DESCRIBING RAW DATA

Applied math for data science (DSC 482/MTH 445) Fall 2022

Table of Contents

Figure 1: From "Person of Interest" (2011-2016) by Jonathan Nolan

1 What is statistics?

"Statistics are all around us, from marketing to sales to healthcare. The ability to collect, analyze, and draw conclusions from data is not only extremely valuable, but it is also becoming commonplace to expect roles that are not traditionally analytical to understand the fundamental concepts of statistics." (DataCamp, n.d.)

- Which information does this "definition" contain?
    - Application of statistics (marketing, sales, healthcare)
    - Input (data), process (collect, analyze), and output (conclusions)
    - Importance (commonplace)
- Okay, but what **is** statistics?

  Statistics is the practice of turning data into information to identify trends and understand features of populations. (Davies, 2016)

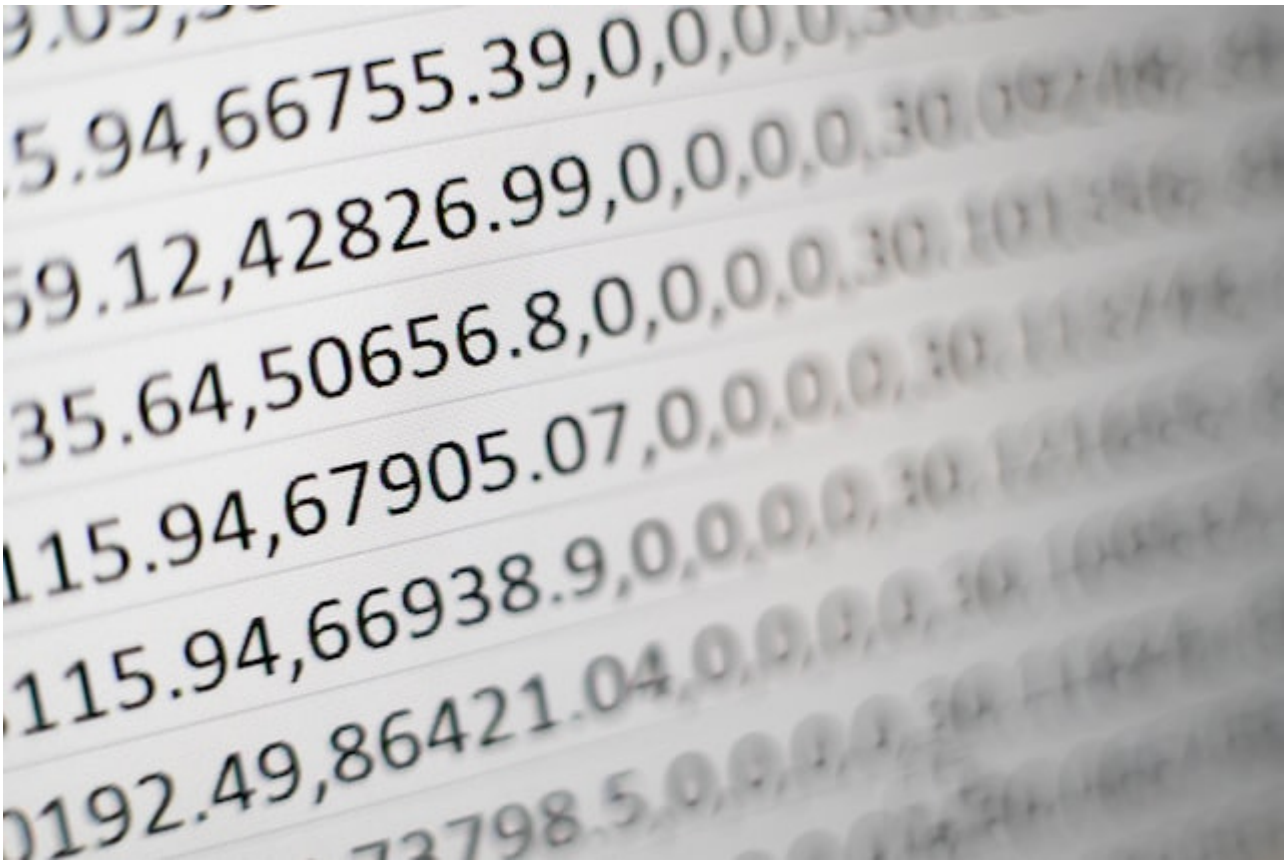
  Statistics is all about using the data from samples to draw conclusions about populations. (Schmuller , 2017)

  "Statistics, the science of collecting, analyzing, presenting, and interpreting data." (Britannica)

2 What is data?

- Lat. "*datum*" = that what is given:
    1. entity - for example a corporation or a school
    2. event - for example an event like an election
    3. process - for example a marketing campaign
- *Raw* data = records or observations that make up a *sample*
- *Big* data = Volume + Velocity + Variety[1]

