# ST447 Project

Candidate ID: 47565

December 2, 2022

## 1 Introduction

The aim of the project is to determine which driving centre will be best suited for our friend to pass the driving test. We will be applying certain statistical techniques to determine the optimal centre for our friend to take the test.

## 2 Setting Coding Environment

We begin our analysis by loading up the relevant libraries in RStudio which will be required for doing the analysis. The required libraries can be loaded by using the following R Code.

```
library(readODS) # loading relevant libraries
library(ggplot2)
library(InformationValue)
library(pROC)
```

## 3 Friend's Profile

We load the profile of our friend by using the R Script, 'XYZprofile.R' and passing our ID in the XYZprofile.

```
source('XYZprofile.R')
XYZprofile(ID)
```

The profile of XYZ:
**Age: 18**
**Gender: Male**
**Home address: St. Albans**

## 4 Gathering Data

Since, we need to determine the optimal driving centre out of St. Albans and Wood Green (London) so we only fetch data corresponding to these locations.

### 4.1 Collecting St. Albans Data

Since, we want to get data corresponding to St. Albans from all the years so we start off by creating an intial dataframe (a table like structure that stores relevant information for our analysis) for the year 2021 corresponding to sheet 2 and then we will keep on adding new dataframes to the intial one to get the final dataframe which contains data from 2007-2021. We use the subset of this dataframe to do our initial analysis and use the entire dataframe to build the logistic regression model.

```
data <- read_ods("dvsa1203.ods",sheet=2) # reading data corresponding to sheet 2
k <- which(data[,1]=='St Albans') # finding index corresponding to St Albans
df1<- data[(k+1):(k+9),2:11] # finding relevant rows and columns
colnames(df1)<-c('Age','Male_Conducted','Male_Passes','Male_Pass_Rate','Female_Conducted',
'Female_Passes','Female_Pass_Rate','Total_Conducted','Total_Passes','Total_Pass_Rate') # renaming
    the columns

for (i in (3:16)){ # reading the data corresponding to sheets 3:16
  data <- read_ods("dvsa1203.ods",sheet=i)
```

```
  k <- which(data[,1]=='St Albans')
  if (i<=8){
  df2<- data[(k+1):(k+9),2:11]
  colnames(df2)<-c('Age','Male_Conducted','Male_Passes','Male_Pass_Rate','Female_Conducted',
  'Female_Passes','Female_Pass_Rate','Total_Conducted','Total_Passes','Total_Pass_Rate')
  df1<-rbind(df1,df2)
  }
  else if (i>8){
    df2<- data[(k+1):(k+9),-c(1,6,10)]
  colnames(df2)<-c('Age','Male_Conducted','Male_Passes','Male_Pass_Rate','Female_Conducted',
  'Female_Passes','Female_Pass_Rate','Total_Conducted','Total_Passes','Total_Pass_Rate')
  df1<-rbind(df1,df2) # performing row bind operation
  }
}

df3 <- df1[df1$Age==18,] # subsetting the data corresponding to age of our friend
df3[names(df3)]<-sapply(df3[names(df3)],as.numeric) # converting the datatype to numeric
rownames(df3)<-NULL # resetting the rownames of our dataframe
```

## 4.2 Collecting Wood Green (London)/Wood Green Data

We can collect the data for Wood Green using a similar approach as mentioned above so that it has information on all 18 year old male and female candidates for the period 2007-21.

```
data <- read_ods("dvsa1203.ods",sheet=2)
j <- which(data[,1]=='Wood Green (London)')
df4 <- data[(j+1):(j+9),2:11]
colnames(df4)<-c('Age','Male_Conducted','Male_Passes','Male_Pass_Rate','Female_Conducted',
'Female_Passes','Female_Pass_Rate','Total_Conducted','Total_Passes','Total_Pass_Rate')

for (i in (3:16)){
  data <- read_ods("dvsa1203.ods",sheet=i)
  l <- which(data[,1]=='Wood Green (London)'| data[,1]=='Wood Green')
  if (i<=8){
  df5<- data[(l+1):(l+9),2:11]
  colnames(df5)<-c('Age','Male_Conducted','Male_Passes','Male_Pass_Rate','Female_Conducted',
  'Female_Passes','Female_Pass_Rate','Total_Conducted','Total_Passes','Total_Pass_Rate')
  df4<-rbind(df4,df5)
  }
  else if (i>8){
    df5<- data[(l+1):(l+9),-c(1,6,10)]
  colnames(df5)<-c('Age','Male_Conducted','Male_Passes','Male_Pass_Rate','Female_Conducted',
  'Female_Passes','Female_Pass_Rate','Total_Conducted','Total_Passes','Total_Pass_Rate')
  df4<-rbind(df4,df5)
  }
}

df6<-df4[df4$Age==18,]
df6[names(df6)]<-sapply(df6[names(df6)],as.numeric)
rownames(df6)<-NULL
```

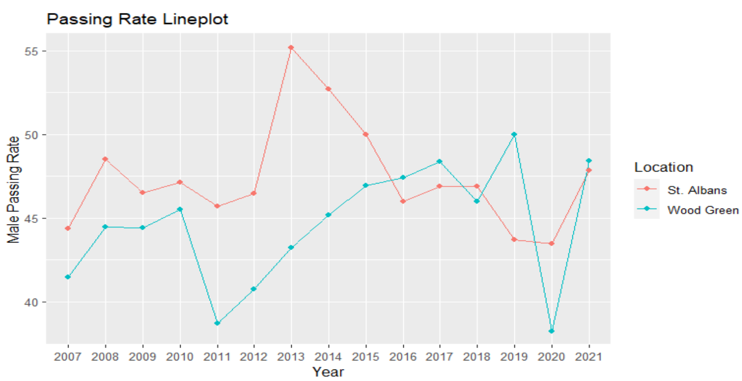
| | Age | Male_Conducted | Male_Passes | Male_Pass_Rate | Female_Conducted | Female_Passes | Female_Pass_Rate | Total_Conducted | Total_Passes | Total_Pass_Rate |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 18 | 696 | 333 | 47.84483 | 713 | 352 | 49.36886 | 1409 | 685 | 48.61604 |
| 2 | 18 | 191 | 83 | 43.45550 | 194 | 92 | 47.42268 | 385 | 175 | 45.45455 |
| 3 | 18 | 508 | 222 | 43.70079 | 532 | 236 | 44.36090 | 1040 | 458 | 44.03846 |

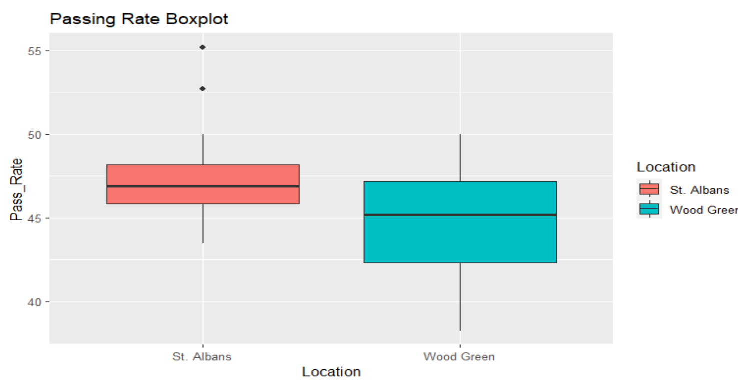
Above figure represents the format of the final dataframe that we will be using for our analysis purpose.

# 5 Plotting Passing Rate in St. Albans and Wood Green

```
Year<-rep(seq(2021,2007,-1),2)
Location<-rep(c('Wood Green','St. Albans'),each=15)
Pass_Rate<-c(df6[,4],df3[,4])

```

```
Pass_DF<-data.frame(Year,Location,Pass_Rate) # creating dataframe for plotting
# Lineplot
ggplot(Pass_DF,aes(x=factor(Year),y=Pass_Rate,group=Location))+
  geom_line(aes(color=Location))+
  geom_point(aes(color=Location))+
  ggtitle('Passing Rate Lineplot')+
  xlab('Year')+
  ylab('Male Passing Rate')
# Boxplot
ggplot(Pass_DF, aes(x=Location, y=Pass_Rate, fill=Location)) +
  geom_boxplot()+
  ggtitle('Passing Rate Boxplot')
```




We can see that the passing rate is higher in St. Albans as compared to Wood Green till the year 2015 through the lineplot. Also, through the boxplot we can see that the median passing rate is higher in St. Albans as compared to London. Through our initial exploratory data analysis it seems that taking a test in St. Albans is more preferable over taking a test in Wood Green but we need to perform statistical analysis to confirm the same.

# 6 Statistical Analysis

We assume $X_i$ as a random variable denoting an 18 year old male candidate taking the test in the specified location. Clearly, $X_i$ can take up two values: 0 (fails the driving test) and 1(passes the driving test). Then, $X_i$ can be modeled as a Bernoulli Random Variable with probability of success i.e. passing the driving test as $p$. By central limit theorem, we can estimate the expected passing probability as:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbf{X}_i \approx p = \frac{(\sum_{i=2007}^{2021}\textbf{Passing Candidates})}{(\sum_{i=2007}^{2021}\textbf{Total Candidates})}$$

Also, the above approximation converges asymptotically to normal distribution N(0,1). Hence, we can construct the confidence interval using the standard error:

$$SE(p) = \sqrt{\frac{p(1-p)}{n}}$$

Our 95% Confidence Interval can be constructed using the below mentioned formula:

$$(p - 1.96SE(p), p + 1.96SE(p))$$

## 6.1 Calculating Expected Passing Rate in St. Albans

We calculate the expected passing rate in St. Albans by using the aforementioned approach.

```
count<-sum(df3[,2]) # calculating the total number of tests conducted
pass_count<-sum(df3[,3]) # calculating the total number of passing candidates
pass_prob<-pass_count/count # calculating the expected probability
cat('Expected Passing Rate in St. Albans: ',pass_prob*100,'%','\n')
cat('95 % Confidence Interval for Expected Passing Rate in St. Albans: (',round(pass_prob-1.96*se_
   pass_prob,4)*100,'%',',',round(pass_prob+1.96*se_pass_prob,4)*100,'%',')')
```

```
Expected Passing Rate in St. Albans:  47.45672 %
95 % Confidence Interval for Expected Passing Rate in St. Albans: ( 46.32 % , 48.59 % )
```

### 6.2 Calculating Expected Passing Rate in London

We calculate the expected passing rate in London in the similar manner as used for St. Albans.

```
count<-sum(df3[,2])
pass_count<-sum(df3[,3])
pass_prob<-pass_count/count
cat('Expected Passing Rate in St. Albans: ',pass_prob*100,'%','\n')
cat('95 % Confidence Interval for Expected Passing Rate in London: (',round(pass_prob_london-1.96*se_pass_prob_london,4)*100,'%',',',round(pass_prob_london+1.96*se_pass_prob_london,4)*100,'%',')')
```

```
Expected Passing Rate in London:  45.20918 %
95 % Confidence Interval for Expected Passing Rate in London: ( 43.75 % , 46.67 % )
```

### 6.3 Testing Statistical Significance of our observation

1. We set our significance level at 5% for the permutation test.
2. $H_0$: $F_x = F_y$.
3. $H_1$: $F_x \neq F_y$.

#### 6.3.1 Performing the Permutation Test

```
set.seed(42) # for reproducibility
pass_home<-df3[,4] # creating St. Albans sample
pass_london<-df6[,4] # creating Wood Green sample
T_Obs<-abs(mean(pass_home)-mean(pass_london)) # choosing test statistic as absolute mean difference
pass<-c(pass_home,pass_london) # combining two samples
k<-0
for (i in 1:5000){
  pass_per<-sample(pass,30)
  T_Cal<-abs(mean(pass_per[1:15])-mean(pass_per[16:30]))
  if(T_Cal>T_Obs){
    k<- (k+1)
  }
}
cat('P-Value:',k/5000) # print p-value
```

```
P-Value: 0.032
```

Clearly, our p-value is less than 0.05 so we reject $H_0$. This means that there is significant difference between the means of two distributions and hence we can suggest our friend to take the test in St. Albans as the expected passing probability is higher in St. Albans as compared to Wood Green.

## 7 Fitting Logistic Regression

We build a logistic regression model using the entire data that is available to us for St. Albans and Wood Green. For building the model we consider the following variables: Age, Gender (Binary Variable), Location(Binary Variable), Response (Binary Variable) and Year. We fit the logistic regression model on Response based on other variables. Since, we have multiple predictor variables our logistic regression model can be written down as:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3+\beta_4 X_4}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3+\beta_4 X_4}}$$

