Analysis of Longitudinal Cardiovascular Data

EDA and Propensity Score Matching

2024/01/07

Contents

Look at the data

```
load("L:/TGH statistical analysis/Jinyu - Practicum student/programs/1 - data setup.RData")
glimpse(raw_long_d)
```

```
## Rows: 1,241
## Columns: 24
## $ ptid          <chr> "ADAI080255", "ADAI080255", "ADAI080255", "ANDR081224", ~
```

```
## $ age           <dbl> 52, 52, 52, 73, 73, 73, 76, 51, 51, 51, 51, 51, 51, 51, ~
## $ sex           <chr> "1_male", "1_male", "1_male", "1_male", "1_male", "1_mal~
## $ bav_confirmed <chr> "1_BAV", "1_BAV", "1_BAV", "1_BAV", "1_BAV", "1_BAV", "1~
## $ nc_sinus      <chr> "1_YES", "1_YES", "1_YES", "2_NO", "2_NO", "2_NO", "2_NO~
## $ raa_type      <chr> "1_clamp", "1_clamp", "1_clamp", "2_hemiarch", "2_hemiar~
## $ died          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ lka_d         <chr> "02MAY2016", "02MAY2016", "02MAY2016", "04SEP2014", "04S~
## $ yr2death      <dbl> 8.421629, 8.421629, 8.421629, 16.826831, 16.826831, 16.8~
## $ ao_reop       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ yr2ao_reop    <dbl> 8.421629, 8.421629, 8.421629, 16.826831, 16.826831, 16.8~
## $ bsa_baseline  <dbl> 2.23, 2.23, 2.23, 1.96, 1.96, 1.96, 1.74, 2.09, 2.09, 2.~
## $ dateor        <chr> "11/30/2007", "11/30/2007", "11/30/2007", "11/06/1997", ~
## $ bsa_echo      <dbl> NA, 2.26, 2.28, 1.93, 1.94, 1.94, NA, NA, 1.84, 1.84, NA~
## $ mdate         <chr> "11/30/2007", "12/04/2007", "07/10/2013", "03/06/2013", ~
## $ root          <dbl> 39, 24, 41, 35, 32, 35, 54, 33, 36, 39, NA, NA, 43, 41, ~
## $ aa            <dbl> 50, NA, 42, 27, 27, NA, NA, 47, NA, 29, NA, NA, NA, 30, ~
## $ arch          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 32.2, 31.5, NA, ~
## $ src_root      <dbl> 3, 3, 3, 3, 3, 3, 2, 3, 3, 3, NA, NA, 3, 3, NA, NA, NA, ~
## $ src_aa        <dbl> 3, NA, 3, 3, 3, NA, NA, 3, NA, 3, NA, NA, NA, 3, NA, NA,~
## $ src_arch      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 2, 2, NA, NA, 2,~
## $ day2vst       <dbl> 0, 4, 2049, 5599, 5602, 6018, -1, 0, 6, 27, 1630, 1706, ~
## $ yr2vst        <dbl> 0.000000000, 0.010951403, 5.609856263, 15.329226557, 15.~
## $ date_base     <chr> "11/30/2007", "11/30/2007", "11/30/2007", NA, NA, NA, "0~
```

```
glimpse(work_pt_d)
```

```
## Rows: 406
## Columns: 10
## $ ptid     <fct> ADAI080255, ANDR081224, AQUC020122, AREA112454, ARSU110342, A~
## $ age      <int> 52, 73, 76, 51, 63, 75, 65, 72, 36, 58, 79, 73, 71, 53, 34, 8~
## $ sex      <fct> 1_male, 1_male, 1_male, 1_male, 1_male, 2_female, 1_male, 1_m~
## $ bav      <fct> BAV, BAV, BAV, BAV, BAV, TAV, BAV, BAV, TAV, BAV, BAV, TAV, T~
## $ nc_sinus <fct> 1_YES, 2_NO, 2_NO, 1_YES, 1_YES, 1_YES, 2_NO, 1_YES, 2_NO, 2_~
## $ raa_type <fct> 1_clamp, 2_hemiarch, 2_hemiarch, 1_clamp, 2_hemiarch, 2_hemia~
## $ bsa      <dbl> 2.20, 1.90, 1.74, 2.06, 2.00, 1.40, 2.40, 1.77, 2.10, 1.81, 1~
## $ diabetes <fct> 0_Negative, 1_Diabetes, 0_Negative, 0_Negative, 0_Negative, 0~
## $ hyper    <fct> 0_Negative, 1_Hypertension, 0_Negative, 0_Negative, 0_Negativ~
## $ chlstrl  <fct> 0_Negative, 0_Negative, 0_Negative, 0_Negative, 0_Negative, 1~
```

Observations:

- `raw_long_d` the raw data in long format, which had multiple rows of information for some patients, because the ptid of the first three rows in `raw_long_d` are repeated. It contains 1241 rows with 24 columns.

- `work_pt_d` is a smaller dataset which contains 405 rows with 10 columns. Some patients in this dataset also appears in `raw_long_d`, e.g., patient with ID ADAI080255.
- The uniqueness of `ptid` in `work_pt_d` is undetermined.
- Both `raw_long_d` and `work_pt_d` contains
  - `ptid`: unique patient ID number
  - `age`: the patient's age

  - `sex`: the patient's biological sex
  - `nc_sinus`: A column indicates Whether or not the patient had replacement of the non-coronary sinus of the aortic root.
  - `raa_type`: A column indicates type of ascending aorta replacement.
    * 1 = Clamp on
    * 2 = Hemi-arch

- Values in columns like `sex`, `bav_confirmed`, `nc_sinus`, and `raa_type` in both datasets are strings with numerical number of categories.
- The column `bav_confirmed` in `raw_long_d` indicates exposure group, but we are not sure whether this column consistent with the column `bav` in `work_pt_d`.
- Columns `bsa_baseline` in `raw_long_d` is numerical with similar values to the column `bsa` in `work_pt_d`.

Next, we will perform the following steps:

1. Determine the number of unique patient in the data `raw_long_d`.
2. Check whether each patient has unique row of data in `work_pt_d`.
3. Add a column to `raw_long_d` to record the number of visit for each patient and clean categorical columns.
4. Determine whether the column `bav_confirmed` in `raw_long_d` is consistent with the column `bav` in `work_pt_d`.
5. Determine whether column `bsa_baseline` in `raw_long_d` is consistent with the column `bsa` in `work_pt_d`.
6. Determine the number of patients that are contained in both datasets.
7. Merge `work_pt_d` with `raw_long_d` to obtain a full long raw dataframe and clean columns.

Step 1:

```
long_unique <- raw_long_d %>%
  summarise(n_distinct(ptid))

print(paste("The number of unique patient' ID in raw_long_d is",
            long_unique))
```

```
## [1] "The number of unique patient' ID in raw_long_d is 298"
```

The `raw_long_d` dataset contains information from 298 patients.

Step 2:

```r
# check whether pid in work_pt_d is unique
n_uniques <- work_pt_d %>%
  summarise(n_distinct(ptid)) %>%
  pull()

print(paste("The number of unique patient' ID in work_pt_d is", n_uniques))
```

```
## [1] "The number of unique patient' ID in work_pt_d is 406"
```

The output above tells that each row of data in `work_pt_d` come from different individuals.

