SAL 608 Assignment 2

Andrew Fish

2025-11-02

```r
##packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.2
## v ggplot2   4.0.0     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()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(readr)
library(performance)
library(lmtest)
```

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

```r
library(ggplot2)
library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.5.2
```

```r
##reading in file
keys <- read_csv('data/keys_to_the_game.csv', show_col_types = FALSE)
```

#1.

```r
##summary of opponent stats
keys %>%
  select(contains('opp_')) %>%
  summary()
```

```
##     opp_fga       opp_fg_pct       opp_fga3      opp_fg3_pct
##  Min.   :38.00   Min.   :23.10   Min.   : 5.00   Min.   : 0.00
##  1st Qu.:53.00   1st Qu.:38.30   1st Qu.:18.50   1st Qu.:26.70
##  Median :57.00   Median :42.90   Median :22.00   Median :33.30
##  Mean   :57.61   Mean   :43.03   Mean   :22.23   Mean   :33.63
##  3rd Qu.:62.00   3rd Qu.:47.50   3rd Qu.:26.00   3rd Qu.:40.70
##  Max.   :86.00   Max.   :64.50   Max.   :50.00   Max.   :72.00
##     opp_fta       opp_ft_pct       opp_orb        opp_drb         opp_ast
##  Min.   : 2.00   Min.   :  0.0   Min.   : 2.0   Min.   :13.00   Min.   : 2.00
##  1st Qu.:12.00   1st Qu.: 61.5   1st Qu.: 8.0   1st Qu.:21.00   1st Qu.:10.00
##  Median :17.00   Median : 71.4   Median :10.0   Median :24.00   Median :12.00
##  Mean   :17.23   Mean   : 70.1   Mean   :10.3   Mean   :24.55   Mean   :12.61
##  3rd Qu.:21.00   3rd Qu.: 80.0   3rd Qu.:13.0   3rd Qu.:28.00   3rd Qu.:15.00
##  Max.   :44.00   Max.   :100.0   Max.   :23.0   Max.   :41.00   Max.   :28.00
##     opp_stl          opp_blk          opp_tov
##  Min.   : 0.000   Min.   : 0.000   Min.   : 2.0
##  1st Qu.: 4.000   1st Qu.: 2.000   1st Qu.:10.0
##  Median : 6.000   Median : 3.000   Median :12.0
##  Mean   : 6.028   Mean   : 3.409   Mean   :12.7
##  3rd Qu.: 8.000   3rd Qu.: 5.000   3rd Qu.:15.0
##  Max.   :17.000   Max.   :13.000   Max.   :26.0
```

```r
##summary non opponent stats
keys %>%
  select(!(contains('opp_'))) %>%
  summary()
```

```
##       fga            fg_pct           fga3          fg3_pct
##  Min.   :38.00   Min.   :16.70   Min.   : 7.00   Min.   : 6.70
##  1st Qu.:53.00   1st Qu.:39.70   1st Qu.:19.00   1st Qu.:27.30
##  Median :57.00   Median :44.00   Median :22.00   Median :34.80
##  Mean   :57.34   Mean   :44.04   Mean   :22.21   Mean   :34.41
##  3rd Qu.:61.00   3rd Qu.:49.10   3rd Qu.:26.00   3rd Qu.:41.30
##  Max.   :82.00   Max.   :68.00   Max.   :43.00   Max.   :71.40
##       fta            ft_pct            orb             drb
##  Min.   : 2.00   Min.   :  0.00   Min.   : 1.00   Min.   :10.0
##  1st Qu.:15.00   1st Qu.: 64.15   1st Qu.: 7.00   1st Qu.:21.5
##  Median :19.00   Median : 72.20   Median :10.00   Median :25.0
##  Mean   :19.75   Mean   : 71.60   Mean   :10.33   Mean   :25.0
##  3rd Qu.:24.00   3rd Qu.: 79.30   3rd Qu.:13.00   3rd Qu.:28.0
##  Max.   :46.00   Max.   :100.00   Max.   :26.00   Max.   :41.0
##       ast             stl              blk              tov
##  Min.   : 2.00   Min.   : 0.000   Min.   : 0.000   Min.   : 3.0
##  1st Qu.:11.00   1st Qu.: 4.000   1st Qu.: 2.000   1st Qu.: 9.0
##  Median :13.00   Median : 6.000   Median : 4.000   Median :11.0
##  Mean   :13.71   Mean   : 6.429   Mean   : 3.939   Mean   :11.8
##  3rd Qu.:16.00   3rd Qu.: 8.000   3rd Qu.: 5.000   3rd Qu.:14.0
##  Max.   :29.00   Max.   :15.000   Max.   :14.000   Max.   :24.0
##     win
##  Mode :logical
##  FALSE:296
##  TRUE :443
##
##
```

```
##
```

```r
##win percentage
mean(keys$win)
```

```
## [1] 0.5994587
```

