Biostat 203B Homework 4

Due Mar 8 @ 11:59PM

AUTHOR
Zongzhe Lin UID:206328707

Display machine information:

```
sessionInfo()
```

```
R version 4.3.2 (2023-10-31 ucrt)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 11 x64 (build 22631)

Matrix products: default


locale:
[1] LC_COLLATE=Chinese (Simplified)_China.utf8
[2] LC_CTYPE=Chinese (Simplified)_China.utf8
[3] LC_MONETARY=Chinese (Simplified)_China.utf8
[4] LC_NUMERIC=C
[5] LC_TIME=Chinese (Simplified)_China.utf8

time zone: America/Tijuana
tzcode source: internal

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

loaded via a namespace (and not attached):
 [1] htmlwidgets_1.6.4 compiler_4.3.2    fastmap_1.1.1     cli_3.6.1
 [5] tools_4.3.2       htmltools_0.5.7   rstudioapi_0.15.0 yaml_2.3.8
 [9] rmarkdown_2.26    knitr_1.45        jsonlite_1.8.7    xfun_0.40
[13] digest_0.6.33     rlang_1.1.1       evaluate_0.23
```

Display my machine memory.

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

```
Totalram:  63.845 GiB
Freeram:   52.780 GiB
```

Load database libraries and the tidyverse frontend:

```
library(bigrquery)
```

```
Warning: 程辑包'bigrquery'是用R版本4.3.3 来建造的
```

```r
library(dbplyr)
library(DBI)
```

```
Warning: 程辑包'DBI'是用R版本4.3.3 来建造的
```

```r
library(gt)
```

```
Warning: 程辑包'gt'是用R版本4.3.3 来建造的
```

```r
library(gtsummary)
```

```
Warning: 程辑包'gtsummary'是用R版本4.3.3 来建造的
```

```r
library(tidyverse)
```

```
Warning: 程辑包'ggplot2'是用R版本4.3.3 来建造的
```

```
Warning: 程辑包'tidyr'是用R版本4.3.3 来建造的
```

```
Warning: 程辑包'readr'是用R版本4.3.3 来建造的
```

```
Warning: 程辑包'stringr'是用R版本4.3.3 来建造的
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.3     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.0     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.1
✔ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::ident()  masks dbplyr::ident()
✖ dplyr::lag()    masks stats::lag()
✖ dplyr::sql()    masks dbplyr::sql()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
errors
```

```r
library(dplyr)
library(tidyr)
library(lubridate)
library(forcats)
library(shiny)
library(DT)
```

```
Warning: 程辑包'DT'是用R版本4.3.3 来建造的
```

```

载入程辑包：'DT'
```

```
The following objects are masked from 'package:shiny':
    dataTableOutput, renderDataTable
```

```r
library(ggplot2)
```

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.

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** add this token to your git repository.

```r
# path to the service account token
satoken <- "biostat-203b-2024-winter-313290ce47a6.json"
# BigQuery authentication using service account
bq_auth(path = satoken)
```

Connect to BigQuery database `mimic4_v2_2` in GCP (Google Cloud Platform), using the project billing account `biostat-203b-2024-winter`.

```r
# connect to the BigQuery database `biostat-203b-2024-winter.mimic4_v2_2`
con_bq <- dbConnect(
    bigrquery::bigquery(),
    project = "biostat-203b-2024-winter",
    dataset = "mimic4_v2_2",
    billing = "biostat-203b-2024-winter"
)
con_bq
```

```
<BigQueryConnection>
  Dataset: biostat-203b-2024-winter.mimic4_v2_2
  Billing: biostat-203b-2024-winter
```

List all tables in the `mimic4_v2_2` database.

```r
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"        "hcpcsevents"        "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"
```

anchor+### Q1.2 `icustays` data

Connect to the `icustays` table.

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

```
# Source:   table<icustays> [?? x 8]
# Database: BigQueryConnection
   subject_id  hadm_id  stay_id first_careunit last_careunit
        <int>    <int>    <int> <chr>          <chr>
 1   10004113 29879900 35200789 Neuro Stepdown Neuro Stepdown
 2   10018328 23786647 31269608 Neuro Stepdown Neuro Stepdown
 3   10019607 24546857 34178938 Neuro Stepdown Neuro Stepdown
 4   10023994 21824032 30439823 Neuro Stepdown Neuro Stepdown
 5   10028930 26238833 33941522 Neuro Stepdown Neuro Stepdown
 6   10067921 27475639 31097260 Neuro Stepdown Neuro Stepdown
 7   10088776 26138210 39177309 Neuro Stepdown Neuro Stepdown
 8   10088776 27242516 35695123 Neuro Stepdown Neuro Stepdown
 9   10103795 20269407 30619907 Neuro Stepdown Neuro Stepdown
10   10103795 22741814 31411464 Neuro Stepdown Neuro Stepdown
   intime              outtime                los
   <dttm>              <dttm>               <dbl>
 1 2173-03-20 20:16:36 2173-03-21 21:43:59  1.06
 2 2154-04-24 23:03:44 2154-05-02 15:55:21  7.70
 3 2167-03-19 17:37:39 2167-03-21 13:12:28  1.82
 4 2146-07-19 15:32:22 2146-07-20 14:05:53  0.940
 5 2172-08-28 13:11:37 2172-08-29 14:19:35  1.05
 6 2128-07-15 00:55:00 2128-07-16 11:41:23  1.45
 7 2156-01-01 17:29:32 2156-01-02 13:35:41  0.838
 8 2155-05-01 15:30:41 2155-05-02 14:59:41  0.978
 9 2176-09-12 00:26:56 2176-09-17 14:52:28  5.60
10 2176-07-01 19:45:36 2176-07-11 22:07:48 10.1
# ℹ more rows
```

Q1.3 `admissions` data

Connect to the `admissions` table.

