ds105-practice

Review practice file 7_dataframe_review.org

README

- **Open Emacs on this Org-mode file to code along!**
- **Look at your notes later to check what you did not get**
- **Challenge me to review/repeat topics that went by you**

TODO What is a data

What is a data frame (technically)?

> A rectangular data structure that has the VARIABLES (observables) of a data set as COLUMNS, and their values (observations) as ROWS as records. Variables can have different data types (unlike vectors).
>
> A `matrix` is also a rectangular data structure but its entries have only ONE data type and columns and rows are not really different.

TODO Creating a data frame of numeric values (numbers)

How can you create a data frame of two vectors with values `1 2 3 4 5 6 7 8 9 10`? What are the properties of this data frame?

```
data.frame(1:10,1:10)
```

```
   X1.10 X1.10.1
1      1       1
2      2       2
3      3       3
4      4       4
5      5       5
6      6       6
7      7       7
8      8       8
9      9       9
10    10      10
```

- Rows are automatically numbered
- Columns have default names `X1.10` and `X1.10.1`

TODO Creating a data frame from survey data

You've bought a new car. The car company sends you a survey. What kind of variables and corresponding data types do you expect?

- Name (of customer): `character`
- Make (of car): `character`
- Type (of car): `character`
- Year (of build): `character`
- Customer (returning or not): `logical`
- Price (of car): `numeric`
- Happiness (of customer): `factor`

How would you create such a data frame for a survey?

1. Create vectors with `c` or `factor`
2. Add vectors to data frame with `data.frame`
3. Store data frame in R object

Listing 1:

```
survey <- data.frame(
  "Name"="Birkenkrahe",
  "Make"="GMC",
  "Type"="Equinox",
  "Year"="2022",
  "Customer"=TRUE,
  "Price"=2,
  "Happiness"=factor("happy",
                order=TRUE,
                levels=c("unhappy", "neutral", "happy")))
survey
class(survey)
```

```
        Name Make    Type Year Customer Price Happiness
1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
[1] "data.frame"
```

TODO Which commands are used to explore data frames

Which R commands are used to explore data frames?

```
str(survey)  # data frame structure: variables and values
```

```
'data.frame':    1 obs. of  7 variables:
$ Name     : chr "Birkenkrahe"
$ Make     : chr "GMC"
$ Type     : chr "Equinox"
$ Year     : chr "2022"
$ Customer : logi TRUE
$ Price    : num 2
```

```
$ Happiness: Ord.factor w/ 3 levels "unhappy"<"neutral"<..: 3
```

```
head(survey) # or tail: tabular view of the top (or the bottom)
```

```
        Name Make    Type Year Customer Price Happiness
1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
```

```
summary(survey)  # statistical view of each variable incl. NA
```

```
   Name               Make               Type               Year
Length:1           Length:1           Length:1           Length:1
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character



Customer           Price     Happiness
Mode:logical   Min.   :2   unhappy:0
TRUE:1         1st Qu.:2   neutral:0
               Median :2   happy  :1
               Mean   :2
               3rd Qu.:2
               Max.   :2
```

TODO What do you do with missing values (`NA`)?

What about `NA`?

```
survey <- data.frame(
  "Name"="Birkenkrahe",
  "Make"="GMC",
  "Type"="Equinox",
  "Year"="2022",
  "Customer"=TRUE,
  "Price"=2,
  "Happiness"=factor("happy",
                order=TRUE,
                levels=c("unhappy", "neutral", "happy")))
s_na <- data.frame(survey,"missing"=NA) # add NA column to data frame
str(s_na)
```

```
'data.frame':    1 obs. of  8 variables:
$ Name     : chr "Birkenkrahe"
$ Make     : chr "GMC"
$ Type     : chr "Equinox"
$ Year     : chr "2022"
$ Customer : logi TRUE
$ Price    : num 2
$ Happiness: Ord.factor w/ 3 levels "unhappy"<"neutral"<..: 3
$ missing  : logi NA
```

```
summary(s_na$"missing")
```

```
  Mode    NA's
logical       1
```

TODO How do you extract values from a data frame?

1. Look at the variables to remind yourself of the data structure

```
   survey <- data.frame(
```

```
  "Name"="Birkenkrahe",
  "Make"="GMC",
  "Type"="Equinox",
  "Year"="2022",
  "Customer"=TRUE,
  "Price"=2,
  "Happiness"=factor("happy",
                order=TRUE,
                levels=c("unhappy", "neutral", "happy"))) ## calling a
named code block
str(survey)
```

```
'data.frame':    1 obs. of  7 variables:
$ Name     : chr "Birkenkrahe"
$ Make     : chr "GMC"
$ Type     : chr "Equinox"
$ Year     : chr "2022"
$ Customer : logi TRUE
$ Price    : num 2
$ Happiness: Ord.factor w/ 3 levels "unhappy"<"neutral"<..: 3
```

2. How do you get values from a data frame? For example:
   1. the first row
   2. the third column
   3. the fourth through fifth column
   4. a named column (like `Happiness` OR `Customer`)?
   5. two named columns (like `Happiness` AND `Customer`)

```
## using rownames (numbers)
survey[1,]  # first row
survey[,3]  # third column
survey[,4:5] # fourth through fifth column

## using colnames
survey[,"Type"]
survey[,c("Year","Customer")]
survey$Happiness
```

```
       Name Make    Type Year Customer Price Happiness
1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
[1] "Equinox"
 Year Customer
1 2022     TRUE
[1] "Equinox"
 Year Customer
1 2022     TRUE
[1] happy
Levels: unhappy < neutral < happy
```

TODO How do you add another row to the data frame?

How can you add another row to the data frame?

- add rows with the index operator `[]`
- add rows with `rbind(data_frame, vector)`

Tip: the index of row two (for all columns) would be `survey[2,]`

Tip: before messing with a data frame, make a copy

1. Add new row using `[]`. The values are stored in `row2`

   Listing 1:

```
row2 <- c("Birkenkrahe", "Kia", "Rio", "2023", FALSE, 1, "neutral")

## make a copy "new_survey" of the "survey" data frame
new_survey <- survey # always store intermediate results

## add row to your copy using [ ]
new_survey[2,] <- row2
new_survey
```

```
         Name Make    Type Year Customer Price Happiness
1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
2 Birkenkrahe  Kia     Rio 2023    FALSE     1   neutral
         Name Make    Type Year Customer Price Happiness
1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
2 Birkenkrahe  Kia     Rio 2023    FALSE     1   neutral
```

2. Add the same row again using `rbind`. The values are stored in `row2`

```
## add row using rbind
new_survey <- rbind(new_survey, row2)
new_survey
```

```
         Name Make    Type Year Customer Price Happiness
1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
2 Birkenkrahe  Kia     Rio 2023    FALSE     1   neutral
3 Birkenkrahe  Kia     Rio 2023    FALSE     1   neutral
```

TODO How do you remove a row from a data frame?

1. The data frame `new_survey` now has a double record in row 3. Print that row on its own first using `[]` to make sure, then repeat the command but add − before the index value

```
new_survey[3,]
new_survey[-3,]
```

```
         Name Make Type Year Customer Price Happiness
3 Birkenkrahe  Kia  Rio 2023    FALSE     1   neutral
         Name Make    Type Year Customer Price Happiness
1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
2 Birkenkrahe  Kia     Rio 2023    FALSE     1   neutral
```

2. Now overwrite `new_survey` accordingly

```
## overwriting new_survey with itself after removing row 3
new_survey <- new_survey[-3,]
new_survey
```

```
         Name Make    Type Year Customer Price Happiness
1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
2 Birkenkrahe  Kia     Rio 2023    FALSE     1   neutral
```

TODO How do you name rows of a data frame?

1. To name observations (rows) of a data frame, use `rownames`.
   - Save `new_survey` in a copy named `df`
   - Print all row names of `df` with `rownames`

```
df <- new_survey
rownames(df)
```

```
[1] "1" "2"
```

2. Now overwrite `rownames(df)` with new names, e.g. `Car_1` and `Car_2` and print the whole data frame to see the new names

```
rownames(df) <- c("Car_1", "Car_2")
df
```

```
            Name Make    Type Year Customer Price Happiness
Car_1 Birkenkrahe  GMC Equinox 2022     TRUE     2     happy
Car_2 Birkenkrahe  Kia     Rio 2023    FALSE     1   neutral
```

3. Now you can use the row names to index rows - print the second row only, using `[]`

```
df["Car_2",]
```

```
            Name Make Type Year Customer Price Happiness
Car_2 Birkenkrahe  Kia  Rio 2023    FALSE     1   neutral
```

TODO How do you rename column names?

1. For a data frame, the `names` function returns the same values as `colnames`. Print the column names of `df` using both functions

```
names(df)
colnames(df)
```

```
[1] "Name"      "Make"      "Type"      "Year"      "Customer"  "Price"
[7] "Happiness"
[1] "Name"      "Make"      "Type"      "Year"      "Customer"  "Price"
[7] "Happiness"
```

2. How can you check if these two vectors are really `identical`?

```
identical(names(df),colnames(df))
```

```
[1] TRUE
```

3. To change a column vector name means overwriting it. For example, change the name of the column `Customer` to `Account`.
   - Find the index of the column using `which`
   - Print the current `colnames` using the index value you found
   - Then overwrite its `colnames` value with the new name `Account`
   - Print the data frame to check the result

```
index <- which(colnames(df)=="Customer")
colnames(df)[index]
```

```
colnames(df)[index] <- "Account"
df
```

```
[1] "Customer"
            Name Make    Type Year Account Price Happiness
Car_1 Birkenkrahe  GMC Equinox 2022    TRUE     2     happy
Car_2 Birkenkrahe  Kia     Rio 2023   FALSE     1   neutral
```

TODO How can you `subset` observations?

1. How can you `subset` observations? E.g. for car types from 2023?

   Reminder: the arguments of `subset` are: input data frame, and a logical condition on the subset.

```
subset(df, Year==2023)
```

```
            Name Make Type Year Account Price Happiness
Car_2 Birkenkrahe  Kia  Rio 2023   FALSE     1   neutral
```

2. How can you extract the `Make` only from that `subset`?

   - The subset is a data frame, too. Store it in `dfs`
   - Now extract the column that corresponds to `Make`

```
dfs <- subset(df, Year==2023)
dfs[,"Make"]
subset(df, Year==2023)[,"Make"]
```

```
[1] "Kia"
[1] "Kia"
```

TODO How can you clean up after a session?

Remove objects from the current session using `rm`.

- Run `ls()` to see your currently loaded R objects
- Remove `new_survey` by feeding it to `rm`
- Run `ls()` again to see your currently loaded R objects
- Run `rm(list=ls())` to remove all remaining objects
- Run `ls()` again to see the result

```
ls()
rm(new_survey)
ls()
```

```
[1] "df"         "dfs"        "idx"        "index"      "new_survey"
[6] "row2"       "s_na"       "survey"     "tg"
[1] "df"     "dfs"    "idx"    "index"  "row2"   "s_na"   "survey" "tg"
```