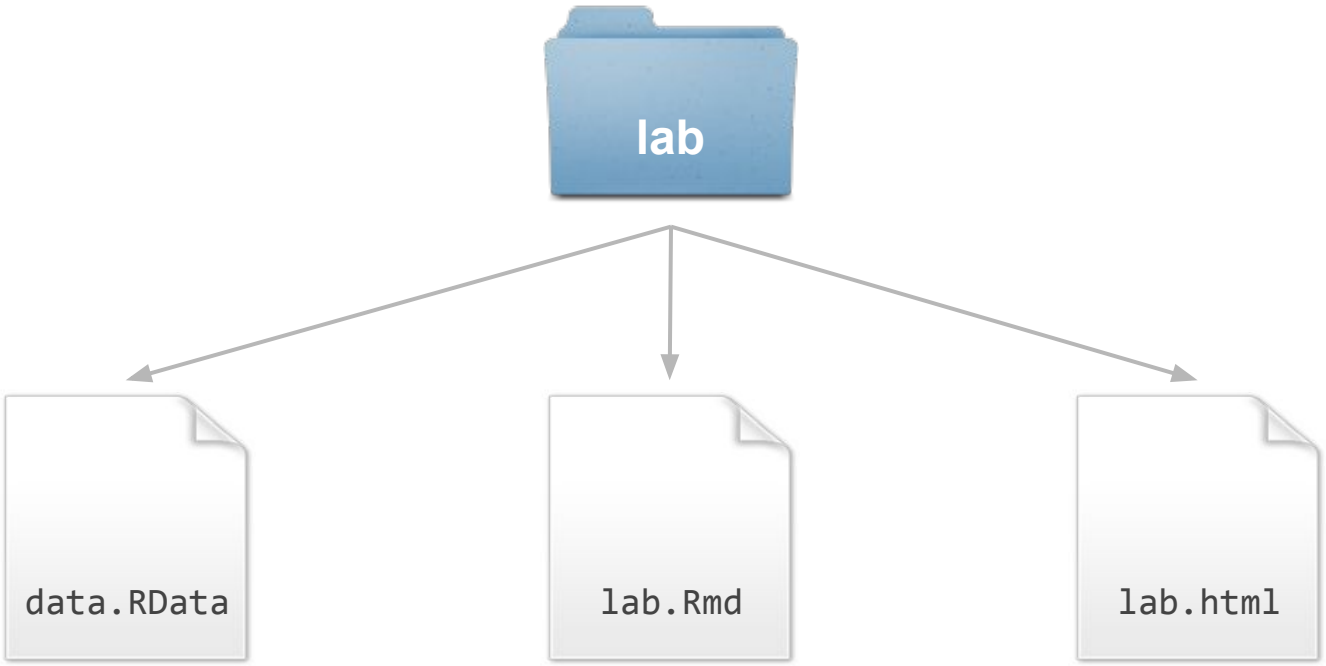
# Filesystem

Stat 133 by Gaston Sanchez

# How does a data analysis project look like from the files standpoint?

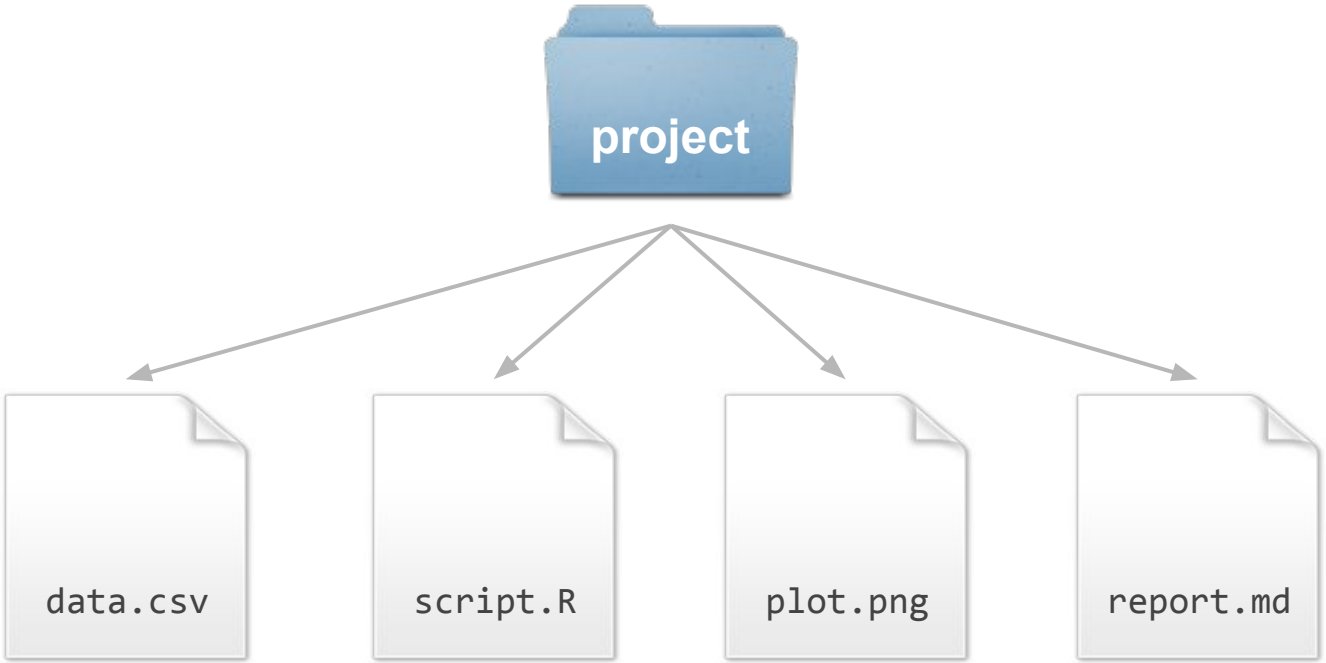
# Lab of last week



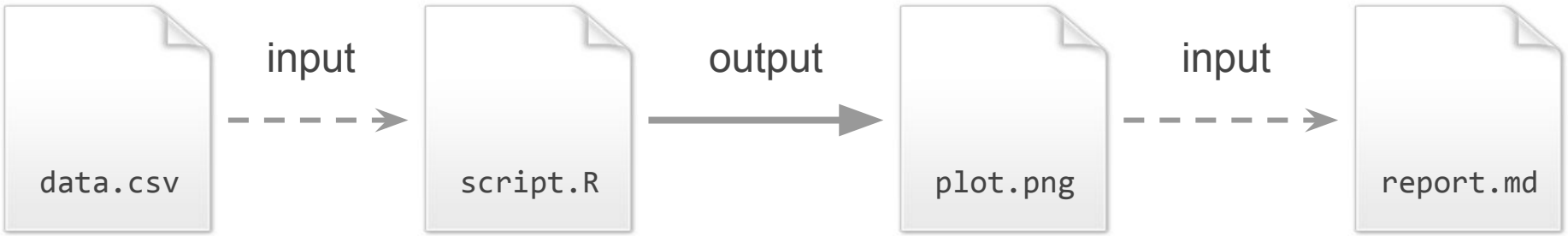
# Relationships among files

# A Toy Project

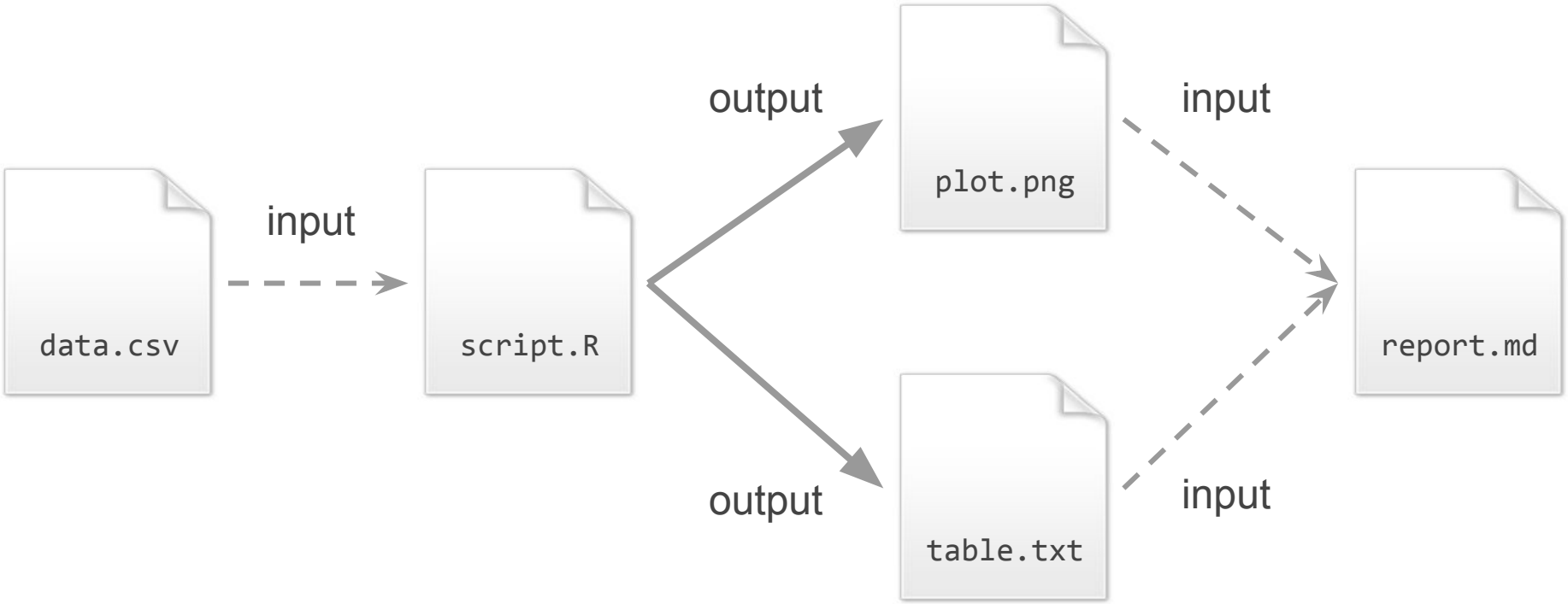
# Basic Project example



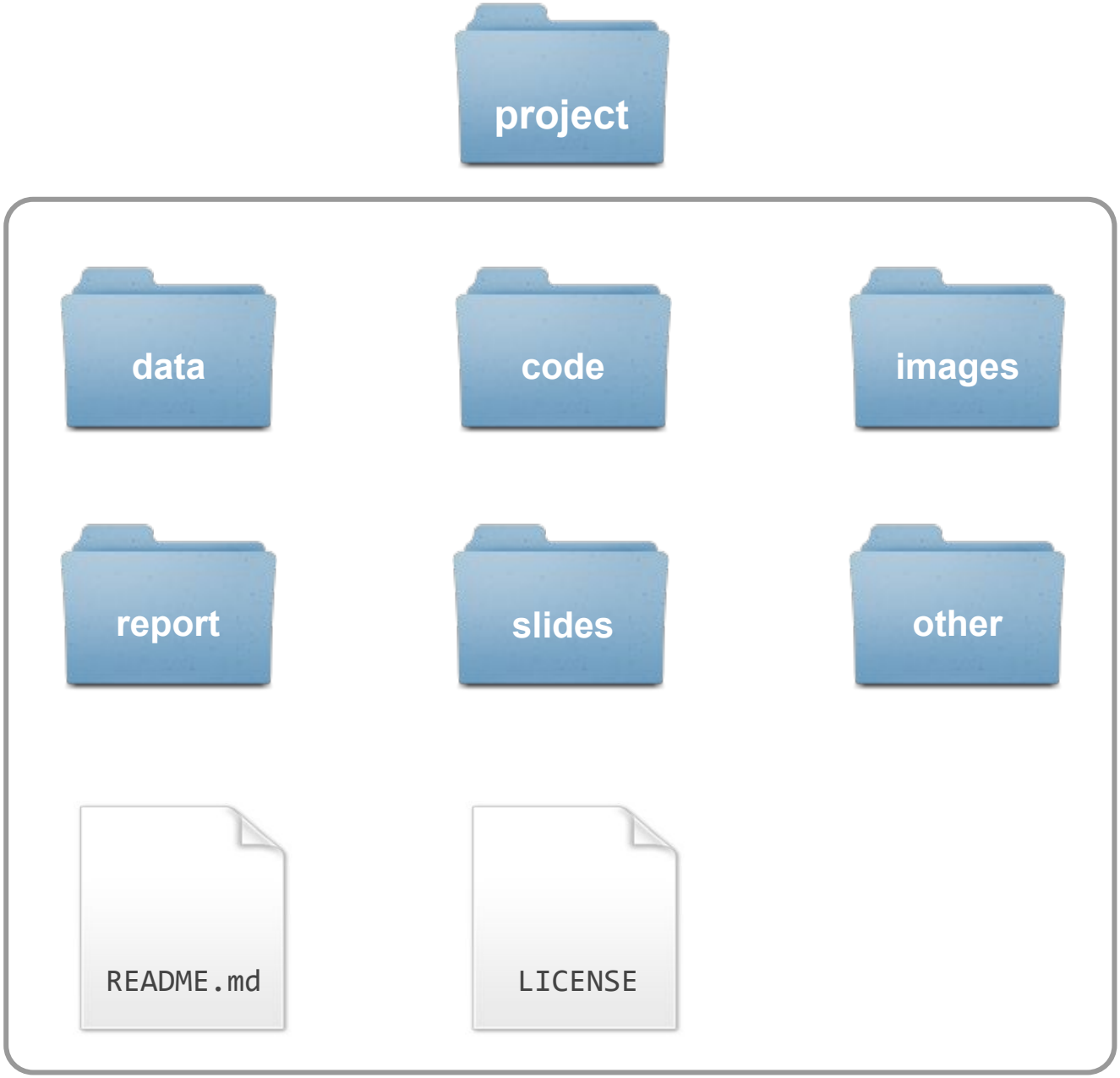
# Relationships among files



# Another example



# A More Complete Project


project
data
code
images
report
slides
other
README.md
LICENSE

# Project File-Structure

```
project/
```

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
```

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
```

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
```

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
  report/
    doc.tex
    doc.pdf
  README.md
```

This is the typical view we have of a project's structure

But there's another way to look at a project's files

There’s a network of relationships among files

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
  report/
    doc.tex
    doc.pdf
  README.md
```

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
  report/
    doc.tex
    doc.pdf
  README.md
```

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
  report/
    doc.tex
    doc.pdf
  README.md
```

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
  report/
    doc.tex
    doc.pdf
  README.md
```

# Project File-Structure

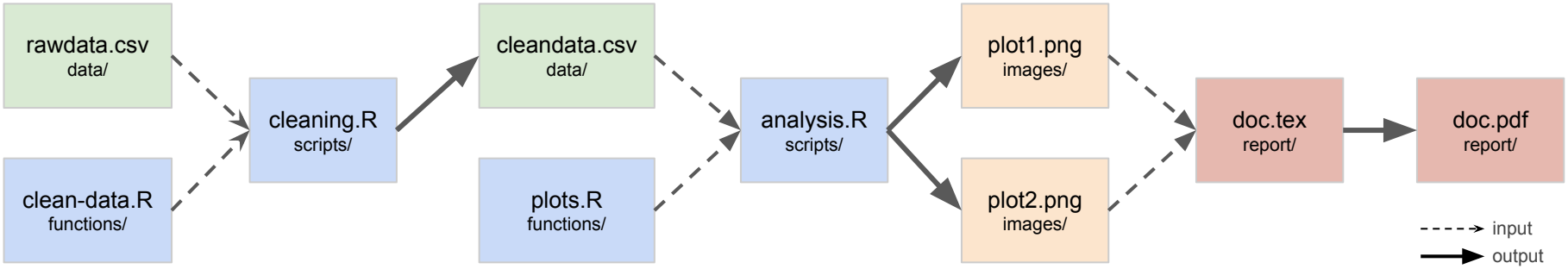
```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
  report/
    doc.tex
    doc.pdf
  README.md
```

# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
  report/
    doc.tex
    doc.pdf
  README.md
```

# Drawing a flow diagram showing the inputs and outputs

# Project Workflow




Files can be inputs
Files can be outputs
Or both

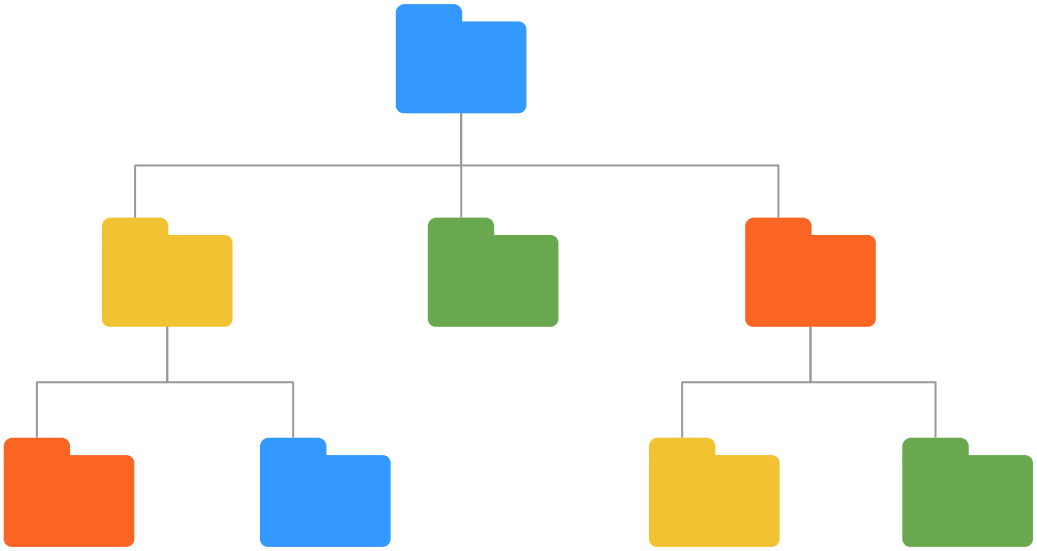
# Project File-Structure

