University of Applied Sciences and Arts Northwestern Switzerland
School of Business

member of swissuniversities

Data Science - Exercises

Holger Wache

Exercise C

Transformation and Normalisation

Prerequisites

If not installed so far, please install the following 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”, also when you start R (or R-Studio)

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

Deleting columns

Deleting a column, e.g. dep_delay

```
> my_flight <- subset(flights,select=-dep_delay)
> my_flights
# A tibble: 336,776 × 18
    year month   day dep_time sched_dep_time arr_time sched_arr_time arr_delay carrier flight tailnum
   <int> <int> <int>    <int>          <int>    <int>          <int>     <dbl> <chr>    <int> <chr>
 1  2013     1     1      517            515      830            819        11 UA        1545 N14228
 2  2013     1     1      533            529      850            830        20 UA        1714 N24211
 3  2013     1     1      542            540      923            850        33 AA        1141 N619AA
 ...
```

Deleting several columns, e.g. dep_delay and flight

```
> my_flight <- subset(flights,select=-c(dep_delay,flight))
> my_flights
# A tibble: 336,776 × 17
    year month   day dep_time sched_dep_time arr_time sched_arr_time arr_delay carrier tailnum origin
   <int> <int> <int>    <int>          <int>    <int>          <int>     <dbl> <chr>   <chr>   <chr>
 1  2013     1     1      517            515      830            819        11 UA      N14228  EWR
 2  2013     1     1      533            529      850            830        20 UA      N24211  LGA
 3  2013     1     1      542            540      923            850        33 AA      N619AA  JFK
 ...
```

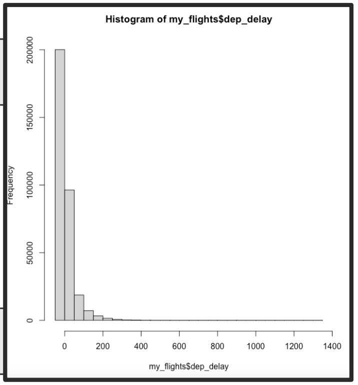
Transforming departure delay (1/2)

First remove all rows with missing values and remove rows with extreme negative delay

```
> my_flights <- filter(flights, ! is.na(dep_time))
> my_flights <- filter(my_flights, dep_delay > -29)
```

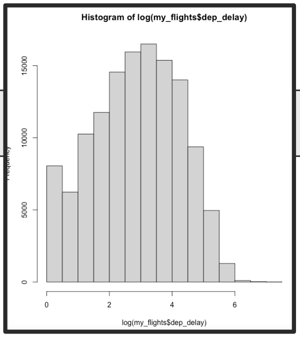
We now analyse the distribution

```
> hist(my_flights$dep_delay)
```



Looks imbalanced, looks like a logarithmic distribution; converting it to a more uniform distribution ..

```
> hist(log(my_flights$dep_delay))
```

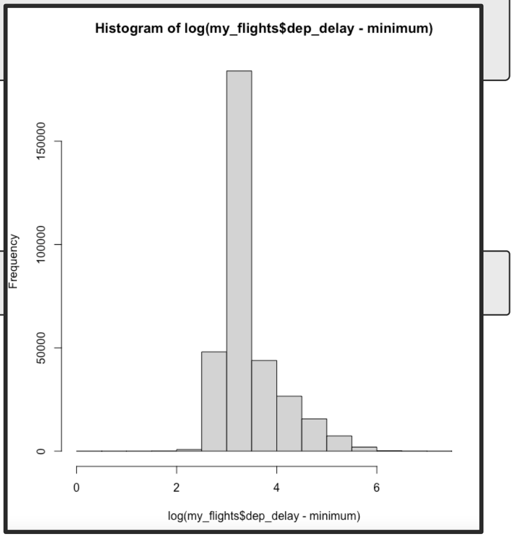


Now it looks better, but we produce NA (for the negative delays; log is not defined for negative inputs)

Transforming departure delay (2/2)

In order to remove negative values – and don't want to delete them – we simply shift the delays by the most negative value (i.e. the minimum). Then all values are positive.

```
> minimum <- min(my_flights$dep_delay,na.rm = TRUE)
> hist(log(my_flights$dep_delay - minimum))
```



Looks better now; we keep it!

```
> my_flights$dep_delay <- log(my_flights$dep_delay - minimum)
```

Normalising departure time (1/2)

First remove all rows with missing values

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

We apply the min-max normalisation to “dep_time” (assuming a range of 0000-2359).

The new min is 0 and the new max is 1. Then $\frac{v-0000}{(2359-0000)} * (1-0)+0 = \frac{v}{2359}$

```
> my_flights$dep_time <- my_flights$dep_time / 2359
> my_flights
# A tibble: 328,521 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>    <dbl>          <int>     <dbl>    <int>          <int>     <dbl> <chr>    <int> <chr>
 1  2013     1     1    0.219            515         2      830            819        11 UA        1545 N14228
 2  2013     1     1    0.226            529         4      850            830        20 UA        1714 N24211
 3  2013     1     1    0.230            540         2      923            850        33 AA        1141 N619AA
...
```

Normalising departure time (2/2)

However the coding of time in integer is not continuous. E.g. 1178 would never exists. We need a (self-defined) conversion function "time_conversion", which translates that into continuous numbers

```
> time_conversion <- function(x) {
h <- trunc(x/100,0)
m <- x-(h*100)
r <- m+(h*60)
return(r)
}
```

```
> my_flights$dep_time <- time_conversion(my_flights$dep_time) / (24*60)
> my_flights
# A tibble: 328,521 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>    <dbl>          <int>     <dbl>    <int>          <int>     <dbl> <chr>    <int> <chr>
 1  2013     1     1    0.220            515         2      830            819        11 UA        1545 N14228
 2  2013     1     1    0.231            529         4      850            830        20 UA        1714 N24211
 3  2013     1     1    0.238            540         2      923            850        33 AA        1141 N619AA
...
```