or equivalently, the log odds is linear in $X_1, X_2, X_3$ and $X_4$

$$log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4$$

### 7.1 Gathering Data for St. Albans

We create two different dataframes for male and female candidates and merge them later on to get the final dataframe for St. Albans.

```r
# Creating Male St. Albans Dataset
rownames(df1)<-NULL # resetting the rownames
df1[names(df1)]<-sapply(df1[names(df1)],as.numeric) # converting all columns into numeric type
df1['Year']<-rep(seq(2021,2007,-1),each=9) # repeating sequence of year 9 times

Male_Data<-df1[,c('Age','Year','Male_Conducted','Male_Passes')] # subsetting relevant data

rows<-1:nrow(Male_Data)
Male_Data_1<-Male_Data[rep(rows,Male_Data[rows,'Male_Conducted']),] # repeating rows equivalent to test conducted
rownames(Male_Data_1)<-NULL

responses<-c() # creating empty response vector
for (i in 1:nrow(Male_Data)){
  ones<-c()
  zero<-c()
  total<-Male_Data[i,3] # value equal to total number of test conducted
  pass<-Male_Data[i,4] # value equal to total number of passed candidates

  ones<-rep(1,pass) # repeating 1 equivalent to number of pass
  zero<-rep(0,total-pass) # repeating 0 equivalent to number of total-pass

  labels<-c(ones,zero) # combining the ones and zero vector created above
  responses<-c(responses,labels) # adding label to the response to create final response
}
male_gender_label_home<-rep(1,nrow(Male_Data_1)) # creating binary repsonse for gender for males
male_age_home<-Male_Data_1$Age # creating age vector of males
male_year_home<-Male_Data_1$Year # creating year vector

home_male_data<-data.frame(Age=male_age_home,Year=male_year_home,
Gender=male_gender_label_home, Responses=responses,Location=rep(0,length(male_age_home))) # creating final dataframe for males

# Creating Female St. Albans Dataset
Female_Data<-df1[,c('Age','Year','Female_Conducted','Female_Passes')]

rows<-1:nrow(Female_Data)
Female_Data_1<-Female_Data[rep(rows,Female_Data[rows,'Female_Conducted']),]
rownames(Female_Data_1)<-NULL

responses<-c()
for (i in 1:nrow(Female_Data)){
  ones<-c()
  zero<-c()
  total<-Female_Data[i,3]
  pass<-Female_Data[i,4]

  ones<-rep(1,pass)
  zero<-rep(0,total-pass)

  labels<-c(ones,zero)
  responses<-c(responses,labels)
}

female_gender_label_home<-rep(0,nrow(Female_Data_1))
female_age_home<-Female_Data_1$Age
female_year_home<-Female_Data_1$Year

home_female_data<-data.frame(Age=female_age_home,Year=female_year_home,
Gender=female_gender_label_home, Responses=responses,Location=rep(0,length(female_age_home)))

home_data<-rbind(home_male_data,home_female_data) # combing male and female data for St. Albans
```

## 7.2 Gathering Data for Wood Green

We use the earlier approach to build the dataframe for Wood Green.

```
# Creating Male Wood Green Dataset
rownames(df4)<-NULL
df4[names(df4)]<-sapply(df4[names(df4)],as.numeric)
df4['Year']<-rep(seq(2021,2007,-1),each=9)

Male_Data<-df4[,c('Age','Year','Male_Conducted','Male_Passes')]

rows<-1:nrow(Male_Data)
Male_Data_1<-Male_Data[rep(rows,Male_Data[rows,'Male_Conducted']),]
rownames(Male_Data_1)<-NULL

responses<-c()
for (i in 1:nrow(Male_Data)){
  ones<-c()
  zero<-c()
  total<-Male_Data[i,3]
  pass<-Male_Data[i,4]

  ones<-rep(1,pass)
  zero<-rep(0,total-pass)

  labels<-c(ones,zero)
  responses<-c(responses,labels)
}

male_gender_label_london<-rep(1,nrow(Male_Data_1))
male_age_london<-Male_Data_1$Age
male_year_london<-Male_Data_1$Year
london_male_data<-data.frame(Age=male_age_london,Year=male_year_london,Gender=male_gender_label_
    london,
Responses=responses,Location=rep(1,length(male_age_london)))

#Creating Female Wood Green Dataset
Female_Data<-df4[,c('Age','Year','Female_Conducted','Female_Passes')]

rows<-1:nrow(Female_Data)
Female_Data_1<-Female_Data[rep(rows,Female_Data[rows,'Female_Conducted']),]
rownames(Female_Data_1)<-NULL

responses<-c()
for (i in 1:nrow(Female_Data)){
  ones<-c()
  zero<-c()
  total<-Female_Data[i,3]
  pass<-Female_Data[i,4]

  ones<-rep(1,pass)
  zero<-rep(0,total-pass)

  labels<-c(ones,zero)
  responses<-c(responses,labels)
}

female_gender_label_london<-rep(0,nrow(Female_Data_1))
female_age_london<-Female_Data_1$Age
female_year_london<-Female_Data_1$Year
london_female_data<-data.frame(Age=female_age_london,Year=female_year_london,Gender=female_gender_
    label_london,
Responses=responses,Location=rep(1,length(female_age_london)))

london_data<-rbind(london_male_data,london_female_data)
```

## 7.3 Combining Data from St. Albans and Wood Green

```r
mod_data<-rbind(home_data,london_data) # creating final dataframe for logistic model
```

