

BIOSTATS 540 – Introductory Biostatistics
Fall 2022
Introduction to R
04 – Introduction to { ggplot2 }

Introduction.

Who doesn't like producing a pretty graph. There are lots of ways to do this in R. Using the package ggplot2 is one of the best. In this lesson, I will introduce you to how to make some basic graphs using the package **{ggplot2}**.

Dataset Used: **framingham_didactic.xlsx**

Packages Used: **tidyverse, ggplot2, ggExtra**

		Page
1	Introduction to the Framingham Heart Study Didactic Data: framingham_didactic.xlsx	2
2	Introduction to ggplot2	3
3	Single Discrete Variable: Bar Chart	9
4	Single Continuous Variable: Box and Whisker Plot	10
5	XY Scatter Plot	12

1. Introduction to the Framingham Heart Study Didactic Data [framingham_didactic.xlsx](#)

Source:

The Framingham Didactic dataset was contributed by Dr. Amy Nowacki, Associate Professor, Cleveland Clinic. Please refer to this resource as: Amy S. Nowacki, “Framingham Didactic Dataset”, *TSHS Resources Portal* (2015). Available at <https://www.causeweb.org/tshs/framingham-didactic/>.

Description:

Cardiovascular disease (CVD) is the leading cause of death and serious illness in the United States. In 1948, the Framingham Heart Study - under the direction of the National Heart Institute (now known as the National Heart, Lung, and Blood Institute or NHLBI) was initiated.

The objective of the Framingham Heart Study was to identify the common factors or characteristics that contribute to 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.

In this illustration, we will explore the fitting of a multiple predictor model to a different transformation of systolic blood pressure. $Y = -1/\sqrt{\text{sbp}}$. The predictors of interest will be sex (sex), body mass index (bmi), serum cholesterol (scl), and age (age).

Data dictionary/Codebook

Position	Variable	Label	Type	Codings
1	id	Patient identifier		
2	sex	Sex at birth	numeric	1=male 2=female
3	sbp	Systolic Blood Pressure (mm Hg)	numeric	
4	dbp	Diastolic Blood Pressure (mm Hg)	numeric	
5	scl	Serum cholesterol (mg/100 ml)	numeric	
6	age	Age at baseline exam, years	numeric	
7	bmi	Body Mass index (kg/m ²)	numeric	
8	month	Month of year of baseline exam	numeric	
9	followup	Follow-up: days since baseline	numeric	
10	chdfate	Event of CHD at follow up	numeric	1 = event had developed CHD 0 = otherwise

2. Introduction to ggplot2

The **grammar of a ggplot** consists of:

- 1st- Some minimal components (required); plus
- 2nd – Additional specifications (as you like, but not required)

Grammar of a ggplot graph

Required (minimal) components:

<code>data =</code>	data to use (Tip – this must be a dataframe)
<code>aes()</code>	define x-axis, y-axis, 3 rd variable, as appropriate
<code>geom_XXX()</code>	type of plot (e.g., dot, box, scatterplot, etc.)

Additional Specifications:

<code>aesthetic</code>	aesthetics pertaining to position, size, color, shape, fill, etc.
<code>scale</code>	customize scale; e.g. – log scale, color scale, size, shape
<code>stat =</code>	show statistics (e.g., regression line, overlay normal)
<code>facets</code>	plot by group or create multi-panel graphs.

The easiest way (and best for trouble shooting) to build a ggplot plot is to proceed **layer by layer**.

How to Build Your Plot Layer by Layer:

Step 1: Attach packages. Either of the following works

```
library(ggplot2) or
library(tidyverse)
```

Step 2 (optional, recommended): Create each plot in an R Script or in its own R Markdown chunk

Step 3 (optional, recommended): Code your graph line by line, layering as you go.

First line: # Execute and correct as needed
`data = dataframename`

Second line is added # Execute and correct as needed
`data = dataframename +`
`aes()`

Third line is added: # Execute and correct as needed
`data = dataframename +`
`aes() +`
`geom_xxx()`

Tips

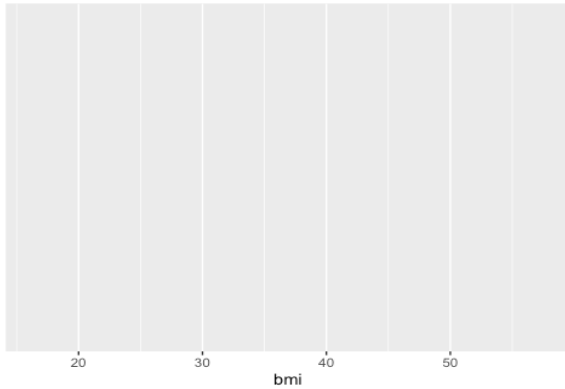
- The continuation `+` MUST BE at the end of the line (not at the start of the next line)
- As you add lines, execute all of the accumulating layers
- Check your work and correct errors, layer by layer

Illustration of Layer by Layer

Note: In what follows the red color of the + at the end of each line is my coloring for emphasis

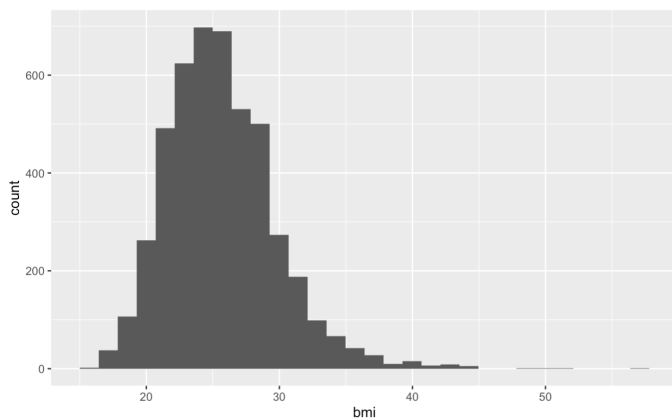
```
# Attach ggplot2 or tidyverse
library(ggplot2)

# Minimal Components: data= and aes( )
# Tell R which data to use. Specify single variable for histogram
ggplot(data=framinghamdf) +
  aes(x=bmi)
```



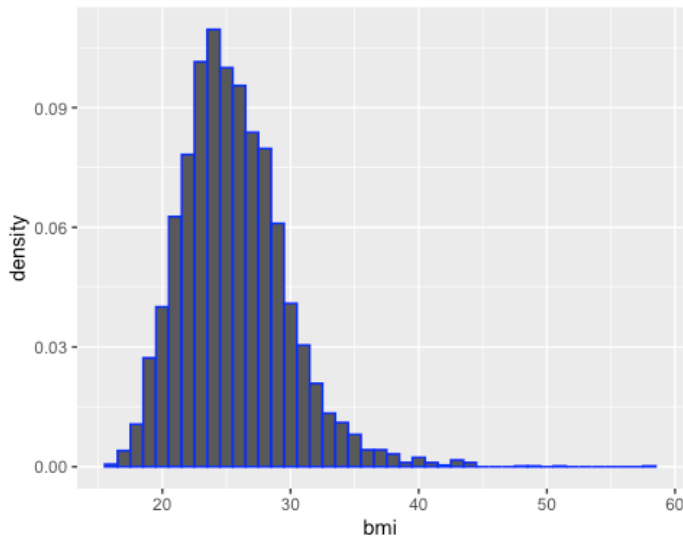
Yes, a blank plot to start!

```
# Next Layer: + geom_XXX( )
# Tell R which kind of plot to produce
ggplot(data=framinghamdf) +
  aes(x=bmi) +
  geom_histogram( )
```



geom_histogram default plot

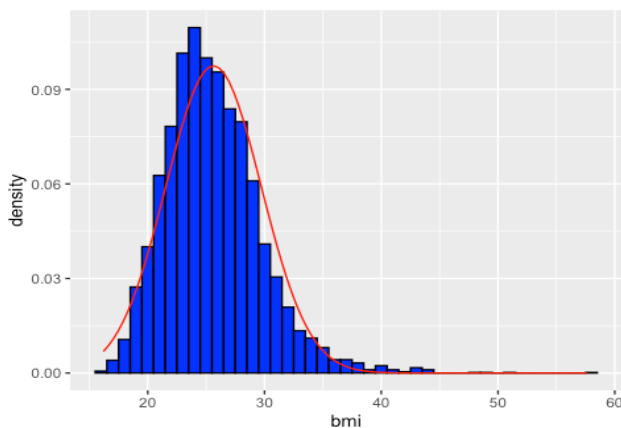
```
# Along the way: fine tune geom_histogram specifications
# binwidth, color, y-axis
ggplot(data=framinghamdf) +
  aes(x=bmi) +
  geom_histogram(binwidth=1, colour="blue", aes(y=..density..))
```



This looks a little better...

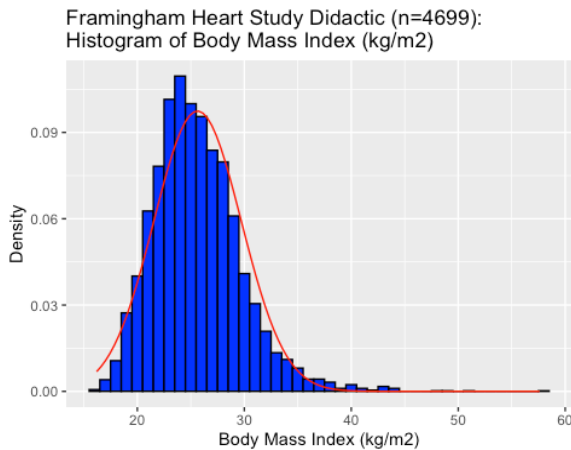
```
# Next Layer: + stat_function( )
# Note: This is actually an argument of the geom_SOMETHING( )
# Here we are telling R overlay a statistical calculation, in particular an overlay normal curve
# Tip: Be sure to include the option na.rm=TRUE. Reason: Calculations will not
# happen if there are missing values

ggplot(data=framinghamdf) +
  aes(x=bmi) +
  geom_histogram(binwidth=1, colour="blue", aes(y=..density..)) +
  stat_function(fun=dnorm, color="red", args=list(mean=mean(framinghamdf$bmi, na.rm=TRUE),
    sd=sd(framinghamdf$bmi, na.rm=TRUE)))
```



Pretty!

```
# Next Layers: + ADD TITLE, LABELS, AXIS LIMITS, etc
ggplot(data=framinghamdf) +
  aes(x=bmi) +
  geom_histogram(binwidth=1, colour="blue", aes(y=..density..)) +
  stat_function(fun=dnorm, color="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")
```



Don't forget to save your graph!

How to Save Your Graph

STEP 1: Attach the packages {ggplot2}

```
library(ggplot2)
```

STEP 2: Edit your R Script or your R Markdown code so as to create your graph as an object

Create your graph as an object using whatever name you like. Don't forget. If you want to see your plot as it develops, layer by layer, you will have to print it. For example, if your graph object is named `p`, then to print it, simply type `p`

```
p <- ggplot(stuff) +
  next layer +
  next layer +
  etc
```

STEP 3: Use the command `ggsave()` to save your graph

```
ggsave(file="FILENAME.extension",p, option, option). For example
ggsave(file="mygraph.tiff",p, width=7, height=5, units="in")
```

Initialize Session

```
setwd("/cloud/project/BIOSTATS 690F")
getwd()

## [1] "/cloud/project/BIOSTATS 690F"

rm(list = ls())
```

Load framingham_didactic. Inspect

```
library(tidyverse)
library(readxl)
framingham <- read_excel("framingham_didactic.xlsx")
glimpse(framingham)

## Rows: 4,699
## Columns: 10
## $ id      <dbl> 2642, 4627, 2568, 4192, 3977, 659, 2290, 4267, 2035, 3587, 10...
## $ sex     <dbl> 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 1, 2, 1...
## $ sbp     <dbl> 120, 130, 144, 92, 162, 212, 140, 174, 142, 115, 202, 130, 13...
## $ dbp     <dbl> 80, 78, 90, 66, 98, 118, 85, 102, 94, 70, 124, 94, 88, 72, 10...
## $ scl     <dbl> 267, 192, 207, 231, 271, 182, 276, 259, 242, 242, 260, 326, 1...
## $ age     <dbl> 55, 53, 61, 48, 39, 61, 44, 39, 47, 60, 58, 47, 43, 59, 56, 5...
## $ bmi     <dbl> 25.0, 28.4, 25.1, 26.2, 28.4, 33.3, 25.3, 27.9, 26.6, 30.8, 2...
## $ month   <dbl> 8, 12, 8, 11, 11, 2, 6, 11, 5, 10, 3, 10, 1, 9, 9, 9, 8, 1...
## $ followup <dbl> 18, 35, 109, 147, 169, 199, 201, 209, 265, 278, 290, 300, 300...
## $ chdfate <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
```

Be sure to attach the package ggplot2

```
library(ggplot2)
```

Create sexf, a factor version of the variable sex

```
framingham$sexf <- factor(framingham$sex,
                          levels=c(1,2),
                          labels=c("Male", "Female"))

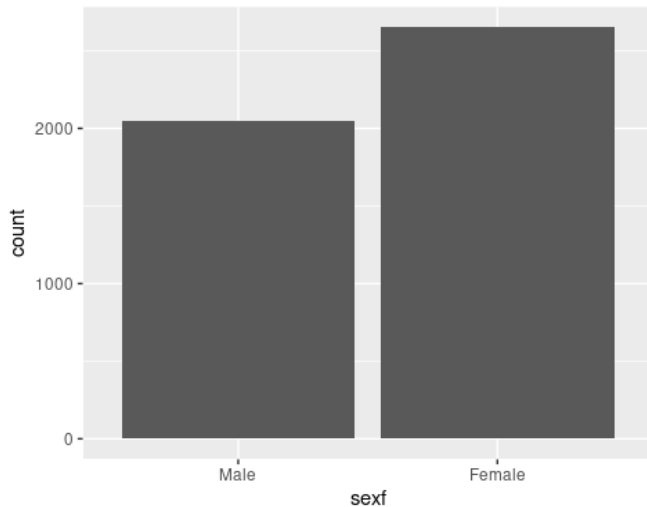
table(framingham$sex,framingham$sexf)           # show.

##
##      Male Female
## 1  2049      0
## 2     0  2650
```


3. Single Discrete Variable: Bar Chart

BAR Graph - default

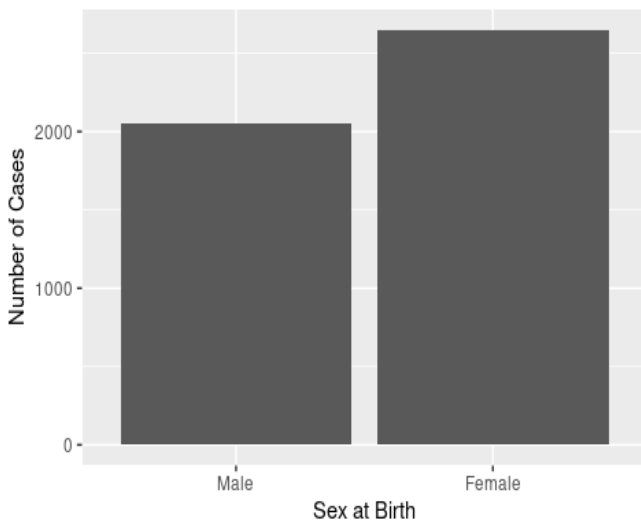
```
ggplot(data=framingham) +           # required component: data=
  aes(x=sexf) +                     # required component: aes( )
  geom_bar(na.rm=TRUE)              # required component: geom_SOMETHING
```



Bar Graph with aesthetics

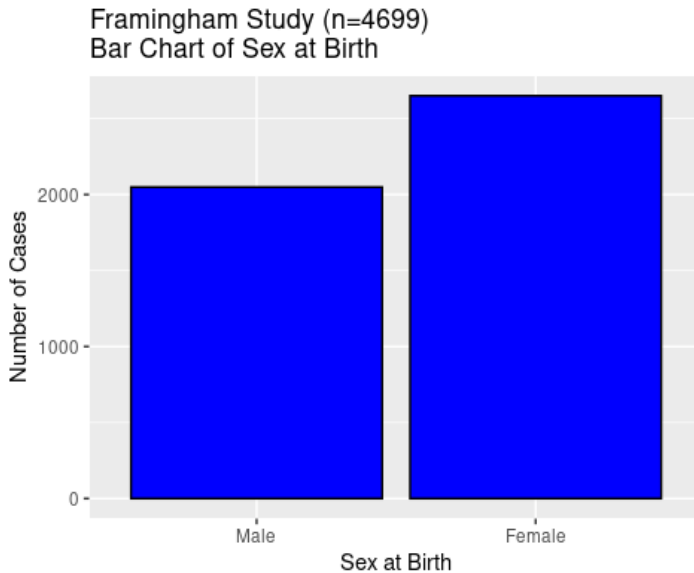
```
ggplot(data=framingham) +           # required component: data=
  aes(x=sexf) +                     # required component: aes( )
  geom_bar(na.rm=TRUE) +            # required component: geom_SOMETHING
  ggtitle("Framingham Study (n=4699)\nBar Chart of Sex at Birth") +
  xlab("Sex at Birth") +
  ylab("Number of Cases")
```

Framingham Study (n=4699)
Bar Chart of Sex at Birth



Bar Graph with Aesthetics + COLOR

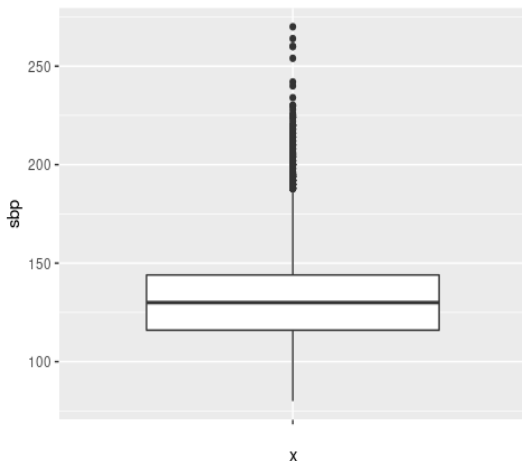
```
ggplot(data=framingham) +           # required component: data=
  aes(x=sexf) +                       # required component: aes( )
  geom_bar(na.rm=TRUE,
           color="black",
           fill="blue") +             # required component: geom_SOMETHING
  ggtitle("Framingham Study (n=4699)\nBar Chart of Sex at Birth") +
  xlab("Sex at Birth") +
  ylab("Number of Cases")
```



4. Single Continuous Variable: Box and Whisker Plot

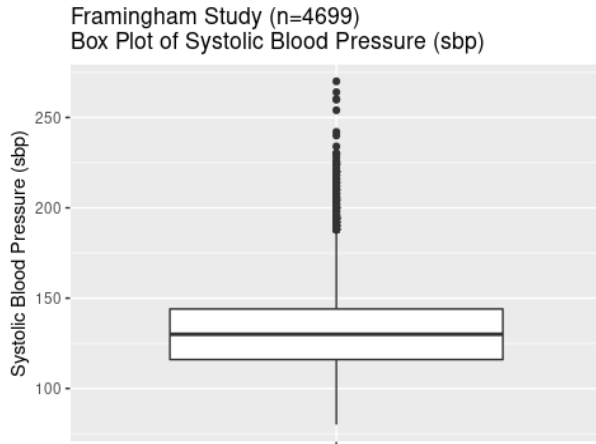
Box and Whisker Plot - default

```
ggplot(data=framingham) +           # required component: data=
  aes(x="", y=sbp) +                 # required component: aes( )
  geom_boxplot(na.rm=TRUE)           # required component: geom_SOMETHING
```

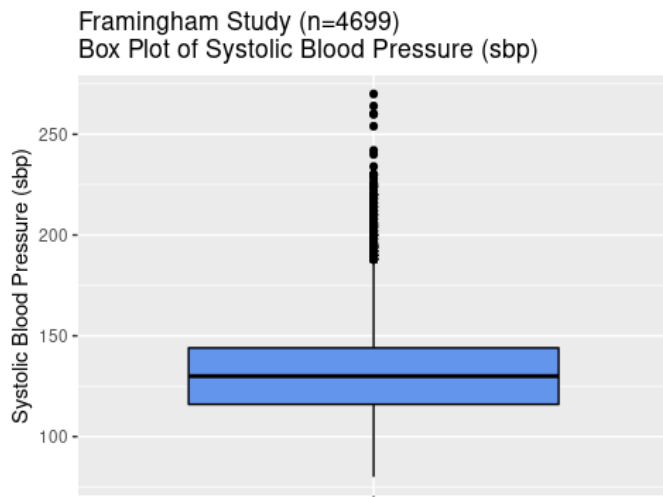


Box and Whisker Plot with Aesthetics

```
ggplot(data=framingham) +           # required component: data=
  aes(x="", y=sbp) +                 # required component: aes( )
  geom_boxplot(na.rm=TRUE) +         # required component: geom_SOMETHING
  ggtitle("Framingham Study (n=4699)\nBox Plot of Systolic Blood Pressure (sbp)") +
  xlab("") +
  ylab("Systolic Blood Pressure (sbp)")
```



```
# BOX PLOT - Single continuous variable (with optional aesthetics) + color!
ggplot(data=framingham) +           # required component: data=
  aes(x="", y=sbp) +                 # required component: aes( )
  geom_boxplot(na.rm=TRUE,
               color="black",
               fill="cornflowerblue") + # required component: geom_SOMETHING
  ggtitle("Framingham Study (n=4699)\nBox Plot of Systolic Blood Pressure (sbp)") +
  xlab("") +
  ylab("Systolic Blood Pressure (sbp)")
```



5. XY Scatterplot

Preliminary (for illustration only): Use `sample_n()` in package `{tidyverse}` to obtain random sample

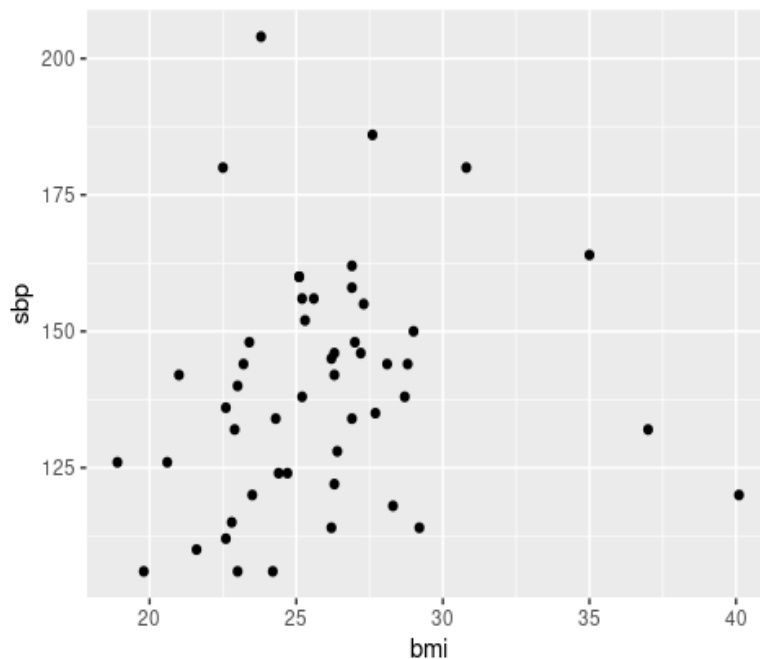
```
library(tidyverse)

temp <- framingham %>%
  sample_n(50)
glimpse(temp)

## Rows: 50
## Columns: 11
## $ id      <dbl> 1337, 3926, 112, 4201, 1925, 1696, 4646, 412, 3346, 856, 2838...
## $ sex     <dbl> 1, 2, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 1, 1, 2, 2, 2, 1, 2...
## $ sbp     <dbl> 134, 160, 138, 106, 144, 132, 155, 122, 120, 106, 134, 106, 1...
## $ dbp     <dbl> 92, 88, 85, 78, 96, 96, 100, 84, 75, 64, 82, 70, 80, 86, 98, ...
## $ scl     <dbl> 305, 167, 271, 292, 209, 165, 200, 283, 344, 241, 275, 167, 2...
## $ age     <dbl> 56, 55, 43, 57, 45, 55, 44, 34, 54, 51, 53, 40, 55, 37, 43, 6...
## $ bmi     <dbl> 24.3, 25.1, 25.2, 23.0, 28.8, 37.0, 27.3, 26.3, 23.5, 24.2, 2...
## $ month   <dbl> 3, 11, 1, 11, 5, 4, 12, 1, 9, 2, 8, 8, 4, 11, 2, 3, 1, 3, 8, ...
## $ followup <dbl> 5577, 10882, 7553, 5000, 10028, 5943, 11688, 11688, 9497, 504...
## $ chdfate <dbl> 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0...
## $ sexf    <fct> Male, Female, Male, Female, Male, Male, Female, Male, Female,...
```

Scatterplot - default

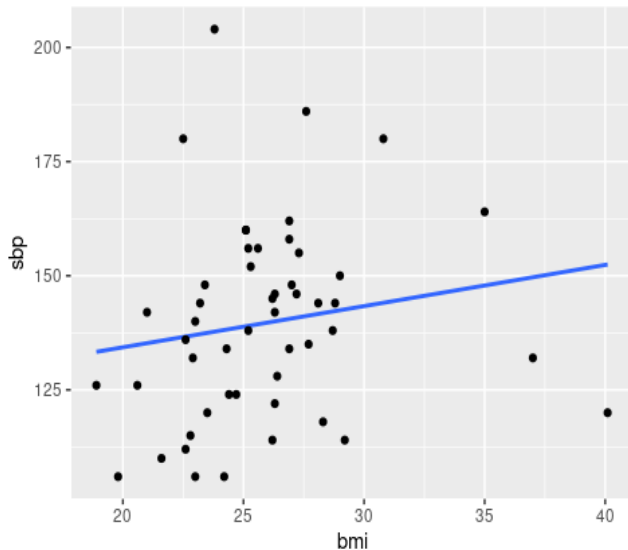
```
ggplot(data=temp) +                # required component: data=
  aes(x=bmi, y=sbp) +              # required component: aes( )
  geom_point(na.rm=TRUE)          # required component: geom_SOMETHING
```



Scatterplot with overlay linear model straight line

```
ggplot(data=temp) + # required component: data=
  aes(x=bmi, y=sbp) + # required component: aes( )
  geom_smooth(method=lm, se=FALSE) + # TIP - put line on first, then points
  geom_point(na.rm=TRUE) # required component: geom_SOMETHING

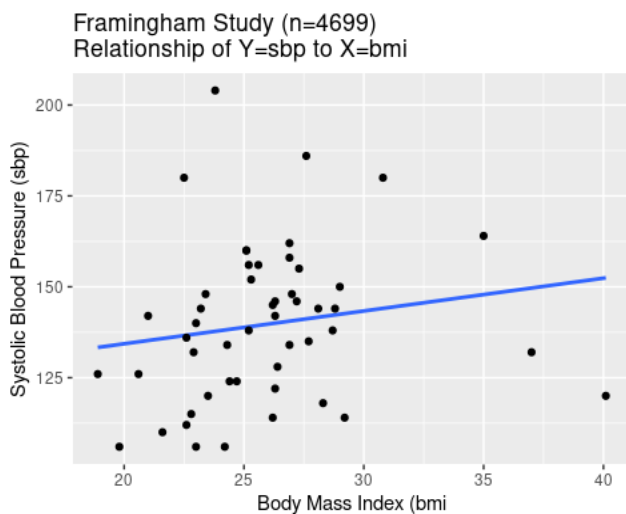
## `geom_smooth()` using formula 'y ~ x'
```



Scatterplot with overlay straight line and aesthetics

```
ggplot(data=temp) + # required component: data=
  aes(x=bmi, y=sbp) + # required component: aes( )
  geom_smooth(method=lm, se=FALSE) + # TIP - put line on first, then points
  geom_point(na.rm=TRUE) + # required component: geom_SOMETHING
  ggtitle("Framingham Study (n=4699)\nRelationship of Y=sbp to X=bmi") +
  xlab("Body Mass Index (bmi)") +
  ylab("Systolic Blood Pressure (sbp)")

## `geom_smooth()` using formula 'y ~ x'
```



Add `theme_bw()` to obtain cleaner background

```
ggplot(data=temp) + # required component: data=
  aes(x=bmi, y=sbp) + # required component: aes( )
  geom_smooth(method=lm, se=FALSE) + # TIP - put line on first, then points
  geom_point(na.rm=TRUE) + # required component: geom_SOMETHING
  ggtitle("Framingham Study (n=4699)\nRelationship of Y=sbp to X=bmi") +
  xlab("Body Mass Index (bmi)") +
  ylab("Systolic Blood Pressure (sbp)") +
  theme_bw()

## `geom_smooth()` using formula 'y ~ x'
```

