Prob A1 i160011 - This was the first assignment of the course

Probability and statitics (National University of Computer and Emerging Sciences)

Assignment #1
MT-206 Probability and Statistics

Date of submission: 26th Feb. 2020

Name: Maaz Muhammad
Role Number: i16-0011
Section: C

Problem #1.

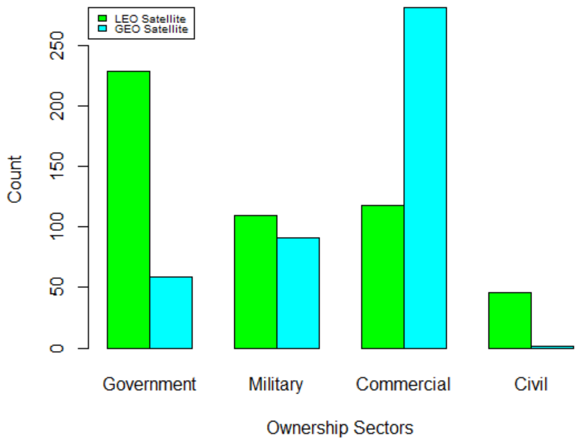
According to the Union of Concerned Scientists (www.ucsusa.org), as of November 2012, there were 502 low Earth orbit (LEO) and 432 geosynchronous orbit (GEO) satellites in space. Each satellite is owned by an entity in either the government, military, commercial, or civil sector. A breakdown of the number of satellites in orbit for each sector is displayed in the accompanying table. Use this information to construct pair of graphs (i. Rectangles, Pie Chart) that compare the ownership sectors of LEO and GEO satellites in orbit. What observations do you have about the data?

Ownership Sectors	LEO Satellites	GEO Satellites
Government	229	59
Military	109	91
Commercial	118	281
Civil	46	1

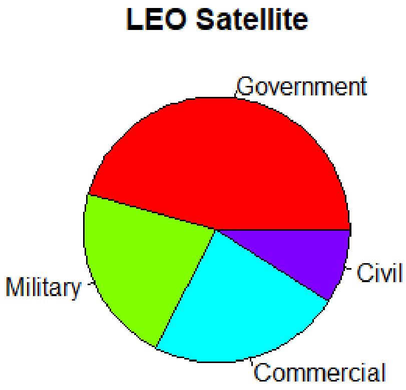
Solution 1:

R-Codes and Graphs

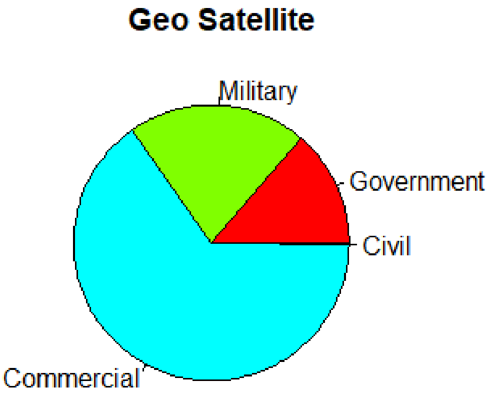
```
sects=c("Government","Military","Commercial","Civil");

countL=c(229,109,118,46);

countG=c(59,91,281,1);

colours=c("green","cyan")

top=c("LEO Satellite", "GEO Satellite");

dm=matrix(c(countL,countG),nrow=2,ncol=4,byrow = TRUE)

counts=rbind(countL,countG);

barplot(counts,beside=TRUE, width=.3, names.arg = sects, xlab = "Ownership
Sectors",ylab="Count",col = colours)

legend("topleft",top,cex =0.6,fill = colours)
```



```
x <- c(229,109,118,46);

labels <-c("Government","Military","Commercial","Civil");

pie(x, labels, main = "LEO Satellite", col = rainbow(length(x)),radius = 0.6)
```




```
x <- c(59,91,281,1);

labels <- c("Government","Military","Commercial","Civil");

pie(x, labels, main = "Geo Satellite", col = rainbow(length(x)),radius = 0.6)
```



Observations:

A large portion of Leo satellites are owned by the Government sector but most Geo Satellites are being used in the Commercial sector. This is mainly apparent from how they take considerable portions in the pie charts which give a good overview of the data.

Problem #2.

Do social robots walk or roll? According to the United Nations, social robots now outnumber industrial robots worldwide. A social (or service) robot is designed to entertain, educate, and care for human users. In a paper published by the International Conference on Social Robotics (Vol. 6414, 2010), design engineers investigated the trend in the design of social robots. Using a random sample of 106 social robots obtained through a web search, the engineers found that 63 were built with legs only, 20 with wheels only, 8 with both legs and wheels, and 15 with neither legs nor wheels.
a. What type of graph is used to describe the data?
b. Identify the variable measured for each of the 106 robot designs.
c. Use graph to identify the social robot design that is currently used the most.
d. Compute class relative frequencies for the different categories shown in the graph.

