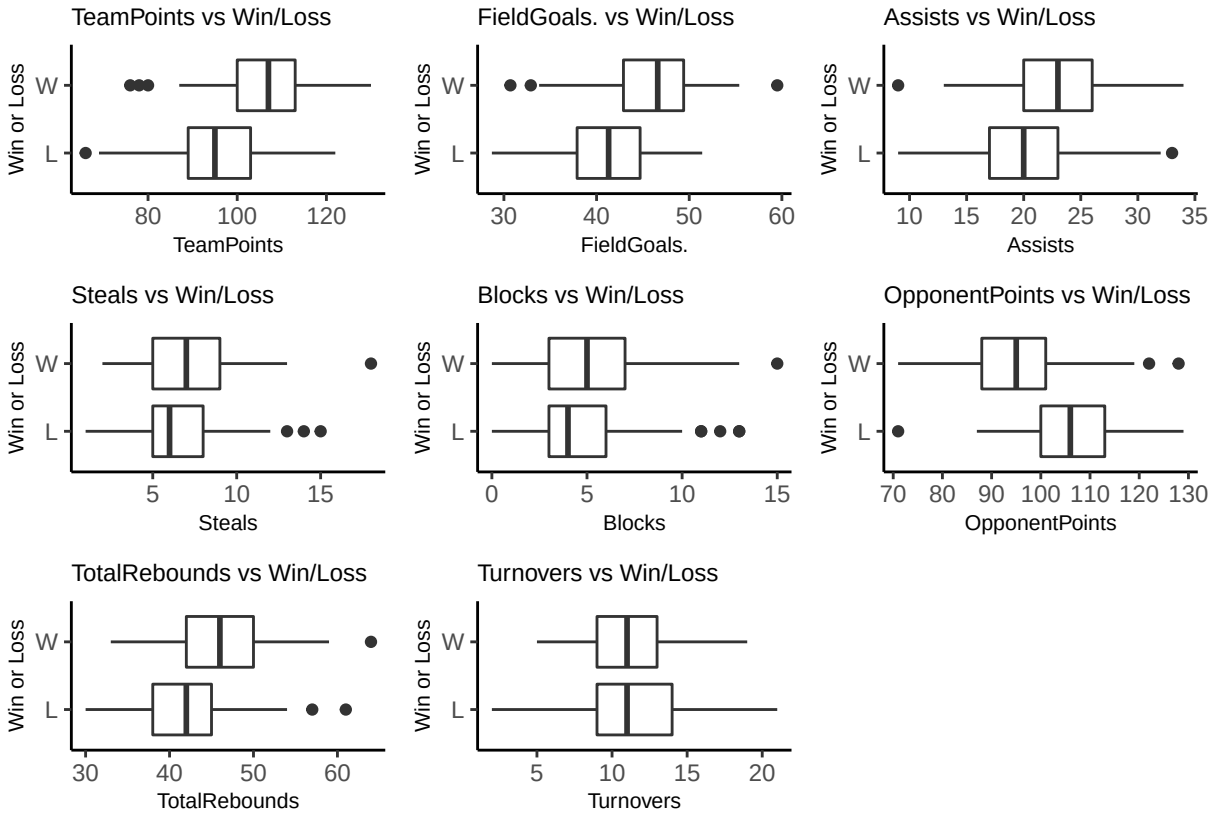
Data Analysis Assignment 3

Q1 EDA



After using boxplots for continuous variables and table for categorical variable, we can tell that the higher the number of total points scored in the game(TeamPoints), the higher the number of field goals made in the game(FieldGoals.), the higher the total number of assists(Assists), the higher the total number of steals (balls stolen from the opposing team while the opposing team has possession) in the game(Steals), the higher the total number of blocks (direct prevention of a made field goal after the ball has been shot by an opposing player) in the game (Blocks) and the higher total number of rebounds grabbed in the game(TotalRebounds), the more likely the tean will win the game, since the median is higher. For the Turnovers(Total number of times the ball was lost back to the opposing team while the team had possession), there seems to have no obvious difference between win and loss. While the lower the number of total points scored by the opposing

	Away	Home
L	77	52
W	46	71

	Away	Home
L	0.6260163	0.4227642
W	0.3739837	0.5772358

team in the game(OpponentPoints) and the lower the total number of times the ball was lost back to the opposing team, the more likely the team will lose the game.

The probability of wining changes for the location of the game. When the game is a home game, the team is more likely to win, while when it's a away game, the team is more likely to lose.

