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. 5v5 HDCF% Within 1 (NaturalStatTrick)
7. %Leading (morehockeystats)
8. 5v5 Shooting% (Corsica1)
9. 5v5 Save% (Corsica1)
10. PP% (FoxSports)
11. PK% (FoxSports)
12. Regular Season Wins (NHL.com)
13. 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("corsica_5v5_08-17.csv")
corsica1$Season <- strtoi(substr(corsica1$Season, 6, 9))
corsica1 <- corsica1[, c('Team', 'Season', 'CF.', 'GF.', 'xGF.', 'Sh.', 'Sv.')]

corsica2 <- read.csv("corsica_all_08-17.csv")
corsica2$Season <- strtoi(substr(corsica2$Season, 6, 9))
```

```r
corsica2 <- corsica2[, c('Team', 'Season', 'allGF.', 'allGF', 'allGA')]

stattrick <- read.csv("misc_stats_playoffwins.csv")
stattrick <- stattrick[, c('Team', 'Season', 'HDCF', 'Time.Led', 'Playoff.Wins', 'GF_EN', 'GA_EN', 'PP.

raw_data <- merge(merge(corsica1, corsica2, by=c("Team", "Season")), stattrick, 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
for (year in 2008:2017){
  season <- adj_data$Season==year

  adj_data$`5v5_Sh%`[season] <- round(adj_data$`5v5_Sh%`[season] - mean(adj_data$`5v5_Sh%`[season]), 2)
  adj_data$`5v5_Sv%`[season] <- round(adj_data$`5v5_Sv%`[season] - mean(adj_data$`5v5_Sv%`[season]), 2)
  adj_data$`Time_Led`[season] <- round(adj_data$`Time_Led`[season] - mean(adj_data$`Time_Led`[season]),
  adj_data$`PP%`[season] <- round(adj_data$`PP%`[season] - mean(adj_data$`PP%`[season]), 2)
  adj_data$`PK%`[season] <- round(adj_data$`PK%`[season] - mean(adj_data$`PK%`[season]), 2)
  adj_data$Wins[season] <- round(adj_data$Wins[season] - mean(adj_data$Wins[season]), 2)
  adj_data$ROWins[season] <- round(adj_data$ROWins[season] - mean(adj_data$ROWins[season]), 2)
}

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    -0.58
## 2  ANA   2009 50.32  50.87   52.73    51.10   50.96     53.54    -0.27
## 4  ANA   2011 50.21  50.66   46.53    44.08   44.35     44.88     1.47
## 6  ANA   2013 53.82  52.97   55.28    50.07   47.93     48.77     0.66
## 7  ANA   2014 56.44  56.17   58.44    51.45   49.80     52.62     5.03
## 8  ANA   2015 50.78  50.35   51.47    52.13   50.96     52.19     2.42
##   5v5_Sh% 5v5_Sv%   PP%   PK% Wins ROWins Playoff_Wins
```

```
## 1   -0.72    1.65 -1.10  0.85    6   3.20            2
## 2    0.44    0.38  4.66 -1.37    1  -0.70            7
## 4    0.06    0.13  5.50 -0.67    6   6.97            2
## 6    0.64    0.93  3.35 -0.24    6   3.23            3
## 7    2.10    0.36 -1.90  0.12   13  15.93            7
## 8    0.75   -0.45 -2.92 -0.37   10   7.67           11
```

```
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     0.66
## 7   ANA   2014 56.44  56.17   58.44    51.45   49.80     52.62     5.03
## 8   ANA   2015 50.78  50.35   51.47    52.13   50.96     52.19     2.42
## 9   ANA   2016 53.35  52.59   49.42    53.01   52.42     52.22     4.73
## 10  ANA   2017 52.76  52.48   53.11    52.02   49.67     52.26     3.46
## 30  BOS   2013 54.51  54.30   55.42    53.24   54.39     54.22     5.03
##    5v5_Sh% 5v5_Sv%    PP%   PK% Wins ROWins Playoff_Wins
## 6     0.64    0.93  3.35 -0.24    6   3.23            3
## 7     2.10    0.36 -1.90  0.12   13  15.93            7
## 8     0.75   -0.45 -2.92 -0.37   10   7.67           11
## 9    -0.82   -0.11  4.44  5.88    5   5.57            3
## 10    0.10    0.66 -0.41  3.83    5   5.30           11
## 30   -0.64    1.10 -3.35  5.36    4   3.23           14
```

Correlations

