
#statistical Hypothesis testing

#one-sample T-testing

```
x<-rnorm(100)#sample vector
t.test(x,mu=5)#one sample test
```

OUTPUT:

```
One Sample t-test
data:  x
t = -52.268, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 5
95 percent confidence interval:
 -0.2733968  0.1123422
sample estimates:
 mean of x
-0.0805273
```

#two-sample T-testing

```
x<-rnorm(100)
y<-rnorm(100)
t.test(x,y)
```

OUTPUT:

```
Welch Two Sample t-test
data:  x and y
t = -0.056338, df = 197.95, p-value = 0.9551
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2814629  0.2658276
sample estimates:
  mean of x   mean of y
-0.04196545 -0.03414780
```

#directional Hypothesis

```
t.test(x,mu=2,alternative = 'greater')
```

OUTPUT:

```
One Sample t-test
data:  x
t = -20.982, df = 99, p-value = 1
alternative hypothesis: true mean is greater than 2
95 percent confidence interval:
 -0.2035555       Inf
sample estimates:
  mean of x
```

```
-0.04196545
```

#one-sample u-test

wilcox.test(y,exact=FALSE)

OUTPUT:

```
Wilcoxon signed rank test with continuity
        correction
data:  y
V = 2373, p-value = 0.6024
alternative hypothesis: true location is not equal to 0
```

#two-sample u-test

wilcox.test(x,y)

OUTPUT:

```
Wilcoxon rank sum test with continuity
        correction
data:  x and y
W = 4878, p-value = 0.7666
alternative hypothesis: true location shift is not equal to 0
```

#correlation Test

cor.test(matcars$mpg,matcars$hp)

OUTPUT:

```
Pearson's product-moment correlation
data:  mtcars$mpg and mtcars$hp
t = -6.7424, df = 30, p-value = 1.788e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8852686 -0.5860994
sample estimates:
       cor
-0.7761684
```

#Chi-Square Test

library(MASS)

#create Dataframes

print(str(survey))

OUTPUT:

```
'data.frame':237 obs. of  12 variables:
 $ Sex   : Factor w/ 2 levels "Female","Male": 1 2 2 2 2 1 2 1 2 2 ...
 $ Wr.Hnd: num  18.5 19.5 18 18.8 20 18 17.7 17 20 18.5 ...
```

```
 $ NW.Hnd: num  18 20.5 13.3 18.9 20 17.7 17.7 17.3 19.5 18.5 ...
 $ W.Hnd : Factor w/ 2 levels "Left","Right": 2 1 2 2 2 2 2 2 2 2 ...
 $ Fold  : Factor w/ 3 levels "L on R","Neither",..: 3 3 1 3 2 1 1 3 3 3 ...
 $ Pulse : int  92 104 87 NA 35 64 83 74 72 90 ...
 $ Clap  : Factor w/ 3 levels "Left","Neither",..: 1 1 2 2 3 3 3 3 3 3 ...
 $ Exer  : Factor w/ 3 levels "Freq","None",..: 3 2 2 2 3 3 1 1 3 3 ...
 $ Smoke : Factor w/ 4 levels "Heavy","Never",..: 2 4 3 2 2 2 2 2 2 2 ...
 $ Height: num  173 178 NA 160 165 ...
 $ M.I   : Factor w/ 2 levels "Imperial","Metric": 2 1 NA 2 2 1 1 2 2 2 ...
 $ Age   : num  18.2 17.6 16.9 20.3 23.7 ...
NULL
```

#create a data frame from the main data set

stu_data=data.frame(survey$Smoke,survey$Exer)

#create a contingency table with the needed variables

stu_data=table(survey$Smoke,survey$Exer)

print(stu_data)

OUTPUT:

```
      Freq None Some
 Heavy    7    1    3
 Never   87   18   84
 Occas   12    3    4
 Regul    9    1    7
```

