Statistics 147 LAB #2

Summer 2020; 10 pts

NAME: Wesley Chang **ID: (last 4 #s only)** 0996

This lab is designed to give the student practice generating charts and graphs, descriptive statistics, and to solve problems involving the Binomial and Normal distributions using **SAS** and **R**.

Data Files needed for this lab: cartest1.dat and ages.dat. They are available on iLearn under **Data Files**.

ages.dat

```
Father  Mother
   23      20
   32      28
   35      33
   19      21
   20      18
   23      22
   23      21
   26      24
   25      26
   25      24
   25      25
   29      27
   24      22
   26      18
   24      21
   39      32
   31      30
   22      27
   25      24
   29      28
```

cartest1.dat

```
  Car BrandA BrandB
   1    125    133
   2     64     65
   3     94    103
   4     38     37
   5     90    102
   6    106    115
```

1 SAS

1.1 Descriptive Statistics: Graphical & Numerical Methods

REMINDER: Each time you make a change to your SAS program, remember to save the file and then execute it again to update the output.

1. Using SAS and the data file **ages.dat**, generate a horizontal bar chart and a vertical bar chart (histogram), with midpoints from 15 to 35 in increments of 5, for the variable **Mother**.

   NOTE: One can use **PROC CONTENTS** to get information about the data set. We will add this to our program, just to see how it works.

   PROC CHART or **PROC GCHART** can be used to create many different types of charts. **VBAR (VBAR3D)** creates a vertical bar chart, **HBAR (HBAR3D)** creates a horizontal bar chart, **BLOCK** creates a block chart, **PIE** creates a pie chart and **STAR** creates a star chart. In each case, the variable listed determines the values that label the bars or sections. Options are used to control the kind of statistics presented and any grouping: (See Lecture Notes for a listing of the available options.)

   NOTE: The actual data starts on line 2, so you will need to use a **firstobs** command in your infile statement.

   You will need the following code: (For this lab, you don't need to type in all the comments, but I would recommend doing it at some point!)

```
/* Set up format of the output */
options nocenter ps = 55 nocenter ls = 78 nodate nonumber formdlim='*';
 /* ls = linesize,  ps = pagesize
   nocenter      justifies the output so it is not centered on the page
   nodate        suppresses printing of today's date on each page of output
   nonumber      suppresses printing of page number on each page of output
   formdlim      overrides the internal page breaks and replaces them
                           with the designated symbol*/


/* Use DM to clear all windows except the editor window */
DM log "odsresults; clear; out; clear; log; clear;";
ods graphics off;

/* Create titles */
title1 'Statistics 147, Summer 2020';
title2 'Lab 2';
title3 'Your Name Goes Here';

/* Create SAS data set called age_level
   Open data file ages.dat using an infile statement. Note the actual data begins on line 2 */

data age_level;
   /* Be sure to change the path to the location of your data file */
   infile 'PATH\TO\YOUR\FILE\ages.dat' firstobs = 2;

   /* Input the name of the variables
      Father   age of the father
      Mother   age of the mother     */
   input Father Mother;
/*Finish the data step with a run statement*/
run;
```

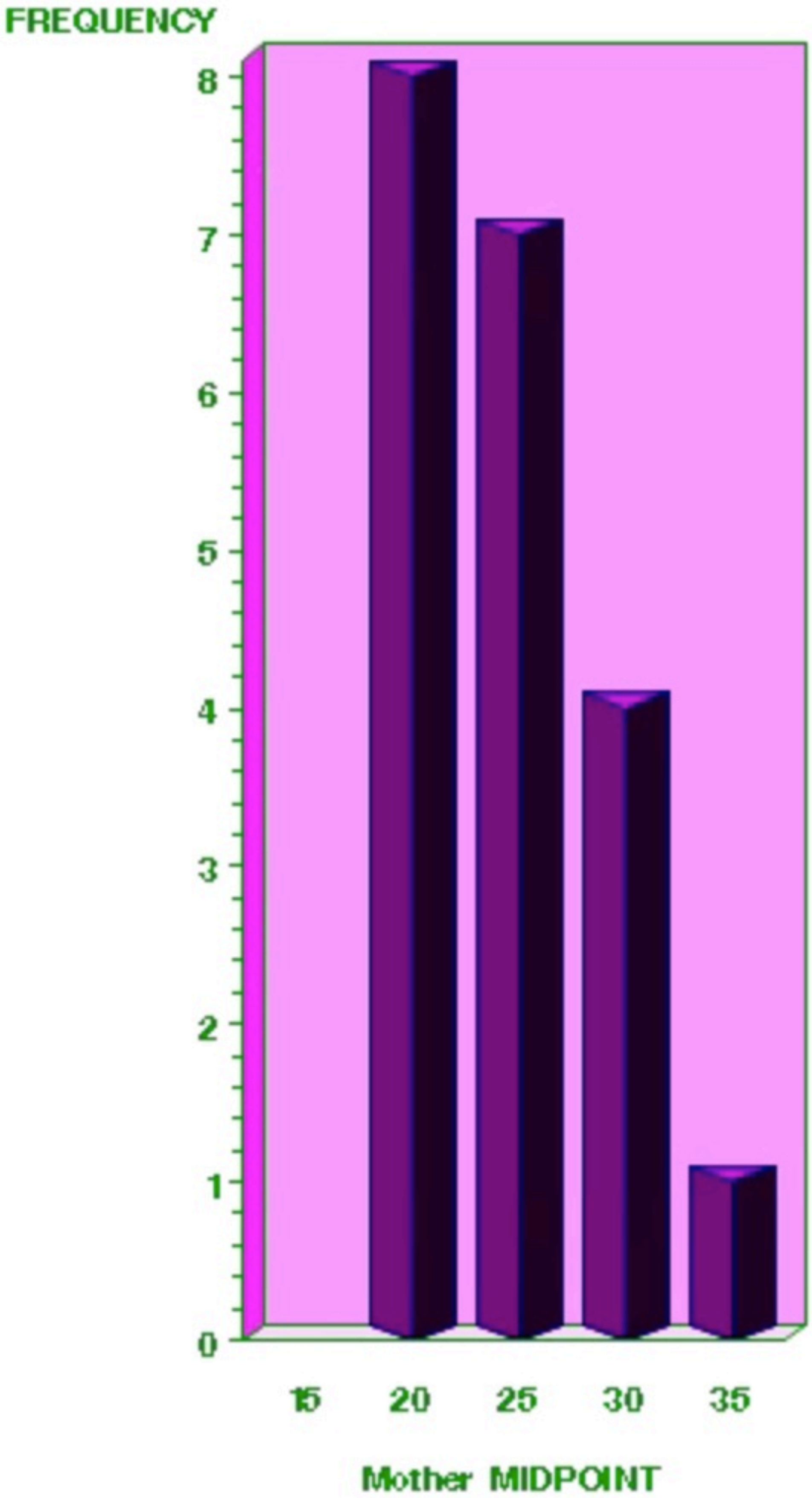
```
/* Print the data as a check  */
proc print data = age_level;
/*Add a title for the question number*/
    title4 'Question 1';
run;

/* Create  horizontal and vertical 3-D bar charts with midpoints
   from 15 to 35 by 5
    caxis     color of the axis
    cfr       color of the frame
    coutline  color of the outline of the bars
    shape     shape of the bars
    ctext     color of the text on the chart  */

proc gchart data = age_level;
   /* Create new title5 to describe the output*/
   title5 'Horizontal Bar Chart for Mother';
   /*Specify HORIZONTAL barchart*/
   hbar3d Mother / midpoints = 15 to 35 by 5
                   caxis = orange
                   cfr=verylightpurplishblue
                   coutline = verydarkblue
                   shape = hexagon
                   ctext = red;
run;

proc gchart data = age_level;
   /* Modify title5 to reflect new PROC output*/
   title5 'Vertical Bar Chart for Mother';
   /*Specify VERTICAL barchart*/
   vbar3d Mother / midpoints = 15 to 35 by 5
                  caxis = green
                  cfr=verylightpurple
                  coutline = verydarkblue
                  shape = prism
                  ctext = green;
   /* Use pattern command to change the colors of the bars */
   pattern color = purple;
run;

quit;
```

To save your program, select **File** → **Save as**. In the box next to **Save in**, click on **Google** drive, your user drive, your dropbox and/or your **flash** drive. In the box next to **File name**, type **sas_lab2su20** and click on **Save**.

To execute your program, select **Run** → **Submit** or simply click on the Running Stickman on the main toolbar. The **Graph** window should open - scroll down through this window to see both bar charts. Sketch your bar

Question 1

Vertical Bar Chart for Mother



charts in the space provided. (When you have finished your sketches, minimize or close the graph window.)

See Above

2. (Return to the **Program Editor** window.) Next, generate a plot of the variables **Father** (y-axis) vs **Mother** (x-axis) by adding the following lines of code **right before** the quit statement:

```
/* Use symbol statement to set up the format of the plot symbols
   value       symbol of the data points
   height      height of the symbol of the data points
   cv          color of the symbols*/
symbol1  value = +
        height=3
        cv = blue;

/* Use proc gplot to generate high resolution plot
    plot vertical*horizontal
    caxis   color of the axes
    ctext    color of the text on the plot   */
proc gplot data = age_level;
   /* Modify title4 and title5 */
   title4 'Question 2';
   title5 'Plot of Mother vs Father';

   plot Father*Mother /
      caxis = darkgreen
      ctext = darkred;
run;
```

This plot will appear in the **Output** window. You may need to scroll up or down through the **Output** window to see all your output. When your plot appears on the screen, have your TA or lab mate check it, then write their initials here: ____________________

3. One can use **proc univariate** or **proc means** to generate descriptive statistics.

    (i) Use **proc univariate** to generate descriptive statistics for the variable **Father**.

    Simply add the following lines of code to your program file **before the quit** statement, save your file (click on the black diskette in the main toolbar), and re-execute (click on the Running Stickman).

```
/* Use proc univariate to generate descriptive statistics */
proc univariate data = age_level;
    /* Modify title4 and title5 */
    title4 'Question 3';
    title5 'Descriptive Statistics for Father';
    /*make sure the variable/column is specified as 'Father'*/
    var Father;
run;
```

Complete the table below.

Variable	Mean	Median	Stdev	Max	Min
Father	26.250	25.000	4.962	39	19

(ii) Use **proc means** to generate the mean, median and standard deviation for the variable **Mother**. Add the following lines of code **right before** the run statement.

```
/* Use proc means to generate the mean, median and standard deviation for the variable Mother */
proc means mean median stddev data = age_level;
    /* Modify title5 */
    title5 'Descriptive Statistics for Mother';
    /*make sure the variable is specified as 'Mother'*/
    var Mother;
run;
```

Complete the table below.

Variable	Mean	Median	Stdev
Mother	24.550	24.000	4.310

1.2 Probability Distributions

RECALL:

♠ For **discrete** random variables, **probability density** = **probability mass function** = $P(X = x)$.

♣ For random variables, **survival function** = $P(X > a) = 1 - P(X \leq a)$

∇ $P(X \geq a) = 1 - P(X < a)$

▶ For **discrete** random variables only, $P(X < a) = P(X \leq a - 1)$.

▼ For **continuous** random variables only, $P(X \leq a) = P(X < a)$ and $P(X \geq a) = P(X > a)$.

NOTE: One can use the **PDF**, **CDF** and **SDF** functions in SAS to calculate probability density (mass) functions, cumulative distribution functions [$P(X \leq x)$], and survival distribution functions [$P(X > x)$]. The **quantile** function can be used to generate percentiles. (See Appendix D in the Class Lecture Notes.)

1. Suppose X is a continuous random variable that represents the lifelength (in months) of a battery. Also suppose X is normally distributed with a mean of 60 months and a standard deviation of 6 months.

   (i) Find the probability that a battery selected at random will last between 48 and 62 (inclusively) months.

   We want to use SAS to compute $P(48 \leq X \leq 62)$.

   NOTE: $P(48 \leq X \leq 62) = P(X \leq 62) - P(X < 48) = p1$.

   We write the following SAS code (right before the quit statement):

```
/* Create temporary SAS dataset called norm1 */
data norm1;
    /* X is Normal with mu and sigma
       Use an input statement to read in the values of mu, sigma, x1 and x2 */
        /*60 6     48 62*/
    input mu sigma x1 x2;
    /* P(x1 <= X <= x2) = P(X <= x2) - P(X < x1) */
    /* Use cdf function
    /* Format: cdf(’Normal’,x,mu,sigma) */
    p1 = cdf(’Normal’,x2,mu,sigma) - cdf(’Normal’,x1,mu,sigma);
    /*Alternatively, you could write the values directly*/
      /*p1 = cdf(’Normal’,62,mu,sigma) - cdf(’Normal’,48,mu,sigma);*/
datalines;
60 6 48 62
;

/* Print the results */
/* NOTE: The ‘noobs‘ stands for "no observation numbers" in the printout
   Try running it without the ‘noobs‘ to see the difference*/
proc print noobs data = norm1;
    /* Modify  title4 and title5 */
    title4 ’Probability Question 1’;
    title5 ’Part (i)’;
run;
```

   Save and execute your program. Complete the following:

mu	**sigma**	**x1**	**x2**	**p1**
60	6	48	62	0.60781

   (ii) Find the probability that a battery selected at random will last more than 63 months.

   To accomplish this task, add the following lines of code right **before** the **quit** statement.

```
data part_ii;
   /* Input variable list */
   input mu sigma x3;
   /* Use sdf function to get P(X > x)
      Format: sdf(’Normal’,x,mu,sigma) */
   p2 = sdf(’Normal’,x3,mu,sigma);
   /*Alternatively, you could write p2 = sdf(’Normal’,63,mu,sigma);*/
```

```
datalines;
60 6 63
;
run;
/* Print the results */
proc print noobs data = part_ii;
   /* Revise title5 */
   title5 'Part (ii)';
   /* If you want to only print a few columns, name them here */
   var mu sigma x3 p2;
run;
```

Save and execute your file. Complete the following:

mu	sigma	x3	p2
60	6	63	0.30854

(iii) Find the 96% percentile for the lifelength of the battery. (i.e., Find x such that $P(X \leq x) = 0.96$.) To accomplish this task , add the following lines of code right **before** the **quit** statement.

```
data part_iii;
    /* Input mu, sigma and the appropriate probability */
    input mu sigma prob1;

    /* Find x such that P(X <= x) = prob1 */
    /* X is Normal with mu and sigma
       Use quantile function
       Format: quantile('distribution',percentile,mu,sigma) */
    x4 = quantile('Normal',prob1,mu,sigma);
datalines;
60 6 0.96
;
run;

/* Print the results */
proc print noobs data = part_iii;
    /* Revise title5';
    title5 'Part (iii)';
run;
```

Be sure to save and execute your file to generate the new output. Complete the following:

mu	sigma	prob1	x4
60	6	0.96	70.5041

2. Suppose the probability that a person who regularly watches college football is female is 0.45. Twenty college football watchers are selected at random.

(i) What is the probability that exactly 9 are female?

(ii) What is the probability that at most 11 are female?

(iii) What is the probability that more than 10 are female?

(iv) What is the probability that at least 9 are female?

(v) What is the probability that between, and including, 8 and 12 are female?

We add the following lines of code to our existing SAS code, right **before** the **quit** statement.

```
data binom1;
    /* Input the values of p, n, x1, x2, x3, x4, x5 and x6 */
    /*    0.45 20 9  11 10 8  12 7 */
    input p    n  x1 x2 x3 x4 x5 x6;

    /* Note: x1 = 9, x2 = 11, x3 = 10, x4 = 8, x5 = 12, x6 = 7 */
    /* Create variables using pdf, cdf and sdf functions*/
       Format: P(X = x) = pdf(’Binom’,x,p,n)    P(X <= x) = cdf(’Binom",x,p,n)
       P(X > x) = sdf(’Binom’,x,p,n)*/

    /* Part (i)    p1 = P(X = 9) = pdf(’Binom’,x1,p,n)*/
    p1 = pdf(’Binom’,x1,p,n);

    /* Part (ii)   p2 = P(X <= 11) = cdf(’Binom’,x2,p,n) */
    p2 = cdf(’Binom’,x2,p,n);

    /*Part (iii)  p3 = P(X > 10) = 1 - P(X <= 10) = 1 - cdf(’Binom’,x3,p,n)
                                                  = sdf(’Binom’,x3,p,n)*/
    p3 =  1 - cdf(’Binom’,x3,p,n);

    p3a = sdf(’Binom’,x3,p,n);
    /*Part (iv)   p4 = P(X >= 9) = 1 - P(X < 9) = 1 - P(X <= 8) = 1 - cdf(’Binom’,x4,p,n)*/
    p4 =  1 - cdf(’Binom’,x4,p,n);
    /*Part (v)    p5 = P(8 <= X <= 12) = P(X <= 12) - P(X <= 7)
                     = cdf(’Binom’,x5,p,n) - cdf(’Binom’,x6,p,n);*/
    p5 = cdf(’Binom’,x5,p,n) - cdf(’Binom’,x6,p,n);
datalines;
0.45 20 9 11 10 8 12 7
;
run;

/* Print the results */
proc print noobs data = binom1;
    /* Revise title4 and title5 */
    title4 ’Probablity Question 2’;
    title5 ’Part (i) P(X = 9)’;
    var p1;
run;

proc print noobs data = binom1;
    /* Modify title5 */
    title5 ’Part (ii) P(X <= 11)’;
    var p2;
```

```
run;

proc print noobs data = binom1;
    /* Modify title5 */
    title5 'Part (iii) P(X > 10)';
    var p3 p3a;
run;

proc print noobs data = binom1;
    /* Modify title5 */
    title5 'Part (iv) P(X >= 9)';
    var p4;
run;

proc print noobs data = binom1;
    /* Modify title5 */
    title5 'Part (v) P(8 <= X <= 12) = P(X <= 12) - P(X <= 7) ';
    var p5;
run;
```

Be sure to save and execute your file to generate the new output. Complete the following:

(i) What is the probability that exactly 9 are female? 0.17705

(ii) What is the probability that at most 11 are female? 0.86924

(iii) What is the probability that more than 10 are female? 0.24929

(iv) What is the probability that at least 9 are female? 0.58569

(v) What is the probability that between, and including, 8 and 12 are female? 0.68996

Exit SAS.

2 R

1. **To invoke R**: Double-click on the *R* icon.
2. When **R** opens you should see the main menu and a window called the **R Console**. The **R Console** window is where all your output will appear.
3. **R** is command-line driven. All commands must follow the > prompt (similar to the **MTB** prompt in Minitab.
4. You always tell **R** to execute a command line by pressing the *Enter* key.
5. Commands are separated either by a semi-colon (;), or by a newline. Elementary commands can be grouped together into one compound expression by braces (and).
6. The R Console window allows command editing. The left and right arrow keys, home, end, backspace, insert, and delete work exactly as one would expect. The up and down arrow keys can be used to scroll through recent commands. Thus, if you make a mistake all you need to do is press the up key to recall your last command and edit it.

NOTE: The backslash (/) character has a special meaning to R. To specify a Windows path in the RConsole, either

1. use double backslashes (\\), or
2. use a forward slash (/). (It is suggested that you use this option.)

NOTE: R programs end in **.r**.

NOTE: Comments in **R** begin with a # sign.

```
> # This is a comment
```

NOTE: To quit **R**, type

```
> quit()   # or q()
```

NOTE: To list the contents of your workspace,

```
> ls()                         # List the contents of the workspace.
> rm(list=ls())                # This completely clears the workspace.
> ls()
character(0)                   # This means "nothing to see here"
```

Setting your working Directory:

NOTE: Be sure you create your **RSpace** directory before setting your working directory.

To set the current **working directory**, use the **setwd** command. (You might need to do this each time you start a new **R** session, depending on your system. Be sure you change the path to the location of your **data files**!)

```
> setwd("PATH/TO/THE/FOLDER/WITH/YOUR/DATAFILES") # set working directory
```

To check the current working directory,

```
> getwd()    # Displays the location of the current working directory
```

If you would like to see the contents of your working directory,

```
> dir()    # Displays the contents of your current working directory
```

Open R now and create/locate your working directory!

2.1 Basic Charts & Graphs

Let's use an *R script* to enter our commands. Open **R**. From the main menu select **File → New script**. The **R Editor** window will open. (It will say *untitled* until you save the script.)

★ Move the cursor to the **R Editor** window and type in the following titles in your code:

```
# Statistics 147
# Lab #2 Summer 2020
# Your name goes here
```

From the main menu in **R**, select **File → Save As**. The *Save Script As* window will open. Select a destination where you would like to save your script. In the box next to *Filename*, type **lab2_R_147_su20** and click **Save**.

2.1.1 Bar Chart

One can use the **barplot** command in **R** to generate a **bar chart**. For options, see pages 88-89 in the *Class Lecture Notes.*

Example 2.1. A random sample of 15 students was selected from a statistics course and their eye colors recorded as follows:

```
blue  green brown hazel hazel green brown brown brown hazel brown blue brown green hazel
```

NOTE: hazel is not a recognized color, so we will replace it with **yellow**.

♠ Complete the following by typing in the given **R** code in the **R Editor** window.

```
eyecolor1 <- c("blue", "green", "brown", "yellow", "yellow", "green", "brown",
"brown", "brown", "yellow", "brown", "blue", "brown", "green", "yellow")
# Print the data as a check
eyecolor1
```

Save your **R** script. (Select **File → Save**). To execute your script, from the main menu, select **Edit → Run All**. Complete the following from the **R** console window.

You should see the following in the **R Console** window.

```
 [1] "blue"  "green" "brown" "yellow" "yellow" "green" "brown" "brown" "brown"
[10] "yellow" "brown" "blue"  "brown" "green" "yellow"
```

♠ Create and print a frequency table. Type the following code in the **R Editor** window.

```
# Create frequency table
eyecolor_table1 <- table(eyecolor1)
# Print table
eyecolor_table1
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

blue	brown	green	yellow
2	6	3	4

♠ Create and print the labels for the horizontal axis. Type the following code in the **R Editor** window.

```
Create labels for x axis
colors1 <- c("blue", "brown", "green", "yellow")
# Print the labels
colors1
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
> Create labels for x axis
> colors1 <- c("blue", "brown", "green", "yellow")
> # Print the labels
> colors1

[1] ____________________________________________
```

♠ Create a bar chart. Type the following code in the **R Editor** window.

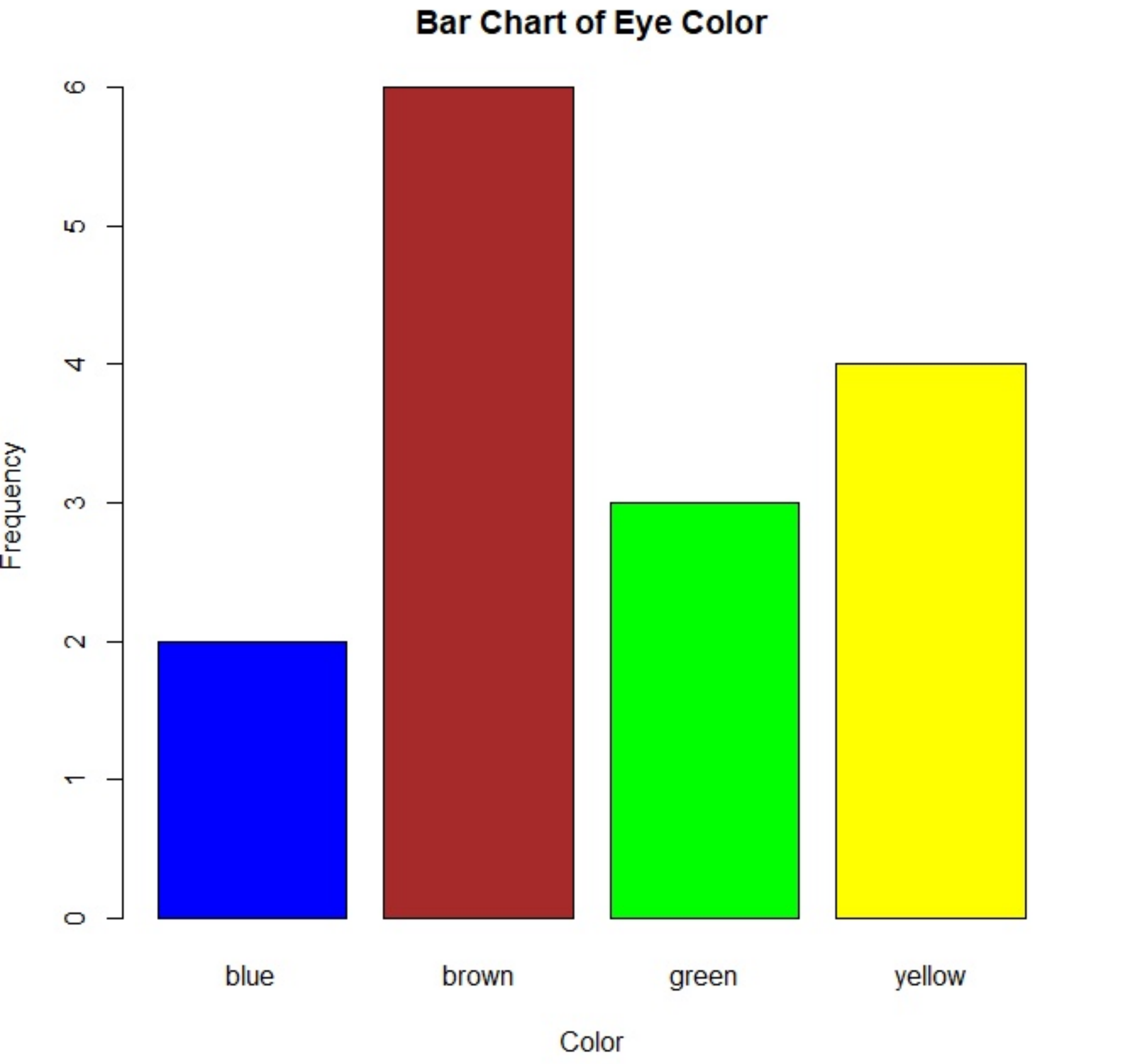
```
# Create bar chart: xlab = x-axis label, ylab = y-axis label, main = main title,
# col = color of the bars
barplot(eyecolor_table1,xlab = "Color",ylab = "Frequency",main = "Bar Chart of  Eye Color",
        col=colors1)

# Note, the above, single line code is a pain to read and debug.
# Try writing each parameter on it’s own line
barplot(eyecolor_table1,
    xlab = "Color",
    ylab = "Frequency",
    main = "Bar Chart of  Eye Color",
    col=colors1)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.


Bar Chart of Eye Color
Frequency
0
1
2
3
4
5
6
blue
brown
green
yellow
Color

♠ Sketch your bar chart in the space provided:

See Above

Example 2.2. Refer to Example 2.1. Create a pie chart for the data.

One can use the **pie** command in **R** to create a pie chart. For options, see page 93 in the *Class Lecture Notes.*

Type the following code in the **R Editor** window.

```
# Create pie chart: main = main title for chart, col = color of the slices,
# labels = labels for the slices
pie(eyecolor_table1,
    main = "Pie Chart for Eye Color Data",
    col = colors1,
    labels = colors1)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

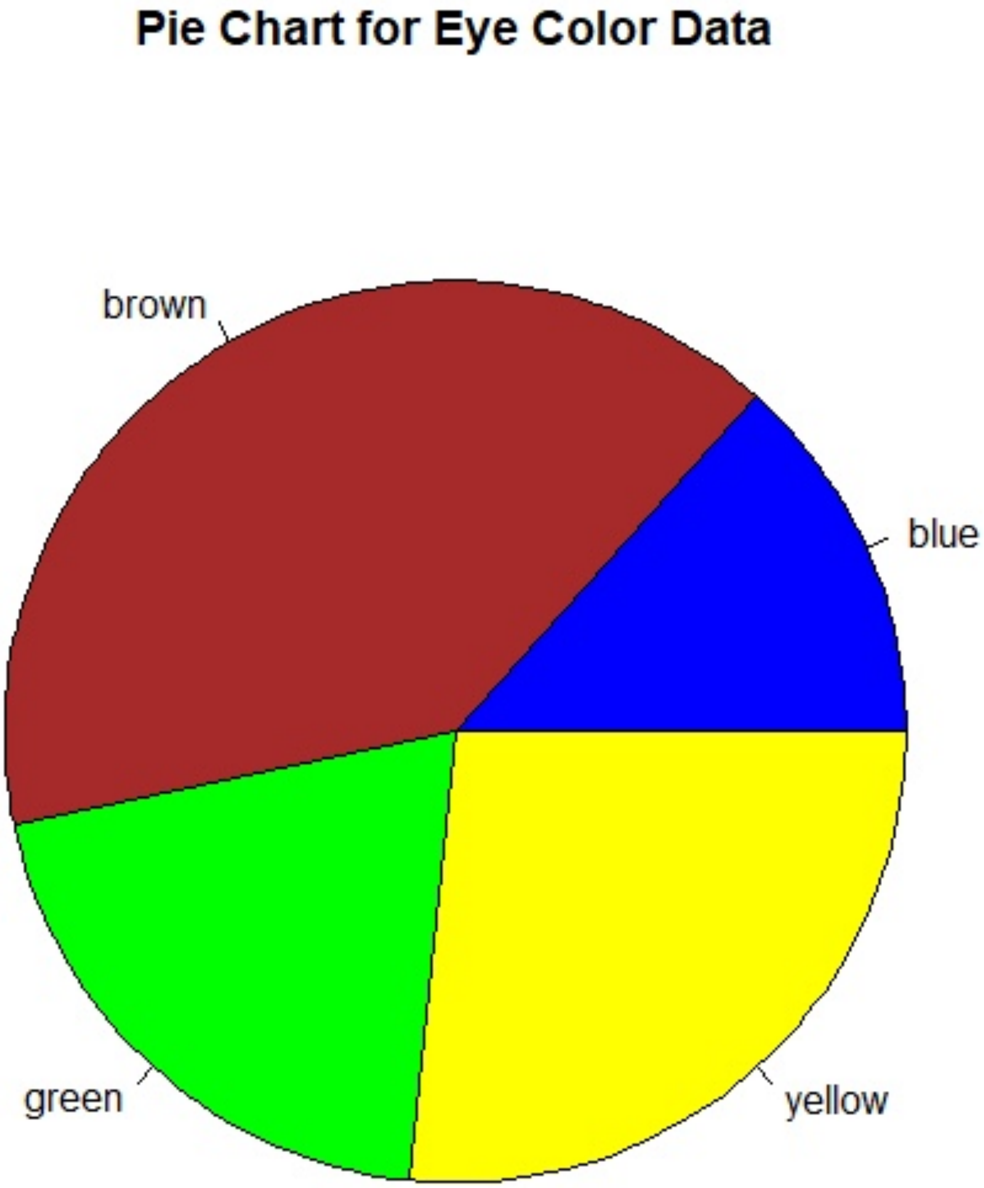
Sketch your pie chart in the space provided.

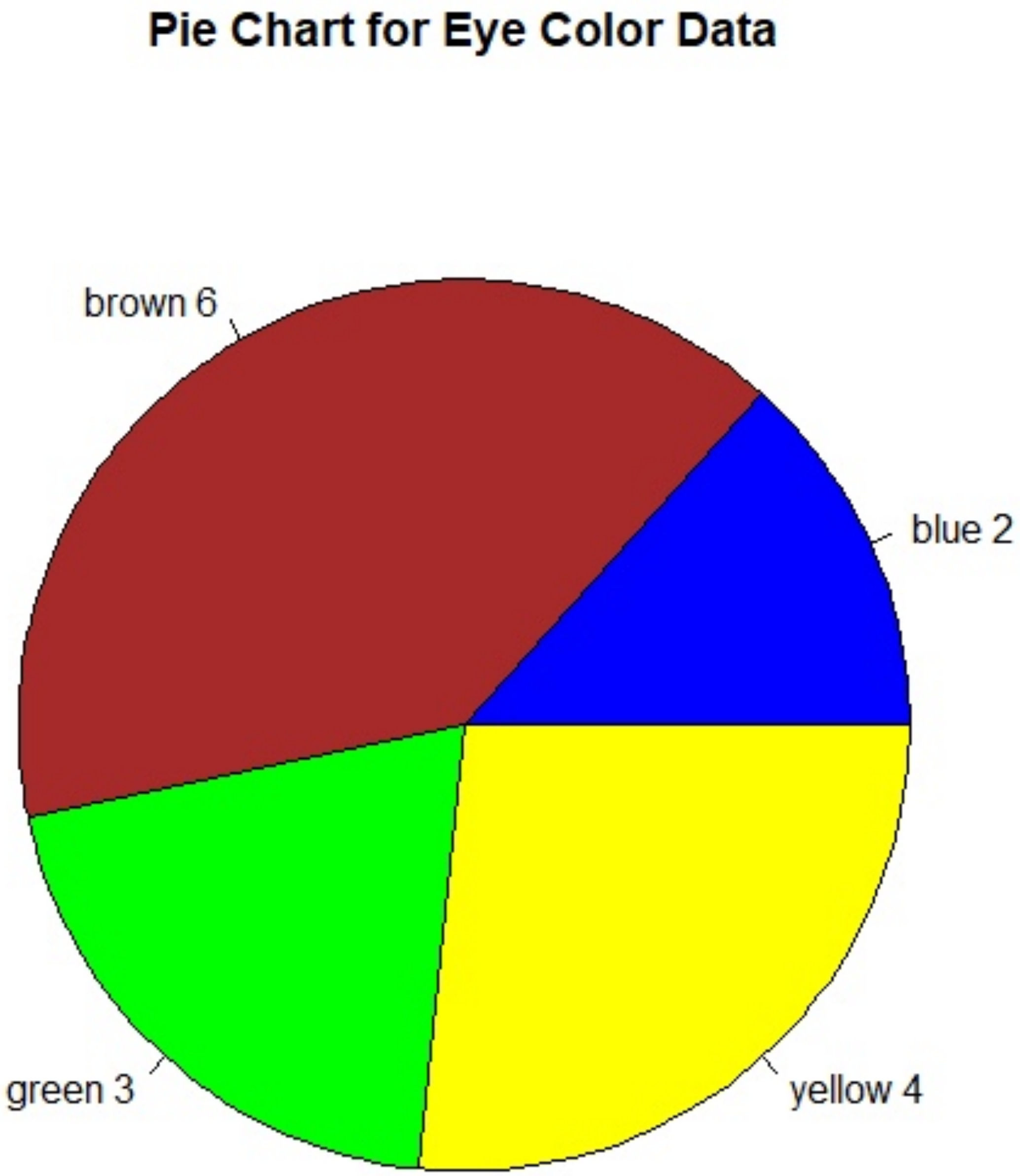
See Below

Example 2.3. Refer to Example 2.2. Create a new pie chart displaying color and frequency for each of the slices.

This can be accomplished typing the following commands in the **R Editor** window.

```
# Create vector of slice frequencies
slice1 <- c(2,6,3,4)
slice1 # print to check

# Create new label by pasting together the colors and slice frequencies
labels1 = paste(colors1,slice1)
labels1 # print to check
```


Pie Chart for Eye Color Data
brown
blue
green
yellow


Pie Chart for Eye Color Data
brown 6
blue 2
green 3
yellow 4

```
# Create pie chart

pie(eyecolor_table1,
    main = "Pie Chart for Eye Color Data",
    col = colors1,
    labels = labels1)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Sketch your new pie chart in the space provided:

See Above

2.1.2 Histograms

To create histograms in **R**, one can use the **hist** command. For options, see pages 80 - 81 in the *Class Lecture Notes*.

Example 2.4. In a recent study, data was recorded concerning the ages at which married couples had their first child. Twenty couples were selected at random and the following data was recorded in the data file, **ages.dat**.

♠ Use the **read.table** command to read the data into **R**. Use the names() function to obtain thye column names. Use the attach() function to make the columns individually accessible. Type the following lines of code in the **R Editor** window. **Be sure to change the path below to the location of your data file.**

```
ages1_data <- read.table(" PATH/TO/YOUR/DATAFILES/ages.dat", header= TRUE)
# Print the data
ages1_data

# Confirm that the variables Father and Mother have not yet been created.
# Try printing them and see that they don't yet exist
Father
Mother

# Use the names() function to obtain column names
names(ages1_data)

# Use the attach() function to make columns individually accessIble
attach(ages1_data)

# Show that the columns of data are now attached as separate vectors
Father
Mother
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

- ▲ highlight the new text you just typed.
- ▲ From the main menu, select **Edit** → **Run line or selection**.

When the output appears in the **R Console**, have your neighbor, TA or Luke initial here. ________________

♠ Use **hist** command in **R** to create a histogram for the ages of the mother. Type the following in the **R Editor** woindow.

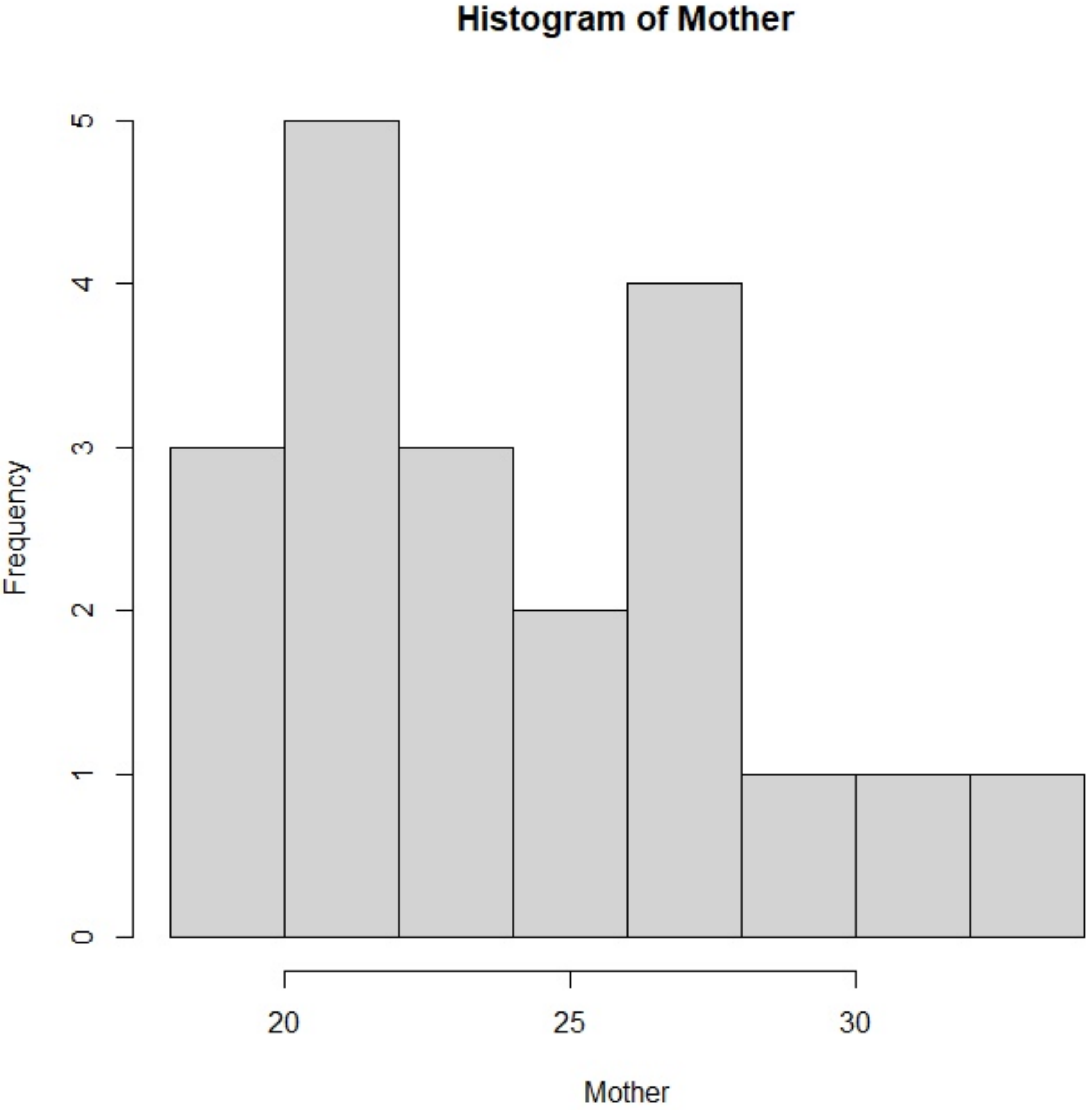
```
# Create histogram for variable Mother using the hist() command
hist(Mother)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

- ▲ highlight the new text you just typed.
- ▲ From the main menu, select **Edit** → **Run line or selection**.

Sketch your histogram in the space provided on the next page.

See Below


Histogram of Mother
Frequency
0
1
2
3
4
5
20
25
30
Mother

♠ The histogram is pretty boring, so let's spice it up! Modify the histogram so that the interval breaks occur from 15 to 35 in increments of 5. Add some color and an appropriate title.

We can accomplish this by typing the following in the **R Editor** window. (You do not need to type in the comments at this time, but I encourage you to add them at some point!)

```
# Create colors for the bars
colors1m <- c("brown","green","blue","yellow","red")
# Print the colors
colors1m

# Create interval breaks
brks1 <- c(15,20,25,30,35)
# Print the breaks
brks1

# Create histogram
# main = main title, breaks = interval breaks, col = colors of the bars
# To include the number of counts, you can just set labels=TRUE.
# xlim = range of the horizontal axis
# ylim = range for the vertical axis
# To see more, type ‘?hist‘ in the R console
hist(Mother,
    main= "Age of the Mother",
    breaks = brks1,
    col = colors1m,
    labels = TRUE,
    xlim = c(15,35),
    ylim = c(0,10))
```

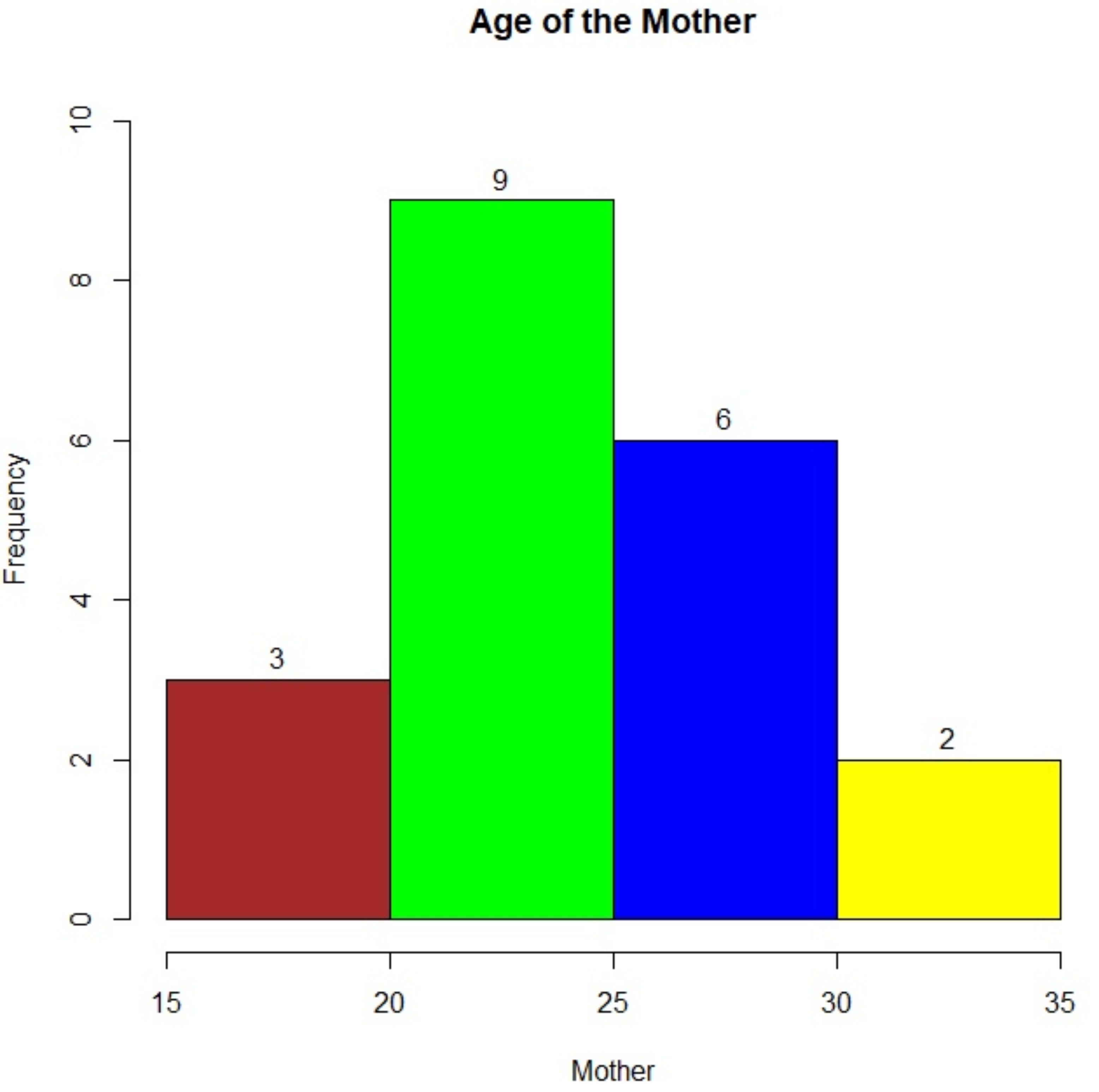
Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Sketch your histogram in the space provided:

See Below


Age of the Mother
Frequency
0
2
4
6
8
10
3
9
6
2
15
20
25
30
35
Mother

2.1.3 Scatterplots & Plots

One can use the **plot** command in **R** to create a scatterplot.

NOTE: For plot symbols, refer to page 106 in the *Class Lecture Notes*.

Example 2.5. Create a scatterplot of the ages of the Father versus the age of the Mother.

Type the following in the **R Editor** window.

```
# Use plot(x,y,xlab = x-axis label, ylab = y-axis label,main = title,pch = symbol_number)
plot(Father, Mother,
    xlab = "Father",
    ylab = "Mother",
    main = "Ages of Father vs Mother",
    pch = 8)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

When your plot appears on the screen, have Luke, your TA or lab mate check it, then write their initials here. ____________

Example 2.6. Consider the data file, **cartest1.dat**. Note that the data file includes headings in Line 1.

♠ To read in this data file, type the following in the **R Editor** window: **(To read in your data, be sure to change the path!!)**

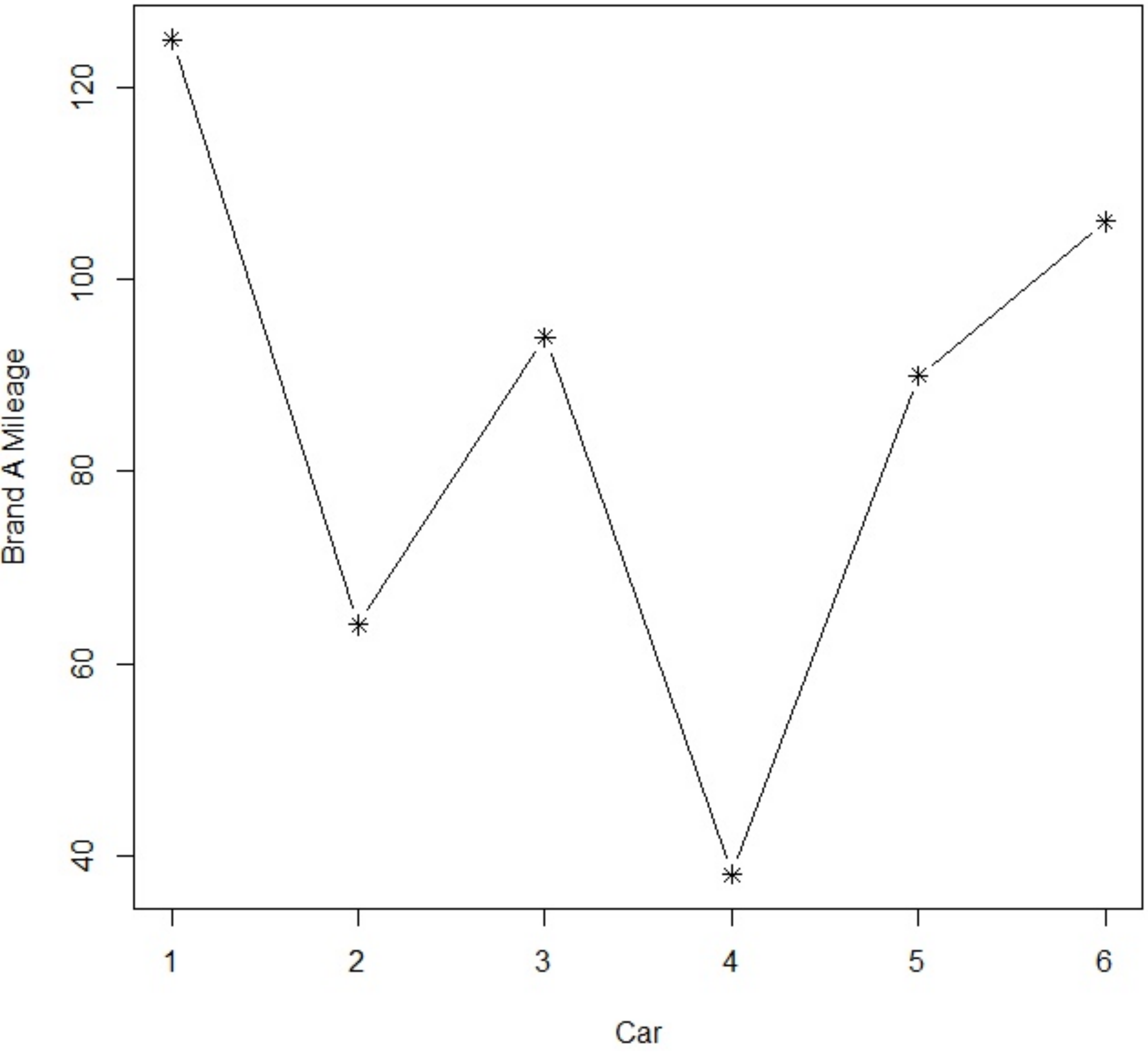
```
cartest = read.table(file = "PATH/TO/YOUR/DATAFILES/cartest1.dat", header = TRUE)
# Print the data
cartest

# Make column names accessible using the attach() function
attach(cartest)

# Use names(0 function to obtain column names
names (cartest)

# Print to check
Car
BrandA
BrandB
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

When your output appears in the **R Console** window, have Luke, your TA or lab mate check it, then write their initials here. ____________


Brand A Mileage
40
60
80
100
120
1
2
3
4
5
6
Car

♣ Create a plot of Car vs BrandA.

NOTE: For plot symbols, refer to page 106 in the *Class Lecture Notes*, or Google the term ”R plot point shapes”.

Type the following in the **R Editor** window.

```
# plot (variable, type = "o" = both point and line, ylab = label of y-axis,
# xlab = label of x-axis, pch = type of symbol)
plot(BrandA,
    type = "b",
    ylab = "Brand A Mileage",
    xlab = "Car",
    pch = 8)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

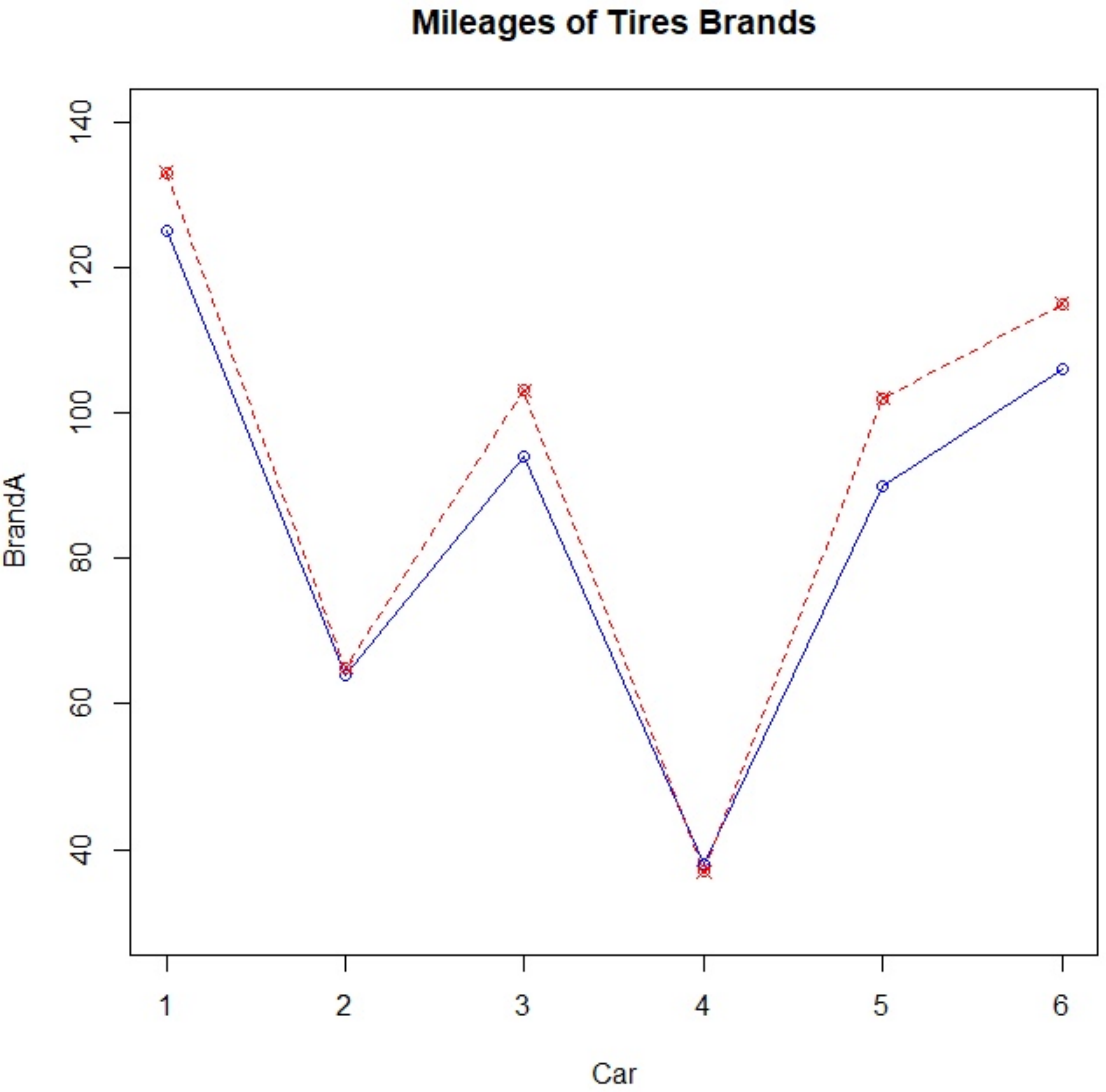
Sketch your plot in the space provided.

See Above

Example 2.7. Refer to Example 2.6. Plot a plot of Car vs BrandA and Car vs BrandB on the same axis.

Type the following in the **R Editor** window.

```
# Plot the cars data with type = o to get both the points and the connecting
# line and an overlay
plot(BrandA,
    type="o",
    col="blue",
    main = "Mileages of Tires Brands",
    xlab = "Car",
    ylim = c(30,140))

# graph line with type = "o" to get overlay, pch = 13 = circle cross symbol
# and lty = 2 = dashed line
lines(BrandB,
    type="o",
    pch=13,
    lty=2,
    col="red")
```


Mileages of Tires Brands
BrandA
140
120
100
80
60
40
1
2
3
4
5
6
Car

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

- ▲ highlight the new text you just typed.
- ▲ From the main menu, select **Edit** → **Run line or selection**.

Sketch your plot in the space provided.

See Above

2.2 Descriptive Statistics

Example 2.8. Refer to Example 2.4 and datafile **ages.dat**. Using **R**, find the mean, median and standard deviation of the variable, **Father**.

NOTE: We have already read in this data file so we don't have to read it in again during this session. Type the following in the **R Editor** window.

```
# Use mean function to generate mean of Father
mean_Father <- mean(Father)
# Print result
mean_Father

# Use the median function to generate the median of Father
median_Father <- median(Father)
# Print result
median_Father

# Use the sd function to generate the standard deviation of Father
sd_Father <- sd(Father)
# Print result
sd_Father
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

- ▲ highlight the new text you just typed.
- ▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window..

```
> # Use mean function to generate mean of Father
> mean_Father <- mean(Father)
> # Print result
> mean_Father

[1] ____________________
> # Use the median function to generate the median of Father
> median_Father <- median(Father)
> # Print result
> median_Father

[1] ___________________
> # Use the sd function to generate the standard deviation of Father
> sd_Father <- sd(Father)
> # Print result
> sd_Father

[1] ___________________
```

NOTE: Among many user-written packages, the package *pastecs* has an easy to use function called **stat.desc** to display a table of descriptive statistics for a list of variables. You can download the package and then load it.

♠ **Step 1:** From the main menu in R, select **Packages** → **Set CRAN Mirror** to open the CRAN mirror. In the CRAN mirror, select a location: USA and click OK.

♠ **Step 2:** Select **Packages** → **Install package**. A list of packages will open. Scroll down to, and select, the package you want to install (in this case **pastecs**) and click ok. (Agree to use a personal library so it will save in your system directory)

♠ **Step 2:** Select **Packages** → **Load package**. Scroll down the list of your available packages, select **pastecs** and click ok.

♠ **Easier Alternative:** You can use the `install.packages()` function from within the **R Console** to install any package.

```
> install.packages("pastecs")
```

♠ **Easier Alternative:** You can load any package that is already installed with the `library()` function.

```
library("pastecs")
```

Example 2.9. Refer to Example 2.8 and datafile **ages.dat**. Using the *pastecs* package in **R**, find the descriptive statistics for the variable, **Mother**.

Type the following in the **R Editor** window.

```
# Use stat.desc (from the pastecs package) to generate DESCriptive STATistics for Mother
# To see more, type ‘?stat.desc‘ in the R Console
stat.desc(Mother)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following table.

nbr.val	**nbr.null**	**nbr.na**	**min**	**max**	**range**
20.000	0.000	0.000	18.000	33.000	15.000
sum	**median**	**mean**	**SE.mean**	**CI.mean.0.95**	**var**
491.000	24.000	24.550	0.964	2.017	18.576
std.dev	**coef.var**				
4.310	0.1755				

NOTE: nbr.val = sample size, nbr.null = # of null values, nbr.na = # of missing values, CI.mean.0.95 = $t_{\alpha/2,n-1}s/\sqrt{n}$.

2.3 Probability Distributions

1. Suppose X is a discrete random variable such that $X \sim b(x; n = 6, p = 0.30)$. Use **R** to generate the pdf and cdf of X.

   In **R**,

   ♠ Use dbinom(x,size = n,prob = p) to generate $P(X = x)$ = pdf of X

   ♠ Use pbinom(x,size = n,prob = p) to generate $P(X \leq x)$ = cdf of X

   ♣ Use the **seq** command to generate the sequence from 0 to 6. Type the following in the **R Editor** window.

```
# Use seq function to generate sequence 0 to 6
x <- seq(0,6)
x   # Print the values
```

   Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

   ▲ highlight the new text you just typed.

   ▲ From the main menu, select **Edit** → **Run line or selection**.

   Complete the following.

```
> # Use seq function to generate sequence 0 to 6
> x <- seq(0,6)
> x   # Print the values

[1] _____________________________
```

   ★ Use the **dbinom** function to generate the pdf.

   Type the following in the **R Editor** window.

```
# Use the dbinom function to generate pdf: size = sample size, prob = p
pdf_x <- dbinom(x,size = 6,prob = 0.30)
pdf_x   # Print the values
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
> # Use the dbinom function to generate pdf: size = sample size, prob = p
> pdf_x <- dbinom(x,size = 6,prob = 0.30)
> pdf_x   # Print the values

[1] __________________________________________________________________
```

▲ Use the **pbinom** function to generate the cdf.
Type the following in the **R Editor** window.

```
# Use the pbinom function to generate cdf: size = sample size, prob = p
cdf_x <- pbinom(x,size = 6,prob = 0.30)
cdf_x   # Print the values
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
># Use the pbinom fuinction to generate cdf: size = sample size, prob = p
> cdf_x <- pbinom(x,size = 6,prob = 0.30)
> cdf_x   # Print the values

[1] __________________________________________________________________
```

√ Use the **cbind** function to combine the three columns of data.
Type the following in the **R Editor** window.

```
all_together <- cbind(x,pdf_x,cdf_x)
all_together  # Print the values
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

When your output appears on the screen, have Luke, your TA, or your lab mate check it, then write their initials here. ____________

■ Create a plot of x vs the pdf of x.
Type the following in the **R Editor** window.

```
# Use the plot function to generate a plot of x vs pdf_x
# Format: plot(x_variable,y_variable, main = "Title of Graph",
# col = color of plot points & line
# type = "o" to get points and connecting line on same axes,
# pch = symbol_type (see p. 89 in notes)
# cex = magnification of the plot symbols
```

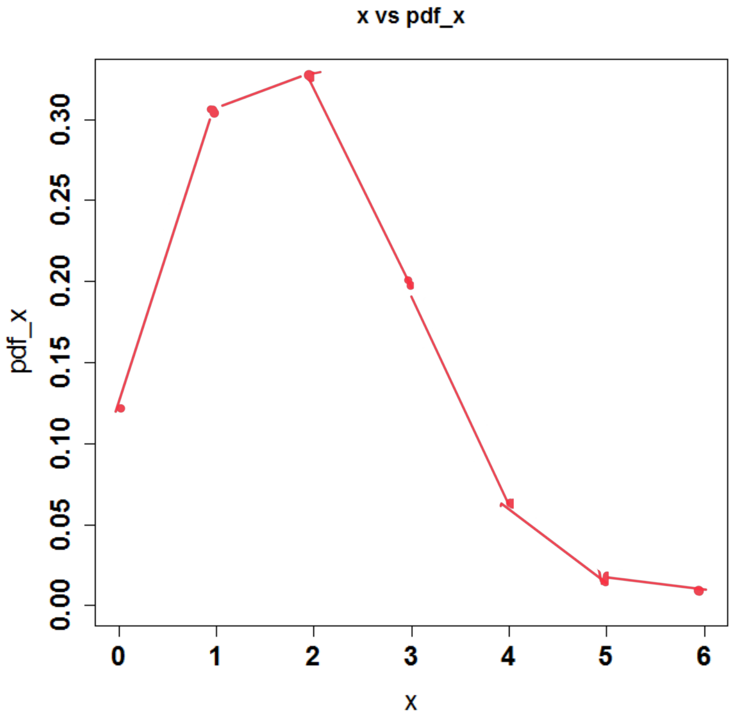
```
plot(x,pdf_x,
    main = "x vs pdf_x",
    col = "blue",
    type = "o",
    pch = 18,
    cex =1.5)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Sketch your plot.



∇ Create a plot of x vs the cdf of x.
Type the following in the **R Editor** window.

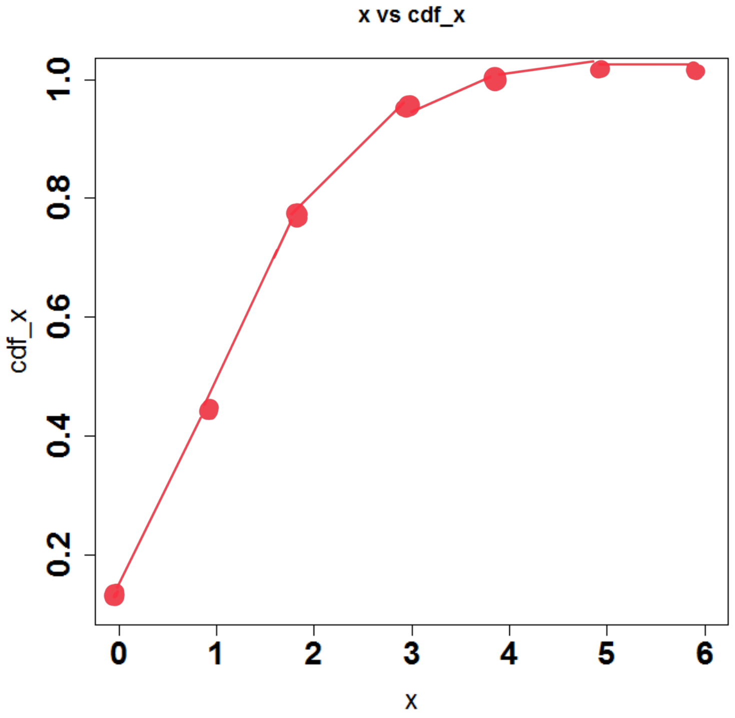
```
# Use the plot function to generate a plot of x vs cdf_x
# Format: plot(x_variable,y_variable, main = "Title of Graph",
# col = color of plot points & line
# type = "o" to get points and connecting line on same axes,
# pch = symbol_type (see p. 106 in notes)
# cex = magnification of the plot symbols
plot (x,cdf_x,
    main = "x vs cdf_x",
    col = "blue",
    type = "o",
    pch = 17,
    cex =1.5)
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save.**) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Sketch your plot.



2. Ruihan believes that 30% of *tablet pc* owners prefer the *Microsoft Surface Pro*. To test her claim, she selects a random sample of 25 *tablet pc* owners. Let X = # of owners that selected the *Microsoft Surface Pro*. Use this information to complete the following using **R**.

♣ What is the probability that exactly 9 owners selected the *Microsoft Surface Pro*?

Type the following in the **R Editor** window.

```
# P(X = 9)
# Use dbinom (x,size = sample size, prob = probability of success)
exactly9 <- dbinom(9,25,0.30)
exactly9   # Print the answer
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save.**) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window..

```
>  # P(X = 9)
> # Use dbinom (x,size = sample size, prob = probability of success)
> exactly9 <- dbinom(9,25,0.30)
> exactly9   # Print the answer

[1] ______________________
```

♣ What is the probability that no more than 9 owners selected the *Microsoft Surface Pro*?
Type the following in the **R Editor** window.

```
# P(X <= 9)
# Use pbinom (x,size = sample size, prob = probability of success)
nomorethan9 <- pbinom(9,size=25,prob=0.30)
nomorethan9   # Print the answer
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
> # P(X <= 9)
> # Use pbinom (x,size = sample size, prob = probability of success)
> nomorethan9 <- pbinom(9,size=25,prob=0.30)
> nomorethan9   # Print the answer

[1] _______________________
```

♣ What is the probability that more than 9 owners selected the *Microsoft Surface Pro*?
Type the following in the **R Editor** window.

```
# P(X > 9)
# Use pbinom (x,size = sample size, prob = probability of success, lower = FALSE)
# lower = FALSE = P(X > x)
morethan9 <- pbinom(9,size=25,prob = 0.30, lower = FALSE)
morethan9   # Print the answer
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

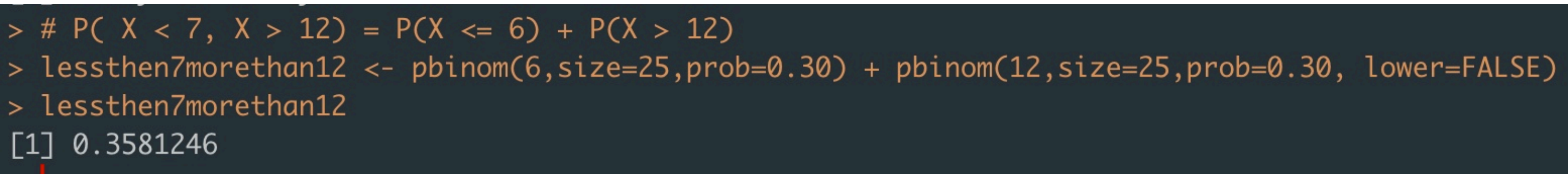
▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
> # P(X > 9)
> # Use pbinom (x,size = sample size, prob = probability of success, lower = FALSE)
> # lower = FALSE = P(X > x)
> morethan9 <- pbinom(9,size=25,prob = 0.30, lower = FALSE)
> morethan9   # Print the answer

[1] _________________________
```

♣ What is the probability that between 9 and 12, inclusively, owners selected the *Microsoft Surface Pro*?
Type the following in the **R Editor** window.

```
# P( 9 <= X <= 12) = P(X <= 12) - P(X < 9) = P(X <= 12) - P(X <= 8)
# Calculate between9and12 = P(X <= 12) - P(X <= 8)
between9and12 <- pbinom(12,size = 25,prob = 0.30) - pbinom(8,size=25,prob = 0.30)
between9and12   # Print the answer
```



```
> # P( X < 7, X > 12) = P(X <= 6) + P(X > 12)
> lessthen7morethan12 <- pbinom(6,size=25,prob=0.30) + pbinom(12,size=25,prob=0.30, lower=FALSE)
> lessthen7morethan12
[1] 0.3581246
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

- ▲ highlight the new text you just typed.
- ▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
> # P( 9 <= X <= 12) = P(X <= 12) - P(X < 9) = P(X <= 12) - P(X <= 8)
> # Calculate between9and12 = P(X <= 12) - P(X <= 8)
> between9and12 <- pbinom(12,size = 25,prob = 0.30) - pbinom(8,size=25,prob = 0.30)
> between9and12   # Print the answer

[1] _______________________
```

♣ (Your turn!) What is the probability that less than 7 or more than 12 owners selected the *Microsoft Surface Pro*? Be sure to write your **R** commands and the results of each step in the space provided.
Commands:

see Above
____

____

____

____

____

ANSWER: ___________

NOTE: Suppose $X \sim N(\mu, \sigma)$. In **R**,

- ♠ Use dnorm(x,mean,sd) to generate $P(X = x)$ = pdf of X
- ♠ Use pnorm(x,mean,sd) to generate $P(X \leq x)$ = cdf of X
- ♠ Use pnorm(x,mean,sd,lower = FALSE) to generate $P(X > x)$ = 1 - cdf of X

NOTE: You do not have to include the words mean and sd, but you can if you wish.

3. *Microsoft* claims that the *Surface Pro 4* has a battery life that is normally distributed with a mean of 8 hours (when web browsing) and a standard deviation of 1.25 hours. Luke wants to buy a *Surface Pro 4* for use in his research project. He goes to a local store and a salesperson selects a *Surface Pro 4* at random for him. Luke takes the *Surface Pro 4* to his office, fully charges the battery and then uses it until it runs out of power. Let X = the battery life of the *Surface Pro 4*.

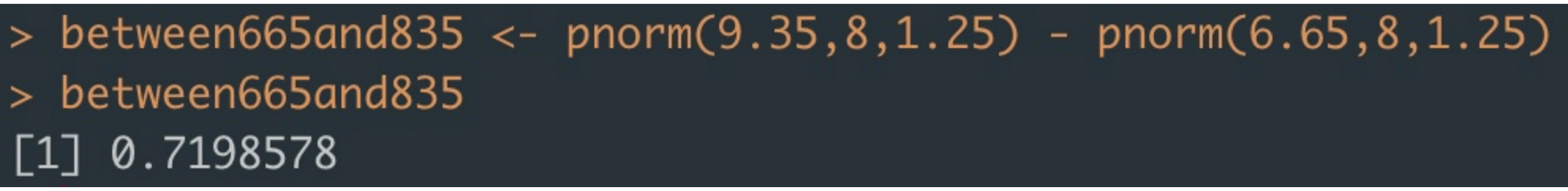
♣ Find the probability that the battery life is less than 6.4 hours?
Type the following in the **R Editor** window.

```
# P(X < 6.4)
# Use pnorm(x,mean,sd)
lower1 <- pnorm(6.4,mean = 8,sd = 1.25)
lower1  # Print the answer
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

- ▲ highlight the new text you just typed.
- ▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.



```
> between665and835 <- pnorm(9.35,8,1.25) - pnorm(6.65,8,1.25)
> between665and835
[1] 0.7198578
```

```
> # P(X < 6.4)
> # Use pnorm(x,mean,sd)
> lower1 <- pnorm(6.4,mean = 8,sd = 1.25)
> lower1  # Print the answer

[1] ___________________________
```

♣ Find the probability that the battery life is greater than 6.4 hours?
Type the following in the **R Editor** window.

```
# P(X > 6)
# Use pnorm(x,mu,sigma,lower = FALSE) to get upper tail
upper1 <- pnorm(6.4,8,1.25,lower = FALSE)
upper1  # Print the answer
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

▲ highlight the new text you just typed.

▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
> # P(X > 6)
> # Use pnorm(x,mu,sigma,lower = FALSE) to get upper tail
> upper1 <- pnorm(6.4,8,1.25,lower = FALSE)
> upper1  # Print the answer

[1] ____________________________
```

♣ (Your turn!) Find the probability that the battery life is between 6.65 and 8.35 hours. Be sure to write your **R** commands and the results of each step in the space provided.
Commands:

See Above

```
______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________
```

ANSWER: ___________

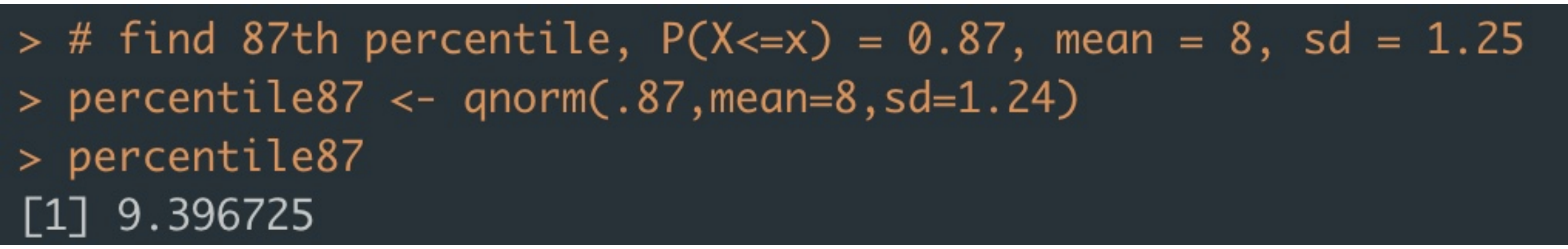
NOTE: To generate the quantiles (inverse cumulative probabilities) in **R**, put a **q** followed by the designated name of the distribution.

For example

```
qnorm(quantile,mean,sd)
```

4. Refer to Question 3.

♠ Find the 95th percentile (quantile). (i.e., find x such that $P(X \leq x) = 0.95$.)
Type the following in the **R Editor** window.



```
> # find 87th percentile, P(X<=x) = 0.87, mean = 8, sd = 1.25
> percentile87 <- qnorm(.87,mean=8,sd=1.24)
> percentile87
[1] 9.396725
```

```
# Use qnorm(quantile, mean,sd) to generate 95th percentile
quantile95 <- qnorm(0.95,mean=8,sd =1.25)
quantile95   # Print the answer
```

Make sure your cursor is in the **R Editor** window. Save your **R** script (Select **File** → **Save**.) and then

- ▲ highlight the new text you just typed.
- ▲ From the main menu, select **Edit** → **Run line or selection**.

Complete the following from the **R Console** window.

```
> # Use qnorm(quantile, mean,sd) to generate 95th percentile
> quantile95 <- qnorm(0.95,mean=8,sd =1.25)
> quantile95   # Print the answer

[1] __________________________
```

♠ (Your turn!) Find the 87th percentile (quantile). (i.e., find x such that $P(X \leq x) = 0.87$.) Be sure to include your **R** commands and the results!

Commands:

```
                          See Above
______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________

______________________________________________________________________
```

ANSWER: ____________

NOTE:

- ♣ Be sure to save your **R** script. You can open it any time and run any of the code.
- ♣ If you want to save the contents of the **R Console** window, copy and paste it into Word/Notepad/LaTeX.

Quit **R** by typing **q()** or **quit()** next to the **R** prompt.

You have now successfully completed Lab #2. Please straighten your work area (Please make sure the desktop is clean and that the chair is pushed in.), log off your account, and turn in your worksheet. Don't forget your flash drive (if you used one)! Thanks and have a good day!

Luke and Ruihan