# Econometrics 1 *Applied Econometrics with R*

## Lecture 3: Basics of R (2)

---


黄嘉平

中国经济特区研究中心 讲师

办公室：文科楼1726

E-mail: huangjp@szu.edu.cn

Tel: (0755) 2695 0548

Office hour: Mon./Tue. 13:00-14:00

# Matrix

- Create a matrix from a vector

```
> x <- 1:12
> y <- matrix(x, nrow = 3, ncol = 4)
```

- Elements of a matrix

```
> y[2, 3]
> y[2,]
> y[, 4]
> y[3, c(1,3)]
```

# Basic matrix algebra

- Try `+, -, *, /, ^, sqrt(), exp(), log()` with a matrix
- Create a matrix $z = \begin{pmatrix} 1 & 5 \\ 2 & 6 \end{pmatrix}$
- Matrix multiplication

```
> z %*% z
```

- Transpose

```
> t(z)
```

# Basic matrix algebra

- Solving an equation of the form

$$A \times X = B \quad \text{or} \quad A \times \boldsymbol{x} = \boldsymbol{b}$$

- Use

```
> solve(A, B)
```

- Try to solve the following problem

“今有雉兔同笼，上有三十五头，下有九十四足，问雉兔各几何？”

——《孙子算经》

# Basic matrix algebra

- Inverse

```
> solve(z)
```

Since $A \times A^{-1} = I$ , $A^{-1}$ can be seen as the solution of equation $A \times X = B$ where $B = I$ . `solve()` returns the solution of this type of equations. See the help document and learn its usage using `?solve`

- Check your result

```
> z %*% solve(z)
```

# Other matrix operations

- Size of a given matrix: `dim(), nrow(), ncol()`
- Patterned matrix — the identity matrix

```
> diag(4)
```

- Combining matrices

```
> cbind(z, diag(2))
> rbind(z, solve(z))
```

# Write your codes: script files and projects

# Save your code in a script file

- One advantage of R is that it is interactive, but you also may end up with a chaos after a long time working with it.

- It is **highly recommended** that you save your script

- An R script is a text file with extension .R

- Select “File > New File > R Script” from the menu bar of RStudio to create an R script file. It opens a panel.

- Write your commands (now it is a program) and save.

- Use `source()` to run a script file.
  You can also select some lines and click “Run”.

# Practice

- Matrix multiplication is not communicative, i.e.,

$$AB \neq BA$$

  Let us check this through an example.

- Let $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $B = \begin{bmatrix} 0 & -1 \\ 6 & 7 \end{bmatrix}$,

  create a new script file, write your program to calculate $AB$ and $BA$, and save your code.

# Practice

```
> A <- matrix(c(1, 3, 2, 4), 2, 2)
> B <- matrix(c(0, -1, 6, 7), 2, 2, brow =
TRUE)
> A %*% B
     [,1] [,2]
[1,]   12   13
[2,]   24   25
> B %*% A
     [,1] [,2]
[1,]   -3   -4
[2,]   27   40
```