Step 3:

```r
raw_long <- raw_long_d %>%
  group_by(ptid) %>%
  mutate(visit = row_number(),
         total_visit = n())%>% # count the number of visit
  relocate(visit, .after = died) %>%
  ungroup() %>%
  mutate_at(vars(bav_confirmed),  ~str_replace(., "^[01]_", "")) %>%
  mutate(bav_confirmed = factor(bav_confirmed, levels = c("TAV", "BAV"))) %>% # ordering the exposure g
  mutate_at(vars(nc_sinus, raa_type, sex), ~str_replace(., "^[12]_", ""))
```

Step 4:

```r
# Check consistency of BAV values between work_pt_d and raw_long_d
work_pt_d %>%
  left_join(raw_long, by = "ptid") %>%  # Filter rows with matching ptid
  select(ptid, bav, bav_confirmed) %>%    # Select relevant columns
  filter(bav != bav_confirmed) %>%        # Filter rows with inconsistent values
  nrow()
```

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

The column `bav_confirmed` in `raw_long_d` is consistent with the column `bav` in `work_pt_d`.

Step 5:

```r
work_pt_d %>%
  select(ptid, bsa) %>%
  right_join(raw_long %>%
               filter(visit == 1) %>%
               select(ptid, bsa_baseline),
             by = "ptid") %>%
  group_by(ptid) %>%
  filter(bsa != bsa_baseline) %>%
  nrow()
```

```
## [1] 151
```

151 out of 298 patients had inconsistent record of bsa values, so we will keep these two columns during the further merge step.

Step 6:

```r
work_pt_d %>%
  select(ptid) %>%
  filter(ptid %in% raw_long$ptid) %>%
  nrow()
```

```
## [1] 298
```

298 ptid in `work_pt_d` are also in `raw_long_d`, meaning that every patient in `raw_long_d` are included in `work_pt_d`.

Step 6:

```r
data <- raw_long %>%
  left_join(select(work_pt_d,
                   ptid, diabetes, hyper, chlstrl, bsa),
            by = "ptid") %>%
  relocate(diabetes, hyper, chlstrl, bsa, .before = died) %>%
  mutate_at(vars(diabetes, hyper, chlstrl), ~str_replace(., "^[01]_", "")) %>%
  select(-c(bsa_echo, aa, arch))
```

Supplementary Information

- Prefix `src_` indicates the source. For example, `src_root` == 1 means that the root size is measured by MRI, 2 indicates CT, and 3 indicates ECHO.
- Columns `bsa_echo`, `aa`, and `arch` contains lots of missing values and can be ignored for now, so we removed them from the data.
- The column `ao_reop` is a derived competing risk indicator, where we are interested in reoperation on aorta while considering death as the competing event. Therefore, 0 = alive, 1= reoperation on aorta (before death if any) and 3 = death (dead before reoperation on aorta if any).

```r
write.csv(data, file = "L:/TGH statistical analysis/Jinyu - Practicum student/data/full_data.csv")
```

Data Cleaning

Check missing

```r
sapply(data, function(x) sum(is.na(x)))
```

```
##          ptid           age           sex bav_confirmed      nc_sinus
##             0             0             0             0             0
##      raa_type      diabetes         hyper       chlstrl           bsa
##             0             0             0             0            18
##          died         visit         lka_d      yr2death       ao_reop
##             0             0             0             0             0
##    yr2ao_reop  bsa_baseline        dateor         mdate          root
##             0             0             0             0           148
##      src_root        src_aa      src_arch       day2vst       yr2vst
##           148           521          1033             0             0
##     date_base   total_visit
##           353             0
```

Observations:

- The number of missing in columns `root` and `src_root` are the same, i.e., 148. Since this is a long table, the number of patients without root measurement might be less than 148.
- The number of missing values in columns `src_aa` and `src_arch` is over 500, so we will ignore these two columns for further analysis for now.
- Over one third of these records had `date_base` missing, so there is need to explore the reason of missing.

Empty Root Size

Remove patients who had only one visit.

```
# Remove patients who had only one visit.
data <- data %>%
  filter(!(total_visit == 1))

# Find records with missing root
root_na <- data %>%
  filter(is.na(root)) %>% # find those with missing root size
  select(ptid, age, sex, bav_confirmed, visit, total_visit) %>%
  group_by(ptid) %>%
  summarise(visit_start = min(visit), # the first visit that root size appear to be empty
            n = n(), total = max(total_visit))
head(root_na)
```

```
## # A tibble: 6 x 4
##   ptid       visit_start     n total
##   <chr>            <int> <int> <int>
## 1 AREA112454           4    11    18
## 2 ARSU110342           1     1     4
## 3 ASSJ060731           7     1     9
## 4 ATWA101031           1     1     3
## 5 AYOC083067           2     1     2
## 6 BAEC032439           2     1     2
```

Patients whose root size was never measured at any of the visit

Patients whose root size was never measured at any of the visit should be removed from the data

```
to_exclude <- root_na %>%
  filter(n == total)

data1 <- data %>%
  filter(!(ptid %in% to_exclude$ptid))
nrow(to_exclude)
```

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

```r
nrow(data1)
```

```
## [1] 1182
```

One patient who had no root measurement at any visit were excluded.

Patients had no root measurement at their first visit.

1. Remove the first record of patients whose root size was measured from their second visit.
2. Remove patients whose root size was only measured once among multiple visits.
3. Only retain records that has root size data for patients whose root size was measured more than once but the number of measurements are less than their total number of visit.

```r
# let rm denote root missing and "_1" indicating no root measure during the first visit
rm_1 <- root_na %>%
  filter(!(ptid %in% to_exclude$ptid) &
    visit_start == 1)

# Patients who didn't have root size at the first visit only
rm_1_only <- rm_1 %>%
  filter(n == 1 & total > 1)

data.frame(
  case = c("Root size was not measured at the first visit",
           "Root size was not measured at the first visit only among all visits",
           "Root size was measured at some visits"),
  N = c(nrow(rm_1), nrow(rm_1_only), nrow(rm_1) - nrow(rm_1_only)))
```

```
##                                                                 case  N
## 1                       Root size was not measured at the first visit 34
## 2 Root size was not measured at the first visit only among all visits 28
## 3                                Root size was measured at some visits  6
```

Table 1: Root size missing pattern in patients whose first measurement was not taken at their first visit.

