Introduction to working with R on cloud
Elementary exploring *Titanic* data and searching *Migrant stories*

Barbora Hladká – Martin Holub

Course Tutorial
NPFL 142 – Artificial Intelligence for Humanities
https://ufal.mff.cuni.cz/courses/npfl142
Charles University

Info on working with R system on cloud..... 2
Task 1: Titanic dataset basic analysis..... 3
Exercise 1.1 – Getting a data set..... 3
Exercise 1.2 – Loading a data set into a table in R and showing its structure..... 3
Exercise 1.3 – Exploration of passengers' gender..... 4
Exercise 1.4 – Exploration of survived passengers based on gender..... 5
Practice Exercises..... 7
Exercise 1.5 – Elementary analysis of numerical attributes..... 7
Practice Exercises..... 9
Task 2: Migrants dataset basic analysis..... 10
Dataset description..... 10
Exercise 2.1 – Getting a data set..... 10
Exercise 2.2 – Directions of migration..... 11
Practice Exercises..... 12
Task 3: Searching in Migrants' stories..... 13
Exercise 3.1 – Migrants' vocabulary..... 13
Practice Exercises..... 15
Appendix: Table with special characters used in regular expressions..... 16

Info on working with R system on cloud

We will be computing on the Artificial Intelligence Cluster (AIC) administered by ÚFAL MFF UK. Namely, we will be using RStudio provided by the JupyterLab Notebook installed at AIC. You have already tried out to log in https://aic.ufal.mff.cuni.cz/jlab when you did the homework assignment #0.

An identical setup on cloud is an advantage of a shared computing environment. We will not have to troubleshoot problems that would occur during system installation on local devices. All required libraries will be pre-installed, allowing us to focus directly on the code details.

Naming folders and scripts in students' `Home` folders

The tutorials are organized into Tasks. Each Task is a series of Exercises using a specific dataset. From the beginning of the course, all datasets are placed in the `DATA` folder in students' `Home` folders.[1] For each Task, students should:

1. Create a new folder in their `Home` and name it by the given label for the dataset associated with the Task.
2. Copy a specific data file from `DATA` into the new folder.
3. Then, in this folder, students will create their R script to address all Exercises of the given Task.

Students can name their scripts according to their own convention. However, they should always place the given Task number in the name of the respective script. For example, the dataset *Titanic* having the label *titanic* is associated with Task 1 (see below). So students will create a new folder `titanic` and copy the data file `DATA/titanic.csv` to it. Then the script `titanic/titanic.T1.R` will contain their own code for the Task 1 (see below).

Students can download the contents of their folders in RStudio to their local drives using Output pane > Files > More > Export.

[1] Note: On Unix-like systems, folders are called "directories".

© Barbora Hladká & Martin Holub, 2025

Task 1: *Titanic* dataset basic analysis

Data description – Titanic - Machine Learning from Disaster. We will be working with the `train.csv` file downloaded from the Kaggle web site and renamed as `titanic.csv`.

Exercise 1.1 – Getting a data set

Run RStudio and do the following steps:

- Create a new folder `titanic` in your `Home` folder
  (Output pane > Files > New folder)
- Copy `DATA/titanic.csv` to folder `titanic`
- Set folder `titanic` as your working folder
  (Output pane > Files > More > Set as Working Directory)

Exercise 1.2 – Loading a data set into a table in R and showing its structure

In RStudio, create a blank R script (Output pane > Files > New Blank File > R Script) and enter the new file name `titanic.T1.R`. Then the script is open in the Source pane (upper-left) and you can add the commands listed below to the script.