# Absolute path and relative path

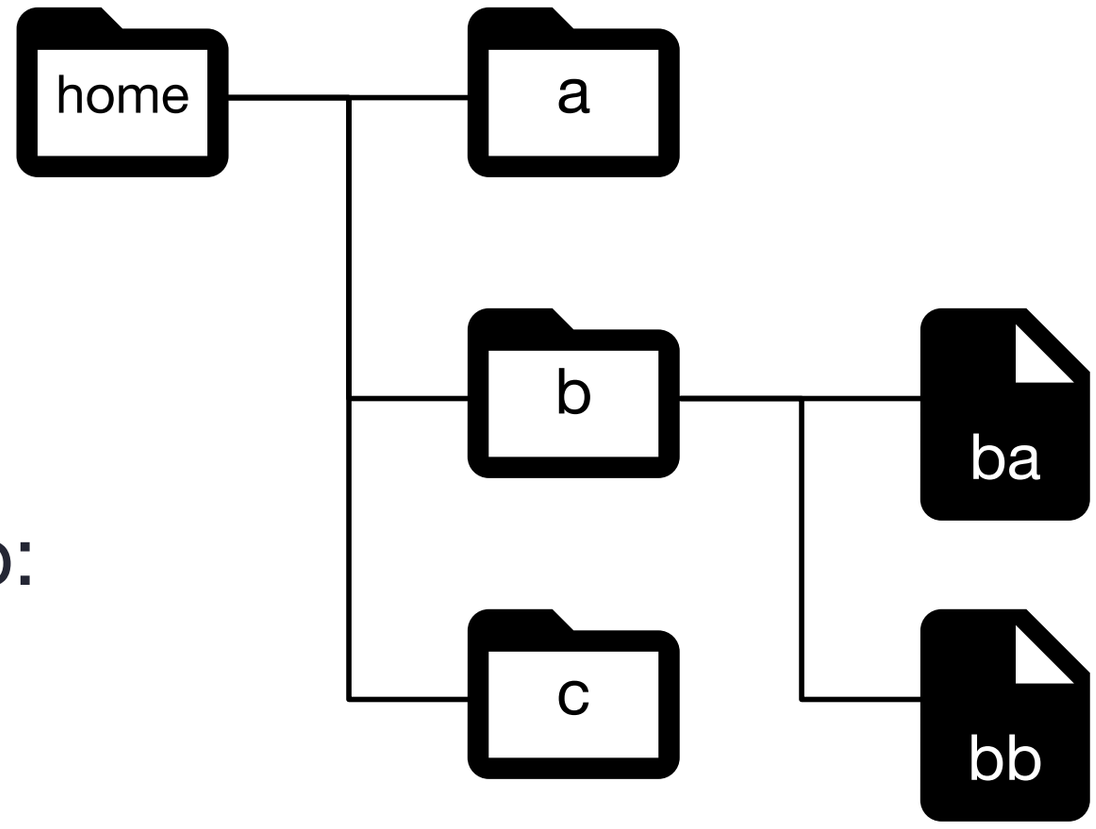
The path of file “ba”

Absolute path:
“home/b/ba”

Relative path if you are at folder b:
“./ba”

Relative path if you are at folder c:
“../b/ba”

Try: find the R program in your computer, and specify its absolute path (and pay attention to “/” and “\”).



# Current directory, and project in RStudio

- The current directory is the default location when you open your R program.

- Use `getwd()` to see your current directory.
  Use `setwd('C://file/path')` to set your current directory.

- Current directory affects the relative path of your scripts.

- Working with RStudio **projects** can make things simple.

# Create a project in RStudio

1. Select “File > New Project…” from the menu bar of RStudio

2. Select “New Directory”, then “Empty Project”

3. “Directory Name”: choose a name, e.g. “AER”

   “Create project as a subdirectory of:”: choose where you want to put your project, e.g. an existing `path`

   Click “Create Project”
   (RStudio will make a new directory “AER” under the `path` you have chosen)

4. When you work with this project, you can put all your data files in the “AER” directory, and use relative paths.

# Data management in R

# Data frame

- A typical way of storing data in R is using data frame

  `> library("AER")`
  if not installed, use `install.packages("AER")`

```
> data("Journals")
> class(Journals)
> View(Journals)
> str(Journals)
> summary(Journals)
> names(Journals)
> rownames(Journals)
```

# The structure of a data frame

`rownames`

`names` or `colnames`

| | title | publisher | society | price | pages | charpp | citations | foundingyear | subs | field |
|---|---|---|---|---|---|---|---|---|---|---|
| **APEL** | Asian-Pacific Economic Literature | Blackwell | no | 123 | 440 | 3822 | 21 | 1986 | 14 | General |
| **SAJoEH** | South African Journal of Economic History | So Afr ec history assn | no | 20 | 309 | 1782 | 22 | 1986 | 59 | Economic History |
| **CE** | Computational Economics | Kluwer | no | 443 | 567 | 2924 | 22 | 1987 | 17 | Specialized |
| **MEPiTE** | MOCT-MOST Economic Policy in Transitional Economics | Kluwer | no | 276 | 520 | 3234 | 22 | 1991 | 2 | Area Studies |
| **JoSE** | Journal of Socio-Economics | Elsevier | no | 295 | 791 | 3024 | 24 | 1972 | 96 | Interdisciplinary |
| **LabEc** | Labour Economics | Elsevier | no | 344 | 609 | 2967 | 24 | 1994 | 15 | Labor |
| **EDE** | Environment and Development Economics | Cambridge Univ Pres | no | 90 | 602 | 3185 | 24 | 1995 | 14 | Development |
| **RoRPE** | Review of Radical Political Economics | Elsevier | no | 242 | 665 | 2688 | 27 | 1968 | 202 | Specialized |
| **EoP** | Economics of Planning | Kluwer | no | 226 | 243 | 3010 | 28 | 1987 | 46 | Area Studies |
| **Mt** | Metroeconomica | Blackwell | no | 262 | 386 | 2501 | 30 | 1949 | 46 | General |
| **JoCP** | Journal of Consumer Policy | Kluwer | no | 279 | 578 | 2200 | 32 | 1978 | 57 | Consumer Economics |

