Guide for Module 1.1 exercises

Data Preparation, Summarization and Visualization

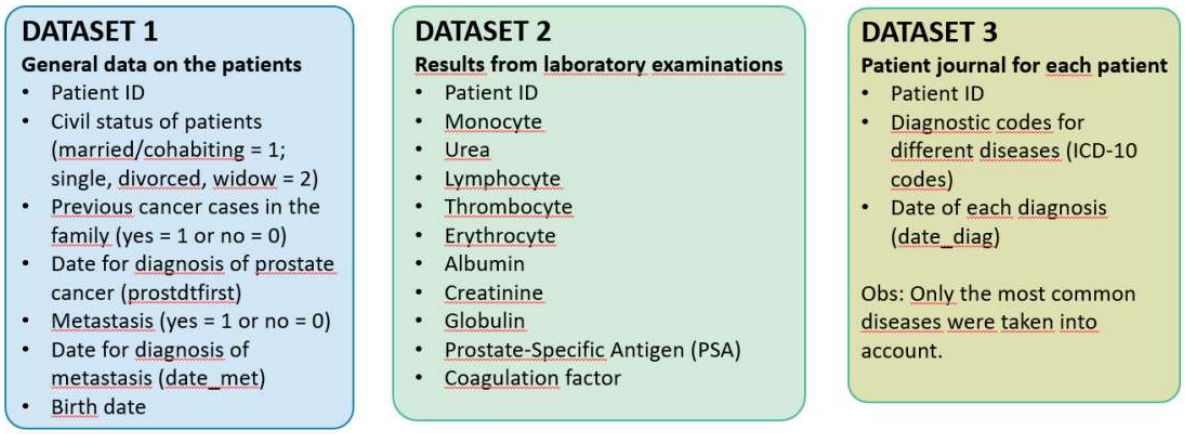
Study description:

Urologists from the Urology department at Odense University Hospital have selected 1159 patients that were diagnosed with prostate cancer (PC) at their first visit with them.

Some of those patients were also diagnosed with metastasis (metastasis = 1) and others do not present metastasis (metastasis = 0). Metastatic cases were detected by the appropriate method (e.g. CT scans, MRI).

The urologists have requested a bunch of laboratory examinations from these patients (measured in blood and urine), as well as retrieved their medical history (i.e. patient journals).

They got in contact with us, the AI team, to help them to identify ways of predicting which patients will present metastasis and which will not. This could be further used as a prediction tool in an earlier stage, and would also avoid overtreatment and invasive examinations.



Can PSA (Prostate-Specific Antigen) be used as a biomarker to detect metastatic prostate cancer?

Previous studies say the following:

“Based on results from some small, single centre studies published in the beginning of the 1990s, PSA levels above 100 ng/mL have been used as a proxy for metastatic prostate cancer” Thomsen et al., 2020

“PSA < 20 ng/ml have high predictive value in ruling out skeletal metastasis” Kamaleshwaran et al., 2012

TASK 1 (DATASET 1) - Opening data and making basic statistics

• Open DATASET 1 and inspect it briefly (e.g. see which variables you have in this dataset, how the data is structured, etc).

Tips in R:

- Make sure to install the appropriate packages for the exercise. I recommend the dplyr and tidyr packages. To install and open the package, you need to do the following:

  install.packages("dplyr")

  library(dplyr)

- To open a csv file in R, you can use the following:

  mydata <- read.csv("dataset.csv")

- The function names(mydata) can be used to see the name of the variables in the dataset. You can also click at the dataset on the Environment tab to see how it looks like.

- Are the variables correctly recognized by the software (e.g. are categorical variables recognized as categorical, and numerical variables recognized as numerical)?

  - If you are using R, one tip is to use the str() function. The summary() function also gives you a good overview. Inside the parenthesis, you add the name of your dataset.

  - If not, transform the variables to their appropriate format In R, this can be done with the functions "as.factor", "as.numeric", etc. One example:

    df$Sex <- as.factor(df$Sex)

    df$Date_visit <- as.Date(df$Date_visit, format = "%d-%m-%Y")

- Do we have any missing data? Which variables are affected?

  - Tip: In R, you can use:

    data %>% summarise_all(funs(sum(is.na(.)))) #It prints the amount of NA for each of the variables

- With information available in DATASET 1, make a summary table describing the study population, e.g.:

  - Age of patients at diagnosis of prostate cancer (mean and standard deviation)
  - Age of patients at diagnosis of metastasis (for those that metastasis = 1)
  - Percentage of patients with and without metastasis
  - Previous cancer in the family
    - 1 = Patient reported at least one case of aggressive cancer in the family; 0 = no aggressive cancer in the family
  - Civil status
    - 1 = Married/cohabiting; 2 = Single/divorced/widow

One example for the table is the following:

Patients' characteristics		
Age at time of PC diagnosis	Mean ± standard deviation	
Age at time of metastasis	Mean ± standard deviation	
Metastasis		
Yes=1	N of patients with metastasis (%)	
No=0	N of patients without metastasis (%)	
Previous cancer in the family		
Yes=1	N of patients with previous cancer in family (%)	
No=0	N of patients without previous cancer in family (%)	
Civil status		
Married or cohabiting = 1	N of patients married or cohabiting (%)	
Single, divorced or widow = 2	N of patients not married or cohabiting (%)	

• Tip 1: In R, you can use the package lubridate and the function year to calculate the age at time of PC diagnosis or the age at time of metastasis:

library(lubridate)

df$age_PCD <- year(as.period(interval(start=df$birthdate, end=df$PCDdate)))

• Tip 2: In R, you can use the package psych and the function describe() to obtain mean and standard deviation of the quantitative variables of interest as well as the function summary() to obtain the number of specific characteristics for the qualitative variables of interest.

TASK 2 (DATASET 2) – Inspecting biochemical data

Open DATASET 2. What can you say about:

1) Missing data

• Tip 1: In R, you can use:

data %>% summarise_all(funs(sum(is.na(.)))) #It prints the amount of NA for each of the variables

• Tip 2: If you are using R, the package VIM has some nice visualization options.

aggr_plot <- aggr(data_mis, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE, labels=names(data_mis), cex.axis=.7, gap=3, ylab=c("Bar graph of missing data","Pattern"))

2) Correlation between variables

• Tip: If you are using R, a nice visualization graph:

library(ggplot2)

library(GGally)

ggcorr(df, hjust=0.76, size=5, color="grey50", layout.exp = 1)

Discuss what you see in pairs or within your group.

TASK 3 (DATASET 3) – Preparing the data

• Open DATASET 3 – how does the data look like?

• What do the codes mean? Inspect them in e.g. Medinfo: https://medinfo.dk/sks/brows.php

You can inspect the codes in R using:

summary(as.factor(df$code))

• Consider here to create a new variable called e.g. "disease", where you assign to each code the respective disease it refers to. One example in R is to use the function mutate, from dplyr package (though you should do whatever makes you more comfortable):

```
library(dplyr)

df2 <- df %>% mutate(disease = case_when(code == "XXX" ~ "Disease1", code == "YYY" ~ "Disease2", TRUE ~ "other"))
```

• Create binary variables for each of the diseases (consider renaming them instead of using the codes) indicating whether the patients have/do not have a diagnosis for that specific disease. At the same time, transform the data, so that it has the same shape as DATASET1 and DATASET2 (i.e. one row per patient)

• In R, one of the options is to do the following:

```
library(dplyr)

library(tidyr)

df3 <- df2 %>% mutate(diag = 1) %>% select(ID, disease, diag)

df4 <- spread(df3, disease, diag)

df4[is.na(df4)] <- 0
```

TASK 4 (DATASET 1-3) – Merging the data

• Merge all datasets (DATASET 1, DATASET 2, and DATASET 3) together. For DATASET 2 (i.e. the one with biochemistry examinations), remember to use the new dataset you created in Task 2.

In R, merge() function can be used. When you want to keep all observations of one of the datasets, you can use all.x = TRUE

Make sure that, for all the patients that did not get a diagnosis for a certain disease, they then have the binary variable for that specific disease equal to 0. This can be done with the following coding:

```
df[c(x:y)][is.na(df[c(x:y)])] <- 0
```

<where x:y are the number of all the columns corresponding to diseases>

The idea is to have a dataset like the following:

ID	civil_status	cancer_family	prostdtfirst	metastasis	date_met	birthdate	diabetes
1	1	1	2017-03-16	0	NA	1935-01-01	1
2	2	0	2005-03-31	0	NA	1942-01-01	0
3	2	0	2017-08-08	0	NA	1959-01-01	0
4	2	0	2008-01-02	0	NA	1934-02-01	0
5	2	0	2006-05-08	0	NA	1949-03-01	0
6	2	0	2019-11-13	0	NA	1951-03-01	1
7	1	1	2012-05-01	0	NA	1927-04-01	0
8	1	1	2013-10-20	0	NA	1930-04-01	0
9	1	1	2015-10-27	0	NA	1935-04-01	1
10	1	0	2015-11-13	0	NA	1937-04-01	0
11	1	0	2017-11-13	0	NA	1940-04-01	0
12	1	1	2014-10-21	1	2014-10-21	1945-10-09	0
13	2	0	2018-06-21	0	NA	1952-04-01	0

• Inspect the data and variables. If you are using R, one tip is to use the str function. Once more, make sure that the variables are correctly recognized by the software (e.g. categorical variables are recognized as factor and numerical variables are recognized as numerical).

• Let's use one of the discussed methods to impute the missing values in the dataset.

Because of time constraints, I suggest you to use the mean/median substitution. You are of course welcome to use others. Just remember to briefly justify your choice (in the portfolio) and also to briefly mention the main limitations of the chosen method.

One example in R is the following:

```
df$albumine[is.na(df$albumine)] <- mean(df$albumine, na.rm = TRUE)
```

After imputing missing data, I suggest you to inspect the data again, to make sure your data has no missing data anymore.

For the report, make sure to include (at least) the following:

• Start your report by describing the module's objective and the clinical motivation behind this work.

• Include the summary table from TASK 1 and describe the data you use in this Module (DATASET 1, DATASET 2, DATASET 3).

Remember to always save the script!