raw_data_pred) <- c('Team', 'Season', 'GF%', 'AdjGF%', '5v5_xGF%', '5v5_CF%', 'Time_Led', '5v5_

raw_data_pred
```

```
##     Team Season   GF% AdjGF% 5v5_xGF% 5v5_CF% Time_Led 5v5_Sh% 5v5_Sv%  PP%
## 1    ANA   2018 52.50  52.22    48.41   48.62    23.90    8.16   93.35 17.8
```

```
## 2   ARI   2018 45.08  45.80     46.18   48.00     16.77     7.28    92.15 16.9
## 3   BOS   2018 55.86  55.38     53.48   53.80     21.40     7.82    92.34 23.5
## 4   BUF   2018 41.68  41.83     45.05   47.58     13.30     6.16    91.63 19.1
## 5   CAR   2018 47.07  46.97     53.12   54.47     18.47     7.04    90.97 18.4
## 6   CBJ   2018 51.08  51.13     51.94   51.55     20.60     7.44    92.85 17.2
## 7   CGY   2018 47.06  46.94     52.62   53.45     17.13     6.79    91.92 16.0
## 8   CHI   2018 47.30  45.83     49.59   52.36     17.55     7.08    91.82 16.0
## 9   COL   2018 51.93  53.08     46.29   47.57     21.23     8.21    93.24 21.9
## 10  DAL   2018 50.99  51.29     53.54   51.04     19.55     7.69    92.73 19.3
## 11  DET   2018 45.34  45.88     48.58   48.80     16.95     7.53    91.98 17.5
## 12  EDM   2018 46.64  46.30     50.89   50.62     15.53     7.44    91.74 14.8
## 13  FLA   2018 50.20  50.53     48.72   49.14     20.65     7.71    92.32 18.9
## 14  L.A   2018 53.99  55.16     48.00   50.05     18.17     7.80    93.09 20.4
## 15  MIN   2018 52.19  51.32     53.58   47.18     22.35     8.19    92.72 20.4
## 16  MTL   2018 44.52  44.35     51.24   50.49     15.33     6.38    92.19 21.2
## 17  N.J   2018 50.31  51.08     50.32   48.63     19.48     7.91    91.73 21.4
## 18  NSH   2018 56.13  56.95     50.93   51.52     25.03     8.19    93.55 21.2
## 19  NYI   2018 47.11  46.74     46.87   47.47     18.58     8.84    91.71 23.2
## 20  NYR   2018 46.44  45.34     47.12   45.90     15.20     7.55    92.18 21.2
## 21  OTT   2018 43.54  42.03     45.87   47.12     14.17     7.69    90.84 16.6
## 22  PHI   2018 51.34  51.82     50.54   49.79     18.83     7.88    92.48 20.7
## 23  PIT   2018 52.12  52.00     52.69   52.23     19.47     7.29    91.03 26.2
## 24  S.J   2018 52.22  52.22     51.92   50.80     20.57     7.50    91.64 20.6
## 25  STL   2018 50.11  49.76     51.46   51.73     20.55     7.17    92.87 15.4
## 26  T.B   2018 55.34  55.58     52.76   51.62     23.67     9.35    92.93 23.9
## 27  TOR   2018 54.00  54.02     51.10   49.86     22.30     9.01    92.81 25.0
## 28  VAN   2018 45.76  47.02     46.57   47.68     16.13     7.21    92.15 21.4
## 29  VGK   2018 54.27     NA     50.67   50.96     22.28     8.38    92.19 21.4
## 30  WPG   2018 55.83  56.03     52.79   51.50     27.07     8.53    92.54 23.4
## 31  WSH   2018 51.82  51.47     46.88   47.99     21.93     9.14    92.47 22.5
##      PK% ROWins
## 1  83.2     40
## 2  79.5     27
## 3  83.7     47
## 4  77.9     24
## 5  77.5     33
## 6  76.2     39
## 7  81.8     35
## 8  79.1     32
## 9  83.3     41
## 10 80.8     38
## 11 77.5     25
## 12 76.7     31
## 13 80.2     41
## 14 85.0     43
## 15 81.3     42
## 16 74.1     27
## 17 81.8     39
## 18 81.9     47
## 19 73.2     32
## 20 81.4     31
## 21 76.2     26
## 22 75.8     40
## 23 80.0     45
```

```
## 24 84.8      40
## 25 79.7      41
## 26 76.1      48
## 27 81.4      42
## 28 78.3      31
## 29 81.4      47
## 30 81.8      48
## 31 80.3      46
```