Root size was not measured at the first visit	N = 34
Root size was not measured at the first visit only	28
Root size was measured at some visits	6

28 out 34 patients whose root size was measured at every visit after the first visit. For these patients, we will remove their record for the first visit and set their second record as the first visit, so 28 rows are expected to be removed from the data. Next, we need to pull out the remaining 6 individuals' records to see their missing patterns.

```r
data2 <- data1 %>%
  filter(!(ptid %in% rm_1_only$ptid & visit == 1)) %>%
  mutate(visit = ifelse(ptid %in% rm_1_only$ptid, visit - 1, visit))
  # decrease each visit by 1 for patients' id in rm_1_only
cat(paste(nrow(data1) - nrow(data2), "records were removed."))
```

```
## 28 records were removed.
```

```
rm_1_remaining <- rm_1 %>%
  filter(!(ptid %in% rm_1_only$ptid)) %>%
  mutate(n_measured = total - n)
rm_1_remaining
```

```
## # A tibble: 6 x 5
##   ptid       visit_start     n total n_measured
##   <chr>            <int> <int> <int>      <int>
## 1 COLE010634           1     6     9          3
## 2 CRAG031833           1     3     4          1
## 3 GRE2111134           1     3     9          6
## 4 LICR122828           1     2     3          1
## 5 LOMB050746           1     2     9          7
## 6 THOJ091031           1     2     5          3
```

Observations and thinking

- Among the 6 individuals, 2 of them had root size measurement only once during their three to four visits. So, we will exclude all records pertaining to these two patients, resulting in the removal of 7 rows.
- Among patients who had 9 total visits, one of them had their root size being measured for three times, while the other two of them had six to seven times of root size measurements. Also, the last patient who had 3 times of root size measurement out of 5 total visits. For these four individuals, we will only retain those records that has some values in the `root` column, so 13 rows are expected to be removed.

```
remove_all <- rm_1_remaining %>%
  filter(n_measured == 1) %>%
  select(ptid) %>%
  pull()

remove_partial <- rm_1_remaining %>%
  select(ptid) %>%
  filter(!(ptid %in% remove_all)) %>%
  pull()

data3 <- data2 %>%
  filter(!(ptid %in% remove_all)) %>%  # 7 records are expected to be removed
  filter(!(ptid %in% remove_partial & is.na(root))) %>%  # 13 records are expected to be removed
  group_by(ptid) %>%
  mutate(visit = ifelse(ptid %in% remove_partial, row_number(), visit),
         total_visit = n()) %>%
  ungroup()
```

Patients whose root size was measured at first visit only.

```
data3 %>%
  filter(is.na(root)) %>% # find those with missing root size
  select(ptid, age, sex, bav_confirmed, visit, total_visit) %>%
```

```
  group_by(ptid) %>%
  summarise(visit_start = min(visit), # the first visit that root size appear to be empty
            n = n(), total = max(total_visit)) %>%
  filter(n == total) %>%
  nrow()
```

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

The above result indicates that every patient who had root size measurement at their first visit must had root size measurement at their follow-up visits.

Patients whose root size was not measured at their last visit only.

```
root_na <- data3 %>%
  filter(is.na(root)) %>% # find those with missing root size
  select(ptid, age, sex, bav_confirmed, visit, total_visit) %>%
  group_by(ptid) %>%
  summarise(visit_start = min(visit), # the first visit that root size appear to be empty
            n = n(), total = max(total_visit))

root_na %>%
  filter(n == 1 & visit_start == total) %>%
  head()
```

```
## # A tibble: 6 x 4
##   ptid       visit_start     n total
##   <chr>            <dbl> <int> <int>
## 1 AYOC083067           2     1     2
## 2 BAEC032439           2     1     2
## 3 BIDR012533           2     1     2
## 4 CASC010227           8     1     8
## 5 CHAR112653           7     1     7
## 6 CING082330           6     1     6
```

Case 1: Patients who had only two visits.
* Remove all records for them.
Case2: Patients who had more than two visits.
* Remove their record corresponding to their last visit.

```
remove_all <- root_na %>%
  filter(n == 1 & visit_start == total) %>%
  filter(visit_start == 2) %>%
  select(ptid) %>%
  pull()

remove_last <- root_na %>%
  filter(n == 1 & visit_start == total) %>%
  filter(visit_start > 2) %>%
  select(ptid) %>%
  pull()
```

```
data4 <- data3 %>%
  filter(!(ptid %in% remove_all)) %>%
  filter(!(ptid %in% remove_last & is.na(root)))

nrow(data3) - nrow(data4)
```

```
## [1] 24
```

24 records were removed.

Patients whose root size was measured more than two times.

These patients all had root size initial measurement at their first visit.

Case 1: Only one of the visit after the initial visit had no root size measurement.
* Remove the record that has the root column empty.
* Recount their visit times.

```
to_remove <- data4 %>%
  filter(is.na(root)) %>% # find those with missing root size
  select(ptid, age, sex, bav_confirmed, visit, total_visit) %>%
  group_by(ptid) %>%
  summarise(visit_start = min(visit), # the first visit that root size appear to be empty
            n = n(), total = max(total_visit)) %>%
  filter(n == 1) %>%
  select(ptid) %>%
  pull()

cat(paste(length(to_remove), "records are expected for removal."))
```

```
## 17 records are expected for removal.
```

```
data5 <- data4 %>%
  filter(!(ptid %in% to_remove & is.na(root))) %>%
  group_by(ptid) %>%
  mutate(visit = ifelse(ptid %in% to_remove, row_number(), visit),
         total_visit = n()) %>%
  ungroup()

cat(paste(nrow(data4) - nrow(data5),
          "records were removed."))
```

```
## 17 records were removed.
```

Patients who had no root size measurement at the last few consecutive visits

Remove records that had no root size measurement.

```
rm_end <- data5 %>%
  filter(is.na(root)) %>% # find those with missing root size
  select(ptid, age, sex, bav_confirmed, visit, total_visit) %>%
  group_by(ptid) %>%
  summarise(visit_start = min(visit), # the first visit that root size appear to be empty
            n = n(), total = max(total_visit)) %>%
  filter(n == total-visit_start+1)

cat(paste(sum(rm_end$n), "records are expected to be removed. "))
```

```
## 9 records are expected to be removed.
```

```
data6 <- data5 %>%
  filter(!(ptid %in% rm_end$ptid & is.na(root)))

cat(paste(nrow(data5) - nrow(data6),
          "records were removed."))
```

```
## 9 records were removed.
```

Root size was not measured multiple times at visits between the initial and last

Remove all records without root size and update visit count.

```
rest <- data6 %>%
  filter(is.na(root)) %>% # find those with missing root size
  select(ptid, visit, total_visit) %>%
  group_by(ptid)

head(rest, 10)
```

```
## # A tibble: 10 x 3
## # Groups:   ptid [1]
##    ptid       visit total_visit
##    <chr>      <dbl>       <int>
##  1 AREA112454     4          18
##  2 AREA112454     5          18
##  3 AREA112454     8          18
##  4 AREA112454     9          18
##  5 AREA112454    10          18
##  6 AREA112454    12          18
##  7 AREA112454    13          18
##  8 AREA112454    14          18
##  9 AREA112454    15          18
## 10 AREA112454    16          18
```

```
cat(paste(nrow(rest), "records are expected to be removed. "))
```

```
## 43 records are expected to be removed.
```

```
data7 <- data6 %>%
  filter(!(ptid %in% rest$ptid & is.na(root))) %>%
  group_by(ptid) %>%
  mutate(visit = ifelse(ptid %in% rest$ptid, row_number(), visit),
         total_visit = n()) %>%
  ungroup()

cat(paste(nrow(data6) - nrow(data7),
          "records were removed."))
```

```
## 43 records were removed.
```

Check whether all records without root size information were removed successfully

```
data7 %>%
  filter(is.na(root)) %>%
  nrow()
```

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

Create the binary outcome column

The outcome = 1 if
a. the absolute value of aortic root size (**The last visit?**) is greater than 4.5cm or
b. the growth is greater than 5mm over time.

Check values in root column

```
data7 %>%
  filter(visit == 1) %>%
  group_by(bav_confirmed) %>%
  summarise(n = n_distinct(ptid),
            mean = round(mean(root), 2), median = median(root),
            sd = sd(root), iqr = IQR(root),
            max = max(root), min = min(root))
```

```
## # A tibble: 2 x 8
##   bav_confirmed     n  mean median    sd   iqr   max   min
##   <fct>         <int> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 TAV              69  35.0     34  8.61     8    67 20
## 2 BAV             165  35.7     36  6.80     7    63  3.65
```

Observations:

- The number of BAV patients is over two times of the number of patients with TAV.

- Exposure groups had similar mean, standard deviation and interquartile range, suggesting that the baseline measurement achieved balance between the two exposure groups.

- The minimum root size from the BAV group is only one fifth of the minimum value from the TAV group.

- According to the cutoff criteria, the unit of root size in our data should be mm.

```
base_root <- data7 %>%
  select(ptid, root, visit) %>%
  group_by(ptid) %>%
  filter(visit == 1) %>%
  rename(baseroot = root) %>%
  select(ptid, baseroot)

last_root <- data7 %>%
  select(ptid, root, visit, total_visit) %>%
  group_by(ptid) %>%
  filter(visit == total_visit) %>%
  rename(lastroot = root) %>%
  select(ptid, lastroot)

outcome_df <- data7 %>%
  left_join(base_root, by = "ptid") %>%
  left_join(last_root, by = "ptid") %>%
  mutate(rt_growth = abs(lastroot - baseroot)) %>%
  mutate(outcome = ifelse(lastroot > 45 | rt_growth > 5,  1, 0)) %>%
  mutate(outcome = factor(outcome)) %>%
  relocate(lastroot, .after = root) %>%
  relocate(rt_growth, .after = lastroot) %>%
  select(-baseroot)

outcome_df %>%
  filter(is.na(outcome))
```

```
## # A tibble: 0 x 30
## # i 30 variables: ptid <chr>, age <dbl>, sex <chr>, bav_confirmed <fct>,
## #   nc_sinus <chr>, raa_type <chr>, diabetes <chr>, hyper <chr>, chlstrl <chr>,
## #   bsa <dbl>, died <dbl>, visit <dbl>, lka_d <chr>, yr2death <dbl>,
## #   ao_reop <dbl>, yr2ao_reop <dbl>, bsa_baseline <dbl>, dateor <chr>,
## #   mdate <chr>, root <dbl>, lastroot <dbl>, rt_growth <dbl>, src_root <dbl>,
## #   src_aa <dbl>, src_arch <dbl>, day2vst <dbl>, yr2vst <dbl>, date_base <chr>,
## #   total_visit <int>, outcome <fct>
```

Tidy the data

The following steps were implemented in the next chunk:

- Remove all patient who only visited once
- Convert columns of date to Date format.

- Locate date columns together.

- Set “Survived” as the reference level for death.
- Rename `bav_confirmed` as `exposure`.
- Factor `ao_reop`
- Remove columns that are not useful for now.
- Create the binary column by the cutoff.
  - outcome = 1 if

```
data8 <- outcome_df %>%
  filter(!(total_visit == 1)) %>%  # 9 patients were removed
  mutate(lka_d = as.Date(lka_d, format = "%d%b%Y")) %>%
  mutate(across(c(dateor, mdate, date_base),
                ~as.Date(., format = "%m/%d/%Y"))) %>%
  relocate(lka_d, .before = date_base) %>%
  relocate(dateor, .after = lka_d) %>%
  relocate(mdate, .after = dateor) %>%
  mutate(died = factor(died, labels = c("Survived", "Died"))) %>%
  rename(Exposure = bav_confirmed) %>%
  mutate(ao_reop = factor(ao_reop, levels = c(0, 1, 3),
                          labels = c("Alive",
                                     "Reoperation on aorta",
                                     "Death"))) %>%
  select(-c(src_arch, src_aa, src_arch, src_root))
```

Check values in categorical columns

```
cat_cols <- data8[, sapply(data8,
                  function(x)is.factor(x) || is.character(x))] %>%
  select(-ptid)

lapply(cat_cols, table)
```

```
## $sex
##
## female   male
##    328    704
##
## $Exposure
##
## TAV BAV
## 296 736
##
## $nc_sinus
##
##  NO YES
## 654 378
##
## $raa_type
##
```

```
## 0_no_replacement            clamp         hemiarch
##               20              620              392
##
## $diabetes
##
## Diabetes Negative
##       98      934
##
## $hyper
##
## Hypertension     Negative
##          567          465
##
## $chlstrl
##
##                hyperlipidemia       Negative
##              4            465            563
##
## $died
##
## Survived     Died
##      870      162
##
## $ao_reop
##
##                Alive Reoperation on aorta                Death
##                  849                   27                    0
##
## $outcome
##
##   0   1
## 699 333
```