We suppose using `tidyverse` package.

```
library(tidyverse)
```

Load the *Titanic* dataset into your R environment and look at its structure.

```
dataset <- read_csv("titanic.csv")
print(dataset)
# A tibble: 891 × 12
   PassengerId Survived Pclass Name   Sex     Age SibSp Parch Ticket  Fare Cabin
         <dbl>    <dbl>  <dbl> <chr>  <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
 1           1        0      3 Braun… male     22     1     0 A/5 2…  7.25 NA
 2           2        1      1 Cumin… fema…    38     1     0 PC 17… 71.3  C85
 3           3        1      3 Heikk… fema…    26     0     0 STON/…  7.92 NA
 4           4        1      1 Futre… fema…    35     1     0 113803 53.1  C123
 5           5        0      3 Allen… male     35     0     0 373450  8.05 NA
 . . .
```

Check the number of examples, i.e. passengers onboard.

```
nrow(dataset)              # number of rows in the tibble
[1] 891
```

Check the number of attributes.

```
ncol(dataset)              # number of columns in the tibble
[1] 12
```

Check the attribute names.

```
colnames(dataset)
 [1] "PassengerId" "Survived"    "Pclass"      "Name"        "Sex"
 [6] "Age"         "SibSp"       "Parch"       "Ticket"      "Fare"
[11] "Cabin"       "Embarked"
```

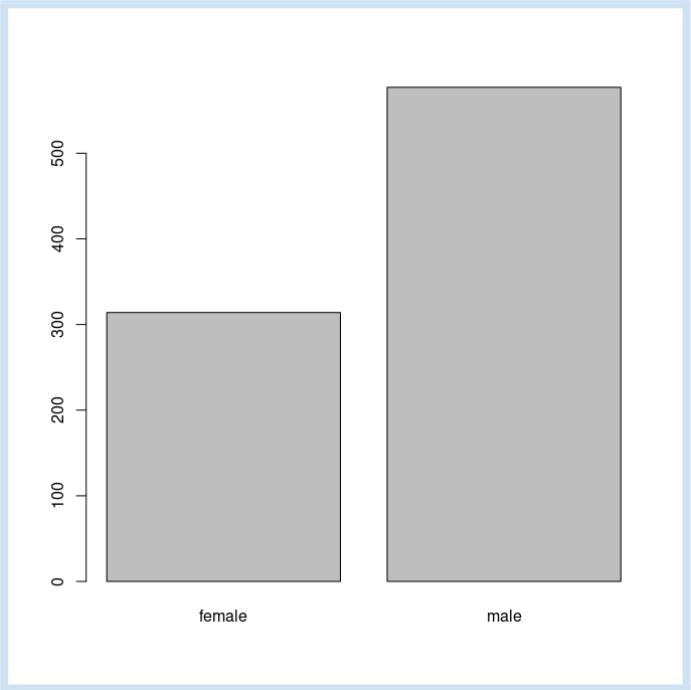
Exercise 1.3 – Exploration of passengers' gender

Explore the distribution of values of a given attribute.

```
table(dataset$Sex)              # frequency table for Sex attribute
female   male
   314    577
```

Visualize the distribution.

```
barplot(table(dataset$Sex))
```



© Barbora Hladká & Martin Holub, 2025

Calculate the proportion of men and women.

```
N <- nrow(dataset)
table(dataset$Sex)/N
  female     male
0.352413 0.647587
```

Proportion in rounded percentage.

```
round(table(dataset$Sex)/N * 100, 1)
female   male
  35.2   64.8
```

Exercise 1.4 – Exploration of survived passengers based on gender

Research Question: What did survival depend on during the sinking of the Titanic?

Explore the distribution of values of the `Survived` and Sex attributes.

```
survived_sex <- table(dataset$Survived, dataset$Sex)
survived_sex                              # contingency table for Survived and Sex attributes
      female male
  0     81  468
  1    233  109
```

