Introduction to R
2021-22
Data Visualization with ggplot2

Summary

In this illustration, you will learn how to produce some basic graphs (hopefully some useful ones!) using the package **ggplot2**. You will be using an R dataset that you import directly into R Studio.

Before You Begin: Be sure to have downloaded from the course website: ***framingham.Rdata***

Before You Begin: Be sure to have installed (one time) the following packages. Recall. There are two ways to do this: (1) From the tab **PACKAGES > INSTALL**; be sure to click on box "install dependencies"; or (2) from the console pane using the command **install.packages("*nameofpackage*")**.

__**#1. Hmisc**
__**#2. stargazer**
__**#3. summarytools**
__**#4. ggplot2**

Introduction
Framingham Heart Study (Didactic Dataset)

The dataset you are using in this illustration **(framingham.Rdata)** is a subset of the data from the Framingham Heart Study, Levy (1999) National Heart Lung and Blood Institute, Center for Bio-Medical Communication.

The objective of the Framingham Heart Study was to identify the common factors or characteristics that contribute to cardiovascular disease (CVD) by following its development over a long period of time in a large group of participants who had not yet developed overt symptoms of CVD or suffered a heart attack or stroke. The researchers recruited 5,209 men and women between the ages of 30 and 62 from the town of Framingham, Massachusetts, and began the first round of extensive physical examinations and lifestyle interviews that they would later analyze for common patterns related to CVD development. Since 1948, the subjects have continued to return to the study every two years for a detailed medical history, physical examination, and laboratory tests, and in 1971, the study enrolled a second generation - 5,124 of the original participants' adult children and their spouses - to participate in similar examinations. In April 2002 the Study entered a new phase: the enrollment of a third generation of participants, the grandchildren of the original cohort. This step is of vital importance to increase our understanding of heart disease and stroke and how these conditions affect families. Over the years, careful monitoring of the Framingham Study population has led to the identification of the major CVD risk factors - high blood pressure, high blood cholesterol, smoking, obesity, diabetes, and physical inactivity - as well as a great deal of valuable information on the effects of related factors such as blood triglyceride and HDL cholesterol levels, age, gender, and psychosocial issues. With the help of another generation of participants, the Study may close in on the root causes of cardiovascular disease and help in the development of new and better ways to prevent, diagnose and treat cardiovascular disease.

This dataset is a HIPAA de-identified subset of the 40-year data. It consists of measurements of 9 variables on n=4699 patients who were free of coronary heart disease at their baseline exam.

Coding Manual

Position	Variable	Variable Label	Codes
1.	id	Patient identifier	
2.	sex	Patient gender	1 = male 2 = female
3.	sbp	Systolic blood pressure, mm Hg	
4.	dbp	Diastolic blood pressure, mm Hg	
5.	scl	Serum cholesterol, mg/100 ml	
6.	age	Age at baseline exam, years	
7.	bmi	Body mass index, kg/m^2	
8.	month	Month of year of baseline exam	
9.	followup	Subject's follow-up, days since baseline	
10.	chdfate	Event of CHD at end of follow-up	1 = patient developed CHD at follow-up 0 = otherwise

1.
Introduction to ggplot2

__1a. *Good to know!* Build Your Plot Layer by Layer

At first glance, the code to produce a plot using ggplot( ) might appear to be daunting. Mercifully, once you know what you are looking at, it's all quite lovely!

ggplot( ) plots are built in layers. There are 3 required layers. Then there are several optional layers

Required layers

- **Layer #1**: Tell R what data you are plotting
- **Layer #2**: Define your X-axis, Y-axis and, optionally, any other variables (e.g., grouping variable)
  Note – This layer is also called "aesthetic mappings"
- **Layer #3**: Tell R what kind of plot to produce (e.g., scatter, bar, histogram, etc.)
  *Note – This layer is variously referred to as "geometrics" or geom_*

Additional layers
You will definitely want to familiarize yourself with the array of additional layers that are possible. These include

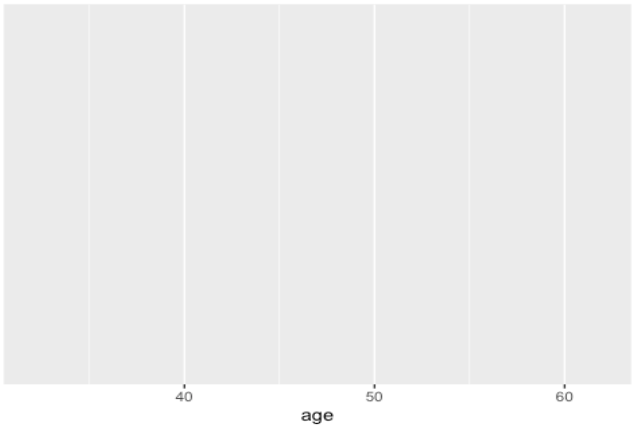
- **FACETS**: Facets allow you to produce separate plots for each level of some grouping variable (s).
- **AXIS, LABELS, LIMITS**: These utilities allow you to set the scale and tick marks of your axes.
- **THEME and OTHER LABELS**: These utilities allow you to set the overall design (e.g. blank background or pale gray with grid lines, etc) and other labeling such as legends

1b. Illustration #1 – How Layering Works

```
load(file="framingham.Rdata")      # Assumes the data are in the working directory
library(ggplot2)

# For this illustration, I am using a random sample of n=100 observations
smalldf <- framinghamdf[sample(nrow(framinghamdf),100),]
smalldf$chdfate <- factor(smalldf$chdfate,
                          levels=c(0,1),
                          labels=c("0=Other", "1=Event of CHD"))

# LAYER 1, required:  Specify the data
ggplot(data=smalldf)
```

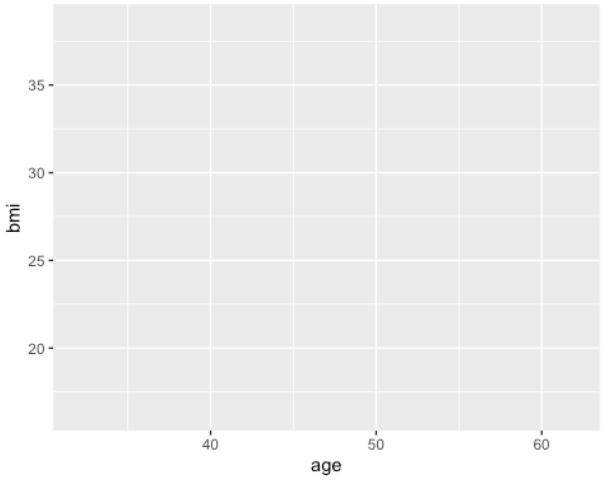

Whoa!! Nothing. Actually, the blank result is correct. This is because all we have accomplished is to notify R that the data that will be used in plotting is smalldf.

```
# LAYER 2, required:  Specify the x-axis using aes()
# Note:  The continuation characther + must go at the END OF THE LINE
ggplot(data=smalldf) +
    aes(x=age)
```



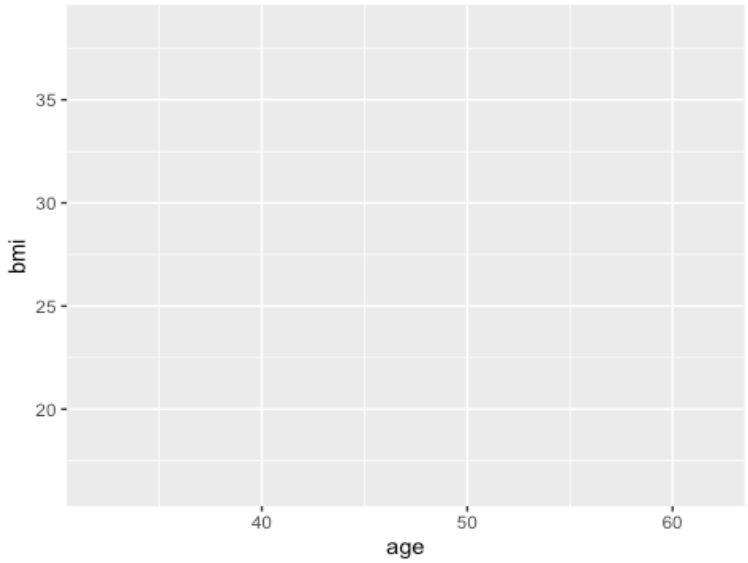
This layer 2 command added the specification of the x-axis. Still no actual plot, however. Stay tuned.

```
# Another example of Layer 2:  x-axis and  y-axis specified separately
ggplot(data=smalldf) +
    aes(x=age) +
    aes(y=bmi)
```



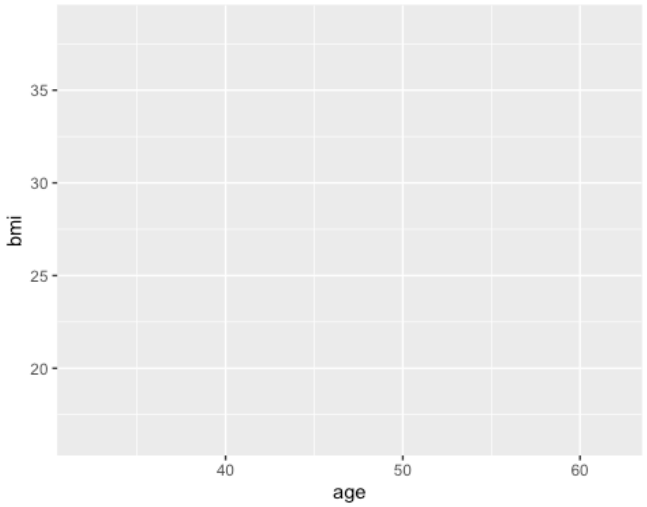
This layer 2 command added the specification of the y-axis

```
# Another example of Layer 2:  Specify both x and y axes at once
ggplot(data=smalldf) +
    aes(x=age, y=bmi)
```



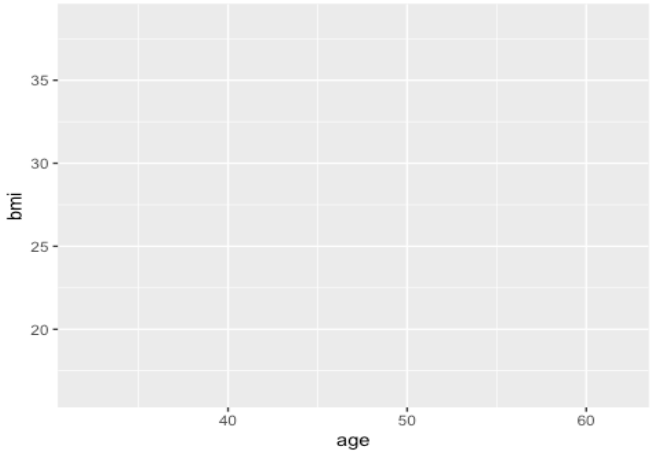
Yes, you're right. In layer 2, you can specify both the x and y axes in one aes( ) command

```
# Another example of Layer 2: Specify a (factor) grouping variable using color=
ggplot(data=smalldf) +
    aes(x=age, y=bmi) +
    aes(color=chdfate)
```



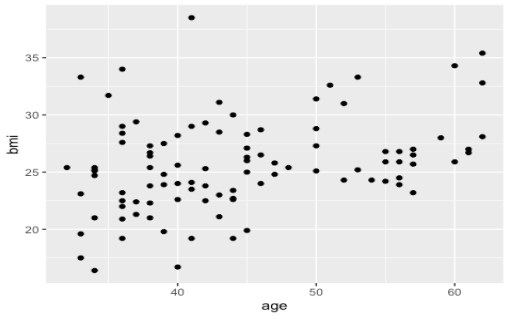
This layer 2 command added specification of a third (grouping) variable. However, because we are still in "Layer 2" work, there is not plot. This is why this picture looks the same as the picture on the bottom of the previous page.

```
# Another example of Layer 2: Specify a (factor) grouping variable using color= and shape=
ggplot(data=smalldf) +
    aes(x=age, y=bmi) +
    aes(color=chdfate, shape=chdfate)
```



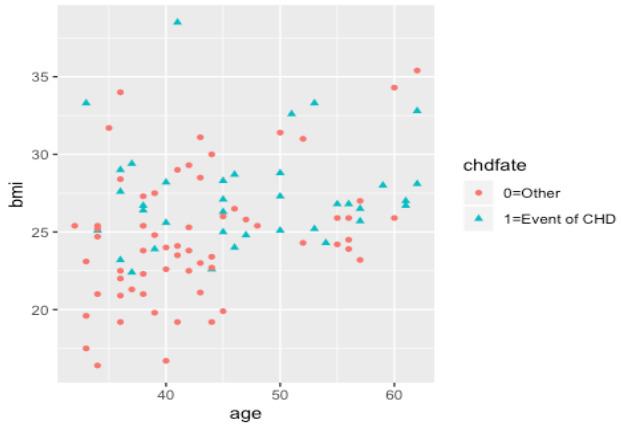
Again, the layer 2 specification of a grouping variable does not produce anything to see just yet.

```
# LAYER 3, required:  Plot!  Choose your geom_FILLINNAME()
# Example:  Simple XY scatter plot using geom_point()
ggplot(data=smalldf) +
    aes(x=age, y=bmi) +
    geom_point()

## Warning: Removed 1 rows containing missing values (geom_point).
```



Finally!!! We get a plot! Addition of the layer geom_point( ) produces a scatter plot.

```
# Another example of LAYER 3
# Example:  Simple XY scatter plot using geom_point()
# Separate color and shape for grouping variable chdfate
ggplot(data=smalldf) +
    aes(x=age, y=bmi) +
    aes(color=chdfate, shape=chdfate) +
    geom_point()

## Warning: Removed 1 rows containing missing values (geom_point).
```



Thought you might like to see the result of having previously coded aes(color=chdfate, shape=chdfate) in layer 2. Now our layer 3 scatter plot shows the AGE-BMI scatter in separate colors and shape depending on the group (chdfate = 0 or 1).

__1c. Guidelines for Layering (required)

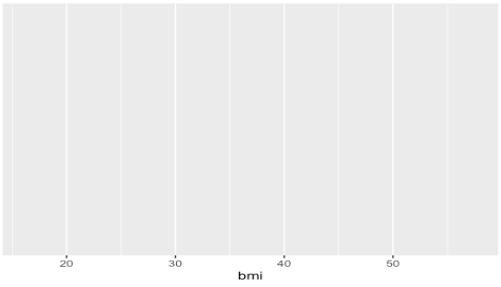
Hack: ***The continuation character "+" must go at the end of the line, NOT the start of the next line!***

	Layer	Examples
1	**Dataset** **data=DATAFRAMENAME** Key: **data=** tells R the object (dataframe) where you will find the variables you want to plot	**Example** `ggplot(data=framinghamdf)`
2	**Aesthetic mappings, aes( )** **aes(x=XVAR, y=YVAR, color=ZVAR, shape=ZVAR)** Key: **aes( )** tells R how to map your X and/or Y variables to the features of your graph. Important: What is put into **aes( )** will depend on whether you are doing a single variable plot or a multiple variable plot. It may also depend on the particular plot	**Examples** `ggplot(data=framinghamdf, aes(x=bmi))` Single Variable Plots aes(x=factor(chdfate)) aes(x=bmi) aes(x=" ", y=age) Multiple Variable Plots aes(x=factor(chdfate), y=bmi) aes(x=age,y=bmi) aes(x=age, y=bmi,color=chdfate) aes(x=age, y=bmi, shape=chdfate)
3	**geom_** Key: **geom_FILLIN( )** tells R what kind of plot to produce (e.g. box plot, histogram, xy scatter, etc) Geoms can have additional arguments: For example: stat: add a statistical transformation or calculation to your plot position: choose how you want things to be positioned or overlapped	**Examples** `p <- ggplot(data=framinghamdf, aes(x=bmi)) + geom_histogram(aes(y=..density..))` Some Other geom_ geom_bar( ) geom_histogram( ) geom_boxplot( ) geom_point( ) geom_smooth( )
4	**Axis labels, axis limits, annotations** Axis Labels **xlab(" ")** **ylab(" ")** **ggtitle(" ")** Axis Limits **xlim(#,#)** **scale_x_continuous( )** **ylim(#,#)** **scale_y_continuous( )**	**Examples** `xlab("Body Mass Index (kg/m2)")` ***Hack: Use \n to insert return so as to have text over multiple lines***
5	**theme_** Key: This is the final bit of customization. Do you want a gray background? Gray plot area? Etc?	**Example** `theme_bw()`

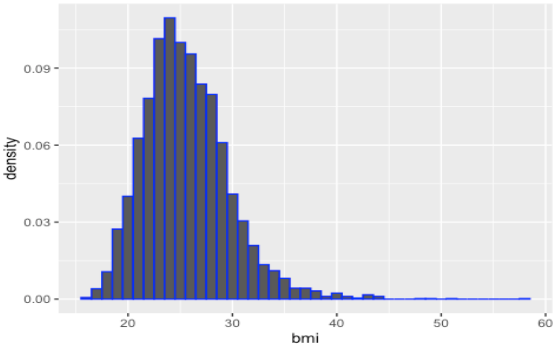
__1d Illustration #2 – Histogram with Overlay Normal Distribution

Hack: ***The continuation character "+" must go at then end of the line, NOT the start of the next line!***