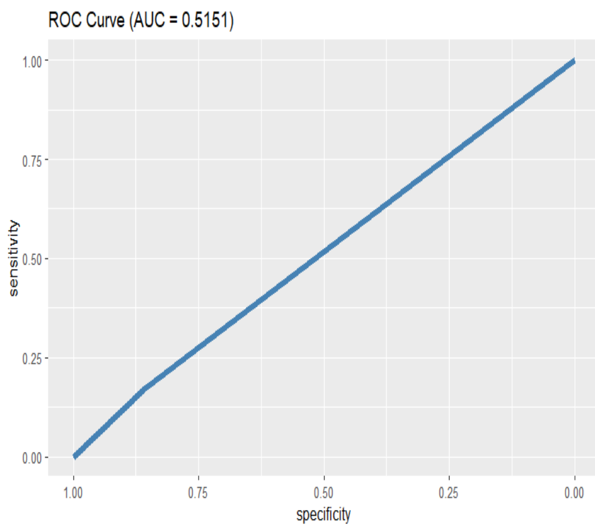
## 7.4 Building Logistic Regression Model

```r
set.seed(1) # setting seed for reproducibility
samples <- sample(c(TRUE, FALSE), nrow(mod_data), replace=TRUE, prob=c(0.7,0.3)) # creating samples for splitting data into training and testing sets
train <- mod_data[samples, ] # creating training dataset
test <- mod_data[!samples, ]  # creating testing dataset

log_mod<-glm(Responses~.,family=binomial,data=train) # fitting logistic resgression model

summary(log_mod) # printing summary for logistic regression model

X_test<-test[,-4] # creating dataframe having only features from testing dataset
y_test<-test[,4] # creating response vector from testing dataset

pred_prob<-predict(log_mod,X_test,type='response') # predicting probability for test dataset
pred<-ifelse(pred_prob>=0.5,1,0) # assigning labels based on predicted probability

res<-data.frame(Age=18,Year=2022,Gender=1,Location=c(0,1)) # creating dataframe for predicting outcome of our friend

table(factor(pred),factor(y_test)) # creating confusion matrix
cat('Accuracy:',round((17567+2876)/(17567+2876+13914+2886),4)*100,'%','\n') # printing accuracy of model

pass_values<-predict(log_mod,res,type='response',se.fit = TRUE) # predicting for our friend

cat('Predicted probability associated with St. Albans:',round(pass_values$fit[1],4)*100,'%','with a
Standard Error of:',round(pass_values$se.fit[1],4),'\n') # printing probability of our friend being in Class 1 in St. Albans
cat('Predicted probability associated with London:',round(pass_values$fit[2],4)*100,'%','with a
Standard Error of:',round(pass_values$se.fit[2],4)) # printing probability of our friend being in Class 1 in London
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)    
(Intercept) -17.024358   3.270909  -5.205 1.94e-07 ***
Age          -0.034966   0.002838 -12.319  < 2e-16 ***
Year          0.008682   0.001624   5.346 8.99e-08 ***
Gender        0.199101   0.013703  14.530  < 2e-16 ***
Location     -0.163105   0.014523 -11.231  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 119656  on 86997  degrees of freedom
Residual deviance: 118982  on 86993  degrees of freedom
AIC: 118992

Number of Fisher Scoring iterations: 4


        0     1
  0 17567 13914
  1  2886  2876
Accuracy: 54.89 %
Predicted Probability by Model associated with St. Albans: 52.53 % with a Standard
Error of: 0.0044
Predicted Probability by Model associated with London: 48.45 % with a Standard Error
of: 0.0049
```

We can clearly see that the predicted passing probability by the model for our friend in St. Albans is 52.53% while the predicted passing probability associated with London is 48.45%. Thus, we suggest our friend to take the test in St. Albans based on our Logistic Regression Model.

```
roc_mod<-roc(y_test,pred)
auc<-round(auc(y_test,pred),4)

ggroc(roc_mod, colour = 'steelblue', size = 2) +
  ggtitle(paste0('ROC Curve ', '(AUC = ', auc, ')'))
```




# 8 Conclusion

We can clearly see that St. Albans is the optimal driving centre as per initial exploratory data analysis, statistical analysis and logistic regression model. Hence, we can suggest our friend to take the test in St. Albans over London.

# 9 Strengths

1. We get the same result via initial exploratory data analysis, statistical calculations and logistic regression model. Hence, we can be more confident in giving suggestion to our friend.

2. Since, we have built a Logistic Regression Model so we can predict the chances of passing the test in two locations in near future as well.

# 10 Weakness

1. We do not check for the normal distribution of the data while building Logistic Regression Model.

2. Accuracy for our Logistic Regression Model is not high.

# 11 Suggestions

1. Our Logistic Regression Model can be improved by having more variables.