

DATA ANALYTICS – 4027

LAB-8

Name: Hari Krishna P

Reg No: 19BCE7675

DATE: 09/11/2021

Contents:

➤ **Graph Plotting**

Submitted to:

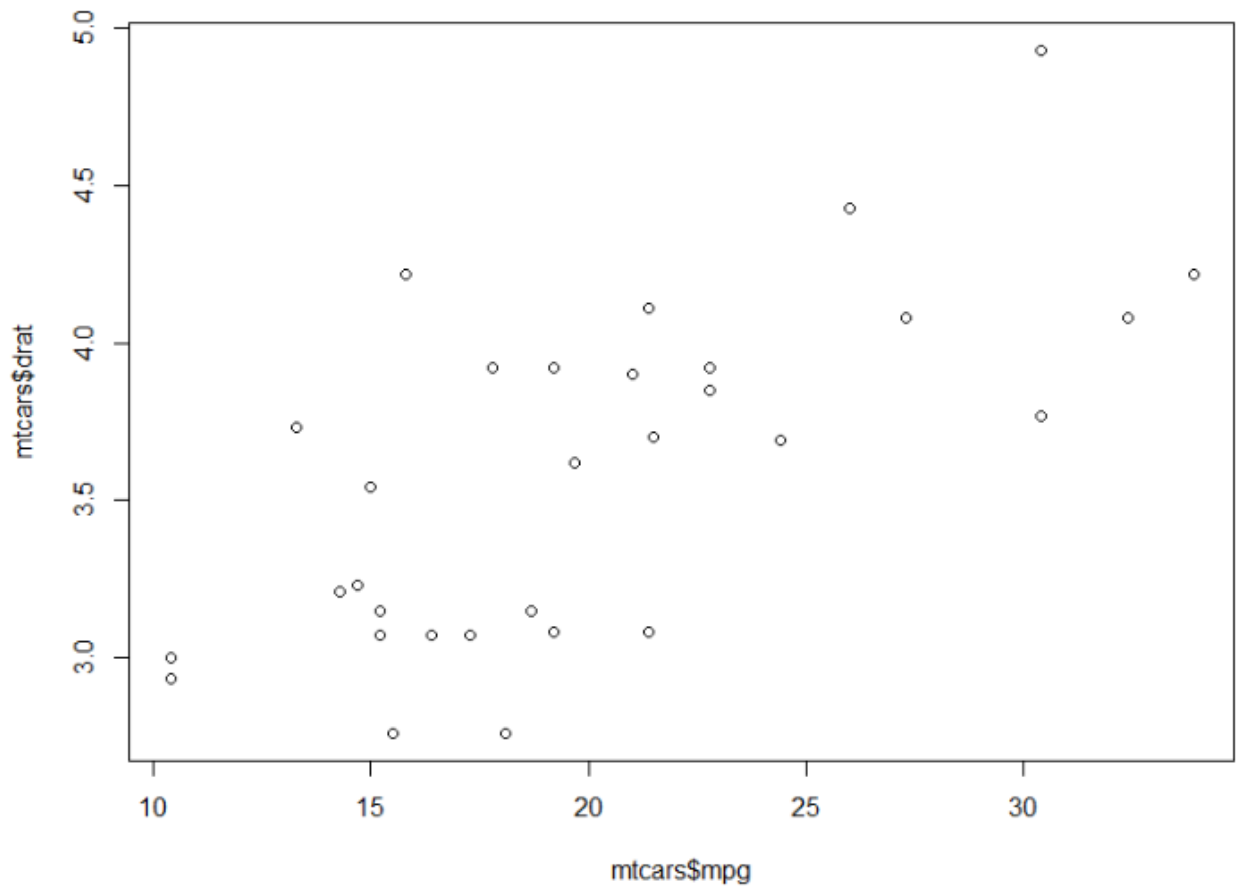
Prof . Hari Seetha

EX-8

Note: use the mtcars dataset, provided by R

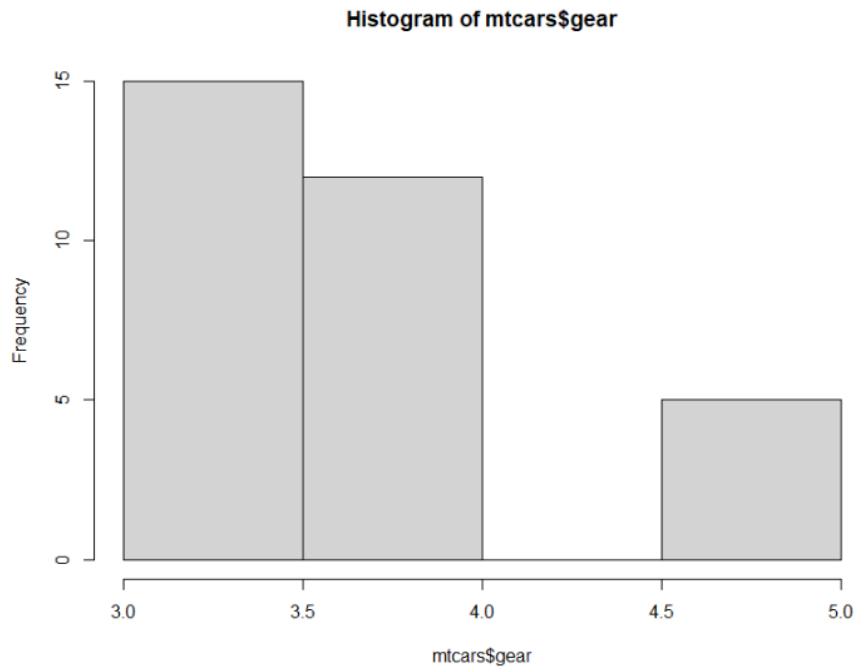
1. Plot Miles/(US) gallon versus Rear axle ratio by `plot(mpg, drat)` . On which axis does mpg appear?
 - a. mpg appears on the x axis
 - b. mpg appears on the y axis

`plot(mtcars$mpg,mtcars$drat)`

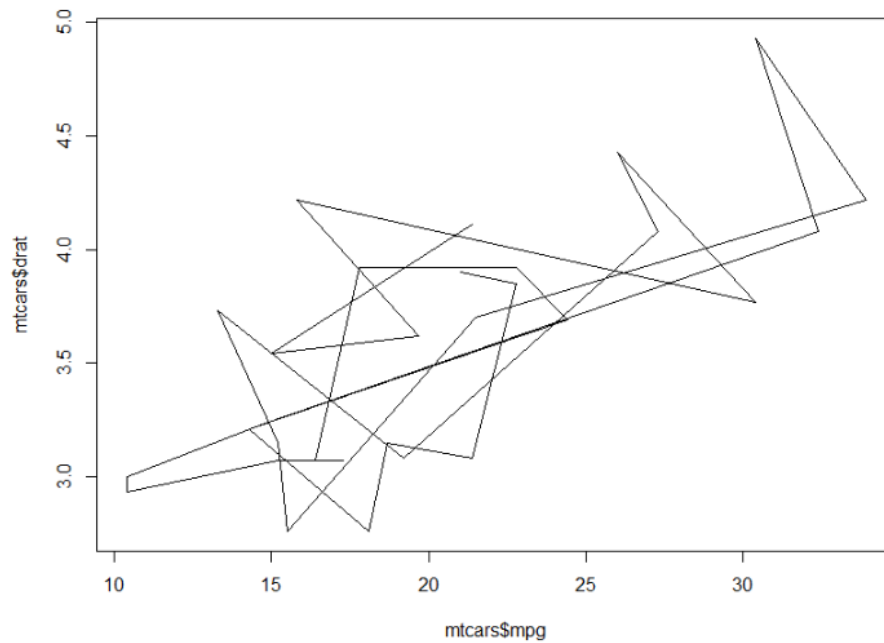


2. Produce a histogram with `hist(gear)` . What do you see?
 - a. frequencies
 - b. probability density

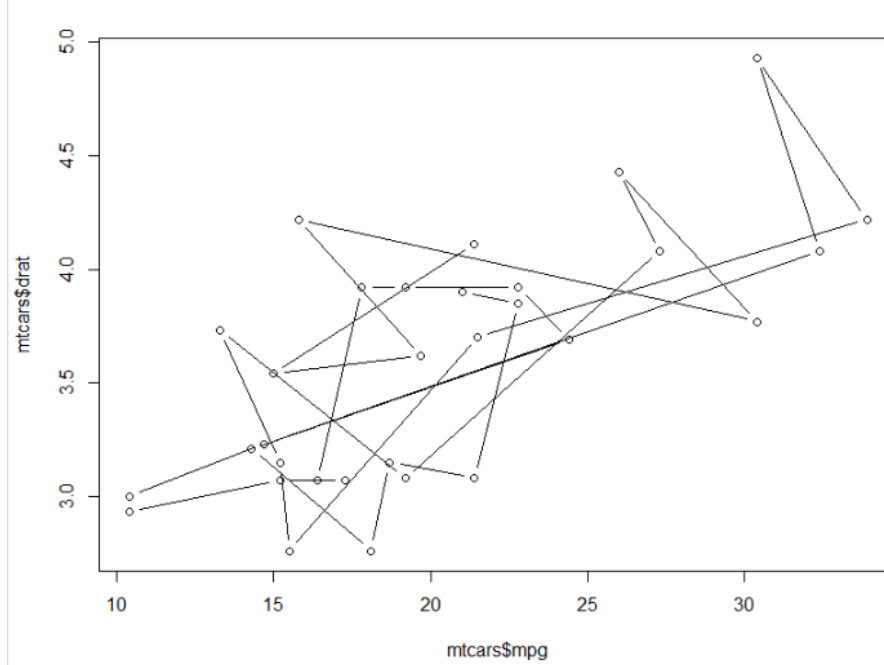
`hist(gear)`



3. Change type of visualization of our scatterplot in Exercise 1 `plot(mpg, drat, type="l")`. If we want to see lines what we have to type into **"l"**:
`plot(mpg, drat, type="l")`
-

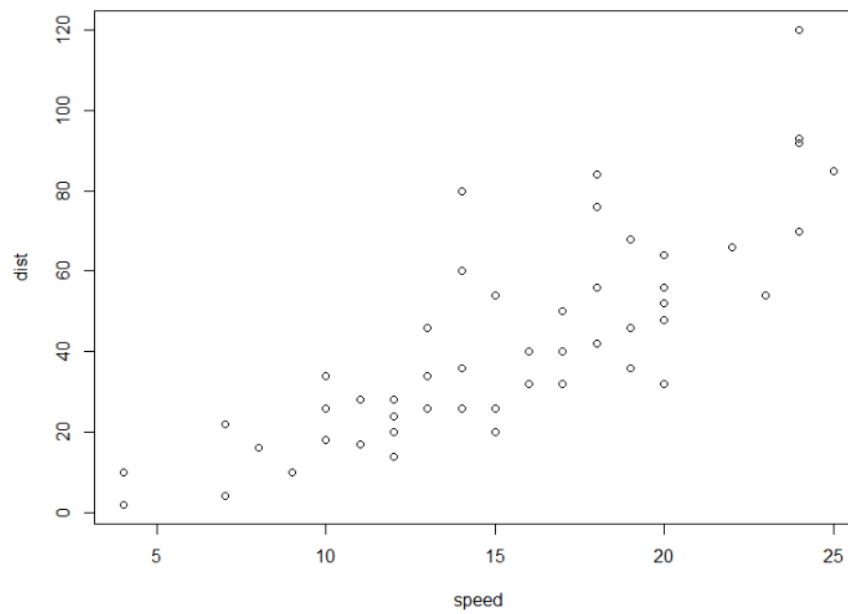


4. Now we want to see both point and lines in our plot. What we have to type into `plot(mpg, drat, type="b")`:



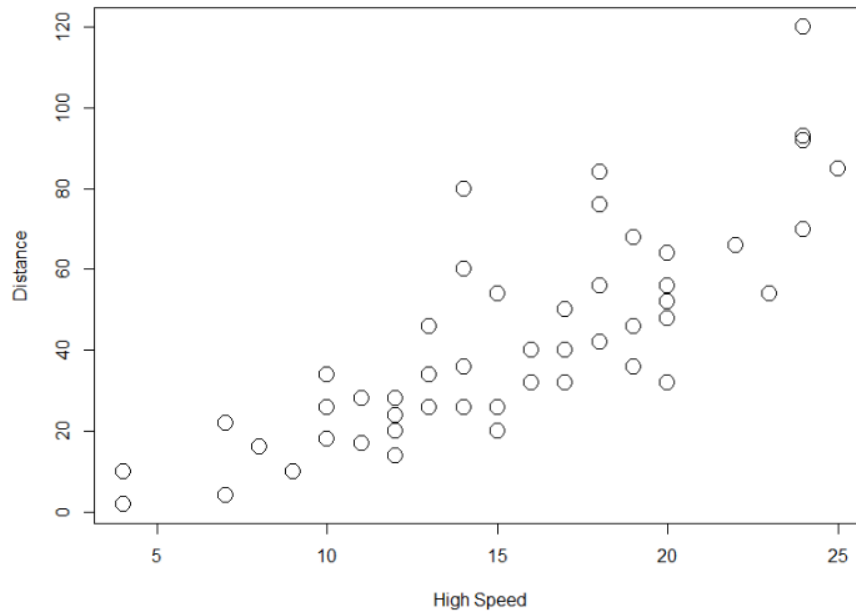
5. a) Load the `cars` dataset and create a scatterplot of the data.

`plot(cars)`



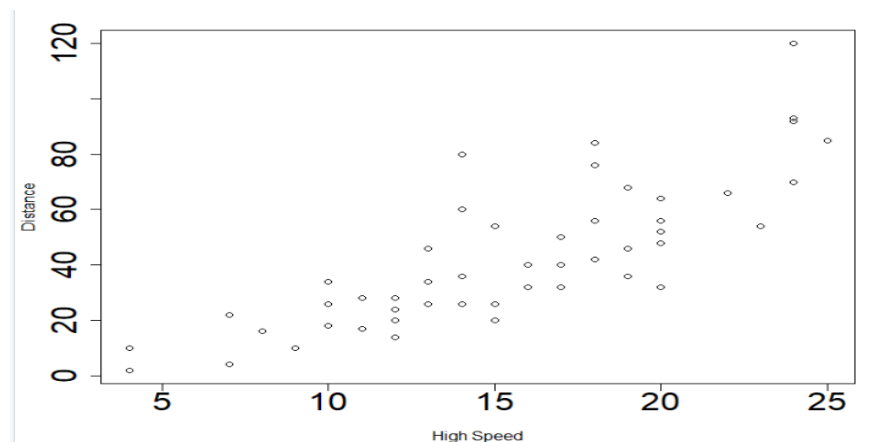
b) Using the argument `lab` of the function `plot` create a new scatterplot where the thickmarks of the x and y axis specify every integer.

`plot(cars, lab=c(20,10,6))`

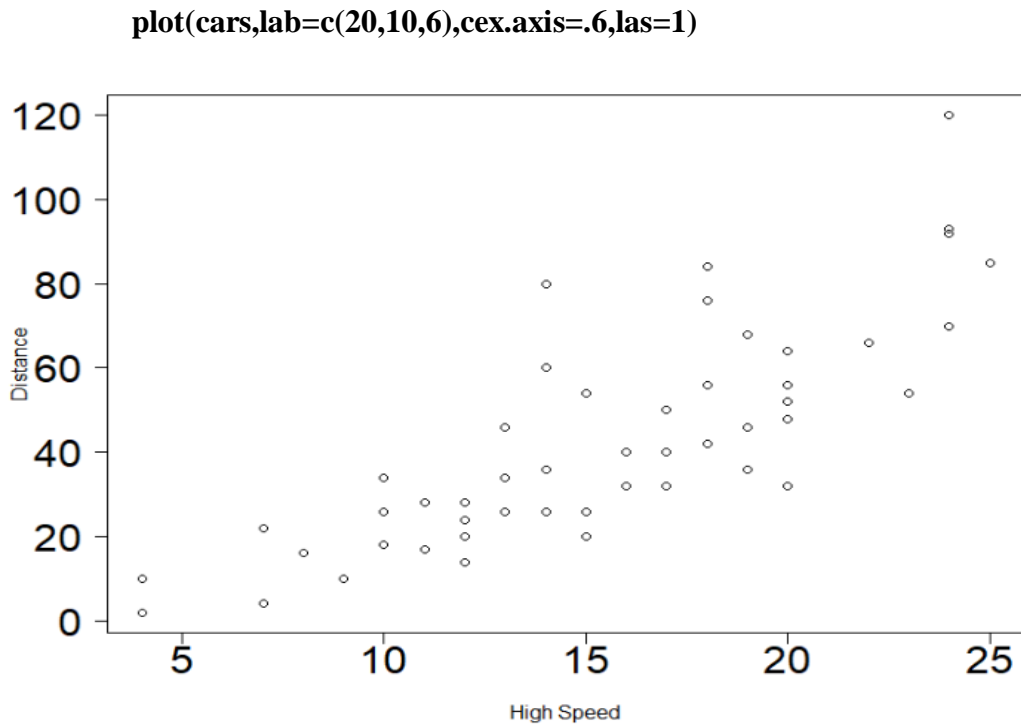


- The previous plot didn't show all the numbers associated to the new thickmarks, so we are going to fix them. Recreate the same plot from the previous question and using the argument `cex.axis` control the size of the numbers associated to the axes thickmarks so they can be small enough to be visible.

`plot(speed,dist,cex.axis=2,las = 1,xlab = "High Speed",ylab = "Distance")`

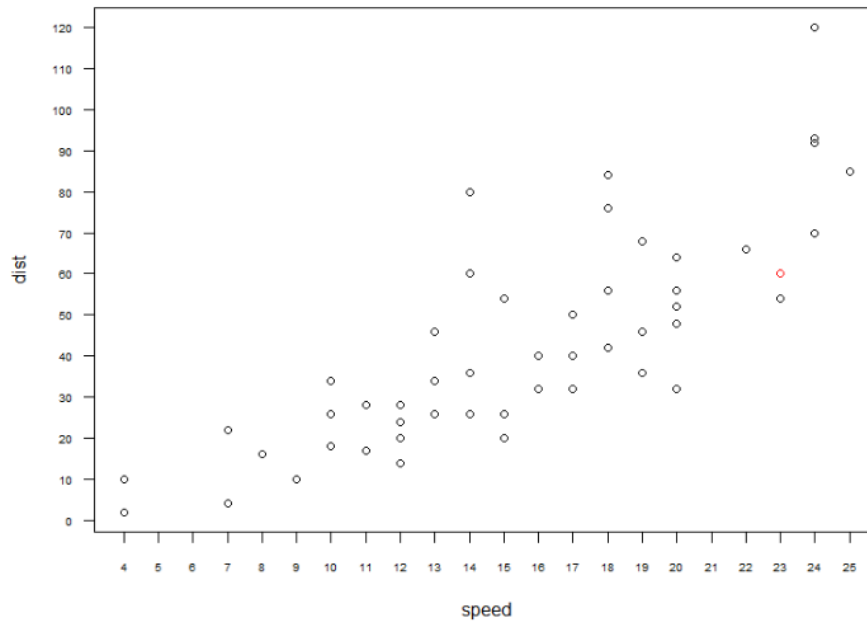


7. On the previous plot the numbers associated to the y-axis thickmarks aren't easy to read. Recreate the plot from the last exercise and use the argument `las` to change the orientation of the labels from vertical to horizontal.



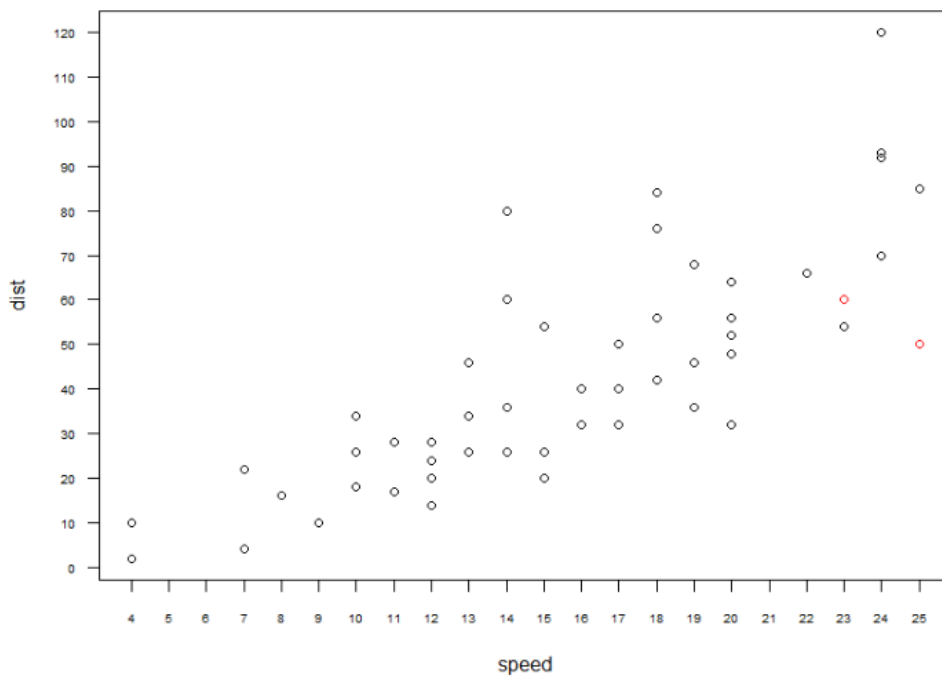
8. Suppose you want to add two new observations to the previous plot, but you want to identify them on the graph. Using the `points` function add the new observations to the last plot using red to identify them. The values of the new observation are speed = 23, 26 and dist = 60, 61.

```
points(x=c(23,26),y=c(60,61),col="red")
```



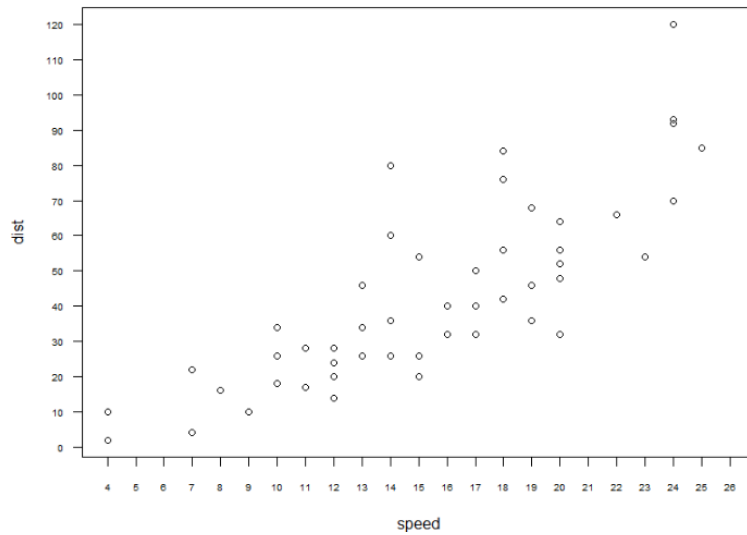
9. As you could see the previous plot doesn't show one of the new observations because is out the x-axis range.

`points(x=c(25,27),y=c(50,51),col="red")`



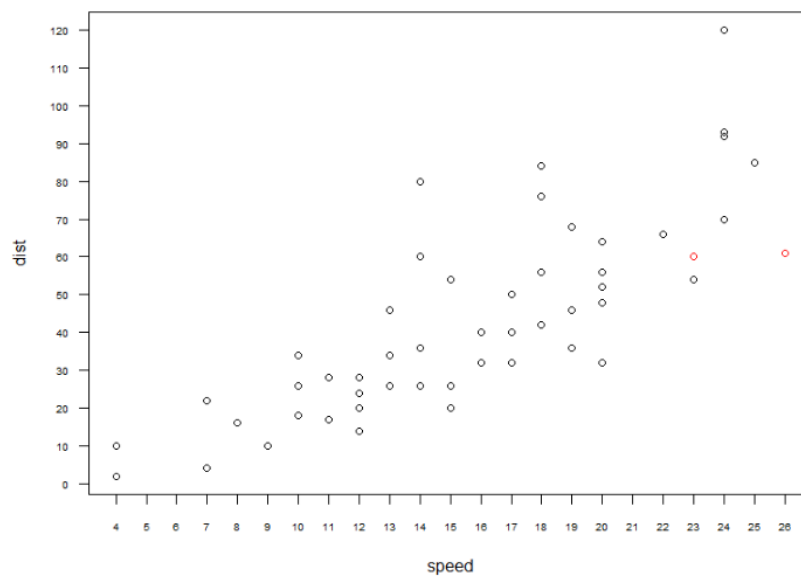
10. a) Create again the plot for the old observations with an x-axis range that includes all the values from 4 to 26.

```
plot(cars, lab=c(20,10,6), cex.axis=.6, las=1, xlim=c(4,26))
```



11. b) Add the two new observations using the `points` function.

```
points(x=c(23,26), y=c(60,61), col="red")
```

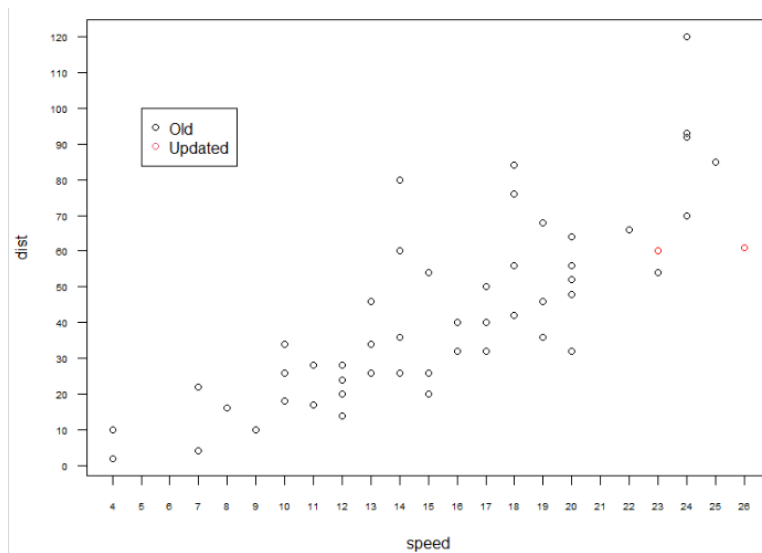


12. Add to the previous plot a legend on the top left corner that shows which color is assigned to old observations and which one to new ones.

```
plot(cars, lab=c(20,10,6), cex.axis=.6, las=1, xlim=c(4,26))
```

```
points(x=c(23,26), y=c(60,61), col="red")
```

```
legend(5,100,c("Old", "Updated"), col=1:2, pch=1)
```



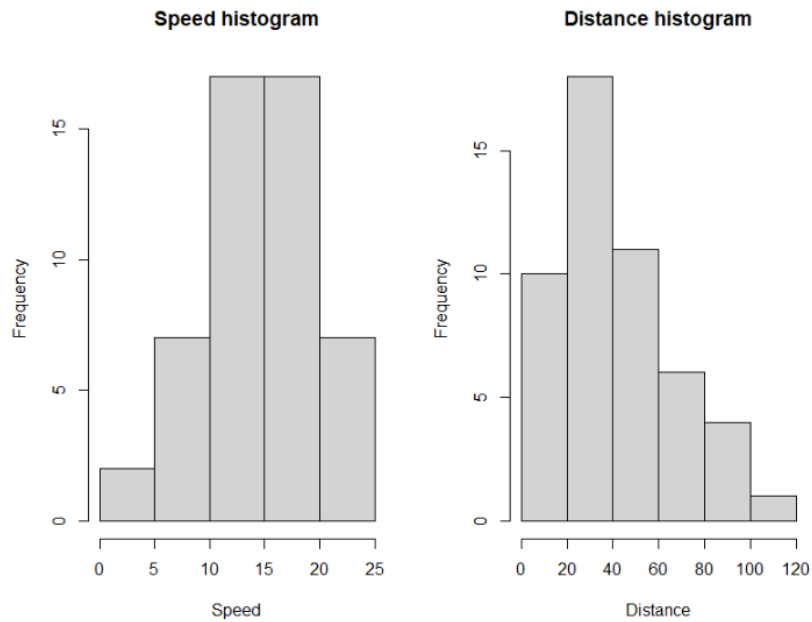
13. This exercise will test your skills to create more than one plot in the same layout. Using the functions `par` and `mfrow`.

Create on the same layout two histograms, one for each column of the `cars` data

```
par(mfrow=c(1,2))
```

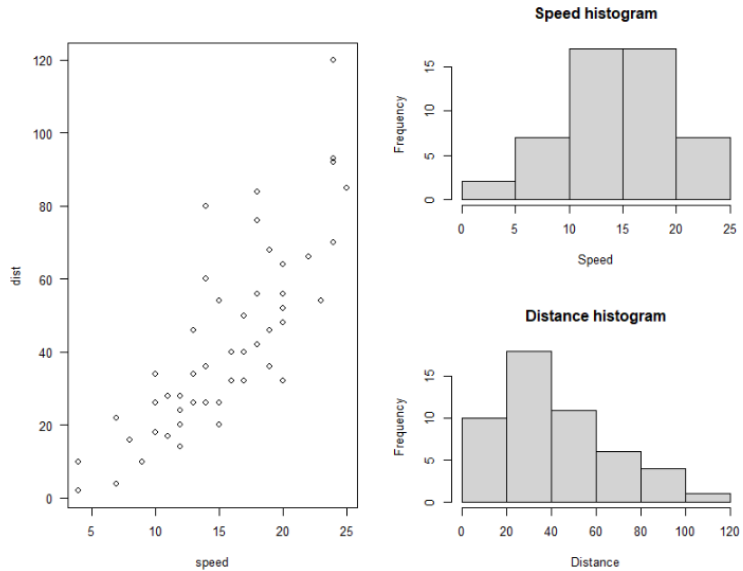
```
hist(cars[,1], main="Speed histogram", xlab="Speed")
```

```
hist(cars[,2], main="Distance histogram", xlab="Distance")
```



14. Using the function `layout` print on the same layout 3 plots, on the left side a scatterplot of cars, on the top right the histogram of the column speed of the data `cars`, and on the bottom right an histogram of the col

Refer <https://statisticsglobe.com/r-layout-function-arrange-plots/>



```
layout(matrix(c(1,1,2,3), ncol=2))
plot(cars, las=1)
hist(cars[,1], main="Speed histogram", xlab="Speed")
hist(cars[,2], main="Distance histogram", xlab="Distance")
```

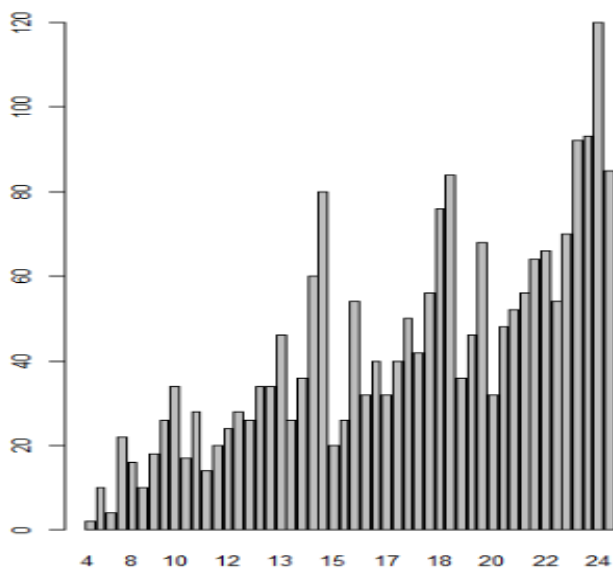
14. Generate Bar plot with name of the cars on X-Axis and hp on y-axis

```
layout(matrix(c(1,2,2,3), ncol=2))
```

```
plot(cars, las=1)
```

```
hist(cars[,1], main="Speed histogram", xlab="Speed")
```

```
hist(cars[,2], main="Distance histogram", xlab="Distance  
Cars
```



15. Generate a pie chart that shows cyl of cars present in the dataset using different colors, legend, title

```
x <- c(1,2,3,4)
```

```
mycars <- c("ASTONMARTIN", "BMW", "FERRARI", "MESARATTI")
```

```
colors <- c("RED", "BLUE", "GREEN", "YELLOW")
```

```
pie(x, label = mycars, main = "RACE CARS", col = colors)
```

```
legend("topright", mycars, fill = colors)
```

RACE CARS

