IDS 702 HW 4 - KEY

Your Name Here

Instructions: Use this template to complete your assignment. When you click "Render," you should get a PDF document that contains both your answers and code. You must show your work/justify your answers to receive credit. Submit your rendered PDF file on Gradescope. **Remember to render frequently**, as this will help you to catch errors in your code before the last minute.

Add your name in the Author section in the header

Load data

```
library(tidyverse)
library(modelsummary)
library(caret)
library(pROC)

nba <- read.csv("https://raw.githubusercontent.com/anlane611/datasets/refs/heads/main/nba_gar
```

Exercise 1

```
## create subset
nba_cho <- nba |> filter(Team=="CHO")

## create factor variable for Home
nba_cho$Home <- factor(nba_cho$Home)

## create new Win variable (include quality control check)
nba_cho$Win <- ifelse(nba_cho$WINorLOSS=="W",1,0)
nba_cho$Win <- factor(nba_cho$Win,
```

```r
                       levels = c(0,1),
                       labels = c("Loss","Win"))
table(nba_cho$WINorLOSS, nba_cho$Win)
```

```
    Loss Win
  L  175   0
  W    0 153
```

```r
## format date variable
nba_cho <- nba_cho |>
  mutate(Date_clean = as.Date(Date, "%Y-%m-%d"))
```

Exercise 2

```r
## code to fill in the table
nba_cho |> count(Win)
```

```
   Win   n
1 Loss 175
2  Win 153
```

```r
nba_cho |>
  group_by(Win) |>
  count(Home) |>
  mutate(perc=n/sum(n)*100)
```

```
# A tibble: 4 x 4
# Groups:   Win [2]
  Win   Home      n  perc
  <fct> <fct> <int> <dbl>
1 Loss  Away    103  58.9
2 Loss  Home     72  41.1
3 Win   Away     61  39.9
4 Win   Home     92  60.1
```

```
nba_cho |>
  group_by(Win) |>
  summarise(mean(TeamPoints), sd(TeamPoints),
            mean(FieldGoals.), sd(FieldGoals.),
            mean(Assists), sd(Assists),
            mean(Steals), sd(Steals),
            mean(Blocks), sd(Blocks),
            mean(OpponentPoints), sd(OpponentPoints),
            mean(TotalRebounds), sd(TotalRebounds),
            mean(Turnovers), sd(Turnovers))
```

```
# A tibble: 2 x 17
  Win   `mean(TeamPoints)` `sd(TeamPoints)` `mean(FieldGoals.)` `sd(FieldGoals~`
  <fct>              <dbl>            <dbl>               <dbl>            <dbl>
1 Loss                96.9             11.3               0.417           0.0489
2 Win                109.              11.4               0.463           0.0513
# ... with 12 more variables: `mean(Assists)` <dbl>, `sd(Assists)` <dbl>,
#   `mean(Steals)` <dbl>, `sd(Steals)` <dbl>, `mean(Blocks)` <dbl>,
#   `sd(Blocks)` <dbl>, `mean(OpponentPoints)` <dbl>,
#   `sd(OpponentPoints)` <dbl>, `mean(TotalRebounds)` <dbl>,
#   `sd(TotalRebounds)` <dbl>, `mean(Turnovers)` <dbl>, `sd(Turnovers)` <dbl>
```

Variable	Wins (N= 153)	Losses (N= ___)
Home games - N (%)	92 (60)	
Team Points - mean (SD)	109 (11.4)	
Field Goal Percentage - mean (SD)		
Assists - mean (SD)		
Steals - mean (SD)		
Blocks - mean (SD)		
Opponent Points - mean (SD)		
Total Rebounds - mean (SD)		
Turnovers - mean (SD)		

Exercise 3

Similar statistics likely to be highly correlated which could lead to issues with multicollinearity. E.g., Opp.X3PointShots, Opp.X3PointShotsAttempted, Opp.X3PointShots.