Q2

1. We shouldn't include both FieldGoals. and FieldGoals as predictors in the logistic model, since FieldGoadls. contains the information of both FieldGoals and FieldGoalsAttempted.
2. We shouldn't include both FieldGoals. and X3PointShots as predictors in the logistic model, since FieldGoadls. includes 3 point shots, it contains the whole information of X3PointShots.

Q3

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-29.80	3.94	-7.56	0.00
HomeHome	1.11	0.40	2.75	0.01
TeamPoints	-0.01	0.03	-0.34	0.74
FieldGoals.	0.44	0.08	5.78	0.00
Assists	-0.11	0.05	-2.23	0.03
Steals	0.39	0.08	4.74	0.00
Blocks	0.04	0.07	0.60	0.55
TotalRebounds	0.28	0.04	6.22	0.00
Turnovers	-0.17	0.06	-3.03	0.00

Fitted Model

log(p_win/1-p_win) =β0 + β1Home_Home + β2TeamPoints +β3 FieldGoals+β4Assists+ β5Steals + β6Blocks + β7TotalRecounds + β8Turnovers + ϵ

log(p_win/1-p_win) = -29.80 + 1.11Home_Home - 0.01TeamPoints + 0.44 FieldGoals. - 0.11Assists + 0.39Steals + 0.04Blocks + 0.28TotalRecounds - 0.17Turnovers + ϵ

β_1: The odds of wining a NBA game when the game is at home are 3.03 times higher than the game is away from home.

β_2: As we increase TeamPoints by 1 unit, we increase the log-odds of wining an NBA game by -0.01 (increase the odds for wining an NBA game by a multiplicative effect of 0.99/ the odds of winning an NBA game increase 0.99 times). — **not significant**

β_3: As we increase FieldGoals. by 1 percent, we increase the log-odds of wining an NBA game by 0.44 (increase the odds for wining an NBA game by a multiplicative effect of 1.55/ the odds of winning an NBA game increase 1.55 times).

β_4: As we increase Assists by 1 unit, we increase the log-odds of wining an NBA game by -0.11 (increase the odds for wining an NBA game by a multiplicative effect of 0.90/ the odds of winning an NBA game increase 0.90 times).

β_5: As we increase Steals by 1 unit, we increase the log-odds of wining an NBA game by 0.39 (increase the odds for wining an NBA game by a multiplicative effect of 1.48/ the odds of winning an NBA game increase 1.48 times).

β_6: As we increase Blocks by 1 unit, we increase the log-odds of wining an NBA game by 0.04 (increase the odds for wining an NBA game by a multiplicative effect of 1.04/ the odds of winning an NBA game increase 1.04 times). — **not significant**

β_7: As we increase TotalRebounds by 1 unit, we increase the log-odds of wining an NBA game by 0.28 (increase the odds for wining an NBA game by a multiplicative effect of 1.32/ the odds of winning an NBA game increase 1.32 times).

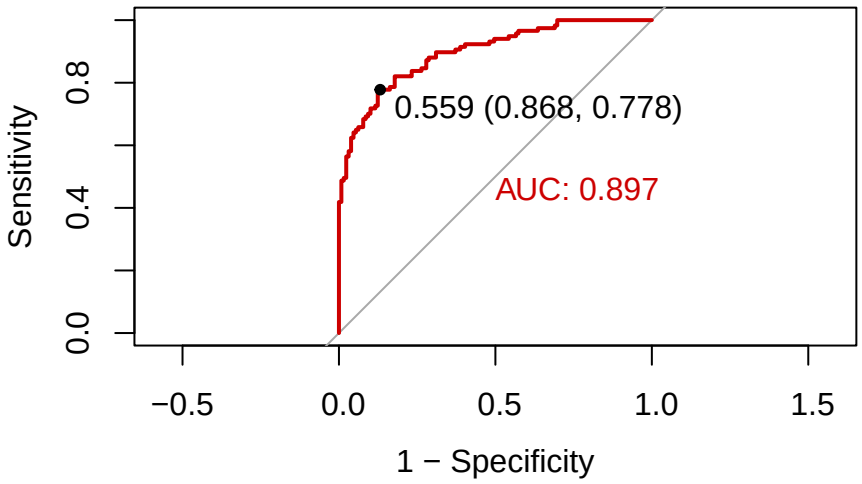
β_8: As we increase TurnOverss by 1 unit, we increase the log-odds of wining an NBA game by -0.17 (increase the odds for wining an NBA game by a multiplicative effect of 0.84/ the odds of winning an NBA game increase 0.84 times).

Q4

	x
HomeHome	1.27
TeamPoints	2.03
FieldGoals.	3.44
Assists	1.53
Steals	1.53
Blocks	1.16
TotalRebounds	2.27
Turnovers	1.29

Since all VIFs are between 1 and 5(<10), which means they are moderately correlated, we don't need to worry about multicollinearity.

Q5



	0	1
0	106	23
1	23	94

	x
Accuracy	0.8130081

	x
Sensitivity	0.8034188
Specificity	0.8217054

The accuracy of this model is **0.813**.

AUC:0.897

Q6

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	9.19	11.61	0.79	0.43
HomeHome	1.53	0.68	2.25	0.02
TeamPoints	0.14	0.06	2.26	0.02
FieldGoals.	0.40	0.16	2.56	0.01
Assists	-0.01	0.08	-0.12	0.91
Steals	0.35	0.18	1.96	0.05
Blocks	-0.09	0.10	-0.91	0.36
TotalRebounds	0.17	0.09	1.83	0.07
Turnovers	-0.55	0.13	-4.27	0.00
Opp.FieldGoals.	-0.86	0.17	-5.08	0.00
Opp.TotalRebounds	-0.36	0.09	-3.94	0.00
Opp.TotalFouls	0.10	0.10	0.96	0.34
Opp.Turnovers	0.61	0.16	3.73	0.00

log(p_win/1-p_win) = 9.19 + 1.53Home_Home + 0.14TeamPoints + 0.40 FieldGoals - 0.01Assists + 0.35Steals - 0.09Blocks + 0.17TotalRebounds - 0.55Turnovers - 0.86Opp.FieldGoals. - 0.36Opp.TotalRebounds + 0.10Opp.TotalFouls + 0.61Opp.Turnovers $+ \epsilon$

Significant coefficients includes β_1, β_2, β_3, β_8, β_9, β_{10}, and β_{12}

β_1: The odds of wining a NBA game when the game is at home are 4.62 times higher than the game is away from home.

β_2: As we increase TeamPoints by 1 unit, we increase the log-odds of wining an NBA game by 0.14 (increase the odds for wining an NBA game by a multiplicative effect of 1.15/ the odds of winning an NBA game increase 1.15 times).

β_3: As we increase FieldGoals. by 1 percent, we increase the log-odds of wining an NBA game by 0.40 (increase the odds for wining an NBA game by a multiplicative effect of 1.49/ the odds of winning an NBA game increase 1.49 times).

β_8: As we increase TurnOvers by 1 unit, we increase the log-odds of wining an NBA game by -0.55 (increase the odds for wining an NBA game by a multiplicative effect of 0.58/ the odds of winning an NBA game increase 0.58 times).

β_9: As we increase Opp.FieldGoals. by 1 percent, we increase the log-odds of wining an NBA game by -0.86 (increase the odds for wining an NBA game by a multiplicative effect of 0.42/ the odds of winning an NBA game increase 0.42 times).

β_{10}: As we increase Opp.TotalRebounds by 1 unit, we increase the log-odds of wining an NBA game by -0.36 (increase the odds for wining an NBA game by a multiplicative effect of 0.70/ the odds of winning an NBA game increase 0.70 times).

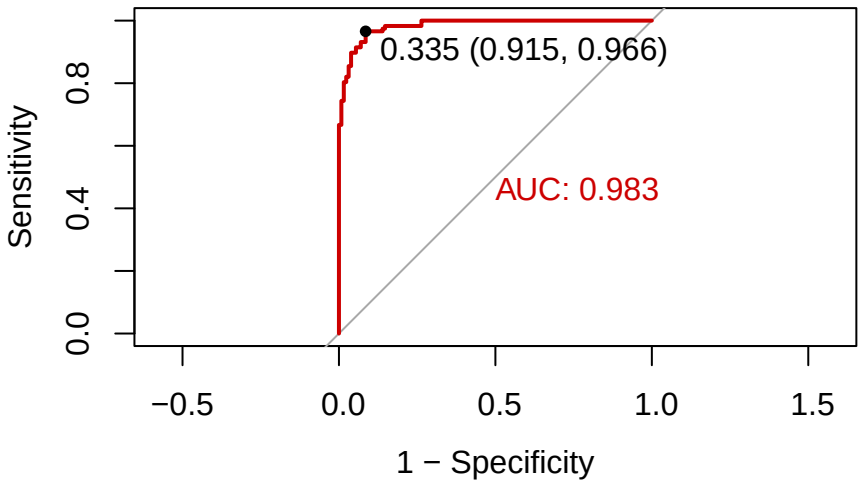
β_{12}: As we increase Opp.Turnovers by 1 unit, we increase the log-odds of wining an NBA game by 0.61 (increase the odds for wining an NBA game by a multiplicative effect of 1.84/ the odds of winning an NBA game increase 1.84 times).

