Introduction to R

Session 02: Working with data in R

Álvaro Pérez[1]

Instituto Tecnológico Autónomo de México

Fall 2024

[1]Based on PhD Romero Londoño's notes.

Working with Data

- R is all about working with data!
- data.frames:
    - data.frames are an object type
    - Most of the time, you'll be doing calculations using them
    - Conceptually, data.frames are basically spreadsheets
    - Technically, they're a list of vectors

data.frame

- It's a collection of vectors of the same length
- Note the use of = here, not <-

```
df <- data.frame(
RacePosition = 1:5,
WayTheySayHi = as.factor(c('Hi','Hello','Hey','Yo','Hi
    ')),
NumberofKids = c(3,5,1,0,2))
df
```

Glancing at Data

- If we just want a quick overview:
  - Arrow in the Environment tab
  - ‘head()‘ (look at the head of the data - first six rows)
  - ‘str()‘ (structure)

```
str(df)
```

Getting at Data

- Now we have a data frame, ‘df‘. How do we use it?
- We can pull the vectors back out with ‘$‘ Note autocompletion of variable names.
- We can treat it just like the vectors we had before

```
df$NumberofKids
df$NumberofKids[2]
df$NumberofKids >= 3
```

Practice

- Create ‘df2 <- data.frame(a = 1:20, b = 0:19*2,‘ ‘c = sample(101:200,20,replace=TRUE))‘
- What is the average of ‘c‘?
- What is the sum of ‘a‘ times ‘b‘?
- Did you get any values of ‘c‘ 103 or below? (make a logical)
- What is on the 8th row of ‘b‘?
- How many rows have ‘b‘ above 10 AND ‘c‘ below 150?

Practice Answers

```r
df2 <- data.frame(
a = 1:20,
b = 0:19*2,
c = sample(101:200,20,replace=TRUE))

mean(df2$c)

sum(df2$a*df2$b)

sum(df2$c <= 103) > 0

df2$b[8]

sum(df2$b > 10 & df2$c < 150)
```

Working With Data Frames

- We can manipulate data frames:
  - Create new variables
  - Change variables
  - Rename variables
- It's very common that you'll have to work with data before analyzing it
- "data cleaning" is very important and a big part of statistical analysis

Creating New Variables

- data.frames are just lists of vectors
- So create a vector and tell R where in that list to stick it
- Use descriptive names so you know what the variable is

```r
df$State <- c('Alaska','California',
                              'California','Maine',
                              'Florida')
df
```

Another approach - DPLYR and Tidyverse

- We just saw the base-R way to do it
- Can use **dplyr** (data pliers) for data manipulation instead
- dplyr syntax is inspired by SQL
- tidyverse isn't a part of base R. It's in a package, so we'll need to install it

Variable creation with dplyr

- The **mutate** command will “mutate” our data frame to have a new column in it
- The pipe ‘%>%‘ says “take df and send it to that mutate command to use”
- Or we can stick the data frame itself in the ‘mutate‘ command
- Thus these two are equivalent:

```
library(tidyverse)
df1 <- df %>% mutate(State = c('Alaska','California',
                                 'California','Maine','
                                     Florida'))

df2 <- mutate(df,State = c('Alaska','California',
                                 'California','Maine','
                                     Florida'))

identical(df1,df2)
df <- df1
```

Creating New Variables

- We can create multiple new variables in one mutate command

```r
df <- df %>% mutate(
MoreThanTwoKids = NumberofKids > 2,
One = 1,
KidsPlusPosition = NumberofKids + RacePosition)

df
```

Manipulating Variables

- We can't really **change** variables, but we can overwrite them
- We can drop variables with '-' in the dplyr 'select' command
- Note we chain multiple dplyr commands with '%>%'

```r
df <- df %>%
  select(-KidsPlusPosition,-WayTheySayHi,-One) %>%
  mutate(State = as.factor(State),
         RacePosition = RacePosition - 1)
df$State[3] <- 'Alaska'
str(df)
```

Renaming Variables

- Sometimes it will make sense to change the names of the variables we have.
- Names are stored in ‘names(df)‘ which we can edit directly
- Or the ‘rename()‘ command in dplyr has us covered

```r
names(df)
#two ways of renaming
#names(df) <- c('Pos','Num.Kids','State','mt2Kids
    ')

df <- df %>% rename(Pos = RacePosition, Num.Kids=
    NumberofKids,
                    mt2Kids = MoreThanTwoKids)

names(df)
```

Practice

- Create a data set ‘data‘ with three variables: ‘a‘ is all even numbers from 2 to 20, ‘b‘ is ‘c(0,1)‘ over and over, and ‘c‘ is any ten-element numeric vector of your choice.
- Rename them to ‘EvenNumbers‘, ‘Treatment‘, ‘Outcome‘.
- Add a logical variable called Big that’s true whenever EvenNumbers is greater than 15
- Increase Outcome by 1 for all the rows where Treatment is 1.
- Create a logical AboveMean that is true whenever Outcome is above the mean of Outcome.
- Display the data structure

Practice Answers

```r
data <- data.frame(a = 1:10*2,
                   b = c(0,1),
                   c = sample(1:100,10,replace=FALSE))
                       %>%
  rename(EvenNumbers = a, Treatment = b, Outcome = c)

data <- data %>%
  mutate(Big = EvenNumbers > 15,
         Outcome = Outcome + Treatment,
         AboveMean = Outcome > mean(Outcome))

str(data)
```

dplyr package

- select(): used to select columns of a data frame that you want to focus on
- filter(): used to extract subsets of rows from a data frame
- arrange(): used to reorder rows of a data frame according to one of the variables/columns
- rename(): renaming a variable
- mutate(): compute transformations of variables in a data frame
- group_by():used to generate summary statistics from the data frame within strata defined by a variable.
- %>%: stringing together multiple dplyr functions in a sequence of operations.
- summarize()

Example

- Download the database "1976-2020-president.csv" from my github repository
- In R, upload the dataframe
- Preserve only the votes for the Democratic and the Republican Party from de 1976 elections
- Create a new dataframe with only the state, the party, the votes for each party and the total of votes
- Sort states by number of votes
- Rename de variable "party_simplified.ªs "party"
- Create new variable that is the percentage of the vote (use the dataframe of the second step)
- Group by party and calculate de total aumount of votes for each party in that election

Answers

```r
#install.packages("dplyr")
library(dplyr)
# Rename database
US_elections <- X1976_2020_president
head(US_elections)
str(US_elections)
# filter()
elect_1976 <- filter(US_elections, year == 1976 & (
    party_simplified == "DEMOCRAT" | party_simplified
    == "REPUBLICAN"))
# select()
votes_1976 <- select(elect_1976,state | party_
    simplified | candidatevotes)
# arrange()
arrange(votes_1976,candidatevotes)
# rename()
votes_1976 <- rename(votes_1976,"party"=party_
    simplified)
```

```r

# mutate()
elect_1976 <- elect_1976 %>%
  mutate(per_votes = candidatevotes/totalvotes)
# group_by()
votes_1976_party <- votes_1976 %>%
  group_by(party) %>%
  summarise(sum_votes=sum(candidatevotes,na.rm = T))
```