Biostat 203B Homework 3

Due Feb 21 @ 11:59PM

Wenqiang Ge UID:106371961

Table of contents

Display machine information for reproducibility:

```
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: x86_64-pc-linux-gnu
Running under: Ubuntu 24.04.1 LTS

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.12.0
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0

locale:
 [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C           LC_TIME=C.UTF-8
 [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8    LC_MESSAGES=C.UTF-8
 [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
[10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C

time zone: America/Los_Angeles
tzcode source: system (glibc)

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

```
loaded via a namespace (and not attached):
 [1] compiler_4.4.2    fastmap_1.2.0     cli_3.6.3         tools_4.4.2
 [5] htmltools_0.5.8.1 rstudioapi_0.17.1 yaml_2.3.10       rmarkdown_2.29
 [9] knitr_1.49        jsonlite_1.8.9    xfun_0.50         digest_0.6.37
[13] rlang_1.1.5       evaluate_1.0.3
```

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

```
# Load necessary libraries
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 1.9.4     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 beco
```

```
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
```

```
library(duckdb)
```

```
Loading required package: DBI
```

```
library(ggplot2)
library(dplyr)
library(lubridate)
library(readr)
library(duckdb)
library(DBI)
library(scales)
```

```

Attaching package: 'scales'

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

    discard

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

    col_factor
```

Display your machine memory.

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

```
Totalram:  7.686 GiB
Freeram:   6.747 GiB
```

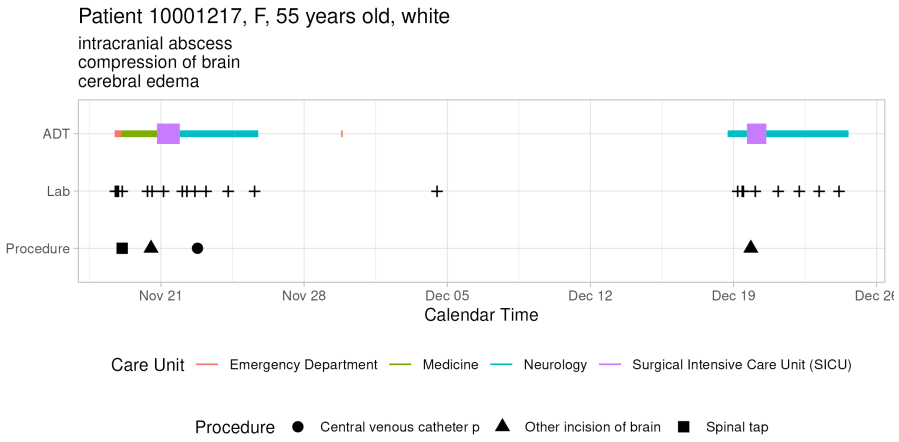
In this exercise, we use tidyverse (ggplot2, dplyr, etc) to explore the MIMIC-IV data introduced in homework 1 and to build a cohort of ICU stays.

0.1 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.

0.1.1 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:

```
# Load datasets as tbles
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.
```

```
admissions <- 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.
```

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

```
Rows: 2413581 Columns: 7
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (2): eventtype, careunit
dbl  (3): subject_id, hadm_id, transfer_id
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.
```

```
procedures <- 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.
```

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

```
Rows: 6364488 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): icd_code
dbl (4): subject_id, hadm_id, seq_num, 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.
```

```
# Filter for patient 10063848
patient_id <- 10063848
patient_data <- patients %>% filter(subject_id == patient_id)
admissions_data <- admissions %>% filter(subject_id == patient_id)
transfers_data <- transfers %>% filter(subject_id == patient_id)
procedures_data <- procedures %>% filter(subject_id == patient_id)
diagnoses_data <- diagnoses %>% filter(subject_id == patient_id)
d_icd_procedures <- read_csv("~/mimic/hosp/d_icd_procedures.csv.gz")
```

```
Rows: 86423 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.
```

```
d_icd_diagnoses <- 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.
```

```
rm(patients,admissions,
   transfers,procedures,diagnoses,
   patient_id)
```

```
#Connect to DuckDB
con <- dbConnect(duckdb::duckdb(), dbdir = ":memory:")

#  Load `data` into DuckDB
dbWriteTable(con, "patients", patient_data, overwrite = TRUE)
dbWriteTable(con, "admissions", admissions_data, overwrite = TRUE)
dbWriteTable(con, "transfers", transfers_data, overwrite = TRUE)
dbWriteTable(con, "procedures", procedures_data, overwrite = TRUE)
dbWriteTable(con, "diagnoses", diagnoses_data, overwrite = TRUE)
dbWriteTable(con, "d_icd_procedures", d_icd_procedures, overwrite = TRUE)
dbWriteTable(con, "d_icd_diagnoses", d_icd_diagnoses, overwrite = TRUE)
```

```
#  labevents
dbExecute(con, "CREATE TABLE Lab AS
                SELECT subject_id,hadm_id,labevent_id,
                charttime,
                FROM read_parquet(
    '~/biostat-203b-2025-winter/hw3/labevents_parquet/' ||
    'part-0.parquet'
  )
                WHERE subject_id = 10063848")
```

```
[1] 1114
```

```
# ADT
dbExecute(con, "
  CREATE TABLE ADT AS
  SELECT
    subject_id,hadm_id,
    careunit,eventtype,
    intime,outtime,
  FROM transfers
")
```

```
[1] 15
```

```
# Procedures
dbExecute(con, "
  CREATE TABLE Procedure AS
  SELECT
    p.subject_id,p.hadm_id,chartdate,
    dp.icd_code,dp.icd_version,dp.long_title,
  FROM procedures p
  LEFT JOIN d_icd_procedures dp
  ON p.icd_code = dp.icd_code and p.icd_version = dp.icd_version
")
```

```
[1] 6
```

```
#ADT history
dbExecute(con, "
  CREATE TABLE ADT_history AS
  SELECT
    a.subject_id,a.hadm_id,a.careunit,
    a.eventtype,a.intime,a.outtime,
    p.chartdate,p.long_title,
    p.chartdate,
  FROM ADT a
  LEFT JOIN Procedure p
  ON a.subject_id = p.subject_id and a.hadm_id = p.hadm_id
```

```
")
```

```
[1] 42
```

```
#patients info & subtitle shows top 3 diagnoses.
dbExecute(con, "
  CREATE TABLE title AS
  SELECT
    p.subject_id,p.gender,p.anchor_age,
    d.icd_code,d.icd_version,
    dd.long_title,
  FROM patients p
  LEFT JOIN diagnoses d
  on p.subject_id = d.subject_id
  LEFT JOIN d_icd_diagnoses dd
  ON d.icd_code = dd.icd_code and d.icd_version = dd.icd_version
")
```

```
[1] 34
```

```
info <- dbGetQuery(con, "
  SELECT t.subject_id,t.gender,t.anchor_age,a.race,
  FROM title t
  left join admissions a
  on t.subject_id = a.subject_id
  limit 1
")
```

```
top_diagnoses <- dbGetQuery(con, "
  SELECT dd.long_title
  FROM (
    SELECT icd_code
    FROM diagnoses
    LIMIT 3
  ) AS t
  LEFT JOIN d_icd_diagnoses dd
    ON t.icd_code = dd.icd_code
")

top_diagnoses_vector <- top_diagnoses$long_title

rm(top_diagnoses)
```

```
# Collect ADT Data
adt_data <- tbl(con, "ADT_history") %>%
  collect() %>%
  mutate(intime = as.POSIXct(intime, format="%Y-%m-%d %H:%M:%S"),
```

```
        outtime = as.POSIXct(outtime, format="%Y-%m-%d %H:%M:%S")) %>%
  filter(careunit != "UNKNOWN", outtime > intime)
# Remove unknown care units and ensure valid time intervals

# Collect Procedure Data
procedure_data <- tbl(con, "Procedure") %>%
  collect() %>%
  mutate(chartdate = as.POSIXct(chartdate, format="%Y-%m-%d"))
# Convert chart date to POSIXct format

# Collect Lab Data (Ensure One Cross Per Day)
lab_data <- tbl(con, "Lab") %>%
  collect() %>%
  mutate(charttime = as.POSIXct(charttime, format="%Y-%m-%d %H:%M:%S"))
# Convert lab event timestamps to POSIXct format

# Extract unique procedure names (long_title)
unique_shapes <- unique(procedure_data$long_title)
num_shapes <- length(unique_shapes)
# Count the number of unique procedure types

# Preset shapes (circle, triangle, square)
base_shapes <- c(15,16,17)
# Default shapes for the first three procedures

# Additional backup shapes (ggplot2 shape codes)
extra_shapes <- c(9,10,11,12,13,14)
# Extra shapes in case there are more procedure types

# Determine the final set of shapes to use
if (num_shapes > length(base_shapes)) {
  final_shapes <- c(base_shapes,
                    extra_shapes[1:(num_shapes - length(base_shapes))])
  # Use backup shapes if needed
} else {
  final_shapes <- base_shapes[1:num_shapes]
  # Use only the default shapes if they are enough
}

# Map procedure names to their corresponding shapes
shape_mapping <- setNames(final_shapes, unique_shapes)
```

```
# Create Final Plot
final_plot <- ggplot() +
```

```
  # Procedure Events
  geom_point(data = procedure_data, aes(x = chartdate,
                                        y = "Procedure",
                                        shape = long_title), size = 5) +

  # Lab Events (Cross Shape)
  geom_point(data = lab_data, aes(x = charttime, y = "Lab"),
             shape = 3, size = 4, stroke = 1.5) +

  # ADT timeline
  geom_segment(data = adt_data, aes(x = intime, xend = outtime,
                                    y = "ADT", yend = "ADT",
                                    color = careunit,
                                    size = ifelse(careunit %in% c("
                                    Surgical Intensive Care Unit (SICU)",
                                    "Medical Intensive Care Unit (MICU)"),
                                    10, 5))) +

  # Dynamic Shape Assignment
  scale_shape_manual(values = shape_mapping) +
  scale_size_identity() +
  scale_x_datetime(date_labels = "%b %d", date_breaks = "1 week") +

  # Axis Labels and Themes
  labs(title = paste("Patient", info[1], ",",
                     info[2], ",", info[3], "years old,",
                     tolower(info[4])),
       subtitle = paste(top_diagnoses_vector, collapse = "\n"),
       x = "Calendar Time",
       y = "",
       color = "Care Unit",
       shape = "Procedure") +

  theme_minimal() +
  theme(legend.position = "bottom",  # Move legend to the bottom
        axis.text.y = element_text(size = 10),
        # Ensure Y-axis labels are readable
        axis.ticks.y = element_blank()) +
  guides(color = guide_legend(nrow = 6), shape = guide_legend(nrow = 5))
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
# Split legend into multiple rows

# Save and display the plot
```

```
ggsave("ADT_plot.png", final_plot, width = 15, height = 6, dpi = 300)
print(final_plot)
```


```
rm(top_diagnoses_vector)
```

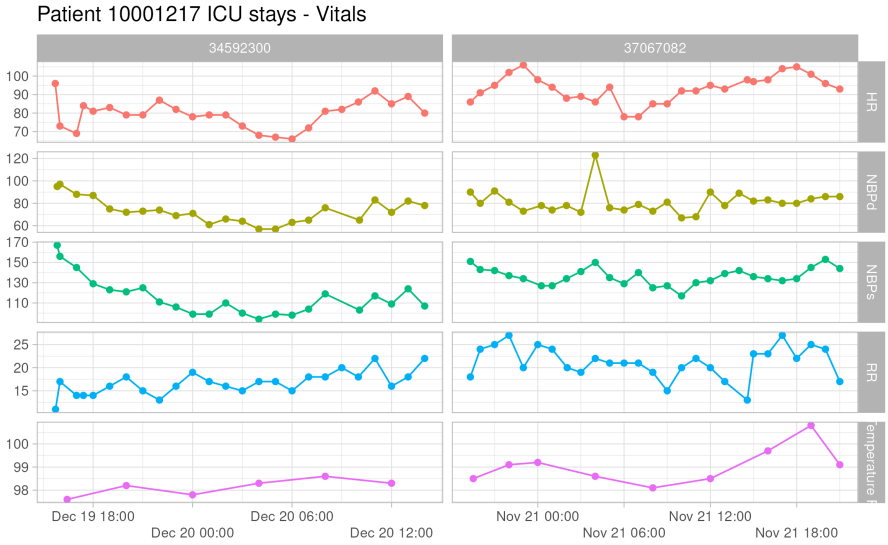


```
rm(admissions_data,lab_data,transfers_data,
   procedure_data,procedures_data,d_icd_diagnoses,
   d_icd_procedures,patient_data,adt_data,diagnoses_data)

rm(final_plot,shape_mapping,unique_shapes)

rm(con,info)
```

0.1.2 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:

```
patient_id <- 10063848
icustays <- read_csv("~/mimic/icu/icustays.csv.gz")
```

```
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.
```

```
patient_data <- icustays %>% filter(subject_id == patient_id)
```

```
#Connect to DuckDB
con <- dbConnect(duckdb::duckdb(), dbdir = ":memory:")

#  Load `data` into DuckDB
dbWriteTable(con, "patients", patient_data, overwrite = TRUE)
dbWriteTable(con, "icustays", icustays, overwrite = TRUE)
```

```
# chartevents
dbExecute(con, "CREATE TABLE chartevents AS
                SELECT subject_id, itemid, charttime, valuenum
                FROM read_parquet(
    '~/biostat-203b-2025-winter/hw3/chartevents_parquet/' ||
    'part-0.parquet'
  )
                WHERE itemid IN (220045, 220179, 220180, 223761, 220210) and
                subject_id = 10063848")
```

```
[1] 298
```

```
# stay
dbExecute(con, "
  CREATE TABLE stay AS
  SELECT subject_id,stay_id,intime,outtime
  FROM icustays
")
```

```
[1] 94458
```

```
# vital
dbExecute(con, "
  CREATE TABLE vital AS
  SELECT c.subject_id,c.itemid,
  c.charttime,c.valuenum,
  s.stay_id,s.intime,s.outtime
  FROM chartevents c
  Left join stay s
  on c.subject_id = s.subject_id
")
```

```
[1] 596
```

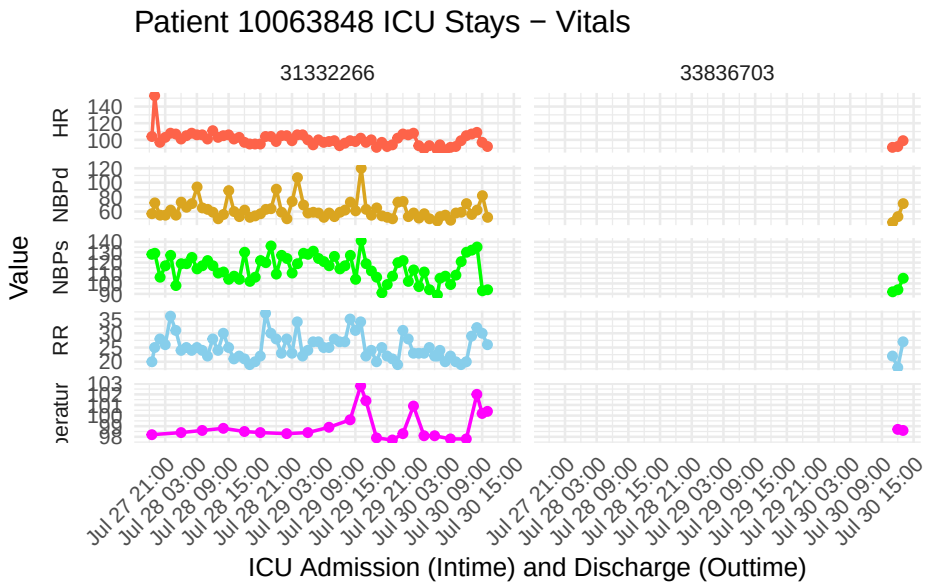
```
# Load data from DuckDB
vital_data <- dbGetQuery(con, "
  SELECT subject_id, stay_id, itemid, charttime,
  valuenum, intime, outtime
  FROM vital
  WHERE charttime BETWEEN intime AND outtime
")

rm(icustays,patient_data)
```

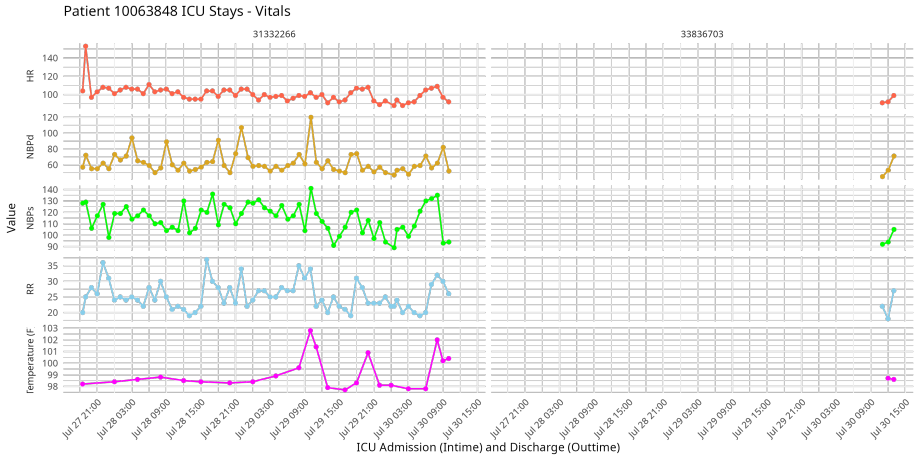
```
# Convert time columns to POSIXct
vital_data <- vital_data %>%
  mutate(charttime = as.POSIXct(charttime, format="%Y-%m-%d %H:%M:%S"),
         intime = as.POSIXct(intime, format="%Y-%m-%d %H:%M:%S"),
```

```
         outtime = as.POSIXct(outtime, format="%Y-%m-%d %H:%M:%S"))


# Define mapping for itemid to readable labels
itemid_labels <- c("220045" = "HR",
                   "220180" = "NBPd",
                   "220179" = "NBPs",
                   "220210" = "RR",
                   "223761" = "Temperature (F)")

# Assign readable labels for itemid
vital_data <- vital_data %>%
  mutate(vital_type = factor(itemid_labels[as.character(itemid)],
                             levels = itemid_labels))

vital_colors <- c("HR" = "#FD6349",
                  "NBPd" = "goldenrod",
                  "NBPs" = "green",
                  "Temperature (F)" = "#FF01FD",
                  "RR" = "skyblue")
```

```
# Adjust each stay_id's x-axis to fit within its intime and outtime range
vital_data_filtered <- vital_data %>%
  group_by(stay_id) %>%
  ungroup()

vital_plot <- ggplot(vital_data_filtered, aes(x = charttime,
                                              y = valuenum,
                                              color = vital_type)) +
  geom_point(size = 1.5) +
  geom_line(aes(group = vital_type), size = 0.7) +
  scale_color_manual(values = vital_colors) +
  scale_x_datetime(labels = date_format("%b %d %H:%M"), breaks = "6 hours") +
  facet_grid(vital_type ~ stay_id, scales = "free_y", switch = "y") +
  labs(title = paste("Patient",
                     unique(vital_data$subject_id), "ICU Stays - Vitals"),
       x = "ICU Admission (Intime) and Discharge (Outtime)",
       y = "Value") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        strip.placement = "outside",
        strip.text.y.right = element_text(size = 12, face = "bold"),
        legend.position = "none")

ggsave("icu_vitals.png", vital_plot, width = 12, height = 6, dpi = 300)
```

```
print(vital_plot)
```






```
rm(vital_data,vital_data_filtered,vital_plot)
rm(vital_colors)
```

0.2 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
```

0.2.1 Q2.1 Ingestion

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

Solution:

```
icustays_tble <- read_csv("~/mimic/icu/icustays.csv.gz")
```

```
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.
```

0.2.2 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:

```
# Count unique subject_id
num_unique_subjects <- icustays_tble%>%
  distinct(subject_id) %>%
  nrow()

print(paste("Number of unique subjects:", num_unique_subjects))
```

```
[1] "Number of unique subjects: 65366"
```

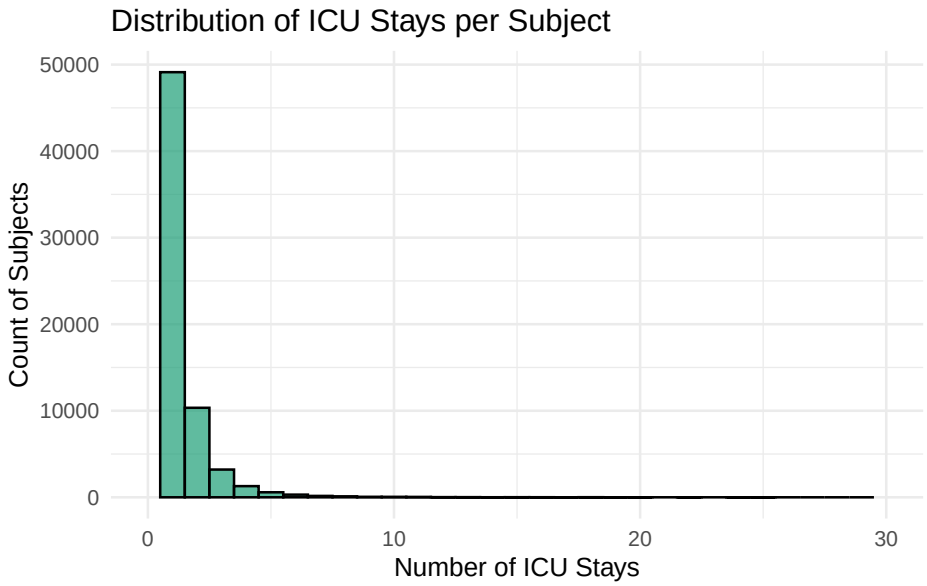
```
library(plotly)
```

```

Attaching package: 'plotly'

The following object is masked from 'package:ggplot2':

    last_plot

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

```
    schema

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

    filter

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

    layout
```

```r
# Count ICU stays per subject
icu_stays_per_subject <- icustays_tble %>%
  group_by(subject_id) %>%
  summarise(num_stays = n(), .groups = "drop")

# Create a histogram to visualize the distribution
p <- ggplot(icu_stays_per_subject, aes(x = num_stays)) +
  geom_histogram(binwidth = 1, fill = "#1B9E76",
                 color = "black", alpha = 0.7) +
                  # Use histogram with bin width 1
  scale_x_continuous(limits = c(0, 30)) +
  # Limit X-axis to focus on the majority of data
  labs(title = "Distribution of ICU Stays per Subject",
       x = "Number of ICU Stays",
       y = "Count of Subjects") +
  theme_minimal()  # Use a clean theme for better readability

print(p)
```

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_bin()`).

Warning: Removed 2 rows containing missing values or values outside the scale range
(`geom_bar()`).
```




```
# Convert to an interactive plot
#ggplotly(p)

rm(icu_stays_per_subject)
```

The histogram shows the distribution of ICU stays per subject (patient). Most subjects have only one ICU stay, with a sharp drop-off as the number of stays increases. The majority of patients (49,124 subjects) had a single ICU stay, while a much smaller proportion experienced multiple ICU stays. This suggests that repeated ICU admissions are uncommon, but some patients do return multiple times, possibly due to chronic conditions, post-surgical complications, or severe illnesses requiring multiple admissions. The long tail of the distribution indicates that a few patients had 10 or more ICU stays, though these cases are rare.

0.3 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
```

0.3.1 Q3.1 Ingestion

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

Solution:

```
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.
```

0.3.2 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:

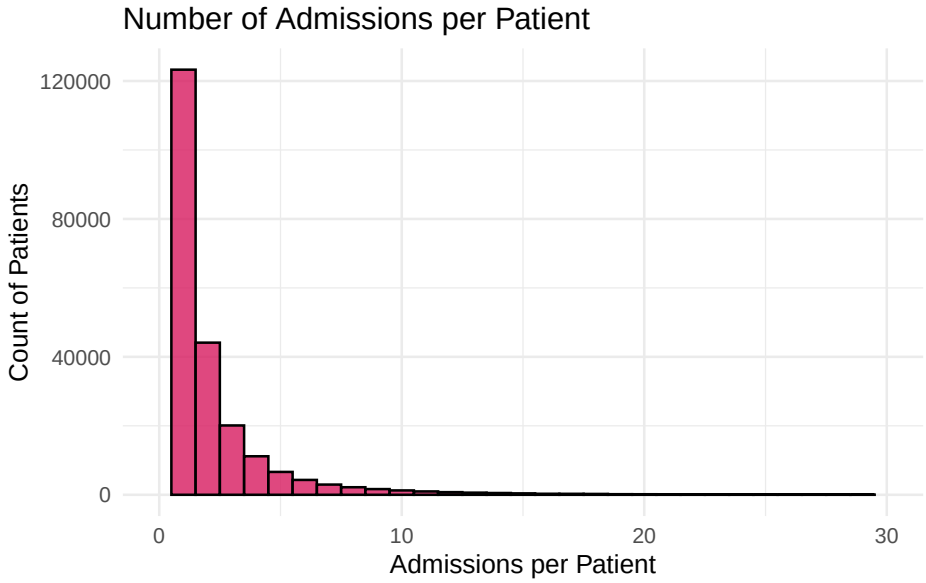
- **number of admissions per patient**

```
# Compute the number of admissions per patient
admissions_per_patient <- admissions_tble %>%
  group_by(subject_id) %>%
  summarise(num_admissions = n(), .groups = "drop")

# Create a histogram to visualize the distribution
p <- ggplot(admissions_per_patient, aes(x = num_admissions)) +
```

```
  geom_histogram(binwidth = 1, fill = "#D81B60",
                 color = "black", alpha = 0.8) +
                  # Use histogram instead of bar chart
  scale_x_continuous(limits = c(0, 30)) +
  # Limit the X-axis to focus on the most relevant range
  labs(title = "Number of Admissions per Patient",
       x = "Admissions per Patient",
       y = "Count of Patients") +
  theme_minimal()  # Use a clean theme for better readability

print(p)
```

```
Warning: Removed 504 rows containing non-finite outside the scale range
(`stat_bin()`).
```

```
Warning: Removed 2 rows containing missing values or values outside the scale range
(`geom_bar()`).
```



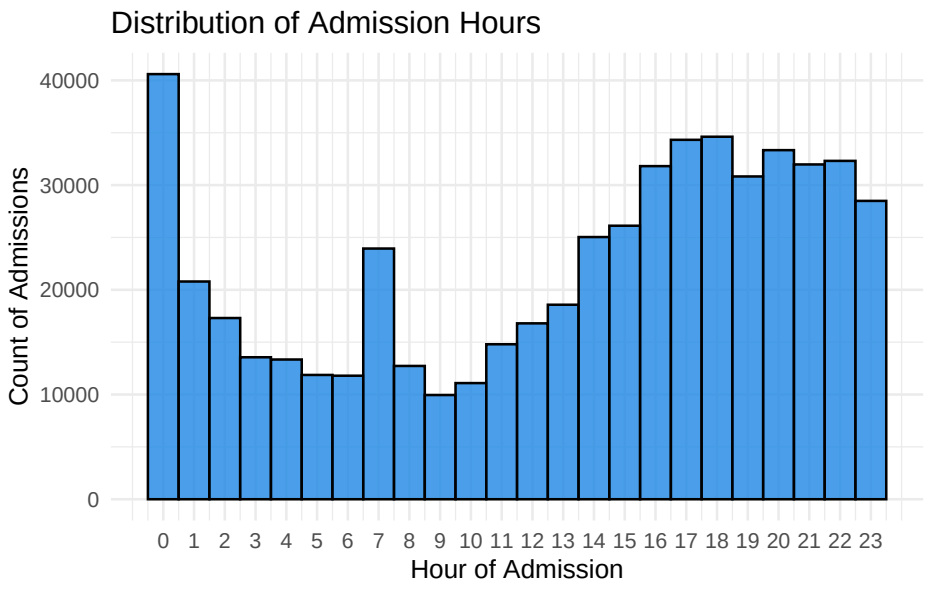

```
# Convert to an interactive plot
#ggplotly(p)

rm(admissions_per_patient)
```

Most patients have only one or very few admissions, while very few patients have a large number of admissions. A small number of patients have multiple hospitalizations, but the frequency drops quickly as the number of admissions increases. Patients with 10+ admissions are very rare. The distribution is

heavily right-skewed, with very few patients experiencing more than 5 hospital stays. This means that most patients have a low number of admissions, and a small subset of patients has a significantly higher number. Also, Most hospital patients are one-time visitors, while a minority (likely with chronic illnesses or complications) are readmitted multiple times.

- **admission hour (anything unusual?)**

```
# Extract hour from admission time
admissions_tble <- admissions_tble %>%
  mutate(admission_hour = hour(admittime))

# Create a histogram of admission hours
p_hour <- ggplot(admissions_tble, aes(x = admission_hour)) +
  geom_histogram(binwidth = 1, fill = "#1E88E5",
                 color = "black", alpha = 0.8) +
  scale_x_continuous(breaks = seq(0, 23, by = 1)) +
  labs(title = "Distribution of Admission Hours",
       x = "Hour of Admission",
       y = "Count of Admissions") +
  theme_minimal()

print(p_hour)
```

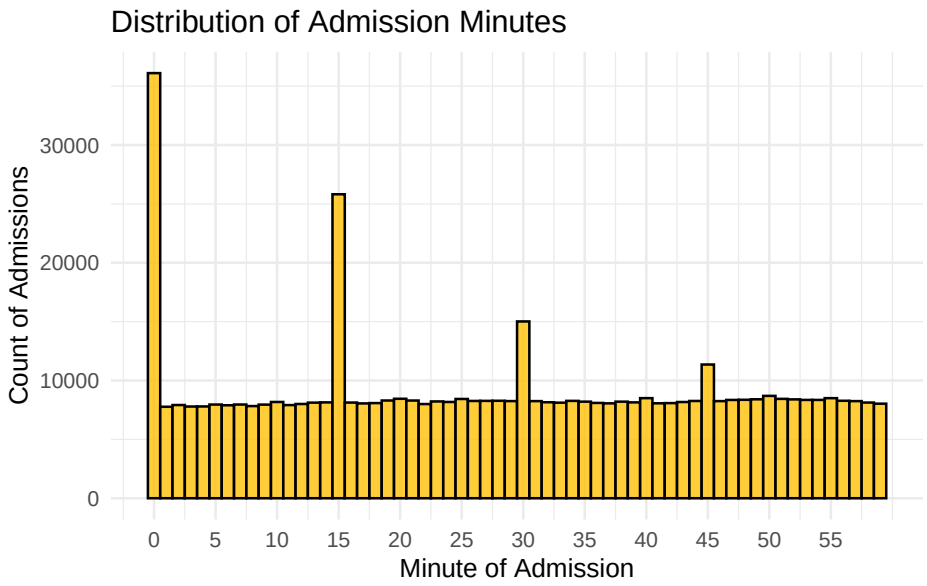



```
# Convert to interactive plot
#ggplotly(p_hour)
```

The distribution of admission hours shows notable patterns. There is a sharp peak at midnight (00:00), suggesting that many admissions are recorded at the start of the day, possibly due to administrative rounding or batch processing of records. Another smaller peak is observed around 07:00–08:00, which could be associated with morning shift changes or scheduled hospital admissions. Admissions increase steadily from late morning and peak between 16:00–19:00, likely reflecting patient arrivals after daytime medical consultations and emergency cases during evening hours. The pattern indicates a combination of systemic recording practices, scheduled admissions, and emergency patient arrivals.

- **admission minute (anything unusual?)**

```
# Extract minute from admission time
admissions_tble <- admissions_tble %>%
  mutate(admission_minute = minute(admittime))

# Create a histogram of admission minutes
p_minute <- ggplot(admissions_tble, aes(x = admission_minute)) +
  geom_histogram(binwidth = 1, fill = "#FFC107",
                 color = "black", alpha = 0.8) +
  scale_x_continuous(breaks = seq(0, 59, by = 5)) +
  labs(title = "Distribution of Admission Minutes",
       x = "Minute of Admission",
       y = "Count of Admissions") +
  theme_minimal()

print(p_minute)
```



```r
# Convert to interactive plot
#ggplotly(p_minute)
```

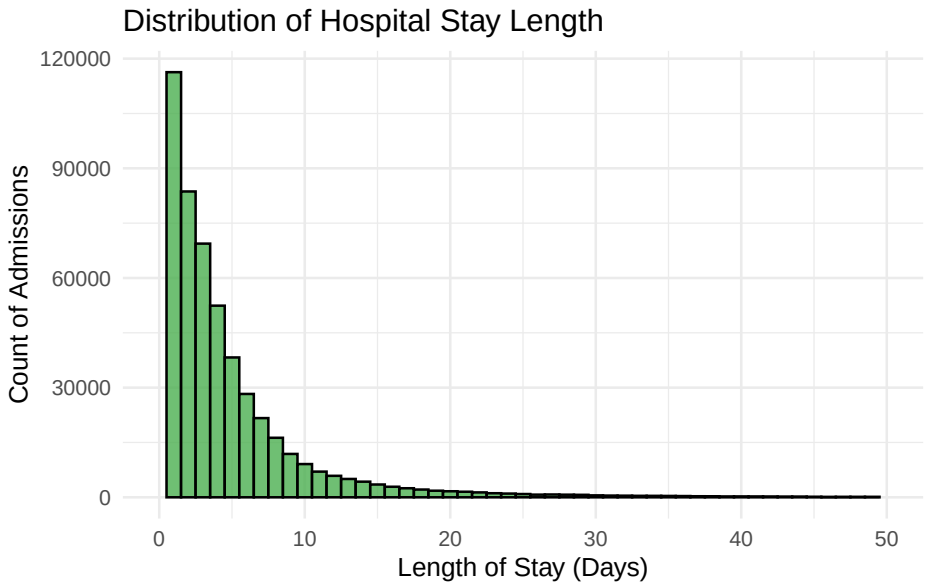
The distribution of admission minutes shows distinct spikes at 0, 15, 30, and 45 minutes, suggesting that admission times are often rounded to quarter-hour intervals. This pattern likely arises due to manual data entry practices, hospital system constraints, or administrative processes that batch-record admissions. The relatively uniform distribution in other minutes indicates that some admissions occur naturally without rounding, but the spikes highlight a systemic bias in how times are logged. This could impact time-based analyses and should be accounted for in further data processing.

- **length of hospital stay (from admission to discharge) (anything unusual?)**

```r
# Calculate length of stay in days
admissions_tble <- admissions_tble %>%
  mutate(length_of_stay = as.numeric(difftime(dischtime,
                                              admittime, units = "days")))

# Create a histogram of hospital stay length
p_los <- ggplot(admissions_tble, aes(x = length_of_stay)) +
  geom_histogram(binwidth = 1, fill = "#4CAF50",
                 color = "black", alpha = 0.8) +
  scale_x_continuous(limits = c(0, 50)) +
  # Adjusted limit for visualization
  labs(title = "Distribution of Hospital Stay Length",
       x = "Length of Stay (Days)",
       y = "Count of Admissions") +
  theme_minimal()

print(p_los)
```

```
Warning: Removed 2105 rows containing non-finite outside the scale range
(`stat_bin()`).
```

```
Warning: Removed 2 rows containing missing values or values outside the scale range
(`geom_bar()`).
```




```
# Convert to interactive plot
#ggplotly(p_los)
```

The distribution of hospital stay length is right-skewed, with most admissions having a short length of stay. The majority of patients are discharged within a few days, with the highest count around 1–3 days, suggesting that many hospital visits are for short-term treatments or observations. As the length of stay increases, the number of admissions decreases, indicating that extended hospitalizations are less common. The presence of a long tail suggests that a small number of patients require prolonged hospital care, possibly due to severe or chronic conditions. This pattern is typical in hospital data, where most cases are routine and resolved quickly, while complex cases require extended stays.

0.4 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
```

0.4.1 Q4.1 Ingestion

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

Solution:

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

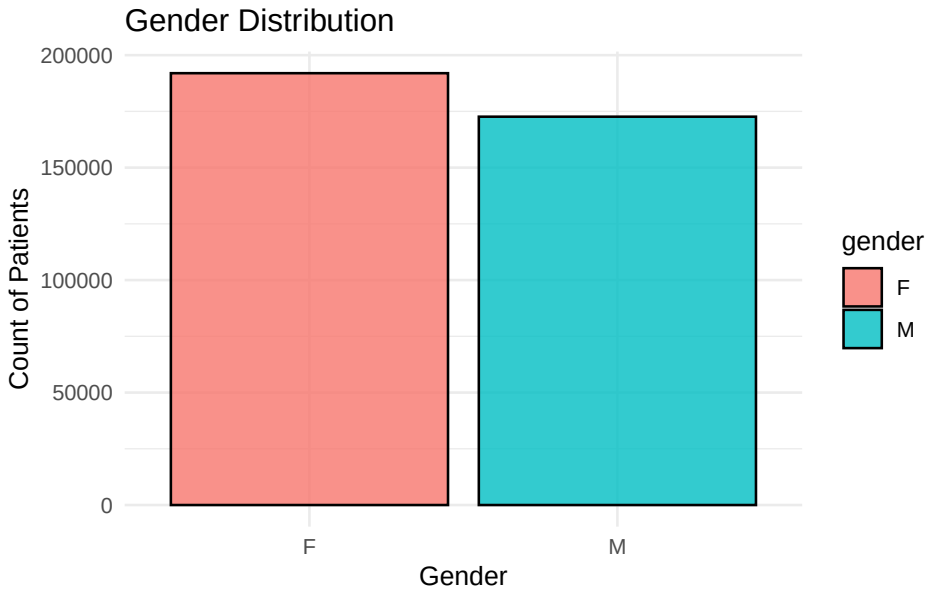
```
Rows: 364627 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (2): gender, anchor_year_group
dbl  (3): subject_id, anchor_age, anchor_year
date (1): dod

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.
```

0.4.2 Q4.2 Summary and visualization

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

Solution

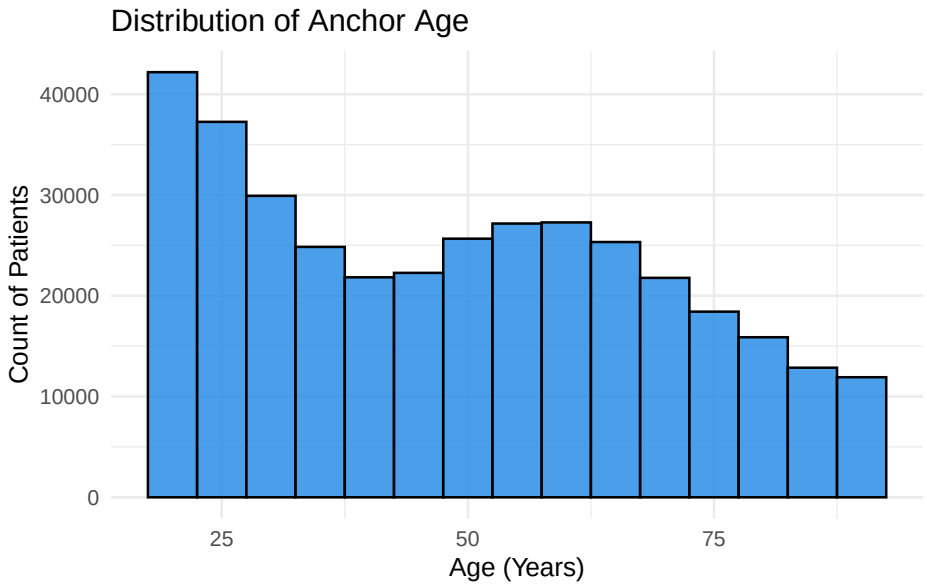
```
# Count the number of patients by gender
gender_dist <- patients_tble %>%
  count(gender)

# Bar plot for gender distribution
p_gender <- ggplot(gender_dist, aes(x = gender, y = n, fill = gender)) +
  geom_bar(stat = "identity", color = "black", alpha = 0.8) +
  labs(title = "Gender Distribution",
       x = "Gender",
       y = "Count of Patients") +
  theme_minimal()


print(p_gender)
```




```
# Convert to interactive plot
#ggplotly(p_gender)
rm(gender_dist)
```

The bar chart indicates that there are more female (F) patients than male (M) patients in the dataset. The difference is noticeable but not extreme, suggesting a slight gender imbalance in hospital admissions. This could be influenced by factors such as longer life expectancy in females, higher healthcare utilization rates among women, or specific hospital demographics.

```
# Histogram of anchor_age
p_age <- ggplot(patients_tble, aes(x = anchor_age)) +
  geom_histogram(binwidth = 5, fill = "#1E88E5",
                 color = "black", alpha = 0.8) +
  labs(title = "Distribution of Anchor Age",
       x = "Age (Years)",
       y = "Count of Patients") +
  theme_minimal()

print(p_age)
```




```
# Convert to interactive plot
#ggplotly(p_age)
```

The histogram of anchor_age shows a bimodal distribution, with a large concentration of patients in the 20–30 age range and another broader peak between 50–70 years old. The younger peak could represent a large number of maternity-related or younger adult admissions, while the older peak likely reflects age-related chronic conditions and elderly care. The drop-off after 80 years suggests a lower number of very elderly patients, possibly due to lower life expectancy or selection bias in hospital records.

```
rm(p,p_hour,p_los,
   p_minute,p_gender,p_age)
```

0.5 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
```

`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
```

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:

```
# Connect to DuckDB
con <- dbConnect(duckdb::duckdb(), dbdir = ":memory:")

# Load and filter lab events directly in DuckDB
dbExecute(con, "CREATE TABLE filtered_labevents AS
                SELECT subject_id, itemid, storetime, valuenum
                FROM read_parquet(
    '~/biostat-203b-2025-winter/hw3/labevents_parquet/' ||
    'part-0.parquet'
  )
                WHERE itemid IN (50912, 50971, 50983, 50902,
          50882, 51221, 51301, 50931)")
```

```
[1] 32679896
```

```
#  Load ICU stays dataset from CSV and store it in DuckDB
dbExecute(con, "CREATE TABLE icustays AS
                SELECT subject_id, stay_id, intime, outtime
                FROM read_csv_auto('~/mimic/icu/icustays.csv.gz')")
```

```
[1] 94458
```

```
# Join lab events with ICU stays and filter by time condition
dbExecute(con, "CREATE TABLE merged_labevents AS
                SELECT l.subject_id, i.stay_id,
                l.itemid, l.storetime, l.valuenum
                FROM filtered_labevents l
                INNER JOIN icustays i ON l.subject_id = i.subject_id
                WHERE l.storetime < i.intime")
```

```
[1] 20122551
```

```
# Select the most recent lab measurement per `itemid` before ICU admission
dbExecute(con, "CREATE TABLE latest_labs AS
                SELECT subject_id, stay_id, itemid, storetime, valuenum
                FROM (
                    SELECT *, ROW_NUMBER() OVER (PARTITION BY
                    subject_id, stay_id, itemid ORDER BY storetime DESC) AS rn
                    FROM merged_labevents
                ) WHERE rn = 1")
```

```
[1] 677237
```

```
# Pivot `itemid` to become separate columns
dbExecute(con, "CREATE TABLE labevents_pivoted AS
                SELECT subject_id, stay_id,
                       MAX(CASE
                       WHEN itemid = 50882 THEN valuenum END) AS bicarbonate,
                       MAX(CASE
                       WHEN itemid = 50902 THEN valuenum END) AS chloride,
                       MAX(CASE
                       WHEN itemid = 50912 THEN valuenum END) AS creatinine,
                       MAX(CASE
                       WHEN itemid = 50931 THEN valuenum END) AS glucose,
                       MAX(CASE
                       WHEN itemid = 50971 THEN valuenum END) AS potassium,
                       MAX(CASE
                       WHEN itemid = 50983 THEN valuenum END) AS sodium,
                       MAX(CASE
                       WHEN itemid = 51221 THEN valuenum END) AS hematocrit,
                       MAX(CASE
                       WHEN itemid = 51301 THEN valuenum END) AS wbc,
                FROM latest_labs
                GROUP BY subject_id, stay_id")
```

```
[1] 88086
```

```
# Fetch the final processed table into R
labevents_tble <- dbGetQuery(con, "SELECT * FROM labevents_pivoted")
```

```
# Arrange for better readability
labevents_tble <- labevents_tble %>% arrange(subject_id, stay_id)

# View final dataset
options(width = 1000)
print(head(labevents_tble, 10))
```

```
   subject_id  stay_id bicarbonate chloride creatinine glucose potassium sodium hematocrit
1    10000032 39553978          25       95        0.7     102       6.7    126       41.1
2    10000690 37081114          26      100        1.0      85       4.8    137       36.1
3    10000980 39765666          21      109        2.3      89       3.9    144       27.3
4    10001217 34592300          30      104        0.5      87       4.1    142       37.4
5    10001217 37067082          22      108        0.6     112       4.2    142       38.1 1
6    10001725 31205490          NA       98         NA      NA       4.1    139         NA
7    10001843 39698942          28       97        1.3     131       3.9    138       31.4 1
8    10001884 37510196          30       88        1.1     141       4.5    130       39.7 1
9    10002013 39060235          24      102        0.9     288       3.5    137       34.9
10   10002114 34672098          18       NA        3.1      95       6.5    125       34.3 1
```

```
# Disconnect from DuckDB
dbDisconnect(con, shutdown = TRUE)

rm(con)
gc()
```

```
           used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  1897387 101.4    3896407 208.1  3896407 208.1
Vcells 17233095 131.5   53504516 408.3 83585364 637.8
```

0.6 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
```

`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
```

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). Re-

trieve 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: 94,363 × 7
   subject_id  stay_id heart_rate non_invasive_blood_pressure_diastolic non_invasive_blood_pressure_sy…¹ respiratory_rate temperature_fahrenheit
        <dbl>    <dbl>      <dbl>                                 <dbl>                             <dbl>            <dbl>                 <dbl>
 1   10000032 39553978         91                                    48                                84               24                  98.7
 2   10000690 37081114         80                                    63                               107               27                  97.7
 3   10000980 39765666         77                                   127                               158               24                  98
 4   10001217 34592300         96                                    97                               167               17                  97.6
 5   10001217 37067082         86                                    90                               151               18                  98.5
 6   10001725 31205490         86                                    56                                73               19                  97.7
 7   10001843 39698942        131                                    85                               112               17                  97.9
 8   10001884 37510196         60                                    49                               180               16                  98.1
 9   10002013 39060235         80                                    70                               104               14                  97.2
10   10002114 34672098        111                                    80                               112               22                  97.9
# ℹ 94,353 more rows
# ℹ abbreviated name: ¹non_invasive_blood_pressure_systolic
# 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:

```
#Connect to DuckDB
con <- dbConnect(duckdb::duckdb(), dbdir = ":memory:")

# Load ICU stays dataset from CSV into DuckDB
duckdb_table_icustays <- tbl(
  con, sql("SELECT * FROM read_csv_auto('~/mimic/icu/icustays.csv.gz')")
  )

# Query ICU stays dataset within DuckDB
result_icustays <- duckdb_table_icustays %>%
  select(subject_id, stay_id, intime,outtime) %>%  # Select necessary columns
  collect()  # Bring the filtered results into memory as a tibble

dbDisconnect(con)
gc()
```

```
           used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  1922044 102.7    3896407 208.1  3896407 208.1
Vcells 17666050 134.8   53504516 408.3 83585364 637.8
```

```
rm(con,duckdb_table_icustays)
```

```
# Connect to DuckDB
con <- dbConnect(duckdb::duckdb(), dbdir = ":memory:")

#  Load and filter only required `itemid` values (Reduce Memory Usage)
dbExecute(con, "CREATE TABLE chartevents_duckdb AS
```

```
                SELECT subject_id, itemid, storetime, valuenum
                FROM read_parquet(
    '~/biostat-203b-2025-winter/hw3/chartevents_parquet/' ||
    'part-0.parquet'
  )
                WHERE itemid IN (220045, 220179, 220180, 223761, 220210)")
```

```
[1] 30200193
```

```
#  Load `icustays_tble` into DuckDB
dbWriteTable(con, "icustays", result_icustays, overwrite = TRUE)
```

```
# Filter the measurements in between the icu time
dbExecute(con, "CREATE TABLE latest_vitals_raw AS
                SELECT c.subject_id, i.stay_id,
                c.itemid, c.storetime, c.valuenum
                FROM chartevents_duckdb c
                INNER JOIN icustays i
                ON c.subject_id = i.subject_id
                WHERE c.storetime BETWEEN i.intime AND i.outtime")
```

```
[1] 30129155
```

```
rm(result_icustays)
```

```
dbExecute(con, "CREATE TABLE ranked_vitals AS
        SELECT *,
        ROW_NUMBER() OVER (PARTITION BY subject_id, stay_id,
        itemid ORDER BY storetime ASC) AS row_num,
        FIRST_VALUE(storetime) OVER (PARTITION BY subject_id, stay_id,
        itemid ORDER BY storetime ASC) AS first_storetime
        FROM latest_vitals_raw")
```

```
[1] 30129155
```

```
dbExecute(con, "VACUUM")
```

```
[1] 0
```

```
dbExecute(con, "CREATE TABLE first_vitals AS
        SELECT *
        FROM ranked_vitals
        WHERE storetime = first_storetime")
```

```
[1] 643570
```

```
# Select the first measurement (`storetime`) per `itemid` and `stay_id`
dbExecute(con, "CREATE TABLE latest_vitals_filtered AS
    SELECT subject_id, stay_id, itemid, storetime,
    round(AVG(valuenum),2) AS valuenum
```

```
    FROM first_vitals
    GROUP BY subject_id, stay_id, itemid, storetime")
```

```
[1] 467599
```

```
# Pivot `itemid` into separate columns (one row per ICU stay)
latest_vitals <- dbGetQuery(con, "SELECT * FROM latest_vitals_filtered")

rm(con)
chartevents_tble <- latest_vitals %>%
  select(-storetime) %>%
  pivot_wider(names_from = itemid, values_from = valuenum,
              values_fill = list(valuenum = NA))

# Create a named vector mapping `itemid` to readable names
itemid_mapping <- c(
  "220045" = "heart_rate",
  "220179" = "non-invasive_blood_pressure_systolic",
  "220180" = "non-invasive_blood_pressure_diastolic",
  "223761" = "temperature_fahrenheit",
  "220210" = "respiratory_rate"
)

# Rename columns based on `itemid_mapping`
colnames(chartevents_tble) <- c(
  "subject_id", "stay_id",
  itemid_mapping[colnames(chartevents_tble)[-c(1:2)]]
)

rm(latest_vitals)
```

```
# Arrange for better readability
chartevents_tble <- chartevents_tble %>% arrange(subject_id, stay_id)

# Ensure columns appear in the correct order
desired_order <- c("subject_id", "stay_id",
                   "heart_rate",
                   "non-invasive_blood_pressure_diastolic",
                   "non-invasive_blood_pressure_systolic",
                   "respiratory_rate",
                   "temperature_fahrenheit")

chartevents_tble <- chartevents_tble[, desired_order]

# View the final dataset
options(width = 1000)
```

```
print(head(chartevents_tble, 10))
```

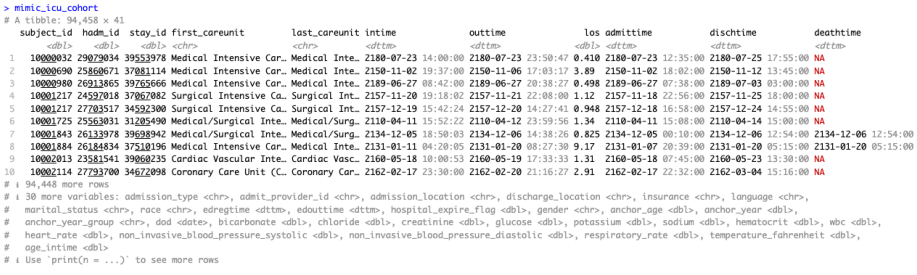
```
# A tibble: 10 x 7
   subject_id  stay_id heart_rate `non-invasive_blood_pressure_diastolic` `non-invasive_bloo
        <dbl>    <dbl>      <dbl>                                   <dbl>
 1   10000032 39553978       91                                      48
 2   10000690 37081114       78                                      56.5
 3   10000980 39765666       76                                     102
 4   10001217 34592300       79.3                                    93.3
 5   10001217 37067082       86                                      90
 6   10001725 31205490       86                                      56
 7   10001843 39698942      124.                                     78
 8   10001884 37510196       49                                      30.5
 9   10002013 39060235       80                                      62
10   10002114 34672098      110.                                     80
```

0.7 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 x 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 18:00:53 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
```

Solution:

```
# Connect to DuckDB
con <- dbConnect(duckdb::duckdb(), dbdir = ":memory:")
```

```r
# Write data into DuckDB
dbWriteTable(con, "icustays_tble", icustays_tble, overwrite = TRUE)
dbWriteTable(con, "admissions_tble", admissions_tble, overwrite = TRUE)
dbWriteTable(con, "patients_tble", patients_tble, overwrite = TRUE)

# Load ICU stays data
icustays_tble <- dbGetQuery(con, "SELECT * FROM icustays_tble")

# Load admissions data
admissions_tble <- dbGetQuery(con, "SELECT * FROM admissions_tble")

# Load patients data
patients_tble <- dbGetQuery(con, "SELECT * FROM patients_tble")
```

```r
# Write data into DuckDB
dbWriteTable(con, "labevents_tble", labevents_tble, overwrite = TRUE)
dbWriteTable(con, "chartevents_tble", chartevents_tble, overwrite = TRUE)

# Load lab/chart events
labevents_tble <- dbGetQuery(con, "SELECT * FROM labevents_tble")
chartevents_tble <- dbGetQuery(con, "SELECT * FROM chartevents_tble")
```

```r
# Merge ICU stays and admissions**
mimic_icu_cohort <- icustays_tble %>%
  inner_join(admissions_tble, by = "subject_id") %>%
  inner_join(patients_tble, by = "subject_id")
```

```
Warning in inner_join(., admissions_tble, by = "subject_id"): Detected an unexpected many-to
i Row 1 of `x` matches multiple rows in `y`.
i Row 48 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship = "many-to-many"` to silence
```

```r
# *Add lab events (most recent lab tests)**
mimic_icu_cohort <- mimic_icu_cohort %>%
  left_join(labevents_tble, by = c("subject_id", "stay_id"))
```

```r
# Add chart events (first recorded vital signs)**
mimic_icu_cohort <- mimic_icu_cohort %>%
  left_join(chartevents_tble, by = c("subject_id", "stay_id"))

#*Ensure each ICU stay corresponds to only one row**
mimic_icu_cohort <- mimic_icu_cohort %>%
  distinct(subject_id, stay_id, .keep_all = TRUE)

# Sort by subject_id and stay_id**
mimic_icu_cohort <- mimic_icu_cohort %>%
```

```
  arrange(subject_id, stay_id)

# Display the final dataset
options(width = 1000)
print(head(mimic_icu_cohort, 10))
```

```
   subject_id hadm_id.x  stay_id                                    first_careunit
1    10000032  29079034 39553978                Medical Intensive Care Unit (MICU)
2    10000690  25860671 37081114                Medical Intensive Care Unit (MICU)
3    10000980  26913865 39765666                Medical Intensive Care Unit (MICU)
4    10001217  27703517 34592300               Surgical Intensive Care Unit (SICU)
5    10001217  24597018 37067082               Surgical Intensive Care Unit (SICU)
6    10001725  25563031 31205490 Medical/Surgical Intensive Care Unit (MICU/SICU) Medical/Su
7    10001843  26133978 39698942 Medical/Surgical Intensive Care Unit (MICU/SICU) Medical/Su
8    10001884  26184834 37510196                Medical Intensive Care Unit (MICU)
9    10002013  23581541 39060235     Cardiac Vascular Intensive Care Unit (CVICU)     Cardia
10   10002114  27793700 34672098                          Coronary Care Unit (CCU)
```

```
# Disconnect from the database
dbDisconnect(con, shutdown = TRUE)
gc()
```

```
           used  (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  1956484 104.5    3896407 208.1  3896407 208.1
Vcells 24277669 185.3   74197602 566.1 83585364 637.8
```

```
rm(admissions_tble,duckdb_table_icustays,patients_tble)
```

```
Warning in rm(admissions_tble, duckdb_table_icustays, patients_tble): object 'duckdb_table_i
```

```
rm(base_shapes,patient_id,final_shapes,
   itemid_labels,itemid_mapping,desired_order,
   extra_shapes)

rm(icustays_tble,labevents_tble,chartevents_tble)
```

0.8 Q8. Exploratory data analysis (EDA)

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

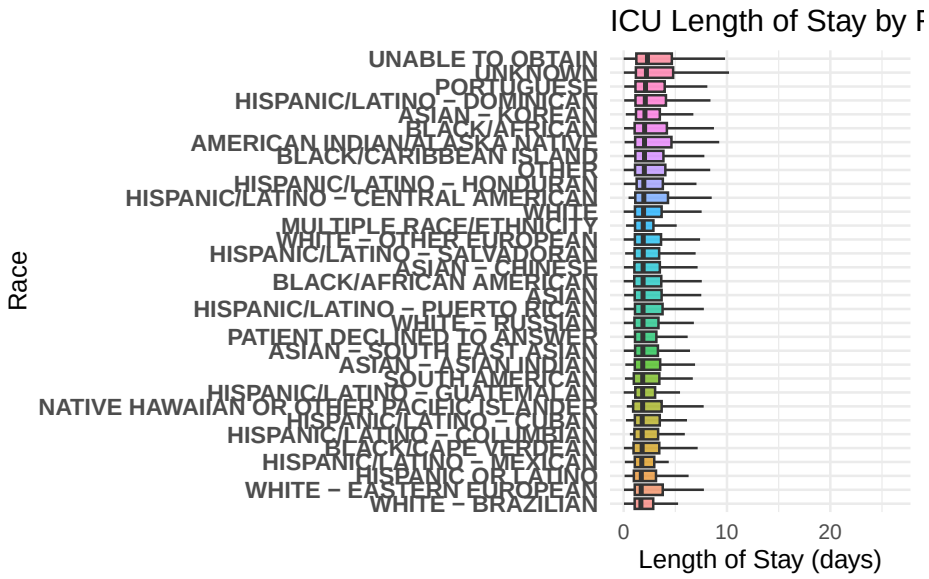
- Length of ICU stay `los` vs demographic variables (race, insurance, marital_status, gender, age at intime)
- Length of ICU stay `los` vs the last available lab measurements before ICU stay
- Length of ICU stay `los` vs the first vital measurements within the ICU stay

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

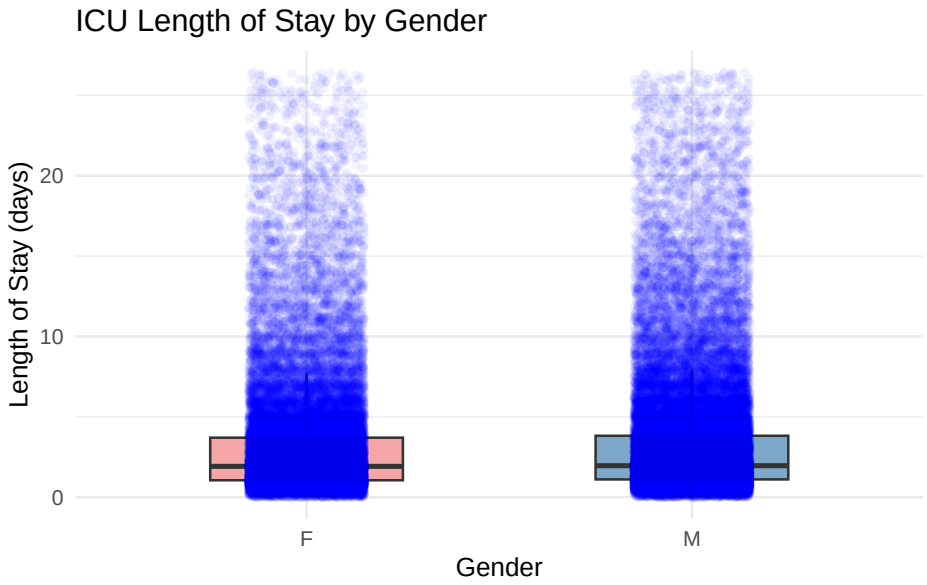
Solution:

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

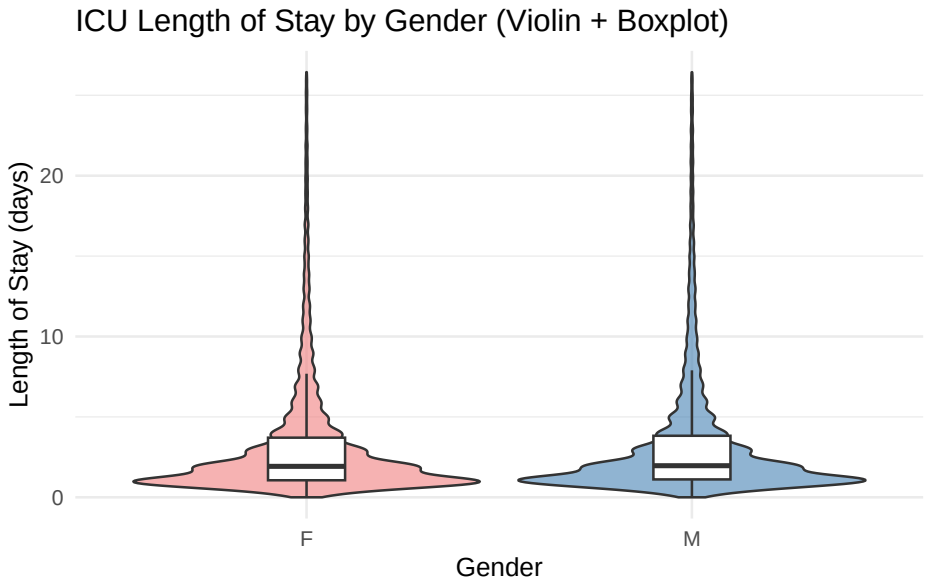
```
# LOS vs Gender
# Remove extreme outliers for better visualization (e.g., focus on LOS < 50 days)
mimic_icu_cohort_filtered <- mimic_icu_cohort %>%
  filter(los < quantile(los, 0.99, na.rm = TRUE))  # Remove top 1% outliers

# Calculate median LOS per race and order the factor levels
race_order <- mimic_icu_cohort_filtered %>%
  group_by(race) %>%
  summarise(median_los = median(los, na.rm = TRUE)) %>%
  arrange(median_los) %>%
  pull(race)

mimic_icu_cohort_filtered$race <- factor(mimic_icu_cohort_filtered$race,
                                         levels = race_order)

# Improved boxplot with better visibility
ggplot(mimic_icu_cohort_filtered, aes(x = race, y = los, fill = race)) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) +
  # Reduce opacity for better visibility
  coord_flip() +  # Flip to horizontal for better readability
  theme_minimal() +  # Cleaner theme
  theme(
    axis.text.y = element_text(size = 10, face = "bold"),  # Bigger labels
    legend.position = "none"  # Hide legend to declutter
  ) +
  labs(
    title = "ICU Length of Stay by Race (Filtered Outliers)",
    x = "Race",
    y = "Length of Stay (days)"
  )
```




```
# LOS vs Gender
# Improved boxplot + jitter
ggplot(mimic_icu_cohort_filtered, aes(x = gender, y = los,
                                      fill = gender)) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7, width = 0.5) +
  # Hide extreme outliers, adjust width
  geom_jitter(width = 0.15, alpha = 0.05, size = 1.2,
              color = "blue") +  # More transparency, smaller points
  theme_minimal() +
  labs(title = "ICU Length of Stay by Gender", x = "Gender",
       y = "Length of Stay (days)") +
  scale_fill_manual(values = c("F" = "lightcoral", "M" = "steelblue")) +
  theme(legend.position = "none")
```

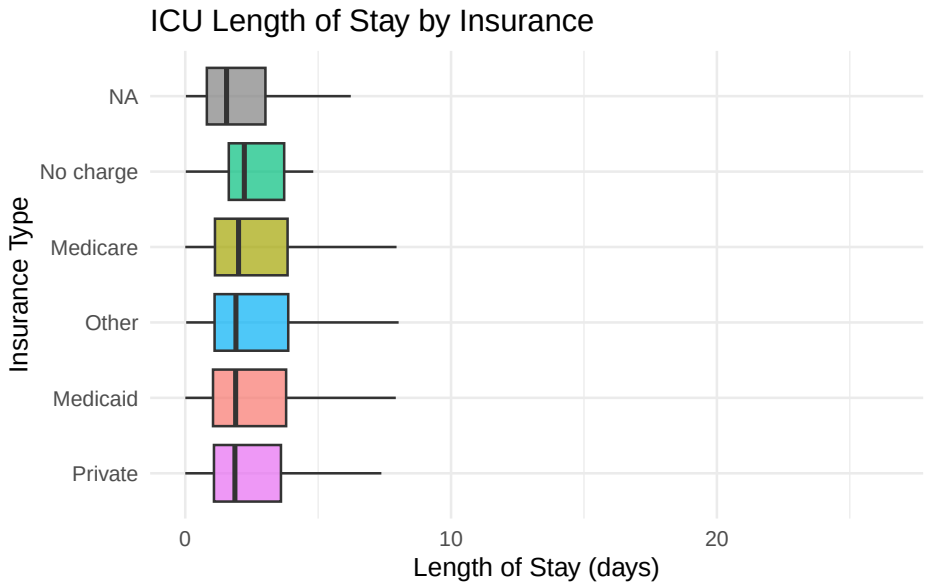



```
# Alternative: Violin plot for better visualization
ggplot(mimic_icu_cohort_filtered, aes(x = gender, y = los, fill = gender)) +
  geom_violin(alpha = 0.6, trim = TRUE) +  # Smooth density distribution
  geom_boxplot(width = 0.2, fill = "white", outlier.shape = NA) +
  # Overlay boxplot inside violin
  theme_minimal() +
  labs(title = "ICU Length of Stay by Gender (Violin + Boxplot)",
       x = "Gender", y = "Length of Stay (days)") +
  scale_fill_manual(values = c("F" = "lightcoral", "M" = "steelblue")) +
  theme(legend.position = "none")
```

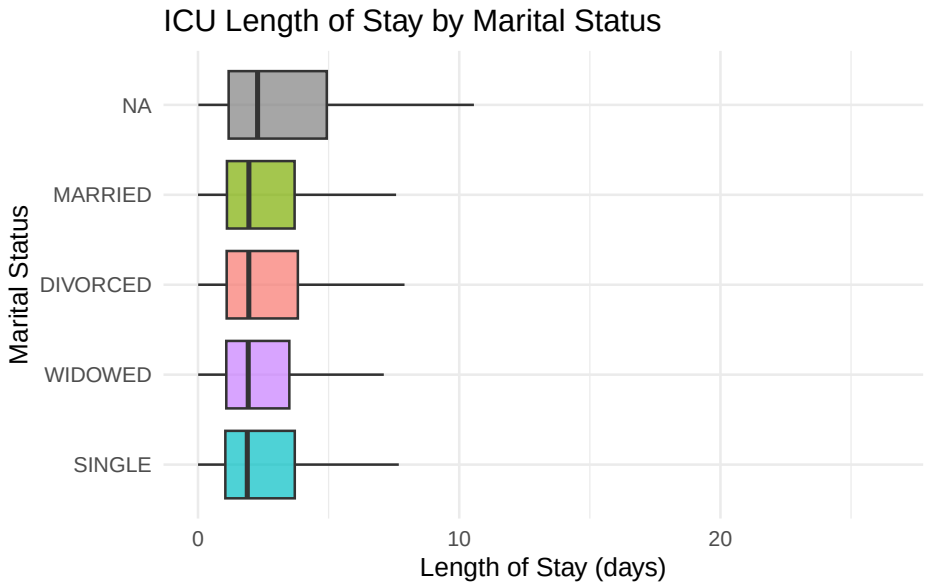


Most ICU stays are relatively short, with patients mainly concentrated at the bottom, while the slender tail indicates that a small number of patients have abnormally prolonged ICU stays. The distribution shapes of the two genders are similar, indicating that the hospitalization time patterns of males and females in the ICU are basically the same, with no significant gender differences.

```
# LOS vs Insurance
ggplot(mimic_icu_cohort_filtered, aes(x = reorder(insurance,
                                                  los, median, na.rm = TRUE),
                                      y = los, fill = insurance)) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) +
  coord_flip() +
  theme_minimal() +
  labs(title = "ICU Length of Stay by Insurance",
       x = "Insurance Type", y = "Length of Stay (days)") +
  theme(legend.position = "none")
```



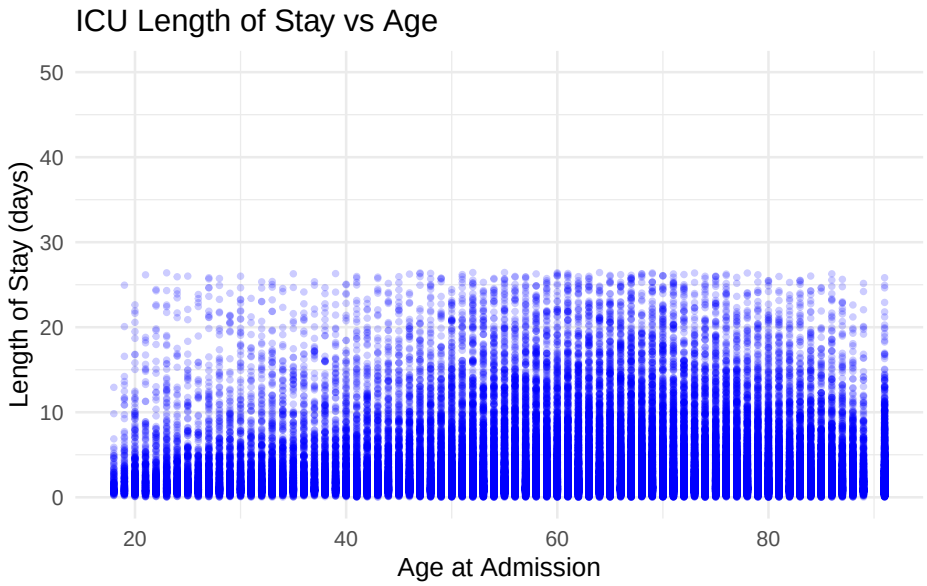

```
# LOS vs Marital Status
ggplot(mimic_icu_cohort_filtered, aes(x = reorder(marital_status, los,
                                                  median, na.rm = TRUE),
                                      y = los, fill = marital_status)) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) +
  coord_flip() +
  theme_minimal() +
  labs(title = "ICU Length of Stay by Marital Status", x = "Marital Status",
       y = "Length of Stay (days)") +
  theme(legend.position = "none")
```




```
# Scatterplot: LOS vs Age
ggplot(mimic_icu_cohort_filtered, aes(x = anchor_age, y = los)) +
  geom_point(alpha = 0.2, color = "blue", size = 0.8) +
  geom_smooth(method = "loess", color = "red", se = TRUE) +
  theme_minimal() +
  labs(title = "ICU Length of Stay vs Age", x = "Age at Admission",
       y = "Length of Stay (days)") +
  scale_y_continuous(limits = c(0, 50))
```

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

```
Warning: Failed to fit group -1.
Caused by error in `predLoess()`:
! workspace required (13114052915) is too large probably because of setting 'se = TRUE'.
```

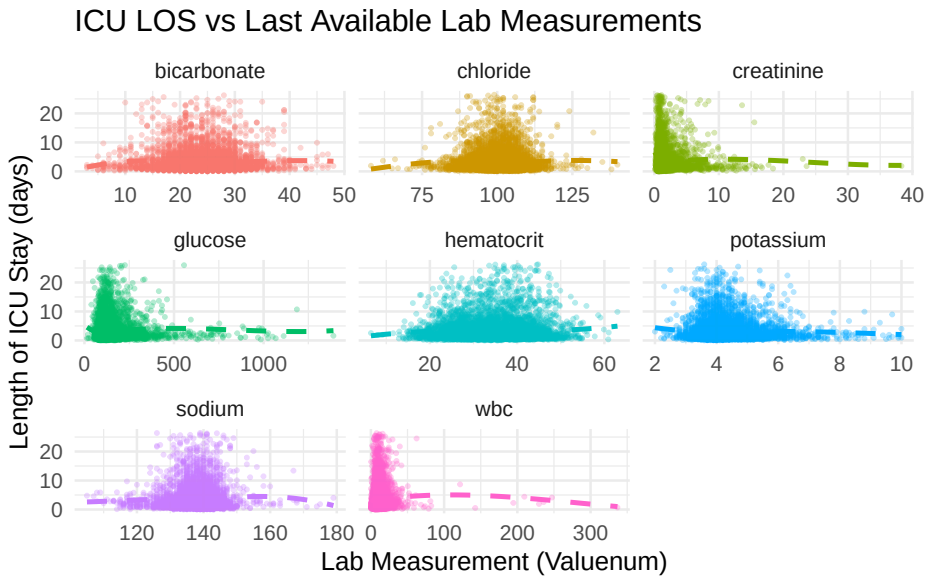


0.8.0.2 Length of ICU stay los vs the last available lab measurements before ICU stay

```
# Convert from wide format to long format
mimic_icu_long <- mimic_icu_cohort_filtered %>%
  pivot_longer(cols = c("bicarbonate", "chloride", "creatinine", "glucose",
                        "potassium", "sodium", "hematocrit", "wbc"),
               names_to = "item_name", values_to = "valuenum") %>%
  filter(!is.na(valuenum) & !is.na(los))  # Remove missing values
```

```
# Sample a smaller subset to improve performance
mimic_icu_sampled <- mimic_icu_long %>%
  slice_sample(n = 50000)  # Adjust this number as needed

# LOS vs Valuenum (Grouped by Lab Test)
ggplot(mimic_icu_sampled, aes(x = valuenum, y = los, color = item_name)) +
  geom_point(alpha = 0.3, size = 0.5) +
  geom_smooth(method = "loess", se = FALSE, linetype = "dashed") +
  facet_wrap(~ item_name, scales = "free_x") +  # Facet by lab test
  theme_minimal() +
  labs(title = "ICU LOS vs Last Available Lab Measurements",
       x = "Lab Measurement (Valuenum)", y = "Length of ICU Stay (days)",
       color = "Lab Test") +
  theme(legend.position = "none")  # Hide legend if unnecessary
```

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




```
rm(mimic_icu_long,mimic_icu_sampled)
```

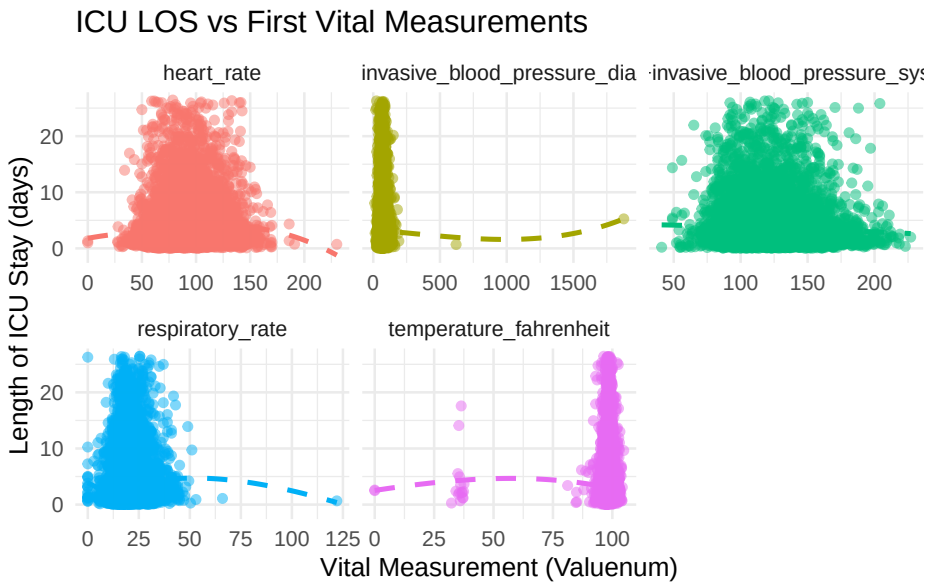
0.8.0.3 Length of ICU stay los vs the first vital measurements within the ICU stay

```
# Convert from wide format to long format using correct column names
mimic_icu_long <- mimic_icu_cohort_filtered %>%
  pivot_longer(cols = c("heart_rate", "non-invasive_blood_pressure_diastolic",
                        "non-invasive_blood_pressure_systolic",
                        "respiratory_rate",
                        "temperature_fahrenheit"),
               names_to = "item_name", values_to = "valuenum") %>%
  filter(!is.na(valuenum) & !is.na(los))  # Remove missing values
```

```
# Sample a smaller subset to improve performance
mimic_icu_sampled <- mimic_icu_long %>%
  slice_sample(n = 50000)  # Adjust this number as needed

# LOS vs Valuenum (Grouped by Lab Test)
ggplot(mimic_icu_sampled, aes(x = valuenum, y = los, color = item_name)) +
  geom_point(alpha = 0.5) +  # Reduce opacity for better visualization
  geom_smooth(method = "loess", se = FALSE, linetype = "dashed") +
  facet_wrap(~ item_name, scales = "free_x") +
  theme_minimal() +
  labs(title = "ICU LOS vs First Vital Measurements",
       x = "Vital Measurement (Valuenum)", y = "Length of ICU Stay (days)",
       color = "Vital Test") +
```

```
  theme(legend.position = "none")  # Hide redundant legend
```

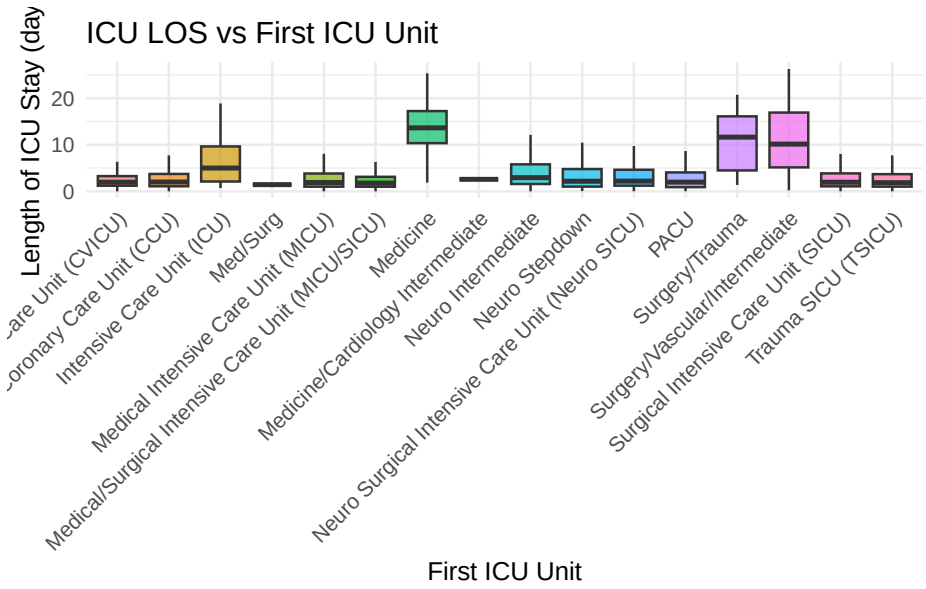
```
`geom_smooth()` using formula = 'y ~ x'
```




```
rm(mimic_icu_long,mimic_icu_sampled)
```

0.8.0.4 Length of ICU stay los vs first ICU unit

```
ggplot(mimic_icu_cohort_filtered, aes(x = first_careunit, y = los,
                                      fill = first_careunit)) +
  geom_boxplot(alpha = 0.7, outlier.shape = NA) +  # Avoid extreme outliers
  theme_minimal() +
  labs(title = "ICU LOS vs First ICU Unit",
       x = "First ICU Unit", y = "Length of ICU Stay (days)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```


ICU LOS vs First ICU Unit
Length of ICU Stay (day
20
10
0
-are Unit (CVICU)
-oronary 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)
PACU
Surgery/Trauma
Surgery/Vascular/Intermediate
Surgical Intensive Care Unit (SICU)
Trauma SICU (TSICU)
First ICU Unit