- **Regression analysis test**

1. **Linear Regression**

 # Height vector

x <- c(153, 169, 140, 186, 128,

```
    136, 178, 163, 152, 133)
```

 # Weight vector

```
y <- c(64, 81, 58, 91, 47, 57,
    75, 72, 62, 49)
```

 # Create a linear regression model

```
model <- lm(y~x)
```

 # Print regression model

```
print(model)
```

OUTPUT :

```
Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)          x
  -39.7137      0.6847
```

 # Find the weight of a person With height 182

```
df <- data.frame(x = 182)
res <-  predict(model, df)
cat("\nPredicted value of a person
        with height = 182")
print(res)
```

OUTPUT :

```
Predicted value of a person
        with height = 182
    1
84.9098
```

 # Output to be present as PNG file

```
    png(file = "linearRegGFG.png")
```

 # Plot

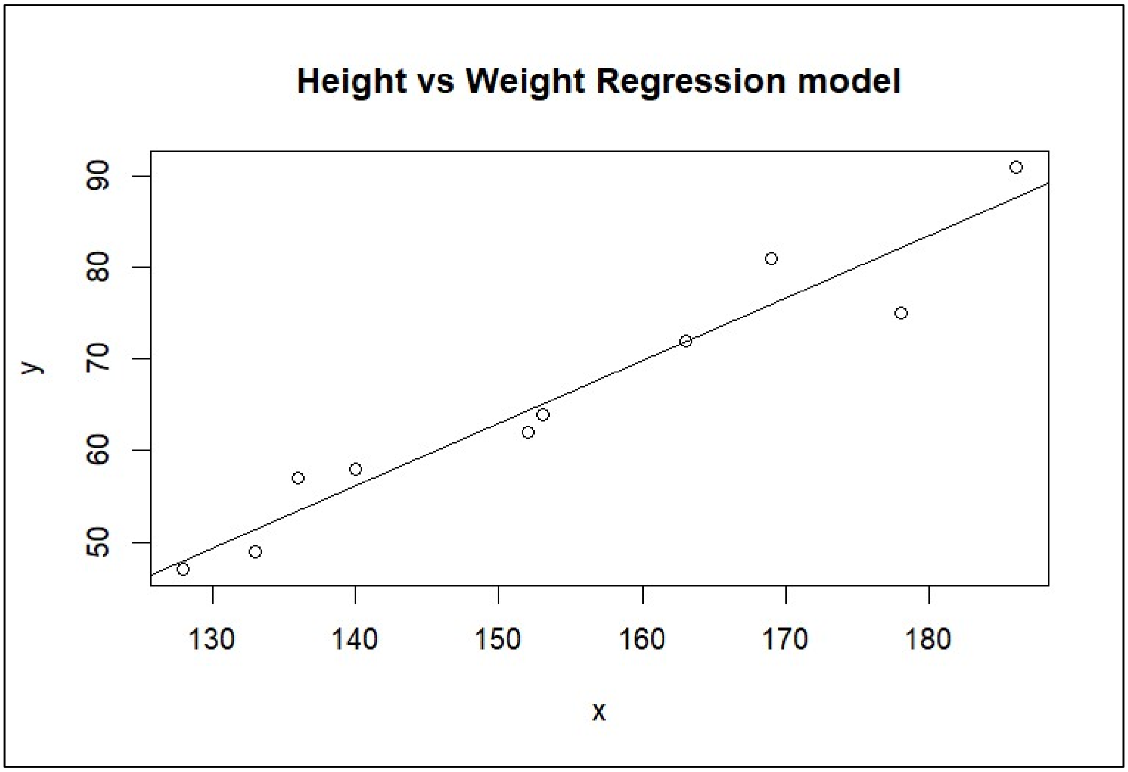
```
plot(x, y, main = "Height vs Weight Regression model")
abline(lm(y~x))
```

plot(out)

 # Save the file.

dev.off()

OUTPUT :



2. **Multiple Linear Regression**

 # Using airquality dataset

input <- airquality[1:50,c("Ozone", "Wind", "Temp")]

 # Create regression model

model <- lm(Ozone~Wind + Temp,data = input)

 # Print the regression model

cat("Regression model:\n")

print(model)

OUTPUT :

```
Regression model:
Call:
lm(formula = Ozone ~ Wind + Temp, data = input)
```

```
Coefficients:
(Intercept)        Wind        Temp
   -58.239       -0.739       1.329
```

 # Output to be present as PNG file

png(file = "multipleRegGFG.png")

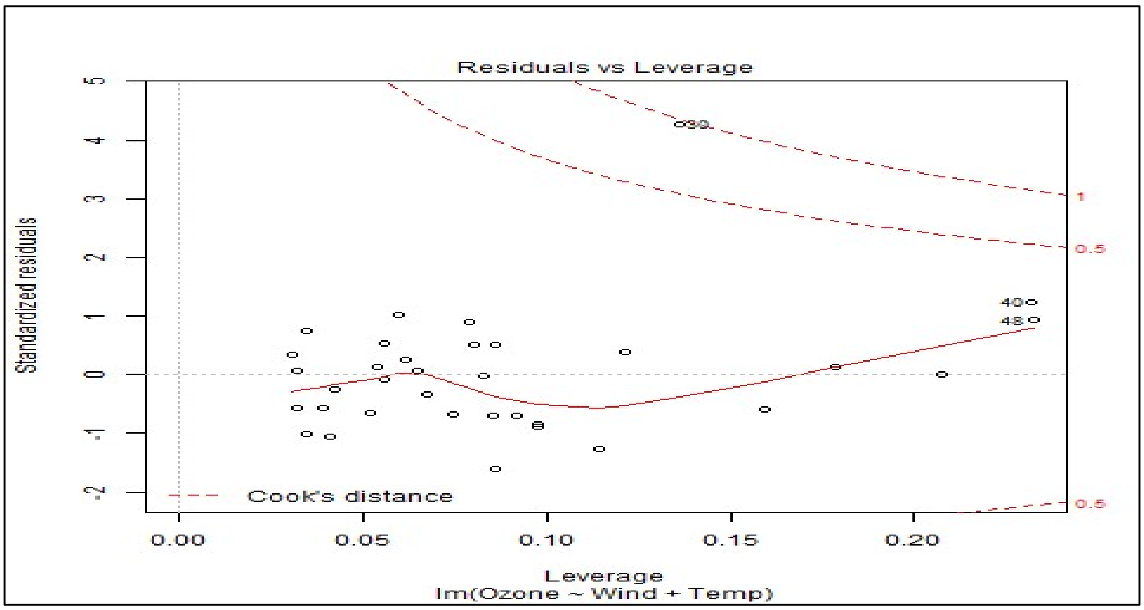
 # Plot

plot(model)

 # Save the file.

dev.off()

OUTPUT :



3. **Logistic Regression**

 # Using mtcars dataset

 # To create the logistic model

model <- glm(formula = vs ~ wt,family = binomial,data = mtcars)

 # Creating a range of wt values

x <- seq(min(mtcars$wt),max(mtcars$wt),0.01)

 # Predict using weight

y <- predict(model, list(wt = x), type = "response")

 # Print model

print(model)

OUTPUT :

```
Call:  glm(formula = vs ~ wt, family = binomial, data = mtcars)

Coefficients:
(Intercept)          wt
      5.715      -1.911

Degrees of Freedom: 31 Total (i.e. Null);  30 Residual
Null Deviance:          43.86
Residual Deviance: 31.37       AIC: 35.37
```

 # Output to be present as PNG file

png(file = "LogRegGFG.png")

 # Plot

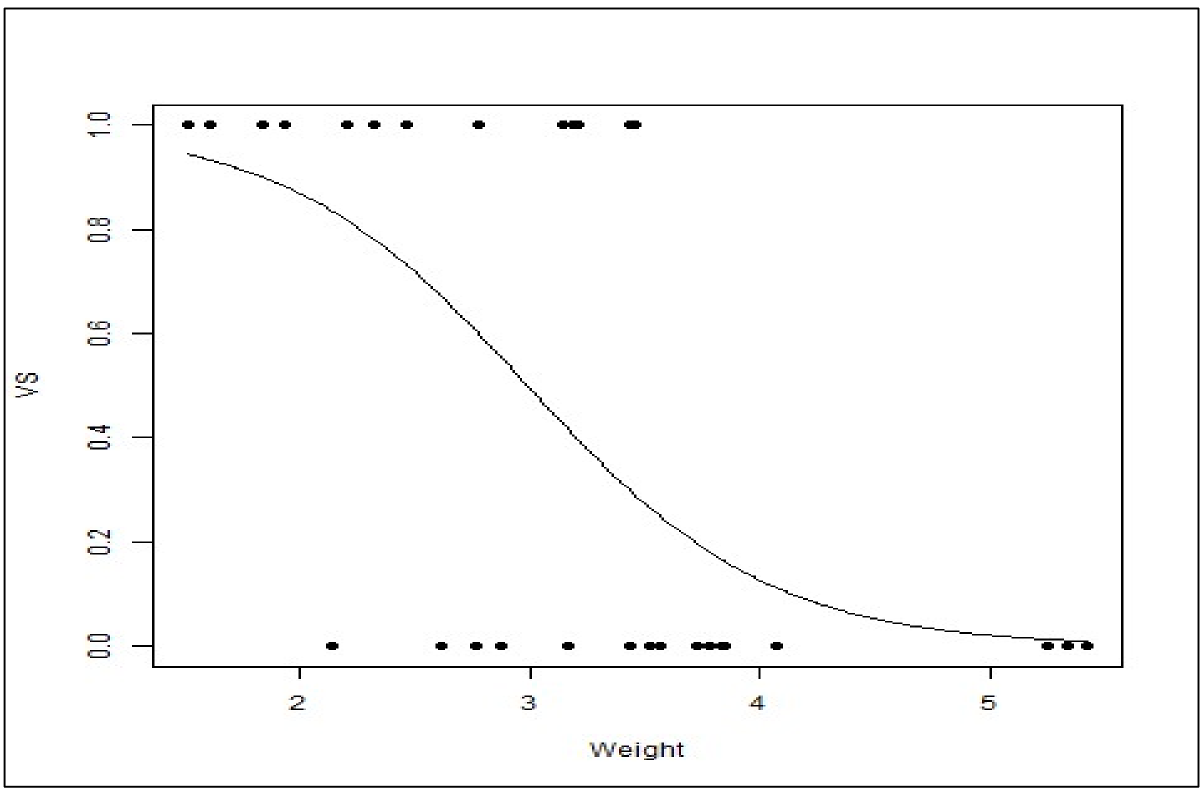
plot(mtcars$wt, mtcars$vs, pch = 16, xlab = "Weight", ylab = "VS")

lines(x, y)

 # Saving the file

dev.off()

OUTPUT :



1. **Performing One Way ANOVA test in R**

 # Installing the package

install.packages("dplyr")

 # Loading the package

library(dplyr)

 # Variance in mean within group and between group

boxplot(mtcars$disp~factor(mtcars$gear) xlab = "gear", ylab = "disp")

 # Step 1: Setup Null Hypothesis and Alternate Hypothesis

H0 = mu = mu01 = mu02(There is no difference

between average displacement for different gear)

H1 = Not all means are equal

 # Step 2: Calculate test statistics using aov function

mtcars_aov <- aov(mtcars$disp~factor(mtcars$gear))

summary(mtcars_aov)

OUTPUT :

