Atlas test

Wojciech Banas

2025-02-06

Package load

```
#install.packages("pacman")
pacman::p_load(dplyr, kableExtra,
               ggplot2, stringr,
               ggsci, data.table, ggmap,
               lubridate, rnaturalearthdata,
               sf, rnaturalearth, broom,
               readxl)

options(scipen = 9) # disable scientific notation
`%nin%` <- purrr::negate(`%in%`)
theme_set(theme_bw())
```

Data load

```
# Set the directory containing the files- assumes the unpacked test_data is in the same
↪  folder as this .rmd file
directory_path <- "test_data/data/dest"

setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
# Get a list of all files in the directory
file_list <- list.files(directory_path, full.names = TRUE)

# Function create clean names for files
clean_name <- function(filename) {
  name <- basename(filename)
  name <- str_remove(name, "\\.csv\\.gz$")  # Remove .csv.gz
  name <- str_remove(name, "\\.csv$")  # Remove .csv
  name
}

# Loop through each file and load it into memory
for (file in file_list) {
  file_name <- clean_name(file)  # Get cleaned file name
  cat("Loading file: \n", file_name, "\n")
  if (str_detect(file, "\\.csv\\.gz$")) {
    assign(file_name, read.csv(gzfile(file)), envir = .GlobalEnv)  # Read .csv.gz
  } else if (str_detect(file, "\\.csv$")) {
    assign(file_name, read.csv(file), envir = .GlobalEnv)  # Read .csv
```

```
  }
}
```

```
## Loading file:
##  conditions
## Loading file:
##  dictionary_loinc
## Loading file:
##  dictionary_rxnorm
## Loading file:
##  dictionary_snomed
## Loading file:
##  encounters
## Loading file:
##  medications
## Loading file:
##  observations
## Loading file:
##  patients
```

```
# Read UK population estimates from
↪  https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/data


pop_estimates <- read_xlsx("ukpopulationestimates183820231.xlsx",sheet = 7, skip = 1)
```

Task 1

Instructions: The data files contain errors and bugs that reflect common problems and challenges associated with electronic health records. Load and clean the data, both individually and linked. Where patients fail a quality control check, exclude them from the analyses. The output of this task is a clean clean patient demographics table for use in the next steps and sanitized condition, observation and medication files.

Goals: * Remove patients with erroneous birth or death dates from the patients table or sex (not always essential to remove, but here I assume it is, given that we need to calculate adjusted prevalence). Missing data from other fields in that table is left as is, as these aren't explicitly relevant. * Remove entries from other tables that do not map to the patients table * Remove duplicated entries * Remove entries that do not match the expected field type

Sanity checks and cleanup

Here I go through each table, have a look how many patients it describes, and doing some pre-processing if needed.

Patients

- Removing patients with incorrect birth/death dates
- Removing patients with incorrect gender data
- Cleaning up ethnicity
- Providing overview of N unique patients

```
#str(patients)

cat("N rows in patients table:")
```

```
## N rows in patients table:
```

```
nrow(patients)
```

```
## [1] 1162
```

```
cat("N unique patients in patients table:")
```

```
## N unique patients in patients table:
```

```
uniqueN(patients$Id)
```

```
## [1] 1162
```

```
cat("N unique entries by ID and passport:")
```

```
## N unique entries by ID and passport:
```

```
nrow(distinct(patients,
              Id, PASSPORT))
```

```
## [1] 1162
```

```
cat("No duplicated patient IDs detected")
```

```
## No duplicated patient IDs detected
```

```
cat("Errors in birth/death dates:")
```

```
## Errors in birth/death dates:
```

```
patients %>%
  mutate(across(c(BIRTHDATE, DEATHDATE), ~ ymd(.x), .names = "DATE_{.col}"),
         to_remove = case_when(is.na(DATE_BIRTHDATE) ~ "Erroneous birth date",
                               is.na(DATE_DEATHDATE) & (DEATHDATE != "") ~ "Erroneous
                               ↪  death date",
                               DATE_BIRTHDATE > ymd("2025-01-01") ~ "Born in 2025 or
↪  later",# Assuming birth dates in 2025 or later are erroneous
                               DATE_BIRTHDATE < ymd("1850-01-01") ~ "Born before 1850",#
↪  Assuming birth dates before 1850 are erroneous
                               DATE_BIRTHDATE > DATE_DEATHDATE ~ "Died before birth",
                               T~ "No errors")) %>%
  pull(to_remove) %>%
  table()
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(c(BIRTHDATE, DEATHDATE), ~ymd(.x), .names =
##   "DATE_{.col}")`.
## Caused by warning:
## !  2 failed to parse.
```

```
## .
## Born in 2025 or later     Died before birth  Erroneous birth date
##                    15                     1                     4
##             No errors
##                  1142
```

```
cat("These records are removed- assuming correct date of birth is crucial for this
↪  project\n given we need to calculated adjusted prevalence")
```

```
## These records are removed- assuming correct date of birth is crucial for this project
##  given we need to calculated adjusted prevalence
```

```
patients_clean <- patients %>%
  mutate(across(c(BIRTHDATE, DEATHDATE), ~ ymd(.x), .names = "DATE_{.col}"),
         to_remove = case_when(is.na(DATE_BIRTHDATE) ~ "Erroneous birth date",
                               is.na(DATE_DEATHDATE) & (DEATHDATE != "") ~ "Erroneous
                               ↪  death date",
                               DATE_BIRTHDATE > ymd("2025-01-01") ~ "Born in 2025 or
↪  later",# Assuming birth dates in 2025 or later are erroneous
                               DATE_BIRTHDATE < ymd("1850-01-01") ~ "Born before 1850",#
↪  Assuming birth dates before 1850 are erroneous
                               DATE_BIRTHDATE > DATE_DEATHDATE ~ "Died before birth",
                               T~ "No errors")) %>%
  filter(to_remove == "No errors") %>%
  select(-to_remove, -BIRTHDATE, -DEATHDATE)
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(c(BIRTHDATE, DEATHDATE), ~ymd(.x), .names =
##   "DATE_{.col}")`.
## Caused by warning:
## !  2 failed to parse.
```

```
cat("Unique levels of race:")
```

```
## Unique levels of race:
```

```
patients_clean %>%
  pull(RACE) %>%
  table()
```

```
## .
##     ???-    asian    black hawaiian   native    other    white  XJniDSe
##        6       72       95       16        2       11      925       15
```

```
cat("Levels ``???-`` and ``XJniDSe`` are relabelled to ``unknown``")
```

```
## Levels ``???-`` and ``XJniDSe`` are relabelled to ``unknown``
```

```
patients_clean <- patients_clean %>%
  mutate(RACE = ifelse(RACE %in% c("???-",
                                   "XJniDSe"),
                       "Unknown",
                       RACE))


cat("Unique levels of gender:")
```

```
## Unique levels of gender:
```

```
patients_clean %>%
  pull(GENDER) %>%
  table()
```

```
## .
```

```
##        8293.3      F      M
##     20      4    586    532
```

```
cat("Levels ``8293.3 `` and `` `` are  removed. \n Assuming correct gender is crucial as
↪  we need to calculate adjusted prevalence")
```

```
## Levels ``8293.3 `` and `` `` are  removed.
##  Assuming correct gender is crucial as we need to calculate adjusted prevalence
```

```
patients_clean <- patients_clean %>%
  mutate(GENDER = ifelse(GENDER %in% c("8293.3",
                                       ""),
                         "Unknown",
                         GENDER)) %>%
  filter(GENDER != "Unknown")

cat("N unique patients in the new patients table:")
```

```
## N unique patients in the new patients table:
```

```
uniqueN(patients_clean$Id)
```

```
## [1] 1118
```

conditions

- Providing overview of N rows and patients
- Removing records that do not match patients in patients table

```
cat("N rows of data in conditions table")
```

```
## N rows of data in conditions table
```

```
nrow(conditions)
```

```
## [1] 38100
```

```
cat("N rows of distinct data in conditions table")
```

```
## N rows of distinct data in conditions table
```

```
nrow(distinct(conditions))
```

```
## [1] 38069
```

```
cat("N duplicated rows removed:")
```

```
## N duplicated rows removed:
```

```
nrow(conditions) - nrow(distinct(conditions))
```

```
## [1] 31
```

```
conditions_clean <- distinct(conditions)


cat("N records in conditions table without a match in patients table:")
```

```
## N records in conditions table without a match in patients table:
```

```
sum(unique(conditions_clean$PATIENT) %nin% unique(patients_clean$Id))
```

```
## [1] 45
```

```
cat("Records of that patient are removed")
```

```
## Records of that patient are removed
```

```
patient_to_remove <- unique(conditions$PATIENT)[unique(conditions_clean$PATIENT) %nin%
↪  unique(patients_clean$Id)]
conditions_clean <- conditions_clean %>%
  filter(PATIENT %nin% patient_to_remove)

cat("N records with incorrect date")
```

```
## N records with incorrect date
```

```
conditions_clean <- conditions_clean %>%
  mutate(across(c(START, STOP), ~ as.Date(.)))

conditions_clean %>%
  filter(is.na(START)) %>%
  nrow()
```

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

```
conditions_clean %>%
  filter(is.na(STOP)) %>%
  nrow()
```

```
## [1] 7894
```

```
cat("N records in patients table without a match in conditions table:")
```

```
## N records in patients table without a match in conditions table:
```

```
sum(unique(patients_clean$Id) %nin% unique(conditions_clean$PATIENT))
```

```
## [1] 16
```

observations

- Providing overview of N rows and patients
- Removing records that do not match patients in patients table

```
cat("N rows of data in observations table")
```

```
## N rows of data in observations table
```

```
nrow(observations)
```

```
## [1] 531144
```

```
cat("N rows of distinct data in observations table")
```

```
## N rows of distinct data in observations table
```

```r
nrow(distinct(observations))
```

```
## [1] 530501
```

```r
cat("N records in observations table without a match in patients table:")
```

```
## N records in observations table without a match in patients table:
```

```r
sum(unique(observations$PATIENT) %nin% unique(patients_clean$Id))
```

```
## [1] 45
```

```r
cat("Records of that patient are removed")
```

```
## Records of that patient are removed
```

```r
patient_to_remove <- unique(observations$PATIENT)[unique(observations$PATIENT) %nin%
↪  unique(patients_clean$Id)]
observations_clean <- observations %>%
  filter(PATIENT %nin% patient_to_remove)

cat("N rows without a linked encounter")
```

```
## N rows without a linked encounter
```

```r
cat("N records in patients table without a match in observations table:")
```

```
## N records in patients table without a match in observations table:
```

```r
sum(unique(patients_clean$Id) %nin% unique(observations_clean$PATIENT))
```

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

```r
observations_clean <- observations_clean %>%
  mutate(DATE = ymd_hms(DATE))

cat("N records with incorrect date")
```

```
## N records with incorrect date
```

```r
observations_clean %>%
  filter(is.na(DATE)) %>%
  nrow()
```

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

```r
cat("N records without a valid encounter ID, which are removed")
```

```
## N records without a valid encounter ID, which are removed
```

```r
observations_clean %>%
  filter(ENCOUNTER == "") %>% nrow()
```

```
## [1] 30480
```

```r
observations_clean <- observations_clean %>%
  filter(ENCOUNTER != "")
```

medications

- Providing overview of N rows and patients
- Removing records that do not match patients in patients table

```
cat("N rows of data in medications table")
```

```
## N rows of data in medications table
```

```
nrow(medications)
```

```
## [1] 56430
```

```
cat("N rows of distinct data in medications table")
```

```
## N rows of distinct data in medications table
```

```
nrow(distinct(medications))
```

```
## [1] 56422
```

```
cat("N records in observations table without a match in patients table:")
```

```
## N records in observations table without a match in patients table:
```

```
sum(unique(medications$PATIENT) %nin% unique(patients_clean$Id))
```

```
## [1] 42
```

```
cat("Records of that patient are removed")
```

```
## Records of that patient are removed
```

```
patient_to_remove <- unique(medications$PATIENT)[unique(medications$PATIENT) %nin%
↪  unique(patients_clean$Id)]
medications_clean <- medications %>%
  filter(PATIENT %nin% patient_to_remove)

medications_clean <- medications_clean %>%
  mutate(START = ymd_hms(START))

cat("N records with incorrect date")
```

```
## N records with incorrect date
```

```
medications_clean %>%
  filter(is.na(START)) %>%
  nrow()
```

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

```
cat("N records in patients table without a match in observations table:")
```

```
## N records in patients table without a match in observations table:
```

```
sum(unique(patients_clean$Id) %nin% unique(medications_clean$PATIENT))
```

```
## [1] 43
```

Task 2

Provide a comprehensive description of the patients, observations, and conditions tables - for example, provide the number of unique patients, the most frequent ontology terms (e.g. LOINC, SNOMED and RxNorm) and other information you think is important to describe.

Demographics table

- Creating a table describing ethnicity, race, gender, age, death
- Plotting patients on a map

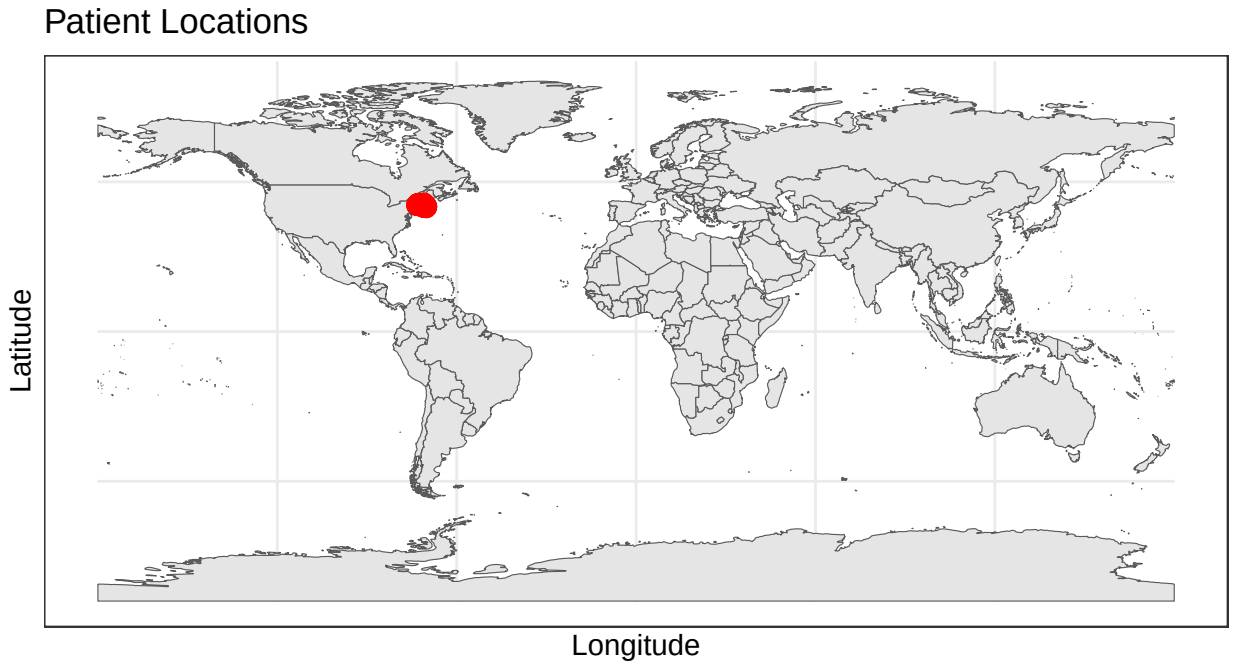
```
n_patients <- nrow(patients_clean)

pat_ethn <- patients_clean %>%
  group_by(ETHNICITY) %>%
  summarise(n = n()) %>%
  rename(item = ETHNICITY)

pat_race <- patients_clean %>%
  group_by(RACE) %>%
  summarise(n = n())%>%
  rename(item = RACE)

pat_gend <- patients_clean %>%
  group_by(GENDER) %>%
  summarise(n = n())%>%
  rename(item = GENDER)

pat_died <- patients_clean %>%
  mutate(dead = !is.na(DATE_DEATHDATE)) %>%
  group_by(dead) %>%
  summarise(n = n())%>%
  rename(item = dead)

#observations_clean$DATE %>% max()
#encounters$START %>% ymd_hms() %>% max()

patients_clean <- patients_clean %>%
  mutate(end_date = as.Date("2021-11-19"), # assuming this is the extract end date, as
  ↪  the max dates of observations and encounters is then
         age = ifelse(!is.na(DATE_DEATHDATE),
                      difftime(DATE_DEATHDATE, DATE_BIRTHDATE),
                      difftime(end_date, DATE_BIRTHDATE)),
         age = age/365.25,
         age_group = cut(age,
                         breaks = c(seq(0, 90, by = 10), Inf),
                         labels = c("0-9", "10-19", "20-29", "30-39", "40-49",
                                    "50-59", "60-69", "70-79", "80-89", "90 or more"),
                         right = FALSE),
         age_group = factor(age_group, levels = c("0-9", "10-19", "20-29", "30-39",
         ↪  "40-49",
                                    "50-59", "60-69", "70-79", "80-89", "90 or more"),
                            ordered = TRUE))

pat_age <- patients_clean %>%
```

```
  group_by(age_group) %>%
  summarise(n = n()) %>%
  rename(item = age_group)

rbind(pat_ethn, pat_race, pat_gend, pat_died, pat_age) %>%
  mutate(perc = n/n_patients) %>%
  rowwise() %>%
  mutate(perc = paste0(n, " (", round(100*perc,2), ")")) %>%
  ungroup() %>%
  rename(`N (%)` = perc) %>%
  select(-n) %>%
  kbl(booktabs = T,
      caption = paste0("Demographics table of the population (N = ", n_patients, ")"))
      ↪  %>%
  pack_rows("Ethnicity", 1,
            nrow(pat_ethn)) %>%
  pack_rows("Race", nrow(pat_ethn) +1,
            nrow(pat_ethn)  +nrow(pat_race)) %>%
  pack_rows("Gender", 1+nrow(pat_ethn)+nrow(pat_race),
            nrow(pat_ethn)+nrow(pat_race)+ nrow(pat_gend)) %>%
  pack_rows("Died", 1+nrow(pat_ethn)+nrow(pat_race)+ nrow(pat_gend),
            nrow(pat_ethn)+nrow(pat_race)+ nrow(pat_gend)+nrow(pat_died))%>%
  pack_rows("Age", 1+ nrow(pat_ethn)+nrow(pat_race)+ nrow(pat_gend)+nrow(pat_died),
            nrow(pat_ethn)+nrow(pat_race)+ nrow(pat_gend)+nrow(pat_died)+nrow(pat_age))
```

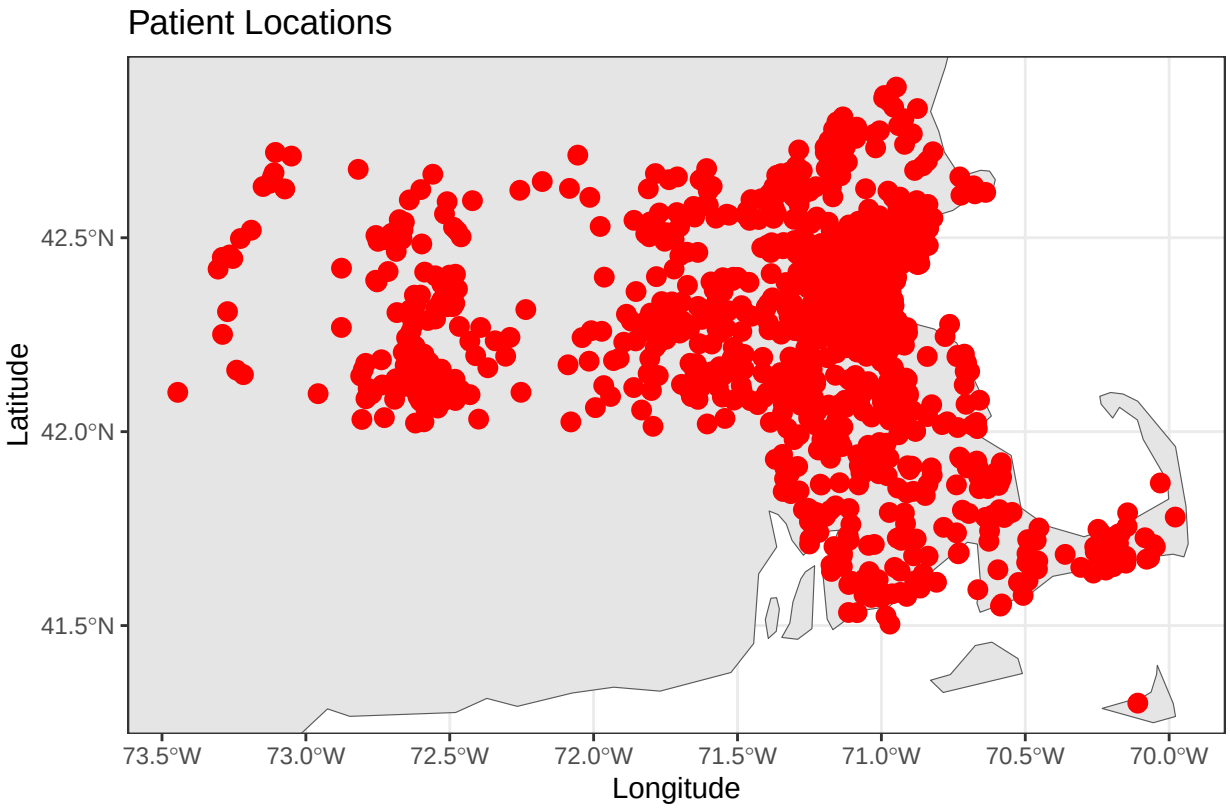
```
# Load a map
world <- ne_countries(scale = "medium", returnclass = "sf")

# Convert data to spatial format
patients_sf <- st_as_sf(patients_clean, coords = c("LON", "LAT"), crs = 4326)

# Plot the map
ggplot() +
  geom_sf(data = world) +
  geom_sf(data = patients_sf, color = "red", size = 3) +
  labs(title = "Patient Locations", x = "Longitude", y = "Latitude")
```

Table 1: Demographics table of the population (N = 1118)

item	N (%)
Ethnicity	
hispanic	102 (9.12)
nonhispanic	1016 (90.88)
Race	
Unknown	21 (1.88)
asian	72 (6.44)
black	94 (8.41)
native	2 (0.18)
other	11 (0.98)
white	918 (82.11)
Gender	
F	586 (52.42)
M	532 (47.58)
Died	
FALSE	962 (86.05)
TRUE	156 (13.95)
Age	
0-9	116 (10.38)
10-19	146 (13.06)
20-29	146 (13.06)
30-39	119 (10.64)
40-49	133 (11.9)
50-59	165 (14.76)
60-69	132 (11.81)
70-79	75 (6.71)
80-89	47 (4.2)
90 or more	39 (3.49)




```
bbox <- st_bbox(patients_sf)
ggplot() +
  geom_sf(data = world) +
  geom_sf(data = patients_sf, color = "red", size = 3) +
    coord_sf(xlim = c(bbox["xmin"], bbox["xmax"]), ylim = c(bbox["ymin"], bbox["ymax"]),
    ↪  expand = TRUE) +
  labs(title = "Patient Locations", x = "Longitude", y = "Latitude")
```



conditions

- Cleaning up SNOMED dictionary and joining it to the conditions table
- Providing an overview of the conditions
- Histogram of the number of conditions (observations) per patient
- Histogram of the events over time

```
# first, clean up the snomed dictionary
duped_snomed_codes <- dictionary_snomed$CODE[dictionary_snomed$CODE %>%
  duplicated()]
to_remove <- dictionary_snomed %>%
  filter(CODE %in% duped_snomed_codes) %>%
  group_by(CODE) %>%
  mutate(n_th = row_number()) %>%
  filter(n_th >1) # assuming no difference between e.g. "Epilepsy (disorder)" and
  ↪  "Epilepsy"

dictionary_snomed_clean <- dictionary_snomed %>%
  filter(DESCRIPTION %nin% to_remove$DESCRIPTION)

conditions_clean <- conditions_clean %>%
  left_join(dictionary_snomed_clean, by = "CODE")

patients_per_condition <- conditions_clean %>%
  distinct(PATIENT, DESCRIPTION) %>%
  pull(DESCRIPTION) %>%
  table() %>%
  as.data.frame() %>%
```

Table 2: Prevalence of the top 10 most frequent SNOMED items(N = 1118)

SNOMED item	N patients (%)
Stress (finding)	861 (77.01)
Full-time employment (finding)	802 (71.74)
Viral sinusitis (disorder)	707 (63.24)
Part-time employment (finding)	666 (59.57)
Limited social contact (finding)	638 (57.07)
Social isolation (finding)	627 (56.08)
Received higher education (finding)	503 (44.99)
Acute viral pharyngitis (disorder)	491 (43.92)
Not in labor force (finding)	484 (43.29)
Body mass index 30+ - obesity (finding)	452 (40.43)

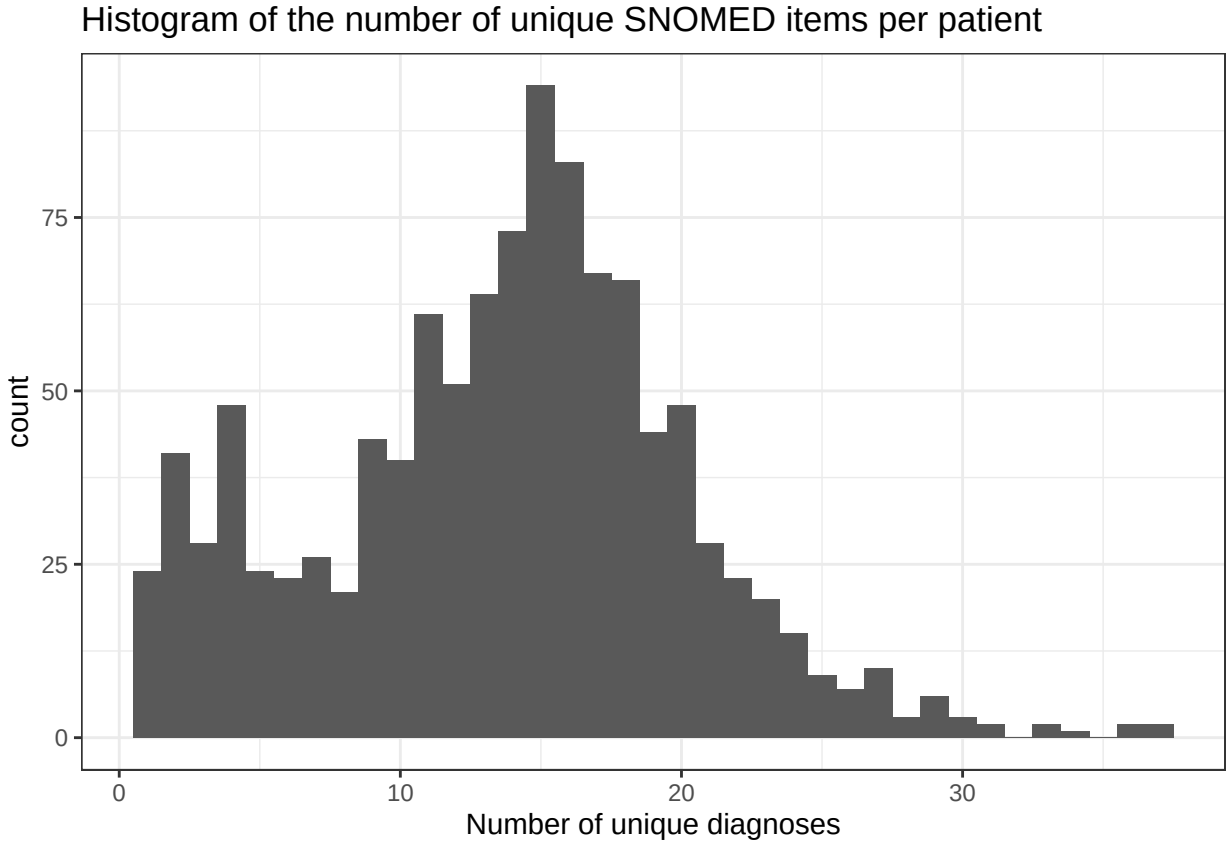
```
  arrange(desc(Freq)) %>%
  mutate(perc = Freq/uniqueN(patients_clean$Id)) %>%
  rowwise() %>%
  mutate(Freq_perc = paste0(Freq, " (", round(perc*100, 2), ")")) %>%
  ungroup() %>%
  select(-perc) %>%
  rename("SNOMED item" = ".",
         "N patients (%)" = Freq_perc)

head(patients_per_condition, 10) %>%
  select(-Freq) %>%
  kbl(booktabs = T,
      caption = paste0("Prevalence of the top 10 most frequent SNOMED items" , "(N = ",
      ↪  n_patients, ")"))
```

```
patients_per_condition %>%
  filter(Freq == min(Freq)) %>%
  select(-Freq) %>%
  kbl(booktabs = T,
      caption = paste0("Prevalence of the least frequent SNOMED items" , " (N = ",
      ↪  n_patients, ")"))
```

```
conditions_clean %>%
  distinct(PATIENT, DESCRIPTION) %>%
  group_by(PATIENT) %>%
  mutate(n = n()) %>%
  distinct(PATIENT, n) %>%
  ggplot(aes(n)) +
  geom_histogram(bins = (conditions_clean %>%
                           distinct(PATIENT, DESCRIPTION) %>%
                           group_by(PATIENT) %>%
                           mutate(n = n()) %>%
                           distinct(PATIENT, n) %>% pull(n) %>% max())) +
  labs(title = "Histogram of the number of unique SNOMED items per patient",
       x = "Number of unique diagnoses")
```

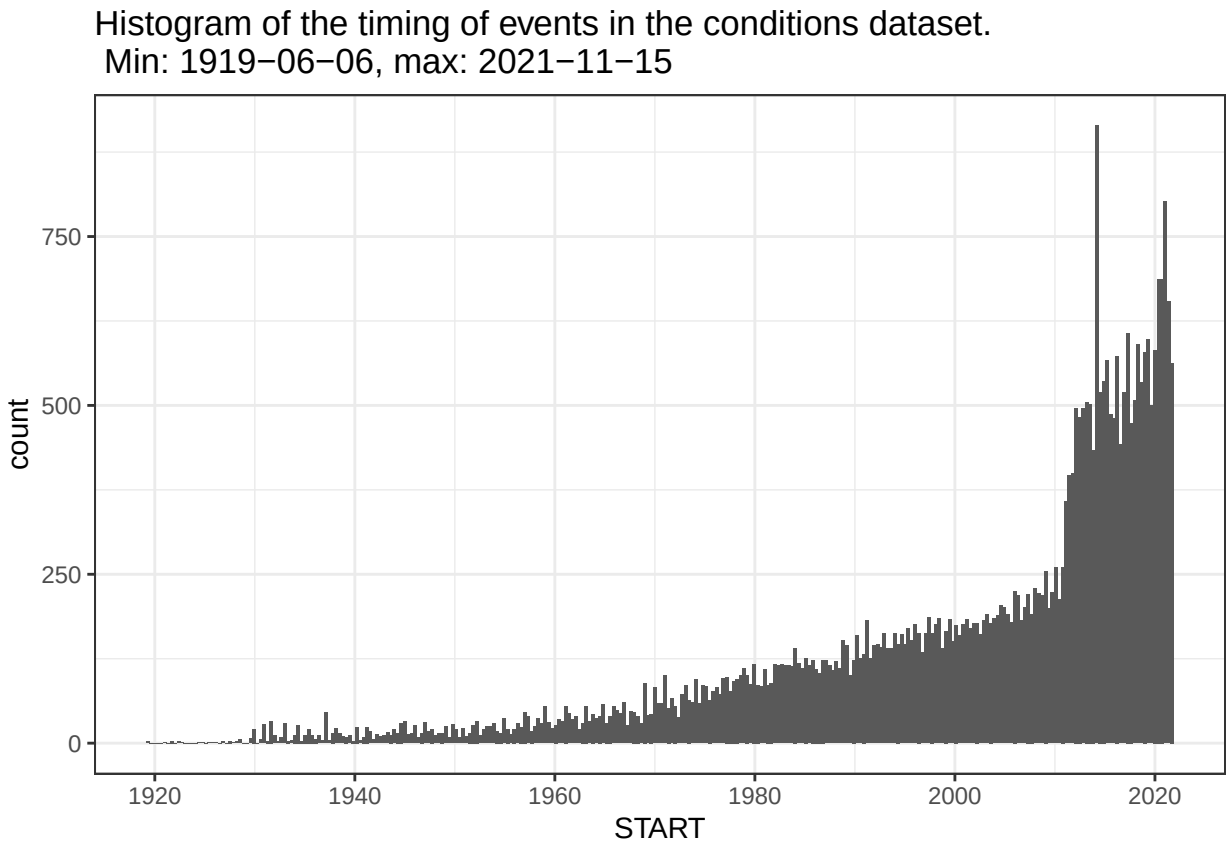
Table 3: Prevalence of the least frequent SNOMED items (N = 1118)

SNOMED item	N patients (%)
Acquired coagulation disorder (disorder)	1 (0.09)
At risk for suicide (finding)	1 (0.09)
Chronic kidney disease stage 2 (disorder)	1 (0.09)
Cystic Fibrosis	1 (0.09)
Dislocation of hip joint (disorder)	1 (0.09)
Fracture of the vertebral column with spinal cord injury	1 (0.09)
History of lower limb amputation (situation)	1 (0.09)
Infection caused by Pseudomonas aeruginosa	1 (0.09)
Injury of heart (disorder)	1 (0.09)
Injury of kidney (disorder)	1 (0.09)
Major depression disorder	1 (0.09)
Male Infertility	1 (0.09)
Microalbuminuria due to type 2 diabetes mellitus (disorder)	1 (0.09)
Non-small cell carcinoma of lung TNM stage 2 (disorder)	1 (0.09)
Proliferative diabetic retinopathy due to type II diabetes mellitus (disorder)	1 (0.09)
Pyelonephritis	1 (0.09)
Spina bifida occulta (disorder)	1 (0.09)
Tear of meniscus of knee	1 (0.09)




```
conditions_clean %>%
```

```
ggplot(aes(START)) +
geom_histogram(bins = 300) +
labs(title = paste0("Histogram of the timing of events in the conditions dataset. \n
↪  Min: ",
                    min(conditions_clean$START), ", max: ",
                     ↪  max(conditions_clean$START)))
```



observations

- Cleaning up loinc dictionary and joining it to the observations table
- Providing an overview of the observations
- Histogram of the number of observations per patient
- Histogram of the events over time

```
cat("Categories of observations")
```

```
## Categories of observations
```

```
observations_clean$CATEGORY %>%
  table()
```

```
## .
##           exam        imaging     laboratory      procedure social-history
##            483             42         159166            459            378
##         survey        therapy    vital-signs
##         210830             27         111926
```

```
cat("Types of observations")
```

```
## Types of observations
```

```
observations_clean$TYPE %>%
  table()
```

```
## .
## numeric    text
##  289891  193420
```

```
cat("Types of Units")
```

```
## Types of Units
```

```
observations_clean$UNITS %>%
  table()
```

```
## .
##                                  %                 /a               /min
##             178754             15272              8589             25028
##             [iU]/L              [pH]               {#}           {count}
##                 12                88             10377                63
##              {INR}         {nominal}          {score}         {T-score}
##                142              6581             37347                86
##            10*3/uL           10*6/uL               Cel                cm
##               7370              3059              1194             13796
##                 fL              g/dL               g/L              K/uL
##              10826             10527              1729                 8
##                 kg             kg/m2              kU/L           m[IU]/L
##              13840             11120              1155                 1
##              mg/dL              mg/g              mg/L            mL/min
##              52345              1992               142              2209
## mL/min/{1.73_m2}                mm           mm[Hg]            mmol/L
##               1992                65             28099             27118
##              ng/dl             ng/mL                pg             pg/mL
##                  1               302              3055               570
##                 pH             ratio                 s               U/L
##                513                 7               142              6869
##              ug/dL              ug/L             ug/mL
##                428               356               142
```

```
cat("Number of unique text-type observations")
```

```
## Number of unique text-type observations
```

```
observations_clean %>%
  filter(TYPE == "text") %>%
  pull(VALUE) %>%
  uniqueN()
```

```
## [1] 1748
```

```
# Numeric observations convert to numeric without errors
#observations_clean %>%
 # filter(TYPE == "numeric") %>%
```

```
  #mutate(VALUE = as.numeric(VALUE))

duped_loinc_codes <- dictionary_loinc$CODE[dictionary_loinc$CODE %>%
  duplicated()]
to_remove <- dictionary_loinc %>%
  filter(CODE %in% duped_loinc_codes) %>%
  arrange(CODE) %>%
  group_by(CODE) %>%
  mutate(string_length = str_length(DESCRIPTION)) %>%
  filter(string_length != max(string_length)) # assuming the correct description is the
  ↪  longer one

dictionary_loinc_clean <- dictionary_loinc %>%
  filter(DESCRIPTION %nin% to_remove$DESCRIPTION)

observations_clean <- observations_clean %>%
  left_join(dictionary_loinc_clean, by = "CODE")


observations_per_patient <- observations_clean %>%
  distinct(PATIENT, DESCRIPTION) %>%
  pull(DESCRIPTION) %>%
  table() %>%
  as.data.frame() %>%
  arrange(desc(Freq)) %>%
  mutate(perc = Freq/uniqueN(patients_clean$Id)) %>%
  rowwise() %>%
  mutate(Freq_perc = paste0(Freq, " (", round(perc*100, 2), ")")) %>%
  ungroup() %>%
  select(-perc) %>%
  rename("loinc item" = ".",
         "N patients (%)" = Freq_perc)


head(observations_per_patient, 10) %>%
  select(-Freq) %>%
  kbl(booktabs = T,
      caption = paste0("Prevalence of the top 10 most frequent loinc items" , "(N = ",
      ↪  n_patients, ")"))
```

```
observations_per_patient %>%
  filter(Freq == min(Freq)) %>%
  select(-Freq) %>%
  kbl(booktabs = T,
      caption = paste0("Prevalence of the least frequent loinc items" , " (N = ",
      ↪  n_patients, ")"))
```

```
observations_clean %>%
  distinct(PATIENT, DESCRIPTION) %>%
  group_by(PATIENT) %>%
  mutate(n = n()) %>%
  distinct(PATIENT, n) %>%
  ggplot(aes(n)) +
```

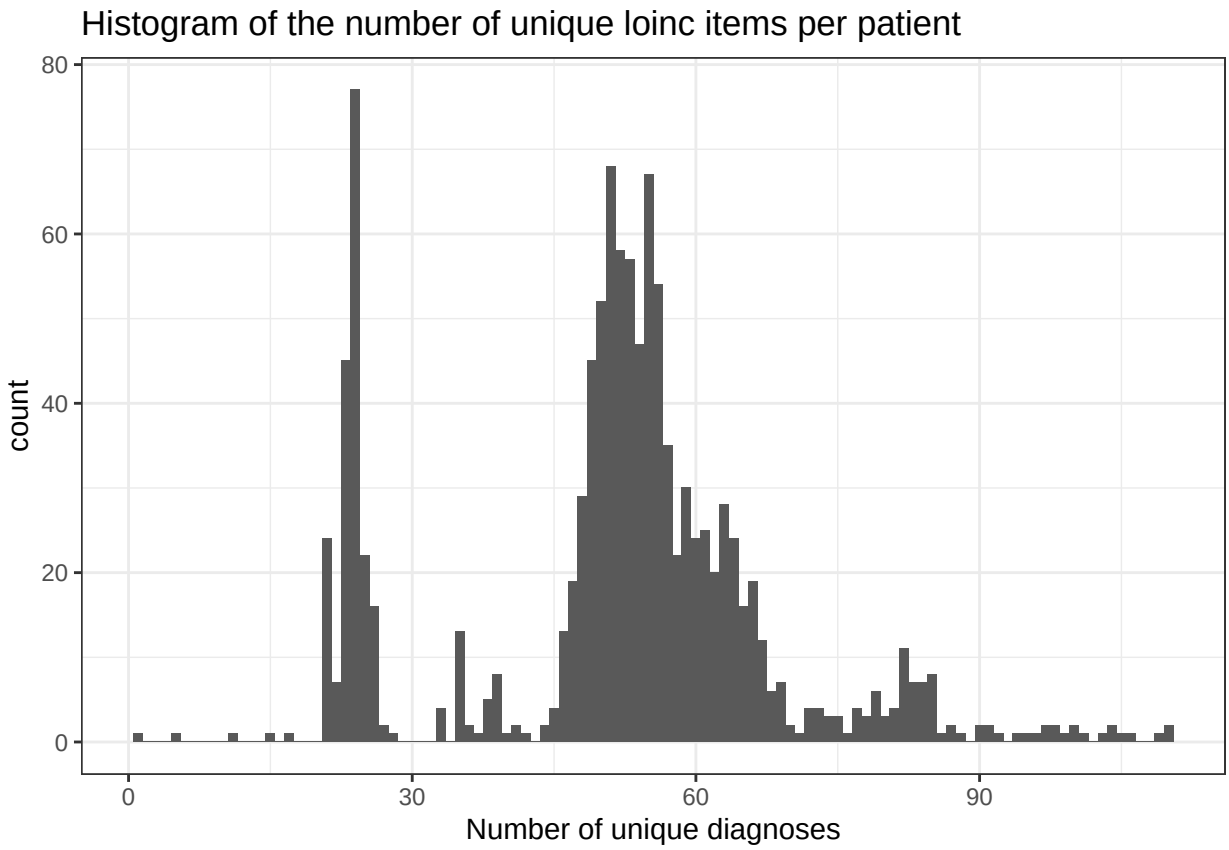
Table 4: Prevalence of the top 10 most frequent loinc items(N = 1118)

loinc item	N patients (%)
Erythrocyte distribution width [Entitic volume] by Automated count	1115 (99.73)
Erythrocytes [#/volume] in Blood by Automated count	1115 (99.73)
Hemoglobin [Mass/volume] in Blood	1115 (99.73)
Leukocytes [#/volume] in Blood by Automated count	1115 (99.73)
MCH [Entitic mass] by Automated count	1115 (99.73)
MCHC [Mass/volume] by Automated count	1115 (99.73)
MCV [Entitic volume] by Automated count	1115 (99.73)
Platelet distribution width [Entitic volume] in Blood by Automated count	1115 (99.73)
Platelet mean volume [Entitic volume] in Blood by Automated count	1115 (99.73)
Platelets [#/volume] in Blood by Automated count	1115 (99.73)

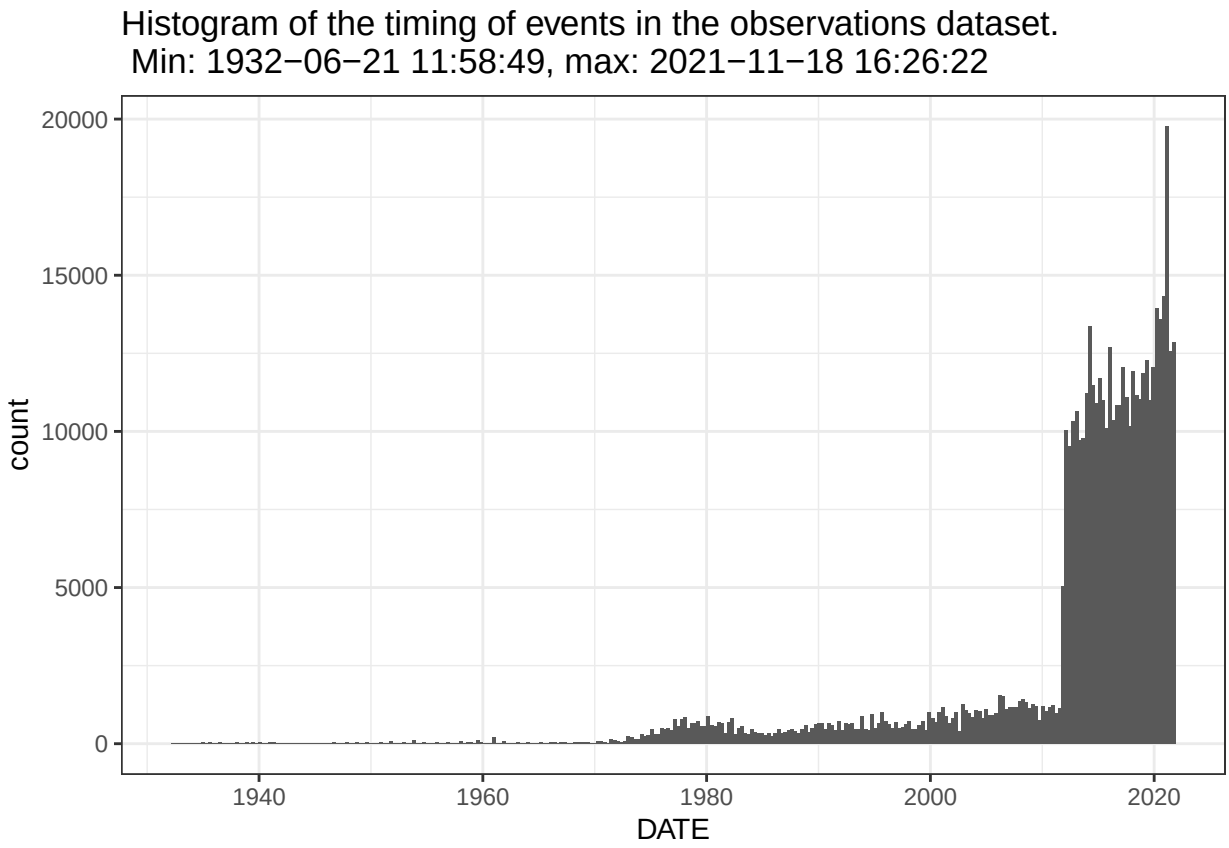
Table 5: Prevalence of the least frequent loinc items (N = 1118)

loinc item	N patients (%)
Thyrotropin [Units/volume] in Serum or Plasma	1 (0.09)
Thyroxine (T4) free [Mass/volume] in Serum or Plasma	1 (0.09)
Weight difference [Mass difference] –pre dialysis - post dialysis	1 (0.09)

```
geom_histogram(bins = (observations_clean %>%
                         distinct(PATIENT, DESCRIPTION) %>%
                         group_by(PATIENT) %>%
                         mutate(n = n()) %>%
                         distinct(PATIENT, n) %>% pull(n) %>% max())) +
labs(title = "Histogram of the number of unique loinc items per patient",
     x = "Number of unique diagnoses")
```




```
observations_clean %>%
  ggplot(aes(DATE)) +
  geom_histogram(bins = 300) +
  labs(title = paste0("Histogram of the timing of events in the observations dataset. \n
  ↪  Min: ",
                      min(observations_clean$DATE), ", max: ",
                       ↪  max(observations_clean$DATE)))
```



Medications

- Cleaning up rxnorm dictionary and joining it to the observations table
- Providing an overview of the medications
- Histogram of the number of medications per patient
- Histogram of the events over time

```
# first, clean up the rxnorm dictionary
duped_rxnorm_codes <- dictionary_rxnorm$CODE[dictionary_rxnorm$CODE %>%
  duplicated()]

to_remove <- dictionary_rxnorm %>%
  filter(CODE %in% duped_rxnorm_codes) %>%
  group_by(CODE) %>%
  mutate(n_th = row_number()) %>%
  filter(n_th >1) # assuming no difference between the duplicates

dictionary_rxnorm_clean <- dictionary_rxnorm %>%
  mutate(CODE = as.character(CODE)) %>%
  filter(DESCRIPTION %nin% to_remove$DESCRIPTION)

medications_clean <- medications_clean %>%
  mutate(CODE = str_remove(CODE, "\\.0$")) %>%
  left_join(dictionary_rxnorm_clean, by = "CODE")

cat("N observations in medications that have no valid description, which will be
↪  removed")
```

Table 6: Prevalence of the top 10 most frequent rxnorm items(N = 1118)

SNOMED item	N patients (%)
Acetaminophen 325 MG Oral Tablet	471 (42.13)
Naproxen sodium 220 MG Oral Tablet	237 (21.2)
Amoxicillin 250 MG / Clavulanate 125 MG Oral Tablet	213 (19.05)
lisinopril 10 MG Oral Tablet	202 (18.07)
Hydrochlorothiazide 25 MG Oral Tablet	182 (16.28)
amLODIPine 2.5 MG Oral Tablet	151 (13.51)
Ibuprofen 200 MG Oral Tablet	139 (12.43)
Simvastatin 10 MG Oral Tablet	133 (11.9)
Acetaminophen 21.7 MG/ML / Dextromethorphan Hydrobromide 1 MG/ML / doxylamine succinate 0.417 MG/ML Oral Solution	124 (11.09)
Acetaminophen 160 MG Chewable Tablet	106 (9.48)

```
## N observations in medications that have no valid description, which will be removed
```

```
medications_clean %>%
  filter(is.na(DESCRIPTION)) %>% nrow()
```

```
## [1] 62
```

```
medications_clean <- medications_clean %>%
  filter(!is.na(DESCRIPTION))

medication_per_patient <- medications_clean %>%
  distinct(PATIENT, DESCRIPTION) %>%
  pull(DESCRIPTION) %>%
  table() %>%
  as.data.frame() %>%
  arrange(desc(Freq)) %>%
  mutate(perc = Freq/uniqueN(patients_clean$Id)) %>%
  rowwise() %>%
  mutate(Freq_perc = paste0(Freq, " (", round(perc*100, 2), ")")) %>%
  ungroup() %>%
  select(-perc) %>%
  rename("SNOMED item" = ".",
         "N patients (%)" = Freq_perc)

head(medication_per_patient, 10) %>%
  select(-Freq) %>%
  kbl(booktabs = T,
      caption = paste0("Prevalence of the top 10 most frequent rxnorm items" , "(N = ",
      ↪  n_patients, ")")) %>%
  kable_styling(latex_options = "scale_down")
```

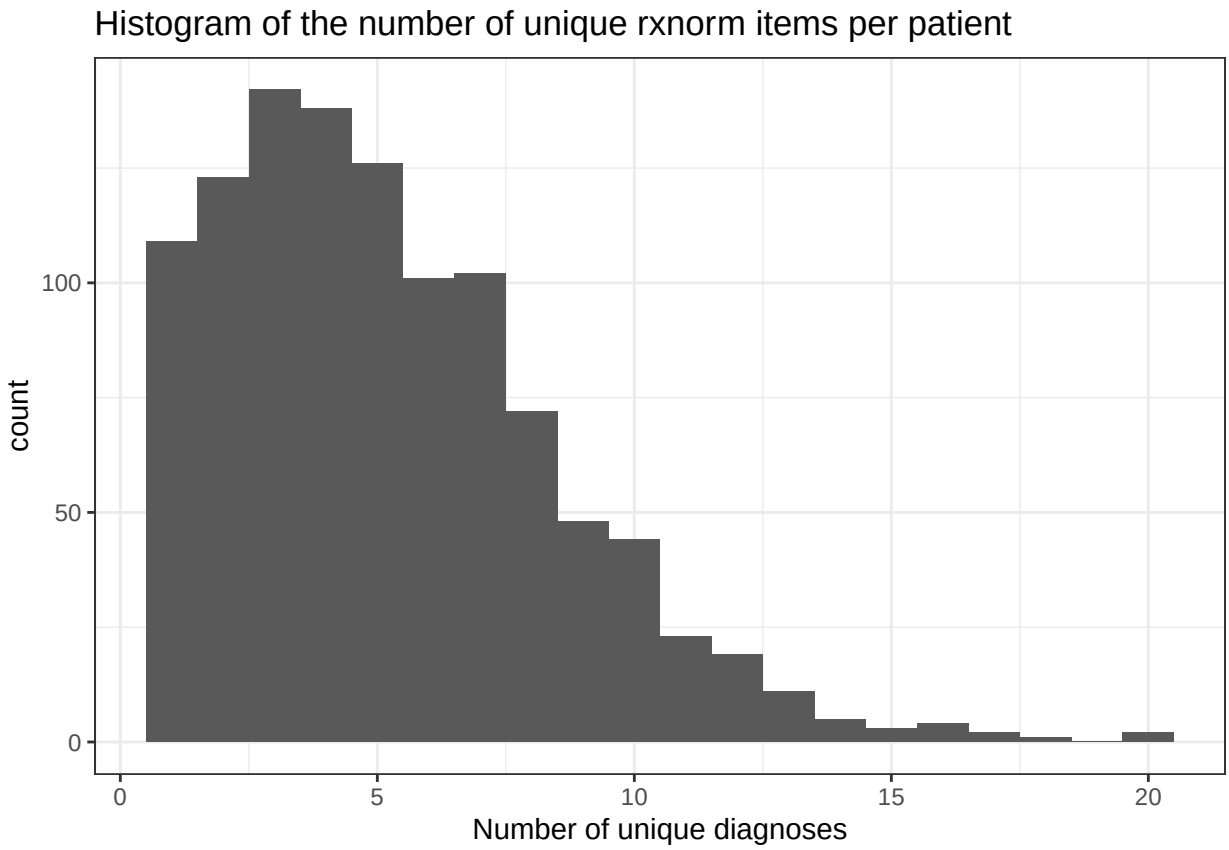
```
medication_per_patient %>%
  filter(Freq == min(Freq)) %>%
  select(-Freq) %>%
  kbl(booktabs = T,
      caption = paste0("Prevalence of the least frequent rxnorm items" , " (N = ",
      ↪  n_patients, ")"))
```

```
medications_clean %>%
  distinct(PATIENT, DESCRIPTION) %>%
  group_by(PATIENT) %>%
  mutate(n = n()) %>%
```

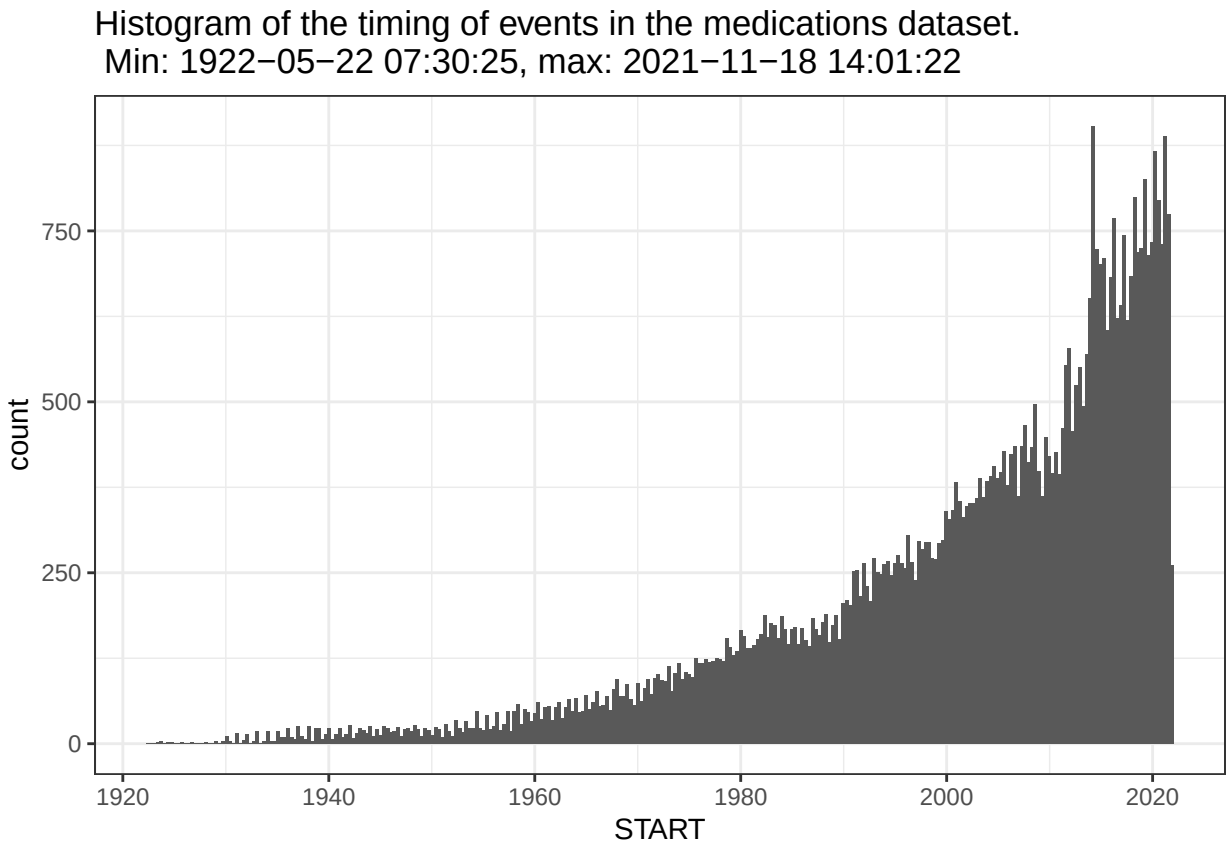
Table 7: Prevalence of the least frequent rxnorm items (N = 1118)

SNOMED item	N patients (%)
1 ML Epoetin Alfa 4000 UNT/ML Injection [Epogen]	1 (0.09)
1 ML Vasopressin (USP) 20 UNT/ML Injection	1 (0.09)
10 ML Doxorubicin Hydrochloride 2 MG/ML Injection	1 (0.09)
10 ML Fentanyl 0.05 MG/ML Injection	1 (0.09)
100 ML Epirubicin Hydrochloride 2 MG/ML Injection	1 (0.09)
20 ML Ciprofloxacin 10 MG/ML Injection	1 (0.09)
5 ML fulvestrant 50 MG/ML Prefilled Syringe	1 (0.09)
5 ML hyaluronidase-oysk 2000 UNT/ML / trastuzumab 120 MG/ML Injection	1 (0.09)
Acetaminophen 325 MG / oxyCODONE Hydrochloride 2.5 MG Oral Tablet	1 (0.09)
Albuterol 5 MG/ML Inhalation Solution	1 (0.09)
Ampicillin 100 MG/ML Injectable Solution	1 (0.09)
Azithromycin 250 MG Oral Capsule	1 (0.09)
baricitinib 2 MG Oral Tablet	1 (0.09)
Carboplatin 10 MG/ML Injectable Solution	1 (0.09)
Clindamycin 300mg	1 (0.09)
Pancreatin 600 MG Oral Tablet	1 (0.09)
Penicillin G 375 MG/ML Injectable Solution	1 (0.09)
predniSONE 20 MG Oral Tablet	1 (0.09)
Pulmozyme (Dornase Alfa)	1 (0.09)
remifentanil 2 MG Injection	1 (0.09)
vancomycin 1000 MG Injection	1 (0.09)

```
distinct(PATIENT, n) %>%
ggplot(aes(n)) +
geom_histogram(bins = (medications_clean %>%
                         distinct(PATIENT, DESCRIPTION) %>%
                         group_by(PATIENT) %>%
                         mutate(n = n()) %>%
                         distinct(PATIENT, n) %>% pull(n) %>% max())) +
labs(title = "Histogram of the number of unique rxnorm items per patient",
     x = "Number of unique diagnoses")
```




```
medications_clean %>%
  ggplot(aes(START)) +
  geom_histogram(bins = 300) +
  labs(title = paste0("Histogram of the timing of events in the medications dataset. \n
  ↪  Min: ",
                      min(medications_clean$START), ", max: ",
                       ↪  max(medications_clean$START)))
```



Task 3a

Using the cleaned data from the first task, explore and compare the distribution of systolic and diastolic blood pressure

- Providing the number of observations
- Showing the histogram and the potentially erroneous data, which are then removed
- Showing updated histogram and overlaid density
- Showing correlation between systolic and diastolic BP per encounter
- Showing summary statistics for BP

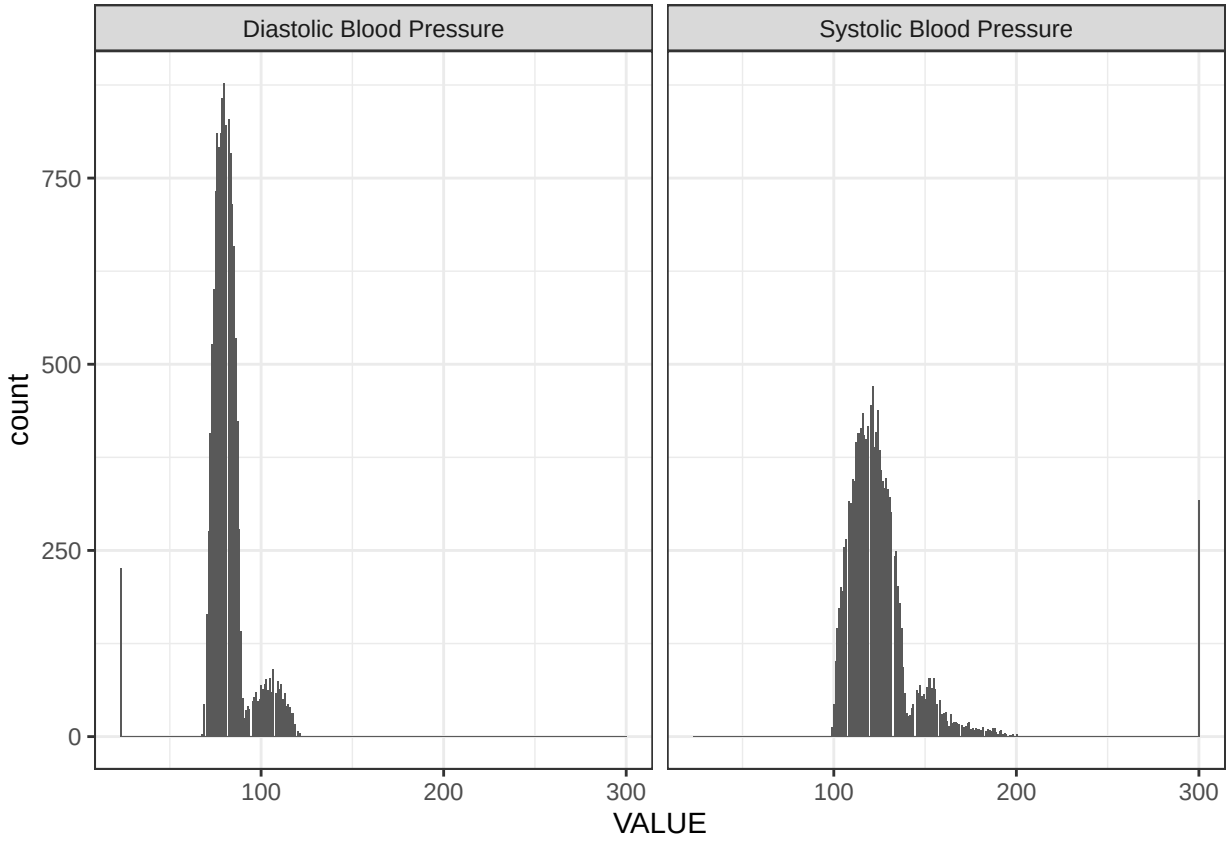
```
cat("N observations of either systolic or diastolic BP")
```

```
## N observations of either systolic or diastolic BP
```

```
task_3a <- observations_clean %>%
  filter(str_detect(DESCRIPTION, "Diastolic|Systolic")) %>%
  mutate(VALUE = as.numeric(VALUE))

task_3a %>%
  pull(DESCRIPTION) %>%
  table()
```

```
## .
## Diastolic Blood Pressure  Systolic Blood Pressure
##                    13932                     13932
```

```
task_3a %>%
  ggplot(aes(VALUE)) +
  geom_histogram(bins = 300) +
  facet_wrap(~DESCRIPTION)
```




```
cat("N unique observations of Diastolic BP <50 and its values")
```

```
## N unique observations of Diastolic BP <50 and its values
```

```
task_3a %>%
  filter(DESCRIPTION == "Diastolic Blood Pressure") %>%
  filter(VALUE < 50) %>%
  group_by( VALUE) %>%
  summarise(n = n()) %>%
  arrange(desc(n))
```

```
## # A tibble: 2 x 2
##   VALUE     n
##   <dbl> <int>
## 1  23     226
## 2  43.3     1
```

```
cat("N unique observations of Systolic BP > 270 and its values")
```

```
## N unique observations of Systolic BP > 270 and its values
```

```
task_3a %>%
  filter(DESCRIPTION == "Systolic Blood Pressure") %>%
```

```
  filter(VALUE > 270) %>%
  group_by( VALUE) %>%
  summarise(n = n()) %>%
  arrange(desc(n))
```

```
## # A tibble: 1 x 2
##   VALUE     n
##   <dbl> <int>
## 1   300   317
```

```
cat("N observations of Systolic BP equal 300. These observations are removed")
```

```
## N observations of Systolic BP equal 300. These observations are removed
```

```
task_3a %>%
  filter(DESCRIPTION == "Systolic Blood Pressure") %>%
  filter(VALUE == 300) %>%
  summarise(n = n()) %>%
  pull(n)
```

```
## [1] 317
```

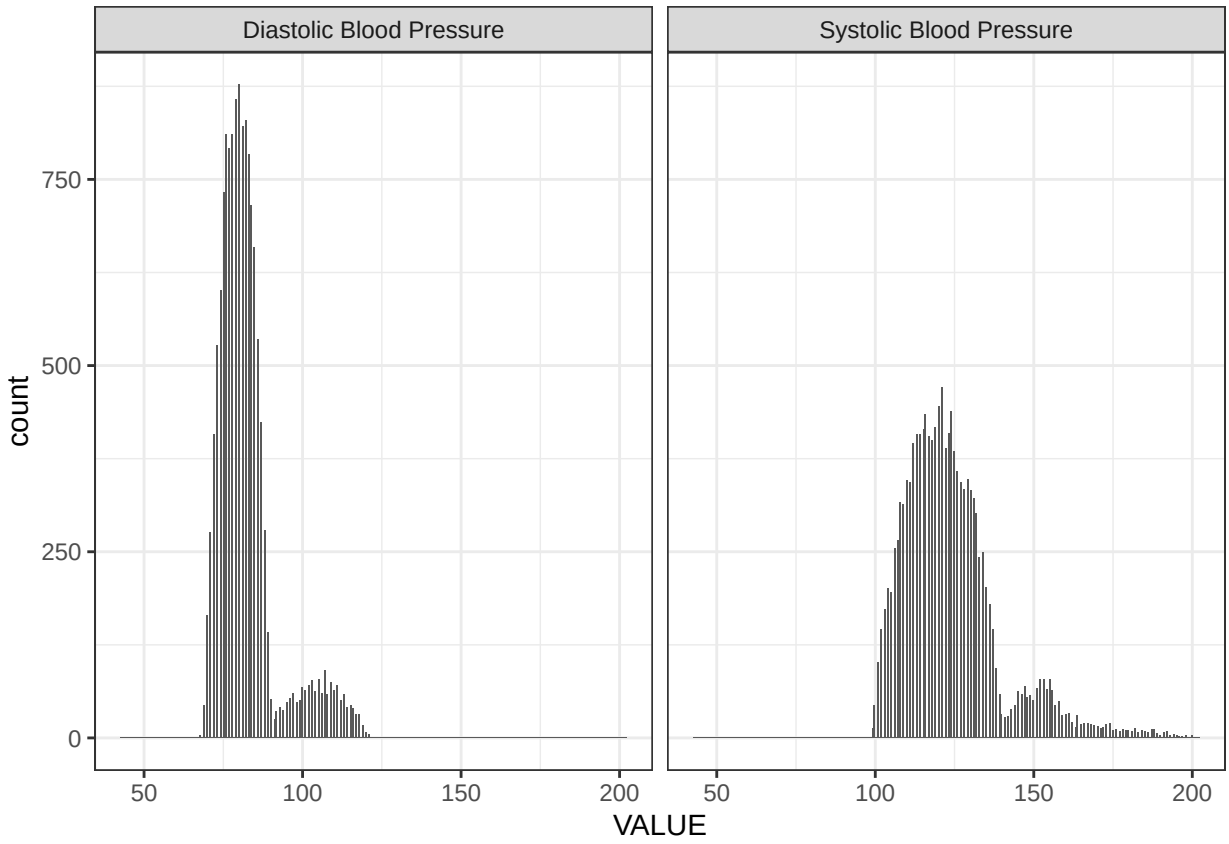
```
cat("N observations of Diastolic BP equal 23 These observations are removed")
```

```
## N observations of Diastolic BP equal 23 These observations are removed
```

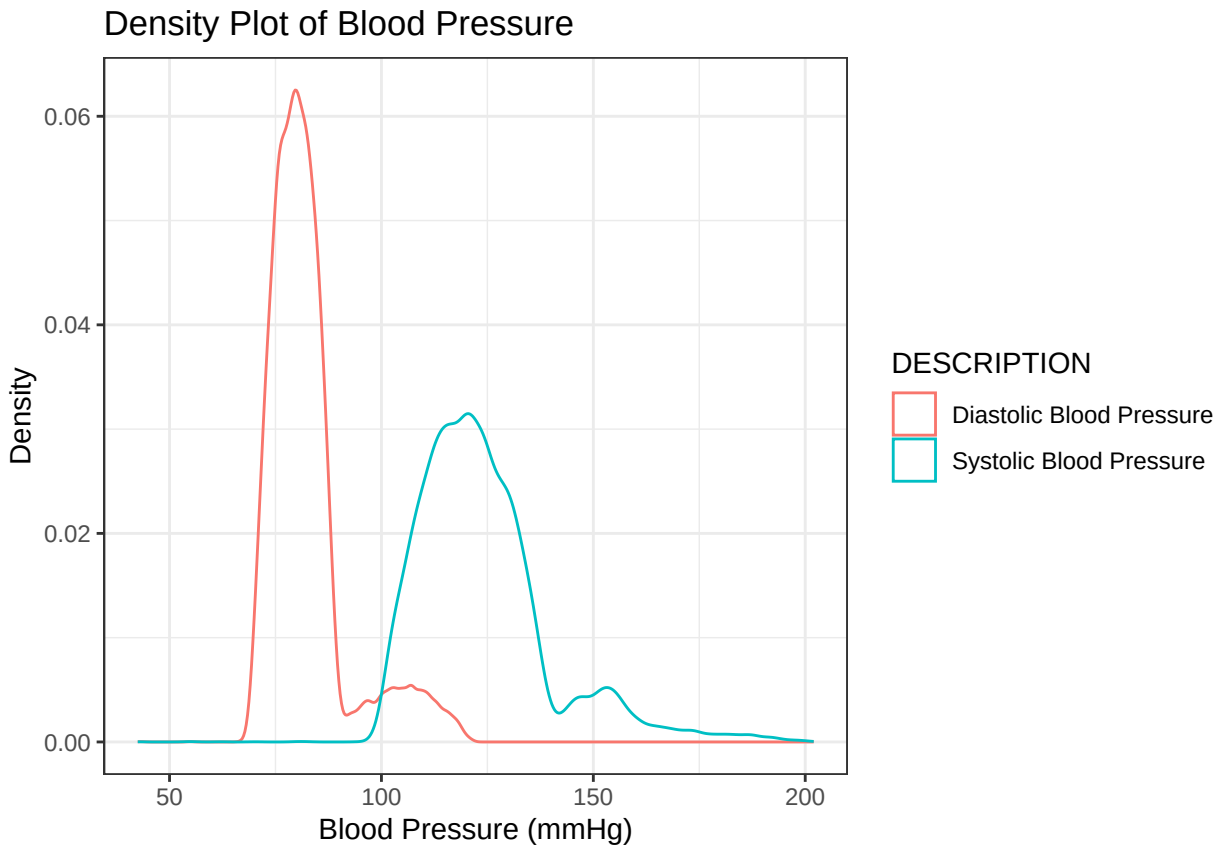
```
task_3a %>%
  filter(DESCRIPTION == "Diastolic Blood Pressure") %>%
  filter(VALUE == 23) %>%
  summarise(n = n()) %>%
  pull(n)
```

```
## [1] 226
```

```
task_3a <- task_3a %>%
  filter(!((DESCRIPTION == "Diastolic Blood Pressure" & VALUE == 23 )|
           (DESCRIPTION == "Systolic Blood Pressure" & VALUE == 300)))

task_3a %>%
  ggplot(aes(VALUE)) +
  geom_histogram(bins = 300) +
  facet_wrap(~DESCRIPTION)
```




```
ggplot(task_3a, aes(x = VALUE, color = DESCRIPTION)) +
  geom_density(alpha = 0.6) +
  labs(title = "Density Plot of Blood Pressure", x = "Blood Pressure (mmHg)", y =
  ↪  "Density")
```




```
task_3a_agg <- task_3a %>%
  group_by(ENCOUNTER, DESCRIPTION) %>%
  summarise(Average_BP = mean(VALUE, na.rm = TRUE), .groups = "drop")  # Using mean, but
  ↪  median also works

# Convert to wide format
task_3a_agg_wide <- task_3a_agg %>%
  tidyr::pivot_wider(names_from = DESCRIPTION, values_from = Average_BP)

# Rename columns for clarity
colnames(task_3a_agg_wide) <- c("encounter_ID", "Diastolic_BP", "Systolic_BP")

ggplot(task_3a_agg_wide, aes(x = Systolic_BP, y = Diastolic_BP)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") +
  labs(title = "Scatter Plot: Systolic vs. Diastolic BP per encounter",
       x = "Systolic BP (mmHg)",
       y = "Diastolic BP (mmHg)")
```

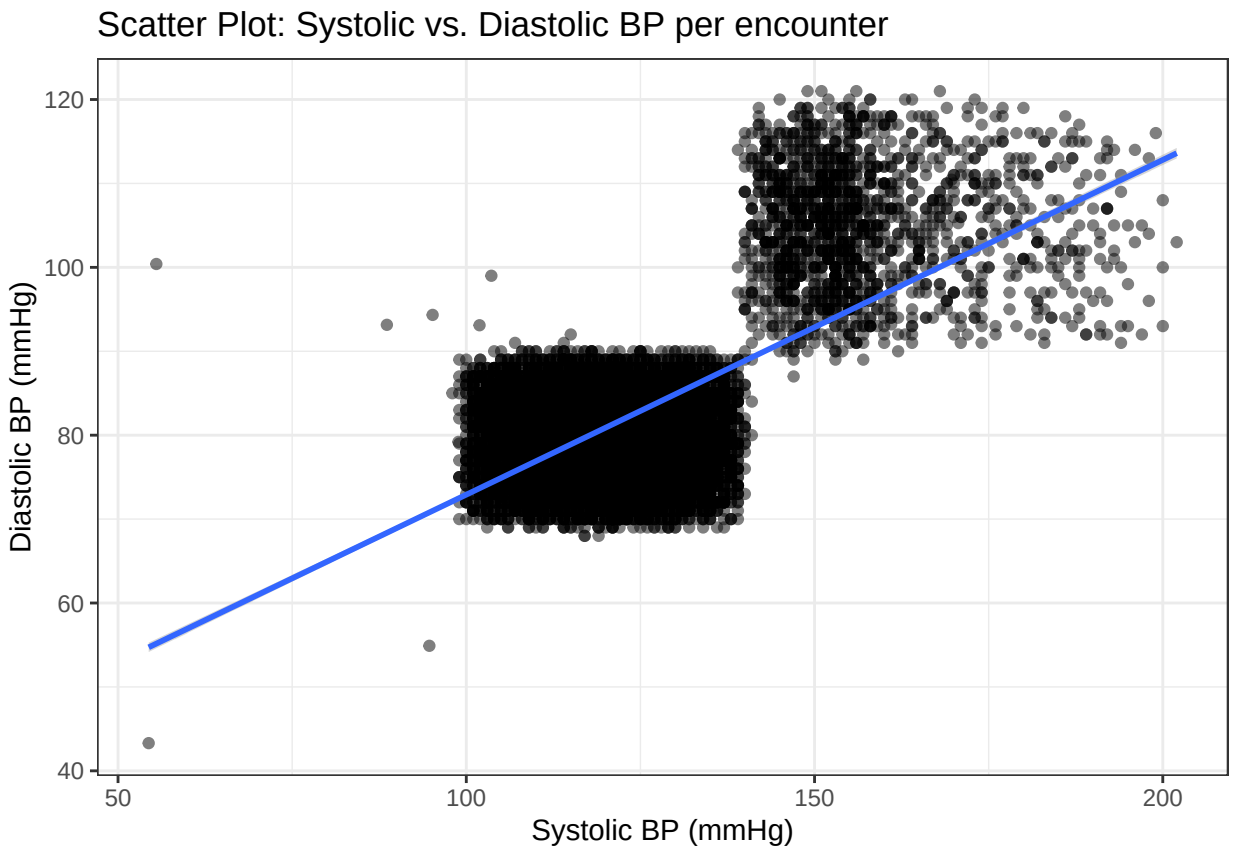
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## Warning: Removed 386 rows containing non-finite outside the scale range
## (`stat_smooth()`).
```

```
## Warning: Removed 386 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

Table 8: Summary statistics of blood pressure values across the whole datset.

DESCRIPTION	Mean	SD	Median	IQR	Min	Max
Diastolic Blood Pressure	82.41974	9.565119	80	9	43.3	121
Systolic Blood Pressure	123.88837	15.711768	121	17	42.5	202




```
task_3a %>%
  group_by(DESCRIPTION) %>%
  summarise(Mean = mean(VALUE),
            SD = sd(VALUE),
            Median = median(VALUE),
            IQR = IQR(VALUE),
            Min = min(VALUE),
            Max = max(VALUE)) %>%
  kbl(booktabs = T,
      caption = "Summary statistics of blood pressure values across the whole datset.")
```

Task 3b

Using the cleaned data from the first task, explore and compare the distribution of BMI in patients with diagnosed hypertension

- Assuming that patients never "loose" the diagnosis of hypertension
- Showing N patients with hypertension

- Annotating observations with hypertension status, ensuring date is accounted for
- Calculating BMI values for those who do not have these recorded (younger patients). BMI was calculated by hand, using height and weight from the same encounter. encounters without height are removed, as imputation of height would be troublesome and inaccurate
- Showing histograms of BMI for patients with BMI and when BMI was calculated by hand
- Showing BMI by hypertension status, and overlaid density plots
- Showing summary metrics of BMI among patients with and without hypertension
- Showing results of simple logistic regression with BMI as predictor and hypertension diagnosis as outcome (not really describing the distribution but additional way to describe the association)

```
cat("N patients with hypertension")
```

```
## N patients with hypertension
```

```
conditions_clean %>%
  filter(str_detect(DESCRIPTION, "(?i)Hypertension")) %>%
  distinct(PATIENT, .keep_all = T) %>%
  pull(DESCRIPTION) %>% table()
```

```
## .
## Hypertension
##          282
```

```
# Pulling patients with hypertension and their first diagnosis
hypertension_ids_dates <- conditions_clean %>%
  filter(str_detect(DESCRIPTION, "Hypertension")) %>%
  distinct(PATIENT, START) %>%
  group_by(PATIENT) %>%
  mutate(min_start = min(START)) %>%
  ungroup() %>%
  filter(START == min_start) %>%
  select(-min_start) %>%
  mutate(hypertension = T)

# Addding hypertension diagnosis, ensuring the timing of the diagnosis is taken into
↪  account
observations_clean <- observations_clean %>%
  left_join(hypertension_ids_dates, by = c("PATIENT")) %>%
  mutate(START = ifelse(is.na(START),
                        ("2030-01-01"),
                        as.character(START)),
         START = ymd(START)) %>%
  mutate(hypertension = ifelse(START <= DATE,
                               T,
                               F)) %>%
  select(-START)

patients_with_BMI_values <- observations_clean %>%
  filter(DESCRIPTION == "Body Mass Index") %>%
  pull(PATIENT) %>%
  unique()

cat("N patients without a BMI record. BMI is calculated for \n those patients using
↪  weight and height measurements")
```

```
## N patients without a BMI record. BMI is calculated for
##  those patients using weight and height measurements
```

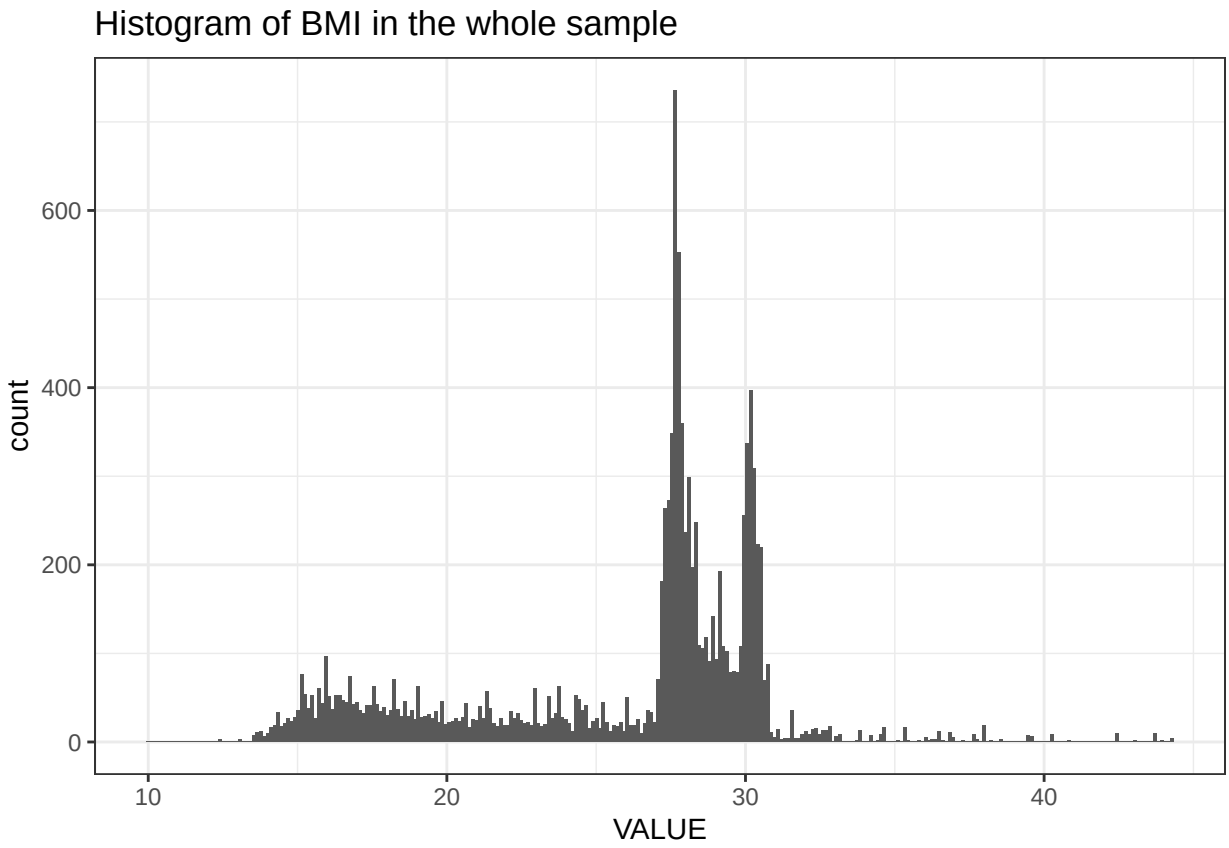
```
((observations_clean$PATIENT %>% unique()) %nin% patients_with_BMI_values ) %>%
  sum()
```

```
## [1] 41
```

```
cat("Body height is sometimes missing for encounters of patients without BMI. \n These
↪  encoutners are removed as there is enough data already for this comparison. \n Height
↪  could be imputed as mean, but for those young patients this \n may not work well.
↪  Carrying forward last value also may not work well. \n Therefore entries with weight
↪  but no height are removed")
```

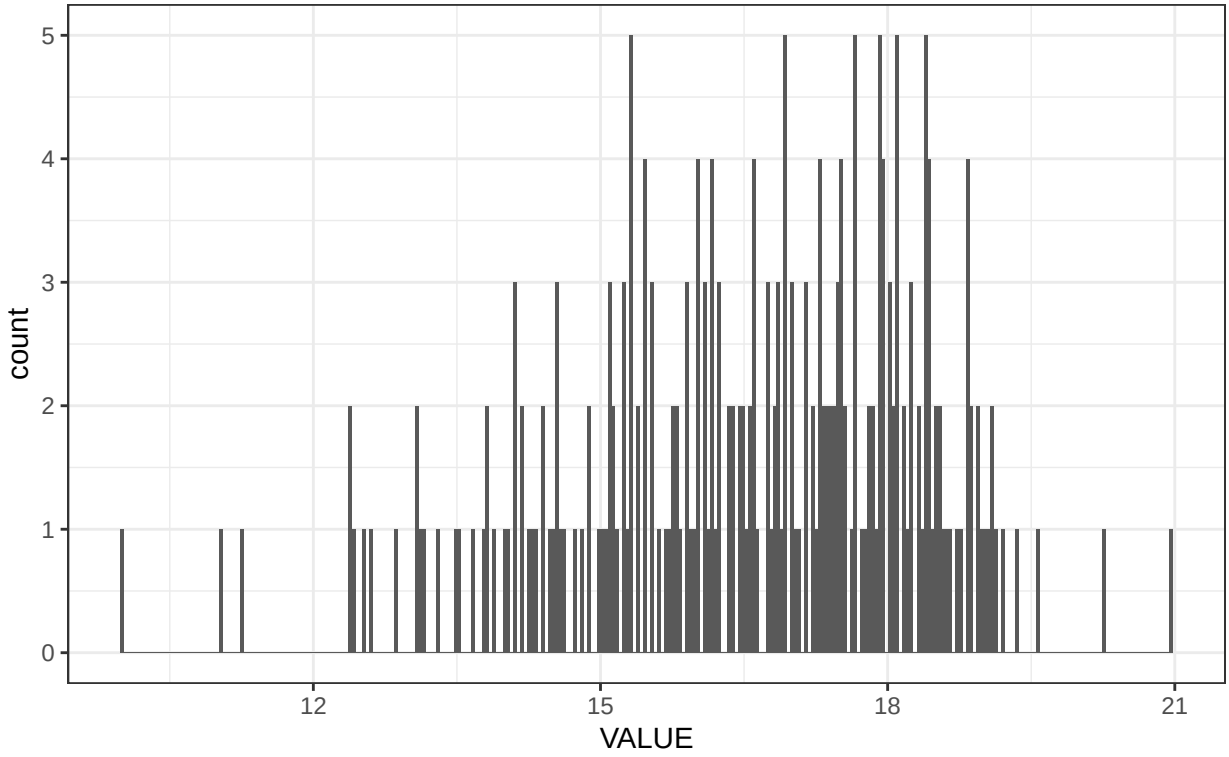
```
## Body height is sometimes missing for encounters of patients without BMI.
##  These encoutners are removed as there is enough data already for this comparison.
##  Height could be imputed as mean, but for those young patients this
##  may not work well. Carrying forward last value also may not work well.
##  Therefore entries with weight but no height are removed
```

```
BMI_by_hand <- observations_clean %>%
  filter(PATIENT %nin% patients_with_BMI_values) %>%
  filter(DESCRIPTION %in% c("Body Height", "Body Weight")) %>%
  tidyr::pivot_wider(id_cols = c(PATIENT, ENCOUNTER, DATE, hypertension), names_from =
↪  DESCRIPTION, values_from = VALUE) %>%
  filter(!is.na(`Body Height`)) %>%
  mutate(across(contains("Body"), ~ as.numeric(.)),
         BMI_by_hand = `Body Weight`/(`Body Height`/100)^2,
         VALUE = BMI_by_hand,
         DESCRIPTION = "Body Mass Index",
         by_hand = T) %>%
  select(DATE, PATIENT, ENCOUNTER, VALUE, DESCRIPTION, by_hand, hypertension)

task_3b <- observations_clean %>%
  filter(DESCRIPTION == "Body Mass Index") %>%
  select(DATE, PATIENT, ENCOUNTER, VALUE, DESCRIPTION, hypertension) %>%
  mutate(by_hand = F) %>%
  rbind(BMI_by_hand) %>%
  mutate(VALUE = as.numeric(VALUE))


task_3b %>%
  ggplot(aes(VALUE)) +
  geom_histogram(bins = 300) +
  labs(title = "Histogram of BMI in the whole sample")
```




```
BMI_by_hand %>%
  ggplot(aes(VALUE)) +
  geom_histogram(bins = 300) +
  labs(title = "Histogram of BMI in the subset where BMI was calculated by hand\nThis
  ↪  population is quite different from the rest.")
```

Histogram of BMI in the subset where BMI was calculated by hand
This population is quite different from the rest.



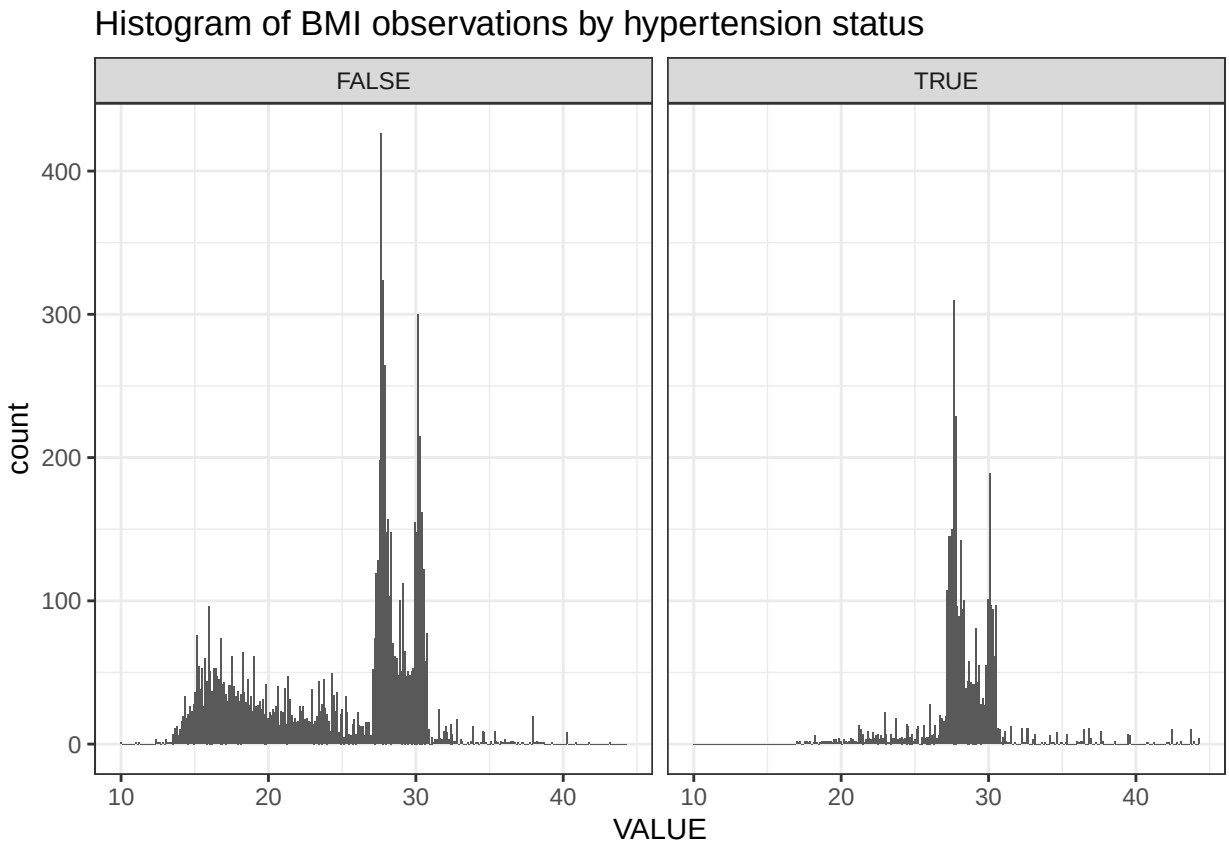

```
cat("Mean age across patients for whom BMI was missing vs not missing.\n This explains
↪  the small height and odd BMI distribution")
```

```
## Mean age across patients for whom BMI was missing vs not missing.
##  This explains the small height and odd BMI distribution
```

```
task_3b %>%
  left_join(patients_clean %>%
              select(Id, DATE_BIRTHDATE), by = c("PATIENT" = "Id")) %>%
  mutate(age = as.numeric(difftime(DATE, DATE_BIRTHDATE, units = "days"))/365.25) %>%
  group_by(by_hand) %>%
  summarise(mean_age = mean(age))
```

```
## # A tibble: 2 x 2
##   by_hand mean_age
##   <lgl>      <dbl>
## 1 FALSE     44.6
## 2 TRUE       0.586
```

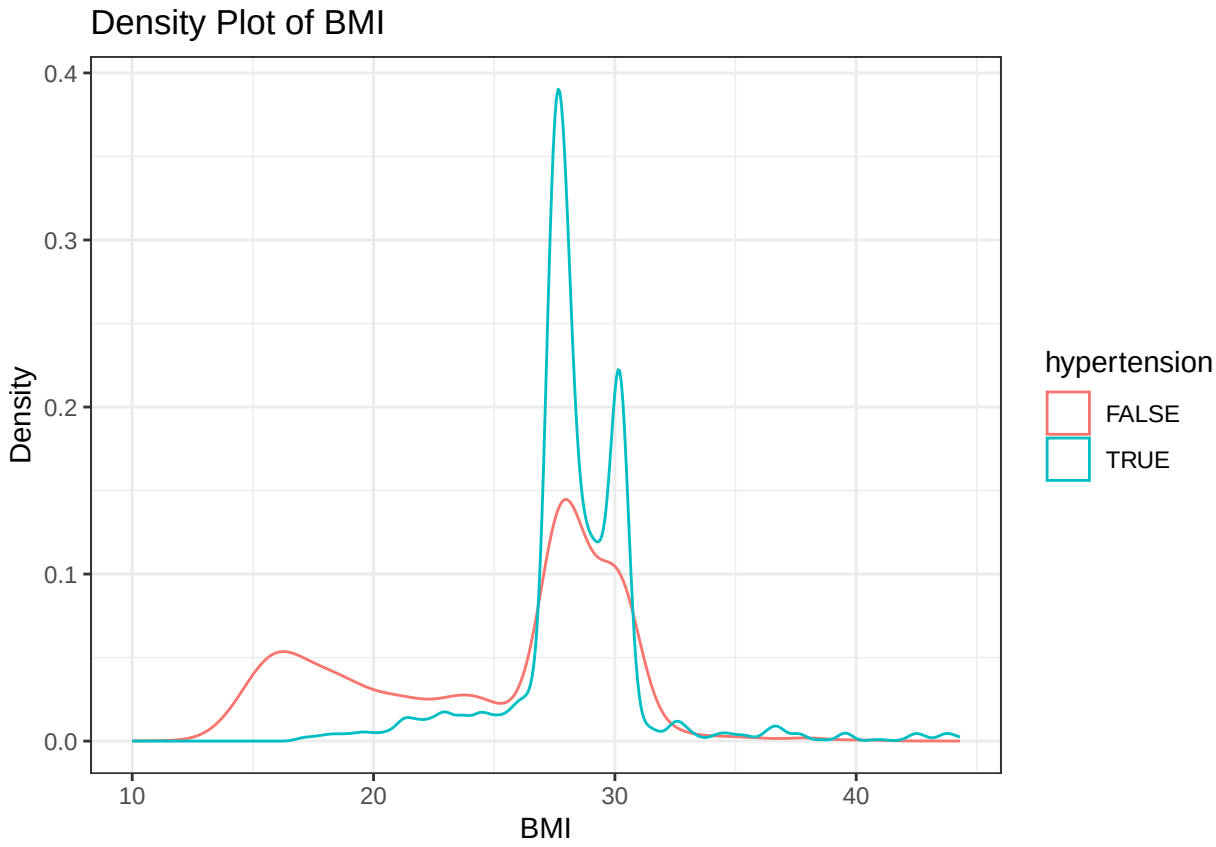
```
task_3b %>%
  ggplot(aes(VALUE)) +
  geom_histogram(bins = 300) +
  facet_wrap(~hypertension) +
  labs(title = "Histogram of BMI observations by hypertension status")
```




```
ggplot(task_3b, aes(x = VALUE, color = hypertension)) +
  geom_density(alpha = 0.6) +
  labs(title = "Density Plot of BMI", x = "BMI", y = "Density")
```

Table 9: Summary statistics of BMI values across the hypertension status.

hypertension	Mean	SD	Median	IQR	Min	Max
FALSE	24.81667	5.526189	27.5	9.5	9.989877	43.2
TRUE	28.36653	2.966182	28.1	2.3	17.000000	44.3




```
task_3b %>%
  group_by(hypertension) %>%
  summarise(Mean = mean(VALUE),
            SD = sd(VALUE),
            Median = median(VALUE),
            IQR = IQR(VALUE),
            Min = min(VALUE),
            Max = max(VALUE)) %>%
  kbl(booktabs = T,
      caption = "Summary statistics of BMI values across the hypertension status.")
```

```
model <- glm(hypertension ~ VALUE, data = task_3b, family = binomial)

cat("Results of logistic regression with hypertension as outcome and BMI as predictor
↪  variable")
```

```
## Results of logistic regression with hypertension as outcome and BMI as predictor variable
```

```
tidy(model, exponentiate = TRUE, conf.int = TRUE)
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 (Intercept)  0.00383   0.161       -34.5 1.52e-261  0.00278   0.00523
## 2 VALUE        1.19      0.00579      30.7 1.37e-207  1.18      1.21
```

Task 4

Report the crude, and adjusted (to the UK population as much as possible) prevalence of hypertension.

- Not accounting for recovery from hypertension
- Including all patients, regardless if died within the period
- Using UK estimates for mid 2022

```
crude_prevalence <- conditions_clean %>%
  filter(str_detect(DESCRIPTION, "Hypertension")) %>%
  distinct(PATIENT) %>%
  nrow() / n_patients

cat(paste0("Crude prevalence in the whole dataset (N = ", n_patients, ") : ",
↪  round(crude_prevalence*100, 2), "%"))
```

```
## Crude prevalence in the whole dataset (N = 1118) : 25.22%
```

```
patients_hypertension <- conditions_clean %>%
  filter(str_detect(DESCRIPTION, "Hypertension")) %>%
  pull(PATIENT) %>%
  unique()


US_ests <- patients_clean %>%
  mutate(hypertension = Id %in% patients_hypertension) %>%
  group_by(age_group, GENDER) %>%
  mutate(n = n()) %>%
  group_by(hypertension, GENDER, age_group) %>%
  mutate(n_hypertensive = n()) %>%
  distinct(hypertension, n, n_hypertensive, age_group) %>%
  arrange(age_group, GENDER) %>%
  filter(hypertension) %>%
  mutate(hypertension_prevalence = n_hypertensive/n,
         GENDER = ifelse(GENDER == "F",
                         "Females",
                         "Males"))



UK_ests_today <- pop_estimates %>%
  filter(Sex %in% c("Males", "Females"),
         Age != "All Ages") %>%
  mutate(Age = str_replace(Age, "\\+", ""),
         Age = as.numeric(Age),
         age_group = factor(cut(Age,
```

```
                                breaks = c(seq(0, 90, by = 10), Inf),
                                labels = c("0-9", "10-19", "20-29", "30-39", "40-49",
                                           "50-59", "60-69", "70-79", "80-89", "90 or
                                           ↪  more"),
                                right = FALSE), # Ensures inclusive lower bounds
                              ordered = TRUE)) %>%
  group_by(Sex, age_group) %>%
  summarise(sum_n = sum(`Mid-2022`))
```

```
## `summarise()` has grouped output by 'Sex'. You can override using the `.groups`
## argument.
```

```
UK_pop_total <- UK_ests_today$sum_n %>%
  sum()

adjusted_prevalence <- UK_ests_today %>%
  mutate(prop = sum_n / UK_pop_total,
         ) %>%
  inner_join(US_ests, by = c("Sex" = "GENDER",
                             "age_group")) %>%
  mutate(prev_times_weight = hypertension_prevalence * prop) %>%
  pull(prev_times_weight) %>%
  sum()

cat(paste0("Adjusted prevalence, to the UK population in mid-2022 is: ",
↪  round(adjusted_prevalence, 4)*100, "%"))
```

```
## Adjusted prevalence, to the UK population in mid-2022 is: 25.59%
```