# Extract a subset of data from a data frame

- Column vector

```
> Journals[[4]]
> Journals$price
> Journals[ ,4]
> Journals[ , “price”]
```

- Column slice

```
> Journals[4]
> Journals[“price”]
> Journals[4:5]
> Journals[ , c(“price”, “pages”)]
```

- Row slice

```
> Journals[3, ]
> Journals[c(3, 12), ]
> Journals[c(“CE”, “REE”), ]
```

- A subset

```
> Journals[c(3, 12), 1:5]
```

- A selection with multiple columns is always treated as a data frame

# Create a data frame

- Create data frame with new data

```
> mydata <- data.frame(one = 1:10, two=
11:20, three = 21:30)
> rownames(mydata) <- 101:110
```

- Converting data to data.frame

```
> mymatrix <- matrix(1:30, ncol = 3)
> mydata2 <- as.data.frame(mymatrix)
> names(mydata2) <- c(“one”, “two”, “three”)
```

# Data import: from excel to R

- Get a sample data:

  广东统计年鉴2016
  http://www.gdstats.gov.cn/tjnj/2016/directory.html

  找到你感兴趣的数据，点击“Excel下载”

- Clean your data file and save it to a .csv file.

- Use `read.csv()` to import data from your .csv file.

- Practice!

# Data export

- Use `write.csv()` to save a data frame or a matrix into a .csv file.

- Data can also be saved as a binary file.

```
> save(mydata, file = “mydatafile.RData”)
```

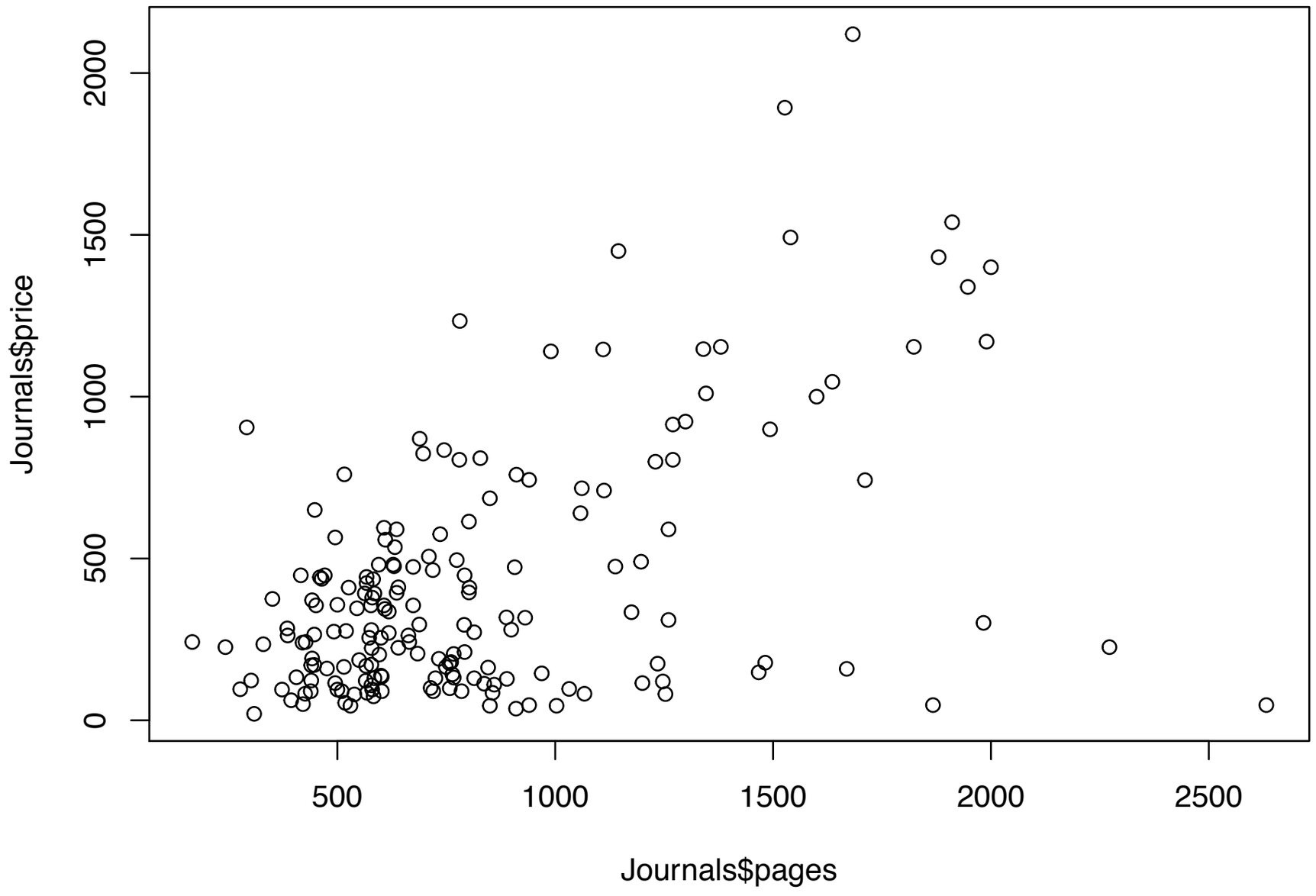
- Load .RData type of data

```
> load(“mydatafile.RData”)
```

