STAT 3027

2. Vector

① When are [brackets] used and when are (parentheses) used?
↓ index ↓ combine

② which methods are there to remove names?

③ what are the difference between <-, =, and ==?

★ V <- c(2.71, 5, 3.14)

combination

length(V)

V

index	value
i	v[i]
1	2.71
2	5
3	3.14

words <- c("tree", "ant", "chainsaw")

words[1]

★ Basic Types. Specifying Constants

numeric (real number) 3.14e2 = 3.14×10² = 314

character

Escape sequence

\" (double quote)

String = paste (sep = " ", " ")

cat (string)

cat (sep = " ", ",")

logical : TRUE and FALSE

Sum (V>3)
=> 2

Vector (mode = "logical", length = 0)
creates a vector of the given mode & length
logical(0)

=5

FALSE ... FALSE

numric =5

0 0000

chal

change a vector's type
a(.numric()
as. character ()
as. logical

integer
complex
raw

as. numeric (V>3)

W= c ("34", "12", "45")
W
"34" "12" "45"
Sum (W)
↳ errol.
W.numbers = as.numric(w)
W.numbers
34 12 45

sum (w. numbers)

91

sun (as. numaric (w))

91

★ Names attribute and a few functions

name (x) gets or set.

name (v) = c("e", "five", "pi").

v

e five pi

2.71 5.00 3.14

name (v) = NULL # remove names

== y = c (burger = 2.50 ; fries =1.50)

y

> burger fries

2.5 1.5

names (y) = NULL

2.5 1.5

A Few Functions!

x <_ c (R, 11, 16, 11)

★ Operators

%/% : quotient

%% remainder

sqrt(sum((x - mean(x)) ^2) / (n-1))

seq (10, 15, by=2)
  c(10, 12, 14)
seq (10, 15, length.out = 3)
  10.0, 12.5, 15.0

matching : %in%
  1:3 %in% c(2,7)
    FALSE TRUE FALSE

☆ Indexing

4. X[V]
  ↓
  negative ⇒ exclude.
  indice = which (x < 14)
    index   [illegible]
X[indices]
  X[x<14]
☆ which can be omitted.

x[(x%%2)==0]

3. Vector (continued) and List

SORTING Functions
sort (x, decreasing = FALSE)
  x = c(12, 11, 16, 11)
  ⇒ 11, 11, 12, 16

```
v = rank (x, ties.method = "average")
        第几大
                          "first"

v = order (x, ..., decreasing = FALSE)
   the index of smallest elements in x.
                 x[v] is sorted.
```

Structure, summary, quantile

```
str (object)
  summary (object)
quantile

  v = quantile (x, probs = c(0.25, 0.75))
```

NULL and NA special values

```
names (x)
 x[3] = NA
   sum (x)     = NA
   sum (x, na.rm = TRUE)

      is.null()  ,  is.na()   — fn
```

File input/output

```
 scan(file = " ", what = numeric())

 write (x, file = "data")
```

List : (not necessarily of the same type
can be named)

$ operator
y$8 ⇒ null

unlict (y, usenames=FALSE)

4. Data Frame work
(R's fundamental data structure)
A data frame is (≈) a list of vector

Catergorical Variables

Factor ⇒ a vector of catergorical values
factor (x, levels, labels = levels)

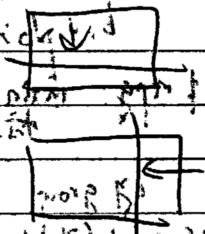
table (...)

Manipulation Examples
m$_
m[, 1]
row colum

m [1:3, 1:3]
dim (m)

n.rows = dim (m)[1]
n.cols = length (m)

head(m)



tail(m)

rownames(m)

m$hp[30] = 25

M = median(m$hp)

sorted = m [order(mcyl, mdisp)]

write.table(x, file =)

read.csv( )

Formula.