By splitting the summary statistics up by team we can see that the opponent tends to have worse versions (lower values for points and higher for turnovers). This makes sense as overall the opposing team losses roughly 40%

#2.

```r
##log model win v everything
log_mod <- glm(as.numeric(win) ~ .,
               data = keys,
               family = binomial())
summary(log_mod)
```

```
##
## Call:
## glm(formula = as.numeric(win) ~ ., family = binomial(), data = keys)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.77415    6.32368   0.597 0.550622
## fga          0.39403    0.13574   2.903 0.003698 **
## fg_pct       0.61928    0.10933   5.664 1.48e-08 ***
## fga3         0.07333    0.04818   1.522 0.127966
## fg3_pct      0.16702    0.03270   5.108 3.25e-07 ***
## fta          0.35341    0.07236   4.884 1.04e-06 ***
## ft_pct       0.06518    0.02251   2.896 0.003782 **
## orb          0.13031    0.17184   0.758 0.448255
## drb          0.32038    0.13169   2.433 0.014979 *
## ast         -0.15247    0.07430  -2.052 0.040163 *
## stl          0.06398    0.10648   0.601 0.547929
## blk          0.17774    0.09598   1.852 0.064047 .
## tov         -0.25728    0.12713  -2.024 0.042990 *
## opp_fga     -0.62375    0.15421  -4.045 5.24e-05 ***
## opp_fg_pct  -0.51432    0.09897  -5.197 2.03e-07 ***
## opp_fga3    -0.15538    0.04569  -3.400 0.000673 ***
## opp_fg3_pct -0.14940    0.02626  -5.690 1.27e-08 ***
## opp_fta     -0.49800    0.08380  -5.942 2.81e-09 ***
## opp_ft_pct  -0.05976    0.02016  -2.964 0.003041 **
## opp_orb      0.14832    0.17801   0.833 0.404727
## opp_drb     -0.01214    0.13765  -0.088 0.929733
## opp_ast      0.07284    0.06662   1.093 0.274233
## opp_stl      0.07587    0.10415   0.728 0.466319
## opp_blk     -0.31028    0.12132  -2.558 0.010539 *
## opp_tov      0.13057    0.13805   0.946 0.344242
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 995.03  on 738  degrees of freedom
## Residual deviance: 193.35  on 714  degrees of freedom
## AIC: 243.35
##
## Number of Fisher Scoring iterations: 8
```

```r
##predicted values from the model attaching to original data
expected <- keys %>%
  mutate(ExpectedProb = round(predict(log_mod, type = 'response'), 8))
```

To test validity of model we will check that all assumptions hold

```r
##binary response
unique(keys$win)
```

```
## [1] FALSE  TRUE
```

