# RDataWrangling

Çağrı Çebişli

2023-01-22

## Useful Links

https://www.uvm.edu/~tdonovan/RforFledglings/data-wrangling-with-dplyr.html

## For Importing Data

### Folders, Working Directory, Shortcuts

- Shortcut for chunk: Command + Option + I
- Invisible Chunk: r,warning=FALSE,error=FALSE,message=FALSE,include=FALSE, Echo=FALSE

**Sets Working Directory to Saved Folder:**

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

library(here): read_xlsx(here("folder name", "data to import.xlsx"))

After you set your working directory to saved folder, you can move around with using here package. Thus, this operation is good for conducting reproducible research. You do not talk about /c:cagri etc., only move around folders, so folder name is important here.

### Exploratory Data Analysis

**EDA Report DlookR**

library(dlookr)

EDA Report

data %>% eda_report(output_format = "html", output_file = "EDA_diamonds.html")

Very good exploratory data analysis page. Do take notes while reading this, write down ideas and disaggregations.

**Crosstabs**

Good crosstab to see frequency, include NA to understand overall column.

**Some x-tab examples**

(table(data$column,useNA = "always"))

prop.table(table(data$column,useNA="always"))

CD_WFP %>% group_by(province) %>% summarize(AvgTR_Month = mean(Arrival_Differences,na.rm = T), CampRes = mean(camp_residence_how_long,na.rm = T))

---

aggregate(flowers[, 4:7], by = list(nitrogen = flowers$nitrogen, treat = flowers$treat), FUN = mean) ***

## Column Re-arrange

Extract month from character saved date variable.

Dots in format can change into: % , / , etc.

month(as.POSIXlt(data$date_as_character_variable, format="%d.%m.%Y"))

## Select Columns - Arrange - Filter

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
starwars %>%
  mutate(name, bmi = mass / ((height / 100)  ^ 2)) %>%
  select(name:mass, bmi)
```

```
## # A tibble: 87 x 4
##    name               height  mass   bmi
##    <chr>               <int> <dbl> <dbl>
##  1 Luke Skywalker        172    77  26.0
##  2 C-3PO                 167    75  26.9
##  3 R2-D2                  96    32  34.7
##  4 Darth Vader           202   136  33.3
##  5 Leia Organa           150    49  21.8
##  6 Owen Lars             178   120  37.9
##  7 Beru Whitesun lars    165    75  27.5
##  8 R5-D4                  97    32  34.0
##  9 Biggs Darklighter     183    84  25.1
## 10 Obi-Wan Kenobi        182    77  23.2
## # ... with 77 more rows
```

```
starwars %>%
  arrange(desc(mass))
```

```
## # A tibble: 87 x 14
##    name        height  mass hair_~1 skin_~2 eye_c~3 birth~4 sex   gender homew~5
##    <chr>        <int> <dbl> <chr>   <chr>   <chr>     <dbl> <chr> <chr>  <chr>
##  1 Jabba Desi~    175  1358 <NA>    green-~ orange    600   herm~ mascu~ Nal Hu~
##  2 Grievous       216   159 none    brown,~ green,~    NA   male  mascu~ Kalee
##  3 IG-88          200   140 none    metal   red        15   none  mascu~ <NA>
##  4 Darth Vader    202   136 none    white   yellow     41.9 male  mascu~ Tatooi~
##  5 Tarfful        234   136 brown   brown   blue       NA   male  mascu~ Kashyy~
##  6 Owen Lars      178   120 brown,~ light   blue       52   male  mascu~ Tatooi~
##  7 Bossk          190   113 none    green   red        53   male  mascu~ Trando~
##  8 Chewbacca      228   112 brown   unknown blue      200   male  mascu~ Kashyy~
##  9 Jek Tono P~    180   110 brown   fair    blue       NA   male  mascu~ Bestin~
## 10 Dexter Jet~    198   102 none    brown   yellow     NA   male  mascu~ Ojom
## # ... with 77 more rows, 4 more variables: species <chr>, films <list>,
## #   vehicles <list>, starships <list>, and abbreviated variable names
## #   1: hair_color, 2: skin_color, 3: eye_color, 4: birth_year, 5: homeworld
```

```
starwars %>%
  group_by(species) %>%
  summarise(
    n = n(),
    mass = mean(mass, na.rm = TRUE)
  ) %>%
  filter(
    n > 1,
    mass > 50
  )
```

```
## # A tibble: 8 x 3
##   species      n  mass
##   <chr>    <int> <dbl>
## 1 Droid        6  69.8
## 2 Gungan       3  74
## 3 Human       35  82.8
## 4 Kaminoan     2  88
## 5 Mirialan     2  53.1
## 6 Twi'lek      2  55
## 7 Wookiee      2 124
## 8 Zabrak       2  80
```

## Subset

subset(flowers, treat == "tip" & nitrogen == "medium" & block == 2, select = c("treat", "nitrogen", "leafarea"))

## Reshape- Long and Wide Data

### Wide Data into Long Data

```
wide_data <- data.frame(subject = c("A", "B", "C", "D"),
                sex = c("M", "F", "F", "M"),
                control = c(12.9, 5.2, 8.9, 10.5),
                cond1 = c(14.2, 12.6, 12.1, 12.9),
                cond2 = c(8.7, 10.1, 14.2, 11.9))
wide_data
```

```
##   subject sex control cond1 cond2
## 1       A   M    12.9  14.2   8.7
## 2       B   F     5.2  12.6  10.1
## 3       C   F     8.9  12.1  14.2
## 4       D   M    10.5  12.9  11.9
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
my_long_df <- melt(data = wide_data, id.vars = c("subject", "sex"),
                   measured.vars = c("control", "cond1", "cond2"),
                   variable.name = "condition", value.name = "measurement")
my_long_df
```

```
##    subject sex condition measurement
## 1        A   M   control        12.9
## 2        B   F   control         5.2
## 3        C   F   control         8.9
## 4        D   M   control        10.5
## 5        A   M     cond1        14.2
## 6        B   F     cond1        12.6
## 7        C   F     cond1        12.1
## 8        D   M     cond1        12.9
## 9        A   M     cond2         8.7
## 10       B   F     cond2        10.1
## 11       C   F     cond2        14.2
## 12       D   M     cond2        11.9
```

### Long Data into Wide Data

```
long_data <- data.frame(
             subject = rep(c("A", "B", "C", "D"), each = 3),
             sex = rep(c("M", "F", "F", "M"), each =3),
             condition = rep(c("control", "cond1", "cond2"), times = 4),
             measurement = c(12.9, 14.2, 8.7, 5.2, 12.6, 10.1, 8.9,
                             12.1, 14.2, 10.5, 12.9, 11.9))
head(long_data,5)
```

```
##   subject sex condition measurement
## 1       A   M   control        12.9
## 2       A   M     cond1        14.2
## 3       A   M     cond2         8.7
## 4       B   F   control         5.2
## 5       B   F     cond1        12.6
```

```
my_wide_df <- dcast(data = long_data, subject + sex ~ condition,
                    value.var = "measurement")
my_wide_df
```

```
##   subject sex cond1 cond2 control
## 1       A   M  14.2   8.7    12.9
## 2       B   F  12.6  10.1     5.2
## 3       C   F  12.1  14.2     8.9
## 4       D   M  12.9  11.9    10.5
```