	(1)				
	Est.	S.E.	2.5 %	97.5 %	p
(Intercept)	1.2×10^{-13}	3.8×10^{-13}	1.0×10^{-16}	5.2×10^{-11}	<0.01
HomeHome	2.46	0.83	1.28	4.82	<0.01
TeamPoints	0.998	0.022	0.955	1.043	0.92
FieldGoals.	1.7×10^{17}	1.1×10^{18}	9.4×10^{11}	1.0×10^{23}	<0.01
Assists	0.96	0.04	0.88	1.04	0.32
Steals	1.4	0.1	1.3	1.7	<0.01
Blocks	1.077	0.065	0.958	1.215	0.22
TotalRebounds	1.31	0.05	1.22	1.42	<0.01
Turnovers	0.837	0.041	0.758	0.919	<0.01

Exercise 4

```r
## model here
nba_mod1 <- glm(Win ~ Home+TeamPoints+FieldGoals.+
                  Assists+Steals+Blocks+TotalRebounds+Turnovers,
                data=nba_cho, family="binomial")

modelsummary(nba_mod1,
             fmt = fmt_significant(2),
             shape = term ~ model + statistic,
             statistic = c("std.error","conf.int","p.value"),
             exponentiate = TRUE,
             gof_map=NA)
```

Exercise 5

The OR is extremely high for FieldGoals. because the variable is a proportion, so 1 unit increase is not appropriate. We can multiply the variable by 100 to be a percentage

```r
nba_cho <- nba_cho |>
  mutate(FieldGoalsPerc = FieldGoals.*100)

nba_mod2 <- glm(Win ~ Home+TeamPoints+FieldGoalsPerc+
                  Assists+Steals+Blocks+TotalRebounds+Turnovers,
```

	(1)				
	Est.	S.E.	2.5 %	97.5 %	p
(Intercept)	1.2×10^{-13}	3.8×10^{-13}	1.0×10^{-16}	5.2×10^{-11}	<0.01
HomeHome	2.46	0.83	1.28	4.82	<0.01
TeamPoints	0.998	0.022	0.955	1.043	0.92
FieldGoalsPerc	1.487	0.096	1.317	1.698	<0.01
Assists	0.96	0.04	0.88	1.04	0.32
Steals	1.4	0.1	1.3	1.7	<0.01
Blocks	1.077	0.065	0.958	1.215	0.22
TotalRebounds	1.31	0.05	1.22	1.42	<0.01
Turnovers	0.837	0.041	0.758	0.919	<0.01

```
                 data=nba_cho, family="binomial")

modelsummary(nba_mod2,
             fmt = fmt_significant(2),
             shape = term ~ model + statistic,
             statistic = c("std.error","conf.int","p.value"),
             exponentiate = TRUE,
             gof_map=NA)
```

Exercise 6

The odds of winning are 2.46 times higher for home games compared to away games. The relationship between playing at home and winning is statistically significant (p<0.01, 95% CI: [1.28,4.82])

For each additional field goal percent increase, the odds of winning increase by 1.49, which is statistically significant (p<0.01, 95% CI: [1.32,1.7])

etc for steals, total rebounds, and turnovers

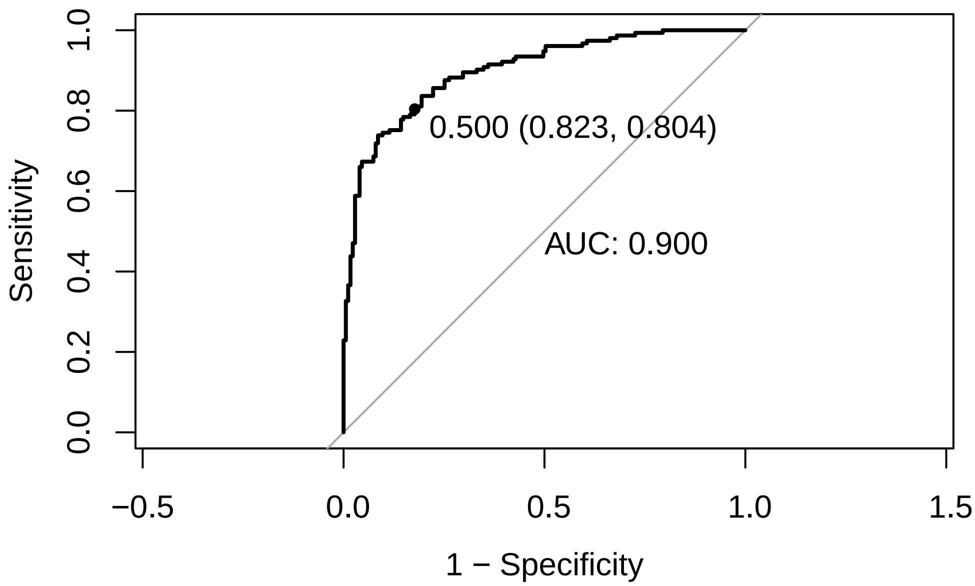
Exercise 7

```
nba_cho_fitted <- factor(ifelse(fitted(nba_mod2)>0.5,1,0),levels=c(0,1),labels=c("Loss","Win"
confusionMatrix(table(nba_cho_fitted,nba_cho$Win), positive = "Win", mode="everything")
```

```
Confusion Matrix and Statistics


nba_cho_fitted Loss Win
          Loss  144  30
          Win    31 123

               Accuracy : 0.814
                 95% CI : (0.7676, 0.8547)
    No Information Rate : 0.5335
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6265

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.8039
            Specificity : 0.8229
         Pos Pred Value : 0.7987
         Neg Pred Value : 0.8276
              Precision : 0.7987
                 Recall : 0.8039
                     F1 : 0.8013
             Prevalence : 0.4665
         Detection Rate : 0.3750
   Detection Prevalence : 0.4695
      Balanced Accuracy : 0.8134

       'Positive' Class : Win
```

```r
roc(nba_cho$Win, fitted(nba_mod2), print.thres=0.5, print.auc=T,
    legacy.axes=T,plot=T)
```

```
Setting levels: control = Loss, case = Win


Setting direction: controls < cases
```




```
Call:
roc.default(response = nba_cho$Win, predictor = fitted(nba_mod2),     plot = T, print.thres =

Data: fitted(nba_mod2) in 175 controls (nba_cho$Win Loss) < 153 cases (nba_cho$Win Win).
Area under the curve: 0.8997
```

Exercise 8

```
nba_mod_reduced <- nba_mod2

nba_cho <- nba_cho |>
  mutate(Opp.FieldGoalsPerc = Opp.FieldGoals.*100)

nba_mod_full <- glm(Win ~ Home+TeamPoints+FieldGoalsPerc+
                  Assists+Steals+Blocks+TotalRebounds+Turnovers+
                    OpponentPoints+Opp.FieldGoalsPerc+Opp.Assists+
                    Opp.Steals+Opp.Blocks+Opp.TotalRebounds+Opp.Turnovers,
                data=nba_cho, family="binomial")
```

```
Warning: glm.fit: algorithm did not converge
```

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

```r
anova(nba_mod_reduced, nba_mod_full, test="Chisq")
```

```
Analysis of Deviance Table

Model 1: Win ~ Home + TeamPoints + FieldGoalsPerc + Assists + Steals + 
    Blocks + TotalRebounds + Turnovers
Model 2: Win ~ Home + TeamPoints + FieldGoalsPerc + Assists + Steals + 
    Blocks + TotalRebounds + Turnovers + OpponentPoints + Opp.FieldGoalsPerc + 
    Opp.Assists + Opp.Steals + Opp.Blocks + Opp.TotalRebounds + 
    Opp.Turnovers
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)    
1       319     259.08                          
2       312       0.00  7   259.08 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Exercise 9

Bonus

A team always wins when TeamPoints>OpponentPoints, so there is perfect separation. Removing OpponentPoints eliminates the error.