# Graphics

# Basic plotting function: `plot()`

- Scatter plot is probably the most common graphical display in statistics.



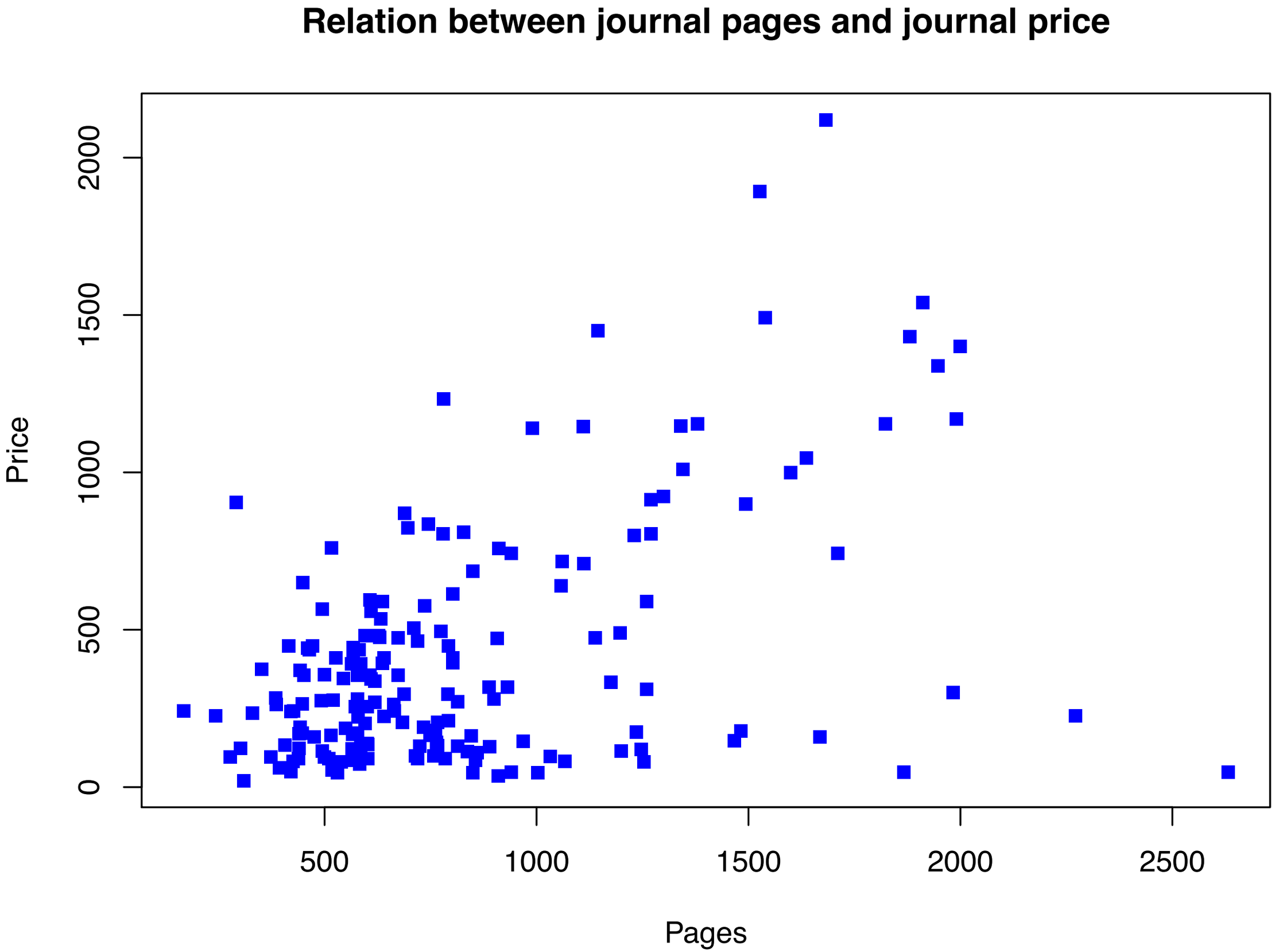
- Horizontal axis x, vertical axis y → `plot(x, y)`

```
> plot(Journals$pages, Journals$price)
```

