R Statistics

Dr. Kundan Kumar, PhD

- Mean, Median & Mode
- Linear Regression
- Logistic Regression
- Normal Distribution
- Chi-Square Test
- T test

Mean,Media & Mode

Statistical analysis in R is performed by using many in-built functions. Most of these functions are part of the R base package. We discuss mean, median and mode functions here

Mean

Syntax

The basic syntax for calculating mean in R is

```
mean(x, trim = 0, na.rm = FALSE, ...)
```

- ✓ x is the input vector.
- ✓ trim is used to drop some observations from both end of the sorted vector.
- ✓ na.rm is used to remove the missing values from the input vector.

Mean,Median & Mode

Median

The middle most value in a data series is called the median. The median() function is used in R to calculate this value.

Syntax median(x, na.rm = FALSE)

Example

```
# Create the vector.
x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Find the median.
result <- median(x)
print(result)
```

Output

```
[1] 5.6
```

Creating Dataframe in R

Mode

The mode is the value that has highest number of occurrences in a set of data. Unike mean and median, mode can have both numeric and character data. Here, we will create a function to calculate mode because R does not have an in built function for mode

Example

```
#mode
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}
# Create the vector with numbers.
v <- c(2,1,2,3,1,2,3,4,1,5,5,3,2,3)
getmode(v)
```

Output

```
[1] 2
```

Linear Regression

Regression analysis is a very widely used statistical tool to establish a relationship between two variables. One of these variables is called predictor (independent) variable whose value is gathered through experiments and the other is called dependent variable. These two variables are related through an equation of the following form;

$$y = ax + b$$

Linear Regression

Creating A linear regression model in R

- ✓ The lm() functions is used in R to great a linear regression
- ✓ The **predict()** function in R to make prediction.

Syntax

lm(formula,data)

- **formula** is a symbol presenting the relation between x and y.
- **data** is the vector on which the formula will be applied.

Linear Regression

Example

```
#Linear Relation
x <- c(151, 174, 138, 186, 128,
       136, 179, 163, 152, 131)
y <- c(63, 81, 56, 91, 47, 57,
       76, 72, 62, 48)

# Apply the lm() function.
relation <- lm(y~x)

print(relation)
```

Output

```
Coefficients:
(Intercept)            x
   -38.4551       0.6746
```

Linear Regression

Get summary of the relationship

```
print(summary(relation))
```

Output

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

Residuals:
             Estimate Std. Error t value
(Intercept) -38.45509    8.04901  -4.778
x             0.67461    0.05191  12.997
            Pr(>|t|)
(Intercept)  0.00139 **
x            1.16e-06 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.253 on 8 degrees of freedom
Multiple R-squared:  0.9548,    Adjusted R-squared:  0.9491
F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.164e-06
```

>

Linear Regression

The predict() Function

Syntax predict(object, newdata)

- ✓ **object** is the formula which is already created using the lm() function.
- ✓ newdata is the vector containing the new value for predictor variable.

Example

```
# Find y for x=160.
a <- data.frame(x = 160)
y_pred <-  predict(relation,a)
print(y_pred)
```

Output

```
69.48258
```

Multiple Regression

Multiple regression is an extension of linear regression into relationship between more than two variables. In R, it is created using **lm()** function.It is represented mathematically as:

$$y = a + b1x1 + b2x2 + ...bnxn$$

Syntax

```
lm(y ~ x1+x2+x3...,data)
```

Example

```
input <- mtcars[,c("mpg","disp","hp","wt")]

# Create the relationship model.
model <- lm(mpg~disp+hp+wt, data = input)

# Show the model.
print(model)
```

Output

```
Coefficients:
(Intercept)        disp          hp          wt
  37.105505   -0.000937   -0.031157   -3.800891
```

Logistic Regression

The Logistic Regression is a regression model in which the response variable (dependent variable) has categorical values such as True/False or 0/1.
The **glm()** function to create the logistic regression model .

Syntax

```
glm(formula,data,family)
```

family is R object to specify the details of the model. It's value is binomial for logistic regression.

Logistic Regression

Example

```
#Logistic
#Create the data set
input <- mtcars[,c("am","cyl","hp","wt")]
#Build the model
am.data = glm(formula = am ~ cyl + hp + wt, data = input,
              family = binomial)
#Gives summary of the model
print(summary(am.data))
```

Logistic Regression

Output

```
Call:
glm(formula = am ~ cyl + hp + wt, family = binomial, data = i
nput)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.17272  -0.14907  -0.01464   0.14116   1.27641

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 19.70288    8.11637   2.428   0.0152 *
cyl          0.48760    1.07162   0.455   0.6491
hp           0.03259    0.01886   1.728   0.0840 .
wt          -9.14947    4.15332  -2.203   0.0276 *
```

Normal Distribution

R has four in built functions to generate normal distribution.They are;

dnorm(x, mean, sd):This is the PDF
pnorm(x, mean, sd):This is the CDF
qnorm(p, mean, sd): The quantile function
rnorm(n, mean, sd)

dnorm

```
# Specify x-values for dnorm function
x_dnorm <- seq(- 5, 5, by = 0.05)
# Probability are stored
y_dnorm <- dnorm(x_dnorm,mean = 2.5,sd=0.5)
plot(x_dnorm,y_dnorm)
```

Normal Distribution

pnorm

```
#generate a sequence
x_pnorm <- seq(- 10, 10, by = 0.2)
#Commulative probability of x_pnorm
y_pnorm <- pnorm(x_pnorm,2.5,2)
plot(x_pnorm,y_pnorm)
```

Normal Distribution

qnorm

This function takes the probability value and gives a number whose cumulative value matches the probability value. This can be used directly to get Z value from N(0,1)

Example

```
#qnorm
qnorm(0.95)
```

Output

```
> qnorm(0.95)
[1] 1.644854
```

Normal Distribution

rnorm

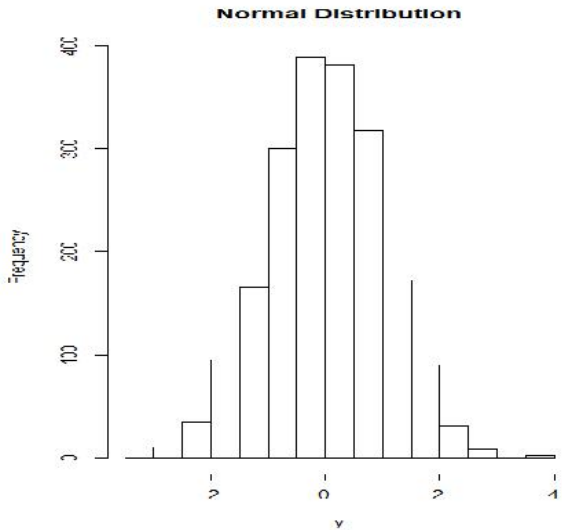
This function is used to generate random numbers whose distribution is normal

Example

```
#rnorm
#Get 2000 samples from the standard norm distribution
y <- rnorm(2000)

#Plot a histogram
hist(y, main="Normal DIstribution")
```

Output



Chi Square Tests

Chi-Square test is a statistical method to determine if two categorical variables have a significant correlation between them. Both those variables should be from same population and they should be categorical like – Yes/No, Male/Female, Red/Green. We use chisq.test() function for chi square test

Syntax: chisq.test(data)

data is the data in form of a table containing the count value of the variables in the observation.

Chi Square Tests

We will take the Cars93 data in the "MASS" library which represents the sales of different models of car in the year 1993.

```
# Load the library.
library("MASS")
# Create a table with the needed variables.
car.data = table(Cars93$AirBags, Cars93$Type)
print(car.data)

# Perform the Chi-Square test.
print(chisq.test(car.data))
```

Chi Square Tests

Output

```
                    Compact Large Midsize Small Sporty Van
  Driver & Passenger       2     4       7     0      3   0
  Driver only              9     7      11     5      8   3
  None                     5     0       4    16      3   6
> # Perform the Chi-Square test.
> print(chisq.test(car.data))

        Pearson's Chi-squared test

data:  car.data
X-squared = 33.001, df = 10, p-value = 0.0002723
```

The result shows the p-value of less than 0.05 which indicates a string correlation.

Hypothesis Testing

A hypothesis test is a process that uses sample statistics to test a claim about the value of a population parameter.

"H subzero" or "H naught"

A **null hypothesis** H_0 is a statistical hypothesis that contains a statement of equality such as $\leq$, $=$, or $\geq$.

"H sub-a"

A **alternative hypothesis** H_a is the complement of the null hypothesis. It is a statement that must be true if H_0 is false and contains a statement of inequality such as $>$, $\neq$, or $<$.

Source:**Larson & Farber,** ***Elementary Statistics: Picturing the World*****, 3e**

T test: One Sampe t test

- ✓ The one-sample t test is used to compare one mean (of a sample) to a reference value (an a priori chosen value). Here, we will use the t.test() function perform t test in R.
- ✓ Example: Consider the following; We have: a sample of measurements of daily energy intake (in Joules) from 11 students. Now, we want to know whether these data conform with the recommended value of 7725 Joules per day.

  Data: 5260,5470,5640,6180,6390,6515,6805,7515,7515,8230,8770

Syntax

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

T test: One Sampe t test

Example

```
#Create data
daily.intake <-c(5260,5470,5640,6180,6390,6515,
                 6805,7515,7515,8230,8770)
#Set the mean
t_mu <- 7725
#ttest
t.test(daily.intake,mu=t_mu)
```

Output

```
        One Sample t-test

data:  daily.intake
t = -2.8208, df = 10, p-value = 0.01814
alternative hypothesis: true mean is not equal to 772
95 percent confidence interval:
 5986.348 7520.925
sample estimates:
mean of x
 6753.636
```

Interpretation of Output: Discussion

T test:Two-Sample t test

- ✓ Instead of comparing one mean to a reference value, we might more often want to compare two means (two samples). We could also say we want to test the Null-hypothesis that two samples come from distributions with the same mean.
- ✓ A two-sample t-test is used to test the difference between two population means.
- ✓ In R, t.test() function is used here as well but with some parameter tweaked.
- ✓ To get the classical two-sample t test we have to explicitly say that the variances are assumed to be equal (var.equal = T).
- ✓ Example: We will use the **sleep** dataset in r

T test: Two-Sample t test

Example: Here , we use the sleep dataset of the datasets package

```
View(sleep)# View data set
#Separate extra according to group
#to have two groups for the test
t.test(extra~group,data=sleep,var.equal=T)
```

Output

```
        Two Sample t-test

data:  extra by group
t = -1.8608, df = 18, p-value = 0.07919
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.363874  0.203874
sample estimates:
mean in group 1 mean in group 2
           0.75            2.33
```

Thank You

FOLLOW ME ON


kundankumar011


Kundan-Rwanda


kundan.kumar011


@kundankumar011


kundan.kumar011


+250 – 786183431/ 735805278


+250 - 786183431


kundan.kumar011@gmail.com