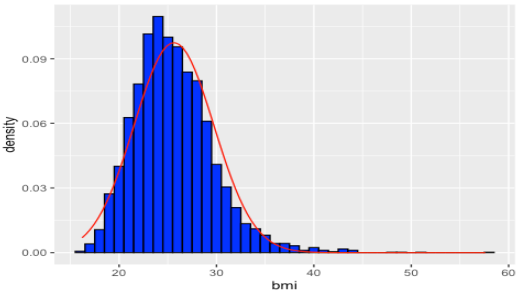
```
# Layer 1 (dataset) and Layer 2 (aesthetics) can be combined into one instruction
ggplot(data=framinghamdf, aes(x=bmi)) +
```



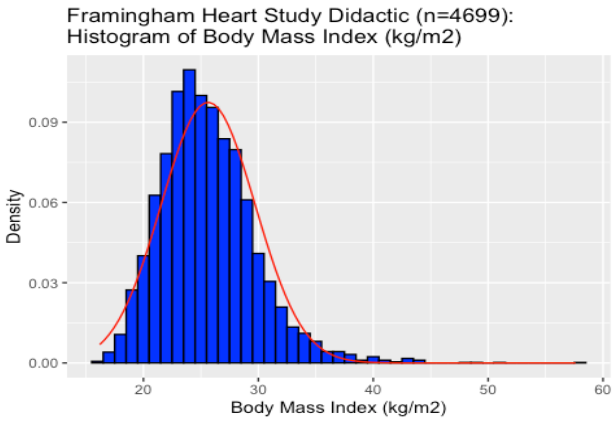

```
# Layer 3.   GEOM_  +
# Tell R which kind of plot to produce
geom_histogram(binwidth=1, colour="blue", aes(y=..density..)) +
```



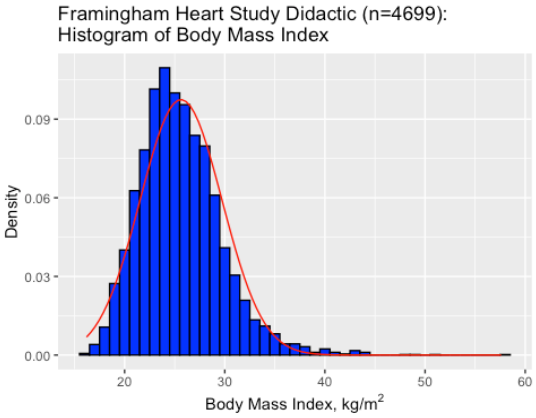

```
# LAYER 3, continued. STAT +   Note:  This is actually an argument of the GEOM_
# Here we are telling R overlay a statistical calculation, in particular an overlay normal curve
# IMPORTANT:  Be sure to include na.rm=TRUE since calculations will not happen if there are missing values
                                                                                +
 stat_function(fun=dnorm, color="red", args=list(mean=mean(framinghamdf$bmi,na.rm=TRUE),
      sd=sd(framinghamdf$bmi,na.rm=TRUE))) +
```



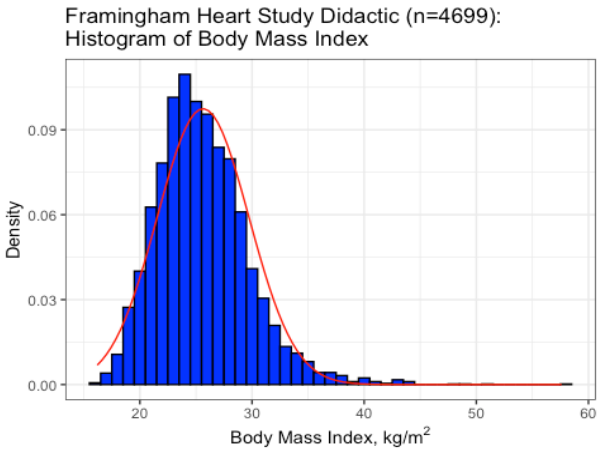
```
# LAYER 4. ADD TITLE, LABELS, AXIS LIMITS, etc +
ggtitle("Framingham Heart Study Didactic (n=4699):  \nHistogram of Body Mass Index (kg/m2)") +
    xlab("Body Mass Index (kg/m2)") +
    ylab("Density") +
```



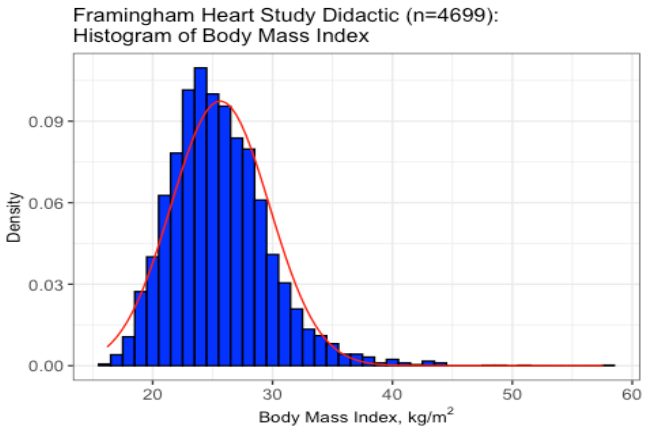

```
# LAYER 4, EXTRA.   Carol decides to go back in and edit so as to the superscript for meters squared
ggtitle("Framingham Heart Study Didactic (n=4699):  \nHistogram of Body Mass Index") +
    xlab(expression("Body Mass Index, kg/m"^{2} ))  +
    ylab("Density") +
```



```
# LAYER 5  THEME +
# Final customizations to make your graph especially good looking!
# Here we use the theme theme_bw() to get rid of the grey in the plotting area
theme_bw() +
```




```
# LAYER 5 EXTRA. Fine tune the appearance of title and axis titles

theme(axis.text=element_text(size=10),
          axis.title=element_text(size=10),
          plot.title=element_text(size=12))
```



How about that! So pretty!

2.
Preliminaries

```
setwd("/Users/cbigelow/Desktop")
library(Hmisc)
library(stargazer)
library(summarytools)
library(ggplot2)
```

Input data. Check. Label variables. Label variable values.

```
load(file="framingham.Rdata")
str(framinghamdf)

## 'data.frame':    4699 obs. of  10 variables:
##  $ id      : int  2642 4627 2568 4192 3977 659 2290 4267 2035 3587 ...
##  $ sex     : int  1 1 1 1 1 2 1 1 1 1 ...
##  $ sbp     : int  120 130 144 92 162 212 140 174 142 115 ...
##  $ dbp     : int  80 78 90 66 98 118 85 102 94 70 ...
##  $ scl     : int  267 192 207 231 271 182 276 259 242 242 ...
##  $ age     : int  55 53 61 48 39 61 44 39 47 60 ...
##  $ bmi     : num  25 28.4 25.1 26.2 28.4 ...
##  $ month   : int  8 12 8 11 11 2 6 11 5 10 ...
##  $ followup: int  18 35 109 147 169 199 201 209 265 278 ...
##  $ chdfate : int  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "datalabel")= chr ""
##  - attr(*, "time.stamp")= chr "17 Apr 2014 14:25"
##  - attr(*, "formats")= chr  "%8.0g" "%8.0g" "%8.0g" "%8.0g" ...
##  - attr(*, "types")= int  252 251 252 252 252 251 254 251 252 251
##  - attr(*, "val.labels")= chr  "" "" "" "" ...
##  - attr(*, "var.labels")= chr  "" "" "" "" ...
##  - attr(*, "version")= int 12

label(framinghamdf$bmi) <- "bmi: Body Mass Index (kg/m2)"
label(framinghamdf$age) <- "age: Age (years)"
label(framinghamdf$chdfate) <- "chdfate: Event of CHD (0/1)"
framinghamdf$chdfate <- factor(framinghamdf$chdfate,
                               levels=c(0,1),
                               labels=c("0=Other", "1=Event of CHD"))
```

Descriptives on the variables used in this illustration

```
summarytools::freq(framinghamdf$chdfate)

## Frequencies
##
##                       Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## -------------------- ------ --------- -------------- --------- --------------
##              0=Other   3226     68.65          68.65     68.65          68.65
##       1=Event of CHD   1473     31.35         100.00     31.35         100.00
##                 <NA>      0                               0.00         100.00
##                Total   4699    100.00         100.00    100.00         100.00
```

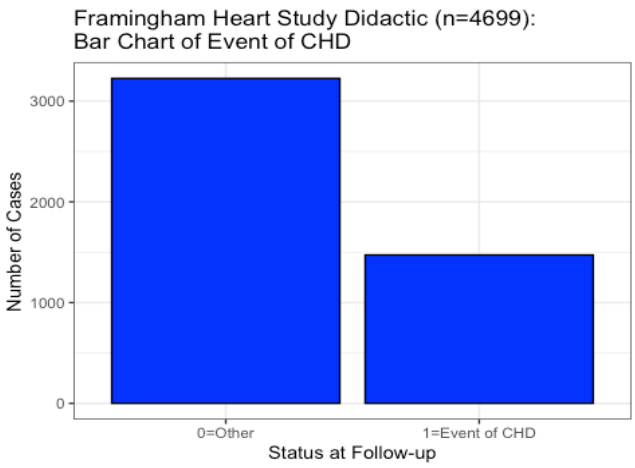
```
stargazer(framinghamdf[c("bmi","age")],type="text",summary.stat=c("n","mean","sd", "min", "max"))
```

```
##
## ===========================================
## Statistic   N    Mean  St. Dev.  Min    Max
## -------------------------------------------
## bmi       4,690 25.632  4.095   16.200 57.600
## age       4,699 46.041  8.504     30     68
## -------------------------------------------
```

3.
Single Variable Graphs

__3a. Discrete: Bar Chart

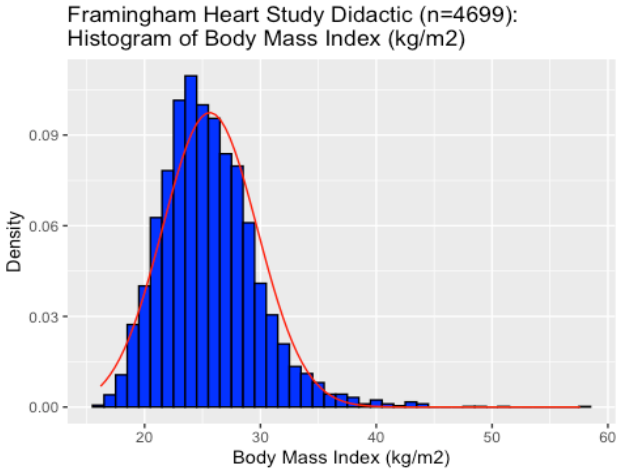
```
# SINGLE DISCRETE VARIABLE:  BAR CHART
# ggplot(data=DATAFRAME, aes(x=factor(NOMINALVARIABLE))) + geom_bar() + options

p1 <- ggplot(data=framinghamdf,aes(x=factor(chdfate))) +
    geom_bar(color="black", fill="blue",show.legend = FALSE) +
    ggtitle("Framingham Heart Study Didactic (n=4699):  \nBar Chart of Event of CHD") +
    xlab("Status at Follow-up") +
    ylab("Number of Cases") +
    theme(legend.position = "none") +
    theme_bw()
p1
```




```
# Want to save your graph?  Use the command ggsave( ) to save to your working directory.
# ggsave(file="NAME.EXTENSION", ROBJECTGRAPHNAME, options)
ggsave(file="barchart.tiff",p1, width=7, height=5, units="in")
```

_3b. Continuous: Histogram (I added an overlay normal for fun!)

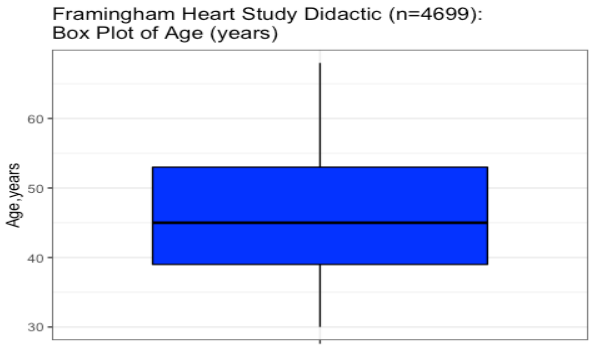
```
# SINGLE CONTINUOUS VARIABLE: HISTOGRAM WITH OVERLAY NORMAL
# ggplot(data=DATAFRAME, aes(x=CONTINUOUSVARIABLE)) + geom_histogram() + stat_function() + options
# TIP:  For overlay normal be sure to include option na.rm=TRUE in mean and variance calculations

p2 <- ggplot(data=framinghamdf, aes(x=bmi)) +
    geom_histogram(binwidth=1, colour="black", fill="blue", aes(y=..density..)) +
    stat_function(fun=dnorm, colour="red",
                  args=list(mean=mean(framinghamdf$bmi,na.rm=TRUE),sd=sd(framinghamdf$bmi, na.rm=TRUE))) +
    ggtitle("Framingham Heart Study Didactic (n=4699):  \nHistogram of Body Mass Index (kg/m2)") +
    xlab("Body Mass Index (kg/m2)") +
    ylab("Density") +
    theme_bw() +
    theme(axis.text=element_text(size=10),
          axis.title=element_text(size=10),
          plot.title=element_text(size=12))
```



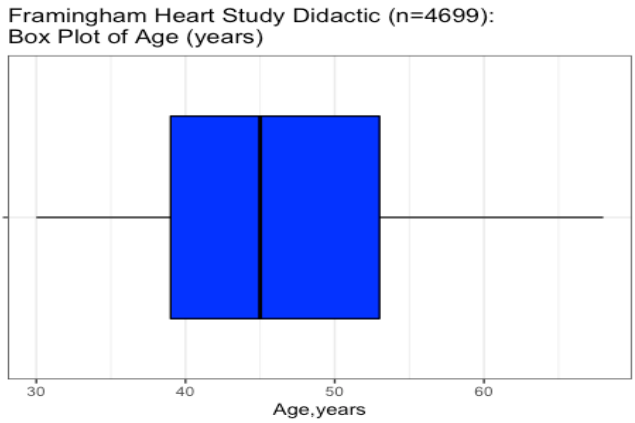

```
ggsave(file="histogram.tiff",p2, width=7, height=5, units="in")
```

_3c. Continuous: Box Plot

```
# SINGLE CONTINUOUS VARIABLE:  BOX PLOT - Vertical
# ggplot(data=DATAFRAME, aes(x="",y=CONTINUOUSVARIABLE)) + geom_boxplot

p3 <-ggplot(data=framinghamdf, aes(x="", y=age)) +
    geom_boxplot(color="black", fill="blue") +
    xlab("") +
    ylab("Age,years") +
    ggtitle("Framingham Heart Study Didactic (n=4699):  \nBox Plot of Age (years)") +
    theme_bw()
p3
```



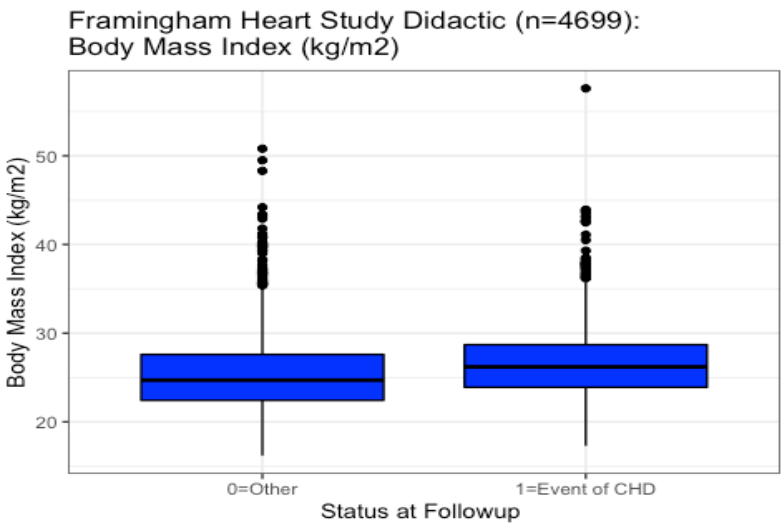
```
# SINGLE CONTINUOUS VARIABLE:  BOX PLOT - Horizontal
# ggplot(data=DATAFRAME, aes(x="",y=CONTINUOUSVARIABLE)) + geom_boxplot + coord_flip()
p4 <-ggplot(data=framinghamdf, aes(x="", y=age)) +
    geom_boxplot(color="black", fill="blue") +
    coord_flip() +
    xlab("") +
    ylab("Age,years") +
    ggtitle("Framingham Heart Study Didactic (n=4699):  \nBox Plot of Age (years)") +
    theme_bw()
p4
```



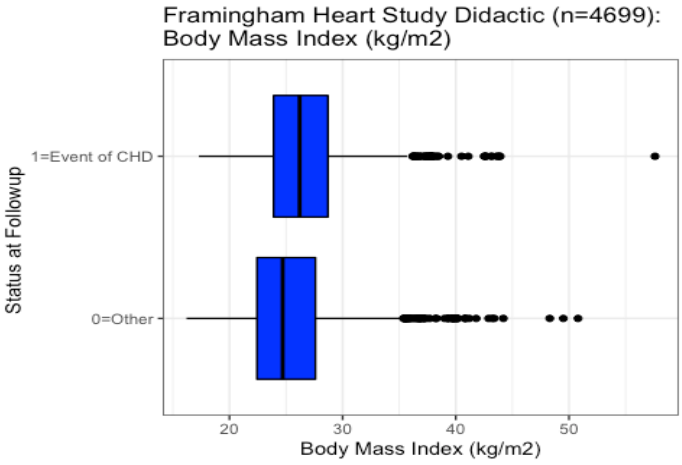
4.
Multiple Variable Graphs

_4a. Continuous, by Group (Discrete): Side-by-Side Box Plot

```
# CONTINUOUS VARIABLE, BY GROUP:  SIDE-BY-SIDE BOX PLOT - Vertical
# ggplot(data=DATAFRAME, aes(x=factor(DISCRETEVARIABLE),y=CONTINUOUSVARIABLE)) + geom_boxplot() + options

p5 <-ggplot(data=framinghamdf, aes(x=factor(chdfate), y=bmi)) +
    geom_boxplot(color="black", fill="blue") +
    ggtitle("Framingham Heart Study Didactic (n=4699):  \nBody Mass Index (kg/m2)") +
    xlab("Status at Followup ") +
    ylab("Body Mass Index (kg/m2)") +
    theme(legend.position = "none") +
    theme_bw()
p5
```



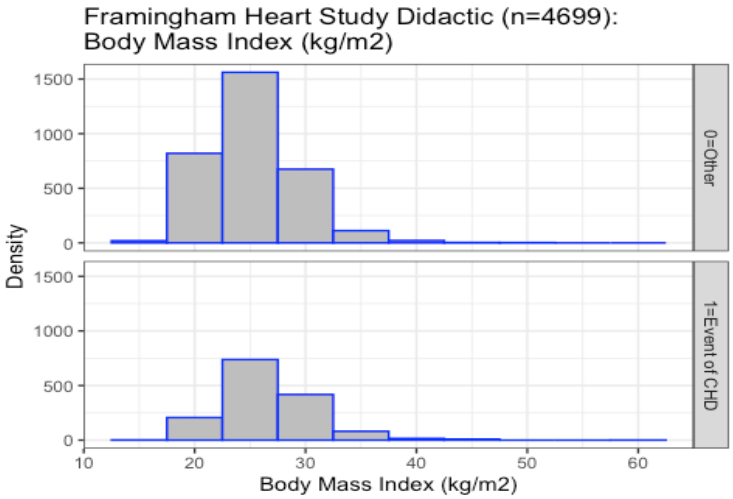

```
# CONTINUOUS VARIABLE, BY GROUP:  SIDE-BY-SIDE BOX PLOT - Horizontal
# ggplot(data=DATAFRAME, aes(x=factor(DISCRETEVARIABLE),y=CONTINUOUSVARIABLE)) + geom_boxplot() +
# coord_flip() + options

p6 <-ggplot(data=framinghamdf, aes(x=factor(chdfate), y=bmi)) +
    geom_boxplot(color="black", fill="blue") +
    ggtitle("Framingham Heart Study Didactic (n=4699):  \nBody Mass Index (kg/m2)") +
    xlab("Status at Followup ") +
    ylab("Body Mass Index (kg/m2)") +
    coord_flip() +
    theme(legend.position = "none") +
    theme_bw()
p6
```



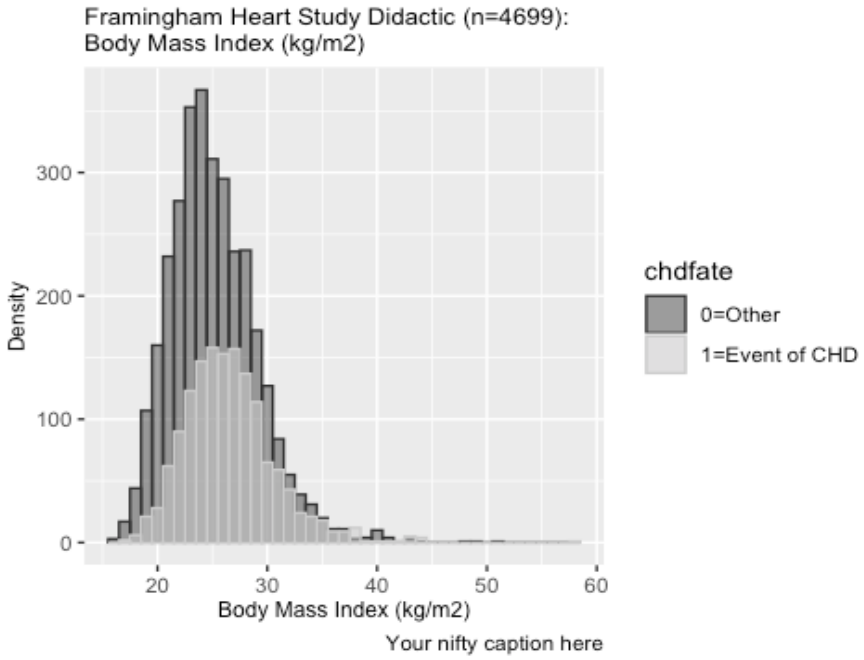

```
ggsave(file="boxplot_vertical.tiff", p5, width=7, height=5, units="in")
```

_4b. Continuous, by Group (Discrete): Side-by-Side Histogram

```
# CONTINUOUS VARIABLE, BY GROUP:  SIDE-BY-SIDE BOX HISTOGRAM - Separate Panels
# ggplot(data=framinghamdf, aes(x=bmi)) + geom_histogram() + facet_grid(GROUPVARIABLE ~ .) + options

p7 <-ggplot(data=framinghamdf, aes(x=bmi)) +
    geom_histogram(binwidth=5, color="blue", fill="grey") +
    facet_grid(chdfate ~ .) +
    scale_color_grey() +scale_fill_grey() +
    ggtitle("Framingham Heart Study Didactic (n=4699):  \nBody Mass Index (kg/m2)") +
    xlab("Body Mass Index (kg/m2)") +
    ylab("Density") +
    theme(axis.title=element_text(size=9),
                plot.title=element_text(size=10)) +
    theme_bw()

p7
```



```
# CONTINUOUS VARIABLE, BY GROUP:  SIDE-BY-SIDE BOX HISTOGRAM - Overlay, slight transparency
# ggplot(data=framinghamdf, aes(x=CONTINUOUSVARIABLE,fill=GROUPVARIABLE,color=GROUPVARIABLE)) +
#     geom_histogram(binwidth=#,position="identity",alpha=0.5) + options
p8 <-ggplot(data=framinghamdf, aes(x=bmi, fill=chdfate, color=chdfate)) +
    geom_histogram(binwidth=1,position="identity", alpha=0.5) +
    ggtitle("Framingham Heart Study Didactic (n=4699):  \nBody Mass Index (kg/m2)") +
    labs(y="Density", x="Body Mass Index (kg/m2)",caption="Your nifty caption here") +
    scale_color_grey()+scale_fill_grey() +
    theme(axis.title=element_text(size=9),
                plot.title=element_text(size=10))
p8
```

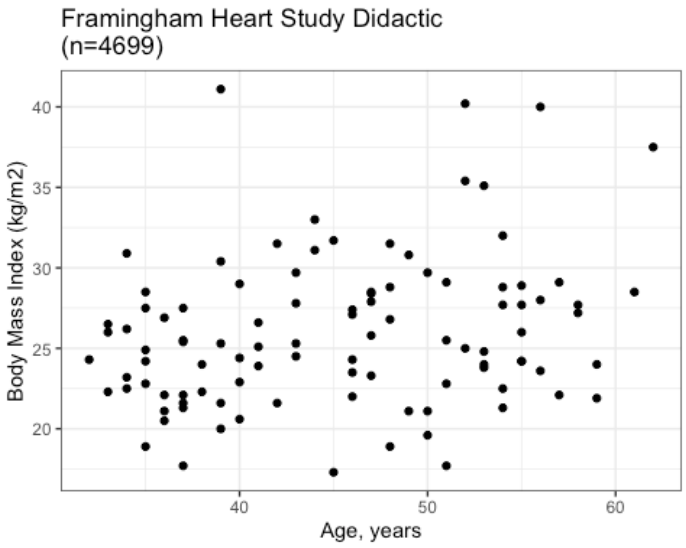


ASIDE: For the next plots I want to work with a random sample size of n=100 from my dataframe. In the command that follows, I take a random sample of n=100 and store this in a new dataframe smalldf.

```
# KEY:
# NEWDATATFRAME <- OLDDATAFRAME[sample(nrow(OLDDATAFRAME), 100), ]
smalldf <- framinghamdf[sample(nrow(framinghamdf),100), ]
```

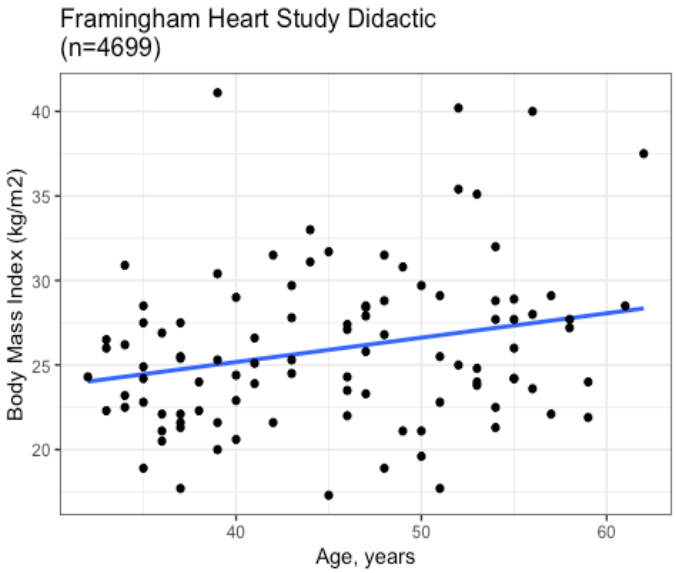
_4c. Continuous: X-Y Plot (Scatterplot)

```
# TWO CONTINUOUS VARIABLES:  XY SCATTERPLOT
# ggplot(data=DATAFRAME, aes(x=XVARIABLE, y=YVARIABLE)) + geom_point() + options

p9 <- ggplot(data=smalldf, aes(x=age,y=bmi)) +
    geom_point() +
    xlab("Age, years") +
    ylab("Body Mass Index (kg/m2)") +
    ggtitle("Framingham Heart Study Didactic \n(n=4699)") +
    theme_bw()
p9
```



_4d. Continuous: X-Y Plot (Scatterplot), with Overlay Linear Regression Model Fit

```
# TWO CONTINUOUS VARIABLES:  XY SCATTERPLOT with OVERLAY LINEAR REGRESSION MODEL FIT
# TIP:  Plot linear regression first so that data points are on top
# ggplot(data=DATAFRAME, aes(x=XVARIABLE, y=YVARIABLE)) + geom_point() + options

p10 <- ggplot(data=smalldf, aes(x=age,y=bmi)) +
    geom_smooth(method=lm, se=FALSE) +
    geom_point() +
    xlab("Age, years") +
    ylab("Body Mass Index (kg/m2)") +
    ggtitle("Framingham Heart Study Didactic \n(n=4699)") +
    theme_bw()
p10
```



_4e. Continuous: X-Y Plot, by Group (Discrete)

```
# TWO CONTINUOUS VARIABLES, GROUP:  XY SCATTERPLOT – BY GROUP
# ggplot(data=DATAFRAME, aes(x=XVARIABLE, y=YVARIABLE, color=GROUPVARIABLE)) + geom_point() + options

p11 <- ggplot(data=smalldf, aes(x=age,y=bmi,color=chdfate)) +
    geom_point() +
    xlab("Age, years") +
    ylab("Body Mass Index (kg/m2)") +
    ggtitle("Framingham Heart Study Didactic \n(n=4699)") +
    theme_bw()
p11
```