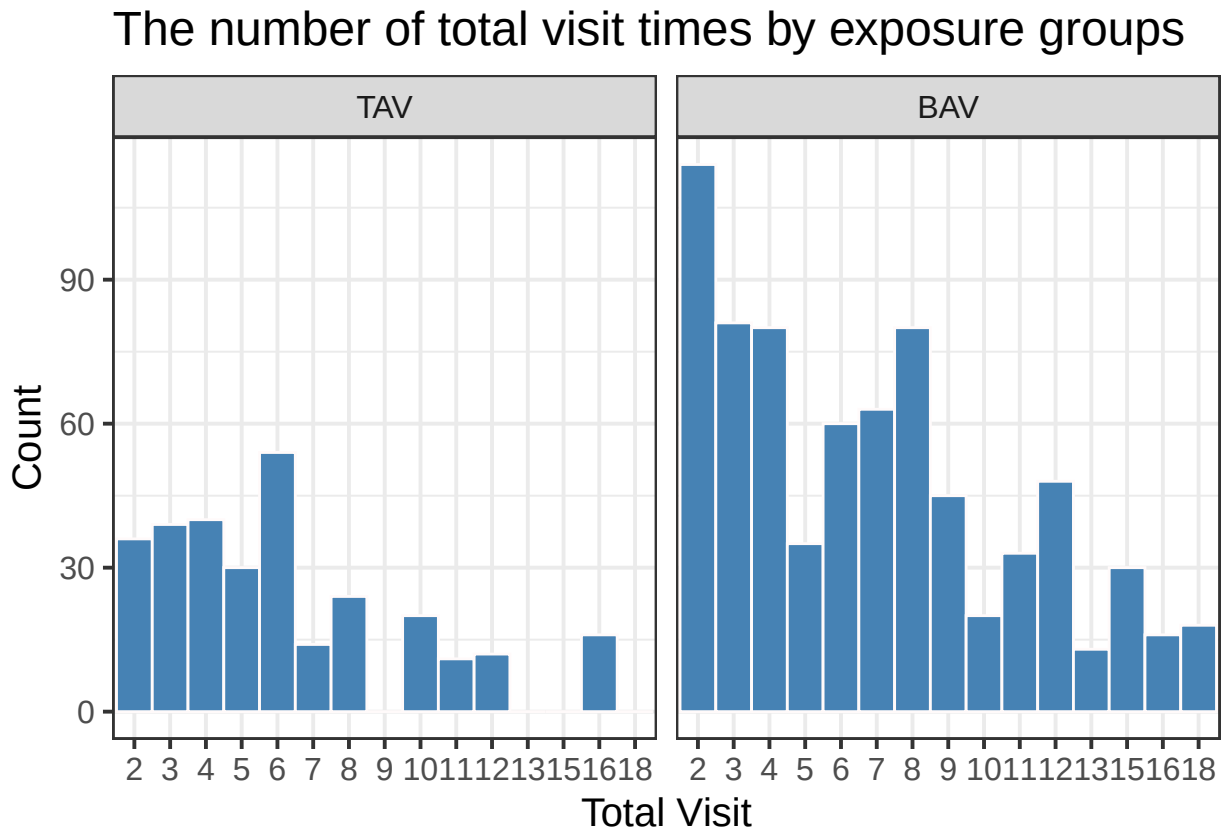
Clean `raa_type`, `diabetes`, `hypertension`, `chlstrl`.

```
library(stringr)
clean_data <- data8 %>%
  mutate(raa_type = ifelse(raa_type == "0_no_replacement",
                           "No replacement", raa_type),
         diabetes = ifelse(diabetes == "Diabetes", "Positive", diabetes),
         hyper = ifelse(hyper=="Hypertension", "Positive", hyper),
         chlstrl = ifelse(chlstrl=="", "Unknown", chlstrl)) %>%
  mutate_at(vars(sex, nc_sinus, raa_type, chlstrl), ~str_to_sentence(.))
```

EDA

Total Visit

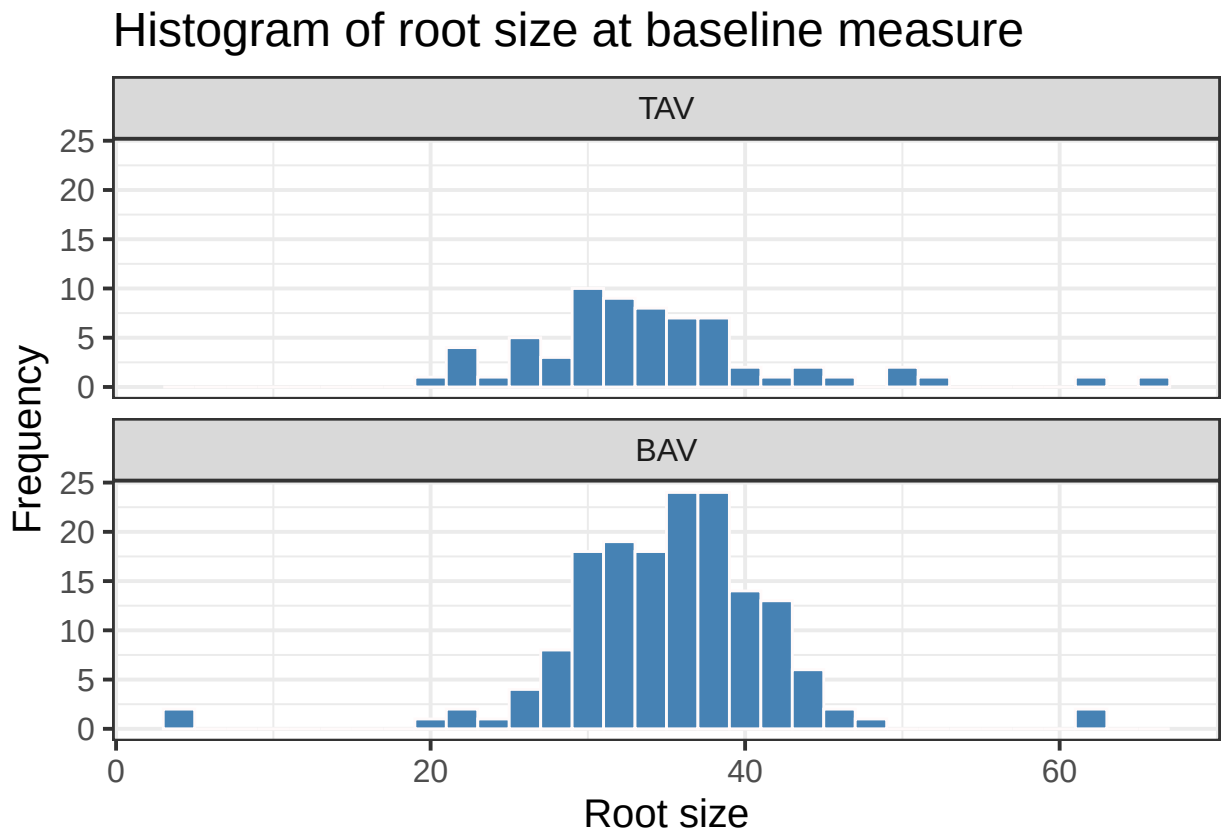
```r
clean_data %>%
  group_by(Exposure) %>%
  select(total_visit) %>%
  table() %>%
  as.data.frame() %>%
  ggplot(aes(x = total_visit, y = Freq))+
  geom_col(width = 1, fill = "steelblue", color = "snow1")+
  facet_wrap(~Exposure)+
  labs(title = "The number of total visit times by exposure groups",
       y = "Count", x = "Total Visit")
```



Baseline measurement at initial visit

Continuous outcome

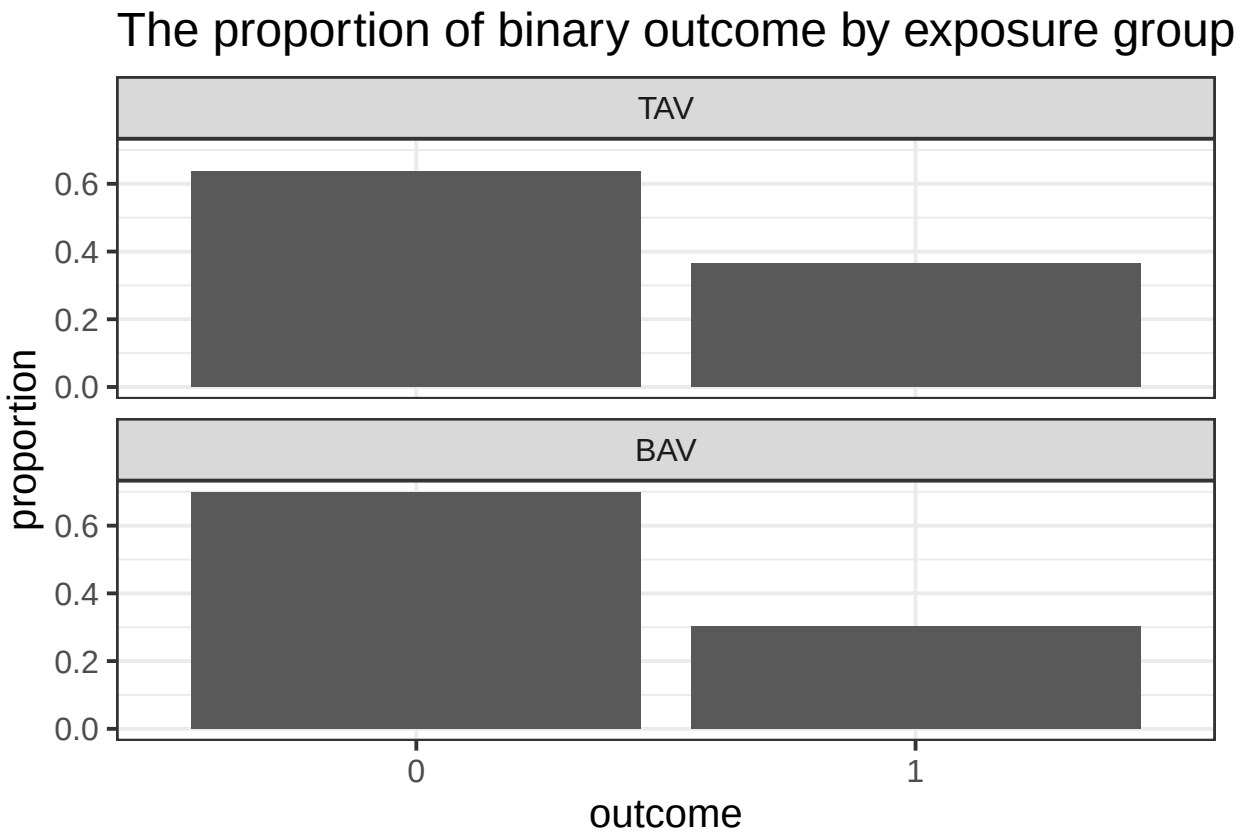
```r
clean_data %>%
  filter(visit == 1) %>%
  ggplot(aes(x = root))+
  geom_histogram(binwidth = 2, fill = "steelblue", color = "snow1")+
  facet_wrap(~Exposure, nrow = 2)+
  labs(x = "Root size", y = "Frequency",
       title = "Histogram of root size at baseline measure")
```



The end of treatment: Binary outcome

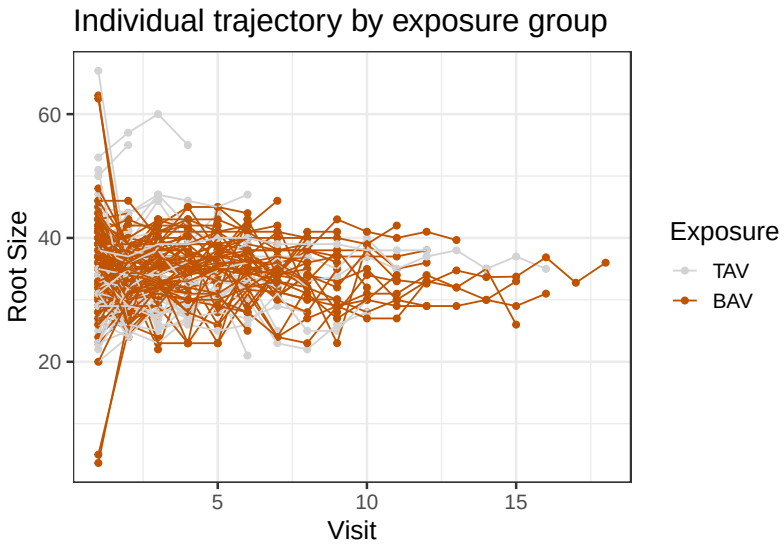
```
group_total <- clean_data %>%
  filter(visit == total_visit) %>%
  group_by(Exposure) %>%
  summarise(N=n()) %>%
  select(N) %>%
  pull()


clean_data %>%
  filter(visit == total_visit) %>%
  group_by(Exposure, outcome) %>%
  summarise(n = n()) %>%
  mutate(proportion = ifelse(Exposure == "TAV",
                             n/group_total[1], n/group_total[2])) %>%
  ggplot()+
  geom_col(aes(x = outcome, y = proportion))+
  facet_wrap(~Exposure, nrow = 2)+
  labs(title = "The proportion of binary outcome by exposure groups")
```

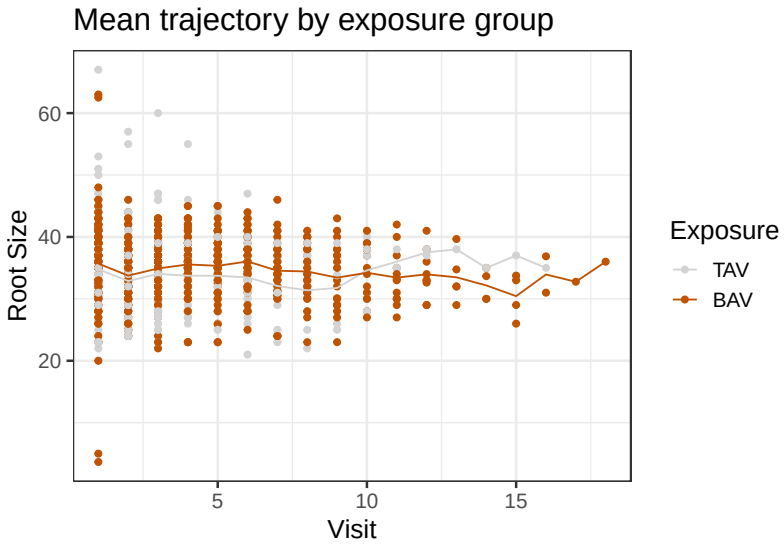


Mean trajectory by exposure group

```
clean_data %>%
  ggplot(aes(y = root, x= visit, color = Exposure)) +
  geom_point(size = 1.5) +
  geom_line(aes(group = ptid))+
  labs(title = "Individual trajectory by exposure group",
       y = "Root Size", x = "Visit")+
  scale_color_manual(values = c("TAV" = "lightgrey", "BAV" = "#bf5300"))
```




```
clean_data %>%
  ggplot(aes(y = root, x= visit, color = Exposure)) +
  geom_point(size = 1.5) +
  stat_summary(aes(group = Exposure), fun = mean, geom="line") +
  labs(title = "Mean trajectory by exposure group",
       y = "Root Size", x = "Visit")+
  scale_color_manual(values = c("TAV" = "lightgrey", "BAV" = "#bf5300"))
```



Propensity Score Matching

```
match_df <- clean_data %>%
  group_by(ptid) %>%
  slice(1) %>%
  filter(yr2vst == 0)
```

Pre-analysis using non-matched data

Difference-in-means: Root Size

```
covs <- c("age", "sex", "bsa_baseline")

match_df %>%
  group_by(Exposure) %>%
  summarise(N = n(),
            mean_rs = round(mean(root),2),
            std_error = round(sd(root) / sqrt(N), 2)) %>%
  rename(`Mean Root Size` = mean_rs,
         `Standard Error` = std_error) %>%
  cbind(match_df %>%
          group_by(Exposure) %>%
          select(one_of(covs)) %>%
          summarise_all(funs(mean(., na.rm = T))) %>%
          mutate(age = round(age, 2),
                 bsa_baseline = round(bsa_baseline, 2)) %>%
          select(-Exposure, -sex) %>%
          rename(Age = age, `Baseline BSA` = bsa_baseline))
```

```
##   Exposure  N Mean Root Size Standard Error   Age Baseline BSA
## 1      TAV 22          34.30           1.63 71.27         1.77
## 2      BAV 80          35.82           0.77 61.20         1.98
```

```
with(match_df, t.test(root ~ Exposure))
```

```
##
##  Welch Two Sample t-test
##
## data:  root by Exposure
## t = -0.84696, df = 31.031, p-value = 0.4035
## alternative hypothesis: true difference in means between group TAV and group BAV is not equal to 0
## 95 percent confidence interval:
##  -5.206195  2.150854
## sample estimates:
## mean in group TAV mean in group BAV
##          34.29545          35.82312
```

The difference-in-means is **NOT statistically significant** at the level of 0.05.

Match patients who had their first measurement on the operation date and had at least 2 measurement records by exposure groups.

```
library(optmatch)
pps <- glm(Exposure ~ age + sex + bsa_baseline, family = binomial(), data = match_df)
summary(pps)
```

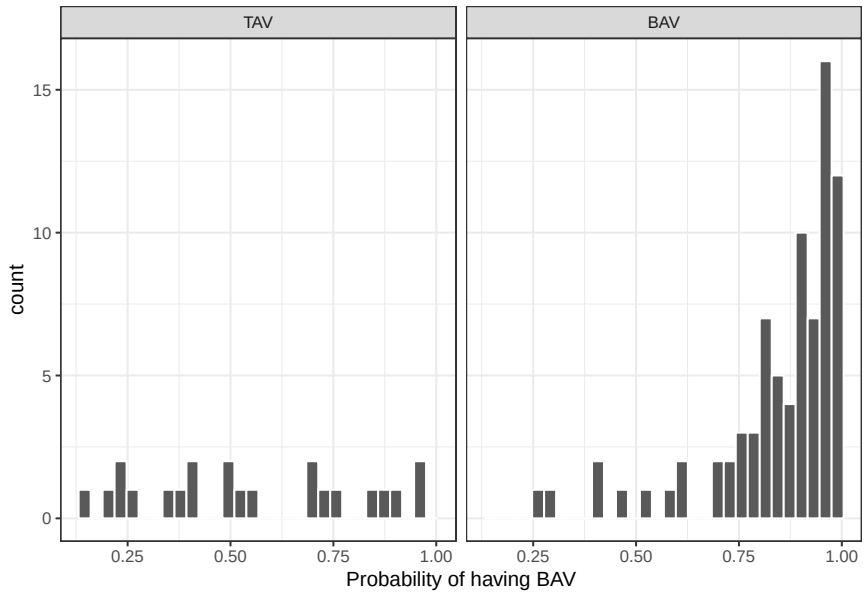
```
##
## Call:
```

```
## glm(formula = Exposure ~ age + sex + bsa_baseline, family = binomial(),
##     data = match_df)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.30119    3.42093  -0.088   0.9298
## age          -0.07132    0.02863  -2.491   0.0127 *
## sexMale       1.24221    0.63447   1.958   0.0502 .
## bsa_baseline  2.98148    1.52156   1.959   0.0501 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 106.364  on 101  degrees of freedom
## Residual deviance:  77.696  on  98  degrees of freedom
## AIC: 85.696
##
## Number of Fisher Scoring iterations: 5
```

```
pps_df <- data.frame(score = predict(pps, type = "response"),
                     Exposure = pps$model$Exposure)
head(pps_df)
```

```
##       score Exposure
## 1 0.9797893      BAV
## 2 0.9716684      BAV
## 3 0.9866162      BAV
## 4 0.7835908      BAV
## 5 0.9103047      BAV
## 6 0.7094509      TAV
```

```
pps_df %>%
  ggplot(aes(x = score)) +
  geom_histogram(color = "white") +
  facet_wrap(~Exposure) +
  xlab("Probability of having BAV") +
  theme_bw()
```




```r
# install.packages("MatchIt")
library(MatchIt)
mod_match <- matchit(Exposure ~ age + sex + bsa_baseline,
                     method = "nearest", data = match_df)
summary(mod_match) # 22 matched pairs
```

```
##
## Call:
## matchit(formula = Exposure ~ age + sex + bsa_baseline, data = match_df,
##     method = "nearest")
##
## Summary of Balance for All Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.8484        0.5514          1.8194     0.3910    0.3340
## age                61.2000       71.2727         -0.7988     1.6415    0.2032
## sexFemale           0.2000        0.6364         -1.0909          .    0.4364
## sexMale             0.8000        0.3636          1.0909          .    0.4364
## bsa_baseline        1.9843        1.7709          1.1136     0.6397    0.2605
##              eCDF Max
## distance       0.5727
## age            0.3727
## sexFemale      0.4364
## sexMale        0.4364
## bsa_baseline   0.4943
##
## Summary of Balance for Matched Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.9766        0.5514          2.6053     0.0012    0.6511
## age                48.5455       71.2727         -1.8023     1.1411    0.4545
## sexFemale           0.0000        0.6364         -1.5909          .    0.6364
## sexMale             1.0000        0.3636          1.5909          .    0.6364
## bsa_baseline        2.1255        1.7709          1.8507     0.3333    0.4339
##              eCDF Max Std. Pair Dist.
```

```
## distance       1.0000          2.6053
## age            0.7727          1.8023
## sexFemale      0.6364          1.5909
## sexMale        0.6364          1.5909
## bsa_baseline   0.7727          2.2303
##
## Sample Sizes:
##           Control Treated
## All            22      80
## Matched        22      22
## Unmatched       0      58
## Discarded       0       0
```

```r
# create a dataframe containing only the matched pairs
matched_pairs <- match.data(mod_match)  %>% # column "distance" represent the propensity score
  mutate(sex = ifelse(sex == "female", 1, 0))
dim(matched_pairs)
```

