EDA Jorick Baron

Jorick Baron

```
library(tidyr)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(stringr)
library(knitr)
library(kableExtra)
```

In this EDA we will explore the data downloaded from here. For future reference we will describe the data in the codebook below.

```
codebook <- read.delim("Data/codebook.csv", sep = ",")
kable(codebook, caption = "Table 1: Codebook", align = "lcccr", booktabs = T) %>%
  kable_styling(latex_options = c("scale_down"))
```

Table 1: Table 1: Codebook

Name	Fullname	Description	Type	Unit
sample_id	Sample ID	Unique string identifying each subject	string	NA
patient_cohort	Patient's Cohort	Cohort 1, previously used samples; Cohort 2, newly added samples	string	NA
sample_origin	Sample Origin	BPTB: Barts Pancreas Tissue Bank; ESP: Spanish National Cancer Research Centre; LIV: Liverpool University;UCL: University College	string	NA
age	Age	Age in years	int	years
sex	Sex	M = male, F = female	char	NA
diagnosis	Diagnosis (1=Control, 2=Benign, 3=PDAC)	1 = control, 2 = benign hepatobiliary disease; 3 = Pancreatic ductal adenocarcinoma, i.e. pancreatic cancer	int	NA
stage	Stage	For those with pancratic cancer, what stage was it? One of IA, IB, IIA, IIIB, III, IV	string	NA
benign_sample_diagnosis	Benign Samples Diagnosis	For those with a benign, non-cancerous diagnosis, what was the diagnosis?	string	NA
plasma_CA19_9	Plasma CA19-9 U/ml	Blood plasma levels of CA 19–9 monoclonal antibody that is often elevated in patients with pancreatic cancer.	float	plasma units/milliliter
creatinine	Creatinine mg/ml	Urinary biomarker of kidney function	float	mg/ml
LYVE1	LYVE1 ng/ml	Urinary levels of Lymphatic vessel endothelial hyaluronan receptor 1, a protein that may play a role in tumor metastasis	float	ng/ml
REG1B	REG1B ng/ml	Urinary levels of a protein that may be associated with pancreas regeneration.	float	ng/ml
TFF1	TFF1 ng/ml	Urinary levels of Trefoil Factor 1, which may be related to regeneration and repair of the urinary tract	float	ng/ml
REG1A	REG1A ng/ml	Urinary levels of a protein that may be associated with pancreas regeneration.	float	ng/ml

First we will load in the data and to check if it has loaded in properly we look at the structure of the loaded data.

```
data <- read.table(file="Data/source/Debernardi_et_al_2020_data.csv", sep = ",", header = T, na.strings
str(data)
```

```
## 'data.frame':    590 obs. of  14 variables:
##  $ sample_id              : chr  "S1" "S10" "S100" "S101" ...
##  $ patient_cohort         : chr  "Cohort1" "Cohort1" "Cohort2" "Cohort2" ...
##  $ sample_origin          : chr  "BPTB" "BPTB" "BPTB" "BPTB" ...
##  $ age                    : int  33 81 51 61 62 53 70 58 59 56 ...
##  $ sex                    : chr  "F" "F" "M" "M" ...
##  $ diagnosis              : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ stage                  : chr  NA NA NA NA ...
##  $ benign_sample_diagnosis: chr  NA NA NA NA ...
##  $ plasma_CA19_9          : num  11.7 NA 7 8 9 NA NA 11 NA 24 ...
##  $ creatinine             : num  1.832 0.973 0.78 0.701 0.215 ...
##  $ LYVE1                  : num  0.89322 2.03758 0.14559 0.0028 0.00086 ...
##  $ REG1B                  : num  52.9 94.5 102.4 60.6 65.5 ...
##  $ TFF1                   : num  654.3 209.5 461.1 142.9 41.1 ...
##  $ REG1A                  : num  1262 228 NA NA NA ...
```

Thus far it seems to have loaded correctly.

We will also check the first few records maybe catch some errors.

```
head(data)
```

```
##   sample_id patient_cohort sample_origin age sex diagnosis stage
## 1        S1        Cohort1          BPTB  33   F         1  <NA>
## 2       S10        Cohort1          BPTB  81   F         1  <NA>
## 3      S100        Cohort2          BPTB  51   M         1  <NA>
## 4      S101        Cohort2          BPTB  61   M         1  <NA>
## 5      S102        Cohort2          BPTB  62   M         1  <NA>
## 6      S103        Cohort2          BPTB  53   M         1  <NA>
##   benign_sample_diagnosis plasma_CA19_9 creatinine      LYVE1     REG1B
## 1                    <NA>          11.7    1.83222 0.89321920  52.94884
## 2                    <NA>            NA    0.97266 2.03758500  94.46703
## 3                    <NA>           7.0    0.78039 0.14558890 102.36600
## 4                    <NA>           8.0    0.70122 0.00280488  60.57900
## 5                    <NA>           9.0    0.21489 0.00085956  65.54000
## 6                    <NA>            NA    0.84825 0.00339300  62.12600
##       TFF1    REG1A
## 1 654.2822 1262.000
## 2 209.4882  228.407
## 3 461.1410       NA
## 4 142.9500       NA
## 5  41.0880       NA
## 6  59.7930       NA
```

The data seems to have quite a few NAs, reading further into the description most would be expected i.e. no stage if there is no cancer nor a diagnosis.

Let's check that nothing went wrong with those two anyway.

```
healthy <- subset(data, diagnosis == 1, select = c(diagnosis, stage, benign_sample_diagnosis))
cancerfree <- subset(data, diagnosis == 2, select = c(diagnosis, stage, benign_sample_diagnosis))
cancerous <- subset(data, diagnosis == 3, select = c(diagnosis, stage, benign_sample_diagnosis))
```

```
stage_na_count <- sum(is.na(data$stage))
bsd_na_count <- sum(is.na(data$benign_sample_diagnosis))
paste("all these numbers should be the same number", stage_na_count - nrow(cancerfree), bsd_na_count - r
```

```
## [1] "all these numbers should be the same number 183 183 183"
```

Those numbers lined up to expectations.

The NAs in columns "plasma_CA19_9" and "REG1A" are supposed to be there because not every patient had been fully tested:
"REG1A ... Only assessed in 306 patients", "plasma_CA19_9 ... Only assessed in 350 patients" see Debernardi et al 2020 documentation.csv in the source files.
However to make sure everything is correct these numbers will be tested.

```
n_plasma_CA19_9 <- nrow(data) - sum(is.na(data$plasma_CA19_9))
n_REG1A <- nrow(data) - sum(is.na(data$REG1A))
paste("REG1A:", n_REG1A, "plasma_CA19_9:", n_plasma_CA19_9)
```

```
## [1] "REG1A: 306 plasma_CA19_9: 350"
```

These numbers are correct.

Are there more NAs?

```
sum(is.na(data[, c(1:6, 10:13)]))
```

```
## [1] 0
```

0 NAs remaining.

```
boxplot(data[c(4, 9:14)])
```