

# Biostat 203B Homework 4

Due Mar 9 @ 11:59PM

Wenqiang Ge UID:106371961

## Table of contents

0.1	Q1. Compile the ICU cohort in HW3 from the Google BigQuery database . . . . .	2
0.2	Q2. Shiny app . . . . .	23

Display machine information:

```
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_3.6.4          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_1.9.0     xfun_0.50          digest_0.6.37
[13] rlang_1.1.5       evaluate_1.0.3
```

Display my machine memory.

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

```
Totalram: 7.686 GiB
```

```
Freeram: 6.176 GiB
```

Load database libraries and the tidyverse frontend:

```
library(bigrquery)
library(dbplyr)
library(DBI)
library(gt)
library(gtsummary)
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 dplyr::filter() masks stats::filter()
```

```
x dplyr::ident()  masks dbplyr::ident()
```

```
x dplyr::lag()    masks stats::lag()
```

```
x dplyr::sql()    masks dbplyr::sql()
```

```
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(forcats)
```

## 0.1 Q1. Compile the ICU cohort in HW3 from the Google BigQuery database

Below is an outline of steps. In this homework, we exclusively work with the BigQuery database and should not use any MIMIC data files stored on our local computer. Transform data as much as possible in BigQuery database and collect() the tibble **only at the end of Q1.7**.

### 0.1.1 Q1.1 Connect to BigQuery

Authenticate with BigQuery using the service account token. Please place the service account token (shared via BruinLearn) in the working directory (same folder as your qmd file). Do **not** ever add this token to your Git repository. If you do so, you will lose 50 points.

```
# path to the service account token
```

```
satoken <- "biostat-203b-2025-winter-4e58ec6e5579.json"
```

```
# BigQuery authentication using service account
bq_auth(path = satoken)
```

Connect to BigQuery database `mimiciv_3_1` in GCP (Google Cloud Platform), using the project billing account `biostat-203b-2025-winter`.

```
# connect to the BigQuery database `biostat-203b-2025-mimiciv_3_1`
con_bq <- dbConnect(
  bigrquery::bigquery(),
  project = "biostat-203b-2025-winter",
  dataset = "mimiciv_3_1",
  billing = "biostat-203b-2025-winter"
)
con_bq
```

```
<BigQueryConnection>
  Dataset: biostat-203b-2025-winter.mimiciv_3_1
  Billing: biostat-203b-2025-winter
```

List all tables in the `mimiciv_3_1` database.

```
dbListTables(con_bq)
```

[1] "admissions"	"caregiver"	"chartevents"
[4] "d_hcpcs"	"d_icd_diagnoses"	"d_icd_procedures"
[7] "d_items"	"d_labitems"	"datetimeevents"
[10] "diagnoses_icd"	"drgcodes"	"emar"
[13] "emar_detail"	"hpcsevents"	"icustays"
[16] "ingredientevents"	"inputevents"	"labevents"
[19] "microbiologyevents"	"omr"	"outputevents"
[22] "patients"	"pharmacy"	"poe"
[25] "poe_detail"	"prescriptions"	"procedureevents"
[28] "procedures_icd"	"provider"	"services"
[31] "transfers"		

### 0.1.2 Q1.2 icustays data

Connect to the `icustays` table.

```
# full ICU stays table
icustays_tble <- tbl(con_bq, "icustays") |>
  arrange(subject_id, hadm_id, stay_id) |>
  # show_query() |>
  print(width = Inf)
```

```
# Source:      SQL [?? x 8]
# Database:    BigQueryConnection
# Ordered by:  subject_id, hadm_id, stay_id
```

```

subject_id  hadm_id  stay_id  first_careunit
<int>      <int>      <int>  <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 more rows

```

### 0.1.3 Q1.3 admissions data

Connect to the admissions table.

```

# # TODO
admissions_tble <- tbl(con_bq, "admissions") |>
  print(width = Inf)

```

```
# Source:   table<`admissions`> [?? x 16]
```

# Database: BigQueryConnection

	subject_id	hadm_id	admittime	dischtime
	<int>	<int>	<dtm>	<dtm>
1	10106244	26713233	2147-05-09 10:34:00	2147-05-12 13:43:00
2	13700703	20448599	2172-09-25 01:01:00	2172-10-03 13:25:00
3	15443666	27961368	2168-12-30 23:30:00	2169-01-05 16:02:00
4	16299919	26977065	2193-05-15 08:37:00	2193-05-17 16:03:00
5	14149715	24191358	2181-10-25 19:37:00	2181-10-29 14:38:00
6	14446098	20543394	2182-04-04 20:11:00	2182-05-07 19:00:00
7	10584718	23485217	2165-02-12 15:41:00	2165-03-06 08:20:00
8	12224488	25909420	2158-10-29 15:59:00	2158-11-01 15:45:00
9	15845632	28189199	2124-10-05 02:44:00	2124-10-12 15:00:00
10	18131667	28337235	2195-11-18 02:58:00	2195-11-27 13:34:00

  

	deathtime	admission_type	admit_provider_id
	<dtm>	<chr>	<chr>
1	NA	DIRECT EMER.	<NA>
2	NA	OBSERVATION ADMIT	<NA>
3	NA	OBSERVATION ADMIT	<NA>
4	NA	OBSERVATION ADMIT	<NA>
5	NA	OBSERVATION ADMIT	P00230
6	NA	URGENT	P004G6
7	2165-03-06 08:20:00	EW EMER.	P004G6
8	NA	EW EMER.	P004G6
9	NA	EW EMER.	P004G6
10	NA	EW EMER.	P004G6

