Inlämningsuppgift 5 - från data till inferens.

Statistiska Metoder med R

Jacob Widaeus

2024-10-13

Bakgrund

Till detta arbete används ett allmänt tillgängligt dataset som heter "jasa" och inkluderas i paketet survival. Den finns även bifogad som .rda fil som en del i inlämningen.

Jag kommer använda tidyverse till de flesta datamanipulationer och ggplot till grafritningar, då jag använt det tidigare och planerar använda det i framtiden.

Datan beskriver överlevnad av patienter på väntelistan till Standford hjärttransplantationsprogram. Den kommer i följande format:

Variable	Description
birth.dt	Birth date
accept.dt	Acceptance into program
tx.date	Transplant date
fu.date	End of followup
fustat	Dead or alive
surgery	Prior bypass surgery
age	Age (in years)
futime	Followup time
wait.time	Time before transplant
transplant	Transplant indicator
mismatch	Mismatch score
hla.a2	Particular type of mismatch
mscore	Another mismatch score
reject	Rejection occurred

Hypotes

Min hypotes är att ålder, tidigare kirurgi, organavstötning är de variabler som ökar risken mest för att predicera död i de som erhåller hjärttransplantation.

Dependencies

Detta arbete bygger på följande paket:

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(survival)
library(fs)
```

Import

Först importeras datan genom att skapa en funktion för att skapa en ny mapp “data” och spara den i en csv fil där datum inkluderas.

```
initiate <- function(output_path) {
  dir_path <- dirname(output_path)
  if (!dir.exists(dir_path)) dir.create(dir_path)
  today_date <- Sys.Date()
  file_name <- basename(output_path)
  file_extension <- tools::file_ext(file_name)
  file_base <- tools::file_path_sans_ext(file_name)
```

```
  # Remove any existing date from the file_base
  file_base <- sub("_\\d{4}-\\d{2}-\\d{2}$", "", file_base)

  new_file_name <- paste0(file_base, "_", today_date, ".", file_extension)
  new_output_path <- file.path(dir_path, new_file_name)

  # Assuming 'jasa' is a data frame that you want to write to CSV
  write.csv(jasa, new_output_path)
}
```

base_dir definieras som bas-filsökvägen. Denna är datorberoende.

```
base_dir <- "C:/Users/oesma/Desktop/kiRstat/block5"
```

```
output_path <- file.path(base_dir, paste0("data/jasa_", Sys.Date(), ".csv"))
```

Därefter läses .csv filen in, och samtidigt sparas som en .rda fil för redundans - även här med datum. Om den som granskar detta ska återskapa, se till att ha rätt path och använd helst .r filen då .qmd inte alltid samarbetar väl med importfunktioner.

```
initiate(output_path)
data <- read.csv(file.path(base_dir, paste0("data/jasa_", Sys.Date(), ".csv")))
save(data, file = file.path(base_dir, paste0("data/jasa_", Sys.Date(), ".rda")))
```

Översikt och städa upp klasser

Först bildar jag mig en översikt över datan med glimpse()

```
glimpse(data)
```

```
Rows: 103
Columns: 15
$ X          <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, ~
$ birth.dt   <chr> "1937-01-10", "1916-03-02", "1913-09-19", "1927-12-23", "19~
$ accept.dt  <chr> "1967-11-15", "1968-01-02", "1968-01-06", "1968-03-28", "19~
$ tx.date    <chr> NA, NA, "1968-01-06", "1968-05-02", NA, NA, "1968-08-31", N~
$ fu.date    <chr> "1968-01-03", "1968-01-07", "1968-01-21", "1968-05-05", "19~
$ fustat     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ surgery    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,~
```

```
$ age        <dbl> 30.84463, 51.83573, 54.29706, 40.26283, 20.78576, 54.59548,~
$ futime     <int> 49, 5, 15, 38, 17, 2, 674, 39, 84, 57, 152, 7, 80, 1386, 0,~
$ wait.time  <int> NA, NA, 0, 35, NA, NA, 50, NA, NA, 11, 25, NA, 16, 36, NA, ~
$ transplant <int> 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1,~
$ mismatch   <int> NA, NA, 2, 3, NA, NA, 4, NA, NA, 2, 1, NA, 3, 1, NA, 2, NA,~
$ hla.a2     <int> NA, NA, 0, 0, NA, NA, 0, NA, NA, 0, 0, NA, 0, 0, NA, 0, NA,~
$ mscore     <dbl> NA, NA, 1.11, 1.66, NA, NA, 1.32, NA, NA, 0.61, 0.36, NA, 1~
$ reject     <int> NA, NA, 0, 0, NA, NA, 1, NA, NA, 1, 0, NA, 1, 1, NA, 1, NA,~
```

Man kan redan nu se att flera av klasserna av variablerna är fel. Jag konverterar datumen till datumklasser, och de binära variablerna som representerar TRUE/FALSE till booleans.

```
# Convert character strings that are dates to Date type
data_cleaned <- data %>%
  mutate(across(where(~ is.character(.) && any(!is.na(as.Date(.)))), as.Date))

# Convert all 0 and 1 integer columns (including those with NAs) to booleans
data_cleaned <- data_cleaned %>%
  mutate(across(where(~ all(. %in% c(0, 1, NA)) && is.numeric(.)), ~ as.logical(.)))
```

Faktorisera alla kolumner som har färre än 10 diskreta variabler, men inte är logical.

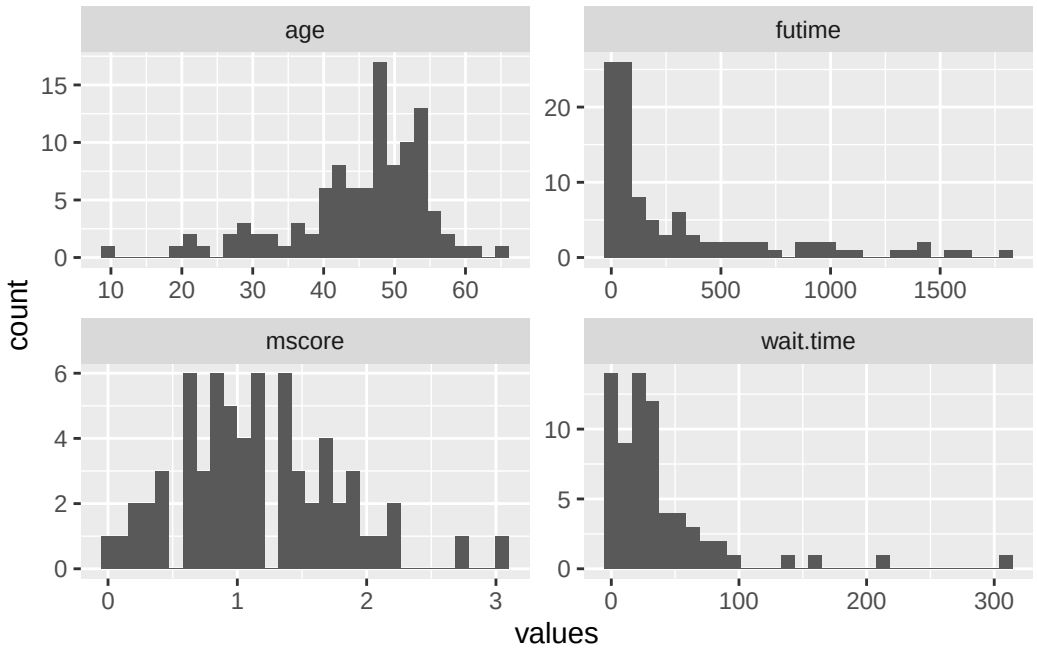
```
data_cleaned <- data_cleaned %>%
  mutate(across(where(~ n_distinct(.) < 10 && !is.logical(.)), as.factor))
```

EDA - Exploratory data analysis

Barplots för att skildra numeriska variabler.

```
data_cleaned %>%
  select_if(is.numeric) %>%
  select(-X) %>%  # Remove the variable 'X'
  gather(key = "variables", value = "values") %>%
  ggplot(aes(x = values)) +
  facet_wrap(~variables, scales = "free") +
  geom_histogram(bins = 30)
```

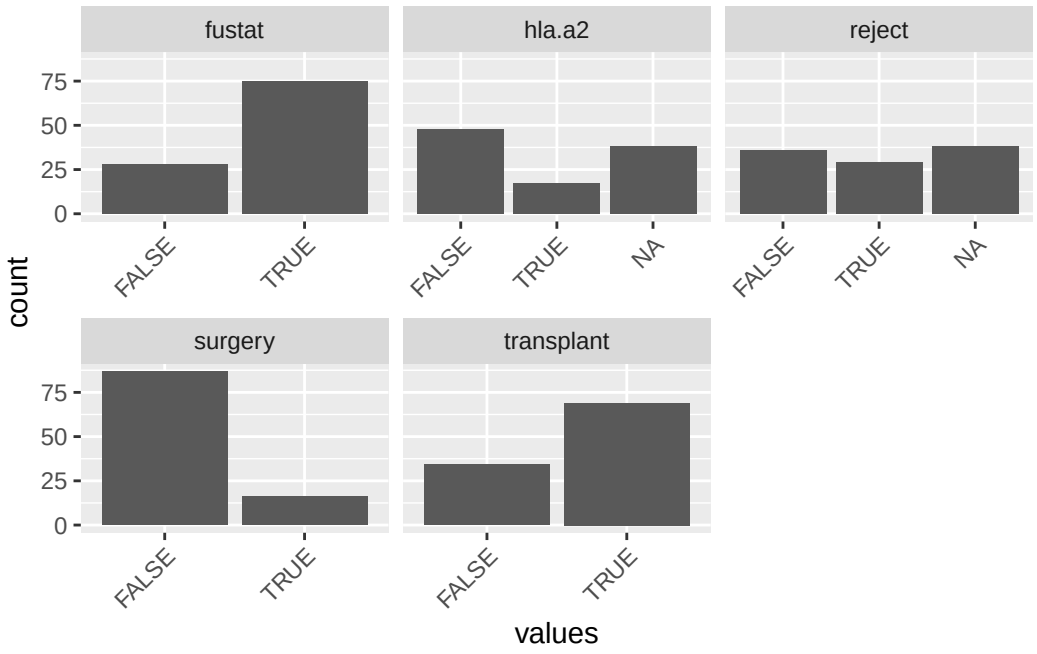
```
Warning: Removed 72 rows containing non-finite outside the scale range
(`stat_bin()`).
```



Man ser att alla numeriska variabler är skewed och ej normalfördelade.

Barplot for logistiska variabler

```
data_cleaned %>%
  select(where(is.logical)) %>%
  pivot_longer(cols = everything(), names_to = "variables", values_to = "values") %>% #nolint
  ggplot(aes(x = values)) +
  facet_wrap(~variables, scales = "free_x") +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Hur många av de som **inte** är NA på reject är NA på hla.a2?

```
count_na_hla_a2 <- data_cleaned %>%
  filter(!is.na(reject)) %>%
  summarize(count = sum(is.na(hla.a2)))

print(count_na_hla_a2)
```

```
  count
1     0
```

Alltså, de som inte har fått rejection finns inga värden på hla provtagning. Det verkar som att man endast provtagit de som fått en rejection.

Reflektion