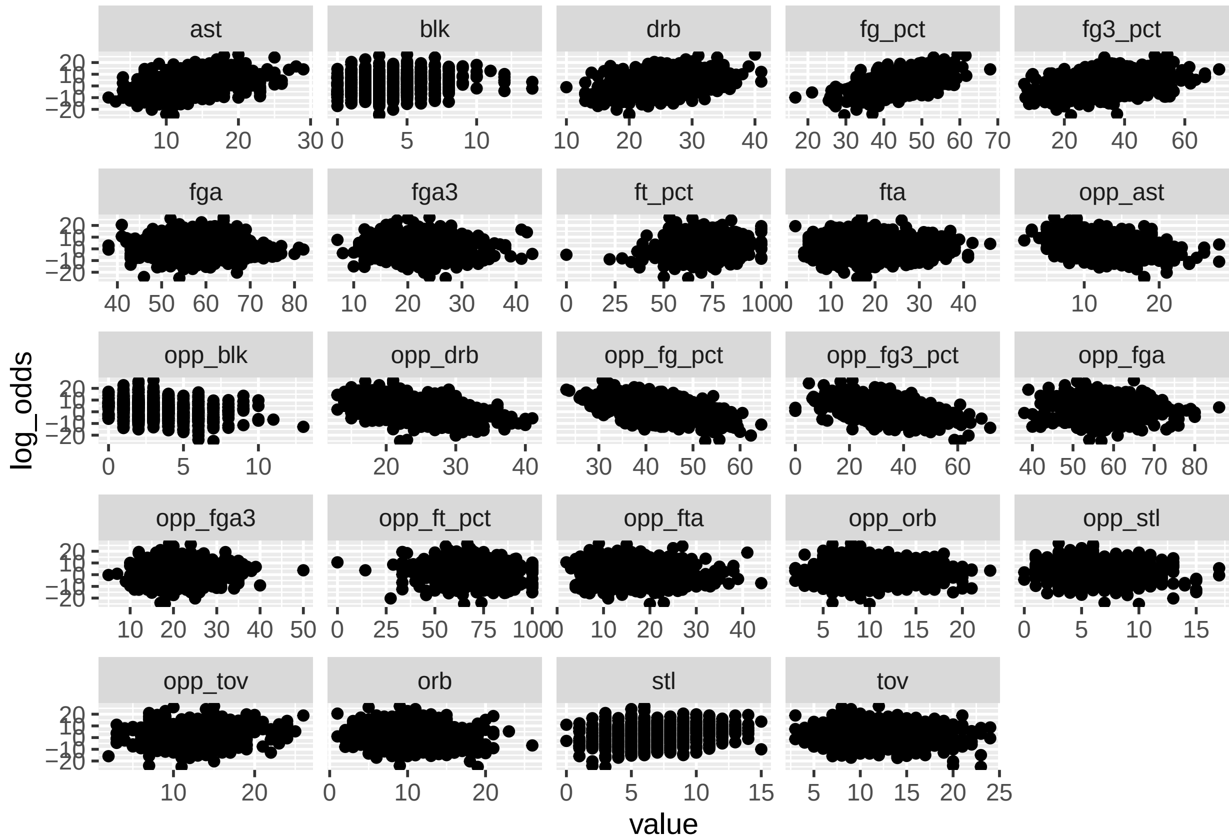
We have false and true as the two values of the win variable so it is binary.

```r
##independence
dwtest(log_mod, alternative = 'two.sided')
```

```
##
##  Durbin-Watson test
##
## data:  log_mod
## DW = 2.0074, p-value = 0.9146
## alternative hypothesis: true autocorrelation is not 0
```

No issues with independence

```r
##graphs log odds against all variables to visualize if linear relationship
keys %>%
  select(fga:opp_tov) %>%
  mutate(log_odds = log(log_mod$fitted.values / (1 - log_mod$fitted.values))) %>%
  pivot_longer(-log_odds) %>%
  ggplot(aes(value, log_odds)) +
  geom_point() +
  facet_wrap(~ name, scales = 'free_x')
```



All variables are linear with log odds so no issues

```r
##sample size
nrow(keys) > 10 * ((length(keys) - 1) / min(mean(keys$win), 1 - mean(keys$win)))
```

```
## [1] TRUE
```

Sample is large enough

For this question since only looking at the predictions and their validity we do not need to test for multicollinearity. Based on the assumptions this model and its predictions appear to be valid.

#3.

```r
##from performance package
check_collinearity(log_mod)
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##          Term  VIF     VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
##          fga3 1.84 [ 1.67,  2.05]     1.36      0.54     [0.49, 0.60]
##       fg3_pct 2.37 [ 2.14,  2.66]     1.54      0.42     [0.38, 0.47]
##        ft_pct 2.17 [ 1.96,  2.43]     1.47      0.46     [0.41, 0.51]
##           ast 2.30 [ 2.07,  2.58]     1.52      0.43     [0.39, 0.48]
##           stl 2.11 [ 1.91,  2.36]     1.45      0.47     [0.42, 0.52]
##           blk 1.39 [ 1.29,  1.54]     1.18      0.72     [0.65, 0.78]
##      opp_fga3 2.06 [ 1.87,  2.31]     1.44      0.48     [0.43, 0.54]
##   opp_fg3_pct 2.01 [ 1.82,  2.24]     1.42      0.50     [0.45, 0.55]
##    opp_ft_pct 1.78 [ 1.62,  1.98]     1.33      0.56     [0.50, 0.62]
##       opp_ast 1.81 [ 1.65,  2.02]     1.35      0.55     [0.50, 0.61]
##       opp_stl 2.47 [ 2.22,  2.77]     1.57      0.40     [0.36, 0.45]
##       opp_blk 1.67 [ 1.52,  1.85]     1.29      0.60     [0.54, 0.66]
##
## Moderate Correlation
##
##         Term  VIF     VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
##          fta 8.38 [ 7.37,  9.56]     2.90      0.12     [0.10, 0.14]
##          tov 7.18 [ 6.32,  8.18]     2.68      0.14     [0.12, 0.16]
##   opp_fg_pct 8.62 [ 7.57,  9.83]     2.94      0.12     [0.10, 0.13]
##      opp_fta 7.16 [ 6.31,  8.16]     2.68      0.14     [0.12, 0.16]
##      opp_tov 7.30 [ 6.42,  8.31]     2.70      0.14     [0.12, 0.16]
##
## High Correlation
##
##      Term   VIF     VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
##       fga 27.90 [24.35, 31.99]     5.28      0.04     [0.03, 0.04]
##    fg_pct 10.17 [ 8.92, 11.62]     3.19      0.10     [0.09, 0.11]
##       orb 13.87 [12.14, 15.86]     3.72      0.07     [0.06, 0.08]
##       drb 11.72 [10.27, 13.40]     3.42      0.09     [0.07, 0.10]
##   opp_fga 37.91 [33.06, 43.48]     6.16      0.03     [0.02, 0.03]
##   opp_orb 12.95 [11.34, 14.81]     3.60      0.08     [0.07, 0.09]
##   opp_drb 11.53 [10.11, 13.18]     3.40      0.09     [0.08, 0.10]
```

Examining the VIF for the variables we can see that there are plenty of terms with isses. For the most part the opp versions of the terms have issues. From our initial investigation I am deciding to remove the opp versions of the terms from the problematic zones above to account for correlation issues. Would remove all opp variables but don't want to commit ommited variable bias

```r
##removing unwanted variables that are correlation issues
new_data <- keys %>%
  select(!(c(opp_fga, opp_fg_pct, opp_fta, opp_tov, opp_drb, opp_orb)))
```

```r
##reruning with updated data
new_log <- glm(win ~ .,
               data = new_data,
               family = binomial())
summary(new_log)
```

```
##
## Call:
## glm(formula = win ~ ., family = binomial(), data = new_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.9687121  2.4424819  -3.672 0.000241 ***
## fga         -0.1760862  0.0328527  -5.360 8.33e-08 ***
## fg_pct       0.2424885  0.0348233   6.963 3.32e-12 ***
## fga3        -0.0008706  0.0283566  -0.031 0.975507
## fg3_pct      0.0766011  0.0183162   4.182 2.89e-05 ***
## fta         -0.0039545  0.0197782  -0.200 0.841524
## ft_pct       0.0479421  0.0114440   4.189 2.80e-05 ***
## orb          0.3938041  0.0570794   6.899 5.23e-12 ***
## drb          0.3004725  0.0416743   7.210 5.59e-13 ***
## ast          0.0326566  0.0477056   0.685 0.493631
## stl          0.4541729  0.0589722   7.701 1.35e-14 ***
## blk          0.0720217  0.0576734   1.249 0.211744
## tov         -0.3753132  0.0582812  -6.440 1.20e-10 ***
## opp_fga3    -0.0128930  0.0260094  -0.496 0.620104
## opp_fg3_pct -0.0951500  0.0151304  -6.289 3.20e-10 ***
## opp_ft_pct  -0.0206414  0.0105561  -1.955 0.050536 .
## opp_ast     -0.1161702  0.0422820  -2.748 0.006005 **
## opp_stl     -0.0423176  0.0664495  -0.637 0.524230
## opp_blk     -0.2410310  0.0724570  -3.327 0.000879 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 995.03  on 738  degrees of freedom
## Residual deviance: 413.86  on 720  degrees of freedom
## AIC: 451.86
##
## Number of Fisher Scoring iterations: 6
```

```r
##double checking
check_collinearity(new_log)
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##         Term  VIF   VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
##          fga 3.00 [2.68, 3.38]     1.73      0.33     [0.30, 0.37]
##       fg_pct 2.44 [2.19, 2.74]     1.56      0.41     [0.37, 0.46]
##         fga3 1.58 [1.44, 1.75]     1.26      0.63     [0.57, 0.69]
```

```
##      fg3_pct 1.70 [1.55, 1.89]     1.30       0.59      [0.53, 0.64]
##          fta 1.23 [1.15, 1.36]     1.11       0.81      [0.73, 0.87]
##       ft_pct 1.09 [1.04, 1.23]     1.05       0.91      [0.81, 0.96]
##          orb 2.97 [2.65, 3.35]     1.72       0.34      [0.30, 0.38]
##          drb 2.28 [2.06, 2.56]     1.51       0.44      [0.39, 0.49]
##          ast 2.05 [1.85, 2.29]     1.43       0.49      [0.44, 0.54]
##          stl 1.60 [1.47, 1.78]     1.27       0.62      [0.56, 0.68]
##          blk 1.12 [1.06, 1.25]     1.06       0.89      [0.80, 0.95]
##          tov 2.96 [2.64, 3.33]     1.72       0.34      [0.30, 0.38]
##     opp_fga3 1.29 [1.19, 1.42]     1.13       0.78      [0.70, 0.84]
##  opp_fg3_pct 1.41 [1.30, 1.56]     1.19       0.71      [0.64, 0.77]
##   opp_ft_pct 1.14 [1.07, 1.26]     1.07       0.88      [0.79, 0.93]
##      opp_ast 1.44 [1.32, 1.59]     1.20       0.70      [0.63, 0.76]
##      opp_stl 2.12 [1.91, 2.37]     1.46       0.47      [0.42, 0.52]
##      opp_blk 1.39 [1.29, 1.54]     1.18       0.72      [0.65, 0.78]
```

#4.

```r
##new df that has required variables
data <- keys %>%
  select(fg_pct, fg3_pct, ft_pct, fta, orb, tov, opp_fg_pct, opp_fg3_pct, opp_ft_pct, opp_fta, opp_orb,
```

```r
lim_log <- glm(win ~ .,
               data = data,
               family = binomial())
summary(lim_log)
```

```
##
## Call:
## glm(formula = win ~ ., family = binomial(), data = data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.01829    2.53713   1.584    0.113
## fg_pct       0.45773    0.05197   8.807  < 2e-16 ***
## fg3_pct      0.09958    0.02241   4.444 8.81e-06 ***
## ft_pct       0.06139    0.01513   4.057 4.97e-05 ***
## fta          0.12048    0.02604   4.626 3.73e-06 ***
## orb          0.35593    0.05926   6.006 1.90e-09 ***
## tov         -0.45610    0.05978  -7.630 2.36e-14 ***
## opp_fg_pct  -0.48166    0.05342  -9.016  < 2e-16 ***
## opp_fg3_pct -0.11884    0.01969  -6.035 1.59e-09 ***
## opp_ft_pct  -0.06305    0.01512  -4.169 3.06e-05 ***
## opp_fta     -0.19384    0.03198  -6.061 1.35e-09 ***
## opp_orb     -0.44898    0.06574  -6.830 8.50e-12 ***
## opp_tov      0.45384    0.06026   7.531 5.03e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 995.03  on 738  degrees of freedom
## Residual deviance: 244.54  on 726  degrees of freedom
## AIC: 270.54
##
## Number of Fisher Scoring iterations: 8
```

```r
unique(keys$win)
```

```
## [1] FALSE  TRUE
```

binary response variable holds

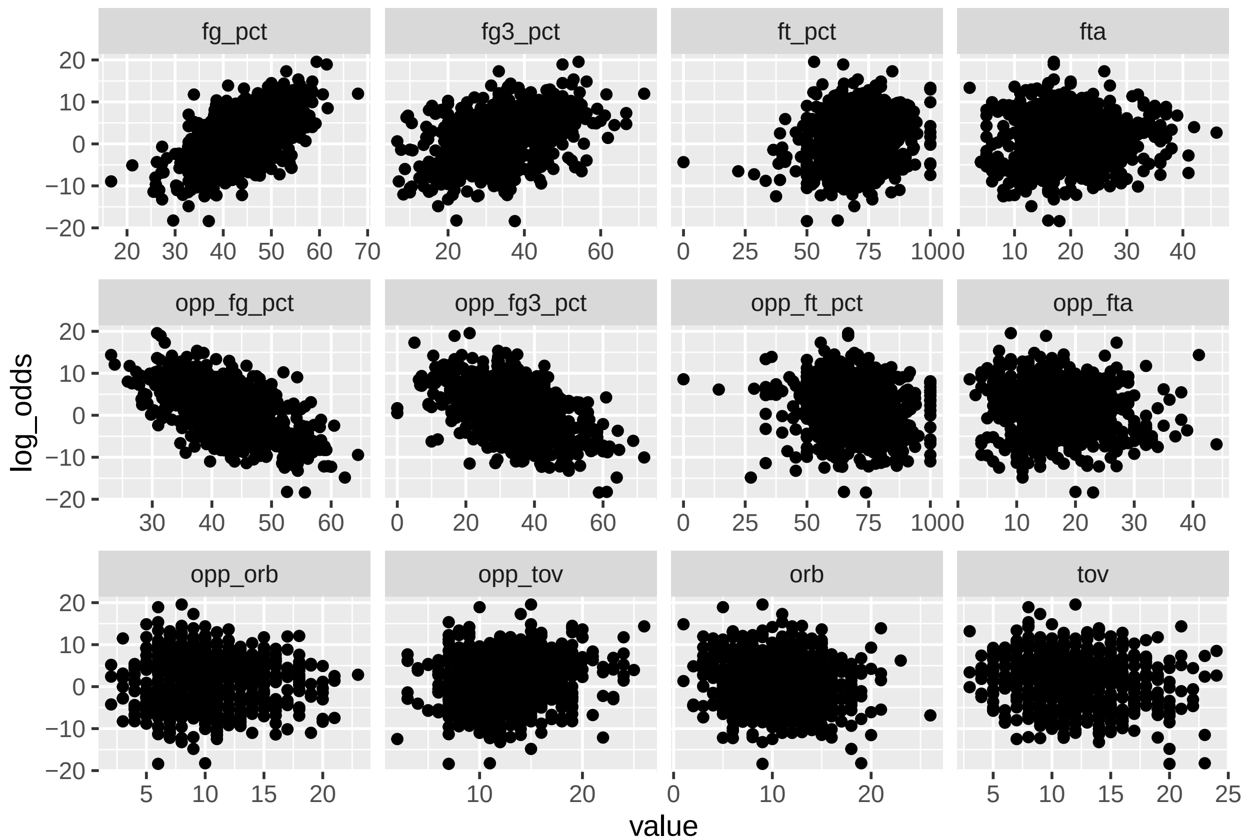
```r
dwtest(lim_log, alternative = 'two.sided')
```

```
##
##  Durbin-Watson test
```

```
##
## data:  lim_log
## DW = 2.0137, p-value = 0.8485
## alternative hypothesis: true autocorrelation is not 0
```

No independence issues

```
##log odds relationship between variables
data %>%
  select(!win) %>%
  mutate(log_odds = log(lim_log$fitted.values / (1 - lim_log$fitted.values))) %>%
  pivot_longer(-log_odds) %>%
  ggplot(aes(value, log_odds)) +
  geom_point() +
  facet_wrap(~ name, scales = 'free_x')
```



All show linear relationship

```
##checking sample size
nrow(data) > 10 * ((length(data) - 1) / min(mean(data$win), 1 - mean(data$win)))
```

```
## [1] TRUE
```

Large enough sample

```r
check_collinearity(lim_log)
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##          Term  VIF   VIF 95% CI adj. VIF Tolerance Tolerance 95% CI
##        fg_pct 2.97 [2.65, 3.35]     1.72      0.34     [0.30, 0.38]
##       fg3_pct 1.55 [1.42, 1.72]     1.24      0.65     [0.58, 0.70]
##        ft_pct 1.11 [1.05, 1.24]     1.05      0.90     [0.81, 0.95]
##           fta 1.28 [1.19, 1.41]     1.13      0.78     [0.71, 0.84]
##           orb 1.85 [1.68, 2.06]     1.36      0.54     [0.48, 0.60]
##           tov 1.90 [1.72, 2.12]     1.38      0.53     [0.47, 0.58]
##    opp_fg_pct 3.22 [2.87, 3.64]     1.80      0.31     [0.27, 0.35]
##   opp_fg3_pct 1.43 [1.32, 1.59]     1.20      0.70     [0.63, 0.76]
##    opp_ft_pct 1.26 [1.17, 1.39]     1.12      0.79     [0.72, 0.85]
##       opp_fta 1.35 [1.25, 1.50]     1.16      0.74     [0.67, 0.80]
##       opp_orb 2.35 [2.11, 2.64]     1.53      0.43     [0.38, 0.47]
##       opp_tov 1.77 [1.61, 1.97]     1.33      0.56     [0.51, 0.62]
```

No correlation issues

#5.

```r
##already have the data set so don't need to create a new one
##will run the probit model and compare accuracy using performance package
prob_mod <- glm(win ~.,
                data = data,
                family = binomial(link = 'probit'))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(prob_mod)
```

```
##
## Call:
## glm(formula = win ~ ., family = binomial(link = "probit"), data = data)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.326500   1.253892   1.855 0.063536 .
## fg_pct       0.214752   0.022158   9.692  < 2e-16 ***
## fg3_pct      0.040965   0.010535   3.889 0.000101 ***
## ft_pct       0.028424   0.007331   3.877 0.000106 ***
## fta          0.053193   0.012237   4.347 1.38e-05 ***
## orb          0.162694   0.026876   6.054 1.42e-09 ***
## tov         -0.210840   0.026640  -7.914 2.48e-15 ***
## opp_fg_pct  -0.220163   0.022035  -9.992  < 2e-16 ***
## opp_fg3_pct -0.057676   0.009406  -6.132 8.70e-10 ***
## opp_ft_pct  -0.030424   0.007239  -4.203 2.63e-05 ***
## opp_fta     -0.106442   0.015480  -6.876 6.16e-12 ***
## opp_orb     -0.200380   0.029291  -6.841 7.86e-12 ***
## opp_tov      0.203300   0.026428   7.693 1.44e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 995.03  on 738  degrees of freedom
## Residual deviance: 269.39  on 726  degrees of freedom
## AIC: 295.39
##
## Number of Fisher Scoring iterations: 8
```

```r
##default method for performance_accuracy() is k folds CV
##logit first
performance_accuracy(lim_log)
```

```
## # Accuracy of Model Predictions
##
## Accuracy (95% CI): 97.73% [97.03%, 98.12%]
## Method: Area under Curve
```

```r
##probit
performance_accuracy(prob_mod)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## # Accuracy of Model Predictions
##
## Accuracy (95% CI): 97.75% [96.26%, 99.24%]
## Method: Area under Curve
```

Testing another form of cv using training and test data sets

```r
set.seed(18) ##Yamamoto WS MVP
n <- nrow(data)
prop <- .6
train <- sample(n, size = n * prop)
train_data <- data[train, ]
test_data <- data[-train, ]
```

Rerunning logit and probit using training data

```r
logit <- glm(win ~.,
             data = train_data,
             family = binomial())

probit <- glm(win ~ .,
              data = train_data,
              family = binomial(link = 'probit'))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
test_data %>%
  mutate(log_pred = predict(logit,
                            .,
                            type = 'response'),
         pro_pred = predict(probit,
                            .,
                            type = 'response')) %>%
  summarize(log_brier = BrierScore(win, log_pred),
            pro_brier = BrierScore(win, pro_pred))
```

```
## # A tibble: 1 x 2
##   log_brier pro_brier
##       <dbl>     <dbl>
## 1    0.0506    0.0534
```

Logit and Probit created similar models in terms of accuracy, but according to their Brier Scores we should be using the logit model since it has the lower score. As well looking at the confidence intervals from the accuracy we see that the logit has a higher lower and upper bound for the 95% CI.