# Technical Exercise

Dennis Valentine

2024-01-11

```
md <- read.csv(file = "data/MedicalDictionary.csv")


patient <- read.csv(file = "data/Patient.csv") %>%
  select(-X) %>% # cleaning data
  filter(acceptable == 1) %>% # QC
  unique() %>% #remove duplicate rows
  mutate(regstartdate = as.Date(regstartdate), regenddate = as.Date(regenddate )) %>%
  filter(regstartdate >= as.Date("1945-01-01") & regstartdate <= as.Date("2023-01-01")) %>%   # removin
  filter( (regenddate >= as.Date("1945-01-01") & regenddate <= as.Date("2023-01-01")) | is.na(regenddat
  filter( ! (regenddate <=  regstartdate) | is.na(regenddate)) %>%
  group_by(patid) %>%
  add_count()

patient %>%
  arrange(patid) %>%
  filter(n > 1) # YOB is different, gender is different -- this causes problems later on. At this point
```

```
## # A tibble: 47 x 8
## # Groups:   patid [23]
##    patid  pracid gender   yob regstartdate regenddate acceptable     n
##    <chr>   <int>  <int> <int> <date>       <date>          <int> <int>
##  1 2novgj      5      2  1927 1995-02-15   NA                  1     2
##  2 2novgj      5      2  1905 1995-02-15   NA                  1     2
##  3 3Uo7ad      2      2  1939 1971-03-18   NA                  1     2
##  4 3Uo7ad      2      3  1939 1971-03-18   NA                  1     2
##  5 3l4V31      1      1  1984 2009-10-04   2022-12-21          1     2
##  6 3l4V31      1      3  1984 2009-10-04   2022-12-21          1     2
##  7 6U3z90      1      2  1995 2013-01-01   NA                  1     2
##  8 6U3z90      1      2  1983 2013-01-01   NA                  1     2
##  9 6b7BbE      3      2  1971 2003-12-20   NA                  1     2
## 10 6b7BbE      3      2  1987 2003-12-20   NA                  1     2
## # i 37 more rows
```

```
obvs <- read.csv(file = "data/Observation.csv") %>%
  select(-X) %>% # medcodeid links to the dictionary
  mutate(enterdate   = as.Date(enterdate)) %>%
  filter(enterdate >= as.Date("1945-01-01") & enterdate <= as.Date("2023-01-01"))   # removing mistakes
# sum(obvs$obsid == "") # checking the data
# sum(is.na(obvs$obsid))
```

```
problem <- read.csv(file = "data/Problem.csv")
problem %>% head()
```

```
##    patid      obsid pracid probstatusid
## 1 pRU7Ac iyitr4xt3w      1           NA
## 2 pRU7Ac tb8wrof3hn      1           NA
## 3 pRU7Ac 6ag35gwpvu      1           NA
## 4 pRU7Ac n6lsususnr      1            4
## 5 pRU7Ac 4z6yc59xkb      1           NA
## 6 pRU7Ac 2rjsz4fxdf      1           NA
```

```
# Consultation.csv
cons <- read.csv(file = "data/Consultation.csv") %>%
  unique()
cons %>% head()
```

```
##    patid       consid pracid   consdate  enterdate
## 1 L63MBY PoWOR7uxwUCm      2 2005-11-03 2005-11-09
## 2 NyVtqH h09V9pxx4WLi      2 2021-11-27 2021-11-27
## 3 Pi4M3Y 6DpqDB8Wlrwg      1 2008-02-17 2008-02-24
## 4 rNQp7i 9mmfc43njihC      4 1974-09-06 1974-09-07
## 5 Pz3mEl ODLCWbdLBN3v      5 2016-06-15 2016-06-25
## 6 uXpwGz NIEZTVRrrxYQ      4 1985-05-18 1985-05-31
```

```
# Practice.csv
practice <- read.csv(file = "data/Practice.csv")
practice
```

```
##   pracid     region
## 1      1 North-East
## 2      2 North-West
## 3      3   Midlands
## 4      4 South East
## 5      5 South West
```

## Joins

Join across the tables to generate a table or dataframe with the following information. In comments, explain how you dealt with any inconsistencies in the data.

```
p1 <- patient %>% # I've already cleaned up the data as I was exporing it.
  transmute(patid, age_at_2023 = 2023 - as.integer(yob), gender, regstartdate,  regenddate) %>%
  unique() # try to remove multiple patid


df <- left_join(x = p1, y = cons %>% select(patid:pracid), by = "patid") %>% # 1: many mapping, -- gend
  left_join(y =  obvs, by = c("patid", "consid", "pracid")) %>%  # obvs[30817,] == FyHqc6 person ID cau
  select(-obsdate ) %>% # this is subjective - people might not recall correctly, skewing answers. But
  left_join(y = md, by = c("medcodeid" = "aurum_code")) %>% # dates causing issues -- just take the min
   # not asked for term but this is one of those times I'll do more because from experience, clients do
  select(-pracid)
```

```
## Warning in left_join(x = p1, y = cons %>% select(patid:pracid), by = "patid"): Detected an unexpecte
## i Row 1 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 1216 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

```
## Warning in left_join(., y = obvs, by = c("patid", "consid", "pracid")): Detected an unexpected many-
## i Row 1 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 30817 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

```
df %>% summary() # someone is aged 118 in 2023??
```

```
##     patid             age_at_2023         gender       regstartdate
##  Length:39335       Min.   : 23.00   Min.   :1.000   Min.   :1946-01-15
##  Class :character   1st Qu.: 42.00   1st Qu.:1.000   1st Qu.:1993-01-25
##  Mode  :character   Median : 63.00   Median :2.000   Median :2005-05-11
##                     Mean   : 61.46   Mean   :1.584   Mean   :2002-02-07
##                     3rd Qu.: 80.00   3rd Qu.:2.000   3rd Qu.:2015-04-12
##                     Max.   :118.00   Max.   :3.000   Max.   :2022-12-16
##
##    regenddate            consid             obsid
##  Min.   :1966-02-26   Length:39335       Length:39335
##  1st Qu.:2009-01-29   Class :character   Class :character
##  Median :2015-07-25   Mode  :character   Mode  :character
##  Mean   :2011-11-09
##  3rd Qu.:2020-01-25
##  Max.   :2022-12-22
##  NA's   :31902
##    enterdate           medcodeid              Term
##  Min.   :1950-08-26   Length:39335       Length:39335
##  1st Qu.:2006-08-05   Class :character   Class :character
##  Median :2015-06-14   Mode  :character   Mode  :character
##  Mean   :2011-05-28
##  3rd Qu.:2020-05-12
##  Max.   :2023-01-01
##  NA's   :530
```

```
# In a database as big as CPRD anything that can go wrong would have, at some level, gone wrong. I thin
```

## Counts

Number of consultations for each patient (save as .csv, and print number for patient 02A27z)

```
cons_per_patient <- df %>%
  filter(!is.na(consid)) %>%
  count(patid, consid) %>%
  add_count(patid)  %>%
  distinct(patid, n_consultations = nn)
```

```
## Storing counts in ‘nn‘, as ‘n‘ already present in input
## i Use ‘name = "new_name"‘ to pick a new name.
```

```
write.csv(x = cons_per_patient, file = "data/cons_per_patient.csv", row.names = FALSE)

cons_per_patient %>%
  filter(patid == "02A27z")
```

```
## # A tibble: 1 x 2
## # Groups:   patid [1]
##   patid  n_consultations
##   <chr>            <int>
## 1 02A27z              30
```

Number of observations for each patient (save as .csv, and print number for patient 02A27z)

```
obvs_per_patient <- df %>%
  filter(!is.na(medcodeid )) %>%
  count(patid, medcodeid  ) %>%
  add_count(patid)  %>%
  distinct(patid, n_obvs = nn)
```

```
## Storing counts in ‘nn‘, as ‘n‘ already present in input
## i Use ‘name = "new_name"‘ to pick a new name.
```

```
write.csv(x = obvs_per_patient, file = "data/obvs_per_patient.csv", row.names = FALSE)

df %>%
  filter(patid == "02A27z") %>%
  filter(!is.na(medcodeid )) %>%
  count(medcodeid) %>%
  nrow()
```

```
## [1] 91
```

Mean number of observations per consultation

```
n_obvs_per_con <- df %>%
  ungroup() %>%
  filter(!is.na(medcodeid) & !is.na(consid)) %>%
  distinct(consid, medcodeid) %>%
  group_by(consid) %>%
  add_count(name = "n_consid") %>%
  distinct(consid, n_consid) %>%
  ungroup() %>%
  summarise(mean_obvs = mean(n_consid))
# mean = 3.46 observations (i.e. 4 observations per consultation)
```

Please also display number of consultations and observations per patient as a histogram.

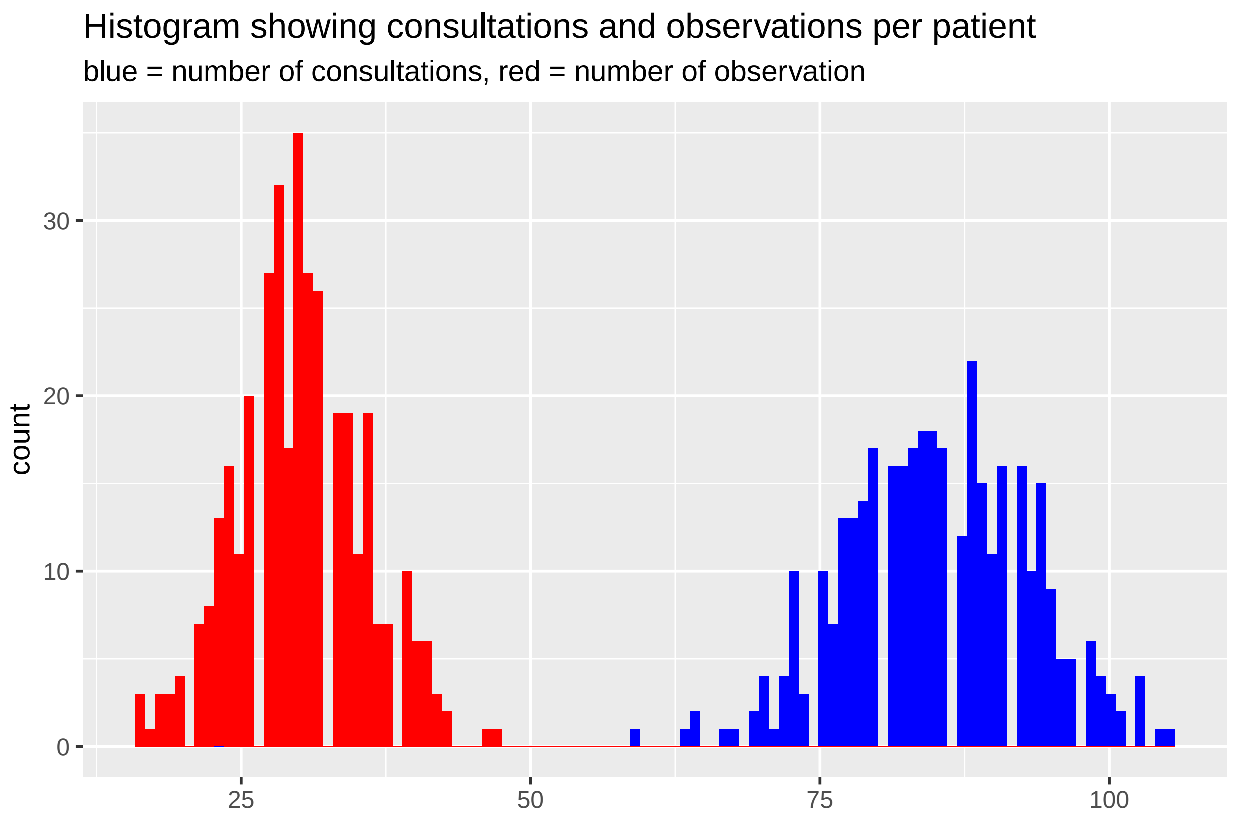
```
pat_cons <- df %>%
  distinct(patid, consid) %>%
  filter(!is.na(consid)) %>%
  summarise( n_cons = n())

pat_obvs <-   df %>%
  distinct(patid, medcodeid) %>%
  filter(!is.na(medcodeid)) %>%
  summarise( n_obvs = n())

plotting_df <- full_join(x = pat_obvs, y = pat_cons, by = "patid")
summary(plotting_df)
```

```
##     patid               n_obvs           n_cons     
##  Length:364         Min.   : 37.00   Min.   :16.00  
##  Class :character   1st Qu.: 80.00   1st Qu.:27.00  
##  Mode  :character   Median : 85.00   Median :30.00  
##                     Mean   : 84.99   Mean   :30.06  
##                     3rd Qu.: 91.00   3rd Qu.:34.00  
##                     Max.   :105.00   Max.   :47.00
```

```
ggplot(data = plotting_df) +
  geom_histogram(aes(x = n_obvs), bins = 105, fill = "blue") +
  geom_histogram(aes(x = n_cons), bins = 105, fill = "red" ) +
  ggtitle(label = "Histogram showing consultations and observations per patient", subtitle = "blue = nu
  xlab("")
```



```
# I should have done it differently - I should have done it on 2 different histograms and glued it toge
```

3. Identify how many patients have each of the following conditions:

   - Migraine
   - Type 2 diabetes
   - Stomach ulcer

```
# Phenotyping is subjective so code lists are shared in databases like:
# UCL/HDR-UK: https://phenotypes.healthdatagateway.org/
# LSHTM: https://datacompass.lshtm.ac.uk/view/keywords/Code_list.html
# Cambridge: https://www.phpc.cam.ac.uk/pcu/research/research-groups/crmh/cprd_cam/codelists/v11/
# Birmingham
# QOF https://digital.nhs.uk/data-and-information/data-collections-and-data-sets/data-collections/quali

# If this was a normal coding system like Read or ICD then I'd phenotype based
# on existing publications and cite the paper. I would also get green light
# from the client for projects


migrane_c <- md %>%
  filter(grepl(pattern = "Migraine", x = Term, ignore.case = TRUE)) %>%
  select(aurum_code) %>%
  unlist(use.names = FALSE)

t2d_c <- md %>%  # known different subtypes.
```

```r
  filter(grepl(pattern = "diabetes", x = Term, ignore.case = TRUE)) %>%
  filter(grepl(pattern = "1", x = Term, ignore.case = TRUE)) %>%
  select(aurum_code) %>%
  unlist(use.names = FALSE)


su_c <- md %>%
  filter(grepl(pattern = "Stomach|ulcer", x = Term, ignore.case = TRUE)) %>%
  filter(aurum_code == "NDWL524827") %>%  # on;y 1 code
  select(aurum_code) %>%
  unlist(use.names = FALSE)

# since i'm running the same code >=3 times I'll write a function
count_disease <- function(x){
  x %>%
    add_count() %>%
    filter(n > 1) %>%
    distinct(patid) %>%
    nrow() %>%
    print()
}




df %>%
  filter(medcodeid %in% migrane_c) %>%
  count_disease() # 20 distinct patients
```

```
## [1] 20
```

```r
df %>%
  filter(medcodeid %in% t2d_c) %>%
  count_disease() # 15
```

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

```r
df %>%
  filter(medcodeid %in% su_c) %>%
  count_disease() # 27
```

```
## [1] 27
```

## Stats

Choose one of the above conditions. Choose an appropriate approach to statistically test if there are gender differences in the presence of this condition (1 = Male, 2 = Female, 3 = Unspecified/Other) and show your output.

```r
cases <- df %>%
  filter(medcodeid %in% su_c) %>%
  add_count() %>%
  filter(n > 1) %>%
  distinct(patid) %>%
  mutate(status = 1)

test_df <- left_join(x = df %>% distinct(patid, gender), y = cases, by = "patid") %>%
  mutate(status = ifelse(test = is.na(status), yes = 0, no = status))

sum(test_df$status) # these must be issues with the gender again
```

```
## [1] 29
```

```
test_table <- table(gender = test_df$gender, disease_status = test_df$status)
chisq.test(test_table)
```

```
## Warning in chisq.test(test_table): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  test_table
## X-squared = 1.0147, df = 2, p-value = 0.6021
```

## Version control & dependency control

We do not expect you to provide robust version control procedures or dependency control for this brief exercise. However, please explain how you would do so given the appropriate time and resoure.

I use git for version control - GitHub for personal use and my PhD while BitBucket for my professional life at Cegedim. Dependencey control could be done in a few different ways. I experimented with packrat before I discovered Docker images. I've build a few containers over at dockerhub. Here is a link: https://hub.docker.com/repository/docker/dendendocks/c3-olap-1536/general. In this instance, as it was part of my PhD, the docker file isn't publicly available.