R tutorial

sample <- sample( 1:20,40,replace = TRUE)

table(sample)[table(sample) > 2]

?? what is || or | or &&

Note that a single ‘|’ was used in the previous example. There is a difference between ‘||’ and ‘|’. A single bar will perform a vector operation, term by term, while a double bar will evaluate to a single TRUE or FALSE result:

http://www.cyclismo.org/tutorial/R/vectorIndexing.html#id3

**>** (c(**TRUE**,**TRUE**))|(c(**FALSE**,**TRUE**))

[1] TRUE TRUE

**>** (c(**TRUE**,**TRUE**))||(c(**FALSE**,**TRUE**))

[1] TRUE

**>** (c(**TRUE**,**TRUE**))&(c(**FALSE**,**TRUE**))

[1] FALSE TRUE

**>** (c(**TRUE**,**TRUE**))&&(c(**FALSE**,**TRUE**))

[1] FALSE

How to know the current work directory

Getwd()

Get to know what inside

Dir()

How to set the directry you wanna work, you need to

setwd("/Users/wenyi.yan/Desktop/MISC")

getwd() # check which dir currently

setwd("/Users/wenyi.yan/Documents/SCU/Edmunds") # set the working dir

pur\_data <- read.csv( file = "CARS\_PAID\_Final.txt", header = T, sep = "~") # read.csv could read txt file. tell R how to separete the colume by "~" "|" or " space ", and the header or not

can\_data <- read.csv("Cancelled\_VINs~.txt",header = F, sep = "~")

* need to add the “” and / to identify which directory you wanna pick

how to read the table into R

data1 <- read.table("42414",header = T)

how to write table or other csv. Remember don’t need to tell to save which directy. R just save thing in current directy. write.csv( date1,’final\_report’) remember to quote it

write.csv(cs\_data,'cs\_data.csv')

you could read.table or read.csv file ( first is the name of file, that is 42414, and tell the computer is false or True.

Then could use

names()

dim()

class()

attributes()

typeof()

lapply(data1,class) \* The *lapply* command returns a list. The entries in the list have the same names as the entries in the list that is passed to it. The values of each entry are the results from applying the function.

data1$C

to check the entire dataset

different with table and data. Fame ,table can assign the row name ,howvern the dataframe can/t use index to represent the row

you could check by column and row

data1[1:2,]

data[,4:6]

or you could remove the column

data[-1,]

data[,-31]

or you could check (no idea what the na.rm mean?)

summary(data1)

max(data1$MED\_SCORE,na.rm = FALSE)

min(data1$MED\_SCORE,na.rm = FALSE)

mean(data1$MED\_SCORE,na.rm = FALSE)

if you want to know which index is the max and you wanna to remove it you could do that

which

how to use package,

first go to tool \_> install package …

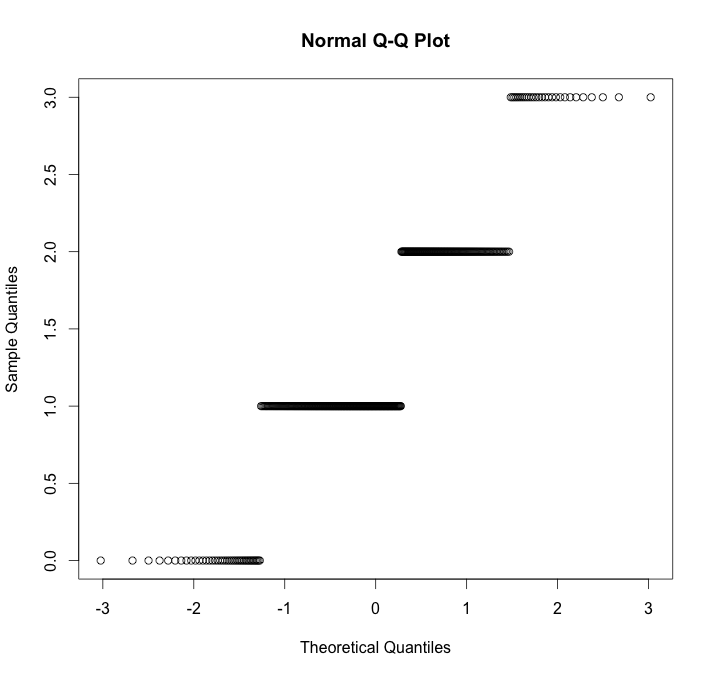
then to activate it library( ggplot)

how to plot the data

first we randomly to choose some data

numberWhite <- rhyper(400,4,5,3)

qqnorm(numberWhite)