Q7

	0	1
0	120	9
1	9	108

	x
Accuracy	0.9268293

	x
Sensitivity	0.9230769
Specificity	0.9302326



The accuracy of this model is **0.927**.

AUC for model that includes Opp.FieldGoals., Opp.TotalRebounds, Opp.TotalFouls, and Opp.Turnovers as predictors is higher(0.983 > 0.897), the accuracy is also higher(0.927>0.813). Therefore, this new model is better.

Q8

	0	1
0	37	2
1	9	34

	x
Accuracy	0.8658537

	x
Sensitivity	0.9444444
Specificity	0.8043478

The accuracy of this model is **0.866**. The sensitivity of this model is **0.944** and the specificity is **0.804**(1-specificity is 0.196). Therefore, the model do well in predicting data for the 2017/2018 season.

Q9

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
233	82.32	NA	NA	NA
231	80.78	2	1.54	0.46

	0	1		x		x
0	39	2	Accuracy	0.8902439	Sensitivity	0.9444444
1	7	34			Specificity	0.8478261

- **P value = 0.46(>0.05)**, which means that there is no statistically significance between the model that including Opp.Assists and Opp.Blocks and the original model. However, When using this model to predict data for the 2017/2018 season, the accuracy is **0.890**(>0.866), the sensitivity is **0.944**, and the specificity is **0.848**(>0.804). Since *accuracy increases and 1-specificity decreases*, including Opp.Assists and Opp.Blocks in the model at the same time improves the model.

Resid. Df	Resid. Dev	Df	Deviance	Pr(>Chi)
233	82.32	NA	NA	NA
232	79.60	1	2.72	0.1

	0	1		x		x
0	39	2	Accuracy	0.8902439	Sensitivity	0.9444444
1	7	34			Specificity	0.8478261

- **FreeThrows.** may improve our model, because the predictor FieldGoals. includes both number of field goals made and attempted in the game (also includes 3 point shots but not free throws). Though the p=value of the model including FreeThrows. is 0.10, which is not statistically significant enough, when using this model to predict data for the 2017/2018 season, the accuracy is **0.890**(>0.866), the sensitivity is **0.944**, and the specificity is **0.848**(>0.804). Since *accuracy increases and 1-specificity decreases*, including Opp.Assists and Opp.Blocks in the model at the same time improves the model(the model that I selected in Q7).

Q10

- Although during exploratory data analysis, for the Turnovers(Total number of times the ball was lost back to the opposing team while the team had possession), there seems to have no obvious difference between win and loss. In each model, the coefficient of Turnovers is statistically significant.
- In order to have a higher chance of wining, team should try to get higher FieldGoals/FieldGoalsAttempted and lower the total number of times the ball was lost back to the opposing team while the team had possession(Turnovers). In the meantime, if Opp.FieldGoals/Opp.FieldGoalsAttempted is lower, the number of offensive rebounds grabbed by the opposing team in the game is lower, and the higher the total number of times the ball was won back from the opposing team while the opposing team had possession, the higher the possibility the team will win.