5. Graphics

1. Comon parameters

formula, data.

main, sub, xlab, ylab

xlim, ylim,

pch

cex

par

2. Numeric data

```
boxplot (x)          boxplox ( x ~g)
boxplot (mtcars$mpg, main = "Gas mileage", ylab = "
                          标.题                ylim = c (0, 40))
                     by group                    y轴范围
boxplot ( mpg ~ factor (cyl), data =      , xlab.
        ylab =          );

Stripchart (x, method=" overplot")
Strip chart (x ~g)
      overplot
      jitter       > duplicate
      stack

hist (x, breaks =" Sturges", freq = NULL)
                                   density
☆ cex. axis = 2
调尺寸 cex. lab = 2.

plot (x, y)          } ⇒ 4.17
points (x,y)
   x = 1:5;  y = 2*x; plot(x, y)

  ? points
  lines (x = c(1, 3, 5, 7, 9)
         y = c (8, 1, 4, 1, 8)
               col = "red")
```

plot(density(x))

rug(x) add the data points, xistan

pairs (x)

curses (expr = x*sin(1/x), from = -pi/6, to = pi/6, n=200)

点数

curve (expr = x*1, add = TRUE, col = "red")

Legends

legend(x, y, lengend, col = par

legend ("top", legend = c("x* sin(1/x)", "x"),
col = c("black", "red"), lty c(1, 1))

? legend

expression(..) ? par

? plotmath

e.g. frac(x, y)

stem(plot())?

categorical data

bar plot. (height, names. arg = NULL)

AT variable mosaicplot(x)

count = table( );

barplot (counts, names.arg = c( ))

Multiple figures

matrix (data, nrow, ncol, byrow = FALSE)

layout (mat)

layout. Show(3)

Write graphical output to a file

dev. off(6)

x = rnorm (100)

pdf ("[illegible]. pdf")

plot.

dev. off(x)

Test and Intervals

One mean or the Difference of Two mean

out = t. test (x, y = NULL, alternativo = "two sided".
mu = 0 conf level = .95) To test mu = 0

$ parameter : degree of freedom

$ statistic Student's test statistic

$ p. value

$ conf. int

$ estimate

str(out)

out$p. value enter

two means

$H_0: \mu_x - \mu_y = \mu_0$

F Test for Equality of Variances

out = var.test(x, y, ~~ratio~~ ratio = 1, alternative = "two sided", conf.level = .95)

$H_0 \quad \frac{\sigma_x^2}{\sigma_y^2} = \text{ratio}.$

$ parameter $\quad (n_x - 1) \quad (n_y - 1)$

$ statistic $\quad f = \frac{s_x^2}{s_y^2}$

$ p.value

$ conf.int.

chi-Squared Tests

Goodness of fit

counts = c(..) probs = c(...)

out = chisq.test(x = counts, p = probs)

$ parameter

$ statistic

Independence / Homogenity

$ expected. expected counts under H_0

chisq.test();

One Proportion or the Difference of Two Proportions out = prop.test (x, n= p0, alternative = "two sided", conf.level = .95

z-test $H_0: p = p_0 \Rightarrow$

$estimate $\hat{p} = x/n$

2-vetors x and n

$H_0: p_1 - p_2 = 0 \quad (p_1 = p_2)$

$$\hat{p} = \frac{x_1 + x_2}{n_1 + n_2}$$

7. Regression

1. Simple linear Regression

$y = mx + b$

cor(x, y) / correlation ⇒ 数

lm(y~x, data)

A = cor (x) ⇒ matrix A

cor (x [, i], y[j,])

2. lm (y ~ x , data)

Call:

anova (m)

m$coefficients [1]

ablive (a, b) $y = a + bx$

= ablive (reg=m)

ablive (a = mean (car$dist)

horizontal

---

predict (mode)

d = data.frame (speed = seq (from 0, to 25, by 1))

y.hat = predict (m, ~~new~~ newdata=d)

points (x = d$speed, y = y.hat, pch = 19, cex=3)

---

plot (m$ fitted. values, m$residual) no pattern.
↓
ε → indepent & normal

abline (0, 0)

QQplot.