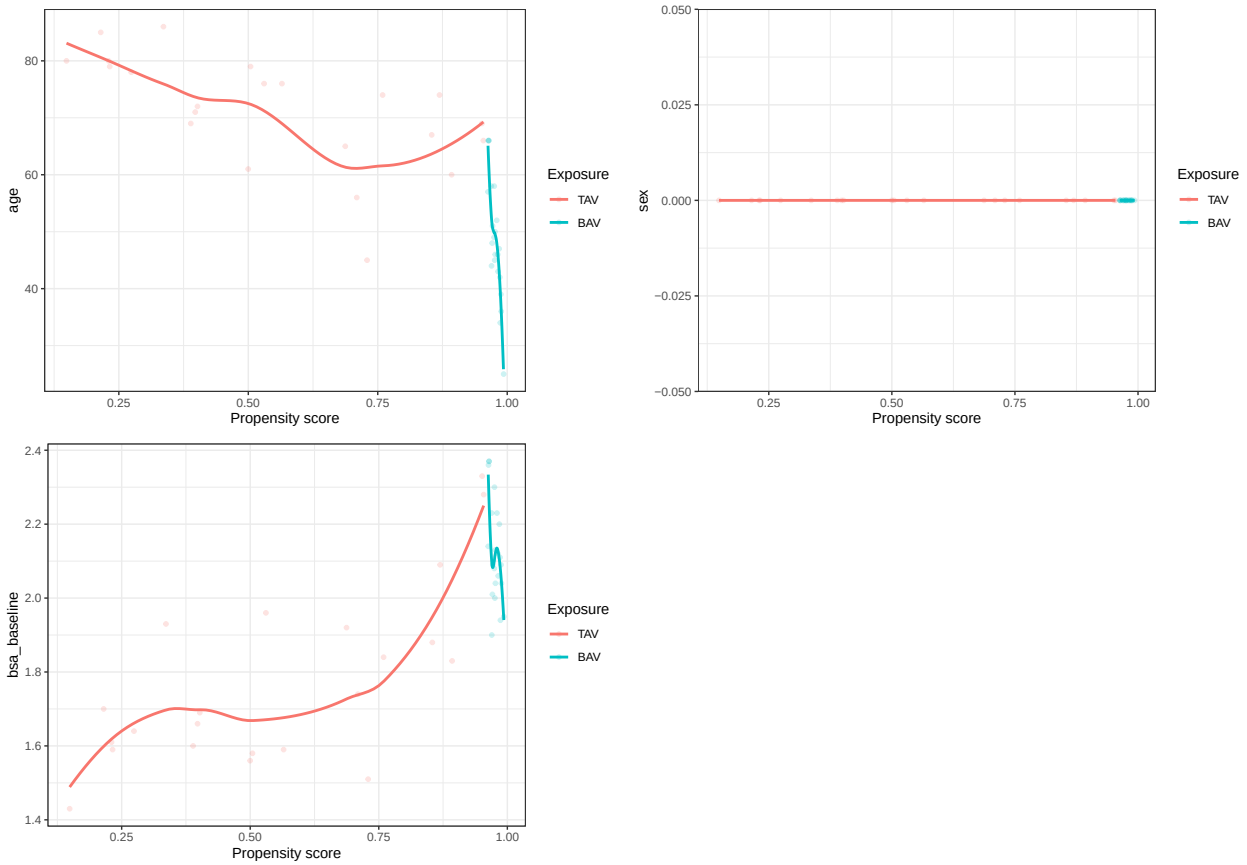
```
## [1] 44 30
```

There are 44 records in the matched pair dataframe, meaning that we had 22 matched pairs.

Examining covariate balance in the matched sample

```r
balance_check <- function(data, variable) {
  data$variable <- data[[variable]]
  data$Exposure <- as.factor(data$Exposure)
  support <- c(min(data$variable), max(data$variable))
  ggplot(data, aes(x = distance, y = variable, color = Exposure)) +
    geom_point(alpha = 0.2, size = 1.3) +
    geom_smooth(method = "loess", se = F) +
    xlab("Propensity score") +
    ylab(variable) +
    theme_bw() +
    ylim(support)
}


balance_check(matched_pairs, "age")
balance_check(matched_pairs, "sex")
balance_check(matched_pairs, "bsa_baseline")
```



Difference in Means

```
matched_pairs %>%
  group_by(Exposure) %>%
  summarise(N = n(),
            mean_rs = mean(root),
            std_error = sd(root) / sqrt(N)) %>%
  rename(`Mean Root Size` = mean_rs,
         `Standard Error` = std_error) %>%
  cbind(match_df %>%
          group_by(Exposure) %>%
          select(one_of(covs)) %>%
          summarise_all(funs(mean(., na.rm = T))) %>%
          select(-c(sex, Exposure)) %>%
          rename(Age = age, `Baseline BSA` = bsa_baseline))
```

```
##   Exposure  N Mean Root Size Standard Error      Age Baseline BSA
## 1      TAV 22       34.29545      1.6305588 71.27273     1.770909
## 2      BAV 22       38.51818      0.9291489 61.20000     1.984250
```

Test the difference using t-test with the null hypothesis that there is no difference between exposure groups.

```r
lapply(c("age", "bsa_baseline", "root"), function(v) {
    t.test(matched_pairs[[v]] ~ matched_pairs$Exposure)
})
```

```
## [[1]]
##
##  Welch Two Sample t-test
##
## data:  matched_pairs[[v]] by matched_pairs$Exposure
## t = 7.4017, df = 41.818, p-value = 3.994e-09
## alternative hypothesis: true difference in means between group TAV and group BAV is not equal to 0
## 95 percent confidence interval:
##  16.52989 28.92466
## sample estimates:
## mean in group TAV mean in group BAV
##          71.27273          48.54545
##
##
## [[2]]
##
##  Welch Two Sample t-test
##
## data:  matched_pairs[[v]] by matched_pairs$Exposure
## t = -6.0128, df = 33.599, p-value = 8.673e-07
## alternative hypothesis: true difference in means between group TAV and group BAV is not equal to 0
## 95 percent confidence interval:
##  -0.474430 -0.234661
## sample estimates:
## mean in group TAV mean in group BAV
##          1.770909          2.125455
##
##
## [[3]]
##
##  Welch Two Sample t-test
##
## data:  matched_pairs[[v]] by matched_pairs$Exposure
## t = -2.2501, df = 33.337, p-value = 0.03116
## alternative hypothesis: true difference in means between group TAV and group BAV is not equal to 0
## 95 percent confidence interval:
##  -8.0394559 -0.4059987
## sample estimates:
## mean in group TAV mean in group BAV
##          34.29545          38.51818
```

The difference in means **after matching are statistically significant** between exposure groups, meaning that we need to reject the null hypothesis of no difference.

TableOne For Matched Pairs

```r
library(tableone)
tb1_data <- clean_data %>%
  select(-c(lka_d, dateor, mdate, date_base, lastroot)) %>%
  mutate(total_visit = factor(total_visit))

# Vector of variables to summarize
myVars <- names(tb1_data) %>% as.array()

# Vector of categorical variables that need transformation
catVars <- c('sex', 'bav_confirmed', 'nc_sinus', 'raa_type',
             'diabetes', 'hyper', 'chlstrl', 'died', 'ao_reop',
             "total_visit", "outcome")

table1 <- CreateTableOne(vars = myVars, strata = 'Exposure',
                         data = tb1_data, factorVars = catVars)

table1_df <- as.data.frame(print(table1, showAllLevels = TRUE, smd=TRUE)) %>%
  select(-test) %>% filter(!(level %in% c("BAV", "TAV"))) %>%
  mutate_at(vars(level), ~str_replace(., "^[0]_", "")) %>%
  mutate(level = str_replace_all(str_to_sentence(level), "_", " ")) %>%
  mutate(level = str_to_sentence(level))
```

Questions

1. When the number of visits are not consistent across group and individuals, how to construct covariance matrix?

2. How to communicate data for EDA?