```r
#College Admission
# get working Directory
getwd()

# to import the csv file
library(rio)

library(tidyverse)

# import the dataset
data<-read.csv("College_admission.csv")

# view the dataset
view(data)

#1. Find the missing values. (if any, perform missing value
treatment)
is.na(data)

#Finding structure of dataset

str(data)

#Factoring
f<-factor(c(data$gpa))
as.numeric(f)
View(data)

#2. Find outliers (if any, then perform outlier treatment)
hist(data$gre,xlab = "gre",main = "Histogram of gre",breaks =
sqrt(nrow(data)))

#or using ggplot
library(ggplot2)
ggplot(data) + aes(x=gre) +
geom_histogram(bins=30L,fill="red")+ theme_minimal()

#Boxplots also useful to detect potential outliers
boxplot(data$ses,ylab="ses")
boxplot(data$admit,ylab="admit")

#3. Find the structure of the data set and if required,
# transform the numeric data type to factor and vice-versa.
#To extract exact values of outliers
```

```
boxplot.stats(data$gre)$out

#To extract row number corresponding to outliers

out <- boxplot.stats(data$gre)$out
out_ind <- which(data$gre %in% c(out))
out_ind

#4. Find whether the data is normally distributed or not.
# Use the plot to determine the same.

#Variables for this outliers
data[out_ind,]


shapiro.test(data$gre)


#5. Normalize the data if not normally distributed.

library(caret)
da<-as.data.frame(scale(data[,2]))
summary(data$gre)


#6. Use variable reduction techniques to identify significant
variables.
library(olsrr)

model <-lm(admit~ gre + gpa + ses + Gender_Male + Race +
rank,data = data)
ols_step_all_possible(model)

#7. Run logistic model to determine the factors that influence
the admission
# process of a student (Drop insignificant variables)

head(data)
summary(data)
sapply(data, sd)

data_logit <-glm(admit~gre + gpa+rank ,data=data,family =
"binomial")

#8. Calculate the accuracy of the model and run validation
#techniques.
```

```
library(ROCR)
library(Metrics)

library(caret)
split<-createDataPartition(y=data$admit,p=0.6,list = FALSE)
new_train <- data[split]
new_test <- data[split]

log_predict<-predict(data_logit,newdata=data,type="response")
log_predict<-ifelse(log_predict>0.5,1,0)
pr<-prediction(log_predict,data$admit)
perf<-performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf)
auc(data$admit,log_predict)

#9.  Try other modelling techniques like decision tree and SVM
and select a champion model
library(rpart)
library(rpart.plot)
fit<-rpart(admit~.,data=data,method='class')
rpart.plot(fit,extra=106)

#10. Determine the accuracy rates for each kind of model

#Confusion matrix
pu<-predict(fit,data,type='class')
tm<-table(data$admit,pu)
tm
```