Stat 305 Project Template - Insurance Policy

Rohit Nair

Teammates' Names: Ted Zybin, Ibrahim Alwishah

Research Question

How are the provided variables associated with the prediction that a customer is of 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)
library(e1071)
```

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
```

```r
# 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
```

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

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

```r
# how many rows are incomplete?
sum( !complete.cases(df)  )
```

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

```r
# take age and score

age_score = select(df, age, score);

sum( !complete.cases(age_score));
```

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

```r
# get only the rows with both age and score
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, marital 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 the `age` which is numerical and uses the unit of years, `education_num` which is a unit-less numerical indicator of the number of years of education, `occupation` which is a unit-less categorical variable which assigns an occupation group to an individual, `score` which is a proprietary insurance numerical score rounded to two decimal places with no units, `cap_gain` which is a numerical variable with unit USD, `marital_status` which is a categorical variable with no units, and `hours_per_week` which is a numerical variable with uses the unit of hours.

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

Data Transformation

```r
# check that variables you think should be factors are factors!
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: chr  "Never-married" "Married-civ-spouse" "Divorced" "Married-civ-spouse" ...
##  $ occupation    : chr  "Group 2" "Group 5" "Group 1" "Group 1" ...
##  $ 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    : chr  "Low" "Low" "Low" "Low" ...
```

```r
# setting marital status, occupation, and value flag as factors
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

```r
# filtering out individuals less than 25 years of age and with a cap gain of 0.
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
```

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

total_favstats = data.frame()

# row binding favstats for each column in num_df
for (i in 1:ncol(num_df)){
  total_favstats <- rbind(total_favstats,favstats(num_df[,i]))
}

# column binding variable names
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)
```r
scores_by_valueflag <- df %>%
  filter(age >= 25) %>%
  select(occupation,score,value_flag) %>%
  mutate_at(vars(occupation,value_flag),list(factor))


print(str(scores_by_valueflag));
```

```
## 'data.frame':    40410 obs. of  3 variables:
##  $ occupation: Factor w/ 6 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
```

```
# boxplot showing distribution of scores by occupation
ggplot(scores_by_valueflag, aes(x=occupation,y=score)) +
  geom_boxplot(aes(fill=value_flag)) +
  xlab("Occupation (in Categorical Groups)") +
  ylab("Insurance Scores") +
  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`.

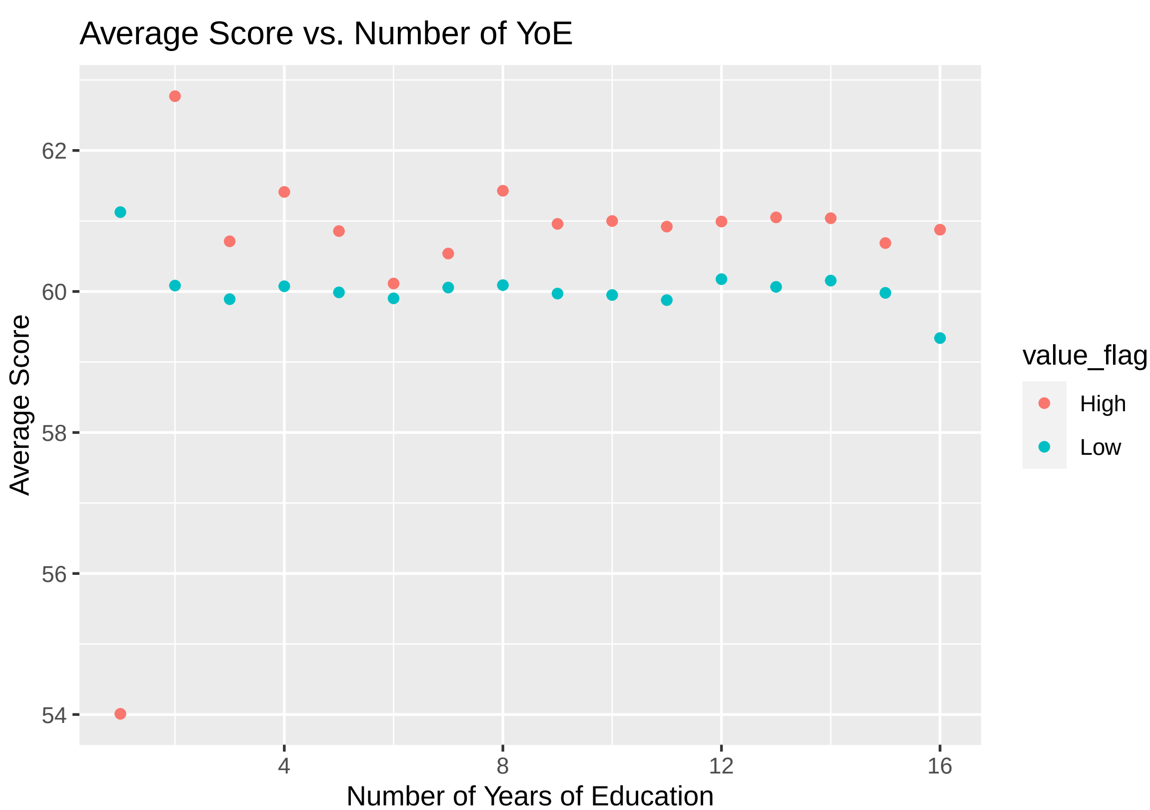
```
# getting the average scores for education level and value flag groups
education_score <- df %>%
  select(education_num,score,value_flag) %>%
  group_by(education_num,value_flag) %>%
  summarize(avg_score = mean(score))
```

```
## `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.

```r
# getting value_flag counts for individuals of a certain age and occupation
age_value_by_occupation <- df %>%
  filter(age >= 25) %>%
  select(age,occupation,value_flag) %>%
  group_by(value_flag,occupation,age) %>%
  summarize(count=count(value_flag == "High"| value_flag == "Low"))
```

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

```r
age_value_by_occupation
```

```
## # A tibble: 688 x 4
## # Groups:   value_flag, occupation [12]
##    value_flag occupation   age count
##    <fct>      <fct>      <int> <int>
##  1 High       Group 1       25     1
##  2 High       Group 1       26     2
##  3 High       Group 1       27     7
##  4 High       Group 1       28     9
##  5 High       Group 1       29    10
##  6 High       Group 1       30     9
##  7 High       Group 1       31     9
##  8 High       Group 1       32     8
##  9 High       Group 1       33     5
## 10 High       Group 1       34    14
## # ... with 678 more rows
```

```r
# facet wrap
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,cap_gain != 0) %>%
  group_by(marital_status,occupation,value_flag) %>%
  summarize(
            avg_education_num = mean(education_num),
            avg_cap_gain = mean(cap_gain))
```

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

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

marital_status	occupation	value_flag	avg_education_num	avg_cap_gain
Divorced :12	Group 1 :11	High:32	Min. : 7.235	Min. : 1847
Married-AF-spouse : 2	Group 2 :12	Low :35	1st Qu.: 9.099	1st Qu.: 3239
Married-civ-spouse :12	Group 3 :12	NA	Median :10.333	Median : 4348
Married-spouse-absent: 8	Group 4 : 9	NA	Mean :10.589	Mean :12057
Never-married :12	Group 5 :13	NA	3rd Qu.:11.550	3rd Qu.:20021
Separated :10	Group NA:10	NA	Max. :16.000	Max. :53648
Widowed :11	NA	NA	NA	NA

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

Bootstrap Confidence Interval

The population is all individuals older than or equal to the age of 25 and greater than 0 USD in capital gains with a value flag of High.

