STAA 551 Case Study

Matt Poiesz, Nathan Mitchell, Danielle Contreras

Every year, women from around the world gather in and out of the United States to compete in golf tournaments hosted by the Ladies Professional Golf Association (LPGA). At the end of tournament season, the 60 best golfers are invited to the CME Group Tour Championship which, recently, has led to two million dollars in prize money. Clearly, it pays to be the best. Thus, our goal was to predict what characteristics a golfer must possess in order to be the best and, in turn, earn the most money.

For this study, we utilized the average amount of prize money earned per round, where a "round" represents a full course, typically 18 holes, as a conduit for determining the overall skill level of a player. After all, if a golfer is golfing consistently well, she will, on average, earn more money per round she plays. We have taken the amount of prize money earned per round ("przrnd") to be our response variable.

Though there were many different possible predictor variables to choose from, we ended up with three in our final model. To predict the amount of prize money earned, we used (1) the number of rounds played by the golfer ("rounds"), which is a quantitative variable, (2) the percentage of greens she hit in regulation ("pctgrn"), another quantitative variable, and (3) the average number of putts per round ("aveputt"), which is also a quantitative variable.

Before any analysis could begin, we had to go through the process of cleaning up the provided dataset. In this case, this cleanup was fairly straightforward–all we had to do was remove the "Golfer" column which stored the name of the golfer. Table 1 provides summary information about each remaining predictor and the response variable.

Figure 1 was used to investigate pairwise relationships between each of the variables. Of particular interest is the relatively high correlation values between our three final predictors and the response. Sand save percentage ("pctsndsv") and percentage of fairways hit ("pctfrwy") have very low correlation with the response. Average sand shots per round ("avesand") and average drive distance ("avedist") have moderate correlation with prize money, although both are noticeably lower than the final predictors.

Figure 1 also displays evidence that the relationship between the outcome variable of interest and many of the predictor variables appears to be nonlinear in nature. Given our goal was to fit a linear regression model to these data, we attempted to coerce this relationship to be linear. To do this, we performed a log transformation on our response variable.

Table 1: Summary Statistics

variable	Mean	SD	Median	Min	Max
rounds	67.28	18.80	65.50	28.00	102.00
avedist	246.89	8.64	245.90	224.80	268.50
pctfrwy	67.68	5.29	68.35	50.30	78.80
pctgrn	62.96	3.48	63.00	53.70	70.40
aveputt	25.62	2.21	25.43	21.22	30.98
avesand	0.85	0.24	0.83	0.37	1.71
pctsndsv	37.89	8.19	37.15	18.90	60.00
przrnd	3900.82	3879.98	2064.50	241.00	18991.00

