# R for Health Data Science

Week 07: Modern graphics with ggplot2

Sam Stewart

2021-04-09

```
library(Hmisc)
library(kableExtra)
library(tidyverse)
library(ggplot2)
library(boomer)


dat = read.csv("data/framinghamFirst.csv",header=TRUE,
               na.strings=".",stringsAsFactors=FALSE)
dat$BMIGroups = cut(dat$BMI,breaks=c(0,18.5,25,30,Inf),
                    labels=c("Underweight","Normal","Overweight","Obese"))
dat$SEX = factor(dat$SEX,levels=1:2,labels=c("Male","Female"))
dat$DIABETES = factor(dat$DIABETES,levels=0:1,labels=c("No Diabetes","Diabetes"))
dat$HYPERTEN = factor(dat$HYPERTEN,levels=0:1,labels=c("Normotensive","Hypertensive"))

# I want a dataset that's a bit easier to see, so I'll use dat.reduced on occasion
# in this session
set.seed(11)
dat.reduced = dat %>% sample_n(200)
```

We'll be exploring a different approach to graphics from the *tidyverse*, namely the *ggplot* library, `ggplot2`. The **G**rammar of **G**raphics library is an approach to creating graphics in a more structured, readable-way, similar to the principles behind `dplyr` and the tidyverse. As with the visualization lecture in week 3 we'll explore

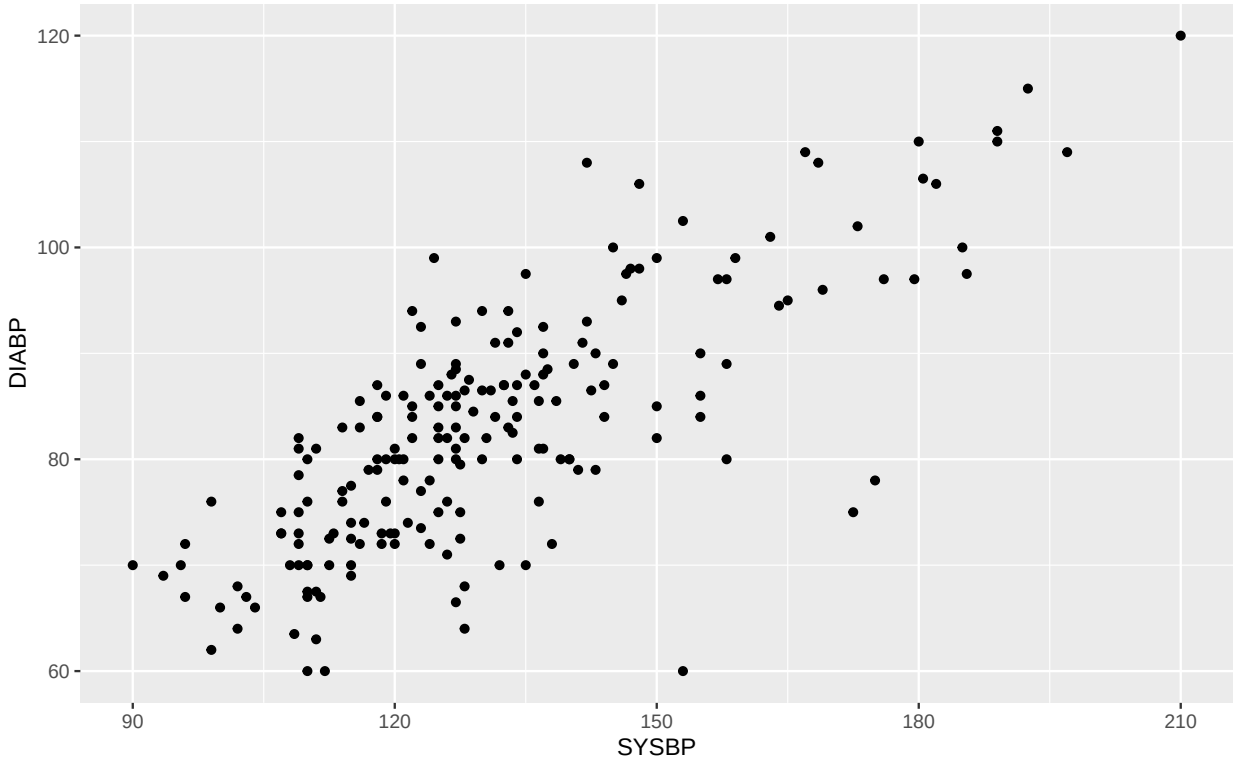
1. Line/Dot plots
2. Box/Violin plots
3. Histograms

The R4DS book, specifically chapters 3 and 28, has information on ggplot, though the library is used throughout the book for graphing. You can also check out Hadley's own website at https://ggplot2.tidyverse.org/ for a brief introduction, and more importantly two extremely useful cheat-sheets.

## 1 Structure of `ggplot`

The idea of ggplot is that you define a drawing space, and then add *aesthetics*, denoted with `aes()` that explain what to add. For example, the command below creates a scatter plot of systolic against diastolic blood pressure.

```
ggplot(dat.reduced)+
  geom_point(aes(x=SYSBP,y=DIABP))
```




The general structure is

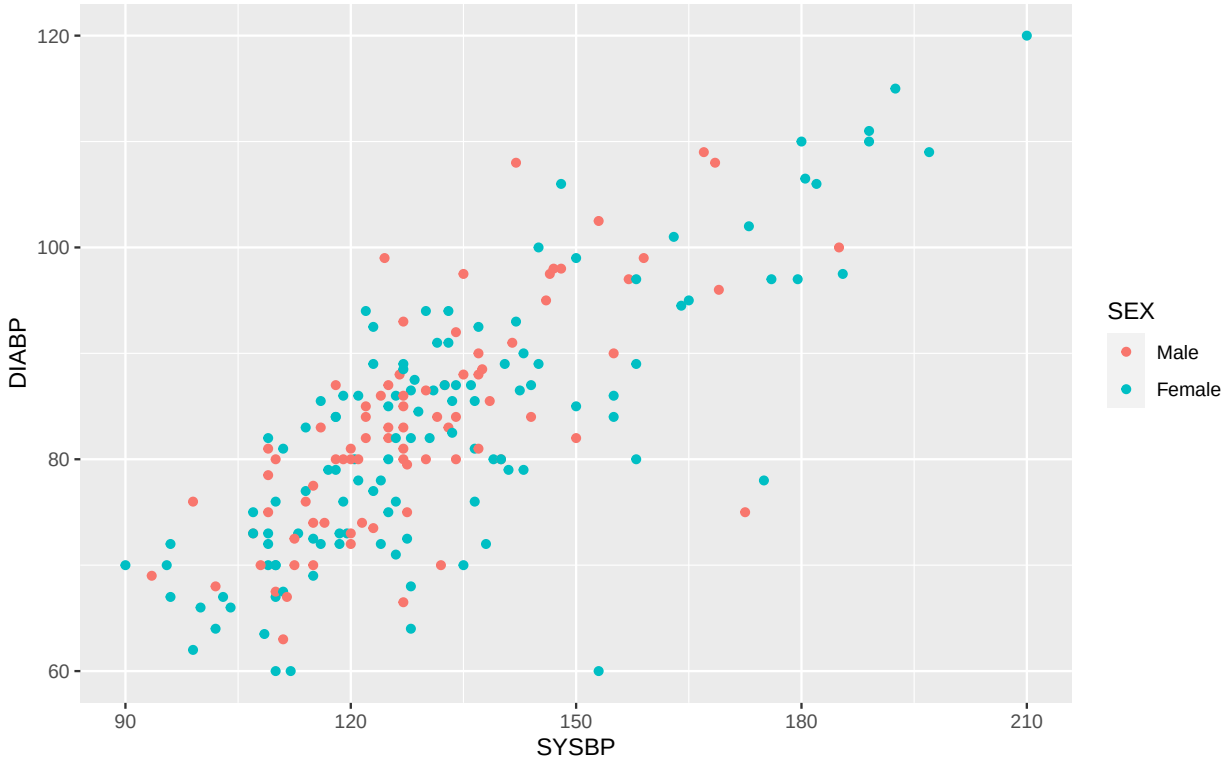
```
ggplot(data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

It has a similar assembly-line like process as `dplyr`, only were using `+` here instead of `%>%`. Note as well that you can specify the `aes()` part in the `ggplot()` function and it will then pass that to all the subfunctions

```
ggplot(dat.reduced,aes(x=SYSBP,y=DIABP))+
  geom_point()
```

The command is broken into two parts - `ggplot(dat,aes())` establishes a plotting region - using the dataset `dat.reduced` and defining the x and y axes. The second part adds the scatter plot. Note that ggplot always has the dataset first, so it can work with `dplyr`.

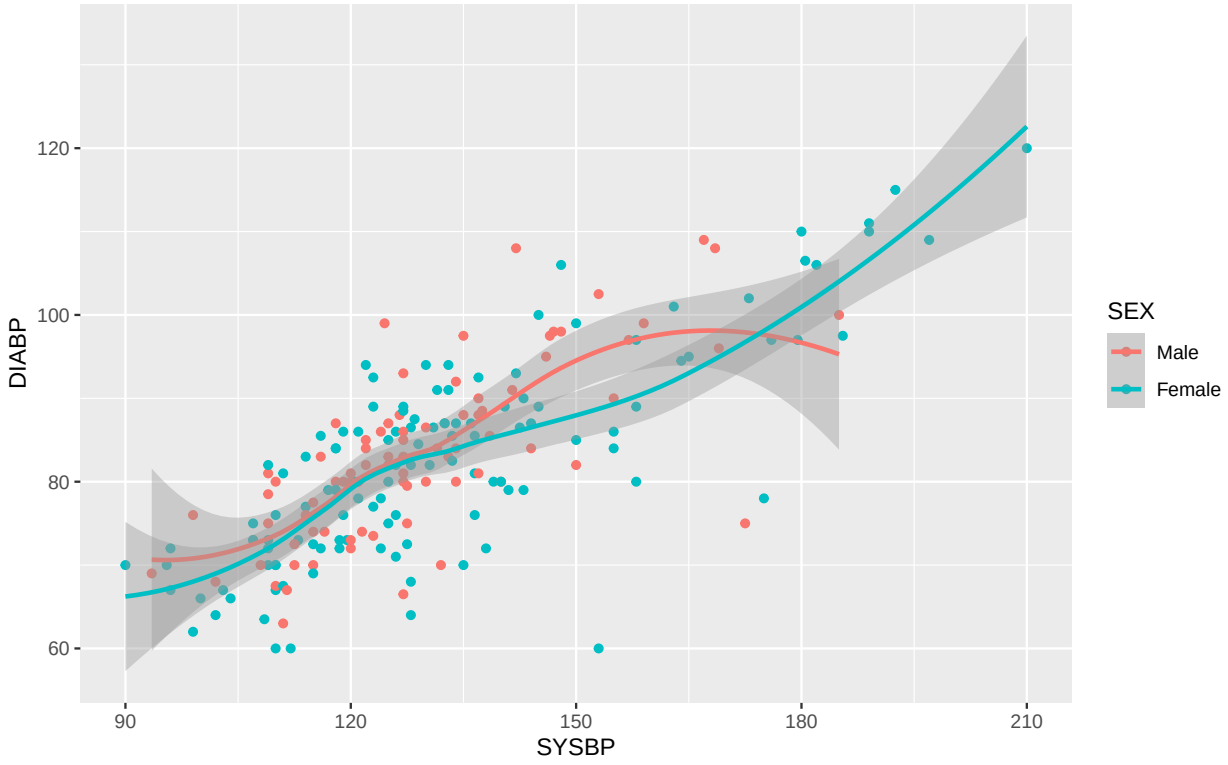
```
dat.reduced%>%
  ggplot(aes(x=SYSBP,y=DIABP))+
  geom_point(aes(color=SEX))
```




In this command I've piped the dataset in, along with coloring the dots by the variable SEX. When you add a component you can add to the `aes()` command with new values, though don't try to over-write them (if you try `geom_point(aes(x=TOTCHOL,color=SEX))`) you'll get a weird result).

The nice thing about ggplot is how it's built in parts. We're going to modify this one in parts, adding lines of best fit, and then modifying the axis titles.

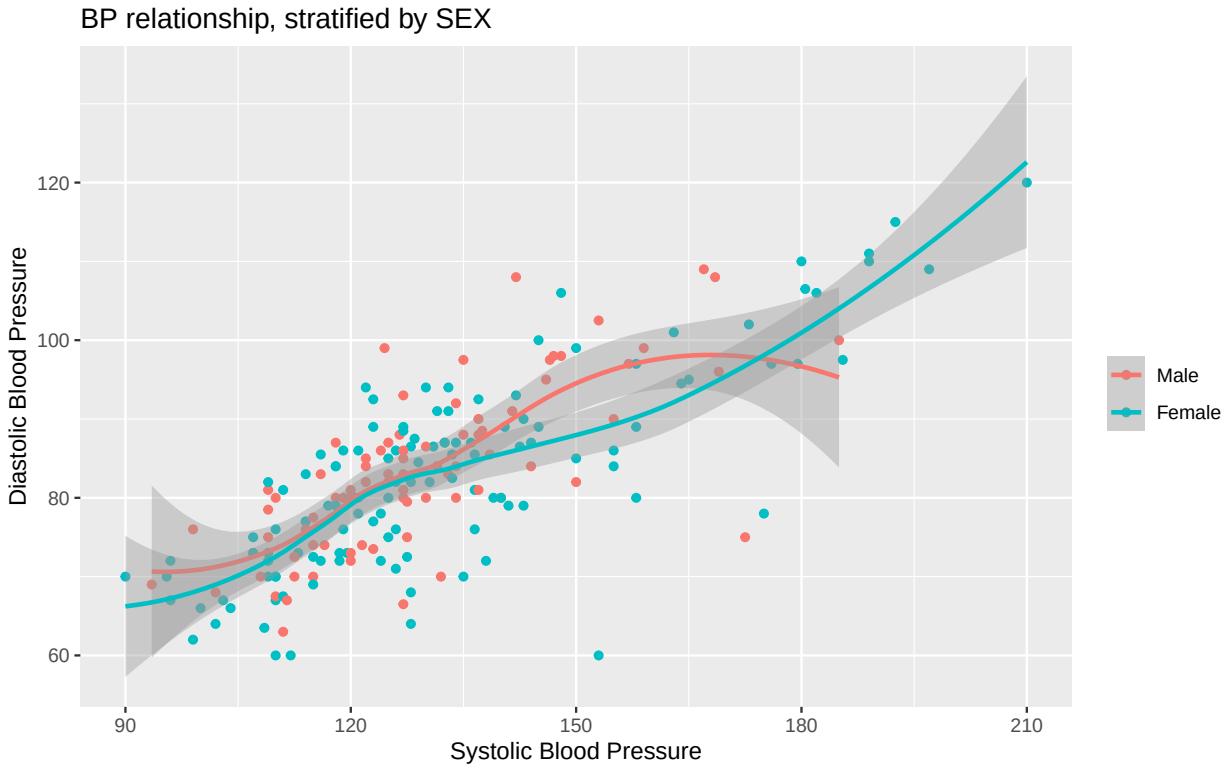
```
dat.reduced%>%
  ggplot(aes(x=SYSBP,y=DIABP,color=SEX,group=SEX))+
  geom_point()+
  geom_smooth()
```




Note that I've added the color command to the base `ggplot` function - that's because I want both `geom_point` and `geom_smooth` to make use of it. I also added a `group` variable also grouping by SEX. This is common when you want to stratify a plot

- `group` tells the function that the data is stratified by a variable but doesn't colour it that way
- `color` tells you to color the dots but doesn't split the data.

```
dat.reduced%>%
  ggplot(aes(x=SYSBP,y=DIABP,color=SEX,group=SEX))+
  geom_point()+
  geom_smooth()+
  labs(x="Systolic Blood Pressure",
       y='Diastolic Blood Pressure',
       title='BP relationship, stratified by SEX',
       color='')
```

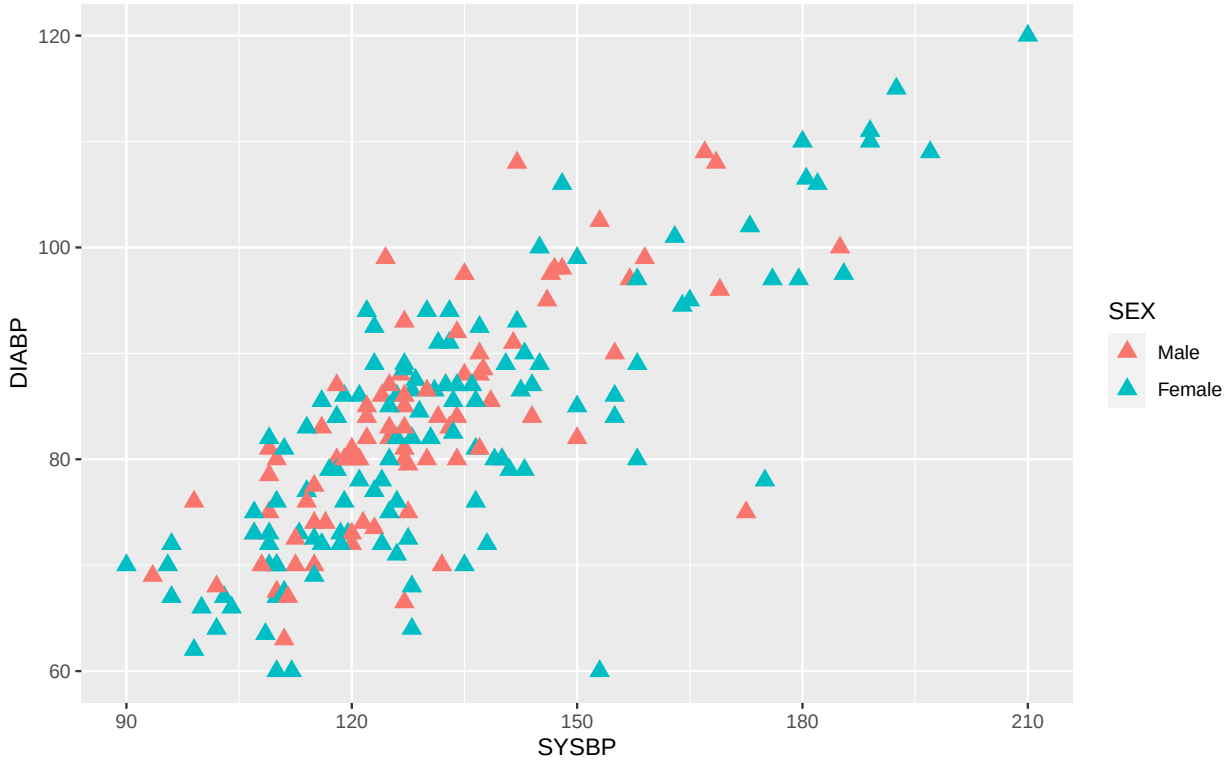



`labs()` is used to control the labels, and can assign a new title to any of the `aes` components, as well as `title, subtitle, caption`. There are also specific functions called `xlab(), ylab(), ggtitle()` if you want to target a single text element.

## 2 Line/Dot Plots

`geom_point` and `geom_smooth` are the best ways to draw dot plots (as shown above). For `geom_point` we've covered the effect of `x, y, group, colour` above, but in addition `shape` and `size` are useful as well - `shape` being the equivalent of `pch` and size matching `cex`.

```
dat.reduced%>%
  ggplot(aes(x=SYSBP,y=DIABP,color=SEX,group=SEX))+
  geom_point(size=3,shape=17)
```




```
dat.reduced%>%
  ggplot(aes(x=SYSBP,y=DIABP,color=SEX,group=SEX))+
  geom_point(aes(size=BMIGroups,shape=SEX))
```

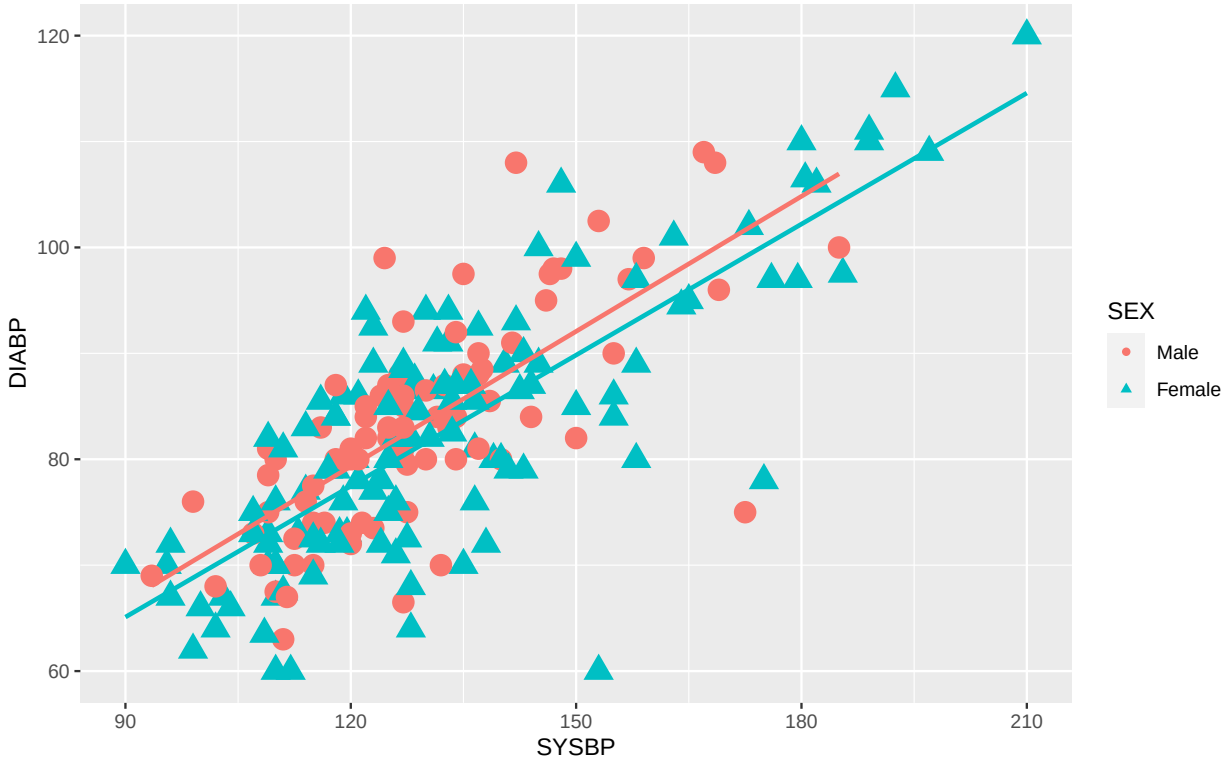
As you can see `size` and `shape` can either be a constant or a variable, depending on your need.

`geom_smooth` is used to add lines to the plot - `loess` is the default method (which is essentially a rolling average, details here). Other methods you might consider are `lm` and `glm`.

```
dat.reduced%>%
  ggplot(aes(x=SYSBP,y=DIABP,color=SEX,group=SEX))+
  geom_point(aes(shape=SEX,size=2))+
  guides(size=FALSE)+
  geom_smooth()
```


The default also has confidence intervals included, they can be surpressed with the `se` option.

```
dat.reduced%>%
  ggplot(aes(x=SYSBP,y=DIABP,color=SEX,group=SEX))+
  geom_point(aes(shape=SEX,size=2))+
  guides(size=FALSE)+
  geom_smooth(se=FALSE,method='lm',show.legend=FALSE)
```




Note that I have also included the command `show.legend=FALSE` here - ggplot tends to add everything to the legend, but you can surpress certain parts if desired.

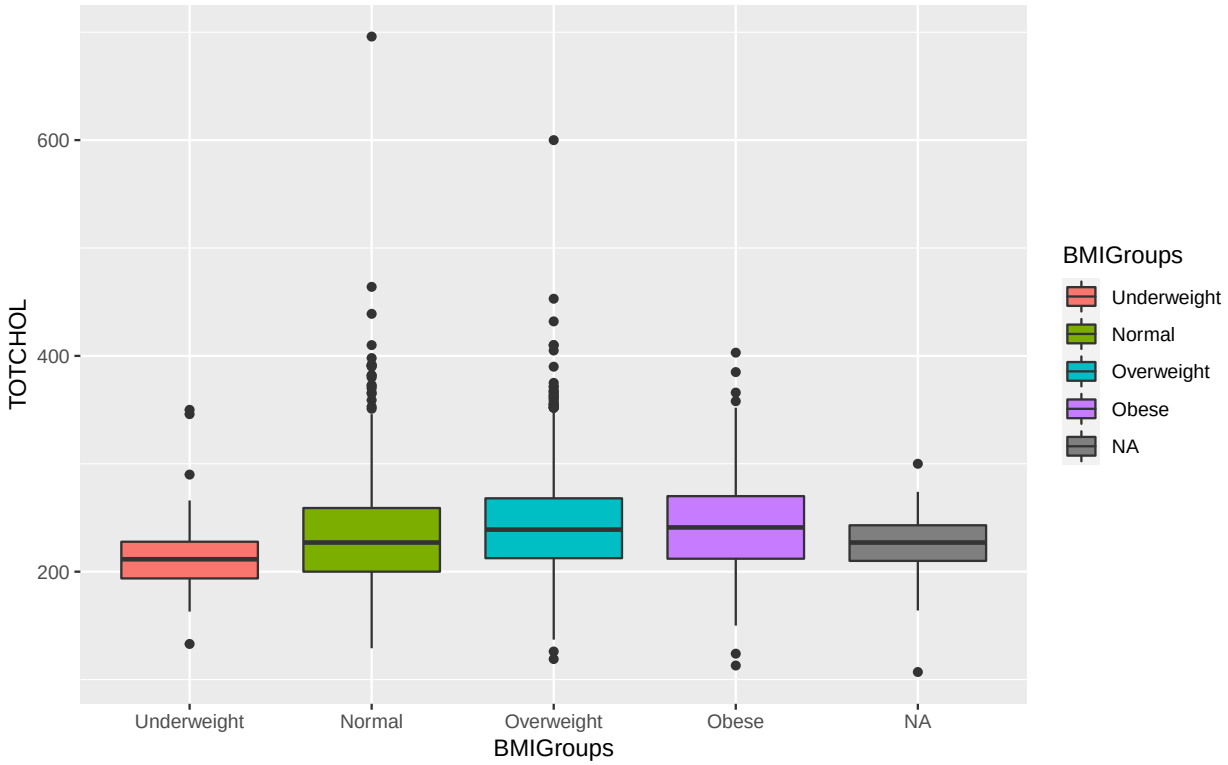
# 3 Box and Violin Plots

```
geom_boxplot(
  mapping = NULL,
  data = NULL,
  stat = "boxplot",
  position = "dodge2",
  ...,
  outlier.colour = NULL,
  outlier.color = NULL,
  outlier.fill = NULL,
  outlier.shape = 19,
  outlier.size = 1.5,
  outlier.stroke = 0.5,
  outlier.alpha = NULL,
  notch = FALSE,
  notchwidth = 0.5,
  varwidth = FALSE,
  na.rm = FALSE,
  orientation = NA,
  show.legend = NA,
  inherit.aes = TRUE
)
```

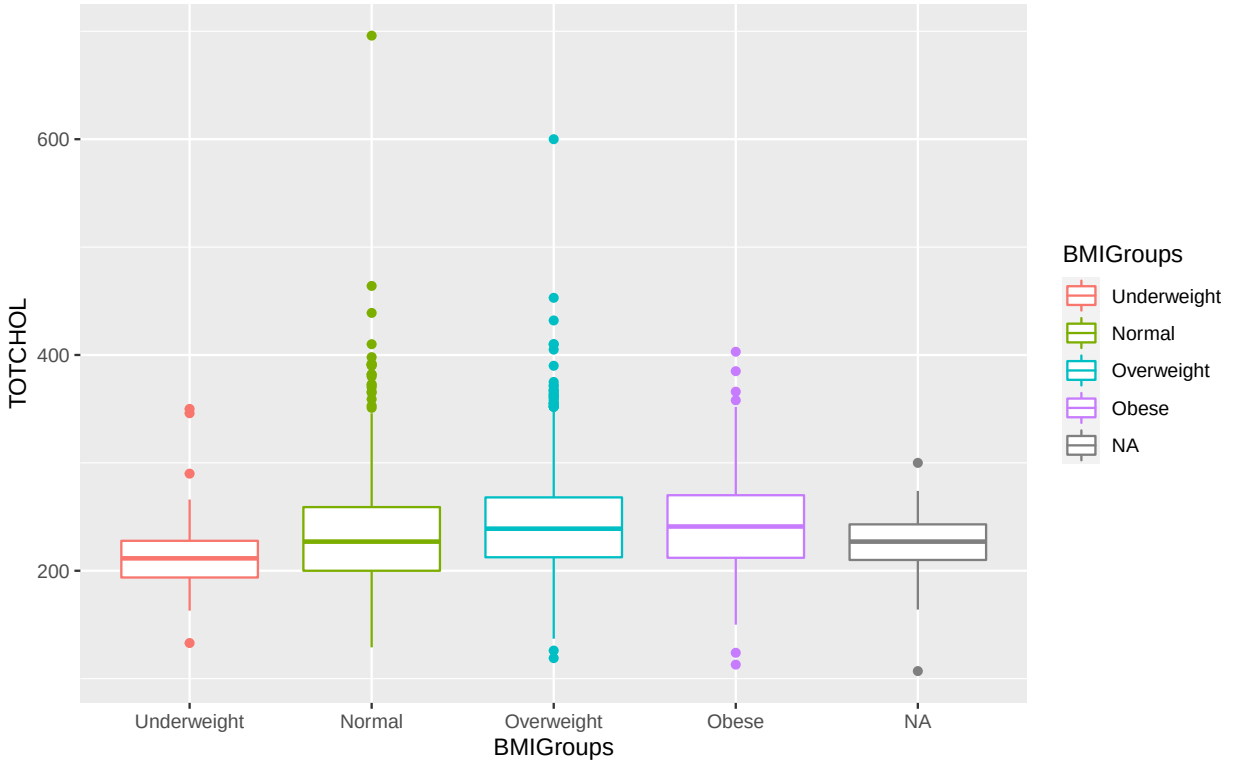
Boxplots are achieved with the command `geom_boxplot()` - the variable you're box-plotting is the `y` value and if you want to stratify that goes on the `x` variable.

Boxplot takes more arguments than most, though most are about management of outliers. The `fill` argument is useful for filling in the boxplots, though `colour` can be used instead if you want coloured outlines and white filling.

```
dat %>%
ggplot(aes(x=BMIGroups,y=TOTCHOL))+
  geom_boxplot(aes(fill=BMIGroups))
```



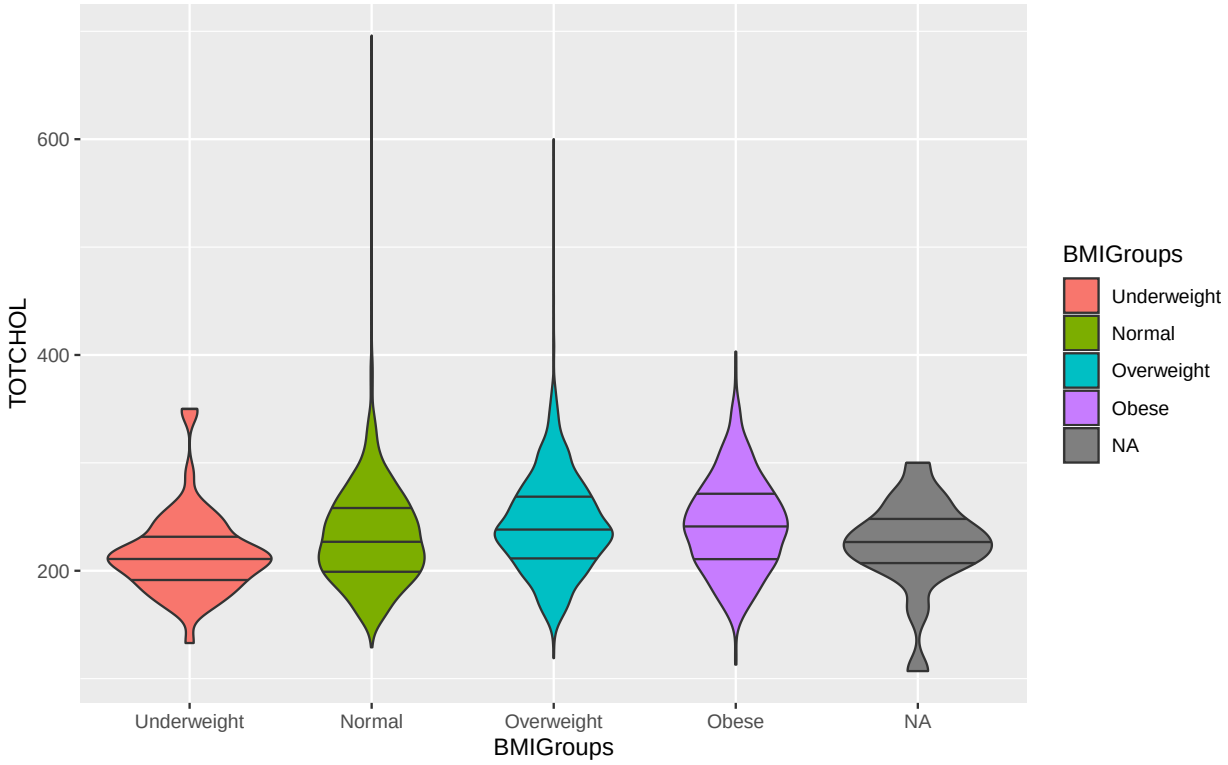

```
dat %>%
ggplot(aes(x=BMIGroups,y=TOTCHOL))+
  geom_boxplot(aes(colour=BMIGroups))
```




There is a great blog post here from the STHDA that outlines all the control elements for boxplots in ggplot2, it'll probably have any additional arguments you might need.

I tend to prefer violin plots to boxplots, since they communicate a bit more about the distribution of the variable. `geom_violin` is the command, and it takes essentially the same arguments.

```
dat %>%
  ggplot(aes(x=BMIGroups,y=TOTCHOL))+
  geom_violin(aes(fill=BMIGroups),
              draw_quantiles=seq(0,1,by=0.25))
```

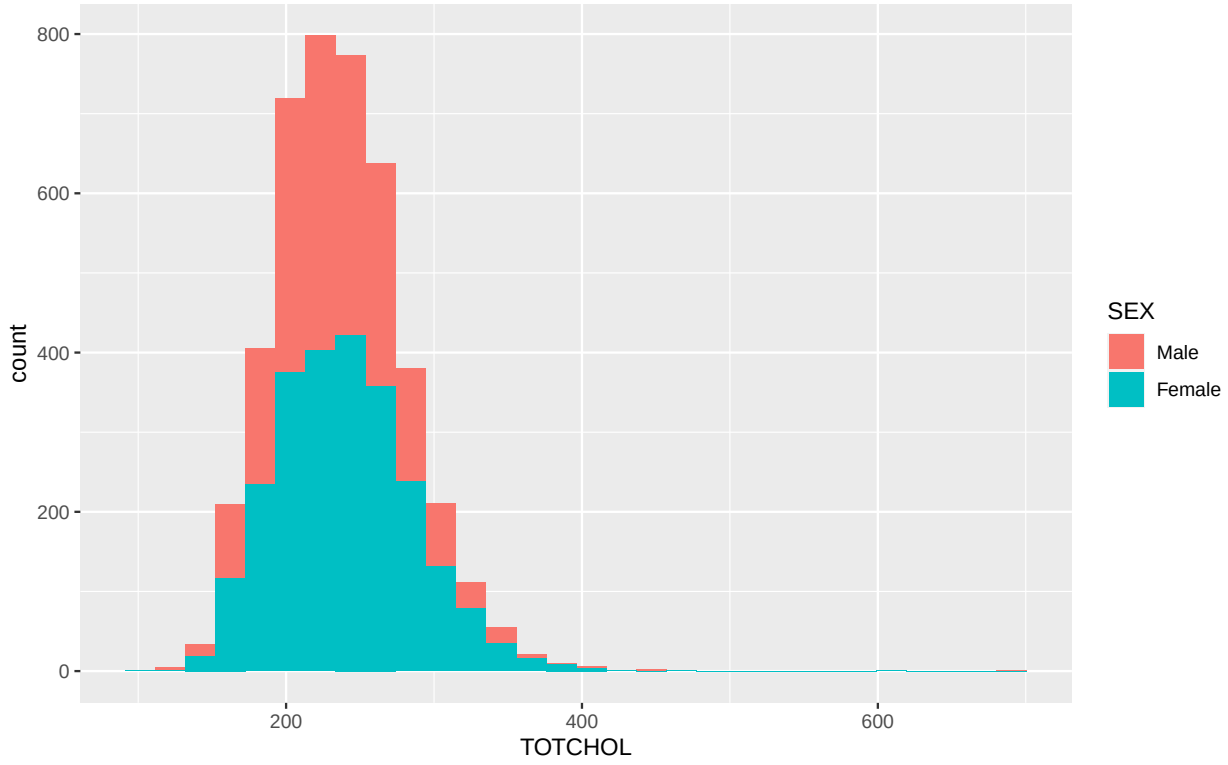



I added the `draw_quantiles` to get the median, Q1 and Q3, since those are the borders of a typical boxplot.

# 4 barplots and histograms

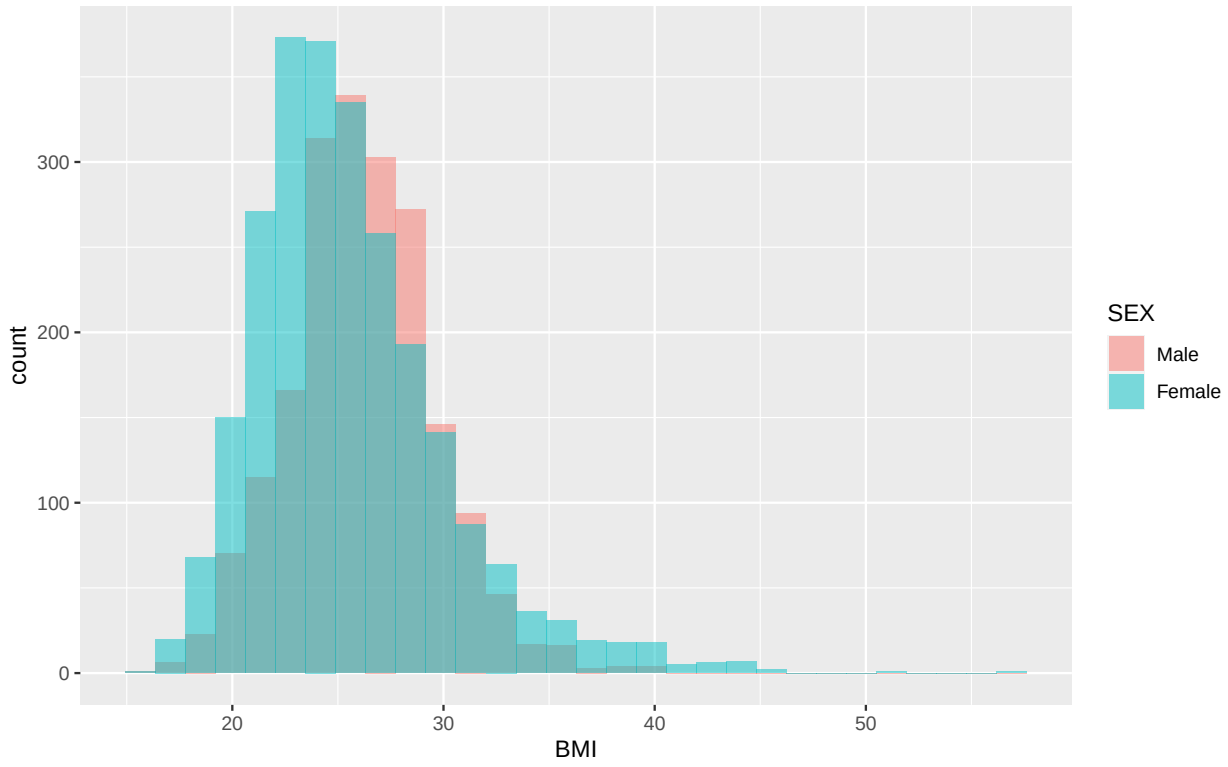
Histograms are very similar to boxplots, except that they are done using the `geom_histogram` command.

```
dat %>%
ggplot(aes(x=TOTCHOL))+
  geom_histogram(aes(group=SEX,fill=SEX))
```




The default behaviour is stacked histograms, which I don't find useful - we can do overlapping histograms as well.

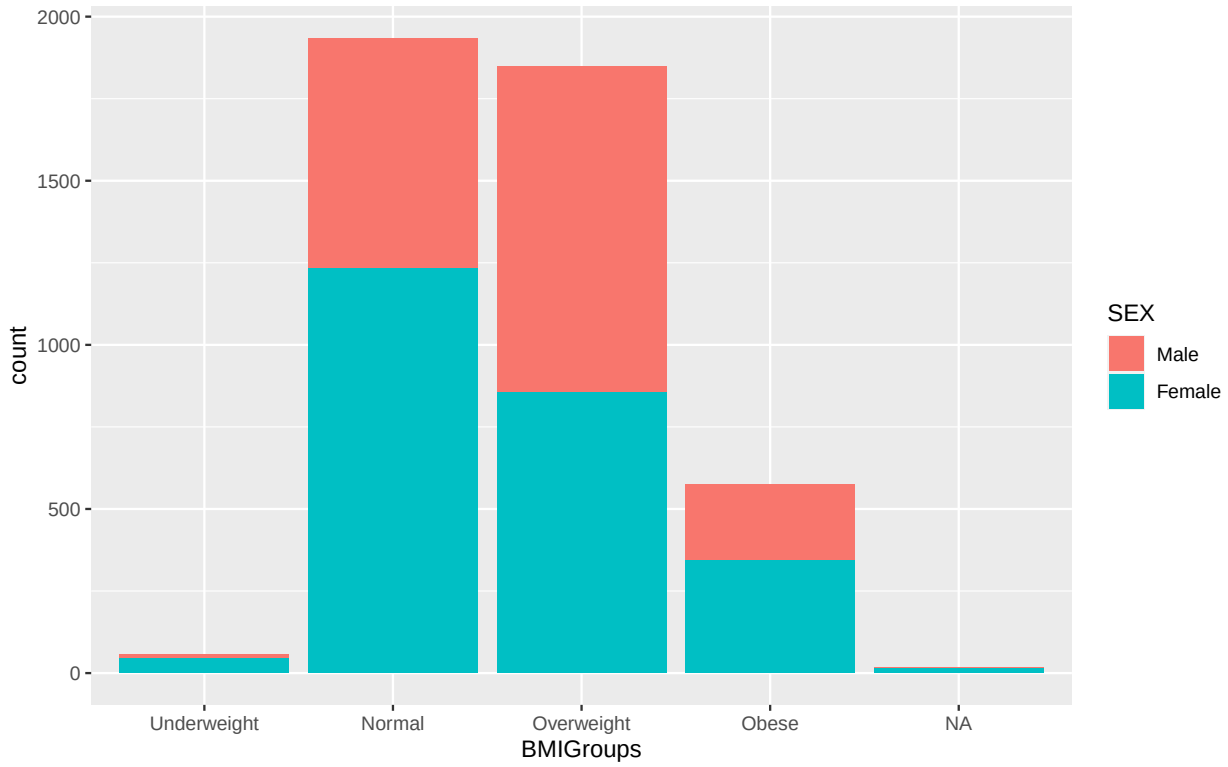
```
dat %>%
ggplot(aes(x=BMI,group=SEX,fill=SEX))+
  geom_histogram(position='identity',alpha=0.5)
```



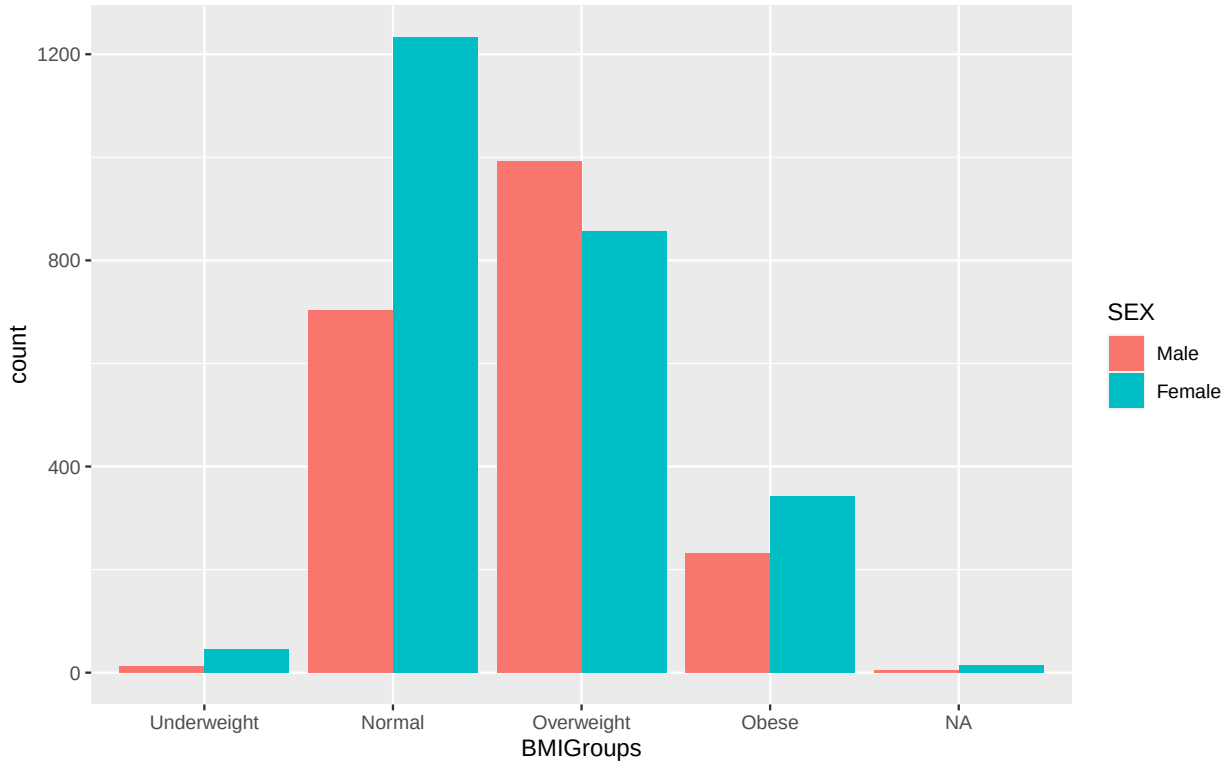

When doing overlapping histograms the `alpha` value is useful - this controls the alpha-channel, or the translucency, of a color, on a [0,1] scale, with 0 being completely translucent. As with boxplot STHDA has a useful blog post outlining all the options for histograms.

For barplots the `geom_bar` command will create plots for categorical data

```
dat %>%
  ggplot(aes(x=BMIGroups,group=SEX,fill=SEX))+
  geom_bar()
```




```
dat %>%
  ggplot(aes(x=BMIGroups,group=SEX,fill=SEX))+
  geom_bar(position='dodge')
```



And not surprisingly, STHDA has another blog post that provides a more complete breakdown of the functionality of the command.

# 5 Breakout Activity

As with last week, we're going to re-visit old commands and address them with the tidyverse - in this case we're going back to week 3 - try to create the following plots

1. A scatterplot of ~~systolic vs diastolic~~ Total Cholesterol (TOTCHOL) vs BMI
2. A histogram of systolic blood pressure
3. A ~~box~~ violin plot of systolic blood pressure stratified by SEX
4. A scatterplot of systolic vs diastolic for people with normal BMI *(hint: this is solved with a `dplyr` command, not a `ggplot` command)*
5. Separate systolic vs diastolic scatter plots by BMI Category *(We didn't learn how to do this, look up `facet_wrap` or `facet_grid`)*