Introduction to R and RStudio Notes

RStudio Overview

R is the name of the programming language itself and RStudio is a convenient interface.

Your RStudio window has four panels:

1. **R Markdown Panel** (this document) in the upper left panel. Is the “master file” of notes, commands, and outputs. Different texts can be formatted (titles, bold, italicized, etc) and be generated into a html file.
2. **Console Panel** in the bottom left has the same text at the top of the console telling you the version of R that you’re running. Below that information is the *prompt* or request for command.
3. **Workspace Panel** in the upper right has a history of commands previously entered and datasets that are being used.
4. **Lower right panel** shows the plots you generate and also where you can browse your files, access help, manage packages, etc.

R Packages

R is an open-source programming language, meaning that users can contribute packages that make our lives easier, and we can use them for free. For this lab, and many others in the future, we will use the following R packages:

- `statsr` : for data files and functions used in this course
- `dplyr` : for data wrangling
- `ggplot2` : for data visualization

First install these packages:

- install.packages(“dplyr”)
- install.packages(“ggplot2”)
- install.packages(“shiny”)
- install_github(“StatsWithR/statsr”)

Next, you need to load the packages in your working environment with the `library` function. Note that you only need to **install** packages once, but you need to **load** them each time you relaunch RStudio.

```
library(dplyr)
library(ggplot2)
library(statsr)
```

Three ways to run the commands:

1. Highlight the codes to run -> Run -> Run current Chunk
2. Type the code into the console
3. Run to run the entire rmd file

Simple Data Commands

About the Data: The data set refers to Dr. John Arbuthnot, an 18th century physician, writer, and mathematician. He was interested in the ratio of newborn boys to newborn girls, so he gathered the baptism records for children born in London for every year from 1629 to 1710.

Load Data Sets

```
data(arbuthnot)
```

Data is successfully loaded when it appears in the Workspace Panel as ‘arbuthnot’ with 82 observations and 3 variables.

View Data

```
arbuthnot
```

```
## # A tibble: 82 × 3
##     year  boys girls
##    <int> <int> <int>
##  1  1629  5218  4683
##  2  1630  4858  4457
##  3  1631  4422  4102
##  4  1632  4994  4590
##  5  1633  5158  4839
##  6  1634  5035  4820
##  7  1635  5106  4928
##  8  1636  4917  4605
##  9  1637  4703  4457
## 10  1638  5359  4952
## # ℹ 72 more rows
```

View Dimensions of a DataFrame

```
dim(arbuthnot)
```

```
## [1] 82  3
```

Output indicates that there are 82 rows and 3 columns

View Names of Columns of Dataset

```
names(arbuthnot)
```

```
## [1] "year"  "boys"  "girls"
```

View Data in a Column

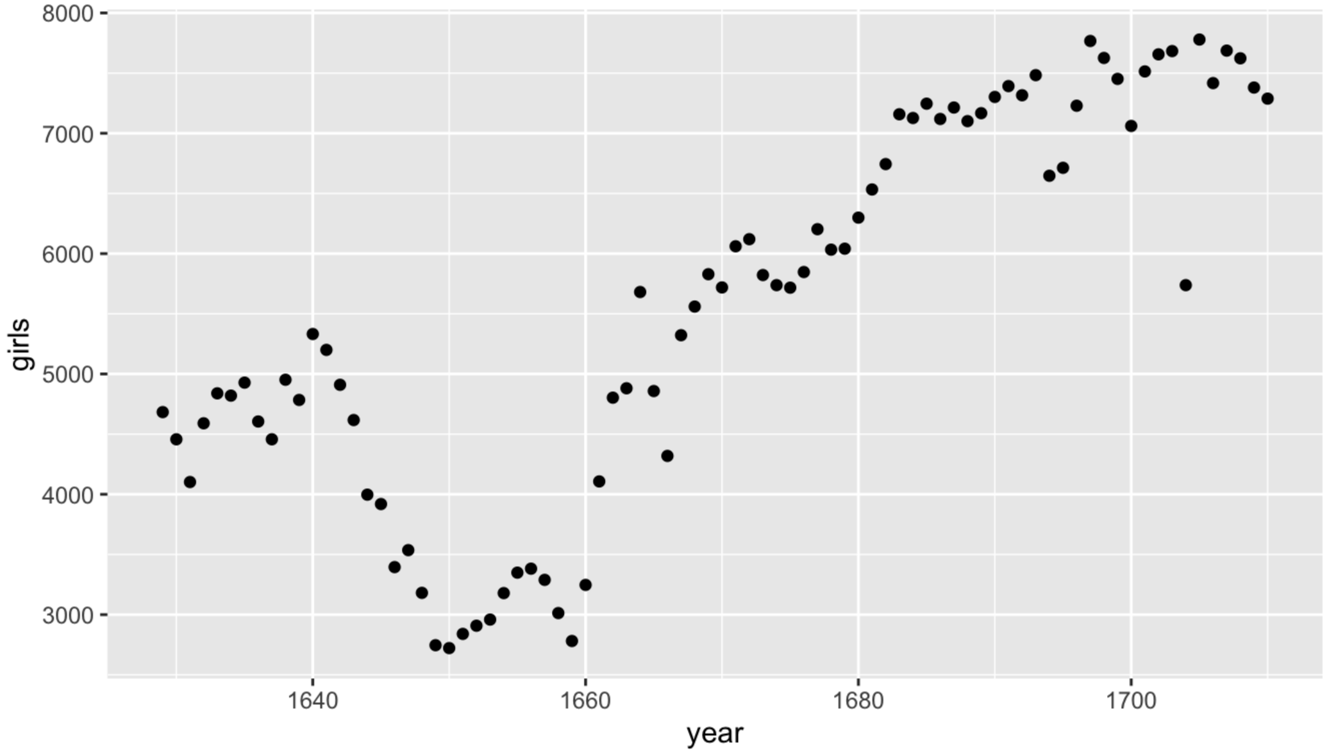
```
arbuthnot$year
```

```
##  [1] 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643
## [16] 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658
## [31] 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673
## [46] 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688
## [61] 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703
## [76] 1704 1705 1706 1707 1708 1709 1710
```

This command will only show the years baptized. The dollar sign basically says “go to the data frame that comes before me, and find the variable that comes after me”.

Create Scatter Plot

```
ggplot(data = arbuthnot, aes(x = year, y = girls)) +
  geom_point()
```



Use `ggplot()` function to build plots. You should see the plot appear under the *Plots* tab of the lower right panel of RStudio.

- The first argument is always the dataset.
- Then provide the variables from the dataset to be assigned to `aes`thetic elements of the plot, e.g. the x and the y axes.
- Finally, we use another layer, separated by a `+` to specify the `geom`etric object for the plot. Since we want to scatterplot, we use `geom_point`.

Find Synatx for a Function To find more syntax for the `ggplot` function just type in a question mark followed by the name of the function that you’re interested in.

```
?ggplot
```

Notice that the help file replaces the plot in the lower right panel. You can toggle between plots and help files using the tabs at the top of that panel.

More extensive help for plotting with the `ggplot2` package can be found at http://docs.ggplot2.org/current/ (http://docs.ggplot2.org/current/).

R as a big calculator

Simple Addition

```
5218 + 4683
```

```
## [1] 9901
```

Add Vectors

```
arbuthnot$boys + arbuthnot$girls
```

```
##  [1]  9901  9315  8524  9584  9997  9855 10034  9522  9160 10311 10150 10850
## [13] 10670 10370  9410  8104  7966  7163  7332  6544  5825  5612  6071  6128
## [25]  6155  6620  7004  7050  6685  6170  5990  6971  8855 10019 10292 11722
## [37]  9972  8997 10938 11633 12335 11997 12510 12563 11895 11851 11775 12399
## [49] 12626 12601 12288 12847 13355 13653 14735 14702 14730 14694 14951 14588
## [61] 14771 15211 15054 14918 15159 13632 13976 14861 15829 16052 15363 14639
## [73] 15616 15687 15448 11851 16145 15369 16066 15862 15220 14928
```

Adding a new variable to the data frame

```
arbuthnot <- arbuthnot %>%
  mutate(total = boys + girls)
```

The `%>%` operator is called the **piping** operator. It takes the output of the current line and pipes it into the following line of code.

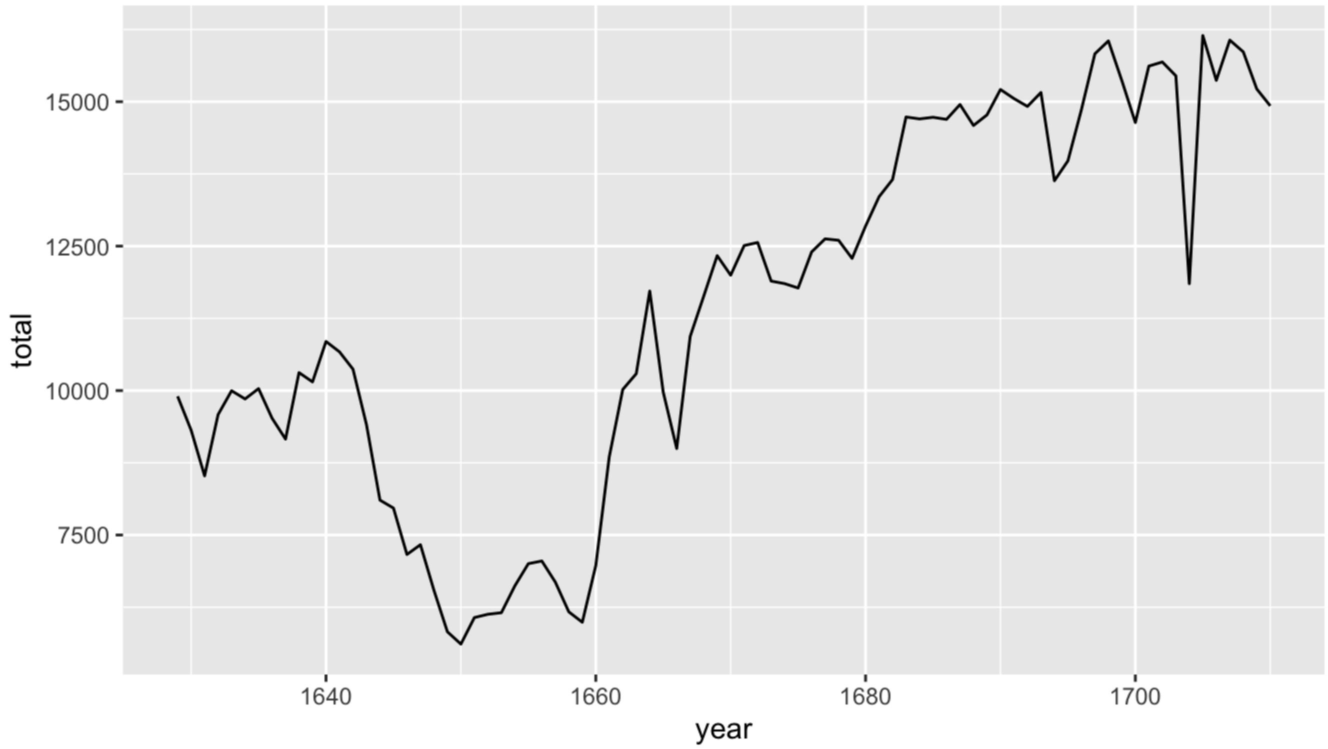
It can be read as *“Take the `arbuthnot` dataset and **pipe** it into the `mutate` function. Mutate a new variable called `total` that is the sum of the variables called `boys` and `girls`. Then overwrite the old `arbuthnot` dataset with the new arbuthnot dataset containing the new variable (total).”*

This is essentially equivalent to going through each row and adding up the boys and girls counts for that year and recording that value in a new column called total.

`<-` performs an *assignment* takes the output of one line of code and saving it into an object in your workspace. In this case, you already have an object called `arbuthnot`, so this command updates that data set with the new mutated column.

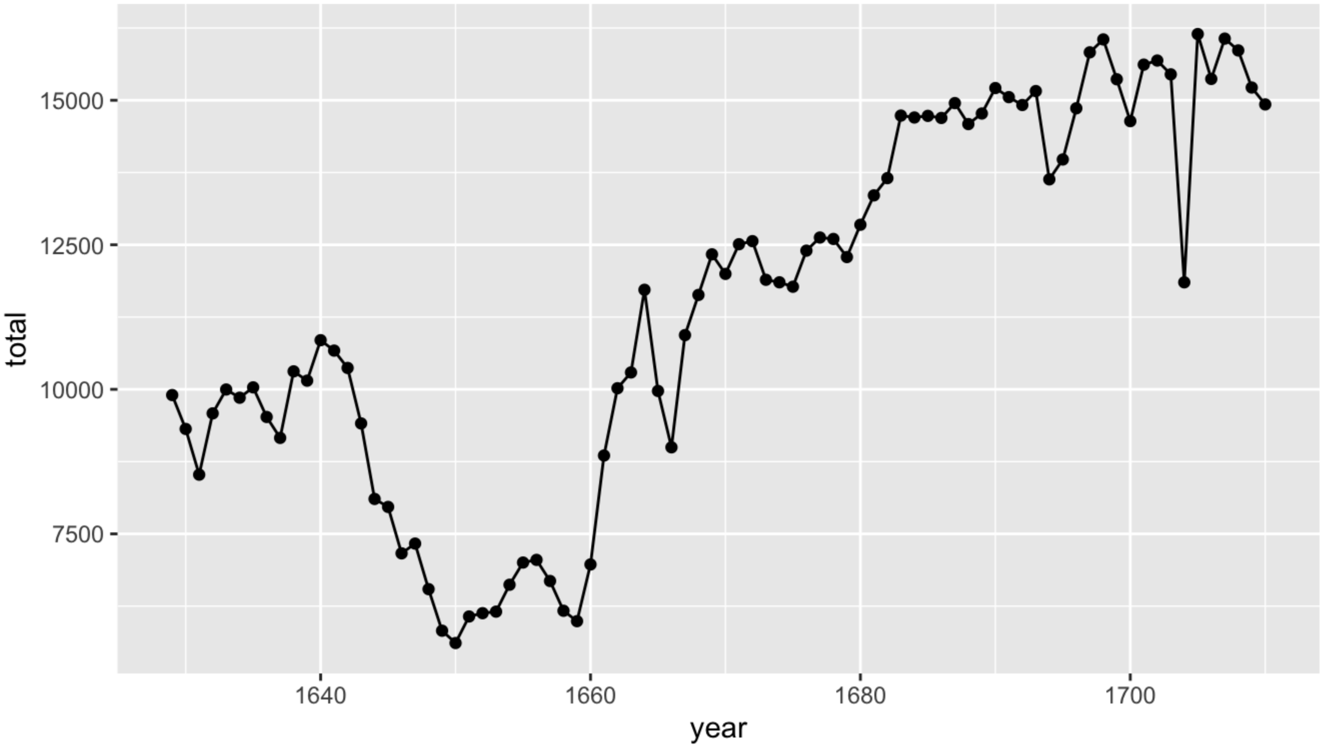
** Create a Line Graph**

```
ggplot(data = arbuthnot, aes(x = year, y = total)) +
  geom_line()
```



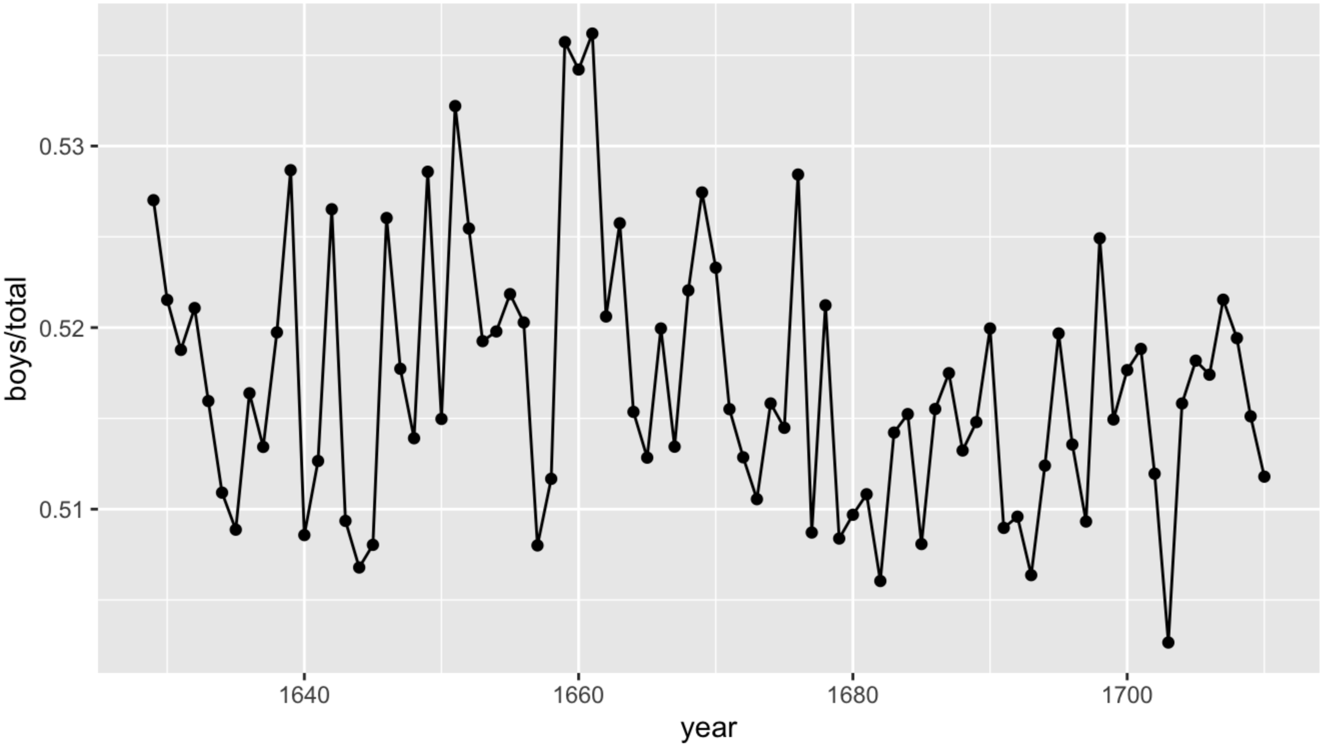
Note that using `geom_line()` instead of `geom_point()` results in a line plot instead of a scatter plot. You want both? Just layer them on:

```
ggplot(data = arbuthnot, aes(x = year, y = total)) +
  geom_line() +
  geom_point()
```



Graph Using Arithmetics

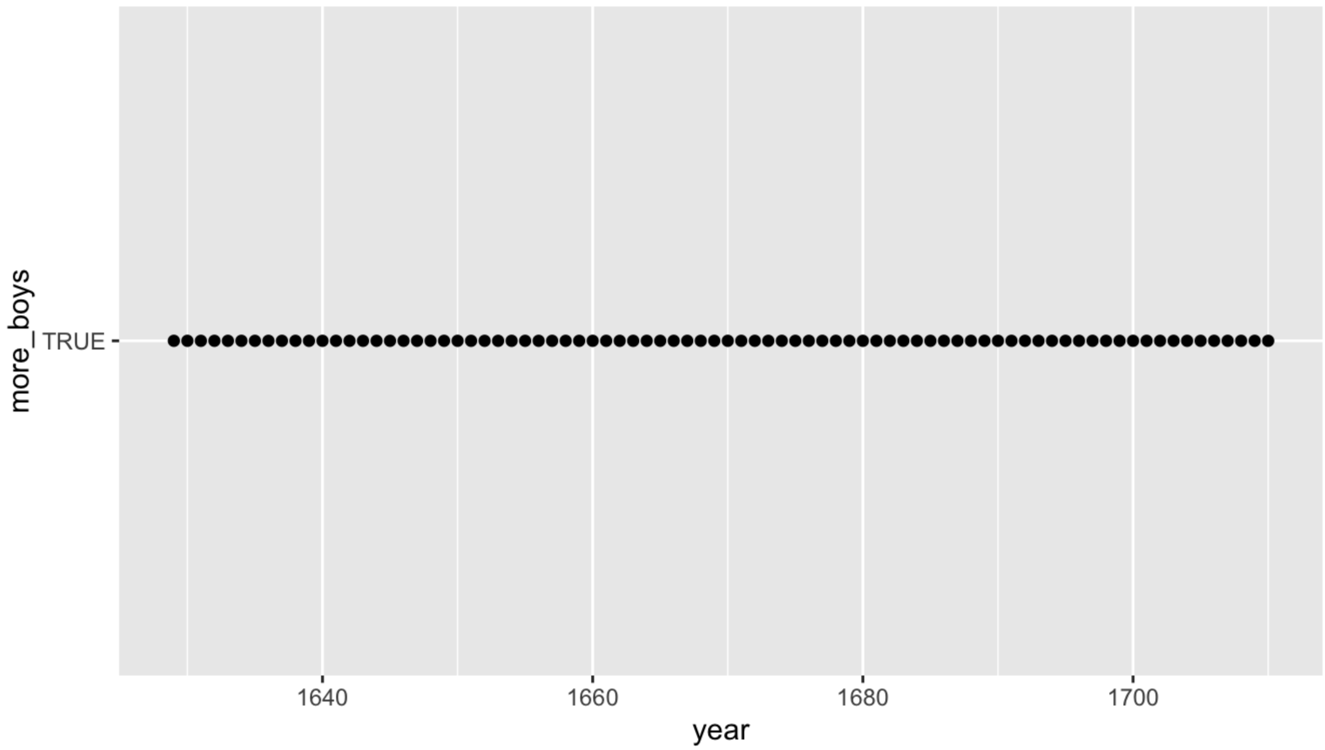
```
ggplot(data = arbuthnot, aes(x = year, y = boys/total)) +
  geom_line() +
  geom_point()
```



Graph using Comparisons

```
arbuthnot <- arbuthnot %>%
  mutate(more_boys = boys > girls)

ggplot(data = arbuthnot, aes(x = year, y = more_boys)) +
  geom_point()
```



R can make comparisons like greater than, `>` , less than, < , and equality, `==` . For example, we can ask if boys outnumber girls in each year with the expression This command add a new variable to the `arbuthnot` data frame containing the values of either `TRUE` if that year had more boys than girls, or `FALSE` if that year did not (the answer may surprise you). Here,we’ve asked R to create *logical* data.

Resources for learning R and working in RStudio

- The following cheathseets may come in handy throughout the course. Note that some of the code on these cheatsheets may be too advanced for this course, however majority of it will become useful as you progress through the course material.
    - Data wrangling cheatsheet (http://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf)
    - Data visualization cheatsheet (http://www.rstudio.com/wp-content/uploads/2015/12/ggplot2-cheatsheet-2.0.pdf)
    - R Markdown (http://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf)
- While you will get plenty of exercise working with these packages in the labs of this course, if you would like further opportunities to practice we recommend checking out the relevant courses at DataCamp (https://www.datacamp.com/courses).