

Biostat 203B Homework 3

Due Feb 23 @ 11:59PM

Jiaye Tian and UID: 306541095

Table of contents

Q1. Visualizing patient trajectory	4
Q1.1 ADT history	5
Q1.2 ICU stays	15
Q2. ICU stays	21
Q2.1 Ingestion	21
Q2.2 Summary and visualization	22
Q3. admissions data	26
Q3.1 Ingestion	26
Q3.2 Summary and visualization	27
Q4. patients data	33
Q4.1 Ingestion	33
Q4.2 Summary and visualization	34
Q5. Lab results	36
Q6. Vitals from charted events	40
Q7. Putting things together	44
Q8. Exploratory data analysis (EDA)	45

#clear workspace, only keep 'icu_cohort' tibble: rm(list = setdiff(ls(), "icu_cohort"))

#if you want to interactive plot: library(plotly) p <- icu |> ... ggplotly(p)

Display machine information for reproducibility:

```
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: aarch64-apple-darwin20
Running under: macOS Sonoma 14.7.3
```

Matrix products: default

BLAS: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib

LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/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_4.4.2	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).

```
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      1.0.4
```

```
-- 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 become
```

```
library(ggplot2)
library(lubridate)
```

Display your machine memory.

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

```
Totalram: 16.000 GiB
Freeram:   1.172 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.

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.

```
patients_tble <- read_csv("~/mimic/hosp/patients.csv.gz") |>
  print(width = Inf)
```

```
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.
```

```
# A tibble: 364,627 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
7	10000108	M	25	2163	2014 - 2016	NA
8	10000115	M	24	2154	2017 - 2019	NA
9	10000117	F	48	2174	2008 - 2010	NA

```
10 10000161 M 60 2163 2020 - 2022 NA
# i 364,617 more rows
```

```
admissions_tble <- read_csv("~/mimic/hosp/admissions.csv.gz") |>
  print(width = Inf)
```

```
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
```

```
dtm (5): admittime, disctime, 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.
```

```
# A tibble: 546,028 x 16
```

	subject_id	hadm_id	admittime		disctime		deathtime
	<dbl>	<dbl>	<dtm>		<dtm>		<dtm>
1	10000032	22595853	2180-05-06 22:23:00		2180-05-07 17:15:00		NA
2	10000032	22841357	2180-06-26 18:27:00		2180-06-27 18:49:00		NA
3	10000032	25742920	2180-08-05 23:44:00		2180-08-07 17:50:00		NA
4	10000032	29079034	2180-07-23 12:35:00		2180-07-25 17:55:00		NA
5	10000068	25022803	2160-03-03 23:16:00		2160-03-04 06:26:00		NA
6	10000084	23052089	2160-11-21 01:56:00		2160-11-25 14:52:00		NA
7	10000084	29888819	2160-12-28 05:11:00		2160-12-28 16:07:00		NA
8	10000108	27250926	2163-09-27 23:17:00		2163-09-28 09:04:00		NA
9	10000117	22927623	2181-11-15 02:05:00		2181-11-15 14:52:00		NA
10	10000117	27988844	2183-09-18 18:10:00		2183-09-21 16:30:00		NA
	admission_type	admit_provider_id	admission_location		discharge_location		
	<chr>	<chr>	<chr>		<chr>		
1	URGENT	P49AFC	TRANSFER FROM HOSPITAL		HOME		
2	EW EMER.	P784FA	EMERGENCY ROOM		HOME		
3	EW EMER.	P19UTS	EMERGENCY ROOM		HOSPICE		
4	EW EMER.	P060TX	EMERGENCY ROOM		HOME		
5	EU OBSERVATION	P39NWO	EMERGENCY ROOM		<NA>		
6	EW EMER.	P42H7G	WALK-IN/SELF REFERRAL		HOME HEALTH CARE		
7	EU OBSERVATION	P35NE4	PHYSICIAN REFERRAL		<NA>		
8	EU OBSERVATION	P40JML	EMERGENCY ROOM		<NA>		
9	EU OBSERVATION	P47EY8	EMERGENCY ROOM		<NA>		
10	OBSERVATION ADMIT	P13ACE	WALK-IN/SELF REFERRAL		HOME HEALTH CARE		
	insurance	language	marital_status		race		edregtime

	<chr>	<chr>	<chr>	<chr>	<dtm>
1	Medicaid	English	WIDOWED	WHITE	2180-05-06 19:17:00
2	Medicaid	English	WIDOWED	WHITE	2180-06-26 15:54:00
3	Medicaid	English	WIDOWED	WHITE	2180-08-05 20:58:00
4	Medicaid	English	WIDOWED	WHITE	2180-07-23 05:54:00
5	<NA>	English	SINGLE	WHITE	2160-03-03 21:55:00
6	Medicare	English	MARRIED	WHITE	2160-11-20 20:36:00
7	Medicare	English	MARRIED	WHITE	2160-12-27 18:32:00
8	<NA>	English	SINGLE	WHITE	2163-09-27 16:18:00
9	Medicaid	English	DIVORCED	WHITE	2181-11-14 21:51:00
10	Medicaid	English	DIVORCED	WHITE	2183-09-18 08:41:00
	edouttime		hospital_expire_flag		
	<dtm>			<dbl>	
1	2180-05-06	23:30:00		0	
2	2180-06-26	21:31:00		0	
3	2180-08-06	01:44:00		0	
4	2180-07-23	14:00:00		0	
5	2160-03-04	06:26:00		0	
6	2160-11-21	03:20:00		0	
7	2160-12-28	16:07:00		0	
8	2163-09-28	09:04:00		0	
9	2181-11-15	09:57:00		0	
10	2183-09-18	20:20:00		0	

i 546,018 more rows

```
transfers_tble <- read_csv("~/mimic/hosp/transfers.csv.gz") |>
  print(width = Inf)
```

Rows: 2413581 Columns: 7

-- Column specification -----

Delimiter: ","

chr (2): eventtype, careunit

dbl (3): subject_id, hadm_id, transfer_id

dtm (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.

A tibble: 2,413,581 x 7

	subject_id	hadm_id	transfer_id	eventtype	careunit
	<dbl>	<dbl>	<dbl>	<chr>	<chr>
1	10000032	22595853	33258284	ED	Emergency Department

```

2  10000032 22595853      35223874 admit      Transplant
3  10000032 22595853      36904543 discharge UNKNOWN
4  10000032 22841357      34100253 discharge UNKNOWN
5  10000032 22841357      34703856 admit      Transplant
6  10000032 22841357      38112554 ED          Emergency Department
7  10000032 25742920      35509340 admit      Transplant
8  10000032 25742920      35968195 ED          Emergency Department
9  10000032 25742920      38883756 discharge UNKNOWN
10 10000032 29079034      32952584 ED          Emergency Department

```

```

      intime      outtime
      <dtm>      <dtm>
1 2180-05-06 19:17:00 2180-05-06 23:30:00
2 2180-05-06 23:30:00 2180-05-07 17:21:27
3 2180-05-07 17:21:27 NA
4 2180-06-27 18:49:12 NA
5 2180-06-26 21:31:00 2180-06-27 18:49:12
6 2180-06-26 15:54:00 2180-06-26 21:31:00
7 2180-08-06 01:44:00 2180-08-07 17:50:44
8 2180-08-05 20:58:00 2180-08-06 01:44:00
9 2180-08-07 17:50:44 NA
10 2180-07-22 16:24:00 2180-07-23 05:54:00
# i 2,413,571 more rows

```

```

procedures_icd_tble <- read_csv("~/mimic/hosp/procedures_icd.csv.gz") |>
  print(width = Inf)

```

```

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.

```

```

# A tibble: 859,655 x 6

```

```

  subject_id hadm_id seq_num chartdate icd_code icd_version
    <dbl>    <dbl>   <dbl> <date>    <chr>      <dbl>
1  10000032 22595853     1 2180-05-07 5491         9
2  10000032 22841357     1 2180-06-27 5491         9
3  10000032 25742920     1 2180-08-06 5491         9

```



```

4  10000068 25022803      1 2160-03-03 8938      9
5  10000117 27988844      1 2183-09-19 OQS734Z    10
6  10000280 25852320      1 2151-03-18 8938      9
7  10000560 28979390      1 2189-10-16 5551      9
8  10000635 26134563      1 2136-06-19 3734      9
9  10000635 26134563      2 2136-06-19 3728      9
10 10000635 26134563      3 2136-06-19 3727      9
# i 859,645 more rows

```

```

diagnoses_icd_tble <- read_csv("~/mimic/hosp/diagnoses_icd.csv.gz") |>
  print(width = Inf)

```

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.

A tibble: 6,364,488 x 5

	subject_id	hadm_id	seq_num	icd_code	icd_version
	<dbl>	<dbl>	<dbl>	<chr>	<dbl>
1	10000032	22595853	1	5723	9
2	10000032	22595853	2	78959	9
3	10000032	22595853	3	5715	9
4	10000032	22595853	4	07070	9
5	10000032	22595853	5	496	9
6	10000032	22595853	6	29680	9
7	10000032	22595853	7	30981	9
8	10000032	22595853	8	V1582	9
9	10000032	22841357	1	07071	9
10	10000032	22841357	2	78959	9

i 6,364,478 more rows

```

d_icd_procedures_tble <- read_csv("~/mimic/hosp/d_icd_procedures.csv.gz") |>
  print(width = Inf)

```

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.

```

```

# A tibble: 86,423 x 3
  icd_code icd_version
  <chr>      <dbl>
1 0001          9
2 0002          9
3 0003          9
4 0009          9
5 001         10
6 0010          9
7 0011          9
8 0012          9
9 0013          9
10 0014          9
  long_title
  <chr>
1 Therapeutic ultrasound of vessels of head and neck
2 Therapeutic ultrasound of heart
3 Therapeutic ultrasound of peripheral vascular vessels
4 Other therapeutic ultrasound
5 Central Nervous System and Cranial Nerves, Bypass
6 Implantation of chemotherapeutic agent
7 Infusion of drotrecogin alfa (activated)
8 Administration of inhaled nitric oxide
9 Injection or infusion of nesiritide
10 Injection or infusion of oxazolidinone class of antibiotics
# i 86,413 more rows

```

```

d_icd_diagnoses_tble <- read_csv("~/mimic/hosp/d_icd_diagnoses.csv.gz") |>
  print(width = Inf)

```

```

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.

```
# A tibble: 112,107 x 3
  icd_code icd_version long_title
  <chr>      <dbl> <chr>
1 0010          9 Cholera due to vibrio cholerae
2 0011          9 Cholera due to vibrio cholerae el tor
3 0019          9 Cholera, unspecified
4 0020          9 Typhoid fever
5 0021          9 Paratyphoid fever A
6 0022          9 Paratyphoid fever B
7 0023          9 Paratyphoid fever C
8 0029          9 Paratyphoid fever, unspecified
9 0030          9 Salmonella gastroenteritis
10 0031          9 Salmonella septicemia
# i 112,097 more rows
```

```
lab_events <- open_dataset("~/203b/hw/hw3/labevents_pq",
                           format = "parquet") %>%
  select(subject_id, charttime, itemid, storetime, valuenum) %>%
  collect()
print(lab_events, width = Inf)
```

```
# A tibble: 158,374,764 x 5
  subject_id charttime          itemid storetime          valuenum
  <int> <dtm>          <int> <dtm>          <dbl>
1 10000032 2180-03-23 04:51:00 50931 2180-03-23 08:56:00      95
2 10000032 2180-03-23 04:51:00 51071 2180-03-23 09:00:00     NA
3 10000032 2180-03-23 04:51:00 51074 2180-03-23 09:00:00     NA
4 10000032 2180-03-23 04:51:00 51075 2180-03-23 09:00:00     NA
5 10000032 2180-03-23 04:51:00 51079 2180-03-23 09:00:00     NA
6 10000032 2180-03-23 04:51:00 51087 NA                      NA
7 10000032 2180-03-23 04:51:00 51089 2180-03-23 09:15:00     NA
8 10000032 2180-03-23 04:51:00 51090 2180-03-23 09:00:00     NA
9 10000032 2180-03-23 04:51:00 51092 2180-03-23 09:00:00     NA
10 10000032 2180-03-23 04:51:00 50853 2180-03-25 04:06:00     15
# i 158,374,754 more rows
```

```

subject_id <- 10063848

adt_data <- transfers_tble %>%
  filter(subject_id == !!subject_id & !is.na(intime) & !is.na(outtime)) %>%
  mutate(intime = as.POSIXct(intime,
                             format="%Y-%m-%d %H:%M:%S",
                             tz="UTC"),
         outtime = as.POSIXct(outtime,
                              format="%Y-%m-%d %H:%M:%S",
                              tz="UTC"))

lab_data <- lab_events %>%
  filter(subject_id == !!subject_id) %>%
  distinct(subject_id, charttime, itemid, .keep_all = TRUE) %>%
  mutate(charttime = as.POSIXct(charttime, tz="UTC"))

patient_info <- admissions_tble %>%
  filter(subject_id == !!subject_id) %>%
  select(subject_id, race) %>%
  distinct() %>%
  left_join(patients_tble %>% select(subject_id, gender, anchor_age),
            by = "subject_id")

diagnoses_icd <- diagnoses_icd_tble %>%
  mutate(icd_code = str_remove(icd_code, "^0+"),
         icd_version = as.character(icd_version))

d_icd_diagnoses <- d_icd_diagnoses_tble %>%
  mutate(icd_code = str_remove(icd_code, "^0+"),
         icd_version = as.character(icd_version))

diagnosis_data <- diagnoses_icd_tble %>%
  filter(subject_id == !!subject_id) %>%
  left_join(d_icd_diagnoses_tble, by = c("icd_code", "icd_version")) %>%
  left_join(admissions_tble %>% select(subject_id, admittime),
            by = "subject_id") %>%
  arrange(admittime) %>%
  slice(1:3)

```

Warning in left_join(., admissions_tble %>% select(subject_id, admittime), : Detected an unequal number of rows in the left and right tables.
 i Row 1 of `x` matches multiple rows in `y`.
 i Row 3309 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.

```
procedure_data <- procedures_icd_tble %>%
  filter(subject_id == !!subject_id & !is.na(chartdate)) %>%
  left_join(d_icd_procedures_tble, by = "icd_code") %>%
  mutate(chartdate = as.POSIXct(chartdate,
                                format="%Y-%m-%d",
                                tz="UTC"),
         procedure_label = ifelse(is.na(long_title),
                                "Unknown Procedure",
                                str_trunc(long_title, 30,
                                           side = "right")))

title_text <- paste("Patient", subject_id, ",",
                  patient_info$gender, ",",
                  patient_info$anchor_age, "years old,",
                  patient_info$race)

subtitle_text <- diagnosis_data %>%
  filter(!is.na(long_title)) %>%
  pull(long_title) %>%
  paste(collapse = "\n")

procedure_data <- procedure_data %>%
  mutate(procedure_label = str_trunc(long_title, 30, side = "right"))

ggplot() +

  geom_point(data = procedure_data,
            aes(x = chartdate, y = "Procedure",
               shape = factor(procedure_label)),
            size = 4) +

  # ADT
  geom_segment(data = adt_data,
             aes(x = intime, xend = outtime, y = "ADT",
                color = careunit,
                linewidth = ifelse(str_detect(careunit,
                                              "ICU|CCU|SICU"), 5, 2))
             ) +
  scale_linewidth_identity() +
```

```

# Lab
geom_point(data = lab_data,
           aes(x = charttime, y = "Lab"),
           shape = 3, size = 3, color = "black") +

scale_x_datetime(date_labels = "%b %d", date_breaks = "7 days") +

theme_minimal() +
labs(title = title_text,
     subtitle = subtitle_text,
     x = "Calendar Time",
     y = "Event Type",
     color = "Care Unit",
     shape = "Procedure") +

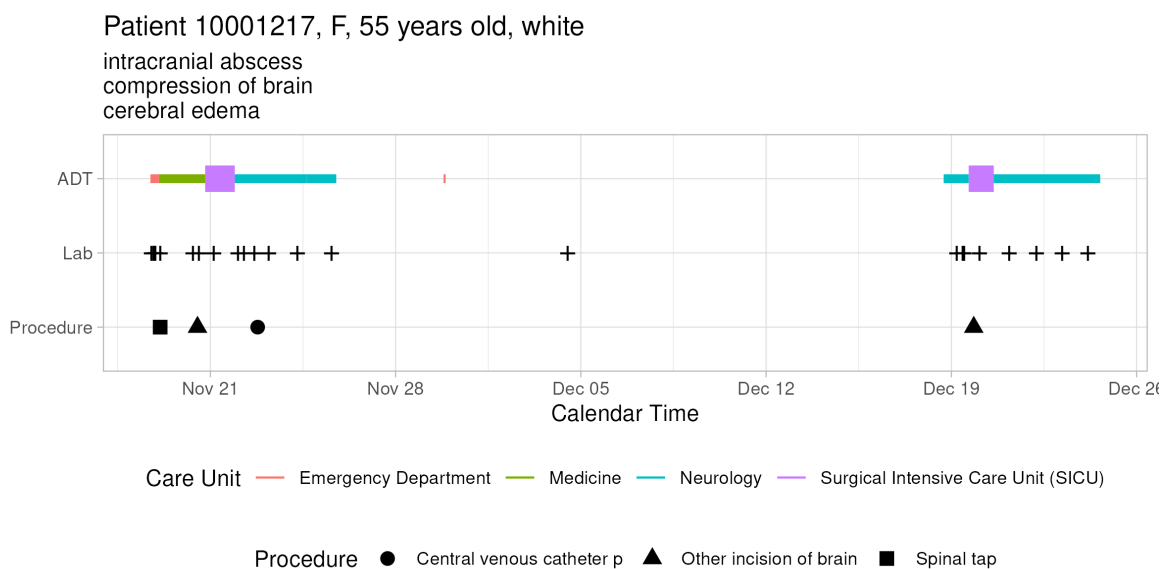
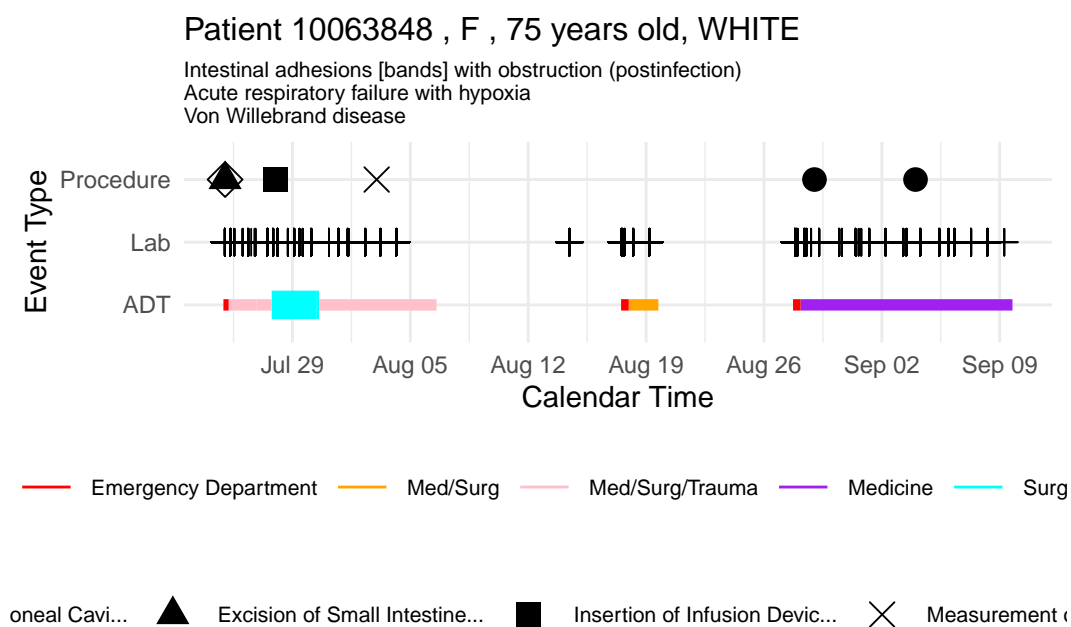
# Care Unit
scale_color_manual(values = c("red",
                              "orange",
                              "pink",
                              "purple",
                              "cyan")) +

# Procedure
scale_shape_manual(values = c(16, 17, 15, 4, 5)) +

guides(
  color = guide_legend(title = "Care Unit", nrow = 1),
  shape = guide_legend(title = "Procedure", nrow = 1)
) +

theme(
  legend.position = "bottom",
  legend.box = "vertical",
  legend.spacing.y = unit(0.5, "cm"),
  legend.text = element_text(size = 8), # legend
  plot.title = element_text(size = 12, hjust = 0), #
  plot.subtitle = element_text(size = 8, hjust = 0), #
  legend.key.size = unit(0.8, "cm") # legend
)

