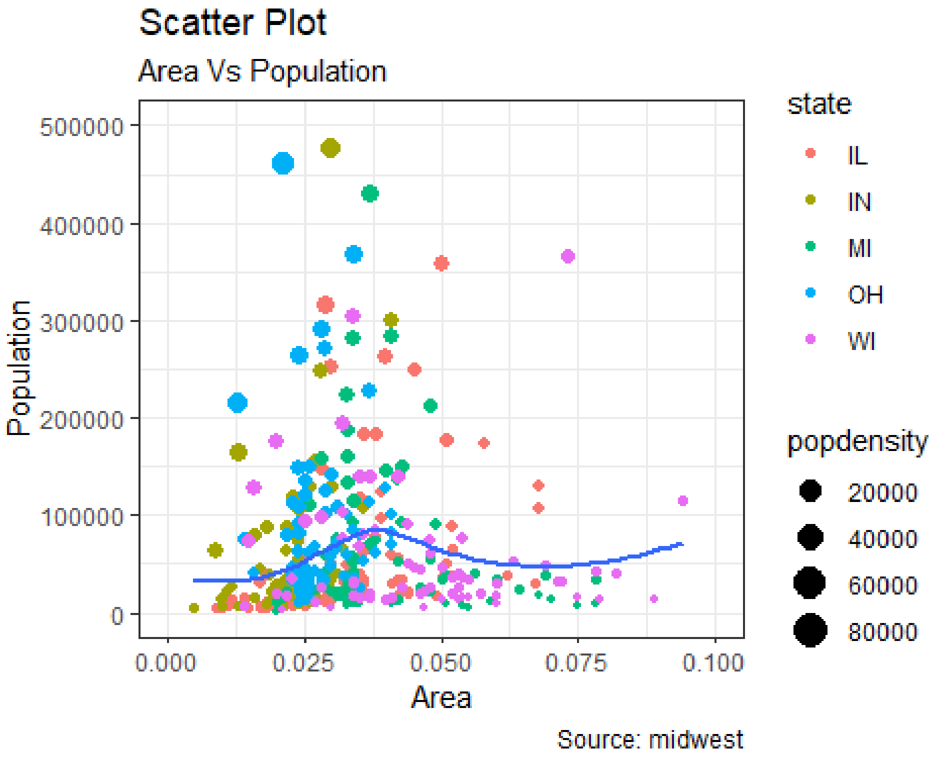
Initialize library and load dataset

```
options(scipen = 999)
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.1.3

theme_set(theme_bw())
data("midwest", package = "ggplot2")

midwest <- read.csv("http://goo.gl/G1K41K")
```

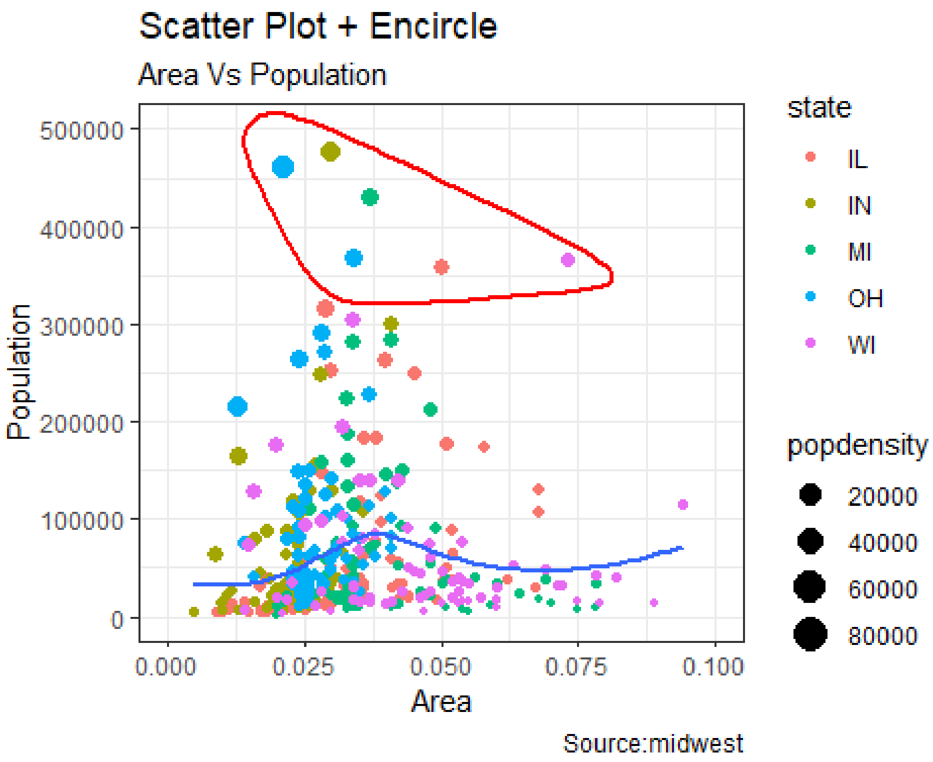
Scatter plot

```
gg <- ggplot(midwest, aes(x= area, y= poptotal)) + geom_point(aes(col= state,
size = popdensity)) + geom_smooth(method= "loess", se= F) + xlim(c(0,.1)) +
ylim(c(0,500000)) + labs(title = "Scatter Plot", subtitle = "Area Vs
Population", x = "Area", y = "Population", caption = "Source: midwest")
plot(gg)

## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 15 rows containing non-finite values (stat_smooth).

## Warning: Removed 15 rows containing missing values (geom_point).
```



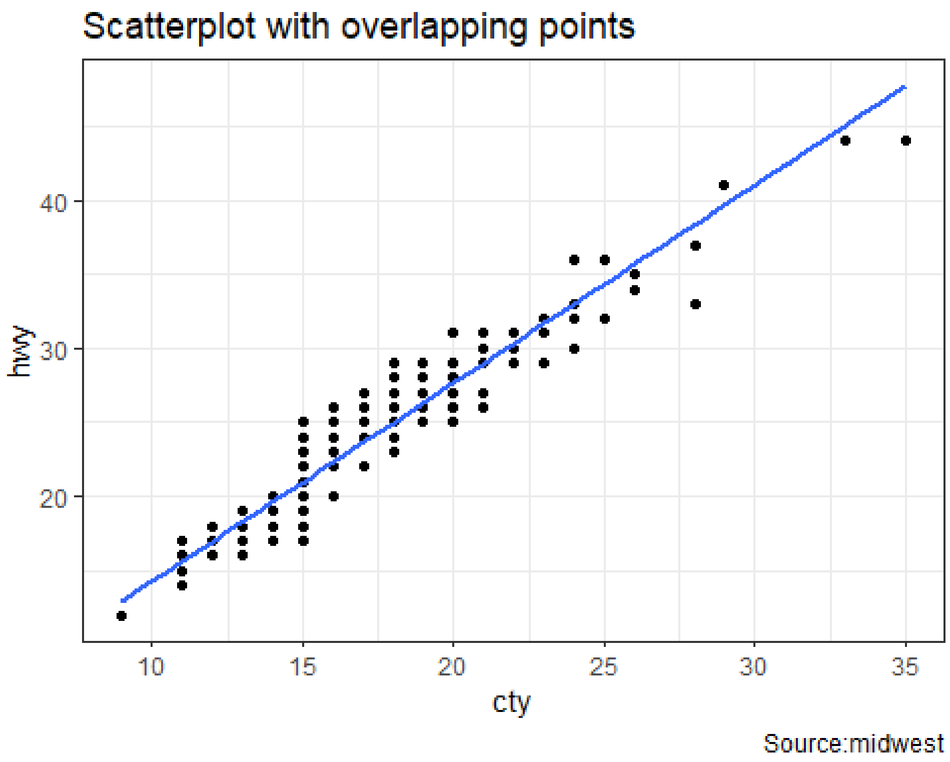
Scatterplot with Encircling

```
options(scipen = 999)
library(ggplot2)
library(ggalt)

## Warning: package 'ggalt' was built under R version 4.1.3

## Registered S3 methods overwritten by 'ggalt':
##   method                  from
##   grid.draw.absoluteGrob  ggplot2
##   grobHeight.absoluteGrob ggplot2
##   grobWidth.absoluteGrob  ggplot2
##   grobX.absoluteGrob      ggplot2
##   grobY.absoluteGrob      ggplot2

midwest_select <- midwest[midwest$poptotal > 350000 &
                            midwest$poptotal <= 500000 &
                            midwest$area > 0.01 &
                            midwest$area < 0.1, ]

ggplot(midwest, aes(x= area, y= poptotal)) + geom_point(aes(col= state, size
= popdensity)) + geom_smooth(method= "loess", se= F) + xlim(c(0,.1)) +
ylim(c(0, 500000)) + geom_encircle(aes(x=area, y=poptotal), data=
midwest_select, color= "red", size = 2, expand= 0.08) + labs(title = "Scatter
Plot + Encircle", subtitle = "Area Vs Population", x = "Area", y =
"Population", caption = "Source:midwest")

## `geom_smooth()` using formula 'y ~ x'

## Warning: Removed 15 rows containing non-finite values (stat_smooth).

## Warning: Removed 15 rows containing missing values (geom_point).
```

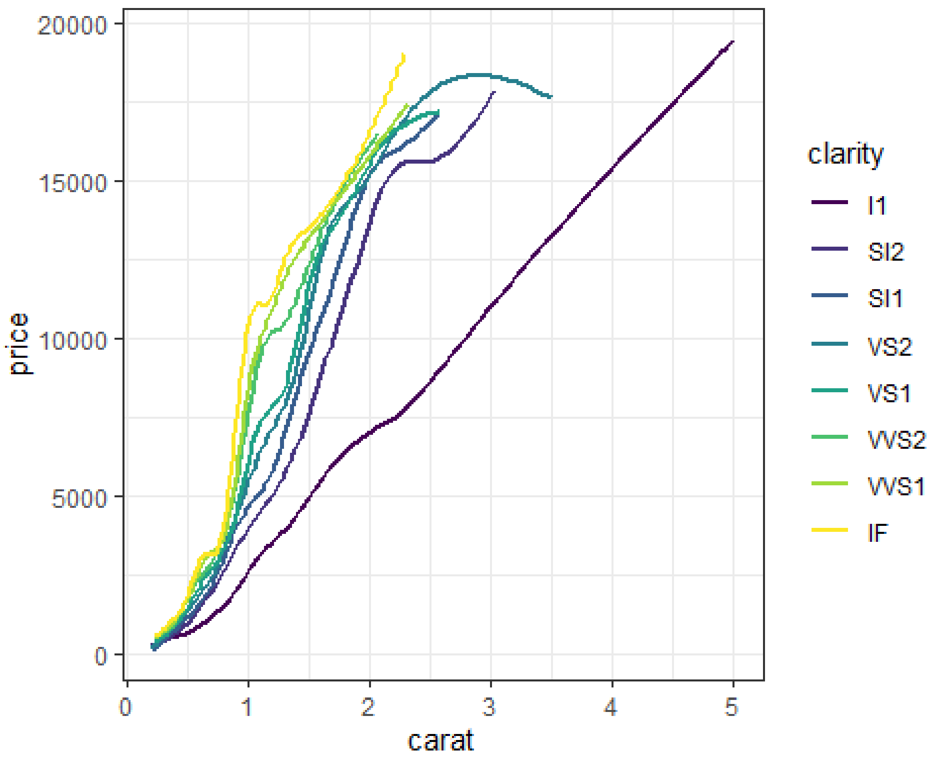


Jitter Plot

```
library(ggplot2)
data(mpg, package= "ggplot2")
theme_set(theme_bw())

g <- ggplot(mpg, aes(cty, hwy))

g + geom_point() + geom_smooth(method = "lm", se = F) + labs(title =
"Scatterplot with overlapping points", x = "cty", y = "hwy", caption =
"Source:midwest")

## `geom_smooth()` using formula 'y ~ x'
```

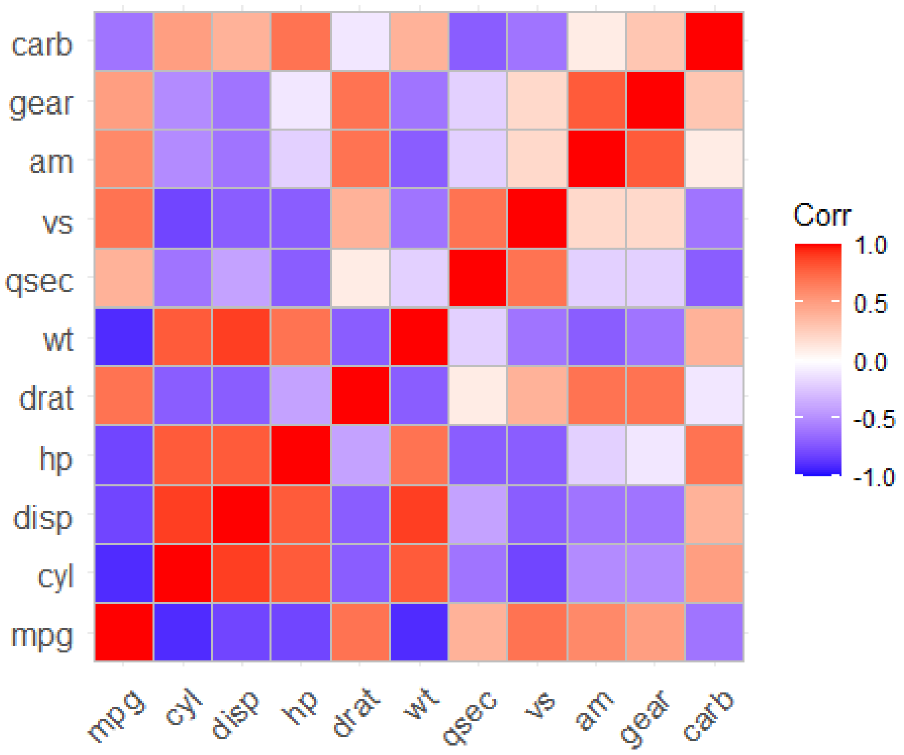


```
library(ggplot2)
data("diamonds", package = "ggplot2")
# View(diamonds)
ggplot(diamonds, aes(x=carat, y=price, color=clarity)) +
geom_smooth(se=FALSE)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



## How to plot correlograms in r?

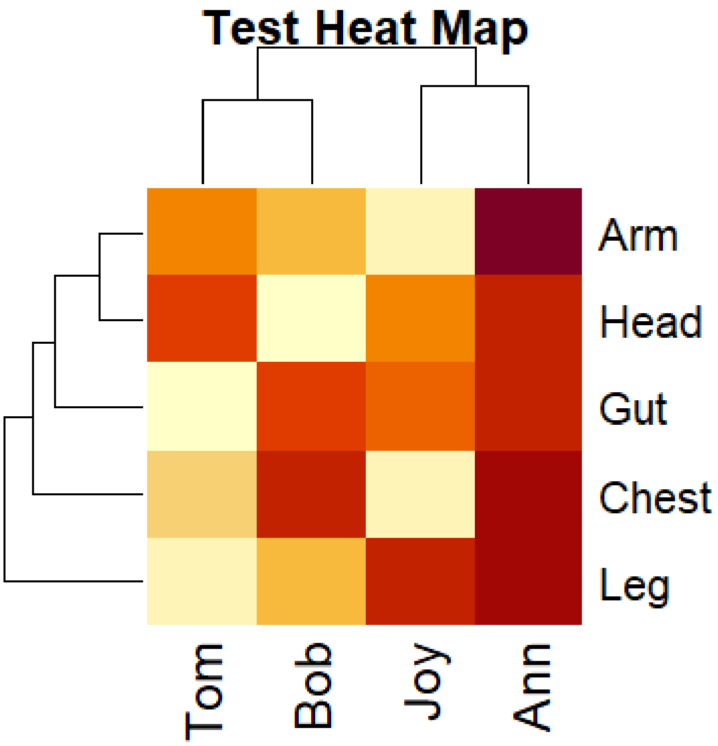
```
library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 4.1.3

data(mtcars)
corr <- round(cor(mtcars), 1)
ggcorrplot(corr)
```



```
library(maps)

## Warning: package 'maps' was built under R version 4.1.3

map("state", interior = FALSE)
map("state", boundary = TRUE, col="gray", add = TRUE)
```

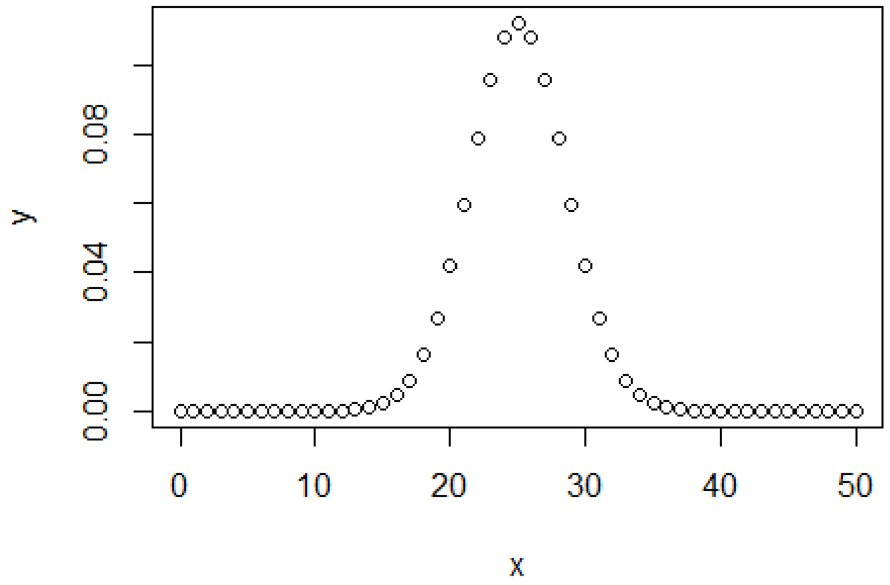
## how to make a heatmap in R

```
ds = data.frame(rnorm(5, 50, 20),rnorm(5, 50, 20),rnorm(5, 50, 20),rnorm(5,
50, 20))
rn = c("Arm","Leg","Chest","Gut","Head")
cn = c("Ann","Bob","Tom","Joy")
x = data.matrix(ds, rownames.force = FALSE)
heatmap(x, labRow=rn, labCol=cn, main = "Test Heat Map")
```



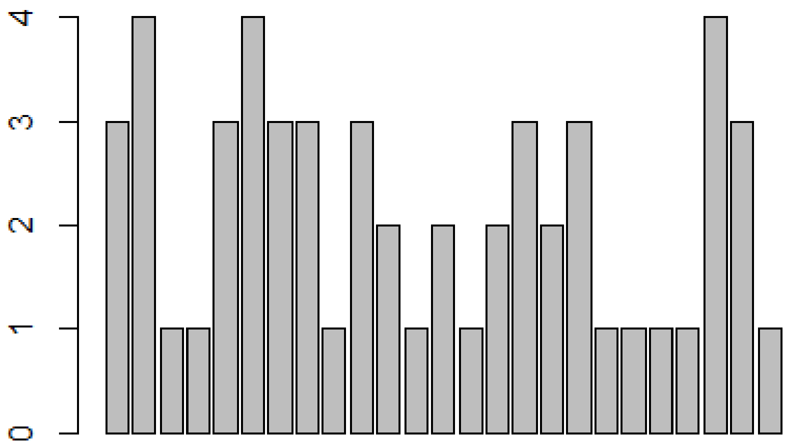
## Create a sample of 50 numbers which are incremented by 1.Create the binomial distribution,plot the graph for this sample

```
# Create a sample of 50 numbers which are incremented by 1.
x <- seq(0,50,by = 1)