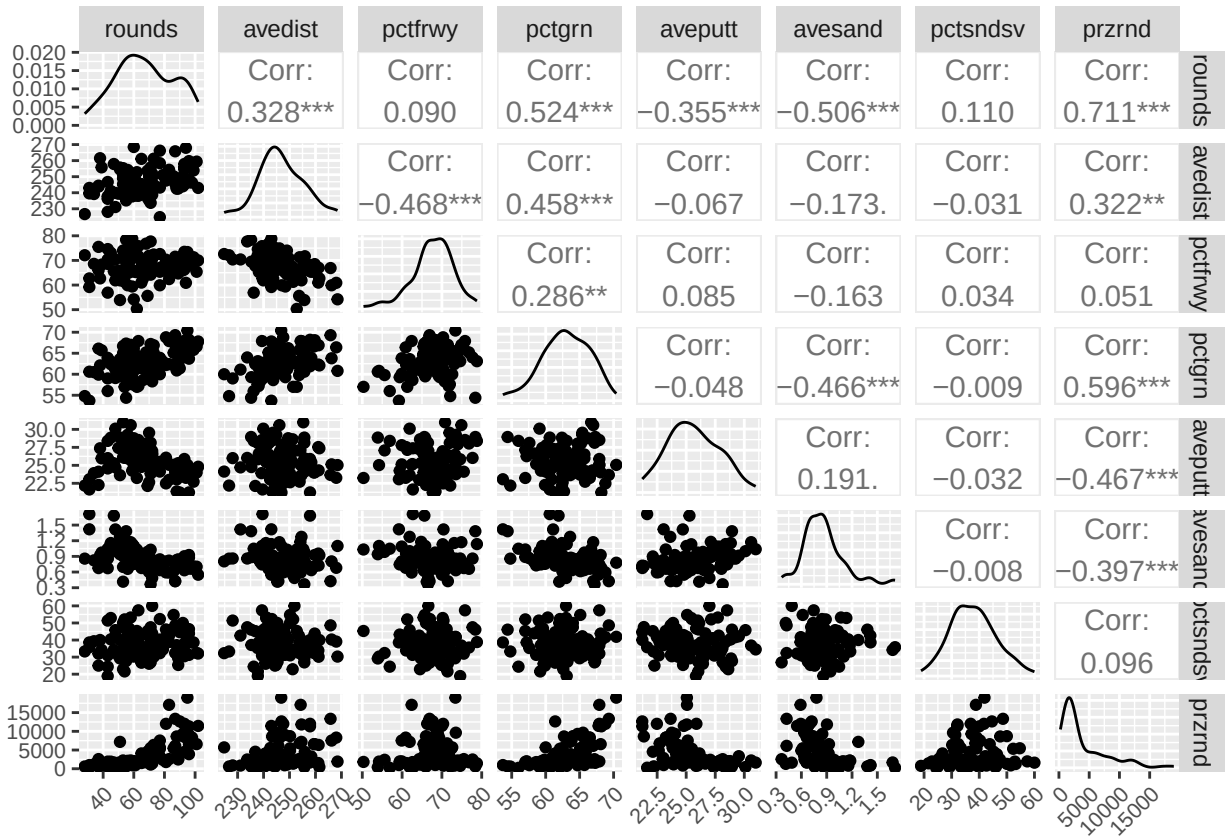


Figure 1: Pairwise relationships among variables in the dataset.

As can be seen in Figure 2, once our response variable had been transformed, the relationship between itself and many of the predictors went from non-linear to being far more linear in nature.

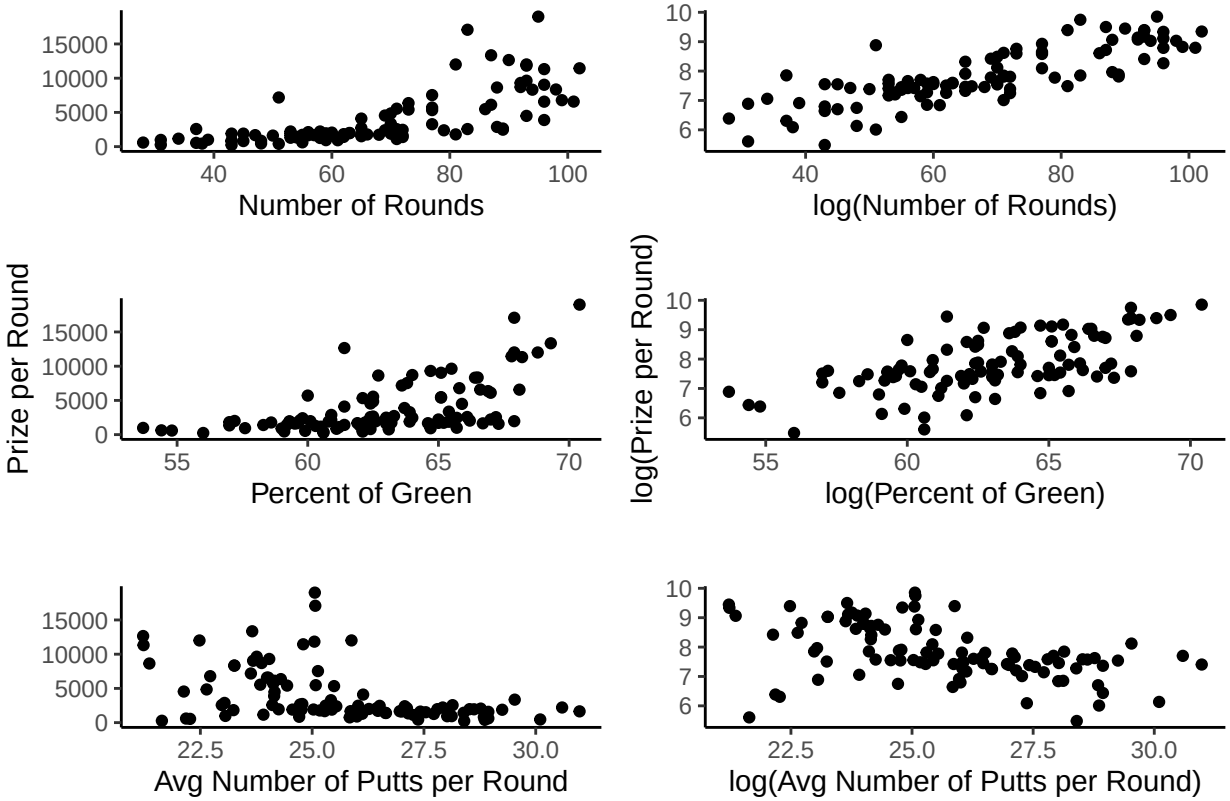


Figure 2: Prize Money per Round vs. Prize Money per Round

After transforming our response, our next step was to transform each of the predictors. To do this, we decided upon standardization by first subtracting off each predictor's mean then dividing by each predictor's standard deviation. This process was done for all predictors. Implementing these transformations can be beneficial in many different ways, but the one we cared the most about was interpretability. Most of the predictors were in different units from one another and all predictors were in different units than the response. By standardizing our predictors, we were able to compare the association each of them had on the response.

In order for these comparisons to mean anything in terms of the response variable, we also chose to standardize the response. Thus, each variable in our model, each predictor and the response, was standardized before any model was fit. Our next step was fitting a model.

In determining what our final model was going to look like, our first step was to fit a model with every possible predictor and all possible pairwise interactions. This meant that we fit a model that utilized the number of rounds played, the average drive distance, the percentage of fairways hit, the percentage of greens in regulation, the average number of putts per round, the average number of sand shots per round, the sand save percentage, the amount of prize money won per round, and all possible interactions between these variables. In total, this model had 28 terms not including the intercept.

Going into this model fitting, we knew a few things. First, most of these predictors were going to have very little association with the response variable. There were a few interactions we had hoped would be useful, such as a possible interaction between the percentage of fairways hit and the average drive distance or between sand save percentage and average sand shots per round, but most of the other interactions were

shots in the dark just to see if anything would surprise us. Because we knew that most of our predictors were going to be of little use, we decided upon using a default horseshoe prior when fitting our model.

The idea behind a horseshoe prior is that all non-essential predictors will receive a coefficient that has been shrunken down to 0, or nearly 0, which, in effect, removes it from the model. We knew that most of the predictors were not going to make the cut and planned on trimming our model down after this larger model had been fit, so trimming the number of predictors down using a horseshoe prior seemed like an obvious choice.

As we expected, most of the predictors had received a near-0 slope. This included every interaction that had been included in the model. The predictors that were left are given in the formula provided below:

$$\hat{prz\hat{r}nd} = \hat{\beta}_0 + \hat{\beta}_1 rounds + \hat{\beta}_2 pctgrn + \hat{\beta}_3 aveputt + \epsilon$$

As can be seen, only three of the 28 predictors ended up having any meaningful contribution or association toward the prediction of earned prize money.

This model was able to provide us with very valuable information regarding which characteristics of a golfer lead to higher payouts, on average. An additional benefit of this model is the sparsity resulting from the removal of the majority of predictors, which simplifies interpretation and inference.

After model fitting, assumptions of linearity, homoscedasticity, and normality were assessed with a residuals vs. fitted values plot and qq-plot. Each model assumption was satisfied.

With the assumptions checked, we can move onto discussing the variables.

First, the number of rounds a golfer has played. When looking at increases in prize money earned, the number of rounds a golfer has played has been estimated to have the most impact among all predictors. This, however, is not a surprise. When comparing two golfers to one another, the one with the higher number of rounds played can be described as being more experienced. It seems likely that a golfer with more experience will, on average, be a better golfer and thus win more prize money. The model seems to support this notion.

Second, the percent of greens in regulation. This statistic refers to whether or not the golfer is able to hit the ball "in the green," or within the area of the golf course that is the most near to the hole, within at least par minus two hits. When comparing two golfers to one another, the one that is better at golf is expected to hit the ball in the green within regulation on average more often than the one that is not as good at golf. Thus, it is expected that a higher percentage of greens in regulation will lead to a higher payout for that golfer. This relationship has been confirmed by the model that has been fit, though this relationship is not quite as strong as the relationship between the number of rounds and the expected payout.

Third, the average putts per round. This statistic is pretty straight forward. It is the average number of putts per round that a golfer has. A better golfer will have a lower average number of putts per round rather than a large number for the average putts per round. This is because, in golf, the goal is to hit the ball as few times as possible. This relationship is supported by the model that has been fit. This relationship is not as strong as the others, but there is a negative slope. The negative slope supports the idea that a better golfer will have a lower average number of putts per round.