3 What is data to R?



- Data can be *read in* from external files (e.g. `read.table`[2])
- Data can be *stored* as R objects (e.g. `data.frame`)
- Data can be *summarized* and *analyzed* using R functions (e.g. `summary`)

4 Statistical variables

- In statistics, *variables* are characteristics of individuals in a population
- An *individual* and *population* are (useful) abstractions (**Why?**)
- First step: identify and categorize the available variables

5 Example: data frames

- We use the `data.frame` function to create a data frame from scratch
- Data are supplied as *vectors* of the same length
- Data are grouped by *variable* (column vector)

- R code[3]:

```
mydata <- data.frame (
  person = c("Peter", "Lois", "Meg", "Chris", "Stewie"),
  age = c(42, 40, 17, 14, 1),
```

```
  sex = factor(c("M", "F", "F", "M", "M")))
mydata
```

- R variables: character vector, numbers, factor with levels
- To see the structure of an R object, use the function `str`.

```
str(mydata)
```

```
'data.frame':   5 obs. of  3 variables:
 $ person: chr  "Peter" "Lois" "Meg" "Chris" ...
 $ age   : num  42 40 17 14 1
 $ sex   : Factor w/ 2 levels "F","M": 2 1 1 2 2
```

- To extract portions of the data, use index operators `$` and `[]`.

```
## extract row 2 in column 2
mydata[2,2]

## extract rows 2 to 5 in column 2
mydata[2:4,2]

## extract age
mydata$age

## extract Lois' age (row 2, column 2)
mydata$age[2]

## extract persons older than 40
mydata$person[mydata$age >= 40]

## extract age of persons older than 40
mydata$age[mydata$age >= 40]
```

```
[1] 40
[1] 40 17 14
[1] 42 40 17 14  1
[1] 40
[1] "Peter" "Lois"
[1] 42 40
```

- In the last command, we extract from the column vector `mydata$person` only those values that are greater 40.
- An alternative extraction method uses the `subset` function (Kabacoff, 2017).

```
old <- subset(x=mydata, mydata$age >= 40)
old$person
old_male <- subset(x=mydata, mydata$age >= 40 & mydata$sex == "M")
old_male$person
```

```
[1] "Peter" "Lois"
[1] "Peter"
```

- To extract elements with multiple conditions, you need to build logical expressions.

```
## extract persons who are older than 40 and male
mydata$person[mydata$age >= 40 & mydata$sex == "M"]
```

```
[1] "Peter"
```

- To report size of data frames - number of records and variables, or rows and columns, use `nrow`, `ncol` and `dim`.

```
nrow(mydata) # retrieve number of rows or records

ncol(mydata) # retrieve number of columns or variables

dim(mydata)  # retrieve both number of rows and columns
```

```
[1] 5
[1] 3
[1] 5 3
```

6 Practice: data frames

7 Numeric variables

- *Numerical* variables are variables whose observations are naturally recorded as numbers.
- There are *continous* and *discrete* numerical variables.
    1. Continuous variables can be recorded as values in some interval, up to any number of decimals. Example: an observation of rainfall amount of 15 mm or of 15.42135 mm. The number of decimals provide the *precision* of the measurement.
    2. Discrete variables can only take on distinct numeric values. If the range is restricted, there is a finite number of possible values. Example: number of heads in 20 coin flips. The possible outcomes are restricted to integers in the interval [0,20].

8 Categorical variables

- *Categorical* variables can only take a finite number of possibilities (or categories) but they are not always recorded as numeric values
- There are *nominal* and *ordinal* categorical variables.
    1. Nominal variables cannot be logically ranked. Example: sex, with possible values **male** or **female**, and their order is irrelevant.
    2. Ordinal variables can be naturally ranked. Example: dose of a drug, with possible values low, medium, and high. These amounts can be ordered in increasing or decreasing order.

9 Example: chick weights

- The data frame `chickwts` is available in the automatically loaded `datasets` package. You can check that with `search()`[4].

```
search()
```

```
 [1] ".GlobalEnv"            "package:insuranceData" "package:robustbase"
 [4] "package:grid"          "package:MASS"          "ESSR"
 [7] "package:stats"         "package:graphics"      "package:grDevices"
[10] "package:utils"         "package:datasets"      "package:methods"
[13] "Autoloads"             "package:base"
```

- You can check the structure of `chickwts` with `str`.

```
str(chickwts)
```

```
'data.frame':   71 obs. of  2 variables:
 $ weight: num  179 160 136 227 217 168 108 124 143 140 ...
 $ feed  : Factor w/ 6 levels "casein","horsebean",..: 2 2 2 2 2 2 2 2 2 2 ...
```

- You can look at the first five records of the data set in two different ways, with the `head` function, or by extraction with the index operator[5].

```
chickwts[1:5, ]
```

```
  weight      feed
1    179 horsebean
2    160 horsebean
3    136 horsebean
4    227 horsebean
5    217 horsebean
```

- You can look at the meaning and origin of this data set with the `help` function (the help is better invoked from the *R console*)

```
help(chickwts)  # opens info sheet in default browser
```

- In the help, you see that these data contain the weights of 71 chicks in grams after six weeks, alongside 6 types of food given to them.

- `weight` is a *numeric* measurement that can fall anywhere on the continuum - it's a continuous variable. However, the recorded values seem to have been rounded.

```
chickwts$weight # show all values of chick weights
```

```
 [1] 179 160 136 227 217 168 108 124 143 140 309 229 181 141 260 203 148 169 213
[20] 257 244 271 243 230 248 327 329 250 193 271 316 267 199 171 158 248 423 340
[39] 392 339 341 226 320 295 334 322 297 318 325 257 303 315 380 153 263 242 206
[58] 344 258 368 390 379 260 404 318 352 359 216 222 283 332
```

- `feed` is a *categorical* variable with six non-numeric possible outcomes. Since these outcomes are not naturally ordered, it is a *nominal* categorical variable. The printout shows the levels in alphabetical order.

```
chickwts$feed
```

```
 [1] horsebean horsebean horsebean horsebean horsebean horsebean horsebean
 [8] horsebean horsebean horsebean linseed   linseed   linseed   linseed
[15] linseed   linseed   linseed   linseed   linseed   linseed   linseed
[22] linseed   soybean   soybean   soybean   soybean   soybean   soybean
[29] soybean   soybean   soybean   soybean   soybean   soybean   soybean
[36] soybean   sunflower sunflower sunflower sunflower sunflower sunflower
[43] sunflower sunflower sunflower sunflower sunflower sunflower meatmeal
[50] meatmeal  meatmeal  meatmeal  meatmeal  meatmeal  meatmeal  meatmeal
[57] meatmeal  meatmeal  meatmeal  casein    casein    casein    casein
[64] casein    casein    casein    casein    casein    casein    casein
[71] casein
Levels: casein horsebean linseed meatmeal soybean sunflower
```

10 Univariate and multivariate data

- Data related to only one dimension are called *univariate*
- For example, `chickwts$weight` is univariate: each measurement can be expressed with a single number, and stored as a *vector*.

- When measuring entities with more than one component associated with each observation, we measure *multivariate* data, and stored as *array*.
- For example, *spatial coordinates* have at least two components, a horizontal x- and a vertical y-coordinate. Each component on its own is not particularly useful. They are stored as a *matrix*.

11 Example: quake locations

- The built-in data set `quakes` give the locations of 1000 seismic events recorded off the coast of Fiji.
- Look at the first five events and read the descriptions in the help.

```
head(x=quakes, n=5)
```

```
     lat   long depth mag stations
1 -20.42 181.62   562 4.8       41
2 -20.62 181.03   650 4.2       15
3 -26.00 184.10    42 5.4       43
4 -17.97 181.66   626 4.1       19
5 -20.42 181.96   649 4.0       11
```

- The data set records spatial location data, depth in km, the magnitude on the Richter scale, and the number of observation stations that recorded the event.
- You can easily plot longitude and latitude of these 1,000 events:

```
plot(x=quakes$long,
     y=quakes$lat,
     xlab="Longitude",
     ylab="Latitude")
```

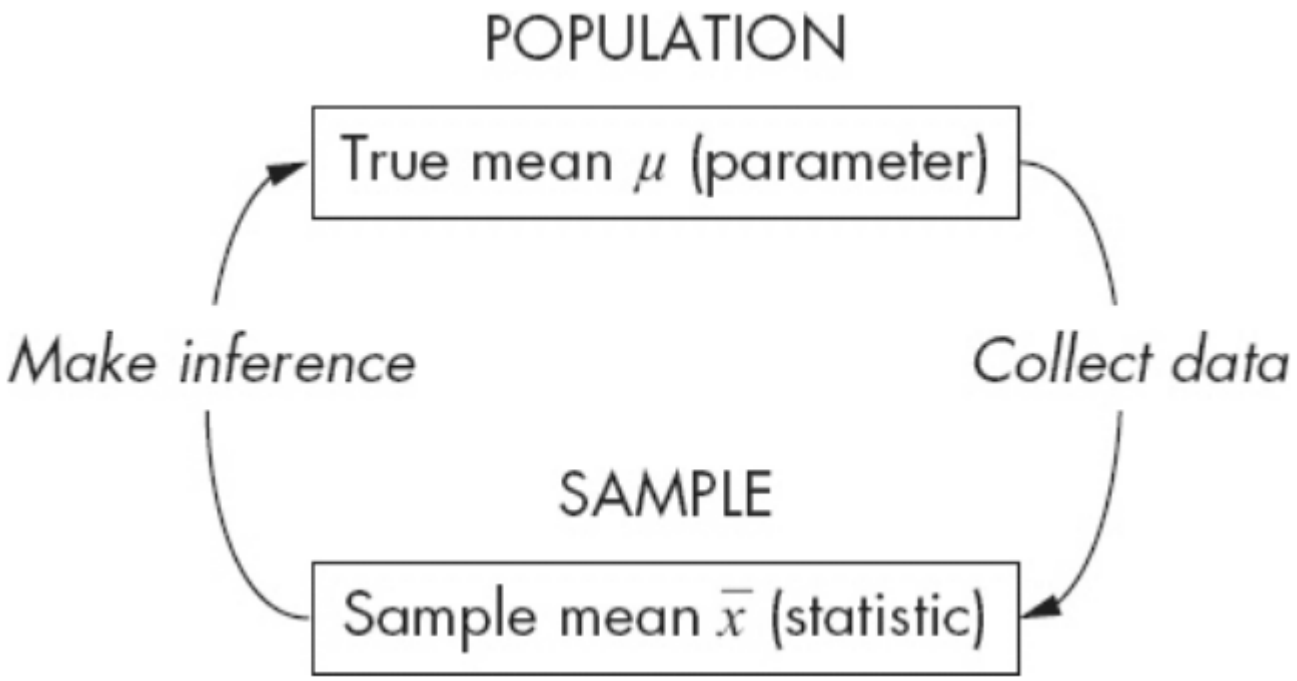
12 Parameter vs statistic

- *Statistics* is concerned with understanding *population* features
- A population is a collection of individuals or entities or events
- *Parameters* are population characteristics
- Populations cannot be accessed directly - instead, *samples* are taken
- *Statistics* are estimates of parameters of interest using the sample

13 Example: cat lovers

Example: let's say you wanted to know the average age of women in the US who own cats.

1. Population: all women in the US who own at least 1 cat
2. Parameter: mean age of US women who own at least 1 cat
3. Sample: randomly identify a smaller number of women with cat(s)
4. Statistic: mean age of women in the sample



14 Practice: statistical variables

15 References

- DataCamp (n.d.). Introduction to Statistics. URL: datacamp.com.
- Davies TD (2016). Book of R. NoStarch Press. URL: nostarch.com
- Kabacoff (2017). Quick-R: Subsetting Data. URL: stamethods.net.
- Schmuller J (2017). Statistical Analysis with R for Dummies. URL: wiley.com

16 Glossary: concepts

TERM	MEANING
Statistics	Data analysis techniques
Data	Entities, events, or processes
Raw data	Data originating from samples
Big data	Volume, Velocity, Variety
Variable	Characteristic of an individual in a population
vector	n-tuple of values of the same type
factor	vector of categorical variables
numeric variable	numbers
continuous numeric variable	potentially infinite numbers, with decimal point
discrete numeric variable	finite set of integer values
categorical variable	finite set of non-numeric values
nominal categorical variable	not naturally ordered categorical variable
ordinal categorical variable	naturally ordered categorical variable
univariate data	single dimension (vector)
multivariate data	more than one dimension (array)
population	individual or collective of interest
parameter	population characteristic of interest
sample	some data from a population
statistic	sample characteristic of interest

17 Glossary: code

CODE	MEANING
`read.table`	R function to read tabular data
`data.frame`	R function to create a data frame

CODE	MEANING
`summary`	R function to get summary statistics
`c`	R function to create vectors
`<-`	R assignment operator (right to left)
`factor`	R function to create factor vector
`$`	Accessor operator
`[]`	Index operator
`subset`	R function to extract subset of values
`nrow`	R function to return no. of rows
`ncol`	R function to return no. of columns
`dim`	R function to return object dimensions
`head`, `tail`	display beginning/end of data set
`str`	display structure of data set

Footnotes:

[1] This is the "3V" definition of big data. You'll find other attributes, like "value" or "veracity", which are not directly measurable, however.

[2] /You can get help on any of the examples with `?` or `help()`.

[3] Recall that a data frame consists of vectors. It is created with the `data.frame` function - its arguments are vectors of any type. Numerical or character vectors are created with the `c` function. Its arguments are values of any one type - characters or numbers. Factors are vectors, and they are created using the `factor` function. The difference is that their levels can be ordered explicitly.

[4] Also interesting: the related function `searchpath()` which returns the path searched by R to find packages

```
searchpaths()
```

[5] The `head` function prints 6 rows by default. To print only 5 rows, you need to restrict its range with `head(x=chickwts,n=5)`

Author: MARCUS BIRKENKRAHE
Created: 2022-09-22 Thu 14:57