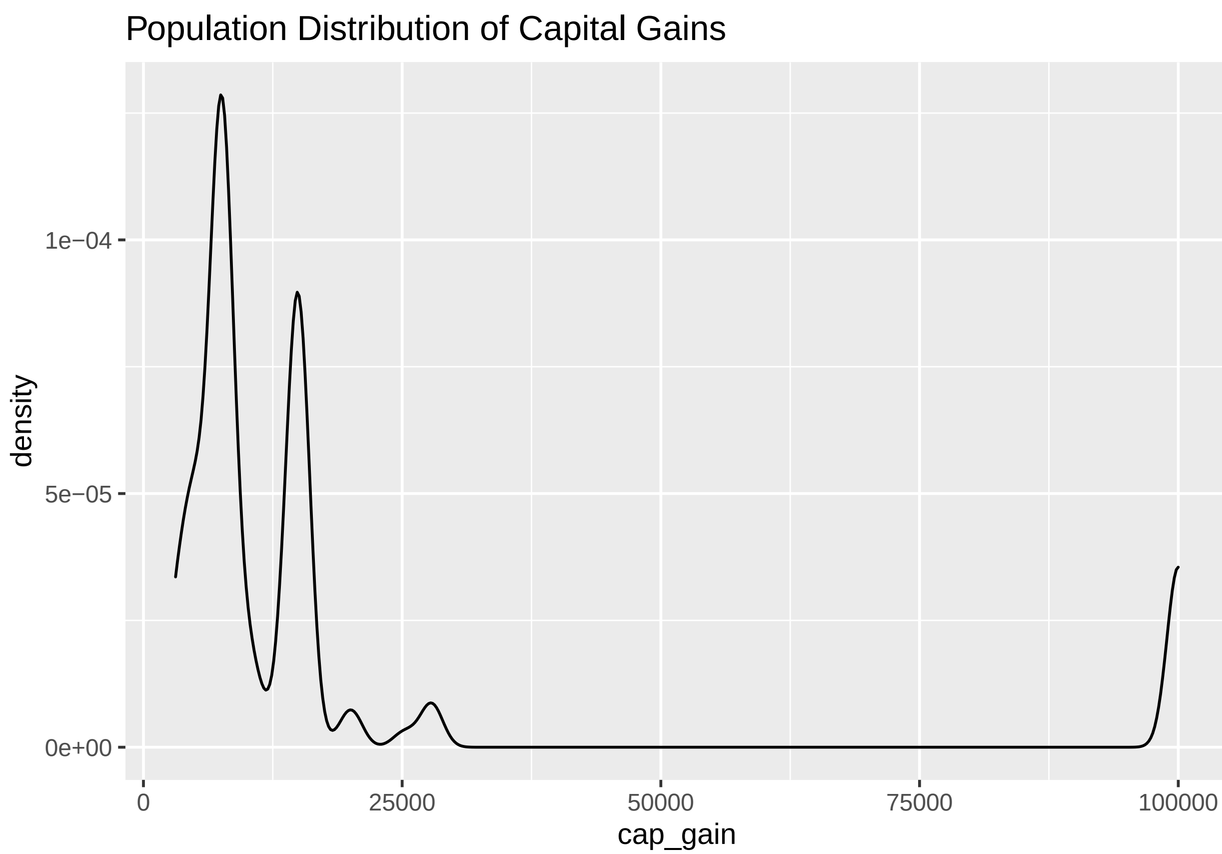
```
# find the population of all individuals 25 or older with greater than 0 USD in cap gains and with a va
population <- df %>%
  filter(age >= 25, cap_gain > 0,value_flag=="High")

head(population);
```

```
##   age education_num     marital_status occupation cap_gain hours_per_week score
## 1  31            14      Never-married    Group 5    14084             50 63.93
## 2  42            13 Married-civ-spouse    Group 5     5178             40 59.70
## 3  44             9           Divorced    Group 3    14344             40 51.69
## 4  44            13 Married-civ-spouse    Group 5    15024             60 59.65
## 5  32             9 Married-civ-spouse    Group 3     7688             40 56.10
## 6  40            14      Never-married    Group 5    14084             55 61.17
##   value_flag
## 1       High
## 2       High
## 3       High
## 4       High
## 5       High
## 6       High
```

```
#population size
dim(population)[1]
```

```
## [1] 2462
```

```
# detecting population skew
ggplot(population,aes(x = cap_gain)) +
  geom_density() +
  ggtitle("Population Distribution of Capital Gains")
```




```
set.seed(101)
# select a sample, without replacement
sample = population %>%
  sample_n(size = 200,replace = FALSE)
head(sample)
```

```
##   age education_num     marital_status occupation cap_gain hours_per_week score
## 1  46            13 Married-civ-spouse    Group 5     3103             37 63.98
## 2  46            14 Married-civ-spouse    Group 5    99999             50 59.28
## 3  46            13 Married-civ-spouse    Group 5    15024             55 60.48
## 4  51            14 Married-civ-spouse    Group 5     7298             50 61.26
## 5  40            14 Married-civ-spouse    Group 5    15024             30 55.30
## 6  46            13           Divorced    Group 3     8614             40 59.61
##   value_flag
## 1       High
## 2       High
## 3       High
## 4       High
## 5       High
## 6       High
```

```
# take 1000 random bootstrap samples from the original sample with replacement

bootstrap_samplemeans <- do(1000)*mean(~cap_gain,
                     data = sample_n(population,size=200,
```

```
                                          replace = TRUE))
bootstrap_samplemeans <- bootstrap_samplemeans %>%
                         rename(boostrap_samplemeans = mean)
head(bootstrap_samplemeans)
```

```
##   boostrap_samplemeans
## 1             21828.93
## 2             17872.95
## 3             21766.25
## 4             19839.31
## 5             19064.01
## 6             20245.12
```

```
confidenceinterval = quantile(bootstrap_samplemeans$boostrap_samplemean,
                              probs = c(0.025,0.975))
confidenceinterval
```

```
##     2.5%    97.5%
## 15497.60 22486.46
```

Hypothesis Testing

The average capital gains value within a a sample of this population is assumed to be greater than \$20000 USD. Is this supported by the data?

H_0 : The average capital gains amount for an individual equal to or above the age of 25 flagged as being high value is greater than or equal to \$20000.

H_a : The average capital gains amount for an individual equalt to or above the age of 25 flagged as being high value is less than \$20000.

$$H_0 : \mu >= 20000$$

$$H_a : \mu < 20000$$

To compute the p-value, we assume that the null hypothesis is true and we'll set the significance level at 5%.

```
# claimed value of the mean
omean <- 20000

# store the current sample
sample_cap_gains <- sample$cap_gain

# compute the sample mean
mean_sample <- mean(sample_cap_gains)
mean_sample
```

```
## [1] 16703.65
```

```
# store the error in estimating the average capital gains for the sample
# distance between null value and sample mean
c <- abs(omean - mean_sample)
```

```
# shifted sample
new_cap_gains <- sample_cap_gains - mean_sample + 20000

# check that the shifted data has 20000 as the mean.
mean(new_cap_gains)
```

```
## [1] 20000
```

```
# combine actual and shifted sample in one dataframe
both <- data.frame(sample_cap_gains,new_cap_gains)

# generate 1000 bootstrap samples
# from the new sample
bootstrap <- do(1000)*mean(~new_cap_gains,
              data = sample_n(both,
                              size = nrow(both),
                              replace = T))

# computing the p-value knowing that the alternative is greater
p_value <-  sum(bootstrap$mean >= mean_sample)/1000

# print the p-value:
cat("p-value is equal to:",p_value)
```

```
## p-value is equal to: 0.98
```

Since the p-value is greater than 0.05, We fail to reject the null hypothesis. The data do not provide convincing evidence at the .05 significance level that the average capital gains value is less than 20000.

Polynomial Regression

Is there an association between `education_num` and average`cap_gain` for all individuals 25 and older flagged as high value? We will set `education_num` as the predictor variable and average`cap_gain` as the response variable.

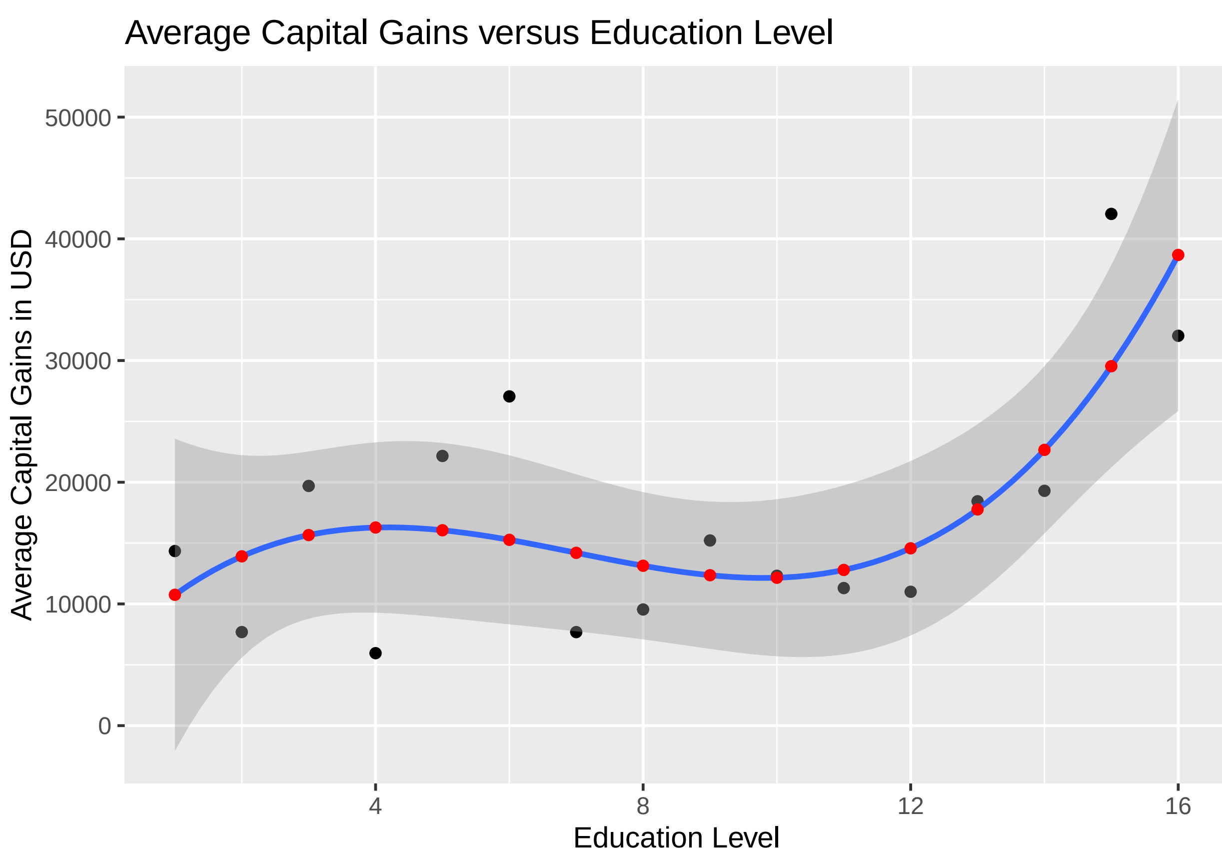
```
pop_data <- population %>%
  group_by(education_num) %>%
  summarize(avg_cap_gain = mean(cap_gain))
head(pop_data)
```

```
## # A tibble: 6 x 2
##   education_num avg_cap_gain
##           <int>        <dbl>
## 1             1       14344
## 2             2        7688
## 3             3       19695.
## 4             4        5955.
## 5             5       22158.
## 6             6       27054.
```

```
model = lm(avg_cap_gain ~ poly(education_num,3),data = pop_data)
#vector of estimates
model$coefficients;
```

```
##             (Intercept) poly(education_num, 3)1 poly(education_num, 3)2
##                17235.70                18511.79                16148.91
## poly(education_num, 3)3
##                14192.40
```

```
# model performance
ggplot(pop_data, aes(x = education_num,y = avg_cap_gain)) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ poly(x, 3)) +
  geom_point(aes(x = education_num,
                 y = model$fitted.values),
             color = "red") +
  ylab("Average Capital Gains in USD") +
  xlab("Education Level") +
  labs(title = "Average Capital Gains versus Education Level")
```



One more thing

Value Flag Prediction using an SVM Model

```
set.seed(1)

population <- df %>%
  filter(age >= 25, cap_gain > 0, hours_per_week > 0,
         score > 0, education_num > 0) %>%
  mutate_at(vars(marital_status,occupation,value_flag),list(factor))

sample <- sample(c(TRUE,FALSE),nrow(population),replace = TRUE,prob = c(0.7, 0.3))

train <- population[sample,]
test <- population[!sample,]


X_test <- population[!names(population) %in% c("value_flag")]
y_test <- population[names(population) %in% c("value_flag")]

model <- svm(value_flag ~ ., data = train)
summary(model)
```

```
##
## Call:
## svm(formula = value_flag ~ ., data = train)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1
##
## Number of Support Vectors:  1190
##
##  ( 595 595 )
##
##
## Number of Classes:  2
##
## Levels:
##  High Low
```

```
pred <- predict(model,X_test)

# accuracy of Support Vector Machine Model
mean(data.frame(pred) == y_test)
```

```
## [1] 0.8755893
```

Conclusion

This is a general overview of what risk assessment for an insurance policyholder would look like. With the data provided, we were able to select pertinent variables, both numerical and categorical, and identify the various associations between the variables which would contribute to the predictions of a policyholder being high or low value. Graphs were made to further analyze patterns. Bootstrap hypothesis testing was used to look at sample distributions of capital gains and affirm it's connection to value flagging. The association between level of education and capital gains also was explored in order to ascertain their relative weights within a possible prediction model. In the end, based on the nature of the variables provided and the associations found, a support vector machine model was used to predict whether a potential customer would be high or low value