8)a	9) c logistical regre
data <- c(4, 7, 8, 9, 7, 10, 5, 7, 6, 8) mean_value <- mean(data) cat("Mean:", mean_value, "\n") median_value <- median(data) cat("Median:", median_value, "\n") mode_value <- as.numeric(names(sort(table(data), decreasing = TRUE)[1])) cat("Mode:", mode_value, "\n")	age <- c(20, 30, 40, 50, 60) buy <- c(0, 1, 0, 1, 1) model <- glm(buy ~ age, family = binomial) summary(model) predicted_probs <- predict(model, newdata = data.frame(age = c(25, 35)), type = "response") print(predicted_probs) age_seq <- seq(min(age), max(age), length.out = 100) fitted_probs <- predict(model, newdata = data.frame(age = age_seq), type = "response") plot(age, buy, main = "Logistic Regression: Age vs. Buying", xlab = "Age", ylab = "Buying Probability", pch = 16, col = "blue", ylim = c(0, 1)) lines(age_seq, fitted_probs, col = "red", lwd = 2)
8)b input <- mtcars[,c("am","mpg","hp")] print(head(input)) input <- mtcars result <- aov(mpg~hp*am,data = input) print(summary(result)) result <- aov(mpg~hp+am,data = input) print(summary(result)) result1 <- aov(mpg~hp*am,data = input) result2 <- aov(mpg~hp+am,data = input) print(anova(result1,result2))	9)d poisiion regression input <- warpbreaks print(head(input)) output <-glm(formula = breaks ~ wool+tension, data = warpbreaks, family = poisson) print(summary(output))
9)a Linear regression height <- c(150, 160, 170, 180, 190) weight <- c(50, 60, 70, 80, 90) model <- lm(weight ~ height) plot(height, weight, main = "Height vs. Weight",xlab = "Height (cm)", ylab = "Weight (kg)", pch = 16) abline(model, col = "red")	10)a time series snowfall <- c(790,1170.8,860.1,1330.6,630.4,9 11.5,683.5,996.6,783.2,982,881.8,1021) snowfall_timeseries<- ts(snowfall,start = c(2013,1),frequency = 12) print(snowfall_timeseries) png(file = "snowfall.png") plot(snowfall_timeseries) dev.off()
9)b Multiple regresssion age <- c(25, 30, 35, 40, 45) height_multi <- c(150, 160, 170, 180, 190) weight_multi <- c(50, 60, 65, 75, 85) multiple_model <- lm(weight_multi ~ height_multi + age) summary(multiple_model) library(ggplot2) ggplot(data = data.frame(weight_multi, fitted = fitted(multiple_model)), aes(x = fitted, y = weight_multi)) +geom_point() +geom_smooth(method = "lm", color = "blue", se = FALSE) +ggtitle("Multiple Regression: Residuals vs Fitted Values") + theme_minimal()	10)b non linear leat square xvalues <- c(1.6, 2.1, 2, 2.23.19, 2.21) yvalues <- c(5.19, 7.43, 6.94, 8.11, 18) png(file = "nls.png") plot(xvalues, yvalues) model <- nls(yvalues ~ b1*xvalues^2 + b2, start = list(b1 = 1, b2 = 3)) lines(seq(min(xvalues), max(xvalues), length.out = 100), predict(model, newdata = data.frame(xvalues = seq(min(xvalues), max(xvalues), length.out = 100)))) dev.off() print(sum(resid(model)^2)) print(confint(model))

10) Decision tree

```
data("iris")
install.packages("caret")
install.packages("C50")
library(caret)
library(C50)
set.seed(7)
inTraininglocal <- createDataPartition(iris$Species, p = .70,
list = FALSE)
training <- iris[inTraininglocal, ]
testing <- iris[-inTraininglocal, ]
model <- C5.0(Species ~ ., data = training)
summary(model)
pred <- predict(model, testing[, -5])  # use predict() instead of
predict.C5.0()
a <- table(testing$Species, pred)
accuracy <- sum(diag(a)) / sum(a)
print(accuracy)
plot(model)
```

```
library("MASS")
print(str(Cars93))
car_data<-
data.frame(Cars93$AirBags,
Cars93$Type)
car_data = table(Cars93$AirBags,
Cars93$Type)
print(car_data)
print(chisq.test(car_data))
```

12)b t test

```
x  <- c(0.593, 0.142, 0.329, 0.691, 0.231,
0.793,  0.519, 0.392, 0.418)
t.test(x, alternative="greater", mu=0.3)
```

11) Normal Dist

dnorm

```
x <- seq(-10, 10, by = .1)
y <- dnorm(x, mean = 2.5, sd = 0.5)
png(file = "dnorm.png")
plot(x,y)
dev.off()
```

pnorm

```
x <- seq(-10,10,by = .2)
y <- pnorm(x, mean = 2.5, sd = 2)
png(file = "pnorm.png")
plot(x,y)
dev.off()
```

qnorm

```
x <- seq(-10,10,by = .2)
y <- pnorm(x, mean = 2.5, sd = 2)
png(file = "pnorm.png")
plot(x,y)
dev.off()
```

rnorm

```
y <- rnorm(50)
png(file = "rnorm.png")
hist(y, main = "Normal DIstribution")
dev.off()
```

```
x <- seq(0,50,by = 1)

y <-
dbinom(x,50,0.5)

png(file =
"dbinom.png")

plot(x,y)

dev.off()

plot(x,y)
```

12) c t test

```
install.packages("randomForest")
library(party)
print(head(readingSkills))
library(party)
library(randomForest)
output.forest <-
randomForest(nativeSpeaker ~ age +
shoeSize + score,
                    data = readingSkills)
print(output.forest)
```

Week 2 associate

Step 1 : load the data set (open weka and click on ‘Explore’ , click open file and load the contact lenses arff dataset)
Step 2 select the aprori alg (go to ‘associate’ tab
, clich ‘ choose ‘ and click weka association apriori from the list) .   step 3 configure and run apriori (click on the ‘associator’ and set minisupped and minmetric, click start to run alg)