Next, we can examine the values of the coefficients to get a better understanding of the magnitude of these relationships. As a reminder, the table below shows a few summary statistics for each variable.

First, we can examine the intercept of this model. When a golfer has played 67.28 rounds, has gotten 63% of their greens within regulation, and has an average number of putts of 25.62, she is expected to win \$3,879.98 (one standard deviation of prize money).

The distribution of prize money, however, is not symmetric, it has a very heavy right tail and is thus very positively skewed. This means that the maximum amount of prize money that can be earned is much greater than the mean amount of prize money. In fact, the mean is far closer to the minimum value than it is the

maximum. This implies that if a golfer is only average at playing golf, she will not be making very much per round compared to her peers that are better than her but instead closer to her peers that are worse.

Next, we can examine the relationship between the average number of rounds a golfer has played and her expected winnings. When two golfers differ by one standard deviation in average number of rounds played, we expect, on average, the one who has played more rounds of golf to win 70% more in terms of standardized earnings (when they have an equal percentage of greens in regulation and average number of putts).

This relationship is quite interesting to look at when we consider the spread of rounds played and money earned. This coefficient tells us that when a golfer plays 18 more rounds of golf, her expected amount of earnings will increase by a factor of roughly 1.7. 18 rounds of golf seems to be a lot of rounds, but the average is roughly 70 and the maximum from the dataset is just over 100, so 18 may not be that many rounds for nearly double the amount of winnings.

Next, we can look at the relationship between the response and the percent green. When two golfers differ by one standard deviation in the percentage of greens they have gotten within regulation, we expect, on average, the one who has played more rounds of golf to win 39% more in terms of standardized earnings (when they have played the same number of rounds and have the same average number of putts per round).

Once again, we can look at what these values mean in context. For reference, one standard deviation of percent green is 3.48% and the mean is nearly 63%. Thus, it can be said that, if a golfer is able to get her percentage of greens in regulation up by not even four percent, she can increase her earnings by a factor of nearly 1.5

Finally, we can examine the relationship between the response and average putt distance. When two golfers differ by one standard deviation in her average of number of putts per round, we expect, on average, the one who has fewer putts per round to earn 23% less than her peer in terms of standardized earnings (when they have played the same number of rounds and have the same percentage of greens within regulation).

To summarize, each coefficient, as well as its standard deviation, is shown in table 2 below.

Table 2

Table 2: Model Estimated Coefficients

Name	$\hat{\beta}_i$	MAD SD	$e^{\hat{\beta}_i}$
Intercept	0	0.047	1
rounds	0.543	0.062	1.722
pctgrn	0.342	0.055	1.408
aveputt	-0.263	0.051	0.769

To conclude, our analysis has shown that, if we want to get a good estimate of how much a golfer will make per round, on average, a good place to start would be to look at how many rounds of golf she has played.

This may be good for predictive purposes, but ultimately does a poor job of answering our overall question which pertained to which characteristics of a golfer might lead her to be better than her peers. What our analysis has shown is that better golfers have played, on average, more rounds of golf. This isn't exactly breaking news as playing more golf tends to improve a player's ability to play golf and thus makes them, on average, a better golf player. Regardless, if one wants to predict whether or not a golf player is good or not, looking at how many rounds she has played is a good place to start.

If we want to get into the actual characteristics of a good golfer, we can examine the other two predictors that we included in the model, percent green and average putt distance.

First, percent greens. It has been found that a good golfer, or at least one that has made more money per round of golf she has played, is able to get the ball into the green at least two hits before par (in regulation)

most of the time. This is unsurprising as getting the ball close to the hole in fewer shots than par typically means that you are a pretty good golfer. Regardless, it is good to know that increasing how often you are able to achieve a green in regulation does lead to earning more prize money per round on average.

