Fachhochschule Südwestfalen
University of Applied Sciences

Programmierung R - Exercise

Software Development

May 3, SS 2022 || Hannah Behrens

Wir geben Impulse

Today’s problem - tourism data

We will work with the Australian `tourism` data provided by Hyndman and Athanasopoulos (2021): The file can be downloaded **here** or **here** and read into R with `readxl::read_excel()` (Wickham and Bryan 2019) (see announcement from May 2, 2022 in moodle).

The original tourism data was published by the Tourism Research Division (Tourism Research Australia) of the Australian Trade and Investment Commission of the Australian Government.

Task - Understanding the tourism data set

Your turn

Make yourself familiar with the Australian tourism data set by making a sketch of how the variables (especially `State`, `Region` and `Purpose`) relate to each other.

Understanding the tourism data set - answer

See `https://otexts.com/fpp3/hts.html` by Hyndman and Athanasopoulos (2021)

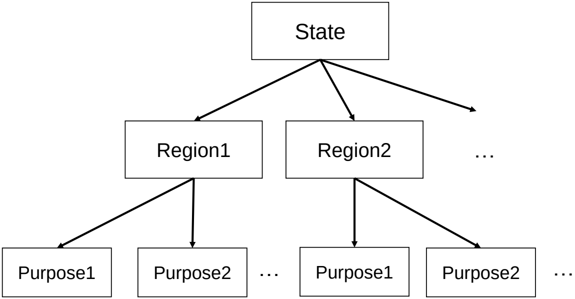


Figure 1: Sketch of the variables of the Australian tourism data set based on Hyndman and Athanasopoulos (2021).

Today’s problem - tourism data

Aim:

We are interested in considering the overnight trips of a specific region in a specific state in Australia. Optionally, we want to select a specific purpose for the overnight trips, e.g. we are interested in the time-dependent visits of business people in Adelaide in South Australia.

On the one hand, we want to get and save the filtered data and on the other hand, we want to visualize the resulting time series in a nice ggplot. Furthermore, we are also interested in the total number of overnight trips for each region in a state like South Australia.

Task - Making a plan

Your turn

Make a plan to solve the problem, i.e. make a plan for the implementation. What do you have to do? Which functions are needed? Make a sketch with some notes.

Making a plan - answer

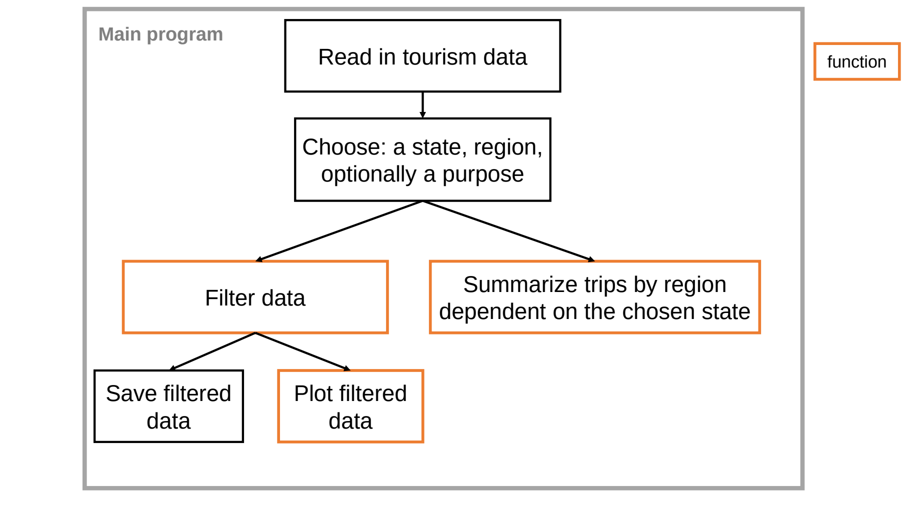

Main program
Read in tourism data
Choose: a state, region, optionally a purpose
Filter data
Summarize trips by region dependent on the chosen state
Save filtered data
Plot filtered data
function

Figure 2: A plan for the implementation.

Making a plan

Concretely,

- create a folder, e.g. named `aus_tourism`
- read in the data in a main program named `main_program_aus_tourism` and call the following functions:
    - `selectedData()`,
    - `plotTs()` and
    - `summarizedRegion()`

  which we have to write.

Task - Coding

Your turn

Write the corresponding program and its needed functions. For each R function create an own R file. Start with:

```r
library(ggplot2)
library(readxl)
library(dplyr)

getwd()
setwd("aus_tourism") # data set and functions are saved in folder "aus_tourism"
tourism <- read_excel("tourism.xlsx")
```

Coding - selectedData() und plotTs() - answer

```r
selectedData <- function(dataset, state, region, purpose = NULL){

  filtered_Data <- dataset %>%
    filter(State == state, Region == region)

  if (!is.null(purpose)) { # filter either for a specific purpose or not
    filtered_Data <- filtered_Data %>%
      filter(Purpose == purpose)
  }
  return(filtered_Data)
}
```

Make use of the R package `dplyr` (Wickham, François, et al. (2021)).

```r
plotTs <- function(filtered_data){
    ts_plot <- ggplot(data = filtered_data, mapping = aes(x = as.Date(Quarter),
                                                        y = Trips,
                                                        color = Purpose))+
      geom_line()+
      xlab("Time [Quarter]")+
      ylab("Overnight trips ('000)")
    ts_plot # return the created ggplot
}
```

Coding - summarizeRegion() - answer

Make use of the R package `dplyr` (Wickham, François, et al. (2021)).

```r
summarizeRegion <- function(dataset, state){
  sum_R <- dataset %>%
    filter(State == state) %>%
    group_by(Region) %>%
    summarize(Trips = sum(Trips))
  sum_R <- as.data.frame(sum_R)
  return(sum_R)
}
```

Coding - main program - answer

```r
library(ggplot2)
library(readxl)
library(dplyr)

getwd()
setwd("aus_tourism") # data set and functions are saved in folder "aus_tourism"
tourism <- read_excel("tourism.xlsx")

source("selectedData.R")
source("plotTs.R")
source("summarizeRegion.R")

state <- "South Australia"
region <- "Adelaide"
purpose <- "Holiday"
dataset <- tourism

sel_Data <- selectedData(dataset = dataset, state = state, region = region)
plot1 <- plotTs(filtered_data = sel_Data)
sR <- summarizeRegion(dataset = dataset, state = state)
# save the filtered data:
write.table(sel_Data, file = "Filtered_Aus_tourism_data.csv", sep = ",")

png("Tourism_data_South_Aus_Adelaide.png") # additionally, save the plot
plot1
dev.off()
```

Task - Documenting

Your turn

Comment and document your program and your functions as it has been shown in the lecture by Buchwitz (2021).

When documenting your functions, make use of the R package `roxygen2` (Wickham et al. 2021).

Documenting - answer

See the files:

- `main_program_aus_tourism.R`:
- `selectedData.R`,
- `plotTs.R` and
- `summarizedRegion.R`

Task - Testing

Your turn

Test your program and functions by filtering

1) South Australia as state, Adelaide as region and **do not** select a specific purpose. Plot the filtered time series.
2) South Australia as state, Adelaide as region and select Holiday as purpose. Plot the filtered time series.

Do some more tests to ensure that your functions work appropriately.

Testing 1) - answer - South Australia as state and Adelaide as region

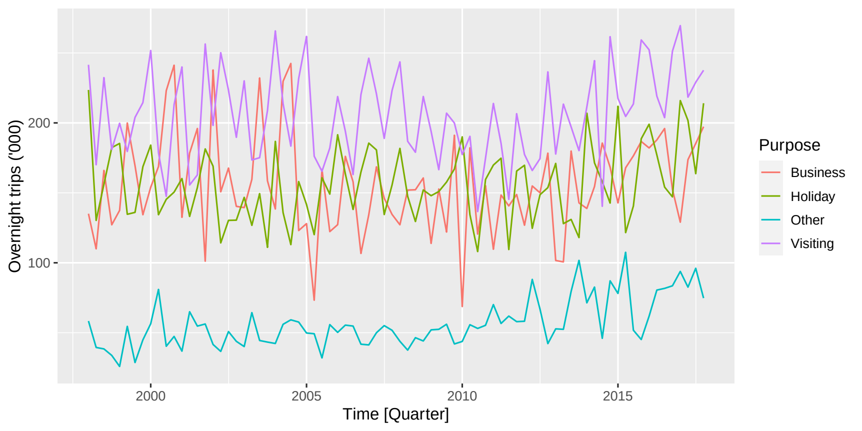
```r
state <- "South Australia"
region <- "Adelaide"
dataset <- tourism
```

```r
sel_Data <- selectedData(dataset = dataset, state = state, region = region)

plot1 <- plotTs(filtered_data = sel_Data)
sR <- summarizeRegion(dataset = dataset, state = state)

head(sel_Data) # filtered data
```

```
## # A tibble: 6 x 5
##   Quarter    Region   State           Purpose  Trips
##   <chr>      <chr>    <chr>           <chr>    <dbl>
## 1 1998-01-01 Adelaide South Australia Business  135.
## 2 1998-04-01 Adelaide South Australia Business  110.
## 3 1998-07-01 Adelaide South Australia Business  166.
## 4 1998-10-01 Adelaide South Australia Business  127.
## 5 1999-01-01 Adelaide South Australia Business  137.
## 6 1999-04-01 Adelaide South Australia Business  200.
```

Testing 1) - answer - South Australia as state and Adelaide as region

```
plot1 # time series plot
```



Testing 1) - answer - South Australia as state and Adelaide as region

```
sR # number of trips by region (dependent on the filtered state)
```

```
##                              Region Trips
## 1                          Adelaide 45906
## 2                    Adelaide Hills  2299
## 3                           Barossa  3850
## 4                      Clare Valley  3112
## 5                    Eyre Peninsula  7086
## 6                Fleurieu Peninsula 12544
## 7  Flinders Ranges and Outback 10327
## 8                   Kangaroo Island  1842
## 9                   Limestone Coast  9728
## 10                      Murraylands  5727
## 11                        Riverland  6369
## 12                  Yorke Peninsula  9361
```

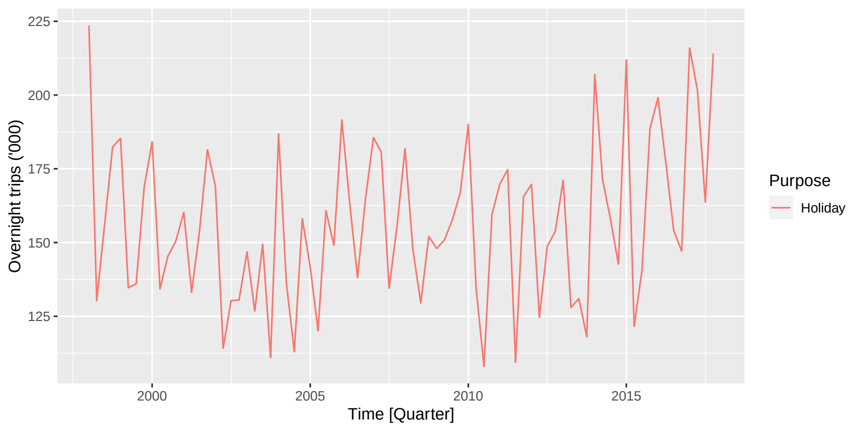
Testing 2) - answer - South Australia as state and Adelaide as region and Holiday as purpose

```r
purpose <- "Holiday"
sel_Data_purpose <- selectedData(dataset = dataset, state = state,
                                 region = region, purpose = purpose)

plot_purpose <- plotTs(filtered_data = sel_Data_purpose)
sR2 <- summarizeRegion(dataset = dataset, state = state)

head(sel_Data_purpose)
```

```
## # A tibble: 6 x 5
##   Quarter    Region   State           Purpose Trips
##   <chr>      <chr>    <chr>           <chr>   <dbl>
## 1 1998-01-01 Adelaide South Australia Holiday  224.
## 2 1998-04-01 Adelaide South Australia Holiday  130.
## 3 1998-07-01 Adelaide South Australia Holiday  156.
## 4 1998-10-01 Adelaide South Australia Holiday  182.
## 5 1999-01-01 Adelaide South Australia Holiday  185.
## 6 1999-04-01 Adelaide South Australia Holiday  135.
```

Testing 2) - answer - South Australia as state and Adelaide as region and Holiday as purpose

```
plot_purpose # time series plot
```



Testing 2) - answer - South Australia as state and Adelaide as region and Holiday as purpose

```
# no difference expected since the number of trips has been summarized by region
# (is independent of purpose):
unique(sR == sR2)
```

```
##      Region Trips
## [1,]   TRUE  TRUE
```

Task - Maintaining

Your turn

You have shown your program to a colleague. She/he recommends to compute also the percentage of trips of a selected Australian state. Create a function named `summarizePercTrips()`.

Maintaining - answer

```r
summarizePercTrips <- function(dataset, state){
   sum_T <- dataset %>%
   filter(State == state) %>%
    summarize(Trips = round(sum(Trips))) # sum of trips of the chosen state

   sum_T_total <- dataset %>% # total sum of trips (of all states)
     summarize(Trips = sum(Trips))

   rel_num_Trips <- sum_T / sum_T_total # sum of trips of chosen state / total sum
    return(rel_num_Trips)
}
```

What can be improved?

So far, we have implemented a program with some functions in order to extract information we are interested in from the Australian tourism data. To make our program more user-friendly, it is desired

- to have a user interface, where we can select a state, region and purpose
  - to show all possible regions of a selected state
  - to show all purposes after selecting a state and region
- that the time series plot will automatically be updated when the input changes
- …

→ All in all, we want to examine the tourism data set interactively.

A smart solution that allows these features is a **Shiny Web App** (Chang et al. (2021)) as we will see in a future exercise!

For home

Document the function `summarizePercTrips()`.

References

Buchwitz, B. 2021. *Computational Statistics*. `https://bchwtz.github.io/bchwtz-cswr/`.

Hyndman, R. J., and G. Athanasopoulos. 2021. *Forecasting: Principles and Practice*. 3rd ed. Springer-Lehrbuch. Melbourne, Australia: OTexts.

Wickham, Hadley, and Jennifer Bryan. 2019. *Readxl: Read Excel Files*. `https://CRAN.R-project.org/package=readxl`.

Wickham, Hadley, Peter Danenberg, Gábor Csárdi, and Manuel Eugster. 2021. *Roxygen2: In-Line Documentation for r*. `https://CRAN.R-project.org/package=roxygen2`.