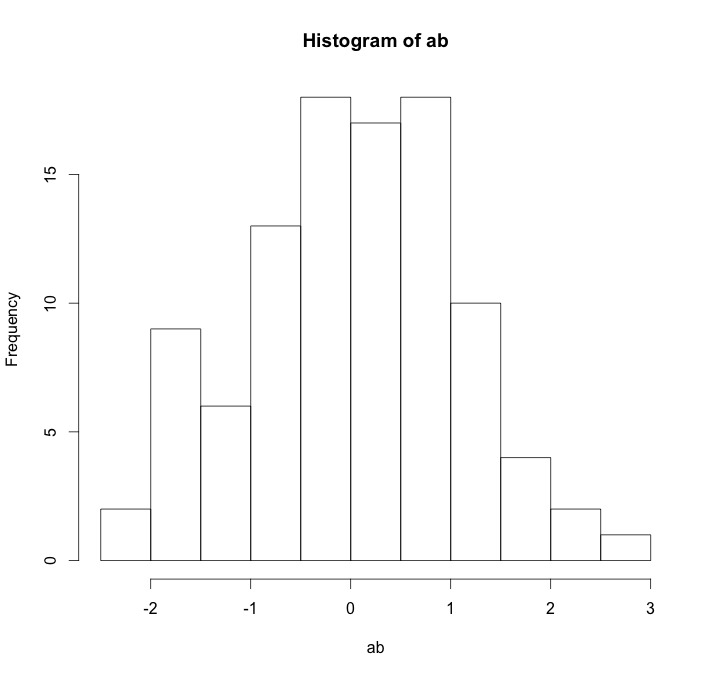
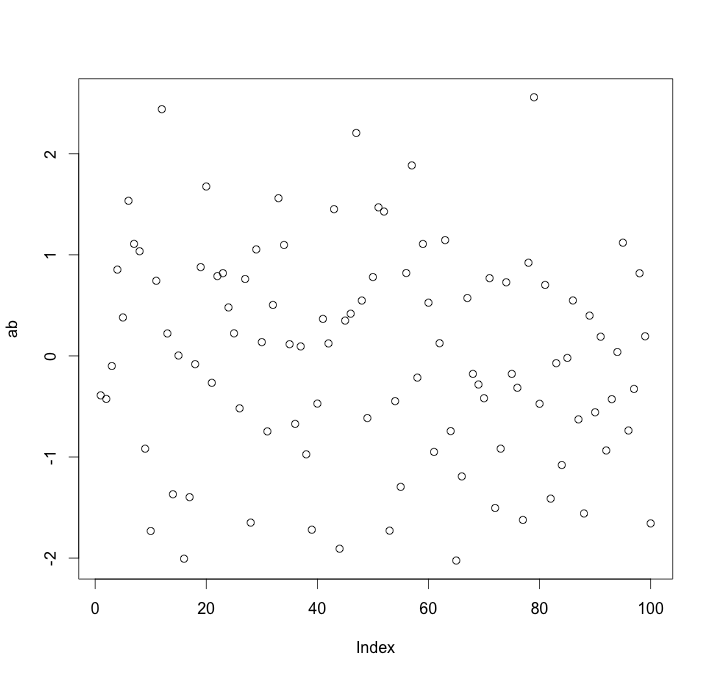
x1 <- runif(8,min = 15,max = 25) ( 8 numbers, min is 15, max is 25)

ab <- rnorm(100,mean = 0, sd = 1)

then we plot it in two ways, one using

plot(ab)

hist(ab)



how to create a list you don’t need to type , just duplicate

a <- c( “a”,”b”)

rep( a, 5)

ababababab

a <- c( “a”,”b”)

rep( a, each = 5)

aaaaabbbbb

> paste0(1:12)

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"

> paste(1:12)

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"

> as.character(1:12)

[1] "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "11" "12"

> > seq(1,20,by = 1)

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

d <- data.frame(value=c(d1,d2), condition=rep(c("A","B"),each=100))

> d

value condition

1 -37.53911331 A

2 12.29651333 A

3 -31.65605031 A

4 -7.88668879 A

5 56.48428691 A

x <- 1:20 matrix(rep(x,2), ncol = 2)

[,1] [,2]

[1,] 1 1

[2,] 2 2

[3,] 3 3

[4,] 4 4

[5,] 5 5

[6,] 6 6

[7,] 7 7

[8,] 8 8

[9,] 9 9

[10,] 10 10

[11,] 11 11

[12,] 12 12

[13,] 13 13

[14,] 14 14

[15,] 15 15

[16,] 16 16

[17,] 17 17

[18,] 18 18

[19,] 19 19

[20,] 20 20

how to create a data.frame by yourself

data.frame(a = col1,b=col2)

a b

1 1 2

2 1 2

3 1 2

giving a data frame , each column would be different type

first column is factor , second is numeric, third is integer

> simple

trial mass velocity

1 A 10.0 12

2 A 11.0 14

3 B 5.0 8

4 B 6.0 10

5 A 10.5 13

6 B 7.0 11

> simple$trial

[1] A A B B A B

Levels: A B

> simple$mass

[1] 10.0 11.0 5.0 6.0 10.5 7.0

> simple$velocity

[1] 12 14 8 10 13 11

> attributes(simple)

$names

[1] "trial" "mass" "velocity"

$class

[1] "data.frame"

$row.names

[1] 1 2 3 4 5 6

class(simple$trial)

[1] "factor" > class(simple$mass)

> class(simple$mass)

[1] "numeric"

> class(simple$velocity)

[1] "integer"

don“t 课 ow

http://www.cyclismo.org/tutorial/R/types.html

**>** a = c(**TRUE**,**FALSE**)

**>** b = c(**FALSE**,**FALSE**)

**>** a|b [1] TRUE FALSE

**>** a||b [1] TRUE

**>** xor(a,b) [1] TRUE FALSE

beside rep you could generate a list of number,

you could use a list of seq to generate a list of number

> ss <- seq(-5,5,by = 0.2)

> ss

[1] -5.0 -4.8 -4.6 -4.4 -4.2 -4.0 -3.8 -3.6 -3.4 -3.2 -3.0 -2.8 -2.6

[14] -2.4 -2.2 -2.0 -1.8 -1.6 -1.4 -1.2 -1.0 -0.8 -0.6 -0.4 -0.2 0.0

[27] 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6

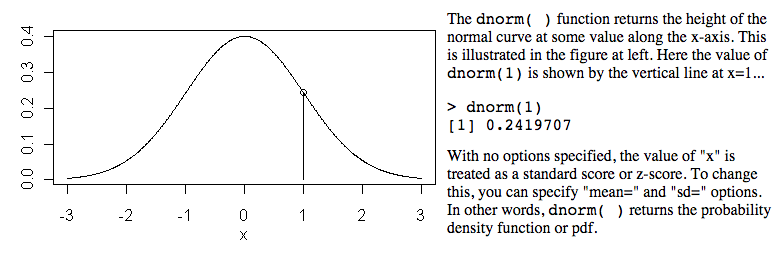
[40] 2.8 3.0 3.2 3.4 3.6 3.8 4.0 4.2 4.4 4.6 4.8 5.0

http://ww2.coastal.edu/kingw/statistics/R-tutorials/prob.html

what is dnorm , rnorm pnorm

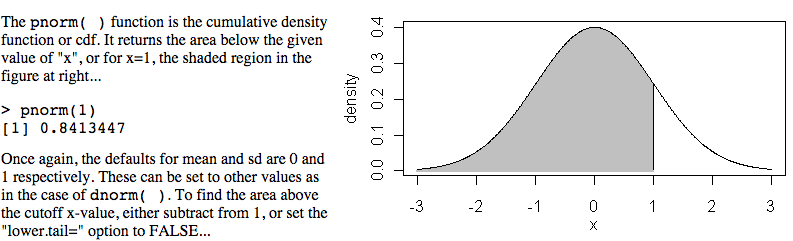
dnorm(0)

when the mean is 0 , how ‘s the height of it.