```
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

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`)

correlations_5_10
```

```
##         GF%    adjGF%   5v5_GF%  5v5_xGF%   5v5_CF% 5v5_HDCF%   Time_Led
## 1 0.3458189 0.3358988 0.2745515 0.2436887 0.2310239 0.2050112 0.07079898
## 2 0.3416835 0.3317128 0.2426668 0.2060818 0.1818608 0.1710590 0.20760090
##       5v5_Sh%   5v5_Sv%         PP%       PK%      Wins    ROWins
## 1 -0.07216727 0.1673556 -0.10352769 0.2168811 0.2557709 0.2028160
## 2 -0.02793233 0.1114788  0.06066142 0.2087697 0.2323212 0.2247324
```

The selected independent variables

Regression

```
indep_vars <- c('GF%', '5v5_xGF%', 'Time_Led', 'PK%', 'Wins')
adj_data_10 <- adj_data_10[, c('Team', 'Season', indep_vars)]

series <- read.csv("all_playoff_series.csv")
series$Home_Won <- round(series$Home_W., 0)
series$`dGF%` <- NA
```

```r
series$`d5v5_xGF%` <- NA
series$`dTime_Led` <- NA
series$`dPK%` <- NA
series$`dWins` <- NA

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),]
  row$`dGF%` <- a1$`GF%` - a2$`GF%`
  row$`d5v5_xGF%` <- a1$`5v5_xGF%` - a2$`5v5_xGF%`
  row$`dTime_Led` <- a1$`Time_Led` - a2$`Time_Led`
  row$`dPK%` <- a1$`PK%` - a2$`PK%`
  row$`dWins` <- a1$`Wins` - a2$`Wins`
  return(row)
}

for (row in 1:nrow(series)) {
  series[row,] <- getDifferences(series[row,], adj_data_10)
}

head(series)
```

```
##   Year Home Away Home_W. Home_Won  dGF% d5v5_xGF% dTime_Led dPK% dWins
## 1 2008  ANA  DAL    0.33        0 -2.03     -0.89     -3.52 -2.5     2
## 2 2009  DET  ANA    0.57        1  4.28      1.46      5.17 -1.4     9
## 3 2010  ARI  DET    0.43        0 -0.02     -0.78      0.70  0.6     6
## 4 2011  ANA  NSH    0.33        0 -2.64     -6.48     -2.45 -3.6     3
## 5 2012  ARI  CHI    0.67        1  0.92     -2.27     -2.30  7.4    -3
## 6 2013  ANA  DET    0.43        0  1.23     -2.32     -0.64 -0.2     6
```

```r
colMeans(series[,c('dGF%', 'd5v5_xGF%', 'dTime_Led', 'dPK%', 'dWins')])
```

```
##      dGF% d5v5_xGF% dTime_Led      dPK%     dWins
## 2.3687333 0.5541333 1.8223333 0.4493333 4.7533333
```

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)

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

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

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

```
##
```

```
## Call:
## glm(formula = Home_Won ~ `dGF%` + `d5v5_xGF%` + dTime_Led + `dPK%` +
##     dWins, family = binomial(link = "logit"), data = dataset)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1816  -1.0449   0.5483   0.9785   2.1232
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.13725    0.26616  -0.516  0.60608
## `dGF%`       0.29988    0.09890   3.032  0.00243 **
## `d5v5_xGF%`  0.08984    0.05662   1.587  0.11259
## dTime_Led   -0.02581    0.08024  -0.322  0.74770
## `dPK%`       0.10085    0.05552   1.817  0.06929 .
## dWins       -0.07547    0.05970  -1.264  0.20618
## ---
## 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: 181.12  on 144  degrees of freedom
## AIC: 193.12
##
## Number of Fisher Scoring iterations: 3
```

```r
summary(weighted_model)
```

```
##
## Call:
## glm(formula = Home_W. ~ `dGF%` + `d5v5_xGF%` + dTime_Led + `dPK%` +
##     dWins, family = binomial(link = "logit"), data = dataset)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.19185  -0.30043   0.00764   0.27583   1.37436
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.01182    0.24747  -0.048    0.962
## `dGF%`       0.06687    0.08705   0.768    0.442
## `d5v5_xGF%`  0.03412    0.05173   0.660    0.510
## dTime_Led    0.03223    0.07448   0.433    0.665
## `dPK%`       0.04796    0.05042   0.951    0.341
## dWins       -0.02058    0.05415  -0.380    0.704
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 149  degrees of freedom
## Residual deviance: 39.097  on 144  degrees of freedom
## AIC: 203.87
##
## Number of Fisher Scoring iterations: 3
```

Cross Validation

```r
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)
    testData <- series_shuffled[testRows,]
    trainData <- series_shuffled[-testRows,]

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

    dcolumns <- c('dGF%', 'd5v5_xGF%', 'dTime_Led', 'dPK%', 'dWins')
    testData$bin_pred <- predict(fold_binary_model, testData[,dcolumns], type='response')
    testData$wgt_pred <- predict(fold_weighted_model, testData[,dcolumns], type='response')

    log_loss_binary[i] <- logLoss(testData$bin_pred, testData$Home_Won)
    log_loss_weighted[i] <- logLoss(testData$wgt_pred, testData$Home_Won)
    mse_binary[i] <- mse(testData$bin_pred, testData$Home_Won)
    mse_weighted[i] <- mse(testData$wgt_pred, testData$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
##                   0.5631314                   0.5951806
##   Mean Squared Error Binary Mean Squared Error weighted
##                   0.2249620                   0.2311865
```

Sandbox Testing

```r
newdata <- data.frame(matrix(c(1,2,3,4,5,2,-3,-1,-5,4),nrow=2, byrow = TRUE))
colnames(newdata) <- c('dGF%', 'd5v5_xGF%', 'dTime_Led', 'dPK%', 'dWins')
predict(binary_model, newdata, type='response')
```

```
##         1         2
## 0.5722277 0.3572479
```

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%` + `dTime_Led` + `dPK%` + `dWins`, data=seaso

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%` + dTime_Led + `dPK%` +
##     dWins, family = binomial(link = "logit"), data = seasons_for_2016)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2532  -1.0160   0.4617   0.9744   2.0123
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.33312    0.31562  -1.055  0.29122
## `dGF%`       0.32936    0.11087   2.971  0.00297 **
## `d5v5_xGF%`  0.05263    0.06192   0.850  0.39535
## dTime_Led    0.05890    0.08764   0.672  0.50154
## `dPK%`       0.12837    0.06362   2.018  0.04362 *
## dWins       -0.09091    0.06959  -1.306  0.19141
## ---
## 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: 139.32  on 114  degrees of freedom
## AIC: 151.32
##
## 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 0.6790362
## 19   STL  CHI        1 0.4906760
## 29   DAL  STL        0 0.4218549
## 39   DAL  MIN        1 0.5593192
## 49   T.B  DET        1 0.7750206
```

```
## 59   FLA  NYI        0 0.4347668
## 69   L.A  S.J        0 0.4209995
## 79   S.J  NSH        1 0.4616089
## 89   T.B  NYI        1 0.5365245
## 99   PIT  NYR        1 0.8219678
## 109  WSH  PHI        1 0.7944419
## 119  WSH  PIT        0 0.3516794
## 129  PIT  T.B        1 0.5783330
## 139  PIT  S.J        1 0.6160323
## 149  STL  S.J        0 0.3788325
```

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

```
## [1] 0.489087
```

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

```
## [1] 0.1787061
```