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

```
# Source:   table<admissions> [?? x 16]
# Database: BigQueryConnection
   subject_id  hadm_id admittime           dischtime           deathtime
        <int>    <int> <dttm>              <dttm>              <dttm>
 1   10106244 26713233 2147-05-09 10:34:00 2147-05-12 13:43:00 NA
 2   13700703 20448599 2172-09-25 01:01:00 2172-10-03 13:25:00 NA
 3   15443666 27961368 2168-12-30 23:30:00 2169-01-05 16:02:00 NA
 4   16299919 26977065 2193-05-15 08:37:00 2193-05-17 16:03:00 NA
 5   19596808 26675460 2128-06-04 16:13:00 2128-06-05 11:36:00 NA
 6   15868868 27142491 2191-08-07 13:06:00 2191-08-11 19:55:00 NA
 7   11532813 24829571 2160-12-18 06:59:00 2160-12-25 14:40:00 NA
 8   14527596 28165014 2165-11-28 19:02:00 2165-12-07 17:23:00 NA
 9   15902500 27166666 2124-10-26 19:37:00 2124-10-28 16:30:00 NA
10   16625434 25780525 2135-09-05 17:38:00 2135-10-14 15:08:00 NA
   admission_type    admit_provider_id admission_location
   <chr>             <chr>             <chr>
 1 DIRECT EMER.      <NA>              PHYSICIAN REFERRAL
 2 OBSERVATION ADMIT <NA>              EMERGENCY ROOM
 3 OBSERVATION ADMIT <NA>              EMERGENCY ROOM
 4 OBSERVATION ADMIT <NA>              EMERGENCY ROOM
 5 EW EMER.          P0038O            EMERGENCY ROOM
 6 URGENT            P004U7            WALK-IN/SELF REFERRAL
 7 OBSERVATION ADMIT P004U7            EMERGENCY ROOM
 8 URGENT            P005HP            TRANSFER FROM HOSPITAL
 9 URGENT            P005HP            TRANSFER FROM HOSPITAL
10 URGENT            P005HP            TRANSFER FROM HOSPITAL
   discharge_location       insurance language marital_status
   <chr>                    <chr>     <chr>    <chr>
 1 HOME                     Other     ENGLISH  SINGLE
 2 HOME                     Other     ENGLISH  MARRIED
 3 HOME HEALTH CARE         Other     ENGLISH  SINGLE
 4 HOSPICE                  Medicare  ENGLISH  WIDOWED
 5 HOME                     Medicare  ENGLISH  MARRIED
 6 HOME HEALTH CARE         Medicare  ENGLISH  SINGLE
 7 ACUTE HOSPITAL           Medicare  ENGLISH  SINGLE
 8 HOME                     Medicaid  ?        MARRIED
 9 HOME                     Medicare  ENGLISH  MARRIED
10 SKILLED NURSING FACILITY Medicare  ?        SINGLE
   race                   edregtime           edouttime
   <chr>                  <dttm>              <dttm>
 1 WHITE                  NA                  NA
 2 WHITE                  2172-09-24 17:38:00 2172-09-25 03:07:00
 3 BLACK/AFRICAN AMERICAN 2168-12-30 11:19:00 2168-12-31 01:22:00
 4 BLACK/AFRICAN AMERICAN 2193-05-15 04:36:00 2193-05-15 14:27:00
 5 WHITE                  2128-06-04 13:25:00 2128-06-04 16:59:00
 6 WHITE                  2191-08-06 18:50:00 2191-08-07 16:30:00
 7 WHITE                  2160-12-17 16:20:00 2160-12-18 08:43:00
 8 BLACK/CARIBBEAN ISLAND NA                  NA
 9 BLACK/CAPE VERDEAN     NA                  NA
10 WHITE                  NA                  NA
   hospital_expire_flag
                  <int>
 1                    0
 2                    0
 3                    0
 4                    0
 5                    0
 6                    0
 7                    0
 8                     0
 9                     0
10                     0
# ℹ more rows
```

Q1.4 `patients` data

Connect to the `patients` table.

```
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
```

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 before the ICU stay and pivot lab items to become variables/columns. Write all steps in *one* chain of pipes.

```
# Define lab item IDs
lab_item_ids <- c(50912, 50971, 50983, 50902, 50882, 51221, 51301, 50931)

# Process the lab events data
labevents_subset <- tbl(con_bq, "labevents") %>%
  # Select the necessary columns
  select(subject_id, itemid, storetime, valuenum) %>%
  # Keep only the rows with item IDs that match the lab_item_ids list
  filter(itemid %in% lab_item_ids) %>%
  # Include only subjects who are also in the ICU stays table
  semi_join(tbl(con_bq, "icustays") %>% select(subject_id), by = "subject_id") %>%
  # Join to get stay IDs and ICU admission times for matching subjects
  left_join(tbl(con_bq, "icustays") %>% select(subject_id, stay_id, intime), by = "subject_id"
  # Filter out lab records that occur after the ICU admission time
  filter(storetime < intime) %>%
  # Order the results by subject, stay, and storetime in descending order
  arrange(subject_id, stay_id, desc(storetime)) %>%
  # Group by subject, stay, and item to prepare for taking the last record
  group_by(subject_id, stay_id, itemid) %>%
```

```r
  # Select the last record before ICU admission for each group
  slice_max(order_by = storetime, n = 1, with_ties = FALSE) %>%
  # Ungroup for the next operations
  ungroup() %>%
  # Remove the storetime column as it's no longer needed
  select(-storetime) %>%
  # Pivot data from long to wide format, making lab item IDs into columns
  pivot_wider(names_from = itemid, values_from = valuenum, names_prefix = "lab_") %>%
  # Rename the new columns with more descriptive names
  rename(
    creatinine = lab_50912,
    potassium = lab_50971,
    sodium = lab_50983,
    chloride = lab_50902,
    bicarbonate = lab_50882,
    hematocrit = lab_51221,
    white_blood_cell_count = lab_51301,
    glucose = lab_50931
  ) %>%
  # Remove the ICU admission time as it's not needed in the final output
  select(-intime) %>%
  # Order the final table by subject and stay ID
  arrange(subject_id, stay_id) %>%
  # Print the table, setting the width to Inf to avoid truncated output
  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 10]
# Database:   BigQueryConnection
# Ordered by: subject_id, stay_id
   subject_id  stay_id hematocrit chloride creatinine sodium glucose bicarbonate
        <int>    <int>      <dbl>    <dbl>      <dbl>  <dbl>   <dbl>       <dbl>
 1   10000032 39553978       41.1       95        0.7    126     102          25
 2   10000980 39765666       27.3      109        2.3    144      89          21
 3   10001217 34592300       37.4      104        0.5    142      87          30
 4   10001217 37067082       38.1      108        0.6    142     112          22
 5   10001725 31205490       NA         98       NA      139      NA          NA
 6   10001884 37510196       39.7       88        1.1    130     141          30
 7   10002013 39060235       34.9      102        0.9    137     288          24
 8   10002155 31090461       25.5       98        2.8    135     117          23
 9   10002155 32358465       22.4       85        1.4    120     133          26
10   10002155 33685454       39.7      105        1.1    139     138          24
   white_blood_cell_count potassium
                    <dbl>     <dbl>
 1                    6.9       6.7
 2                    5.3       3.9
 3                    5.4       4.1
 4                   15.7       4.2
 5                   NA          4.1
 6                 12.2          4.5
 7                  7.2          3.5
 8                 17.9          4.9
 9                  9.8          5.7
10                  7.9          4.6
# ℹ more rows
```

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 during ICU stay and pivot chart events to become variables/columns. Write all steps in *one* chain of pipes.

```
# Define the chart event items IDs for the vitals of interest
vital_item_ids <- c(220045, 220179, 220180, 223761, 220210)

# Connect to the chartevents table and perform the operations
chartevents_subset <- tbl(con_bq, "chartevents") |>
  # Join with icustays_tble to get only the subjects who are in ICU
  inner_join(icustays_tble, by = c("subject_id", "hadm_id", "stay_id")) |>
  # Filter for the specific chart event items
  filter(itemid %in% vital_item_ids) |>
  # Further restrict to measurements within the ICU stay time range
  filter(charttime >= intime & charttime <= outtime) |>
  group_by(subject_id, hadm_id, stay_id, itemid) |>
  # Use a window function to rank rows
  mutate(rn = row_number()) |>
  filter(rn == 1) |>
  ungroup() |>
  # Pivot to have one row per ICU stay
  pivot_wider(
    id_cols = c(subject_id, stay_id),
    names_from = itemid,
    values_from = value
  ) |>
  # Rename the columns to the names of the vitals
  rename(
    heart_rate = `220045`,
    systolic_bp = `220179`,
    diastolic_bp = `220180`,
    temperature_f = `223761`,
    respiratory_rate = `220210`
  ) |>
  # Select the columns of interest
  select(subject_id, stay_id, heart_rate, systolic_bp,
         diastolic_bp, temperature_f, respiratory_rate) |>
  # Print the resulting table with an unlimited width
  print(width = Inf)
```

```
# Source:   SQL [?? x 7]
# Database: BigQueryConnection
   subject_id  stay_id heart_rate systolic_bp diastolic_bp temperature_f
```

```
        <int>    <int> <chr>      <chr>       <chr>        <chr>
 1   10054716 33668354 105        137         53           99
 2   10067389 34081592 60         134         67           97.8
 3   10080961 39385708 75         143         104          99.7
 4   10092227 33365234 92         115         61           98.1
 5   10103748 30137539 86         75          54           98.3
 6   10172391 31128519 42         113         56           98.3
 7   10195266 37270142 106        108         82           98.4
 8   10215797 35517385 76         110         60           97.2
 9   10223372 39703863 52         131         45           100
10   10268465 37580514 96         127         75           98.8
   respiratory_rate
   <chr>
 1 13
 2 24
 3 25
 4 11
 5 30
 6 18
 7 18
 8 19
 9 14
10 14
# ℹ more rows
```

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 >= 18), (iv) merge in the labevents and chartevents tables, (v) `collect` the tibble.

```r
final_tibble <- icustays_tble |>
  # Join on admissions table
  left_join(admissions_tble, by = c("subject_id", "hadm_id")) |>
  # Join on patients table
  left_join(patients_tble, by = "subject_id") |>
  # Filter adults only
  filter(anchor_age >= 18) |>
  # Join on labevents table.
  left_join(labevents_subset, by = c("subject_id", "stay_id")) |>
  # Join on chartevents table.
  left_join(chartevents_subset, by = c("subject_id",  "stay_id")) |>
  # Remove potential duplicate rows resulting from the join
  distinct(subject_id, hadm_id, stay_id, .keep_all = TRUE) |>
  # Collect the data into a local tibble
  collect()
```

```
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?
```

```r
final_tibble
```

```
# A tibble: 73,181 × 40
   subject_id  hadm_id  stay_id first_careunit last_careunit intime
        <int>    <int>    <int> <chr>          <chr>         <dttm>
 1   10619216 23605772 30896840 Neuro Stepdown Neuro Stepdo… 2173-10-27 08:46:43
 2   10943097 25582886 30861231 Neuro Stepdown Neuro Stepdo… 2142-11-29 18:24:15
 3   11117134 29518408 33008527 Trauma SICU (… Neuro Stepdo… 2170-05-26 10:17:00
 4   11724187 21323952 31715618 Neuro Surgica… Neuro Stepdo… 2170-01-09 20:07:00
 5   11888962 29876201 36383794 Neuro Surgica… Neuro Stepdo… 2141-02-07 04:26:00
 6   12274603 25305605 33723954 Neuro Stepdown Neuro Stepdo… 2128-12-03 21:04:32
 7   12358448 21325236 34156555 Neuro Stepdown Neuro Stepdo… 2123-08-14 14:08:10
 8   12459868 23684918 33739667 Neuro Stepdown Neuro Stepdo… 2184-03-14 00:41:00
 9   13362926 23772748 31987058 Neuro Stepdown Neuro Stepdo… 2118-09-18 19:14:23
10   14004235 28438913 38616973 Neuro Stepdown Neuro Stepdo… 2160-07-13 18:08:00
# ℹ 73,171 more rows
# ℹ 34 more variables: outtime <dttm>, los <dbl>, admittime <dttm>,
#   dischtime <dttm>, deathtime <dttm>, admission_type <chr>,
#   admit_provider_id <chr>, admission_location <chr>,
#   discharge_location <chr>, insurance <chr>, language <chr>,
#   marital_status <chr>, race <chr>, edregtime <dttm>, edouttime <dttm>,
#   hospital_expire_flag <int>, gender <chr>, anchor_age <int>, …
```

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` and `fct_collapse` from the `forcats` package can be useful.

```r
final_tibble <- final_tibble %>%
  mutate(
    # Lump infrequent care unit levels into "Other"
    first_careunit = fct_lump_n(first_careunit, n = 4, other_level = "Other"),
    last_careunit = fct_lump_n(last_careunit, n = 4, other_level = "Other"),
    admission_type = fct_lump_n(admission_type, n = 4, other_level = "Other"),
    admission_location = fct_lump_n(admission_location, n = 3, other_level = "Other"),
    discharge_location = fct_lump_n(discharge_location, n = 4, other_level = "Other"),
    # Collapse race levels into specified categories
    race = fct_collapse(
      race,
      ASIAN = c("Asian", "ASIAN - ASIAN INDIAN", "ASIAN - CHINESE",
                "ASIAN - KOREAN", "ASIAN - SOUTH EAST ASIAN"),
      BLACK = c("Black/African", "BLACK/AFRICAN", "BLACK/AFRICAN AMERICAN",
                "BLACK/CAPE VERDEAN", "BLACK/CARIBBEAN ISLAND"),
      HISPANIC = c("HISPANIC OR LATINO", "HISPANIC/LATINO - CENTRAL AMERICAN",
                   "HISPANIC/LATINO - COLUMBIAN", "HISPANIC/LATINO - CUBAN",
                   "HISPANIC/LATINO - DOMINICAN", "HISPANIC/LATINO - GUATEMALAN",
                   "HISPANIC/LATINO - HONDURAN", "HISPANIC/LATINO - MEXICAN",
                   "HISPANIC/LATINO - PUERTO RICAN", "HISPANIC/LATINO - SALVADORAN"),
      WHITE = c("White", "WHITE - BRAZILIAN", "WHITE - EASTERN EUROPEAN",
                "WHITE - OTHER EUROPEAN", "WHITE - RUSSIAN"),
      Other = c("AMERICAN INDIAN/ALASKA NATIVE", "MULTIPLE RACE/ETHNICITY",
                "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER", "OTHER",
                "PATIENT DECLINED TO ANSWER", "PORTUGUESE",
                "SOUTH AMERICAN", "UNABLE TO OBTAIN", "UNKNOWN")
    ),
    # Create new variable for length of stay
    los_long = los >= 2
  )
```

```
Warning: There was 1 warning in `mutate()`.
ℹ In argument: `race = fct_collapse(...)`.
Caused by warning:
! Unknown levels in `f`: Asian, Black/African, White
```

```
final_tibble <- final_tibble %>%
  mutate(across(c(creatinine, potassium, sodium, chloride, bicarbonate, hematocrit,
                  white_blood_cell_count, glucose, heart_rate, systolic_bp,
                  diastolic_bp, temperature_f, respiratory_rate),
                ~as.numeric(as.character(.))))
```

```
mimic_icu_cohort <- final_tibble %>%
  select(-subject_id, -hadm_id, -stay_id, -outtime, -admittime,
         -dischtime, -deathtime, -admit_provider_id, -edouttime, -edregtime,
         -anchor_year, -anchor_year_group) %>%
  tbl_summary(
    by = los_long, # Stratify by los_long
    type = all_continuous() ~ "continuous2",
    statistic = all_continuous() ~ c("{mean} ({sd})", "{min}, {max}"),
    missing = "no" # Exclude missing data from the main statistics
  )

# Print the summary table
print(mimic_icu_cohort)
```

```
<div id="ksunqwyvso" style="padding-left:0px;padding-right:0px;padding-top:10px;padding-
bottom:10px;overflow-x:auto;overflow-y:auto;width:auto;height:auto;">
  <style>#ksunqwyvso table {
  font-family: system-ui, 'Segoe UI', Roboto, Helvetica, Arial, sans-serif, 'Apple Color
Emoji', 'Segoe UI Emoji', 'Segoe UI Symbol', 'Noto Color Emoji';
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

#ksunqwyvso thead, #ksunqwyvso tbody, #ksunqwyvso tfoot, #ksunqwyvso tr, #ksunqwyvso td,
#ksunqwyvso th {
  border-style: none;
}

#ksunqwyvso p {
  margin: 0;
  padding: 0;
}

#ksunqwyvso .gt_table {
  display: table;
  border-collapse: collapse;
  line-height: normal;
  margin-left: auto;
  margin-right: auto;
  color: #333333;
  font-size: 16px;
  font-weight: normal;
  font-style: normal;
  background-color: #FFFFFF;
  width: auto;
  border-top-style: solid;
  border-top-width: 2px;
  border-top-color: #A8A8A8;
  border-right-style: none;
  border-right-width: 2px;
  border-right-color: #D3D3D3;
  border-bottom-style: solid;
  border-bottom-width: 2px;
  border-bottom-color: #A8A8A8;
  border-left-style: none;
  border-left-width: 2px;
  border-left-color: #D3D3D3;
}

#ksunqwyvso .gt_caption {
  padding-top: 4px;
  padding-bottom: 4px;
}

#ksunqwyvso .gt_title {
  color: #333333;
  font-size: 125%;
  font-weight: initial;
  padding-top: 4px;
  padding-bottom: 4px;
  padding-left: 5px;
  padding-right: 5px;
  border-bottom-color: #FFFFFF;
  border-bottom-width: 0;
}

#ksunqwyvso .gt_subtitle {
  color: #333333;
  font-size: 85%;
  font-weight: initial;
  padding-top: 3px;
  padding-bottom: 5px;
  padding-left: 5px;
  padding-right: 5px;
  border-top-color: #FFFFFF;
  border-top-width: 0;
}

#ksunqwyvso .gt_heading {
  background-color: #FFFFFF;
  text-align: center;
  border-bottom-color: #FFFFFF;
  border-left-style: none;
  border-left-width: 1px;
  border-left-color: #D3D3D3;
  border-right-style: none;
  border-right-width: 1px;
  border-right-color: #D3D3D3;
}

#ksunqwyvso .gt_bottom_border {
  border-bottom-style: solid;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
}

#ksunqwyvso .gt_col_headings {
  border-top-style: solid;
  border-top-width: 2px;
  border-top-color: #D3D3D3;
  border-bottom-style: solid;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
  border-left-style: none;
  border-left-width: 1px;
  border-left-color: #D3D3D3;
  border-right-style: none;
  border-right-width: 1px;
  border-right-color: #D3D3D3;
}

#ksunqwyvso .gt_col_heading {
  color: #333333;
  background-color: #FFFFFF;
  font-size: 100%;
  font-weight: normal;
  text-transform: inherit;
  border-left-style: none;
  border-left-width: 1px;
  border-left-color: #D3D3D3;
  border-right-style: none;
  border-right-width: 1px;
  border-right-color: #D3D3D3;
  vertical-align: bottom;
  padding-top: 5px;
  padding-bottom: 6px;
  padding-left: 5px;
  padding-right: 5px;
  overflow-x: hidden;
}

#ksunqwyvso .gt_column_spanner_outer {
  color: #333333;
  background-color: #FFFFFF;
  font-size: 100%;
  font-weight: normal;
  text-transform: inherit;
  padding-top: 0;
  padding-bottom: 0;
  padding-left: 4px;
  padding-right: 4px;
}

#ksunqwyvso .gt_column_spanner_outer:first-child {
  padding-left: 0;
}

#ksunqwyvso .gt_column_spanner_outer:last-child {
  padding-right: 0;
}

#ksunqwyvso .gt_column_spanner {
  border-bottom-style: solid;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
  vertical-align: bottom;
  padding-top: 5px;
  padding-bottom: 5px;
  overflow-x: hidden;
  display: inline-block;
  width: 100%;
}

#ksunqwyvso .gt_spanner_row {
  border-bottom-style: hidden;
}

#ksunqwyvso .gt_group_heading {
  padding-top: 8px;
  padding-bottom: 8px;
  padding-left: 5px;
  padding-right: 5px;
  color: #333333;
  background-color: #FFFFFF;
  font-size: 100%;
  font-weight: initial;
  text-transform: inherit;
  border-top-style: solid;
  border-top-width: 2px;
  border-top-color: #D3D3D3;
  border-bottom-style: solid;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
  border-left-style: none;
  border-left-width: 1px;
  border-left-color: #D3D3D3;
  border-right-style: none;
  border-right-width: 1px;
  border-right-color: #D3D3D3;
  vertical-align: middle;
  text-align: left;
}

#ksunqwyvso .gt_empty_group_heading {
  padding: 0.5px;
  color: #333333;
  background-color: #FFFFFF;
  font-size: 100%;
  font-weight: initial;
  border-top-style: solid;
  border-top-width: 2px;
  border-top-color: #D3D3D3;
  border-bottom-style: solid;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
  vertical-align: middle;
}

#ksunqwyvso .gt_from_md > :first-child {
  margin-top: 0;
}

#ksunqwyvso .gt_from_md > :last-child {
  margin-bottom: 0;
}

#ksunqwyvso .gt_row {
  padding-top: 8px;
  padding-bottom: 8px;
  padding-left: 5px;
  padding-right: 5px;
  margin: 10px;
  border-top-style: solid;
  border-top-width: 1px;
  border-top-color: #D3D3D3;
  border-left-style: none;
  border-left-width: 1px;
  border-left-color: #D3D3D3;
  border-right-style: none;
  border-right-width: 1px;
  border-right-color: #D3D3D3;
  vertical-align: middle;
  overflow-x: hidden;
}

#ksunqwyvso .gt_stub {
  color: #333333;
  background-color: #FFFFFF;
  font-size: 100%;
  font-weight: initial;
  text-transform: inherit;
  border-right-style: solid;
  border-right-width: 2px;
  border-right-color: #D3D3D3;
  padding-left: 5px;
  padding-right: 5px;
}

#ksunqwyvso .gt_stub_row_group {
  color: #333333;
  background-color: #FFFFFF;
  font-size: 100%;
  font-weight: initial;
  text-transform: inherit;
  border-right-style: solid;
  border-right-width: 2px;
  border-right-color: #D3D3D3;
  padding-left: 5px;
  padding-right: 5px;
  vertical-align: top;
}

#ksunqwyvso .gt_row_group_first td {
  border-top-width: 2px;
}

#ksunqwyvso .gt_row_group_first th {
  border-top-width: 2px;
}

#ksunqwyvso .gt_summary_row {
  color: #333333;
  background-color: #FFFFFF;
  text-transform: inherit;
  padding-top: 8px;
  padding-bottom: 8px;
  padding-left: 5px;
  padding-right: 5px;
}

#ksunqwyvso .gt_first_summary_row {
  border-top-style: solid;
  border-top-color: #D3D3D3;
}

#ksunqwyvso .gt_first_summary_row.thick {
  border-top-width: 2px;
}

#ksunqwyvso .gt_last_summary_row {
  padding-top: 8px;
  padding-bottom: 8px;
  padding-left: 5px;
  padding-right: 5px;
  border-bottom-style: solid;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
}

#ksunqwyvso .gt_grand_summary_row {
  color: #333333;
  background-color: #FFFFFF;
  text-transform: inherit;
  padding-top: 8px;
  padding-bottom: 8px;
  padding-left: 5px;
  padding-right: 5px;
}

#ksunqwyvso .gt_first_grand_summary_row {
  padding-top: 8px;
  padding-bottom: 8px;
  padding-left: 5px;
  padding-right: 5px;
  border-top-style: double;
  border-top-width: 6px;
  border-top-color: #D3D3D3;
}

#ksunqwyvso .gt_last_grand_summary_row_top {
  padding-top: 8px;
  padding-bottom: 8px;
  padding-left: 5px;
  padding-right: 5px;
  border-bottom-style: double;
  border-bottom-width: 6px;
  border-bottom-color: #D3D3D3;
}

#ksunqwyvso .gt_striped {
  background-color: rgba(128, 128, 128, 0.05);
}

#ksunqwyvso .gt_table_body {
  border-top-style: solid;
  border-top-width: 2px;
  border-top-color: #D3D3D3;
  border-bottom-style: solid;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
}

#ksunqwyvso .gt_footnotes {
  color: #333333;
  background-color: #FFFFFF;
  border-bottom-style: none;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
  border-left-style: none;
  border-left-width: 2px;
  border-left-color: #D3D3D3;
  border-right-style: none;
  border-right-width: 2px;
  border-right-color: #D3D3D3;
}

#ksunqwyvso .gt_footnote {
  margin: 0px;
  font-size: 90%;
  padding-top: 4px;
  padding-bottom: 4px;
  padding-left: 5px;
  padding-right: 5px;
}

#ksunqwyvso .gt_sourcenotes {
  color: #333333;
  background-color: #FFFFFF;
  border-bottom-style: none;
  border-bottom-width: 2px;
  border-bottom-color: #D3D3D3;
  border-left-style: none;
  border-left-width: 2px;
  border-left-color: #D3D3D3;
  border-right-style: none;
  border-right-width: 2px;
  border-right-color: #D3D3D3;
}

#ksunqwyvso .gt_sourcenote {
  font-size: 90%;
  padding-top: 4px;
  padding-bottom: 4px;
  padding-left: 5px;
  padding-right: 5px;
}

#ksunqwyvso .gt_left {
  text-align: left;
}

#ksunqwyvso .gt_center {
  text-align: center;
}

#ksunqwyvso .gt_right {
  text-align: right;
  font-variant-numeric: tabular-nums;
}

#ksunqwyvso .gt_font_normal {
  font-weight: normal;
}
```

```
#ksunqwyvso .gt_font_bold {
  font-weight: bold;
}

#ksunqwyvso .gt_font_italic {
  font-style: italic;
}

#ksunqwyvso .gt_super {
  font-size: 65%;
}

#ksunqwyvso .gt_footnote_marks {
  font-size: 75%;
  vertical-align: 0.4em;
  position: initial;
}

#ksunqwyvso .gt_asterisk {
  font-size: 100%;
  vertical-align: 0;
}

#ksunqwyvso .gt_indent_1 {
  text-indent: 5px;
}

#ksunqwyvso .gt_indent_2 {
  text-indent: 10px;
}

#ksunqwyvso .gt_indent_3 {
  text-indent: 15px;
}

#ksunqwyvso .gt_indent_4 {
  text-indent: 20px;
}

#ksunqwyvso .gt_indent_5 {
  text-indent: 25px;
}
</style>
  <table class="gt_table" data-quarto-disable-processing="false" data-quarto-
bootstrap="false">
  <thead>
    <tr class="gt_col_headings">
      <th class="gt_col_heading gt_columns_bottom_border gt_left" rowspan="1" colspan="1"
scope="col" id="&lt;strong&gt;Characteristic&lt;/strong&gt;"><strong>Characteristic</strong>
</th>
      <th class="gt_col_heading gt_columns_bottom_border gt_center" rowspan="1" colspan="1"
scope="col" id="&lt;strong&gt;FALSE&lt;/strong&gt;, N = 38,050&lt;span
class=&quot;gt_footnote_marks&quot; style=&quot;white-space:nowrap;font-style:italic;font-
```

```
weight:normal;&quot;&gt;&lt;sup&gt;1&lt;/sup&gt;&lt;/span&gt;"><strong>FALSE</strong>, N =
38,050<span class="gt_footnote_marks" style="white-space:nowrap;font-style:italic;font-
weight:normal;"><sup>1</sup></span></th>
      <th class="gt_col_heading gt_columns_bottom_border gt_center" rowspan="1" colspan="1"
scope="col" id="&lt;strong&gt;TRUE&lt;/strong&gt;, N = 35,131&lt;span
class=&quot;gt_footnote_marks&quot; style=&quot;white-space:nowrap;font-style:italic;font-
weight:normal;&quot;&gt;&lt;sup&gt;1&lt;/sup&gt;&lt;/span&gt;"><strong>TRUE</strong>, N =
35,131<span class="gt_footnote_marks" style="white-space:nowrap;font-style:italic;font-
weight:normal;"><sup>1</sup></span></th>
    </tr>
  </thead>
  <tbody class="gt_table_body">
    <tr><td headers="label" class="gt_row gt_left">first_careunit</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Cardiac Vascular Intensive Care Unit
(CVICU)</td>
<td headers="stat_1" class="gt_row gt_center">5,827 (15%)</td>
<td headers="stat_2" class="gt_row gt_center">5,755 (16%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Medical Intensive Care Unit (MICU)</td>
<td headers="stat_1" class="gt_row gt_center">8,782 (23%)</td>
<td headers="stat_2" class="gt_row gt_center">7,116 (20%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Medical/Surgical Intensive Care Unit
(MICU/SICU)</td>
<td headers="stat_1" class="gt_row gt_center">7,147 (19%)</td>
<td headers="stat_2" class="gt_row gt_center">5,586 (16%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Surgical Intensive Care Unit (SICU)
</td>
<td headers="stat_1" class="gt_row gt_center">5,654 (15%)</td>
<td headers="stat_2" class="gt_row gt_center">5,507 (16%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Other</td>
<td headers="stat_1" class="gt_row gt_center">10,640 (28%)</td>
<td headers="stat_2" class="gt_row gt_center">11,167 (32%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">last_careunit</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Cardiac Vascular Intensive Care Unit
(CVICU)</td>
<td headers="stat_1" class="gt_row gt_center">5,773 (15%)</td>
<td headers="stat_2" class="gt_row gt_center">5,614 (16%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Medical Intensive Care Unit (MICU)</td>
<td headers="stat_1" class="gt_row gt_center">8,929 (23%)</td>
<td headers="stat_2" class="gt_row gt_center">7,903 (22%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Medical/Surgical Intensive Care Unit
(MICU/SICU)</td>
<td headers="stat_1" class="gt_row gt_center">7,051 (19%)</td>
<td headers="stat_2" class="gt_row gt_center">5,303 (15%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Surgical Intensive Care Unit (SICU)
</td>
<td headers="stat_1" class="gt_row gt_center">5,665 (15%)</td>
<td headers="stat_2" class="gt_row gt_center">5,344 (15%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Other</td>
<td headers="stat_1" class="gt_row gt_center">10,632 (28%)</td>
<td headers="stat_2" class="gt_row gt_center">10,967 (31%)</td></tr>
```

```
    <tr><td headers="label" class="gt_row gt_left">intime</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">2153-11-22 10:59:26.236584 (757647737.588742)
</td>
<td headers="stat_2" class="gt_row gt_center">2153-11-05 13:43:01.640203 (753647601.252698)
</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">2110-01-11 10:16:06, 2211-01-05 10:59:50</td>
<td headers="stat_2" class="gt_row gt_center">2110-01-12 00:54:00, 2211-11-09 03:33:00</td>
</tr>
    <tr><td headers="label" class="gt_row gt_left">los</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">1.1 (0.5)</td>
<td headers="stat_2" class="gt_row gt_center">6.0 (6.2)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">0.0, 2.0</td>
<td headers="stat_2" class="gt_row gt_center">2.0, 110.2</td></tr>
    <tr><td headers="label" class="gt_row gt_left">admission_type</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    EW EMER.</td>
<td headers="stat_1" class="gt_row gt_center">20,735 (54%)</td>
<td headers="stat_2" class="gt_row gt_center">17,937 (51%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    OBSERVATION ADMIT</td>
<td headers="stat_1" class="gt_row gt_center">4,329 (11%)</td>
<td headers="stat_2" class="gt_row gt_center">4,645 (13%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    SURGICAL SAME DAY ADMISSION</td>
<td headers="stat_1" class="gt_row gt_center">4,340 (11%)</td>
<td headers="stat_2" class="gt_row gt_center">3,033 (8.6%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    URGENT</td>
<td headers="stat_1" class="gt_row gt_center">5,494 (14%)</td>
<td headers="stat_2" class="gt_row gt_center">6,959 (20%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Other</td>
<td headers="stat_1" class="gt_row gt_center">3,152 (8.3%)</td>
<td headers="stat_2" class="gt_row gt_center">2,557 (7.3%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">admission_location</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    EMERGENCY ROOM</td>
<td headers="stat_1" class="gt_row gt_center">19,602 (52%)</td>
<td headers="stat_2" class="gt_row gt_center">16,279 (46%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    PHYSICIAN REFERRAL</td>
<td headers="stat_1" class="gt_row gt_center">8,986 (24%)</td>
<td headers="stat_2" class="gt_row gt_center">7,412 (21%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    TRANSFER FROM HOSPITAL</td>
<td headers="stat_1" class="gt_row gt_center">6,839 (18%)</td>
<td headers="stat_2" class="gt_row gt_center">8,959 (26%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Other</td>
<td headers="stat_1" class="gt_row gt_center">2,623 (6.9%)</td>
<td headers="stat_2" class="gt_row gt_center">2,481 (7.1%)</td></tr>
```

```
    <tr><td headers="label" class="gt_row gt_left">discharge_location</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    DIED</td>
<td headers="stat_1" class="gt_row gt_center">3,298 (8.8%)</td>
<td headers="stat_2" class="gt_row gt_center">5,006 (14%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    HOME</td>
<td headers="stat_1" class="gt_row gt_center">12,192 (33%)</td>
<td headers="stat_2" class="gt_row gt_center">5,255 (15%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    HOME HEALTH CARE</td>
<td headers="stat_1" class="gt_row gt_center">10,347 (28%)</td>
<td headers="stat_2" class="gt_row gt_center">7,762 (22%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    SKILLED NURSING FACILITY</td>
<td headers="stat_1" class="gt_row gt_center">6,239 (17%)</td>
<td headers="stat_2" class="gt_row gt_center">7,158 (20%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Other</td>
<td headers="stat_1" class="gt_row gt_center">5,402 (14%)</td>
<td headers="stat_2" class="gt_row gt_center">9,885 (28%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">insurance</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Medicaid</td>
<td headers="stat_1" class="gt_row gt_center">3,060 (8.0%)</td>
<td headers="stat_2" class="gt_row gt_center">2,468 (7.0%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Medicare</td>
<td headers="stat_1" class="gt_row gt_center">16,489 (43%)</td>
<td headers="stat_2" class="gt_row gt_center">16,602 (47%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Other</td>
<td headers="stat_1" class="gt_row gt_center">18,501 (49%)</td>
<td headers="stat_2" class="gt_row gt_center">16,061 (46%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">language</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    ?</td>
<td headers="stat_1" class="gt_row gt_center">3,692 (9.7%)</td>
<td headers="stat_2" class="gt_row gt_center">3,751 (11%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    ENGLISH</td>
<td headers="stat_1" class="gt_row gt_center">34,358 (90%)</td>
<td headers="stat_2" class="gt_row gt_center">31,380 (89%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">marital_status</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    DIVORCED</td>
<td headers="stat_1" class="gt_row gt_center">2,843 (8.0%)</td>
<td headers="stat_2" class="gt_row gt_center">2,561 (7.9%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    MARRIED</td>
<td headers="stat_1" class="gt_row gt_center">16,984 (48%)</td>
<td headers="stat_2" class="gt_row gt_center">15,784 (49%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    SINGLE</td>
<td headers="stat_1" class="gt_row gt_center">11,175 (31%)</td>
<td headers="stat_2" class="gt_row gt_center">9,683 (30%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    WIDOWED</td>
<td headers="stat_1" class="gt_row gt_center">4,699 (13%)</td>
<td headers="stat_2" class="gt_row gt_center">4,339 (13%)</td></tr>
```

```
    <tr><td headers="label" class="gt_row gt_left">race</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Other</td>
<td headers="stat_1" class="gt_row gt_center">5,160 (14%)</td>
<td headers="stat_2" class="gt_row gt_center">5,596 (16%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    ASIAN</td>
<td headers="stat_1" class="gt_row gt_center">1,148 (3.0%)</td>
<td headers="stat_2" class="gt_row gt_center">1,007 (2.9%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    BLACK</td>
<td headers="stat_1" class="gt_row gt_center">4,311 (11%)</td>
<td headers="stat_2" class="gt_row gt_center">3,649 (10%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    HISPANIC</td>
<td headers="stat_1" class="gt_row gt_center">1,492 (3.9%)</td>
<td headers="stat_2" class="gt_row gt_center">1,249 (3.6%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    WHITE</td>
<td headers="stat_1" class="gt_row gt_center">25,939 (68%)</td>
<td headers="stat_2" class="gt_row gt_center">23,630 (67%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">hospital_expire_flag</td>
<td headers="stat_1" class="gt_row gt_center">3,357 (8.8%)</td>
<td headers="stat_2" class="gt_row gt_center">4,972 (14%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">gender</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    F</td>
<td headers="stat_1" class="gt_row gt_center">17,014 (45%)</td>
<td headers="stat_2" class="gt_row gt_center">15,349 (44%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    M</td>
<td headers="stat_1" class="gt_row gt_center">21,036 (55%)</td>
<td headers="stat_2" class="gt_row gt_center">19,782 (56%)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">anchor_age</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">62 (17)</td>
<td headers="stat_2" class="gt_row gt_center">64 (16)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">18, 91</td>
<td headers="stat_2" class="gt_row gt_center">18, 91</td></tr>
    <tr><td headers="label" class="gt_row gt_left">dod</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">2155-12-30 (8868.9129680826)</td>
<td headers="stat_2" class="gt_row gt_center">2155-06-10 (8845.76109740791)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">2110-02-08, 2210-12-04</td>
<td headers="stat_2" class="gt_row gt_center">2110-01-25, 2211-01-17</td></tr>
    <tr><td headers="label" class="gt_row gt_left">hematocrit</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">35 (7)</td>
<td headers="stat_2" class="gt_row gt_center">34 (7)</td></tr>
```

```
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">8, 63</td>
<td headers="stat_2" class="gt_row gt_center">7, 70</td></tr>
    <tr><td headers="label" class="gt_row gt_left">chloride</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">101 (6)</td>
<td headers="stat_2" class="gt_row gt_center">101 (7)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">45, 140</td>
<td headers="stat_2" class="gt_row gt_center">46, 144</td></tr>
    <tr><td headers="label" class="gt_row gt_left">creatinine</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">1.44 (1.68)</td>
<td headers="stat_2" class="gt_row gt_center">1.55 (1.68)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">0.00, 41.50</td>
<td headers="stat_2" class="gt_row gt_center">0.10, 43.00</td></tr>
    <tr><td headers="label" class="gt_row gt_left">sodium</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">138.0 (5.0)</td>
<td headers="stat_2" class="gt_row gt_center">137.9 (5.6)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">106.0, 179.0</td>
<td headers="stat_2" class="gt_row gt_center">74.0, 180.0</td></tr>
    <tr><td headers="label" class="gt_row gt_left">glucose</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">143 (93)</td>
<td headers="stat_2" class="gt_row gt_center">145 (88)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">4, 2,202</td>
<td headers="stat_2" class="gt_row gt_center">6, 2,340</td></tr>
    <tr><td headers="label" class="gt_row gt_left">bicarbonate</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">24.3 (4.9)</td>
<td headers="stat_2" class="gt_row gt_center">24.4 (5.3)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">2.0, 49.0</td>
<td headers="stat_2" class="gt_row gt_center">2.0, 50.0</td></tr>
    <tr><td headers="label" class="gt_row gt_left">white_blood_cell_count</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">10.6 (9.3)</td>
<td headers="stat_2" class="gt_row gt_center">11.5 (10.1)</td></tr>
```

```
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">0.1, 385.9</td>
<td headers="stat_2" class="gt_row gt_center">0.1, 479.2</td></tr>
    <tr><td headers="label" class="gt_row gt_left">potassium</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">4.30 (0.79)</td>
<td headers="stat_2" class="gt_row gt_center">4.33 (0.80)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">1.50, 10.00</td>
<td headers="stat_2" class="gt_row gt_center">1.30, 10.00</td></tr>
    <tr><td headers="label" class="gt_row gt_left">heart_rate</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">83 (33)</td>
<td headers="stat_2" class="gt_row gt_center">86 (18)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">0, 5,409</td>
<td headers="stat_2" class="gt_row gt_center">0, 214</td></tr>
    <tr><td headers="label" class="gt_row gt_left">systolic_bp</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">120 (89)</td>
<td headers="stat_2" class="gt_row gt_center">119 (22)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">0, 12,262</td>
<td headers="stat_2" class="gt_row gt_center">0, 247</td></tr>
    <tr><td headers="label" class="gt_row gt_left">diastolic_bp</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">66 (49)</td>
<td headers="stat_2" class="gt_row gt_center">66 (268)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">0, 9,060</td>
<td headers="stat_2" class="gt_row gt_center">0, 49,155</td></tr>
    <tr><td headers="label" class="gt_row gt_left">temperature_f</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">98.19 (7.03)</td>
<td headers="stat_2" class="gt_row gt_center">98.36 (2.17)</td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">0.00, 991.00</td>
<td headers="stat_2" class="gt_row gt_center">0.00, 105.70</td></tr>
    <tr><td headers="label" class="gt_row gt_left">respiratory_rate</td>
<td headers="stat_1" class="gt_row gt_center"><br /></td>
<td headers="stat_2" class="gt_row gt_center"><br /></td></tr>
    <tr><td headers="label" class="gt_row gt_left">    Mean (SD)</td>
<td headers="stat_1" class="gt_row gt_center">18.8 (5.6)</td>
<td headers="stat_2" class="gt_row gt_center">19.7 (5.8)</td></tr>
```

```
    <tr><td headers="label" class="gt_row gt_left">    Range</td>
<td headers="stat_1" class="gt_row gt_center">0.0, 241.0</td>
<td headers="stat_2" class="gt_row gt_center">0.0, 93.0</td></tr>
  </tbody>
  
  <tfoot class="gt_footnotes">
    <tr>
      <td class="gt_footnote" colspan="3"><span class="gt_footnote_marks" style="white-space:nowrap;font-style:italic;font-weight:normal;"><sup>1</sup></span> n (%)</td>
    </tr>
  </tfoot>
</table>
</div>
```

Q1.9 Save the final tibble

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

```r
# make a directory mimiciv_shiny
if (!dir.exists("mimiciv_shiny")) {
  dir.create("mimiciv_shiny")
}

mimic_icu_cohort |>
  write_rds("mimiciv_shiny/mimic_icu_cohort.rds", compress = "gz")
```

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. The other 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.

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

# Load the dataset
mimic_icu_cohort <- readRDS("mimiciv_shiny/mimic_icu_cohort.rds")



last_careunit_plot <- ggplot(final_tibble, aes(x = last_careunit)) +
  geom_bar() +
  labs(x = "Last Care Unit", y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))


labevents_long <- labevents_subset %>%
  pivot_longer(
    cols = c(creatinine, potassium, sodium, chloride, bicarbonate,
             hematocrit, white_blood_cell_count, glucose),
    names_to = "measurement",
    values_to = "value"
  )

labevents_plot <- ggplot(labevents_long, aes(x = measurement, y = value)) +
  geom_boxplot() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Measurement", y = "Value")
```

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

```
ui <- fluidPage(
  titlePanel("ICU Cohort Data Exploration"),
  navbarPage("ICU Data", id = "nav",
    tabPanel("Patient characteristics",
      sidebarLayout(
        sidebarPanel(
          selectInput("variable",
                      "Variable of interest",
                      choices = c("Last Care Unit" = "last_careunit",
                                  "Lab Events" = "lab_events")),
          checkboxInput("outliers",
                        "Remove outliers in IQR method for measurements?",
                        FALSE)
        ),
        mainPanel(
          plotOutput("variablePlot")
        )
      )
    ),
   tabPanel("Patients' ADT and ICU stay information",
      sidebarLayout(
        sidebarPanel(
          numericInput("patient_id", "Select a Patient ID", value = 1, min = 1)
        ),
        mainPanel(
          plotOutput("patientPlot")  # This line outputs the plot
        )
      )
    )
  )
)


# Define server logic to summarize and view selected dataset ----
server <- function(input, output) {

  # Output plot for the selected variable
  output$variablePlot <- renderPlot({
    # Check if 'last_careunit' is selected and render the pre-created plot
    if (input$variable == "last_careunit") {
      last_careunit_plot
    } else {
      labevents_plot
    }
  })

  # Output patient information for the selected patient ID
  output$patientInfo <- renderPrint({
    req(input$patientID)
    patient_data <- mimic_icu_cohort %>% filter(subject_id == input$patientID)
    patient_data
  })


# Generate the plot for the selected patient ID
  output$patientPlot <- renderPlot({
    # Ensure a patient ID is provided
    req(input$patient_id)

    selected_patient_id <- as.numeric(input$patient_id)
```

```r
    # Filter the final_tibble for the selected patient ID
    patient_data <- final_tibble %>%
      filter(subject_id == selected_patient_id)

    ggplot(patient_data) +
      geom_segment(aes(x = intime, xend = outtime, y = "ADT", yend = "ADT",
                       color = careunit, size = if_else(str_detect(careunit, "ICU|CCU"), 1.2,
      geom_point(aes(x = charttime, y = "Lab"), shape = 3) +
      geom_point(aes(x = chartdate, y = "Procedure", shape = long_title)) +
      labs(x = "Calendar Time", y = "Event Type",
           title = sprintf("Patient %s, %s, %d years old, %s", selected_patient_id,
                           unique(patient_data$gender), unique(patient_data$anchor_age),
                           tolower(unique(patient_data$race))),
           subtitle = stringr::str_c(tolower(patient_data$long_title[1:3]), collapse = "\n"),
           shape = "Procedure") +
      scale_color_discrete(name = "Care Unit") +
      scale_y_discrete(limits = c("Procedure", "Lab", "ADT")) +
      theme_minimal() +
      theme(legend.position = "bottom", legend.box = "vertical")
  })
}




# Create Shiny app
shinyApp(ui, server)
```

Shiny applications not supported in static R Markdown documents