RStudio Code code Log

Copy and paste the code below into the RStudio console to create some graphs. Copy one line at a time. Lines are separated by spaces. Code is in blue.

- **Bar plot**

  - Frequency and Relative Frequency code

```
y  <- c(12,2,1,2,4,5,2,1,1)

x<- c("Back","Wrist","Elbow","Hip","Shoulder","Knee","Hand","Groin","Neck")

barplot(y,names.arg=x,xlab="Categories",ylab="Frequency",col="blue",border="black")
```

- **Pareto Chart**
  - Frequency code

```
y  <- c(12,2,1,2,4,5,2,1,1)

names(y) <- c("Back","Wrist","Elbow","Hip","Shoulder","Knee","Hand","Groin","Neck")

library(qcc)

pareto.chart(y, xlab = "Categories", ylab="Frequency", col=heat.colors(length(y)), cumperc =
seq(0, 100, by = 20),ylab2 = "Cumulative Percentage", main = "Pareto Graph")
```

  - Relative Frequency

```
y  <- c(0.4,0.0667,0.033,0.0667,0.133,0.1667,0.0667,0.0333,0.0333)

names(y) <- c("Back","Wrist","Elbow","Hip","Shoulder","Knee","Hand","Groin","Neck")

library(qcc)

pareto.chart(y, xlab = "Categories", ylab="Relative Frequency", col=heat.colors(length(y)),
cumperc = seq(0, 100, by = 20),ylab2 = "Cumulative Percentage", main = "Pareto Graph")
```

- **Pie Chart**

```
slices <- c(12, 27,20, 8, 20,12)

lbls <- c("Not a high school graduate", "High school diploma", "Some college no degree",
"Assciates degree", "Bachelor's degree","Graduate or professional degree")

pct <- round(slices/sum(slices)*100)

lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels

pie(slices,labels = lbls, col=rainbow(length(lbls)),main="Pie Chart")
```

- **Histogram**
  - Frequency

```
x<-c(155.40, 208.39, 204.49,210.20, 322.61, 193.42, 207.79, 134.66,125.76, 190.72, 236.91,
127.37, 124.67, 121.25,243.80, 151.46, 124.85, 167.65,187.23, 260.00, 140.06, 148.02,
131.76,229.18, 162.46, 216.19, 206.29, 120.53, 125.00, 65.00,210.25, 183.16, 143.52,256.70
,271.20 ,211.09, 105.00, 227.59, 105.23, 147.70,209.29, 256.69, 224.35, 125.21, 206.01,150.56,
262.99, 223.99, 208.40, 123.41)

hist(x, main = "Histogram")
```

  - Relative frequency

```
x<-c(155.40, 208.39, 204.49,210.20, 322.61, 193.42, 207.79, 134.66,125.76, 190.72, 236.91,
127.37, 124.67, 121.25,243.80, 151.46, 124.85, 167.65,187.23, 260.00, 140.06, 148.02,
131.76,229.18, 162.46, 216.19, 206.29, 120.53, 125.00, 65.00,210.25, 183.16, 143.52,256.70
,271.20 ,211.09, 105.00, 227.59, 105.23, 147.70,209.29, 256.69, 224.35, 125.21, 206.01,150.56,
262.99, 223.99, 208.40, 123.41)

hist(x,probability = TRUE,main = "Histogram")
```

- **Mean**

1.To find the mean in R you first have to input the data values
2. Now depending on the type of mean (population or sample) you are looking for you will choose the code accordingly

• Working in the console of RStudio use the following code. Copy the code below one at a time. You can adjust the data values in “x”.

  - Population Mean/ sample mean

```
x<-c(1,2,3,5,6,7,9)

mean(x)
```

- Median

```
x<-c(1,2,3,5,6,7,9)
```

```
median(x)
```

- **Standard Deviation**

  - Population

```
x<-c(1,2,3,5,6,7,9)

sqrt(sum((x-mean(x))^2)/length(x))
```

  - Sample

```
x<-c(1,2,3,5,6,7,9)

sd(x)
```

- **Mean of group data**

1. put in midpoints using code below.

```
x<-c(11.5,14.5,17.5,20.5)
```

2. include frequency using code below

```
f<-c(4,12,30,14)
```

3. find the mean using code below.

```
sum(x*f)/sum(f)
```

- **Weighted mean**

1. put in weights using code below.

```
w<-c(5,3,4)
```

2. include any data values

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

3. find the mean using code below.

```
sum(w*x)/sum(w)
```

- **Standard deviation of grouped data**

1. put in midpoints using code below.

```
x<-c(11.5,14.5,17.5,20.5)
```

2. include frequency using code below

```
f<-c(4,12,30,14)
```

3. find the mean using code below.

```
mean<-sum(x*f)/sum(f)
```

4. Find the standard deviation using the code below.

```
sqrt(sum((x-mean)^2*f)/(sum(f)-1))
```

- **Five -Number Summary**

```
x<-c(24,16,22,28,26,21,24)

fivenum(x)
```

- **Boxplot**

```
x<-c(24,16,22,28,26,21,24)

boxplot(x,horizontal=TRUE)
```

- **Correlation Coefficient**

```
x<-c(24,16,22,28,26,21,24)
Y<-c(1,2,3,4,5,6,7)

cor(x,y)
```

- **Scatter plot R code:**

```
x <- c(12, 15, 5, 17, 8, 10, 14, 16, 16, 9)

y <- c(26.6, 29.3, 10.2, 34.7, 15.8, 22.1, 27.6, 34.9, 32.6, 22.0)

plot(x, y)
```

- **Least-Squares Regression Line**

```
x <- c(12, 15, 5, 17, 8, 10, 14, 16, 16, 9)

y <- c(26.6, 29.3, 10.2, 34.7, 15.8, 22.1, 27.6, 34.9, 32.6, 22.0)

plot(x, y)

model = lm(y ~ x)

abline(model)
```

- **Sum of the Square Residual**

```
y<-c(18,13,9,6,4)

x<-c(1,3,3,6,7)

y_hat=(-2.1675*x+18.67)

y_hat

sum((y-y_hat)^2)
```

- **Discrete Probability Distribution**

R code (Plot Discrete Probability Distribution):

```
 x<-c(0,1,2,3)

y<-c(0.01,0.1,0.38,0.51)

plot(x,y,type="h", lwd=2,col="blue",ylab="p")
```

Mean and standard deviation:

```
x<-c(0,1,2,3)

y<-c(0.01,0.1,0.38,0.51)

weighted.mean(x, y)

library("Weighted.Desc.Stat")

w.sd(x, y)
```

- **Binomial:**

Finding probabilities. Copy and paste the R code on the right.

Probability	R code:
P (x = 5)	dbinom(5, n, p)
P (x ≤ 5)	sum(dbinom(c(0:5),n,p))
P(x<5)	sum(dbinom(c(0:4),n,p))
P(3≤ x≤ 5)	sum(dbinom(c(3:5), n, p))
P(x ≥ 5)	1-sum(dbinom(c(0:4),n,p))
P(x > 5)	1-sum(dbinom(c(0:5),n,p))

- **Normal distribution/ same code 8.1 just be careful with the standard deviation**

Finding the area under the curve

Type of area	R code:
Area to the left	
P(x < a) or P (x ≤ a)	pnorm(a, μ, σ)
Area to the right	
P(x > b) or P(x ≥ b)	pnorm(b, μ, σ, lower.tail=FALSE)
Area in between	
P(a < x < b) or P(a≤ x≤ b)	pnorm(b, μ, σ) - pnorm(a, μ, σ)

Finding the x-value

Percentile, k qnorm(k , μ, σ)

Finding a z-value (area to the right

```
qnorm(k, lower.tail=FALSE)
```

- Chi-Square Test for Independence

```
x<- matrix(c(600, 720, 93, 63, 142, 51, 112, 355, 119, 144, 459, 127), ncol = 4,byrow=TRUE)

 colnames(x)<-c("Married", "Widowed", "Divorced/Separated", "Never Married")

 rownames(x)<-c("Very Happy", "Pretty Happy", "Not Too Happy")

 x<-as.table(x)

chisq.test(x)
```

- Expected count

```
x<- matrix(c(600, 720, 93, 63, 142, 51, 112, 355, 119, 144, 459, 127), ncol = 4,byrow=TRUE)
```

```
 colnames(x)<-c("Married", "Widowed", "Divorced/Separated", "Never Married")

 rownames(x)<-c("Very Happy", "Pretty Happy", "Not Too Happy")

 x<-as.table(x)

c<-chisq.test(x)

round(c$expected,2)
```

- **One-Way ANOVA**
  Example R code
  1. create a data frame with one categorical and one numerical column.
     * For categorical data use “”, see code below:
     ```
      x<-c("Face-Face","Face-Face","Face-Face","Face-Face","Face-Face","Face-Face","Face-Face","Face-Face","Face-Face","Hybrid","Hybrid","Hybrid","Hybrid","Hybrid","Hybrid","Hybrid","Hybrid","Hybrid","Online","Online","Online","Online","Online","Online","Online")
     ```

     * For numerical data use the code below:
     ```
     y<-c(88, 76, 73, 65, 72, 93, 61,85,89,74,81, 55, 90, 68, 77,56, 75, 58,64,60,70,66, 91,91,64)
     ```
  2. Create a table in r
     ```
     df<-data.frame(x,y)
     ```
  3. Use the following to set your model and apply one-way ANOVA in R
     ```
     anova<-aov(y~x,data=df)
     ```
  4. Print your summary
     ```
     summary(anova)
     ```