Recap 1.1: Process Mining Event Log Analysis

```
library(dplyr)
library(stringr)
library(tidyverse)
```

Preparing the data

```
df <- read.table(text = "
Application_Number Event_Time Activity_Name Resource Loan_Goal Requested_Amount
Application_174 '1/3/2016 16:49' 'A_Create Application' User_1 'Existing loan takeover' 11000
Application_174 '1/3/2016 16:49' 'A_Submitted' User_1 'Existing loan takeover' 11000
Application_174 '1/3/2016 16:50' 'A_Concept' User_1 'Existing loan takeover' 11000
Application_524 '1/3/2016 17:13' 'A_Create Application' User_1 'Caravan / Camper' 15000
Application_524 '1/3/2016 17:13' 'A_Submitted' User_1 'Caravan / Camper' 15000
Application_524 '1/3/2016 17:14' 'A_Concept' User_1 'Caravan / Camper' 15000
Application_881 '1/3/2016 17:48' 'A_Create Application' User_1 'Existing loan takeover' 15000
Application_881 '1/3/2016 17:48' 'A_Submitted' User_1 'Existing loan takeover' 15000
Application_881 '1/3/2016 17:49' 'A_Concept' User_1 'Existing loan takeover' 15000
Application_634 '1/3/2016 18:44' 'A_Create Application' User_1 'Home improvement' 10000
Application_634 '1/3/2016 18:44' 'A_Submitted' User_1 'Home improvement' 10000
Application_634 '1/3/2016 18:45' 'A_Concept' User_1 'Home improvement' 10000
Application_797 '1/3/2016 18:50' 'A_Create Application' User_1 'Car' 13500
Application_797 '1/3/2016 18:50' 'A_Submitted' User_1 'Car' 13500
Application_797 '1/3/2016 18:51' 'A_Concept' User_1 'Car' 13500
Application_881 '1/4/2016 13:19' 'A_Accepted' User_13 'Existing loan takeover' 15000
Application_881 '1/4/2016 13:25' 'A_Complete' User_13 'Existing loan takeover' 15000
Application_174 '1/4/2016 13:41' 'A_Accepted' User_7 'Existing loan takeover' 11000
Application_174 '1/4/2016 13:50' 'A_Complete' User_7 'Existing loan takeover' 11000
Application_524 '1/4/2016 18:58' 'A_Accepted' User_8 'Caravan / Camper' 15000
Application_797 '1/4/2016 19:14' 'A_Accepted' User_5 'Car' 13500
Application_524 '1/4/2016 19:14' 'A_Complete' User_8 'Caravan / Camper' 15000
Application_797 '1/4/2016 19:17' 'A_Complete' User_5 'Car' 13500
Application_634 '1/5/2016 15:51' 'A_Accepted' User_7 'Home improvement' 10000
Application_634 '1/5/2016 15:59' 'A_Complete' User_7 'Home improvement' 10000
Application_524 '1/14/2016 9:09' 'A_Validating' User_119 'Caravan / Camper' 15000
Application_524 '1/14/2016 14:15' 'A_Incomplete' User_30 'Caravan / Camper' 15000
Application_524 '1/14/2016 14:41' 'A_Pending' User_113 'Caravan / Camper' 15000
Application_881 '1/15/2016 10:45' 'A_Validating' User_119 'Existing loan takeover' 15000
Application_881 '1/19/2016 9:00' 'A_Incomplete' User_29 'Existing loan takeover' 15000
Application_881 '1/19/2016 9:00' 'A_Denied' User_29 'Existing loan takeover' 15000
Application_174 '1/20/2016 8:10' 'A_Validating' User_114 'Existing loan takeover' 11000
Application_174 '1/20/2016 14:13' 'A_Incomplete' User_99 'Existing loan takeover' 11000
Application_174 '1/21/2016 11:16' 'A_Validating' User_116 'Existing loan takeover' 11000
Application_174 '1/21/2016 12:10' 'A_Incomplete' User_95 'Existing loan takeover' 11000
Application_174 '1/21/2016 15:04' 'A_Validating' User_41 'Existing loan takeover' 11000
Application_174 '1/22/2016 8:12' 'A_Pending' User_95 'Existing loan takeover' 11000
Application_634 '2/4/2016 7:23' 'A_Validating' User_116 'Home improvement' 10000
Application_634 '2/4/2016 8:29' 'A_Incomplete' User_101 'Home improvement' 10000
Application_634 '2/4/2016 17:12' 'A_Pending' User_114 'Home improvement' 10000
Application_797 '2/29/2016 7:02' 'A_Cancelled' User_1 'Car' 13500
", header = TRUE, stringsAsFactors = FALSE)
```

Convert Event_Time to DateTime objects

```
df$Event_Time <- as.POSIXct(df$Event_Time, format="%m/%d/%Y %H:%M")

glimpse(df)
```

```
## Rows: 41
## Columns: 6
## $ Application_Number <chr> "Application_174", "Application_174", "Application_~
## $ Event_Time         <dttm> 2016-01-03 16:49:00, 2016-01-03 16:49:00, 2016-01-~
## $ Activity_Name      <chr> "A_Create Application", "A_Submitted", "A_Concept",~
## $ Resource           <chr> "User_1", "User_1", "User_1", "User_1", "User_1", "~
## $ Loan_Goal          <chr> "Existing loan takeover", "Existing loan takeover",~
## $ Requested_Amount   <int> 11000, 11000, 11000, 15000, 15000, 15000, 15000, 15~
```

1. How many events are there?

```
nrow(df)
```

```
## [1] 41
```

2. How many activities are there?

```
length(unique(df$Activity_Name))
```

```
## [1] 10
```

3. How many cases are there?

```
length(unique(df$Application_Number))
```

```
## [1] 5
```

4. Each case in this event log has a sequence of activities, which we call a trace in process mining.

For each case, write down the corresponding trace. Note that multiple cases may correspond to the same trace, i.e., sequence of activities. How many unique traces are there? In process mining, we call these unique traces also trace variants

```
# manually
traces <- df %>%
  group_by(Application_Number) %>%
  arrange(Event_Time, .by_group = TRUE) %>%
  summarise(trace = str_c(Activity_Name, collapse = " -> ")) %>%
  ungroup()
```

```
trace_variants <- traces %>%
  group_by(trace) %>%
  summarise(count = n()) %>%
  ungroup()

trace_variants
```

```
## # A tibble: 4 x 2
##   trace                                                                count
##   <chr>                                                                <int>
## 1 A_Create Application -> A_Submitted -> A_Concept -> A_Accepted -> A_Com~     1
## 2 A_Create Application -> A_Submitted -> A_Concept -> A_Accepted -> A_Com~     1
## 3 A_Create Application -> A_Submitted -> A_Concept -> A_Accepted -> A_Com~     2
## 4 A_Create Application -> A_Submitted -> A_Concept -> A_Accepted -> A_Com~     1
```

```
nrow(trace_variants)
```

```
## [1] 4
```

with specialized libraries

```
library(bupaR)
```

```
df_ready <- df %>%
  mutate(
    instance_id = as.character(row_number()), # activity instance, required
    lifecycle = "lifecycle" # bupaR needs to know the status of the event; dummy
  )

eventlog <- df_ready %>%
  eventlog(
    case_id = "Application_Number",
    activity_id = "Activity_Name",
    timestamp = "Event_Time",
    activity_instance_id = "instance_id",
    lifecycle_id = "lifecycle", # optional if you have start/complete lifecycle
    resource_id = "Resource" # optional
  )

trace_variants_bupa <- traces(eventlog)
trace_variants_bupa
```

```
## # A tibble: 4 x 3
##   trace                                    absolute_frequency relative_frequency
##   <chr>                                                 <int>              <dbl>
## 1 A_Create Application,A_Submitted,A_Conc~                   2                0.4
## 2 A_Create Application,A_Submitted,A_Conc~                   1                0.2
## 3 A_Create Application,A_Submitted,A_Conc~                   1                0.2
## 4 A_Create Application,A_Submitted,A_Conc~                   1                0.2
```

5. Find the case with the maximum number of events. How many events does the corresponding trace have?

```
trace_variants <- trace_variants %>%
  mutate(num_events = str_count(trace, "->") + 1)  # each "->" means one separator, so add 1

max_trace <- trace_variants %>%
  filter(num_events == max(num_events))
max_trace
```

```
## # A tibble: 1 x 3
##   trace                                                          count num_events
##   <chr>                                                          <int>      <dbl>
## 1 A_Create Application -> A_Submitted -> A_Concept -> A_Accept~     1         11
```

```
cases_max_trace <- traces %>%
  filter(trace == max_trace$trace)
cases_max_trace
```

```
## # A tibble: 1 x 2
##   Application_Number trace
##   <chr>              <chr>
## 1 Application_174    A_Create Application -> A_Submitted -> A_Concept -> A_Acce~
```

6. How many loan goals exist in this event log?

```
length(unique(df$Loan_Goal))
```

```
## [1] 4
```

```
table(df$Loan_Goal)
```

```
##
##                     Car       Caravan / Camper Existing loan takeover
##                       6                      8                     19
##       Home improvement
##                       8
```

7. How many resources exist in this event log?

```
length(unique(df$Resource))
```

```
## [1] 15
```

8. Which of the following statements are correct?

[X] All cases in the event log start with activity A_Create Application.

```r
df %>%
  group_by(Application_Number) %>%
  arrange(Event_Time, .by_group = TRUE) %>%
  slice_head(n = 1) %>% # first event of each case
  summarise(first_activity = first(Activity_Name)) %>%
  ungroup() %>%
  distinct(first_activity)
```

```
## # A tibble: 1 x 1
##   first_activity
##   <chr>
## 1 A_Create Application
```

[ ] All cases in the event log end with A_Cancelled or A_Pending.

```r
df %>%
  group_by(Application_Number) %>%
  arrange(Event_Time, .by_group = TRUE) %>%
  slice_tail(n = 1) %>%                    # last event of each case
  summarise(last_activity = first(Activity_Name)) %>%
  ungroup() %>%
  filter(!last_activity %in% c("A_Cancelled", "A_Pending"))
```

```
## # A tibble: 1 x 2
##   Application_Number last_activity
##   <chr>              <chr>
## 1 Application_881    A_Denied
```

[ ] Requested_Amount for one particular case may change during the process.

```r
df %>%
  group_by(Application_Number) %>%
  summarise(n_unique_amount = n_distinct(Requested_Amount)) %>%
  filter(n_unique_amount > 1)
```

```
## # A tibble: 0 x 2
## # i 2 variables: Application_Number <chr>, n_unique_amount <int>
```

[X] Activity A_Submitted is always executed by the same resource.

```r
df %>%
  filter(Activity_Name == "A_Submitted") %>%
  summarise(n_resources = n_distinct(Resource))
```

```
##   n_resources
## 1           1
```