QTM 150

Week 6 – Working with Data

Umberto Mignozzetti
Mar 05

Recap

You now know:

- The main objects in R.
- How to do basic operations with datasets.
- How to create graphs and plots.

Great job!!

Do you have any questions?

Reminder: The quiz for this class will be posted at 4:00PM. The quiz is due Monday, 11:59 PM.

Today's Agenda

Today we will learn how to work with data.

- We will repeat some of the things that we did on DataCamp.
- This lab will reinforce the DataCamp learning, while helping you improve some gaps in your knowledge.
- Please ask questions!

Importing and Saving Data

Importing and Saving Data

Importing data from GitHub is easy. Let's do another time in here?

```
gss ← read.csv('https://raw.githubusercontent.com/umbertomig/qtm1

# Checking dimension
dim(gss) # This dataset contains 53474 rows and 14 variables
```

```
## [1] 53474    13
```

If GitHub does not work for you, use it in locale: download it to your computer.

Importing and Saving Data

If the dataset is big, you can always work with a chunk of it to get your coding up and running.

- This has the huge advantage of making things faster.

```
# Random number seed
set.seed(123) # This is not strictly required

# Extracting a chunk
gss100←gss[sample(nrow(gss),100),]
dim(gss100)
```

```
## [1] 100  13
```

And we can now save our dataset in our computer.

Importing and Saving Data

To save the dataset, we need the function `write.csv`. Let's check this function, and save our subsample?!

```
# Check your working directory
getwd() #be sure to check your working directory!
```

```
## [1] "/Users/umbertomignozzetti/Dropbox/Academic/Teaching/2021/QTM15
```

```
# Save a new dataset in your working directory
write.csv(gss100, "smallgss.csv")
```

We can go ahead and check the folder to see if the dataset is there!

Observing our data

```
# Print variable names
names(gss100)
```

```
##  [1] "region"   "income"   "happy"    "age"      "finrela"  "marita
##  [7] "degree"   "health"   "wrkstat"  "partyid"  "polviews" "sex"
## [13] "year"
```

```
# Print dataset dimension
dim(gss100)
```

```
## [1] 100  13
```

```
# Print summary
summary(gss100)
```

```
##     region             income             happy                 age
##  Length:100         Length:100         Length:100         Min.   :1
```

Observing our data

```
# First observations
head(gss100)
```

```
##                region         income        happy age       finrela
## 51663 E. NOR. CENTRAL $25000 OR MORE PRETTY HAPPY  79       AVERAGE
## 2986  E. NOR. CENTRAL $15000 - 19999   VERY HAPPY  38       AVERAGE
## 29925        MOUNTAIN $25000 OR MORE PRETTY HAPPY  43 ABOVE AVERAGE
## 29710 W. SOU. CENTRAL $25000 OR MORE PRETTY HAPPY  66 ABOVE AVERAGE
## 37529         PACIFIC $25000 OR MORE PRETTY HAPPY  42       AVERAGE
## 2757  W. SOU. CENTRAL $10000 - 14999   VERY HAPPY  33       AVERAGE
##              marital      degree    health          wrkstat
## 51663       MARRIED HIGH SCHOOL       IAP          RETIRED NOT STR
## 2986        MARRIED HIGH SCHOOL      GOOD    KEEPING HOUSE  STRONG
## 29925 NEVER MARRIED    BACHELOR      <NA> WORKING FULLTIME        C
## 29710       MARRIED HIGH SCHOOL EXCELLENT          RETIRED NOT STR
## 37529       MARRIED HIGH SCHOOL      FAIR WORKING FULLTIME        I
## 2757        MARRIED HIGH SCHOOL EXCELLENT    KEEPING HOUSE   NOT ST
##                   polviews    sex year
```

Extracting parts

Remember that a data.frame has the similar structure as a matrix [rows, columns], and each variable is a vector.

```
#dataset[row, column]
gss100[1,2]
```

```
## [1] "$25000 OR MORE"
```

```
# dataset[rows, columns]
gss100[1:5, c(2,5)]
```

```
##                income       finrela
## 51663 $25000 OR MORE       AVERAGE
## 2986  $15000 - 19999       AVERAGE
## 29925 $25000 OR MORE ABOVE AVERAGE
## 29710 $25000 OR MORE ABOVE AVERAGE
## 37529 $25000 OR MORE       AVERAGE
```

Extracting parts

Remember that a data.frame has the similar structure as a matrix [rows, columns], and each variable is a vector.

```
#dataset$variable, the whole column
gss100[,7]
```

```
##   [1] "HIGH SCHOOL"    "HIGH SCHOOL"    "BACHELOR"       "HIGH SCHO
##   [5] "HIGH SCHOOL"    "HIGH SCHOOL"    "JUNIOR COLLEGE" "BACHELOR'
##   [9] "HIGH SCHOOL"    "LT HIGH SCHOOL" "HIGH SCHOOL"    "HIGH SCHO
##  [13] "HIGH SCHOOL"    "LT HIGH SCHOOL" "JUNIOR COLLEGE" "HIGH SCHO
##  [17] "HIGH SCHOOL"    "LT HIGH SCHOOL" "HIGH SCHOOL"    "BACHELOR'
##  [21] "BACHELOR"       "HIGH SCHOOL"    "HIGH SCHOOL"    "HIGH SCHO
##  [25] "HIGH SCHOOL"    "HIGH SCHOOL"    "BACHELOR"       "BACHELOR'
##  [29] "HIGH SCHOOL"    "HIGH SCHOOL"    "HIGH SCHOOL"    "LT HIGH S
##  [33] "HIGH SCHOOL"    "BACHELOR"       "BACHELOR"       "LT HIGH S
##  [37] "LT HIGH SCHOOL" "BACHELOR"       "HIGH SCHOOL"    "HIGH SCHO
##  [41] "HIGH SCHOOL"    "BACHELOR"       "LT HIGH SCHOOL" "JUNIOR CO
```

R-Base plots

.font130[

We learned `qplot` in the previous class. However, if you need quick (but sadly ugly) plots, you use the R-Base plots.

Here we are going to do a few of these plots.

Histograms:

```
hist(gss100$age)
```

R-Base plots

.font130[

Barplots:

```
barplot(table(gss100$marital))
```

R-Base plots

.font130[

Boxplots (comparing numerical and categorical):

```
boxplot(gss100$age~gss100$sex)
```

Questions?

Have a great weekend!