Exploring a new data set

Introduction to Data Visualization

Table of Contents

1 README

This and the next few sections of the course provide a more detailed description of the objectives of EDA, the reasons for its importance, and some useful tools and techniques. Based on: Pearson, ch. 3 (2016).

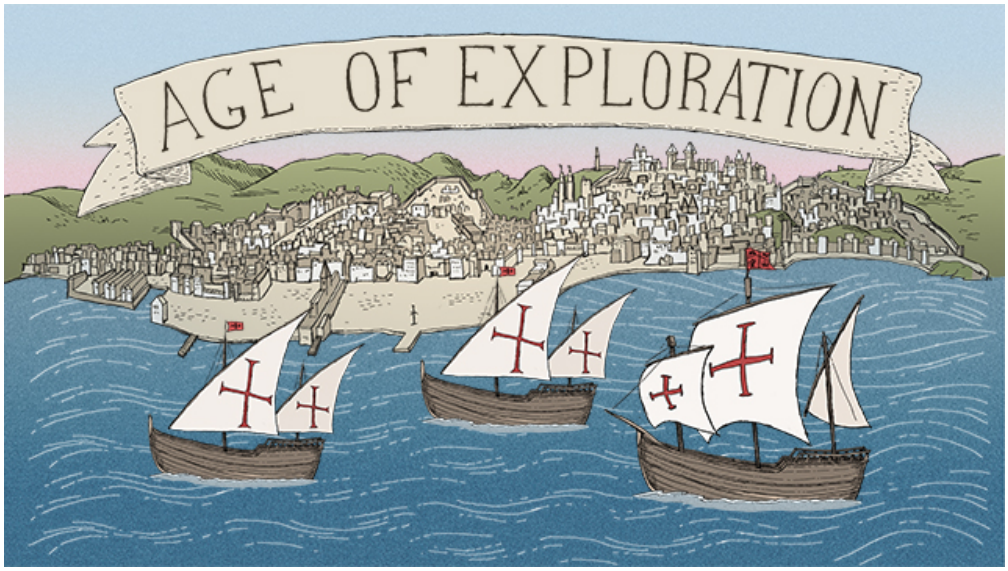
2 Key concepts in exploring data

(Image: May 3, 1808 - Francisco de Goya)

- Velleman and Hoaglin (1991) suggest four R's of EDA:
    1. Revelation

    2. Residuals
    3. Re-expression
    4. Resistance
- **Revelation** refers to data visualization as a way of revealing underlying patterns in the data. All the graphs we've created so far were examples of this activity.
- **Residuals** refers to the differences between observed values of a variable and its predictions from some mathematical model. Models used in EDA (like `mean` and `median`) can reveal patterns but such simple models would not be sufficient for predictive modeling.
- **Re-expression** refers to the application of mathematical transformations to one or more variables. The log transformation is an example but there are many more.
- **Resistance** refers to the presence of outliers or other data anomalies, which alter the analysis results and need to be explained, or removed, or both.

3 Exploration strategy



- The most general advice is "take the data seriously" and not just the models and tools used to analyse them (Strickland, 2022).
- In the era of Gauss (1777-1855), data were either collected directly (*primary data*), or obtained from a trusted friend or colleague (*secondary data*). Data sets were small and easy to know.
- Today, datasets are typically much larger, collected by people with whom we have no connection, or (like event logs) by machines that we did not build ourselves.
- General EDA strategy:
    1. Assess general dataset characteristics
    2. Examine descriptive statistics for each variable
    3. Examine exploratory visualizations
    4. Look for data anomalies
    5. Look at relations between key variables
    6. Summarize the results in form of a *data dictionary*

4 Assess general dataset characteristics

This step can often be achieved with built-in R functions like `summary`, `head`, or `str`. But the intrepid explorer knows how to build his own functions (or he/she can learn it @Lyon in DSC 205)!

1. How many records do we have?
2. How many variables do we have?
3. What type is each variable? Numeric, categorical, logical?
4. How many unique values does each variable have?
5. What value occurs most frequently, and how often does it occur?
6. Are there missing observations? If so, how many?
7. Do the values look like what we were expecting?
8. Do you understand what the variables mean?
9. Do you understand how they observations were obtained?