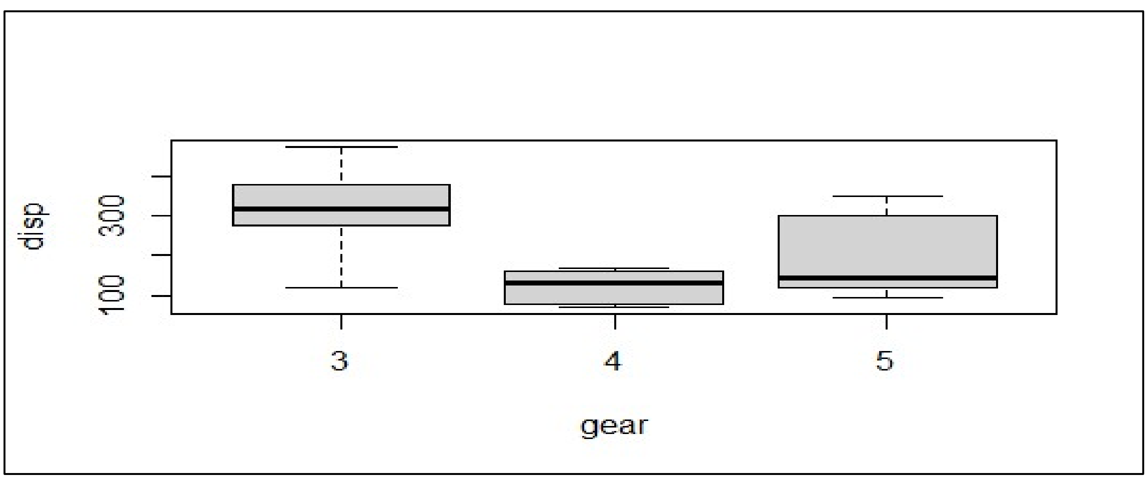
```
                    Df Sum Sq Mean Sq F value   Pr(>F)
factor(mtcars$gear)  2 280221  140110   20.73 2.56e-06 ***
Residuals           29 195964    6757
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

 # Step 3: Calculate F-Critical Value

For 0.05 Significant value, critical value = alpha = 0.05

 # Step 4: Compare test statistics with F-Critical value

and conclude test p < alpha, Reject Null Hypothesis



2. Performing Two Way ANOVA test in R

 # Installing the package

install.packages("dplyr")

 # Loading the package

library(dplyr)

 # Variance in mean within group and between group

```
boxplot(mtcars$disp~mtcars$gear, subset = (mtcars$am == 0),
        xlab = "gear", ylab = "disp", main = "Automatic")
boxplot(mtcars$disp~mtcars$gear, subset = (mtcars$am == 1),
        xlab = "gear", ylab = "disp", main = "Manual")
```

 # Step 1: Setup Null Hypothesis and Alternate Hypothesis

H0 = mu0 = mu01 = mu02(There is no difference between average displacement for different gear)

H1 = Not all means are equal

 # Step 2: Calculate test statistics using aov function

```
mtcars_aov2 <- aov(mtcars$disp~factor(mtcars$gear) *factor(mtcars$am))
summary(mtcars_aov2)
```

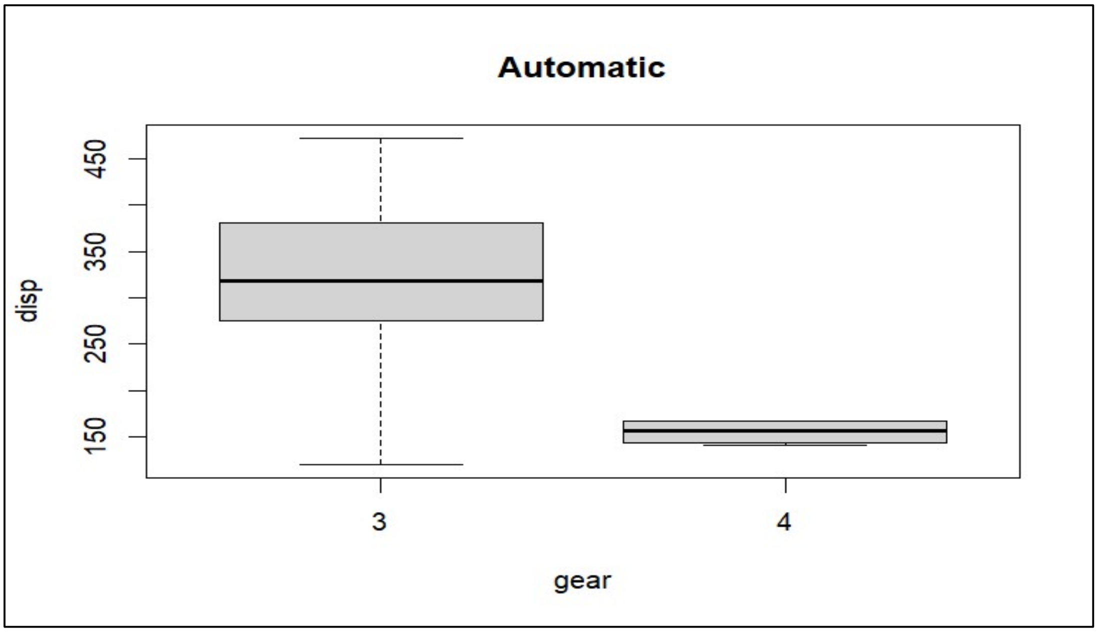
OUTPUT :

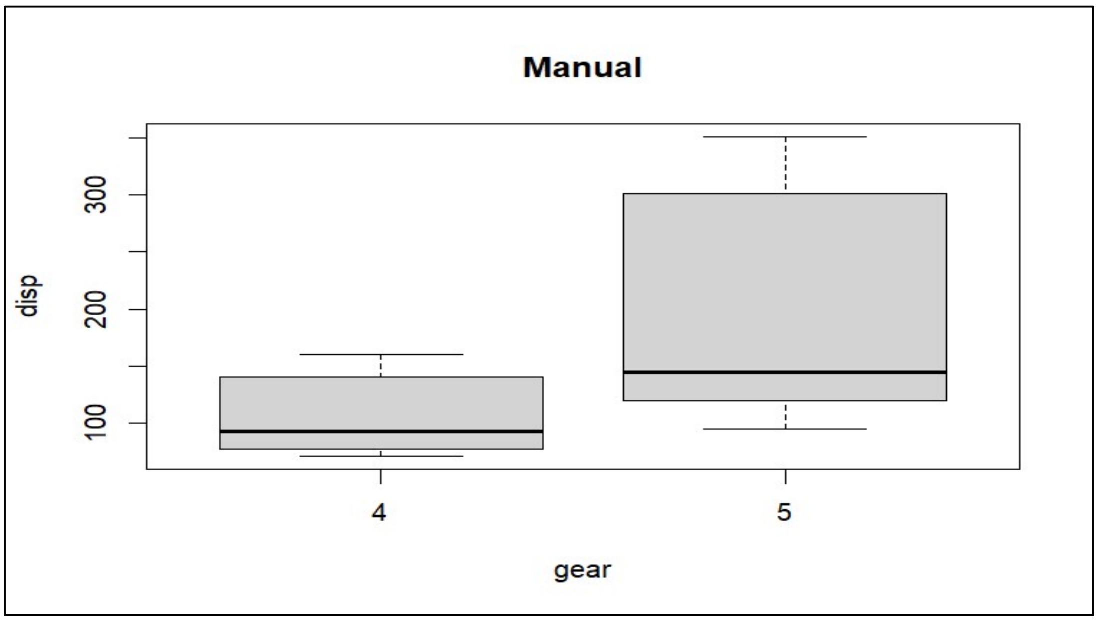
```
                    Df Sum Sq Mean Sq F value   Pr(>F)
factor(mtcars$gear)  2 280221  140110  20.695 3.03e-06 ***
factor(mtcars$am)    1   6399    6399   0.945    0.339
Residuals           28 189565    6770
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

 # Step 3: Calculate F-Critical Value

For 0.05 Significant value, critical value = alpha = 0.05

 # Step 4: Compare test statistics with F-Critical value and conclude test p < alpha, Reject Null Hypothesis


Automatic
disp
450
350
250
150
3
4
gear


Manual
disp
300
200
100
4
5
gear