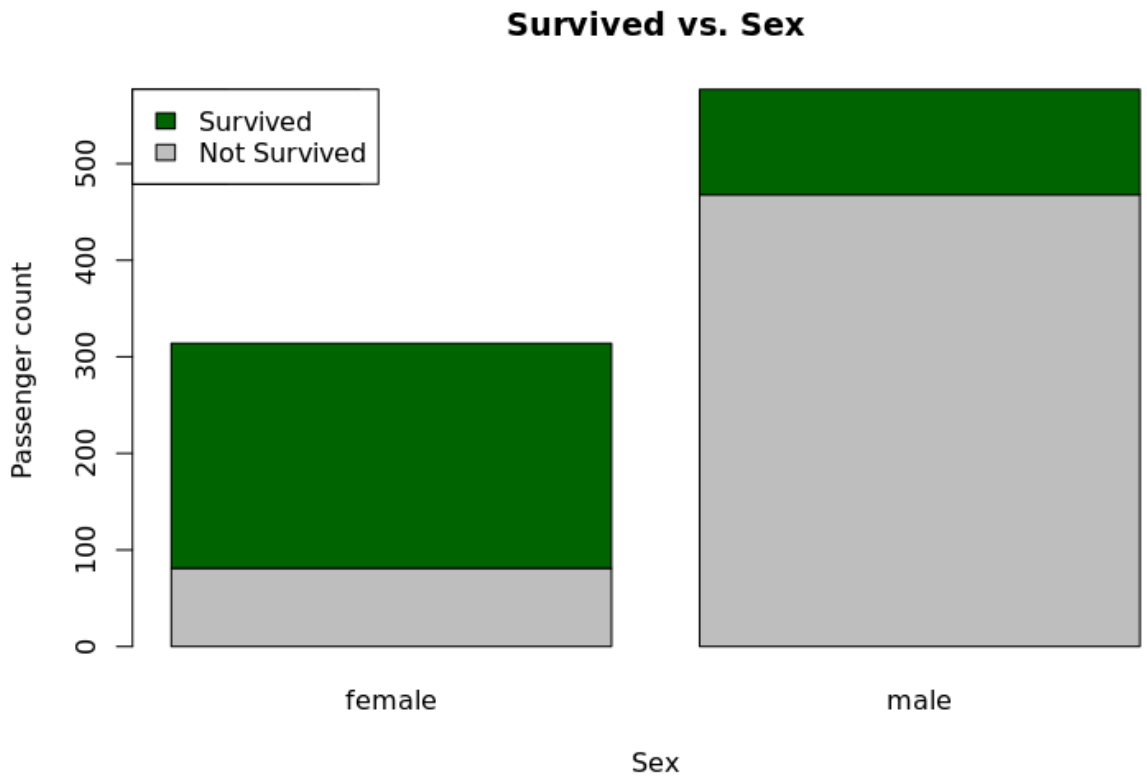
Visualize the contingency table.

```
barplot(survived_sex,
       main = "Survived vs. Sex",
       xlab = "Sex",
       ylab = "Passenger count",
       col = c("grey","darkgreen"),
       legend.text = TRUE,
       args.legend = list(x = "topleft")
)
```

Add an explicit legend.

```
legend(
       legend = c("Survived", "Not Survived"),
       fill   = c("darkgreen","gray"),
       "topleft"
)
```

© Barbora Hladká & Martin Holub, 2025

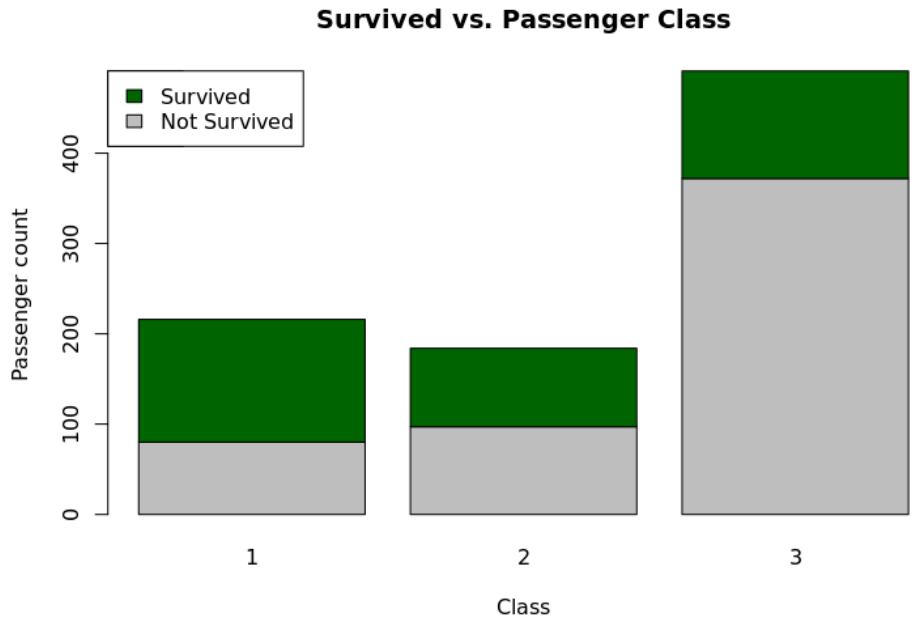


© Barbora Hladká & Martin Holub, 2025

Practice Exercises

Now do the same analysis with the Passenger Class. First, make the corresponding contingency table, and then visualize the distribution. The result should be as follows:

```
survived_class                          # contingency table
      1   2   3
  0  80  97 372
  1 136  87 119
```



What is the proportion of survived people in Class 1, 2, and 3, in percentage?

– Calculate three numbers corresponding to the visualization.

Exercise 1.5 – Elementary analysis of numerical attributes

Attribute Fare is an example of numerical, continuous variable. The numbers are not only integers.

Look at sample values.

```
str(dataset$Fare)                                   # get the structure of the given object
 num [1:891] 7.25 71.28 7.92 53.1 8.05 ...          # sample values
```

Calculate elementary statistics.

```
max(dataset$Fare)                                   # get the maximum
```

```
[1] 512.3292

min(dataset$Fare)                                    # get the minimum
[1] 0

mean(dataset$Fare)                                   # get the average value
[1] 32.20421

median(dataset$Fare)                                 # get the median
[1] 14.4542
```

How do `Fare` values depend on `Pclass`?

```
# to select specific rows from a data frame use subset() function

Fare_C1 <- subset(dataset, Pclass == 1)$Fare         # take Fare values only for Pclass = 1
summary(Fare_C1)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   30.92   60.29   84.15   93.50  512.33

Fare_C2 <- subset(dataset, Pclass == 2)$Fare         # take Fare values only for Pclass = 2
summary(Fare_C2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   13.00   14.25   20.66   26.00   73.50

Fare_C3 <- subset(dataset, Pclass == 3)$Fare         # take Fare values only for Pclass = 2
summary(Fare_C3)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    7.75    8.05   13.68   15.50   69.55


# numbers of different Pclass values correspond to the selected fares:
table(dataset$Pclass)
  1   2   3
216 184 491

length(Fare_C1)
[1] 216

length(Fare_C2)
[1] 184

length(Fare_C3)
[1] 491
```

Practice Exercises

Explore the distribution of passengers who took the first travel class.

5.a What is their average age?

5.b Is there different proportion of women in comparison with the second and third travel classes?

5.c Do men in the first travel class have more expensive fares in comparison with women?

Did passengers with higher fares have greater chance to survive?

5.d Compare the survival rate of passengers with fares greater than the median with the survival rate of all passengers.

5.e Explore the survival rate of men with fares greater, or less than the median (among men).

5.f Do the same for women.

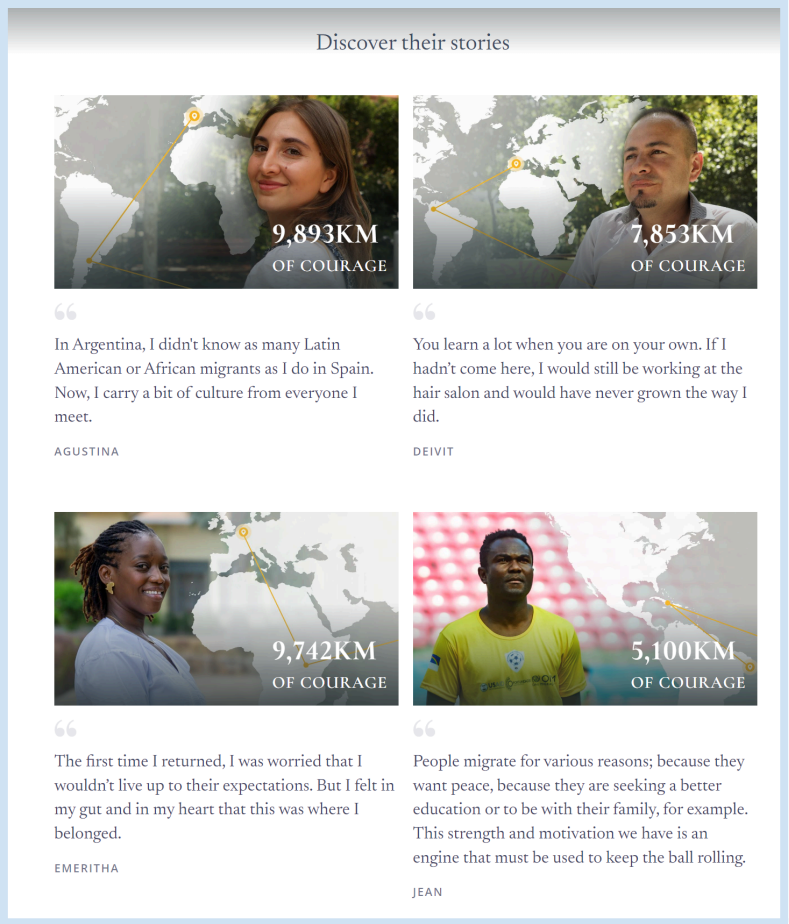
5.g Do the same only for passengers in the third travel class.

© Barbora Hladká & Martin Holub, 2025

Task 2: *Migrants* dataset basic analysis

Dataset description

"I am a migrant" is a campaign launched by the International Organization for Migration (IOM, https://www.iamamigrant.org) to promote diversity and inclusion, and to combat xenophobia and divisive narratives around migration. The platform features first-hand accounts from people on the move. The migrant stories collected by the IOM are written in English, and the exact way they were collected is unknown to us. We assume that the migrants told their stories in interviews in their native languages, and then the stories were transcribed and translated into English. Some of the stories are available as a dataset published in the LINDAT/CLARIAH-CZ repository and they are searchable in TEITOK.



Exercise 2.1 – Getting a data set

In RStudio

- Create a new folder **`migrants`** in **`Home`** folder (Output pane > Files > New folder)
- Copy **`DATA/migrants.tsv`** to **`migrants`**
- Set **`migrants`** as working folder (Output pane > Files > More > Set as Working Directory)

© Barbora Hladká & Martin Holub, 2025

Exercise 2.2 – Directions of migration

In RStudio, create a blank R script (Output pane > Files > New Blank File > R Script) and enter the new file name **`migrants.T2.R`**. Then the script is open in the Source pane (upper-left) and you can add the commands listed below to the script.

We suppose using tidyverse package.

```
library(tidyverse)
```

Load **`migrants.tsv`** dataset into R and explore its structure. See the description of the attributes.

```
dataset <- read_tsv("migrants.tsv")
print(dataset)
```

```
# A tibble: 1,017 × 13
   id_story name                country_or           country_de  conti_or conti_de distance country_or_gdp country_de_gdp gdp_change home_change gender story
      <dbl> <chr>               <chr>                <chr>       <chr>    <chr>    <chr>    <chr>         <chr>          <chr>      <chr>       <chr>  <chr>
 1        1 Abann               South Sudan          New Zealand A        O        far      1 421         43 972         H          im          male   In 2005, I ar…
 2        2 Abass Senghore      Gambia               Libya       A        A        close    757           4 243          H          im          male   I left the co…
 3        3 Abba                Togo                 Niger       LA       A        far      863           568            E          im          male   You will rare…
 4        4 Abbas and his family Iraq                Greece      M        E        far      4 146         18 117         E          im          male   Midway, the b…
 5        5 Abdalrahman         Rwanda               Burundi     A        A        close    798           286            E          im          male   My name is Ab…
 6        6 Abdalsalams family  Syrian Arab Republic Greece      M        E        far      890           18 117         H          im          n      When the war …
 7        7 Abdalwahab          Sudan                Libya       A        A        close    1 415         4 243          E          im          male   In 1986, I go…
 8        8 Abdel               Egypt                Italy       A        E        far      3 609         31 238         H          im          male   Abdel came to…
 9        9 Abdel and his family Syrian Arab Republic Greece      M        E        far      890           18 117         H          im          male   It was a very…
10       10 Abdelhak            Morocco              Morocco     A        A        close    3 108         3 108          E          hc          male   I  went to Be…
# ℹ 1,007 more rows
```

Research Question: What are the directions of migration? Focus on destination countries.
Work with the values of the **`country_de`** attribute. First create a contingency table and then sort the countries by the number of migrants who arrived in these countries in decreasing order.

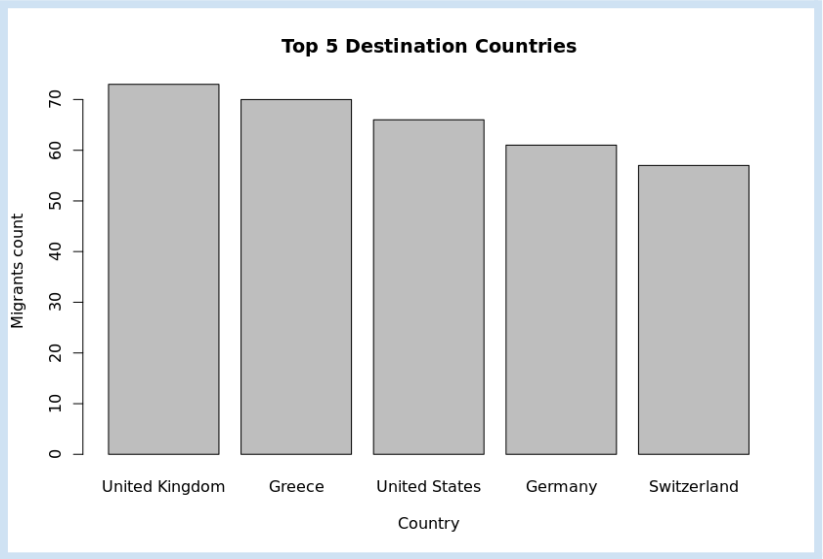
```
destination <- table(dataset$country_de)                          # contingency table
destination[1:5]
Afghanistan     Albania     Algeria   Argentina     Armenia
          8           8           3           5           2
sorted_destination <- sort(destination, decreasing=T)             # sort
```

Check the TOP 3 destination countries.

```
sorted_destination[1:3]
United Kingdom         Greece  United States
            73             70             66
```

Visualize the TOP 5 destination countries.

```
barplot(sorted_destination[1:5],                                  # draw a barplot
        ylab = "Migrants count",
        xlab = "Destination country",
        main = "Top 5 Destination Countries"
       )
```



Practice Exercises

2.a – How many different countries did the migrants come to?

```
101
```

2.b – How many different countries did the migrants leave?

```
134
```

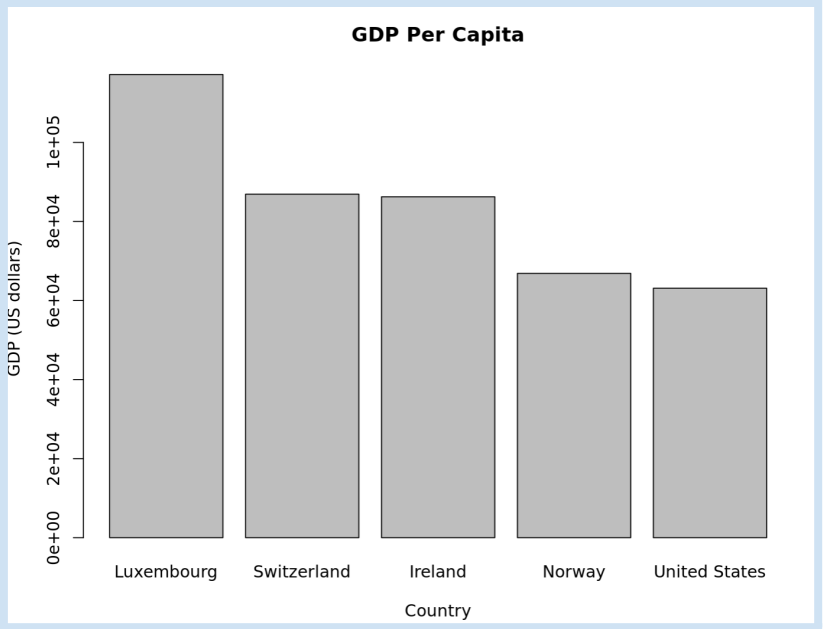
2.c – How many women came to Greece?

```
3
```

2.d – What countries did the people go to from Egypt?

```
Egypt          Estonia        France         Italy          Romania        United Kingdom       United States
3              1              1              3              1              2                    1
```

2.e – Display the five destination countries with the highest Gross Domestic Product.



Task 3: Searching in *Migrants'* stories

Exercise 3.1 – Migrants' vocabulary

Research Question: What are the words that the migrants are using in their stories?

We suppose using `stringr` package.

```
library(stringr)
```

The stories of the migrants can be found in the `story` attribute.

```
names(dataset)
[1] "id_story"       "name"           "country_or"     "country_de"     "conti_or"
[6] "conti_de"       "distance"       "country_or_gdp" "country_de_gdp" "gdp_change"
[11] "home_change"    "gender"          "story"
```

For example, print the first story

```
dataset$story[1]
```

```
[1] "In 2005, I arrived in Auckland, New Zealand. I came with my two daughters and my lovely wife, Mary. We stayed in a resettlement centre for six weeks and received an o
rientation programme organised by the New Zealand government. When I moved into the community, I realised that things were not going to be as easy as I thought. I was face
d with the reality of resettlement challenges. It was very hard to understand and navigate basic resettlement issues, such as neighbourhood connections, networking and acc
ess to available services. Fortunately, there were government and NGOs support services such as NZ Red Cross volunteer programmes to deal with some of the these challenge
s. The place that made me love New Zealand was Mangere Bridge Mountain. I used to go to this mountaintop every day, to enjoy the views of Auckland. I loved it because it g
ave me a sense of belonging and I knew I wanted to stay in New Zealand. I had previously lived for some time in Damascus, Syria, and I was Secretary General for the South
Sudanese Community and Chairman of the Shilluk community. I came to New Zealand with the same spirit. I actively engaged the South Sudanese community members to create act
ivities that could reduce isolation and loneliness in the community and encouraged others to get to know one another. The first administrative work I did in support of com
munity members was to organise a World Refugee Day celebration on June 20, 2006. It was a very successful event, which saw the establishment of the Auckland South Sudanese
Soccer Team. It was because of such events that South Sudanese community members saw the need to continue engaging with one another. I was chosen to lead the community, bu
t I chose to serve in the capacity of deputy chairman on the grounds that I was new to New Zealand. As a deputy chairman I established a lot of networking with the Aucklan
d Resettlement Sectors. Our community became a member of the Auckland Resettled Community Coalition (ARCC) in 2006. We received funding and I ran a parental program for fi
ve years, as well as introducing a healthy eating and healthy living campaign and youth programmes. In 2011 after graduating with a graduate diploma in non-for-profit mana
gement, I worked as a youth worker for nearly four years with the Resettlement Youth Action Network (RYAN). My roles there were to help young people find jobs and to help
them with appropriate education plans (Pathway). Then in 2013, I was elected Chairperson of ARCC. Currently, I am the General Manager, of ARCC after four years of positive
development from a volunteer organisation to one that can pay staff. I am now considering studying again for a Masters degree in community development. I want to keep my f
athers dream alive; he wanted me to get a quality education before he departed this earth. I want readers to learn something from my story. I want to advise those who are
struggling with jobs and life in general to pause and check the missteps. Help yourself first in order to find someone else who can help you. At this point in my journey,
I am proud to have achieved successful outcomes throughout my struggle for a better life. I feel that I am indeed walking in the shoes of my father and keeping his dream a
live."
```

Before we dive into analyzing the stories, we need to understand the difference between a string and a word: a *string* is a sequence of characters of any length; it can include letters, numbers, spaces, and punctuation. A *word*, on the other hand, is simply a smaller part within a string. Words are usually separated by spaces or punctuation.

Find out the number of times that *community* appears in the first story. The function **`str_count(string, pattern = "")`** counts how many times a specific **`pattern`** appears in a **`string`**.

```
str_count(dataset$story[1], "community")
[1] 9
```

Find out the number of times that the string *I feel* appears in the first story.

```
str_count(dataset$story[1], "I feel")
[1] 1
```

Let's focus on temporal expressions, specifically the years mentioned in the first story.

```
str_count(dataset$story[1], "2006")
[1] 2                                    # two occurrences of 2006 in the first story
str_count(dataset$story[1], "2004")
[1] 0                                    # 2004 not in the first story
```

No doubt, searching for years one by one is not effective. Therefore, we will search using regular expressions. *Regular expressions* (regex) are useful because they allow searching for and matching patterns within text rather than searching for exact strings. By combining a selection of simple patterns, we can capture quite complicated strings – see Appendix below.

If we are searching for all the years that are mentioned in a story, let's use square brackets. They allow us to match a character specified inside a set. For example, the regular expression [0123456789] will match a character from the set containing 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. We can use a hyphen character to define a range of characters. Thus [0-9] is the same as [0123456789].

```
str_extract_all(dataset$story[1], "[0-9][0-9][0-9][0-9]") # extract any 4-digit string
[1] "2005" "2006" "2006" "2011" "2013"
```

If we are interested in years 2010+, we can set a regex pattern like `20[1-9][0-9]`

```
str_extract_all(dataset$story[1], "20[1-9][0-9]")          # > 2010
[1] "2011" "2013
```

Practice Exercises

2.f – Display the distribution of *community* frequency in all the stories

```
 0   1   2   3   4   5   6   7   9

839 119  33  13   7   1   3   1   1
```

2.g – Find out what four-digit different numbers greater or equal to 2000 occur in all the stories

```
"2000" "2001" "2002" "2003" "2004" "2005" "2006" "2007" "2008" "2009" "2010" "2011"
"2012" "2013" "2014" "2015" "2016" "2017" "2018" "2019" "2020" "2030"
```

© Barbora Hladká & Martin Holub, 2025

Appendix: Table with special characters used in *regular expressions*

There are 14 meta characters that have a special meaning within a regular expression.

Symbol	Character	Description
square bracket	[]	Defines a character class (set of characters to match) Ex. `[abc]` matches "a", "b", or "c"
backslash	\	Escapes special characters or signals special sequences Ex. `\.` matches a literal dot (.), not any character
caret	^	Matches the start of a string Ex. `^Hello` matches "Hello world" but not "world Hello"
dollar sign	$	Matches the end of a string Ex. `world$` matches "Hello world" but not "world Hello"
period/dot	.	Matches any single character except a newline Ex. `c.t` matches "cat", "cut", "cot", etc.
vertical bar	|	Acts as a logical OR between multiple patterns Ex. `cat|dog` matches either "cat" or "dog".
question mark	?	Matches zero or one of the preceding character or group Ex. `colou?r` matches both "color" and "colour"
asterisk	*	Matches zero or more of the preceding character or group Ex. `ca*t` matches "ct", "cat", "caat", "caaat", etc.
plus sign	+	Matches one or more of the preceding character or group Ex. `ca+t` matches "cat", "caat", "caaat", etc., but not "ct"
curly bracket	{}	Specifies a range for the number of occurrences Ex. `a{2,4}` matches "aa", "aaa", or "aaaa" but not "a" or "aaaaa"
opening parenthesis	()	Groups expressions and captures them for later use Ex. `(ab)+` matches "ab", "abab", "ababab", etc

© Barbora Hladká & Martin Holub, 2025