Placeholder Golf Project

Thomas Schechter and Ian Bogley

12/10/2021

Golf is a sport with diverse and detailed statistics. The purpose of this project will be to use machine learning algorithms to predict whether a player will be successful or not. To do so, we will use data from the PGA 2017 Tour.

In particular, our question will be specified as follows: What players will finish in the top 10 of an event at least once? (Is this accurate phrasing? Is this a good question? May reduce variability of our outcome variable by quite a bit in comparison to predicting number of top 10 finishes.)

To start, let's begin by reading in the data and install necessary packages.

```
#Package unloading
library(pacman)
p_load(tidyverse,data.table,tidymodels,stargazer,ggpubr,
       janitor,sandwich,xtable)

#Read in data
pga_data17 <- fread("PGATOUR_data2.csv")
```

Now, we will clean the data, subsetting to remove null and NA values.

```
#Subset data
pga_data <- pga_data17[1:195,] %>% clean_names() %>%
  mutate(top_ten_finisher = (number_of_top_tens>0))

#check for further na values
lapply(pga_data,FUN = function(x) {sum(is.na(x))})
```

```
## $player
## [1] 0
##
## $events_played
## [1] 0
##
## $points
## [1] 0
##
## $number_of_wins
## [1] 0
##
## $number_of_top_tens
## [1] 0
##
```

```
## $points_behind_lead
## [1] 1
##
## $rounds_played
## [1] 0
##
## $sg_putting_per_round
## [1] 0
##
## $total_sg_putting
## [1] 0
##
## $measured_rounds
## [1] 0
##
## $avg_driving_distance
## [1] 0
##
## $up_and_down_percent
## [1] 0
##
## $par_or_better
## [1] 0
##
## $missed_gir
## [1] 0
##
## $fairway_hit_percent
## [1] 0
##
## $fairways_hit
## [1] 0
##
## $possible_fairways
## [1] 0
##
## $gir_rank
## [1] 0
##
## $going_for_green_in_2_percent
## [1] 0
##
## $attempts_gfg
## [1] 0
##
## $non_attempts_gfg
## [1] 0
##
## $rtp_going_for_the_green
## [1] 0
##
## $rtp_not_going_for_the_grn
## [1] 0
##
```

```
## $hole_outs
## [1] 0
##
## $sand_save_percent
## [1] 0
##
## $number_of_saves
## [1] 0
##
## $number_of_bunkers
## [1] 0
##
## $total_o_u_par
## [1] 0
##
## $three_putt_percent
## [1] 0
##
## $total_3_putts
## [1] 0
##
## $sg_per_round
## [1] 0
##
## $sg_ott
## [1] 0
##
## $sg_apr
## [1] 0
##
## $sg_arg
## [1] 0
##
## $drives_320_percent
## [1] 0
##
## $total_drives_for_320
## [1] 0
##
## $total_drives
## [1] 0
##
## $rough_tendnecy_percent
## [1] 0
##
## $total_rough
## [1] 0
##
## $fairway_bunker_percent
## [1] 0
##
## $total_fairway_bunkers
## [1] 0
##
```

```
## $avg_club_head_speed
## [1] 0
##
## $fastest_ch_speed
## [1] 0
##
## $slowest_ch_speed
## [1] 0
##
## $avg_ball_speed
## [1] 0
##
## $fastest_ball_speed
## [1] 0
##
## $slowest_ball_speed
## [1] 0
##
## $avg_smash_factor
## [1] 0
##
## $highest_sf
## [1] 0
##
## $lowest_sf
## [1] 0
##
## $avg_launch_angle
## [1] 0
##
## $lowest_launch_angle
## [1] 0
##
## $steepest_launch_angle
## [1] 0
##
## $avg_spin_rate
## [1] 0
##
## $highest_spin_rate
## [1] 0
##
## $lowest_spin_rate
## [1] 0
##
## $avg_hang_time
## [1] 0
##
## $longest_act_hang_time
## [1] 0
##
## $shortest_act_hang_time
## [1] 0
##
```

```
## $avg_carry_distance
## [1] 0
##
## $longest_carry_distance
## [1] 0
##
## $shortest_carry_distance
## [1] 0
##
## $avg_score
## [1] 0
##
## $total_strokes
## [1] 0
##
## $total_rounds
## [1] 0
##
## $makes_bogey_percent
## [1] 0
##
## $bogeys_made
## [1] 0
##
## $holes_played
## [1] 0
##
## $age
## [1] 0
##
## $top_ten_finisher
## [1] 0
```

```
#There is one missing value in the "points behind lead" column, we will use the mean value to impute
pga_data$points_behind_lead[33] <- mean(pga_data$points_behind_lead,na.rm = T)
```

Now, let's graph potential factors in top ten finishes and look for correlations.

```
#Begin exploring the relationships in the data

#Correlation b/w rounds played and top tens (Is this looking for issues in representation? Players with
model1 <- lm(rounds_played~number_of_top_tens, pga_data)

tidy(model1)
```

```
## # A tibble: 2 x 5
##   term               estimate std.error statistic   p.value
##   <chr>                 <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)           74.0      1.53      48.5  5.33e-110
## 2 number_of_top_tens     1.65     0.443      3.72 2.60e-  4
```

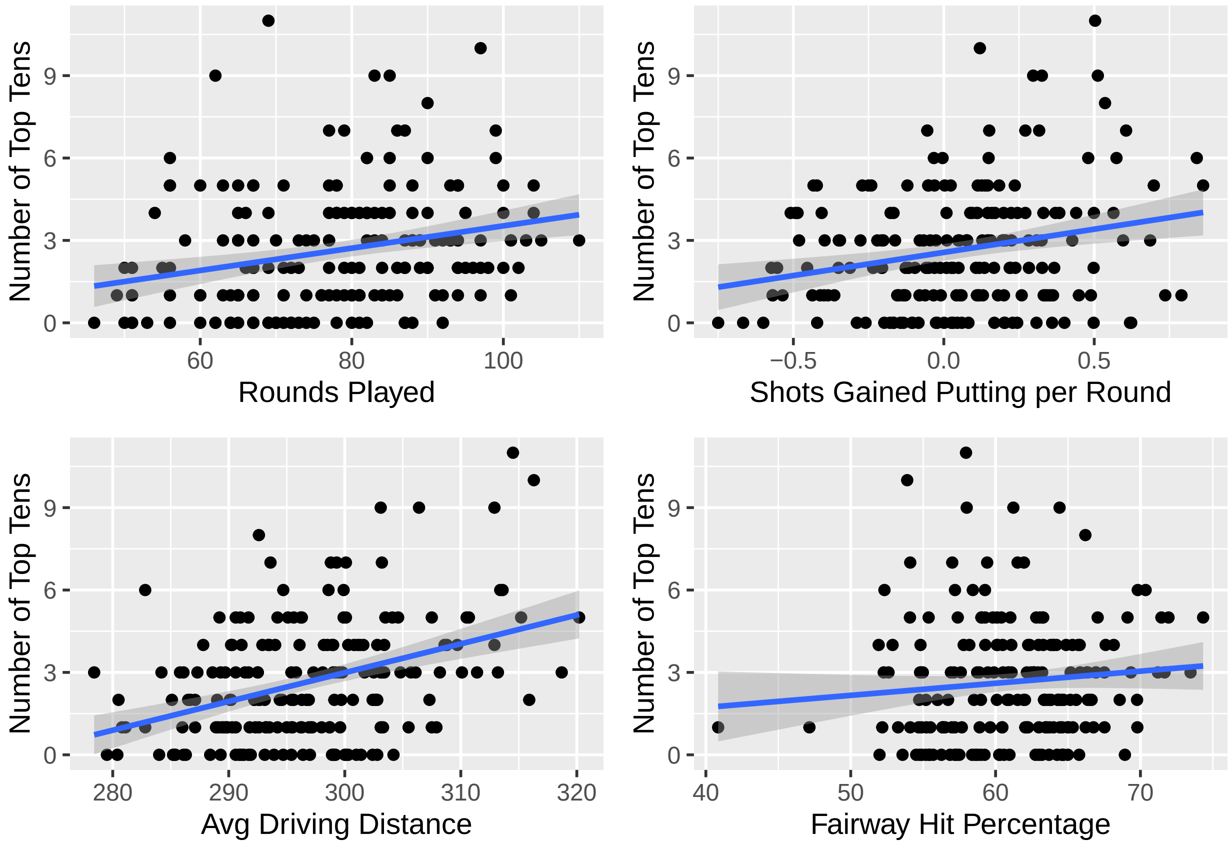
```r
#Plot the rounds played relationship
round_plot <-  ggplot(pga_data, aes(rounds_played, number_of_top_tens)) +
  geom_point() +  geom_smooth(method = "lm") +
  labs(x = "Rounds Played",
       y = "Number of Top Tens")

#Plotting strokes gained putting vs. top ten finishes
putt_plot <- ggplot(pga_data, aes(sg_putting_per_round, number_of_top_tens)) +
  geom_point() +  geom_smooth(method = "lm") +
  labs(x = "Shots Gained Putting per Round",
       y = "Number of Top Tens")

#Plotting average drive distance vs. top ten finishes
drive_plot <- ggplot(pga_data, aes(avg_driving_distance,number_of_top_tens)) +
  geom_point() + geom_smooth(method = "lm") +
  labs(x = "Avg Driving Distance",
       y = "Number of Top Tens")

#Plotting Fairway hit percentage vs top ten finishes
fairway_plot <- ggplot(pga_data, aes(x=fairway_hit_percent,y = number_of_top_tens)) +
  geom_point() + geom_smooth(method = "lm")+
  labs(x = "Fairway Hit Percentage",
       y = "Number of Top Tens")

#Arrange previous plots into a single graphic
ggarrange(round_plot,putt_plot,drive_plot,fairway_plot)
```



Not a significant amount of visual correlation between number of rounds played and number of top ten finishes.

Strokes gained putting seems to have more of an effect on success.

Some relationship appears present in average driving distance, though with more outliers on both ends of the spectrum.

There seems to be even less of a correlation in terms of fairway hit percentage than the previous ones. Even if there is, it is much smaller of an impact than the other factors examined.

Let's examine these relationships as linear regressions, identifying the impact of each and the robustness of the coefficients.

```
for (i in 1:4) {
  #vector of variable names
  variables <- c("rounds_played","sg_putting_per_round",
                 "avg_driving_distance","fairway_hit_percent")

  #models for each variable
  eval(parse(text = paste("model_",i+1,
                          " <- lm(number_of_top_tens ~ ",variables[i],
                          ",pga_data)",sep = "")))

  #robust se for each variable
  eval(parse(text = paste("robust_se_",i+1," <- list(sqrt(diag(vcovHC(model_",i+1,"))))",sep ="")))
}
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

% Date and time: Fri, Dec 17, 2021 - 5:24:30 PM

Table 1:

	Dependent variable:			
	number_of_top_tens			
	(1)	(2)	(3)	(4)
rounds_played	0.041^{***}			
	(0.011)			
sg_putting_per_round		1.689^{***}		
		(0.490)		
avg_driving_distance			0.105^{***}	
			(0.018)	
fairway_hit_percent				0.044
				(0.031)
Constant	−0.528	2.564^{***}	-28.463^{***}	−0.036
	(0.868)	(0.155)	(5.288)	(1.901)
Observations	195	195	195	195
R^2	0.067	0.058	0.152	0.010
Adjusted R^2	0.062	0.053	0.148	0.005
Residual Std. Error (df = 193)	2.132	2.142	2.032	2.196
F Statistic (df = 1; 193)	13.847^{***}	11.874^{***}	34.642^{***}	2.011
Note:		*p<0.1; **p<0.05; ***p<0.01		

We can also go further, collecting the variables with the highest correlation to our dependant variable.

```
cor_coef <- lapply(pga_data[,-c(1,4,5,70)],
                   FUN = function(x) {cor(x=as.numeric(x),
                                          y = pga_data$top_ten_finisher)})

top_var <- cor_coef[order(abs(unlist(cor_coef)),decreasing=T)[1:20]] %>%
  as.data.frame() %>% gather() %>%
  rename(var = "key", cor = "value")
```

```
xtable(top_var)
```

% latex table generated in R 4.0.3 by xtable 1.8-4 package % Fri Dec 17 17:24:30 2021

	var	cor
1	points_behind_lead	-0.41
2	points	0.37
3	avg_score	-0.35
4	total_rounds	0.34
5	holes_played	0.33
6	rtp_going_for_the_green	-0.33
7	total_strokes	0.33
8	attempts_gfg	0.32
9	sg_per_round	0.32
10	number_of_bunkers	0.32
11	rounds_played	0.31
12	number_of_saves	0.31
13	possible_fairways	0.29
14	sg_ott	0.29
15	sg_apr	0.27
16	par_or_better	0.26
17	measured_rounds	0.25
18	total_drives_for_320	0.24
19	bogeys_made	0.23
20	missed_gir	0.22