  

	admission_location	discharge_location	insurance
	<chr>	<chr>	<chr>
1	PHYSICIAN REFERRAL	HOME	Private
2	EMERGENCY ROOM	HOME	Private
3	EMERGENCY ROOM	HOME HEALTH CARE	Medicare
4	EMERGENCY ROOM	HOSPICE	Medicare
5	EMERGENCY ROOM	SKILLED NURSING FACILITY	Medicare
6	TRANSFER FROM HOSPITAL	SKILLED NURSING FACILITY	Medicare
7	TRANSFER FROM SKILLED NURSING FACILITY	DIED	Medicare
8	WALK-IN/SELF REFERRAL	HOME	Medicare
9	PHYSICIAN REFERRAL	HOME	Private
10	PHYSICIAN REFERRAL	HOME HEALTH CARE	Medicare

  

	language	marital_status	race	edregtime
	<chr>	<chr>	<chr>	<dtm>
1	English	SINGLE	WHITE	NA
2	English	MARRIED	WHITE	2172-09-24 17:38:00
3	English	SINGLE	BLACK/AFRICAN AMERICAN	2168-12-30 11:19:00
4	English	WIDOWED	BLACK/AFRICAN AMERICAN	2193-05-15 04:36:00
5	English	SINGLE	WHITE	2181-10-25 08:48:00
6	English	MARRIED	WHITE	NA
7	English	MARRIED	WHITE	NA

```

8 English SINGLE WHITE - OTHER EUROPEAN 2158-10-28 20:22:00
9 English MARRIED WHITE 2124-10-04 19:30:00
10 English SINGLE WHITE 2195-11-17 21:04:00
  edouttime      hospital_expire_flag
  <dtm>          <int>
1 NA 0
2 2172-09-25 03:07:00 0
3 2168-12-31 01:22:00 0
4 2193-05-15 14:27:00 0
5 2181-10-26 15:18:00 0
6 NA 0
7 NA 1
8 2158-10-29 18:01:00 0
9 2124-10-05 04:10:00 0
10 2195-11-18 04:51:00 0
# i more rows

```

#### 0.1.4 Q1.4 patients data

Connect to the `patients` table.

```

# # TODO
patients_tble <- tbl(con_bq, "patients") |>
  print(width = Inf)

```

```

# Source:   table<`patients`> [?? x 6]
# Database: BigQueryConnection
  subject_id gender anchor_age anchor_year anchor_year_group dod
    <int>   <chr>      <int>      <int>   <chr>          <date>
1    10078138 F          18        2110 2017 - 2019      NA
2    10180372 M          18        2110 2008 - 2010      NA
3    10686175 M          18        2110 2011 - 2013      NA
4    10851602 F          18        2110 2014 - 2016      NA
5    10902424 F          18        2110 2017 - 2019      NA
6    11092326 M          18        2110 2008 - 2010      NA
7    11289691 F          18        2110 2017 - 2019      NA
8    11595073 M          18        2110 2011 - 2013      NA
9    11739764 F          18        2110 2017 - 2019      NA
10   11776346 F          18        2110 2008 - 2010      NA
# i more rows

```

#### 0.1.5 Q1.5 labevents data

Connect to the `labevents` table and retrieve a subset that only contain subjects who appear in `icustays_tble` and the lab items listed in HW3. Only keep the last lab measurements (by `storetime`) before the ICU stay and pivot lab items to become variables/columns. Write all steps in *one* chain of pipes.

```

# Define the desired column order (ensuring subject_id and stay_id appear first)
column_order <- c("subject_id", "stay_id",
                  "bicarbonate", "chloride",
                  "creatinine", "glucose",
                  "potassium", "sodium",
                  "hematocrit", "wbc")

# Load labevents data from BigQuery and filter for relevant lab tests
labevents_tble <- tbl(con_bq, "labevents") |>

# Keep only the selected lab test item IDs
filter(itemid %in% c(50912, 50971, 50983, 50902,
                    50882, 51221, 51301, 50931)) |>

# Select relevant columns for processing
select(subject_id, storetime, itemid, valuenum) |>

# Join with ICU stay data to get stay_id and admission time
inner_join(
  select(icustays_tble, subject_id, stay_id, intime),
  by = "subject_id") |>

# Keep only lab results recorded before ICU admission time
filter(storetime < intime) |>

# Group by subject, stay, and itemid to retain the most recent measurement
group_by(subject_id, stay_id, itemid) |>
slice_max(order_by = storetime) |>
ungroup() |> # Remove grouping for subsequent operations

# Convert itemid numeric codes into meaningful lab test names
mutate(itemid = case_when(
  itemid == 51301 ~ "wbc",
  itemid == 51221 ~ "hematocrit",
  itemid == 50983 ~ "sodium",
  itemid == 50971 ~ "potassium",
  itemid == 50931 ~ "glucose",
  itemid == 50912 ~ "creatinine",
  itemid == 50902 ~ "chloride",
  itemid == 50882 ~ "bicarbonate"
)) |>

# Keep only necessary columns after renaming
select(subject_id, stay_id, itemid, valuenum) |>

```

```

# Convert from long format to wide format with mean aggregation for duplicates
pivot_wider(names_from = itemid,
             values_from = valuenum,
             values_fn = mean) |>

# Ensure the final column order follows the defined structure
select(all_of(column_order)) |>

# Arrange rows for better readability
arrange(subject_id, stay_id) |>

# Print the result with a wide display to prevent truncation
print(width = Inf)

```

Warning: ORDER BY is ignored in subqueries without LIMIT  
i Do you need to move arrange() later in the pipeline or use window\_order() instead?

Warning: Missing values are always removed in SQL aggregation functions.  
Use `na.rm = TRUE` to silence this warning  
This warning is displayed once every 8 hours.

Warning: ORDER BY is ignored in subqueries without LIMIT  
i Do you need to move arrange() later in the pipeline or use window\_order() instead?

```

# Source:      SQL [?? x 10]
# Database:    BigQueryConnection
# Ordered by: subject_id, stay_id
  subject_id  stay_id bicarbonate chloride creatinine glucose potassium sodium
    <int>      <int>      <dbl>    <dbl>      <dbl>    <dbl>      <dbl> <dbl>
1  10000032  39553978         25        95         0.7     102         6.7  126
2  10000690  37081114         26       100          1      85         4.8  137
3  10000980  39765666         21       109         2.3     89         3.9  144
4  10001217  34592300         30       104         0.5     87         4.1  142
5  10001217  37067082         22       108         0.6    112         4.2  142
6  10001725  31205490         NA        98         NA      NA         4.1  139
7  10001843  39698942         28        97         1.3    131         3.9  138
8  10001884  37510196         30        88         1.1    141         4.5  130
9  10002013  39060235         24       102         0.9    288         3.5  137
10 10002114  34672098         18        NA         3.1     95         6.5  125
  hematocrit  wbc
    <dbl> <dbl>
1    41.1   6.9
2    36.1   7.1
3    27.3   5.3
4    37.4   5.4
5    38.1  15.7
6     NA    NA

```



```

7      31.4  10.4
8      39.7  12.2
9      34.9   7.2
10     34.3  16.8
# i more rows

```

```
# Check the number of rows
```

```

labevents_tble |>
  tally() |>
  pull(n)

```

Warning: ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

```
[1] 88086
```

#### 0.1.6 Q1.6 chartevents data

Connect to `chartevents` table and retrieve a subset that only contain subjects who appear in `icustays_tble` and the chart events listed in HW3. Only keep the first chart events (by `storetime`) during ICU stay and pivot chart events to become variables/columns. Write all steps in *one* chain of pipes.

```
# # TODO
```

```
# Define the desired column order (ensuring subject_id and stay_id appear first)
```

```

column_order <- c("subject_id", "stay_id",
                  "heart_rate",
                  "non-invasive_blood_pressure_systolic",
                  "non-invasive_blood_pressure_diastolic",
                  "temperature_fahrenheit",
                  "respiratory_rate")

```

```
# Load chartevents data from BigQuery and filter for relevant vital measurements
```

```
chartevents_tble <- tbl(con_bq, "chartevents") |>
```

```
# Keep only the selected vital measurements item IDs
```

```

filter(itemid %in% c(220045, 220179, 220180,
                    223761, 220210)) |>

```

```
# Select relevant columns for processing
```

```
select(subject_id, storetime, itemid, valuenum) |>
```

```
# Join with ICU stay data to get stay_id and ICU intime and outtime
```

```

inner_join(
  select(icustays_tble, subject_id,
        stay_id, intime,outtime),

```

```

    by = "subject_id") |>

# Keep only lab results recorded in ICU
filter(storetime >= intime ) |>
filter(storetime <= outtime ) |>

# Group by subject, stay, and itemid to retain the earliest measurement
group_by(subject_id, stay_id, itemid) |>
slice_min(order_by = storetime) |>
ungroup() |> # Remove grouping for subsequent operations

# Convert itemid numeric codes into meaningful lab test names
mutate(itemid = case_when(
  itemid == 220045 ~ "heart_rate",
  itemid == 220179 ~ "non-invasive_blood_pressure_systolic",
  itemid == 220180 ~ "non-invasive_blood_pressure_diastolic",
  itemid == 223761 ~ "temperature_fahrenheit",
  itemid == 220210 ~ "respiratory_rate"
)) |>

# Keep only necessary columns after renaming
select(subject_id, stay_id, itemid, valuenum) |>

# Convert from long format to wide format with mean aggregation for duplicates
pivot_wider(names_from = itemid,
            values_from = valuenum,
            values_fn = ~ round(mean(.), 1)) |>

# Ensure the final column order follows the defined structure
select(all_of(column_order)) |>

# Arrange rows for better readability
arrange(subject_id, stay_id) |>

# Print the result with a wide display to prevent truncation
print(width = Inf)

```

```

Warning: ORDER BY is ignored in subqueries without LIMIT
i Do you need to move arrange() later in the pipeline or use window_order() instead?
ORDER BY is ignored in subqueries without LIMIT
i Do you need to move arrange() later in the pipeline or use window_order() instead?

# Source:      SQL [?? x 7]
# Database:    BigQueryConnection
# Ordered by:  subject_id, stay_id
  subject_id  stay_id heart_rate `non-invasive_blood_pressure_systolic`

```

	<int>	<int>	<dbl>	<dbl>
1	10000032	39553978	91	84
2	10000690	37081114	78	106
3	10000980	39765666	76	154
4	10001217	34592300	79.3	156
5	10001217	37067082	86	151
6	10001725	31205490	86	73
7	10001843	39698942	124.	110
8	10001884	37510196	49	174.
9	10002013	39060235	80	98.5
10	10002114	34672098	110.	112

	non-invasive_blood_pressure_diastolic`	temperature_fahrenheit
1	48	98.7
2	56.5	97.7
3	102	98
4	93.3	97.6
5	90	98.5
6	56	97.7
7	78	97.9
8	30.5	98.1
9	62	97.2
10	80	97.9

	respiratory_rate
1	24
2	24.3
3	23.5
4	14
5	18
6	19
7	16.5
8	13
9	14
10	21

# i more rows

*# Check the number of rows*

```
chartevents_tble |>
  tally() |>
  pull(n)
```

Warning: ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

[1] 94364

### 0.1.7 Q1.7 Put things together

This step is similar to Q7 of HW3. Using *one* chain of pipes `|>` to perform following data wrangling steps: (i) start with the `icustays_tble`, (ii) merge in admissions and patients tables, (iii) keep adults only (age at ICU intime  $\geq$  18), (iv) merge in the `labevents` and `chartevents` tables, (v) collect the tibble, (vi) sort `subject_id`, `hadm_id`, `stay_id` and `print(width = Inf)`.

```
# # TODO
mimic_icu_cohort <- icustays_tble |>

# Merge with admissions and patients tables
left_join(select(admissions_tble, -subject_id), by = "hadm_id") |>

# Merge with patients table
left_join(patients_tble, by = "subject_id") |>

# Keep only adults (age at ICU intime >= 18)
mutate(age = year(intime) - anchor_year + anchor_age) |>
filter(age >= 18) |>

# Merge with labevents and chartevents tables
left_join(labevents_tble, by = c("subject_id", "stay_id")) |>
left_join(chartevents_tble, by = c("subject_id", "stay_id")) |>

# Collect data into memory
collect() |>

# Sort by subject_id, hadm_id, stay_id
arrange(subject_id, hadm_id, stay_id) |>

# Print the full width to ensure readability
print(width = Inf)
```

Warning: ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window\_order() instead?

```

# A tibble: 94,458 x 41
  subject_id  hadm_id  stay_id first_careunit
    <int>    <int>    <int> <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 admittime      disctime
    <dtm>      <dbl> <dtm>      <dtm>
1 2180-07-23 23:50:47 0.410 2180-07-23 12:35:00 2180-07-25 17:55:00
2 2150-11-06 17:03:17 3.89 2150-11-02 18:02:00 2150-11-12 13:45:00
3 2189-06-27 20:38:27 0.498 2189-06-27 07:38:00 2189-07-03 03:00:00
4 2157-11-21 22:08:00 1.12 2157-11-18 22:56:00 2157-11-25 18:00:00
5 2157-12-20 14:27:41 0.948 2157-12-18 16:58:00 2157-12-24 14:55:00
6 2110-04-12 23:59:56 1.34 2110-04-11 15:08:00 2110-04-14 15:00:00
7 2134-12-06 14:38:26 0.825 2134-12-05 00:10:00 2134-12-06 12:54:00
8 2131-01-20 08:27:30 9.17 2131-01-07 20:39:00 2131-01-20 05:15:00
9 2160-05-19 17:33:33 1.31 2160-05-18 07:45:00 2160-05-23 13:30:00
10 2162-02-20 21:16:27 2.91 2162-02-17 22:32:00 2162-03-04 15:16:00
  deathtime      admission_type      admit_provider_id
    <dtm>      <chr>          <chr>
1 NA          EW EMER.      P060TX
2 NA          EW EMER.      P26QQ4
3 NA          EW EMER.      P060TX
4 NA          EW EMER.      P3610N
5 NA          DIRECT EMER.    P276OU
6 NA          EW EMER.      P32W56
7 2134-12-06 12:54:00 URGENT    P67ATB

```

	admission_location	discharge_location	insurance	language	marital_status
	<chr>	<chr>	<chr>	<chr>	<chr>
1	EMERGENCY ROOM	HOME	Medicaid	English	WIDOWED
2	EMERGENCY ROOM	REHAB	Medicare	English	WIDOWED
3	EMERGENCY ROOM	HOME HEALTH CARE	Medicare	English	MARRIED
4	EMERGENCY ROOM	HOME HEALTH CARE	Private	Other	MARRIED
5	PHYSICIAN REFERRAL	HOME HEALTH CARE	Private	Other	MARRIED
6	PACU	HOME	Private	English	MARRIED
7	TRANSFER FROM HOSPITAL	DIED	Medicare	English	SINGLE
8	EMERGENCY ROOM	DIED	Medicare	English	MARRIED
9	PHYSICIAN REFERRAL	HOME HEALTH CARE	Medicare	English	SINGLE
10	PHYSICIAN REFERRAL	HOME HEALTH CARE	Medicaid	English	<NA>

  

	race	edregtime	edouttime
	<chr>	<dtm>	<dtm>
1	WHITE	2180-07-23 05:54:00	2180-07-23 14:00:00
2	WHITE	2150-11-02 11:41:00	2150-11-02 19:37:00
3	BLACK/AFRICAN AMERICAN	2189-06-27 06:25:00	2189-06-27 08:42:00
4	WHITE	2157-11-18 17:38:00	2157-11-19 01:24:00
5	WHITE	NA	NA
6	WHITE	NA	NA
7	WHITE	NA	NA
8	BLACK/AFRICAN AMERICAN	2131-01-07 13:36:00	2131-01-07 22:13:00
9	OTHER	NA	NA
10	UNKNOWN	2162-02-17 19:35:00	2162-02-17 23:30:00

