# Biostat 203B Homework 3

**Due Feb 21 @ 11:59PM**

Wenjing Zhou and 806542441

Display machine information for reproducibility:

```
sessionInfo()
```

```
R version 4.4.1 (2024-06-14)
Platform: x86_64-apple-darwin20
Running under: macOS 15.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRlapack.dylib;

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Los_Angeles
tzcode source: internal

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
 [1] compiler_4.4.1    fastmap_1.2.0     cli_3.6.3         tools_4.4.1
 [5] htmltools_0.5.8.1 rstudioapi_0.17.0 yaml_2.3.10       rmarkdown_2.29
 [9] knitr_1.49        jsonlite_1.8.9    xfun_0.48         digest_0.6.37
[13] rlang_1.1.4       evaluate_1.0.1
```

Load necessary libraries (you can add more as needed).

```
library(arrow)
```

```
Attaching package: 'arrow'

The following object is masked from 'package:utils':

    timestamp
```

```
library(gtsummary)
library(memuse)
library(pryr)
```

```
Attaching package: 'pryr'

The following object is masked from 'package:gtsummary':

    where
```

```
library(R.utils)
```

```
Loading required package: R.oo

Loading required package: R.methodsS3

R.methodsS3 v1.8.2 (2022-06-13 22:00:14 UTC) successfully loaded. See ?R.methodsS3 for help.

R.oo v1.27.0 (2024-11-01 18:00:02 UTC) successfully loaded. See ?R.oo for help.

Attaching package: 'R.oo'

The following object is masked from 'package:R.methodsS3':

    throw
```

```
The following objects are masked from 'package:methods':

    getClasses, getMethods

The following objects are masked from 'package:base':

    attach, detach, load, save

R.utils v2.12.3 (2023-11-18 01:00:02 UTC) successfully loaded. See ?R.utils for help.


Attaching package: 'R.utils'

The following object is masked from 'package:arrow':

    timestamp

The following object is masked from 'package:utils':

    timestamp

The following objects are masked from 'package:base':

    cat, commandArgs, getOption, isOpen, nullfile, parse, use, warnings
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x purrr::compose()      masks pryr::compose()
x lubridate::duration() masks arrow::duration()
x tidyr::extract()      masks R.utils::extract()
x dplyr::filter()       masks stats::filter()
```

```
x dplyr::lag()          masks stats::lag()
x purrr::partial()      masks pryr::partial()
x dplyr::where()        masks pryr::where(), gtsummary::where()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(ggplot2)
```

Display your machine memory.

```
memuse::Sys.meminfo()
```

```
Totalram:  16.000 GiB
Freeram:    1.480 GiB
```

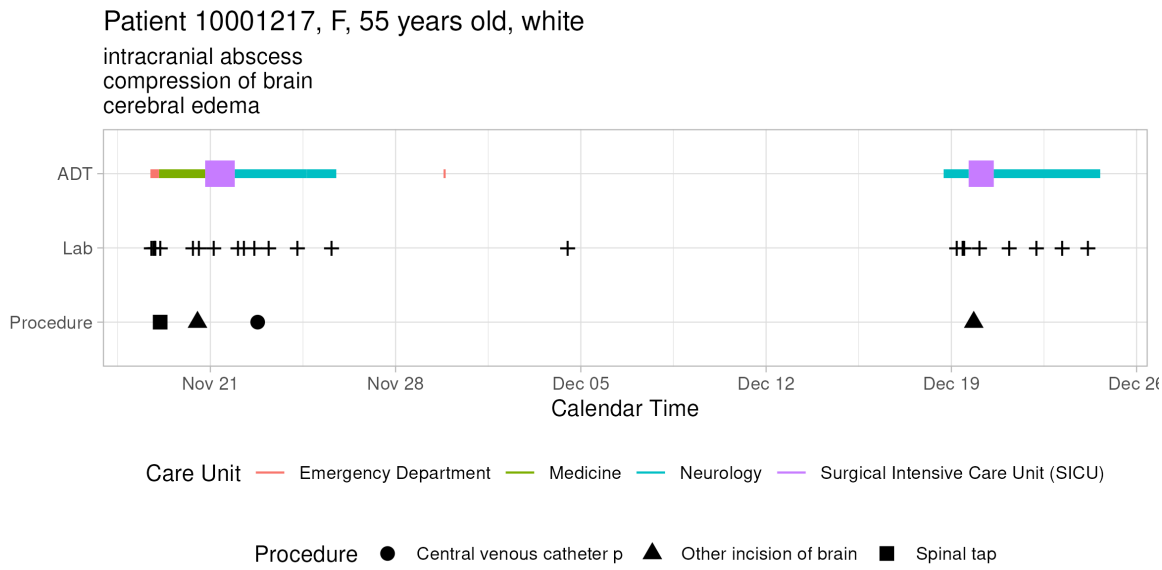
In this exercise, we use tidyverse (ggplot2, dplyr, etc) to explore the MIMIC-IV data introduced in homework 1 and to build a cohort of ICU stays.

## Q1. Visualizing patient trajectory

Visualizing a patient's encounters in a health care system is a common task in clinical data analysis. In this question, we will visualize a patient's ADT (admission-discharge-transfer) history and ICU vitals in the MIMIC-IV data.

### Q1.1 ADT history

A patient's ADT history records the time of admission, discharge, and transfer in the hospital. This figure shows the ADT history of the patient with `subject_id` 10001217 in the MIMIC-IV data. The x-axis is the calendar time, and the y-axis is the type of event (ADT, lab, procedure). The color of the line segment represents the care unit. The size of the line segment represents whether the care unit is an ICU/CCU. The crosses represent lab events, and the shape of the dots represents the type of procedure. The title of the figure shows the patient's demographic information and the subtitle shows top 3 diagnoses.




Do a similar visualization for the patient with `subject_id` 10063848 using ggplot.

Hint: We need to pull information from data files `patients.csv.gz`, `admissions.csv.gz`, `transfers.csv.gz`, `labevents.csv.gz`, `procedures_icd.csv.gz`, `diagnoses_icd.csv.gz`, `d_icd_procedures.csv.gz`, and `d_icd_diagnoses.csv.gz`. For the big file `labevents.csv.gz`, use the Parquet format you generated in Homework 2. For reproducibility, make the Parquet folder `labevents_pq` available at the current working directory `hw3`, for example, by a symbolic link. Make your code reproducible.

**Solution:**

```
library(readr)
library(lubridate)
library(scales)   # for better date formatting on the x-axis

# For reproducibility
set.seed(123)

# ------------------------------
# 1. Read Data Files
# ------------------------------

admissions_tble <- read_csv("~/mimic/hosp/admissions.csv.gz")
transfers <- read_csv("~/mimic/hosp/transfers.csv.gz")

labevents        <- read_csv("~/mimic/hosp/labevents.csv.gz", col_select = c("subject_id", "
```

```
labevents_tble <- open_dataset("labevents_pq",format = "parquet")

procedures_icd   <- read_csv("~/mimic/hosp/procedures_icd.csv.gz")
diagnoses_icd    <- read_csv("~/mimic/hosp/diagnoses_icd.csv.gz")
d_icd_procedures <- read_csv("~/mimic/hosp/d_icd_procedures.csv.gz")
d_icd_diagnoses  <- read_csv("~/mimic/hosp/d_icd_diagnoses.csv.gz")
```

```
library(dplyr)
library(readr)
library(arrow)

#    subject_id
subject_id_target <- 10001217


# patients <- read_csv("patients.csv.gz")
patients <- read_csv("~/mimic/hosp/patients.csv.gz")
```

```
Rows: 364627 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (2): gender, anchor_year_group
dbl  (3): subject_id, anchor_age, anchor_year
date (1): dod

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pt_info <- patients %>%
  select(subject_id, gender, anchor_age) %>%
  filter(subject_id == subject_id_target)

#
print(pt_info)
```

```
# A tibble: 1 x 3
  subject_id gender anchor_age
       <dbl> <chr>       <dbl>
1   10001217 F              55
```

```
admissions <- read_csv("~/mimic/hosp/admissions.csv.gz",
                       col_types = cols(
                         admittime = col_datetime(),
                         dischtime = col_datetime()
                       ))
pt_adm <- admissions %>%
  filter(subject_id == subject_id_target) %>%
  select(subject_id, hadm_id, race) %>%
  distinct(subject_id, .keep_all = TRUE) #keep one record

print(pt_adm)
```

```
# A tibble: 1 x 3
  subject_id  hadm_id race
       <dbl>    <dbl> <chr>
1   10001217 24597018 WHITE
```

```
# ----------------------------
# 3.   transfers.csv.gz
# ----------------------------
# transfers             transfertime
transfers <- read_csv("~/mimic/hosp/transfers.csv.gz",
                      col_types = cols(
                        transfertime = col_datetime()
                      ))
```

```
Warning: The following named parsers don't match the column names: transfertime
```

```
pt_transfers <- transfers %>%
  filter(subject_id == subject_id_target) %>%
  select(subject_id, hadm_id, careunit)

#
print(pt_transfers)
```

```
# A tibble: 12 x 3
   subject_id  hadm_id careunit
        <dbl>    <dbl> <chr>
 1   10001217 24597018 Neurology
 2   10001217 24597018 UNKNOWN
```

```
 3   10001217 24597018 Neurology
 4   10001217 24597018 Medicine
 5   10001217 24597018 Surgical Intensive Care Unit (SICU)
 6   10001217 24597018 Emergency Department
 7   10001217 27703517 Neurology
 8   10001217 27703517 Surgical Intensive Care Unit (SICU)
 9   10001217 27703517 Neurology
10   10001217 27703517 UNKNOWN
11   10001217 27703517 Neurology
12   10001217       NA Emergency Department
```

```
pt_labevents <- open_dataset("labevents_pq", format = "parquet") |>
  # mutate(across(where(is.integer), as.numeric)) |>
  mutate(subject_id = as.numeric(subject_id)) |>
  select(subject_id, hadm_id, storetime) |>
  filter(subject_id == subject_id_target) |>
  collect()

print(pt_labevents, width = Inf)
```

```
# A tibble: 353 x 3
   subject_id hadm_id storetime
        <dbl>   <int> <dttm>
 1   10001217      NA NA
 2   10001217      NA 2157-11-18 11:17:00
 3   10001217      NA 2157-11-18 11:17:00
 4   10001217      NA 2157-11-18 11:17:00
 5   10001217      NA 2157-11-18 11:17:00
 6   10001217      NA 2157-11-18 11:17:00
 7   10001217      NA 2157-11-18 11:17:00
 8   10001217      NA 2157-11-18 11:17:00
 9   10001217      NA 2157-11-18 17:45:00
10   10001217      NA 2157-11-18 11:17:00
# i 343 more rows
```

```
procedures_icd <- read_csv("~/mimic/hosp/procedures_icd.csv.gz")
```

```
Rows: 859655 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (1): icd_code
```

```
dbl  (4): subject_id, hadm_id, seq_num, icd_version
date (1): chartdate

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
pt_procedures <- procedures_icd %>%
  filter(subject_id == subject_id_target) %>%
  select(subject_id, hadm_id, icd_code, chartdate)

print(pt_procedures)
```

```
# A tibble: 4 x 4
  subject_id  hadm_id icd_code chartdate
       <dbl>    <dbl> <chr>    <date>
1   10001217 24597018 0139     2157-11-20
2   10001217 24597018 0331     2157-11-19
3   10001217 24597018 3897     2157-11-22
4   10001217 27703517 0139     2157-12-19
```

```r
# diagnoses_icd  ICD
diagnoses_icd <- read_csv("~/mimic/hosp/d_icd_diagnoses.csv.gz")
```

```
Rows: 112107 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): icd_code, long_title
dbl (1): icd_version

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
pt_diagnoses <- diagnoses_icd %>%
  filter(icd_code %in% pt_procedures$icd_code) %>%
  select(icd_code, long_title)

print(pt_diagnoses)
```

```
# A tibble: 2 x 2
  icd_code long_title
```

```
  <chr>    <chr>
1 0331     Whooping cough due to bordetella parapertussis [B. parapertussis]
2 3897     Deaf, nonspeaking, not elsewhere classifiable
```

```r
combined_data_half <- pt_transfers %>%
  left_join(pt_labevents, by = c("subject_id", "hadm_id")) %>%
  left_join(pt_procedures, by = c("subject_id", "hadm_id"))
```

```
Warning in left_join(., pt_labevents, by = c("subject_id", "hadm_id")): Detected an unexpecte
i Row 1 of `x` matches multiple rows in `y`.
i Row 59 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(., pt_procedures, by = c("subject_id", "hadm_id")): Detected an unexpect
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 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.
```

```r
combined_data <- combined_data_half %>%
  left_join(pt_diagnoses, by = c("icd_code"))
print(combined_data)
```

```
# A tibble: 3,281 x 7
   subject_id  hadm_id careunit  storetime           icd_code chartdate
        <dbl>    <dbl> <chr>     <dttm>              <chr>    <date>
 1   10001217 24597018 Neurology 2157-11-18 19:19:00 0139     2157-11-20
 2   10001217 24597018 Neurology 2157-11-18 19:19:00 0331     2157-11-19
 3   10001217 24597018 Neurology 2157-11-18 19:19:00 3897     2157-11-22
 4   10001217 24597018 Neurology 2157-11-18 19:19:00 0139     2157-11-20
 5   10001217 24597018 Neurology 2157-11-18 19:19:00 0331     2157-11-19
 6   10001217 24597018 Neurology 2157-11-18 19:19:00 3897     2157-11-22
 7   10001217 24597018 Neurology 2157-11-18 20:55:00 0139     2157-11-20
 8   10001217 24597018 Neurology 2157-11-18 20:55:00 0331     2157-11-19
 9   10001217 24597018 Neurology 2157-11-18 20:55:00 3897     2157-11-22
10   10001217 24597018 Neurology 2157-11-18 20:55:00 0139     2157-11-20
# i 3,271 more rows
# i 1 more variable: long_title <chr>
```

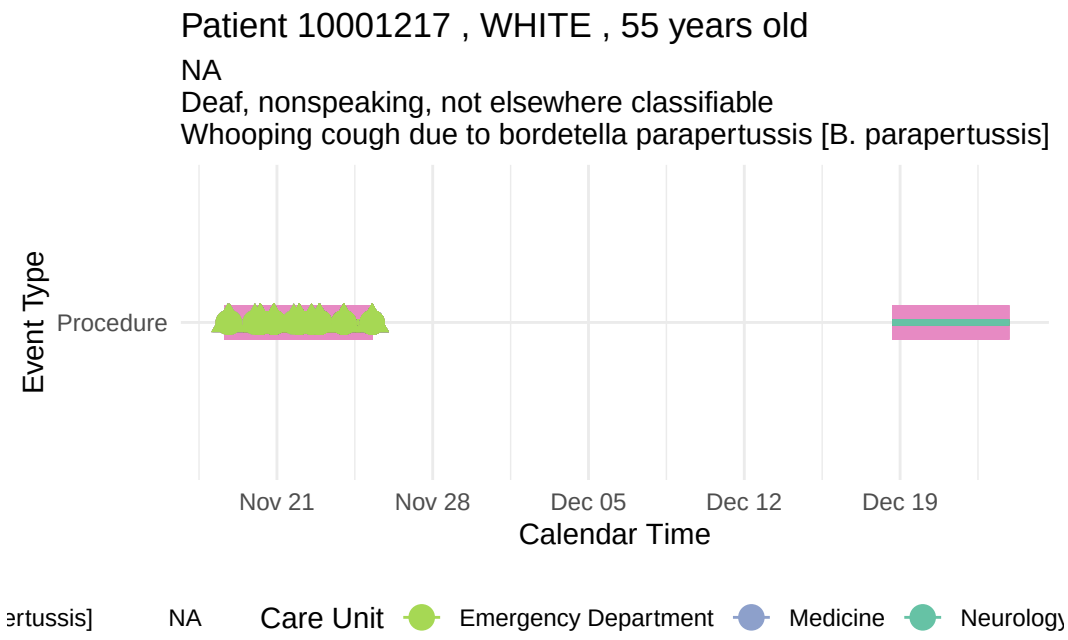
```r
plot_data <- combined_data %>%

  mutate(
    event_time = storetime,
    event_end  = as.POSIXct(chartdate),
    event_type = "Procedure",
    is_icu = ifelse(grepl("ICU|CCU", careunit, ignore.case = TRUE), TRUE, FALSE),
    shape_lable = long_title
  )

unique_careunits <- unique(plot_data$careunit)
care_unit_colors <- setNames(
  RColorBrewer::brewer.pal(n = max(3, length(unique_careunits)), name = "Set2")[1:length(unic
  unique_careunits
)



unique_shapes <- unique(plot_data$shape_lable)

procedure_shapes <- setNames(seq(15, length.out = length(unique_shapes)), unique_shapes)

top3_diagnoses <- plot_data %>%
  count(long_title, sort = TRUE) %>%
  head(3) %>%
  pull(long_title)

ggplot(plot_data, aes(x = event_time, y = event_type, color = careunit)) +

  geom_segment(aes(xend = event_end, yend = event_type, linewidth = ifelse(is_icu, 2, 1))) +

  geom_point(aes(shape = shape_lable), size = 4, na.rm = FALSE, alpha = 1) +
  scale_color_manual(values = care_unit_colors, na.translate = TRUE) +
  scale_shape_manual(values = procedure_shapes, na.translate = TRUE, drop = FALSE) +
  labs(
    title = paste("Patient", subject_id_target, ",", pt_adm$race[1], ",", pt_info$anchor_age
    subtitle = paste(top3_diagnoses, collapse = "\n"),
    x = "Calendar Time",
    y = "Event Type",
    color = "Care Unit",
    shape = "Procedure"
  ) +
  theme_minimal() +
```

```
  theme(legend.position = "bottom")
```

```
Warning: Removed 12 rows containing missing values or values outside the scale range
(`geom_segment()`).
```

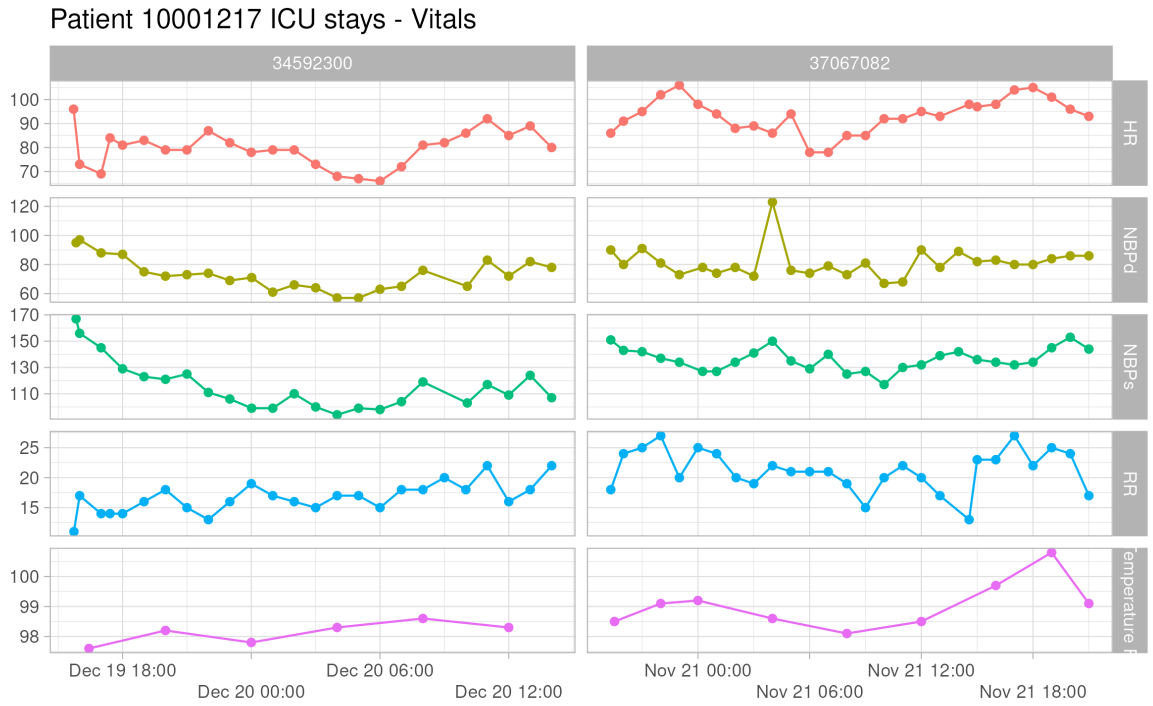
```
Warning: Removed 74 rows containing missing values or values outside the scale range
(`geom_segment()`).
```

```
Warning: Removed 1601 rows containing missing values or values outside the scale range
(`geom_point()`).
```




### Q1.2 ICU stays

ICU stays are a subset of ADT history. This figure shows the vitals of the patient 10001217 during ICU stays. The x-axis is the calendar time, and the y-axis is the value of the vital. The color of the line represents the type of vital. The facet grid shows the abbreviation of the vital and the stay ID.




Do a similar visualization for the patient 10063848.

**Solution:**

```
subject <- 10063848
vitals <- c(220045, 220179, 220180, 223761, 220210)

d_items <- read_csv("~/mimic/icu/d_items.csv.gz") |>
  filter(itemid %in% vitals) %>%
  mutate(itemid = as.integer(itemid))
```

```
Rows: 4095 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (6): label, abbreviation, linksto, category, unitname, param_type
dbl (3): itemid, lownormalvalue, highnormalvalue

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
print(d_items)
```

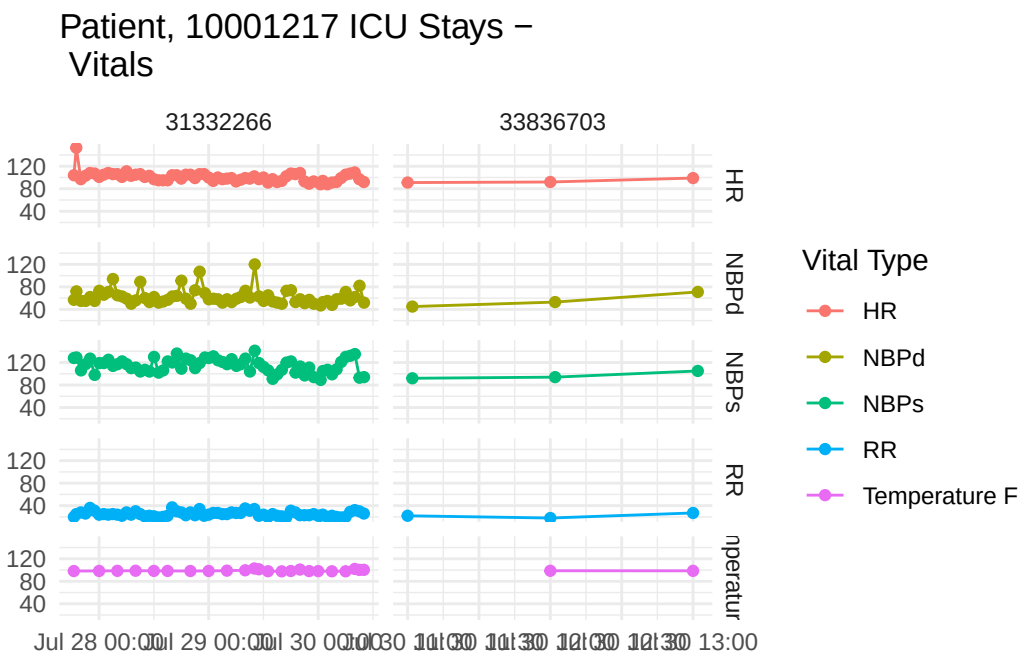
```
# A tibble: 5 x 9
  itemid label  abbreviation linksto category unitname param_type lownormalvalue
   <int> <chr>  <chr>        <chr>   <chr>    <chr>    <chr>                <dbl>
1 220045 Heart~ HR           charte~ Routine~ bpm      Numeric                 NA
2 220179 Non I~ NBPs         charte~ Routine~ mmHg     Numeric                 NA
3 220180 Non I~ NBPd         charte~ Routine~ mmHg     Numeric                 NA
4 220210 Respi~ RR           charte~ Respira~ insp/min Numeric                 NA
5 223761 Tempe~ Temperature~ charte~ Routine~ °F       Numeric                 NA
# i 1 more variable: highnormalvalue <dbl>
```

```
icuvitals <- open_dataset("chartevents_pq", format = "parquet") |>
  to_duckdb() |>
  filter(subject_id %in% subject) |>
  select(subject_id, stay_id, charttime, itemid, valuenum) |>
  left_join(d_items, by = "itemid", copy = TRUE) |>
  filter(itemid %in% vitals) |>
  select(subject_id, stay_id, charttime, itemid, valuenum, abbreviation) |>
  collect()

print(icuvitals)
```

```
# A tibble: 298 x 6
   subject_id  stay_id charttime           itemid valuenum abbreviation
        <dbl>    <dbl> <dttm>               <dbl>    <dbl> <chr>
 1   10063848 31332266 2177-07-27 20:00:00 220045     97   HR
 2   10063848 31332266 2177-07-27 20:02:00 220179    106   NBPs
 3   10063848 31332266 2177-07-27 20:02:00 220180     55   NBPd
 4   10063848 31332266 2177-07-27 20:00:00 220210     28   RR
 5   10063848 33836703 2177-07-30 13:00:00 223761     98.6 Temperature F
 6   10063848 31332266 2177-07-27 19:00:00 220045    153   HR
 7   10063848 31332266 2177-07-27 19:02:00 220179    129   NBPs
 8   10063848 31332266 2177-07-27 19:02:00 220180     72   NBPd
 9   10063848 31332266 2177-07-27 19:00:00 220210     25   RR
10   10063848 33836703 2177-07-30 12:00:00 223761     98.7 Temperature F
# i 288 more rows
```

```
ggplot(data = icuvitals, aes(x = charttime, y = valuenum, color = abbreviation))+
  geom_line() +
```

```
  geom_point() +
  facet_grid(abbreviation ~ stay_id,scale = "free_x") +
  scale_x_datetime(date_labels = "%b %d %H:%M") +
  labs(
    title = paste("Patient,", subject_id_target, "ICU Stays -
Vitals"),
       x = NULL,
       y = NULL,
       color = "Vital Type") +
  theme_minimal()
```




```
  theme(
    legend.position = "none",
    strip.text = element_text(size = 8)
  )
```

```
List of 2
 $ legend.position: chr "none"
 $ strip.text     :List of 11
  ..$ family       : NULL
  ..$ face         : NULL
  ..$ colour       : NULL
  ..$ size         : num 8
```

```
  ..$ hjust        : NULL
  ..$ vjust        : NULL
  ..$ angle        : NULL
  ..$ lineheight   : NULL
  ..$ margin       : NULL
  ..$ debug        : NULL
  ..$ inherit.blank: logi FALSE
  ..- attr(*, "class")= chr [1:2] "element_text" "element"
 - attr(*, "class")= chr [1:2] "theme" "gg"
 - attr(*, "complete")= logi FALSE
 - attr(*, "validate")= logi TRUE
```

## Q2. ICU stays

`icustays.csv.gz` (https://mimic.mit.edu/docs/iv/modules/icu/icustays/) contains data about Intensive Care Units (ICU) stays. The first 10 lines are

```
zcat < ~/mimic/icu/icustays.csv.gz | head
```

```
subject_id,hadm_id,stay_id,first_careunit,last_careunit,intime,outtime,los
10000032,29079034,39553978,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (MI
10000690,25860671,37081114,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (MI
10000980,26913865,39765666,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (MI
10001217,24597018,37067082,Surgical Intensive Care Unit (SICU),Surgical Intensive Care Unit
10001217,27703517,34592300,Surgical Intensive Care Unit (SICU),Surgical Intensive Care Unit
10001725,25563031,31205490,Medical/Surgical Intensive Care Unit (MICU/SICU),Medical/Surgical
10001843,26133978,39698942,Medical/Surgical Intensive Care Unit (MICU/SICU),Medical/Surgical
10001884,26184834,37510196,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (MI
10002013,23581541,39060235,Cardiac Vascular Intensive Care Unit (CVICU),Cardiac Vascular Inte
```

### Q2.1 Ingestion

Import `icustays.csv.gz` as a tibble `icustays_tble`.

```
library(readr)

file_path <- "~/mimic/icu/icustays.csv.gz"

icustays_tble <- read_csv(file_path)
```

```
Rows: 94458 Columns: 8
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (2): first_careunit, last_careunit
dbl  (4): subject_id, hadm_id, stay_id, los
dttm (2): intime, outtime

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(icustays_tble)
```

```
# A tibble: 6 x 8
  subject_id  hadm_id  stay_id first_careunit  last_careunit intime
       <dbl>    <dbl>    <dbl> <chr>           <chr>         <dttm>
1   10000032 29079034 39553978 Medical Intens~ Medical Inte~ 2180-07-23 14:00:00
2   10000690 25860671 37081114 Medical Intens~ Medical Inte~ 2150-11-02 19:37:00
3   10000980 26913865 39765666 Medical Intens~ Medical Inte~ 2189-06-27 08:42:00
4   10001217 24597018 37067082 Surgical Inten~ Surgical Int~ 2157-11-20 19:18:02
5   10001217 27703517 34592300 Surgical Inten~ Surgical Int~ 2157-12-19 15:42:24
6   10001725 25563031 31205490 Medical/Surgic~ Medical/Surg~ 2110-04-11 15:52:22
# i 2 more variables: outtime <dttm>, los <dbl>
```

### Q2.2 Summary and visualization

How many unique `subject_id`? Can a `subject_id` have multiple ICU stays? Summarize the number of ICU stays per `subject_id` by graphs.

**Solution:**

```r
library(dplyr)
library(ggplot2)

# Count the number of unique subject_id
unique_subjects <- icustays_tble %>%
  summarise(unique_subjects = n_distinct(subject_id))
print(unique_subjects)
```

```
# A tibble: 1 x 1
  unique_subjects
            <int>
1           65366
```

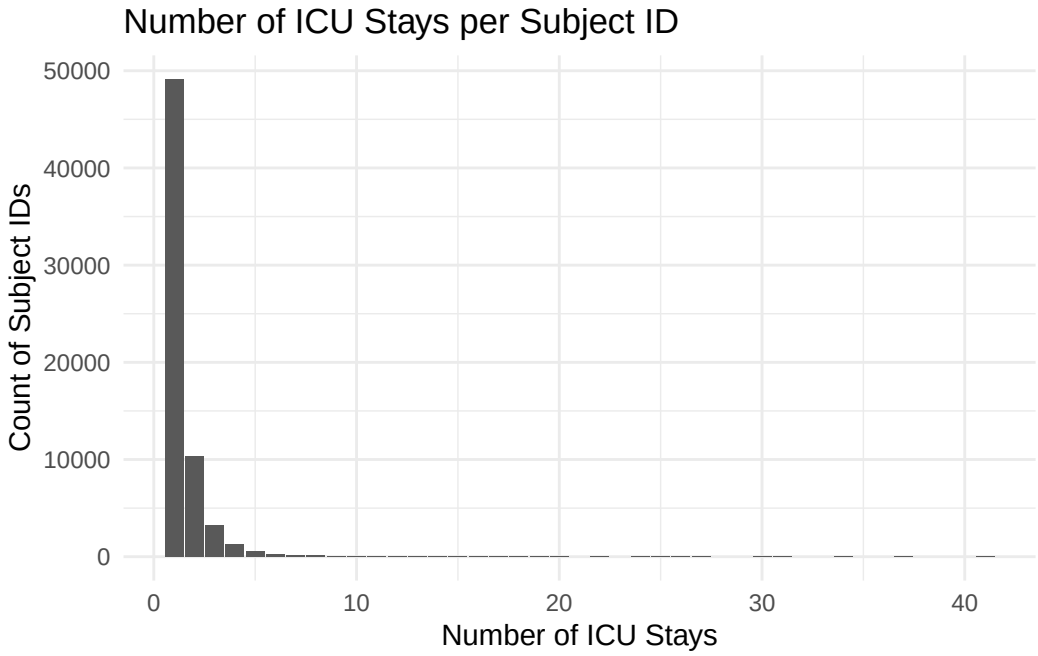
There are 65366 unique `subject_id` in the `icustays_tble`.

```
# Check if a subject_id can have multiple ICU stays
subject_stays <- icustays_tble %>%
  group_by(subject_id) %>%
  summarise(num_stays = n())
print(subject_stays)
```

```
# A tibble: 65,366 x 2
   subject_id num_stays
        <dbl>     <int>
 1   10000032         1
 2   10000690         1
 3   10000980         1
 4   10001217         2
 5   10001725         1
 6   10001843         1
 7   10001884         1
 8   10002013         1
 9   10002114         1
10   10002155         3
# i 65,356 more rows
```

Yes, a `subject_id` can have multiple ICU stays.

```
# Plot the number of ICU stays per subject_id
ggplot(subject_stays, aes(x = num_stays)) +
  geom_bar() +
  labs(title = "Number of ICU Stays per Subject ID",
       x = "Number of ICU Stays",
       y = "Count of Subject IDs") +
  theme_minimal()
```




The graph shows the distribution of the number of ICU stays per `subject_id`. Most patients have only one ICU stay, but there are a few patients with multiple ICU stays.

### Q3. `admissions` data

Information of the patients admitted into hospital is available in `admissions.csv.gz`. See https://mimic.mit.edu/docs/iv/modules/hosp/admissions/ for details of each field in this file. The first 10 lines are

```
zcat < ~/mimic/hosp/admissions.csv.gz | head
```

```
subject_id,hadm_id,admittime,dischtime,deathtime,admission_type,admit_provider_id,admission_l
10000032,22595853,2180-05-06 22:23:00,2180-05-07 17:15:00,,URGENT,P49AFC,TRANSFER FROM HOSPI
10000032,22841357,2180-06-26 18:27:00,2180-06-27 18:49:00,,EW EMER.,P784FA,EMERGENCY ROOM,HO
10000032,25742920,2180-08-05 23:44:00,2180-08-07 17:50:00,,EW EMER.,P19UTS,EMERGENCY ROOM,HOS
10000032,29079034,2180-07-23 12:35:00,2180-07-25 17:55:00,,EW EMER.,P06OTX,EMERGENCY ROOM,HO
10000068,25022803,2160-03-03 23:16:00,2160-03-04 06:26:00,,EU OBSERVATION,P39NWO,EMERGENCY RO
10000084,23052089,2160-11-21 01:56:00,2160-11-25 14:52:00,,EW EMER.,P42H7G,WALK-IN/SELF REFE
10000084,29888819,2160-12-28 05:11:00,2160-12-28 16:07:00,,EU OBSERVATION,P35NE4,PHYSICIAN RE
10000108,27250926,2163-09-27 23:17:00,2163-09-28 09:04:00,,EU OBSERVATION,P40JML,EMERGENCY RO
10000117,22927623,2181-11-15 02:05:00,2181-11-15 14:52:00,,EU OBSERVATION,P47EY8,EMERGENCY RO
```

### Q3.1 Ingestion

Import `admissions.csv.gz` as a tibble `admissions_tble`.

**Solution:**

```
library(readr)

file_path <- "~/mimic/hosp/admissions.csv.gz"

admissions_tble <- read_csv(file_path)
```

```
Rows: 546028 Columns: 16
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (8): admission_type, admit_provider_id, admission_location, discharge_l...
dbl  (3): subject_id, hadm_id, hospital_expire_flag
dttm (5): admittime, dischtime, deathtime, edregtime, edouttime

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(admissions_tble)
```

```
# A tibble: 6 x 16
  subject_id hadm_id admittime           dischtime           deathtime
       <dbl>   <dbl> <dttm>              <dttm>              <dttm>
1   10000032  2.26e7 2180-05-06 22:23:00 2180-05-07 17:15:00 NA
2   10000032  2.28e7 2180-06-26 18:27:00 2180-06-27 18:49:00 NA
3   10000032  2.57e7 2180-08-05 23:44:00 2180-08-07 17:50:00 NA
4   10000032  2.91e7 2180-07-23 12:35:00 2180-07-25 17:55:00 NA
5   10000068  2.50e7 2160-03-03 23:16:00 2160-03-04 06:26:00 NA
6   10000084  2.31e7 2160-11-21 01:56:00 2160-11-25 14:52:00 NA
# i 11 more variables: admission_type <chr>, admit_provider_id <chr>,
#   admission_location <chr>, discharge_location <chr>, insurance <chr>,
#   language <chr>, marital_status <chr>, race <chr>, edregtime <dttm>,
#   edouttime <dttm>, hospital_expire_flag <dbl>
```

### Q3.2 Summary and visualization

Summarize the following information by graphics and explain any patterns you see.

- number of admissions per patient
- admission hour (anything unusual?)
- admission minute (anything unusual?)
- length of hospital stay (from admission to discharge) (anything unusual?)

According to the MIMIC-IV documentation,

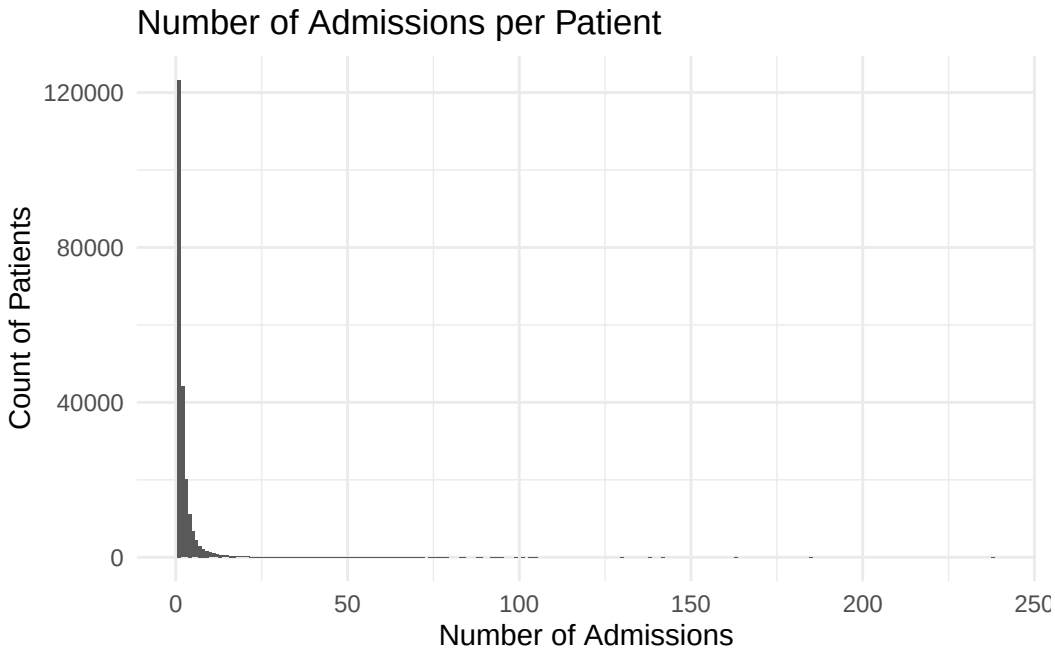
> All dates in the database have been shifted to protect patient confidentiality. Dates will be internally consistent for the same patient, but randomly distributed in the future. Dates of birth which occur in the present time are not true dates of birth. Furthermore, dates of birth which occur before the year 1900 occur if the patient is older than 89. In these cases, the patient's age at their first admission has been fixed to 300.

**Solution:**

```
# 1. Number of admissions per patient
admissions_per_patient <- admissions_tble %>%
  group_by(subject_id) %>%
  summarise(num_admissions = n()) %>%
  arrange(desc(num_admissions))

print(admissions_per_patient)
```

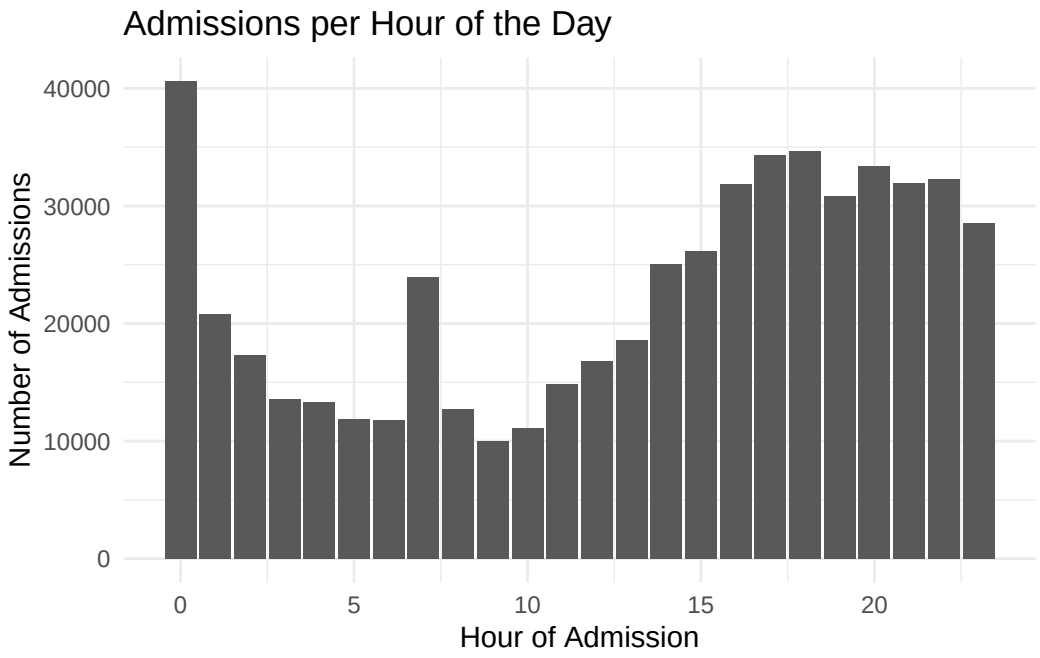
```
# A tibble: 223,452 x 2
   subject_id num_admissions
        <dbl>          <int>
 1   15496609            238
 2   15464144            185
 3   10714009            163
 4   16662316            142
 5   14394983            138
 6   15229574            130
 7   11582633            105
 8   17011846            104
 9   13475033            103
10   11965254            101
# i 223,442 more rows
```

```r
ggplot(admissions_per_patient, aes(x = num_admissions)) +
  geom_bar() +
  labs(title = "Number of Admissions per Patient",
       x = "Number of Admissions",
       y = "Count of Patients") +
  theme_minimal()
```



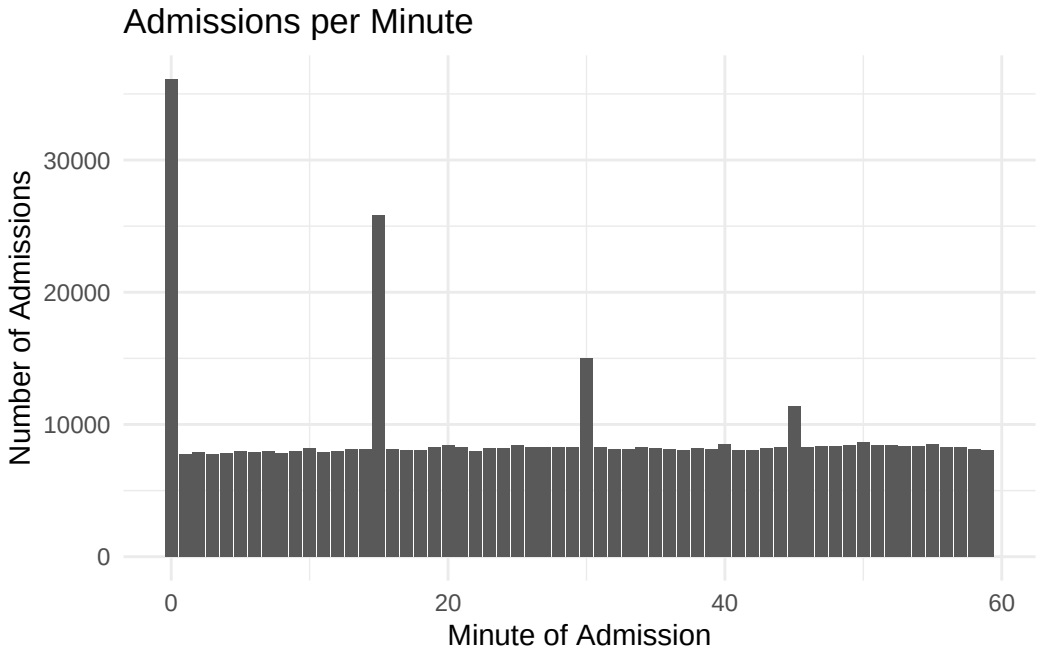

This graph shows the distribution of the number of admissions per patient. The x-axis has a long tail, indicating that most patients have only a few admissions, but there are a small number of patients with many more admissions.

```r
# 2. Admission hour
admissions_tble$admission_hour <- as.numeric(format(as.POSIXct(admissions_tble$admittime), "

# Plot admission hours
ggplot(admissions_tble, aes(x = admission_hour)) +
  geom_bar() +
  labs(title = "Admissions per Hour of the Day",
       x = "Hour of Admission",
       y = "Number of Admissions") +
  theme_minimal()
```



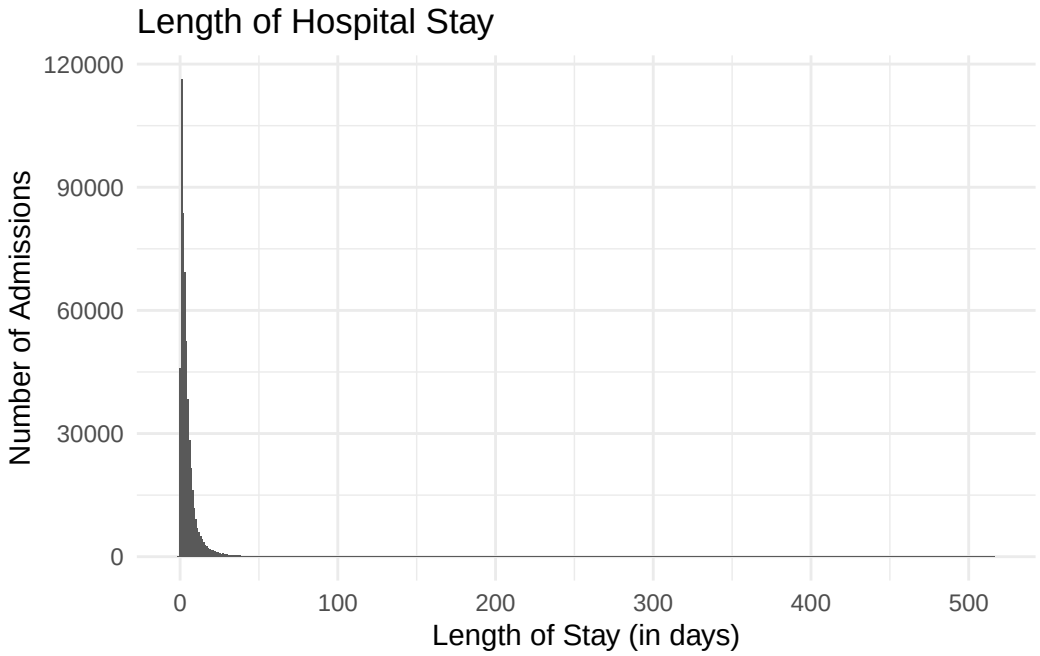

The graph shows the distribution of admissions by hour of the day. There is a peak in admissions around 8 AM, which is likely due to the start of the day shift in hospitals. There are also peaks around 8 PM and 4 AM, which may be due to emergency admissions.

```
# 3. Admission minute
admissions_tble$admission_minute <- as.numeric(format(as.POSIXct(admissions_tble$admittime),

# Plot admission minutes
ggplot(admissions_tble, aes(x = admission_minute)) +
  geom_bar() +
  labs(title = "Admissions per Minute",
       x = "Minute of Admission",
       y = "Number of Admissions") +
  theme_minimal()
```




The graph shows the distribution of admissions by minute of the hour. There is a peak at 0 minutes, which is likely due to the rounding of admission times to the nearest hour. There are also peaks at 30 and 15 minutes, which may be due to the rounding of admission times to the nearest half-hour or quarter-hour.

```
# 4. Length of hospital stay
admissions_tble$admittime <- as.POSIXct(admissions_tble$admittime)
admissions_tble$dischtime <- as.POSIXct(admissions_tble$dischtime)

# Calculate the length of hospital stay in days
admissions_tble$length_of_stay <- as.numeric(difftime(admissions_tble$dischtime, admissions_t

# Plot the length of stay
ggplot(admissions_tble, aes(x = length_of_stay)) +
  geom_histogram(binwidth = 1) +
  labs(title = "Length of Hospital Stay",
       x = "Length of Stay (in days)",
       y = "Number of Admissions") +
  theme_minimal()
```




The graph shows the distribution of the length of hospital stay. Most patients have a short length of stay, but there are a small number of patients with a very long length of stay. This is likely due to the fact that the MIMIC-IV database contains data from a large number of patients, some of whom may have had very long hospital stays.

### Q4. `patients` data

Patient information is available in `patients.csv.gz`. See https://mimic.mit.edu/docs/iv/modules/hosp/patients/ for details of each field in this file. The first 10 lines are

```
zcat < ~/mimic/hosp/patients.csv.gz | head
```

```
subject_id,gender,anchor_age,anchor_year,anchor_year_group,dod
10000032,F,52,2180,2014 - 2016,2180-09-09
10000048,F,23,2126,2008 - 2010,
10000058,F,33,2168,2020 - 2022,
10000068,F,19,2160,2008 - 2010,
10000084,M,72,2160,2017 - 2019,2161-02-13
10000102,F,27,2136,2008 - 2010,
10000108,M,25,2163,2014 - 2016,
10000115,M,24,2154,2017 - 2019,
10000117,F,48,2174,2008 - 2010,
```

### Q4.1 Ingestion

Import `patients.csv.gz` (https://mimic.mit.edu/docs/iv/modules/hosp/patients/) as a tibble `patients_tble`.

```
library(readr)

file_path <- "~/mimic//hosp/patients.csv.gz"

patients_tble <- read_csv(file_path)
```

```
Rows: 364627 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (2): gender, anchor_year_group
dbl  (3): subject_id, anchor_age, anchor_year
date (1): dod

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(patients_tble)
```

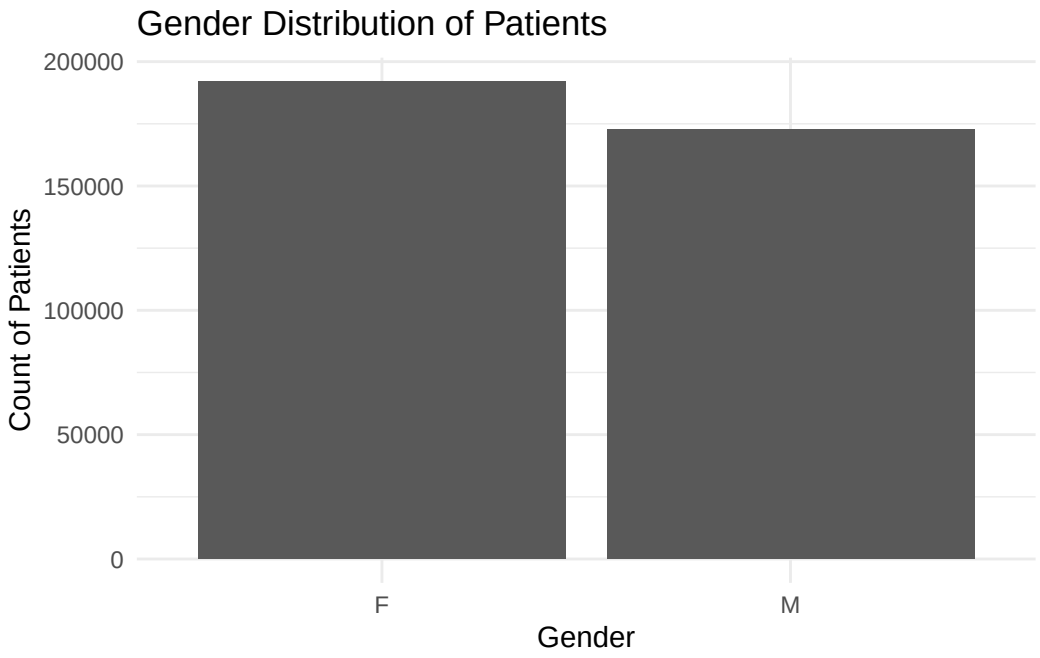
```
# A tibble: 6 x 6
  subject_id gender anchor_age anchor_year anchor_year_group dod
       <dbl> <chr>       <dbl>       <dbl> <chr>              <date>
1   10000032 F              52        2180 2014 - 2016       2180-09-09
2   10000048 F              23        2126 2008 - 2010       NA
3   10000058 F              33        2168 2020 - 2022       NA
4   10000068 F              19        2160 2008 - 2010       NA
5   10000084 M              72        2160 2017 - 2019       2161-02-13
6   10000102 F              27        2136 2008 - 2010       NA
```

### Q4.2 Summary and visualization

Summarize variables `gender` and `anchor_age` by graphics, and explain any patterns you see.

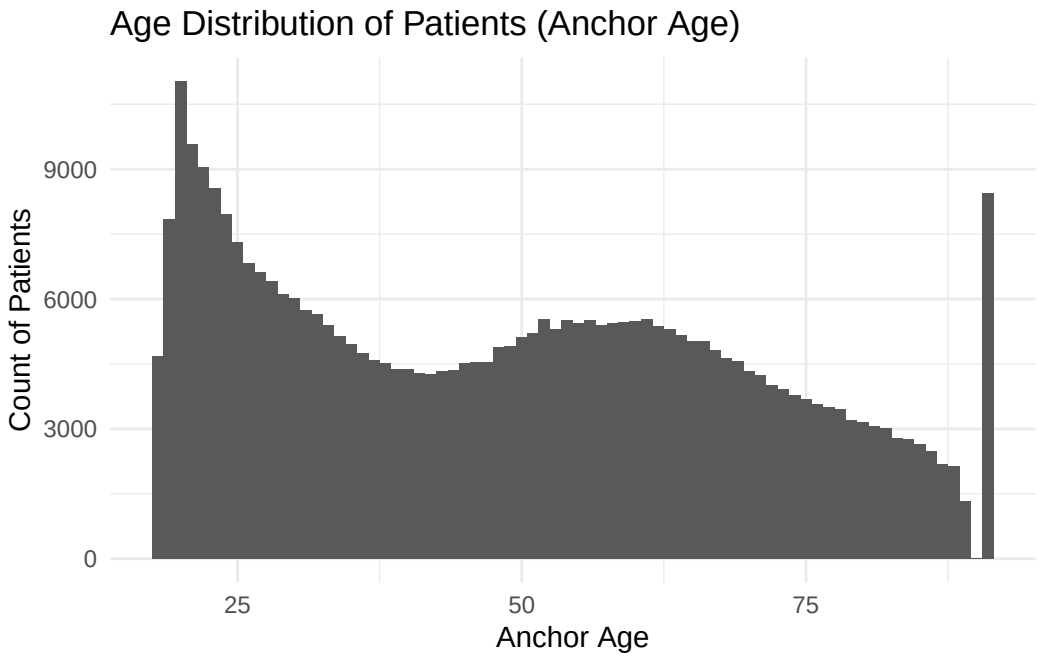
```
# Summary of gender (distribution of male and female patients)
ggplot(patients_tble, aes(x = gender)) +
  geom_bar() +
```

```
  labs(title = "Gender Distribution of Patients",
       x = "Gender",
       y = "Count of Patients") +
  theme_minimal()
```




This graph shows the distribution of gender among patients. Female patients are slightly more than male patients.

```
# Summary of anchor_age (age distribution of patients)
ggplot(patients_tble, aes(x = anchor_age)) +
  geom_histogram(binwidth = 1) +
  labs(title = "Age Distribution of Patients (Anchor Age)",
       x = "Anchor Age",
       y = "Count of Patients") +
  theme_minimal()
```




This graph shows the distribution of age among patients. Most patients are between 50 and 80 years old, with a peak around 60 years old and 20 years old. There are very few patients over 90 years old.

## Q5. Lab results

`labevents.csv.gz` (https://mimic.mit.edu/docs/iv/modules/hosp/labevents/) contains all laboratory measurements for patients. The first 10 lines are

```
zcat < ~/mimic/hosp/labevents.csv.gz | head
```

```
labevent_id,subject_id,hadm_id,specimen_id,itemid,order_provider_id,charttime,storetime,value
1,10000032,,2704548,50931,P69FQC,2180-03-23 11:51:00,2180-03-23 15:56:00,___,95,mg/dL,70,100
2,10000032,,36092842,51071,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,
3,10000032,,36092842,51074,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,
4,10000032,,36092842,51075,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,"
5,10000032,,36092842,51079,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,
6,10000032,,36092842,51087,P69FQC,2180-03-23 11:51:00,,,,,,,ROUTINE,RANDOM.
7,10000032,,36092842,51089,P69FQC,2180-03-23 11:51:00,2180-03-23 16:15:00,,,,,,ROUTINE,PRESU
8,10000032,,36092842,51090,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,MI
9,10000032,,36092842,51092,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,"(
```

`d_labitems.csv.gz` (https://mimic.mit.edu/docs/iv/modules/hosp/d_labitems/) is the dictionary of lab measurements.

```
zcat < ~/mimic/hosp/d_labitems.csv.gz | head
```

```
itemid,label,fluid,category
50801,Alveolar-arterial Gradient,Blood,Blood Gas
50802,Base Excess,Blood,Blood Gas
50803,"Calculated Bicarbonate, Whole Blood",Blood,Blood Gas
50804,Calculated Total CO2,Blood,Blood Gas
50805,Carboxyhemoglobin,Blood,Blood Gas
50806,"Chloride, Whole Blood",Blood,Blood Gas
50808,Free Calcium,Blood,Blood Gas
50809,Glucose,Blood,Blood Gas
50810,"Hematocrit, Calculated",Blood,Blood Gas
```

We are interested in the lab measurements of creatinine (50912), potassium (50971), sodium (50983), chloride (50902), bicarbonate (50882), hematocrit (51221), white blood cell count (51301), and glucose (50931). Retrieve a subset of `labevents.csv.gz` that only containing these items for the patients in `icustays_tble`. Further restrict to the last available measurement (by `storetime`) before the ICU stay. The final `labevents_tble` should have one row per ICU stay and columns for each lab measurement.

```
> labevents_tble
# A tibble: 88,086 × 10
   subject_id  stay_id bicarbonate chloride creatinine glucose potassium sodium hematocrit   wbc
        <dbl>    <dbl>       <dbl>    <dbl>      <dbl>   <dbl>     <dbl>  <dbl>       <dbl> <dbl>
 1   10000032 39553978          25       95        0.7     102       6.7    126        41.1   6.9
 2   10000690 37081114          26      100        1        85       4.8    137        36.1   7.1
 3   10000980 39765666          21      109        2.3      89       3.9    144        27.3   5.3
 4   10001217 34592300          30      104        0.5      87       4.1    142        37.4   5.4
 5   10001217 37067082          22      108        0.6     112       4.2    142        38.1  15.7
 6   10001725 31205490          NA       98       NA        NA       4.1    139        NA    NA
 7   10001843 39698942          28       97        1.3     131       3.9    138        31.4  10.4
 8   10001884 37510196          30       88        1.1     141       4.5    130        39.7  12.2
 9   10002013 39060235          24      102        0.9     288       3.5    137        34.9   7.2
10   10002114 34672098          18       NA        3.1      95       6.5    125        34.3  16.8
# ℹ 88,076 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Hint: Use the Parquet format you generated in Homework 2. For reproducibility, make `labevents_pq` folder available at the current working directory `hw3`, for example, by a symbolic link.

**Solution**

```
library(readr)
library(dplyr)


# Specify the itemids for the required lab tests
dlabitems_tble <-
  read_csv("~/mimic/hosp/d_labitems.csv.gz", show_col_types = FALSE) %>%
  filter(itemid %in% c(
    50912,
    50971,
    50983,
    50902,
    50882,
    51221,
    51301,
    50931
    )) %>%
  mutate(itemid = as.integer(itemid)) %>%
  print(width = Inf)
```

```
# A tibble: 8 x 4
  itemid label             fluid category
   <int> <chr>             <chr> <chr>
1  50882 Bicarbonate       Blood Chemistry
2  50902 Chloride          Blood Chemistry
3  50912 Creatinine        Blood Chemistry
4  50931 Glucose           Blood Chemistry
5  50971 Potassium         Blood Chemistry
6  50983 Sodium            Blood Chemistry
7  51221 Hematocrit        Blood Hematology
8  51301 White Blood Cells Blood Hematology
```

```
# load necessary libraries
library(dplyr)
library(readr)
library(lubridate)
library(data.table)
```

```
Attaching package: 'data.table'
```

```
The following objects are masked from 'package:lubridate':

    hour, isoweek, mday, minute, month, quarter, second, wday, week,
    yday, year

The following objects are masked from 'package:dplyr':

    between, first, last

The following object is masked from 'package:purrr':

    transpose

The following object is masked from 'package:pryr':

    address
```

```
# Load the icustays data
icustays_file_path <- "~/mimic/icu/icustays.csv.gz"
icustays_tble <- read_csv(icustays_file_path)
```

```
Rows: 94458 Columns: 8

-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (2): first_careunit, last_careunit
dbl  (4): subject_id, hadm_id, stay_id, los
dttm (2): intime, outtime

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
icustays_tble <- icustays_tble %>%
  mutate(subject_id = as.integer(subject_id))
```

```
# Load the labevents data
labevents_tble <- open_dataset("labevents_pq",format = "parquet") |>
  # creat a virtual table in DuckDB
  to_duckdb() |>
  # only variables subject_id, itemid, storetime, valuenum are needed
```

```
  select(subject_id, itemid, storetime, valuenum) |>
  # restrict to itemid of interest
  filter(itemid %in% dlabitems_tble$itemid) |>
  # put in the intime of ICU stays, copy = Ture to join data from diff src
  left_join(
    select(icustays_tble, subject_id, stay_id, intime),
    by = c("subject_id"),
    copy = TRUE
    ) |>
  # only keep lab items available before this ICU stay
  filter(storetime < intime) |>
  # group by itemid
  group_by(subject_id, stay_id, itemid) |>
  # only keep the last storetime for each item before intime
  slice_max(storetime, n = 1) |>
  # do not need storetime and intime anymore
  select(-storetime, -intime) |>
  ungroup() |>
  pivot_wider(names_from = itemid, values_from = valuenum) |>
  #more informative column names
  rename_at(
    vars(as.character(dlabitems_tble$itemid)),
    ~str_to_lower(dlabitems_tble$label)
    ) |>
  rename(wbc = `white blood cells`) |>
  #force computation
  show_query() |>
  collect() |>
  # sort for grading purpose
  arrange(subject_id, stay_id) |>
  relocate(subject_id, stay_id) |>
  print(width = Inf)
```

```
<SQL>
SELECT
  subject_id,
  stay_id,
  MAX(CASE WHEN (itemid = 50902.0) THEN valuenum END) AS chloride,
  MAX(CASE WHEN (itemid = 51301.0) THEN valuenum END) AS wbc,
  MAX(CASE WHEN (itemid = 50971.0) THEN valuenum END) AS potassium,
  MAX(CASE WHEN (itemid = 50983.0) THEN valuenum END) AS sodium,
  MAX(CASE WHEN (itemid = 50882.0) THEN valuenum END) AS bicarbonate,
```

```
  MAX(CASE WHEN (itemid = 50931.0) THEN valuenum END) AS glucose,
  MAX(CASE WHEN (itemid = 51221.0) THEN valuenum END) AS hematocrit,
  MAX(CASE WHEN (itemid = 50912.0) THEN valuenum END) AS creatinine
FROM (
  SELECT subject_id, itemid, valuenum, stay_id
  FROM (
    SELECT
      q01.*,
      RANK() OVER (PARTITION BY subject_id, stay_id, itemid ORDER BY storetime DESC) AS col0
    FROM (
      SELECT LHS.*, stay_id, intime
      FROM (
        SELECT subject_id, itemid, storetime, valuenum
        FROM arrow_002
        WHERE (itemid IN (50882, 50902, 50912, 50931, 50971, 50983, 51221, 51301))
      ) LHS
      LEFT JOIN dbplyr_tp5oHqrYHg
        ON (LHS.subject_id = dbplyr_tp5oHqrYHg.subject_id)
    ) q01
    WHERE (storetime < intime)
  ) q01
  WHERE (col01 <= 1)
) q01
GROUP BY subject_id, stay_id
# A tibble: 88,086 x 10
   subject_id  stay_id chloride   wbc potassium sodium bicarbonate glucose
        <dbl>    <dbl>    <dbl> <dbl>     <dbl>  <dbl>       <dbl>   <dbl>
 1   10000032 39553978       95   6.9       6.7    126          25     102
 2   10000690 37081114      100   7.1       4.8    137          26      85
 3   10000980 39765666      109   5.3       3.9    144          21      89
 4   10001217 34592300      104   5.4       4.1    142          30      87
 5   10001217 37067082      108  15.7       4.2    142          22     112
 6   10001725 31205490       98  NA         4.1    139          NA      NA
 7   10001843 39698942       97  10.4       3.9    138          28     131
 8   10001884 37510196       88  12.2       4.5    130          30     141
 9   10002013 39060235      102   7.2       3.5    137          24     288
10   10002114 34672098       NA  16.8       6.5    125          18      95
   hematocrit creatinine
        <dbl>      <dbl>
 1       41.1        0.7
 2       36.1        1
 3       27.3        2.3
 4       37.4        0.5
```

```
 5       38.1        0.6
 6       NA         NA
 7       31.4        1.3
 8       39.7        1.1
 9       34.9        0.9
10       34.3        3.1
# i 88,076 more rows
```

### Q6. Vitals from charted events

`chartevents.csv.gz` (https://mimic.mit.edu/docs/iv/modules/icu/chartevents/) contains all the charted data available for a patient. During their ICU stay, the primary repository of a patient's information is their electronic chart. The `itemid` variable indicates a single measurement type in the database. The `value` variable is the value measured for `itemid`. The first 10 lines of `chartevents.csv.gz` are

```
zcat < ~/mimic/icu/chartevents.csv.gz | head
```

```
subject_id,hadm_id,stay_id,caregiver_id,charttime,storetime,itemid,value,valuenum,valueuom,wa
10000032,29079034,39553978,18704,2180-07-23 12:36:00,2180-07-23 14:45:00,226512,39.4,39.4,kg
10000032,29079034,39553978,18704,2180-07-23 12:36:00,2180-07-23 14:45:00,226707,60,60,Inch,0
10000032,29079034,39553978,18704,2180-07-23 12:36:00,2180-07-23 14:45:00,226730,152,152,cm,0
10000032,29079034,39553978,18704,2180-07-23 14:00:00,2180-07-23 14:18:00,220048,SR (Sinus Rhy
10000032,29079034,39553978,18704,2180-07-23 14:00:00,2180-07-23 14:18:00,224642,Oral,,,0
10000032,29079034,39553978,18704,2180-07-23 14:00:00,2180-07-23 14:18:00,224650,None,,,0
10000032,29079034,39553978,18704,2180-07-23 14:00:00,2180-07-23 14:20:00,223761,98.7,98.7,°F
10000032,29079034,39553978,18704,2180-07-23 14:11:00,2180-07-23 14:17:00,220179,84,84,mmHg,0
10000032,29079034,39553978,18704,2180-07-23 14:11:00,2180-07-23 14:17:00,220180,48,48,mmHg,0
```

`d_items.csv.gz` (https://mimic.mit.edu/docs/iv/modules/icu/d_items/) is the dictionary for the `itemid` in `chartevents.csv.gz`.

```
zcat < ~/mimic/icu/d_items.csv.gz | head
```

```
itemid,label,abbreviation,linksto,category,unitname,param_type,lownormalvalue,highnormalvalue
220001,Problem List,Problem List,chartevents,General,,Text,,
220003,ICU Admission date,ICU Admission date,datetimeevents,ADT,,Date and time,,
220045,Heart Rate,HR,chartevents,Routine Vital Signs,bpm,Numeric,,
220046,Heart rate Alarm - High,HR Alarm - High,chartevents,Alarms,bpm,Numeric,,
220047,Heart Rate Alarm - Low,HR Alarm - Low,chartevents,Alarms,bpm,Numeric,,
```

```
220048,Heart Rhythm,Heart Rhythm,chartevents,Routine Vital Signs,,Text,,
220050,Arterial Blood Pressure systolic,ABPs,chartevents,Routine Vital Signs,mmHg,Numeric,90
220051,Arterial Blood Pressure diastolic,ABPd,chartevents,Routine Vital Signs,mmHg,Numeric,6(
220052,Arterial Blood Pressure mean,ABPm,chartevents,Routine Vital Signs,mmHg,Numeric,,
```

We are interested in the vitals for ICU patients: heart rate (220045), systolic non-invasive blood pressure (220179), diastolic non-invasive blood pressure (220180), body temperature in Fahrenheit (223761), and respiratory rate (220210). Retrieve a subset of `chartevents.csv.gz` only containing these items for the patients in `icustays_tble`. Further restrict to the first vital measurement within the ICU stay. The final `chartevents_tble` should have one row per ICU stay and columns for each vital measurement.

```
> chartevents_tble
# A tibble: 94,424 x 7
   subject_id  stay_id heart_rate non_invasive_blood_pressure_systolic non_invasive_blood_pressure_diastolic respiratory_rate temperature_fahrenheit
        <int>    <dbl>      <dbl>                                <dbl>                                 <dbl>           <dbl>                 <dbl>
 1   10000032 39553978         91                                   84                                    48              24                  98.7
 2   10000690 37081114         79                                  107                                    63              23                  97.7
 3   10000980 39765666         77                                  150                                    77              23                  98
 4   10001217 34592300         96                                  167                                    95              11                  97.6
 5   10001217 37067082         86                                  151                                    90              18                  98.5
 6   10001725 31205490         55                                   73                                    56              19                  97.7
 7   10001843 39698942        118                                  112                                    71              17                  97.9
 8   10001884 37510196         38                                  180                                    12              10                  98.1
 9   10002013 39060235         80                                  104                                    70              14                  97.2
10   10002114 34672098        105                                  104                                    81              22                  97.9
# ℹ 94,414 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Hint: Use the Parquet format you generated in Homework 2. For reproducibility, make `chartevents_pq` folder available at the current working directory, for example, by a symbolic link.

### Solution

```
# Specify the itemids for the required lab tests
d_items <-
  read_csv("~/mimic/icu/d_items.csv.gz", show_col_types = FALSE) %>%
  filter(itemid %in% c(
    220045,
    220179,
    220180,
    223761,
    220210
    )) %>%
  mutate(itemid = as.integer(itemid)) %>%
  print(width = Inf)
```

```
# A tibble: 5 x 9
```

```
  itemid label                                    abbreviation  linksto
   <int> <chr>                                    <chr>         <chr>
1 220045 Heart Rate                               HR            chartevents
2 220179 Non Invasive Blood Pressure systolic     NBPs          chartevents
3 220180 Non Invasive Blood Pressure diastolic    NBPd          chartevents
4 220210 Respiratory Rate                         RR            chartevents
5 223761 Temperature Fahrenheit                   Temperature F chartevents
  category            unitname param_type lownormalvalue highnormalvalue
  <chr>               <chr>    <chr>               <dbl>           <dbl>
1 Routine Vital Signs bpm      Numeric                NA              NA
2 Routine Vital Signs mmHg     Numeric                NA              NA
3 Routine Vital Signs mmHg     Numeric                NA              NA
4 Respiratory         insp/min Numeric                NA              NA
5 Routine Vital Signs °F       Numeric                NA              NA
```

```
chartevents_tble <- open_dataset("chartevents_pq",
                            format = "parquet") |>
  to_duckdb() |>
  select(subject_id, itemid, storetime, valuenum) |>
  filter(itemid %in%d_items$itemid) %>%
  mutate(subject_id = as.integer(subject_id)) %>%
  left_join(
    select(icustays_tble, subject_id, stay_id, intime, outtime),
    by = c("subject_id"),
    copy = TRUE
    ) |>
  filter(storetime > intime & storetime < outtime) |>
  group_by(subject_id, stay_id, itemid) |>
  slice_min(storetime, n = 1) |>
  select(-storetime, -intime, -outtime) |>
  ungroup() |>
  pivot_wider(names_from = itemid, values_from = valuenum) |>
  rename_at(
    vars(as.character(d_items$itemid)),
    ~str_to_lower(d_items$label)
    ) |>
  show_query() |>
  collect() |>
  arrange(subject_id, stay_id) |>
  relocate(subject_id, stay_id) |>
  print(width = Inf)
```

```
<SQL>
SELECT
  subject_id,
  stay_id,
  MAX(CASE WHEN (itemid = 220179.0) THEN valuenum END) AS "non invasive blood pressure systol
  MAX(CASE WHEN (itemid = 220180.0) THEN valuenum END) AS "non invasive blood pressure diasto
  MAX(CASE WHEN (itemid = 223761.0) THEN valuenum END) AS "temperature fahrenheit",
  MAX(CASE WHEN (itemid = 220210.0) THEN valuenum END) AS "respiratory rate",
  MAX(CASE WHEN (itemid = 220045.0) THEN valuenum END) AS "heart rate"
FROM (
  SELECT subject_id, itemid, valuenum, stay_id
  FROM (
    SELECT
      q01.*,
      RANK() OVER (PARTITION BY subject_id, stay_id, itemid ORDER BY storetime) AS col01
    FROM (
      SELECT LHS.*, stay_id, intime, outtime
      FROM (
        SELECT
          CAST(subject_id AS INTEGER) AS subject_id,
          itemid,
          storetime,
          valuenum
        FROM arrow_003
        WHERE (itemid IN (220045, 220179, 220180, 220210, 223761))
      ) LHS
      LEFT JOIN dbplyr_VzfNM4MSLU
        ON (LHS.subject_id = dbplyr_VzfNM4MSLU.subject_id)
    ) q01
    WHERE (storetime > intime AND storetime < outtime)
  ) q01
  WHERE (col01 <= 1)
) q01
GROUP BY subject_id, stay_id
# A tibble: 94,364 x 7
   subject_id  stay_id `non invasive blood pressure systolic`
        <int>    <dbl>                                 <dbl>
 1   10000032 39553978                                    84
 2   10000690 37081114                                   107
 3   10000980 39765666                                   158
 4   10001217 34592300                                   167
 5   10001217 37067082                                   151
 6   10001725 31205490                                    73
```

```
 7   10001843 39698942                                  112
 8   10001884 37510196                                  180
 9   10002013 39060235                                  104
10   10002114 34672098                                  112
   `non invasive blood pressure diastolic` `temperature fahrenheit`
                                     <dbl>                     <dbl>
 1                                      48                      98.7
 2                                      63                      97.7
 3                                     127                      98
 4                                      97                      97.6
 5                                      90                      98.5
 6                                      56                      97.7
 7                                      85                      97.9
 8                                      49                      98.1
 9                                      70                      97.2
10                                      80                      97.9
   `respiratory rate` `heart rate`
                <dbl>        <dbl>
 1                 24           91
 2                 27           80
 3                 24           77
 4                 17           96
 5                 18           86
 6                 19           86
 7                 17          131
 8                 16           60
 9                 14           80
10                 22          111
# i 94,354 more rows
```

## Q7. Putting things together

Let us create a tibble `mimic_icu_cohort` for all ICU stays, where rows are all ICU stays of adults (age at `intime` >= 18) and columns contain at least following variables

- all variables in `icustays_tble`
- all variables in `admissions_tble`
- all variables in `patients_tble`
- the last lab measurements before the ICU stay in `labevents_tble`
- the first vital measurements during the ICU stay in `chartevents_tble`

The final `mimic_icu_cohort` should have one row per ICU stay and columns for each variable.

```
> mimic_icu_cohort
# A tibble: 94,458 × 41
   subject_id  hadm_id  stay_id first_careunit        last_careunit intime              outtime              los admittime           dischtime           deathtime
        <dbl>    <dbl>    <dbl> <chr>                 <chr>         <dttm>              <dttm>             <dbl> <dttm>              <dttm>              <dttm>
 1   10000032 29079034 39553978 Medical Intensive Car… Medical Inte… 2180-07-23 14:00:00 2180-07-23 23:50:47 0.410 2180-07-23 12:35:00 2180-07-25 17:55:00 NA
 2   10000690 25860671 37081114 Medical Intensive Car… Medical Inte… 2150-11-02 19:37:00 2150-11-06 17:03:17 3.89  2150-11-02 18:02:00 2150-11-12 13:45:00 NA
 3   10000980 26913865 39765666 Medical Intensive Car… Medical Inte… 2189-06-27 08:42:00 2189-06-27 20:38:27 0.498 2189-06-27 07:38:00 2189-07-03 03:00:00 NA
 4   10001217 24597018 37067082 Surgical Intensive Ca… Surgical Int… 2157-11-20 19:18:02 2157-11-21 22:08:00 1.12  2157-11-18 22:56:00 2157-11-25 18:00:00 NA
 5   10001217 27703517 34592300 Surgical Intensive Ca… Surgical Int… 2157-12-19 15:42:24 2157-12-20 14:27:41 0.948 2157-12-18 16:58:00 2157-12-24 14:55:00 NA
 6   10001725 25563031 31205490 Medical/Surgical Inte… Medical/Surg… 2110-04-11 15:52:22 2110-04-12 23:59:56 1.34  2110-04-11 15:08:00 2110-04-14 15:00:00 NA
 7   10001843 26133978 39698942 Medical/Surgical Inte… Medical/Surg… 2134-12-05 18:50:03 2134-12-06 14:38:26 0.825 2134-12-05 00:10:00 2134-12-06 12:54:00 2134-12-06 12:54:00
 8   10001884 26184834 37510196 Medical Intensive Car… Medical Inte… 2131-01-11 04:20:05 2131-01-20 08:27:30 9.17  2131-01-07 20:39:00 2131-01-20 05:15:00 2131-01-20 05:15:00
 9   10002013 23581541 39060235 Cardiac Vascular Inte… Cardiac Vasc… 2160-05-18 10:00:53 2160-05-19 17:33:33 1.31  2160-05-18 07:45:00 2160-05-23 13:30:00 NA
10   10002114 27793700 34672098 Coronary Care Unit (C… Coronary Car… 2162-02-17 23:30:00 2162-02-20 21:16:27 2.91  2162-02-17 22:32:00 2162-03-04 15:16:00 NA
# ℹ 94,448 more rows
# ℹ 30 more variables: admission_type <chr>, admit_provider_id <chr>, admission_location <chr>, discharge_location <chr>, insurance <chr>, language <chr>,
#   marital_status <chr>, race <chr>, edregtime <dttm>, edouttime <dttm>, hospital_expire_flag <dbl>, gender <chr>, anchor_age <dbl>, anchor_year <dbl>,
#   anchor_year_group <chr>, dod <date>, bicarbonate <dbl>, chloride <dbl>, creatinine <dbl>, glucose <dbl>, potassium <dbl>, sodium <dbl>, hematocrit <dbl>, wbc <dbl>,
#   heart_rate <dbl>, non_invasive_blood_pressure_systolic <dbl>, non_invasive_blood_pressure_diastolic <dbl>, respiratory_rate <dbl>, temperature_fahrenheit <dbl>,
#   age_intime <dbl>
# ℹ Use `print(n = ...)` to see more rows
```

```
library(tidyr)
library(lubridate)
library(arrow)

patients_tble <- read_csv("~/mimic/hosp/patients.csv.gz")
```

```
Rows: 364627 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (2): gender, anchor_year_group
dbl  (3): subject_id, anchor_age, anchor_year
date (1): dod

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
admissions_tble <- read_csv("~/mimic/hosp/admissions.csv.gz")
```

```
Rows: 546028 Columns: 16
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (8): admission_type, admit_provider_id, admission_location, discharge_l...
dbl  (3): subject_id, hadm_id, hospital_expire_flag
dttm (5): admittime, dischtime, deathtime, edregtime, edouttime

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
icu_cohort <- icustays_tble %>%
  left_join(patients_tble, by = "subject_id") %>%
  filter(anchor_age >= 18)

icu_cohort <- icu_cohort %>%
  left_join(admissions_tble, by = c ("subject_id", "hadm_id"))


icu_cohort <- icu_cohort %>%
  left_join(chartevents_tble, by = c ("subject_id", "stay_id"))


mimic_icu_cohort <- icu_cohort %>%
  left_join(labevents_tble,c ("subject_id", "stay_id"))

mimic_icu_cohort <- mimic_icu_cohort

print(head(mimic_icu_cohort))
```

```
# A tibble: 6 x 40
  subject_id  hadm_id  stay_id first_careunit  last_careunit intime
       <dbl>    <dbl>    <dbl> <chr>           <chr>         <dttm>
1   10000032 29079034 39553978 Medical Intens~ Medical Inte~ 2180-07-23 14:00:00
2   10000690 25860671 37081114 Medical Intens~ Medical Inte~ 2150-11-02 19:37:00
3   10000980 26913865 39765666 Medical Intens~ Medical Inte~ 2189-06-27 08:42:00
4   10001217 24597018 37067082 Surgical Inten~ Surgical Int~ 2157-11-20 19:18:02
5   10001217 27703517 34592300 Surgical Inten~ Surgical Int~ 2157-12-19 15:42:24
6   10001725 25563031 31205490 Medical/Surgic~ Medical/Surg~ 2110-04-11 15:52:22
# i 34 more variables: outtime <dttm>, los <dbl>, gender <chr>,
#   anchor_age <dbl>, anchor_year <dbl>, anchor_year_group <chr>, dod <date>,
#   admittime <dttm>, dischtime <dttm>, deathtime <dttm>, admission_type <chr>,
#   admit_provider_id <chr>, admission_location <chr>,
#   discharge_location <chr>, insurance <chr>, language <chr>,
#   marital_status <chr>, race <chr>, edregtime <dttm>, edouttime <dttm>,
#   hospital_expire_flag <dbl>, ...
```

### Q8. Exploratory data analysis (EDA)

Summarize the following information about the ICU stay cohort `mimic_icu_cohort` using appropriate numerics or graphs:

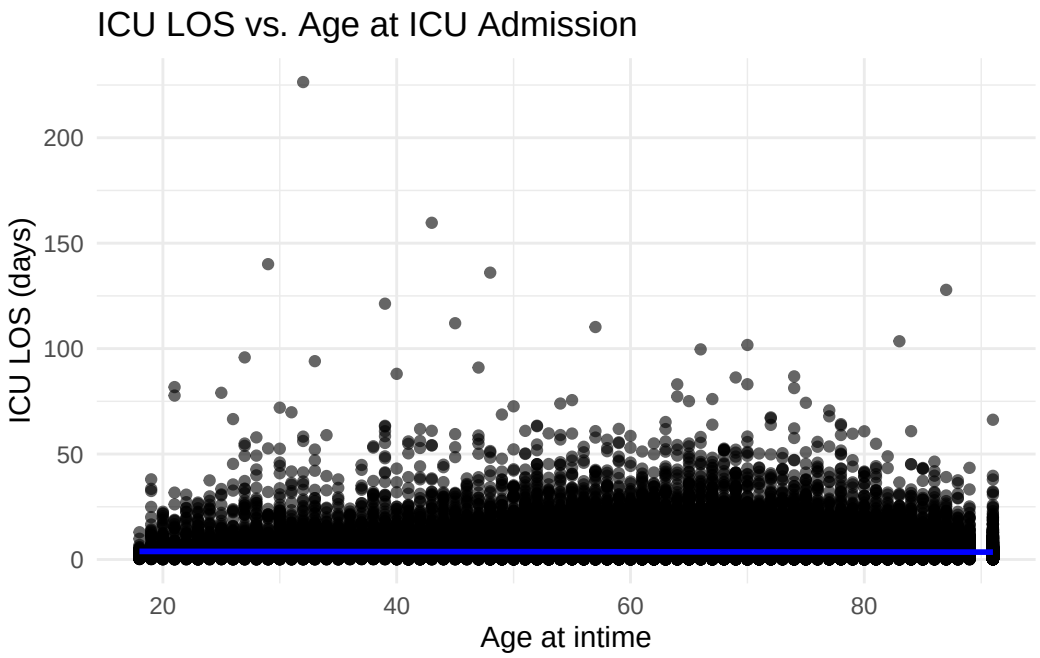
- Length of ICU stay `los` vs demographic variables (race, insurance, marital_status, gender, age at intime)

```
# Scatter plot of Age at intime vs LOS
ggplot(mimic_icu_cohort, aes(x = anchor_age, y = los)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "ICU LOS vs. Age at ICU Admission", x = "Age at intime", y = "ICU LOS (days)")
  theme_minimal()
```

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

```
Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_point()`).
```



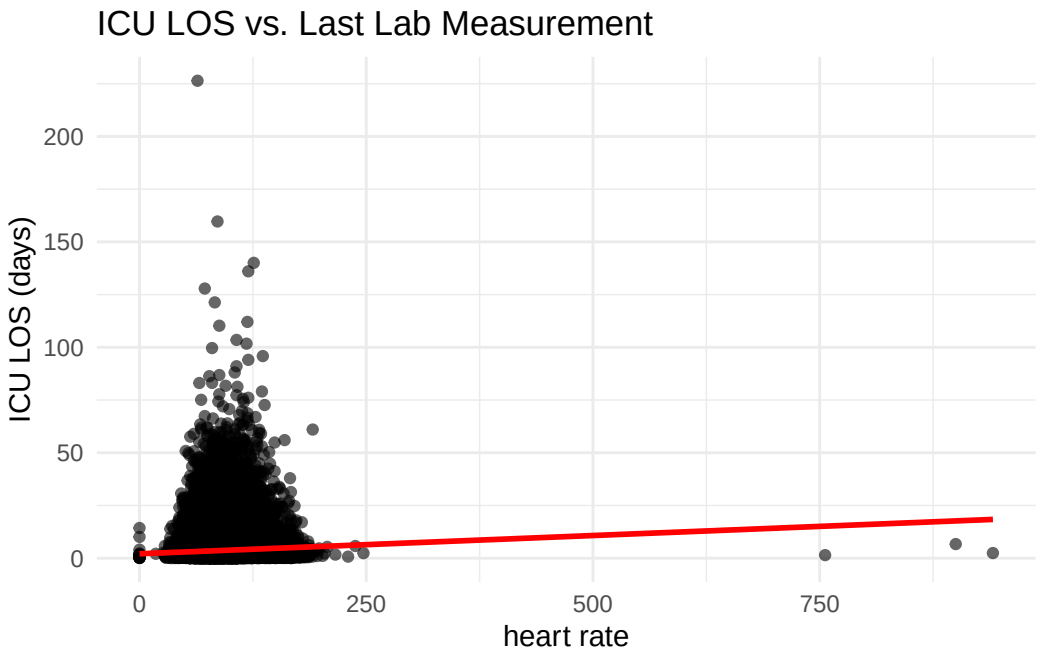

- Length of ICU stay `los` vs the last available lab measurements before ICU stay

```
# Suppose 'last_lab_value' is the last lab measurement before ICU admission
ggplot(mimic_icu_cohort, aes(x = `heart rate`, y = los)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "ICU LOS vs. Last Lab Measurement", y = "ICU LOS (days)") +
  theme_minimal()
```

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

```
Warning: Removed 99 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

```
Warning: Removed 99 rows containing missing values or values outside the scale range
(`geom_point()`).
```



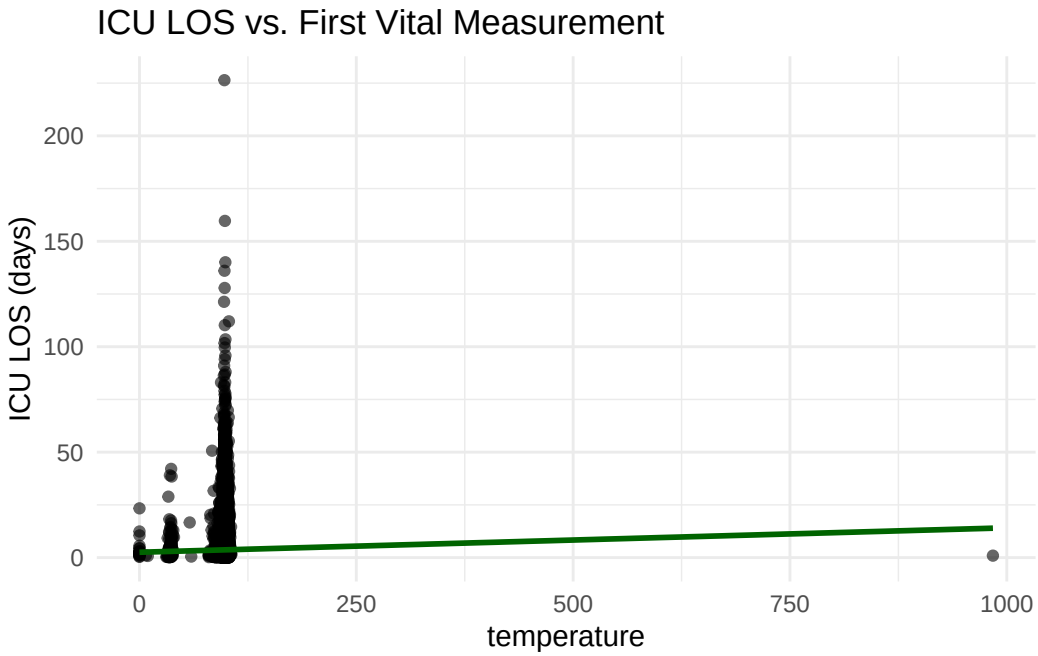

- Length of ICU stay `los` vs the first vital measurements within the ICU stay

```
ggplot(mimic_icu_cohort, aes(x = `temperature fahrenheit`, y = los)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE, color = "darkgreen") +
  labs(title = "ICU LOS vs. First Vital Measurement", x = "temperature", y = "ICU LOS (days)
  theme_minimal()
```

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

```
Warning: Removed 1631 rows containing non-finite outside the scale range
(`stat_smooth()`).
```

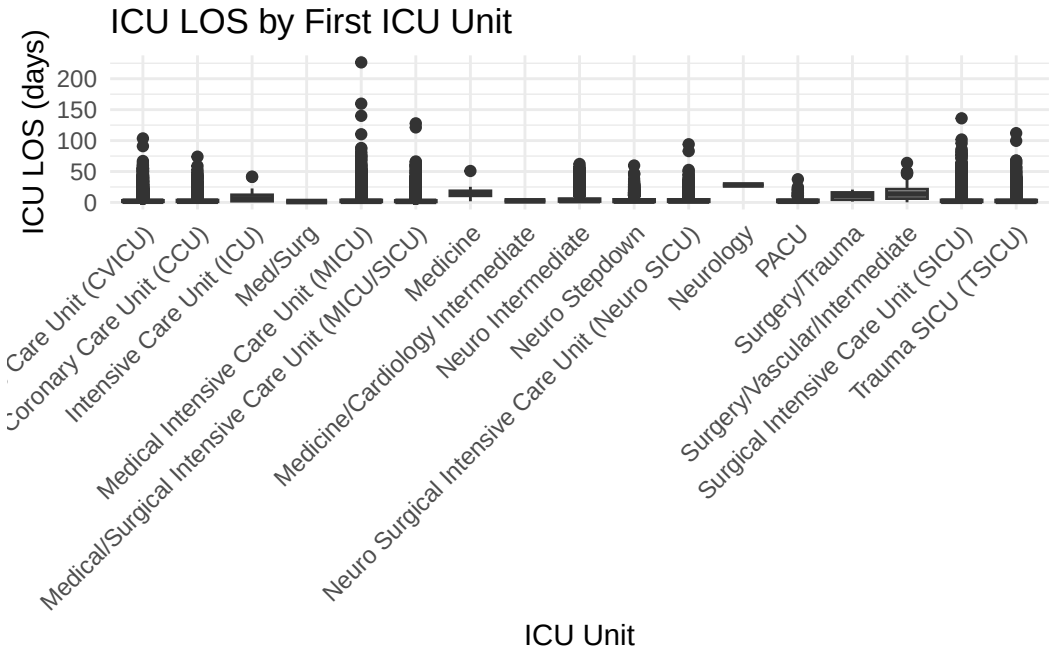
```
Warning: Removed 1631 rows containing missing values or values outside the scale range
(`geom_point()`).
```




- Length of ICU stay `los` vs first ICU unit

```
ggplot(mimic_icu_cohort, aes(x = first_careunit, y = los)) +
  geom_boxplot() +
  labs(title = "ICU LOS by First ICU Unit", x = "ICU Unit", y = "ICU LOS (days)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
Warning: Removed 14 rows containing non-finite outside the scale range
(`stat_boxplot()`).
```


ICU LOS by First ICU Unit
ICU LOS (days)
200
150
100
50
0
Care Unit (CVICU)
Coronary Care Unit (CCU)
Intensive Care Unit (ICU)
Med/Surg
Medical Intensive Care Unit (MICU)
Medical/Surgical Intensive Care Unit (MICU/SICU)
Medicine
Medicine/Cardiology Intermediate
Neuro Intermediate
Neuro Stepdown
Neuro Surgical Intensive Care Unit (Neuro SICU)
Neurology
PACU
Surgery/Trauma
Surgery/Vascular/Intermediate
Surgical Intensive Care Unit (SICU)
Trauma SICU (TSICU)
ICU Unit