2- the r data science environment

R objects:

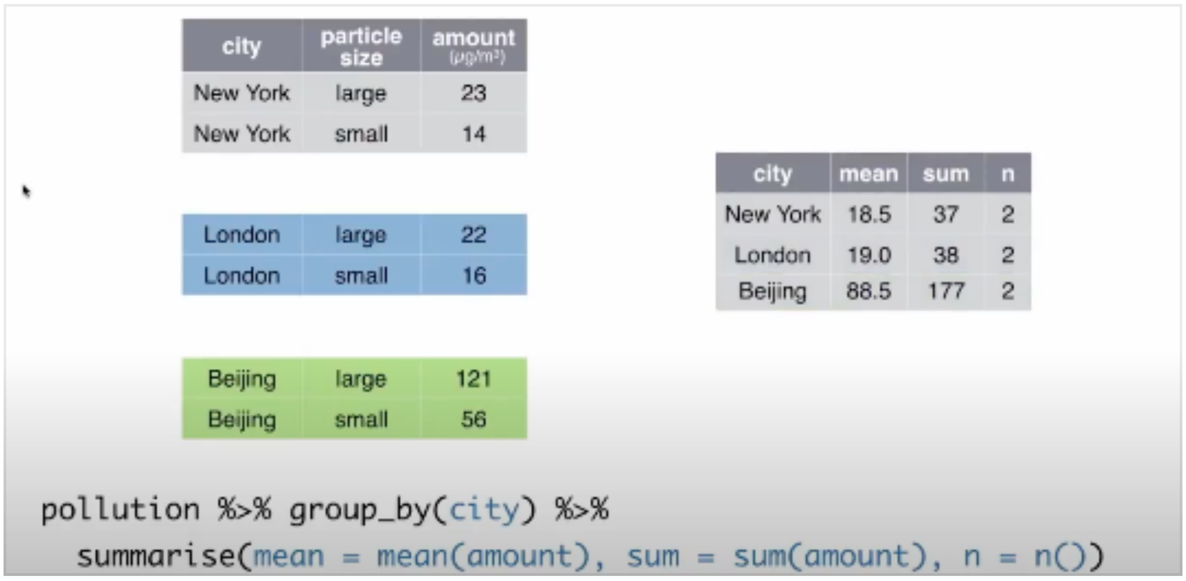
- atomic
    - **vector:** 1-dimensional
    - **matrix**: 2-dimensional
    - **array:** n-dimensional
    - **time-series:** vector with time-index
    - **factor:** vector of categories
- non-atomic
    - **list:** recursive

- check class: is.matrix(x), is.ts(x), is.vector(x)
- coerce to some class: as.list(x), as.array(x)
- everything is a vector in R. even basic scalars are vectors of length 1.
- first element's index is 1

---

pipe operator: %>%

- df %>%
    - do_this_operation %>%
    - then_do_this_operation %>%
    - then_do_this_operation ...



city	particle size	amount ($\mu g/m^3$)
New York	large	23
New York	small	14
London	large	22
London	small	16
Beijing	large	121
Beijing	small	56

city	mean	sum	n
New York	18.5	37	2
London	19.0	38	2
Beijing	88.5	177	2

```
pollution %>% group_by(city) %>%
  summarise(mean = mean(amount), sum = sum(amount), n = n())
```

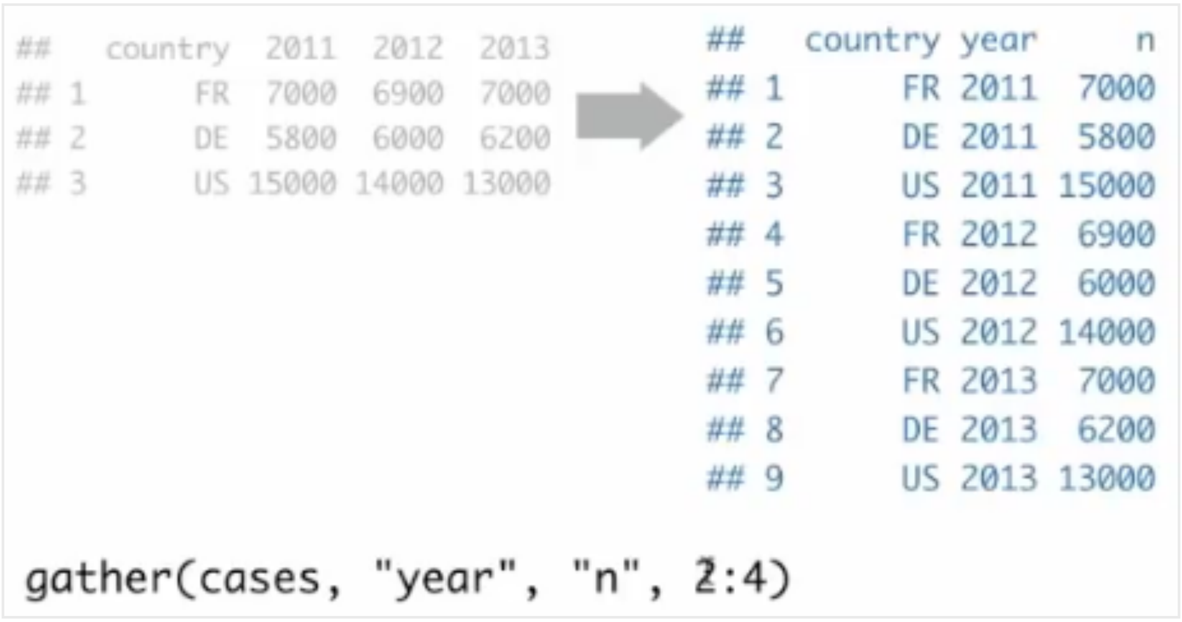
---

tidy data:

- variables/categories are in columns
- observations are in rows

tidyr: library used to wrangle the data

- **gather:** makes observations from the variables

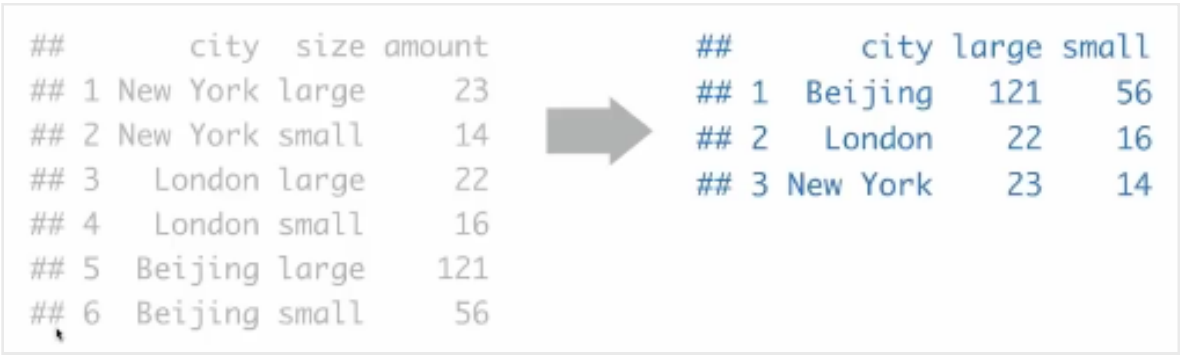


```
##    country  2011  2012  2013
## 1       FR  7000  6900  7000
## 2       DE  5800  6000  6200
## 3       US 15000 14000 13000
```

```
##    country year     n
## 1       FR 2011  7000
## 2       DE 2011  5800
## 3       US 2011 15000
## 4       FR 2012  6900
## 5       DE 2012  6000
## 6       US 2012 14000
## 7       FR 2013  7000
## 8       DE 2013  6200
## 9       US 2013 13000
```

```
gather(cases, "year", "n", 2:4)
```

