Name:-Mithun G

USN:-19BTRCR006

LAB PROGRAM 6

1. What happens when the following code is run?
gender <- c("M" , "M" , "F" , "F" , "F")
whereF<- (gender == "F")
gender[whereF] <- "Female"

In [1]:

```
gender <- c("M" , "M" , "F" , "F" , "F" )
whereF <- (gender == "F" )
gender[whereF] <- "Female"
gender
```

'M' · 'M' · 'Female' · 'Female' · 'Female'

2. Construct the data frame char-num in the following way:
char <- c("2" , "1" , "0")
num<- 0:2
charnum<- data.frame(char, num);
Explore sampling distribution and central limit theorem in R

In [2]:

```
char <- c("2" , "1" , "0" )
num <- 0:2
charnum <- data.frame(char, num);
charnum
```

A data.frame:

3 × 2

char	num
<fct>	<int>
2	0
1	1
0	2

In [3]:

```
summary(charnum)
# since one of the variable is of character datatype, population mean and sample variance c
```

```
 char       num
 0:1   Min.   :0.0
 1:1   1st Qu.:0.5
 2:1   Median :1.0
       Mean   :1.0
       3rd Qu.:1.5
       Max.   :2.0
```

In [4]:

```
# sample variance
var(charnum[['num']])
```

1

3. Use the inbuilt data car and uses the possible graphical plots using ggplot2 graphical packages.

In [5]:

```
df <- datasets::mtcars
head(df)
```

A data.frame: 6 × 11

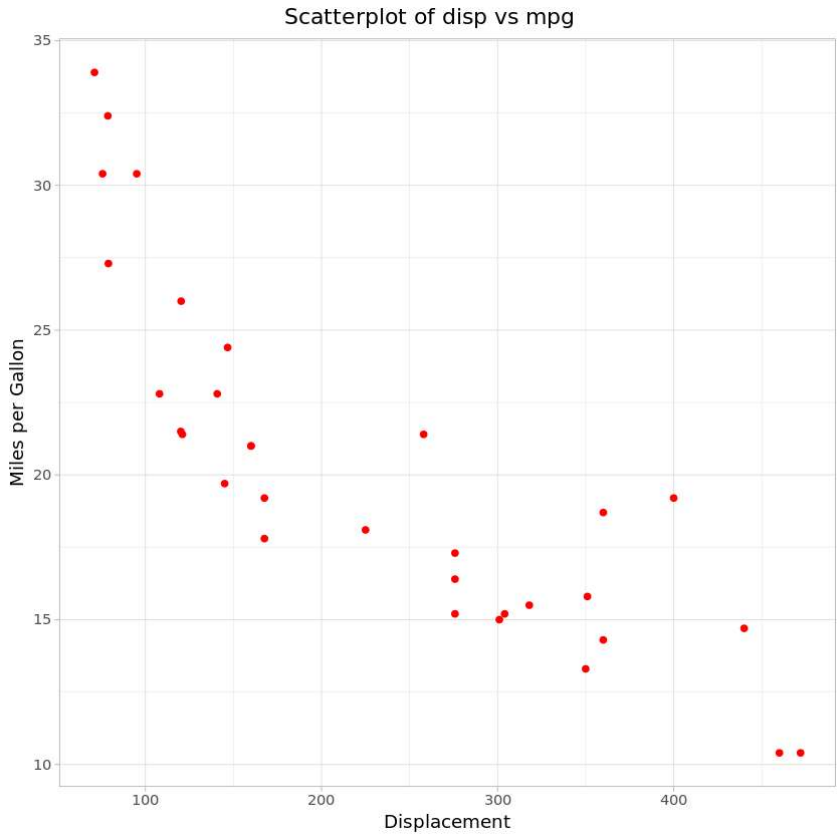
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

install.packages("ggalt") install.packages('GGally') install.packages('ggridges')

```
library('tidyverse')
library('ggalt')
library('GGally')
library('ggridges')
```

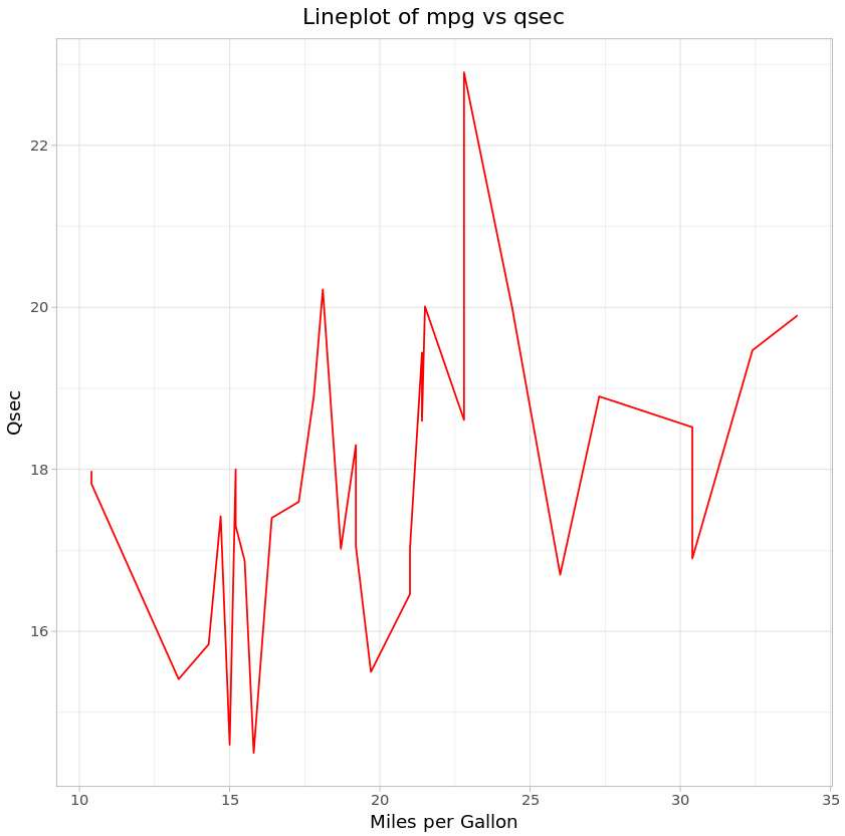
In [96]:

```
# scatterplot
ggplot(data=df) + geom_point(aes(x=disp, y=mpg), color = 'red') + theme_light() +
ggtitle("\t\t\t\tScatterplot of disp vs mpg") + xlab("Displacement") + ylab('Miles per Gall
```



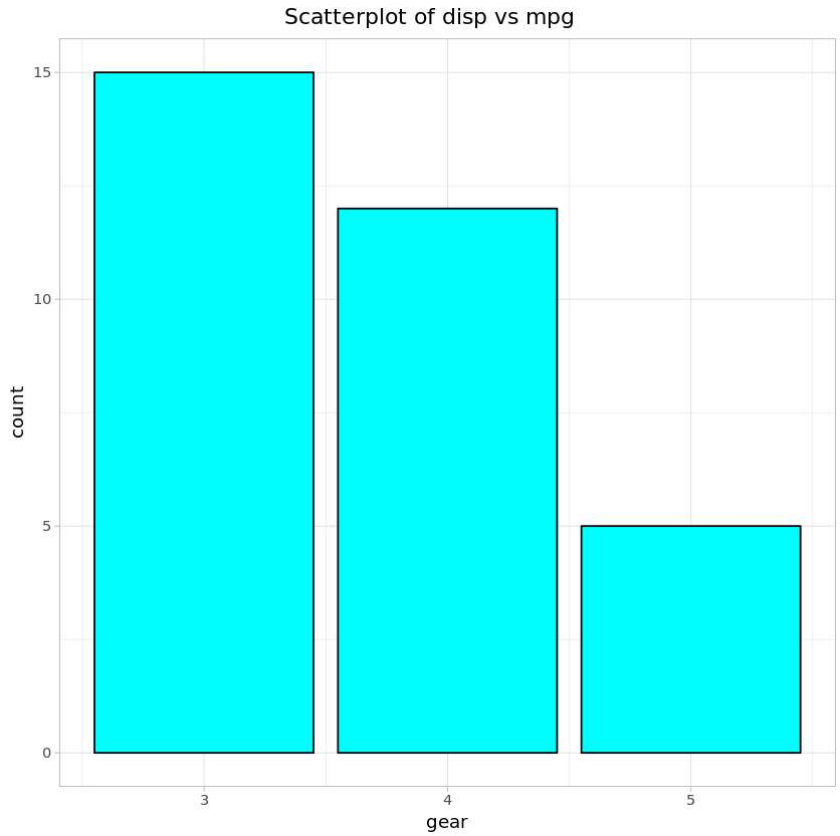
In [57]:

```
# Lineplot
ggplot(data=df) + geom_line(aes(x=mpg, y=qsec), color = 'red') + theme_light() +
ggtitle("\t\t\t\t   Lineplot of mpg vs qsec") + xlab("Miles per Gallon") + ylab('Qsec')
```



