Data Science Academy / Week 5 Homework

Titus Teodorescu

12/05/2021

Part 2

Explore the PSTRE_syntheticData.csv to

- Create two new variables: action sequence variable and time interval sequence for each participant
- Extract the time for the first action for each participant
- Represent each action sequence by n-grams (n=2)

```
## load libraries
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(stringr)
library(purrr)
library(tidyr)

## reading data to a data frame, called df
## despite the ".csv" name (which suggests the data is separated by commas),
the separator in the source .csv file is a tab
df <- read.csv(file="C:/Users/tteodorescu/OneDrive - Educational Testing
Service/ETS/Data Science/Data Science Academy/Week5/Homework/data for
week5/PSTRE_syntheticData.csv",header = TRUE,sep="\t")

print("Original data")

## [1] "Original data"

head(df, n=15L)
```

```
##    TestTakerID Timestamp    Coding
## 1      1001042         0     START
## 2      1001042     44985     SS_Se
## 3      1001042     55852  SS_Se_OK
## 4      1001042     55950 SS_Type_2
## 5      1001042     66546         E
## 6      1001042     90189        SS
## 7      1001042     96096         E
## 8      1001042    101813      Next
## 9      1001042    105379   Next_OK
## 10     1001042    105382       END
## 11     1001079         0     START
## 12     1001079     21234     SS_So
## 13     1001079     27546  SS_So_1B
## 14     1001079     31906  SS_So_OK
## 15     1001079     77760        SS
```

```
df_with_new_variables <- df %>%
  mutate(Next_Timestamp=lag(Timestamp, default = 0)) %>%
## shift the Timestamp column down by 1 row;

## store it to a new column
  mutate(Duration = Timestamp - Next_Timestamp) %>%
## compute the duration of each action
  mutate(Duration = if_else(Duration < 0, 0, Duration)) %>%
## replace negative Duration values with 0
  mutate(Duration = lead(Duration)) %>%
## shift Duration column up by 1 row
  filter(Coding != "END") %>%
## remove the END actions
  group_by(TestTakerID) %>%
## group by TestTakerID
  summarise(action_sequence = paste(Coding, collapse = ","),
## concatenate Coding to a string with a comma separator
    duration_sequence = paste(str_replace(Duration," ",""), collapse = "," ))
## concatenate Duration to a string with a comma separator


print("Revised data with new variables: action_sequence and
duration_sequence")
```

```
## [1] "Revised data with new variables: action_sequence and
duration_sequence"
```

```
head(df_with_new_variables, n=15L)
```

```
## # A tibble: 15 x 3
##    TestTakerID action_sequence                  duration_sequence
##          <int> <chr>                            <chr>
##  1     1001042 START,SS_Se,SS_Se_OK,SS_Type_2,~
44985,10867,98,10596,23643,5907~
```

```
##  2     1001079 START,SS_So,SS_So_1B,SS_So_OK,S~
21234,6312,4360,45854,29414,775~
##  3     1001103 START,SS_Se,SS_Se_OK,SS_Type_2,~
39002,17605,123,31895,4232,3162~
##  4     1001112 START,E,SS,E,Next,Next_OK
54895,7887,7732,2915,1693,1
##  5     1002087 START,SS_Se_OK,SS_Type_200,SS_S~
158089,133,47415,5200,23422,233~
##  6     1002110 START,E,SS,E,E_S,Next,Next_OK
93106,19812,3406,8478,1878,2195~
##  7     1003077 START,SS_So,SS_So_1B,SS_So_OK,E~
53204,9269,5449,21135,35913,293~
##  8     1003110 START,Next,Next_C,Next,Next_C,E~
18045,9435,28212,3413,11492,631~
##  9     1003147 START,Next,Next_OK               110606,4159,4
## 10     1003165 START,SS_So_1B,SS_So_OK,SS_So_O~
89372,11317,21308,31146,6237,11~
## 11     1003192 START,SS_So,SS_So_1B,SS_So_2A,S~
79867,16468,8983,12684,53745,67~
## 12     1005075 START,SS_So,SS_So_1B,SS_So_OK,E~
61604,7828,4354,39734,14213,333~
## 13     1006135 START,E,E_S,Next,Next_OK         132359,11058,2898,2285,2
## 14     1006144 START,E,Next,Next_OK             218033,34781,6298,2
## 15     1006180 START,Next,Next_OK               12123,1841,10

str(df_with_new_variables)

## tibble [1,079 x 3] (S3: tbl_df/tbl/data.frame)
##  $ TestTakerID      : int [1:1079] 1001042 1001079 1001103 1001112 1002087
1002110 1003077 1003110 1003147 1003165 ...
##  $ action_sequence  : chr [1:1079]
"START,SS_Se,SS_Se_OK,SS_Type_2,E,SS,E,Next,Next_OK"
"START,SS_So,SS_So_1B,SS_So_OK,SS,E,SS,E,SS,E,SS,E,E,Next,Next_OK"
"START,SS_Se,SS_Se_OK,SS_Type_2,E,SS,E,SS,E,Next,Next_OK"
"START,E,SS,E,Next,Next_OK" ...
##  $ duration_sequence: chr [1:1079]
"44985,10867,98,10596,23643,5907,5717,3566,3"
"21234,6312,4360,45854,29414,7759,5157,4319,1650,41297,6090,5651,13227,1791,2
" "39002,17605,123,31895,4232,31629,18241,7863,19695,1672,1"
"54895,7887,7732,2915,1693,1" ...

df_duration_of_first_action <- df_with_new_variables %>%
  mutate(duration_of_first_action = str_split(duration_sequence, ",")) %>%
## split by commas
  mutate(duration_of_first_action = map(duration_of_first_action, 1)) %>%
## get the first element
  mutate(duration_of_first_action = unlist(duration_of_first_action)) %>%
  select(TestTakerID, duration_of_first_action)
## remove all other columns
```

```
print("Display duration of the first action")
```

```
## [1] "Display duration of the first action"
```

```
head(df_duration_of_first_action, n=15L)
```

```
## # A tibble: 15 x 2
##    TestTakerID duration_of_first_action
##          <int> <chr>
##  1     1001042 44985
##  2     1001079 21234
##  3     1001103 39002
##  4     1001112 54895
##  5     1002087 158089
##  6     1002110 93106
##  7     1003077 53204
##  8     1003110 18045
##  9     1003147 110606
## 10     1003165 89372
## 11     1003192 79867
## 12     1005075 61604
## 13     1006135 132359
## 14     1006144 218033
## 15     1006180 12123
```

```
df_action_sequence_of_2_grams <- df %>%
  mutate(Next_Coding=lead(Coding)) %>%
## shift the Coding column up by 1 row;

## store it to a new column
  filter(Coding != "END") %>%
## remove the END actions
  unite("two_gram", Coding:Next_Coding, sep=" ") %>%
## combine Coding and Next_Coding with a space separator
  group_by(TestTakerID) %>%
## group by TestTakerID
  summarise(action_sequence = paste(two_gram, collapse = ","))
## concatenate two_gram to a string with a comma separator
```

```
print("Display action sequence of 2-grams. Note that the two entries in the
same 2-gram are separated by a space, while any two adjacent 2-grams are
separated by a comma.")
```

```
## [1] "Display action sequence of 2-grams. Note that the two entries in the
same 2-gram are separated by a space, while any two adjacent 2-grams are
separated by a comma."
```

```
head(df_action_sequence_of_2_grams, n=15L)
```

```
## # A tibble: 15 x 2
##    TestTakerID action_sequence
##          <int> <chr>
##  1     1001042 START SS_Se,SS_Se SS_Se_OK,SS_Se_OK SS_Type_2,SS_Type_2 E,E
SS,S~
##  2     1001079 START SS_So,SS_So SS_So_1B,SS_So_1B SS_So_OK,SS_So_OK SS,SS
E,E ~
##  3     1001103 START SS_Se,SS_Se SS_Se_OK,SS_Se_OK SS_Type_2,SS_Type_2 E,E
SS,S~
##  4     1001112 START E,E SS,SS E,E Next,Next Next_OK,Next_OK END
##  5     1002087 START SS_Se_OK,SS_Se_OK SS_Type_200,SS_Type_200
SS_So_1B,SS_So_1~
##  6     1002110 START E,E SS,SS E,E E_S,E_S Next,Next Next_OK,Next_OK END
##  7     1003077 START SS_So,SS_So SS_So_1B,SS_So_1B SS_So_OK,SS_So_OK E,E
Next,N~
##  8     1003110 START Next,Next Next_C,Next_C Next,Next Next_C,Next_C E,E
Next,N~
##  9     1003147 START Next,Next Next_OK,Next_OK END
## 10     1003165 START SS_So_1B,SS_So_1B SS_So_OK,SS_So_OK SS_So_OK,SS_So_OK
SS_S~
## 11     1003192 START SS_So,SS_So SS_So_1B,SS_So_1B SS_So_2A,SS_So_2A
SS_So_OK,S~
## 12     1005075 START SS_So,SS_So SS_So_1B,SS_So_1B SS_So_OK,SS_So_OK E,E
E_S,E_~
## 13     1006135 START E,E E_S,E_S Next,Next Next_OK,Next_OK END
## 14     1006144 START E,E Next,Next Next_OK,Next_OK END
## 15     1006180 START Next,Next Next_OK,Next_OK END
```