# M1 Mini Assignment 1

*Lars Nielsen*

*9/9/2019*

## Contents



## M1 Mini-Assignment 1

**Link to github .Rmd https://github.com/LarsHernandez/SDS-Projects-2019/blob/master/M1_assignment_1/assignment.Rmd**

You are given 2 datasets from https://nomadlist.com/ - A community page for remote workersworldwide.

- The trips data holds ~46k individual trips of travellers on the platform https://github.com/SDS-AAU/M1-2019/raw/master/data/trips.csv
- People contains some personal information on 4k travelers https://github.com/SDS-AAU/M1-2019/raw/master/data/people.csv
- Finally, you find a countrylist file that holds countrycodes, contrynames and region-associations https://github.com/SDS-AAU/M1-2019/raw/master/data/countrylist.csv

```
library(data.table)
library(ggplot2)
library(magrittr)

# I load the data with fread (to the one doing the peer review i'm sorry but the
# assignment is done in data.table, hope you can understand what is happening, also
# ill just do notes here in the code chunks, and only where i don't think it's
# obvious what is happening)

trips       <- fread("https://github.com/SDS-AAU/M1-2019/raw/master/data/trips.csv")
people      <- fread("https://github.com/SDS-AAU/M1-2019/raw/master/data/people.csv")
countrylist <- fread("https://github.com/SDS-AAU/M1-2019/raw/master/data/countrylist.csv")
```

Your solution approach is more important than the results obtained! Comment your notebook well, explaining all the steps of your analysis. Small technicalexplanations can go as comments in the code. Broader explanations should be inserted asmarkdown cells.Remember that notebooks execute sequentially.

Submission: Wednesday 11.9. 12:00.

Peergrade.io (link + submission details will be sent out onMonday, 9.9)

## 1. Preprocessing

### a. Trips: transform dates into timestamps

(note: in Python, you will have to 'coerce' errors forfaulty dates)

```
trips[,c("date_end", "date_start") := .(as.Date(date_end, "%Y-%m-%d"),
                                        as.Date(date_start, "%Y-%m-%d"))]
class(trips$date_end)
```

```
## [1] "Date"
```

```
head(trips$date_end, 5)
```

```
## [1] "2018-06-15" "2018-06-03" "2017-11-05" "2017-08-07" "2017-03-18"
```

### b. Calculate trip duration in days

(you can use loops, list comprehensions ormap-lambda-functions (python) to create a column that holds the numerical value of theday. You can also use the "datetime" package.)

```
trips[, dur_days := date_end - date_start]

class(trips$dur_days)
```

```
## [1] "difftime"
```

```
head(trips$dur_days, 5)
```

```
## Time differences in days
## [1] 11  3  4 14 29
```

### c. Filter extreme (fake?) observations

for durations as well as dates - start and end (trips thatlast 234565 days / are in the 17th or 23rd century)The minimum duration of a trip is 1 day!Hint: use percentiles/quantiles to set boundaries for extreme values - between 1 and 97, calculate and store the boundaries before subsetting. Rhint: Use percent_rank(as.numeric(variable)) to create percentiles

```
trips[, quantile := dplyr::percent_rank(as.numeric(trips$dur_days))]
trips_s <- trips[quantile >= 0.01 & quantile <= 0.97]

# The range before and after the subsetting by the quantiles:
range(na.omit(trips$dur_days))
```

```
## Time differences in days
## [1] -730484  731122
```

```
range(trips_s$dur_days)
```

```
## Time differences in days
## [1]   1 208
```

### d. Join the countrylist data

to the trips data-frame using the countrycode as a key

```
countrylist[, country_code := alpha_2]
# United Kingdom coded as both UK and GB
trips_s$country_code[trips_s$country_code == "UK"] <- "GB"
# Empty country codes coded as africa in countrylist
trips_s$country_code[trips_s$country_code == ""]   <- "empty"

trips_s[countrylist, on = "country_code",
        c("region", "sub_region") := .(i.region, i.sub_region)]

head(trips_s[,.(country_code, dur_days, region, sub_region)])
```

```
##    country_code dur_days   region                      sub_region
## 1:           MX  11 days Americas Latin America and the Caribbean
## 2:           MX   3 days Americas Latin America and the Caribbean
## 3:           MX   4 days Americas Latin America and the Caribbean
## 4:           JO  14 days     Asia                    Western Asia
## 5:           CN  29 days     Asia                    Eastern Asia
## 6:           VN 167 days     Asia              South-eastern Asia
```