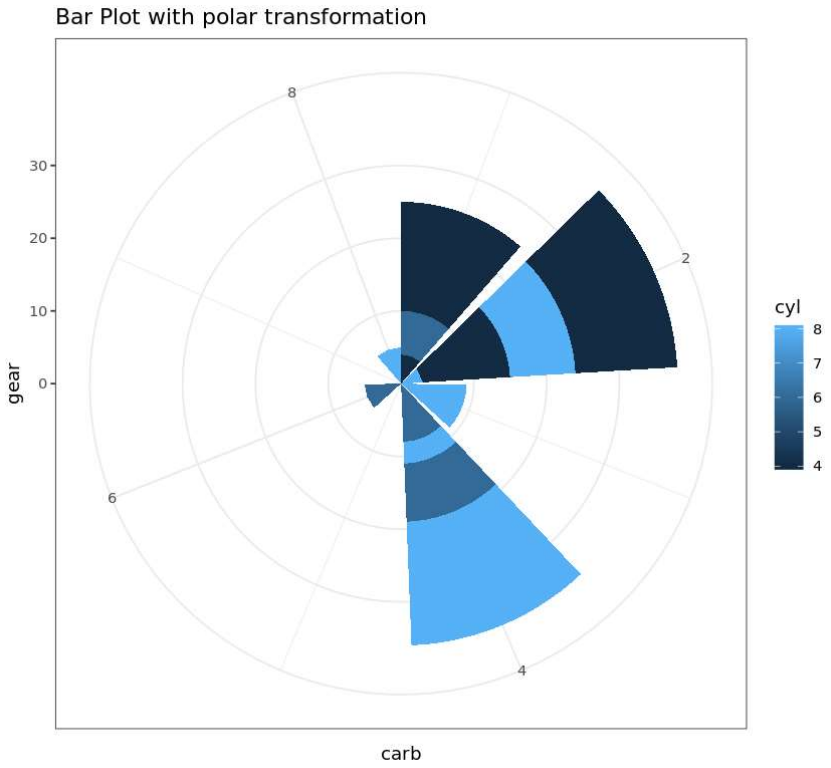
In [66]:

```
# Barplot -> univariate analysis
ggplot(data=df) + geom_bar(aes(gear), color = 'black', fill= 'cyan') + theme_light() +
ggtitle("\t\t\t\t Barplot of gear")
```



In [82]:

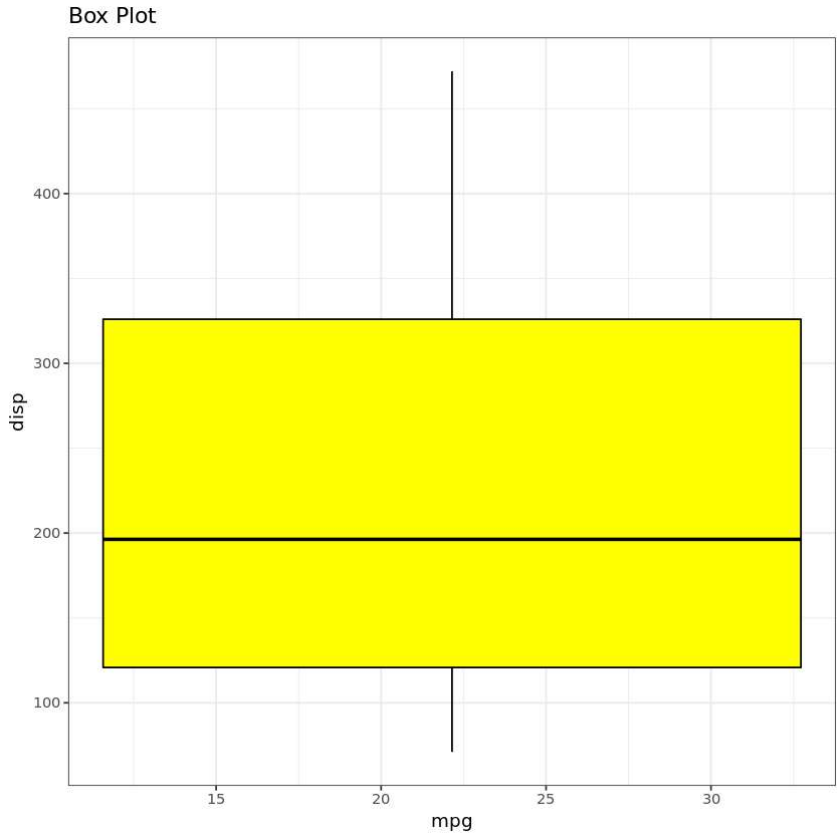
```
# polar transformation
ggplot(df) + geom_bar(aes(x = carb, y = gear, fill = cyl), stat = 'identity')
+ ggtitle("Bar Plot with polar transformation") + theme_bw() + coord_polar("x")
```



In [106]:

```
# boxplot
ggplot(df) + geom_boxplot(aes(x = mpg, y = disp), color='black', fill='yellow') + ggtitle("
```

```
Warning message:
“Continuous x aesthetic -- did you forget aes(group=...)?”
```



1. Modify the code to generate the Fibonacci sequence in the following ways.

a) Change the first two elements to 2 and 2.

b) Change the first two elements to 3 and 2.

c) Change the update rule from summing successive elements to taking differences of successive elements. For example, the third element is defined as the second element minus the first element, and so on.

d) Change the update rule so that each element is defined as the sum of the three preceding elements. Set the third element as 1 in order to start the process.

In [12]:

```
# Fibonacci function
fib <- function(n1,n2){
nterms = 7
count = 2
if(nterms <= 0) {
print("Plese enter a positive integer")
} else {
    if(nterms == 1) {
    print("Fibonacci sequence:")
    print(n1)
} else {
print("Fibonacci sequence:")
print(n1)
print(n2)
while(count < nterms) {
nth = n1 + n2
print(nth)
n1 = n2
n2 = nth
count = count + 1
}
}
}
}
```

In [13]:

```
fib(0,1)  # normal fibonnaci
```

```
[1] "Fibonacci sequence:"
[1] 0
[1] 1
[1] 1
[1] 2
[1] 3
[1] 5
[1] 8
```

In [14]:

```
fib(2,2) # changig first 2 elements to 2,2
```

```
[1] "Fibonacci sequence:"
[1] 2
[1] 2
[1] 4
[1] 6
[1] 10
[1] 16
[1] 26
```

In [15]:

```
fib(3,2) # changinf first 2 elements to 3,2
```

```
[1] "Fibonacci sequence:"
[1] 3
[1] 2
[1] 5
[1] 7
[1] 12
[1] 19
[1] 31
```

In [3]:

```
# Fibonacci function
fib_mod <- function(n1,n2){
nterms = 7
count = 2
if(nterms <= 0) {
print("Plese enter a positive integer")
} else {
    if(nterms == 1) {
    print("Fibonacci sequence:")
    print(n1)
} else {
print("Fibonacci sequence:")
print(n1)
print(n2)
while(count < nterms) {
nth = n2 - n1
print(nth)
n1 = n2
n2 = nth
count = count+1
}
}
}
}
fib_mod(0,1)
```

```
[1] "Fibonacci sequence:"
[1] 0
[1] 1
[1] 1
[1] 0
[1] -1
[1] -1
[1] 0
```