# Create the binomial distribution.
y <- dbinom(x,50,0.5)

# Plot the graph for this sample.
plot(x,y)
```

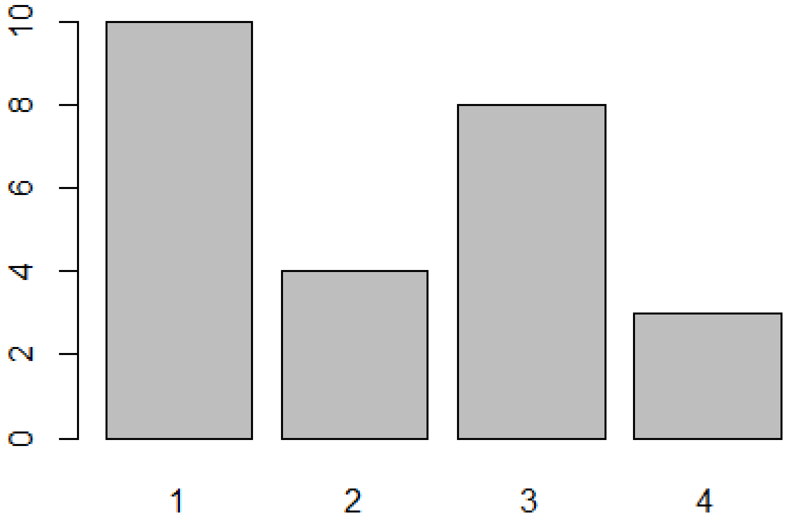


## Suppose, a group of 25 people are surveyed as to their beer-drinking preference. The categories were (1) Domestic can, (2) Domestic bottle, (3) Microbrew and (4) import. The raw data is 3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1 Let's make Suitable plot of both frequencies and proportions

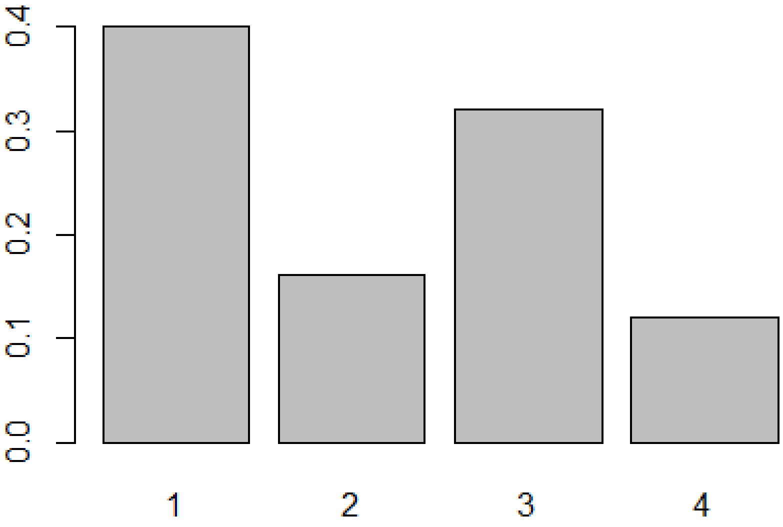
```
beer = c(3, 4, 1, 1, 3, 4, 3, 3, 1, 3, 2, 1, 2, 1, 2, 3, 2, 3, 1, 1, 1, 1, 4,
3, 1)

