

CS 133 - Introduction to Computational and Data Science

Instructor: Renzhi Cao
Computer Science
Department
Pacific Lutheran University
Spring 2017



Homework

- *Already read the book from Page 1 to Page 12.*
- *Read book to Page 22.*
- *Project 2 due date?*

Announcement

- *Project 2 demo and deadline extended to April 25 (next Monday).*
- *Invited talk this Thursday (April 20).*

Demo of R

- *Setting work directory and edit R code*
- *Demo: `ls()`, `dir()`, `getwd()`, `setwd()` in windows*
- *Use Atom to edit R code in Windows*

Get familiar with the R environment

- *Practice:*
- *ls()*
- *dir()*
- *getwd()*
- *setwd() in windows*
- *Use Atom to edit R code in Windows*

Getting help

- *Try to find answer by contacting Dr. Cao*
- *Try to find answer by searching the web*
- *Try to find answer by reading the manual*
- *Try to find answer by reading the FAQ*
- *Try to find answer by inspection or experimentation*
- *Try to find answer by asking a skilled friend*
- *Try to find answer by reading the source code*

Getting help



missing observations in cov/cor



All

Images

News

Shopping

Videos

More

Settings

Tools

About 265,000 results (0.53 seconds)

[\[R\] Error in cor.default\(x1, x2\) : missing observations in cov/cor](#)

<https://stat.ethz.ch/pipermail/r-help/2008-February/155523.html> ▾

Feb 28, 2008 - Previous message: [\[R\] Error in cor.default\(x1, x2\) : missing observations in cov/cor](#);

Next message: [\[R\] Error in cor.default\(x1, x2\) : missing ...](#)

[\[R\] error message when using mice: missing observations in cov/cor](#) Mar 14, 2010

[\[R\] Null values in R.](#) Dec 3, 2008

[correlation with NA \(was Re: \[R\] \(no subject\)\)](#) Mar 16, 2005

[\[R\] na.rm in sd\(\)](#) Jan 29, 2003

[More results from stat.ethz.ch](#)

[R help - Error in cor.default\(x1, x2\) : missing observations in cov/cor](#)

r.789695.n4.nabble.com/Error-in-cor-default-x1-x2-missing-observations-in-cov-cor... ▾

Feb 27, 2008 - 13 posts - 5 authors

Error in cor.default(x1, x2) : missing observations in cov/cor. Hello, I'm trying to do cor(x1,x2) and I get the following error: Error in cor.default(x1, ...

[\[R\] Error in cor.default\(x1, x2\) : missing observations in cov/cor ...](#)

[grokbase.com > Groups > R > r-help > February 2008](#) ▾

Feb 28, 2008 - "Clearly" you must be using a non-standard cor. As Peter Dalgaard pointed out, cor() is not generic and there is no such function as

[maanova: missing observations in cov/cor - Bioconductor Support](#)

<https://support.bioconductor.org/p/14061/> ▾

Treatment<-matest(myData.glowess, model=model.mixed.2,term="Treatment",n. perm=100) dies with:
Error in var(log(Chis)) : missing observations in cov/cor In ...

Getting help

- *What steps will reproduce the problem?*
- *What is the expected output?*
- *What do you see instead?*
- *What version of R do you use?*
- *What operating system?*
- *What other information?*

Getting help

- *Stupid: “Help! Cann’t fit linear model”*
- *Smart: “R 3.0.2 lm() function produces seg fault with large data frame, MAC OS X 10.9.1”*
- *Smarter: “R 3.0.2 lm() function on MAC OS X 10.9.1 – seg fault on large data frame”*

R console input and evaluation

- *<- as assignment operator*
- *# indicate comment*

R console input and evaluation

- `> x <- 1`
- `> print(x)`
- `> x`

- `> x <- 10` *# assign 10 to variable x*

R console input and evaluation

- *When a complete expression is entered, it is evaluated and the result of the evaluated expression is returned.*
- *> x <- 1 ## numeric, nothing printed*
- *> print(x) ## explicit printing*
- *> x ## auto-printing occurs*

- *> y <- 1.5 + 2 ## numeric number*
- *> print(y)*
- *> y*

Practices

- *Create and print a new variable 'b' with value 2017?*
- *Calculate the following formula and save it to variable C:*

$$88 * 75 + 25 / 20$$

print out the value of C

R data types: Objects and Attributes

R has five basic or “atomic” classes of objects:

- numeric (real numbers) - integer
- character
- complex
- logical (True/False)

Numeric and integer

- Decimal values are called **numerics** in R

```
> x <- 10.5    # assign a decimal value
> x           # print the value of x
[1] 10.5
> class(x)    # print the class name of x
[1] "numeric"
```

- If you explicitly want an **integer**, you need to specify the L suffix. So entering 1 in R gives you a numeric object; entering 1L explicitly gives you an integer object

```
> x <- 30L     # assign a integer value
> x           # print the value of x
[1] 30
> class(x)    # print the class name of x
[1] "integer"
```

Numeric and integer

- Inf represents infinity. e.g. $1/0$ is Inf, $1/\text{Inf}$ is 0
- NaN represents an undefined value (“not a number”); e.g. $0/0$; NaN can also be thought of as a missing value

```
> x <- Inf    # assign a infinity value
> x          # print the value of x
[1] Inf
> y <- 0/0
[1] "NaN"
```


Attributes

- R objects can have attributes, which are like metadata for the object.

Some examples of R object attributes are

- names, dimnames
- dimensions (e.g. matrices, arrays)
- class (e.g. integer, numeric)
- length
- other user-defined attributes/metadata

attributes function to access object's attributes.

Characters

- A **character** object is used to represent string values in R.

```
> c <- "Hello World" # assign a character value
> c                  # print the value of c
```

```
> class(c)          # print the class of c
```

- Two character values can be concatenated with the **paste** function.

```
> c1 <- "Hello"
> c2 <- "World"
```

```
> c <- paste(c1,c2) # paste(c1,c2, sep = "")
> print(c)
```

- Extract a substr ? `substr(string,start=*,stop=*)`

```
> c <- "Hello World" # assign a character value
> substr(c,start=0,stop=3)
```

Complex

- A **complex** value in R is defined via the pure imaginary value i .

```
> c <- 1 + 2i      # assign a complex value
> c               # print the value of c

> class(c)        # print the class of c
```

Logical

- A **logical** value is often created via comparison between variables.

```
> x <- 1
> y <- 2
> z = x>y
> print(z)

> class(z)
```

- Standard logical operations are "&" (and), "|" (or), and "!" (negation).

```
> x <- (1>2)
> y <- (2>1)
> z = x&y
> print(z)
```

Practices

- `x <- 2 * 3`
- `x <- 2L + 3L`
- `x <- (100+200 > 300)`
- `x <- paste("aaa", "bb")`
- `x <- (5 > 2) | (6 < 5)`

- Try it by yourself!

Vectors and Lists

- The `c()` function can be used to create vectors of objects by concatenating things together.

```
> x <- c(0.5, 0.6)    ## numeric
```

```
> x <- c(1, 3)        ## integer
```

```
> x <- c(TRUE, FALSE) ## logical
```

```
> x <- c(T, F)        ## logical
```

```
> x <- c("a", "b", "c") ## character
```

```
> x <- 2:13           ## integer
```

```
> x <- c(1+0i, 2+4i) ## complex
```

```
> x[0]               # print the class type of x
```

```
> class(x)           # show the class of variable x
```

```
> x[1]               # print the first element of x
```

Mixing Objects

There are occasions when different classes of R objects get mixed together.

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

```
> y <- c(TRUE, 2) ## numeric
```

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



character

complex

numeric

integer

logical

Vectors and Lists

- `> x <- c(1,3,5,6,7)`
- `> y <- c(2,5,7,3,6)`
- `> x - y`
- `> x + y`

Subsetting a vector

- `> x <- c(1,3,5,6,7,9,10)`
- `> x[2:4]`

Vectors and Lists

- `>x <- c("abc", 10)`
- `>x[2]+3`

- `> as.numeric(x)` # convert x to numeric class
- `> as.integer(x)` # convert x to integer class
- `> as.logical(x)` # convert x to logical class
- `> as.character(x)` # convert x to character class

- `> as.numeric(x[2]) + 3` # convert to numeric class

Vectors and Lists

Lists are special type of vector that contain elements of different classes.

- `x <- list("abc", 10)`
- `> x[[2]] + 3`

- `> x<-c("abc", 10)`
- `> as.numeric(x[2]) + 3`

Vectors and Lists

Getting help in R

- `> ?print`
- `> help(print)`

Simple examples

1. Create a vector `v`, and add two elements: “hello”, 133
2. Print the second element of `v`
3. Convert the second element of `v` to character
4. Setup your working directory to a new 'work' folder in your desktop
5. Create a vector numbers from 1 to 6 and find out its class
6. Create a vector containing following mixed elements {1, 'a', 2, 'b'} and find out its class. Then create a list with the same elements.
7. Get the first two elements from above vector
8. Get the second and third elements from above vector

Exercise

Use source function to run a R script: (getwd, setwd)

```
> source("test.R")
```

In R editor or Atom, write test.R, use source to run that file. Create vector, and put your name or your friend names as the element, and also add your age as element, print out the vector, get familiar with environment setting in R.