

Residual Analysis of NHL Forwards' Points per Game versus Deployment Statistics

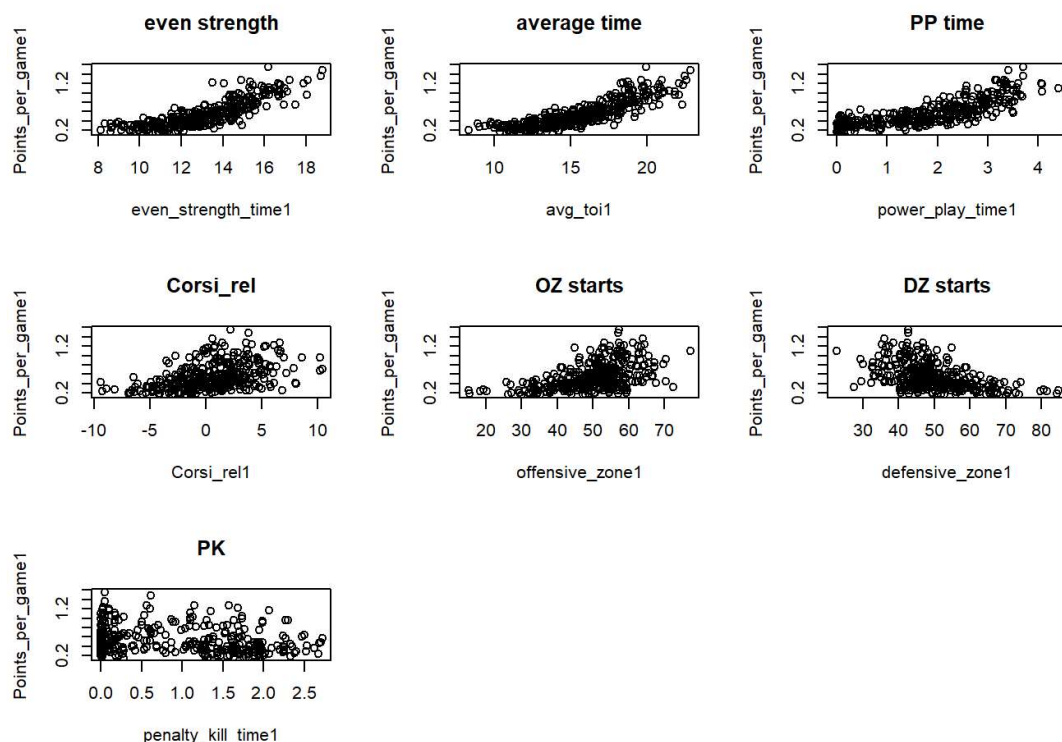
Steven Mey, V00866746

February 6, 2020

The purpose of this exercise was to use NHL deployment data as regressors for points per game, in an effort to determine forwards who performed better and worse than average when compared to their peers with similar on ice deployments. The regressors were: average time on ice per game, average power play time per game, % of starts in the offensive zone, and the amount of even strength ice time per game. These were all used as explanatory variables for Points per game. The resulting rankings can be found in the "model_results.csv" file in this repository. All data used was obtained from : https://www.hockey-reference.com/leagues/NHL_2019_skaters.html (https://www.hockey-reference.com/leagues/NHL_2019_skaters.html)

```
hdat<-read.csv("hockey_master_forwards.csv")
Points_per_game1<-hdat$PTS.G
Corsi_rel1<-hdat$CFPER.rel
offensive_zone1<-hdat$Oz.sper
defensive_zone1<-hdat$Dz.sper
power_play_time1<-hdat$TOI.PP #*nhldat$GP
penalty_kill_time1<-hdat$TOI.PK
even_strength_time1<-hdat$TOI.EV
avg_toi1<-hdat$ATOI

par(mfrow=c(3,3))
plot(even_strength_time1,Points_per_game1,main="even strength")
plot(avg_toi1,Points_per_game1,main="average time")
plot(power_play_time1,Points_per_game1,main="PP time")
plot(Corsi_rel1,Points_per_game1,main="Corsi_rel")
plot(offensive_zone1,Points_per_game1, main="OZ starts")
plot(defensive_zone1,Points_per_game1,main="DZ starts")
