

R - Sommersemester 2007

Oliver Kirchkamp*

17. Juli 2007

*Universität Jena, D-07743 Jena, email oliver@kirchkamp.de

Homepage:

<http://www.kirchkamp.de/>

Literature:

- Venables, Smith: An Introduction to R
- Verzani: Simple R
- Farnsworth: Econometrics in R

Work environment:

- R-commander
- Emacs + ESS

```
library(Rcmdr)
```

Why R:

- free
- extensive number of functions
- not all functions implement all features in a consistent way
- the language itself is rather consistent

1 R-language

1.1 Data

- vector
 - numeric
 - factor
 - character
 - logical
- matrix
- data.frame
- list
- function

1.2 Operators

- = <- ->
- + - * / ^
- :
- %*%

```
7
a=7
a
b<-7
b
b[1]
b[2]
b[3]
b[3]=21
b*3
7+3
7*3
7-3
7/3
7^3
7:3
3:7
```

1.3 Functions

- log(10); pnorm(1); log(10,base=2); log(10,2);

1.4 Getting Help

```
help(mean)
help.search("mean")
mean(b)
mean(b,na.rm=TRUE)
? mean
help.start()
help(mean)
```

1.5 grouping commands

```

...
if (...) { ... ; ... ;
  ...
  ... ; ...
}
...

```

1.6 Assignments, concatenation, and recycling

```

b = 7
b[3]=21
b
#
x = c(7,NA,21)
x
x*x
y = c(x,0,x*x)
y
rbind(x,0,x*x)
cbind(x,5,x*x)
cbind(x,5:6,x*x)
cbind(x,1:6,x*x)

```

1.7 Logical vectors

```

b
b>10

```

1.8 Index vectors

```

b=c(50,3,16,6,8)
order(b)
b[1]
b[c(1,3)]
b[order(b)]
#
b

```

```
b>10
b[b>10]
```

1.9 Working with matrices

```
b=c(50,3,16,6,8)
cbind(b,b^2,1:5)
x=cbind(b,b^2,1:5)
x
x[1,1]
x[1,]
x[,1]
x[3:4,c(2,3)]
colnames(x)
colnames(x)[2]="b^2"
colnames(x)[3]="index"
#
x
x[,"b"]
x[,"b^2"]
x[,c("b","b^2")]
```

1.10 Mixing numbers and other datatypes

Mixing numbers and characters in a matrix does not work:

```
c("a","b","c","d","e")
cbind(x, c("a","b","c","d","e"))
y=cbind(x, c("a","b","c","d","e"))
y[,"b"]
y[,"b"]*2
```

data.frames help here

```
cbind(as.data.frame(x), c("a","b","c","d","e"))
y=cbind(as.data.frame(x), c("a","b","c","d","e"))
y[,"b"]
y$b*2
```

```
y$q = y$b + 3  
y
```

2 Working with datasets and files

2.1 Accessing built-in data

```
library(Ecdat)  
data(Wages)  
edit(Wages)  
Wages[, "lwage"]  
Wages[1,]
```

2.2 Writing and reading CSV data

```
write.table(Wages, file="Wages.csv")  
ls()  
rm(Wages)  
ls()  
Wages  
read.table(file="Wages.csv")  
W=read.table(file="Wages.csv")  
edit(W)
```

2.3 Writing and reading R images

```
Wages  
save(Wages, file="Wages.R")  
rm(Wages)  
Wages  
load("Wages.R")  
Wages
```

2.4 Writing and reading foreign formats

```
library(foreign)
write.dta(W,file="Wages.dta")
```

```
<Stata>
use Wages
desc
```

```
read.dta(file="Wages.dta")
Wages=read.dta(file="Wages.dta")
```

2.5 Writing and reading parts of data

```
Wages[1,]
WF = Wages[Wages$sex=="female",]
WM = Wages[Wages$sex!="female",]
dim(WF)
dim(WM)
WFM=merge(WF,WM,all=TRUE)
dim(WFM)
dim(Wages)
```

2.6 More data manipulation

```
library(Ecdat)
data(Wages)
summary(Wages)
table(Wages$sex)
table(Wages$sex,Wages$ed)
with(Wages,table(sex,ed))
attach(Wages)
table(sex,ed)
detach(Wages)
#
with(subset(Wages,sex=="female"),table(ed))
```

3 Testing and estimation

3.1 Testing

```
library(Ecdat)
data(Wages)
summary(Wages)
#
t.test(Wages$lwage ~ Wages$sex)
wilcox.test(Wages$lwage ~ Wages$sex)
```

3.2 Simple regressions

```
lm(Wages$lwage ~ Wages$ed)
est=lm(Wages$lwage ~ Wages$ed)
est
summary(est)
confint(est)
```

3.3 Different ways to specify datasets

```
lm(Wages$lwage ~ Wages$ed)
with(Wages,lm(lwage ~ ed))
lm(lwage ~ ed, data=Wages)
#
lm(lwage ~ exp,data=Wages,subset=(south=="yes"))
lm(lwage ~ exp,data=Wages,subset=(south=="no"))
#
with(subset(Wages,south=="no"),lm(lwage ~ exp))
with(subset(Wages,south=="yes"),lm(lwage ~ exp))
#
attach(Wages)
lm(lwage ~ ed)
lm(lwage ~ exp,subset=(south=="yes"))
detach(Wages)
```

3.4 Regression equations

```
lm(lwage ~ ed)
lm(lwage ~ ed + sex)
lm(lwage ~ ed * sex)
lm(lwage ~ ed * sex + I(ed^2))
est=lm(lwage ~ ed * sex + I(ed^2))
est
summary(est)
```

3.5 Linear hypotheses

```
library(car)
linear.hypothesis(est,"sexmale=0")
linear.hypothesis(est,c("sexmale=0","ed:sexmale=0"))
coef(est)
coef(est) %*% c(0,0,1,0,30)
linear.hypothesis(est,"sexmale = - 30*ed:sexmale")
linear.hypothesis(est,"sexmale = - 30*ed:sexmale",vcov=hccm)
```

4 Plots

```
library(Ecdat)
data(Wages)
attach(Wages)
hist(lwage)
hist(lwage,breaks=50)
#
library(lattice)
histogram(~ lwage | sex)
densityplot(~ lwage | sex)
stripplot(lwage ~ sex)
boxplot(lwage ~ sex)
#
plot(density(lwage))
plot(ecdf(lwage))
#
plot(ed,lwage)
```

```

plot(lwage ~ ed)
(est=lm(lwage ~ ed))
abline(est)
abline(est,col="red")
(est=lm(lwage ~ ed + I(ed^2)))
points(est$fitted.values ~ ed,col="green")
lines(est$fitted.values ~ ed,col="green")
o = order(ed)
lines(est$fitted.values[o] ~ ed[o],col="blue")
detach(Wages)

```

5 Distributions and simulating stuff

```

dnorm(.3)
plot(dnorm,-2,2)
plot(pnorm,-2,2)
rnorm(1)
rnorm(10)
plot(rnorm(1000),rnorm(1000))
plot(runif(1000),rnorm(1000))

```

Example: Multicollinearity

```

x = runif(1000)
y = 2 * x + 7.3
lm (y ~ x)
#
u = rnorm(1000)
y = 2 * x + 7.3 + u
lm (y ~ x)
#
x2 = 1-x
lm ( y ~ x2 + x)
#
u2 = rnorm(1000,sd=.1)
x2 = 1 - x + u2
lm ( y ~ x2 + x)

```

Let us assume that we are interested in investigating the common distribution of the factors for x and x_2

```
size = 100
beta=array(0,dim=c(size,2))
for (i in 1:size) {
  u1 = rnorm(1000,sd=.01)
  u2 = rnorm(1000,sd=.01)
  x1 = x + u1
  x2 = 1 - x + u2
  est=lm ( y ~ x1 + x2)
  beta[i,]=est$coef[2:3]
}
plot(beta)
```

Not let us make this a function:

```
myEst = function (a) {
  -a
  a*a
}
myEst(7)
#
myEst = function (a) {
  b=2*a
  palim(b)
}
#
myEst(7)
options(error=recover)
myEst(7)
myEst = function(size=200,sd1=.01,sd2=.02) {
  beta=array(0,dim=c(size,2))
  for (i in 1:size) {
    u1 = rnorm(1000,sd=sd1)
    u2 = rnorm(1000,sd=sd2)
    x1 = x + u1
    x2 = 1 - x + u2
    est=lm ( y ~ x1 + x2)
    beta[i,]=est$coef[2:3]
```

```

}
beta
}
plot(myEst(size=100,sd2=1,sd1=1))

```

5.1 Doing things for several subsets

```

size=1000
x = runif(size)
u = rnorm(size)
y = 2*x + 7 + u
i = floor(runif(size,min=0,max=10))
aggregate(x,list(i),mean)
aggregate(cbind(x,u,y),list(i),mean)

```

The function in aggregate can be more complicated

```

aggregate(cbind(x,u,y),list(i),function(x) mean(x))
aggregate(cbind(x,u,y),list(i),function(x) mean(x))
aggregate(cbind(x,u,y),list(i),function(x) {z=mean(x);1/z;})

```

To do more complicated things we need by

```

by(cbind(x,u,y), i, function(x) lm(y ~ x,data=x))
est=by(cbind(x,u,y), i, function(x) lm(y ~ x,data=x))
sapply(est,coef)

```