Introduction of R

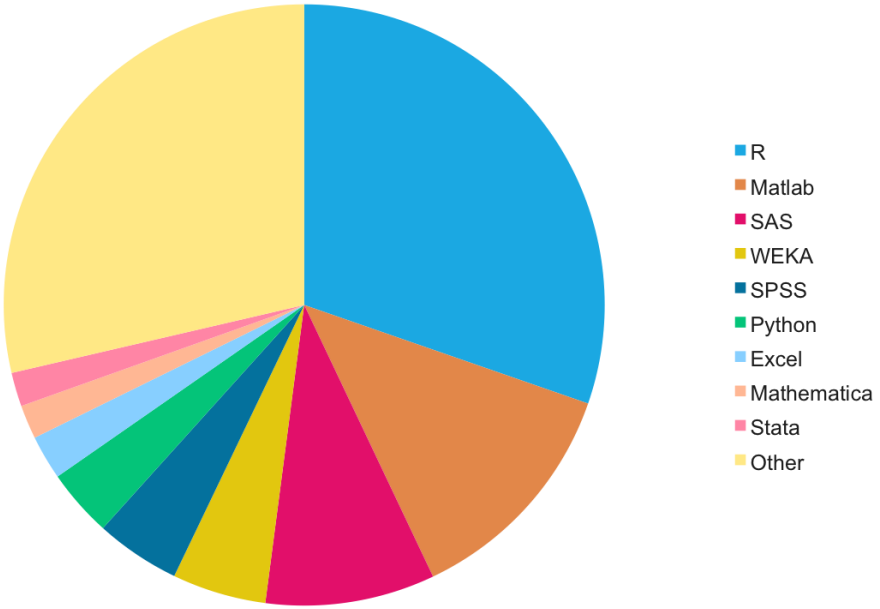
- Overview of R
- Setting up R and basic operations
- Preparing summary statistics
- Basic plotting function

Outline for today's session

- Overview of R
- Setting up R and basic operations
- Preparing summary statistics
- Basic plotting function

What is R?

- Based on S language, written by Robert Gentleman and Ross Ihaka (R&R)
- Programming language for statistical computing & graphics
- Open source

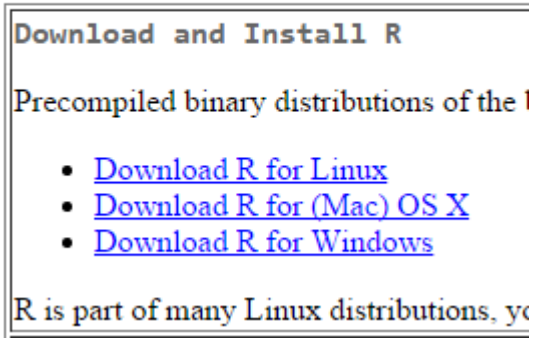


From *http://machinelearningmastery.com/best-programming-language-for-machine-learning/*

R is free!

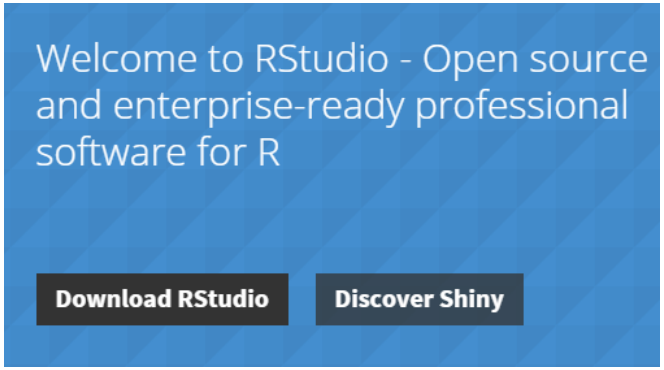
R environment is free.

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



R IDE: rstudio is free.

http://www.rstudio.com/



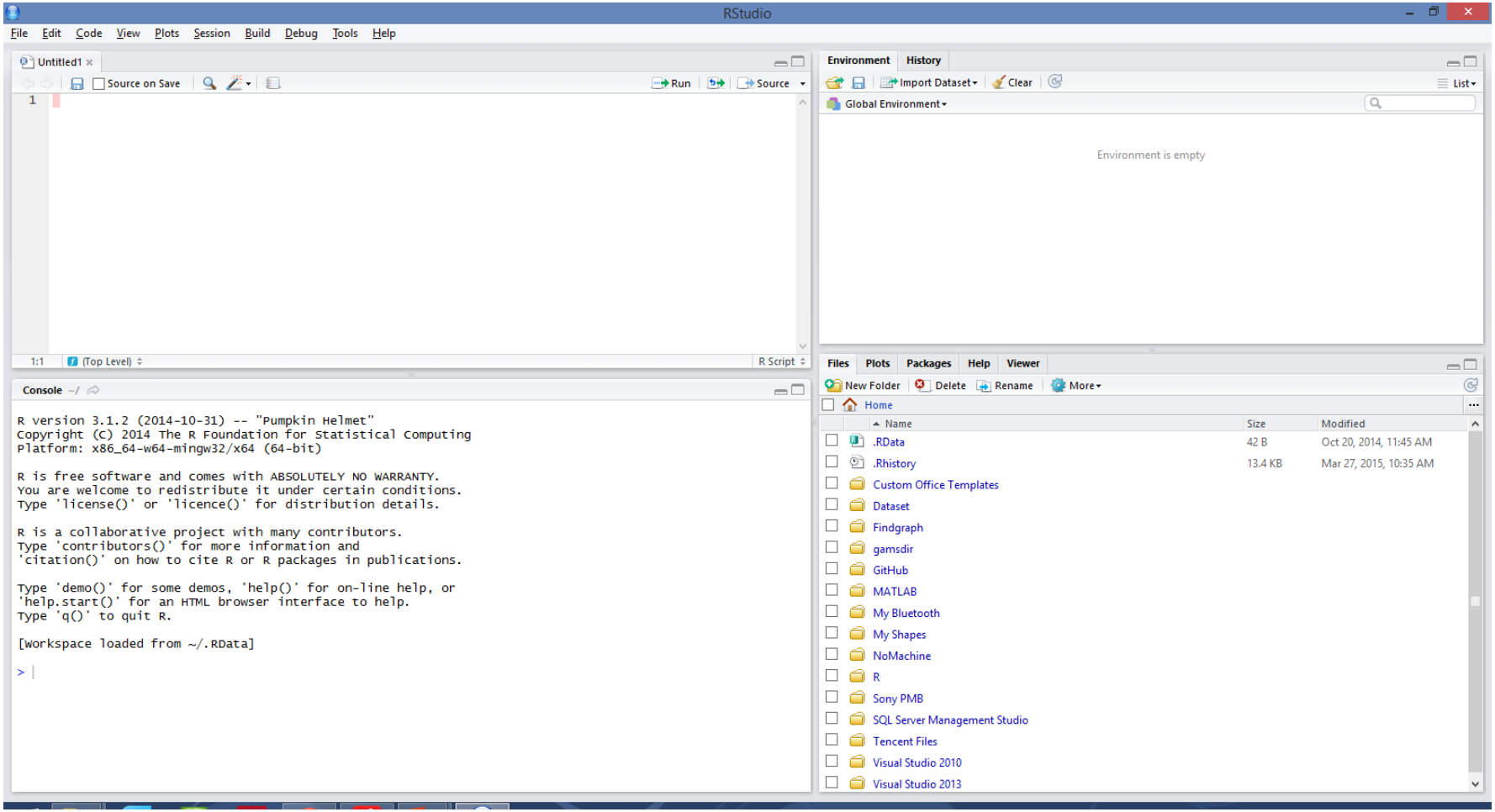
R packages are free!
Get from

Rdocumentation

Outline for today's session

- Overview of R
- Setting up R and basic operations
- Preparing summary statistics
- Basic plotting function

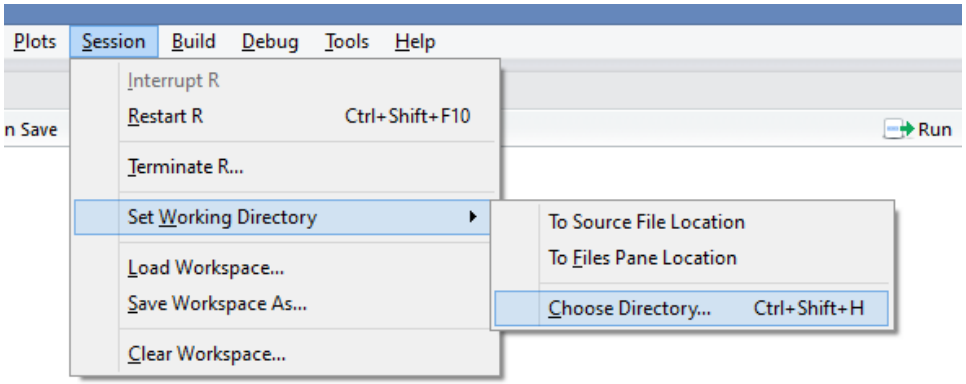
Set up RStudio



Set working directory

This is where you want to save all work.

1. In Rstudio, 'session' -> 'Set Work Directory' -> 'Choose Directory …'



2. Manually specify in R code
   - In windows, `setwd("C:/Users/JIN/Desktop/INDE321")`
   - In mac or linux, `setwd("~/Desktop/INDE321")`
3. Check current working directory, `getwd()`

Some useful tricks

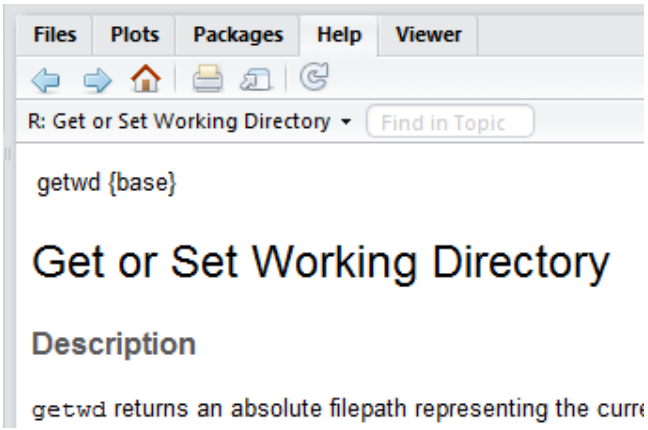
- Comment, #

```
# view the data at row 2, col 5
mydata[2,5]                          # through index
mydata[2,"hp"]                       # through name
```

- Check help document, ?name, or help(name)

  e.g.

```
help(getwd) # check help doc
?getwd      # check help doc
```



R data structures

- Vector: one dimension with same type data
- Matrix: two dimensions with same type data
- Array: more dimensions with same type data
- Data.frame: two dimensions with various type data
- List: can be any format

```
# vector
v1 <- c(1,2,5.3,6,-2,4)                    # numeric vector
v2 <- c("one","two","three")               # character vector
v3 <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) # logical vector

# matrix
m1 <- matrix(1:20, nrow=5,ncol=4)

# arrays
a1 <- array(1:20, dim = c(2,2,5))

# data frames
d1 <- data.frame(v1, v2)

# lists
l1 <- list(v1, m1, d1)
```

- To check data type of a variable

```
# check data type
class(m1)
class(a1)
class(d1)
class(l1)
```

Import data files

- Use R package build-in dataset, e.g.

```
?mtcars
data(mtcars)
```

- Import data from your files
  - csv (.csv), dat (.dat), txt (.txt)

```
# for csv
mydata <- read.csv(file = "./mtcars.csv", header = TRUE)
```

Viewing data

Use mtcars data as example

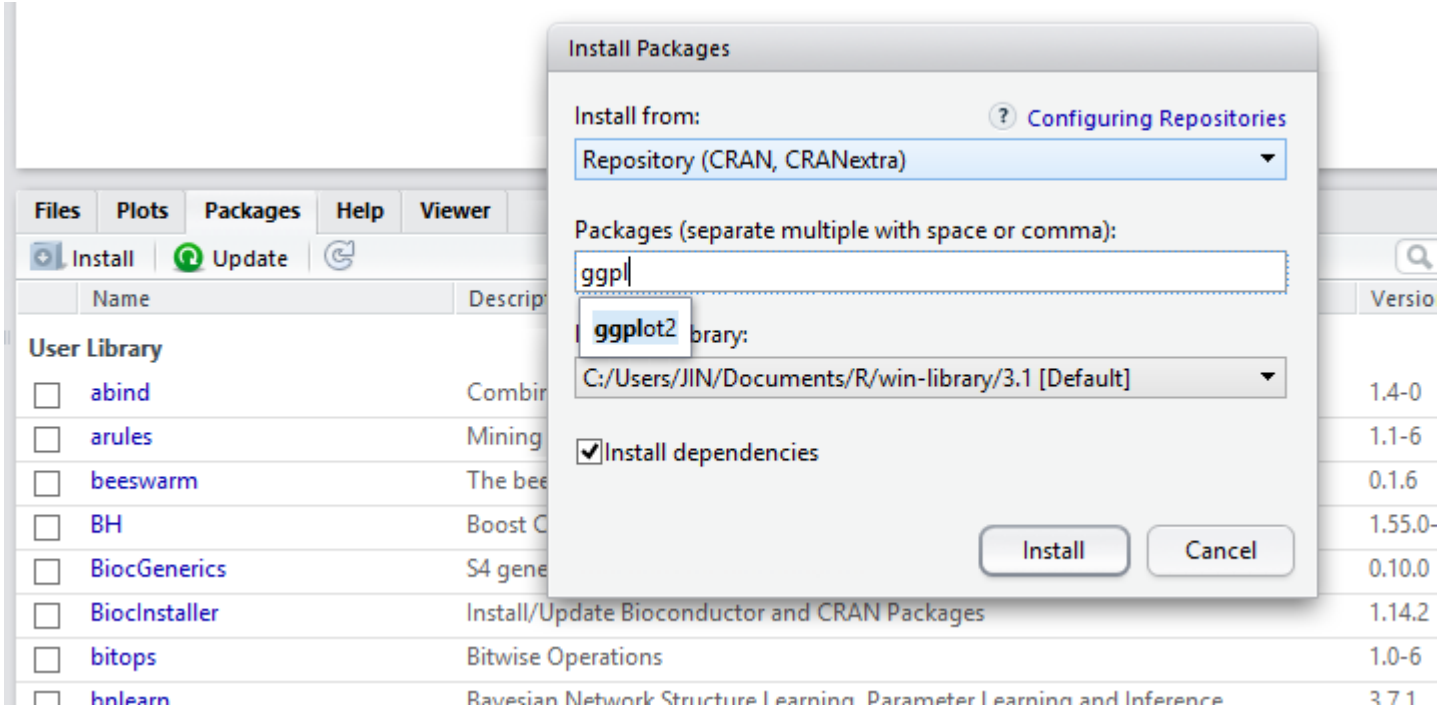
```
data(mtcars)                          # load mtcars
mydata <- mtcars                      # assign mtcars to variable mydata
mydata                                # print data
head(mydata, n = 10)                  # print first 10 rows of data
tail(mydata, n = 5)                   # print last 5 rows of data
dim(mydata)                           # dimsions of an object
class(mydata)                         # data type of an object
names(mydata)                         # list the variables' names in an object
str(mydata)                           # structure of an object
ncol(mydata)                          # number of columns
nrow(mydata)                          # number of rows
```

```
# view the data at row 2, col 5
mydata[2,5]                           # through index
mydata[2,"hp"]                        # through name

# view the data at col 5
mydata[,5]
mydata$hp

# view the data at first 3 rows
mydata[1:3,]
mydata[c(1,2,3),]
```

Install R packages

R is a lightweight statistical programming language, which is different from matlab, SAS and SPSS. It means it comes out with a few basic packages, and you have to install R packages if you want to do more.

e.g. R package ggplot2 is a beautiful plot tool.



Use R packages

Two steps to use R packages after installation.

1. Load the R package

```
library(ggplot2)                    # load package
require(ggplot2)                    # another way to load package
```

2. Call function implemented in the R package

```
ggplot(data = mtcars, aes(x = as.factor(gear), y = mpg, fill = as.factor(gear))) +
  geom_boxplot() +
  xlab("") + ylab("Miles per Gallon") +
  ggtitle("Miles by Gear Number")
```

Exercise

1. Install R package ‘qcc’
2. Load ‘qcc’
3. Check the help document of built-in dataset, ‘pistonrings’
4. Load data pistonrings
5. Show the data type of pistonrings
6. Show the dimension of pistonrings
7. Show the variable names of pistonrings
8. Show the first 8 rows of pistonrings
9. What is the value of row 5, col 2 in pistonrings
10. *Challenge: what is the mean value of diameter in pistonrings*

Outline for today's session

- Overview of R
- Setting up R and basic operations
- Preparing summary statistics
- Basic plotting function

Basic statistical results

- Mean
- Variance
- Standard deviation (how to validate sd with variance)
- Minimum, maximum
- Median
- Quantile

```
mydata <- pistonrings
mean(mydata$diameter)
var(mydata$diameter)
sd(mydata$diameter)                    # check square root of var = sd?
sqrt(var(mydata$diameter)) == sd(mydata$diameter)
min(mydata$diameter)
max(mydata$diameter)
median(mydata$diameter)
quantile(mydata$diameter, 1/2)         # is this equal to median?
quantile(mydata$diameter, 3/4)
quantile(mydata$diameter, 1/4)
IQR(mydata$diameter)                   # quantile(x, 3/4) - quantile(x, 1/4)?
```

Function 'summary()'

- Gives a collection of basic statistics
- Can be used with many functions include model fitting functions (ANOVA, regression model, cluster analysis)

```
summary(mydata$diameter)
```

```
> summary(mydata$diameter)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  73.97   74.00   74.00   74.00   74.01   74.04
```

Function 'table()'

You can generate frequency tables using the table( ) function

```
> table(mtcars$cyl)

 4  6  8
11  7 14
```

```
> table(mtcars[,c("cyl", "hp")])
   hp
cyl 52 62 65 66 91 93 95 97 105 109 110 113 123 150 175 180 205 215 230 245 264 335
  4  1  1  1  2  1  1  1  1   0   1   0   1   0   0   0   0   0   0   0   0   0   0
  6  0  0  0  0  0  0  0  0   1   0   3   0   2   0   1   0   0   0   0   0   0   0
  8  0  0  0  0  0  0  0  0   0   0   0   0   0   2   2   3   1   1   1   2   1   1
```

Outline for today's session

- Overview of R
- Setting up R and basic operations
- Preparing summary statistics
- Basic plotting function

Scatter plot

aes(x variable, y variable, others)

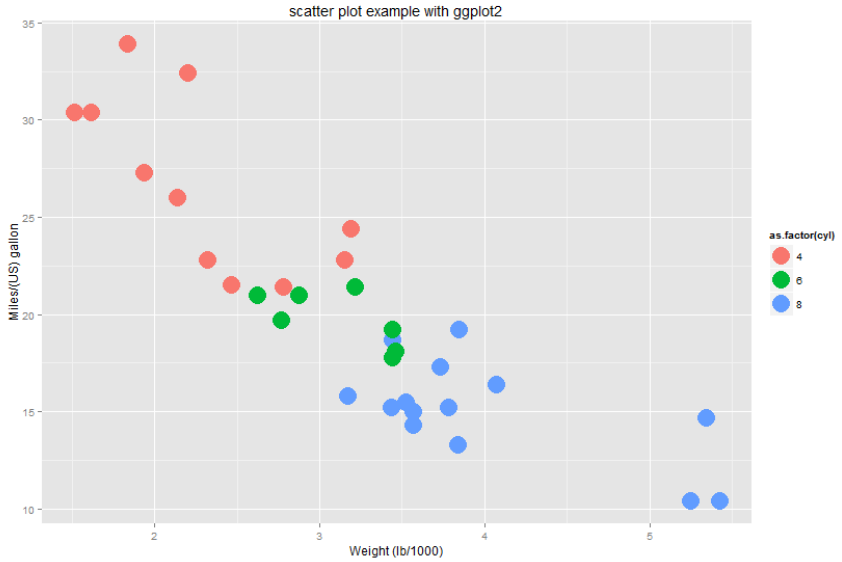
Call ggplot(data_name, …)

as.factor() convert to discrete var

```
# scatter plot
ggplot(data = mydata, aes(x = wt, y = mpg, color = as.factor(cyl))) +
  geom_point(size = 8) +
  xlab("Weight (lb/1000)") + ylab("Miles/(US) gallon") +
  ggtitle("scatter plot example with ggplot2")
```

geom_XXX, layout, e.g.
geom_scatter()
geom_line()
geom_boxplot()
… …

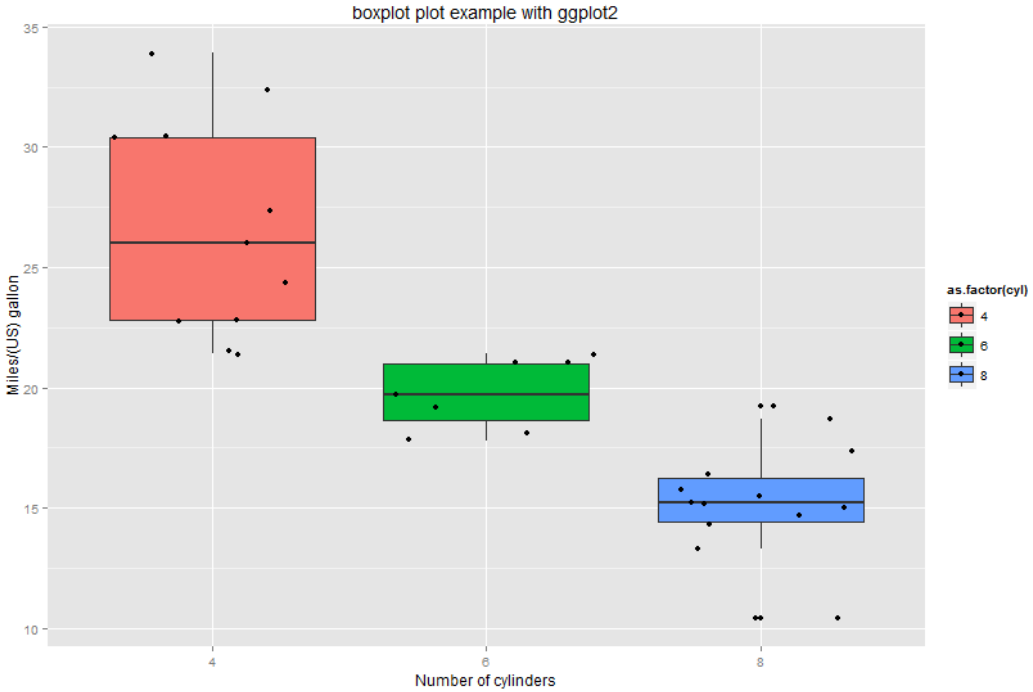
xlab: x axis title
ylab: y axis title
ggtitle: main title



Boxplot plot

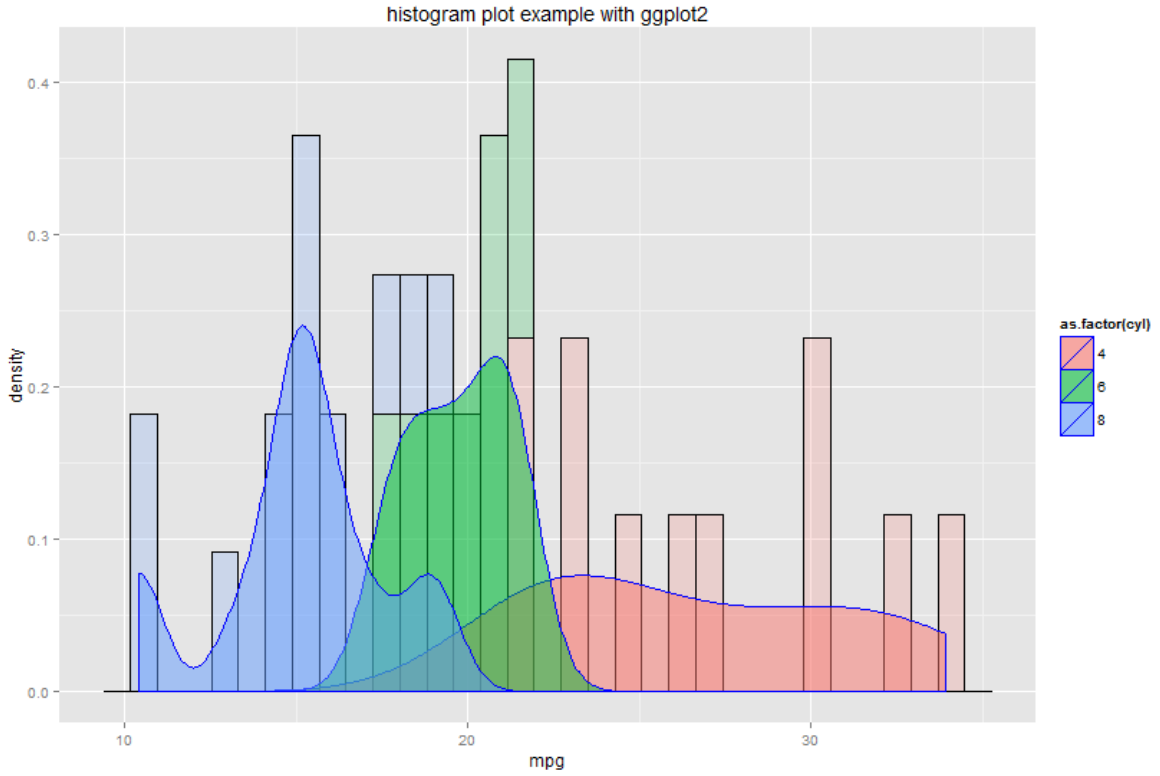
Second layout, geom_jitter(), plot points

```
# boxplot
ggplot(data = mtcars, aes(x = as.factor(gear), y = mpg, fill = as.factor(gear))) +
  geom_boxplot() +
  geom_jitter() +
  xlab("Number of cylinders") + ylab("Miles/(US) gallon") +
  ggtitle("boxplot plot example with ggplot2")
```



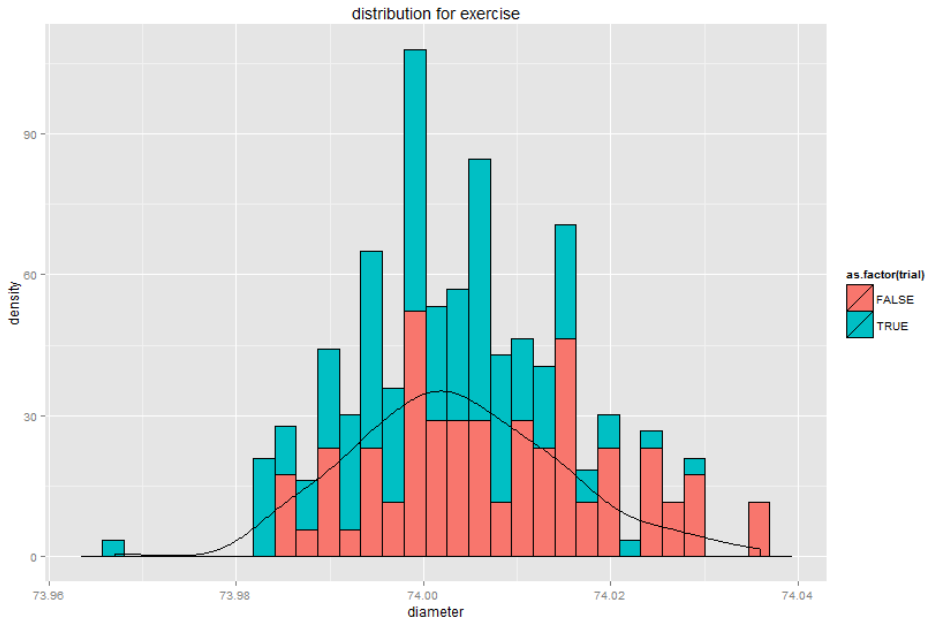
Histogram plot

```
# histogram
ggplot(data = mydata, aes(mpg, fill = as.factor(cyl))) +
  geom_histogram(aes(y = ..density..), color = "black", alpha = 0.2) +
  geom_density(color = "blue", alpha = 0.5) +
  ggtitle("histogram plot example with ggplot2")
```



Exercise

- Plot the histogram and density for variable ‘diameter’ in dataset ‘pistonrings’ in R package ‘qcc’
- Add variable ‘trial’ to show the difference between two groups
- Add main title
- More …



Exercise

- Repeat the application of the above codes on another dataset, data(iris)
- ggplot is nice but sometimes hard to use. In R, there is always more ways to do things. For example, review the examples in http://www.statmethods.net/graphs/boxplot.html and draw the similar figures.

To learn more …

- http://pages.pomona.edu/~jsh04747/courses/RTutorial.pdf
- https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf