



Data Analysis for the Social Sciences with R

Data Management

Prof. Kevin Koehler
kevin.koehler@santannapisa.it

Why data management?



Data management

1. Most data sets we encounter in the real world are messy; they need to be cleaned and re-shaped before analysis
2. We frequently want to merge together information from various data sources
3. Exploratory (descriptive) statistics can be of great help for discovering patterns



Today's class

1. Tidy data and the tidyverse



Today's class

1. Tidy data and the `tidyverse`
2. Data management functions



Today's class

1. Tidy data and the `tidyverse`
2. Data management functions
3. Exercises



What is tidy data



Tidy data

Observations in rows

id	age	vocab	Attitude
1	20	15	8
2	22	16	9
3	20	11	4
4	26	10	4
5	23	12	5
6	20	14	6
7	23	13	8

Variables in columns

id	age	vocab	Attitude
← 20 →	← 15 →	← 8 →	
← 22 →	← 16 →	← 9 →	
← 20 →	← 11 →	← 4 →	
← 26 →	← 10 →	← 4 →	
← 23 →	← 12 →	← 5 →	
← 20 →	← 14 →	← 6 →	
← 23 →	← 13 →	← 8 →	

id	age	vocab	Attitude
1	20	15	8
2	22	16	9
3	20	11	4
4	26	10	4
5	23	12	5
6	20	14	6
7	23	13	8

Values in cells



Messy data

Party	2014 Vote %	2019 Vote %
Nidaa Tounes	37.56%	1.51%
Ennahda	27.79%	19.63%

Table 1: Vote Percentages for Ennahda and Nidaa Tounes (2014 vs. 2019)

Tidy data

party	year	vote
Nidaa Tounes	2014	37.56
Nidaa Tounes	2019	1.51
Ennahda	2014	27.79
Ennahda	2019	19.63

Table 2: Vote Percentages for Ennahda and Nidaa Tounes (2014 vs. 2019)

The tidyverse

The **tidyverse** is a family of functions for data management. They follow a specific structure:

```
new_data <- data %>%  
  function() %>%  
  function()
```

The `%>%` is called a *pipe*. It allows you to stack functions on top of each other.

Data management



Data management functions

The most important data management functions are:

1. Adding or removing variables or observations

```
mutate() # to create new variables  
select() # to select specific variables  
filter() # to filter for values
```

2. Logical operations

```
ifelse() # ifelse(test,yes,no)  
case_when() # series of ifelse statements
```

3. Grouping and summarizing

```
group_by() # to group a data set  
summarize() # to summarize data
```

Data management functions

For example, returning to the Tunisia survey, look at

```
tun22 <- read_csv("tunisia_survey.csv")  
table(tun22$pres2019_2)
```

1	2	97	98	99
434	32	8	9	25

From the codebook, we know that 1 stands for Kais Saied and 2 for Nabil Karoui, the two candidates in the run-off round. 97 stands for blank or invalid votes, 98 for DK/don't remember, and 99 for declined to answer.

Wouldn't it be nice to see names instead of numbers?

Data management functions

```
tun22 <- tun22 %>%  
  mutate(pres2019_2_new = case_when(  
    pres2019_2==1~"Kais Saied",  
    pres2019_2==2~"Nabil Karoui",  
    pres2019_2==97~"blank/invalid",  
    pres2019_2==98~"don't know",  
    pres2019_2==99~"declined to answer"  
  ))
```

New variable with
names instead of
numbers

```
tun22 %>% tabyl(pres2019_2_new)
```

pres2019_2_new	n	percent	valid_percent
Kais Saied	434	0.434	0.85433071
Nabil Karoui	32	0.032	0.06299213
blank/invalid	8	0.008	0.01574803
declined to answer	25	0.025	0.04921260
don't know	9	0.009	0.01771654
<NA>	492	0.492	NA

Output with
janitor package



Data management functions

Next, assume we want to know the average value of Kais Saied voters on question:

To what extent do you agree with the following statement:
“Members of Parliament very quickly lose touch with ordinary people after they assume office.”

From the codebook, we know that this variable is called `mps`.

Data management functions

1. Filtering

```
tun22 %>%  
  select(pres2019_2_new, mps) %>%  
  filter(pres2019_2_new=="Kais Saied") %>%  
  summarize(mps=mean(mps, na.rm = T))
```

A tibble: 1 x 1

	mps
	<dbl>
1	2.38

2. Grouping

```
tun22 %>%  
  select(pres2019_2_new, mps) %>%  
  group_by(pres2019_2_new) %>%  
  summarize(mps=mean(mps, na.rm = T))
```

A tibble: 6 x 2

	pres2019_2_new	mps
	<chr>	<dbl>
1	Kais Saied	2.38
2	Nabil Karoui	2.44
3	blank/invalid	1.62
4	declined to answer	2.04
5	don't know	1.89
6	<NA>	2.77



Data management functions

4. Joining data sets

```
inner_join() # joining data sets keeping common obs  
left_join() # joining data sets keeping all left-hand obs  
right_join() # joining data sets keeping all right-hand obs  
full_join() # joining data sets keeping all obs
```

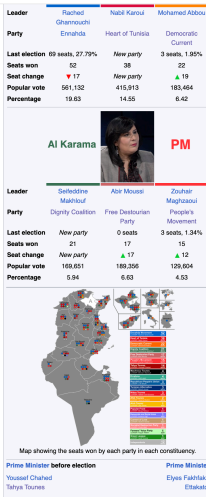
5. Reshaping data sets

```
pivot_longer() # reshape data into long format  
pivot_wider() # reshape data into wide format
```



Data management functions

Results

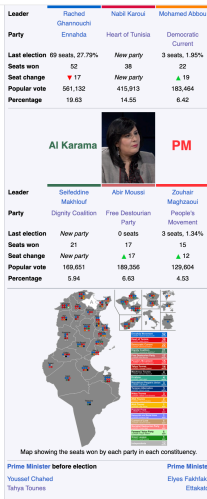


How do we get the data from Wikipedia into R?



Data management functions

Results [εκ]

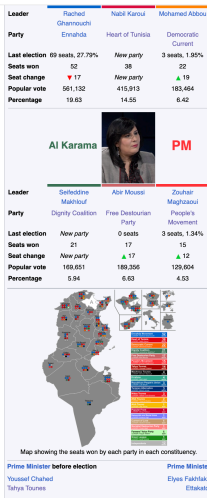


We could:

- manually copy the numbers

Data management functions

Results [web]



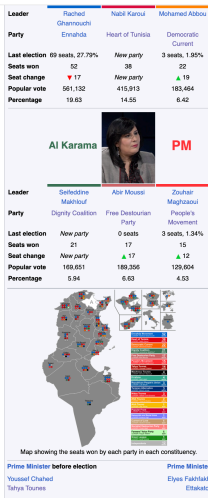
We could:

- manually copy the numbers
- copy paste into Excel, save the file, read it into R



Data management functions

Results [web]



We could:

1. manually copy the numbers
2. copy paste into Excel, save the file, read it into R
3. scrape the tables from Wikipedia using the `rvest` package in R (code is on the GitHub page)



Data management functions

1. Load the result files, inspect them, and remove unnecessary parts

```
res2014 <- read_csv("res2014.csv")
res2014 <- res2014[1:42,]
res2019 <- read_csv("res2019.csv")
res2019 <- res2019[1:22,] %>%
  mutate(Party=ifelse(Party=="Other parties/lists",
                      "Other parties",
                      Party))
```

2. Merge the two result files and select only vote percentages

```
results <- full_join(res2014,res2019,by="Party")
results <- results %>%
  select(Party, Percentage2014, Percentage2019)
```

Is the `results` data set tidy?

Data management functions

3. Reshape the data set to make it tidy

```
results <- results %>%  
  pivot_longer(  
    cols = starts_with("Percentage"),  
    names_to = "Year",  
    values_to = "Percentage",  
    names_prefix = "Percentage"  
  )  
head(results, 4)
```

```
# A tibble: 4 x 3  
  Party      Year Percentage  
  <chr>      <chr>   <chr>  
1 Nidaa Tounes 2014    37.56  
2 Nidaa Tounes 2019     1.51  
3 Ennahda Movement 2014    27.80  
4 Ennahda Movement 2019    19.63
```



Nice tables with stargazer

```
library(stargazer)

tab2014 <- results %>%
  filter(Year==2014 & !is.na(Percentage))

stargazer(tab2014,
           summary = F,
           type = "text") # or type="html" or type="latex"
```

Exercices



Exercises

1. Create a new variable in the survey data which has names instead of numbers for the first-round results
2. Create a new data set which includes the average value of the `mps` variable for the voters of all first-round candidates
3. Create a single data set with the number of seats won by Tunisian political parties in 2014 and 2019
4. Produce a properly formatted table with seat shares in 2014 and 2019 which you could include in a table (using the text processor of your choice)

