R programming

Philip J Cwynar

University of Pittsburgh
School of Information Sciences
and Intelligent Systems Program

Background

R is a programming language and software environment for statistical analysis, graphics representation and reporting.

R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team.

R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems like Linux, Windows and Mac.

Intro

The following important features of R:

R is a well-developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.

R has an effective data handling and storage facility,

R provides a suite of operators for calculations on arrays, lists, vectors and matrices.

R provides a large, coherent and integrated collection of tools for data analysis.

R provides graphical facilities for data analysis and display either directly at the computer or printing at the papers.

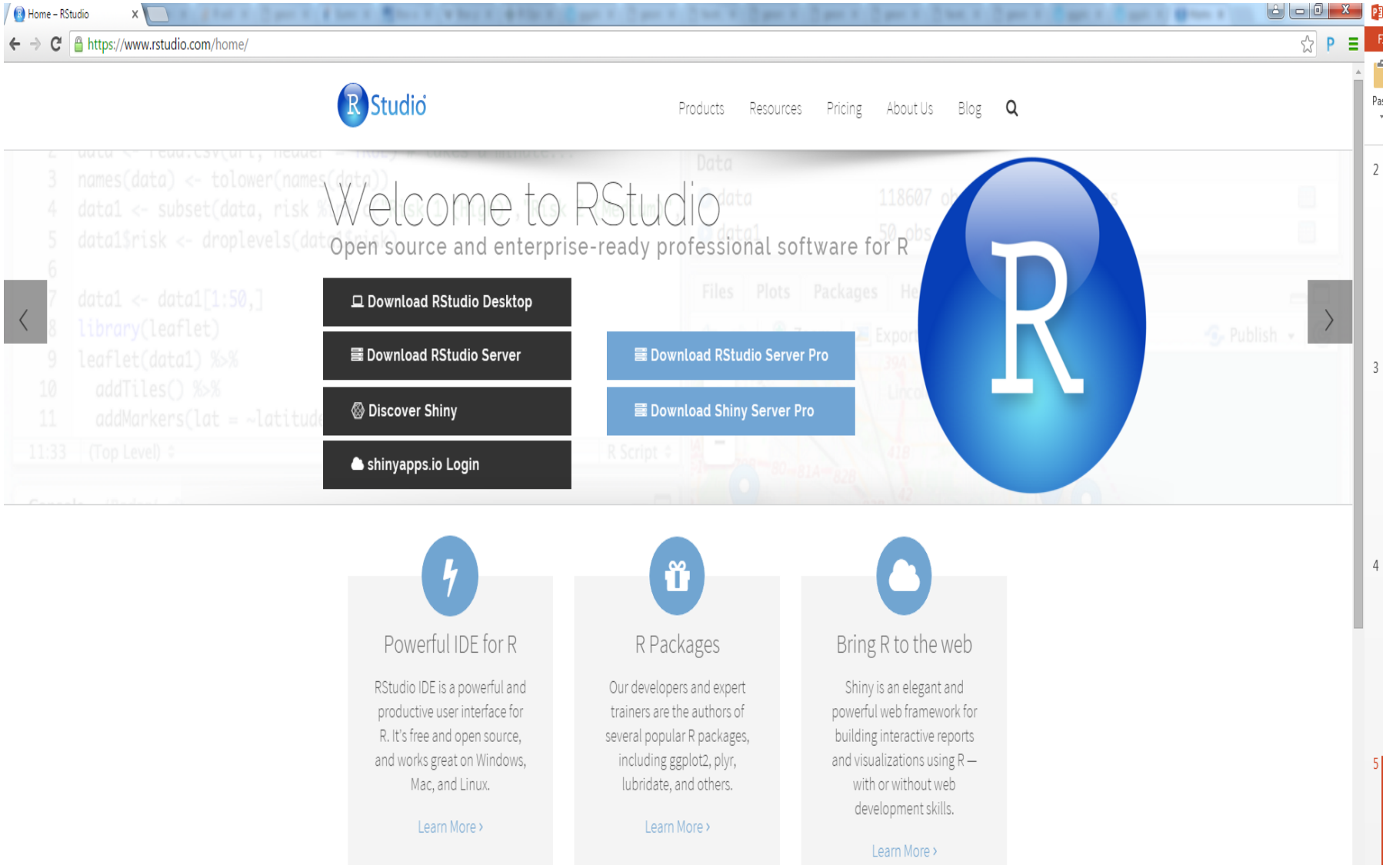
R is world’s most widely used statistics programming language.

It's the # 1 choice of data scientists and supported by a vibrant and talented community of contributors. R is taught in universities and deployed in mission critical business applications.

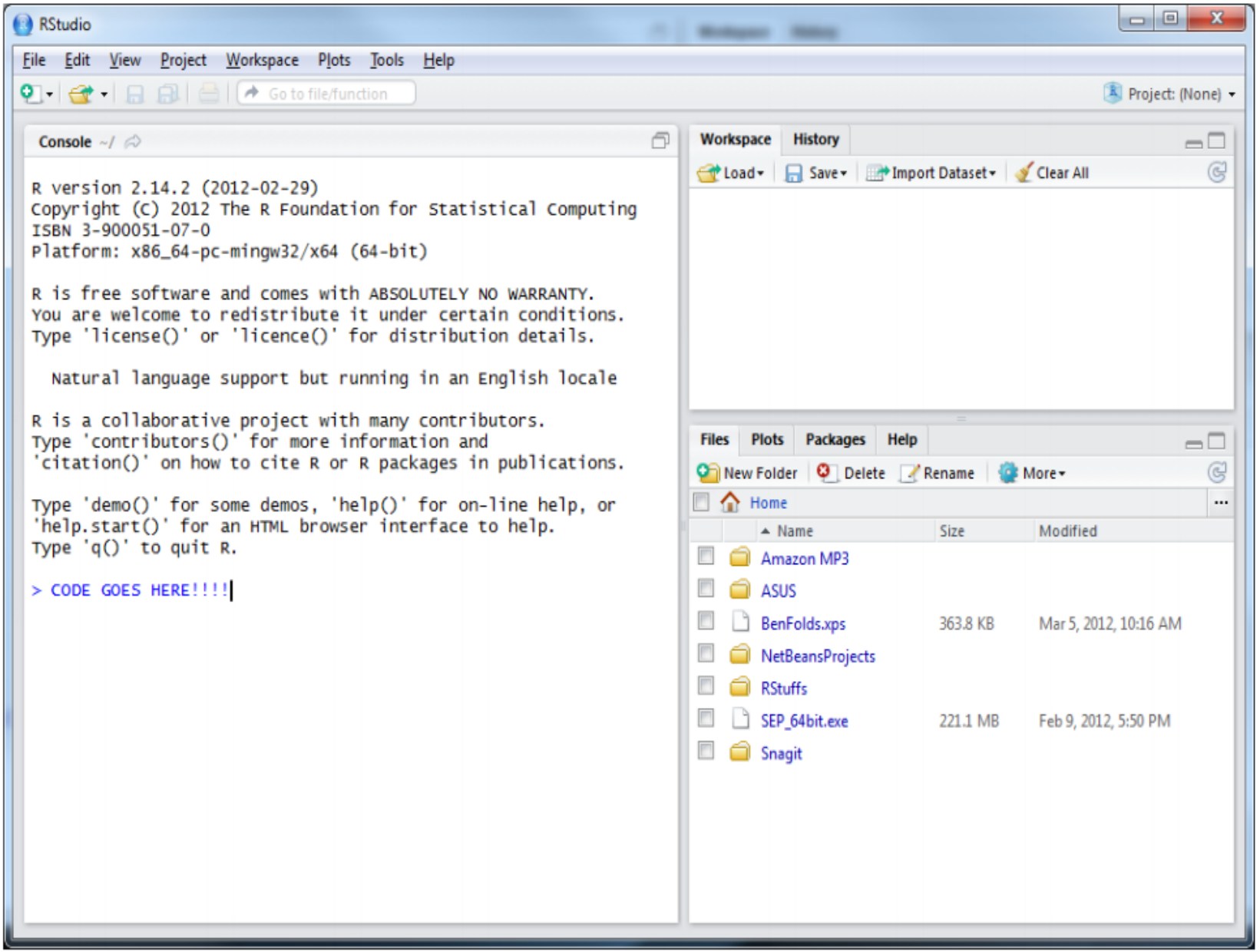
R studio

https://www.rstudio.com/

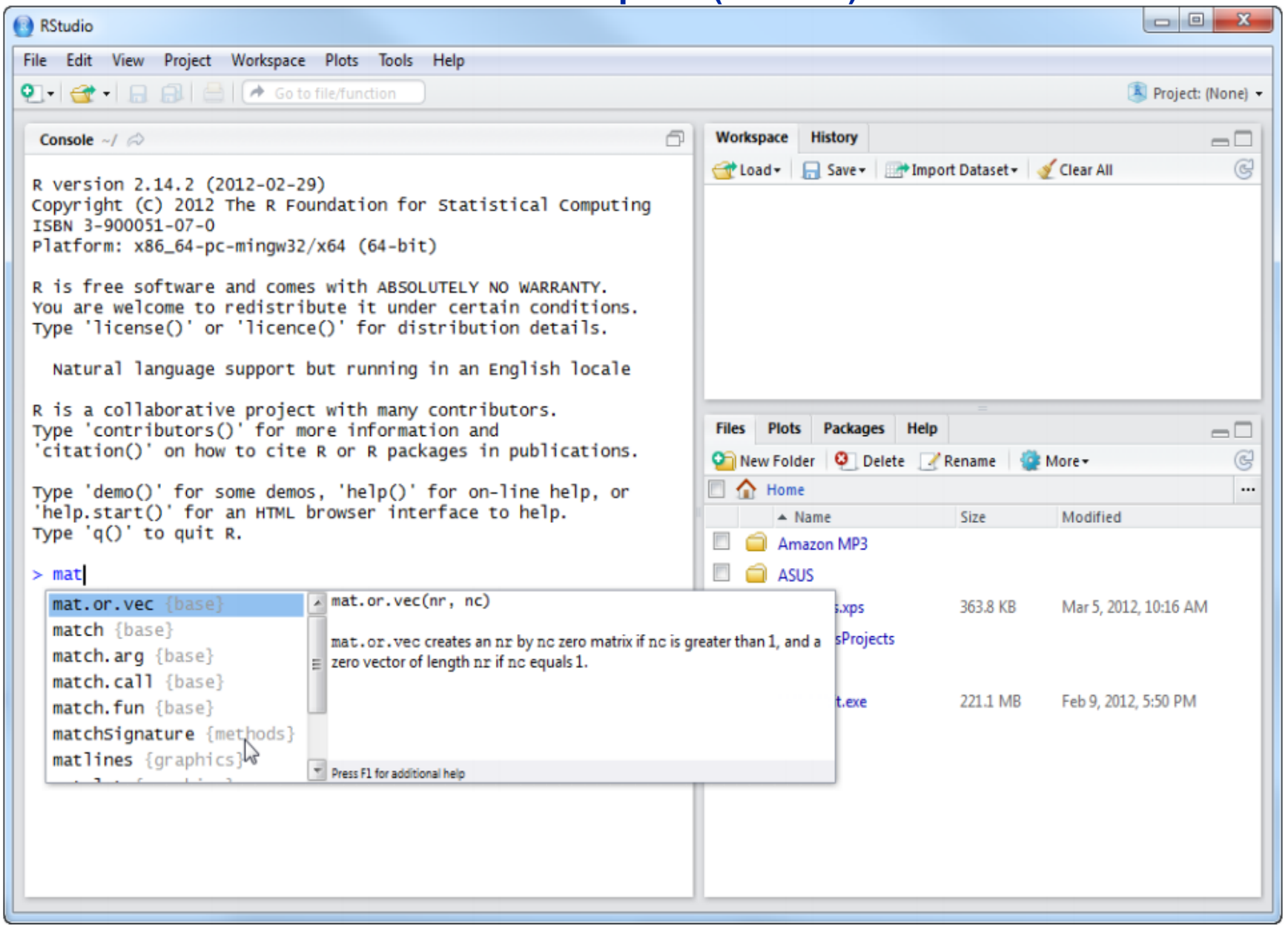
https://www.rstudio.com/products/rstudio/download/


Home – RStudio
https://www.rstudio.com/home/
RStudio
Products
Resources
Pricing
About Us
Blog
Welcome to RStudio
Open source and enterprise-ready professional software for R
Download RStudio Desktop
Download RStudio Server
Discover Shiny
shinyapps.io Login
Download RStudio Server Pro
Download Shiny Server Pro
Powerful IDE for R
RStudio IDE is a powerful and productive user interface for R. It's free and open source, and works great on Windows, Mac, and Linux.
Learn More ›
R Packages
Our developers and expert trainers are the authors of several popular R packages, including ggplot2, plyr, lubridate, and others.
Learn More ›
Bring R to the web
Shiny is an elegant and powerful web framework for building interactive reports and visualizations using R — with or without web development skills.
Learn More ›

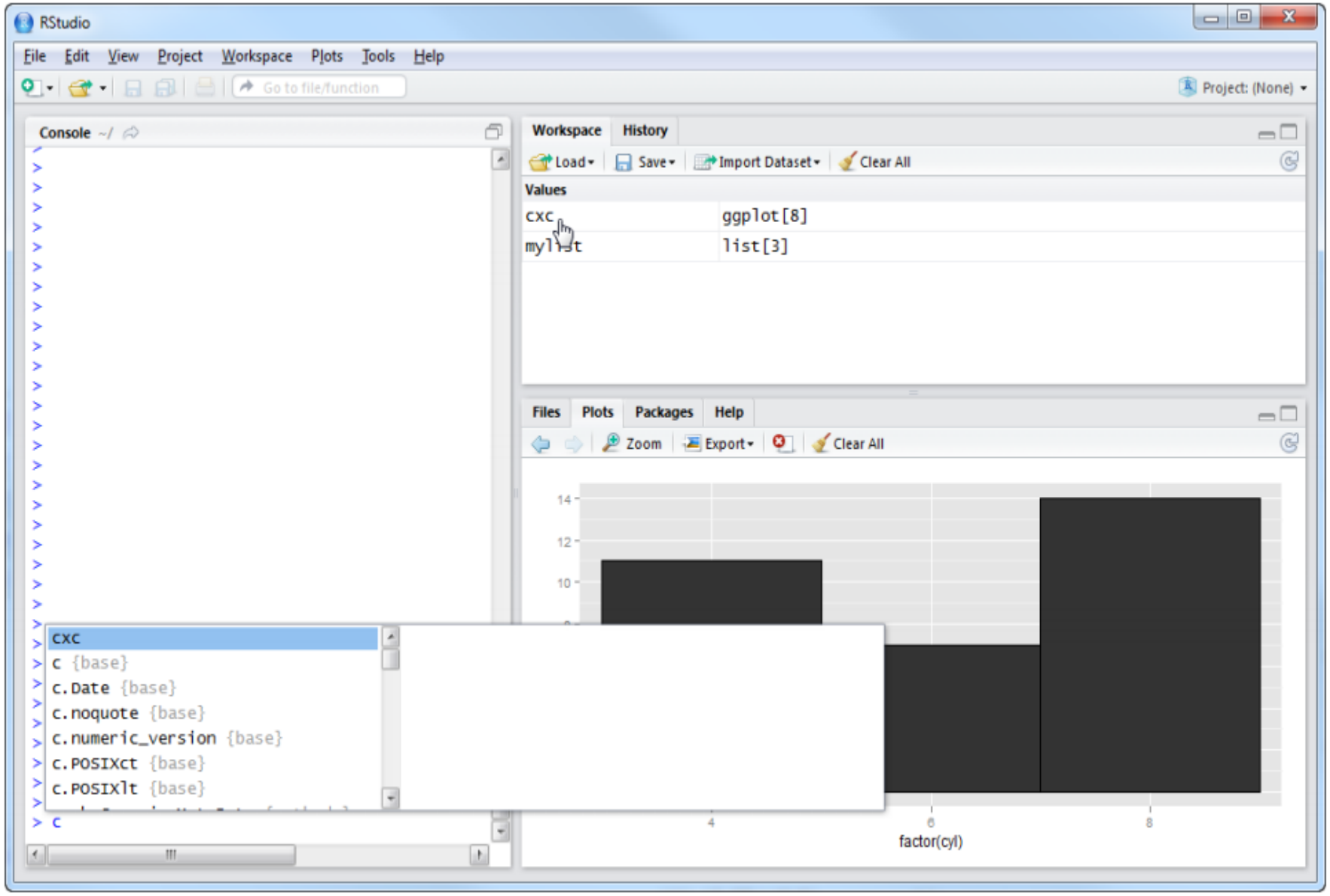
IDE Layout


RStudio
File Edit View Project Workspace Plots Tools Help
Go to file/function
Project: (None)
Console ~/
R version 2.14.2 (2012-02-29)
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-mingw32/x64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
> CODE GOES HERE!!!!
Workspace History
Load Save Import Dataset Clear All
Files Plots Packages Help
New Folder Delete Rename More
Home
Name Size Modified
Amazon MP3
ASUS
BenFolds.xps 363.8 KB Mar 5, 2012, 10:16 AM
NetBeansProjects
RStuffs
SEP_64bit.exe 221.1 MB Feb 9, 2012, 5:50 PM
Snagit

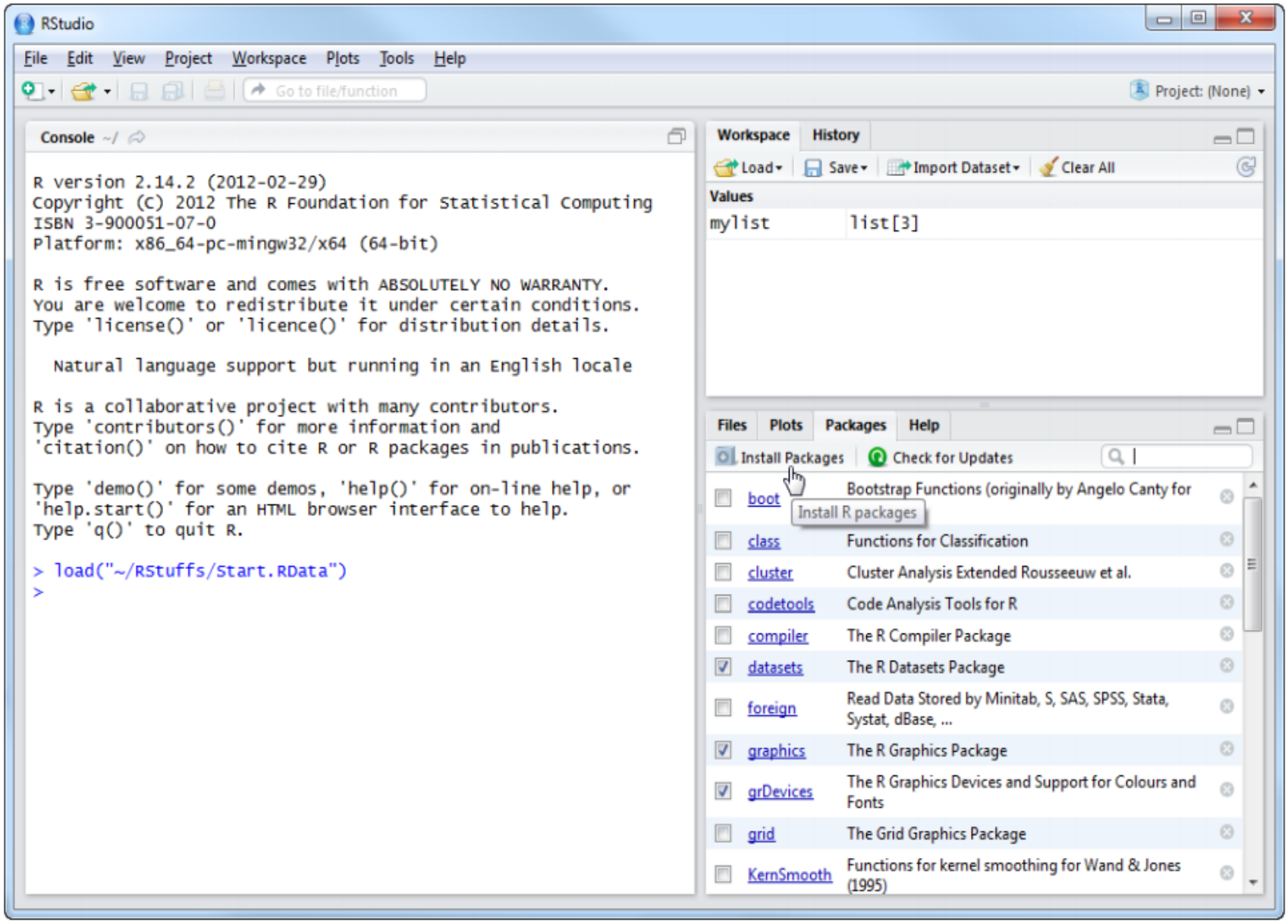
Auto-Complete (use tab)



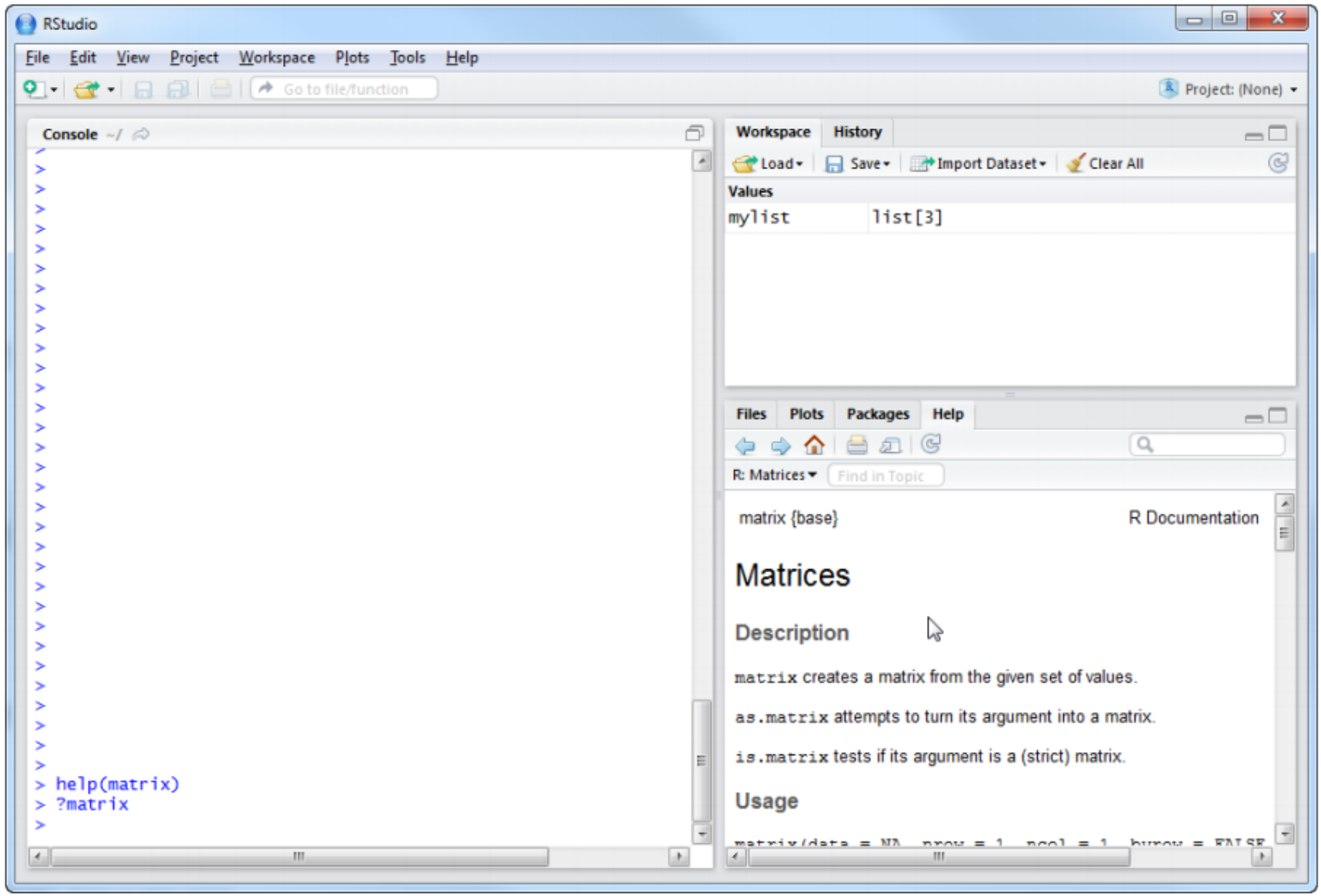
Auto-Complete (even variables!)



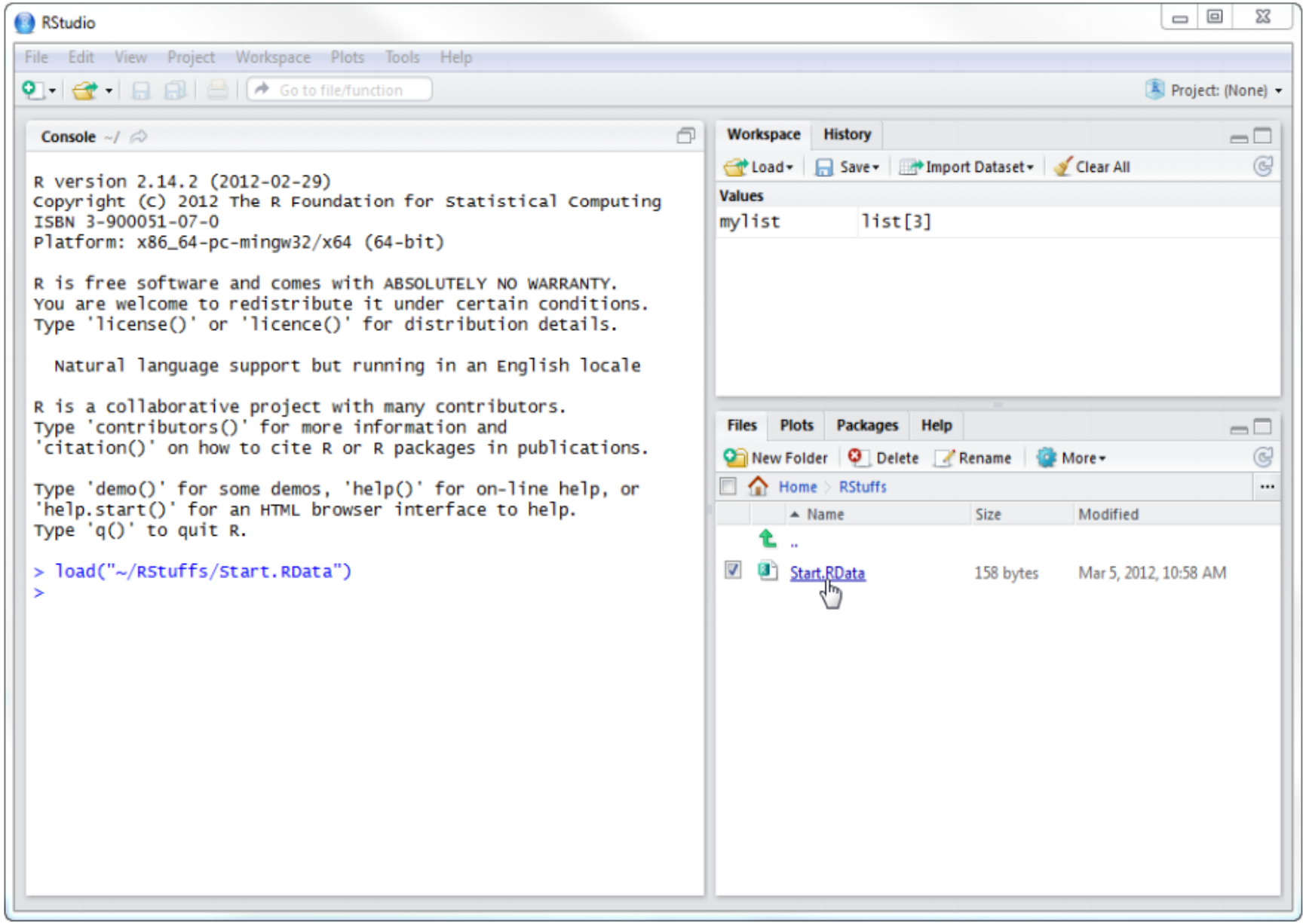
Installing New Packages



Need Help with functions? Use ? Or help()



Loading a file



R 101

R - Data Types

In contrast to other Programming languages like C and java in R the variables are not declared as some data type.

The variables are assigned with R-Objects and the data type of the R-object becomes the data type of the variable ***just like in Python***

There are many types of R-objects.

The frequently used ones are

- **Vectors**
- **Lists**
- **Matrices**
- **Arrays**
- **Factors**
- **Data Frames**

Mode and Class

- R looks at two things:
- Mode
  - How the data is formatted (think int, float, char, matrix, list)
  - Always numeric
- Class
  - Overall structure

```
> x = 1:9
```

```
> dim(x) <- c(3,3)
> x
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> mode(x)
[1] "numeric"
> class(x)
[1] "matrix"
```

R - Data Types cont...

Data Type	Example	Command Line
Logical	TRUE , FALSE	v <- TRUE print(class(v)) it produces following result: [1] "logical"
Numeric	12.3, 5, 999	v <- 23.5 print(class(v)) it produces following result: [1] "numeric"
Integer	2L, 34L, 0L	v <- 2L print(class(v)) it produces following result: [1] "integer"
Complex	3 + 2i	v <- 2+5i print(class(v)) it produces following result: [1] "complex"
Character	'a' , '"good", "TRUE", '23.4'	v <- "TRUE" print(class(v)) it produces following result: [1] "character"
Raw	"Hello" is stored as 48 65 6c 6c 6f	v <- charToRaw("Hello") print(class(v)) it produces following result: [1] "raw"

Variable Assignment

The variables can be assigned values using leftward, rightward and equal to operator. The values of the variables can be printed using **print()** or **cat()**function. The **cat()** function combines multiple items into a continuous print output.

```
# Assignment using equal operator.
var1 = “abcdefg”

# Assignment using leftward operator.
var2 <- 12345

# Assignment using rightward operator.
c(TRUE,1) -> var.3 print(var.1) cat ("var.1
is ", var.1 ,"\n") cat ("var.2 is ", var.
2 ,"\n") cat ("var.3 is ", var.3 ,"\n")
```

Vectors

A **vector** is a sequence of data elements of the same basic type. Members in a vector are officially called **components**. Nevertheless, we will just call them **members** in this site.

When you want to create vector with more than one element, you should use **c()** concatenate function which means to combine the elements into a vector.

```
# Create a vector.
apple <- c('red','green',"yellow")
>apple

 # Get the class of the vector.
 class(apple)
```

When we execute above code, it produces following result:

```
[1] "red" "green" "yellow"

[1] "character"
```

Creating Vectors

The `c()` function can be used to create vectors of objects.

```
> x <- c(0.5, 0.6)       ## numeric
> x <- c(TRUE, FALSE)    ## logical
> x <- c(T, F)           ## logical
> x <- c("a", "b", "c")  ## character
> x <- 9:29              ## integer
> x <- c(1+0i, 2+4i)     ## complex
```

Using the `vector()` function

```
> x <- vector("numeric", length = 10)
> x
 [1] 0 0 0 0 0 0 0 0 0 0
```

Mixing Objects

What about the following?

```
> y <- c(1.7, "a")   ## character
> y <- c(TRUE, 2)    ## numeric
> y <- c("a", TRUE)  ## character
```

When different objects are mixed in a vector, *coercion* occurs so that every element in the vector is of the same class.

Let's Make a Vector!

- Remember me? I'm a Python scalar.

someInt = 12

someString = "hello Pittsburgh!"

- Remember me? I'm a Python list.

someIntList = [1,3,4,19]

someStringList = ["apple", "banana", "kittens"]

- In R, I am handled like this:

> newVector = c(1,2,4,19)

> newVector

[1] 1 2 4 19

> newVector[2]

[1] 2 (Wait a second!) (2 = 2?) (Shouldn't 1 = 2?)

1 is the new 0

- "A single 0 in an index position returns an empty structure;. x[0] returns named numeric(0)."
- So in Python it's 0.. In R, its 1
  - Got that? Good..

http://cran.r-project.org/doc/contrib/R_language.pdf

Matrices

All the elements of a matrix must be of the same type (numeric, logical, character, complex).

- Maybe the most important lesson to be learned today
- Stored as a single vector with columns stacked together

USAGE: matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)

data = the data going into the matrix

nrow = # of rows

ncol = # of columns

byrow = TRUE or FALSE

dimnames = dimension names

Matrices

Matrices are vectors with a *dimension* attribute. The dimension attribute is itself an integer vector of length 2 (nrow, ncol)

```
> m <- matrix(nrow = 2, ncol = 3)
> m
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
> dim(m)
[1] 2 3
> attributes(m)
$dim
[1] 2 3
```

Let's Add Some Labels!

```
>someMatrix = matrix(1:16, 4, 4, FALSE,
   dimnames=list(c("A","B","C","D"),c("W","X","Y","Z")))
> someMatrix
  W X Y Z
A 1 5 9 13
B 2 6 10 14
C 3 7 11 15
D 4 8 12 16
> someMatrix["B","Y"]
[1] 10
```

*Note: Use NULL if you don't want to append labels to either x or y axis

```
# Create a matrix. M = matrix( c('a','a','b','c','b','a'),
nrow=2,ncol=3,byrow = TRUE)
```

Matrix Example

[No Title]

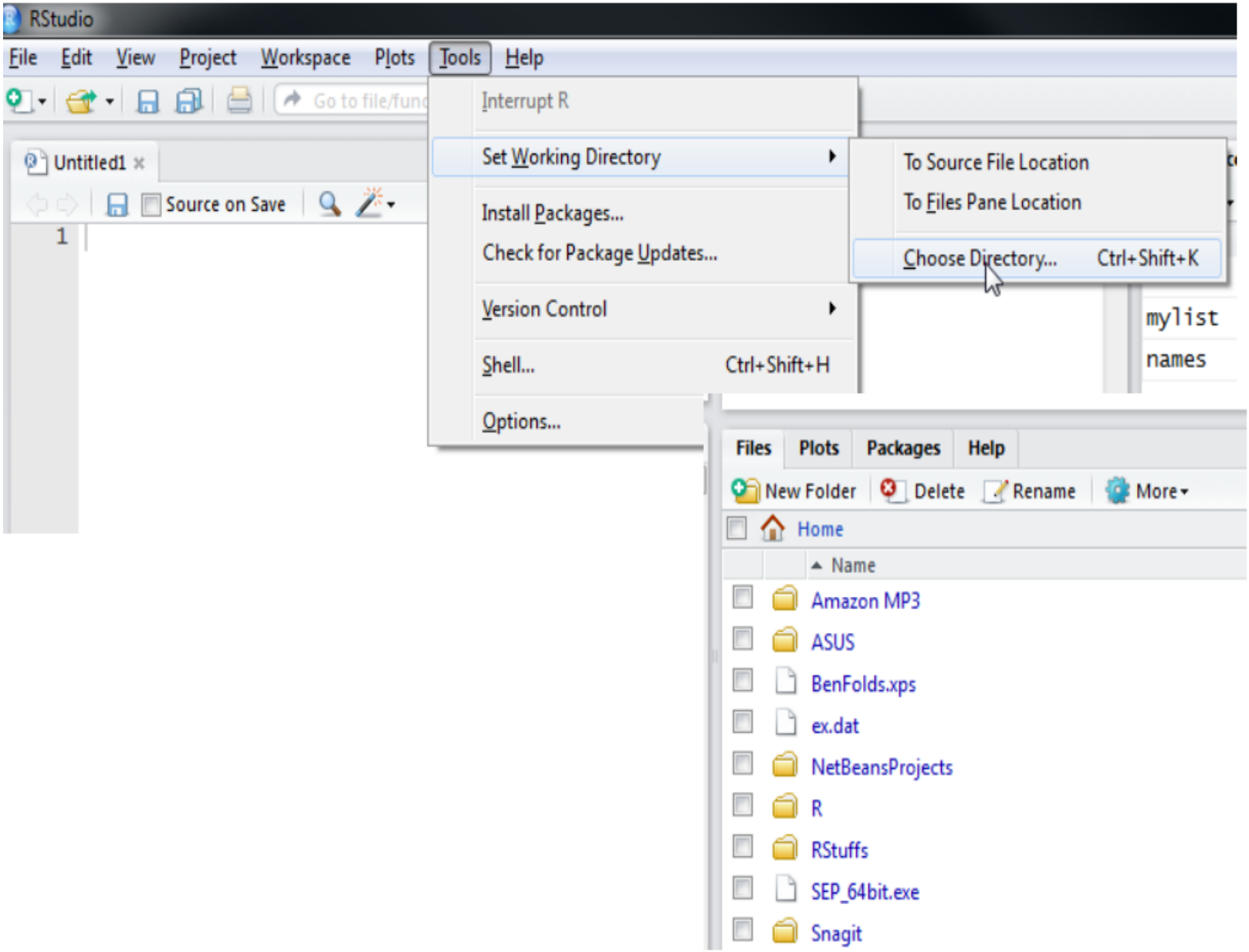
```
> someMatrix = matrix(1:16,4,4)
> someMatrix
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
> someMatrix[10]
[1] 10
> someMatrix[2,3]
[1] 10
```

Data Frames

Data frames are tabular data objects. **Unlike a matrix in data frame each column can contain different modes of data**. The first column can be numeric while the second column can be character and third column can be logical. It is a list of vectors of equal length.

Data Frames are created using the **data.frame()** function.

```
# Create the data frame.
BMI <- data.frame( gender = c("Male", "Male","Female"),
height = c(152, 171.5, 165), weight = c(81,93, 78), Age =c(4
print(BMI)
```


RStudio
File Edit View Project Workspace Plots Tools Help
Interrupt R
Set Working Directory
Install Packages...
Check for Package Updates...
Version Control
Shell... Ctrl+Shift+H
Options...
To Source File Location
To Files Pane Location
Choose Directory... Ctrl+Shift+K
Untitled1
Source on Save
1
mylist
names
Files Plots Packages Help
New Folder Delete Rename More
Home
Name
Amazon MP3
ASUS
BenFolds.xps
ex.dat
NetBeansProjects
R
RStuffs
SEP_64bit.exe
Snagit

Read.Table()

- The majority of data is handled using a table (especially in ggplot)

USAGE: read.table(file, header = FALSE, sep = "", quote = "\"'", dec = ".", row.names, col.names, as.is = !stringsAsFactors, na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names = TRUE, fill = !blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE, comment.char = "#", allowEscapes = FALSE, flush = FALSE, stringsAsFactors = default.stringsAsFactors(), fileEncoding = "", encoding = "unknown", text)

Header

Made a slight change to "BeerAndWinePerCapita.txt“

State,GallonsOfBeer,GallonsOfWine
nEVADA,44,5.75

```
fromFile = read.table("BeerAndWinePerCapita.txt", header = TRUE, sep = ",")
> fromFile
                State GallonsOfBeer GallonsOfWine
1 nEVADA               44          5.75
2 nEW hAMPSHIRE        43.4        6.26
3 nORTH dAKOTA         41.7        1.56
4 mONTANA              41.5        3.06
5 sOUTH dAKOTA         39          1.50
6 wISCONSIN            38.2        2.63
```

```
>mydata <- read.table("Retention.txt",header=TRUE,sep="\t")
>summary(mydata)
```

spend	apret	top10	rejr
Min. : 4125	Min. :18.75	Min. : 8.00	Min. : 0.00
1st Qu.: 7372	1st Qu.:45.37	1st Qu.:22.00	1st Qu.:19.17
Median : 9265	Median :55.71	Median :30.00	Median :27.39
Mean :10975	Mean :56.72	Mean :38.46	Mean :30.65
3rd Qu.:12838	3rd Qu.:68.69	3rd Qu.:49.50	3rd Qu.:36.81
Max. :35863	Max. :95.25	Max. :98.00	Max. :84.07
tstsc	**pacc**	**strat**	**salar**
Min. :48.12	Min. : 8.964	Min. : 7.20	Min. :38640
1st Qu.:61.11	1st Qu.:33.904	1st Qu.:13.40	1st Qu.:54650
Median :64.78	Median :40.850	Median :16.00	Median :61150
Mean :66.16	Mean :43.173	Mean :16.09	Mean :61358
3rd Qu.:70.45	3rd Qu.:51.773	3rd Qu.:18.57	3rd Qu.:67100
Max. :87.50	Max. :76.253	Max. :29.20	Max. :87900

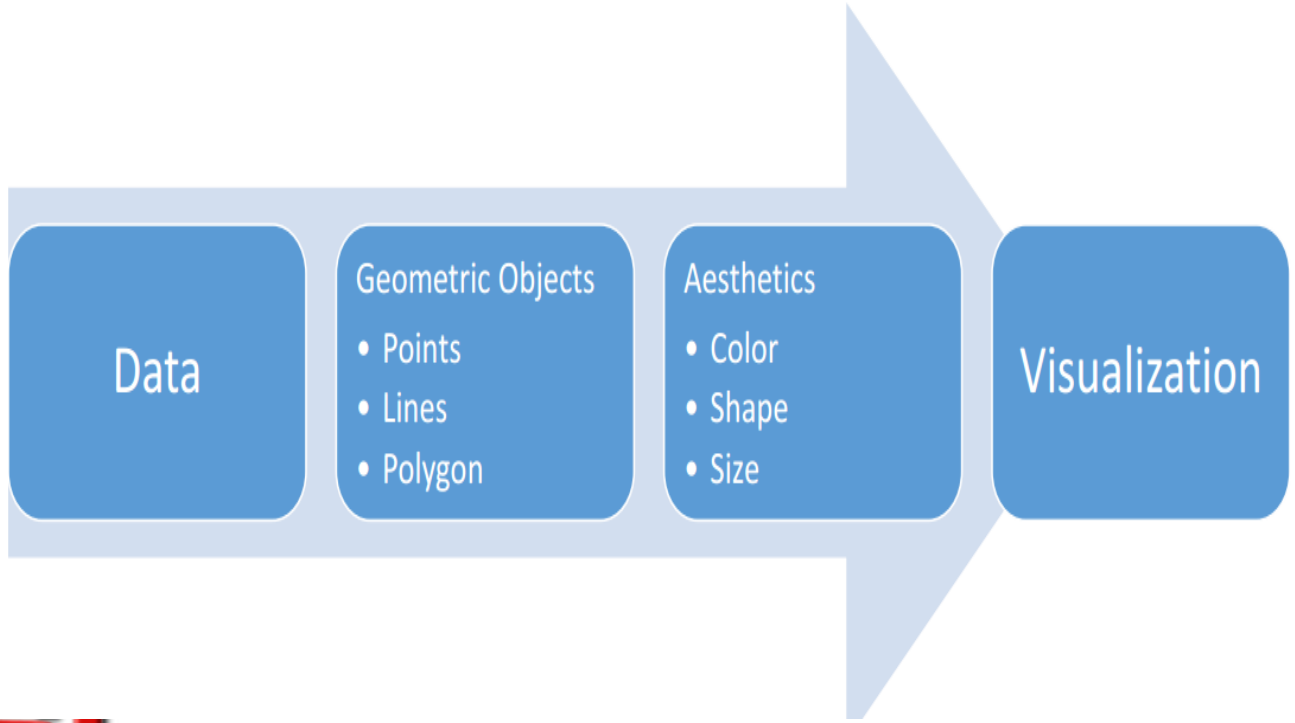
Factors---Types of Variables

- **Nominal (Categorical) –**
  - **Two or more categories**
  - **Have no intrinsic ordering**
  - **Think: (Male/Female), (Blonde/Brunette/Red Hair), and (Pittsburgh/State College/Erie/etc.)**
- **Ordinal**
  - **Similar to Nominal, but with order**
  - **Think: (Low/Medium/High), (Tall/Average/Short), and (Tall Latte/Grande Latte/Venti Latte no whip)**
- **Interval**
  - **Same as Ordinal, but evenly spaced**
  - **Think: (Temperature), (Time), and (Measurements)**

ggplot2

Some Basics of ggplot2

- Extended library of R
- Used to create a variety of visualizations without a lot of background knowledge
- Created in a layered fashion



Terminology

Mapping

Taking the data and the aesthetics and mapping between them to visualize results

Geom

Geometric Objects – Points, lines, polygons (Remind you of something?)

Stat

Statistical Transformation

Example: Taking the data and counting observations to create a histogram

Scale

Gives you an understanding of the range of data in the visualizations

Think: Legend or Axis

Terminology (cont.)

Coord

Coordinate System

Either by Cartesian or polar coordinates

Facet

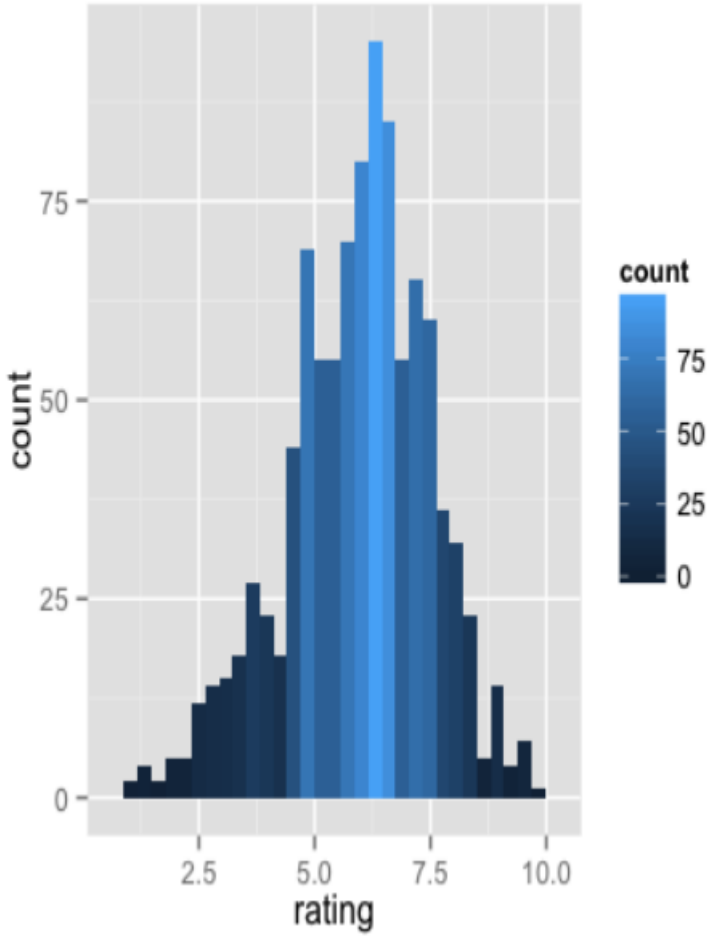
Faceting Specification

Think: Small multiples.. Segmenting data into groups for deeper understanding

A group of nodes in a cluster becoming a single node to understand group to group interaction

Examples

Histogram



- http://docs.ggplot2.org/current/geom_histogram.html

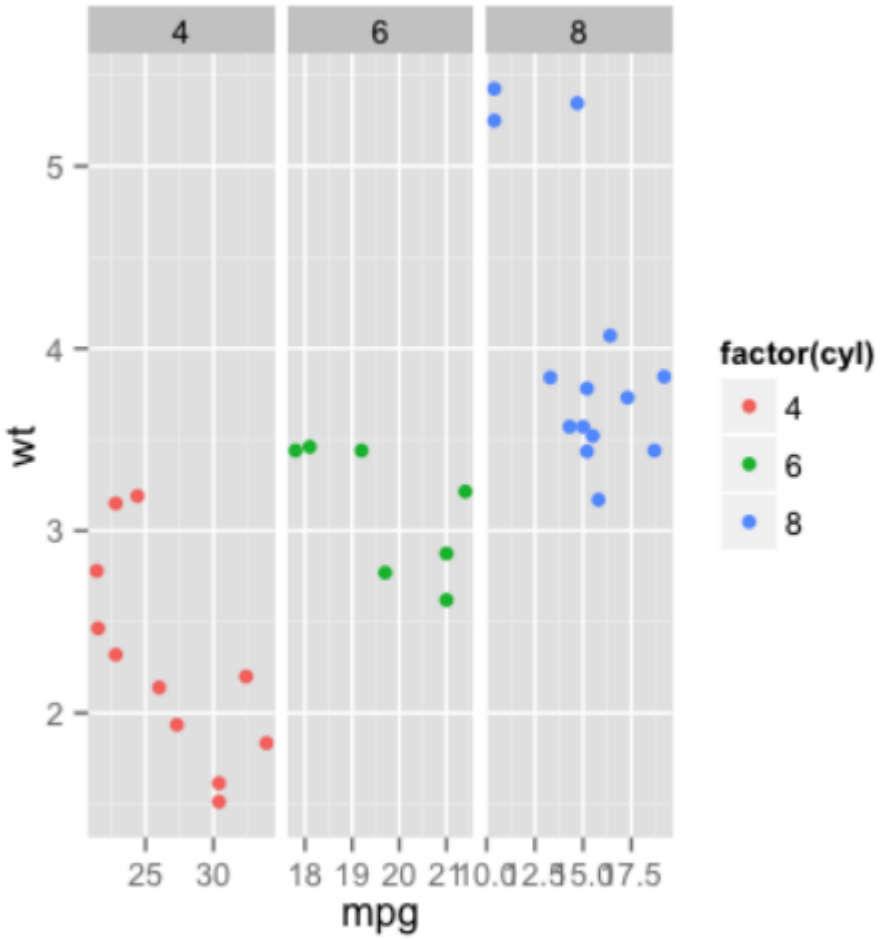
Facet Grid

ggplot2 Quick Reference: **facet**

The faceting approach supported by ggplot2 partitions a plot into a matrix of panels. Each panel shows a different subset of the data. There are two faceting approaches:

- **facet_wrap**(~cell) - univariate: create a 1-d strip of panels, based on one factor, and wrap the strip into a 2-d matrix
- **facet_grid**(row~col) - (usually) bivariate: create a 2-d matrix of panels, based on two factors

Facet Grid

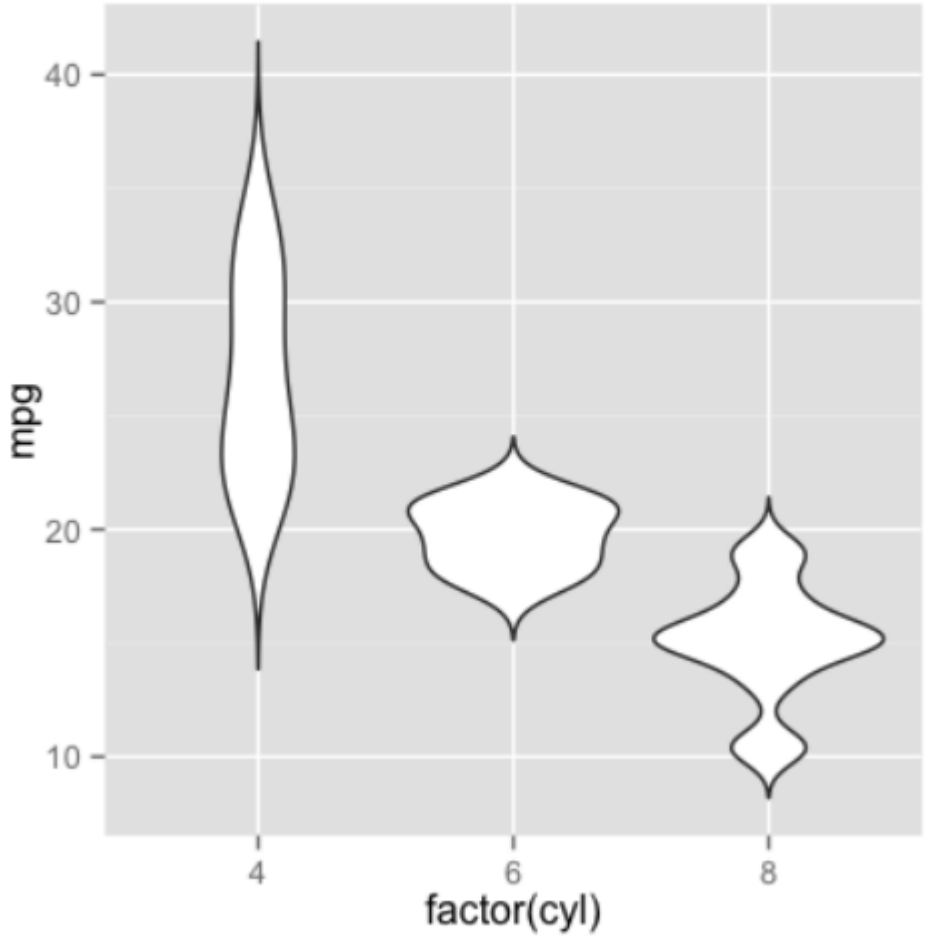


- http://docs.ggplot2.org/current/facet_grid.html
- http://www.cookbook-r.com/Graphs/Facets_(ggplot2)/

Violin Plots

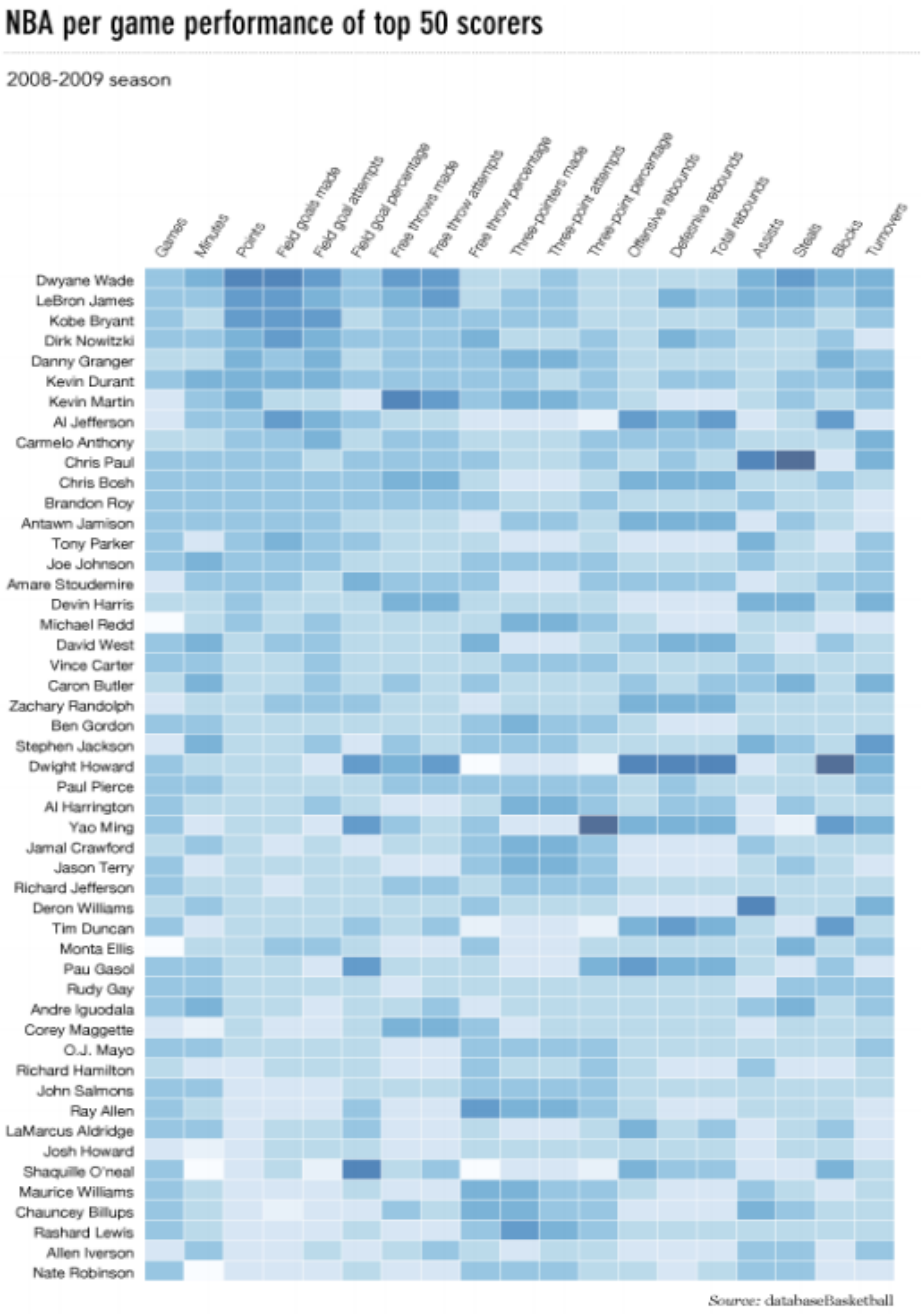
The violin plot is similar to box plots, except that they also show the probability density of the data at different values (in the simplest case this could be a histogram). Typically violin plots will include a marker for the median of the data and a box indicating the interquartile range, as in standard box plots. Overlaid on this box plot is a kernel density estimation.

Violin Plots



- http://docs.ggplot2.org/current/geom_violin.htm

Heat Maps



- https://learnr.wordpress.com/2010/01/26/ggplot2-quick-heatmap-plotting/

Heat Maps cont…

https://learnr.wordpress.com/2010/01/26/ggplot2-quick-heatmap-plotting/

```
nba <- read.csv("http://datasets.flowingdata.com/ppg2008.csv")

nba$Name <- with(nba, reorder(Name, PTS))
```

With

- **"Evaluate an R expression in an environment constructed from data, possibly modifying the original data."**
- **nba$Name <- with(nba, reorder(Name, PTS))**

R Markdown

R Markdown is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R.

http://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf

Useful Links

- https://cran.r-project.org/

- http://zevross.com/blog/2014/08/04/beautiful-plotting-in-r-a-ggplot2-cheatsheet-3/
- http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html
- http://blog.echen.me/2012/01/17/quick-introduction-to-ggplot2/
- http://had.co.nz/ggplot2/
- https://github.com/jennybc/ggplot2-tutorial
- http://www.ceb-institute.org/bbs/wp-content/uploads/2011/09/handout_ggplot2.pdf
- http://www.r-bloggers.com/basic-introduction-to-ggplot2/

Facet

http://sape.inf.usi.ch/quick-reference/ggplot2

Whisker

http://asq.org/learn-about-quality/data-collection-analysis- tools/ overview/box-whisker-plot.html

Violin

http://www.sthda.com/english/wiki/ggplot2-violin-plot-quick-start- guide-r-software-and-data-visualization

Machine learning kaggle titanic

https://www.datacamp.com/courses/kaggle-tutorial-on-machine- learing-the-sinking-of-the-titanic

Kaggle Titanic Assignment

Data Files

File Name	Available Formats
train	.csv (59.76 kb)
gendermodel	.csv (3.18 kb)
genderclassmodel	.csv (3.18 kb)
test	.csv (27.96 kb)
gendermodel	.py (3.58 kb)
genderclassmodel	.py (5.63 kb)
myfirstforest	.py (3.99 kb)

See, fork, and run a random forest benchmark model through Kaggle Scripts

```
VARIABLE DESCRIPTIONS:
survival        Survival
                (0 = No; 1 = Yes)
pclass          Passenger Class
                (1 = 1st; 2 = 2nd; 3 = 3rd)
name            Name
sex             Sex
age             Age
sibsp           Number of Siblings/Spouses Aboard
parch           Number of Parents/Children Aboard
ticket          Ticket Number
fare            Passenger Fare
cabin           Cabin
embarked        Port of Embarkation
                (C = Cherbourg; Q = Queenstown; S = Southampton)
```

Titanic Data Graphics

(1) Go to kaggle.com > Titanic: Machine Learning from Disaster and download (train.csv)

(2) You will submit a single R file (DOCUMENTED)

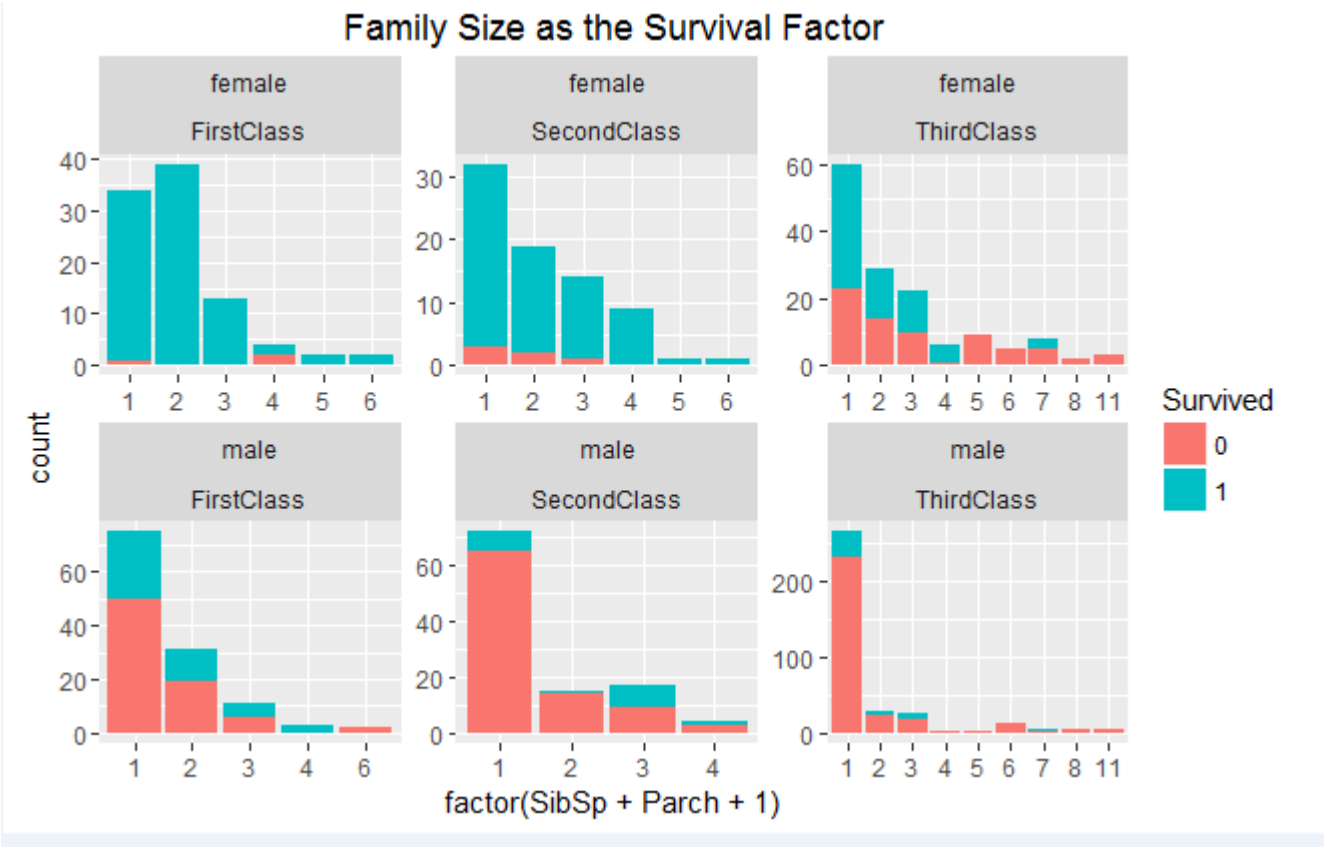
(3) You will create a plot (for each, so 5 plots in total) in ggplot2 using:

- a. Whisker-plot
- b. Histogram
- c. Facet grid
- d. Violin plot
- e. Heatmap

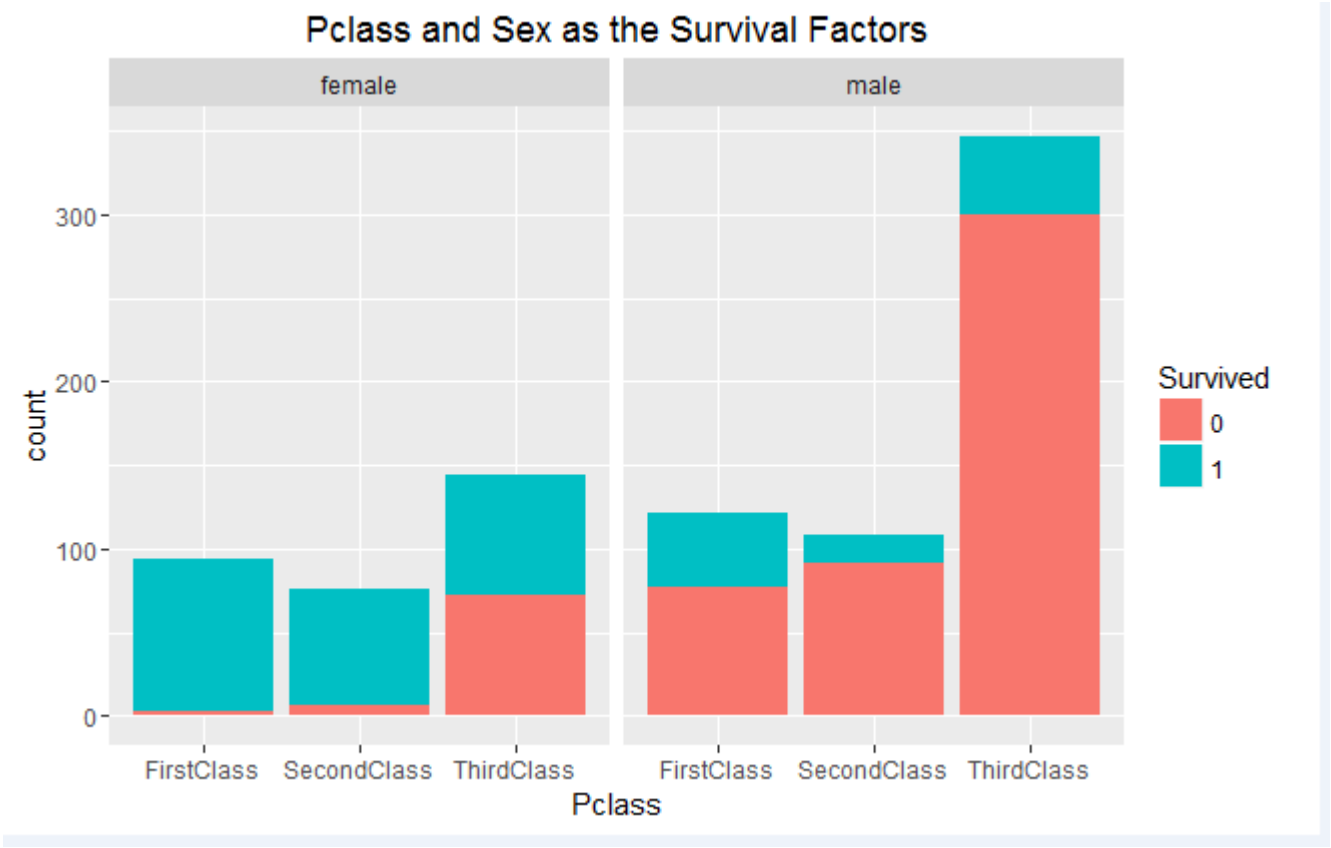
(4) Write up a 500 – 1500 word document talking about the assignment using the graphics to describe the passengers of Titanic.

Examples

```
ggplot(total, aes(factor(SibSp + Parch + 1))) + + geom_bar(aes(fill = Survived)) + +
facet_wrap(~Sex+Pclass, nrow = 2, scales = "free") + + ggtitle("Family Size as the Survival Factor")
```



```
ggplot(total, aes(Pclass)) + + geom_bar(aes(fill = Survived)) + + facet_grid(~Sex) + +
ggtitle("Pclass and Sex as the Survival Factors")
```



Midterm Exam Next week

You can bring with you to the exam one double-sided letter-size sheet of paper with notes.

There are no limits on the font size – you can cram as much information on these two pages as you wish – but the notes have to be handwritten personally by you and this is a strict requirement.

Copied or computer-printed sheets are not allowed.