- Specify graphical parameters

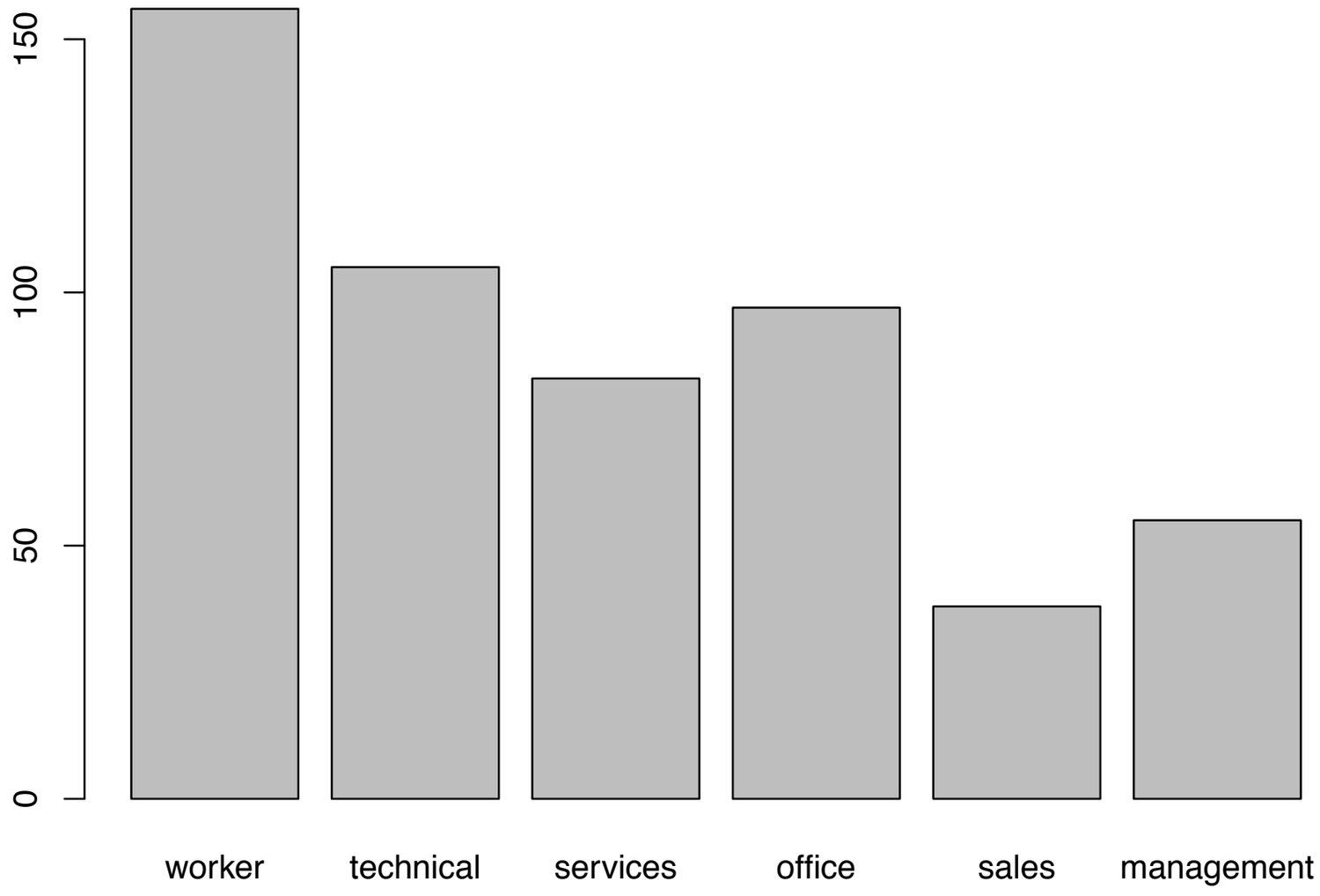
```
> plot(Journals$pages, Journals$price,
main = "Relation between journal pages
and journal price", xlab = "Pages",
ylab = "Price", col = "blue", pch =
15)
```

For further details, use `?plot`


Relation between journal pages and journal price
Price
0
500
1000
1500
2000
500
1000
1500
2000
2500
Pages

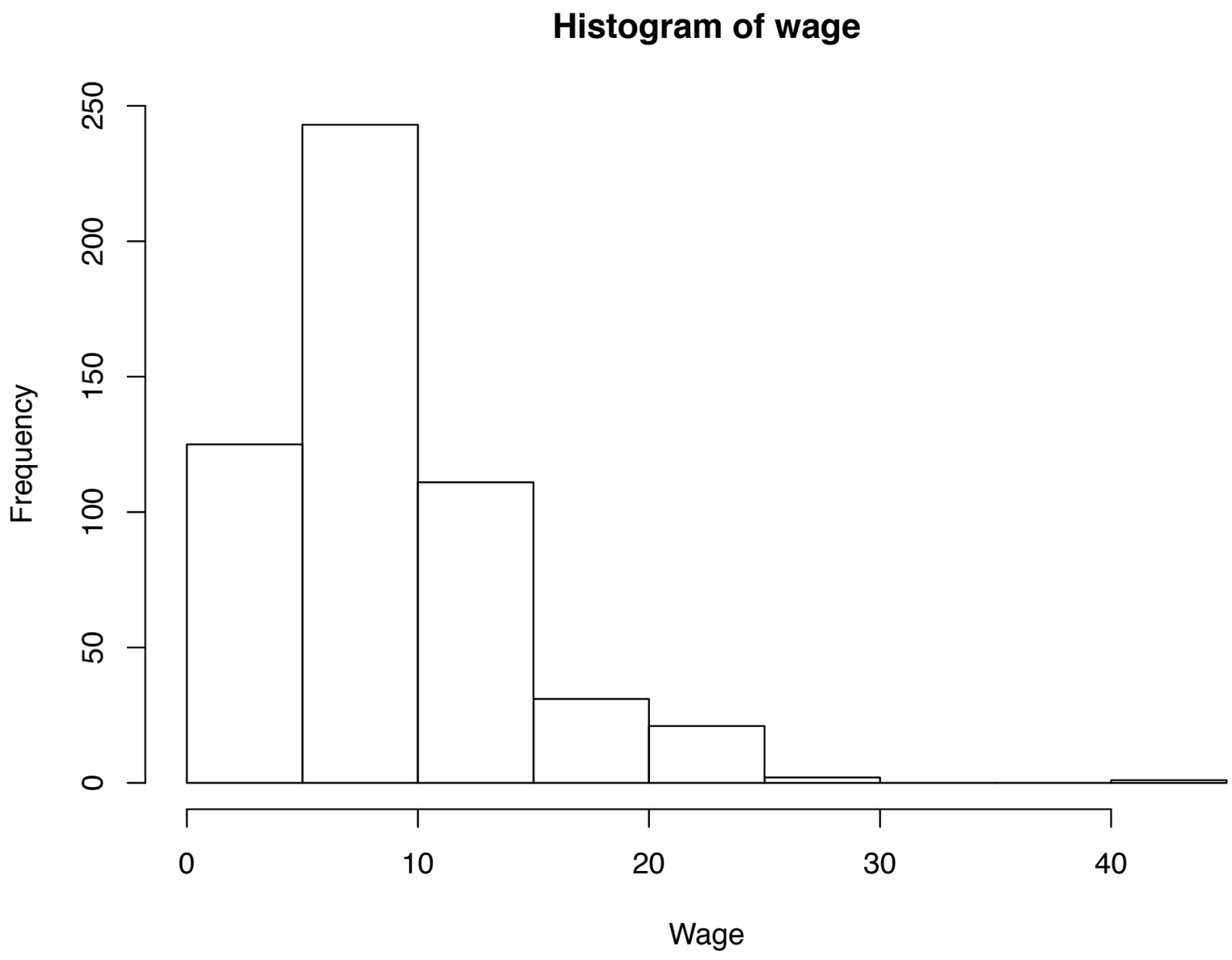
# Bar graph

```
> data(CPS1985, package = "AER")
> barplot(summary(CPS1985$occupation))
```



# Histogram

```
> hist(CPS1985$wage, main = "Histogram of
wage", xlab = "Wage")
```



# Plot a function (1): a general way

- Step 1: specify the domain (with a grid) of your function

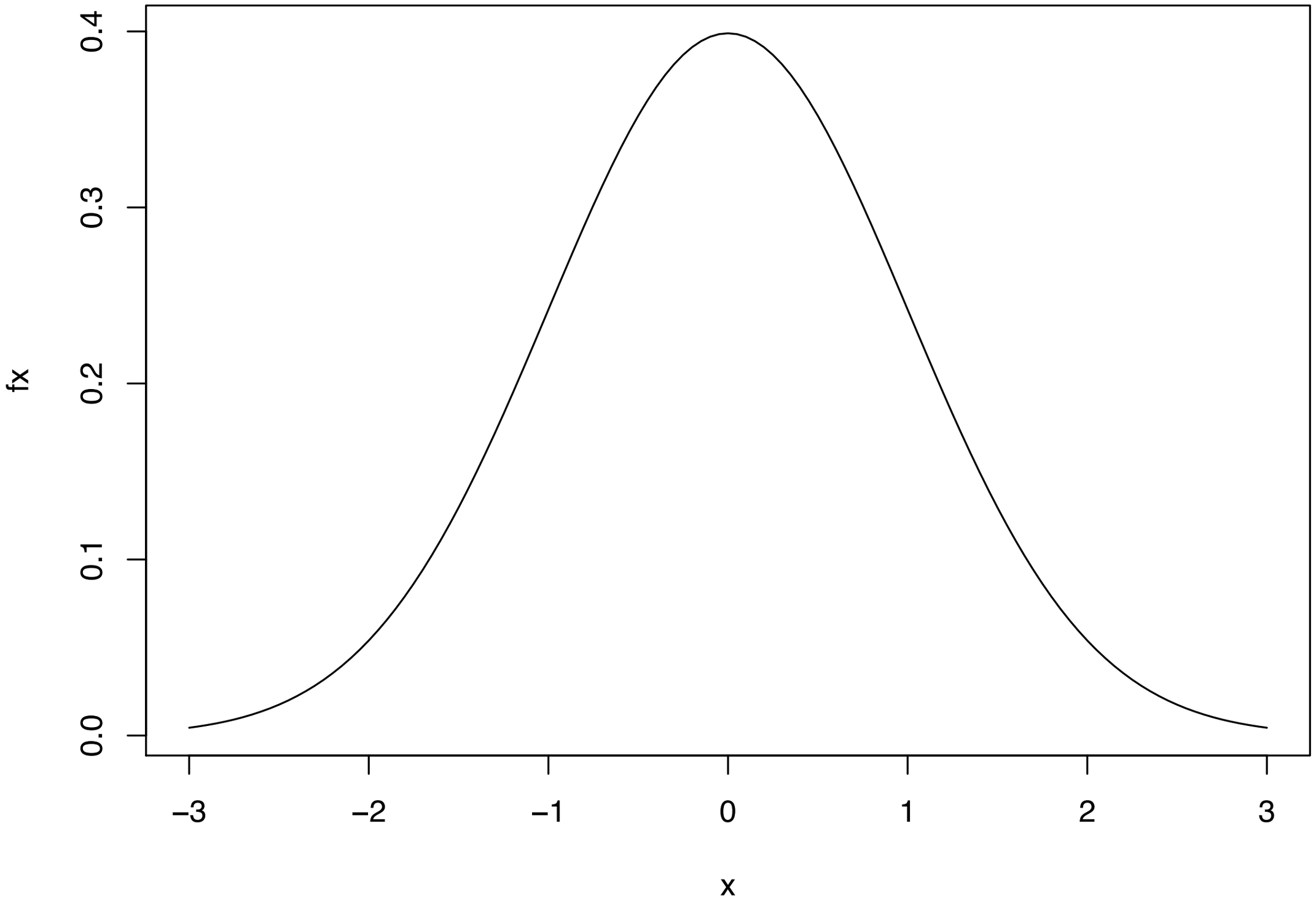
```
> x <- seq(-3, 3, 0.05)
```

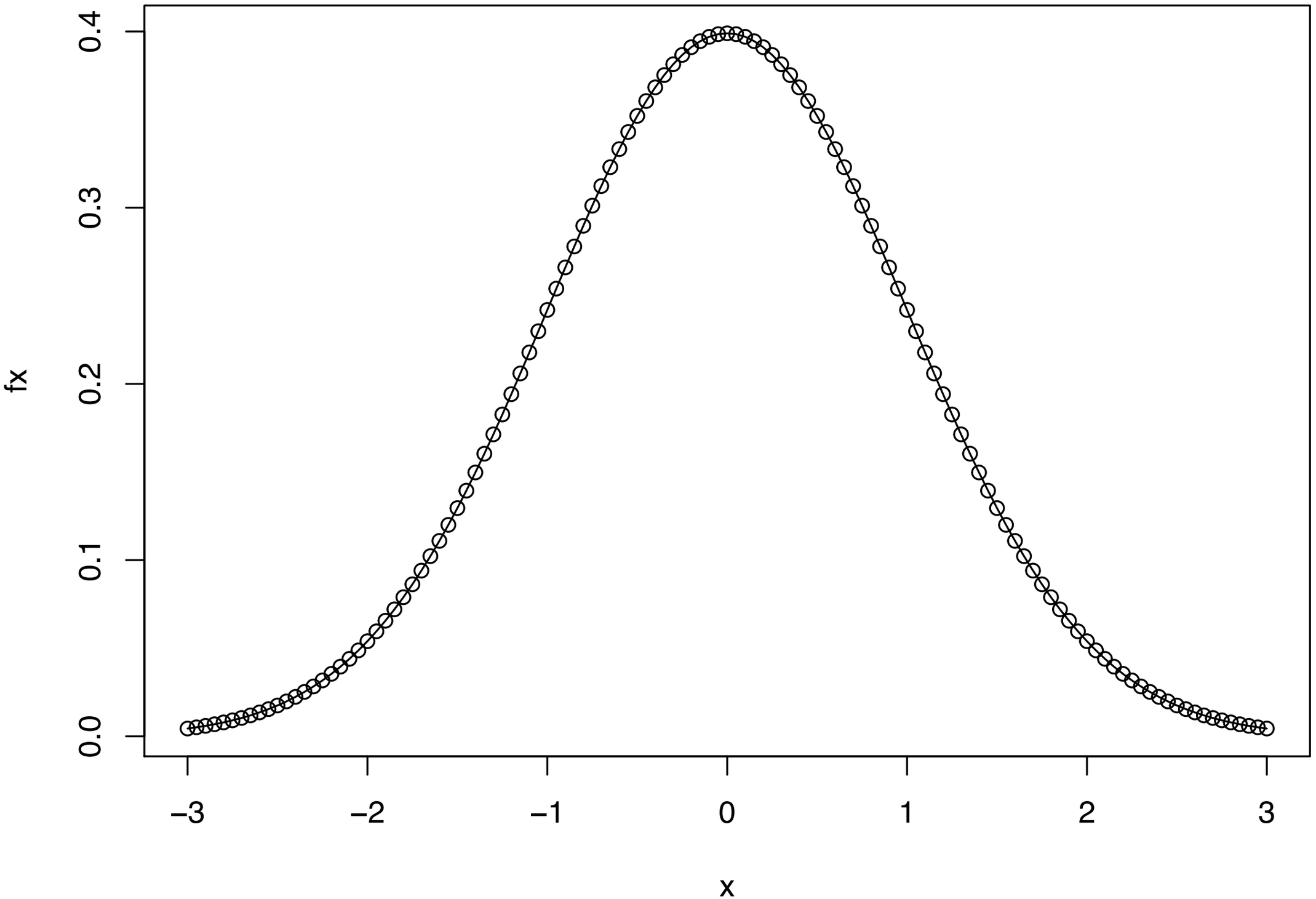
- Step 2: calculate the values over the domain

```
> fx <- exp(-x^2/2) / sqrt(2*pi)
```

- Step 3: plot

```
> plot(x, fx, type = “l")
```


fx
0.0
0.1
0.2
0.3
0.4
−3
−2
−1
0
1
2
3
x


fx
0.0
0.1
0.2
0.3
0.4
−3
−2
−1
0
1
2
3
x

# Plot a function (2): a faster way

- We have plotted the density function of the standard normal distribution.

- This function can be calculated with the built-in command `dnorm()`

- Use `curve()` to draw the function

```
> curve(dnorm, from = -3, to = 3)
```

- Try it!

# References


1. Kleiber, C. and Zeileis, A., *Applied Econometrics with R*, Springer, 2008.

2. Venables, W. N., Smith, D. M., and the R Core Team, *An Introduction to R*. https://cran.r-project.org/manuals.html

3. A base R cheat sheet http://github.com/rstudio/cheatsheets/raw/master/base-r.pdf