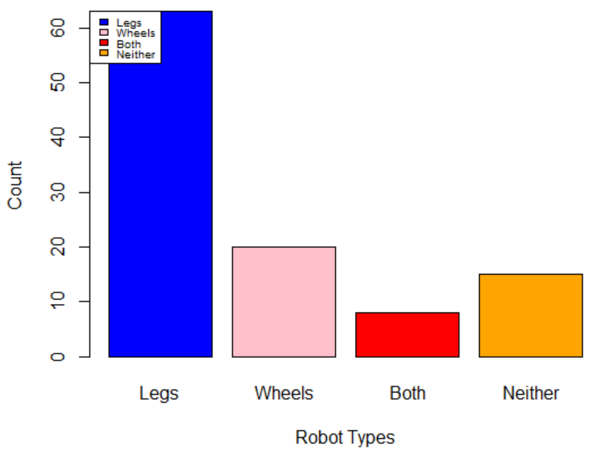
<u>Solution 2:</u>

a: Pie Chart

b: Qualitative Variable *(legs, wheels, both, neither)*

c: R-Code:

```
sects=c("Legs","Wheels","Both","Neither")
count=c(63,20,8,15)
colours = c("blue","pink","red","orange")
barplot(count,width=.3, names.arg = sects, xlab = "Robot Types",ylab="Count",col = colours)
legend("topleft",sects,cex =0.6,fill = colours)
```



d: R-Code:

```
data=c(63,20,8,15)
L&lt<-length(data)
B&lt<-max(data)
S&lt<-min(data)
R&lt<-(B-S)
K=1+3.322*log10(50)
```

```
H<-R/K
breaks<-round(seq(6,B+H,H),0)
breaks
Classes<-cut(data,breaks,right=F);
freq.dist=table(Classes)
fd1=cbind(freq.dist)
fd1
cum.freq.dist=cumsum(fd1)
cfd<-cbind(freq.dist,cum.freq.dist)
cfd
cf<-sum(freq.dist)
rela.freq.dist=freq.dist/cf
rfd<-cbind(freq.dist,cum.freq.dist,rela.freq.dist)
rfd
```

Output:

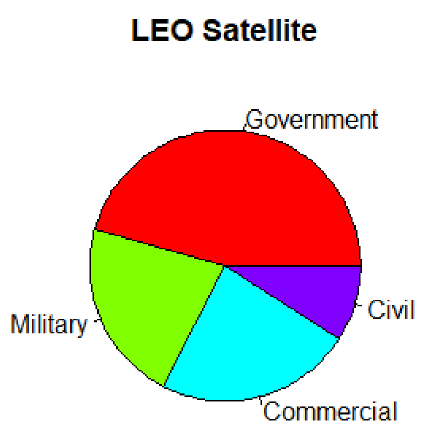
```
       freq.dist cum.freq.dist rela.freq.dist
[5,13)       1           1          0.2
[13,21)      2           3          0.5
[21,29)      0           3          0.0
[29,37)      0           3          0.0
[37,45)      0           3          0.0
[45,53)      0           3          0.0
[53,61)      0           3          0.0
[61,69)      1           4          0.2
```

Problem #3.

Use R Code to construct a pie chart to organize the data given in problem 2. What can you conclude?

<u>Solution 3:</u>

```
x <- c(229,109,118,46)
labels <-c("Government","Military","Commercial","Civil")
pie(x, labels, main = "LEO Satellite", col = rainbow(length(x)),radius = 0.6)
```



Problem #4.

(i) Take a suitable data set (related to CS) to construct the component bar diagram through R-

lang.

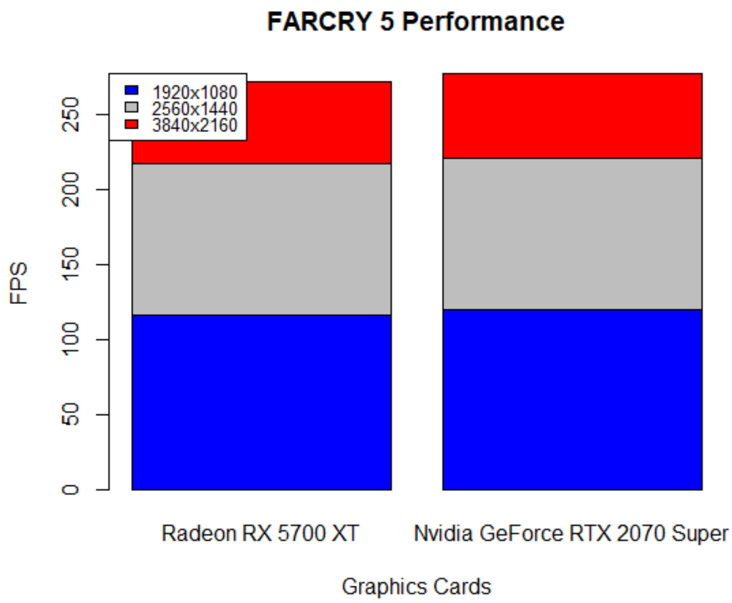
(ii) Use another data set (related to CS) to display the multiple bar diagram through R-lang.

Solution 4:

(i):

https://www.pcmag.com/comparisons/amd-radeon-rx-5700-xt-vs-nvidia-geforce-rtx-2070-super-which-graphics-card

```
colours = c("blue","grey","red")
card <- c("Radeon RX 5700 XT","Nvidia GeForce RTX 2070 Super")
resolution <- c("1920x1080","2560x1440","3840x2160")

# Create the matrix of the values.
Values <- matrix(c(116,120,101,101,55,56), nrow = 3, ncol = 2, byrow = TRUE)

# Create the bar chart
barplot(Values, main = "FARCRY 5 Performance",names.arg = card, xlab = "Graphics Cards", ylab = "FPS", col = colours)

# Add the legend to the chart
legend("topleft", resolution, cex = 0.8, fill = colours)
```



(ii):

https://www.codingdojo.com/blog/7-most-in-demand-programming-languages-of-2018

```
colours = c("blue","grey")
language <- c("Java","Python","JavaScript","C++","C#","PHP","Perl")
year <- c("2018","2017")
```

 # Create the matrix of the values.

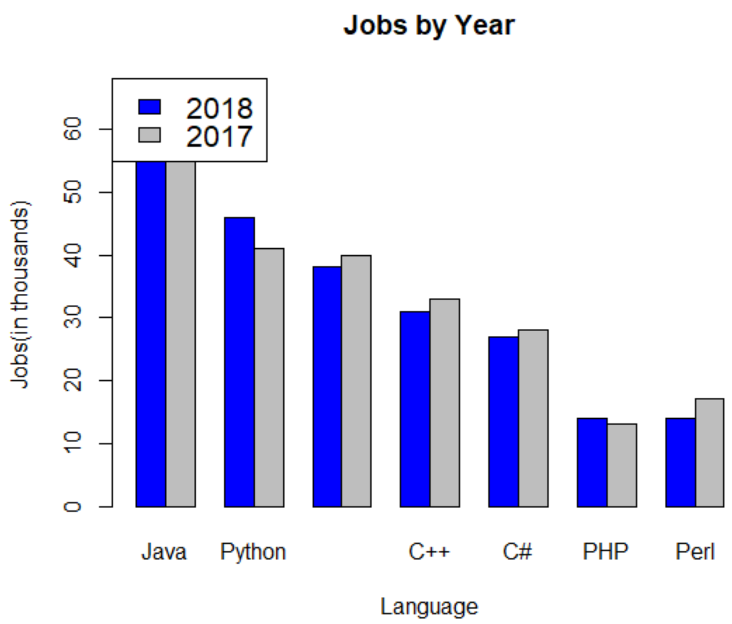
```
Values <- matrix(c(62,46,38,31,27,14,14,68,41,40,33,28,13,17), nrow = 2, ncol = 7, byrow = TRUE)
```

 # Create the bar chart

```
barplot(Values,beside=TRUE, main = "Jobs by Year",names.arg = language, xlab = "Language", ylab = "Jobs(in thousands)", col = colours)
```

 # Add the legend to the chart

```
legend("topleft", year, cex = 1.3, fill = colours)
```



Problem #5.

Collect the data (more than 50 values) on a targeted variable which must be continuous:

(i) Construct the frequency distribution.

(ii) List the mid-points for your frequency distribution, as well as the relative and cumulative

frequencies.

(iii)Draw a histogram depicting the data from the frequency table in # (ii) above and

superimpose a frequency polygon on top of a histogram

(iv) Show the histogram and density curve in one graph.

(v) Show the histogram, density curve and frequency polygon in one graph.

(vi) Also construct an Ogive for the data considered before through R.

Solution 5:

(Note: The following data was taken personally. It is the time taken to run a simple program in Go-lang)

data<-c(1.468,1.594,1.541,1.471,1.473,1.487,1.643,1.507,1.457,1.649,1.620,1.507,1.618,1.566,1.388,1.566,1.494,1.288,1.582,1.517,1.539,1.780,1.423,1.620,1.620,1.525,1.776,1.403,1.487,1.517,1.490,1.528,1.516,1.520,1.436,1.459,1.522,1.549,1.513,1.495,1.505,1.533,1.570,1.503,1.489,1.590,1.471,1.427,1.461,1.380)

(i)

```
L<-length(data)
B<-max(data)
S<-min(data)
R<-(B-S)
K=1+3.322*log10(50)
H<-R/K
breaks<-seq(1,B+H,H)
Classes<-cut(data,breaks,right=F);
freq.dist=table(Classes)
f.d1=cbind(freq.dist)
f.d1
```

Output:

```
  freq.dist
[1,1.07)          0
[1.07,1.15)       0
[1.15,1.22)       0
[1.22,1.3)        1
[1.3,1.37)        0
[1.37,1.44)       6
[1.44,1.52)      21
[1.52,1.59)      13
[1.59,1.67)       7
[1.67,1.74)       0
[1.74,1.81)       2
```

(ii)

```
cum.freq.dist=cumsum(f.d1)
c.f.d<-cbind(freq.dist,cum.freq.dist)
c.f.d
cf<-sum(freq.dist)
rela.freq.dist=freq.dist/cf
r.f.d<-cbind(freq.dist,cum.freq.dist,rela.freq.dist)
r.f.d
i<-1:7
MidPoints<-round(c(breaks[i]+(H/2)),0)
m.p<-cbind(freq.dist,cum.freq.dist,rela.freq.dist,MidPoints)
```

m.p

Output:

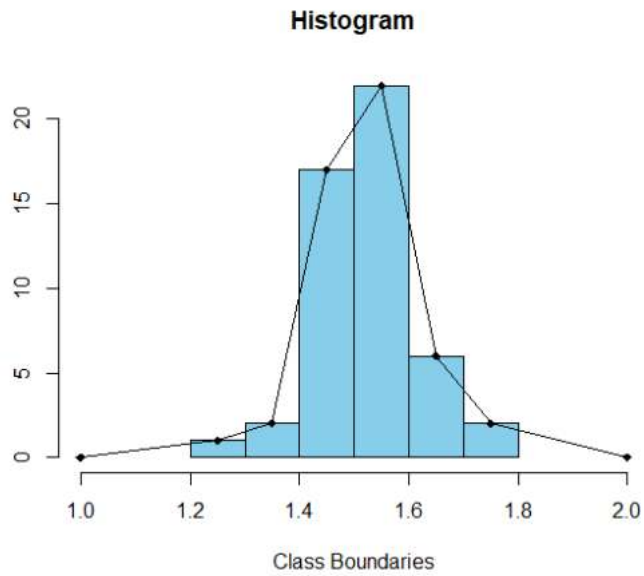
```
freq.dist cum.freq.dist rela.freq.dist
[1,1.07)          0            0             0.00
[1.07,1.15)       0            0             0.00
[1.15,1.22)       0            0             0.00
[1.22,1.3)        1            1             0.02
[1.3,1.37)        0            1             0.00
[1.37,1.44)       6            7             0.12
[1.44,1.52)      21           28             0.42
[1.52,1.59)      13           41             0.26
[1.59,1.67)       7           48             0.14
[1.67,1.74)       0           48             0.00
[1.74,1.81)       2           50             0.04
> i<-1:7
```

(iii)

Histogram<-hist(data, main="Histogram", xlab="Class Boundaries",ylab="", col="skyblue", xlim=c(1,2))

Histogram<-lines(c(1,Histogram$mids, 2),c(0,Histogram$counts,0),lwd=1.5,pch=18,type="o",xlim=c(1,2))

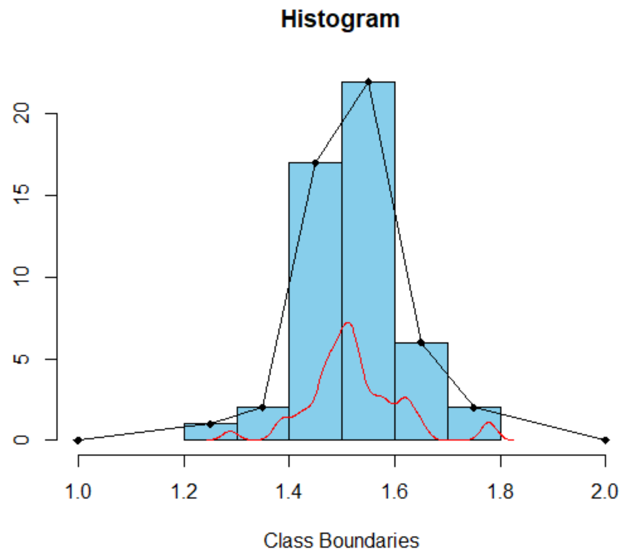
Output:



(iv)

lines(density(data,adjust=0.5),xlim=c(1,2),col="red")

Output:



(v)

Output: