FİNAL PROJECT

Taha BAYAZ

14.09.2020

To be able to run the code blocks in this project, you need to load **tidyverse, knitr, tinytex, scales, randomForest, caret, patchwork** packages.

Part I: Short and Simple

Question 1:

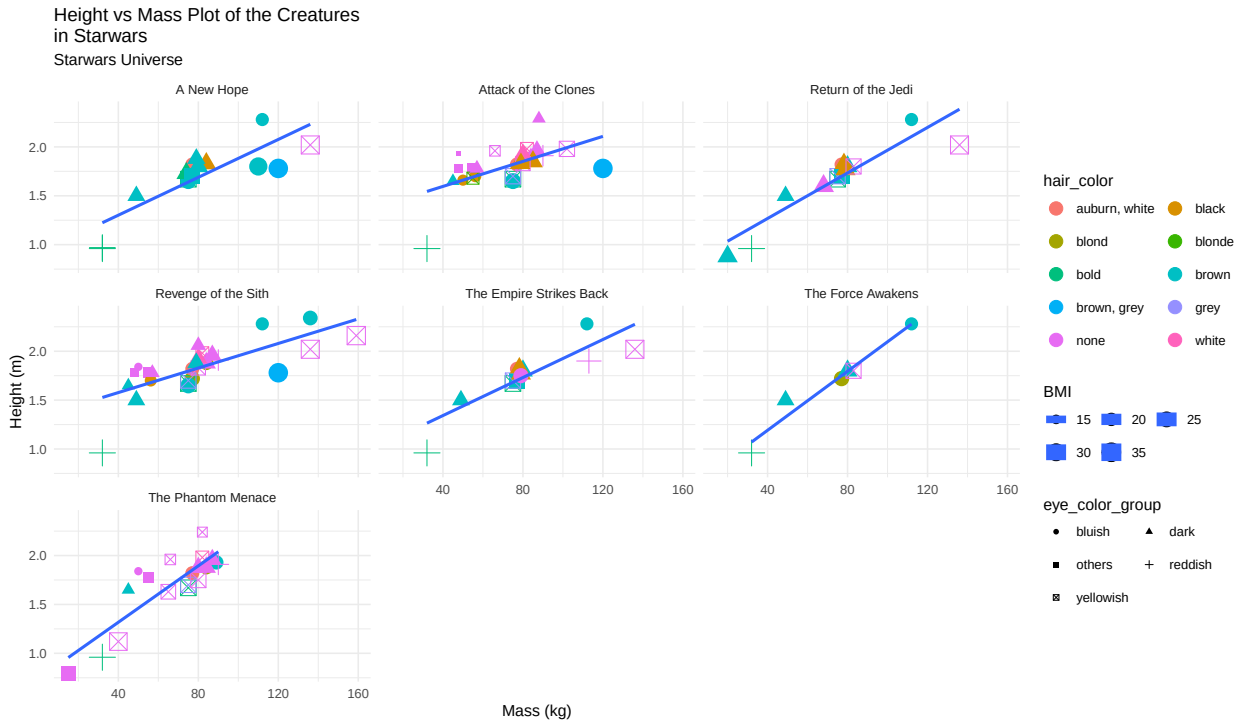
The tests are applied only on the people who show severe symptoms in some countries as a strategy, whereas others apply random testing even for people without any symptoms, which means we shouldn't compare these countries with different testing strategies. Demographics of countries such as population density, average age and rural population are way different from each other to make a decent comparison. Moreover, there is no accepted international standard for how you measure deaths, or their causes. For example, Germany counts deaths in care homes only if people have tested positive for the virus whereas Belgium includes any death in which a doctor suspects coronavirus was involved. Another difference with the countries is the stage of the outbreak an individual country has which means that If a country's first case was early in the global outbreak, then it has had longer for its death toll to grow. Also, It is harder to trust in data that comes from countries with highly controlled political systems which may explain the lower number of cases in China or Iran. Additionally, some countries record the number of people tested, while others record the total number of tests carried out which includes many people to be tested more than once to get an accurate result. Therefore, the testing and Covid19 statistics should have some international standard metrics that takes into account all different characteristics of countries. (Some examples are taken from BBC)

Question 2:

At first, after understanding the data, we need to check for null values whether we can fill these values with other values or we need to remove them and duplicated rows to remove. We need to correct the type of every column (like reordering the categories or changing the type) and check the values in every column whether it is a valid data or there could be some mistake about it. After these preprocessing steps we can check for some correlations of all variables, the distribution of the response variable and create different, valid models that fulfill the assumptions. If we have a project to distribute funds to maximize the impact on the society, firstly we should define how to measure, which would be the increase of the Human Development Index (HDI). For example, if you open classes for women to help them be employed, companies will have more perspectives that make all works more valuable. Another example would be like this: If we invest on health sector, death rate will decrease and qualified workers will live more than before, which makes HDI increase. But, at the end, we should give importance for all subjects rather than the most important one, because the optimal solution, in most cases, would be dividing the fund for more than one subject and marginal benefit to the society diminishes as the funds are increased for a specific policy.

Question 3:

```
starwars %>%  unnest(films) %>%
  drop_na(height, mass) %>%  mutate(bmi = mass / (height / 100)^2 ) %>%
  mutate(eye_color_group = case_when(eye_color %in% c("blue", "blue-gray", "red, blue")
  ~ "bluish", eye_color %in% c("dark", "brown", "black") ~ "dark", eye_color %in% c("red",
  "pink") ~ "reddish", eye_color %in% c("yellow", "green, yellow", "orange", "gold")
  ~ "yellowish",  eye_color %in% c("hazel", "unknown", "white") ~ "others")) %>%
  mutate(hair_color = ifelse(is.na(hair_color), "bold", hair_color)) %>%
  filter(bmi < 50) %>% filter(!is.na(homeworld) | name == "Qui-Gon Jinn") %>%
  mutate(sex = ifelse(is.na(sex), "female", sex), ) %>%
  ggplot(., aes(x = mass, y = height/100, size = bmi)) +
  geom_point(aes(color = hair_color, shape = eye_color_group)) +
  theme_minimal() + scale_y_continuous(breaks = seq(0,6, by = 0.5)) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Height vs Mass Plot of the Creatures
in Starwars", subtitle = st1, x = "Mass (kg)", y = "Height (m)") +
  facet_wrap(~ films) + scale_size_continuous(name="BMI") +
  guides(colour = guide_legend(ncol = 2, byrow = T, override.aes=list(size=4))) +
  guides(size = guide_legend(nrow = 2, byrow = T)) +
  guides(shape = guide_legend(ncol = 2, byrow = T)) +
  theme(legend.direction = "vertical", legend.box = "vertical")
```



When we look at the plots, we see that in every films there are people whose body mass indexes are around 20-25 and all those weights and heights of the actors / actresses are corralated with each other. It means that if you want to be an actor / actress in the next Starwars film, you should be in a fit body regardless of your eye color or hair color.

Part II: Extending Your Group Project

I couldn't find any different plots other than visualizing the distribution of the price. So, I tried to perform a better random forest algorithm. In the project, the R squared of the best random forest algorithm was 0.8313471. At the beginning, we need to load the data and divide to train and test set like in the Assignment 3 - Daimonds dataset.

```
data = readRDS(gzcon(url(paste("https://github.com/pjournal/boun01g-data-mine-r-s",
          "/blob/gh-pages/Project/turkey_car_market_EDA?raw=true", sep = ""))))
set.seed(503)
test = data %>% mutate(data_id = row_number()) %>% sample_frac(0.2)
train = anti_join(data %>% mutate(data_id = row_number()), test, by = "data_id")
train = train[, -18]
test = test[, -18]
categoricals = c("CCM", "Horse_Power", "Model_Year", "Kilometers")
```

After splitting the data, we can perform to create dummy variables for all categorical variables. To do so we can use the `dummyVars` function from `caret` package.

```
dmy = dummyVars("~.", data=train[,-c("Date", "Year", "Month", "Vehicle_Type_Group",
                                     "CCM", "Horse_Power", "Vehicle_Type")])
train_encoded = as.data.frame(predict(dmy, newdata = train))
```

We need to add the numerical and response variables to train data.

```
train_encoded = cbind(train_encoded,sapply(train[, c("CCM", "Horse_Power")], as.numeric))
```

To be able to get better model, we can scale the numerical variables.

```
min_vals = sapply(train_encoded[,categoricals], min)
max_vals = sapply(train_encoded[,categoricals], max)
train_sc = sweep(sweep(train_encoded[,categoricals],2,min_vals),2,max_vals-min_vals,"/")
train_encoded = train_encoded[,-c(37, 93, 95, 96)]
train_encoded = cbind(train_encoded, train_sc)
```

To be able to predict for test dataset, we need to apply the same steps to test dataset.

```
test_encoded = data.frame(predict(dmy, newdata = test))
test_encoded = cbind(test_encoded,sapply(test[,c("CCM","Horse_Power")],as.numeric))
test_sc=sweep(sweep(test_encoded[,categoricals], 2, min_vals),2,max_vals-min_vals,"/")
test_encoded = test_encoded[,-c(37, 93, 95, 96)]
test_encoded = cbind(test_encoded, test_sc)
```

Now, we are ready to create a random forest model and calculate the R squared value.

```
set.seed(1234)
model_rf = randomForest(Price ~ ., data = train_encoded)
pred_rf = predict(model_rf, newdata = test_encoded)
model_rf_r2 = 1 - (sum((pred_rf-test_encoded$Price)^2)/
              sum((test_encoded$Price-mean(train_encoded$Price))^2))
model_rf_r2
```

```
## [1] 0.9211378
```

The R squared value is 0.9211378, which is better than the model that we created in the project.