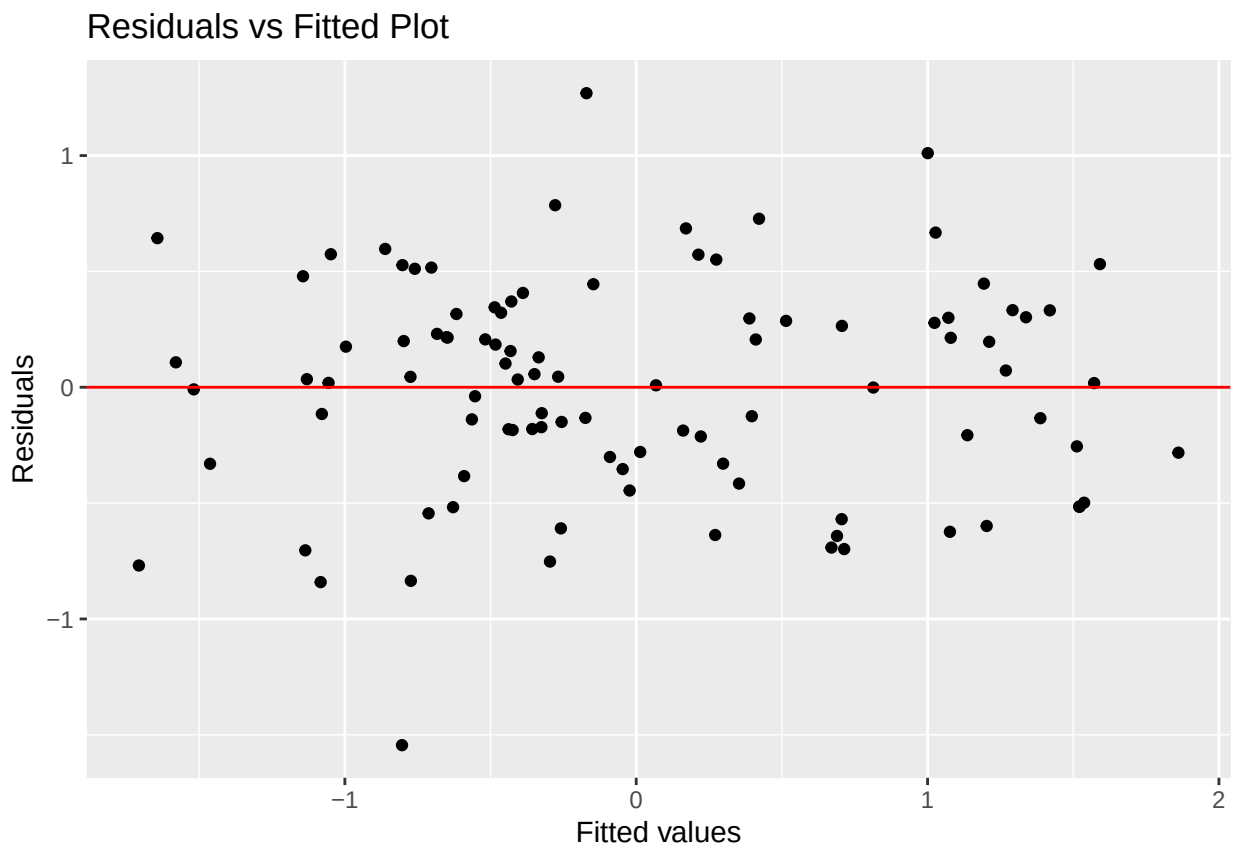
Finally, the average number of putts per round. It has been found that a good golfer tends to complete a round with fewer putts than her opponents. Once again, this isn't new information. The aim of golf is to hit the ball into the hole in as few putts as possible, so it makes sense that, on average, golfers that are able to hit the ball in the hole in fewer putts tend to make more money as they are likely better golfers than their peers that hit the ball in in more putts.

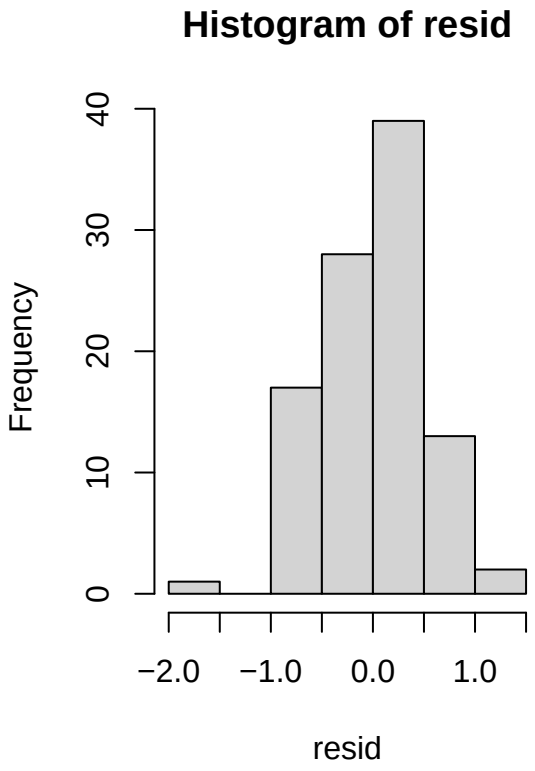
Inference, however, need not be the only goal of a regression analysis. We can instead turn our attention to the predictive power of this model. There are a few ways of analyzing how well this model performed. First, we can inspect the Bayesian R^2 value. It was found that this model achieved an R^2 of 0.772 which, for only three predictors, we found to be acceptable. This value, however, does not tell us the full story.

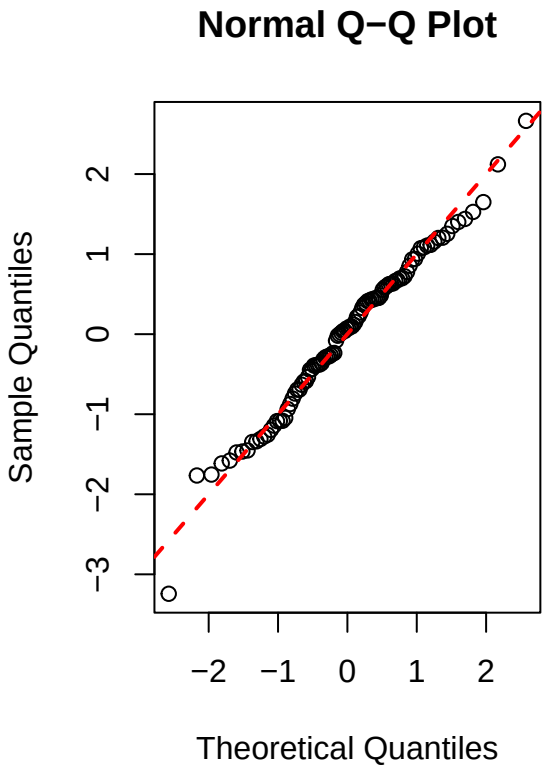
There were few observations in this dataset, meaning it may be the case that our model is fitting the data very well but will not generalize well to unseen data. To see if this is the case, we can calculate the Leave One Out Bayesian R^2 which estimates how the model would perform under cross validation. We found that, using this procedure, the model achieved an R^2 of 0.758. This value is very close to our full-sample R^2, indicating to us that overfitting doesn't appear to be a major problem.

Thus, we conclude that, by using the average number of rounds a golfer has played, her percentage of greens in regulation, and her average number of putts per round, we can fairly accurately predict the average number of money she will win from another round of golf.

Appendix


Residuals vs Fitted Plot
Residuals
Fitted values
1
0
−1
−1
0
1
2


Histogram of resid
Frequency
resid
0
10
20
30
40
−2.0
−1.0
0.0
1.0


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−3
−2
−1
0
1
2
−2
−1
0
1
2

```r
knitr::opts_chunk$set(echo = TRUE)
#packages
library(kableExtra)
library(tidyverse)
library(rstanarm)
library(cowplot)
library(GGally)
library(corrplot)
library(glmnet)
library(caret)
library(tableone)
library(gridExtra)
#data
dat <- read.csv("C:\\Users\\danie\\OneDrive\\Regression Models and Apps\\Case Study\\LPGA.csv")

lpga_ggpairdat <- dat %>%
  select(-1)
#summary table
summary_df <- dat %>%
  select(-Golfer) %>%
  summarise(across(everything(), list(
    Mean = ~mean(.),
    SD = ~sd(.),
    Median = ~median(.),
    Min = ~ min(.),
    Max = ~max(.)
  ))) %>%
  pivot_longer(
    cols = everything(),
    names_to = c("variable", "statistic"),
    names_sep = "_",
    values_to = "value"
  ) %>%
  pivot_wider(
    names_from = statistic,
    values_from = value
  )

summary_table <- kable(summary_df, caption = "Summary Statistics", digits = 2) %>%
                 kable_styling(latex_options = "striped", bootstrap_options = "striped")
summary_table
#ggpairs plot
ggpairs(lpga_ggpairdat) +
  theme(axis.text.x = element_text(angle = 45, hjust=1, size=8)) + theme(axis.text.y = element_text(size
#ggpairs plot
ggpairs(lpga_ggpairdat) +
  theme(axis.text.x = element_text(angle = 45, hjust=1, size=8)) + theme(axis.text.y = element_text(size
suppressWarnings({
plot_1 <- ggplot(lpga_ggpairdat, aes(x=rounds, y=przrnd)) + geom_point() + labs(x="Number of Rounds", y=
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 6),
    axis.title = element_text(size=9)
  )+ theme_classic()
```

```
plot_2 <- ggplot(lpga_ggpairdat, aes(x=rounds, y=log(przrnd))) + geom_point() + labs(x="log(Number of Ro
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 6),
    axis.title = element_text(size=9)
  )+ theme_classic()

##########################################################################################################

plot_3 <- ggplot(lpga_ggpairdat, aes(x=pctgrn, y=przrnd)) + geom_point() + labs(x="Percent of Green", y=
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 6),
    axis.title = element_text(size=9)
  )+ theme_classic()

plot_4 <- ggplot(lpga_ggpairdat, aes(x=pctgrn, y=log(przrnd))) + geom_point() + labs(x="log(Percent of G
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 6),
    axis.title = element_text(size=9)
  )+ theme_classic()

##########################################################################################################

plot_5 <- ggplot(lpga_ggpairdat, aes(x=aveputt, y=przrnd)) + geom_point() + labs(x="Avg Number of Putts
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 6),
    axis.title = element_text(size=9)
  ) + theme_classic()

plot_6 <- ggplot(lpga_ggpairdat, aes(x=aveputt, y=log(przrnd))) + geom_point() + labs(x="log(Avg Number
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 6),
    axis.title = element_text(size=9)
  )+ theme_classic()
})
suppressWarnings({
gridExtra::grid.arrange(plot_1,plot_2,plot_3,plot_4,plot_5, plot_6,
                                    ncol = 2,
                                    nrow = 3)
})
coeff_df <- rbind( c("Intercept", 0.0, 0.047, 1.0), c("rounds", 0.543, 0.062, 1.722), c("pctgrn", 0.342

 kable(coeff_df, booktabs = TRUE, col.names = c("Name", "$\\hat{\\beta_i}$", "MAD SD", "$e^{\\hat{\\beta
#residual plots
set.seed(2024)
lpga_raw <- read.csv("C:\\Users\\danie\\OneDrive\\Regression Models and Apps\\Case Study\\LPGA.csv")
log_data <- lpga_raw %>%
  select(-Golfer) %>%
  mutate("log_przrnd" = log(przrnd)) %>%
  scale() %>%
  data.frame()
fit_2 <- stan_glm(log_przrnd ~ rounds + pctgrn + aveputt, data = log_data, refresh=0)
resid <- residuals(fit_2)
fitted <- fitted(fit_2)
```

```r
df <- data.frame(resid, fitted)
ggplot(data = df, aes(x = fitted, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Fitted values", y = "Residuals", title = "Residuals vs Fitted Plot")


par(mfrow=c(1,2))
hist(resid)
qqnorm(resid / sigma(fit_2))
abline(a=0, b=1, lwd=2, lty=2, col="red")
set.seed(2024)
log_data <- lpga_raw %>%
  select(-Golfer) %>%
  mutate("log_przrnd" = log(przrnd)) %>%
  scale() %>%
  data.frame()

fit <- stan_glm(log_przrnd ~ (rounds+avedist+pctfrwy+pctgrn+aveputt+avesand+pctsndsv)^2, data=log_data,

mean(bayes_R2(fit))
mean(loo_R2(fit))
loo_fit1 <- loo(fit)

exp(coef(fit))[exp(coef(fit)) >= 1.1]
1- exp(coef(fit))[1 - exp(coef(fit)) >= .1]
##########
fit_2 <- stan_glm(log_przrnd ~ rounds + pctgrn + aveputt, data = log_data, refresh=0)

mean(bayes_R2(fit_2))
mean(loo_R2(fit_2))
loo_fit2 <- loo(fit_2)

exp(coef(fit_2))[exp(coef(fit_2)) >= 1.1]
1- exp(coef(fit_2))[1 - exp(coef(fit_2)) >= .1]


##########################################################
#compare loo
loo_compare <- loo_compare(loo_fit1, loo_fit2)
loo_compare

############################################################
#assumptions

resid <- residuals(fit_2)
fitted <- fitted(fit_2)
df <- data.frame(resid, fitted)

ggplot(data = df, aes(x = fitted, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Fitted values", y = "Residuals", title = "Residuals vs Fitted Plot")
```

```r
par(mfrow=c(1,2))
hist(resid)
qqnorm(resid / sigma(fit_2))
abline(a=0, b=1, lwd=2, lty=2, col="red")

######################################################################

#using posterior simulation for parameter estimation
sims <- as.matrix(fit_2)
hist(sims[ ,1], xlab=expression(beta[0]), main="")
abline(v=coef(fit_2)[1], lwd=3, lty=2, col="red")

hist(sims[ ,2], xlab=expression(beta[1]), main="")
abline(v=coef(fit_2)[2], lwd=3, lty=2, col="red")

hist(sims[ ,3], xlab=expression(beta[2]), main="")
abline(v=coef(fit_2)[3], lwd=3, lty=2, col="red")

hist(sims[ ,4], xlab=expression(beta[3]), main="")
abline(v=coef(fit_2)[4], lwd=3, lty=2, col="red")

Median <- apply(sims, 2, median)
MAD_SD <- apply(sims, 2, mad)

intervals <- apply(sims, 2, quantile, probs=c(0.025, 0.975))
print(t(intervals))

ggplot(lpga_ggpairdat, aes(x=rounds, y=przrnd)) + geom_point() + labs(x="Number of Rounds", y="Prize Mo
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 11),
    axis.title = element_text(size=9)
  )+ theme_classic()

ggplot(lpga_ggpairdat, aes(x=rounds, y=log(przrnd))) + geom_point() + labs(x="Number of Rounds", y="Tra
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 11),
    axis.title = element_text(size=9)
  )+ theme_classic()

##############################################################################################

ggplot(lpga_ggpairdat, aes(x=pctgrn, y=przrnd)) + geom_point() + labs(x="Percent of Green", y="Prize Mo
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 11),
    axis.title = element_text(size=9)
  )+ theme_classic()

ggplot(lpga_ggpairdat, aes(x=pctgrn, y=log(przrnd))) + geom_point() + labs(x="Percent of Green ", y="Tra
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 11),
    axis.title = element_text(size=9)
  )+ theme_classic()
```

```
#############################################################################################

ggplot(lpga_ggpairdat, aes(x=aveputt, y=przrnd)) + geom_point() + labs(x="Avg Number of Putts per Round
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 11),
    axis.title = element_text(size=9)
  ) + theme_classic()

ggplot(lpga_ggpairdat, aes(x=aveputt, y=log(przrnd))) + geom_point() + labs(x="Avg Number of Putts per
    axis.title.x = element_text(size = 11),
    axis.title.y = element_text(size = 11),
    axis.title = element_text(size=9)
  )+ theme_classic()

#############################################################################################

lpga_summ_raw1 <- lpga_ggpairdat %>% summarise(across(where(is.numeric), .fns =
                     list(Median = median,
                          Mean = mean,
                          Sd = sd))) %>%
  pivot_longer(everything(), names_sep='_', names_to=c('Variable', '.value'))

lpga_summ_raw2 <- lpga_summ_raw1 %>%
  mutate_at(2:4, round,2) %>%
  slice(1,4,5,8)

lpga_summ_raw2$Variable <- c("Rounds", "% of in Green in Regulation", "Average putts per Round", "Prize


lpga_summ <- lpga_summ_raw2 %>%
  mutate_at(1, as.factor)

kt <- function(lpga_summ) {
  knitr::kable(lpga_summ, digits=3,linesep='',booktabs=TRUE, caption = "Summary Statistics for LPGA", b
}
lpga_summ %>% kt()
#############################################################################################
GGpairs code

lpga_raw <- read.csv("C:\\Users\\danie\\OneDrive\\Regression Models and Apps\\Case Study\\LPGA.csv")

lpga_ggpairdat <- lpga_raw %>%
  select(-1)

ggpairs(lpga_ggpairdat) +
  theme(axis.text.x = element_text(angle = 45, hjust=1, size=8)) + theme(axis.text.y = element_text(siz

coeff_df <- rbind( c("Intercept", 0.0, 0.047, 1.0), c("rounds", 0.543, 0.062, 1.722), c("pctgrn", 0.342

 kable(coeff_df, booktabs = TRUE, col.names = c("Name", "$\\hat{\\beta_i}$", "MAD SD", "$e^{\\hat{\\beta
```