5 Using a custom function for exploration

- The function `BasicSummary` defined below generates a preliminary data summary for a data frame `df`. (On GitHub: tinyurl.com/45n7yub2)
- Results are returned to precision `dgts` (default value `3`)

  The function returns a data frame with one row for each column of `df` and the following columns:

  1. `variable`: the name of the corresponding column of `df`
  2. `type`: the `class` of the variable
  3. `levels`: the number of distinct values of the variable
  4. `topLevel`: the most frequently occurring value
  5. `topCount`: the number of times the most frequent value occurs
  6. `topFrac`: the fraction of records represented by `topCount`
  7. `missFreq`: the number of missing values of the variable
  8. `missFrac`: the fraction of records represented by `missFreq`

```
BasicSummary <- function(df, dgts = 3) {
  m <- ncol(df)
  varNames <- colnames(df)
  varType <- vector("character", m)
  topLevel <- vector("character", m)
  topCount <- vector("numeric", m)
  missCount <- vector("numeric", m)
  levels <- vector("numeric", m)
  for (i in 1:m) {
    x <- df[,i]
    varType[i] <- class(x)
    xtab <- table(x, useNA="ifany")
    levels[i] <- length(xtab)
    nums <- as.numeric(xtab)
    maxnum <- max(nums)
    topCount[i] <- maxnum
    maxIndex <- which.max(nums)
    lvls <- names(xtab)
    topLevel[i] <- lvls[maxIndex]
    missIndex <- which((is.na(x)) | (x=="") | (x==" "))
    missCount[i] <- length(missIndex)
  }
  n <- nrow(df)
  topFrac <- round(topCount/n, digits = dgts)
  missFrac <- round(missCount/n, digits = dgts)
```

```
  summaryFrame <- data.frame(
    variable = varNames,
    type = varType,
    levels = levels,
    topLevel = topLevel,
    topCount = topCount,
    topFrac = topFrac,
    missFreq = missCount,
    missFrac = missFrac)
  return(summaryFrame)
}
```

- This function is only defined for this session. To save it, use `save` and then import it with `load`:

```
save(BasicSummary,file="../data/BasicSummary")
```

- Remove function from session objects, then reload it

```
ls()
rm(BasicSummary)
ls()
load(file="../data/BasicSummary")
ls()
```

```
 [1] "a"            "bar"          "BasicSummary" "baz"          "char"
 [6] "chr"          "foo"          "h"            "hp"           "hpsub"
[11] "idx"          "mat"          "n"            "np"           "p"
[16] "q"            "seq"          "sub"          "vec"          "x"
 [1] "a"     "bar"   "baz"   "char"  "chr"   "foo"   "h"     "hp"    "hpsub"
[10] "idx"   "mat"   "n"     "np"    "p"     "q"     "seq"   "sub"   "vec"
[19] "x"
Error in readChar(con, 5L, useBytes = TRUE) : cannot open the connection
In addition: Warning message:
In readChar(con, 5L, useBytes = TRUE) :
  cannot open compressed file '../data/BasicSummary', probable reason 'No such file or dir
 [1] "a"     "bar"   "baz"   "char"  "chr"   "foo"   "h"     "hp"    "hpsub"
[10] "idx"   "mat"   "n"     "np"    "p"     "q"     "seq"   "sub"   "vec"
[19] "x"
```

6 Running `BasicSummary` on different datasets

- Run `BasicSummary` on the imported data set `df`

```
df <- read.csv(file="https://tinyurl.com/spdnvxbr",
               header=TRUE,
               stringsAsFactors=TRUE)
BasicSummary(df)
```

```
Error in BasicSummary(df) : could not find function "BasicSummary"
```

- Run `BasicSummary` on a real data set from the web, `HollywoodMovies2011` from the `Lock5withR` packagex:

```
library(Lock5withR)  # you may have to install this package
data(HollywoodMovies2011)
options(width=100)
hw <- BasicSummary(HollywoodMovies2011)
head(hw)
```

```
Error in BasicSummary(HollywoodMovies2011) :
  could not find function "BasicSummary"
Error in head(hw) : object 'hw' not found
```

- Run `BasicSummary` on the `Chile` data frame from the `car` package

```
library(car)
data(Chile)
BasicSummary(Chile,dgts=3)
```

```
Loading required package: carData
Error in BasicSummary(Chile, dgts = 3) :
  could not find function "BasicSummary"
```

- A closer look at the last result:
  1. Most of the variables have good explanatory names (except `statusquo`)
  2. R distinguishes `integer` and `numeric` (decimal) numbers
  3. Missing values are counted as a single level: e.g. `income` has `8` levels but the `table` only lists 7 because of the `NA`.

```
table(Chile$income)  # useNA="no" or "ifany"
```

```
2500   7500  15000  35000  75000 125000 200000
 160    494    768    747    269     88     76
```

  4. Missing values may have to be removed - if they show up depends on the precision of the record: add `dgts=5` in the function call.

7 TODO Variable types in practice

8 TODO Numerical vs. ordinal variables

9 TODO Text data vs. character strings

10 References

- Pearson RK (2016). Exploratory Data Analysis. CRC Press.
- Strickland E (9 Feb 2022). Andrew Ng: Unbiggen AI. IEEE Spectrum.

- Velleman PF, Hoaglin DC (1991). Data analysis. In: Hoaglin and Moore (eds.) Perspectives on Contemporary Statistics 21(2), Math. Assoc. of America.

Author: Marcus Birkenkrahe
Created: 2022-11-16 Wed 13:15