N (μ = mean (residuals), σ = sd(residuals))

qqline (x)

e.g x = rnorm (n=100) ; qqnorm (x); qqline(x)

w = rexp (100), random qq

or use plot (m)

---

multiple Linear Regression

m = lm (y ~ x1 + x2 + ... , data=)

Confint (m,

8. Simulation

① repeatable

set. seed(seed)

set.seed(0); a =rnorm(1)

set.seed(0); b=rnorm(1)

② Repeat ~~a~~ calculation n times

replicate(n, expr)

set.seed(0)

③ ? distribution.

a. $\overline{X} \approx N(\mu, \frac{\sigma}{\sqrt{n}})$

t =replicate (N, {x =rnorm (n, mean =

symbols   text(x=col, y=row, labal= useChar, cex=4)

identity(XX, YY, n=1, plot= FALSE)   replicate(, ___)   ↓数字

unique(   )%in%

function

parity = ifelse((X%%2)==0, "even", "odd")

2. loop: while( ){ }

repeat { }   if( ){ break}

Loop one ~~nor~~ or more times: repeat { expression

if (condition) {

break

}

}

input: XX = scan(what = character numeric(), n=1, quiet = TRUE)

skip: if (c) { next }

3. Apply Function   lapplay (X= , FUN=) treating data frame as a list of vectors

☆ } sapply (X= , FUN=) ⇒ vector, matrix, array

multiple argument apply | mapply (FUN, x, y) take the several vectors in and applies FUN to all first elements and then to all second elements

apply (X= , MARGIN, FUN, ...)   =1 rows   =2 columns

tapply (X= , Index, FUN= ..)

subset of vector

tapply(X= mtcar$mpg, INDEX = mtcars$cyl, FUN=)   分类!

tapply(X= ___, INDEX = list(mtcars$cyl, mtcar$gear), FUN=)

4 Matrix   m = matrix (data = vector, nrow= , ncol= , byrow = TRUE,   extra

dimnames = list(c(colname), c(rowname)))   probs=c(

cbind (matrix, ___)   rbind (matrix, ___)

m[row(m) == col(m)]   main diagonal

m [row(m) + col(m) == r - c]   diagonal through (r, c)

m   t   t   reverse.

A * B element-wise product   A%*%B matrix product

solve (a=A, b=b)   A $\vec{x}$ = $\vec{b}$ A%x%x --- matrix - vector product

5. pattern

grep(pattern, x, ignore.case = FALSE, value = FALSE) → indice of elements of charac

grep(pattern =   ... TRUE) ⇒ values

sub (pattern =, replacement = , x=) a copy of x after replacing 1st occure of pattern with replacement

gsub ( ) all   (TRUE ( ),)

\d 0123456789 \w (letter) \s \b 大写取过来

+ 4
1 1 2 3 5 8.

gsub (pattern = "[aeiou]", replacement = "", x=a) # strip vowels
gsub (pattern = "[^aeiou]", replacement = "", x=a) # strip non vowels

— ^ matches the beginning of a line ($ matches the end) e.g
— grep (pattern="^r", x=a) , grep (pattern="^r", ignore.case =TRUE , x=a)
— \\< matches the begining of a word (\\>) end
e.g grep (pattern = "e\\>", x=a)

repetition. {n} exactly n times / {n,} n or more times * ({0,}, + {1,})
{n,m} n–m times inclusive ? ⇒ {0, 1} or "optional"

e.g. grep (pattern = "\\d{4}$", x=a) 4 digits, end of lines
grep (pattern = "□\\d{4}$", x=a) # space, 4 digits, end of line
grep (pattern = " \\d{4, 5}$", x=a) # space, 4 or 5 digits, end of lie

\\N refers to what the Nth enclosed expression matched
— sub (pattern = "(\\w+), (\\w+) +(\\d+)(\\w+).*", replacement= "\\2, \\1, \\4, \\3", x=)

| means or grep (pattern = "Joe |Jack", x=a) grep (pattern ="J(o|a)", x=a)
\\ ⇒ ?regex

— Splitting Strings strsplit (x=a, split=",") strsplit (x=a, split=" +" ⇒ 拆分所有 space
⇒ write(x= csv, file = " ") d=read.csv(" ", header=FALSE, col.names=c( ))

6. ggplot ① qplot (data= , x= ___ (cols) y= ___ , geom= "boxplot")
② ggplot 1. gg1<- ggplot(data = trees, aes (x= __, y=))
gg1 + geom_boxplot()+ geom_jitter() + violin() + coord_flip() (flip coord)
2. + xlab(" ") + ylab(" ") + labs (title=" ") + theme(plot.title = element_text(size=22, vjust=2, hjust
3. ggplot(aes=(x, y, fill= x or y) + guides(fill = FALSE)
4. reorder species by mean ggplot (data, aes(x = reorder (x, y, fun=mean), y= ,)
5. outlier shape + geom_boxplot(outlier.shape=21) 6. change to log scale + scale_y_log10(labels= function(l) {paste (l, " in.")})
7. color brewer – scale_fill_brewer (palette = "Set1")
geom_density() geom_point() + geom_smooth() + geom_line() aes(color = factor( )) shape = factor()
拟合 cofiden int
+ facet_grid(. ~) + scale_x_sqrt (limits = c(0, 50))

7. web scrap TEAM <- "http: __ " lines <- readLines (TEAM)
tem.lines <- grep (pattern =" ", x= lines, value = TRUE) team.names <- sub (x= tem.lines, pattern= " (.*)
replace ="\\1") link<-paste0(___, ___, ___) table<-read.HTMLTable (link)
table [[3]] [,] [ ] → as. numeric (as. character( ))

Vector sort( ) order x[order(x)] na.rm=TRUE list[[ ]]
write.table (csv) (x, file =" ",...) table = read.table (csv) (..)
stopifnot (function( ) == " ")
turn a vector into a character ⇒ paste(v, sep=" ", collapse=" ")
⇒ = data.frame(,)

Web scraping

```
library("XML"  ); TEAM <- "http://www.nfl.com/teams"  ; lines <- readLines(TEAM)
team.lines <- grep(pattern="statistics", x = lines, value=TRUE)
```

- grab just those team names and the abbr from "\t\t\t\t\t\t\t\t<option value=\"/teams/baltimoreravens/statistics?team=BAL\">Statistics</option>"

```
team.names <- sub(x = team.lines, pattern=".*teams/(.*)/statistics.*", replace = "\\1")
team.abbreviation <- sub(x = team.lines, pattern = ".*team=(.*)\">.*", replace="  \\1"  )
```

- specifically i want to get information from:

http://www.nfl.com/teams/baltimoreravens/statistics?season=2014&team=BAL&seasonType=REG#

```
link <- paste0(TEAM, "/", team.names[1], "/statistics?season=2014&team=", team.abbreviation[1],
"&seasonType=REG#"  ); tables <- readHTMLTable(link)
```

- for loop for all the teams

```
rushing <- NULL; receiving <- NULL
for (i in 1:length(team.names)) {
  link <- paste0(TEAM, "/", team.names[i], "/statistics?season=2014&team=", team.abbreviation[i],
"&seasonType=REG#")
  tables <- readHTMLTable(link); print(link)
  rushing[i] <- as.numeric(as.character(tables[[3]][2,3])); receiving[i] <- as.numeric(as.character(tables[[4]][2,3])) }
```

Apply

- lapply (" list apply") applies function FUN to each element of vector or list X, returning a list of the same length

```
lapply(X=mtcars, FUN=mean); sapply(mtcars, mean); sapply(mtcars, mean, simplify=FALSE, USE.NAMES =
FALSE);
```

- sapply - simplified apply, return numbers
- mapply(FUN, ...)(multiple arguments apply") takes the several vectors in ... and applies FUN to all first elements, then to all second elements, etc.

```
mapply(sum, x, y, z); apply(X=m, MARGIN=1, FUN=sum) # keep dimension 1 (rows)
```

- apply- applies the specified function over "margins" (MARGIN=1 for row, MARGIN=2 for column)
- tapply - applied the function over a subset indicated with INDEX

```
tapply(X=mtcars$mpg, INDEX=mtcars$cyl, FUN=mean); tapply(X=mtcars$mpg, INDEX=list(mtcars$cyl,
mtcars$gear), FUN=mean)
tapply(X=mtcars$mpg, INDEX=mtcars$cyl, FUN=quantile, probs=c(.25, .75))
```

Matrix

- main diagonal: m[row(m) == col(m)] • diagonal through (r, c): m[row(m) - col(m) == r - c] • reverse diagonal through (r, c): m[row(m) + col(m) == r + c]
- A * B is an element-wise product; A %*% B is the usual matrix product
- solve(a=A, b=b) gives the solution ⃗x to the system of linear equations, A⃗x = ⃗b

Pattern

grep(pattern, x, ignore.case=FALSE, value=FALSE) 不分大小写，显示满足条件的index。value=TRUE：显示满足条件

的x中的结果。

sub(pattern = "e", replacement = "E", x = a)替换第一次出现的e; gsub(pattern = "e", replacement = "E", x = a)替换所有的e

- \\w, \\s, \\d: words, space, digit
- \\W, \\S, \\D 大写表示不是word, space, digits;
- [aeiou], 其中任何一个
- [^aeiou]不包括所有的
- grep(pattern = "Joe|Jack", x = a)或者
- ^a以a开头
- pattern = “ e\\>” 以e结尾
- . \ | ( ) [{ ^ $ * + ? 遇到这些前面要加\\ (double backslash)

grep(pattern = " \\d{4,5}$", x = a) # space, 4 or 5 digits, end-of-line

sub(pattern = "(\\w+),(\\w+) +(\\d+) (\\w+).*", replacement = "\\2,\\1,\\4,\\3", x=a) 满足pattern条件的内容按照2，1，4，3的顺序排列

sub(pattern=".*<a href=(.*)>.*", replacement="\\1", x=link) 满足pattern条件的只去处括号里的内容

- Splitting Strings for character vector x

  strsplit(x=a, split=“ ,”); strsplit(x=a, split=" +”); strsplit(x=a, split=",|(+)")

ggplot：library(ggplot2)

- box plot或其他图 qplot(data = trees, x = species, y = dbh, geom="boxplot”) • ggplot(data = trees, aes(x =, y =)) + geom_boxplot()
- 加点，加线：ggplot(data = anscombe, aes(x = x1, y=y1)) + geom_point() + geom_smooth(method=lm) + geom_line()
- 不同颜色分组画图：ggplot(data = mtcars, aes(x=mpg, y=qsec, color=factor(cyl), shape=factor(gear)))
- 自己设定颜色：+ scale_color_manual(values = c("pink", “ black”))
- 加标题,调字大小：+ ylab("Diameter") + xlab("Species") + labs(title="Some trees")+ theme(plot.title = element_text(size=22, vjust=2, hjust=1))
- 重新排序，填色，不要注释：ggplot(data = trees, aes(x= reorder(species, dbh,fun=mean), y = dbh, fill = species))+ guides(fill= FALSE)
- 把y变成log形式，填色种类：+ scale_y_log10(labels = function(l) {paste(l, " in.")} + scale_fill_brewer(palette = "Set1")
- 画多张图on same plot： + facet_grid(.~gear)； + facet_grid(carb~gear)；+ facet_wrap(carb~gear)
- scale_x_sqrt(limits=c(0, 50))
- error bar：se <- sqrt(diag(vcov(mod1)))； gg2 <- ggplot(data = sim, aes(x = Species, y = pred, fill=Species)) + geom_bar(stat="identity”)
  - □ 标准差：gg2 + geom_errorbar(aes(ymax = pred + se, ymin = pred - se), width=.25)
  - □ 置信区间：gg2 + geom_errorbar(aes(ymax = pred + qnorm(.975)*se, ymin = pred - qnorm(.975)*se), width=.25)
  - □ 画两端不出头的range：把 geom_errorbar 换成geom_pointrange；粗粗的range：换成geom_crossbar
  - □ 在bar上加字母：+ geom_text(aes(x = Species, y = pred + se + 1 任意高度, label = group要加的字母))