Lecture 3: Introduction to R Part II

Pilsung Kang
School of Industrial Management Engineering
Korea University

AGENDA

01 Conditions and Loops

02 Functions

03 R Graphics

04 Documentation

Conditions and Loops

- Understanding conditions and loops are necessary for efficient data analysis
  - ✓ Example of conditions
    - ▪ Want to remove instances whose value is greater than 3 standard deviations
    - ▪ Want to remove variables with zero variance
    - ▪ Want to replace NULL with a constant value
  - ✓ Example of loops
    - ▪ Want to make a histogram for each variable in a dataframe
    - ▪ Want to compare various machine learning algorithms for the same dataset

Conditions

- if-else condition

  if (condition) {

  statement 1

  } else {

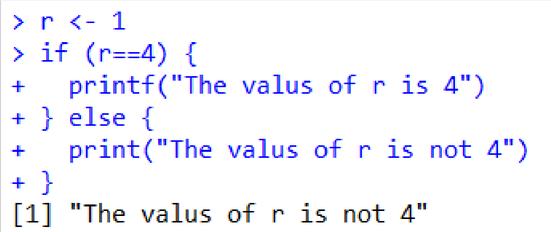
  statement 2

  }

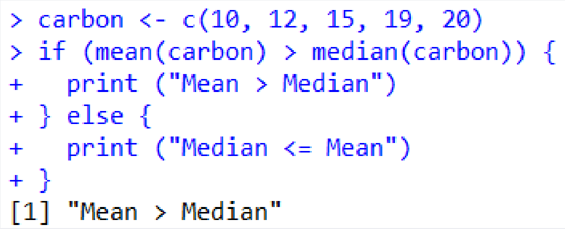
  ✓ condition can be a simple logical comparison to a complex function

  ✓ statement 1: run if the condition is met

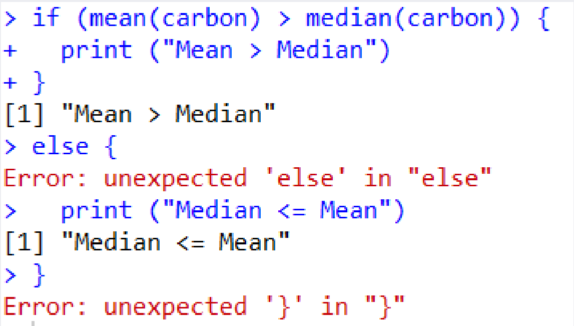
  ✓ statement 2: run if the condition is not met



```
> r <- 1
> if (r==4) {
+   printf("The valus of r is 4")
+ } else {
+   print("The valus of r is not 4")
+ }
[1] "The valus of r is not 4"
```



```
> carbon <- c(10, 12, 15, 19, 20)
> if (mean(carbon) > median(carbon)) {
+   print ("Mean > Median")
+ } else {
+   print ("Median <= Mean")
+ }
[1] "Mean > Median"
```



```
> if (mean(carbon) > median(carbon)) {
+   print ("Mean > Median")
+ }
[1] "Mean > Median"
> else {
Error: unexpected 'else' in "else"
>   print ("Median <= Mean")
[1] "Median <= Mean"
> }
Error: unexpected '}' in "}"
```

Conditions

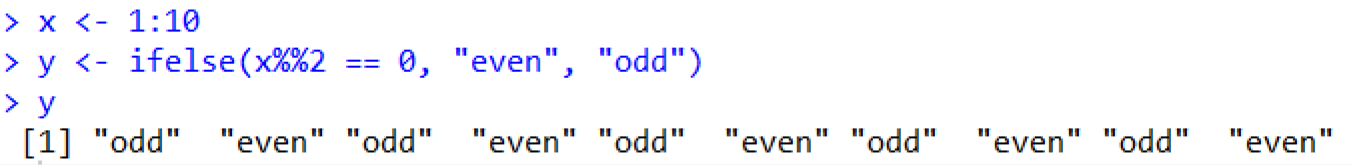
- ifelse: a vectorized condition

  ifelse (condition, statement 1, statement 2)

  ✓ condition: Boolean vector

  ✓ statement 1: run if the condition is met

  ✓ statement 2: run if the condition is not met



```
> x <- 1:10
> y <- ifelse(x%%2 == 0, "even", "odd")
> y
 [1] "odd"  "even" "odd"  "even" "odd"  "even" "odd"  "even" "odd"  "even"
```

Loops

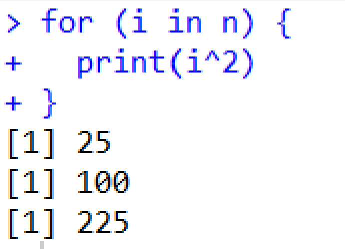
- for loop

```
for (i in x) {
statement
}
```

✓ i: index of loop

✓ x: a set of element for which the loop runs

✓ statement: running part



```
> for (i in n) {
+   print(i^2)
+ }
[1] 25
[1] 100
[1] 225
```

Loop

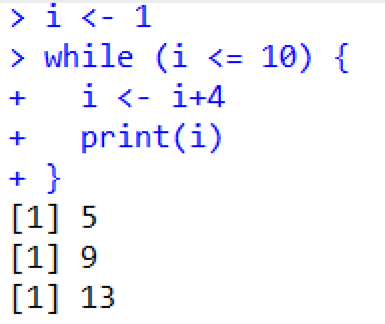
- while loop

  while (condition) {

  statement

  }

  ✓ run the statement until the condition is not met



```
> i <- 1
> while (i <= 10) {
+   i <- i+4
+   print(i)
+ }
[1] 5
[1] 9
[1] 13
```

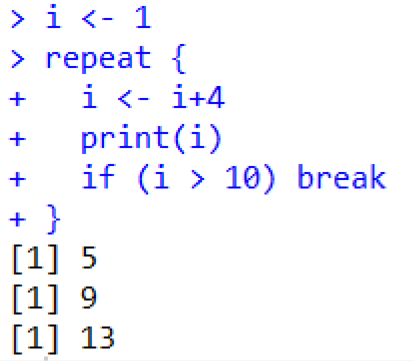
- repeat-break loop

  repeat {

  statement

  condition break

  }



```
> i <- 1
> repeat {
+   i <- i+4
+   print(i)
+   if (i > 10) break
+ }
[1] 5
[1] 9
[1] 13
```

  ✓ run the statement first, check the condition, stop if the condition is met

AGENDA

01 Conditions and Loops

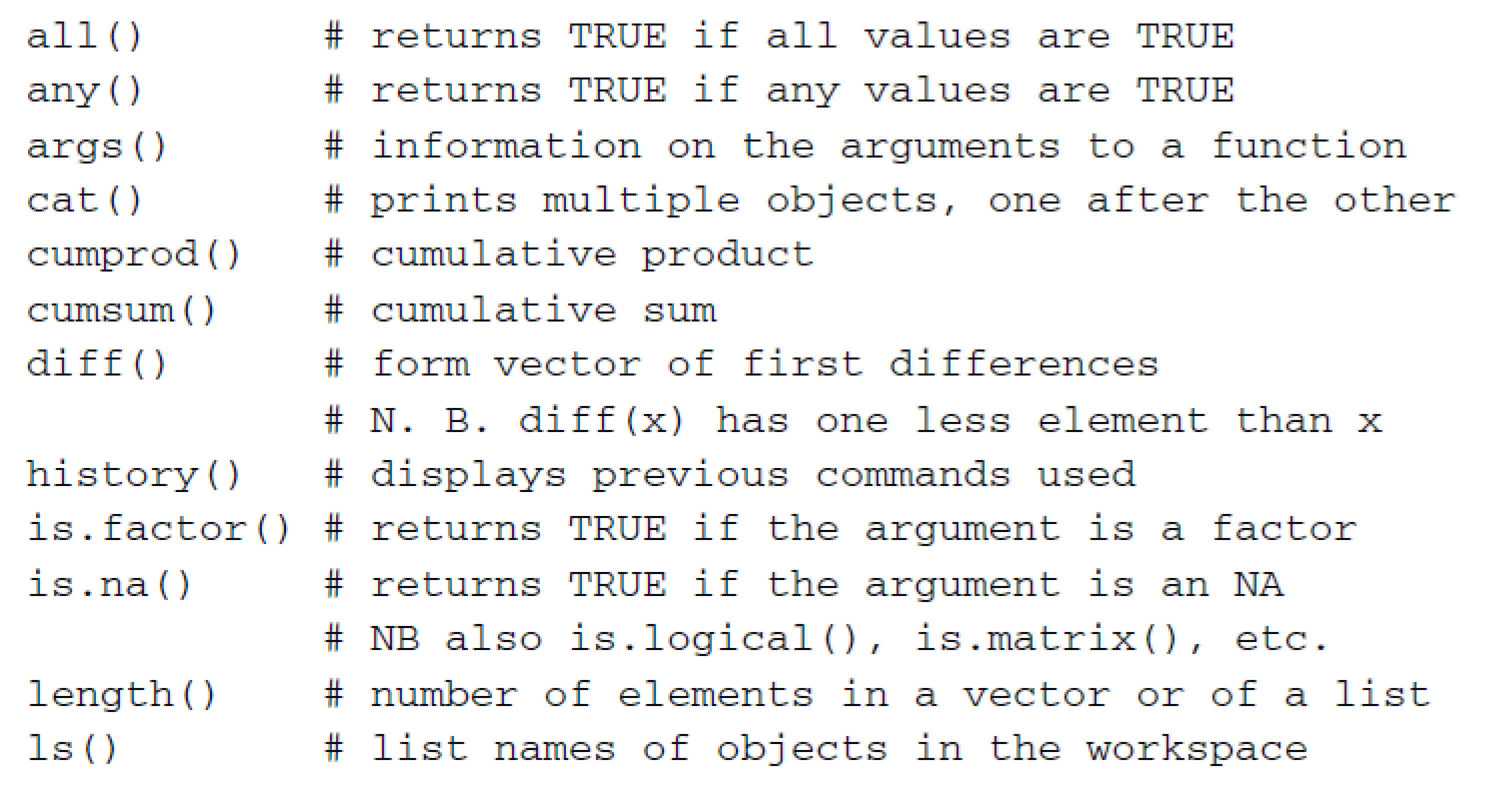
02 Functions

03 R Graphics

04 Documentation

Function

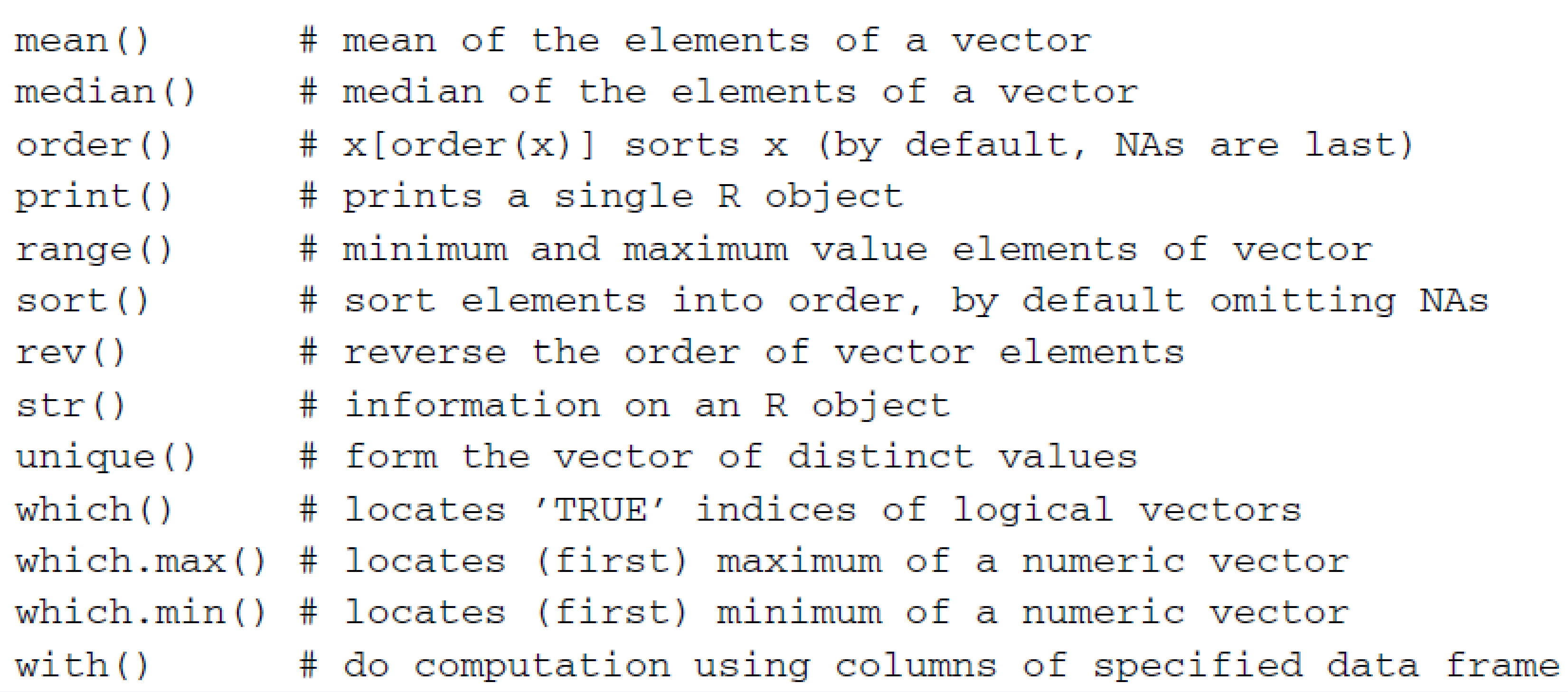
- Why functions?
- An incidental advantage of putting code into functions is that the workspace is not then cluttered with objects that are local to the function



```
all()        # returns TRUE if all values are TRUE
any()        # returns TRUE if any values are TRUE
args()       # information on the arguments to a function
cat()        # prints multiple objects, one after the other
cumprod()    # cumulative product
cumsum()     # cumulative sum
diff()       # form vector of first differences
             # N. B. diff(x) has one less element than x
history()    # displays previous commands used
is.factor()  # returns TRUE if the argument is a factor
is.na()      # returns TRUE if the argument is an NA
             # NB also is.logical(), is.matrix(), etc.
length()     # number of elements in a vector or of a list
ls()         # list names of objects in the workspace
```

Function

- Why functions?
- An incidental advantage of putting code into functions is that the workspace is not then cluttered with objects that are local to the function



```
mean()       # mean of the elements of a vector
median()     # median of the elements of a vector
order()      # x[order(x)] sorts x (by default, NAs are last)
print()      # prints a single R object
range()      # minimum and maximum value elements of vector
sort()       # sort elements into order, by default omitting NAs
rev()        # reverse the order of vector elements
str()        # information on an R object
unique()     # form the vector of distinct values
which()      # locates 'TRUE' indices of logical vectors
which.max()  # locates (first) maximum of a numeric vector
which.min()  # locates (first) minimum of a numeric vector
with()       # do computation using columns of specified data frame
```

Function

- Writing a function

```
function_name <- function(arguments) {
statement 1
statement 2
…
return(object)
}
```

✓ function_name: name that the function is referred to

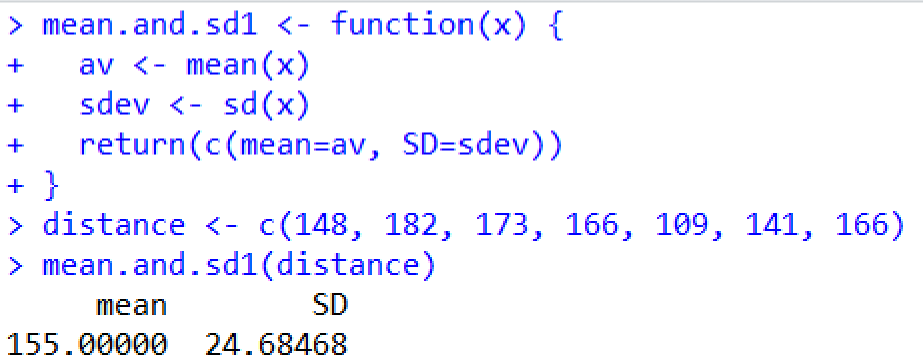
✓ arguments: inputs that a user should provide to run the function

✓ statements: operations running inside the function

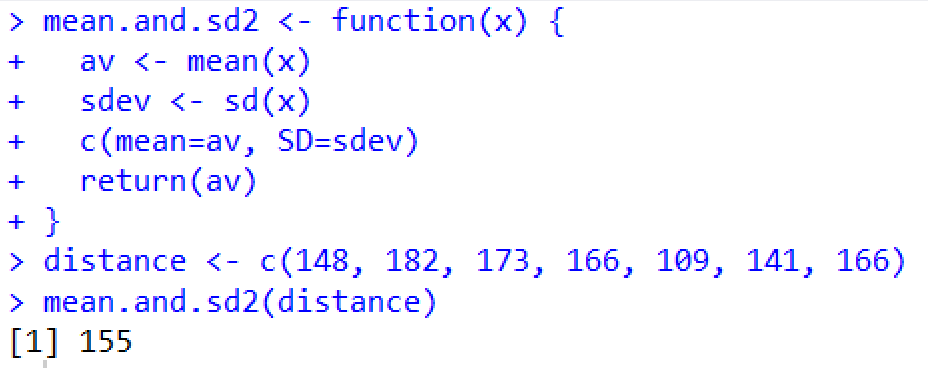
✓ object: function output

Function

- Same operations but different outputs



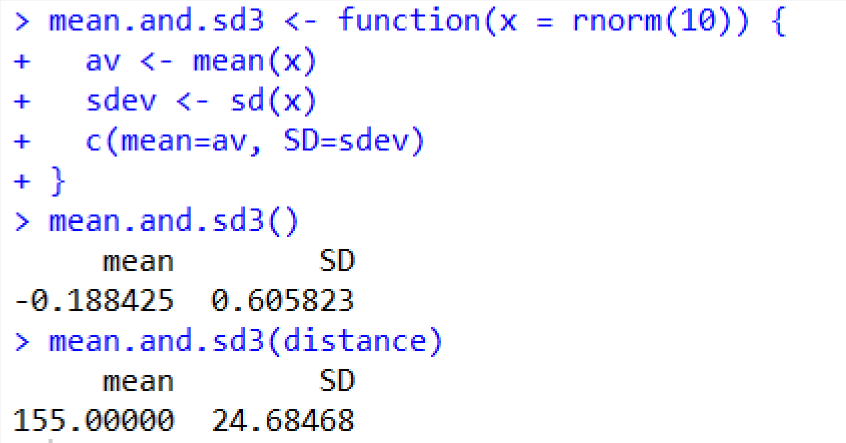
```
> mean.and.sd1 <- function(x) {
+   av <- mean(x)
+   sdev <- sd(x)
+   return(c(mean=av, SD=sdev))
+ }
> distance <- c(148, 182, 173, 166, 109, 141, 166)
> mean.and.sd1(distance)
     mean        SD
155.00000  24.68468
```



```
> mean.and.sd2 <- function(x) {
+   av <- mean(x)
+   sdev <- sd(x)
+   c(mean=av, SD=sdev)
+   return(av)
+ }
> distance <- c(148, 182, 173, 166, 109, 141, 166)
> mean.and.sd2(distance)
[1] 155
```

Function

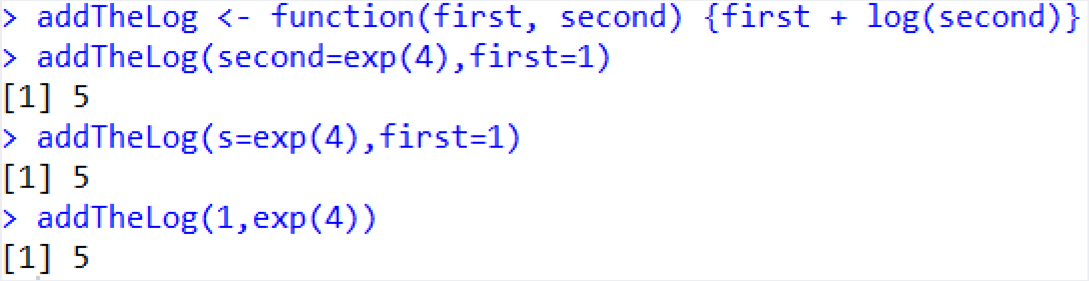
- Writing a function
  - ✓ Functions can accept various types of input arguments
  - ✓ It is possible to provide default arguments
    - ▪ Default arguments are used if a user calls a function but does not provide the required input arguments



```
> mean.and.sd3 <- function(x = rnorm(10)) {
+   av <- mean(x)
+   sdev <- sd(x)
+   c(mean=av, SD=sdev)
+ }
> mean.and.sd3()
     mean        SD
-0.188425  0.605823
> mean.and.sd3(distance)
     mean        SD
155.00000  24.68468
```

Function

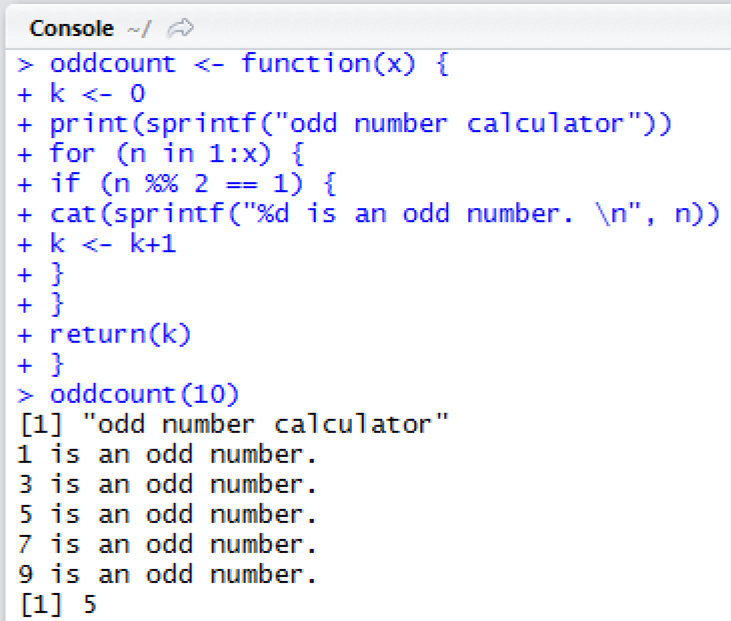
- Function arguments
  - ✓ Each argument has its own name
  - ✓ Name is used to access the corresponding argument within function
  - ✓ Three possible ways to assign the argument
    - Exact name
    - Partially matching names (not recommended)
    - Argument order



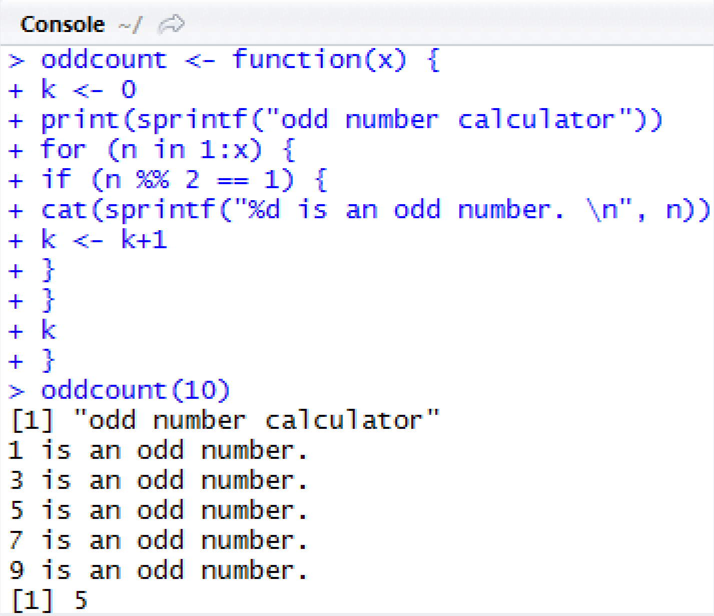
```
> addTheLog <- function(first, second) {first + log(second)}
> addTheLog(second=exp(4),first=1)
[1] 5
> addTheLog(s=exp(4),first=1)
[1] 5
> addTheLog(1,exp(4))
[1] 5
```

Function

- Return object in function
  - ✓ All R object can be the return object
  - ✓ Use return( ) function
  - ✓ If return( ) is not used, the result of the last operation is returned (not recommended)



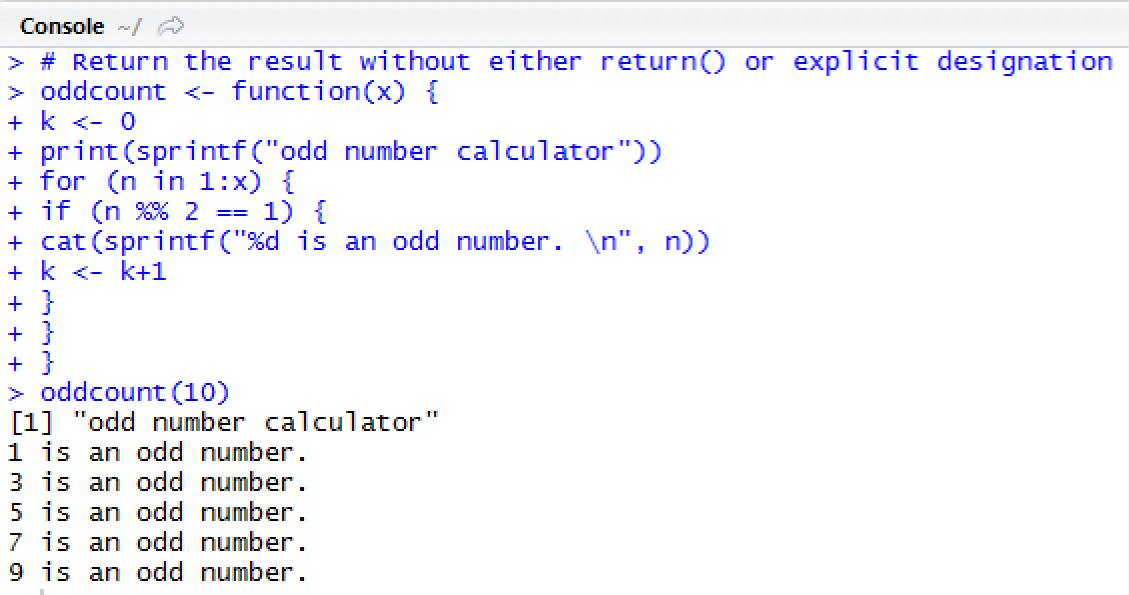
```
> oddcount <- function(x) {
+ k <- 0
+ print(sprintf("odd number calculator"))
+ for (n in 1:x) {
+ if (n %% 2 == 1) {
+ cat(sprintf("%d is an odd number. \n", n))
+ k <- k+1
+ }
+ }
+ return(k)
+ }
> oddcount(10)
[1] "odd number calculator"
1 is an odd number.
3 is an odd number.
5 is an odd number.
7 is an odd number.
9 is an odd number.
[1] 5
```



```
> oddcount <- function(x) {
+ k <- 0
+ print(sprintf("odd number calculator"))
+ for (n in 1:x) {
+ if (n %% 2 == 1) {
+ cat(sprintf("%d is an odd number. \n", n))
+ k <- k+1
+ }
+ }
+ k
+ }
> oddcount(10)
[1] "odd number calculator"
1 is an odd number.
3 is an odd number.
5 is an odd number.
7 is an odd number.
9 is an odd number.
[1] 5
```

Function

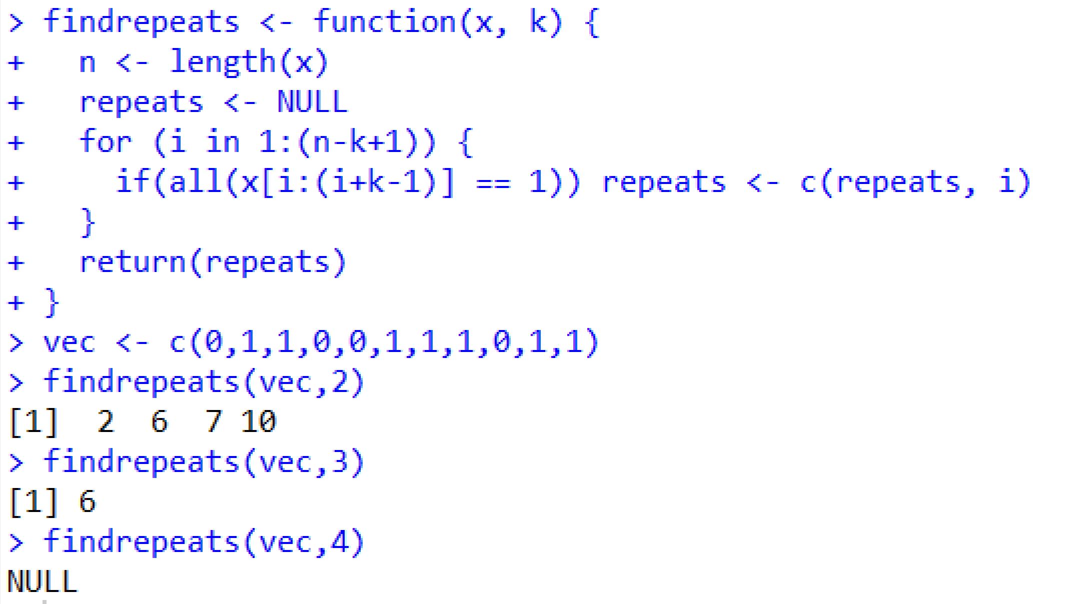
- Return object in function
  - ✓ All R object can be the return object
  - ✓ Use return( ) function
  - ✓ If return( ) is not used, the result of the last operation is returned (not recommended)



```
> # Return the result without either return() or explicit designation
> oddcount <- function(x) {
+ k <- 0
+ print(sprintf("odd number calculator"))
+ for (n in 1:x) {
+ if (n %% 2 == 1) {
+ cat(sprintf("%d is an odd number. \n", n))
+ k <- k+1
+ }
+ }
+ }
> oddcount(10)
[1] "odd number calculator"
1 is an odd number.
3 is an odd number.
5 is an odd number.
7 is an odd number.
9 is an odd number.
```

Function

- Function example 1
  - ✓ Question: from a vector consisting of only 0 and 1, return the indices from which 1 repeatedly appears k times
    - ▪ Ex: (1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1)
      - k=2: answer = (1, 2, 8,11,12)
      - k=3: answer = (1,11)
      - k=4: answer = NULL



```
> findrepeats <- function(x, k) {
+   n <- length(x)
+   repeats <- NULL
+   for (i in 1:(n-k+1)) {
+     if(all(x[i:(i+k-1)] == 1)) repeats <- c(repeats, i)
+   }
+   return(repeats)
+ }
> vec <- c(0,1,1,0,0,1,1,1,0,1,1)
> findrepeats(vec,2)
[1]  2  6  7 10
> findrepeats(vec,3)
[1] 6
> findrepeats(vec,4)
NULL
```

Function

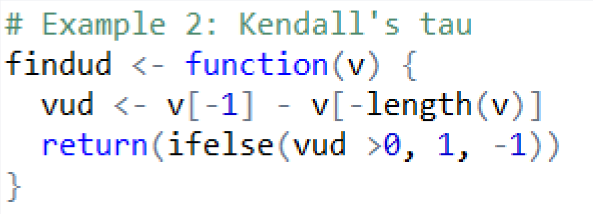
- Function example 2: Kendall’s tau

✓ Raw data: temperature and pressure recorded every hour

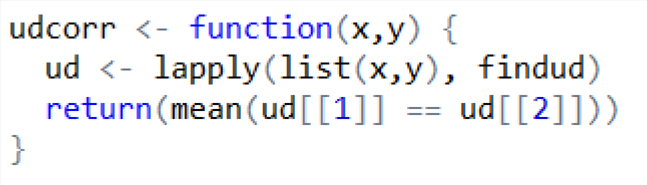
Time	10:00	11:00	12:00	13:00	14:00
Temperature	10	15	13	17	20
Pressure	900	920	890	940	920

✓ What to do

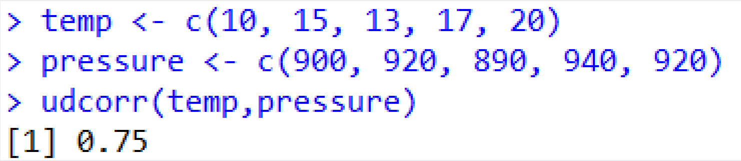
- Determine whether each indicator increases or decreases
- Return the proportion of the events in which the change directions of the two indicators are the same



```
# Example 2: Kendall's tau
findud <- function(v) {
  vud <- v[-1] - v[-length(v)]
  return(ifelse(vud >0, 1, -1))
}
```



```
udcorr <- function(x,y) {
  ud <- lapply(list(x,y), findud)
  return(mean(ud[[1]] == ud[[2]]))
}
```



```
> temp <- c(10, 15, 13, 17, 20)
> pressure <- c(900, 920, 890, 940, 920)
> udcorr(temp,pressure)
[1] 0.75
```

AGENDA

01 Conditions and Loops

02 Functions

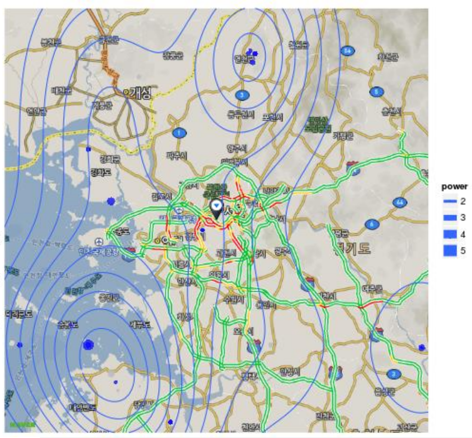
03 R Graphics

04 Documentation

R Graphics

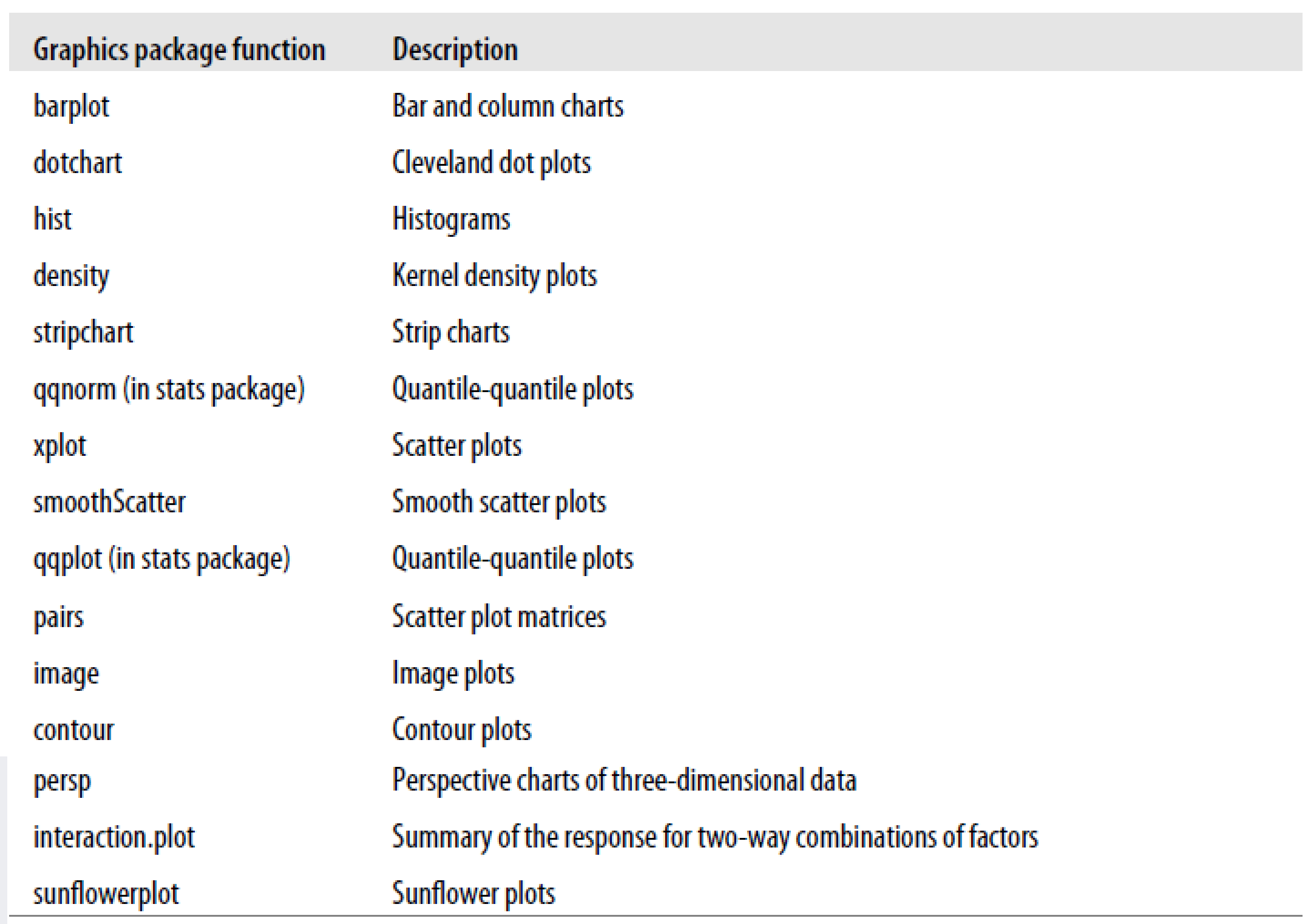
- What we can do with R graphics

London Cycle Hire Journeys
Thicker, yellower lines mean more journeys
Data: 3.2 Million Journeys (from TfL)
Routing: Ollie O'Brien (@oobr) + OpenStreetMap cc-by-sa
Buildings: OS Opendata Crown Copyright 2011
Map: James Cheshire (@spatialanalysis)



R Graphics

- Basic functions that are provided by “graphics” package



Graphics package function	Description
barplot	Bar and column charts
dotchart	Cleveland dot plots
hist	Histograms
density	Kernel density plots
stripchart	Strip charts
qqnorm (in stats package)	Quantile-quantile plots
xplot	Scatter plots
smoothScatter	Smooth scatter plots
qqplot (in stats package)	Quantile-quantile plots
pairs	Scatter plot matrices
image	Image plots
contour	Contour plots
persp	Perspective charts of three-dimensional data
interaction.plot	Summary of the response for two-way combinations of factors
sunflowerplot	Sunflower plots

R Graphics

- Fisher's Iris dataset (default dataset provided by R)
  - ✓ Five variables
    - sepal length in cm, sepal width in cm, petal length in cm, petal width in cm, and
    - Species : Iris Setosa, Iris Versicolour, and Iris Virginica.

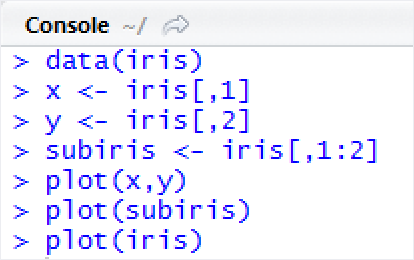
Setosa

Versicolor

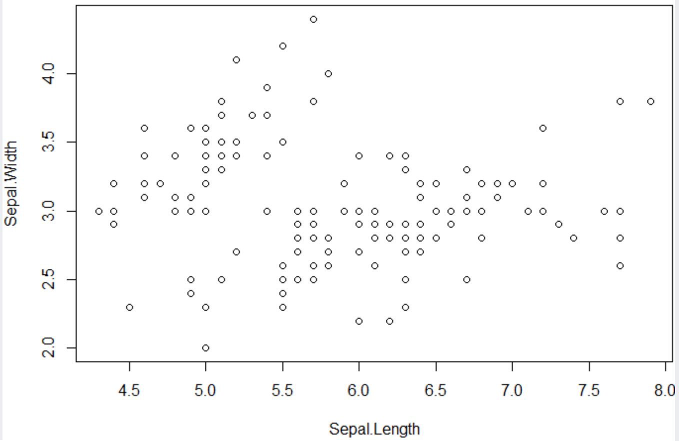
Virginica

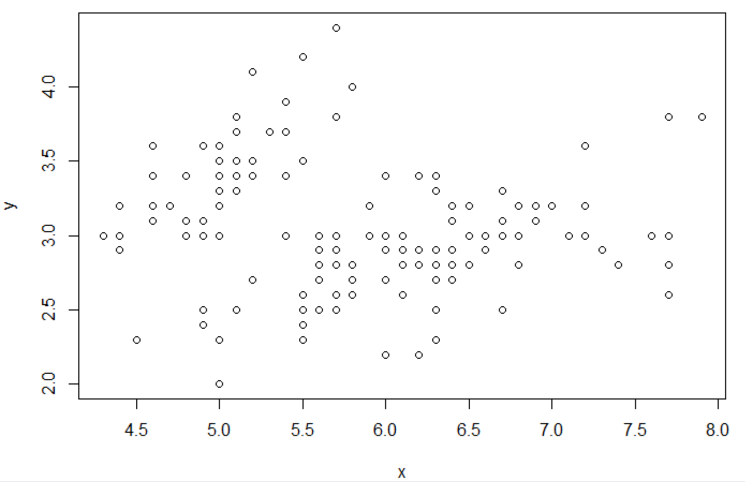
R Graphics

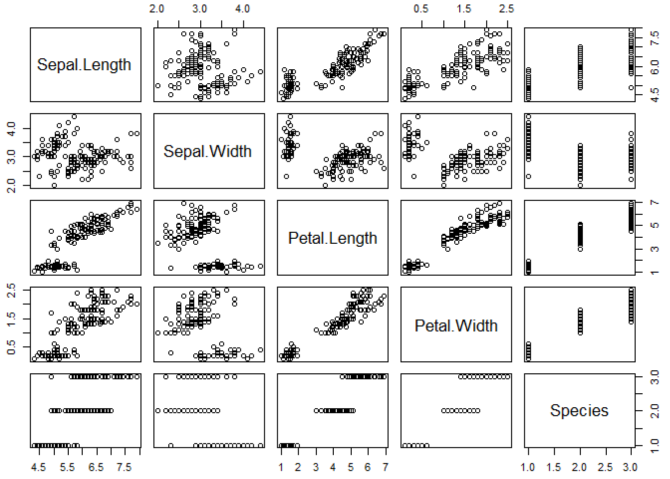
- Polymorphism of R graph functions
  - ✓ polymorphic function: has different operations for different arguments
  - ✓ ex: plot( )



```
> data(iris)
> x <- iris[,1]
> y <- iris[,2]
> subiris <- iris[,1:2]
> plot(x,y)
> plot(subiris)
> plot(iris)
```

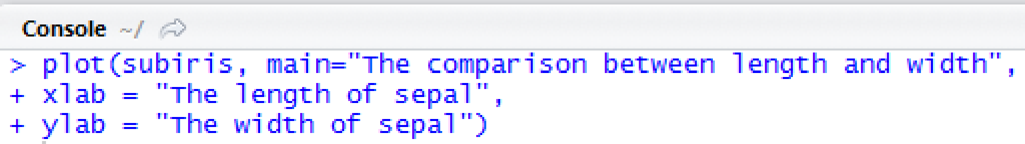




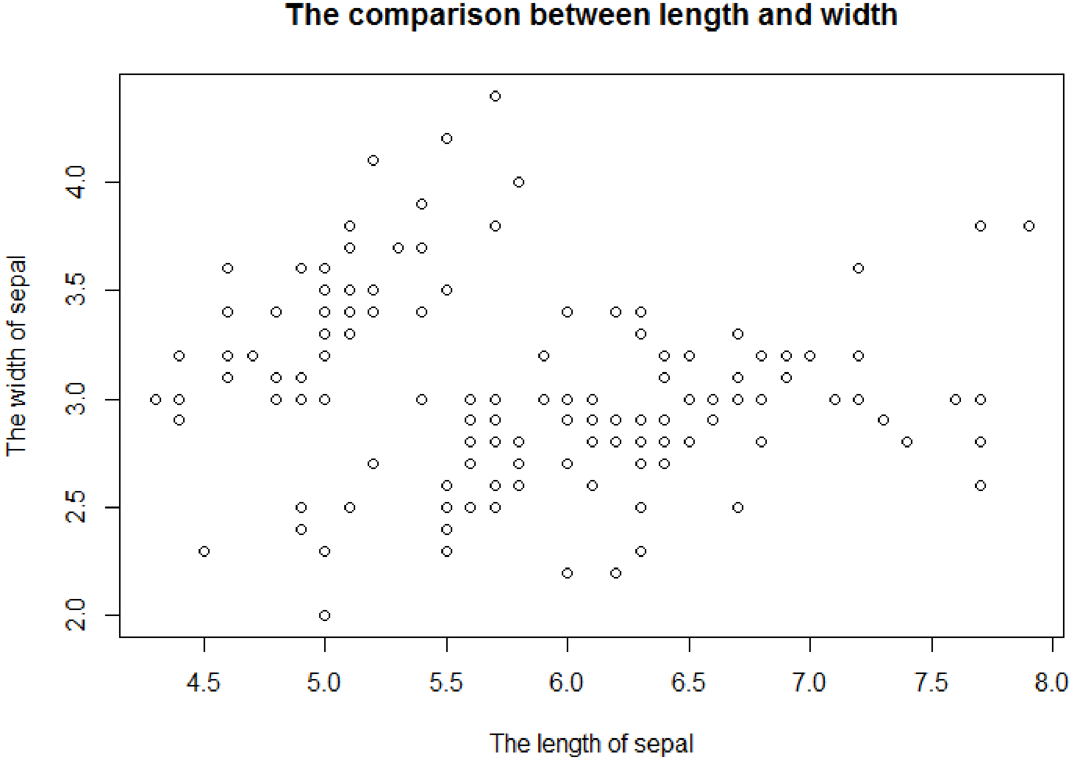


R Graphics

- Titles and labels in a graph
  - ✓ title: main, x-axis label: xlab, y-axis label: ylab

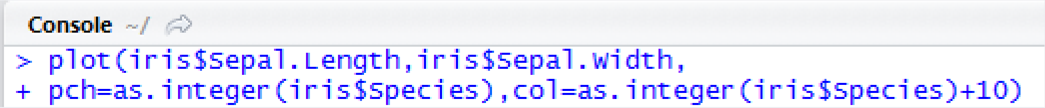


```
> plot(subiris, main="The comparison between length and width",
+ xlab = "The length of sepal",
+ ylab = "The width of sepal")
```

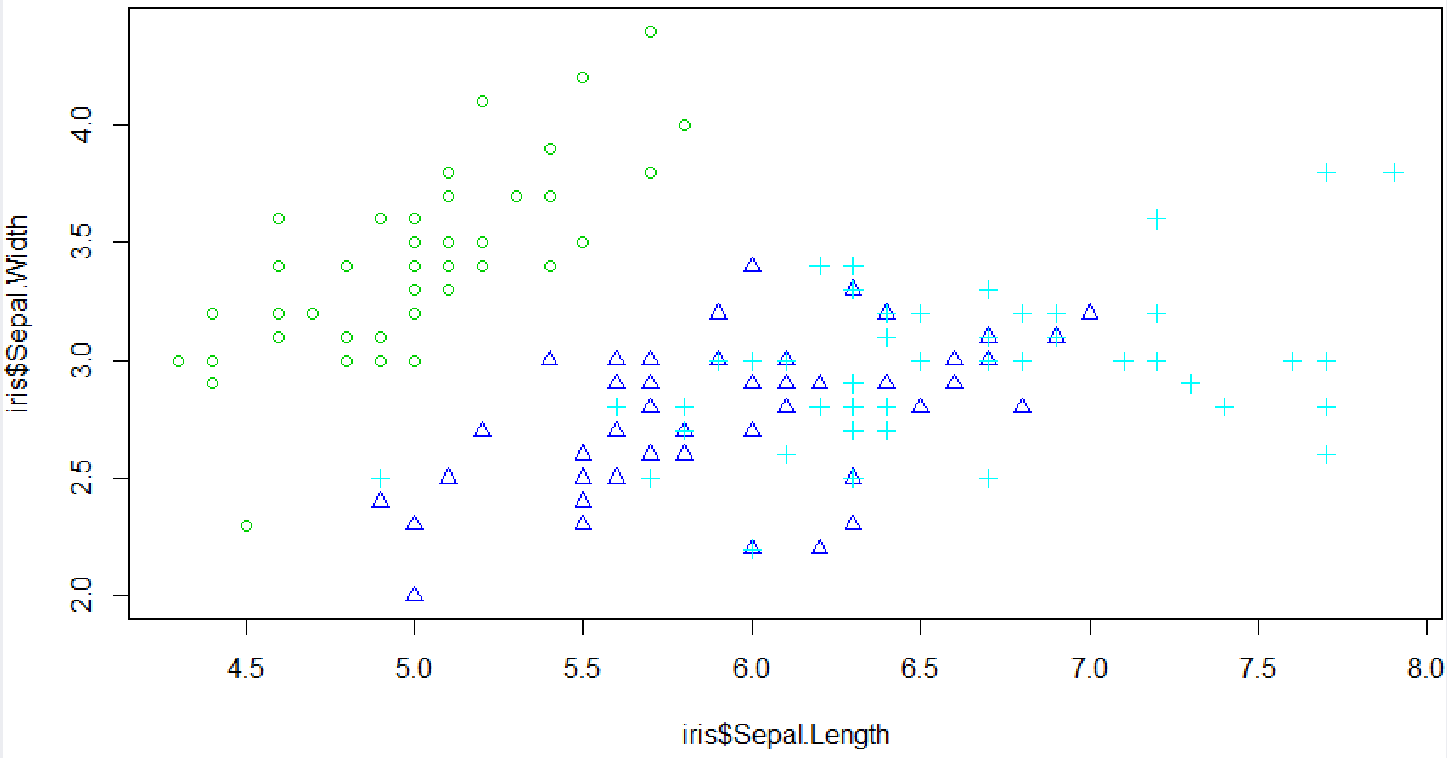


R Graphics

- Scatter plot for various categories
  - ✓ Use pch/col arguments in plot( )

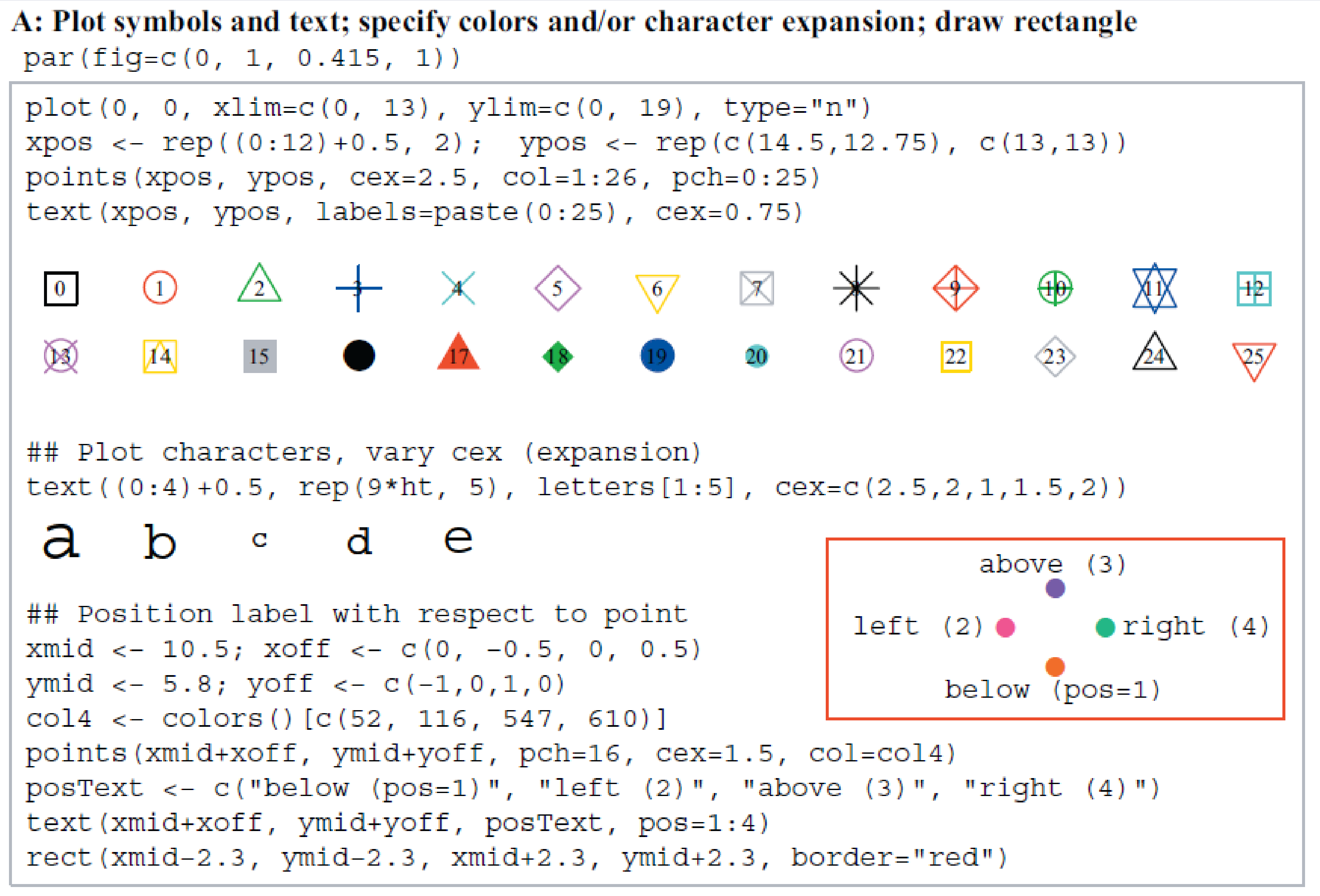


```
> plot(iris$Sepal.Length,iris$Sepal.Width,
+ pch=as.integer(iris$Species),col=as.integer(iris$Species)+10)
```



R Graphics

- Options for better readability



A: Plot symbols and text; specify colors and/or character expansion; draw rectangle

```
par(fig=c(0, 1, 0.415, 1))
```

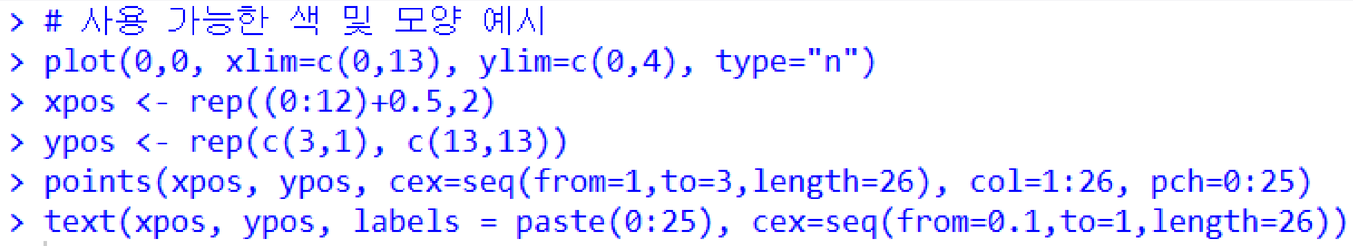
```
plot(0, 0, xlim=c(0, 13), ylim=c(0, 19), type="n")
xpos <- rep((0:12)+0.5, 2);  ypos <- rep(c(14.5,12.75), c(13,13))
points(xpos, ypos, cex=2.5, col=1:26, pch=0:25)
text(xpos, ypos, labels=paste(0:25), cex=0.75)
```

```
## Plot characters, vary cex (expansion)
text((0:4)+0.5, rep(9*ht, 5), letters[1:5], cex=c(2.5,2,1,1.5,2))
```

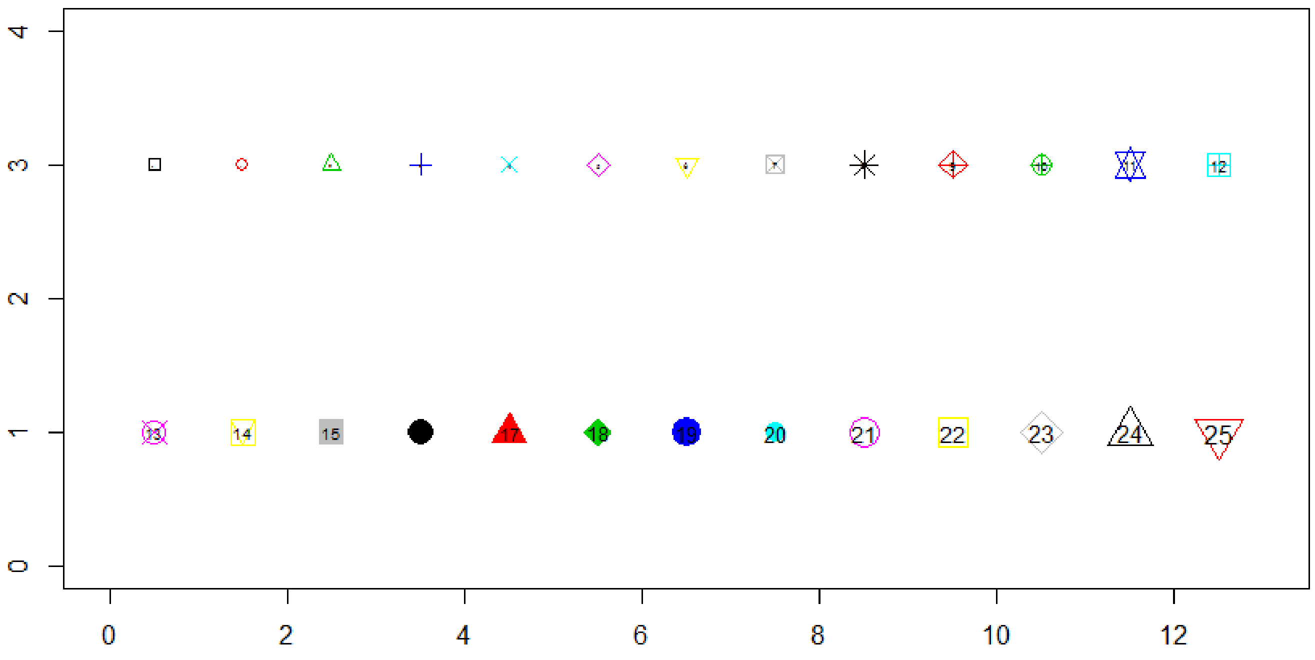
```
## Position label with respect to point
xmid <- 10.5; xoff <- c(0, -0.5, 0, 0.5)
ymid <- 5.8; yoff <- c(-1,0,1,0)
col4 <- colors()[c(52, 116, 547, 610)]
points(xmid+xoff, ymid+yoff, pch=16, cex=1.5, col=col4)
posText <- c("below (pos=1)", "left (2)", "above (3)", "right (4)")
text(xmid+xoff, ymid+yoff, posText, pos=1:4)
rect(xmid-2.3, ymid-2.3, xmid+2.3, ymid+2.3, border="red")
```

R Graphics

- Symbols in graphs
  - ✓ pch: shape, cex: size, col: color

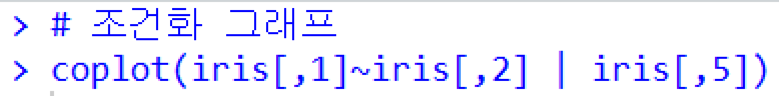


```
> # 사용 가능한 색 및 모양 예시
> plot(0,0, xlim=c(0,13), ylim=c(0,4), type="n")
> xpos <- rep((0:12)+0.5,2)
> ypos <- rep(c(3,1), c(13,13))
> points(xpos, ypos, cex=seq(from=1,to=3,length=26), col=1:26, pch=0:25)
> text(xpos, ypos, labels = paste(0:25), cex=seq(from=0.1,to=1,length=26))
```

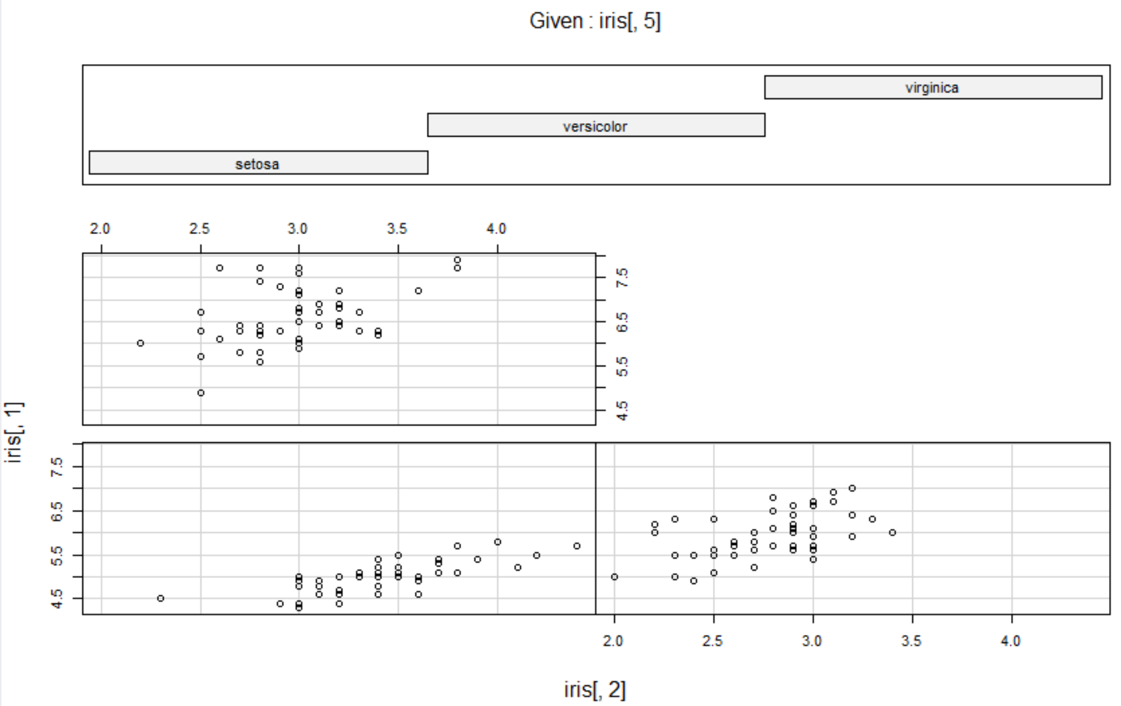


R Graphics

- Conditional plot
  - ✓ coplot(y ~ x | f)
  - ✓ For every f values, draw scatter plot for x and y

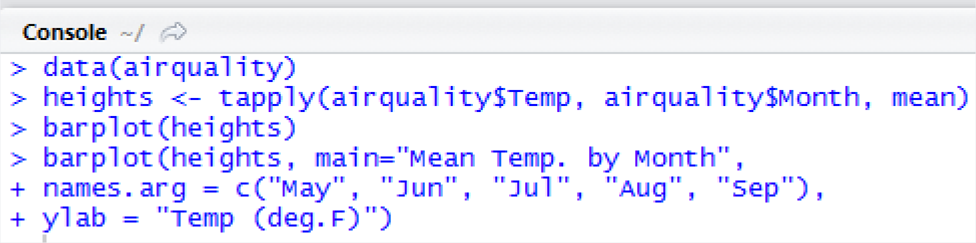


```
> # 조건화 그래프
> coplot(iris[,1]~iris[,2] | iris[,5])
```

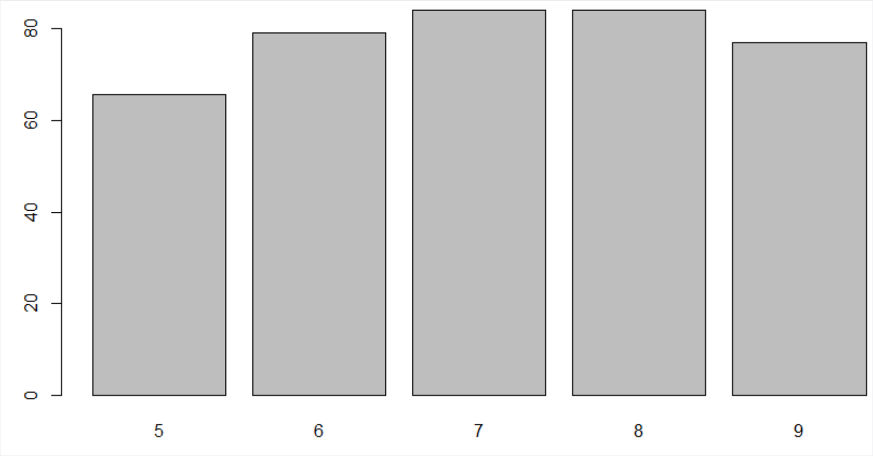


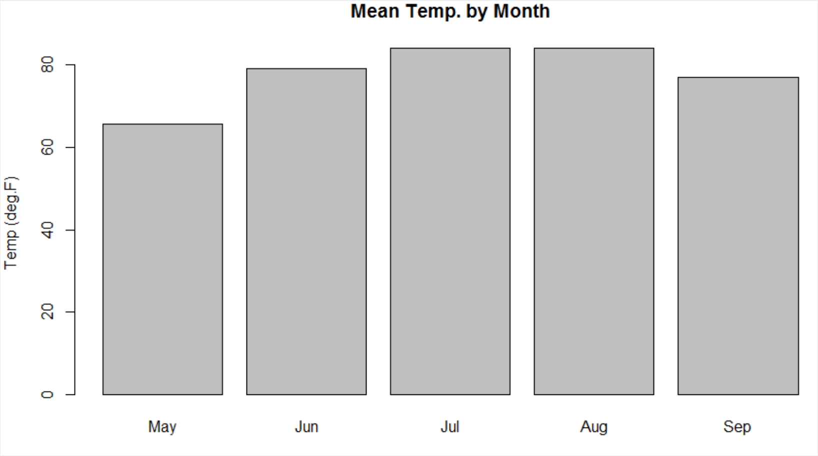
R Graphics

- Bar plots



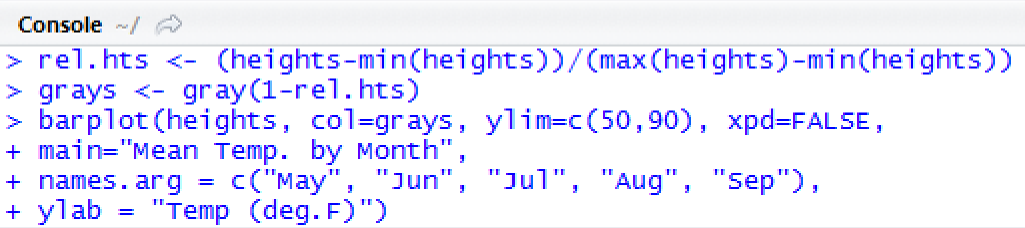
```
> data(airquality)
> heights <- tapply(airquality$Temp, airquality$Month, mean)
> barplot(heights)
> barplot(heights, main="Mean Temp. by Month",
+ names.arg = c("May", "Jun", "Jul", "Aug", "Sep"),
+ ylab = "Temp (deg.F)")
```



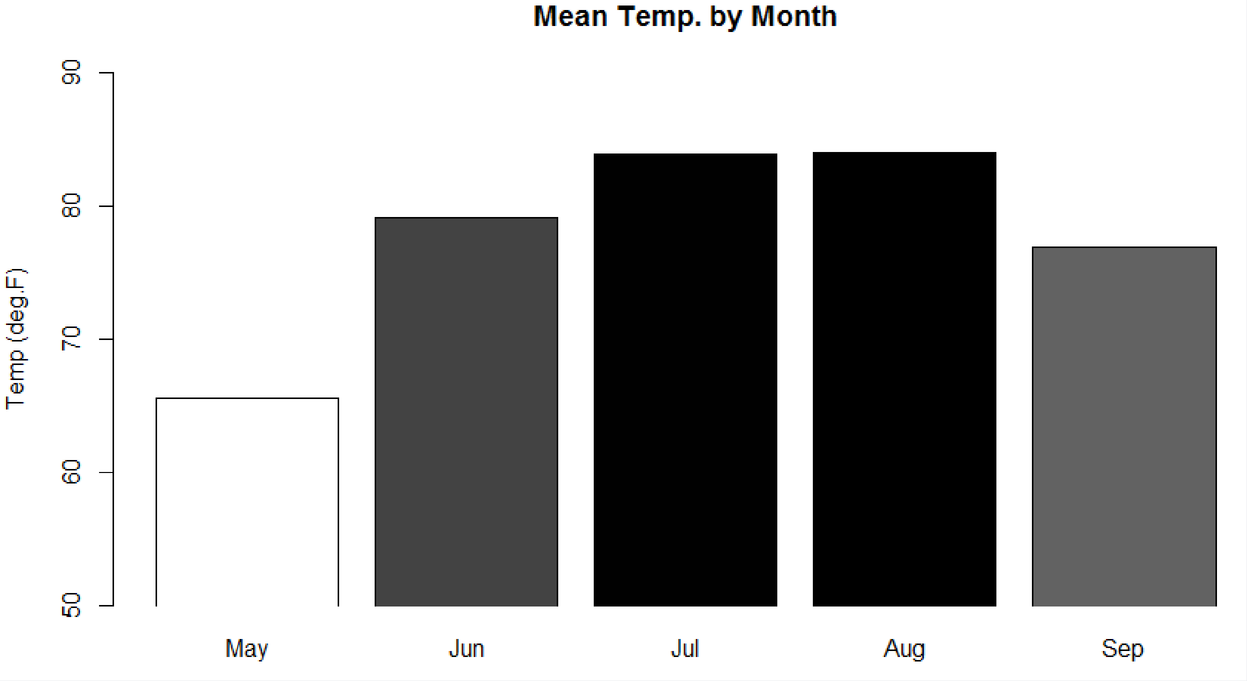


R Graphics

- Bar plots
  - ✓ For better understanding

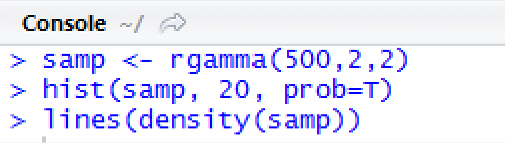


```
> rel.hts <- (heights-min(heights))/(max(heights)-min(heights))
> grays <- gray(1-rel.hts)
> barplot(heights, col=grays, ylim=c(50,90), xpd=FALSE,
+ main="Mean Temp. by Month",
+ names.arg = c("May", "Jun", "Jul", "Aug", "Sep"),
+ ylab = "Temp (deg.F)")
```



R Graphics

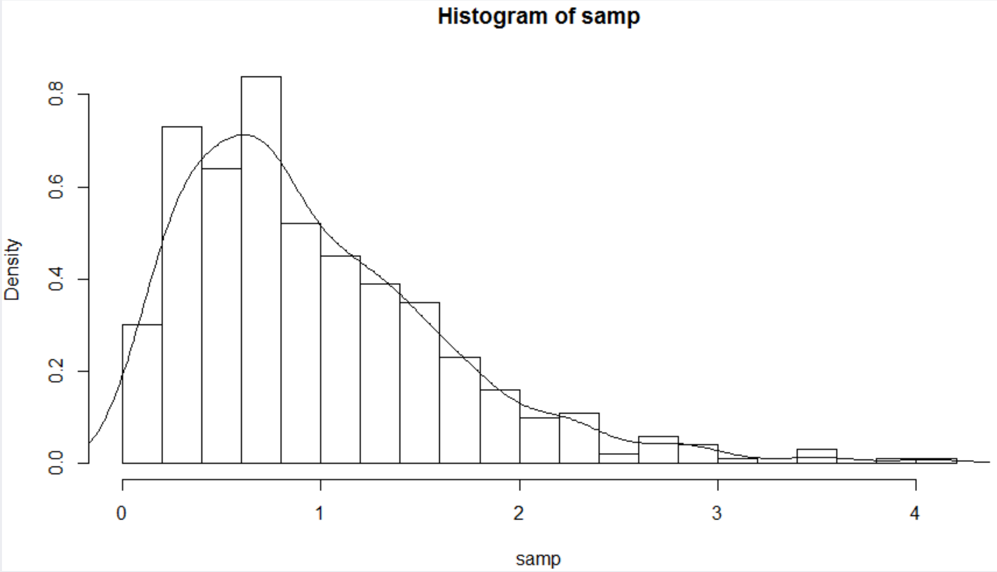
- Histogram
  - ✓ Draw a histogram and an estimated distribution

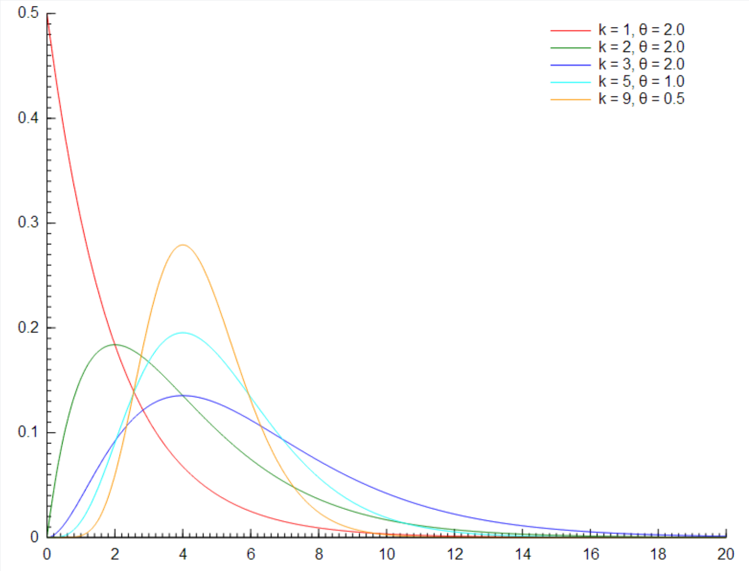


```
> samp <- rgamma(500,2,2)
> hist(samp, 20, prob=T)
> lines(density(samp))
```

$$f(x;k,\theta) = x^{k-1}\frac{e^{-x/\theta}}{\theta^k\,\Gamma(k)} \text{ for } x > 0$$

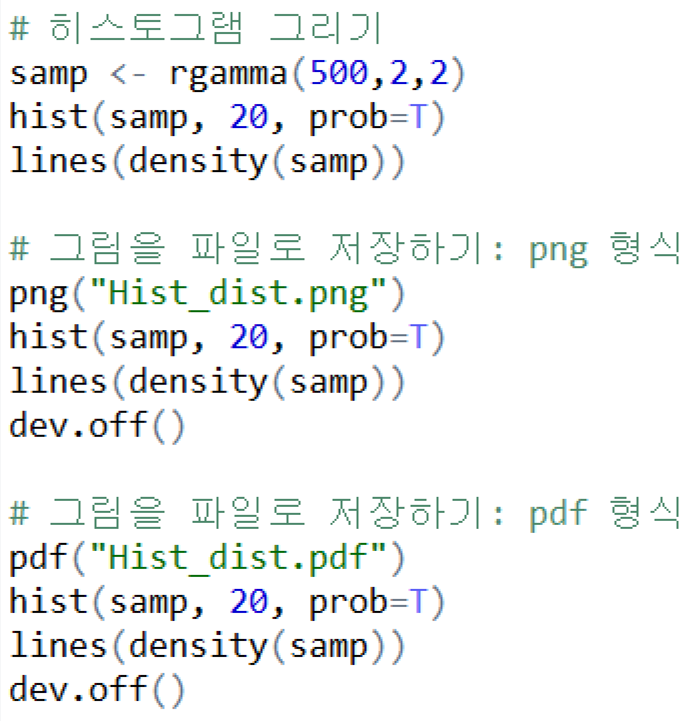
$$\Gamma(z) = \int_0^\infty t^{z-1}\,e^{-t}\,dt$$



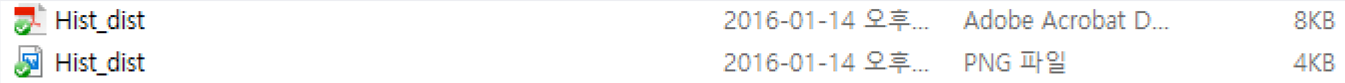


R Graphics

- Save graph object
  - ✓ Use file formats that support help(Device) in Windows



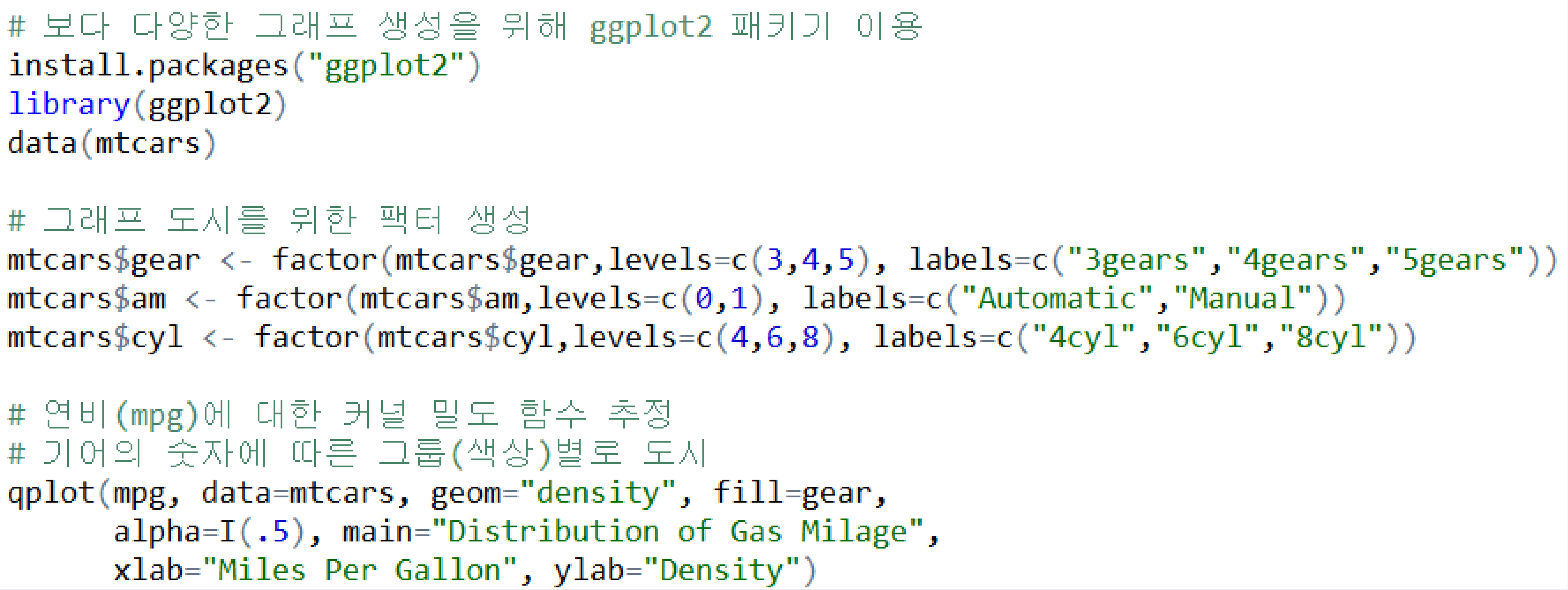
```
# 히스토그램 그리기
samp <- rgamma(500,2,2)
hist(samp, 20, prob=T)
lines(density(samp))

# 그림을 파일로 저장하기: png 형식
png("Hist_dist.png")
hist(samp, 20, prob=T)
lines(density(samp))
dev.off()

# 그림을 파일로 저장하기: pdf 형식
pdf("Hist_dist.pdf")
hist(samp, 20, prob=T)
lines(density(samp))
dev.off()
```



Hist_dist	2016-01-14 오후...	Adobe Acrobat D...	8KB
Hist_dist	2016-01-14 오후...	PNG 파일	4KB

R Graphics

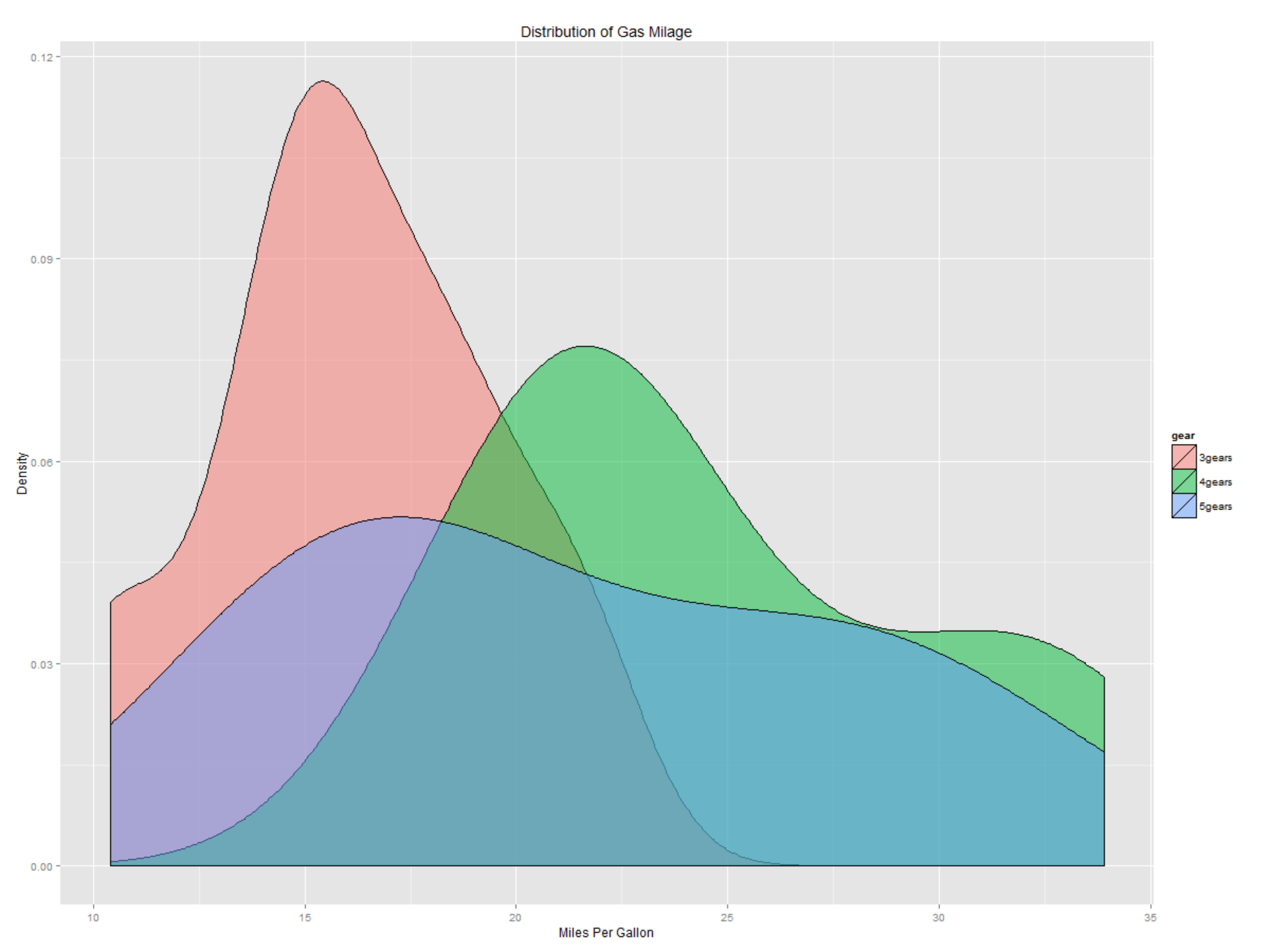
- ggplot2: Make R graph more beautiful and informative!
  - ✓ http://ggplot2.org/



```
# 보다 다양한 그래프 생성을 위해 ggplot2 패키기 이용
install.packages("ggplot2")
library(ggplot2)
data(mtcars)

# 그래프 도시를 위한 팩터 생성
mtcars$gear <- factor(mtcars$gear,levels=c(3,4,5), labels=c("3gears","4gears","5gears"))
mtcars$am <- factor(mtcars$am,levels=c(0,1), labels=c("Automatic","Manual"))
mtcars$cyl <- factor(mtcars$cyl,levels=c(4,6,8), labels=c("4cyl","6cyl","8cyl"))

# 연비(mpg)에 대한 커널 밀도 함수 추정
# 기어의 숫자에 따른 그룹(색상)별로 도시
qplot(mpg, data=mtcars, geom="density", fill=gear,
      alpha=I(.5), main="Distribution of Gas Milage",
      xlab="Miles Per Gallon", ylab="Density")
```

R Graphics

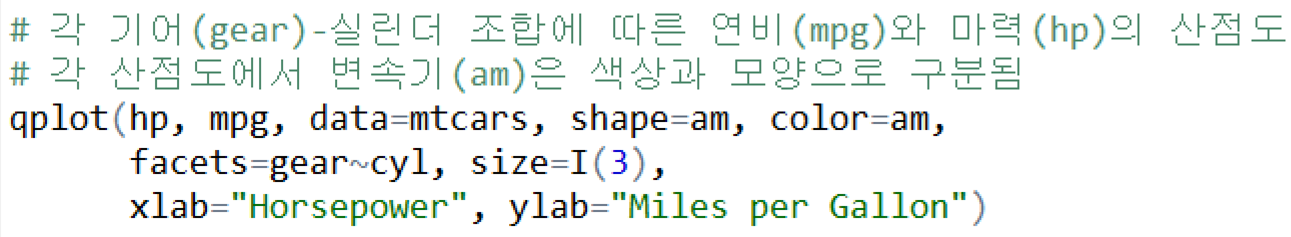
- ggplot2



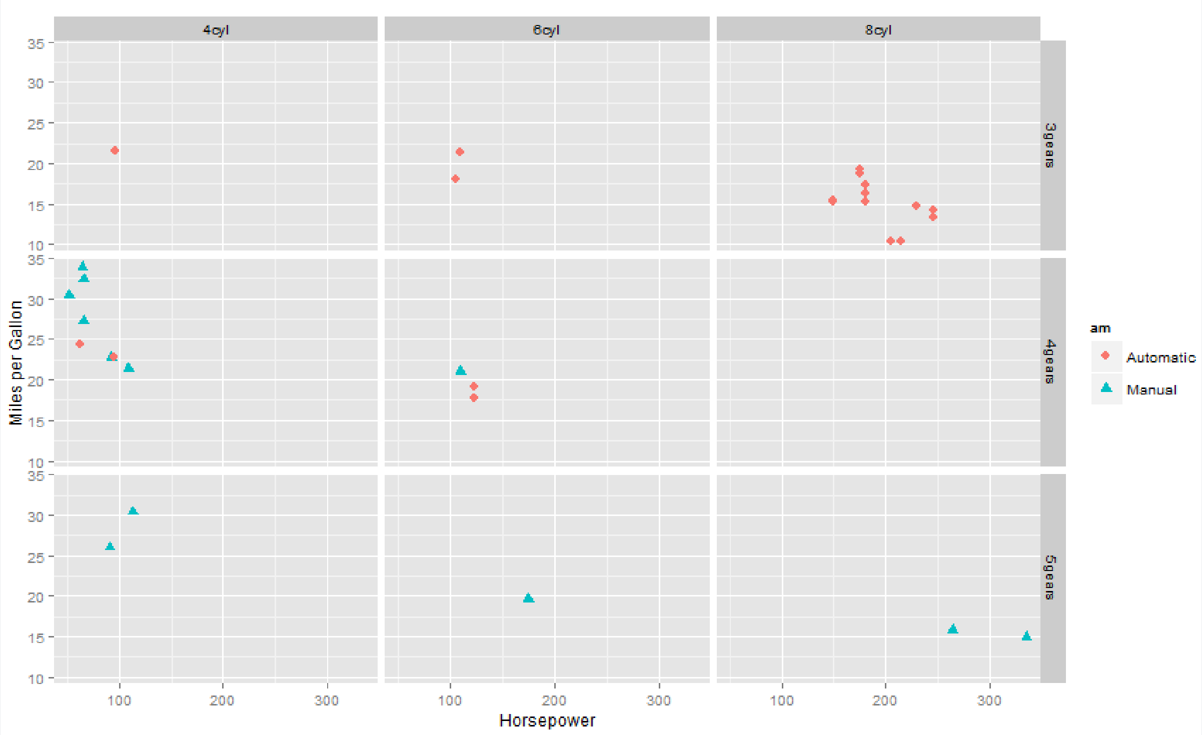
R Graphics

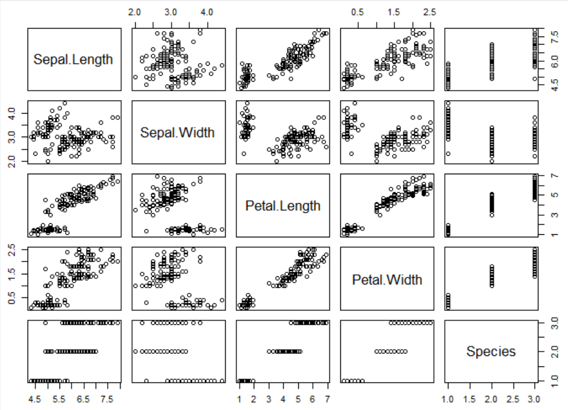
- ggplot2

  ✓ Scatter plot with ggplot2 vs graphics



```
# 각 기어(gear)-실린더 조합에 따른 연비(mpg)와 마력(hp)의 산점도
# 각 산점도에서 변속기(am)은 색상과 모양으로 구분됨
qplot(hp, mpg, data=mtcars, shape=am, color=am,
      facets=gear~cyl, size=I(3),
      xlab="Horsepower", ylab="Miles per Gallon")
```

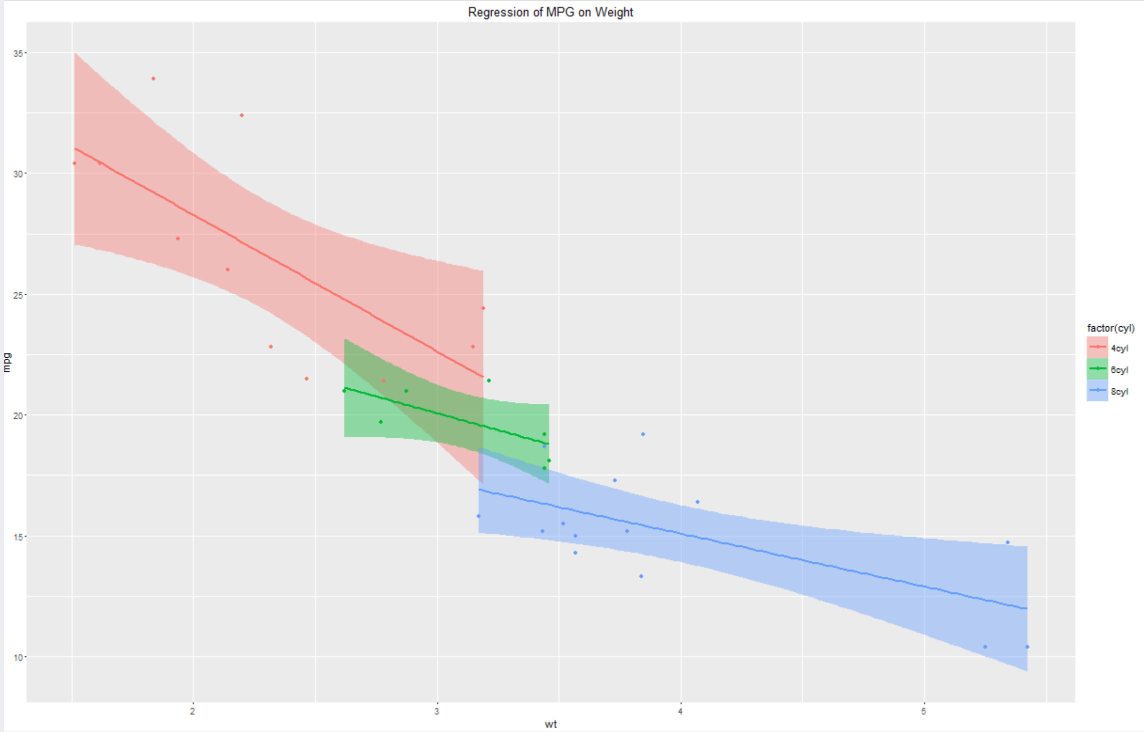




R Graphics

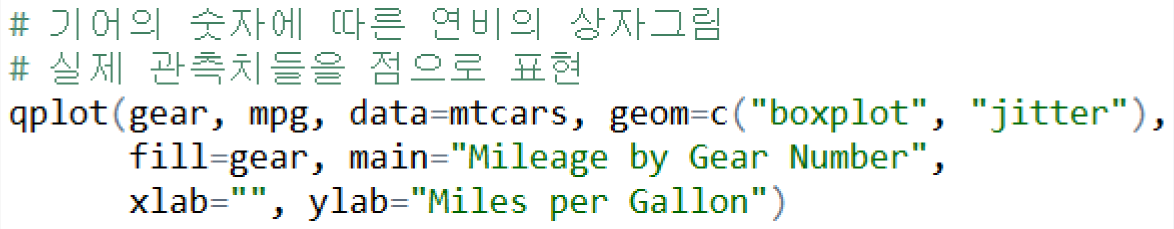
- ggplot2
  - ✓ Can use various conditions and options

```
# 실린더 갯수에 따라 공차중량(wt)과 연비(mpg)를 회귀선으로 표현
p <- ggplot(mtcars, aes(y=mpg, x=wt, colour=factor(cyl)))
p <- p + ggtitle("Regression of MPG on Weight")
p <- p + stat_smooth(method=lm, aes(fill = factor(cyl))) + geom_point()
p
```

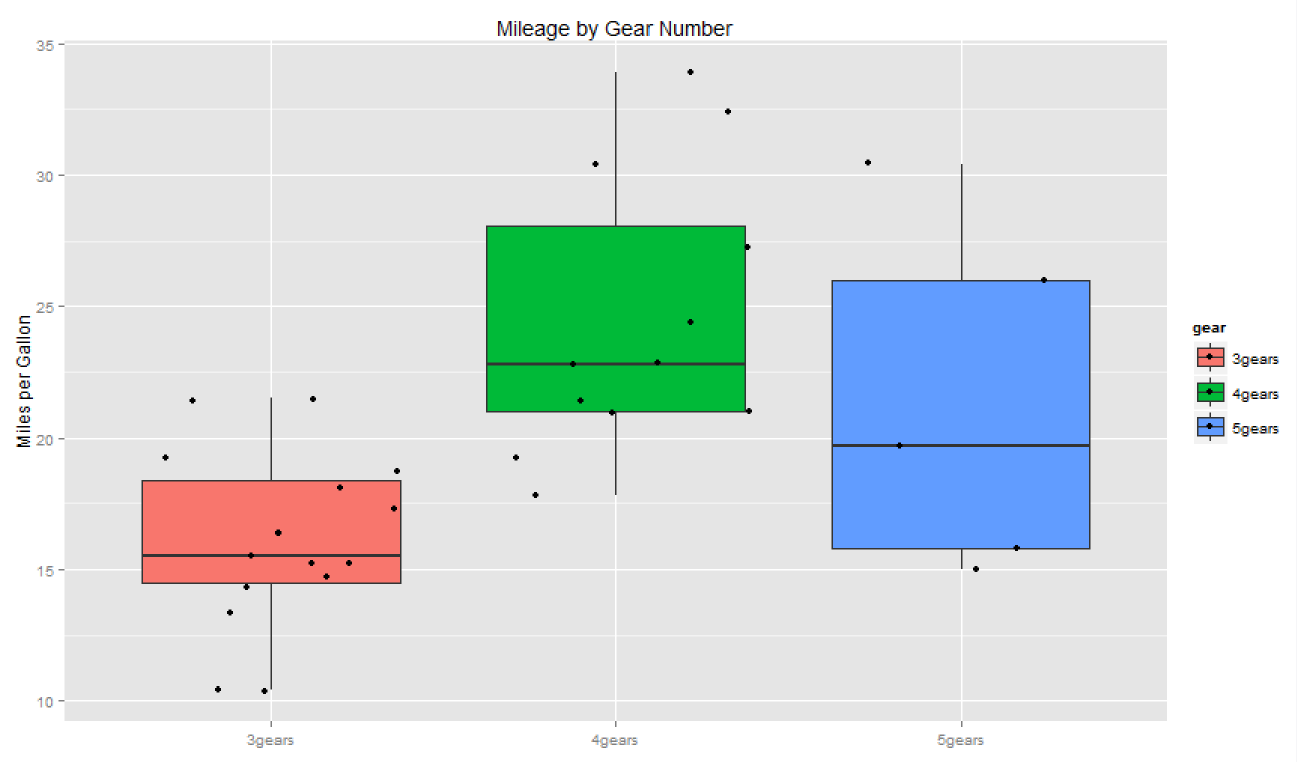


R Graphics

- ggplot2
  - ✓ Can use various conditions and options



```
# 기어의 숫자에 따른 연비의 상자그림
# 실제 관측치들을 점으로 표현
qplot(gear, mpg, data=mtcars, geom=c("boxplot", "jitter"),
      fill=gear, main="Mileage by Gear Number",
      xlab="", ylab="Miles per Gallon")
```



AGENDA

01 Conditions and Loops

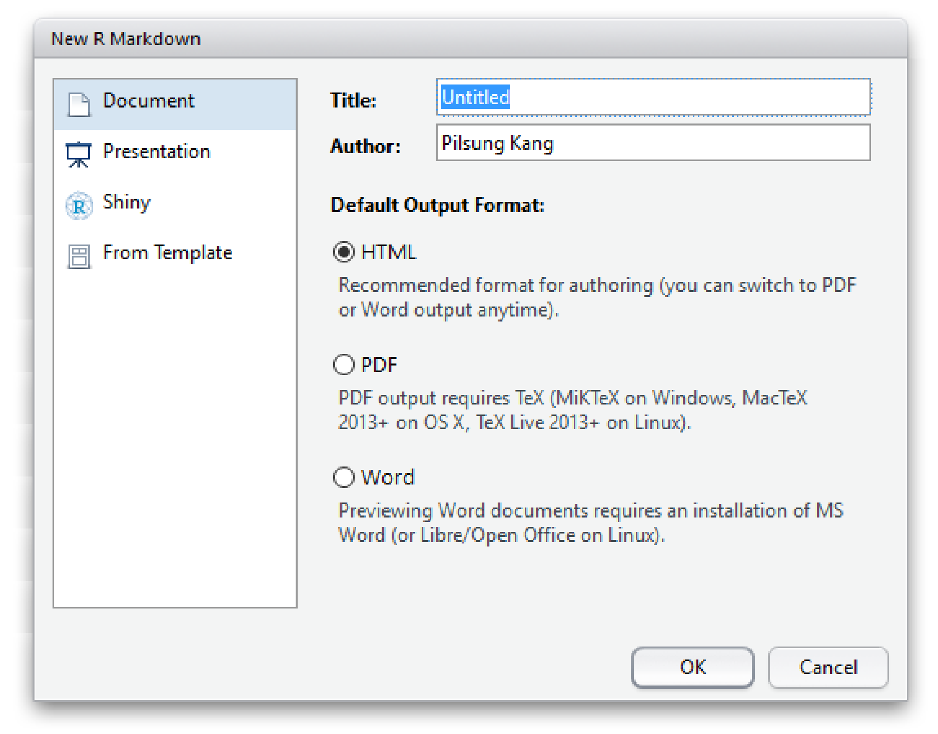
02 Functions

03 R Graphics

04 Documentation

Documentation

- Documentation using R Markdown
  - ✓ install "knitr" package
- Create an RMD fine
  - ✓ File → New File → R Markdown



New R Markdown

Title: Untitled

Author: Pilsung Kang

Default Output Format:

● HTML

Recommended format for authoring (you can switch to PDF or Word output anytime).

○ PDF

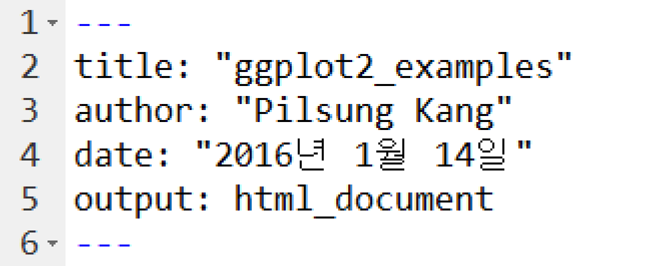
PDF output requires TeX (MiKTeX on Windows, MacTeX 2013+ on OS X, TeX Live 2013+ on Linux).

○ Word

Previewing Word documents requires an installation of MS Word (or Libre/Open Office on Linux).

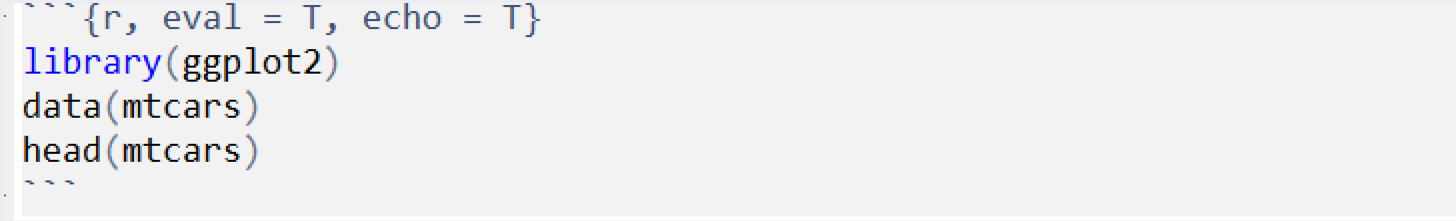
Documentation

- Meta information of the file
  - ✓ Title, author, date, output file format, etc.



```
---
title: "ggplot2_examples"
author: "Pilsung Kang"
date: "2016년 1월 14일"
output: html_document
---
```

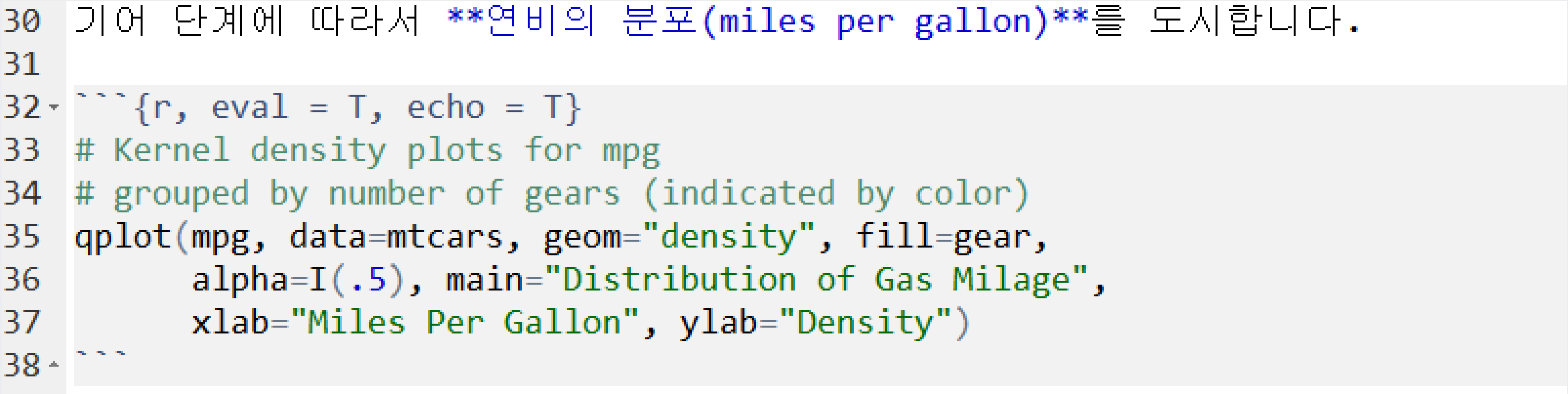
- ✓ How to embed R script in rmd file
- ✓ ```{r, eval = T, echo = T}
  - ▪ eval = T: run R script, F: do not run R script
  - ▪ echo = T: print the script in the output file, F: do not print the script



~~~
```{r, eval = T, echo = T}
library(ggplot2)
data(mtcars)
head(mtcars)
```
~~~

Documentation

- Text in rmd file
  - ✓ Parts without ``` are printed as they are
  - ✓ Use markdown syntax
    - Ex: **pilsung kang** becomes **pilsung kang** in the output file

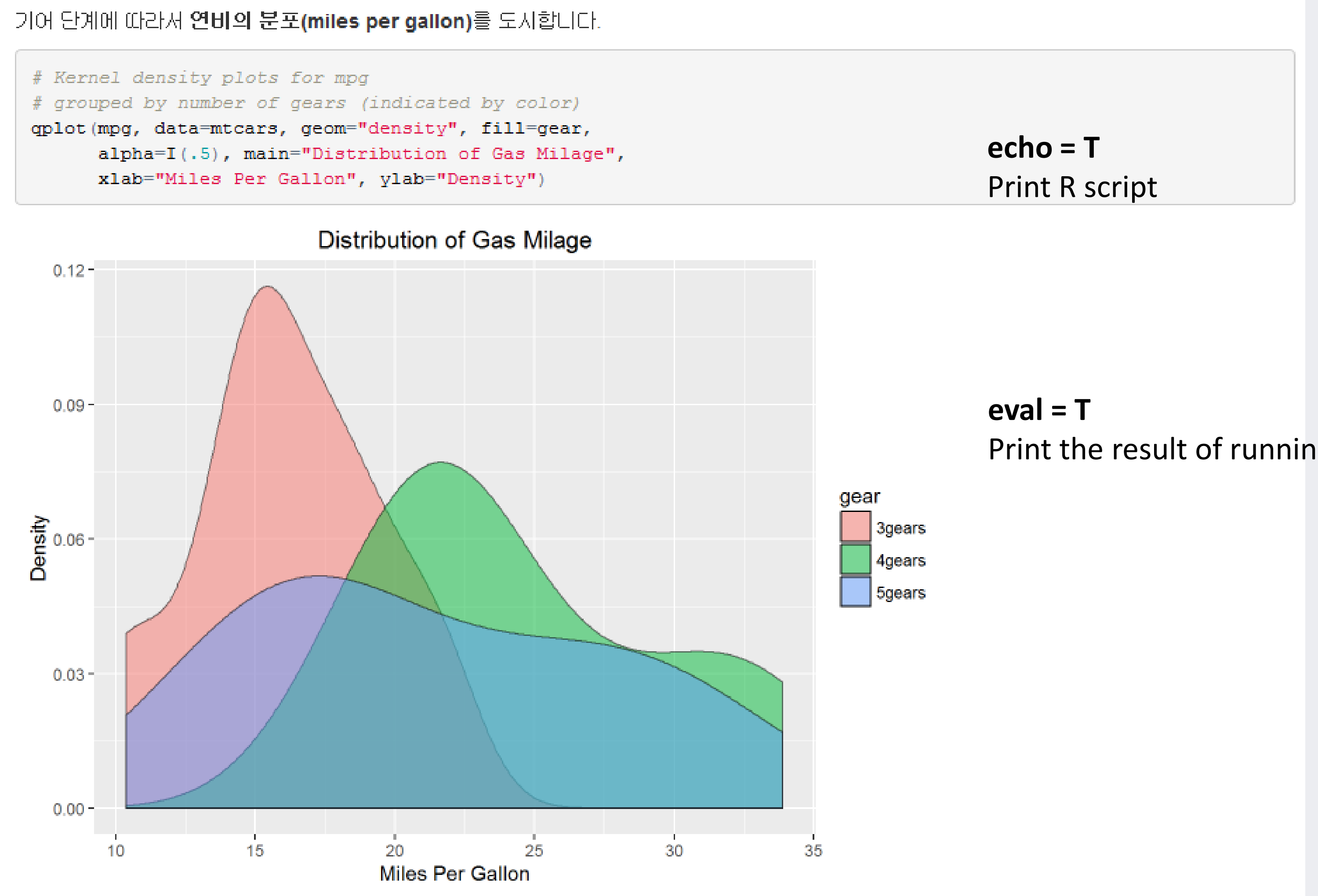


~~~
기어 단계에 따라서 **연비의 분포(miles per gallon)**를 도시합니다.

```{r, eval = T, echo = T}
# Kernel density plots for mpg
# grouped by number of gears (indicated by color)
qplot(mpg, data=mtcars, geom="density", fill=gear,
      alpha=I(.5), main="Distribution of Gas Milage",
      xlab="Miles Per Gallon", ylab="Density")
```
~~~

- For more information about markdown syntax
  - https://gist.github.com/ihoneymon/652be052a0727ad59601
  - http://blog.kalkin7.com/2014/02/05/wordpress-markdown-quick-reference-for-koreans/

Documentation

- Output style
  - ✓ Differences with eval and echo options



기어 단계에 따라서 **연비의 분포(miles per gallon)**를 도시합니다.

```
# Kernel density plots for mpg
# grouped by number of gears (indicated by color)
qplot(mpg, data=mtcars, geom="density", fill=gear,
      alpha=I(.5), main="Distribution of Gas Milage",
      xlab="Miles Per Gallon", ylab="Density")
```

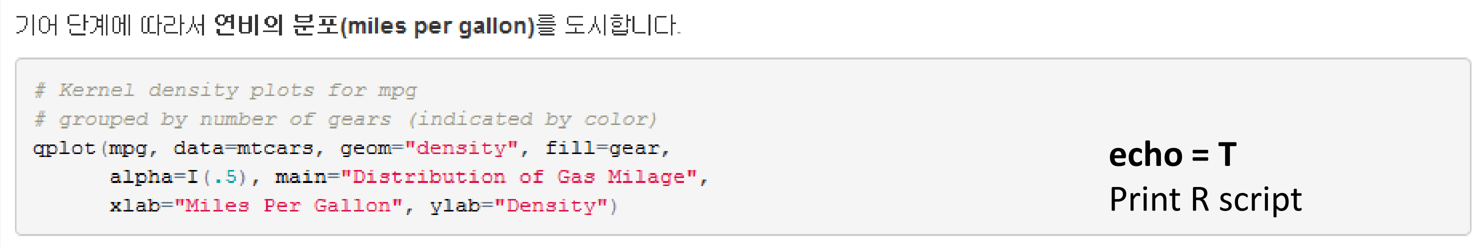
echo = T
Print R script

eval = T
Print the result of running R script

Documentation

- Output style

  ✓ Differences with eval and echo options



기어 단계에 따라서 **연비의 분포(miles per gallon)**를 도시합니다.

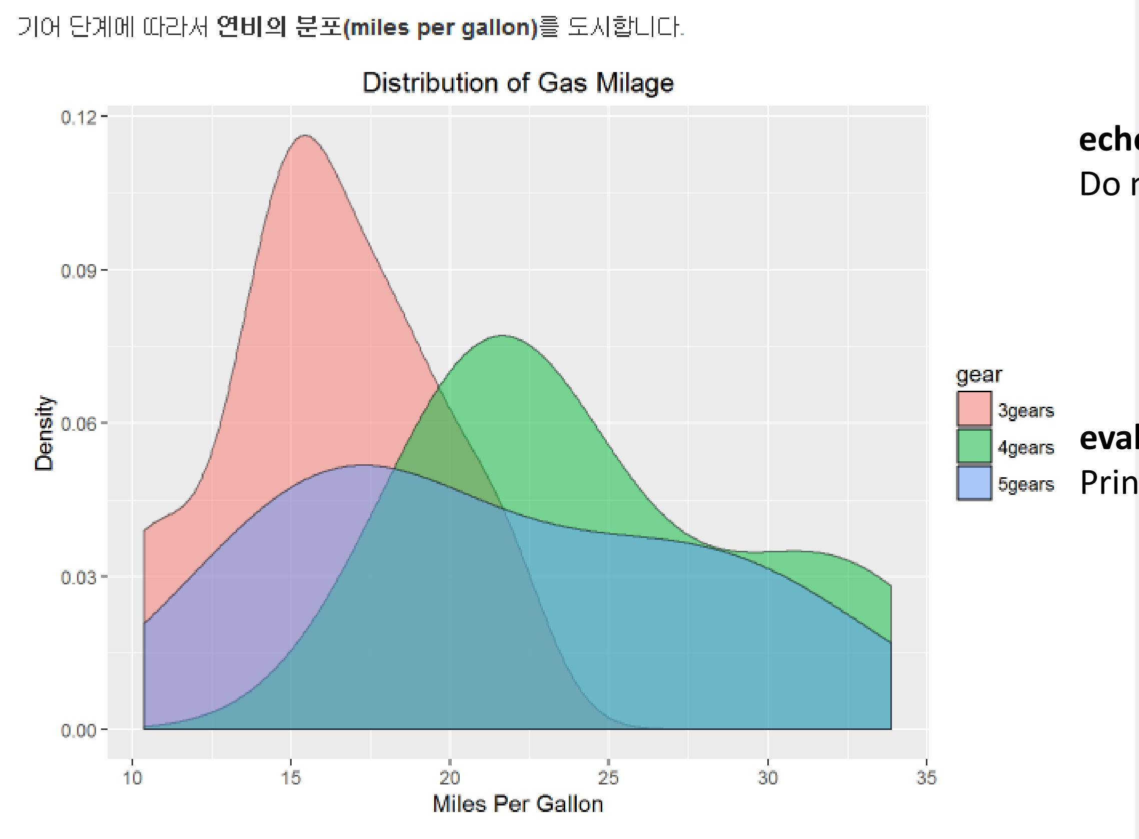
```
# Kernel density plots for mpg
# grouped by number of gears (indicated by color)
qplot(mpg, data=mtcars, geom="density", fill=gear,
      alpha=I(.5), main="Distribution of Gas Milage",
      xlab="Miles Per Gallon", ylab="Density")
```

echo = T
Print R script

eval = F
Nothing appears because R script is not run

Documentation

- Output style
  - ✓ Differences with eval and echo options



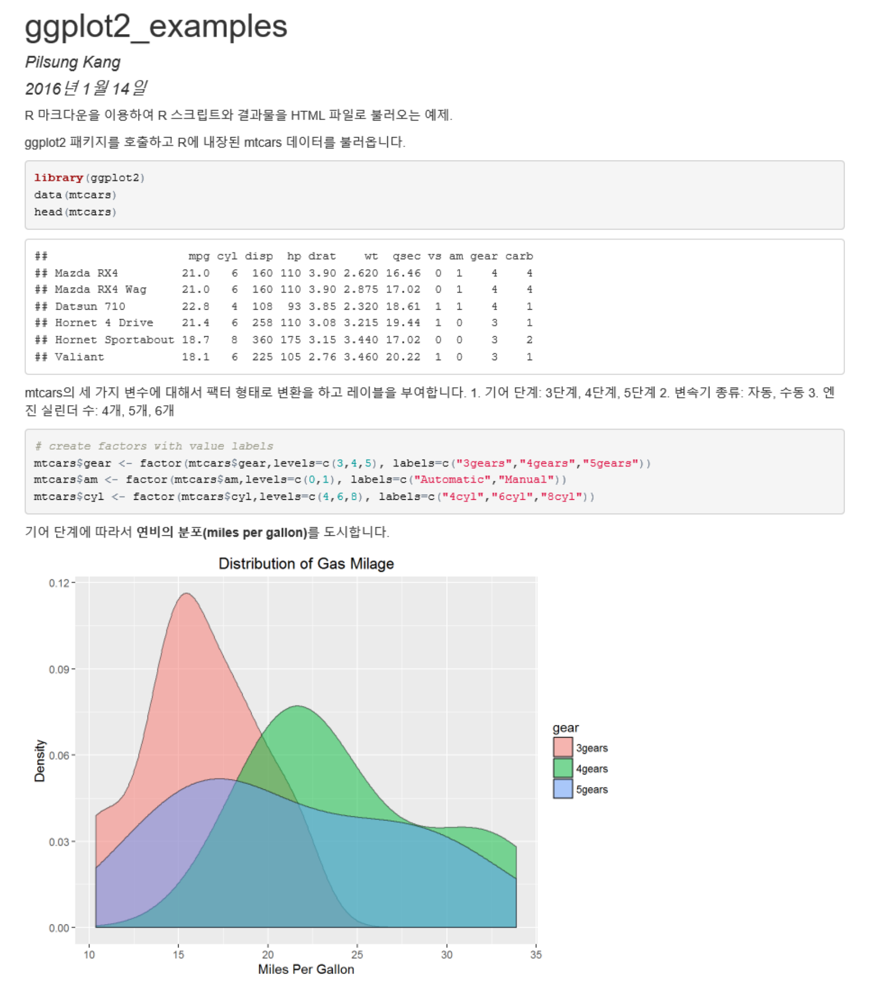
기어 단계에 따라서 **연비의 분포(miles per gallon)**를 도시합니다.

echo = F
Do not print R script

eval = T
Print the result of running R script

Documentation

- Output style
  - ✓ save html file format



ggplot2_examples

Pilsung Kang

2016년 1월 14일

R 마크다운을 이용하여 R 스크립트와 결과물을 HTML 파일로 불러오는 예제.

ggplot2 패키지를 호출하고 R에 내장된 mtcars 데이터를 불러옵니다.

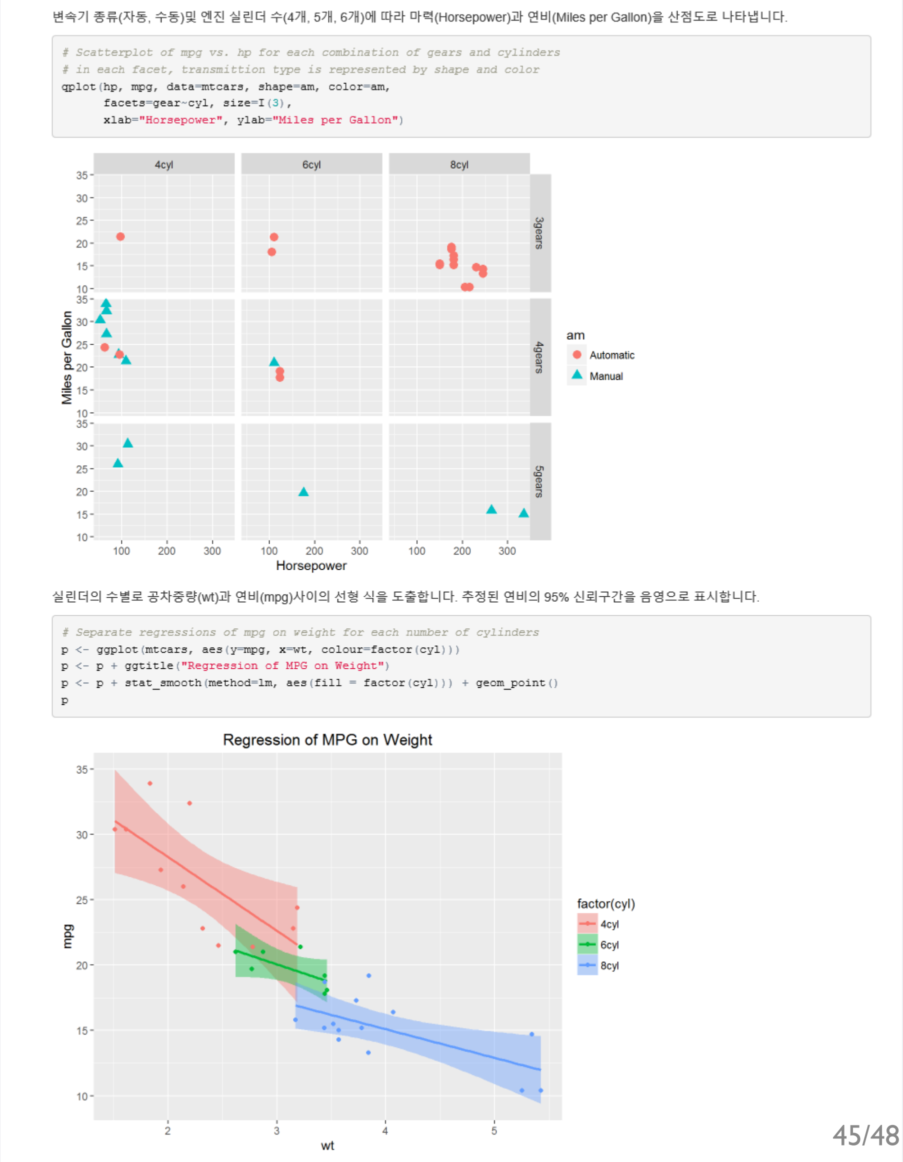
```
library(ggplot2)
data(mtcars)
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

mtcars의 세 가지 변수에 대해서 팩터 형태로 변환을 하고 레이블을 부여합니다. 1. 기어 단계: 3단계, 4단계, 5단계 2. 변속기 종류: 자동, 수동 3. 엔진 실린더 수: 4개, 5개, 6개

```
# create factors with value labels
mtcars$gear <- factor(mtcars$gear,levels=c(3,4,5), labels=c("3gears","4gears","5gears"))
mtcars$am <- factor(mtcars$am,levels=c(0,1), labels=c("Automatic","Manual"))
mtcars$cyl <- factor(mtcars$cyl,levels=c(4,6,8), labels=c("4cyl","6cyl","8cyl"))
```

기어 단계에 따라서 **연비의 분포(miles per gallon)**를 도시합니다.



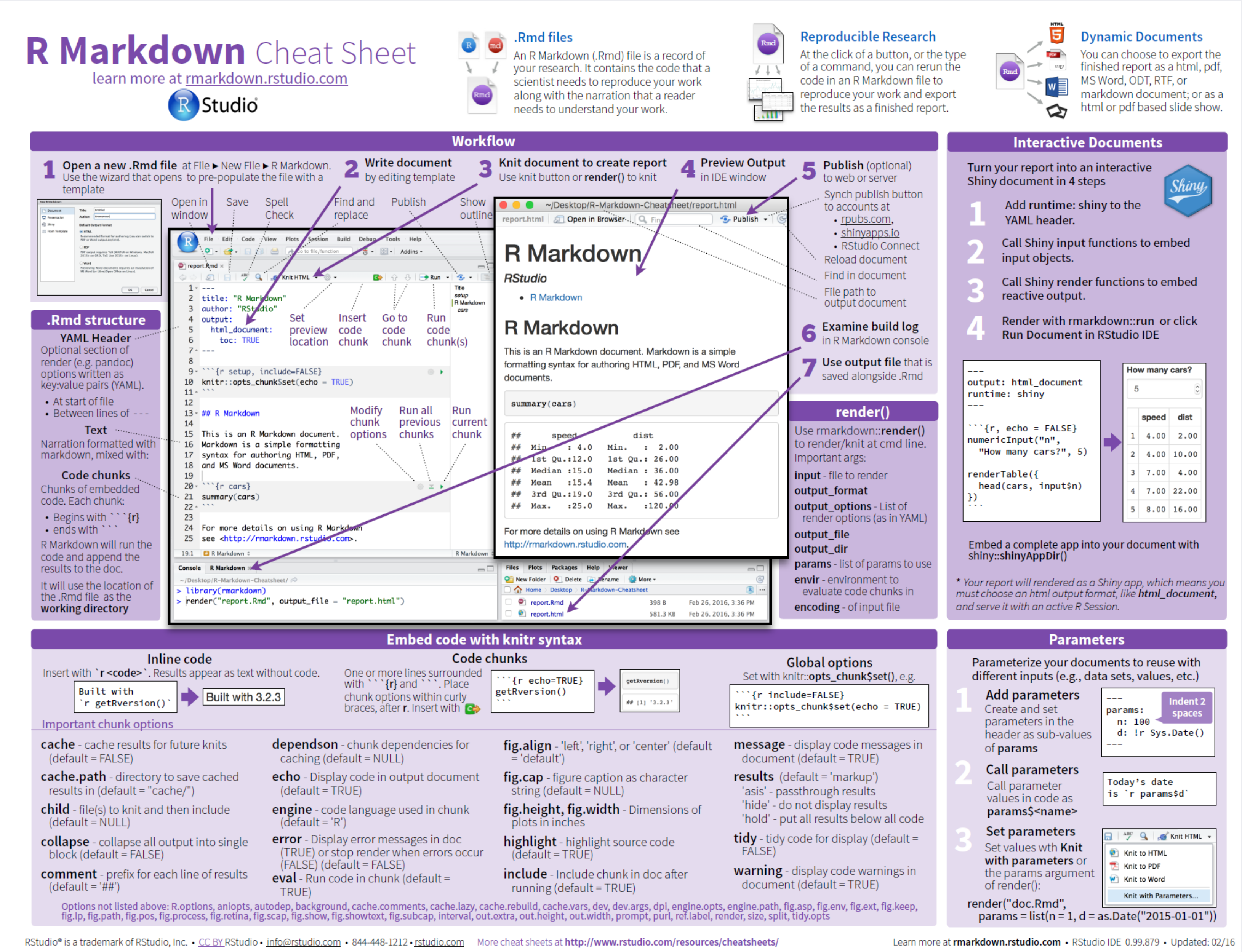
변속기 종류(자동, 수동)및 엔진 실린더 수(4개, 5개, 6개)에 따라 마력(Horsepower)과 연비(Miles per Gallon)을 산점도로 나타냅니다.

```
# Scatterplot of mpg vs. hp for each combination of gears and cylinders
# in each facet, transmittion type is represented by shape and color
qplot(hp, mpg, data=mtcars, shape=am, color=am,
      facets=gear~cyl, size=I(3),
      xlab="Horsepower", ylab="Miles per Gallon")
```

실린더의 수별로 공차중량(wt)과 연비(mpg)사이의 선형 식을 도출합니다. 추정된 연비의 95% 신뢰구간을 음영으로 표시합니다.

```
# Separate regressions of mpg on weight for each number of cylinders
p <- ggplot(mtcars, aes(y=mpg, x=wt, colour=factor(cyl)))
p <- p + ggtitle("Regression of MPG on Weight")
p <- p + stat_smooth(method=lm, aes(fill = factor(cyl))) + geom_point()
p
```

R Markdown Cheat Sheet



R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com

RStudio

.Rmd files

An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

Reproducible Research

At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents

You can choose to export the finished report as a html, pdf, MS Word, ODT, RTF, or markdown document; or as a html or pdf based slide show.

Workflow

1 Open a new .Rmd file at File ▸ New File ▸ R Markdown. Use the wizard that opens to pre-populate the file with a template

2 Write document by editing template

3 Knit document to create report Use knit button or **render()** to knit

4 Preview Output in IDE window

5 Publish (optional) to web or server

6 Examine build log in R Markdown console

7 Use output file that is saved alongside .Rmd

Open in window · Save · Spell Check · Find and replace · Publish · Show outline

Set preview location · Insert code chunk · Go to code chunk · Run code chunk(s)

Modify chunk options · Run all previous chunks · Run current chunk

Synch publish button to accounts at
- rpubs.com,
- shinyapps.io
- RStudio Connect

Reload document

Find in document

File path to output document

~~~
---
title: "R Markdown"
author: "RStudio"
output:
  html_document:
    toc: TRUE
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
~~~

R Markdown

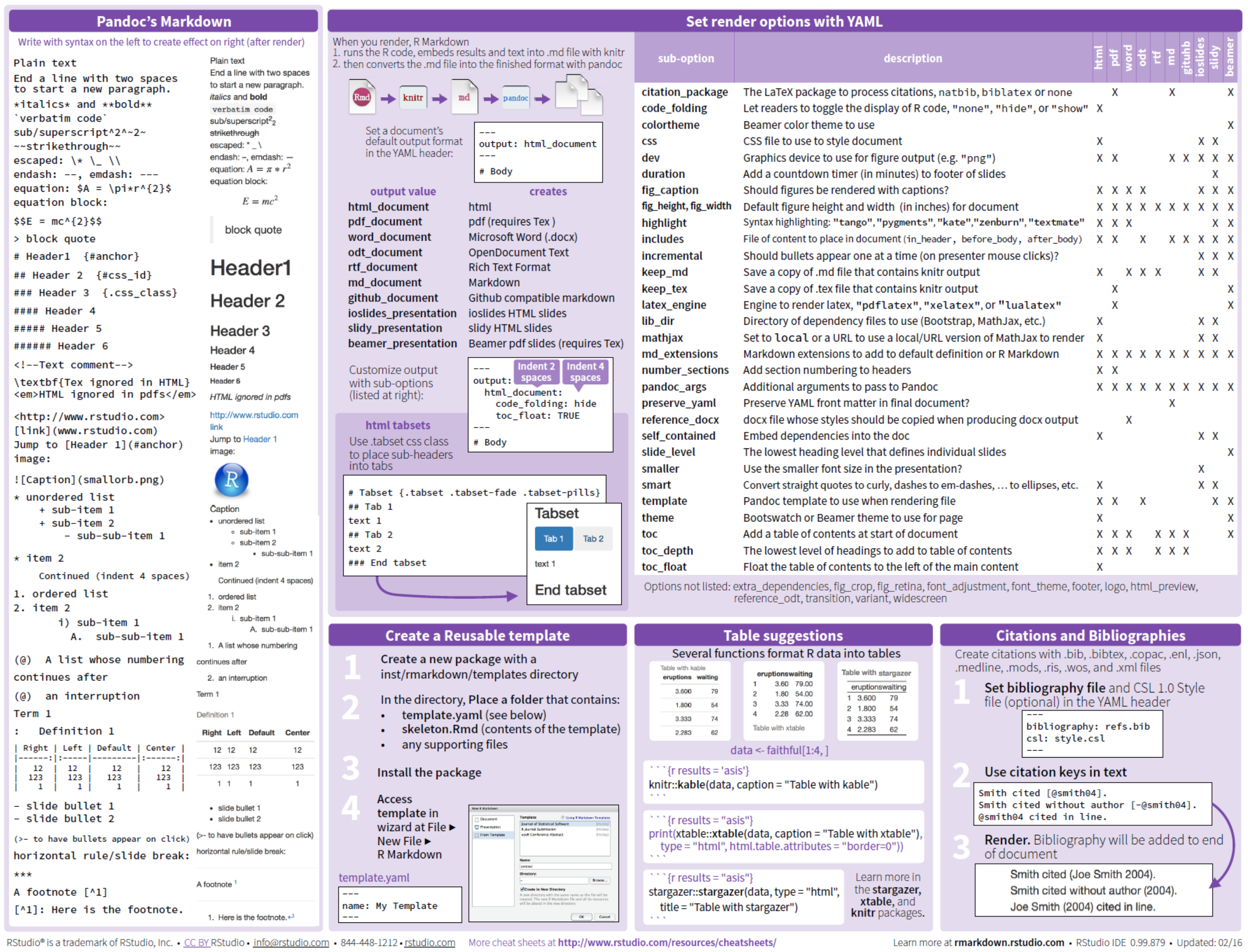
This is an R Markdown document.
Markdown is a simple formatting
syntax for authoring HTML, PDF,
and MS Word documents.

```{r cars}
summary(cars)
```

For more details on using R Markdown
see <http://rmarkdown.rstudio.com>.
```

Console · R Markdown

```
~/Desktop/R-Markdown-Cheatsheet/
> library(rmarkdown)
> render("report.Rmd", output_file = "report.html")
```

~/Desktop/R-Markdown-Cheatsheet/report.html

R Markdown

RStudio

- R Markdown

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents.

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

For more details on using R Markdown see http://rmarkdown.rstudio.com.

Home › Desktop › R-Markdown-Cheatsheet

report.Rmd	398 B	Feb 26, 2016, 3:36 PM
report.html	581.3 KB	Feb 26, 2016, 3:36 PM

.Rmd structure

YAML Header
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).
- At start of file
- Between lines of - - -

Text
Narration formatted with markdown, mixed with:

Code chunks
Chunks of embedded code. Each chunk:
- Begins with ```{r}
- ends with ```

R Markdown will run the code and append the results to the doc.

It will use the location of the .Rmd file as the **working directory**

render()

Use rmarkdown::**render()** to render/knit at cmd line. Important args:

input - file to render

output_format

output_options - List of render options (as in YAML)

output_file

output_dir

params - list of params to use

envir - environment to evaluate code chunks in

encoding - of input file

Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

1. Add **runtime: shiny** to the YAML header.
2. Call Shiny **input** functions to embed input objects.
3. Call Shiny **render** functions to embed reactive output.
4. Render with **rmarkdown::run** or click **Run Document** in RStudio IDE

```
---
output: html_document
runtime: shiny
---

```{r, echo = FALSE}
numericInput("n",
  "How many cars?", 5)

renderTable({
  head(cars, input$n)
})
```
```

How many cars?

5

	speed	dist
1	4.00	2.00
2	4.00	10.00
3	7.00	4.00
4	7.00	22.00
5	8.00	16.00

Embed a complete app into your document with **shiny::shinyAppDir()**

* *Your report will rendered as a Shiny app, which means you must choose an html output format, like* ***html_document,*** *and serve it with an active R Session.*

Embed code with knitr syntax

Inline code

Insert with `` `r <code>` ``. Results appear as text without code.

```
Built with
`r getRversion()`
```

Built with 3.2.3

Code chunks

One or more lines surrounded with ```**{r}** and ```. Place chunk options within curly braces, after **r**. Insert with

```
```{r echo=TRUE}
getRversion()
```
```

```
getRversion()
## [1] '3.2.3'
```

Global options

Set with knitr::**opts_chunk$set()**, e.g.

```
```{r include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```
```

Important chunk options

cache - cache results for future knits (default = FALSE)

cache.path - directory to save cached results in (default = "cache/")

child - file(s) to knit and then include (default = NULL)

collapse - collapse all output into single block (default = FALSE)

comment - prefix for each line of results (default = '##')

dependson - chunk dependencies for caching (default = NULL)

echo - Display code in output document (default = TRUE)

engine - code language used in chunk (default = 'R')

error - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)

eval - Run code in chunk (default = TRUE)

fig.align - 'left', 'right', or 'center' (default = 'default')

fig.cap - figure caption as character string (default = NULL)

fig.height, fig.width - Dimensions of plots in inches

highlight - highlight source code (default = TRUE)

include - Include chunk in doc after running (default = TRUE)

message - display code messages in document (default = TRUE)

results (default = 'markup')
'asis' - passthrough results
'hide' - do not display results
'hold' - put all results below all code

tidy - tidy code for display (default = FALSE)

warning - display code warnings in document (default = TRUE)

Options not listed above: R.options, aniopts, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts

Parameters

Parameterize your documents to reuse with different inputs (e.g., data sets, values, etc.)

1 Add parameters
Create and set parameters in the header as sub-values of **params**

```
---
params:
  n: 100
  d: !r Sys.Date()
---
```

Indent 2 spaces

2 Call parameters
Call parameter values in code as **params$<name>**

```
Today's date
is `r params$d`
```

3 Set parameters
Set values wth **Knit with parameters** or the params argument of render():

Knit HTML
- Knit to HTML
- Knit to PDF
- Knit to Word
- Knit with Parameters...

render("doc.Rmd",
  params = list(n = 1, d = as.Date("2015-01-01"))

More cheat sheets at http://www.rstudio.com/resources/cheatsheets/ Learn more at **rmarkdown.rstudio.com** • RStudio IDE 0.99.879 • Updated: 02/16
```

# R Markdown Cheat Sheet



## Pandoc's Markdown

Write with syntax on the left to create effect on right (after render)

```
Plain text
End a line with two spaces
to start a new paragraph.
*italics* and **bold**
`verbatim code`
sub/superscript^2^~2~
~~strikethrough~~
escaped: \* \_ \\
endash: --, emdash: ---
equation: $A = \pi*r^{2}$
equation block:

$$E = mc^{2}$$

> block quote

# Header1  {#anchor}

## Header 2  {#css_id}

### Header 3  {.css_class}

#### Header 4

##### Header 5

###### Header 6

<!--Text comment-->

\textbf{Tex ignored in HTML}
<em>HTML ignored in pdfs</em>

<http://www.rstudio.com>
[link](www.rstudio.com)
Jump to [Header 1](#anchor)
image:

![Caption](smallorb.png)

* unordered list
    + sub-item 1
    + sub-item 2
        - sub-sub-item 1

* item 2

    Continued (indent 4 spaces)

1. ordered list
2. item 2
      i) sub-item 1
        A.  sub-sub-item 1

(@)  A list whose numbering

continues after

(@)  an interruption

Term 1

:   Definition 1

| Right | Left | Default | Center |
|------:|:-----|---------|:------:|
|   12  |  12  |    12   |    12  |
|  123  |  123 |   123   |   123  |
|    1  |    1 |     1   |     1  |

- slide bullet 1
- slide bullet 2

(>- to have bullets appear on click)

horizontal rule/slide break:

***

A footnote [^1]

[^1]: Here is the footnote.
```

Plain text
End a line with two spaces
to start a new paragraph.
italics and **bold**
`verbatim code`
sub/superscript$^2{}_2$
~~strikethrough~~
escaped: * _ \
endash: –, emdash: —
equation: $A = \pi * r^2$
equation block:

$$E = mc^2$$

> block quote

Header1

Header 2

Header 3

Header 4

Header 5

Header 6

HTML ignored in pdfs

http://www.rstudio.com
link
Jump to Header 1
image:

Caption

- unordered list
    - sub-item 1
    - sub-item 2
        - sub-sub-item 1
- item 2

    Continued (indent 4 spaces)

1. ordered list
2. item 2
    i. sub-item 1
        A. sub-sub-item 1

1. A list whose numbering

continues after

2. an interruption

Term 1

Definition 1

Right	Left	Default	Center
12	12	12	12
123	123	123	123
1	1	1	1

- slide bullet 1
- slide bullet 2

(>- to have bullets appear on click)

horizontal rule/slide break:

A footnote [1]

1. Here is the footnote.↩

Set render options with YAML

When you render, R Markdown
1. runs the R code, embeds results and text into .md file with knitr
2. then converts the .md file into the finished format with pandoc

Rmd → knitr → md → pandoc →

Set a document's default output format in the YAML header:

```
---
output: html_document
---
# Body
```

output value	creates
html_document	html
pdf_document	pdf (requires Tex)
word_document	Microsoft Word (.docx)
odt_document	OpenDocument Text
rtf_document	Rich Text Format
md_document	Markdown
github_document	Github compatible markdown
ioslides_presentation	ioslides HTML slides
slidy_presentation	slidy HTML slides
beamer_presentation	Beamer pdf slides (requires Tex)

Customize output with sub-options (listed at right):

```
---
output:
  html_document:
    code_folding: hide
    toc_float: TRUE
---
# Body
```

Indent 2 spaces
Indent 4 spaces

html tabsets

Use .tabset css class to place sub-headers into tabs

```
# Tabset {.tabset .tabset-fade .tabset-pills}
## Tab 1
text 1
## Tab 2
text 2
### End tabset
```

Tabset
Tab 1 Tab 2
text 1
End tabset

sub-option	description	html	pdf	word	odt	rtf	md	gituhb	ioslides	slidy	beamer
citation_package	The LaTeX package to process citations, `natbib`, `biblatex` or `none`		X				X				X
code_folding	Let readers to toggle the display of R code, `"none"`, `"hide"`, or `"show"`	X									
colortheme	Beamer color theme to use										X
css	CSS file to use to style document	X							X	X	
dev	Graphics device to use for figure output (e.g. `"png"`)	X	X				X	X	X	X	X
duration	Add a countdown timer (in minutes) to footer of slides									X	
fig_caption	Should figures be rendered with captions?	X	X	X	X				X	X	X
fig_height, fig_width	Default figure height and width (in inches) for document	X	X	X	X	X	X	X	X	X	X
highlight	Syntax highlighting: `"tango"`, `"pygments"`, `"kate"`,`"zenburn"`, `"textmate"`	X	X	X						X	X
includes	File of content to place in document (`in_header, before_body, after_body`)	X	X		X		X	X	X	X	X
incremental	Should bullets appear one at a time (on presenter mouse clicks)?								X	X	X
keep_md	Save a copy of .md file that contains knitr output	X		X	X	X			X	X	
keep_tex	Save a copy of .tex file that contains knitr output		X								X
latex_engine	Engine to render latex, `"pdflatex"`, `"xelatex"`, or `"lualatex"`		X								X
lib_dir	Directory of dependency files to use (Bootstrap, MathJax, etc.)	X							X	X	
mathjax	Set to `local` or a URL to use a local/URL version of MathJax to render	X							X	X	
md_extensions	Markdown extensions to add to default definition or R Markdown	X	X	X	X	X	X	X	X	X	X
number_sections	Add section numbering to headers	X	X								
pandoc_args	Additional arguments to pass to Pandoc	X	X	X	X	X	X	X	X	X	X
preserve_yaml	Preserve YAML front matter in final document?						X				
reference_docx	docx file whose styles should be copied when producing docx output			X							
self_contained	Embed dependencies into the doc	X							X	X	
slide_level	The lowest heading level that defines individual slides										X
smaller	Use the smaller font size in the presentation?								X		
smart	Convert straight quotes to curly, dashes to em-dashes, … to ellipses, etc.	X							X	X	
template	Pandoc template to use when rendering file	X	X		X					X	X
theme	Bootswatch or Beamer theme to use for page	X									X
toc	Add a table of contents at start of document	X	X	X		X	X	X			X
toc_depth	The lowest level of headings to add to table of contents	X	X	X		X	X	X			
toc_float	Float the table of contents to the left of the main content	X									

Options not listed: extra_dependencies, fig_crop, fig_retina, font_adjustment, font_theme, footer, logo, html_preview, reference_odt, transition, variant, widescreen

Create a Reusable template

1 Create a new package with a inst/rmarkdown/templates directory

2 In the directory, Place a folder that contains:
- template.yaml (see below)
- skeleton.Rmd (contents of the template)
- any supporting files

3 Install the package

4 Access template in wizard at File ▸ New File ▸ R Markdown

template.yaml

```
---
name: My Template
---
```

New R Markdown
Document
Presentation
From Template
Template:
Using R Markdown Templates
Journal of Statistical Software {rticles}
R Journal Submission {rticles}
useR Conference Abstract {rticles}
Name:
Untitled
Directory:
~
Browse...
☑ Create in New Directory
A new directory with the same name as the file will be created. The new R Markdown file and all its resources will be placed in the new directory.
OK Cancel

Table suggestions

Several functions format R data into tables

Table with kable

eruptions	waiting
3.600	79
1.800	54
3.333	74
2.283	62

	eruptions	waiting
1	3.60	79.00
2	1.80	54.00
3	3.33	74.00
4	2.28	62.00

Table with xtable

Table with stargazer

	eruptions	waiting
1	3.600	79
2	1.800	54
3	3.333	74
4	2.283	62

data <- faithful[1:4,]

~~~
```{r results = 'asis'}
knitr::kable(data, caption = "Table with kable")
```
~~~

~~~
```{r results = "asis"}
print(xtable::xtable(data, caption = "Table with xtable"),
  type = "html", html.table.attributes = "border=0"))
```
~~~

~~~
```{r results = "asis"}
stargazer::stargazer(data, type = "html",
  title = "Table with stargazer")
```
~~~

Learn more in the **stargazer**, **xtable**, and **knitr** packages.

Citations and Bibliographies

Create citations with .bib, .bibtex, .copac, .enl, .json, .medline, .mods, .ris, .wos, and .xml files

1 **Set bibliography file** and CSL 1.0 Style file (optional) in the YAML header

```
---
bibliography: refs.bib
csl: style.csl
---
```

2 **Use citation keys in text**

```
Smith cited [@smith04].
Smith cited without author [-@smith04].
@smith04 cited in line.
```

3 **Render.** Bibliography will be added to end of document

Smith cited (Joe Smith 2004).
Smith cited without author (2004).
Joe Smith (2004) cited in line.

More cheat sheets at **http://www.rstudio.com/resources/cheatsheets/** Learn more at **rmarkdown.rstudio.com** • RStudio IDE 0.99.879 • Updated: 02/16

ANY
questions?