Human Genome Analysis Lab 3 : Graphics with ggplot2

- Learning objectives
- Introduction to R Graphics
  - Grammar of Graphics
  - Installation
  - Learning materials for this lab
- Additional notes on the structure of the complete 23andMe file in R
- Additional Resources
- Exercises
  - Exercise 1
  - Exercise 2
  - Exercise 3
  - Exercise 4
  - Exercise 5
  - Exercise 6

Learning objectives

- Understand basic graphs in ggplot2
- Apply ggplot2 for graphing 23andMe SNP data

Introduction to R Graphics

R provides comprehensive graphics utilities for visualizing and exploring scientific data. To date we have been making a few plots using the R Base Graphics. In addition, several more recent graphics environments extend these utilities. These include the grid, lattice and ggplot2 packages. The ggplot2 envirnoment is by far the most popular and used for many R packages and in many scientific publications.

Grammar of Graphics

ggplot2 is meant to be an implementation of the Grammar of Graphics, hence the gg in ggplot. The basic notion is that there is a grammar to the composition of graphical components in statistical graphics. By direcly controlling that grammar, you can generate a large set of carefully constructed graphics from a relatively small set of operations. As Hadley Wickham (2010), the author of ggplot2 said,

"A good grammar will allow us to gain insight into the composition of complicated graphics, and reveal unexpected connections between seemingly different graphics.

Installation

ggplot2 is a R package, which is part of the tidyverse collections. As in the install of knitr in Lab 1 in the menu select TOOLS and then INSTALL PACKAGES. Install tidyverse (which includes ggplot2 as well as other packages we will use later).

```
install.packages("tidyverse")
```

You only need to install tidyverse once. However, each time you open RStudio and would like to use ggplot you need to load the library for it. You also need to add this line to your Rmd file when you make your report.

```
library(ggplot2)
```

Learning materials for this lab

You can make amazing graphs with ggplot, but there is a long learning curve so we will have multiple lab sessions on ggplot and graphing. Thus I am providing multiple resources that provide different perspectives on learning ggplot. Let me know which one's work best for you.

- Jeff's adaption of a tutoral by Josef Fruehwald, University of York. I have provided the hmtl and Rmd files for this tutorial on the Moodle site. To start I recommend knitting the Rmd file to make sure it works on your computer. Then go through the tutorial section by section writing or copying the code into a new .R file.

- Hadley Wickham and Garrett Grolemund released a new book in 2017 R for Data Science (http://r4ds.had.co.nz/). I recommend reading the short Chapters 1 and 2. We will use Chapter 3.1 to 3.5 as the basis of this week's lab in ggplot. It is helpful to go through the exercises in the Chapter as reinforncement for the reading and preparation for our class problem set.
- Maria Nattestad's Youtube videos (https://www.youtube.com/channel/UC2bWYX9h1KlaGWfTDuhASWg)

Additional notes on the structure of the complete 23andMe file in R

When we display the structure of the 23andMe file there are several important points to notice.

```
SNPs<- read.table("23andMe_complete.txt", header = TRUE, sep = "\t")
str(SNPs)
```

```
## 'data.frame':    960614 obs. of  4 variables:
##  $ rsid      : Factor w/ 960614 levels "i1000009","i2000003",..: 600125 532383 535928 178265 124446 723988 655510 60003
6 813048 254468 ...
##  $ chromosome: Factor w/ 25 levels "1","10","11",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ position  : int  82154 752566 752721 776546 798959 800007 838555 846808 854250 861808 ...
##  $ genotype  : Factor w/ 20 levels "--","A","AA",..: 3 3 15 5 5 8 4 10 5 15 ...
```

First that the object (SNPs) is a dataframe. IA dataframe is a list of vectors of the same length, but they don't have to be of the same type as in a matrix. In this instance the vectors are factors and integers.

To understand how data science works in R, it is necessary to get a solid background in the values that make up data and the structures that hold them. In the previous labs we have seen that objects are containers that can hold a single integer or character, vectors, matrices or dataframes. We loaded the data as rows and columns typical of spreadsheets we see in Excel and similar programs. R extracts these into collections of objects. These can be examine using the following commands.

```
class(SNPs)
```

```
## [1] "data.frame"
```

```
typeof(SNPs)
```

```
## [1] "list"
```

```
str(SNPs)
```

```
## 'data.frame':    960614 obs. of  4 variables:
##  $ rsid      : Factor w/ 960614 levels "i1000009","i2000003",..: 600125 532383 535928 178265 124446 723988 655510 60003
6 813048 254468 ...
##  $ chromosome: Factor w/ 25 levels "1","10","11",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ position  : int  82154 752566 752721 776546 798959 800007 838555 846808 854250 861808 ...
##  $ genotype  : Factor w/ 20 levels "--","A","AA",..: 3 3 15 5 5 8 4 10 5 15 ...
```

```
summary(SNPs)
```

```
##        rsid          chromosome        position            genotype
##  i1000009:     1   2      : 77346   Min.   :        3   CC     :173264
##  i2000003:     1   1      : 76909   1st Qu.: 30718234   GG     :173054
##  i3000001:     1   3      : 63285   Median : 67598882   TT     :148126
##  i3000002:     1   6      : 63245   Mean   : 77262458   AA     :147157
##  i3000003:     1   5      : 56019   3rd Qu.:113837894   AG     :109001
##  i3000004:     1   4      : 55017   Max.   :249218992   CT     :108992
##  (Other) :960608   (Other):568793                       (Other):101020
```

```
class(SNPs$genotype)
```

```
## [1] "factor"
```

```
typeof(SNPs$genotype)
```

```
## [1] "integer"
```

```
str(SNPs$genotype)
```

```
##  Factor w/ 20 levels "--","A","AA",..: 3 3 15 5 5 8 4 10 5 15 ...
```

```
summary(SNPs$genotype)
```

```
##     --      A     AA     AC     AG     AT      C     CC     CG     CT      D
##  21109   6676 147157  25036 109001    569   7188 173264   1003 108992     36
##     DD     DI      G     GG     GT      I     II      T     TT
##    157     17   7061 173054  24727    113    685   6643 148126
```

```
summary(SNPs$chromosome)
```

```
##     1    10    11    12    13    14    15    16    17    18    19     2    20
## 76909 50322 47972 47125 36078 30818 28400 30167 26688 27971 18533 77346 23834
##    21    22     3     4     5     6     7     8     9    MT     X     Y
## 13404 14100 63285 55017 56019 63245 50965 49215 42969  2459 26007  1766
```

```
summary(SNPs$position)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##         3  30718234  67598882  77262458 113837894 249218992
```

Notice the difference in providing a summary of position which is an integer vs the other objects which are factors. Also notice that the chromosomes are not in numerical order since they are factors (characters) instead of numbers. To have the chromosomes order by number we can change the object type of chromosome from a factor to an ordered facto.

```
summary(SNPs$chromosome)
```

```
##     1    10    11    12    13    14    15    16    17    18    19     2    20
## 76909 50322 47972 47125 36078 30818 28400 30167 26688 27971 18533 77346 23834
##    21    22     3     4     5     6     7     8     9    MT     X     Y
## 13404 14100 63285 55017 56019 63245 50965 49215 42969  2459 26007  1766
```

```
SNPs$chromosome = ordered(SNPs$chromosome, levels=c(seq(1, 22), "X", "Y", "MT"))
summary(SNPs$chromosome)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## 76909 77346 63285 55017 56019 63245 50965 49215 42969 50322 47972 47125 36078
##    14    15    16    17    18    19    20    21    22     X     Y    MT
## 30818 28400 30167 26688 27971 18533 23834 13404 14100 26007  1766  2459
```

Additional Resources

We have only scratched the surface here. To learn more, see the ggplot reference site (http://docs.ggplot2.org/), and Winston Chang's excellent Cookbook for R (http://wiki.stdout.org/rcookbook/Graphs/) site. Though slightly out of date, ggplot2: Elegant Graphics for Data Anaysis (http://www.amazon.com/ggplot2-Elegant-Graphics-Data-Analysis/dp/0387981403) is still the definative book on this subject. Here is a nice comparison of basic and ggplot graphs - http://www.fdawg.org/FDAWG/Tutorials/ggplot2.html (http://www.fdawg.org/FDAWG/Tutorials/ggplot2.html). Here is another good tutorial - https://rpubs.com/mccannecology/53464 (https://rpubs.com/mccannecology/53464)

Exercises

Please do the exercises in Chapter 3 before trying the below exercises. Don't forget to load the tidyverse package and the 23andMe file in when you create your report in R_Markdown

```
library(tidyverse)
SNPs<- read.table("23andMe_complete.txt", header = TRUE, sep = "\t")
```

For now don't worry about titles, x and y labels, small/big text or the aspect ratios. We will work on those next week.

Exercise 1

Using ggplot make a make a bar graph of the total SNP counts for each chromosome.

Exercise 2

Order the chromosomes according to number by converting chromosomes from a factor to a order factor as in the example above. Then replot the bar graph

Exercise 3

Show the contribution of each genotype to the chromosome count using a stacked bar graph (with the fill = genotype)

Exercise 4

Make each set of stacked bars the same height to easier to compare proportions across groups.

Exercise 5

Now place genotypes directly beside one another for each chromosome to compare individual values.

Exercise 6

The above graph is pretty hard to read. Try using facet_wrap with the genotype