Appendix: All code for this report

```
knitr::opts_chunk$set(echo = TRUE)
library(ISLR2)
library(knitr)
library(ggplot2)
library(kableExtra)
library(lattice)
library(dplyr)
library(stargazer)
library(gt)
library(janitor)
library(flextable)
library(magrittr)
library(Hmisc)
library(caret)
library(patchwork)
library(leaps)
library(rms)
library(arm)
library(pROC)
library(e1071)
library(caret)
require(gridExtra)
nba <- read.csv("nba_games_stats.csv")
# Set factor variables
nba$Home <- factor(nba$Home)
nba$Team <- factor(nba$Team)
nba$WINorLOSS <- factor(nba$WINorLOSS)
# Convert date to the right format
nba$Date <- as.Date(nba$Date, "%Y-%m-%d")
# Also create a binary variable from WINorLOSS.
# This is not always necessary but can be useful
#particularly for R functions that prefer numeric binary variables
#to the original factor variables
nba$Win <- rep(0,nrow(nba))
nba$Win[nba$WINorLOSS=="W"] <- 1
# Charlotte hornets subset
nba_reduced <- nba[nba$Team == "CHO", ]
#100*FieldGoals., Opp.FieldGoals.
nba_reduced$FieldGoals. <- (nba_reduced$FieldGoals.)*100
nba_reduced$Opp.FieldGoals.<-(nba_reduced$Opp.FieldGoals.)*100
# Set aside the 2017/2018 season as your test data
nba_reduced_train <- nba_reduced[nba_reduced$Date < "2017-10-01",]
nba_reduced_test <- nba_reduced[nba_reduced$Date >= "2017-10-01",]
#boxplot for numerical variables(TeamPoints, FieldGoals.,Assists, Steals, Blocks, OpponentPoints, Total

p1 <- ggplot(nba_reduced_train,aes(x=TeamPoints, y=WINorLOSS, fill=TeamPoints)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Reds") +
  labs(title="TeamPoints vs Win/Loss",
       x="TeamPoints",y="Win or Loss") +
  theme_classic() + theme(legend.position="none",plot.title = element_text(size = 9), axis.title = eleme
```

```
p2 <- ggplot(nba_reduced_train,aes(x=FieldGoals., y=WINorLOSS, fill=FieldGoals.)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Reds") +
  labs(title="FieldGoals. vs Win/Loss",
       x="FieldGoals.",y="Win or Loss") +
  theme_classic() + theme(legend.position="none",plot.title = element_text(size = 9), axis.title = eleme

p3 <- ggplot(nba_reduced_train,aes(x=Assists, y=WINorLOSS, fill=Assists)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Reds") +
  labs(title="Assists vs Win/Loss",
       x="Assists",y="Win or Loss") +
  theme_classic() + theme(legend.position="none",plot.title = element_text(size = 9), axis.title = eleme


p4 <- ggplot(nba_reduced_train,aes(x=Steals, y=WINorLOSS, fill=Steals)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Reds") +
  labs(title="Steals vs Win/Loss",
       x="Steals",y="Win or Loss") +
  theme_classic() + theme(legend.position="none",plot.title = element_text(size = 9), axis.title = eleme

p5 <- ggplot(nba_reduced_train,aes(x=Blocks, y=WINorLOSS, fill=Blocks)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Reds") +
  labs(title="Blocks vs Win/Loss",
       x="Blocks",y="Win or Loss") +
  theme_classic() + theme(legend.position="none",plot.title = element_text(size = 9), axis.title = eleme

p6 <- ggplot(nba_reduced_train,aes(x=OpponentPoints, y=WINorLOSS, fill=OpponentPoints)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Blues") +
  labs(title="OpponentPoints vs Win/Loss",
       x="OpponentPoints",y="Win or Loss") +
  theme_classic() + theme(legend.position="none",plot.title = element_text(size = 9), axis.title = eleme

p7 <- ggplot(nba_reduced_train,aes(x=TotalRebounds, y=WINorLOSS, fill=TotalRebounds)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Reds") +
  labs(title="TotalRebounds vs Win/Loss",
       x="TotalRebounds",y="Win or Loss") +
  theme_classic() + theme(legend.position="none",plot.title = element_text(size = 9), axis.title = eleme

p8 <- ggplot(nba_reduced_train,aes(x=Turnovers, y=WINorLOSS, fill=Turnovers)) +
  geom_boxplot() +
  scale_fill_brewer(palette="Blues") +
  labs(title="Turnovers vs Win/Loss",
       x="Turnovers",y="Win or Loss") +
  theme_classic() + theme(legend.position="none",plot.title = element_text(size = 9), axis.title = eleme

p1+p2+p3+p4+p5+p6+p7+p8+plot_layout(ncol=3)
#tables for factor variable Home
t1 <- table(nba_reduced_train[,c("WINorLOSS","Home")])
```

```
t2 <- apply(table(nba_reduced_train[,c("WINorLOSS","Home")])/sum(table(nba_reduced_train[,c("WINorLOSS"
      2,function(x) x/sum(x))
knitr::kable(list(t1, t2))
nba_glm <- glm(Win ~ Home + TeamPoints + FieldGoals. + Assists + Steals + Blocks + TotalRebounds + Turn
#summary(nba_glm)
knitr::kable(summary(nba_glm)$coefficients,digits = 2)%>% kable_styling(position="center",full_width =
#vif(nba_glm)
knitr::kable(vif(nba_glm),digits = 2)%>% kable_styling(full_width = FALSE,latex_options = c("hold_posit
Conf_mat <- confusionMatrix(as.factor(ifelse(fitted(nba_glm) >= 0.5, "1","0")),
                            as.factor(nba_reduced_train$Win),positive = "1")
t1 <- Conf_mat$table
t2 <- Conf_mat$overall["Accuracy"];
t3 <- Conf_mat$byClass[c("Sensitivity","Specificity")]

knitr::kable(list(t1, t2,t3))
roc(nba_reduced_train$Win,fitted(nba_glm),plot=T,print.thres="best",legacy.axes=T,
    print.auc =T,col="red3")
nba_glm_new <- glm(Win ~ Home + TeamPoints + FieldGoals. + Assists + Steals + Blocks + TotalRebounds +
#summary(nba_glm_new)
knitr::kable(summary(nba_glm_new)$coefficients,digits = 2)%>% kable_styling(position="center",full_widt
Conf_mat <- confusionMatrix(as.factor(ifelse(fitted(nba_glm_new) >= 0.5, "1","0")),
                            as.factor(nba_reduced_train$Win),positive = "1")
t1.1 <- Conf_mat$table
t2.1 <- Conf_mat$overall["Accuracy"];
t3.1 <- Conf_mat$byClass[c("Sensitivity","Specificity")]

knitr::kable(list(t1.1, t2.1,t3.1))%>% kable_styling(position="center",full_width = FALSE,latex_options
roc(nba_reduced_train$Win,fitted(nba_glm_new),plot=T,print.thres="best",legacy.axes=T,
    print.auc =T,col="red3")
pred <- predict(nba_glm_new, nba_reduced_test, type = "response")
pred_new <- as.factor(ifelse(pred >=0.5,'1','0'))

Conf_mat <- confusionMatrix(pred_new,
                            as.factor(nba_reduced_test$Win),positive = "1")
t1<-Conf_mat$table
t2<-Conf_mat$overall["Accuracy"];
t3<-Conf_mat$byClass[c("Sensitivity","Specificity")]

knitr::kable(list(t1, t2,t3))%>% kable_styling(position="center",full_width = FALSE,latex_options = c("
nba_glm_new_1 <- glm(Win ~ Home + TeamPoints + FieldGoals. + Assists + Steals + Blocks + TotalRebounds
#summary(nba_glm_new_1)
#knitr::kable(summary(nba_glm_new_1)$coefficients,digits = 2)%>% kable_styling(position="center",full_w
#anova(nba_glm_new,nba_glm_new_1, test = 'Chisq')
knitr::kable(anova(nba_glm_new,nba_glm_new_1,test = 'Chisq'),digits = 2)%>% kable_styling(position="cen
pred1 <- predict(nba_glm_new_1, nba_reduced_test, type = "response")
pred_new_1 <- as.factor(ifelse(pred1 >=0.5,'1','0'))

Conf_mat <- confusionMatrix(pred_new_1,
                            as.factor(nba_reduced_test$Win),positive = "1")
t1<-Conf_mat$table
t2<-Conf_mat$overall["Accuracy"];
t3<-Conf_mat$byClass[c("Sensitivity","Specificity")]
```

```
knitr::kable(list(t1, t2,t3))%>% kable_styling(position="center",full_width = FALSE,latex_options = c("
nba_glm_new_2 <- glm(Win ~ Home + TeamPoints + FieldGoals. + Assists + Steals + Blocks + TotalRebounds +
#summary(nba_glm_new_2)
#vif(nba_glm_new_2)
#knitr::kable(summary(nba_glm_new_2)$coefficients,digits = 2)%>% kable_styling(position="center",full_w
#anova(nba_glm_new,nba_glm_new_2, test = 'Chisq')
knitr::kable(anova(nba_glm_new,nba_glm_new_2,test = 'Chisq'),digits = 2)%>% kable_styling(position="cen
pred2 <- predict(nba_glm_new_2, nba_reduced_test, type = "response")
pred_new_2 <- as.factor(ifelse(pred1 >=0.5,'1','0'))

Conf_mat <- confusionMatrix(pred_new_2,
                            as.factor(nba_reduced_test$Win),positive = "1")
t1<-Conf_mat$table
t2<-Conf_mat$overall["Accuracy"];
t3<-Conf_mat$byClass[c("Sensitivity","Specificity")]

knitr::kable(list(t1, t2,t3))%>% kable_styling(position="center",full_width = FALSE,latex_options = c("
```