  

	hospital_expire_flag	gender	anchor_age	anchor_year	anchor_year_group
	<int>	<chr>	<int>	<int>	<chr>
1	0	F	52	2180	2014 - 2016
2	0	F	86	2150	2008 - 2010
3	0	F	73	2186	2008 - 2010
4	0	F	55	2157	2011 - 2013
5	0	F	55	2157	2011 - 2013
6	0	F	46	2110	2011 - 2013
7	1	M	73	2131	2017 - 2019
8	1	F	68	2122	2008 - 2010
9	0	F	53	2156	2008 - 2010
10	0	M	56	2162	2020 - 2022

  

	dod	age	bicarbonate	chloride	creatinine	glucose	potassium	sodium
	<date>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2180-09-09	52	25	95	0.7	102	6.7	126
2	2152-01-30	86	26	100	1	85	4.8	137
3	2193-08-26	76	21	109	2.3	89	3.9	144
4	NA	55	22	108	0.6	112	4.2	142
5	NA	55	30	104	0.5	87	4.1	142

6	NA	46	NA	98	NA	NA	4.1	139
7	2134-12-06	76	28	97	1.3	131	3.9	138
8	2131-01-20	77	30	88	1.1	141	4.5	130
9	NA	57	24	102	0.9	288	3.5	137
10	2162-12-11	56	18	NA	3.1	95	6.5	125

  

	hematocrit	wbc	heart_rate	`non-invasive_blood_pressure_systolic`
	<dbl>	<dbl>	<dbl>	<dbl>
1	41.1	6.9	91	84
2	36.1	7.1	78	106
3	27.3	5.3	76	154
4	38.1	15.7	86	151
5	37.4	5.4	79.3	156
6	NA	NA	86	73
7	31.4	10.4	124.	110
8	39.7	12.2	49	174.
9	34.9	7.2	80	98.5
10	34.3	16.8	110.	112

  

	`non-invasive_blood_pressure_diastolic`	temperature_fahrenheit
	<dbl>	<dbl>
1	48	98.7
2	56.5	97.7
3	102	98
4	90	98.5
5	93.3	97.6
6	56	97.7
7	78	97.9
8	30.5	98.1
9	62	97.2
10	80	97.9

  

	respiratory_rate
	<dbl>
1	24
2	24.3
3	23.5
4	18
5	14
6	19
7	16.5
8	13
9	14
10	21

# i 94,448 more rows

### 0.1.8 Q1.8 Preprocessing

Perform the following preprocessing steps. (i) Lump infrequent levels into “Other” level for `first_careunit`, `last_careunit`, `admission_type`, `admission_location`, and `discharge_location`. (ii) Collapse the levels of `race` into `ASIAN`, `BLACK`, `HISPANIC`, `WHITE`, and `Other`. (iii) Create a new variable `los_long` that is `TRUE` when `los` is greater than or equal to 2 days. (iv) Summarize the data using `tbl_summary()`, stratified by `los_long`. Hint: `fct_lump_n` and `fct_collapse` from the `forcats` package are useful.

Hint: Below is a numerical summary of my tibble after preprocessing:

```
# Process mimic_icu_cohort data
mimic_icu_cohort <- mimic_icu_cohort |>
  mutate(
    first_careunit = fct_lump(first_careunit, n = 4),
    last_careunit = fct_lump(last_careunit, n = 4),
    admission_type = fct_lump(admission_type, n = 4),
    admission_location = fct_lump(admission_location, n = 3),
    discharge_location = fct_lump(discharge_location, n = 4)
  ) |>

# Sort by subject_id, hadm_id, stay_id
arrange(subject_id, hadm_id, stay_id) |>

# Print the full width to ensure readability
print(width = Inf)
```