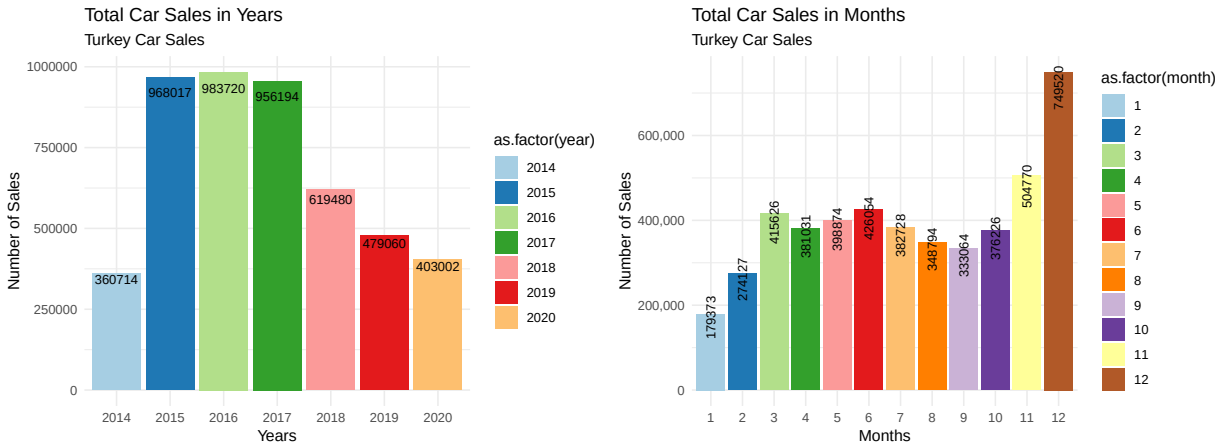
Part III: Welcome to Real Life

To start the analysis, we need to load the prepared data. This data is the transformation and combination of different Excel files.

```
all_data = readRDS(gzcon(url(paste("https://github.com/pjournal/boun01g-data-mine-r-s/",
                           "blob/gh-pages/Final%20TakeHome/all_data?raw=true", sep = ""))))
```

We can plot the number of total car sales in years and months.

```
year_plot = all_data %>%  group_by(year) %>% summarize(total = sum(total_total)) %>%
  ggplot(., aes(x = as.factor(year), y = total)) +
  geom_col(aes(fill = as.factor(year))) +
  geom_text(aes(label = total), size=3, color = "black",
            position = position_stack(vjust = 0.95)) +
  theme_minimal() + scale_fill_brewer(palette = c("Paired"))  +
  labs(title = "Total Car Sales in Years", subtitle = st,
       x = "Years", y = "Number of Sales")
month_plot = all_data %>%  group_by(month) %>% summarize(total = sum(total_total)) %>%
  ggplot(., aes(x = as.factor(month), y = total)) +
  geom_col(aes(fill = as.factor(month))) +
  geom_text(aes(label = total), size=3, color = "black",
            position = position_stack(vjust = 0.95), angle = 90) +
  theme_minimal() +  scale_y_continuous(labels = comma) +
  scale_fill_brewer(palette = c("Paired"))  +
  labs(title = "Total Car Sales in Months", subtitle = st,
       x = "Months", y = "Number of Sales")
(year_plot | month_plot)
```



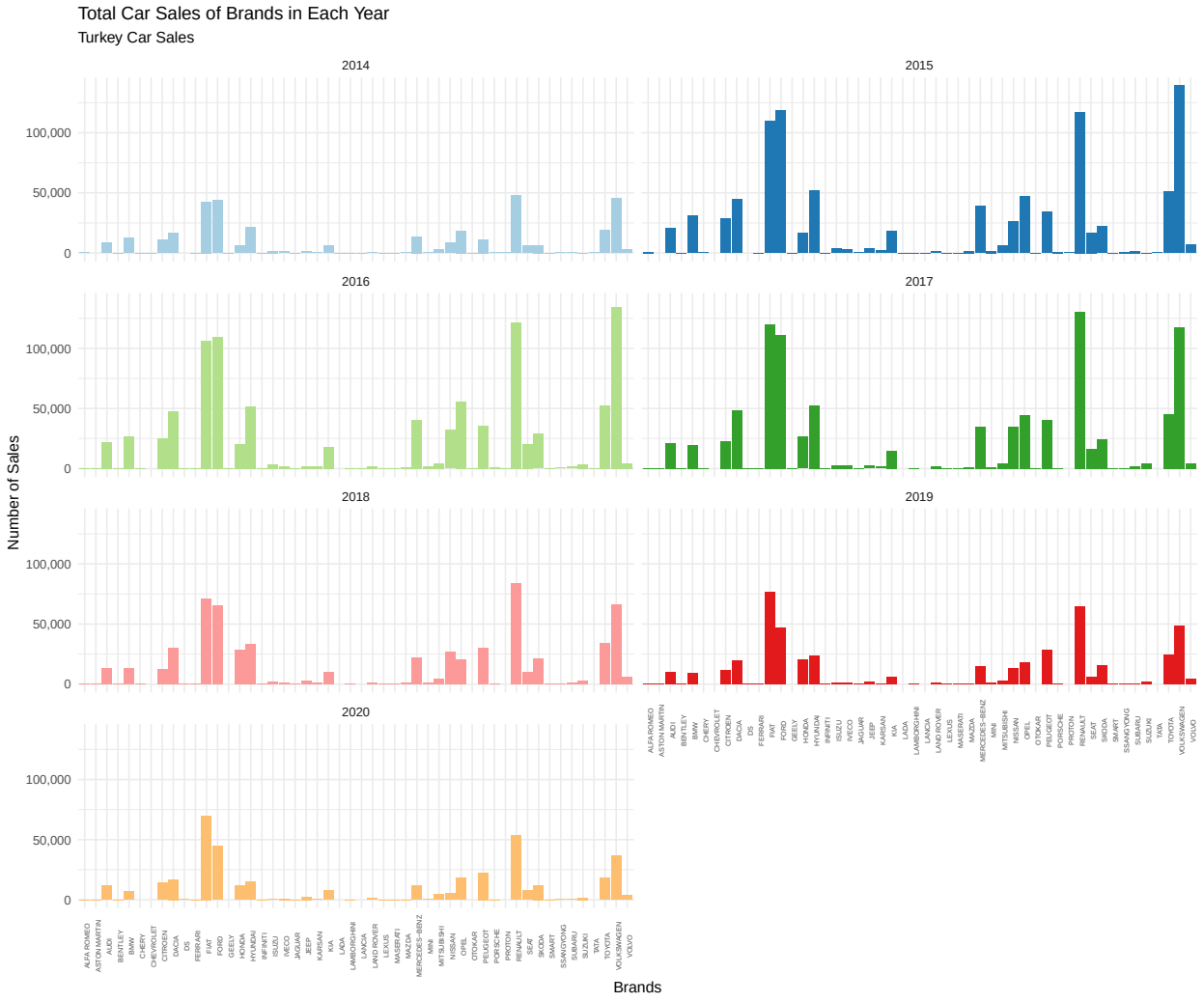
(This plot is prepared with *patchwork* library. You can see the samples from this link)

When we look at the year plot, there is a huge decrease after 2017. For 2020, we have only 8 months but there could be decrease in sales because of COVID-19 (This information is according to this link). The reason for other years could be the increase in car taxes for 2018 and 2019 (This assumption is made according to this link).

When we look at the month plot, the highest sales were realized towards the end of the year and the least sales were realized at the beginning of the year. Car sales are increasing from January to June and from July to December. The reason for this is that the car tax is calculated according to these time intervals (This assumption is made according to this link).

We can plot the sales of brands in each year.

```
all_data %>%
  group_by(year, brand_name) %>%
  summarize(total = sum(total_total)) %>%
  ggplot(., aes(x = as.factor(brand_name), y = total)) +
  geom_col(aes(fill = as.factor(year))) +
  theme_minimal() +
  scale_y_continuous(labels = comma) +
  facet_wrap(~ year, ncol = 2) +
  theme(axis.text.x = element_text(angle = 90, size = 5), legend.position = "none") +
  scale_fill_brewer(palette = c("Paired"))  +
  labs(title = "Total Car Sales of Brands in Each Year",
       subtitle = st,
       x = "Brands",
       y = "Number of Sales")
```



In all years **RENAULT, FIAT, VOLKSWAGEN, FORD** have more sales than the other brands. It means that they are more popular than the other brands.