IDS 702 HW 4

Eric Ortega Rodriguez

Load data

```
library(tidyverse)
library(modelsummary)
```

```
Warning: package 'modelsummary' was built under R version 4.4.1
```

```
library(caret)
library(pROC)

nba <- read.csv(
  paste0(
    "https://raw.githubusercontent.com/anlane611/datasets/refs/",
    "heads/main/nba_games_stats.csv"
  )
)
```

Exercise 1

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

## create factor variable for Home
nba_cho <- nba_cho %>% mutate(Home = as.factor(Home))

## create new Win variable
nba_cho <- nba_cho %>%
  mutate(Win = ifelse(WINorLOSS == "W", 1, 0),
```

```
         Win = factor(Win, levels = c(0, 1), labels = c("Loss", "Win")))

## format date variable
nba_cho <- nba_cho %>%
  mutate(Date_clean = as.Date(Date, "%Y-%m-%d"))
# Quality check
table(nba_cho$Win, nba_cho$WINorLOSS)
```

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

Exercise 2

```
## code to fill in the table
library(dplyr)

# the descriptive statistics for wins and losses
desc_stats <- nba_cho %>%
  group_by(Win) %>%
  summarize(
    Games = n(),
    Home_games = sum(Home == "Home"),
    TeamPoints_mean = mean(TeamPoints, na.rm = TRUE),
    TeamPoints_sd = sd(TeamPoints, na.rm = TRUE),
    FG_percent_mean = mean(FieldGoals / FieldGoalsAttempted, na.rm = TRUE),
    FG_percent_sd = sd(FieldGoals / FieldGoalsAttempted, na.rm = TRUE),
    Assists_mean = mean(Assists, na.rm = TRUE),
    Assists_sd = sd(Assists, na.rm = TRUE),
    Steals_mean = mean(Steals, na.rm = TRUE),
    Steals_sd = sd(Steals, na.rm = TRUE),
    Blocks_mean = mean(Blocks, na.rm = TRUE),
    Blocks_sd = sd(Blocks, na.rm = TRUE),
    OpponentPoints_mean = mean(OpponentPoints, na.rm = TRUE),
    OpponentPoints_sd = sd(OpponentPoints, na.rm = TRUE),
    TotalRebounds_mean = mean(TotalRebounds, na.rm = TRUE),
    TotalRebounds_sd = sd(TotalRebounds, na.rm = TRUE),
    Turnovers_mean = mean(Turnovers, na.rm = TRUE),
    Turnovers_sd = sd(Turnovers, na.rm = TRUE)
```

```
  )

# Getting results
print(desc_stats)
```

```
# A tibble: 2 x 19
  Win   Games Home_games TeamPoints_mean TeamPoints_sd FG_percent_mean
  <fct> <int>      <int>           <dbl>         <dbl>           <dbl>
1 Loss    175         72            96.9          11.3           0.417
2 Win     153         92           109.           11.4           0.463
# i 13 more variables: FG_percent_sd <dbl>, Assists_mean <dbl>,
#   Assists_sd <dbl>, Steals_mean <dbl>, Steals_sd <dbl>, Blocks_mean <dbl>,
#   Blocks_sd <dbl>, OpponentPoints_mean <dbl>, OpponentPoints_sd <dbl>,
#   TotalRebounds_mean <dbl>, TotalRebounds_sd <dbl>, Turnovers_mean <dbl>,
#   Turnovers_sd <dbl>
```

Variable	Wins (N=153)	Losses (N=175)
Home games - N (%)	92 (60.1%)	72 (41.1%)
Team Points - mean (SD)	109.2 (11.4)	96.9 (11.3)
Field Goal Percentage - mean (SD)	46.3 (5.1)	41.7 (4.9)
Assists - mean (SD)	23.3 (4.5)	20.2 (4.5)
Steals - mean (SD)	7.2 (2.8)	6.4 (2.6)
Blocks - mean (SD)	5.5 (2.7)	4.6 (2.6)
Opponent Points - mean (SD)	97.2 (11.3)	107.5 (10.1)
Total Rebounds - mean (SD)	46.6 (6.6)	42.2 (5.8)
Turnovers - mean (SD)	11.4 (3.6)	11.7 (3.7)

Exercise 3

Looking at the codebook, I can see two pairs of variables that may be problematic if they are both included in a model. They are the following:

1. **TotalRebounds** and **OffRebounds**:
   - This is because within`TotalRebounds`, there is both offensive and defensive rebounds.`OffRebounds` represents offensive rebounds. Including both variables in a model could also introduce multicollinearity due to `OffRebounds` being a part of `TotalRebounds`.
2. **FieldGoals** and **FieldGoalsAttempted**:

- `FieldGoals` represents the number of field goals made, and `FieldGoalsAttempted` represents the number attempted. This is also field goal percentage which can also tie into Including both into multicollinearity. More specifically, this is due to one largely depends on the other.

Having these pairs of variables can create redundancy and multicollinearity issues.

Exercise 4

```
## 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)
```

	(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 5

The `FieldGoals` number is very large, which is something that I immediately noticed. This is possibly attributed to our model using the raw count of field goals made under the `FieldGoals.` variable. This did not correctly account for the number of attempts, which is not a good measure of shooting effectiveness. To adjust for this, we created a `FieldGoalPercentage` variable, representing shooting efficiency, by dividing `FieldGoals` by `FieldGoalsAttempted`. This new variable provides a more accurate representation of shooting performance and will be more helpful within our model.

```
# scaled percentage variable
nba_cho$FieldGoals_percent_scaled <- nba_cho$FieldGoals. * 100

# Fitting the logistic regression model
model_adjusted <- glm(
  Win ~ Home + TeamPoints + FieldGoals_percent_scaled
  + Assists + Steals + Blocks + TotalRebounds + Turnovers,
  data = nba_cho,
  family = binomial
)

# summary of the model with odds ratios and additional statistics
modelsummary(
  model_adjusted,
  fmt = fmt_significant(2),
  shape = term ~ model + statistic,
  statistic = c("std.error", "conf.int", "p.value"),
  exponentiate = TRUE,
  gof_map = NA
)
```

	(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_percent_scaled	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

Exercise 6

For the Charlotte Hornets, the statistically significant factors affecting the odds of winning an NBA game can be interpreted as follows:

1. **Playing at Home**
   - The odds ratio for playing at home is approximately **2.46**. This means that the Hornets are about **2.46 times more likely to win** when playing at home compared to away games, highlighting the importance of having a home crowd and a home-court advantage (given p-value < 0.05).

2. **Field Goals Percentage**
   - The odds ratio for field goals percentage is approximately **1.487**. This shows that for each 1% increase in shooting accuracy, the Hornets' odds of winning increase by about **48.7%**. This statistically significant effect (given p-value < 0.05) . This highlights the importance of shooting accuracy

3. **Steals**
   - The odds ratio for steals is around **1.4**. This shows that each additional steal improves the Hornets' odds of winning by **40%**. This positive effect (given p-value < 0.05) shows the value of strong defensive plays. MOre specifcally, taking possesion of the ball.

4. **Total Rebounds**

- The odds ratio for total rebounds is about **1.31**. This means that every additional rebound increases the odds of winning by **31%**. This statistically significant effect (given p-value < 0.05) emphasizes the importance of controlling the boards to increase the chance of winning

5. **Turnovers**
   - The odds ratio for turnovers is approximately **0.837**. This highligths that each additional turnover reduces the Hornets' odds of winning by **16.3%**. This significant negative impact (given p-value < 0.05) highlights the need to minimize turnovers to improve their chances of getting a win

---

In summary, the Hornets are more likely to win when the following factors occur: When they play at home, shoot hoops accurately, and strive to get steals and rebounds. They should also avoid turnovers.

Exercise 7

```
# Predict win probabilities
predicted_probabilities <- predict(model_adjusted, nba_cho, type = "response")

# Classify games as "Win" or "Loss" using a 0.5 cutoff
predicted_classes <- ifelse(predicted_probabilities > 0.5, "Win", "Loss")
predicted_classes <- factor(predicted_classes, levels = c("Loss", "Win"))

# Create the confusion matrix
confusion_matrix <- confusionMatrix(
  table(predicted_classes, nba_cho$Win),
  positive = 'Win',
  mode = 'everything'
)

# Display the confusion matrix
confusion_matrix
```

```
Confusion Matrix and Statistics


predicted_classes 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
```

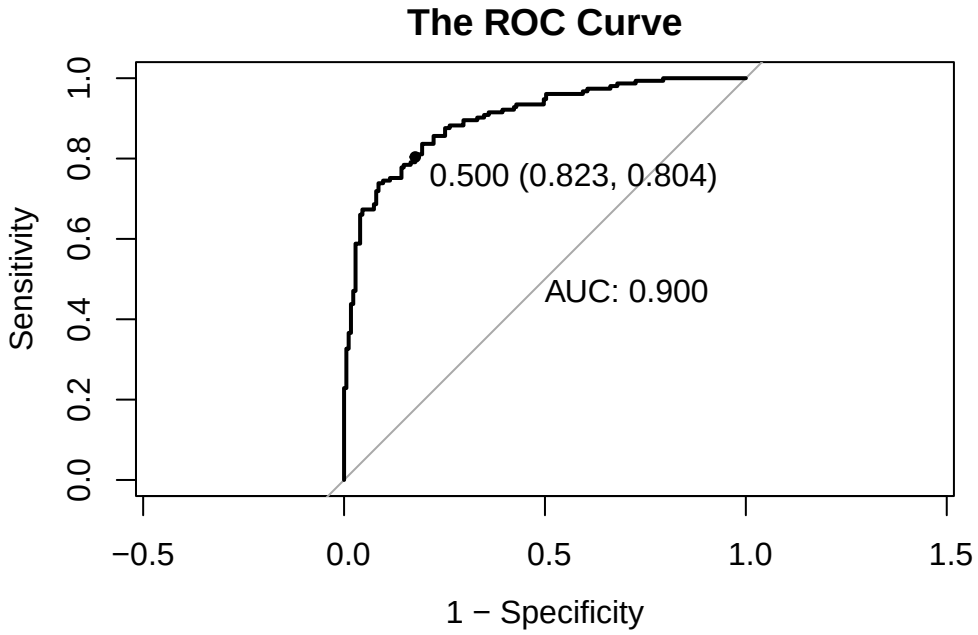
```
accuracy <- confusion_matrix$overall['Accuracy']
cat("Model Accuracy:", accuracy, "\n")
```

```
Model Accuracy: 0.8140244
```

```
roc_curve <- roc(nba_cho$Win, predicted_probabilities,
print.thres=0.5,
print.auc=TRUE,
legacy.axes=TRUE,
main = "The ROC Curve",
plot=TRUE)
```

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

```
Setting direction: controls < cases
```



As seen above, the model has an accuracy of **0.814**. This indicates that it is relatively good at accurately predicting the outcome —whether a win or a loss— in approximately 81% of cases.

Regarding the AUC,is is at approximately **0.9**. This value is a good indicator of the model's effectiveness in distinguishing wins from losses. Given the high AUC score, it reflects the model's good performance in differentiating between the two outcomes.This can also be seen in the AUC graph above.

Exercise 8

```r
# Rescaling the Opp.FieldGoals column to create a percentage if needed
nba_cho$Opp.FieldGoals_percent_scaled <-
  nba_cho$Opp.FieldGoals / nba_cho$Opp.FieldGoalsAttempted * 100

# Fit Model 1, this will include team stats by itself
model_adjusted <- glm(
  Win ~ Home + TeamPoints + FieldGoals_percent_scaled + Assists + Steals + Blocks +
    TotalRebounds + Turnovers,
```

```
  data = nba_cho,
  family = binomial
)

# Fit Model 2, also includes opponent's stats
model_opponent <- glm(
  Win ~ Home + TeamPoints + FieldGoals_percent_scaled + Assists + Steals + Blocks +
    TotalRebounds + Turnovers + OpponentPoints + Opp.FieldGoals_percent_scaled + 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
```

```
# Deviance for each model
deviance_model_team <- deviance(model_adjusted)
deviance_model_opponent <- deviance(model_opponent)

# Degrees of freedom for each model
df_model_team <- df.residual(model_adjusted)
df_model_opponent <- df.residual(model_opponent)

# Change in deviance and degrees of freedom difference
delta_deviance <- deviance_model_team - deviance_model_opponent
delta_df <- df_model_team - df_model_opponent

# Chi-squared test
chi_squared_statistic <- delta_deviance
p_value <- 1 - pchisq(chi_squared_statistic, delta_df)

# result likelihood ratio test
cat("The change in deviance is:", delta_deviance, "\n")
```

```
The change in deviance is: 259.0826
```

```
cat("The degrees of Freedom Difference is:", delta_df, "\n")
```

```
The degrees of Freedom Difference is: 7
```

```
cat("The Chi-Squared Statistic is:", chi_squared_statistic, "\n")
```

```
The Chi-Squared Statistic is: 259.0826
```

```
cat("The p-value is:", p_value, "\n")
```

```
The p-value is: 0
```

```
# Alternative way to compare the models using anova function
anova(model_adjusted, model_opponent, test = "Chisq")
```

```
Analysis of Deviance Table

Model 1: Win ~ Home + TeamPoints + FieldGoals_percent_scaled + Assists +
    Steals + Blocks + TotalRebounds + Turnovers
Model 2: Win ~ Home + TeamPoints + FieldGoals_percent_scaled + Assists +
    Steals + Blocks + TotalRebounds + Turnovers + OpponentPoints +
    Opp.FieldGoals_percent_scaled + 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
```

Based on the results above, including the opponent's statistics significantly improves the model's performance compared to using only the team's statistics. By incorporating both team and opponent factors, the model can better capture the dynamics of each game and understanding of what influences the likelihood of a win. The added complexity allows the model to make more accurate predictions. Ultimately, this adjustment enhances the model's predictive accuracy and reliability when estimating the team's chances of winning.

Exercise 9

Based on the analysis, if Charles Lee (coach of the Charlotte Hornets) approached me to tell him about the factors that are associated with wins. I would include the following:

- **Home Court Advantage**: Playing at home significantly increases the odds of winning. This could be due to being in home crowd and there is no travel fatigue.
- **Field Goal Percentage**: Higher field goal percentages are strongly correlated with winning. Essentially showing that better accuracy, improves the outcome of the game.
- **Steals & Rebounds**: More steals and higher total rebounds positively impact the odds of winning. Steals interrupt the opponent's offense and rebounds allow for more chances of scoring.

In addition to this, I would also tell Coach Lee that there are some factors to watch out for since they can reduce the likelihood of getting a win. Some of these factors include the following:

- **Ball Turnovers**: Turnovers reduce the odds of winning. Reducing turnovers helps the team keep possession of the ball and allows the team to have a higher chance of scoring.
- **Given Opponent Statistics**: As seen above, including the opponent statistics slightly improves the model, but the effect is not as strong as anticipated. It shows that it is crucial to counter the opponents strengths. It would be crucial for Coach Lee to consider the opponent's scoring and rebounding abilities and for him to focus on the Hornets' shooting accuracy, rebounding, defensive plays, and minimizing turnovers—may since this can ultimately lead them to victory.

In conclusion, I would tell Coach Lee to prioritize shooting efficiency, rebounding, defensive pressure (preferably steals), and minimizing turnovers. These areas, which are within the team's control, have the greatest impact on the Hornets' likelihood of winning.

Bonus Question

The warning message I got was the following:

```
Warning: glm.fit: algorithm did not converge
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

The warning message suggests that the model is struggling to converge, meaning it is unable to reliably estimate the coefficients. This often happens when fitted probabilities are pushed to extreme values, close to 0 or 1.

One common reason is **strong separation** in the data, where certain predictors nearly perfectly distinguish between wins and losses. This causes the model to assign extreme values to coefficients, resulting in probabilities that approach 0 or 1. Another potential cause is **multicollinearity**. This is where predictors are highly correlated. For instance, when we include both team and opponent statistics can introduce redundancy, as these variables may reflect the same game conditions.

In this case of this model, when I included both team and opponent statistics, it likely caused redundancy, as certain metrics (like team and opponent points) tend to increase together. This redundancy complicates the model by attempting to account for both sides' statistics, which reduces stability. The warning suggests that removing some redundant predictors would result in a more stable and interpretable model.

To address this, we could remove correlated variables, such as `OpponentPoints`. By reducing multicollinearity, the model may converge more easily and produce more reliable estimates, focusing on the team's own performance metrics.