# A tibble: 94,458 x 41

	subject_id	hadm_id	stay_id	first_careunit
	<int>	<int>	<int>	<fct>
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	Other
	last_careunit		intime	
	<fct>		<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	Other				2162-02-17 23:30:00
	outtime	los	admittime		dischtime
	<dtm>	<dbl>	<dtm>		<dtm>
1	2180-07-23 23:50:47	0.410	2180-07-23 12:35:00		2180-07-25 17:55:00
2	2150-11-06 17:03:17	3.89	2150-11-02 18:02:00		2150-11-12 13:45:00
3	2189-06-27 20:38:27	0.498	2189-06-27 07:38:00		2189-07-03 03:00:00
4	2157-11-21 22:08:00	1.12	2157-11-18 22:56:00		2157-11-25 18:00:00
5	2157-12-20 14:27:41	0.948	2157-12-18 16:58:00		2157-12-24 14:55:00
6	2110-04-12 23:59:56	1.34	2110-04-11 15:08:00		2110-04-14 15:00:00
7	2134-12-06 14:38:26	0.825	2134-12-05 00:10:00		2134-12-06 12:54:00
8	2131-01-20 08:27:30	9.17	2131-01-07 20:39:00		2131-01-20 05:15:00
9	2160-05-19 17:33:33	1.31	2160-05-18 07:45:00		2160-05-23 13:30:00
10	2162-02-20 21:16:27	2.91	2162-02-17 22:32:00		2162-03-04 15:16:00
	deathtime		admission_type		admit_provider_id
	<dtm>		<fct>		<chr>
1	NA		EW EMER.		P060TX
2	NA		EW EMER.		P26QQ4
3	NA		EW EMER.		P060TX
4	NA		EW EMER.		P3610N
5	NA		Other		P276OU
6	NA		EW EMER.		P32W56
7	2134-12-06 12:54:00		URGENT		P67ATB
8	2131-01-20 05:15:00		OBSERVATION ADMIT		P49AFC
9	NA		SURGICAL SAME DAY ADMISSION		P8286C
10	NA		OBSERVATION ADMIT		P46834
	admission_location		discharge_location	insurance	language marital_status
	<fct>		<fct>	<chr>	<chr> <chr>
1	EMERGENCY ROOM		HOME	Medicaid	English WIDOWED
2	EMERGENCY ROOM		Other	Medicare	English WIDOWED
3	EMERGENCY ROOM		HOME HEALTH CARE	Medicare	English MARRIED
4	EMERGENCY ROOM		HOME HEALTH CARE	Private	Other MARRIED
5	PHYSICIAN REFERRAL		HOME HEALTH CARE	Private	Other MARRIED
6	Other		HOME	Private	English MARRIED
7	TRANSFER FROM HOSPITAL		DIED	Medicare	English SINGLE
8	EMERGENCY ROOM		DIED	Medicare	English MARRIED
9	PHYSICIAN REFERRAL		HOME HEALTH CARE	Medicare	English SINGLE
10	PHYSICIAN REFERRAL		HOME HEALTH CARE	Medicaid	English <NA>
	race		edregtime		edouttime
	<chr>		<dtm>		<dtm>
1	WHITE		2180-07-23 05:54:00		2180-07-23 14:00:00

2	WHITE	2150-11-02	11:41:00	2150-11-02	19:37:00		
3	BLACK/AFRICAN AMERICAN	2189-06-27	06:25:00	2189-06-27	08:42:00		
4	WHITE	2157-11-18	17:38:00	2157-11-19	01:24:00		
5	WHITE	NA		NA			
6	WHITE	NA		NA			
7	WHITE	NA		NA			
8	BLACK/AFRICAN AMERICAN	2131-01-07	13:36:00	2131-01-07	22:13:00		
9	OTHER	NA		NA			
10	UNKNOWN	2162-02-17	19:35:00	2162-02-17	23:30:00		
	hospital_expire_flag	gender	anchor_age	anchor_year	anchor_year_group		
	<int>	<chr>	<int>	<int>	<chr>		
1	0	F	52	2180	2014 - 2016		
2	0	F	86	2150	2008 - 2010		
3	0	F	73	2186	2008 - 2010		
4	0	F	55	2157	2011 - 2013		
5	0	F	55	2157	2011 - 2013		
6	0	F	46	2110	2011 - 2013		
7	1	M	73	2131	2017 - 2019		
8	1	F	68	2122	2008 - 2010		
9	0	F	53	2156	2008 - 2010		
10	0	M	56	2162	2020 - 2022		
	dod	age	bicarbonate	chloride	creatinine	glucose	potassium sodium
	<date>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2180-09-09	52	25	95	0.7	102	6.7 126
2	2152-01-30	86	26	100	1	85	4.8 137
3	2193-08-26	76	21	109	2.3	89	3.9 144
4	NA	55	22	108	0.6	112	4.2 142
5	NA	55	30	104	0.5	87	4.1 142
6	NA	46	NA	98	NA	NA	4.1 139
7	2134-12-06	76	28	97	1.3	131	3.9 138
8	2131-01-20	77	30	88	1.1	141	4.5 130
9	NA	57	24	102	0.9	288	3.5 137
10	2162-12-11	56	18	NA	3.1	95	6.5 125
	hematocrit	wbc	heart_rate	`non-invasive_blood_pressure_systolic`			
	<dbl>	<dbl>	<dbl>	<dbl>			
1	41.1	6.9	91	84			
2	36.1	7.1	78	106			
3	27.3	5.3	76	154			
4	38.1	15.7	86	151			
5	37.4	5.4	79.3	156			
6	NA	NA	86	73			
7	31.4	10.4	124.	110			
8	39.7	12.2	49	174.			
9	34.9	7.2	80	98.5			
10	34.3	16.8	110.	112			
	`non-invasive_blood_pressure_diastolic` temperature_fahrenheit						

	<dbl>	<dbl>
1	48	98.7
2	56.5	97.7
3	102	98
4	90	98.5
5	93.3	97.6
6	56	97.7
7	78	97.9
8	30.5	98.1
9	62	97.2
10	80	97.9

```
respiratory_rate
```

	<dbl>
1	24
2	24.3
3	23.5
4	18
5	14
6	19
7	16.5
8	13
9	14
10	21

```
# i 94,448 more rows
```

```
# Collapse race categories into ASIAN, BLACK, HISPANIC, WHITE, and Other
```

```
mimic_icu_cohort <- mimic_icu_cohort |>
```

```
  mutate(
    race = fct_collapse(race, # Collapse race categories
      ASIAN = c("ASIAN", "ASIAN - CHINESE",
                "ASIAN - JAPANESE", "ASIAN - INDIAN"),
      BLACK = c("BLACK", "BLACK/AFRICAN AMERICAN"),
      HISPANIC = c("HISPANIC", "HISPANIC/LATINO"),
      WHITE = c("WHITE", "WHITE - RUSSIAN"),
      Other = c("OTHER", "UNKNOWN", "MIXED",
                "NATIVE AMERICAN", "MIDDLE EASTERN",
                "PACIFIC ISLANDER")
    )
  ) |>
```

```
# Print the full width to ensure readability
print(width = Inf)
```

```
Warning: There was 1 warning in `mutate()`.
```

```
i In argument: `race = fct_collapse(...)`.
```

```
Caused by warning:
```

```
! Unknown levels in `f`: ASIAN - JAPANESE, ASIAN - INDIAN, BLACK, HISPANIC, HISPANIC/LATINO.
```

	subject_id	hadm_id	stay_id	first_careunit
	<int>	<int>	<int>	<fct>
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	Other

outtime			los	admittime		disctime	
<dtm>			<dbl>	<dtm>		<dtm>	
1	2180-07-23	23:50:47	0.410	2180-07-23	12:35:00	2180-07-25	17:55:00
2	2150-11-06	17:03:17	3.89	2150-11-02	18:02:00	2150-11-12	13:45:00
3	2189-06-27	20:38:27	0.498	2189-06-27	07:38:00	2189-07-03	03:00:00
4	2157-11-21	22:08:00	1.12	2157-11-18	22:56:00	2157-11-25	18:00:00
5	2157-12-20	14:27:41	0.948	2157-12-18	16:58:00	2157-12-24	14:55:00
6	2110-04-12	23:59:56	1.34	2110-04-11	15:08:00	2110-04-14	15:00:00
7	2134-12-06	14:38:26	0.825	2134-12-05	00:10:00	2134-12-06	12:54:00
8	2131-01-20	08:27:30	9.17	2131-01-07	20:39:00	2131-01-20	05:15:00
9	2160-05-19	17:33:33	1.31	2160-05-18	07:45:00	2160-05-23	13:30:00
10	2162-02-20	21:16:27	2.91	2162-02-17	22:32:00	2162-03-04	15:16:00
deathtime			admission_type		admit_provider_id		
<dtm>			<fct>		<chr>		
1	NA		EW EMER.		P060TX		
2	NA		EW EMER.		P26QQ4		
3	NA		EW EMER.		P060TX		
4	NA		EW EMER.		P3610N		
5	NA		Other		P276QU		

6	NA	EW EMER.	P32W56
7	2134-12-06 12:54:00	URGENT	P67ATB
8	2131-01-20 05:15:00	OBSERVATION ADMIT	P49AFC
9	NA	SURGICAL SAME DAY ADMISSION	P8286C
10	NA	OBSERVATION ADMIT	P46834

  

	admission_location	discharge_location	insurance	language	marital_status
	<fct>	<fct>	<chr>	<chr>	<chr>
1	EMERGENCY ROOM	HOME	Medicaid	English	WIDOWED
2	EMERGENCY ROOM	Other	Medicare	English	WIDOWED
3	EMERGENCY ROOM	HOME HEALTH CARE	Medicare	English	MARRIED
4	EMERGENCY ROOM	HOME HEALTH CARE	Private	Other	MARRIED
5	PHYSICIAN REFERRAL	HOME HEALTH CARE	Private	Other	MARRIED
6	Other	HOME	Private	English	MARRIED
7	TRANSFER FROM HOSPITAL	DIED	Medicare	English	SINGLE
8	EMERGENCY ROOM	DIED	Medicare	English	MARRIED
9	PHYSICIAN REFERRAL	HOME HEALTH CARE	Medicare	English	SINGLE
10	PHYSICIAN REFERRAL	HOME HEALTH CARE	Medicaid	English	<NA>

  

	race	edregtime	edouttime	hospital_expire_flag	gender
	<fct>	<dtm>	<dtm>	<int>	<chr>
1	WHITE	2180-07-23 05:54:00	2180-07-23 14:00:00		0 F
2	WHITE	2150-11-02 11:41:00	2150-11-02 19:37:00		0 F
3	BLACK	2189-06-27 06:25:00	2189-06-27 08:42:00		0 F
4	WHITE	2157-11-18 17:38:00	2157-11-19 01:24:00		0 F
5	WHITE	NA			0 F
6	WHITE	NA			0 F
7	WHITE	NA			1 M
8	BLACK	2131-01-07 13:36:00	2131-01-07 22:13:00		1 F
9	Other	NA			0 F
10	Other	2162-02-17 19:35:00	2162-02-17 23:30:00		0 M

  

	anchor_age	anchor_year	anchor_year_group	dod	age	bicarbonate
	<int>	<int>	<chr>	<date>	<int>	<dbl>
1	52	2180	2014 - 2016	2180-09-09	52	25
2	86	2150	2008 - 2010	2152-01-30	86	26
3	73	2186	2008 - 2010	2193-08-26	76	21
4	55	2157	2011 - 2013	NA	55	22
5	55	2157	2011 - 2013	NA	55	30
6	46	2110	2011 - 2013	NA	46	NA
7	73	2131	2017 - 2019	2134-12-06	76	28
8	68	2122	2008 - 2010	2131-01-20	77	30
9	53	2156	2008 - 2010	NA	57	24
10	56	2162	2020 - 2022	2162-12-11	56	18

  

	chloride	creatinine	glucose	potassium	sodium	hematocrit	wbc	heart_rate
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	95	0.7	102	6.7	126	41.1	6.9	91
2	100	1	85	4.8	137	36.1	7.1	78
3	109	2.3	89	3.9	144	27.3	5.3	76

4	108	0.6	112	4.2	142	38.1	15.7	86
5	104	0.5	87	4.1	142	37.4	5.4	79.3
6	98	NA	NA	4.1	139	NA	NA	86
7	97	1.3	131	3.9	138	31.4	10.4	124.
8	88	1.1	141	4.5	130	39.7	12.2	49
9	102	0.9	288	3.5	137	34.9	7.2	80
10	NA	3.1	95	6.5	125	34.3	16.8	110.

```

`non-invasive_blood_pressure_systolic`
<dbl>
1      84
2     106
3     154
4     151
5     156
6      73
7     110
8     174.
9     98.5
10    112
`non-invasive_blood_pressure_diastolic` temperature_fahrenheit
<dbl> <dbl>
1      48      98.7
2     56.5     97.7
3     102      98
4      90     98.5
5     93.3     97.6
6      56     97.7
7      78     97.9
8     30.5     98.1
9      62     97.2
10     80     97.9
respiratory_rate
<dbl>
1      24
2     24.3
3     23.5
4      18
5      14
6      19
7     16.5
8      13
9      14
10     21
# i 94,448 more rows

```

```
# Step (iii): Create los_long variable
mimic_icu_cohort <- mimic_icu_cohort |>
  mutate(
    los_long = los >= 2 # TRUE if length of stay is 2 days or more, FALSE otherwise
  )
```

```
mimic_icu_cohort |>
  select(-c(subject_id, hadm_id, stay_id,
            intime, outtime, admittance, dischtime,
            deathtime, admit_provider_id, edregtime,
            edouttime, anchor_age, anchor_year,
            anchor_year_group)) |>
  tbl_summary(by = los_long)
```

14 missing rows in the "los\_long" column have been removed.

The following errors were returned during `tbl\_summary()`:

```
x For variable `dod` (`los_long = FALSE`) and "p75" statistic: * not defined
  for "Date" objects
```

### 0.1.9 Q1.9 Save the final tibble

Save the final tibble to an R data file `mimic_icu_cohort.rds` in the `mimiciv_shiny` folder.

```
# make a directory mimiciv_shiny
if (!dir.exists("mimiciv_shiny")) {
  dir.create("mimiciv_shiny")
}
# save the final tibble
mimic_icu_cohort |>
  write_rds("mimiciv_shiny/mimic_icu_cohort.rds", compress = "gz")
```

Close database connection and clear workspace.

```
if (exists("con_bq")) {
  dbDisconnect(con_bq)
}
rm(list = ls())
```

Although it is not a good practice to add big data files to Git, for grading purpose, please add `mimic_icu_cohort.rds` to your Git repository.

## 0.2 Q2. Shiny app

Develop a Shiny app for exploring the ICU cohort data created in Q1. The app should reside in the `mimiciv_shiny` folder. The app should contain at least two tabs. One tab provides easy access to the graphical and numerical summaries of variables (demographics, lab measurements, vitals) in the ICU cohort, using

Characteristic		TRUE N = 46,337 <sup>1</sup>
first_careunit		
	Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)
	Medical Intensive Care Unit (MICU)	9,837 (21%)
	Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)
	Surgical Intensive Care Unit (SICU)	6,434 (14%)
	Other	16,046 (35%)
last_careunit		
	Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)
	Medical Intensive Care Unit (MICU)	9,837 (21%)
	Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)
	Surgical Intensive Care Unit (SICU)	6,434 (14%)
	Other	16,046 (35%)
los		3.9 (2.7, 6.8)
admission_type		
	EW EMER.	23,012 (50%)
	OBSERVATION ADMIT	7,393 (16%)
	SURGICAL SAME DAY ADMISSION	4,001 (8.6%)
	URGENT	8,691 (19%)
	Other	3,240 (7.0%)
admission_location		
	EMERGENCY ROOM	17,058 (37%)
	PHYSICIAN REFERRAL	11,013 (24%)
	TRANSFER FROM HOSPITAL	13,904 (30%)
	Other	4,362 (9.4%)
discharge_location		
	DIED	6,884 (15%)
	HOME	6,879 (15%)
	HOME HEALTH CARE	10,620 (23%)
	SKILLED NURSING FACILITY	8,785 (19%)
	Other	13,092 (28%)
	Unknown	77
insurance		
	Medicaid	6,768 (15%)
	Medicare	26,330 (58%)
	No charge	5 (<0.1%)
	Other	1,091 (2.4%)
	Private	11,515 (25%)
	Unknown	628
language	24	
	American Sign Language	29 (<0.1%)
	Amharic	14 (<0.1%)
	Arabic	87 (0.2%)
	Armenian	12 (<0.1%)
	Bengali	22 (<0.1%)
	Chinese	550 (1.2%)



the `mimic_icu_cohort.rds` you curated in Q1. The other tab allows user to choose a specific patient in the cohort and display the patient's ADT and ICU stay information as we did in Q1 of HW3, by dynamically retrieving the patient's ADT and ICU stay information from BigQuery database. Again, do **not** ever add the BigQuery token to your Git repository. If you do so, you will lose 50 points.