```
project/
  data/
    rawdata.csv
    cleandata.csv
  code/
    functions/
      clean-data.R
      plots.R
    scripts/
      cleaning.R
      analysis.R
  images/
    plot1.png
    plot2.png
  report/
    doc.tex
    doc.pdf
  README.md
```

You'll be working with files (some of them will be inputs, some outputs, some both)


Lesson 1: Learn how to refer to the files in a project

# File System



# Main Unix Concepts

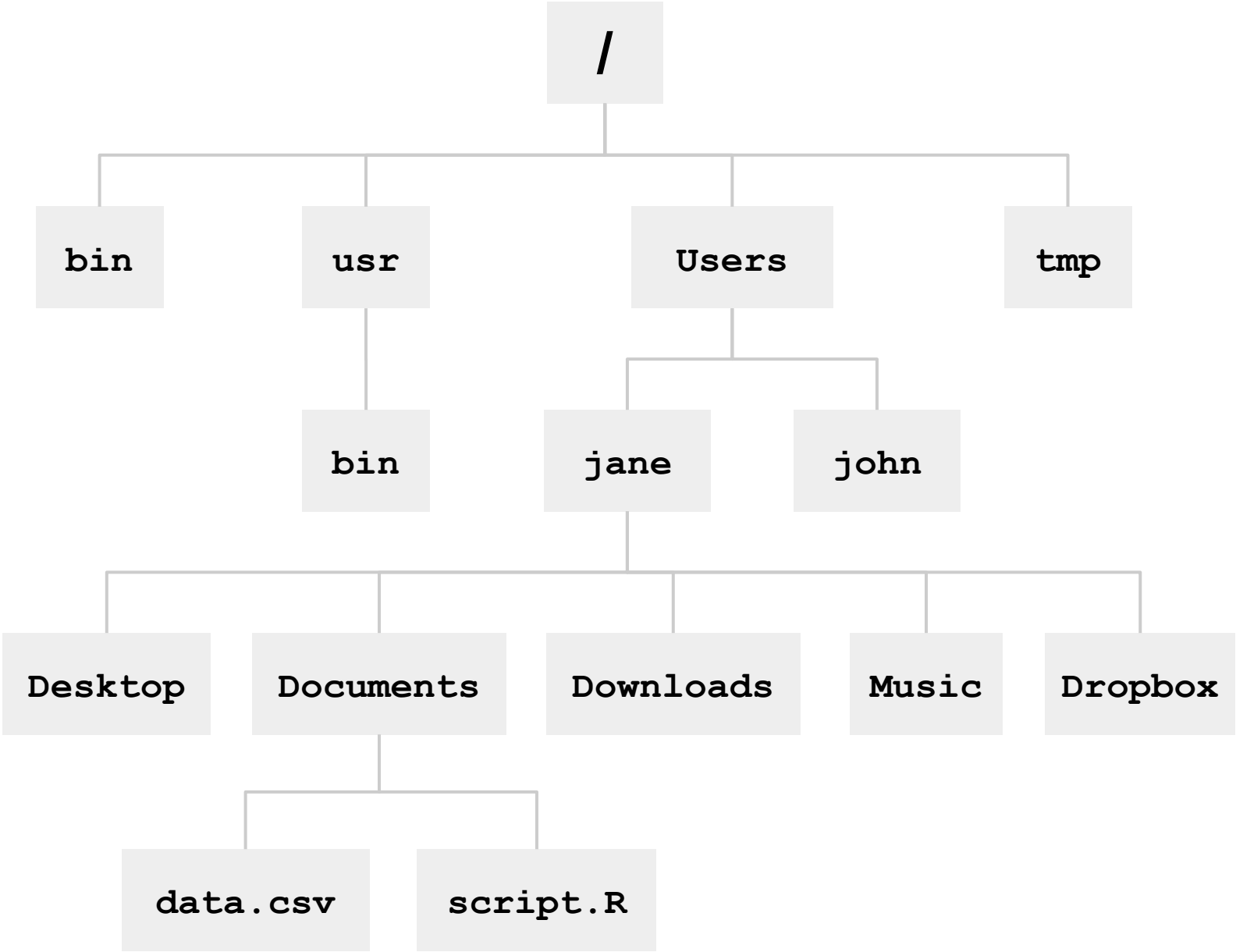
Everything is a **File** (including directories)

Files organized in a tree structure

The filesystem is a hierarchy (of folders & files)

Folder = Directory

No concept of “disk”

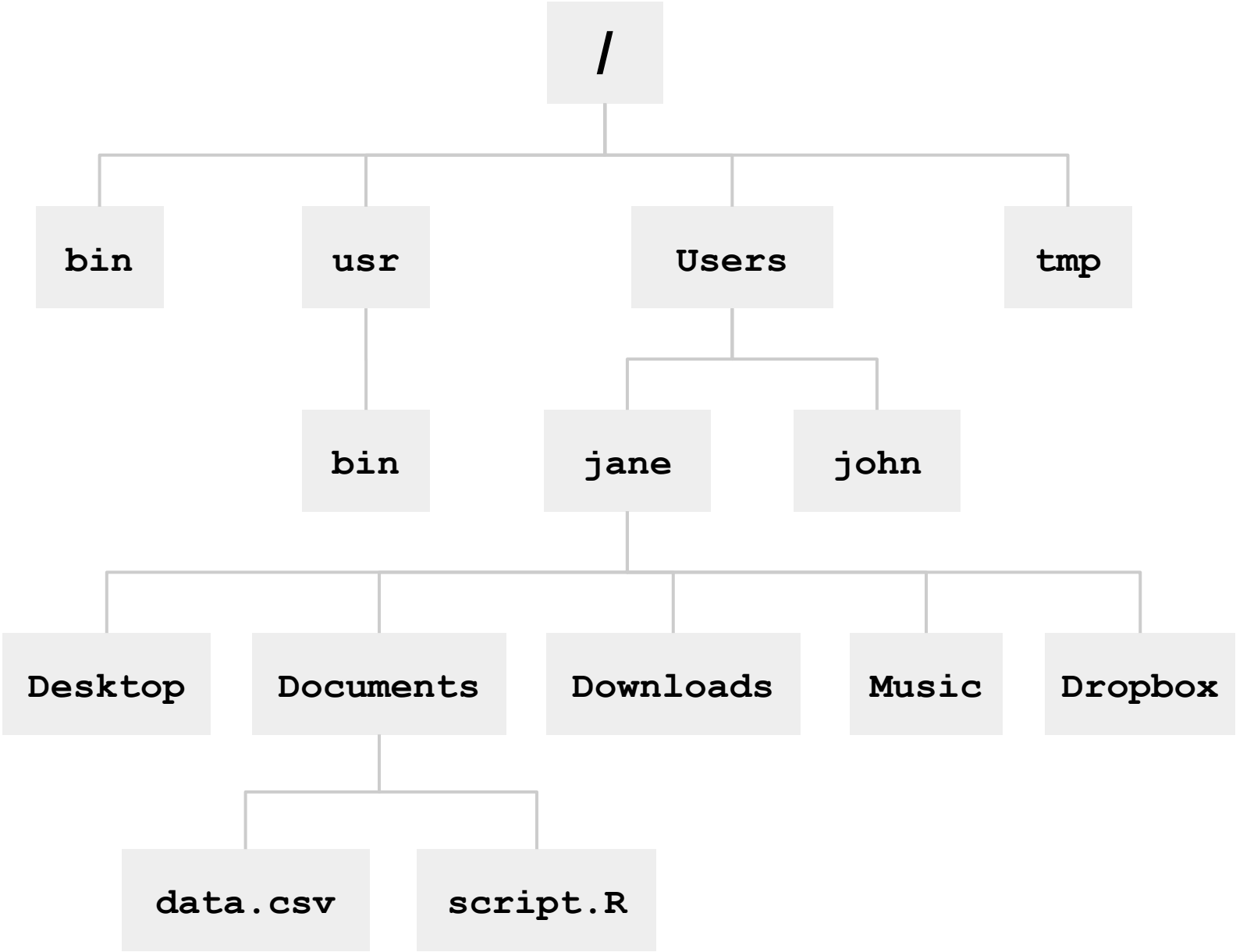

/
bin
usr
Users
tmp
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R

# Typical filesystem in UNIX-like OS

| Directory/Folder | Contents |