- **spread:** makes variables from the observations



```
##         city  size amount
## 1 New York large     23
## 2 New York small     14
## 3   London large     22
## 4   London small     16
## 5  Beijing large    121
## 6  Beijing small     56
```

```
##         city large small
## 1  Beijing   121    56
## 2   London    22    16
## 3 New York    23    14
```

- x -> spread() -> y -> gather() -> x
- split and merge columns with unite() and separate()
  - example: 2000-12-14 into 3 different columns vs

---

dplyr: library used to manipulate data
- **select():** extract existing variables/ columns
- **filter():** extract existing observations
- **mutate():** derive new varibales from existing variables
- **summarise():** change the unit of analysis
- **arranges():** rearranges the dataframe according to the given column
- **bind_cols():** adds second as new columns to the first one
- **bind_rows():** adds second as new rows to the first one
- **union(x,y):** does not include the common rows
- **intersect(x,y)**
- **setdiff(x,y)**
- **left_join(x,y,by='col_name')**
- **inner_join(x,y,by='col_name'):** no NA cells unlike left join

---

```
x <- c(0:10)
x = [ 0  1  2  3  4  5  6  7  8  9 10 ]
```

```
x <- c(0:10, 2)
x = [ 0  1  2  3  4  5  6  7  8  9 10  2 ]
```

```
d <- paste("derya","tınmaz")
d = "derya tınmaz"
```

```
c() -> makes the vectors
all the elements must be the same type, or it casts
automatically
c("abc", 12) -> ["abc" "12"]
```

```
d <- c("derya","tınmaz")
d = [ "derya"  "tınmaz" ]
```

```
a = "derya"
typeof(a)
"character"
length(a)
1
nchar(a)
5
```

```
1:5
[ 1  2  3  4  5]
```

```
seq(from=0,to=13,by=1.5)
rep(c('ping','pong'),3)
```

```
v[-5] -> means all except element 5
```

```
data <- 1:4
names(data) <- c('bir','two','drei','quatre')
data['bir'] -> to access
```

```
mat <- matrix(1:9, nrow=3,ncol=3)
```

```
arr <- array(1:6,c(1,2,3))
```

```
mylist <- list(3,c(2,5,6),greeting='hello',list(3,4,5))
mylist [1]
[[1]] 3
mylist [[1]]
3
mylist$greeting
"hello"
```

```
mylist [[4]][[2]]
4
```

```
square <- function(x) { x * x }
square (5)
25
```

```
if ( 1 < 2 ) 5 else 6
```

```
x <- 0
for (i in 1:10) x <- x + i
```

```
 mean(c(1,2,3,4,5,NA), na.rm = TRUE)
3
```

%% (modulo)
%/% (int division)
%*% (matrix multiplication)

```
cities = c("istanbul","ankara", "izmir")
temps = c(23,32,34)
weather.df = data.frame(city = cities,temperature = temps)
```

- weather.df

```
       city    temperature
1  istanbul         23
2  ankara           32
3   izmir           34
```

- weather.df[1] -> gives first column
- weather.df[2,1] -> ankara
- weather.df[1:2,] ->

```
       city    temperature
1  istanbul        23
2  ankara          32
```

rownames(weather.df) -> "1" "2" "3"

- is not updated if it is cropped

- rownames(weather.df) <- c(3:5)

```
weather.df
        city    temperature
3  istanbul         23
4  ankara           32
5   izmir           34
```

- weather.df[1,] ->

```
 city temperature
3 istanbul         23
```

```
weather.df$city
[1] "istanbul" "ankara"   "izmir"
```

```
dim(weather.df)
[1] 3 2
```

---

install.packages("ggplot2")
library(ggplot2) -> loads the library

---

tidyverse: collection of R packages designed for data science

?ggplot2

```
library(ggplot2)
ggplot(mpg, aes(displ, hwy, colour = class)) +  geom_point()
```

---

library(dplyr)
tibble::as_tibble(diamonds) -> df tp tibble, it is more useful
View(diamonds) -> shows all data