OBP\_2.R

sports2i

2020-10-26

##################################################################  
####### ###########  
####### OBP ~ HardHit% + Whiff% + First Strike% ###########  
####### ###########  
##################################################################  
  
# Setting up the working directory  
setwd("C:/Users/sports2i/Desktop/WD")  
  
# Loading Library  
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## √ ggplot2 3.3.2 √ purrr 0.3.4  
## √ tibble 3.0.4 √ dplyr 1.0.2  
## √ tidyr 1.1.2 √ stringr 1.4.0  
## √ readr 1.4.0 √ forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(lmtest)

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(multcomp)

## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

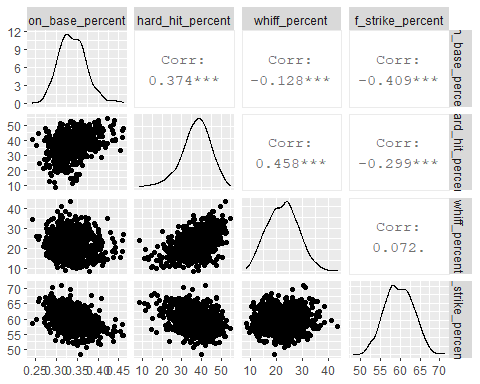
##   
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':  
##   
## geyser

# Reading in data  
mlb <- read.csv(file = "MLB\_OBP3.csv", header = TRUE)  
  
# Data check  
mlb <- mlb[,c(-1,-2,-3,-5,-9)]  
head(mlb)

## on\_base\_percent hard\_hit\_percent whiff\_percent f\_strike\_percent  
## 1 0.293 34.9 23.2 64.0  
## 2 0.360 49.3 23.2 52.1  
## 3 0.356 43.9 32.0 60.3  
## 4 0.297 34.5 17.9 65.5  
## 5 0.334 40.4 16.8 61.9  
## 6 0.337 41.9 18.1 57.3

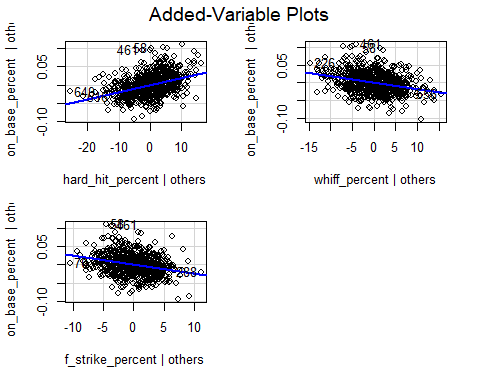
# Matrix scatter plots  
ggpairs(mlb)



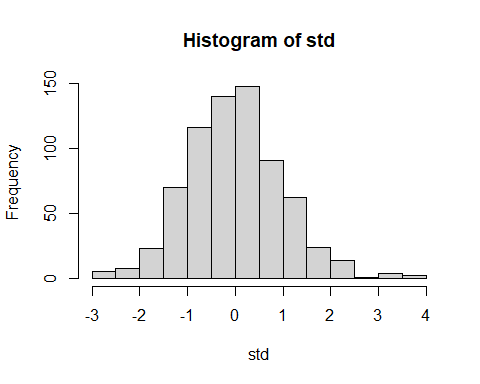
# Fitting model  
mlb\_mlr <- lm(on\_base\_percent ~ ., data = mlb)  
summary(mlb\_mlr)

##   
## Call:  
## lm(formula = on\_base\_percent ~ ., data = mlb)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.083396 -0.018827 -0.000576 0.015773 0.101078   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.4503867 0.0211449 21.300 < 2e-16 \*\*\*  
## hard\_hit\_percent 0.0019804 0.0001711 11.571 < 2e-16 \*\*\*  
## whiff\_percent -0.0018038 0.0002110 -8.551 < 2e-16 \*\*\*  
## f\_strike\_percent -0.0024122 0.0003207 -7.521 1.67e-13 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.02814 on 704 degrees of freedom  
## Multiple R-squared: 0.3084, Adjusted R-squared: 0.3055   
## F-statistic: 104.7 on 3 and 704 DF, p-value: < 2.2e-16

# Test on reduced model  
#reduced.lm <- lm(on\_base\_percent ~ barrel\_batted\_rate   
# + whiff\_percent, data = mlb)  
#anova(mlb\_mlr, reduced.lm)  
  
# Added variable plots  
avPlots(mlb\_mlr)



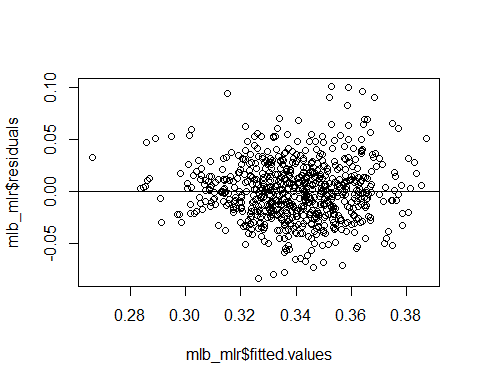
# Std.Residual histogram (normality)  
std <- stdres(mlb\_mlr)  
hist(std)



ks.test(std,"pnorm")

##   
## One-sample Kolmogorov-Smirnov test  
##   
## data: std  
## D = 0.038174, p-value = 0.2535  
## alternative hypothesis: two-sided

# Residual plots  
plot(mlb\_mlr$residuals ~ mlb\_mlr$fitted.values)  
abline(h=0)



bptest(mlb\_mlr)

##   
## studentized Breusch-Pagan test  
##   
## data: mlb\_mlr  
## BP = 7.308, df = 3, p-value = 0.0627

# Try prediction - Joey Votto (2015)  
predict.lm (mlb\_mlr,newdata=data.frame(hard\_hit\_percent = 32.9, f\_strike\_percent = 54.5, whiff\_percent = 24), interval="prediction", level = 0.95)

## fit lwr upr  
## 1 0.3407864 0.2853212 0.3962515

# Real Stats: 0.459  
  
# Shin-Soo Choo (2018)  
predict.lm (mlb\_mlr,newdata=data.frame(hard\_hit\_percent = 40.5, f\_strike\_percent = 58.2, whiff\_percent = 27), interval="prediction", level = 0.95)

## fit lwr upr  
## 1 0.3415009 0.2861721 0.3968296

# Real Stats: 0.376  
  
  
  
# Cross Validation  
n.cv <- 1000  
bias <- rep(NA,n.cv)  
rpmse <- rep(NA,n.cv)  
cvg <- rep(NA,n.cv)  
wid <- rep(NA,n.cv)  
n.test <- round(nrow(mlb)/10)  
  
for(i in 1:n.cv){  
 #split into test and training set  
 test.obs <- sample(1:nrow(mlb),n.test)  
 test.set <- mlb[test.obs,]  
 train.set <- mlb[-test.obs,]  
   
 # fit a lm using training data only  
 train.lm <- lm(on\_base\_percent ~ ., data=train.set)  
   
 # Prediction and prediction intervals  
 test.pred <- predict.lm(train.lm,newdata = test.set,interval="prediction",level = 0.95)   
   
 # calculate results  
 bias[i] <- mean(test.pred[,1] - test.set$on\_base\_percent)  
 rpmse[i] <- sqrt(mean((test.pred[,1] - test.set$on\_base\_percent)^2))  
 cvg[i] <- mean(test.pred[,2] < test.set$on\_base\_percent & test.pred[,3] > test.set$on\_base\_percent)  
 wid[i] <- mean(test.pred[,3]-test.pred[,2])   
}  
mean(bias)

## [1] -4.359154e-06

mean(rpmse)

## [1] 0.02803246

mean(cvg)

## [1] 0.9497887

mean(wid)

## [1] 0.1109161