SAL 608 Assignment 1

Andrew Fish

2025-10-21

#1.

```
##packages using the same ones from the assignment code plus performance as it contains vif and other m
library(readr)
library(GGally)
```

```
## Loading required package: ggplot2
```

```
library(ggfortify)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

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

```
library(MASS)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v stringr   1.5.2
## v forcats   1.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(regclass)
```

```
## Loading required package: bestglm
## Loading required package: leaps
## Loading required package: VGAM
## Loading required package: stats4
## Loading required package: splines
##
## Attaching package: 'VGAM'
##
## The following object is masked from 'package:lmtest':
##
##     lrtest
##
## Loading required package: rpart
## Loading required package: randomForest
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
##
## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

```
library(performance)
library(ggplot2)
```

```
##reading in the data
full_dat <- read_csv('data/ncaaSeason.csv')
```

```
## Rows: 2396 Columns: 61
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Team, League
## dbl (59): Season, #.x, GP, MPG, FGM, FGA, FG%, 3PM, 3PA, 3P%, FTM, FTA, FT%,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
##selecting wanted columns for modeling had to change from code in assignment pdf as weren't valid colu
dat <- full_dat %>%
  select(`Win %`,
         `FG%`,
         `TS%`,
         `ORB%`,
         `DRB%`,
```

```
        `AST%`,
        `TOV%`,
        `STL%`,
        `BLK%`,
        Pace,
        `FT/FGA`) %>%
  ##renaming to proper variable format
  rename(Win_Pct = `Win %`,
        FG_Pct = `FG%`,
        TS_Pct = `TS%`,
        ORBR = `ORB%`,
        DRBR = `DRB%`,
        ASTR = `AST%`,
        TOVR = `TOV%`,
        STLR = `STL%`,
        BLKR = `BLK%`,
        Pace = Pace,
        FT_FGA = `FT/FGA`)
```

The following chunk will model Win_Pct using FG_Pct in a simple linear regression model

```
##simple lin reg model Win_Pct v FG_Pct
mod1 <- lm(Win_Pct ~ FG_Pct,
           data = dat)

##calls summary of model
summary(mod1)
```

```
##
## Call:
## lm(formula = Win_Pct ~ FG_Pct, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38683 -0.08939  0.00317  0.09290  0.42601
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.36129    0.04644  -29.31   <2e-16 ***
## FG_Pct       4.25678    0.10552   40.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1339 on 2394 degrees of freedom
## Multiple R-squared:  0.4047, Adjusted R-squared:  0.4044
## F-statistic:  1627 on 1 and 2394 DF,  p-value: < 2.2e-16
```

From calling summary of our model we see that the R-squared is 0.4047 or 40.47%. This means that 40.47% of the variance in Win_Pct is explained by FG_Pct

#2.

```r
##MLR of dat modeling Win_Pct
mod2 <- lm(Win_Pct ~ .,
           data = dat)

summary(mod2)
```

```
##
## Call:
## lm(formula = Win_Pct ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32623 -0.06096  0.00133  0.06184  0.34803
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.0199640  0.0771356 -26.187  < 2e-16 ***
## FG_Pct       0.3181253  0.1892234   1.681   0.0929 .
## TS_Pct       2.5588443  0.1816918  14.083  < 2e-16 ***
## ORBR         0.0096026  0.0005123  18.744  < 2e-16 ***
## DRBR         0.0124162  0.0006470  19.191  < 2e-16 ***
## ASTR         0.0020392  0.0003867   5.274 1.46e-07 ***
## TOVR        -0.0242521  0.0010823 -22.408  < 2e-16 ***
## STLR         0.0243475  0.0011894  20.470  < 2e-16 ***
## BLKR         0.0123898  0.0007537  16.439  < 2e-16 ***
## Pace        -0.0040503  0.0005659  -7.157 1.10e-12 ***
## FT_FGA       0.3155432  0.0584232   5.401 7.28e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0894 on 2385 degrees of freedom
## Multiple R-squared:  0.7354, Adjusted R-squared:  0.7343
## F-statistic:   663 on 10 and 2385 DF,  p-value: < 2.2e-16
```

mod2 models Win_Pct using the rest of the variables within dat. The R-squared changes to 73.54% meaning that 73.54% of the variance in Win_Pct is explained by all other variables in the data. This value is more than mod1 which makes sense since we are using more predictors we should have more explained variance. Typically the more predictors you have the higher the R-squared which is why it isn't always the best judge of a model since it can be manipulated.

#3. Examining multicollinearity in mod2

```
##using performance package to check multicollinearity
check_collinearity(mod2)
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##    Term  VIF   VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
##    ORBR 1.37 [1.30, 1.44]     1.17      0.73     [0.69, 0.77]
##    DRBR 1.28 [1.22, 1.35]     1.13      0.78     [0.74, 0.82]
##    ASTR 1.26 [1.20, 1.33]     1.12      0.80     [0.75, 0.83]
##    TOVR 1.19 [1.15, 1.26]     1.09      0.84     [0.79, 0.87]
##    STLR 1.16 [1.12, 1.22]     1.08      0.86     [0.82, 0.90]
##    BLKR 1.21 [1.16, 1.27]     1.10      0.83     [0.79, 0.86]
##    Pace 1.07 [1.04, 1.13]     1.03      0.93     [0.88, 0.96]
##  FT_FGA 1.45 [1.38, 1.54]     1.21      0.69     [0.65, 0.72]
##
## Moderate Correlation
##
##    Term  VIF   VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
##  FG_Pct 7.21 [6.70, 7.77]     2.68      0.14     [0.13, 0.15]
##  TS_Pct 8.59 [7.97, 9.26]     2.93      0.12     [0.11, 0.13]
```

From our VIF results we see that FG_Pct and TS_Pct have problematic VIFs. The simple solution to this issue would be to drop one of the two from our regression. Since TS_Pct has the higher VIF we will drop that variable. So our new mod2 will be as follows:

```
##mod2 without TS_Pct
mod2_new <- lm(Win_Pct ~ FG_Pct + ORBR + DRBR + ASTR + TOVR + STLR + BLKR + Pace + FT_FGA,
           data = dat)

summary(mod2_new)
```

```
##
## Call:
## lm(formula = Win_Pct ~ FG_Pct + ORBR + DRBR + ASTR + TOVR + STLR +
##     BLKR + Pace + FT_FGA, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34599 -0.06259  0.00372  0.06389  0.31107
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.6884610  0.0764338 -22.091  < 2e-16 ***
## FG_Pct       2.7100593  0.0868069  31.219  < 2e-16 ***
## ORBR         0.0071878  0.0005023  14.309  < 2e-16 ***
## DRBR         0.0122293  0.0006731  18.170  < 2e-16 ***
## ASTR         0.0035415  0.0003867   9.158  < 2e-16 ***
## TOVR        -0.0274782  0.0011007 -24.965  < 2e-16 ***
## STLR         0.0236731  0.0012366  19.144  < 2e-16 ***
```

```
## BLKR         0.0110245  0.0007777  14.175  < 2e-16 ***
## Pace        -0.0045148  0.0005879  -7.680 2.31e-14 ***
## FT_FGA       0.7168488  0.0530703  13.508  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09303 on 2386 degrees of freedom
## Multiple R-squared:  0.7134, Adjusted R-squared:  0.7124
## F-statistic:   660 on 9 and 2386 DF,  p-value: < 2.2e-16
```

#4. Splitting the Data

```
set.seed(23) ##MJ
##size of dat
n <- nrow(dat)
##proportion to use for training data using 75%
prop <- 0.75

##random sample of dat
train <- sample(n, size = n * prop)

##splitting to trainng and test
dat_train <- dat[train, ]
dat_test <- dat[-train, ]
```

Rerunning the models

```
##simple lin reg model Win_Pct v FG_Pct
mod1 <- lm(Win_Pct ~ FG_Pct,
           data = dat_train)

##calls summary of model
summary(mod1)
```

```
##
## Call:
## lm(formula = Win_Pct ~ FG_Pct, data = dat_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.38637 -0.08538  0.00363  0.09023  0.42462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.34477    0.05347  -25.15   <2e-16 ***
## FG_Pct       4.21987    0.12144   34.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.133 on 1795 degrees of freedom
## Multiple R-squared:  0.4021, Adjusted R-squared:  0.4018
## F-statistic:  1207 on 1 and 1795 DF,  p-value: < 2.2e-16
```

```
##mod2 without TS_Pct
mod2_new <- lm(Win_Pct ~ FG_Pct + ORBR + DRBR + ASTR + TOVR + STLR + BLKR + Pace + FT_FGA,
           data = dat_train)

summary(mod2_new)
```

```
##
## Call:
## lm(formula = Win_Pct ~ FG_Pct + ORBR + DRBR + ASTR + TOVR + STLR +
```

```
##     BLKR + Pace + FT_FGA, data = dat_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34526 -0.06214  0.00291  0.06280  0.30843
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.7338557  0.0878786 -19.730  < 2e-16 ***
## FG_Pct       2.7928576  0.0984432  28.370  < 2e-16 ***
## ORBR         0.0070745  0.0005815  12.166  < 2e-16 ***
## DRBR         0.0117605  0.0007634  15.405  < 2e-16 ***
## ASTR         0.0035192  0.0004383   8.028 1.77e-15 ***
## TOVR        -0.0267035  0.0012703 -21.022  < 2e-16 ***
## STLR         0.0225704  0.0014244  15.845  < 2e-16 ***
## BLKR         0.0115537  0.0008966  12.886  < 2e-16 ***
## Pace        -0.0038967  0.0006673  -5.839 6.22e-09 ***
## FT_FGA       0.7056469  0.0611432  11.541  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09263 on 1787 degrees of freedom
## Multiple R-squared:  0.7112, Adjusted R-squared:  0.7097
## F-statistic: 488.9 on 9 and 1787 DF,  p-value: < 2.2e-16
```

```r
mod3 <- lm(Win_Pct ~ TS_Pct + ORBR + TOVR + Pace + FT_FGA,
           data = dat_train)

summary(mod3)
```

```
##
## Call:
## lm(formula = Win_Pct ~ TS_Pct + ORBR + TOVR + Pace + FT_FGA,
##     data = dat_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37728 -0.07215  0.00082  0.07596  0.33084
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.0610981  0.0837727 -12.666  < 2e-16 ***
## TS_Pct       3.4569463  0.0969783  35.647  < 2e-16 ***
## ORBR         0.0150731  0.0006284  23.986  < 2e-16 ***
## TOVR        -0.0272815  0.0014323 -19.048  < 2e-16 ***
## Pace        -0.0048791  0.0007492  -6.513 9.55e-11 ***
## FT_FGA       0.1629539  0.0733038   2.223   0.0263 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1057 on 1791 degrees of freedom
## Multiple R-squared:  0.6231, Adjusted R-squared:  0.622
## F-statistic: 592.1 on 5 and 1791 DF,  p-value: < 2.2e-16
```

Calculating RMSE of the 3 models

```
rmse(mod1)
```

```
## [1] 0.1329027
```

```
rmse(mod2_new)
```

```
## [1] 0.09237533
```

```
rmse(mod3)
```

```
## [1] 0.1055272
```

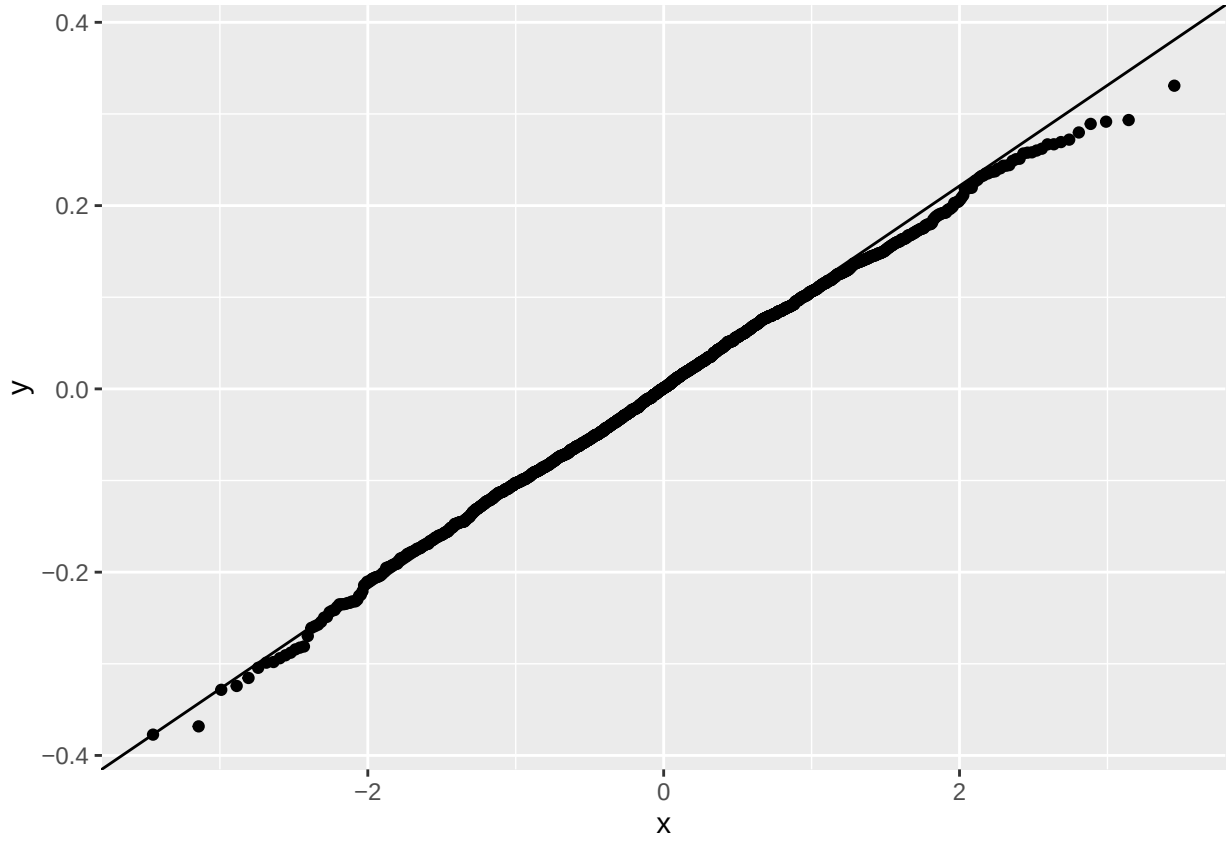
mod2_new which is the model using all predictors besides TS_Pct has the lowest test RMSE with 0.0924

#5.

The slope of TS_Pct in mod3 means that for every 1 percent change in TS_Pct Win_Pct changes 3.4569 percent. If TS_Pct increase so will Win_Pct.

#6. Normality

```
##informal test
mod3_resid <- data.frame(resid = mod3$residuals)
ggplot(mod3_resid, aes(sample = resid)) +
  stat_qq() + stat_qq_line()
```




```
##formal test using Shapiro-Wilk Test
##sig level 0.15
shapiro.test(mod3_resid$resid)
```

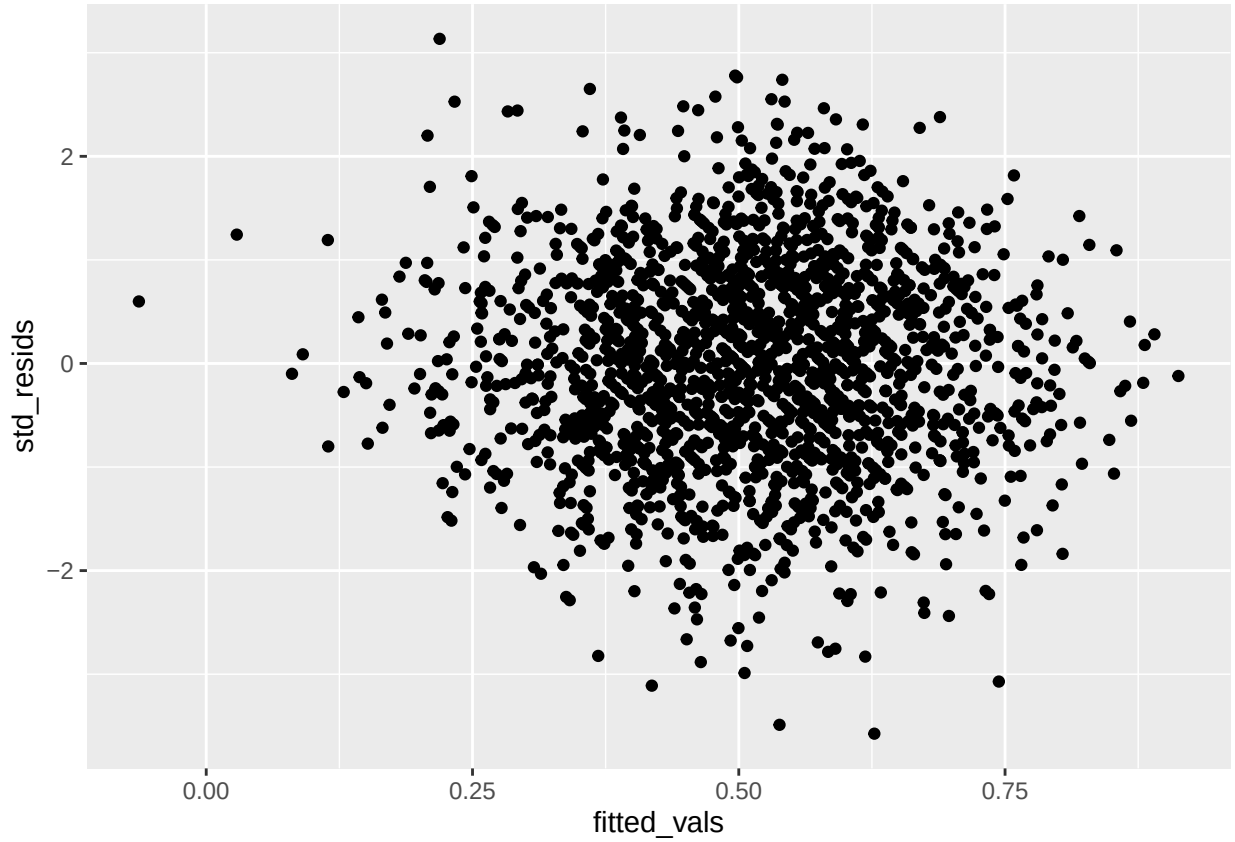
```
##
##  Shapiro-Wilk normality test
##
## data:  mod3_resid$resid
## W = 0.99864, p-value = 0.1685
```

Fail to reject the null, could potentially be an issue with significance level closer to 0.2. Also using the visual test the upper tail begins to diverge from the normal line.

Constant Variance

```
##visual test
data.frame(fitted_vals = mod3$fitted.values,
           std_resids = scale(mod3$residuals)) %>%
```

```
  ggplot(aes(fitted_vals, std_resids)) +
  geom_point()
```



There are possible issues as fitted_vals approach 0 and 1. At the edges it looks to funnel both in and out. This could only appear this way since using 75% sample test data. But if it is a problem the way to fix it would be to transform Win_Pct using a log transform, natural log transom, or a sqaure root transform.

Independence

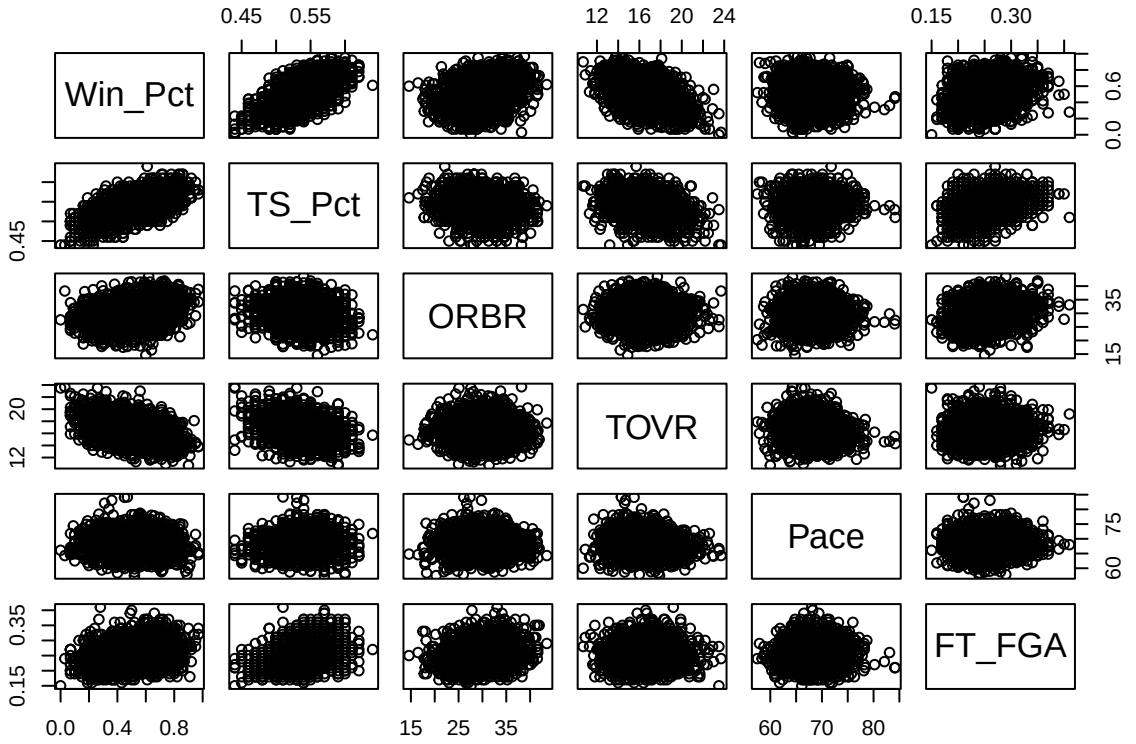
```
##sig level at 0.15
dwtest(mod3, alternative = 'two.sided')
```

```
##
##  Durbin-Watson test
##
## data:  mod3
## DW = 2.0197, p-value = 0.6778
## alternative hypothesis: true autocorrelation is not 0
```

Fail to reject the null

Linearity

```
mod3_dat <- dat[, c("Win_Pct", "TS_Pct", "ORBR", "TOVR", "Pace", "FT_FGA")]
pairs(mod3_dat)
```



From this plot matrix we can see the Win_Pct has a general linear relationship with all the variables.

#7.

```r
##since needing to provide a range of win percentage and not average win percentage will use a predicti
new <- data.frame(TS_Pct = 0.55,
                  ORBR = 26.7,
                  TOVR = 21.2,
                  Pace = 74.3,
                  FT_FGA = 0.24) ##cahnged ORBR since in percent format in data
predict(mod3, newdata = new, interval = 'prediction')
```

```
##         fit       lwr       upr
## 1 0.3408956 0.1327514 0.5490398
```

The range of win percentage would be 13.28%-54.9%