**# Title : Use of R for Correlation and regression analysis.**

df <- read.csv(file.choose())

df

**# Linear regression in R: lm()**

year <- rep(2008:2010,each=4)

quarter <- rep(1:4, 3)

cpi <- c(162.2, 164.6, 166.5, 166, 166.2, 167, 168.6, 169.5, 171,

+ 172.1, 173.3, 174)

plot(cpi, xaxt = "n", ylab = "CPI", xlab = "")

axis(1, labels = paste(year, quarter, sep = "Q"), at = 1:12, las = 3)

**# Co-relation between CPI and Year/ Quarter**

cor(year , cpi)

cor(quarter,cpi)

**# Build a Linear regression model with function lm()**

fit <- lm(cpi ~ year + quarter)

fit

**# What will the CPI be in 2011 ?**

->cpi2011 <- fit$coefficients[[1]]+

+ fit$coefficients[[2]] \* 2011 +

+ fit$coefficients[[3]] \* (1:4)

# An Easier way is to used function predict().

->attributes(fit)

# Function residuals(): differences between observed values and Observed values.

residuals(fit)

summary(fit)

**# 3D plot of the fitted model**

library(scatterplot3d)

s3d <- scatterplot3d(year, quarter,cpi,highlight.3d = T,type = "h", lab = c(2,3) )

s3d$plane3d(fit)

Iris <- read.csv(file.choose())

head(Iris)

set.seed(1234)

cor.test(Iris$SepalLengthCm ,Iris$PetalWidthCm)

**# Iris Example --> Decision Tree**

str(Iris)

ind <- sample(2,nrow(Iris),replace = TRUE,prob = c(0.7,0.3))

train.data <- Iris[ind == 1, ]

test.data <- Iris[ind == 2, ]

**# Build a ctree**

myFormula <- Species ~ Sepal.Length + Sepal.Width + Petal.Length +

+ Petal.Width

Iris\_ctree <- ctree(myFormula, data = train.data)

table(predict(Iris\_ctree), train.data$Species)

print(Iris\_ctree)

plot(Iris\_ctree)

plot(Iris\_ctree,type="simple")

**# Predict on test data**

testPred <- predict(Iris\_ctree, newdata=test.data)

ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))

train.data <- iris[ind==1,]

test.data <- iris[ind==2,]

ind <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))

train.data <- iris[ind==1,]

test.data <- iris[ind==2,]

library(randomForest)

rf <- randomForest(Species ~ ., data=train.data, ntree=100,

+ proximity=T)

table(predict(rf), train.data$Species)

print(rf)