pnorm(1)

when 1 sd, how the propobility of the left side



v <- c(0,1,2)

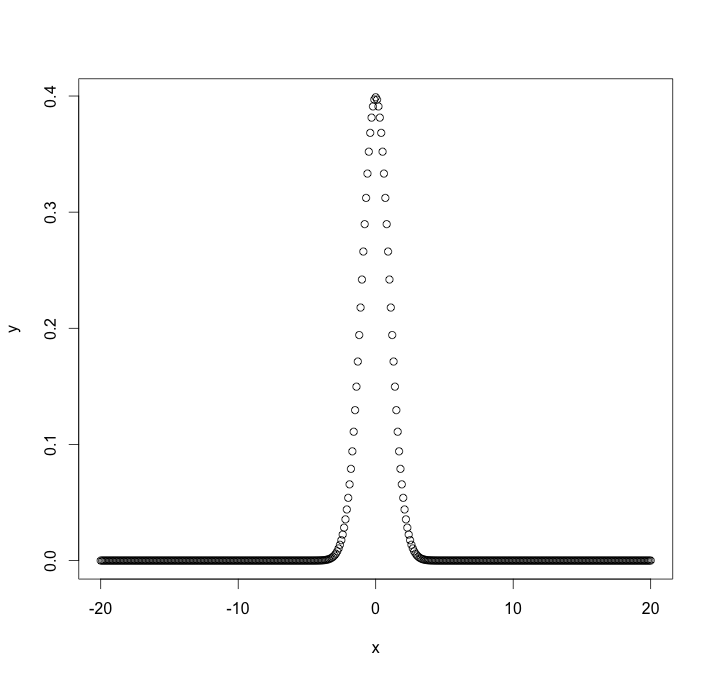
dnorm(v)

[1] 0.39894228 0.24197072 0.05399097

x <- seq(-20,20,by=.1)

y <- dnorm(x)

plot(x,y)

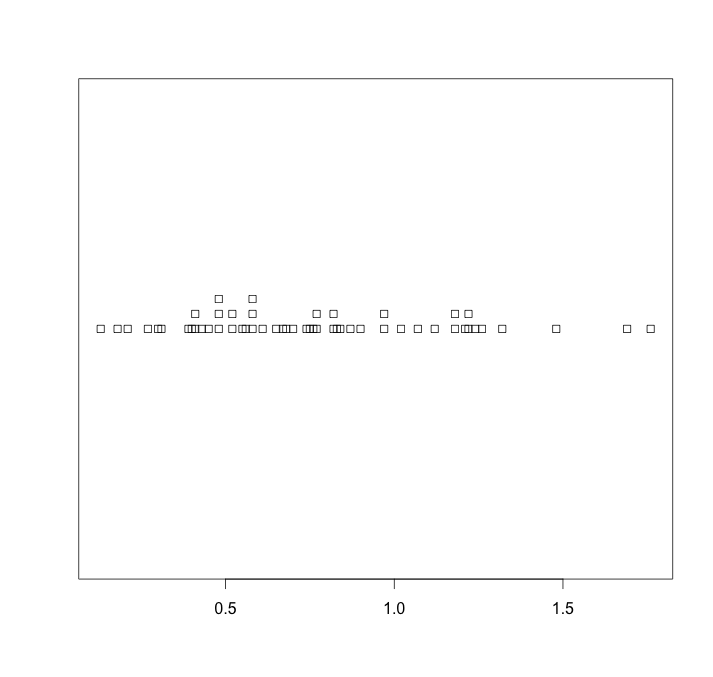


Plot chat

stripchart(w1$vals)

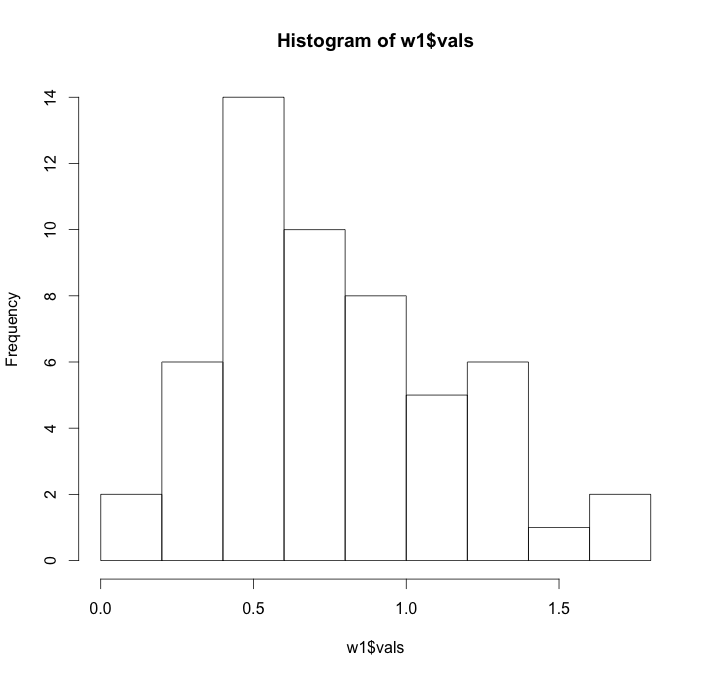
you could get frequcy of the number using stack but you still could get hist

stripchart(w1$vals,method="stack")



hist(w1$vals)

hist(w1$vals,breaks=100)



boxplot(w1$vals)

＊ plot STBM but in different factor C level

boxplot(tree$STBM~tree$C)

take a look at the distribution of a set of number

w1$vals

[1] 0.43 0.40 0.45 0.82 0.52 1.32 0.90 1.18 0.48 0.21 0.27 0.31 0.65

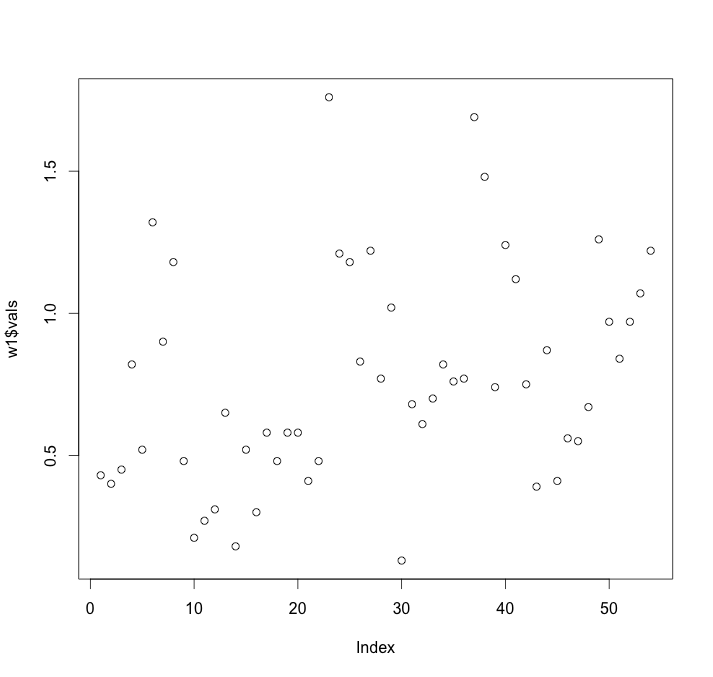
[14] 0.18 0.52 0.30 0.58 0.48 0.58 0.58 0.41 0.48 1.76 1.21 1.18 0.83

[27] 1.22 0.77 1.02 0.13 0.68 0.61 0.70 0.82 0.76 0.77 1.69 1.48 0.74

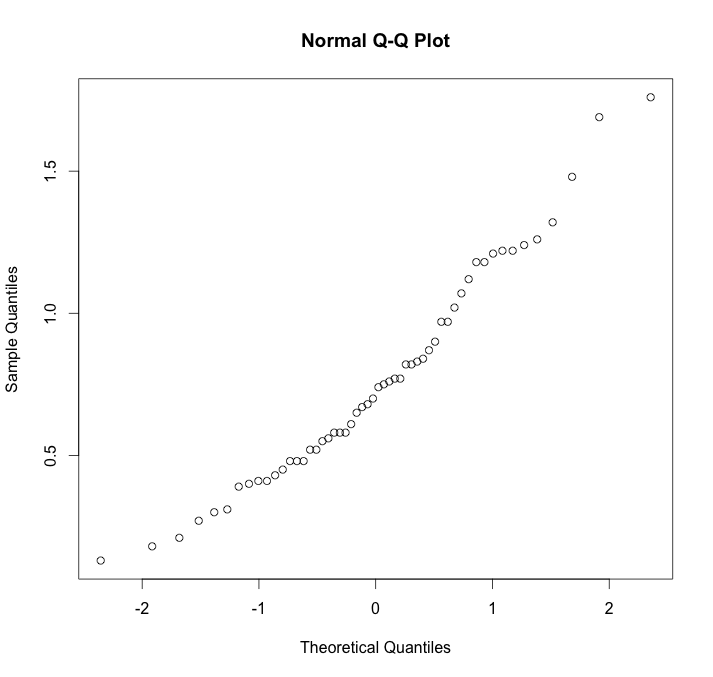
[40] 1.24 1.12 0.75 0.39 0.87 0.41 0.56 0.55 0.67 1.26 0.97 0.84 0.97

[53] 1.07 1.22

plot(w1$vals)



qqnorm(w1$vals)



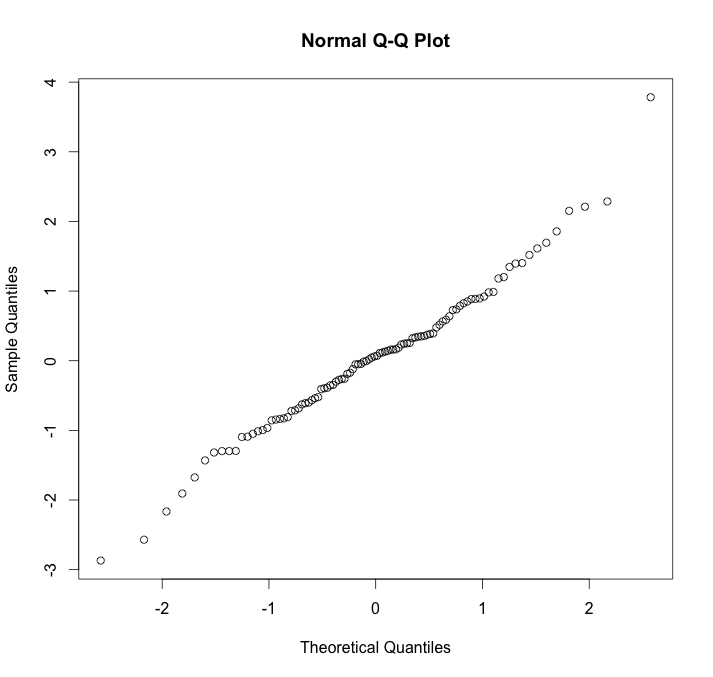
check if the number is normal distribution

d <- rnorm(100,mean = 0 ,sd =1 ) -> generate a list of number

qqnorm(d)

plot(d)

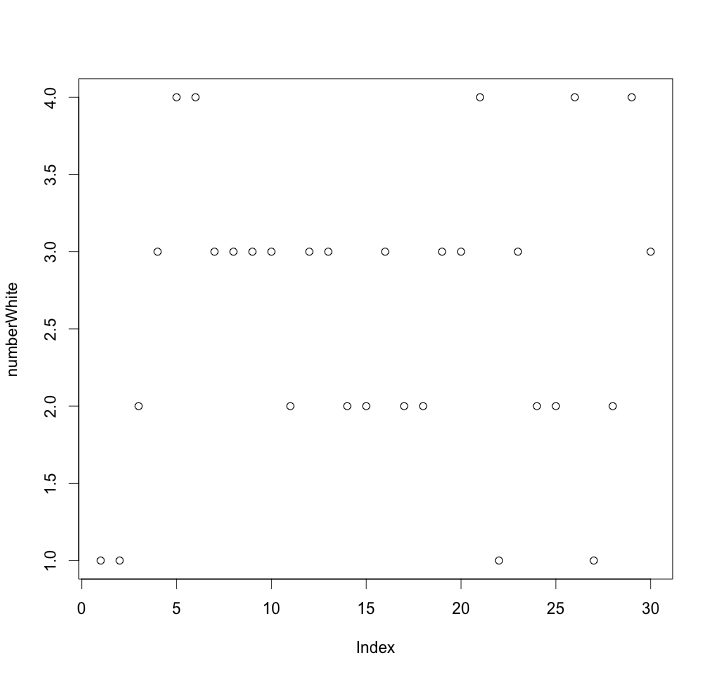
this is normal distribution!!!! PLOT !!!!! tho it is not a bell curve



rhyper what is

numberWhite <- rhyper(30,10,5,4)

plot(numberWhite)

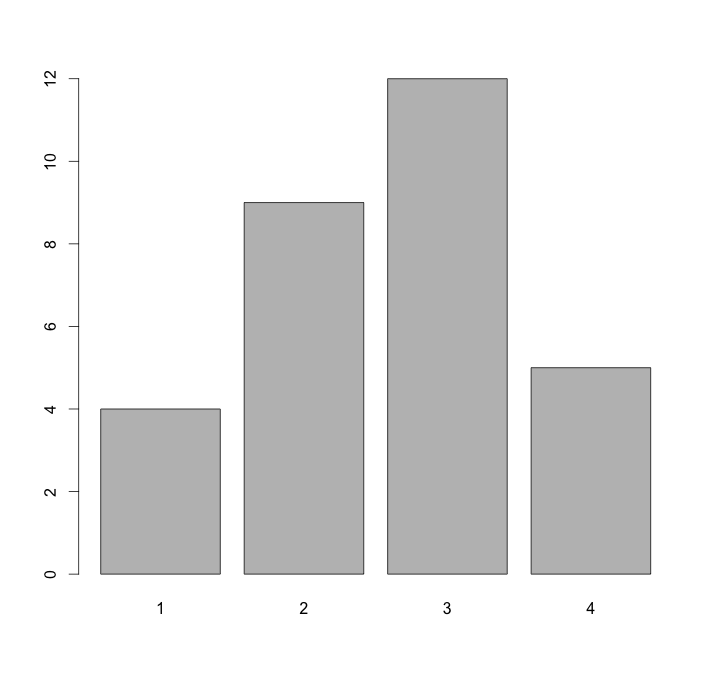


change to factor would be categorize into factor

numberWhite <- rhyper(30,10,5,4)

numberWhite <- as.factor(numberWhite)

plot(numberWhite)



30 is 30 observation, last 2 is mean have 3(2+1) levels ; for 10 and 20 no idea

numberWhite <- rhyper(30,10,20,2)

numberWhite <- as.factor(numberWhite)

must to change to factor then could catergizae to see the hist how to categorize

conditional filter the vector

a = c(7,2,5,3,8,2)

b = a[c(TRUE,FALSE,FALSE,FALSE,TRUE,FALSE)]

b

[1] 7 8

the same as

a = c(7,2,5,3,8,2)

b = a[a>6]

b

[1] 7 8

\*\* index using [] not () to repents

directly to get the coefficient to calculate the number

fit <- lm(rate ~ year)

fit

summary(fit)

attributes(fit)

typeof(fit)

fit$coefficients

fit$coefficients[[1]]

fit$coefficients[[2]]

how to create a data.frame manually

col1 <-c(1:10)

col2 <-c(11,20)

smoke <- data.frame(a=col1,b=col2)

a b

1 1 11

2 2 20

3 3 11

4 4 20

5 5 11

6 6 20

7 7 11

8 8 20

9 9 11

10 10 20

how to create a matix table

smoke <- matrix(c(51,43,22,92,28,21,68,22,9),ncol=3,byrow=TRUE)

> colnames(o) <- c("High","Low","Middle")

> rownames(o) <- c("current","former","never")

> smoke <- as.table(smoke)

> smoke

High Low Middle

current 51 43 22

former 92 28 21

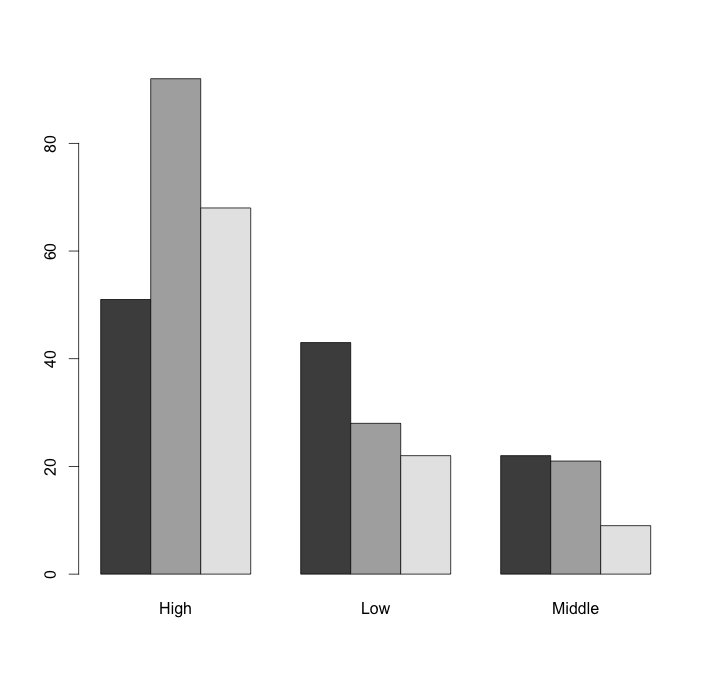
never 68 22 9

if the data is a table, all the number inside is factor , when you plot the data, the pic would be like this

smoke <- matrix(c(51,43,22,92,28,21,68,22,9),ncol=3,byrow=**TRUE**) **>** colnames(o) <- c("High","Low","Middle") **>** rownames(o) <- c("current","former","never") **>** smoke <- as.table(smoke)

smoke High Low Middle current 51 43 22 former 92 28 21 never 68 22 9

**>** barplot(smoke,legend=T,beside=T,main='Smoking Status by SES') **>** plot(smoke,main="Smoking Status By Socioeconomic Status")



how to create a table and assign a col and row name

A = matrix(c( 1, 2, 3, 4, 5, 6),ncol=3,byrow=TRUE)

colnames(A) <- c("A","B","C")

rownames(A) <- c("RA","RB")

A

A B C

RA 1 2 3

RB 4 5 6

>

\*\* no row name to dataframe. Just

how to create a data.frame to assing a name

you can name directly in the data.frame

> d1 <- rnorm(2, sd=30)

> d2 <- rnorm(2, sd=10)

> d <- data.frame(value=c(d1,d2), condition=rep(c("A","B"),each=2))

> d

value condition

1 7.979140 A

2 -18.484701 A

3 -5.036505 B

4 15.427697 B

rep(1:12)

[1] 1 2 3 4 5 6 7 8 9 10 11 12

> c(1:12)

[1] 1 2 3 4 5 6 7 8 9 10 11 12

>

rep!

runif

rexp

lapply

*tapply*

inpute in date must convert let R know

You have to use the *strptime* command to convert it into a time field.

myData <- data.frame(time=c("2014-01-23 14:28:21","2014-01-23 14:28:55", "2014-01-23 14:29:02","2014-01-23 14:31:18"), speed=c(2.0,2.2,3.4,5.5))

myData$time <- strptime(myData$time,"%Y-%m-%d %H:%M:%S")

sample(1,10,replace = T)

[1] 1 1 1 1 1 1 1 1 1 1

sample(0,10,replace = T)

[1] 0 0 0 0 0 0 0 0 0 0

qq plot

cc<-rnorm(100)

qqnorm(cc) 