Graphics in R

Kamarul Imran Musa

2018-06-12

Visual exploration

Introduction to visualization

Data visualization or data visualisation is viewed by many disciplines as a modern equivalent of visual communication. It involves the creation and study of the visual representation of data.

This means "information that has been abstracted in some schematic form, including attributes or variables for the units of information".

For complete references, read these sources:

1. https://en.m.wikipedia.org/wiki/Data_visualization
2. https://en.m.wikipedia.org/wiki/Michael_Friendly

History of data visualization

In his 1983 book *The Visual Display of Quantitative Information* (**ref needed**), the author Edward Tufte defines *graphical displays* and the principles for effective graphical displays. The book defines "excellence in statistical graphics consists of complex ideas communicated with clarity, precision and efficiency".

Processes and objectives of visualization

Visualization is the process of representing data graphically and interacting with these representations. The main objective is to gain insight into the data (http://researcher.watson.ibm.com/researcher/view_group.php?id=143)

What makes good graphics

You may require these to make good graphics:

1. Data
2. Substance rather than about method, graphic design, technology of graphic production or something else
3. No distortion to what the data has to say
4. Presence of many numbers in a small space
5. Coherence for large data sets
6. Encourage the eye to compare different pieces of data
7. Reveal the data at several levels of detail, from a broad overview to the fine structure
8. Serve a reasonably clear purpose: description, exploration, tabulation or decoration
9. Be closely integrated with the statistical and verbal descriptions of a data set.

Graphics packages in R

There are a number of graphics packages in R. A few of the packages are aimed to perform tasks related with graphs. Some provide graphics for certain analyses.

The popular general graphics packages in R include:

1. `graphics`
2. `ggplot2`
3. `lattice`

Some examples of other more specific packages aimed to run graphics for certain analyses include:

1. `survminer::ggsurvlot` to plot survival probability
2. `sjPlot` to plot mixed models results

Preliminaries

We will be using a dataset named `cholest.dta` which is in Stata format.

```
library(foreign)
cholest <- read.dta("cholest.dta")
head(cholest)
```

```
##   chol age exercise  sex categ
## 1  6.5  38        6 male Grp A
## 2  6.6  35        5 male Grp A
## 3  6.8  39        6 male Grp A
## 4  6.8  36        5 male Grp A
## 5  6.9  31        4 male Grp A
## 6  7.0  38        4 male Grp A
```

The data can be summarized as:

```
summary(cholest)
```

```
##       chol            age           exercise         sex       categ
##  Min.   : 6.50   Min.   :28.00   Min.   :2.000   female:40   Grp A:25
##  1st Qu.: 7.60   1st Qu.:36.00   1st Qu.:4.000   male  :40   Grp B:33
##  Median : 8.30   Median :39.00   Median :4.000               Grp C:22
##  Mean   : 8.23   Mean   :39.48   Mean   :4.225
##  3rd Qu.: 8.80   3rd Qu.:43.25   3rd Qu.:5.000
##  Max.   :10.00   Max.   :52.00   Max.   :6.000
```

From variable `sex`, we will create a variable named `male` and label female for 0 and 1 for male)

```
cholest$male <- factor(cholest$sex, labels = c('female', 'male'))
```

Questions to ask before plotting graphs

You must ask yourselves these questions:

1. Which variable or variables do I want to plot?
2. What is (or are) the type of that variable?

- Are they factor (categorical) variables ?
- Are they numerical variables?

3. Am I going to plot

- a single variable?
- two variables together?
- three variables together?

Using the `graphics` package

One variable: Plotting a categorical variable

For categorical variable, we can plot a `barchart` to display the frequencies of the data.

Create a frequency table and name as `count`:

```
counts <- table(cholest$male)
counts
```

```
##
## female   male
##     40     40
```

Now, plot the frequencies for the `counts` object created above

```
barplot(counts, main="Male and Female Distribution",
    xlab = "Sex", ylab = "Freq")
```

One variable: Plotting a numerical variable

A common graphic for numerical variable is a histogram.

Histogram

We create histograms with the function `hist(x)`. In the function,

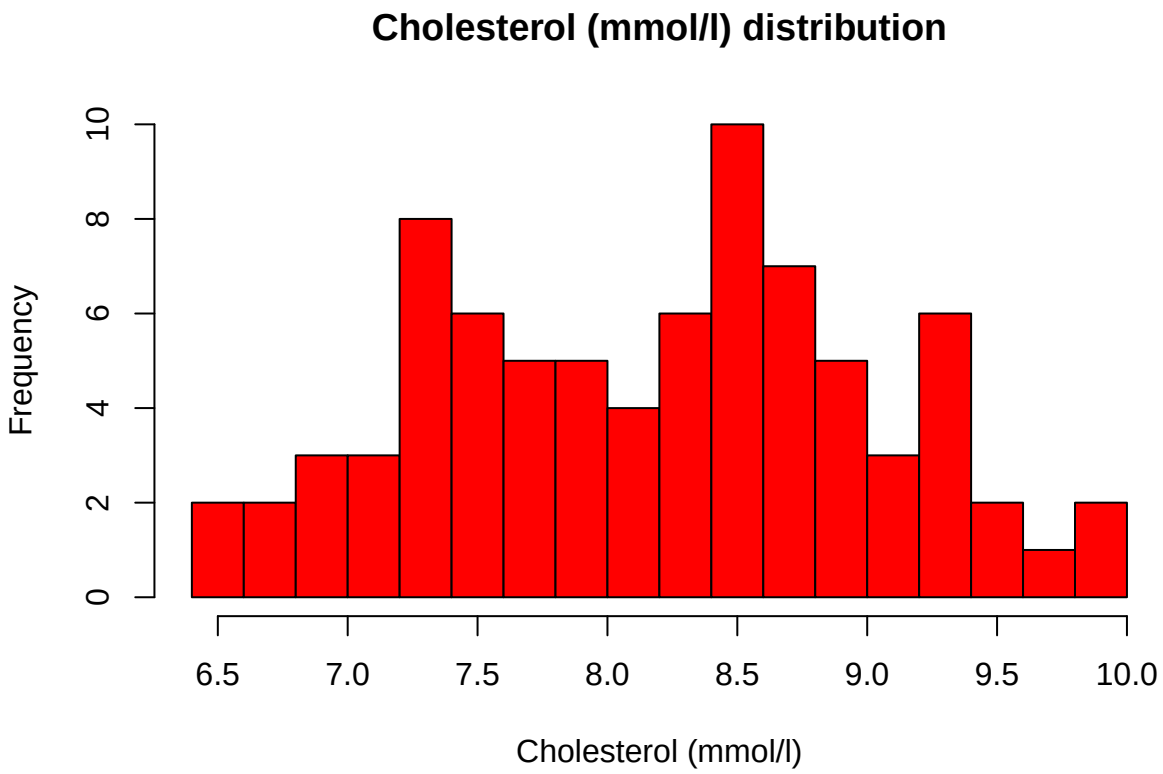
1. the argument `x` is a numeric vector of values to be plotted
2. the argument option `freq=FALSE` plots probability densities instead of frequencies.
3. the argument option `breaks =` controls the number of bins.

```
hist(cholest$chol)
```

Now, inside the `hist()` function, we will

1. set the `col =` argument to `red`
2. set the argument for bin to 12 bins `breaks = 14`
3. label the x-axis as "Cholesterol (mmol/l)"
4. the title is set in `main = 'title of plot'`

```
hist(cholest$chol, breaks = 14, col = "red",
     main = "Cholesterol (mmol/l) distribution", xlab = "Cholesterol (mmol/l)")
```



Kernel density plot

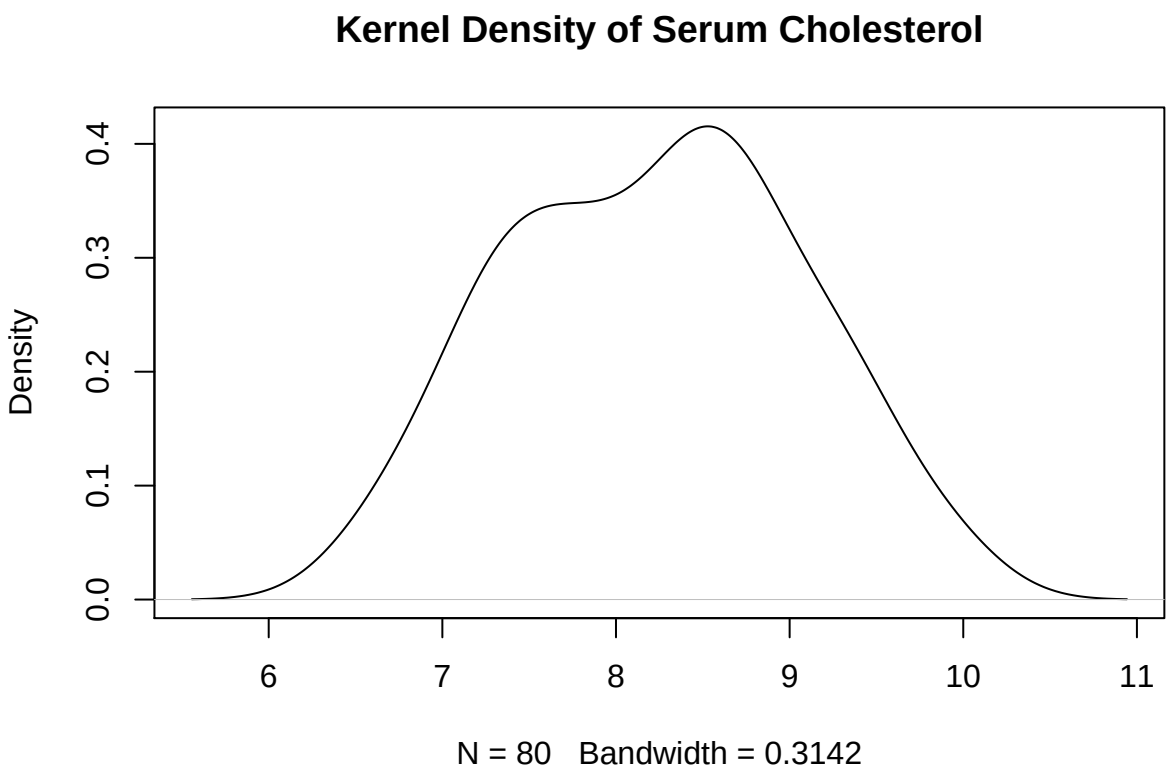
Kernal density plots are usually a much more effective way to view the distribution of a variable.

This can be done using `plot(density(x))`. In the function, the argument for x is a numeric vector.

Below, we

1. create a density plot and named it as `d.plot`. We do not consider missing value by setting the `na.rm = TRUE`
2. next, we plot `d.plot`

```
d.plot <- density(cholest$chol, na.rm = TRUE) # returns the density data
plot(d.plot, main = "Kernel Density of Serum Cholesterol") # plots the results
```



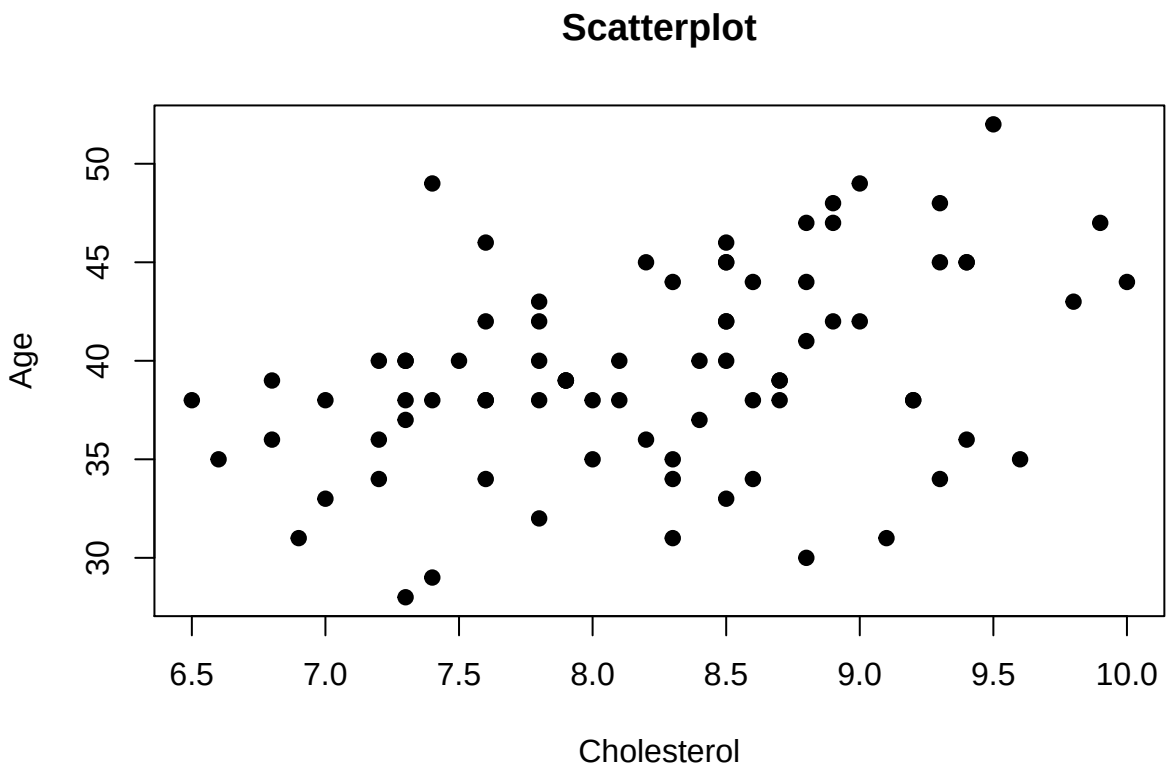
Two variables: Plotting scatter plot

We can plot two numerical variables simultenously. From such plot, we can see the association or relationship between the two variables.

Scatter plot

Scatter plot is one of the most common plots to display the association between 2 numerical variables.

```
plot(cholest$chol, cholest$age, main = "Scatterplot",
    xlab = "Cholesterol", ylab = "Age", pch=19)
```



You can always personalize the graphical parameters such as parameters for *fonts, colours, lines* and *symbols*. You can find the details in the `graphics` documentation. In addition, this website summarizes the parameters in a very nice way: http://www.statmethods.net/advgraphs/parameters.html

Using the `ggplot2` package

The official website for ggplot2 is here http://ggplot2.org/. In their own words, the package is described as

ggplot2 is a plotting system for R, based on the grammar of graphics, which tries to take the good parts of base and lattice graphics and none of the bad parts. It takes care of many of the fiddly details that make plotting a hassle (like drawing legends) as well as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics.

One variable: Plotting a numerical variable

Plot distribution of values of a numerical variable.

Histogram

Load the `ggplot2` package,

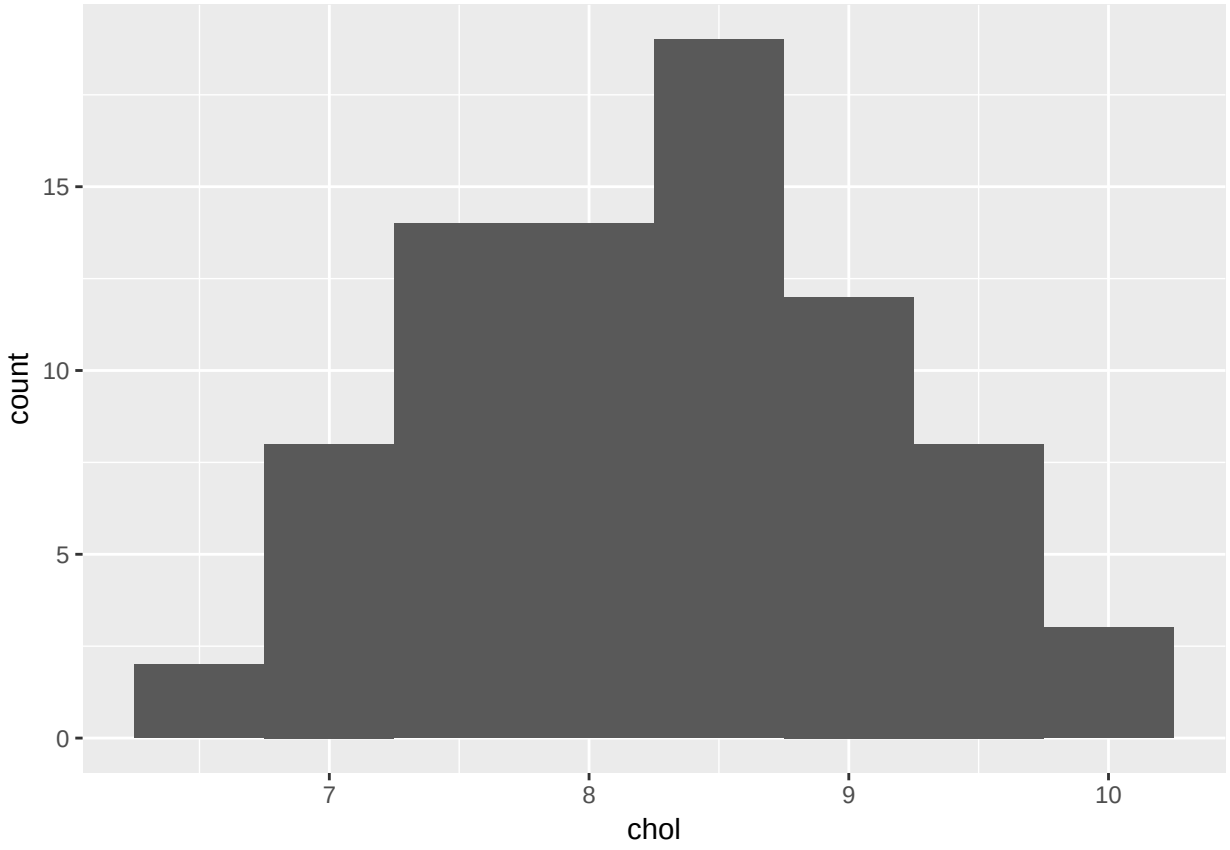
```
library(ggplot2)
```

In `ggplot2`,

1. type `ggplot(data = X)` function to choose the dataset .

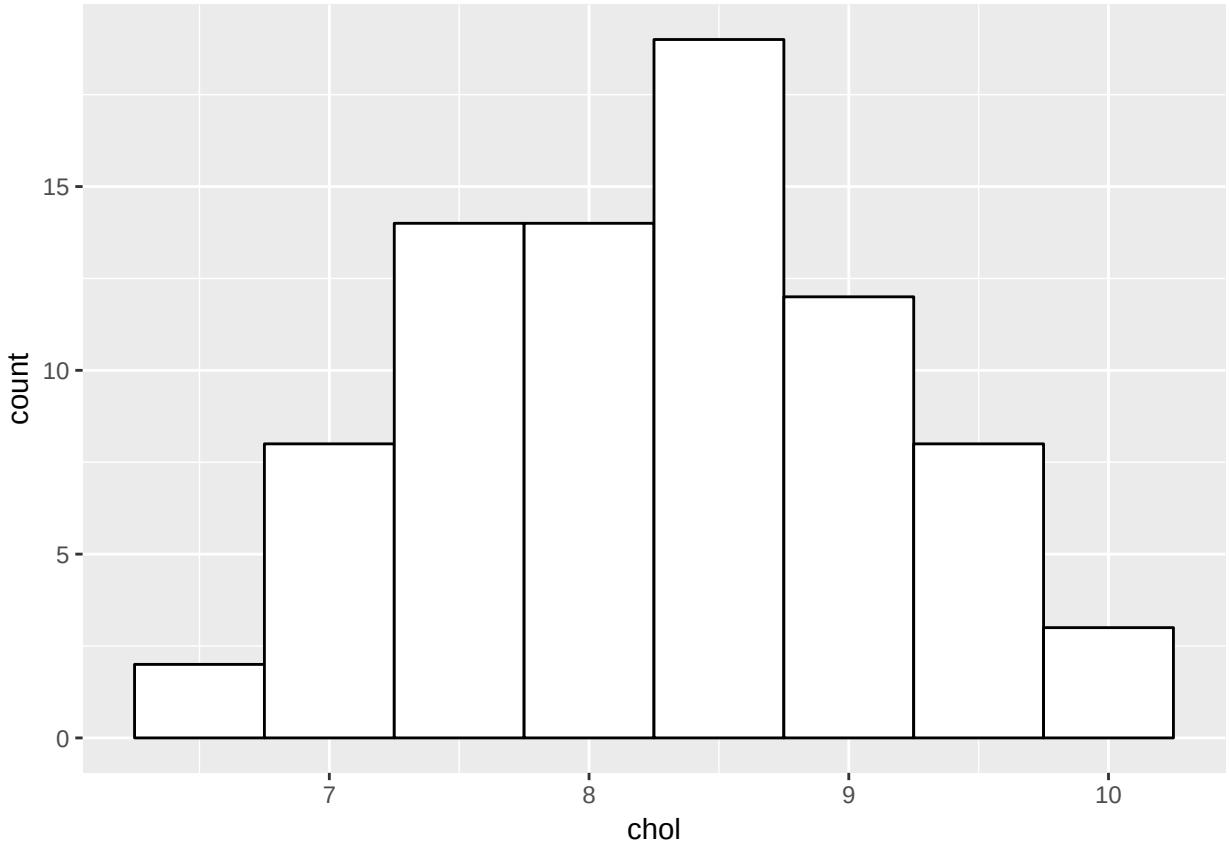
2. the `aes()` for variable or variables to be plotted.
3. then we use `geom_X` to specify the geometric (X) form of the plot.

```
myplot <- ggplot(data = cholest, aes(x = chol))
myplot + geom_histogram(binwidth = 0.5)
```



`ggplot2` has lots of flexibility and personalization. For example, we can set the line color and fill color, the theme, the size, the symbols etc.

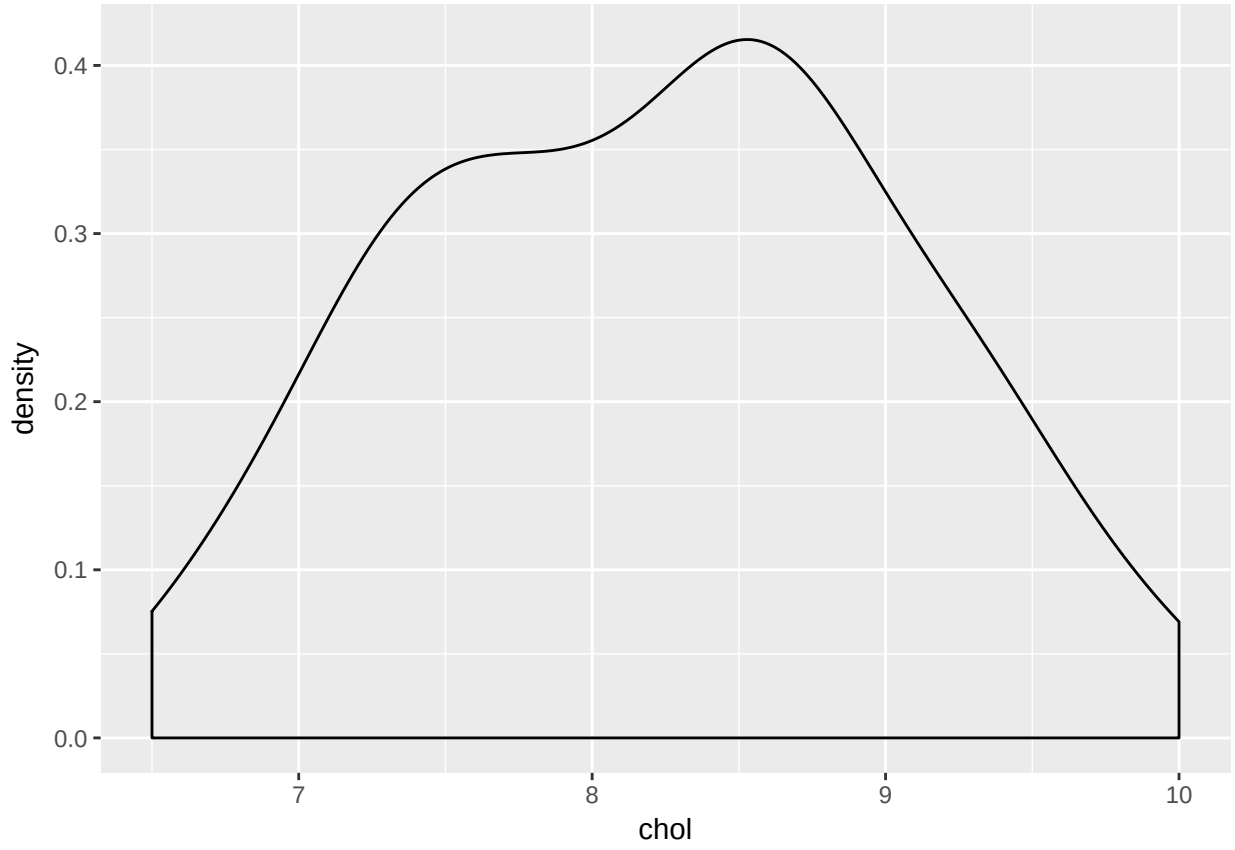
```
ggplot(cholest, aes(x = chol)) + geom_histogram(binwidth = 0.5,
                                                colour = "black", fill = "white")
```



Density curve

Density is useful to examine the distribution of observations.

```
ggplot(data = subset(cholest, !is.na(chol)), aes(x = chol)) + geom_density()
```



Combining the histogram and the density curve

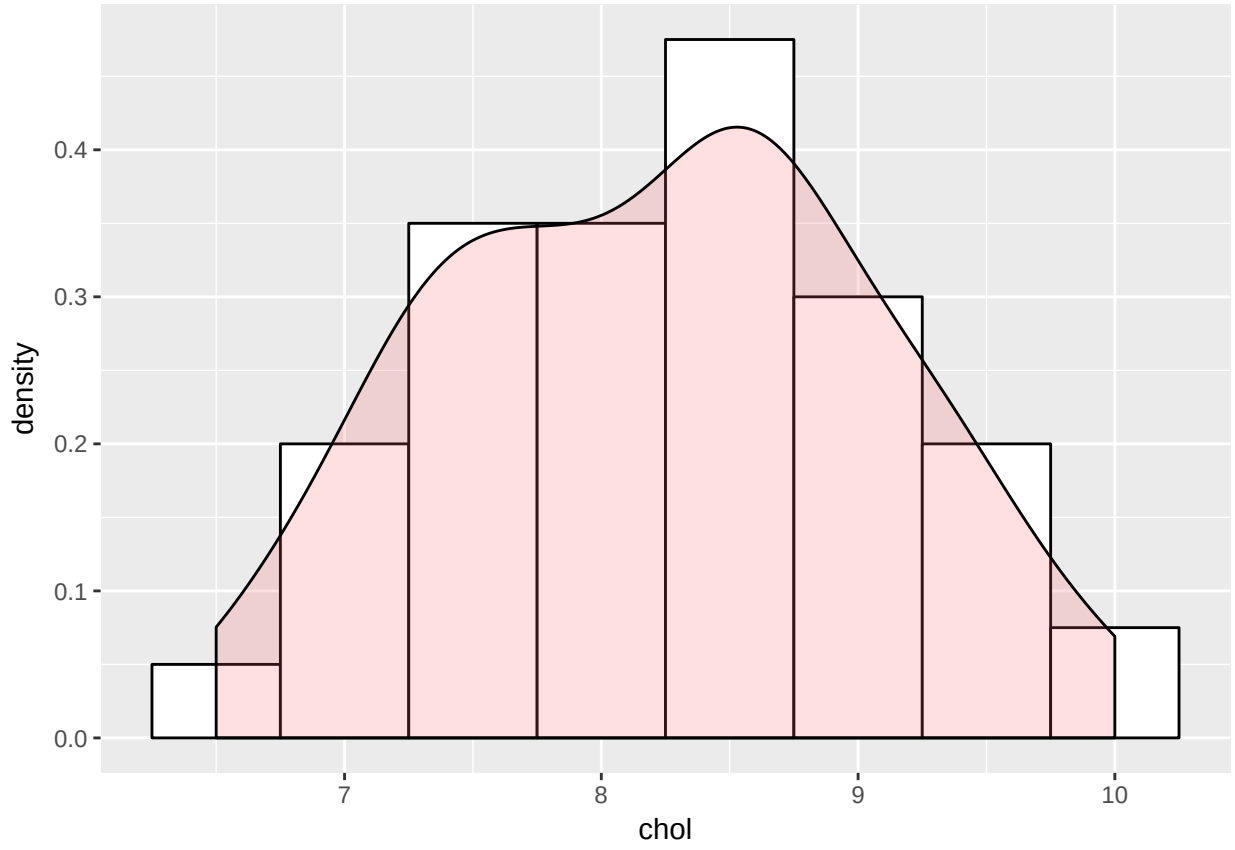
`ggplot2` allows plot to be displayed together. We can combine multiple plots in one single plot by overlaying multiple plots on one another.

Here, we will

1. create a histogram plot
2. create a density curve plot
3. overlay both (the density curve + the histogram).

To do this we need to specify a histogram with density instead of count on y-axis

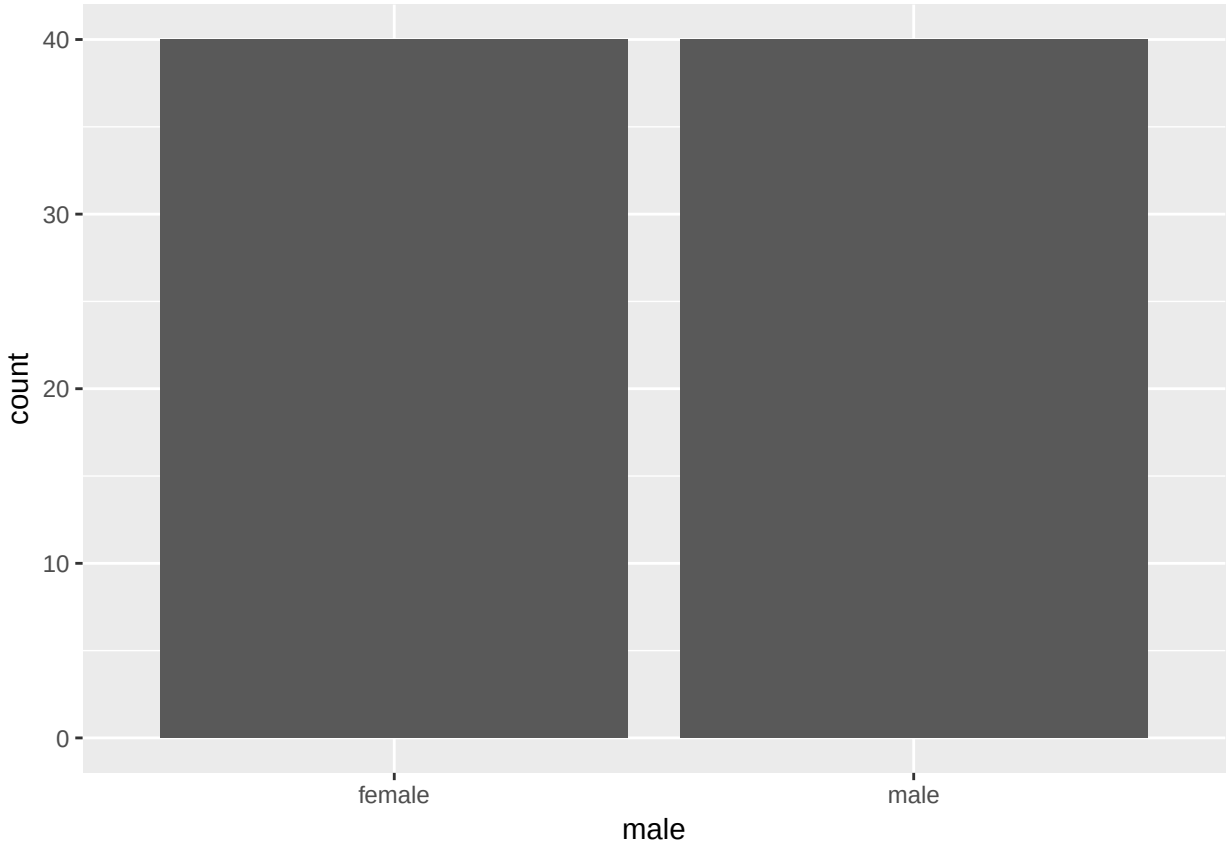
```
ggplot(data = subset(cholest, !is.na(chol)), aes(x = chol)) +
       geom_histogram(aes(y = ..density..), binwidth = 0.5,
                      colour = "black", fill = "white") +
    geom_density(alpha = .2, fill = "#FF6666")
```



One variable: Plotting a categorical variable

Now, let us create a basic barchart using `ggplot2::geom_bar()`

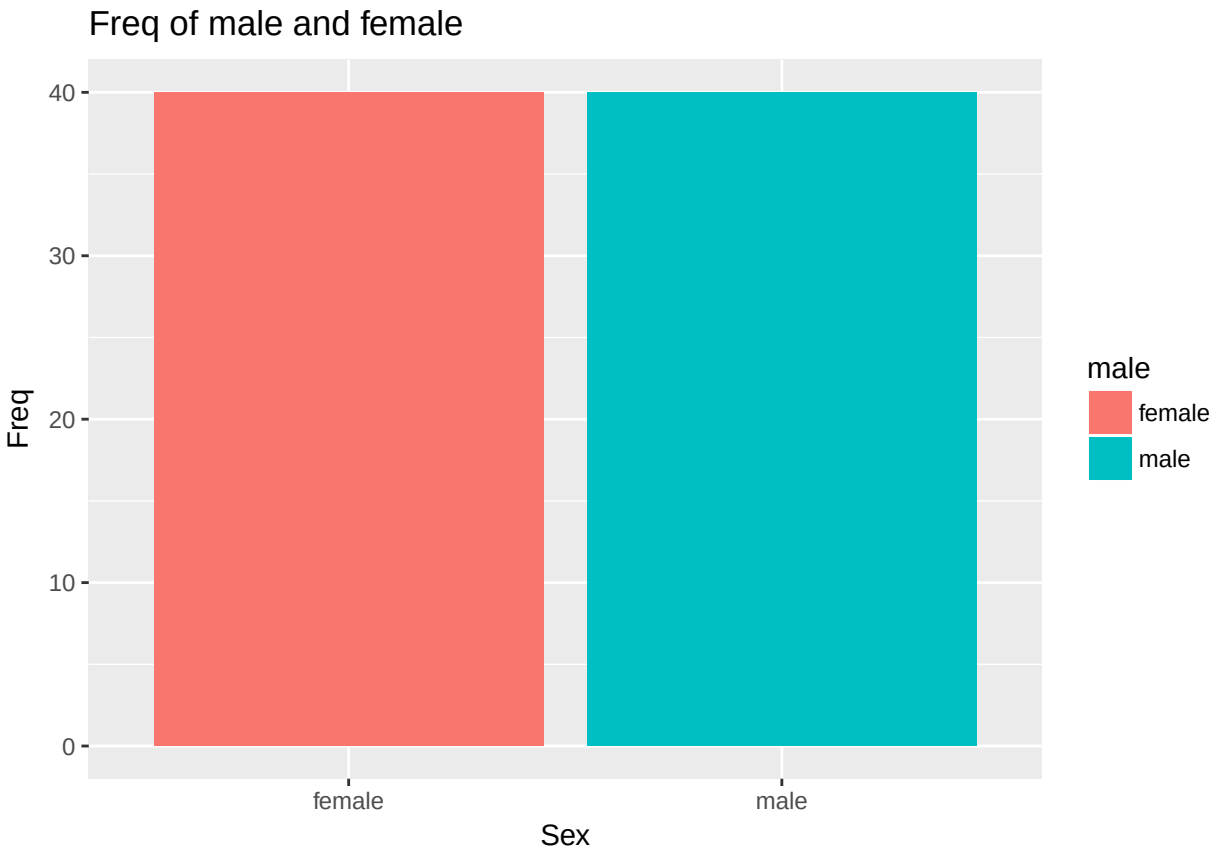
```
sex_bar <- ggplot(data = cholest, aes(male))
sex_bar + geom_bar()
```



The barchart looks OK, but we want to personalize it more - make it prettier and more presentable:

1. add labels to x and y axes `xlab()` and `ylab()`
2. add the title `ggtitle()`

```
ggplot(data = cholest, mapping = aes(male, fill = male)) +
  geom_bar() + xlab('Sex') + ylab('Freq') +
  ggtitle('Freq of male and female')
```



In addition, there is an excellent resource from this website on ggplot2: http://www.cookbook-r.com/Graphs/Bar_and_line_graphs_(ggplot2)/

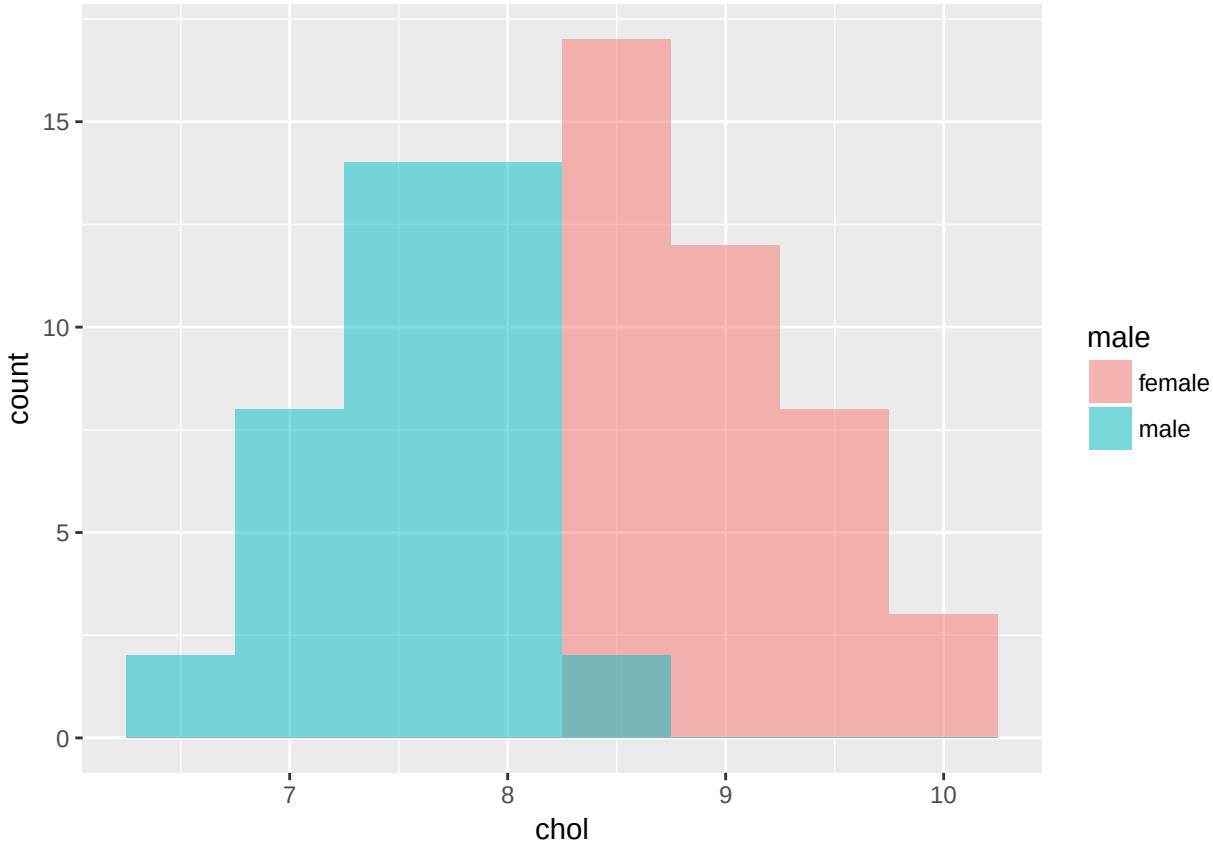
Two variables: Plotting a numerical and a categorical variable

Now, examine the distribution of a numerical variable (`rating`) in two groups (A and B) of the variable `cond` by

1. overlaying two histograms
2. interleaving two histograms
3. overlaying two density curve

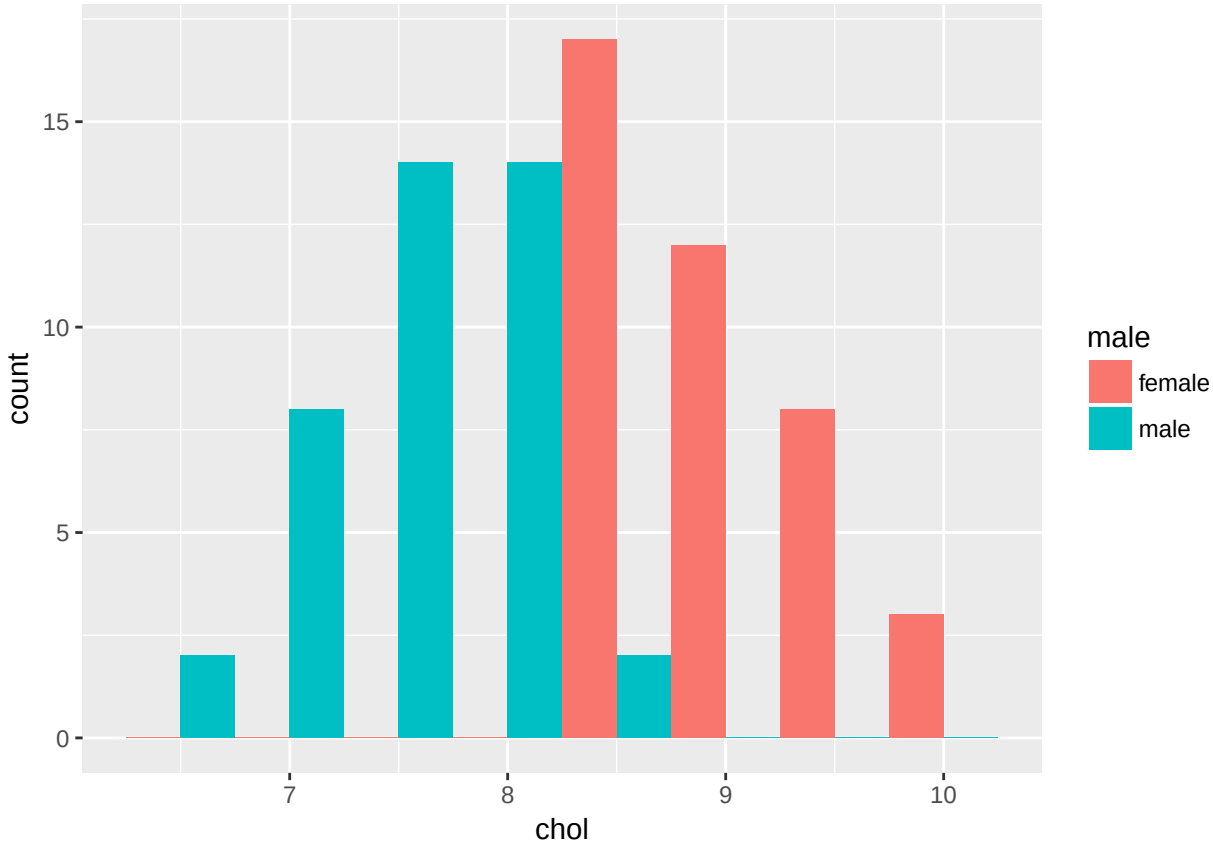
Overlaying histograms

```
ggplot(cholest, aes(x = chol, fill = male)) +
    geom_histogram(binwidth = .5, alpha = .5, position = "identity")
```



Interleaving histograms

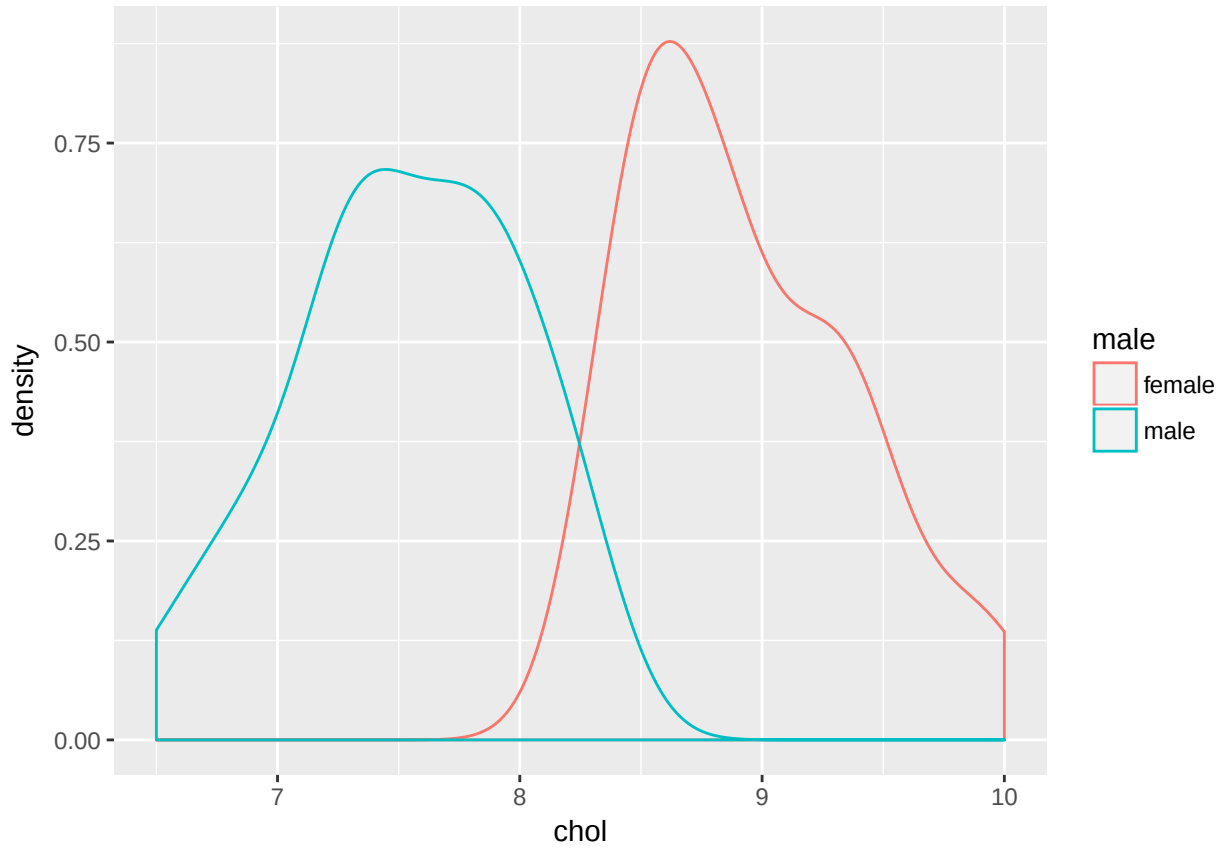
```
ggplot(cholest, aes(x = chol, fill = male)) +
    geom_histogram(binwidth = .5, position = "dodge")
```



Overlaying density plots

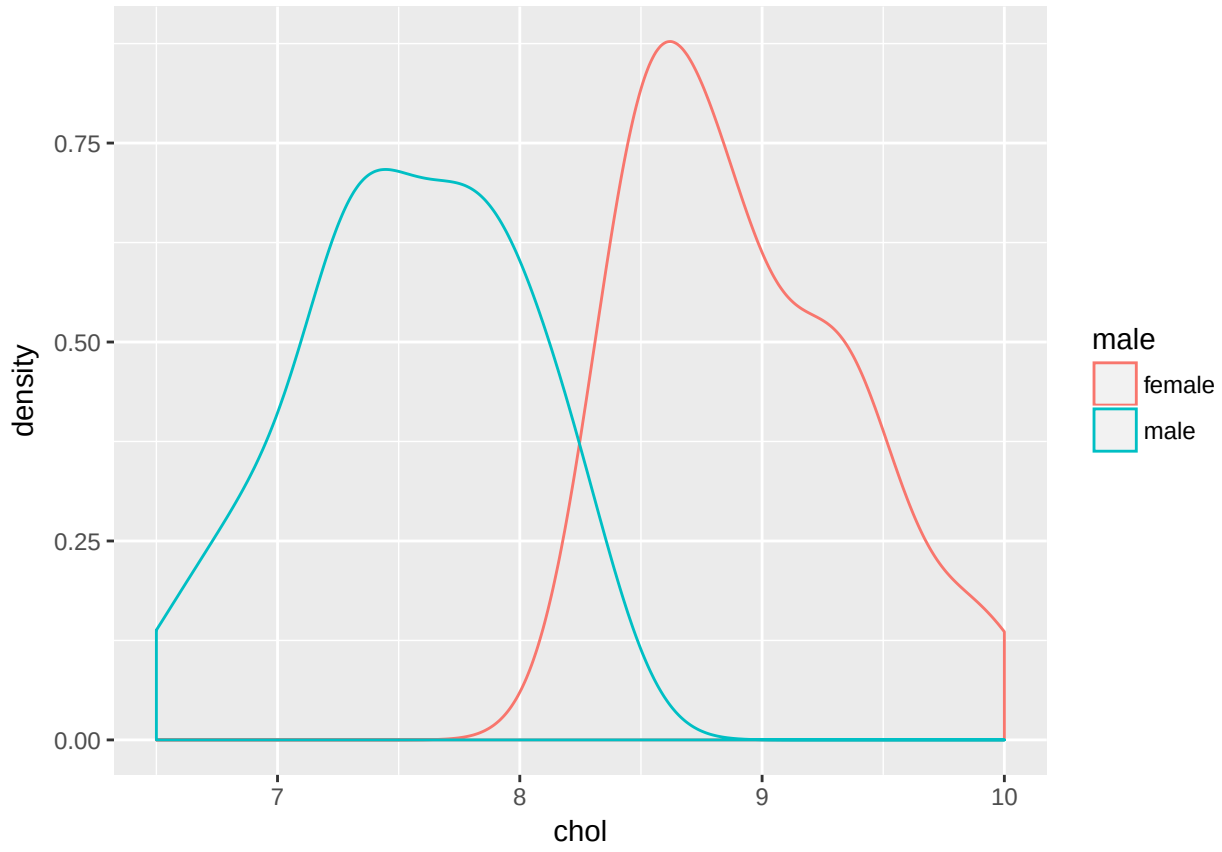
Full transparent

```
ggplot(cholest, aes(x = chol, colour = male)) + geom_density()
```



Now, try set the transparency at 30%

```
# Density plots with semi-transparent fill
ggplot(cholest, aes(x = chol, colour = male)) + geom_density(alpha = .3)
```



Using facets

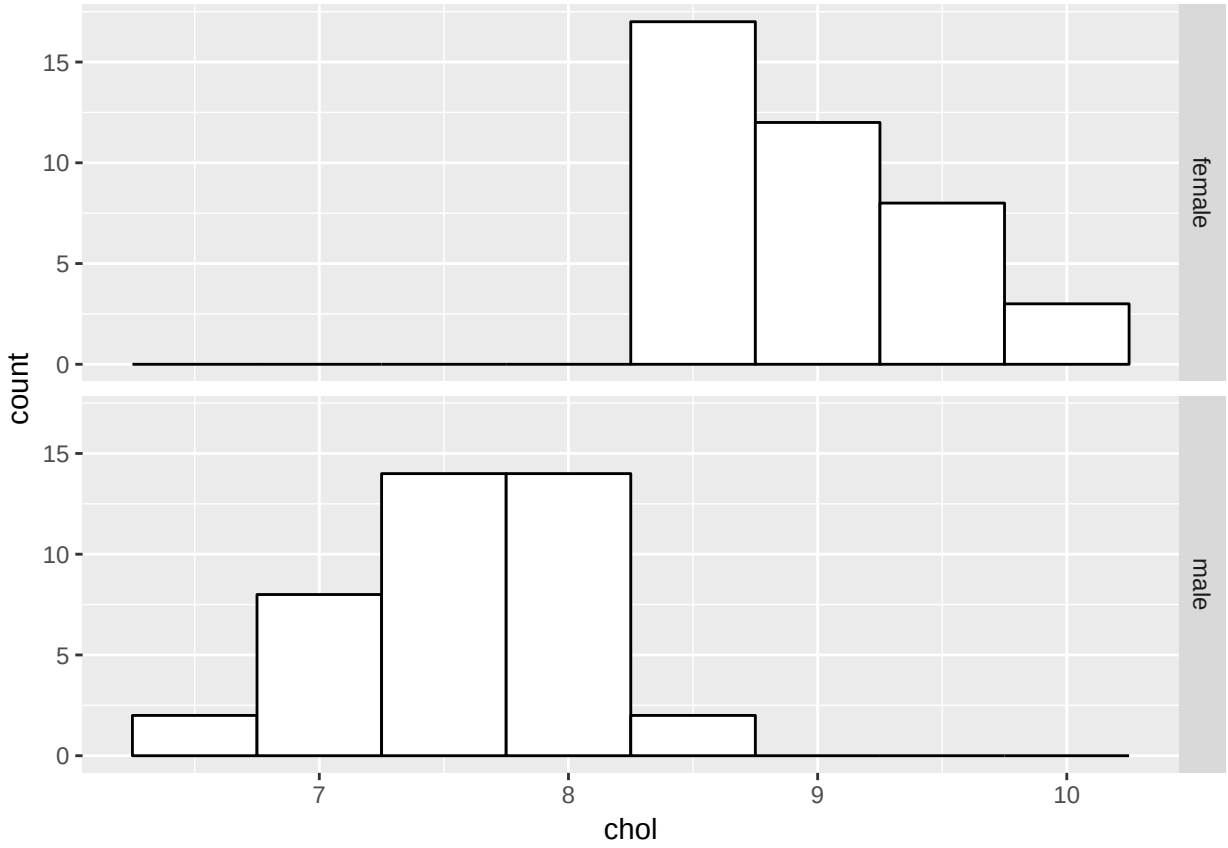
We use `facet_grid()` to split the plot.

There are two types of facetting the plot:

1. Vertical facet
2. Horizontal facet

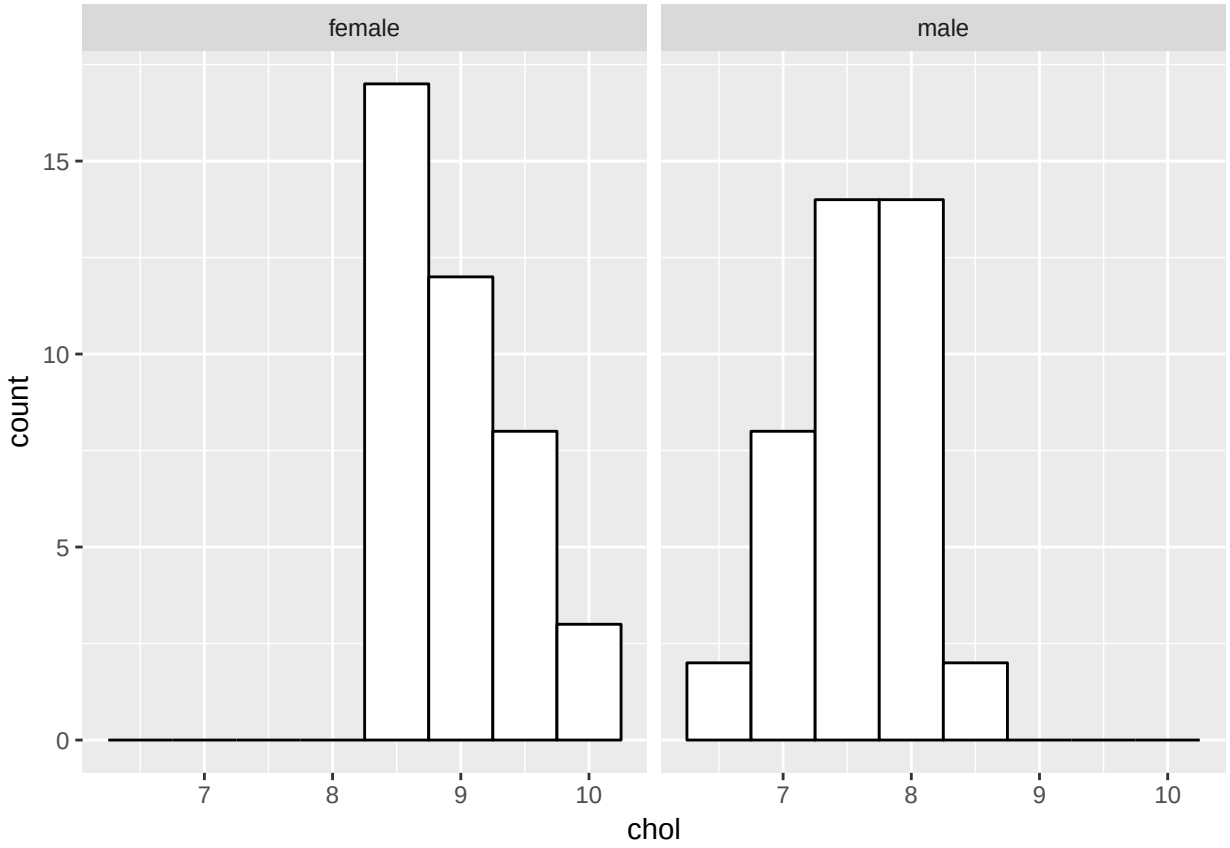
The vertical facets

```
ggplot(data = subset(cholest, !is.na(chol)), aes(x = chol)) + geom_histogram(binwidth = .5,
                                                    colour = "black", fill = "white") +
  facet_grid(male ~ .)
```



The horizontal facets

```
ggplot(data = subset(cholest, !is.na(chol)), aes(x = chol)) + geom_histogram(binwidth = .5,
                                                    colour = "black", fill = "white") +
    facet_grid(. ~ male)
```



Saving plots in ggplot2

This will save the last plot as `.png` and `.jpg` formats,

```
ggsave("myhistogram.png", width = 5, height = 5)
ggsave("myhistogram.jpg", width = 5, height = 5)
```

Using the `lattice` package

`lattice` package can create beautiful plots too. A very useful vignette for `lattice` package can be found here http://lattice.r-forge.r-project.org/Vignettes/src/lattice-intro/lattice-intro.pdf

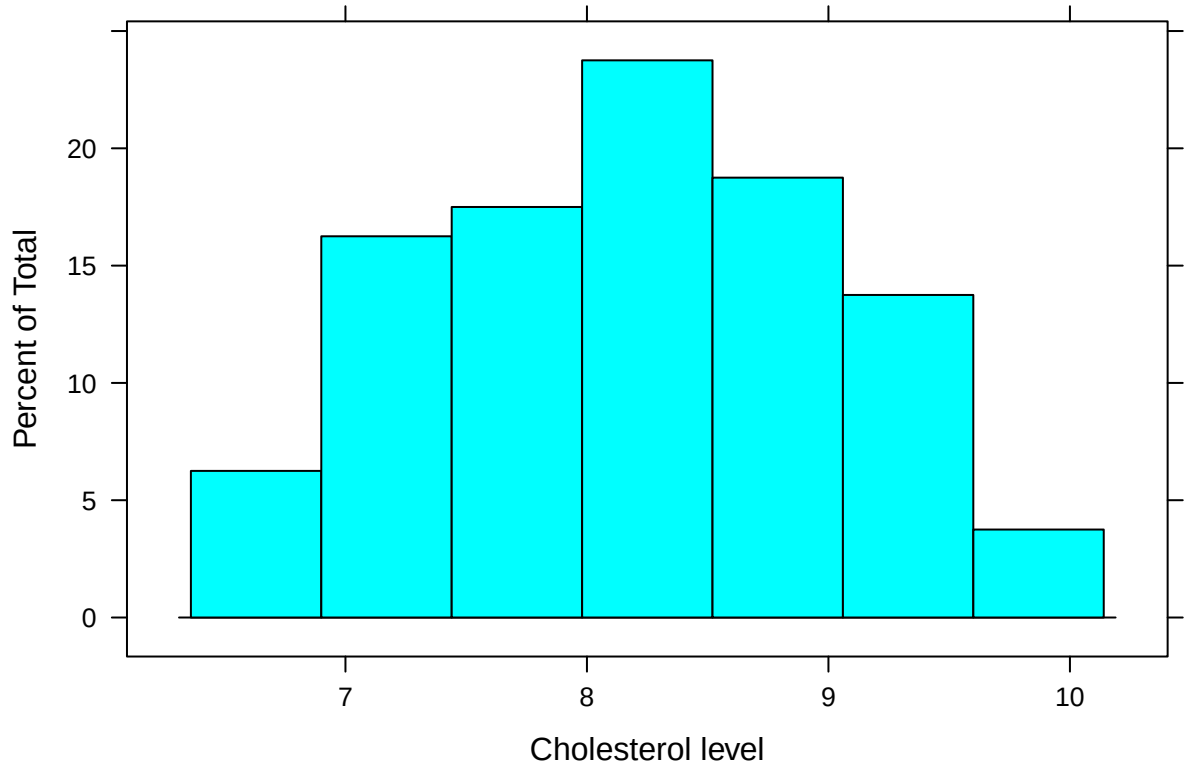
Loading the `lattice` package

```
library(lattice)
```

One numerical variable: Plotting a histogram

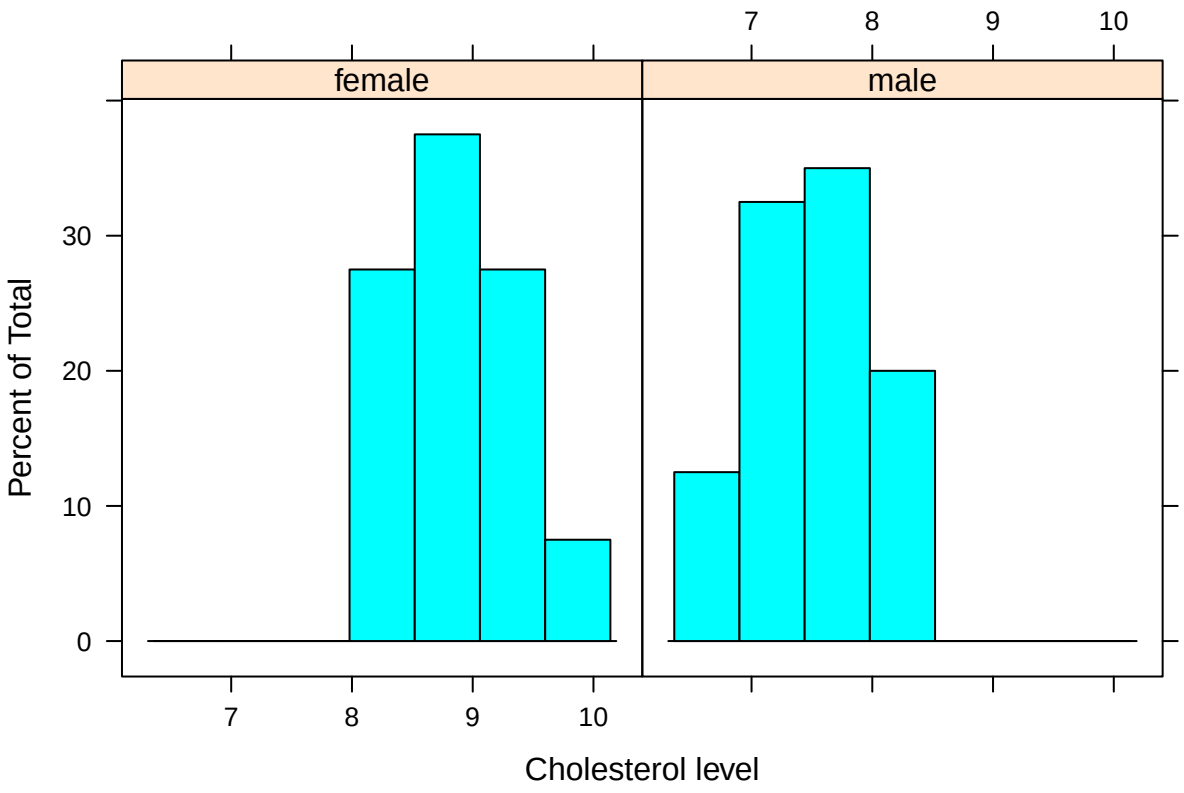
Plot a histogram for variable `chol` and label the x axis

```
histogram(~ chol, data = cholest, xlab = 'Cholesterol level')
```



One numerical and one categorical variable: Plotting histograms

```
histogram(~ chol | male, data = cholest,
          xlab = 'Cholesterol level' )
```

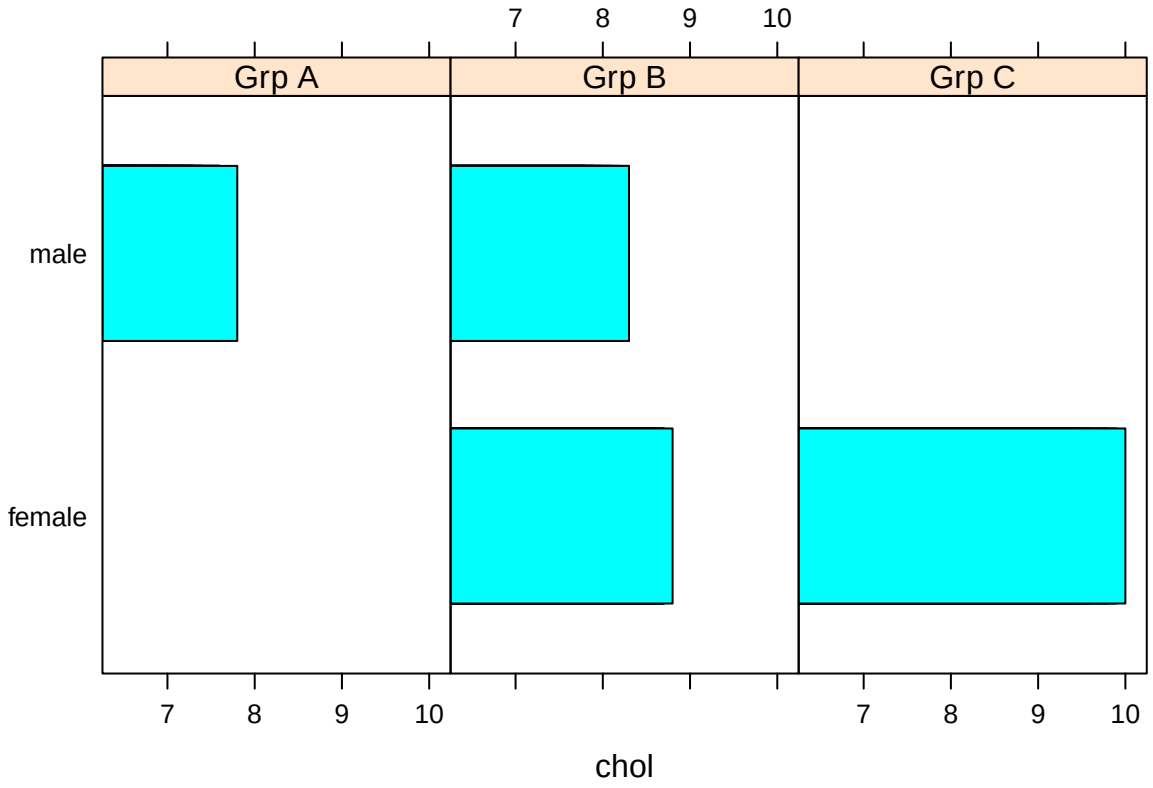


One categorical variable: Plotting a barchart

We use `barchart()` and set

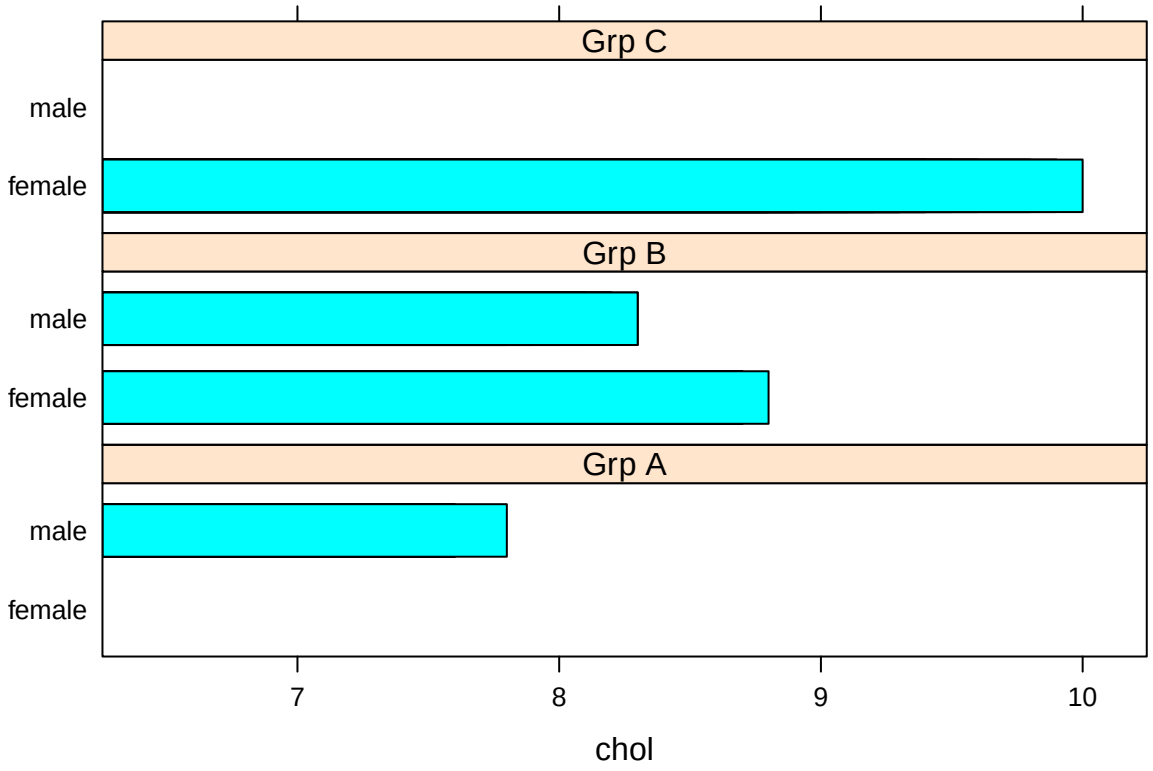
1. dependent variable = `male`
2. independent variable = `chol`
3. 4 bar charts in 4 columns and 1 row

```
barchart(male ~ chol | factor(categ), data = cholest, layout = c(3, 1))
```



Now we change the variables for the x and y axes and also the column and row arrangements - vertical plots, that is 1 column and 4 rows

```
barchart(male ~ chol | factor(categ), data = cholest, layout = c(1, 3))
```


Grp C
male
female
Grp B
male
female
Grp A
male
female
7
8
9
10
chol

Summary