barplot(beer)
```



```
barplot(table(beer))
```

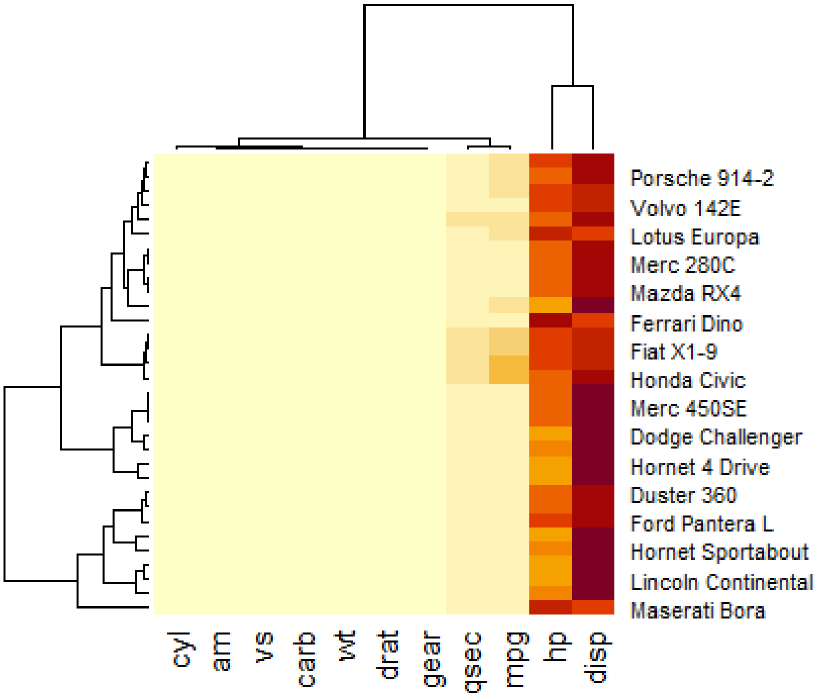


```
barplot(table(beer)/length(beer))
```

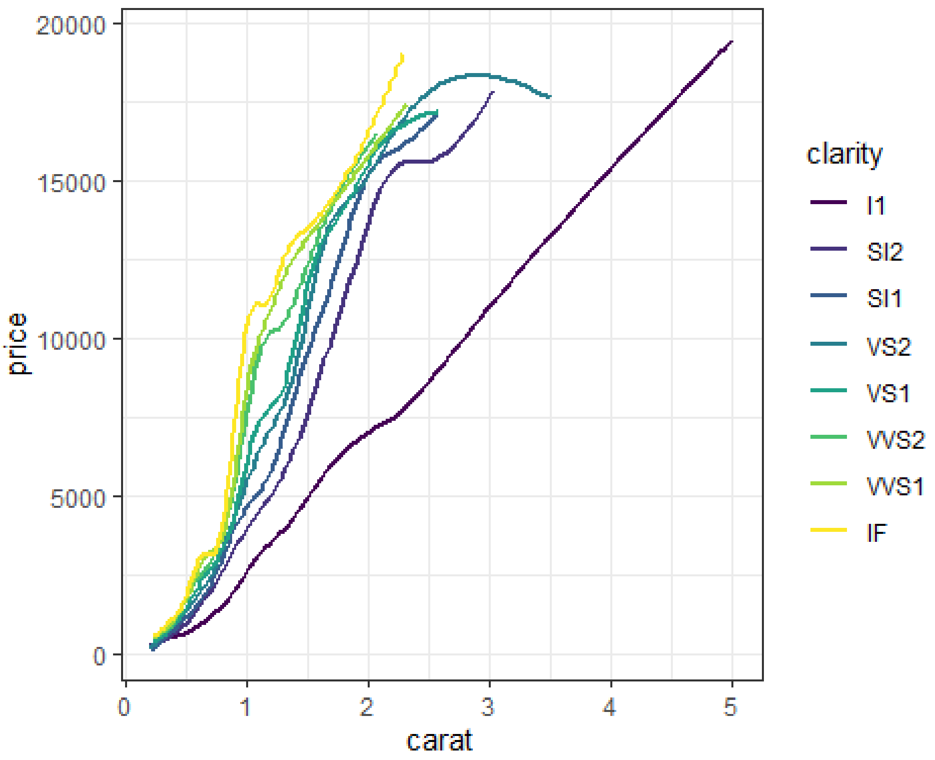


With the help of mtcar data plot a heat map.

```
# The mtcars dataset:
data <- as.matrix(mtcars, packages = "ggplot2")

# Default Heatmap
heatmap(data)
```



```
data("diamonds")

ggplot(diamonds, aes(x=carat, y=price, color=clarity)) +
geom_smooth(se=FALSE)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```


20000
15000
10000
5000
0
price
0
1
2
3
4
5
carat
clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF