DAV Practical

Q1. EDA of Income data set(linear regression)

```
#data loading
data <- read.csv("E:/DAV practical datasets/income.csv")

#Conditional scatterplot
library(lattice)
splom(~data[c(2:5)], groups=NULL, data=data,
      axis.line.tck = 0, axis.text.alpha = 0)

library(ggplot2)
# Visualize the relationship between income and education
ggplot(data, aes(x = Education, y = Income)) +
  geom_point() +
  labs(title = "Income by Education",
       x = "Education",
       y = "Income")

# Visualize the relationship between income and age
ggplot(data, aes(x = Age, y = Income)) +
  geom_point() +
  labs(title = "Income by Age",
       x = "Age",
       y = "Income")

# Visualize the relationship between income and Gender
ggplot(data, aes(x = Gender, y = Income)) +
  geom_point() +
  labs(title = "Income by Gender",
       x = "Gender",
       y = "Income")

#Pearson's correlation
cor.test(data$Income, data$Age)
cor.test(data$Income, data$Education)
cor.test(data$Income, data$Gender)

#Data split
library(caret)
training.samples <- data$Income %>%
  createDataPartition(p=0.7, list=FALSE)
```

```
train_data <- data[training.samples, ]
test_data <- data[-training.samples, ]

#Fitting the model
model <- lm(Income~Age, data=train_data)
summary(model)

#predictions
predictions <- model %>% predict(test_data)
plot(test_data$Income, predictions)
abline(lm(predictions~Income, data=test_data))
```

Q2. EDA of Fish dataset (linear regression)

```
#data loading
data <- read.csv("E:/DAV practical datasets/Fish.csv")

#Conditional scatterplot
library(lattice)
splom(~data[c(2:7)], groups=NULL, data=data,
      axis.line.tck=0, axis.text.alpha=0)

library(ggplot2)
# Visualize the relationship between Weight and Length1
ggplot(data, aes(x = Length1, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Length1",
       y = "Weight")

# Visualize the relationship between Weight and Length2
ggplot(data, aes(x = Length2, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Length2",
       y = "Weight")

# Visualize the relationship between Weight and Length3
ggplot(data, aes(x = Length3, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Length3",
       y = "Weight")
```

```
# Visualize the relationship between Weight and Height
ggplot(data, aes(x = Height, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Height",
       y = "Weight")

# Visualize the relationship between Weight and Length1
ggplot(data, aes(x = Width, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Width",
       y = "Weight")

#Pearson's correlation
cor.test(data$Weight, data$Length1)
cor.test(data$Weight, data$Length2)
cor.test(data$Weight, data$Length3)
cor.test(data$Weight, data$Height)
cor.test(data$Weight, data$Width)

#Data split
library(caret)
training.samples <- data$Weight %>%
  createDataPartition(p=0.7, list=FALSE)

train_data <- data[training.samples, ]
test_data <- data[-training.samples, ]

#Fitting the model
model <- lm(Weight~Length3, data=train_data)
summary(model)

#predictions
predictions <- model %>% predict(test_data)
plot(test_data$Weight, predictions)
abline(lm(predictions~Weight, data=test_data))
```

Q5. EDA of Income dataset (multiple regression)

```
#data loading
data <- read.csv("E:/DAV practical datasets/income.csv")

#Conditional scatterplot
```

```
library(lattice)
splom(~data[c(2:5)], groups=NULL, data=data,
      axis.line.tck = 0, axis.text.alpha = 0)

library(ggplot2)
# Visualize the relationship between income and education
ggplot(data, aes(x = Education, y = Income)) +
  geom_point() +
  labs(title = "Income by Education",
       x = "Education",
       y = "Income")

# Visualize the relationship between income and age
ggplot(data, aes(x = Age, y = Income)) +
  geom_point() +
  labs(title = "Income by Age",
       x = "Age",
       y = "Income")

# Visualize the relationship between income and Gender
ggplot(data, aes(x = Gender, y = Income)) +
  geom_point() +
  labs(title = "Income by Gender",
       x = "Gender",
       y = "Income")

#Pearson's correlation
cor.test(data$Income, data$Age)
cor.test(data$Income, data$Education)
cor.test(data$Income, data$Gender)

#Data split
library(caret)
training.samples <- data$Income %>%
  createDataPartition(p=0.7, list=FALSE)

train_data <- data[training.samples, ]
test_data <- data[-training.samples, ]

#Fitting the model
model <- lm(Income~Age+Education, data=train_data)
summary(model)

#predictions
```

```
predictions <- model %>% predict(test_data)
plot(test_data$Income, predictions)
abline(lm(predictions~Income, data=test_data))
```

Q6. EDA of Fish dataset (Multiple regression)

```
#data loading
data <- read.csv("E:/DAV practical datasets/Fish.csv")

#Conditional scatterplot
library(lattice)
splom(~data[c(2:7)], groups=NULL, data=data,
      axis.line.tck=0, axis.text.alpha=0)

library(ggplot2)
# Visualize the relationship between Weight and Length1
ggplot(data, aes(x = Length1, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Length1",
       y = "Weight")

# Visualize the relationship between Weight and Length2
ggplot(data, aes(x = Length2, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Length2",
       y = "Weight")

# Visualize the relationship between Weight and Length3
ggplot(data, aes(x = Length3, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Length3",
       y = "Weight")

# Visualize the relationship between Weight and Height
ggplot(data, aes(x = Height, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Height",
       y = "Weight")

# Visualize the relationship between Weight and Length1
```

```
ggplot(data, aes(x = Width, y = Weight)) +
  geom_point() +
  labs(title = "Weight",
       x = "Width",
       y = "Weight")

#Pearson's correlation
cor.test(data$Weight, data$Length1)
cor.test(data$Weight, data$Length2)
cor.test(data$Weight, data$Length3)
cor.test(data$Weight, data$Height)
cor.test(data$Weight, data$Width)

#Data split
library(caret)
training.samples <- data$Weight %>%
  createDataPartition(p=0.7, list=FALSE)

train_data <- data[training.samples, ]
test_data <- data[-training.samples, ]

#Fitting the model
model <- lm(Weight~Length1+Length2+Length3+Height+Width, data=train_data)
summary(model)

#predictions
predictions <- model %>% predict(test_data)
plot(test_data$Weight, predictions)
abline(lm(predictions~Weight, data=test_data))
```

Q9. Time series analysis (Gas Production)

```
install.packages("forecast")       # install, if necessary
library(forecast)

# read in gasoline production time series
# monthly gas production expressed in millions of barrels
gas_prod_input <- as.data.frame( read.csv("E:/DAV practical datasets/gas_prod.csv") )

# create a time series object
gas_prod <- ts(gas_prod_input[,2])

#examine the time series
plot(gas_prod, xlab = "Time (months)",
```

```
    ylab = "Gasoline production (millions of barrels)")

# check for conditions of a stationary time series
plot(diff(gas_prod))
abline(a=0, b=0)

# examine ACF and PACF of differenced series
acf(diff(gas_prod), xaxp = c(0, 48, 4), lag.max=48, main="")
pacf(diff(gas_prod), xaxp = c(0, 48, 4), lag.max=48, main="")

# fit a (0,1,0)x(1,0,0)12 ARIMA model
arima_1 <- arima (gas_prod,
              order=c(0,1,0),
              seasonal = list(order=c(1,0,0),period=12))
arima_1

# it may be necessary to calculate AICc and BIC
# http://stats.stackexchange.com/questions/76761/extract-bic-and-aicc-from-arima-object
AIC(arima_1,k = log(length(gas_prod)))   #BIC


# examine ACF and PACF of the (0,1,0)x(1,0,0)12 residuals
acf(arima_1$residuals, xaxp = c(0, 48, 4), lag.max=48, main="")
pacf(arima_1$residuals, xaxp = c(0, 48, 4), lag.max=48, main="")

# fit a (0,1,1)x(1,0,0)12 ARIMA model
arima_2 <- arima (gas_prod,
              order=c(0,1,1),
              seasonal = list(order=c(1,0,0),period=12))
arima_2

# it may be necessary to calculate AICc and BIC
# http://stats.stackexchange.com/questions/76761/extract-bic-and-aicc-from-arima-object
AIC(arima_2,k = log(length(gas_prod)))   #BIC

# examine ACF and PACF of the (0,1,1)x(1,0,0)12 residuals
acf(arima_2$residuals, xaxp = c(0, 48, 4), lag.max=48, main="")
pacf(arima_2$residuals, xaxp = c(0, 48,4), lag.max=48, main="")

# Normality and Constant Variance

plot(arima_2$residuals, ylab = "Residuals")
abline(a=0, b=0)
```

```
hist(arima_2$residuals, xlab="Residuals", xlim=c(-20,20))

qqnorm(arima_2$residuals, main="")
qqline(arima_2$residuals)

# Forecasting

#predict the next 12 months
arima_2.predict <- predict(arima_2,n.ahead=12)
matrix(c(arima_2.predict$pred-1.96*arima_2.predict$se,
     arima_2.predict$pred,
     arima_2.predict$pred+1.96*arima_2.predict$se), 12,3,
    dimnames=list( c(241:252) ,c("LB","Pred","UB")) )

plot(gas_prod, xlim=c(145,252),
   xlab = "Time (months)",
   ylab = "Gasoline production (millions of barrels)",
   ylim=c(360,440))
lines(arima_2.predict$pred)
lines(arima_2.predict$pred+1.96*arima_2.predict$se, col=4, lty=2)
lines(arima_2.predict$pred-1.96*arima_2.predict$se, col=4, lty=2)
```

Q10. Time series analysis (Electricity Production)

```
install.packages("forecast")
library(forecast)

ele_prod_input <- as.data.frame(read.csv("E:/DAV practical datasets/Electric_Production.csv"))

ele_prod <- ts(ele_prod_input[,2])

plot(ele_prod, xlab = "Time", ylab = "Electricity production")

plot(diff(ele_prod))
abline(a=0,b=0)

acf(diff(ele_prod), xaxp = c(0,48,4), lag.max = 48, main="")
pacf(diff(ele_prod), xaxp = c(0,48,4), lag.max = 48, main="")

arima_1 <- arima(ele_prod, order = c(0,1,0), seasonal = list(order=c(1,0,0), period=12))
arima_1

AIC(arima_1,k = log(length(ele_prod)))
```

```
acf(arima_1$residuals, xaxp = c(0,48,4), lag.max = 48, main="")
pacf(arima_1$residuals, xaxp = c(0,48,4), lag.max = 48, main="")

arima_2 <- arima(ele_prod, order = c(0,1,1), seasonal = list(order=c(1,0,0),period=12))
arima_2

acf(arima_2$residuals, xaxp = c(0,48,4), lag.max = 48, main="")
pacf(arima_2$residuals, xaxp = c(0,48,4), lag.max = 48, main="")

plot(arima_2$residuals, ylab = "Residuals")
abline(a=0,b=0)

hist(arima_2$residuals, xlab = "Residuals", xlim = c(-20,20))

qqnorm(arima_2$residuals, main="")
qqline(arima_2$residuals)

#predict the next 12 months
arima_2.predict <- predict(arima_2, n.ahead = 12)
matrix(c(arima_2.predict$pred-1.96*arima_2.predict$se,
         arima_2.predict$pred,
         arima_2.predict$pred+1.96*arima_2.predict$se),12,3,
       dimnames = list(c(241:252), c("LB","Pred","UB")))

plot(ele_prod, xlim = c(145,252), xlab = "Time", ylab = "Electricity Production",
     ylim = c(360,440))

lines(arima_2.predict$pred)
lines(arima_2.predict$pred-1.96*arima_2.predict$se, col=4, lty=2)
lines(arima_2.predict$pred+1.96*arima_2.predict$se, col=4, lty=2)
```

Q13. Visualization for iris dataset

```
# Load the necessary library
library(ggplot2)
View(iris)
# Plot histogram for values of Sepal.Length
ggplot(iris, aes(x = Sepal.Length)) +
  geom_histogram(binwidth = 0.2, fill = "blue") +
  labs(title = "Histogram for Sepal Length", x = "Sepal Length", y = "Frequency")

# Plot scatterplot of Sepal.Width vs Sepal.Length
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(aes(color = Species), size = 3) +
```

```
  labs(title = "Scatterplot of Sepal Width vs Sepal Length", x = "Sepal Length", y = "Sepal Width")

# Plot boxplot for Sepal.Length by Species
ggplot(iris, aes(x = Species, y = Sepal.Length)) +
  geom_boxplot(aes(fill = Species)) +
  labs(title = "Boxplot for Sepal Length by Species", x = "Species", y = "Sepal Length")

# Plot scatter plot matrix for all variables coloured by Species
pairs(iris[,1:4], main = "Scatterplot Matrix for Iris Dataset", pch = 21,
      bg = c("red", "green3", "blue")[unclass(iris$Species)])
```

Q14. Visualization for diamonds dataset

```
library(ggplot2)
View(diamonds)
str(diamonds)
head(diamonds)
summary(diamonds)
ggplot(data=diamonds) + geom_histogram(binwidth=500, aes(x=price))

ggplot(diamonds, aes(carat, price, col = clarity)) +
  geom_point()

ggplot(diamonds, aes(x = carat, y = price, color = cut)) +
  geom_point() +
  labs(x = "Carat", y = "Price", title = "Price vs. Carat with Cut as Color")

ggplot(diamonds, aes(x = cut, y = price)) +
  geom_boxplot() +
  labs(x = "Cut", y = "Price", title = "Boxplot of Prices Grouped by Cut") +
  theme_minimal()

ggplot(diamonds, aes(price, fill = clarity)) +
  geom_density()
```

Q15. Visualization for mtcars dataset

```
# Load the mtcars dataset
data(mtcars)
View(mtcars)
# View the structure of the dataset
str(mtcars)

# Summary statistics of the dataset
```

```r
summary(mtcars)

# Load required libraries
library(ggplot2)

# Plot dot chart grouped by cylinder
ggplot(mtcars, aes(x = factor(cyl), y = mpg, color = factor(cyl))) +
  geom_point() +
  labs(x = "Cylinder", y = "Miles Per Gallon", title = "Dot Chart of MPG by Cylinder")

# Plot bar plot for Distribution of Car Cylinder Counts
ggplot(mtcars, aes(x = factor(cyl))) +
  geom_bar() +
  labs(x = "Cylinder", y = "Count", title = "Distribution of Car Cylinder Counts")

# Plot bar plot for Distribution of Car Gears
ggplot(mtcars, aes(x = factor(gear))) +
  geom_bar(fill = "darkgreen") +
  labs(x = "Gear", y = "Count", title = "Distribution of Car Gears")

ggplot(mtcars, aes(y = mpg, x = 1)) +
  geom_boxplot(fill = "orange") +
  labs(title = "Boxplot of MPG Values", x = "", y = "Miles per Gallon")

# Create histogram of values for mpg
ggplot(mtcars, aes(x = mpg)) +
  geom_histogram(binwidth = 2) +
  labs(title = "Histogram of MPG Values", x = "Miles per Gallon", y = "Frequency")
```