plot(penalty_kill_time1,Points_per_game1,main="PK")
```



```
m4<-lm(Points_per_game1~even_strength_time1+avg_toi1+power_play_time1+offensive_zone1)
summary(m4)
```

```
##
## Call:
## lm(formula = Points_per_game1 ~ even_strength_time1 + avg_toi1 +
##     power_play_time1 + offensive_zone1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29238 -0.09034 -0.00801  0.07441  0.60493
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.6303868   0.0686241   -9.186 < 2e-16 ***
## even_strength_time1  0.0392674   0.0126727    3.099  0.00209 **
## avg_toi1        0.0245018   0.0101857    2.406  0.01664 *
## power_play_time1  0.0793863   0.0120766    6.574 1.68e-10 ***
## offensive_zone1    0.0029168   0.0009486    3.075  0.00226 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1275 on 369 degrees of freedom
## Multiple R-squared:  0.7885, Adjusted R-squared:  0.7862
## F-statistic: 343.9 on 4 and 369 DF,  p-value: < 2.2e-16
```

```
anova(m4)
```

```
## Analysis of Variance Table
##
## Response: Points_per_game1
##              Df Sum Sq Mean Sq  F value    Pr(>F)
## even_strength_time1  1 19.5598  19.5598 1202.8091 < 2.2e-16 ***
## avg_toi1            1  0.8236   0.8236  50.6490 5.794e-12 ***
## power_play_time1    1  1.8340   1.8340 112.7818 < 2.2e-16 ***
## offensive_zone1     1  0.1538   0.1538   9.4554 0.002262 **
## Residuals          369  6.0006   0.0163
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

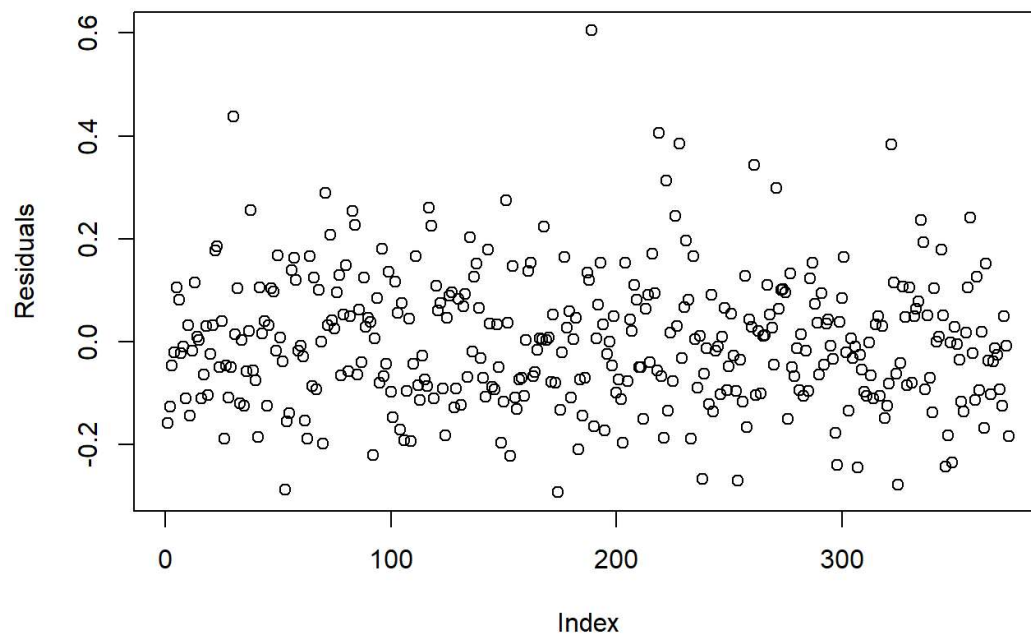
```
library(car)
```

```
## Warning: package 'car' was built under R version 3.6.2
```

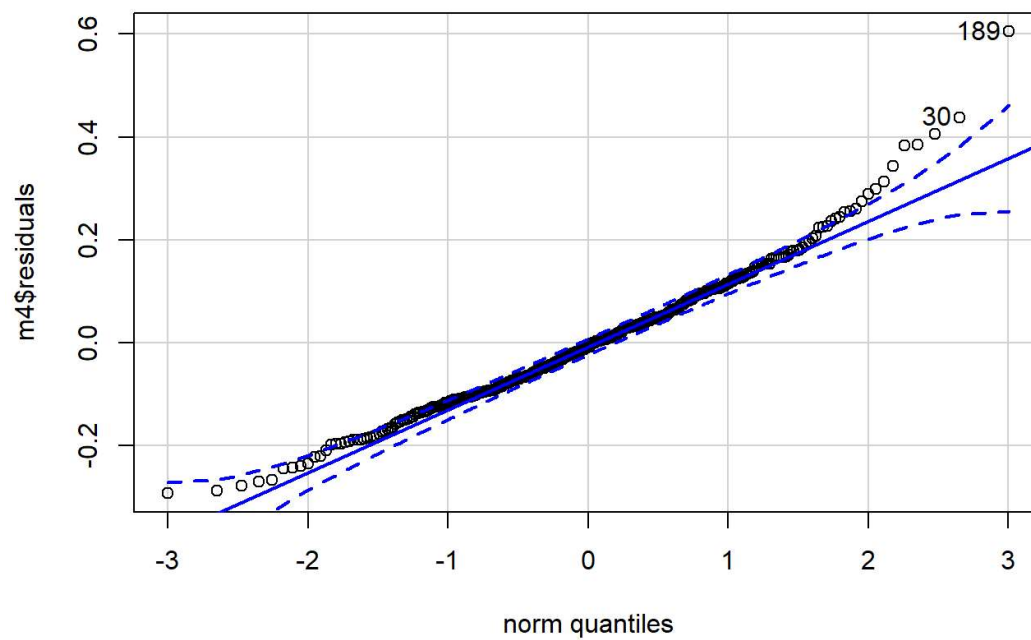
```
## Loading required package: carData
```

```
par(mfrow=c(1,1))
plot(m4$residuals,main="residuals versus index",ylab="Residuals")
```

residuals versus index



```
qqPlot(m4$residuals)
```

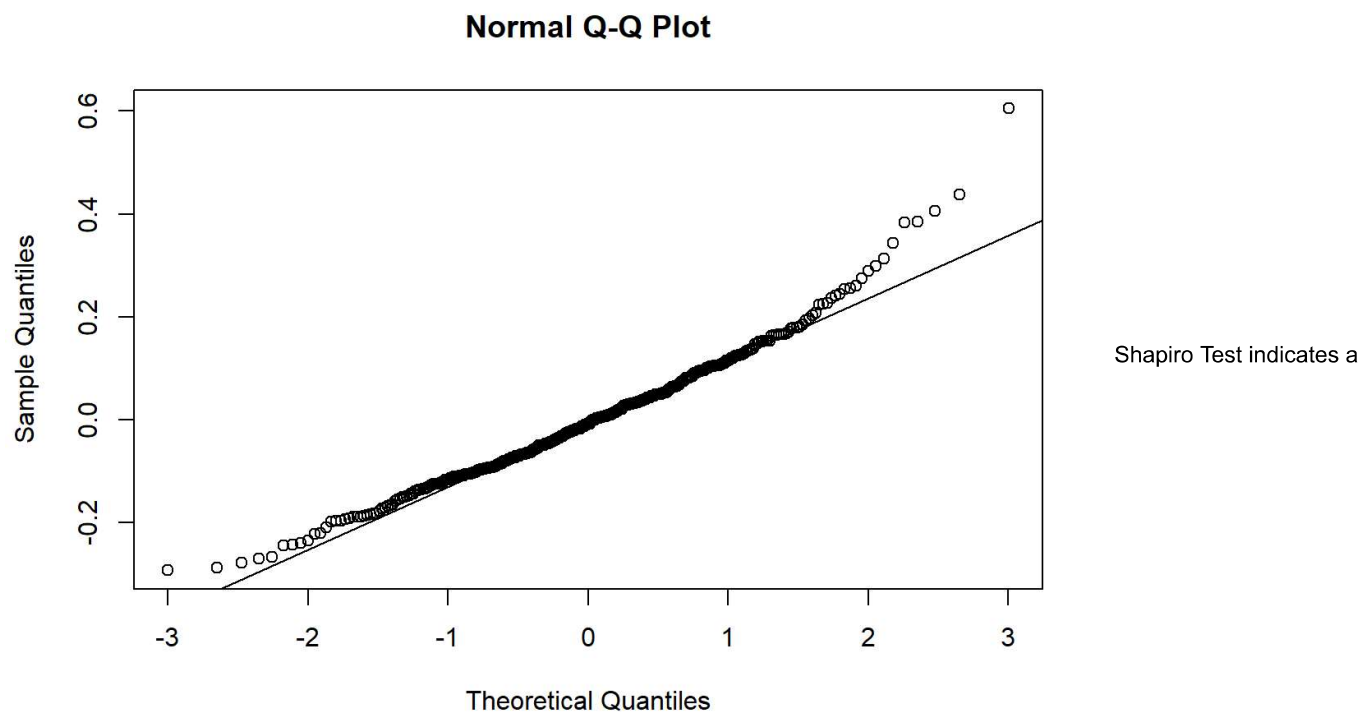


```
## [1] 189 30
```

```
shapiro.test(m4$residuals)
```

```
##  
## Shapiro-Wilk normality test  
##  
## data:  m4$residuals  
## W = 0.97545, p-value = 5.676e-06
```

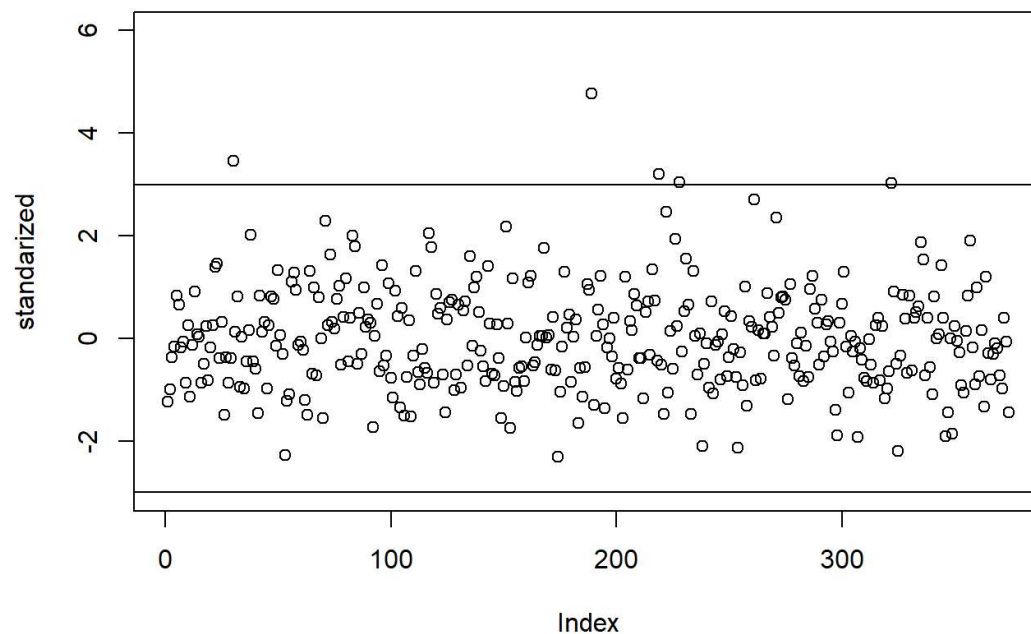
```
qqnorm(m4$residuals)  
qqline(m4$residuals)
```



problem with normality, however the shapiro test has a bias by sample size, and this data is a relatively large sample size. The other tests for normality of residuals appear to indicate no clear issue.

```
sig<-sqrt(var(m4$residuals))  
standarized<-m4$residuals/sig  
plot(standarized,ylim=c(-3,6),main="Standardized Residuals")  
abline(h=3)  
abline(h=-3)
```

Standardized Residuals



```
Resids<-m4$residuals
hdat<-cbind(hdat,Resids)
hdat<-hdat[order(hdat$Resids, decreasing=TRUE),]

#final_model_results<- hdat %>% select(4:11,83)
#write.csv(final_model_results,file="model_results.csv")
```