## 2. People

### a. How many people have at least a "High School" diploma?

```
people_s <- people[education_raw != ""]
table(people_s$education_raw)
```

```
##
##                                Bachelor's Degree
##                                              197
##              Bachelor's Degree, Master's Degree
##                                                9
##                                      High School
##                                               58
##                   High School, Bachelor's Degree
##                                               43
## High School, Bachelor's Degree, Master's Degree
##                                               29
##                                  Master's Degree
##                                              115
```

```
paste0("There are ",people_s[,.N], " individuals in the dataset now that has atleast High School")
```

```
## [1] "There are 451 individuals in the dataset now that has atleast High School"
```

### b. How many people working with "Software Dev" have a "Master's Degree"?

```
res <- people_s[work_raw %like% "Software Dev" & education_raw %like% "Master", .N]

paste0("There are ", res, " induviduals who work with software development and have a masters degree")
```

```
## [1] "There are 57 induviduals who work with software development and have a masters degree"
```

### c. Who is the person ...

with a Master's Degree that has the highest number of followers?[Explore who this person is. :-) ]

```
res <- people_s[education_raw %like% "Master"][order(-followers)]

head(res[,c("username", "followers")])
```

```
##                username followers
## 1:            @levelsio      2182
## 2:                 @aaz       259
## 3:            @neosilky       102
## 4:        @zackllnyoung        60
## 5:            @html5cat        32
## 6: @siddharthkshetrapal        29
```

```
people_s[username == "@levelsio"]
```

```
##      V1  username followers following
## 1: 2043 @levelsio      2182       353
##                                   work_raw
## 1: Software Dev, Startup Founder, Creative
##                                      education_raw
## 1: High School, Bachelor's Degree, Master's Degree
```

## 3. Trips

### a. Which country received the highest number of trips?

```
a <- trips_s[, .N, by = country][order(-N)]
head(a)
```

```
##           country    N
## 1:  United States 6963
## 2:       Thailand 3278
## 3: United Kingdom 2053
## 4:          Spain 1875
## 5:        Germany 1814
## 6:         France 1390
```

### b. Which country received the highest number of trips in 2017?

Use the start of trips as a timereference. (python: use datetimeindex created in 1 as a selector)?Rhint: Use functions from lubridate package to extract year.

```
b <- trips_s[year(date_start) == 2017, .N, by = country][order(-N)]
head(b)
```

```
##           country    N
## 1:  United States 1823
## 2:       Thailand  894
## 3: United Kingdom  612
## 4:          Spain  598
## 5:        Germany  456
## 6:         France  391
```

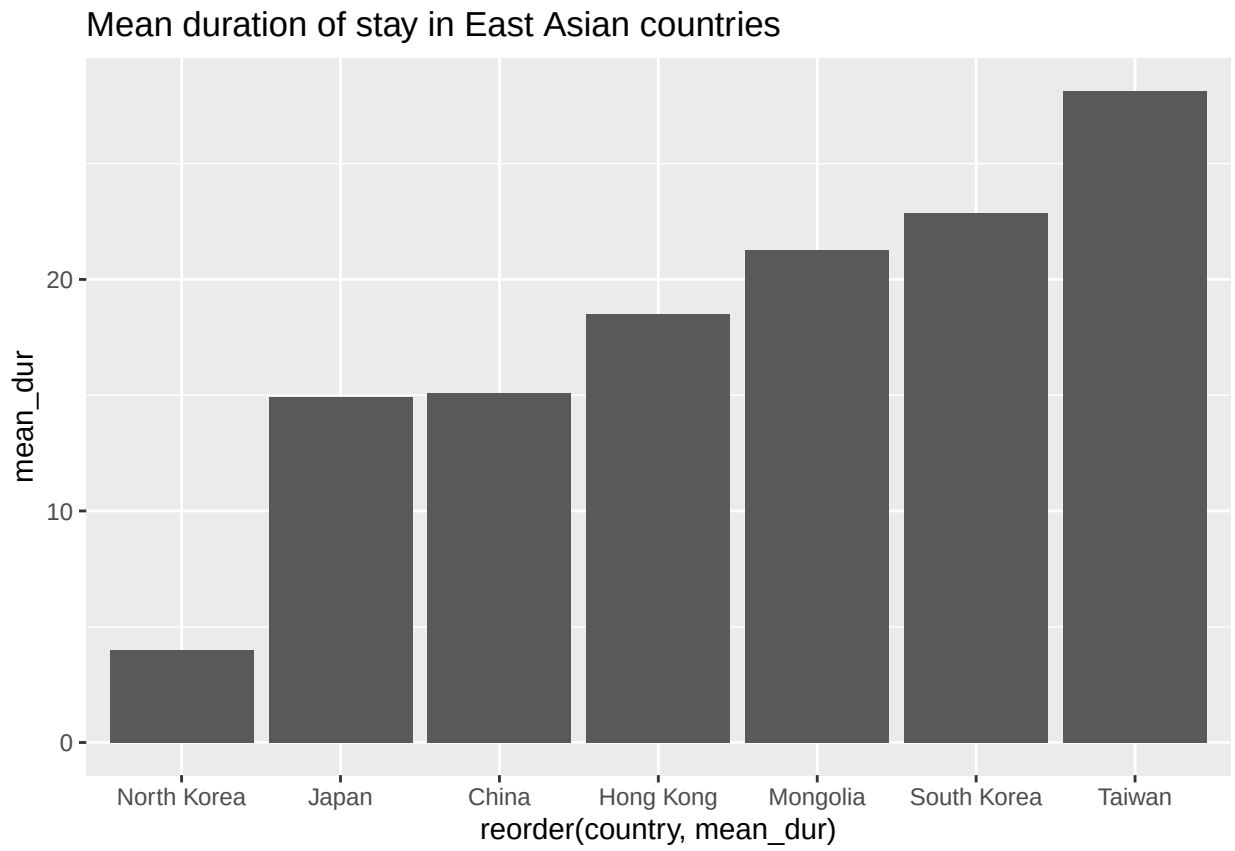
**c. Which is the country in 'Eastern Asia' ...**

where travellers spent on average least time whengoing there? Provide a visualization.

```
c <- trips_s[sub_region == 'Eastern Asia', .(mean_dur = mean(dur_days), total = .N), by = country][orde
c
```

```
##        country      mean_dur total
## 1: North Korea  4.00000 days     9
## 2:       Japan 14.90731 days   971
## 3:       China 15.07692 days  1066
## 4:   Hong Kong 18.50000 days     2
## 5:    Mongolia 21.27778 days    18
## 6: South Korea 22.85465 days   344
## 7:      Taiwan 28.15599 days   359
```

```
ggplot(c, aes(reorder(country, mean_dur), mean_dur)) +
  geom_col() +
  scale_y_continuous() +
  labs(title = "Mean duration of stay in East Asian countries")
```




**d. Do nomads that ...**

indicate working in "Software Dev" tend to have shorter or longer trips onaverage?

```
trips_s[people, on = "username", c("work_raw") := .(i.work_raw)]
trips_s[, dev := work_raw %like% "Software Dev"]
```

```
trips_s[,.(mean_dur = mean(dur_days)), by = dev]
```

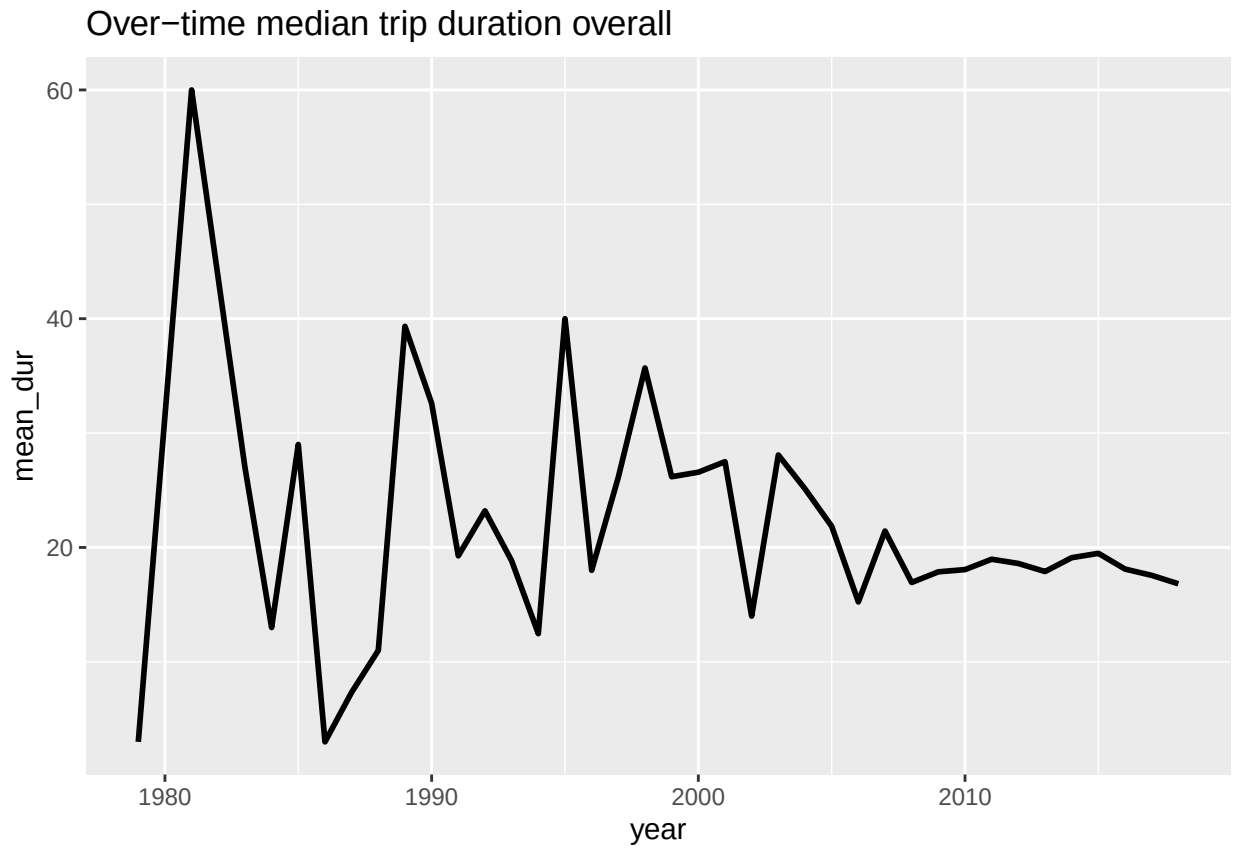
```
##      dev      mean_dur
## 1:  TRUE 16.88626 days
## 2: FALSE 18.36551 days
```

### e. Visualize over-time median trip duration

overall (bonus: and split by world-region).You will get a weird looking plot :-)

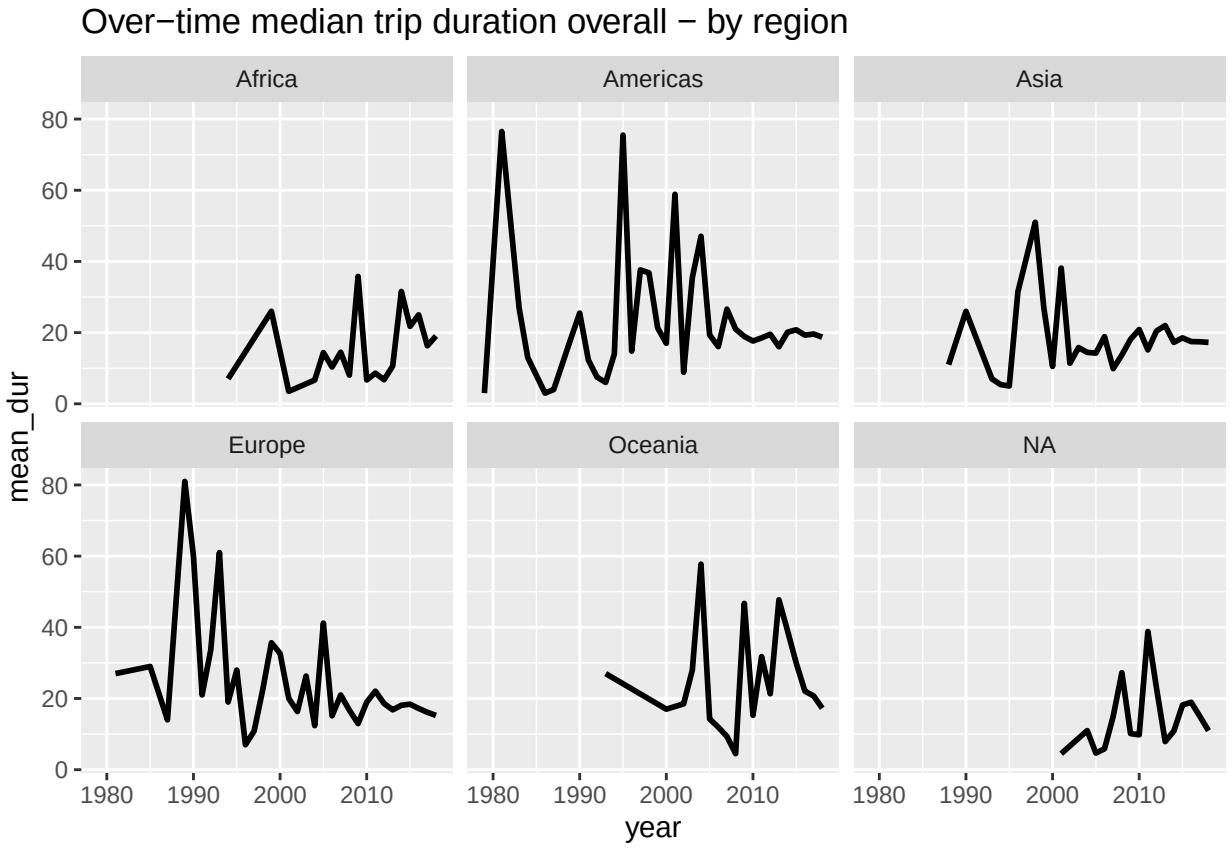
```
e1 <- trips_s[, .(mean_dur = mean(dur_days)), by = year(date_start)][
  year>1970 & year < 2019]

ggplot(e1, aes(year, mean_dur)) +
  scale_y_continuous() +
  geom_line(size=1) +
  labs(title="Over-time median trip duration overall")
```




```
e2 <- trips_s[, .(mean_dur = mean(dur_days)), by = .(year(date_start), region)][
  year>1970 & year < 2019]

ggplot(e2, aes(year, mean_dur)) +
  scale_y_continuous() +
  geom_line(size=1) +
  facet_wrap(~region) +
  labs(title="Over-time median trip duration overall - by region")
```




```
# How manny NA's do we still have?
table(trips_s$region, useNA="always")
```

