Stat 305 Project Template - Insurance Policy

Rohit Nair

Teammates' Names: Ted Zybin, Ibrahim Alwishah

Research Question

How do the provided variables correlate with the prediction that a customer is high value?

External Requirements: Data Read-In and Package Loading

```
# read in the data in this codeblock

# first make sure this Rmd file and your csv file are in the same folder,
# you need to set the working directory under the Session menu (RStudio top)
# to the source file location

df = read.csv("Insurance_policy.csv")
```

```
# load any libraries in this codeblock, not later in the file,
# do not install packages in any Rmd file! Instead install packages
# at your console

library(mosaic)
library(ggplot2)
library(dplyr)
```

Count and Remove Some Missing Values as Appropriate (NAs)

```
# make a mental note on how many rows and columns we have at the start
dim(df)
```

```
## [1] 48842     8
```

```
# we see in the summary how many missing values in each variable
summary(df)
```

```
##       age        education_num   marital_status      occupation
##  Min.   :17.00   Min.   : 1.00   Length:48842       Length:48842
##  1st Qu.:28.00   1st Qu.: 9.00   Class :character   Class :character
##  Median :37.00   Median :10.00   Mode  :character   Mode  :character
##  Mean   :38.64   Mean   :10.08
##  3rd Qu.:48.00   3rd Qu.:12.00
##  Max.   :90.00   Max.   :16.00
##     cap_gain     hours_per_week      score        value_flag
##  Min.   :    0   Min.   : 1.00   Min.   :43.94   Length:48842
##  1st Qu.:    0   1st Qu.:40.00   1st Qu.:57.50   Class :character
##  Median :    0   Median :40.00   Median :60.24   Mode  :character
```

```
##  Mean   : 1079   Mean   :40.42   Mean   :60.23
##  3rd Qu.:    0   3rd Qu.:45.00   3rd Qu.:62.95
##  Max.   :99999   Max.   :99.00   Max.   :76.53
```

```
# more visibly we can see the number of missing values in each variable this way
colSums( is.na(df)  );
```

```
##            age  education_num marital_status     occupation       cap_gain
##              0              0              0              0              0
## hours_per_week          score     value_flag
##              0              0              0
```

```
# how many rows have 0 NAs, in other words, how many rows are complete?
sum( complete.cases(df) );
```

```
## [1] 48842
```

```
# how many rows are incomplete? Wow, 27% of the rows are incomplete rows.
sum( !complete.cases(df)  )
```

```
## [1] 0
```

```
# take age and score
age_score = select(df, age, score);
# only 37 rows are missing, instead of 42!
sum( !complete.cases(age_score));
```

```
## [1] 0
```

```
# take only those rows that have both ozone and temp
age_score = age_score[ complete.cases(age_score), ];
# confirm there are no missing values
colSums(  is.na(age_score))
```

```
##   age score
##     0     0
```

Dataset Description

This dataset contains data about policyholders in a certain insurance provider. It records the age, a score for amount of education, martial status, occupation, capital gains on investments, hours worked per week, an insurance score, and assigns a value flag to each policyholder. The goal of this dataset is to be used for some form of risk assessment.

There were no NAs within the dataset, so no rows were removed.

Some variables of interest include:

- age -> Units: years -> numerical
- education_num -> Units: NA -> numerical
- occupation -> Units: NA -> categorical
- score -> Units: NA -> numerical
- cap_gain -> Units: USD -> numerical
- value_flag -> Units: NA -> categorical

This is observational data, from which you can infer association and correlation, but not causation.

Data Transformation

```
# check that variables you think should be factors are factors!
# str(df);
# darn, it tells me the month is an integer not a factor, which
# can't be right, since I can't average months.
# we might also consider recoding day

# so let's recode it and check it is now a factor.
df$marital_status = as.factor(df$marital_status);
df$occupation = as.factor(df$occupation);
df$value_flag = as.factor(df$value_flag);
str(df)
```

```
## 'data.frame':    48842 obs. of  8 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ education_num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital_status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 5 3 1 3 3 3 4 3 5 3 ...
##  $ occupation    : Factor w/ 6 levels "Group 1","Group 2",..: 2 5 1 1 5 5 1 5 5 5 ...
##  $ cap_gain      : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ hours_per_week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ score         : num  59 55.8 62.8 60.1 53.3 ...
##  $ value_flag    : Factor w/ 2 levels "High","Low": 2 2 2 2 2 2 2 1 1 1 ...
```

Exploratory Data Analysis: Descriptive Statistics and Visualizations

```
num_df <- df %>%
          select(age,education_num,cap_gain,
                         hours_per_week,score) %>%
          filter(age >= 25,cap_gain != 0) %>%
          arrange(age)
head(num_df)
```

```
##   age education_num cap_gain hours_per_week score
## 1  25            10     2174             40 62.28
## 2  25             9     3325             45 62.67
## 3  25            10     2597             48 59.23
## 4  25            12     2354             45 66.82
## 5  25            13     6849             50 60.53
## 6  25             9     7298             84 59.48
```

```
favstats_vec = c()
columns = colnames(num_df)

total_favstats = data.frame()

for (i in 1:ncol(num_df)){
  total_favstats <- rbind(total_favstats,favstats(num_df[,i]))
}

total_favstats <- cbind(names = columns,total_favstats)
rownames(total_favstats) <- NULL

total_favstats
```

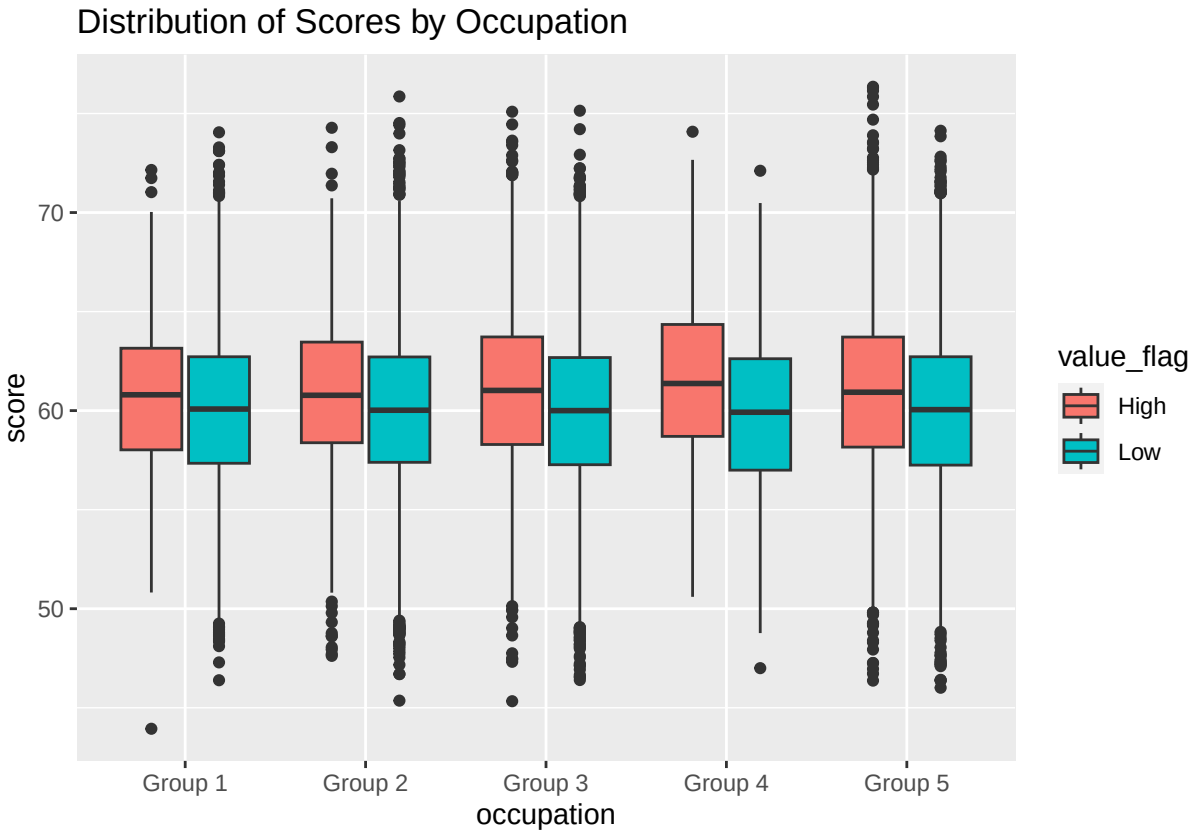
```
##            names    min      Q1  median         Q3      max        mean
## 1            age  25.00   36.00   44.00    53.0000    90.00    45.42797
## 2  education_num   1.00    9.00   11.00    13.0000    16.00    11.15689
## 3       cap_gain 114.00 3674.00 7298.00 14084.0000 99999.00 13460.39969
## 4 hours_per_week   1.00   40.00   40.00    50.0000    99.00    44.01833
## 5          score  43.94   57.91   60.73    63.4475    76.35    60.71336
##             sd    n missing
## 1    12.519084 3818       0
## 2     2.677252 3818       0
## 3 22991.335330 3818       0
## 4    12.201443 3818       0
## 5     4.127331 3818       0
```

1)
```
scores_by_valueflag <- df %>%
  filter(age >= 25, occupation != "Group NA") %>%
  select(occupation,score,value_flag) %>%
  mutate_at(vars(occupation,value_flag),list(factor))


print(str(scores_by_valueflag));
```

```
## 'data.frame':    38635 obs. of  3 variables:
##  $ occupation: Factor w/ 5 levels "Group 1","Group 2",..: 2 5 1 1 5 5 1 5 5 5 ...
##  $ score     : num  59 55.8 62.8 60.1 53.3 ...
##  $ value_flag: Factor w/ 2 levels "High","Low": 2 2 2 2 2 2 2 1 1 1 ...
## NULL
```

```
ggplot(scores_by_valueflag, aes(x=occupation,y=score)) +
  geom_boxplot(aes(fill=value_flag)) +
  ggtitle("Distribution of Scores by Occupation")
```



The graph above explicates the association between type of occupation and score according to the value flag they were labeled with. It's shown that the median scores for all types of occupations never go above 60 for those labeled with `low`.

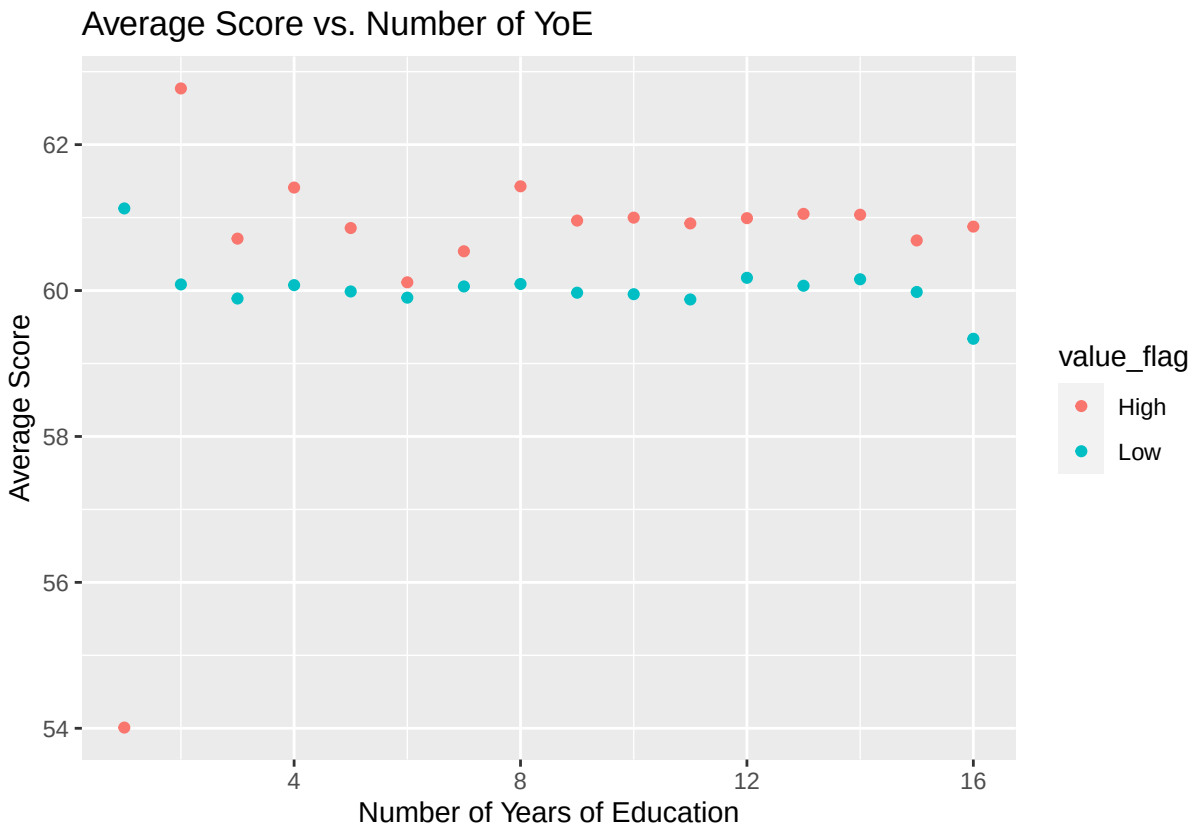
```
education_score <- df %>%
  select(education_num,score,value_flag) %>%
  group_by(education_num,value_flag) %>%
  summarize(avg_score = mean(score))
```

```
## `summarise()` has grouped output by 'education_num'. You can override using the
## `.groups` argument.
```

```
education_score
```

```
## # A tibble: 32 x 3
## # Groups:   education_num [16]
##    education_num value_flag avg_score
##            <int> <fct>          <dbl>
##  1             1 High            54.0
##  2             1 Low             61.1
##  3             2 High            62.8
##  4             2 Low             60.1
##  5             3 High            60.7
##  6             3 Low             59.9
##  7             4 High            61.4
##  8             4 Low             60.1
##  9             5 High            60.9
## 10             5 Low             60.0
## # ... with 22 more rows
```

```
ggplot(data = education_score,mapping = aes(x = education_num,y = avg_score,color = value_flag)) +
  geom_point() +
  xlab("Number of Years of Education") +
  ylab("Average Score") +
  ggtitle("Average Score vs. Number of YoE")
```



This graph shows the association between the number of years of education and the average score. It also shows that those with scores 60 or below tend to be classified as low.

```
age_value_by_occupation <- df %>%
  filter(age >= 25,occupation != "Group NA") %>%
  select(age,occupation,value_flag) %>%
  group_by(age,occupation,value_flag) %>%
  summarize(count=count(value_flag))
```

```
## `summarise()` has grouped output by 'age', 'occupation'. You can override using
## the `.groups` argument.
```

```
age_value_by_occupation
```

```
## # A tibble: 569 x 4
## # Groups:   age, occupation [306]
##      age occupation value_flag count
##    <int> <fct>      <fct>      <int>
##  1    25 Group 1    High           1
##  2    25 Group 1    Low            0
##  3    25 Group 2    High          14
##  4    25 Group 2    Low            0
##  5    25 Group 3    High          26
```

```
##  6    25 Group 3    Low            0
##  7    25 Group 4    High           7
##  8    25 Group 4    Low            0
##  9    25 Group 5    High          27
## 10    25 Group 5    Low            0
## # ... with 559 more rows
```

```
ggplot(data = age_value_by_occupation, mapping = aes(x = age, y = count,color = value_flag)) +
  geom_point() +
  xlab("Age (Years)") +
  ylab("Count of Value_Flag") +
  ggtitle("Value_Flag by Age, based on Occupation") +
  facet_wrap(~occupation,nrow=2)
```

The graph above looks at the total incidence of each value_flag at all ages present in dataset, while divided into facets based on the type of occupation. For Groups 1-3, there's a higher number of people classified as `low` for the bulk of the recorded ages.

```
data <- df %>%
  filter(age >= 25,occupation != "Group NA",cap_gain != 0) %>%
  group_by(marital_status,occupation,value_flag)

knitr::kable(summary(data))
```

age	education_num	marital_status	occupation	cap_gain	hours_per_week	value_flag
Min. :25.00	Min. : 1.00	Divorced : 421	Group 1 : 223	Min. : 114	Min. : 1.00	Min. :43.94	High:2399
1st Qu.:36.00	1st Qu.: 9.00	Married-AF-spouse : 2	Group 2 : 592	1st Qu.: 3908	1st Qu.:40.00	1st Qu.:57.92	Low :1268

age	education_num	marital_status	occupation	cap_gain	hours_per_week	score	value_flag
Median :43.00	Median :11.00	Married-civ-spouse :2537	Group 3 :1052	Median : 7298	Median :40.00	Median :60.76	NA
Mean :44.86	Mean :11.21	Married-spouse-absent: 32	Group 4 : 190	Mean :13689	Mean :44.46	Mean :60.73	
3rd Qu.:52.00	3rd Qu.:13.00	Never-married : 494	Group 5 :1610	3rd Qu.:14344	3rd Qu.:50.00	3rd Qu.:63.45	NA
Max. :90.00	Max. :16.00	Separated : 78	Group NA: 0	Max. :99999	Max. :99.00	Max. :76.35	NA
NA	NA	Widowed : 103	NA	NA	NA	NA	NA

Statistical Analysis: Confidence Interval, Hypothesis Test, and Model or Machine Learning

1) **On your own,** create at least one bootstrap confidence interval. Discuss with your team so that you do NOT do the same confidence interval.

2) **On your own,** do at least one hypothesis test. Discuss with your team so that you do NOT do the same hypothesis test. Clearly state what the population is (remember, the sample is a subset of the population). You want to infer from the sample to the population, so select the appropriate population based on your dataset. Clearly state the hypotheses in terms of population parameters:
H_0 : blah blah blah
H_A : blah blah blah
State your conclusion for each test in a phrase like:
"We reject the null hypothesis. The data provide convincing evidence at the .05 significance level that. . . ."
or
"We fail to reject the null hypothesis. The data *do not* provide convincing evidence at the .05 significance level that. . . ."

3) **On your own,** fit at least one model, or use at least one machine learning technique. Discuss with your team so that you do not do the same thing.

One more thing

Do one more thing that interests you. Perhaps another graph, or another hypothesis test, or try to fit another model, or do a simulation to make a sampling distribution of an estimator, or get another data set from Kaggle and somehow link it to this one. **Be creative and push your boundaries!**

Conclusion

Write one paragraph that summarizes what you did in this Rmd file and reiterate your main finding(s).