```



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

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.

```
icu_stays_tble <- read_csv("~/mimic/icu/icustays.csv.gz") |>
  print(width = Inf)
```

Rows: 94458 Columns: 8

-- Column specification -----

Delimiter: ","

chr (2): first_careunit, last_careunit

dbl (4): subject_id, hadm_id, stay_id, los

dtm (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.

A tibble: 94,458 x 8

	subject_id	hadm_id	stay_id	first_careunit
	<dbl>	<dbl>	<dbl>	<chr>
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	24597018	37067082	Surgical Intensive Care Unit (SICU)
5	10001217	27703517	34592300	Surgical Intensive Care Unit (SICU)
6	10001725	25563031	31205490	Medical/Surgical Intensive Care Unit (MICU/SICU)
7	10001843	26133978	39698942	Medical/Surgical Intensive Care Unit (MICU/SICU)
8	10001884	26184834	37510196	Medical Intensive Care Unit (MICU)
9	10002013	23581541	39060235	Cardiac Vascular Intensive Care Unit (CVICU)
10	10002114	27793700	34672098	Coronary Care Unit (CCU)
	last_careunit		intime	
	<chr>		<dtm>	
1	Medical Intensive Care Unit (MICU)		2180-07-23 14:00:00	
2	Medical Intensive Care Unit (MICU)		2150-11-02 19:37:00	
3	Medical Intensive Care Unit (MICU)		2189-06-27 08:42:00	
4	Surgical Intensive Care Unit (SICU)		2157-11-20 19:18:02	
5	Surgical Intensive Care Unit (SICU)		2157-12-19 15:42:24	
6	Medical/Surgical Intensive Care Unit (MICU/SICU)		2110-04-11 15:52:22	
7	Medical/Surgical Intensive Care Unit (MICU/SICU)		2134-12-05 18:50:03	
8	Medical Intensive Care Unit (MICU)		2131-01-11 04:20:05	
9	Cardiac Vascular Intensive Care Unit (CVICU)		2160-05-18 10:00:53	
10	Coronary Care Unit (CCU)		2162-02-17 23:30:00	
	outtime	los		


```

      <dtm>          <dbl>
1 2180-07-23 23:50:47 0.410
2 2150-11-06 17:03:17 3.89
3 2189-06-27 20:38:27 0.498
4 2157-11-21 22:08:00 1.12
5 2157-12-20 14:27:41 0.948
6 2110-04-12 23:59:56 1.34
7 2134-12-06 14:38:26 0.825
8 2131-01-20 08:27:30 9.17
9 2160-05-19 17:33:33 1.31
10 2162-02-20 21:16:27 2.91
# i 94,448 more rows

```

```

chartevents <- read_csv("~/203b/hw/hw3/chartevents_filtered.csv") |>
  print(width = Inf)

```

```

Rows: 5069858 Columns: 11

```

```

-- Column specification -----
Delimiter: ","
chr  (1): valueuom
dbl  (8): subject_id, hadm_id, stay_id, caregiver_id, itemid, value, valuenu...
dtm  (2): charttime, storetime

```

```

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.

```

```

# A tibble: 5,069,858 x 11

```

```

  subject_id hadm_id stay_id caregiver_id charttime
      <dbl>   <dbl>   <dbl>         <dbl> <dtm>
1    1000032 29079034 39553978      18704 2180-07-23 14:00:00
2    1000032 29079034 39553978      18704 2180-07-23 14:11:00
3    1000032 29079034 39553978      18704 2180-07-23 14:11:00
4    1000032 29079034 39553978      18704 2180-07-23 14:12:00
5    1000032 29079034 39553978      18704 2180-07-23 14:12:00
6    1000032 29079034 39553978      18704 2180-07-23 14:30:00
7    1000032 29079034 39553978      18704 2180-07-23 14:30:00
8    1000032 29079034 39553978      18704 2180-07-23 14:30:00
9    1000032 29079034 39553978      18704 2180-07-23 14:30:00
10   1000032 29079034 39553978      18704 2180-07-23 15:00:00
  storetime          itemid value valuenum valueuom warning
    <dtm>          <dbl> <dbl>   <dbl> <chr>      <dbl>
1 2180-07-23 14:20:00 223761  98.7    98.7 °F          0

```

2	2180-07-23	14:17:00	220179	84	84	mmHg	0
3	2180-07-23	14:17:00	220181	56	56	mmHg	0
4	2180-07-23	14:17:00	220045	91	91	bpm	0
5	2180-07-23	14:17:00	220210	24	24	insp/min	0
6	2180-07-23	14:43:00	220045	93	93	bpm	0
7	2180-07-23	14:43:00	220179	95	95	mmHg	0
8	2180-07-23	14:43:00	220181	67	67	mmHg	0
9	2180-07-23	14:43:00	220210	21	21	insp/min	0
10	2180-07-23	15:34:00	220045	94	94	bpm	0

i 5,069,848 more rows

```
icu_stays <- icu_stays_tble %>%
  filter(subject_id == 10001217) %>%
  mutate(intime = as.POSIXct(intime, tz = "UTC"),
         outtime = as.POSIXct(outtime, tz = "UTC")) %>%
  select(subject_id, stay_id, intime, outtime)

vitals_data <- chartevents %>%
  filter(subject_id == 10001217, !is.na(valuenum)) %>%
  select(subject_id, stay_id, charttime, itemid, valuenum) %>%
  mutate(
    stay_id = as.double(stay_id),
    charttime = as.POSIXct(charttime, tz = "UTC")
  ) %>%
  inner_join(icu_stays, by = "stay_id") %>%
  filter(charttime >= intime & charttime <= outtime)

print(vitals_data)
```

A tibble: 218 x 8

	subject_id.x	stay_id	charttime	itemid	valuenum	subject_id.y
	<dbl>	<dbl>	<dtm>	<dbl>	<dbl>	<dbl>
1	10001217	37067082	2157-11-20 19:19:00	220045	86	10001217
2	10001217	37067082	2157-11-20 19:19:00	220179	151	10001217
3	10001217	37067082	2157-11-20 19:19:00	220181	104	10001217
4	10001217	37067082	2157-11-20 19:19:00	220210	18	10001217
5	10001217	37067082	2157-11-20 19:31:00	223761	98.5	10001217
6	10001217	37067082	2157-11-20 20:00:00	220045	91	10001217
7	10001217	37067082	2157-11-20 20:00:00	220179	143	10001217
8	10001217	37067082	2157-11-20 20:00:00	220181	95	10001217
9	10001217	37067082	2157-11-20 20:00:00	220210	24	10001217
10	10001217	37067082	2157-11-20 21:00:00	220045	95	10001217

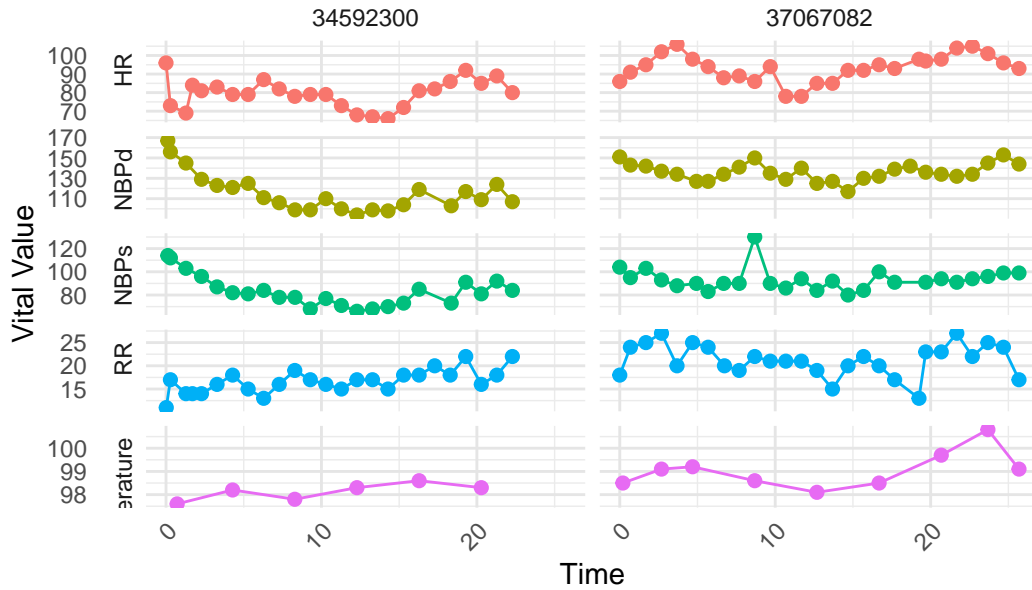
```
# i 208 more rows
# i 2 more variables: intime <dtm>, outtime <dtm>
```

```
vitals_data <- vitals_data %>%
  mutate(vital_label = case_when(
    itemid %in% c(220045) ~ "HR",
    itemid %in% c(220179) ~ "NBPd",
    itemid %in% c(220181) ~ "NBPs",
    itemid %in% c(220210) ~ "RR",
    itemid %in% c(223761) ~ "Temperature",
    TRUE ~ NA_character_
  )) %>%
  filter(!is.na(vital_label)) %>%
  group_by(stay_id) %>%
  mutate(relative_charttime = as.numeric(difftime(charttime, min(charttime), units = "hours"))
  ungroup()
```

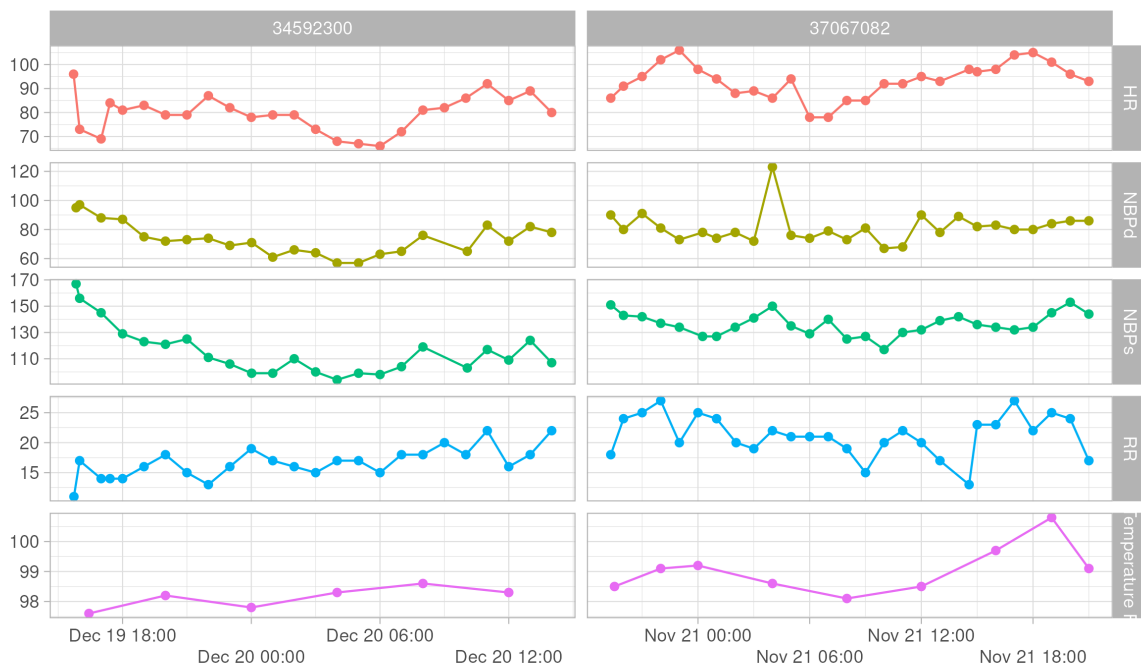
```
ggplot(vitals_data, aes(x = relative_charttime, y = valuenum, color = vital_label)) +
  geom_line() +
  geom_point(size = 2) +
  facet_grid(vital_label ~ stay_id, scales = "free_y", switch = "y") +
  theme_minimal() +
  labs(
    title = paste("Patient", unique(vitals_data$subject_id), "ICU stays - Vitals"),
    x = "Time",
    y = "Vital Value"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    strip.text.y = element_text(angle = 0, hjust = 1),
    legend.position = "none",
    panel.grid.major = element_line(color = "grey90")
  )
```

Warning: Unknown or uninitialised column: `subject_id`.

Patient ICU stays – Vitals



Patient 10001217 ICU stays - Vitals



Do a similar visualization for the patient 10063848.

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 (M
10000690,25860671,37081114,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (M
10000980,26913865,39765666,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (M
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 (M
10002013,23581541,39060235,Cardiac Vascular Intensive Care Unit (CVICU),Cardiac Vascular Int
```

Q2.1 Ingestion

Import icustays.csv.gz as a tibble icustays_tble.

```
icustays_tble <- read_csv("~/mimic/icu/icustays.csv.gz") |>
  print(width = Inf)
```

```
Rows: 94458 Columns: 8
```

```
-- Column specification -----
```

```
Delimiter: ","
```

```
chr (2): first_careunit, last_careunit
```

```
dbl (4): subject_id, hadm_id, stay_id, los
```

```
dtm (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.
```

```
# A tibble: 94,458 x 8
```

```
  subject_id hadm_id stay_id first_careunit
    <dbl>    <dbl>   <dbl> <chr>
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 24597018 37067082 Surgical Intensive Care Unit (SICU)
5  10001217 27703517 34592300 Surgical Intensive Care Unit (SICU)
6  10001725 25563031 31205490 Medical/Surgical Intensive Care Unit (MICU/SICU)
7  10001843 26133978 39698942 Medical/Surgical Intensive Care Unit (MICU/SICU)
8  10001884 26184834 37510196 Medical Intensive Care Unit (MICU)
9  10002013 23581541 39060235 Cardiac Vascular Intensive Care Unit (CVICU)
10 10002114 27793700 34672098 Coronary Care Unit (CCU)
    last_careunit      intime
    <chr>             <dtm>
1 Medical Intensive Care Unit (MICU)      2180-07-23 14:00:00
2 Medical Intensive Care Unit (MICU)      2150-11-02 19:37:00
3 Medical Intensive Care Unit (MICU)      2189-06-27 08:42:00
4 Surgical Intensive Care Unit (SICU)      2157-11-20 19:18:02
5 Surgical Intensive Care Unit (SICU)      2157-12-19 15:42:24
6 Medical/Surgical Intensive Care Unit (MICU/SICU) 2110-04-11 15:52:22
7 Medical/Surgical Intensive Care Unit (MICU/SICU) 2134-12-05 18:50:03
8 Medical Intensive Care Unit (MICU)      2131-01-11 04:20:05
9 Cardiac Vascular Intensive Care Unit (CVICU) 2160-05-18 10:00:53
10 Coronary Care Unit (CCU)                2162-02-17 23:30:00
    outtime      los
    <dtm>        <dbl>
1 2180-07-23 23:50:47 0.410
2 2150-11-06 17:03:17 3.89
3 2189-06-27 20:38:27 0.498
4 2157-11-21 22:08:00 1.12
5 2157-12-20 14:27:41 0.948
6 2110-04-12 23:59:56 1.34
7 2134-12-06 14:38:26 0.825
8 2131-01-20 08:27:30 9.17
9 2160-05-19 17:33:33 1.31
10 2162-02-20 21:16:27 2.91
# i 94,448 more rows

```

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.

```

unique_subjects <- icustays_tble %>%
  distinct(subject_id) %>%

```

```
nrow()
print(unique_subjects)
```

```
[1] 65366
```

```
icu_stay_counts <- icustays_tble %>%
  count(subject_id) %>%
  arrange(desc(n))

print(icu_stay_counts)
```

```
# A tibble: 65,366 x 2
```

	subject_id	n
	<dbl>	<int>
1	12468016	41
2	18358138	37
3	17585185	34
4	17295976	31
5	13269859	30
6	18676703	27
7	12517625	26
8	11281568	25
9	15229355	25
10	15455517	25

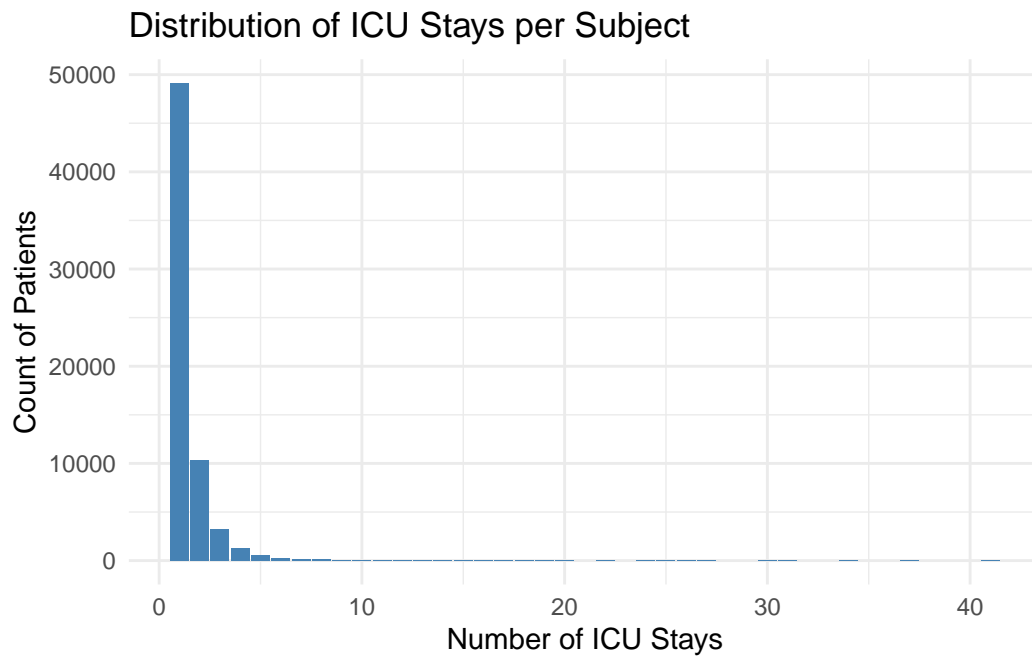
```
# i 65,356 more rows
```

```
icu_summary <- icu_stay_counts %>%
  summarize(
    mean_stays = mean(n),
    median_stays = median(n),
    min_stays = min(n),
    max_stays = max(n)
  )
print(icu_summary)
```

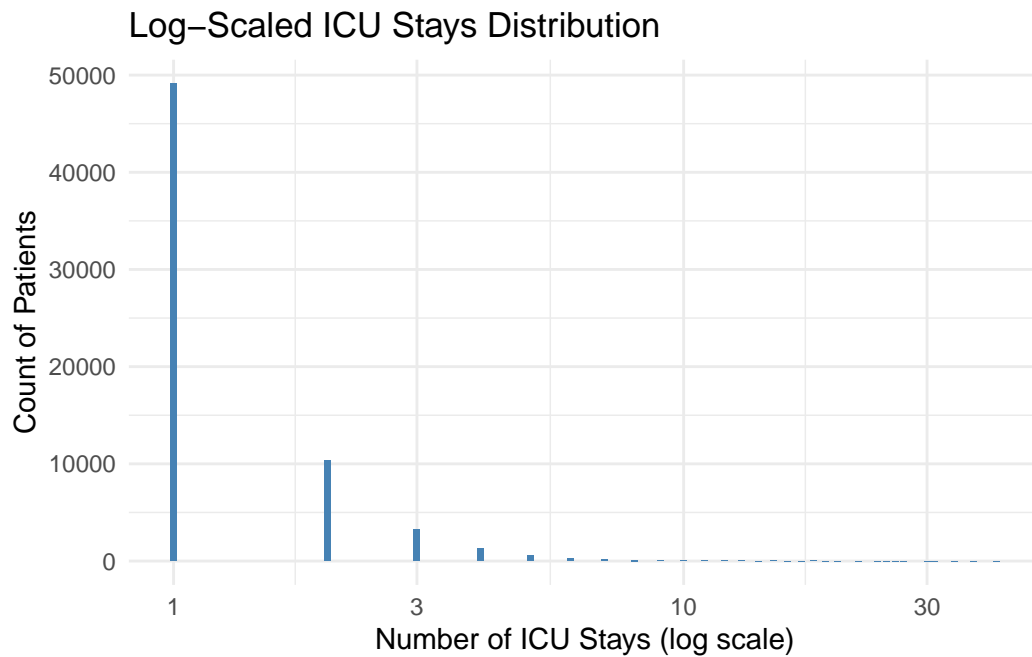
```
# A tibble: 1 x 4
```

	mean_stays	median_stays	min_stays	max_stays
	<dbl>	<dbl>	<int>	<int>
1	1.45	1	1	41

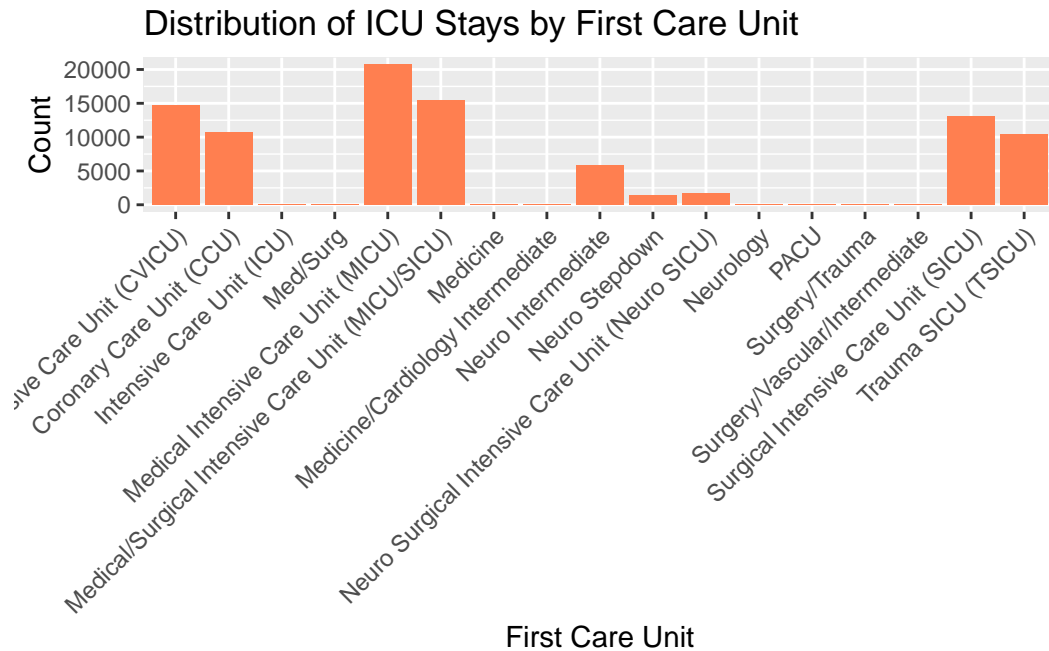
```
ggplot(icu_stay_counts, aes(x = n)) +
  geom_bar(fill = "steelblue") +
  labs(title = "Distribution of ICU Stays per Subject",
       x = "Number of ICU Stays",
       y = "Count of Patients") +
  theme_minimal()
```



```
ggplot(icu_stay_counts, aes(x = n)) +
  geom_bar(fill = "steelblue") +
  scale_x_log10() +
  labs(title = "Log-Scaled ICU Stays Distribution",
       x = "Number of ICU Stays (log scale)",
       y = "Count of Patients") +
  theme_minimal()
```

```
ggplot(icustays_tble, aes(x = first_careunit)) +  
  geom_bar(fill = "coral") +  
  labs(title = "Distribution of ICU Stays by First Care Unit",  
        x = "First Care Unit",  
        y = "Count") +  
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



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_
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,HOS
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.,P060TX,EMERGENCY ROOM,HOS
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 REFER
10000084,29888819,2160-12-28 05:11:00,2160-12-28 16:07:00,,EU OBSERVATION,P35NE4,PHYSICIAN RI
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`.

```
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
```

```
dtm  (5): admittime, disctime, 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.
```

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.

```
unique_hadm <- admissions_tble %>% distinct(hadm_id) %>% nrow()
unique_subjects <- admissions_tble %>% distinct(subject_id) %>% nrow()

cat("Unique hospital admissions (hadm_id):", unique_hadm, "\n")
```

```
Unique hospital admissions (hadm_id): 546028
```

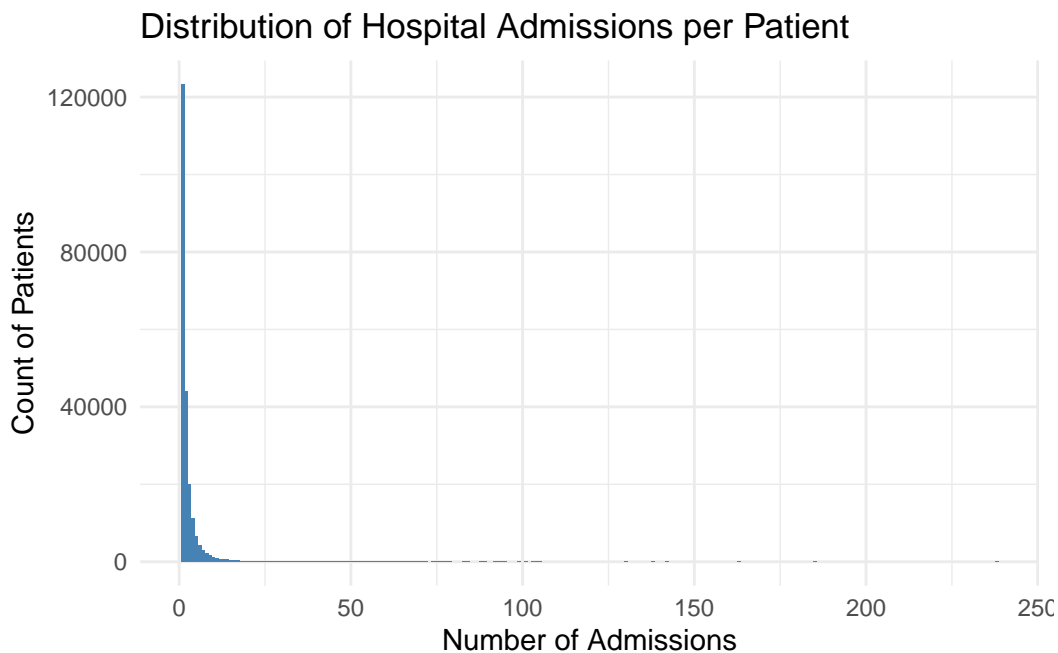
```
cat("Unique patients (subject_id):", unique_subjects, "\n")
```

Unique patients (subject_id): 223452

```
admissions_tble <- admissions_tble %>%  
  mutate(los = difftime(disctime, admittime, units = "days"))  
  
summary(admissions_tble$los)
```

```
Length    Class      Mode  
546028 difftime  numeric
```

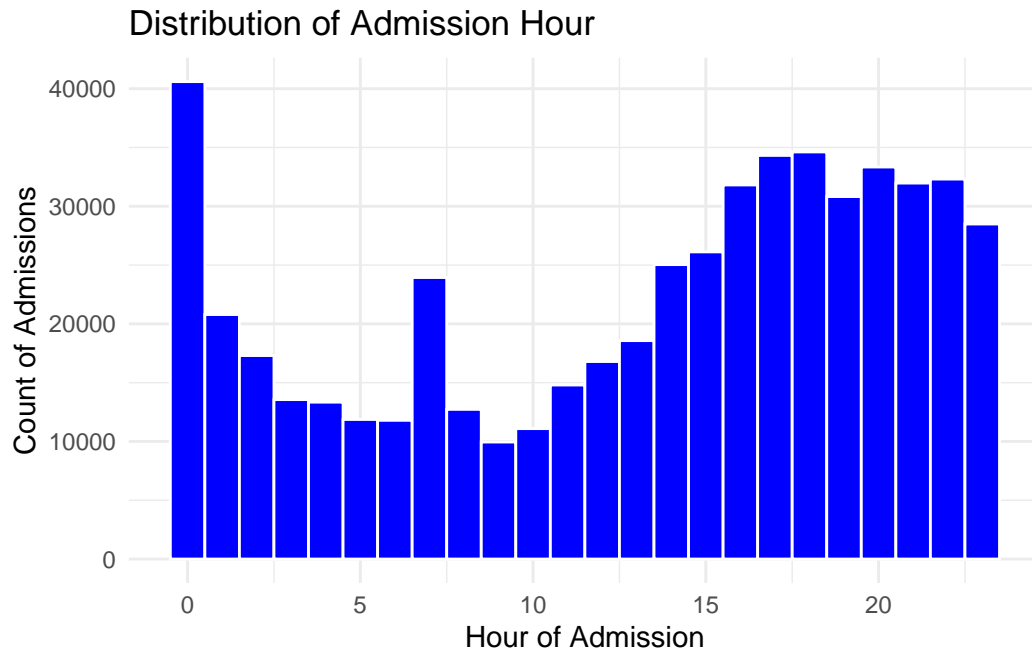
```
admission_counts <- admissions_tble %>%  
  count(subject_id) %>%  
  arrange(desc(n))  
  
ggplot(admission_counts, aes(x = n)) +  
  geom_bar(fill = "steelblue") +  
  labs(title = "Distribution of Hospital Admissions per Patient",  
       x = "Number of Admissions",  
       y = "Count of Patients") +  
  theme_minimal()
```



```
#      1      50+
#
#      50
admissions_tble %>%
  count(subject_id) %>%
  filter(n > 50)
```

```
# A tibble: 119 x 2
  subject_id     n
    <dbl> <int>
1  10108435     53
2  10123949     56
3  10264646     94
4  10312715     51
5  10427568     63
6  10577647     89
7  10578325     74
8  10580201     66
9  10714009    163
10 10913302     78
# i 109 more rows
```

```
#      0      00:00
#      3 - 9
#      0      admittime = "00:00:00"
admissions_tble %>%
  mutate(admit_hour = hour(admittime)) %>%
  ggplot(aes(x = admit_hour)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "white") +
  labs(title = "Distribution of Admission Hour",
       x = "Hour of Admission",
       y = "Count of Admissions") +
  theme_minimal()
```



```
admissions_tble %>%
  filter(hour(admittime) == 0) %>%
  count(admittime)
```

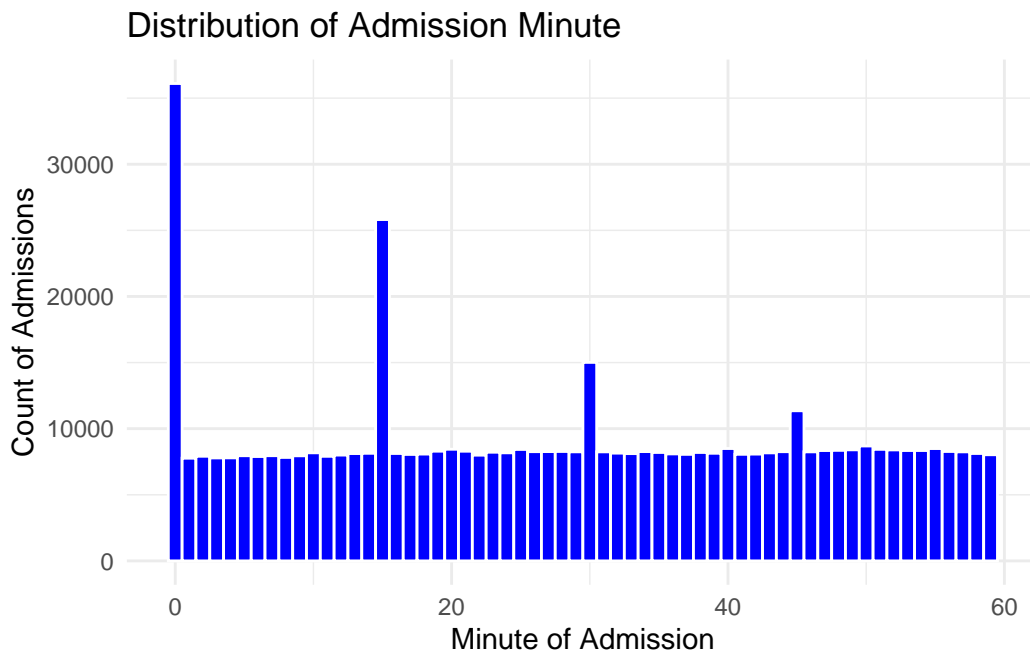
```
# A tibble: 36,153 x 2
```

	admittime	n
	<dtm>	<int>
1	2110-01-12 00:00:00	1
2	2110-01-12 00:34:00	1
3	2110-01-13 00:00:00	1
4	2110-01-23 00:00:00	1
5	2110-01-27 00:10:00	1
6	2110-01-27 00:32:00	1
7	2110-01-28 00:00:00	1
8	2110-01-28 00:12:00	1
9	2110-01-31 00:00:00	1
10	2110-02-03 00:00:00	1

```
# i 36,143 more rows
```

```
admissions_tble %>%
  mutate(admit_minute = minute(admittime)) %>%
  ggplot(aes(x = admit_minute)) +
```

```
geom_histogram(binwidth = 1, fill = "blue", color = "white") +
labs(title = "Distribution of Admission Minute",
     x = "Minute of Admission",
     y = "Count of Admissions") +
theme_minimal()
```



```
# 0 15 30 45          15/30/45
#
#      vs

admissions_tble %>%
  mutate(admit_minute = minute(admittime)) %>%
  count(admit_minute) %>%
  arrange(desc(n))
```

```
# A tibble: 60 x 2
  admit_minute     n
    <int> <int>
1         0 36108
2        15 25818
3        30 15015
4        45 11357
```

```

5          50  8692
6          40  8501
7          55  8499
8          20  8447
9          51  8438
10         25  8431
# i 50 more rows

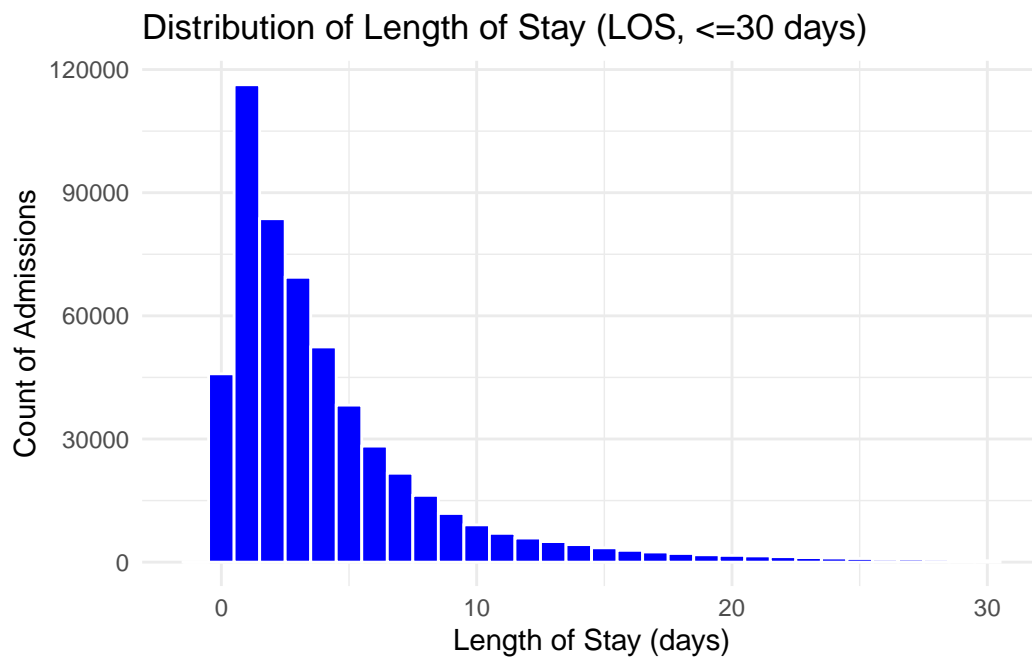
```

```

ggplot(admissions_tble %>% filter(los <= 30), aes(x = los)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "white") +
  labs(title = "Distribution of Length of Stay (LOS, 30 days)",
       x = "Length of Stay (days)",
       y = "Count of Admissions") +
  theme_minimal()

```

Don't know how to automatically pick scale for object of type <difftime>.
Defaulting to continuous.



```

#   LOS      1-7      >30
#           100+
admissions_tble %>%

```



```
filter(los > 100) %>%
select(subject_id, hadm_id, admittance, dischtime, los)
```

```
# A tibble: 242 x 5
  subject_id hadm_id admittance dischtime los
  <dbl>      <dbl> <dtm>      <dtm>      <dbl>
1  10164344 22658293 2194-07-08 15:14:00 2194-11-30 11:00:00 144.8236 days
2  10186976 20911819 2120-12-20 19:41:00 2121-06-24 01:30:00 185.2424 days
3  10201645 24687711 2131-08-07 17:58:00 2132-01-10 11:30:00 155.7306 days
4  10253349 24426241 2189-11-24 08:43:00 2190-05-13 16:08:00 170.3090 days
5  10253349 26415640 2190-05-23 01:00:00 2191-10-20 14:30:00 515.5625 days
6  10337961 26061931 2118-07-26 16:13:00 2118-11-04 16:00:00 100.9910 days
7  10416715 24843066 2181-04-19 14:55:00 2181-08-19 14:05:00 121.9653 days
8  10519706 29552796 2184-08-26 00:19:00 2185-07-13 14:13:00 321.5792 days
9  10636904 22554647 2112-02-11 15:08:00 2112-07-09 19:25:00 149.1785 days
10 10636904 29894505 2113-05-07 00:19:00 2113-09-19 16:18:00 135.6660 days
# i 232 more rows
```

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`.

```
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.

Q4.2 Summary and visualization

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

```
# 1. gender
gender_summary <- patients_tble %>%
  count(gender) %>%
  mutate(percentage = n / sum(n) * 100)

print(gender_summary)
```

```
# A tibble: 2 x 3
  gender      n percentage
  <chr>   <int>     <dbl>
1 F      191984     52.7
2 M      172643     47.3
```

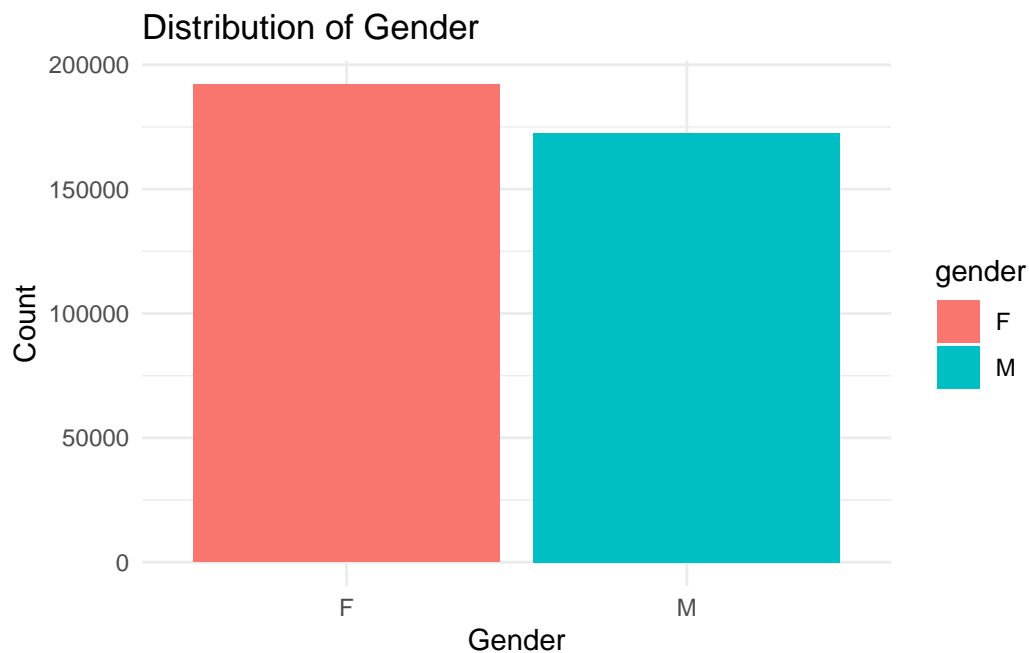
```
# 2. anchor_age
age_summary <- patients_tble %>%
  summarise(
    mean_age = mean(anchor_age, na.rm = TRUE),
    median_age = median(anchor_age, na.rm = TRUE),
    min_age = min(anchor_age, na.rm = TRUE),
    max_age = max(anchor_age, na.rm = TRUE)
  )

print(age_summary)
```

```
# A tibble: 1 x 4
  mean_age median_age min_age max_age
  <dbl>      <dbl>   <dbl>   <dbl>
1    48.9         48     18     91
```

```
#
# 1. gender
gender_plot <- ggplot(patients_tble, aes(x = gender, fill = gender)) +
  geom_bar() +
  labs(
    title = "Distribution of Gender",
    x = "Gender",
    y = "Count"
  ) +
  theme_minimal()

print(gender_plot)
```



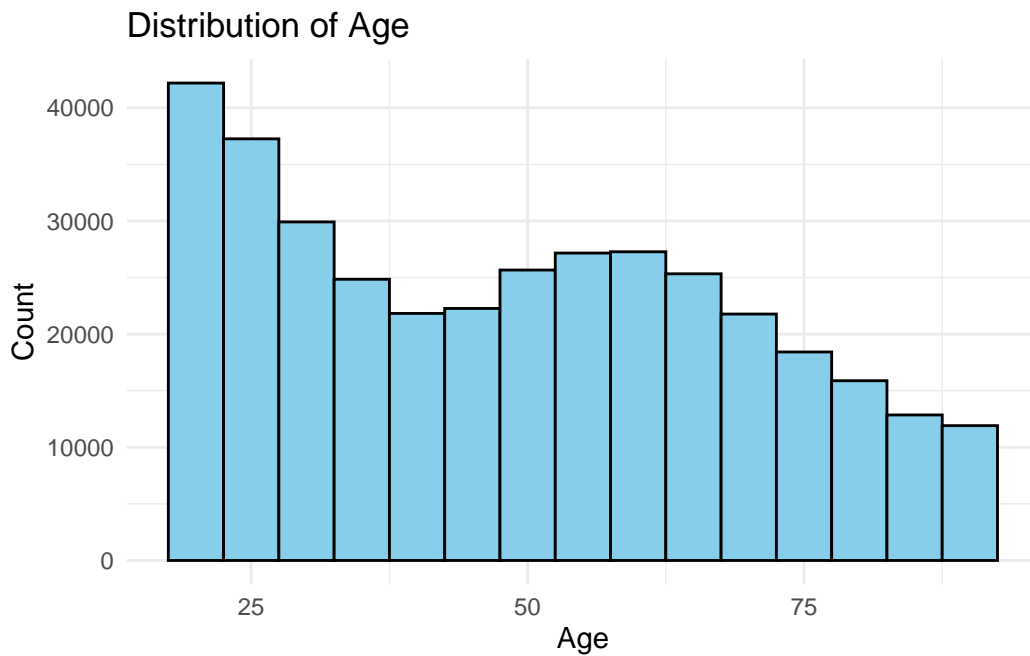
```
# 2. anchor_age
age_plot <- ggplot(patients_tble, aes(x = anchor_age)) +
  geom_histogram(binwidth = 5, fill = "skyblue", color = "black") +
  labs(
```

```

    title = "Distribution of Age",
    x = "Age",
    y = "Count"
) +
theme_minimal()

print(age_plot)

```



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,"I
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,PRES
8,10000032,,36092842,51090,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,M
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 x 10
  subject_id stay_id bicarbonate chloride creatinine glucose potassium sodium hematocrit wbc
    <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.

```
# d_labitems
dlabitems_tble <- read_csv("~/mimic/hosp/d_labitems.csv.gz") %>%
  mutate(itemid = as.character(itemid)) %>%
  select(itemid, label) %>%
  collect()
```

Rows: 1650 Columns: 4

```
-- Column specification -----
Delimiter: ","
chr (3): label, fluid, category
dbl (1): itemid
```

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.

```
# labevents
labevents_tble <- open_dataset("labevents_pq", format = "parquet") |>
#
to_duckdb() |>
#
select(subject_id, itemid, storetime, valuenum) |>
# itemid
filter(itemid %in% c("50912", "50971", "50983", "50902", "50882", "51221", "51301", "50931")) |>
# ICU intime copy=TRUE
left_join(
  select(icustays_tble, subject_id, stay_id, intime),
  by = c("subject_id"),
  copy = TRUE
) |>
# ICU
filter(storetime < intime) |>
# subject_id, stay_id, itemid
group_by(subject_id, stay_id, itemid) |>
# intime storetime
slice_max(storetime, n = 1) |>
# storetime intime
select(-storetime, -intime) |>
ungroup() |>
```

```

# itemid      label copy=TRUE
left_join(dlabitems_tble, by = "itemid", copy = TRUE) |>
select(-itemid) |>
#
pivot_wider(names_from = label, values_from = valuenum) |>
#
rename_with(~ str_to_lower(.)) |>
rename(wbc = 'white blood cells') |>
arrange(subject_id, stay_id) |>
#
show_query() |>
collect()

```

<SQL>

SELECT

```

    subject_id,
    stay_id,
    MAX(CASE WHEN ("label" = 'Bicarbonate') THEN valuenum END) AS bicarbonate,
    MAX(CASE WHEN ("label" = 'Hematocrit') THEN valuenum END) AS hematocrit,
    MAX(CASE WHEN ("label" = 'Chloride') THEN valuenum END) AS chloride,
    MAX(CASE WHEN ("label" = 'Glucose') THEN valuenum END) AS glucose,
    MAX(CASE WHEN ("label" = 'Potassium') THEN valuenum END) AS potassium,
    MAX(CASE WHEN ("label" = 'Creatinine') THEN valuenum END) AS creatinine,
    MAX(CASE WHEN ("label" = 'Sodium') THEN valuenum END) AS sodium,
    MAX(CASE WHEN ("label" = 'White Blood Cells') THEN valuenum END) AS wbc
FROM (
    SELECT subject_id, valuenum, stay_id, "label"
    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 co
            FROM (
                SELECT LHS.*, stay_id, intime
                FROM (
                    SELECT subject_id, itemid, storetime, valuenum
                    FROM arrow_001
                    WHERE (itemid IN ('50912', '50971', '50983', '50902', '50882', '51221', '51301', '5
                ) LHS
            ) LHS
        ) LHS
    LEFT JOIN dbplyr_4h9A09TGtk
        ON (LHS.subject_id = dbplyr_4h9A09TGtk.subject_id)

```

```

    ) q01
    WHERE (storetime < intime)
  ) q01
  WHERE (col01 <= 1)
) LHS
LEFT JOIN dbplyr_4hDGKYgLF1
  ON (LHS.itemid = dbplyr_4hDGKYgLF1.itemid)
) q01
GROUP BY subject_id, stay_id
ORDER BY subject_id, stay_id

```

```

#
print(labevents_tble)

```

```

# A tibble: 88,086 x 10
  subject_id stay_id bicarbonate hematocrit chloride glucose potassium
    <dbl>     <dbl>         <dbl>      <dbl>    <dbl>    <dbl>      <dbl>
1  10000032 39553978           25      41.1      95      102        6.7
2  10000690 37081114           26      36.1     100       85        4.8
3  10000980 39765666           21      27.3     109       89        3.9
4  10001217 34592300           30      37.4     104       87        4.1
5  10001217 37067082           22      38.1     108      112        4.2
6  10001725 31205490          NA       NA       98       NA        4.1
7  10001843 39698942           28      31.4      97      131        3.9
8  10001884 37510196           30      39.7      88      141        4.5
9  10002013 39060235           24      34.9     102      288        3.5
10 10002114 34672098           18      34.3      NA       95        6.5
# i 88,076 more rows
# i 3 more variables: creatinine <dbl>, sodium <dbl>, wbc <dbl>

```

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,w
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,60
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.

```

> charevents_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.

```
d_items_tble <- read_csv("~/mimic/icu/d_items.csv.gz") %>%
  mutate(itemid = as.character(itemid)) %>%
  select(itemid, label) %>%
  collect()
```

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.

```
# chartevents
chartevents_tble <- open_dataset("chartevents_pq", format = "parquet") |>
#
to_duckdb() |>
#
select(subject_id, stay_id, charttime, itemid, valuenum) |>
# itemid
filter(itemid %in% c("220045", "220179", "220180", "223761", "220210")) |>
# ICU intime copy=TRUE
left_join(
  select(icustays_tble, subject_id, stay_id, intime, outtime),
  by = c("subject_id", "stay_id"),
  copy = TRUE
) |>
# ICU
filter(charttime >= intime & charttime <= outtime) |>
# subject_id, stay_id, itemid
group_by(subject_id, stay_id, itemid) |>
# ICU
slice_min(charttime, n = 1) |>
# charttime, intime, outtime
select(-charttime, -intime, -outtime) |>
ungroup() |>
# itemid label copy=TRUE
```

```

left_join(d_items_tble, by = "itemid", copy = TRUE) |>
select(-itemid) |>
#
pivot_wider(names_from = label, values_from = valuenum) |>
#
rename_with(~ str_to_lower(.)) |>
rename(
  heart_rate = 'heart rate',
  systolic_bp = 'non invasive blood pressure systolic',
  diastolic_bp = 'non invasive blood pressure diastolic',
  temperature_f = 'temperature fahrenheit',
  respiratory_rate = 'respiratory rate'
) |>
arrange(subject_id, stay_id) |>
#
show_query() |>
collect()

```

<SQL>

SELECT

```

    subject_id,
    stay_id,
    MAX(CASE WHEN ("label" = 'Respiratory Rate') THEN valuenum END) AS respiratory_rate,
    MAX(CASE WHEN ("label" = 'Heart Rate') THEN valuenum END) AS heart_rate,
    MAX(CASE WHEN ("label" = 'Temperature Fahrenheit') THEN valuenum END) AS temperature_f,
    MAX(CASE WHEN ("label" = 'Non Invasive Blood Pressure systolic') THEN valuenum END) AS systolic_bp,
    MAX(CASE WHEN ("label" = 'Non Invasive Blood Pressure diastolic') THEN valuenum END) AS diastolic_bp
FROM (
    SELECT subject_id, stay_id, valuenum, "label"
    FROM (
        SELECT subject_id, stay_id, itemid, valuenum
        FROM (
            SELECT
                q01.*,
                RANK() OVER (PARTITION BY subject_id, stay_id, itemid ORDER BY charttime) AS col01
            FROM (
                SELECT LHS.*, intime, outtime
                FROM (
                    SELECT subject_id, stay_id, charttime, itemid, valuenum
                    FROM arrow_002
                    WHERE (itemid IN ('220045', '220179', '220180', '223761', '220210'))
                ) LHS
            )
        )
    )

```

```

LEFT JOIN dbplyr_s325IMy9y8
  ON (
    LHS.subject_id = dbplyr_s325IMy9y8.subject_id AND
    LHS.stay_id = dbplyr_s325IMy9y8.stay_id
  )
) q01
WHERE (charttime >= intime AND charttime <= outtime)
) q01
WHERE (col01 <= 1)
) LHS
LEFT JOIN dbplyr_jBzqRAXOpZ
  ON (LHS.itemid = dbplyr_jBzqRAXOpZ.itemid)
) q01
GROUP BY subject_id, stay_id
ORDER BY subject_id, stay_id

```

```

#
print(chartevents_tble)

```

```

# A tibble: 94,424 x 7
  subject_id stay_id respiratory_rate heart_rate temperature_f systolic_bp
    <dbl>    <dbl>          <dbl>      <dbl>          <dbl>          <dbl>
1  10000032 39553978             24         91           98.7            84
2  10000690 37081114             23         79           97.7           107
3  10000980 39765666             23         77           98             150
4  10001217 34592300             11         96           97.6           167
5  10001217 37067082             18         86           98.5           151
6  10001725 31205490             19         55           97.7            73
7  10001843 39698942             17        118           97.9           112
8  10001884 37510196             10         38           98.1           180
9  10002013 39060235             14         80           97.2           104
10 10002114 34672098             22        105           97.9           104
# i 94,414 more rows
# i 1 more variable: diastolic_bp <dbl>

```

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 disctime deathtime
  <dbl> <dbl> <dbl> <chr> <chr> <dtm> <dtm> <dbl> <dtm> <dtm> <dtm>
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 24592018 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 34622098 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 <dtm>, edouttime <dtm>, 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
```

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