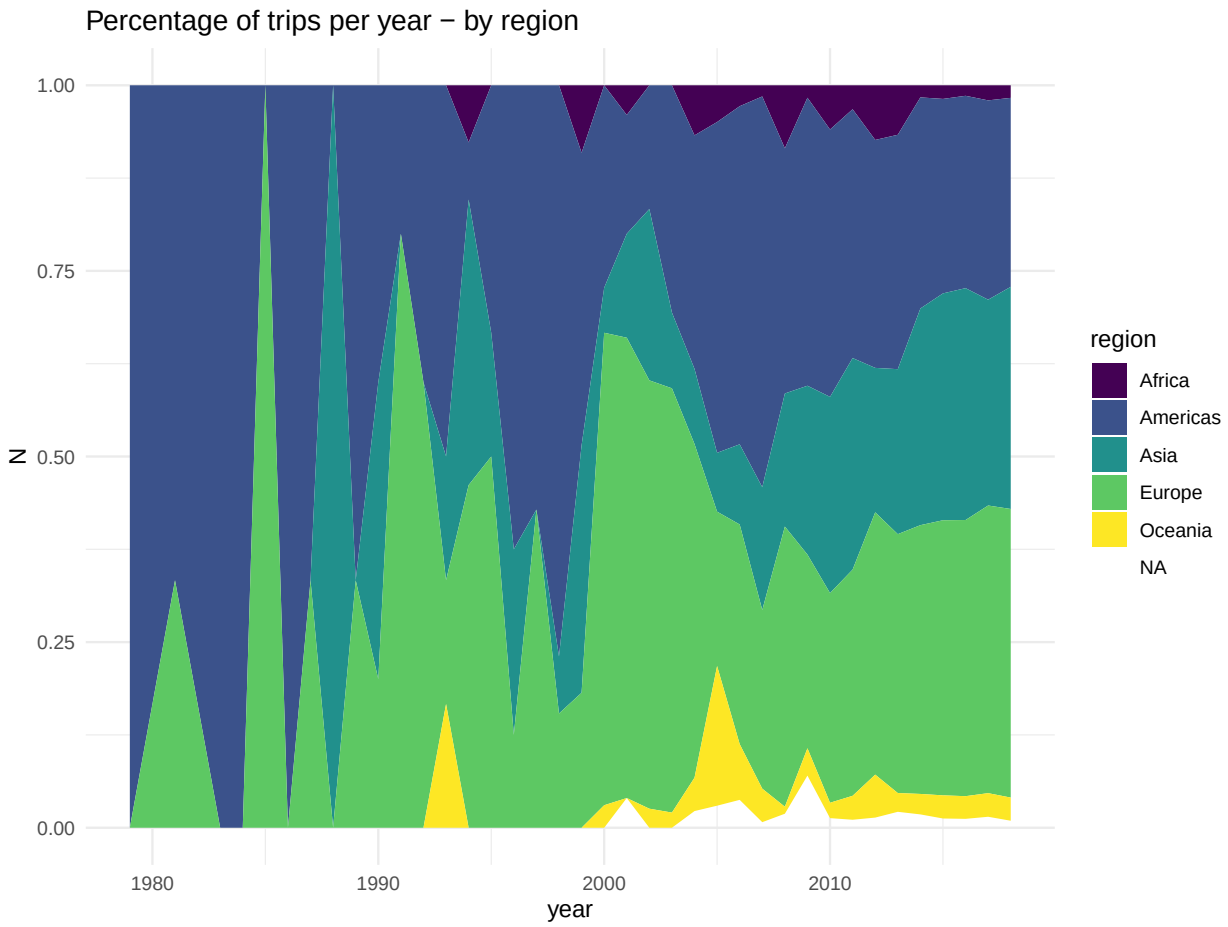
```
##
##   Africa Americas     Asia   Europe  Oceania     <NA>
##      921    11774    12449    16218     1379      594
```

```
# What are those codes that it doesn't know
vec <- sort(table(trips_s$country_code, is.na(trips_s$region))[,2])
subset(vec, vec >0)
```

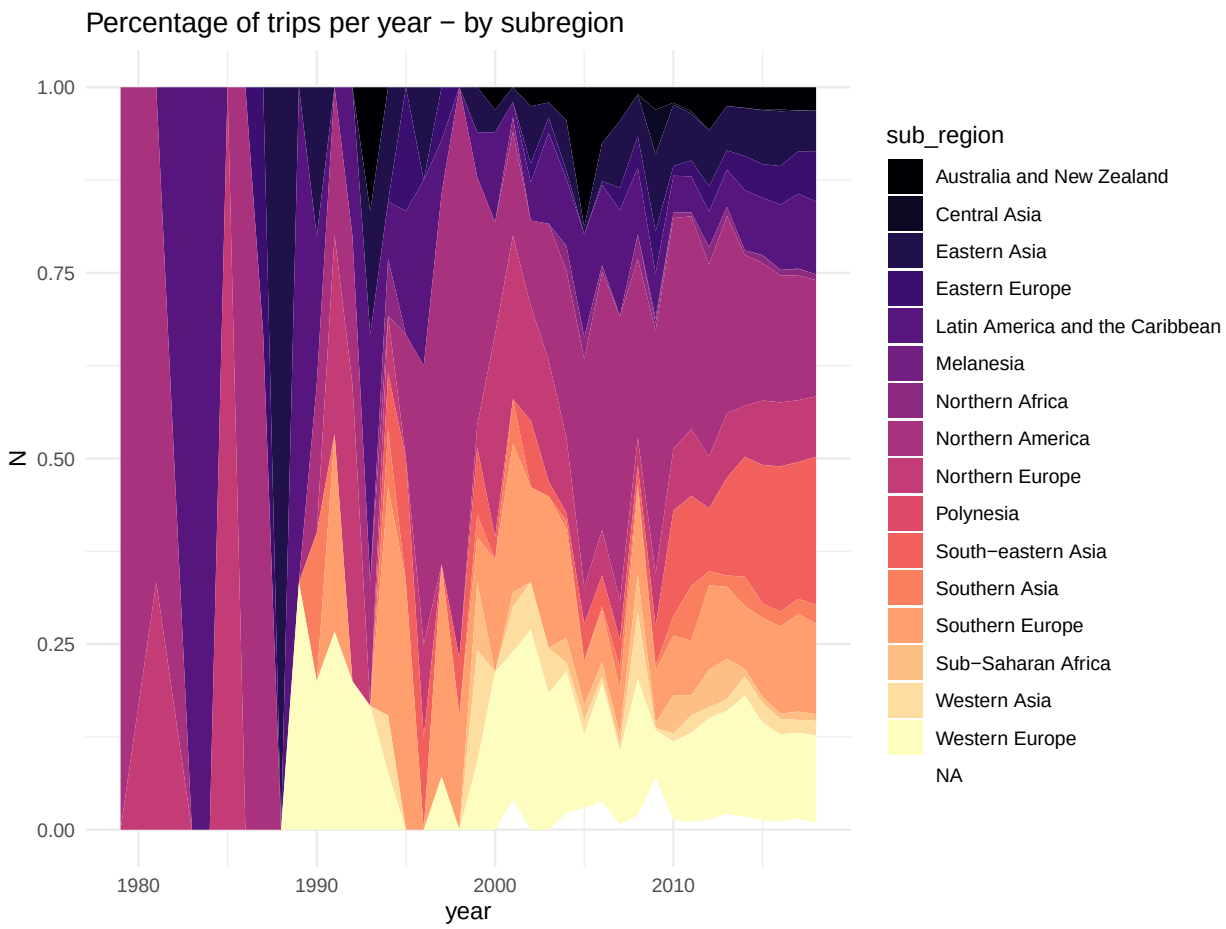
```
##    CT    OI    KS    CB    VB    AA    IA empty
##     1     1     4     5     5    13    15   550
```

```
# Two extra plots

trips_s[, .N, by = .(year(date_start), region)][year>1970 & year < 2019] %>%
  tidyr::complete(year, tidyr::nesting(region), fill = list(N = 0)) %>%
  ggplot(aes(year, N, fill = region)) +
  scale_fill_viridis_d() +
  geom_area(position = "fill") +
  labs(title="Percentage of trips per year - by region") +
  theme_minimal()
```




```
trips_s[, .N, by = .(year(date_start), sub_region)][year>1970 & year < 2019] %>%
  tidyr::complete(year, tidyr::nesting(sub_region), fill = list(N = 0)) %>%
  ggplot(aes(year, N, fill = sub_region)) +
  scale_fill_viridis_d(option="magma") +
  geom_area(position = "fill") +
  labs(title="Percentage of trips per year - by subregion") +
  theme_minimal()
```


Percentage of trips per year − by subregion
1.00
0.75
0.50
0.25
0.00
N
1980
1990
2000
2010
year
sub_region
Australia and New Zealand
Central Asia
Eastern Asia
Eastern Europe
Latin America and the Caribbean
Melanesia
Northern Africa
Northern America
Northern Europe
Polynesia
South−eastern Asia
Southern Asia
Southern Europe
Sub−Saharan Africa
Western Asia
Western Europe
NA