|---|---|
| `/` | Root |
| `/bin` | Binaries and programs (UNIX stuff) |
| `/sbin` | System binaries |
| `/dev` | Devices: hard drives, keyboard, mouse |
| `/etc` | System configurations |
| `/home` | User home directories (except on Mac) |
| `/lib` | Libraries of code |
| `/tmp` | Temporary files |
| `/var` | Various (files the system uses) |
| `/usr`<br>`/usr/bin`<br>`/usr/etc`<br>`/usr/lib`<br>`/usr/local` | User programs, tools and libraries |

# Mac-only files and directories

| Directory/Folder | Contents |
|---|---|
| `/Applications` | Mac programs |
| `/Library` | Mac libraries of code |
| `/Network` | Networked devices |
| `/System` | Mac OS X |
| `/Users` | User home directories |
| `/Volumes` | Mounted volumes (hard drive, dvd, etc) |
| `.DS_Store` | Holds folder view options, icon positions |
| `~/.MacOSX` | Directory for Mac OS X to store options |
| `~/.Trash` | User trash can |


/
bin
usr
Users
tmp
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R

# Main Unix Concepts

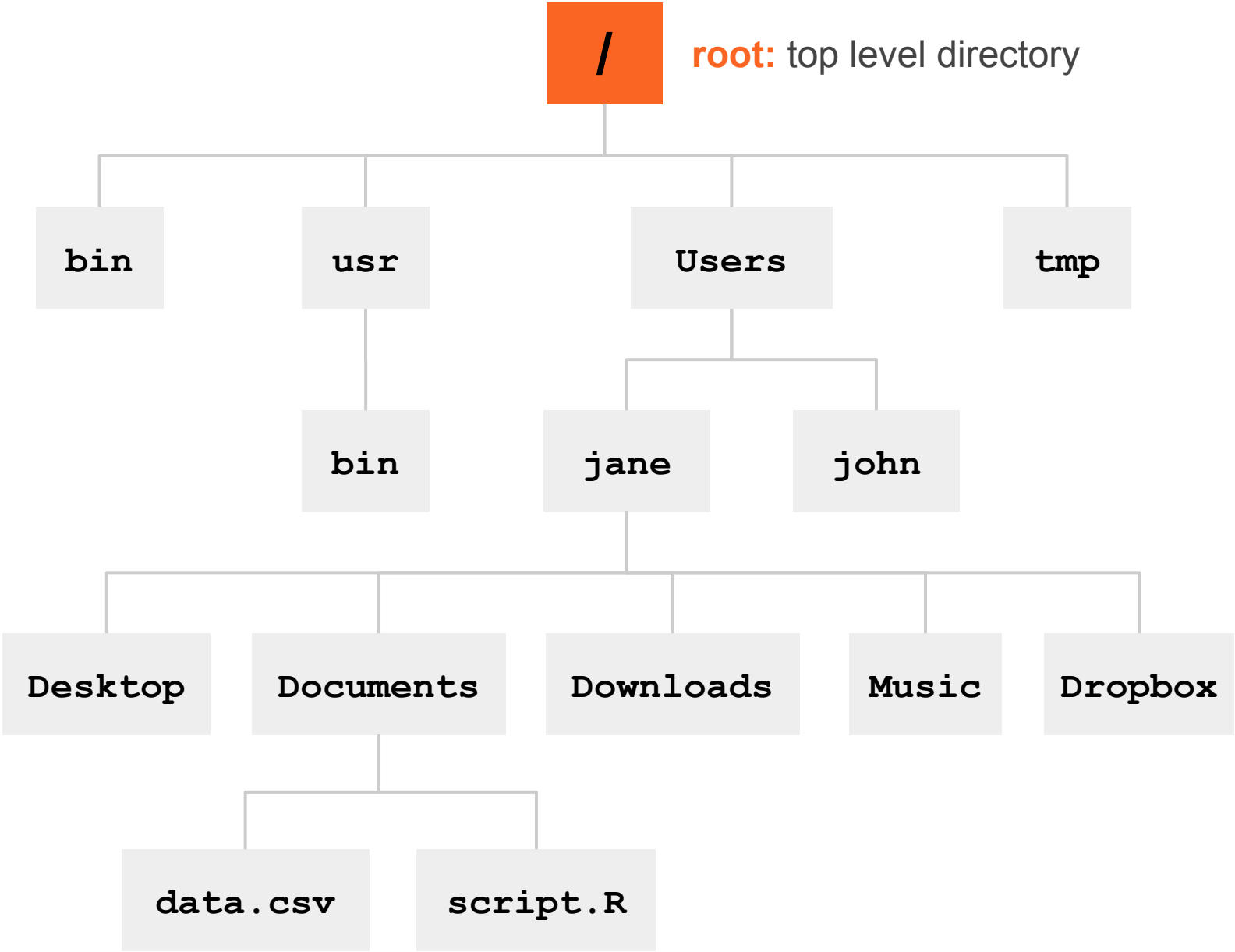
At any given time we are inside a directory

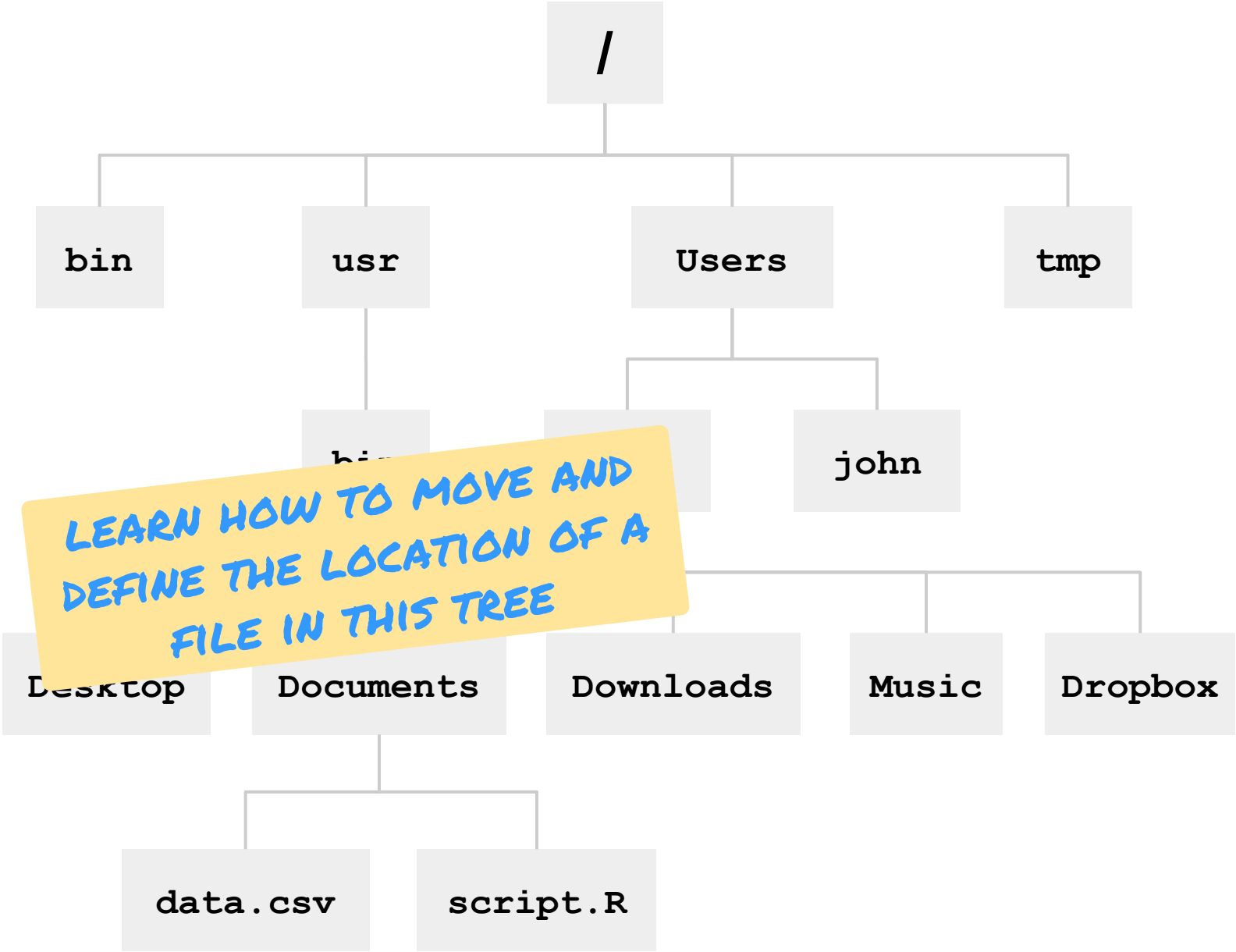
The current directory is the **working directory**

From that directory we can move up or down

When a new **R session** is started, a working directory will be associated to the session

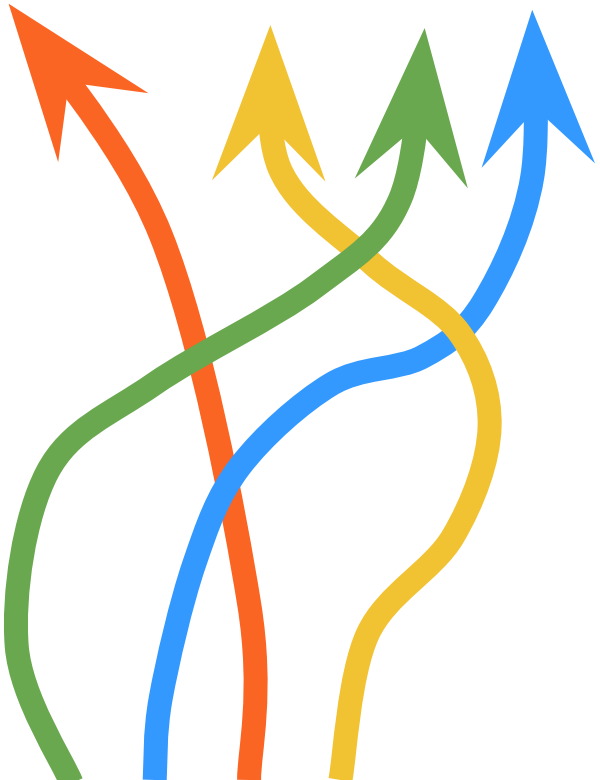
When a **terminal** is started the working directory is the **home directory**


/
root: top level directory
bin
usr
Users
tmp
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R

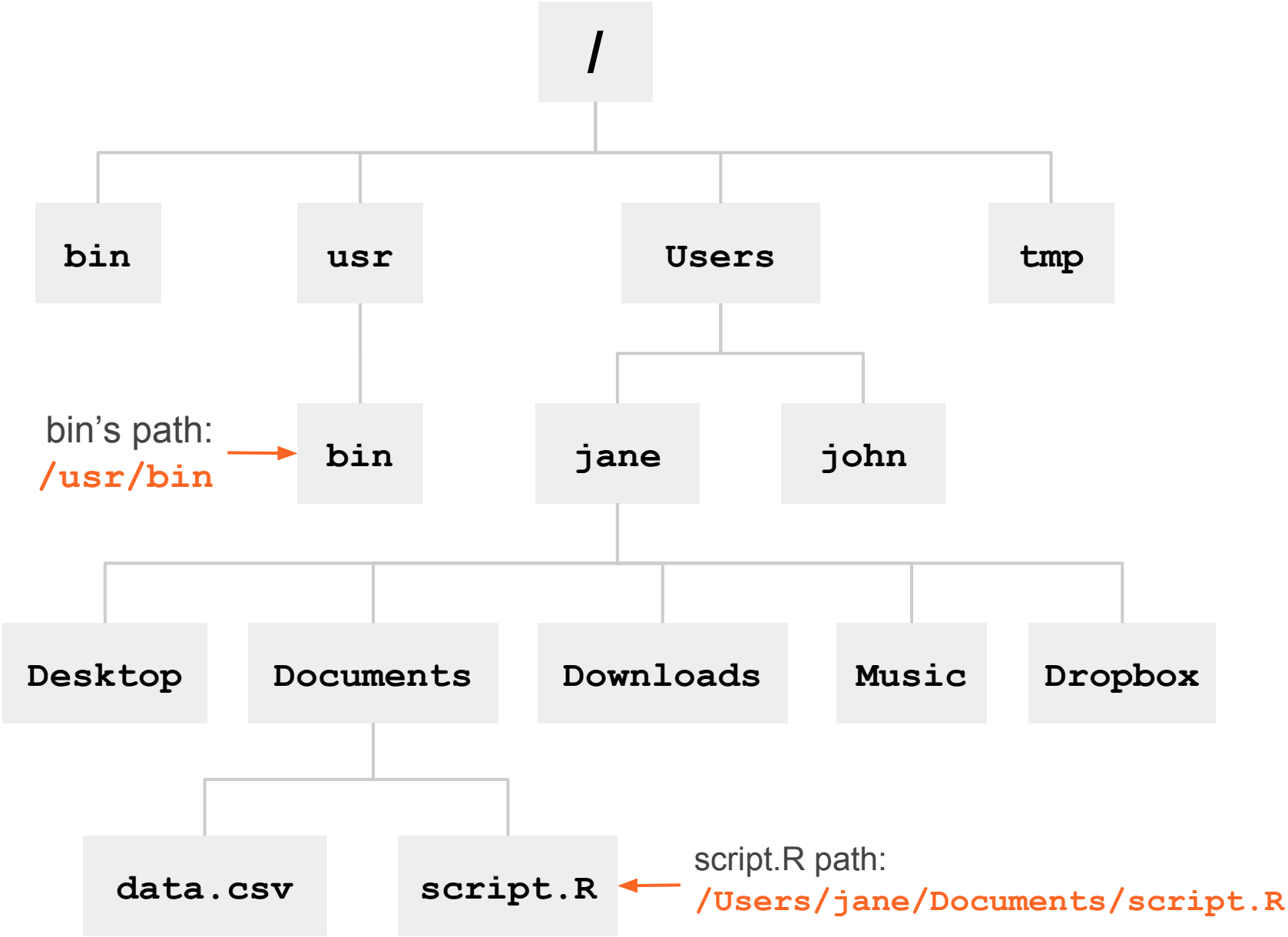


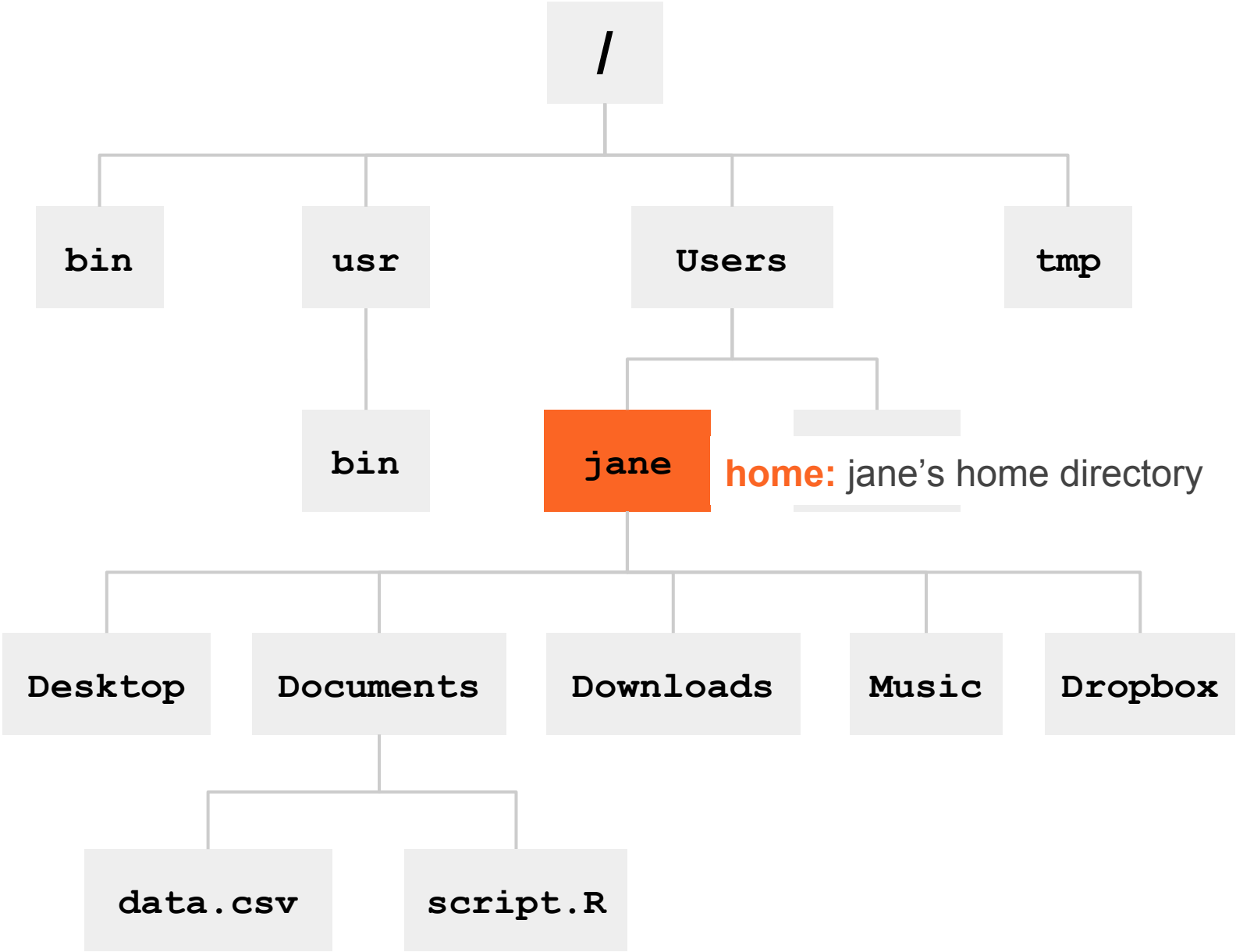

LEARN HOW TO MOVE AND DEFINE THE LOCATION OF A FILE IN THIS TREE

# Paths



Each file and directory has a unique name in the filesystem called a **path**


/
bin
usr
Users
tmp
bin's path:
/usr/bin
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R
script.R path:
/Users/jane/Documents/script.R


/
bin
usr
Users
tmp
bin
jane
home: jane's home directory
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R

# Special Directories

| | |
|---|---|
| `/` | root directory |
| `~` | home directory (i.e. `/home/user`) |
| `.` | current directory |
| `..` | parent directory |

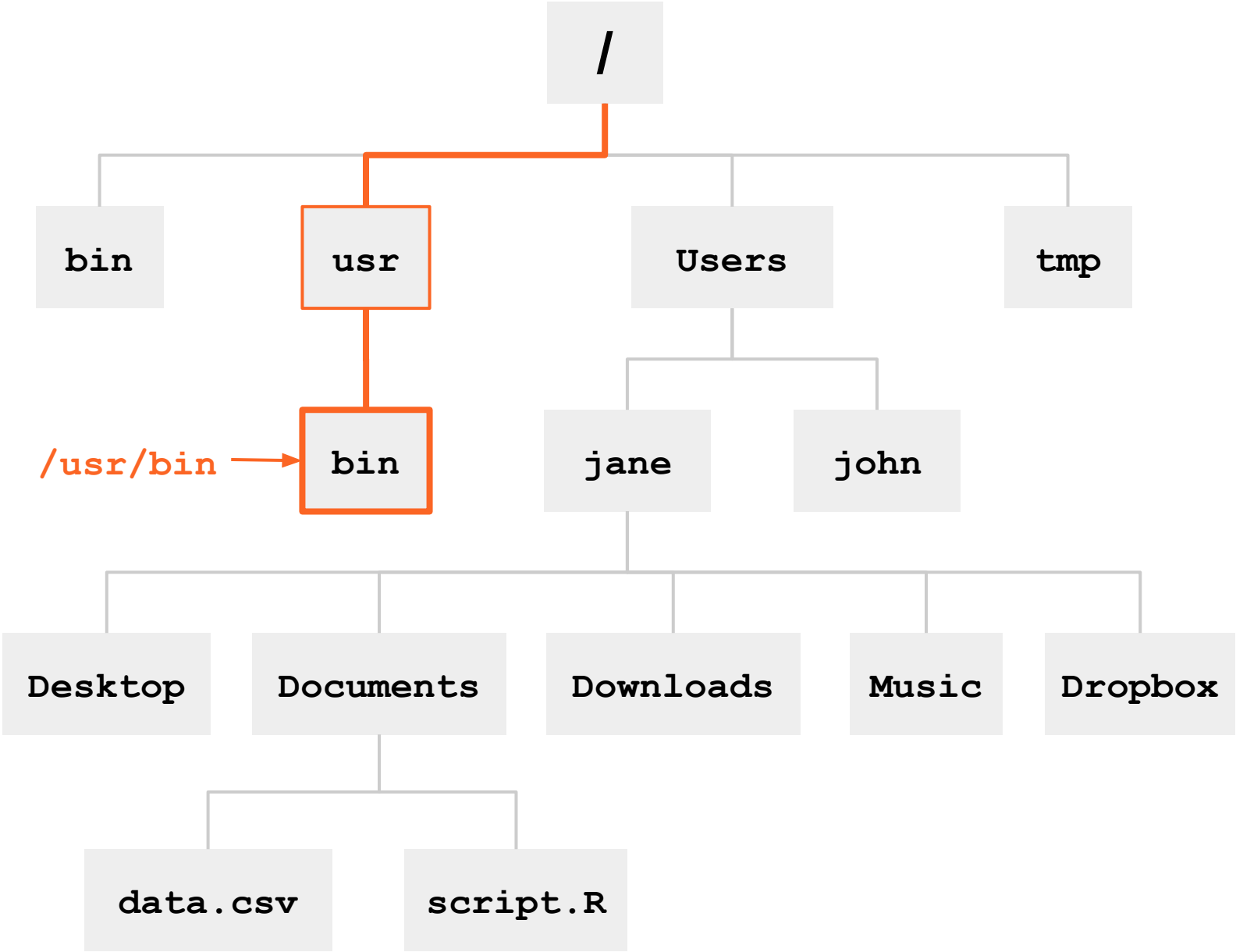
# Absolute Paths

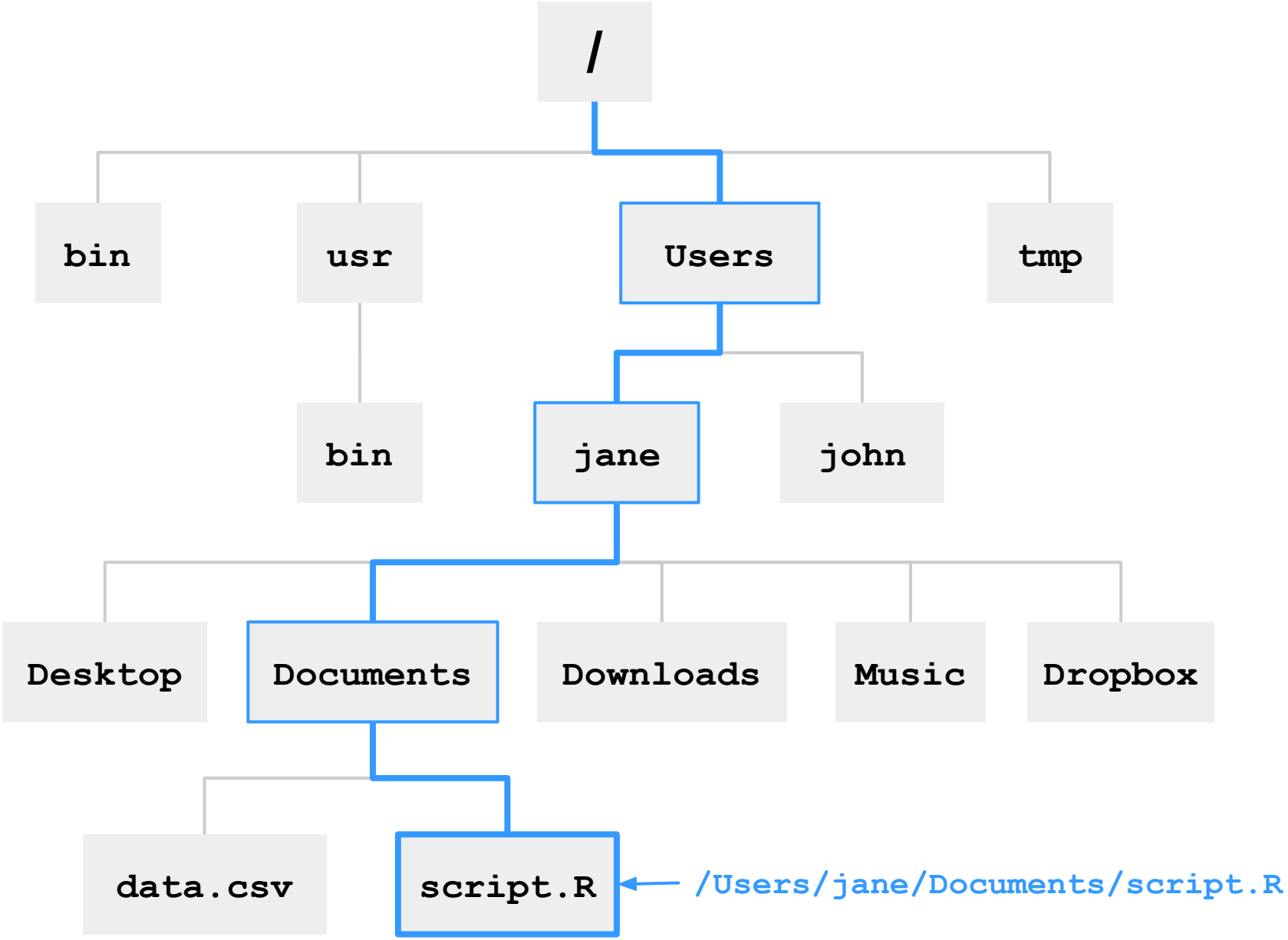
**Absolute paths:** an absolute pathname begins with the root directory and follows the tree branch by branch

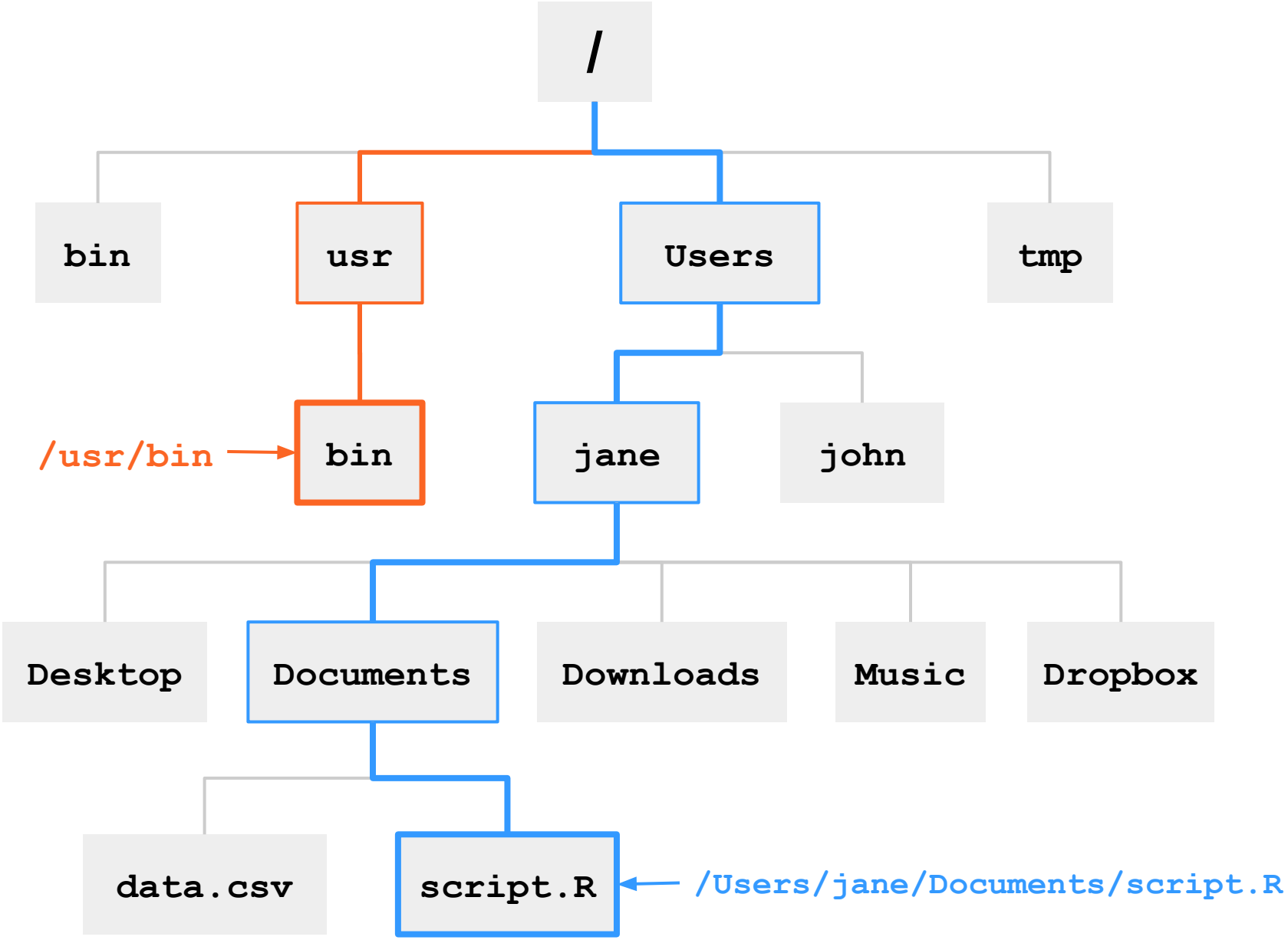
```
/A/B/C/D
```

The firts slash is the root directory

Subsequent slashes separate directories


/
bin
usr
Users
tmp
/usr/bin
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R

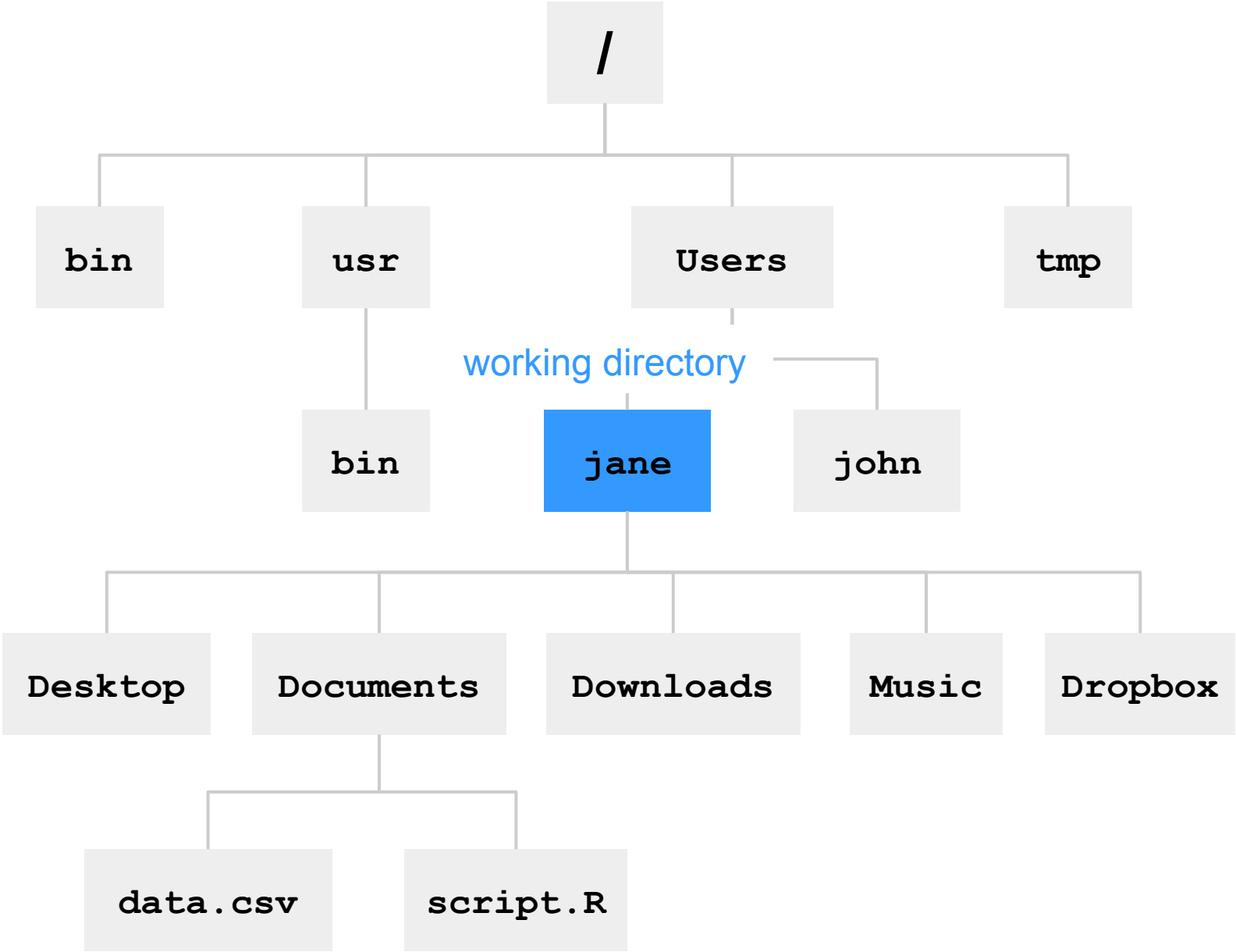

/
bin
usr
Users
tmp
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R
/Users/jane/Documents/script.R

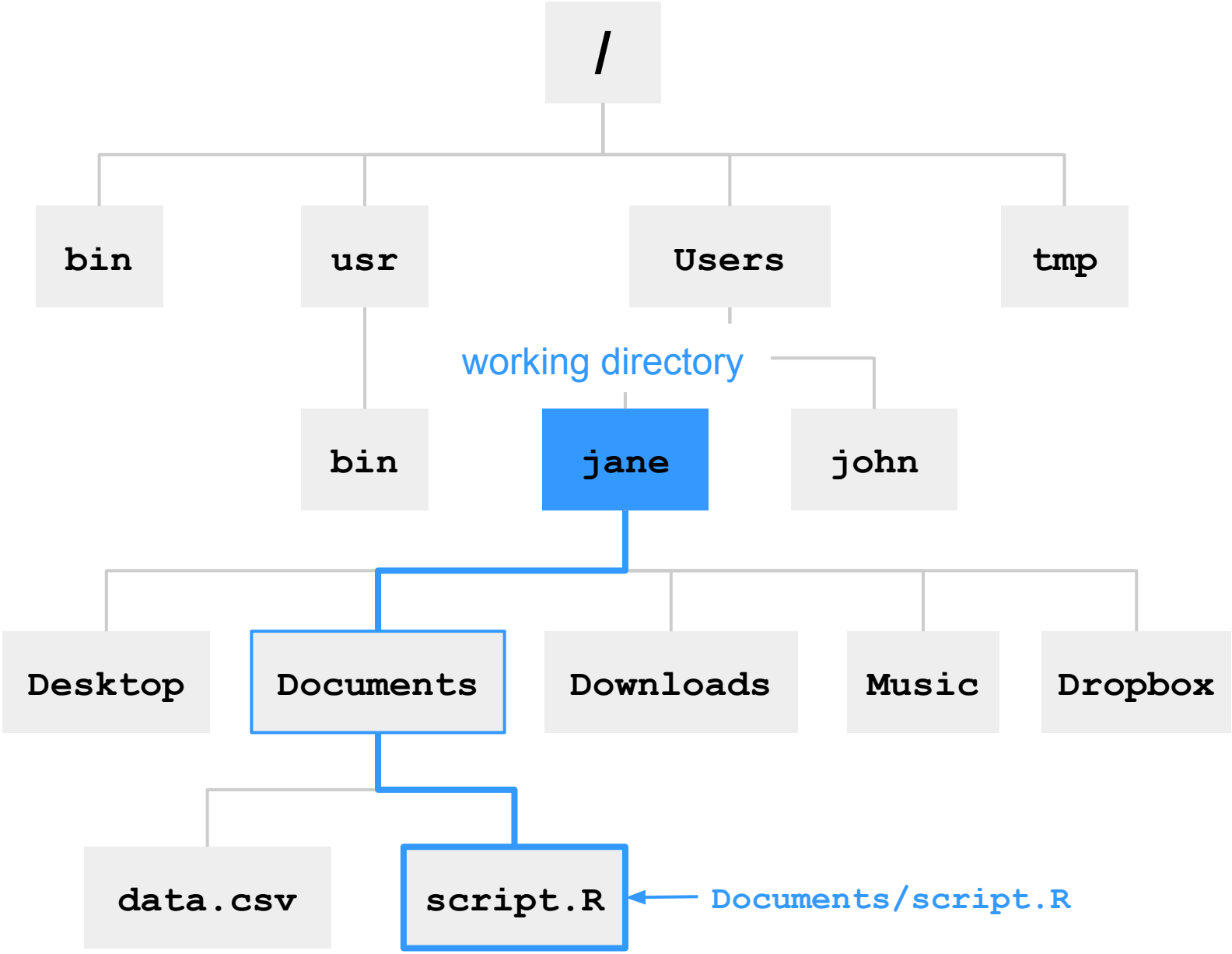

/
bin
usr
Users
tmp
/usr/bin
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R
/Users/jane/Documents/script.R

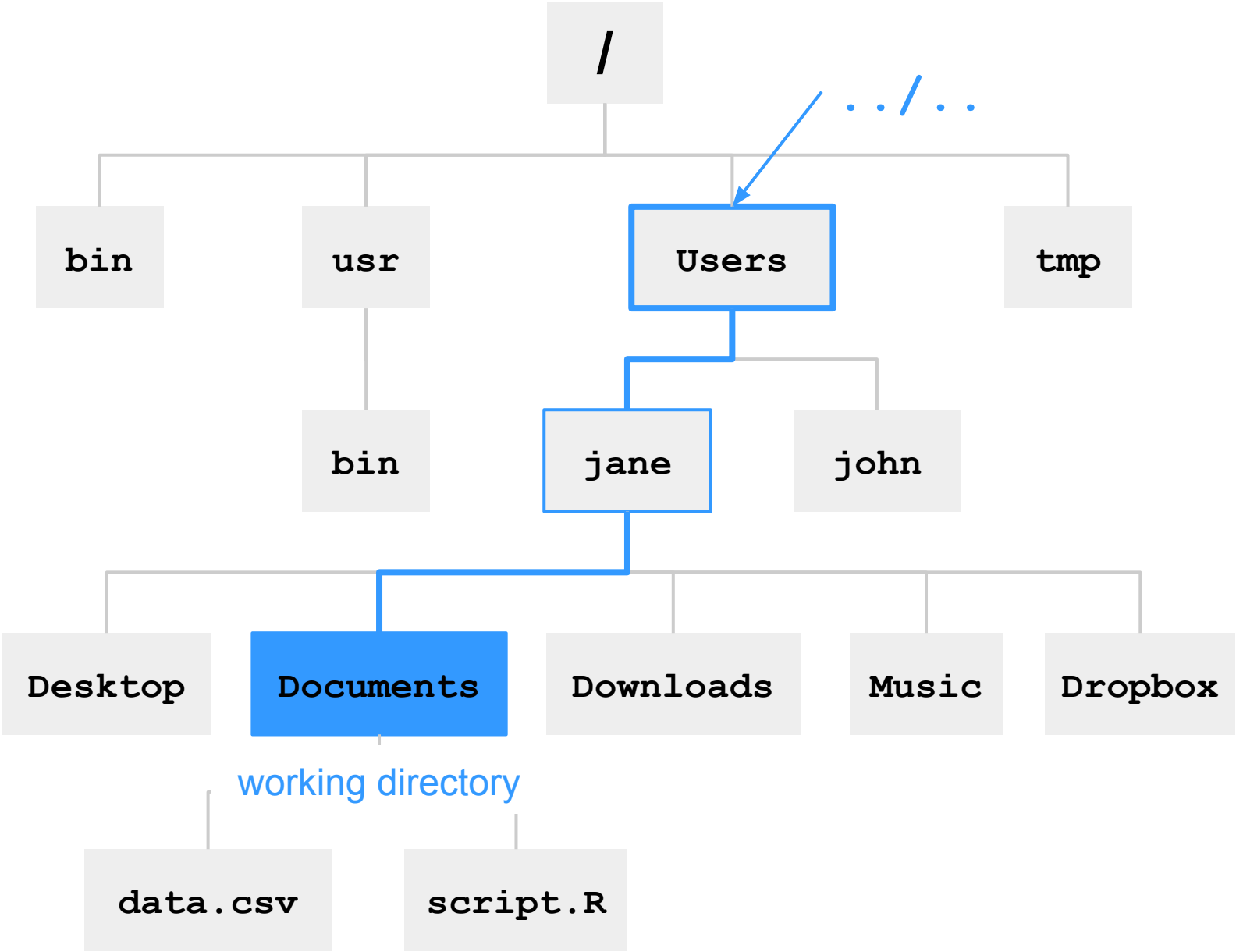
## Relative Paths

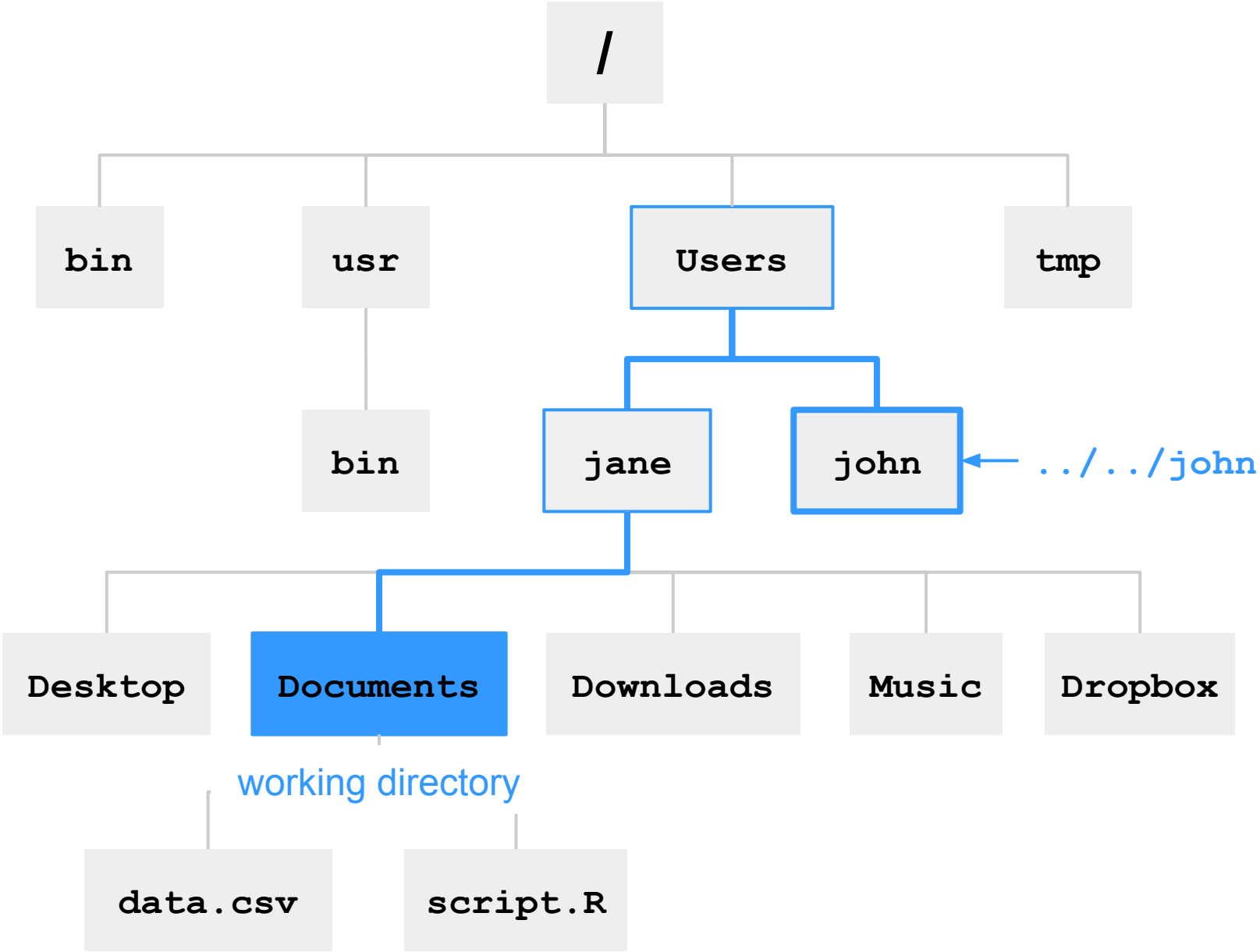
Relative paths: a relative pathname begins at some working directory, moving either up or down the tree

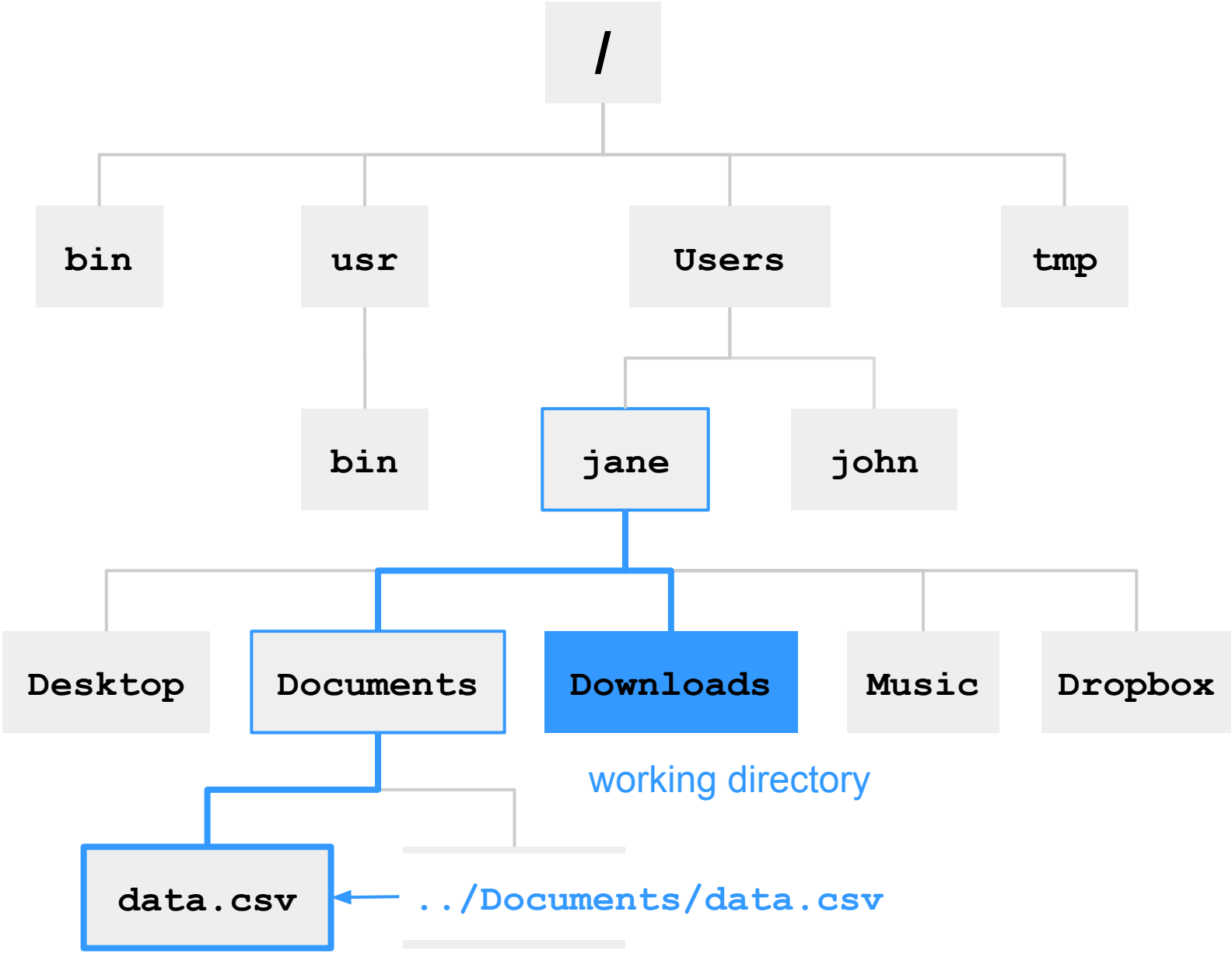
```
../C

./B
```


/
bin
usr
Users
tmp
working directory
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R

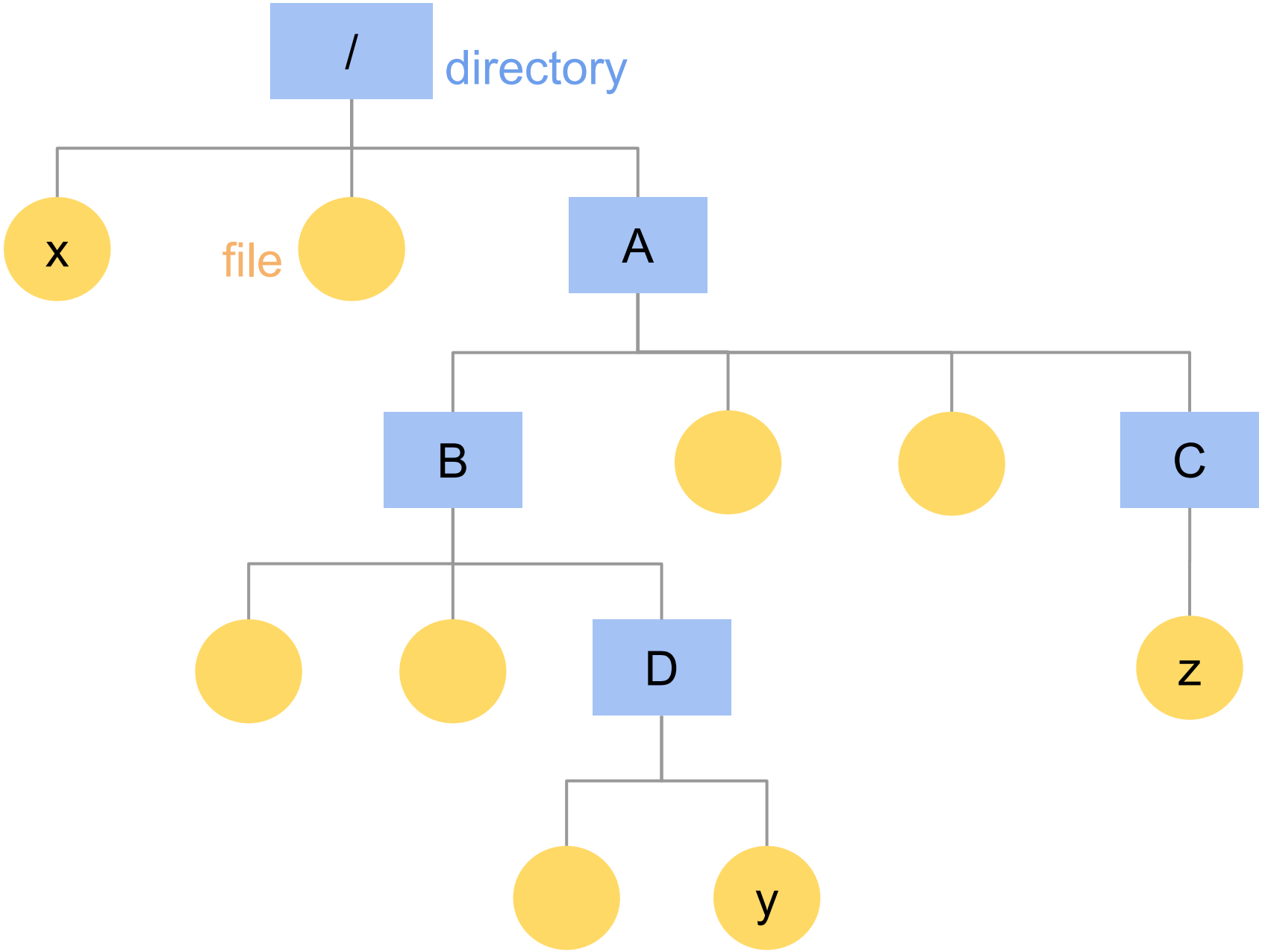

/
bin
usr
Users
tmp
working directory
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
data.csv
script.R
Documents/script.R


/
../..
bin
usr
Users
tmp
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
working directory
data.csv
script.R


/
bin
usr
Users
tmp
bin
jane
john
../../john
Desktop
Documents
Downloads
Music
Dropbox
working directory
data.csv
script.R


/
bin
usr
Users
tmp
bin
jane
john
Desktop
Documents
Downloads
Music
Dropbox
working directory
data.csv
../Documents/data.csv

# Your turn


/
directory
x
file
A
B
C
D
z
y

# Write the following Absolute Paths

From root to C:

From root to z:

From A to y:

From A to z:

# Write the following Relative Paths

From D to C:

To x from within C:

To y in D from within C: