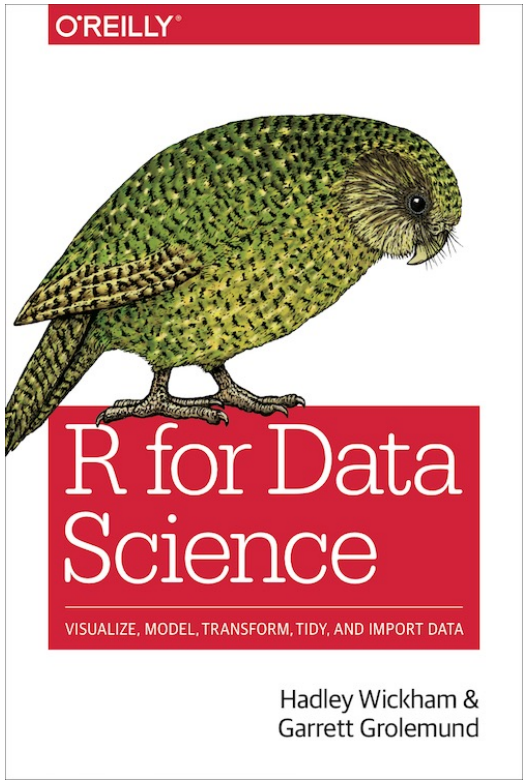
University of Applied Sciences and Arts Northwestern Switzerland
School of Business

Data Science - Exercises

Holger Wache

Exercise B

Data Wrangling with the following book



- "R for Data Science: Import, Tidy, Transform, Visualize, and Model Data" by Garrett Grolemund and Hadley Wickham
- https://r4ds.had.co.nz/index.html
- Want to buy? http://amzn.to/2aHLAQ1

- HERE: Chapter 5 "Data Transformation"

Exercise B

Missing Values

Prerequisites

First, install additional packages

1. an additional package containing flights and
2. an additional package containing nice functions

```
install.packages("nycflights13")
install.packages("tidyverse")
```

After the installation these packages need to be "activated"

```
library(nycflights13)
library(tidyverse)
```

See the data “flights”

```
> flights
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier flight
tailnum
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl> <chr>    <int> <chr>
 1  2013     1     1      517            515         2      830            819        11 UA        1545 N14228
 2  2013     1     1      533            529         4      850            830        20 UA        1714 N24211
 3  2013     1     1      542            540         2      923            850        33 AA        1141 N619AA
 4  2013     1     1      544            545        -1     1004           1022       -18 B6         725 N804JB
 5  2013     1     1      554            600        -6      812            837       -25 DL         461 N668DN
 6  2013     1     1      554            558        -4      740            728        12 UA        1696 N39463
 7  2013     1     1      555            600        -5      913            854        19 B6         507 N516JB
 8  2013     1     1      557            600        -3      709            723       -14 EV        5708 N829AS
 9  2013     1     1      557            600        -3      838            846        -8 B6          79 N593JB
10  2013     1     1      558            600        -2      753            745         8 AA         301 N3ALAA
# … with 336,766 more rows, and 7 more variables: origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>

> count(flights)
# A tibble: 1 x 1
      n
  <int>
1 336776
```

Filtering the data

The function “filter()” allows to select some data lines (sets) with respect to some conditions.
For example all flights in the first month:

```
> filter(flights, month == 1)
...
```

Several conditions can also be combined:

```
> filter(flights, month == 1 & day == 1)
...
```

Identifying Missing Values: Deleting the data row

You can also simply remove the rows with missing values with the function “filter()” . Just negate the “is,na()” test, i.e. select all data lines which DON’T contain a missing value:

```
> filter(flights, ! is.na(dep_time))
...
```

… and counting them

```
count(filter(flights, ! is.na(dep_time)))
# A tibble: 1 x 1
      n
  <int>
1 328521
```

Remembering the data set

You can save the data set and can make it available with a (different) name

```
> my_flights <- filter(flights, ! is.na(dep_time))
```

The variable “my_flights” now contains the data set where the rows with missing values in the ”dep_time” are removed.

Identifying Missing Values: Replacing them (1/4)

Instead of deleting the rows with missing values you can also replace the missing values.

For example for the “dep_time”, there is a scheduled departure time “sched_dep_time” given. That value can be used for replacement.
First make a copy of “flights”

```
> flights_with_replaced_dep_time <- flights
```

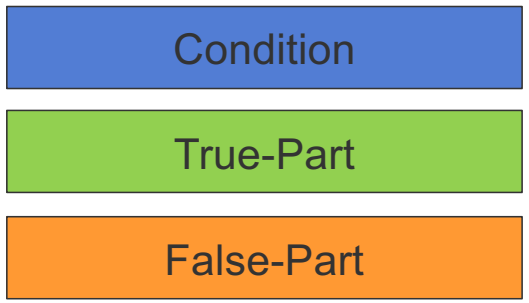
Identifying Missing Values: Replacing them (2/4)

Then address the attribute departure time `dep_time` in `flights_with_replaced_dep_time`, i.e. `flights_with_replaced_dep_time$dep_time`.

The function “ifelse”

- test a condition (here: `is.na(flights$dep_time)`).
- If the test succeeds, then the value from `flights$sched_dep_time` taken.
- Otherwise the original `flights$dep_time`

```
flights_with_replaced_dep_time$dep_time <-
        ifelse(is.na(flights$dep_time),
                flights$sched_dep_time,
                flights$dep_time)
```



Identifying Missing Values: Replacing them (3/4)

Check it with the following example:

```
> filter(flights, tailnum == "N18120")
# A tibble: 134 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier flight tailnum
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl> <chr>    <int> <chr>
 1  2013     1     1     1842           1422       260     1958           1535       263 EV        4633 N18120
 2  2013     1     1       NA           1630        NA       NA           1815        NA EV        4308 N18120
 3  2013     1     2      836            751        45     1059           1001        58 EV        4420 N18120
```

That would be with all NA's. But in

```
> filter(flights_with_replaced_dep_time, tailnum == "N18120")
# A tibble: 134 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier flight tailnum
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl> <chr>    <int> <chr>
 1  2013     1     1     1842           1422       260     1958           1535       263 EV        4633 N18120
 2  2013     1     1     1630           1630        NA       NA           1815        NA EV        4308 N18120
 3  2013     1     2      836            751        45     1059           1001        58 EV        4420 N18120
```

… we replaced one NA.

Identifying Missing Values: Replacing them (4/4)

You can also replace NA's with other values, e.g. 12:00

```
replacement <- 1200

flights_with_replaced_dep_time$dep_time <-
  ifelse(is.na(flights$dep_time),
         replacement,
         flights$dep_time)
```

… or replacing it by the mean (remove NA: "na.rm = TRUE"; convert the mean to an integer: "as.integer"):

```
replacement <- as.integer(mean(flights$dep_time, na.rm = TRUE))

flights_with_replaced_dep_time$dep_time <-
  ifelse(is.na(flights$dep_time),
         replacement,
         flights$dep_time)
```

Exercise B

Outliers

Identifying and Eliminating Outliers (1/2)

Lets analyse the departure delay:

```
> ggplot(flights) + geom_point(mapping = aes(x = flight, y = dep_delay))
```

There are some flights which really depart earlier than scheduled.

```
> arrange(flights, dep_delay)
year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay carrier flight tailnum
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>     <dbl> <chr>    <int> <chr>
 1  2013    12     7     2040           2123       -43       40           2352        48 B6          97 N592JB
 2  2013     2     3     2022           2055       -33     2240           2338       -58 DL        1715 N612DL
 3  2013    11    10     1408           1440       -32     1549           1559       -10 EV        5713 N825AS
 4  2013     1    11     1900           1930       -30     2233           2243       -10 DL        1435 N934DL
 5  2013     1    29     1703           1730       -27     1947           1957       -10 F9         837 N208FR
```

We catch them:

```
> minus_delay <- filter(flights, dep_delay <= 0)
```

Identifying and Eliminating Outliers (2/2)

Analyse the distribution of the negative departure delay:

```
> boxplot(minus_delay$dep_delay)
```

Some values are out of range. We remove all lines with have a lower negative departure delay than 29:

```
> my_flights <- filter(flights, dep_delay > -29)
```