Biostat 203B Homework 4

Due Mar 9 @ 11:59PM

AUTHOR

Sophia Luo, 106409469

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] htmlwidgets_1.6.4 compiler_4.4.2    fastmap_1.2.0     cli_3.6.3
 [5] tools_4.4.2       htmltools_0.5.8.1 rstudioapi_0.17.1 yaml_2.3.10
 [9] rmarkdown_2.29    knitr_1.49        jsonlite_1.8.9    xfun_0.50
[13] digest_0.6.37     rlang_1.1.4       evaluate_1.0.1
```

Display my machine memory.

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

```
Totalram:  13.653 GiB
Freeram:   10.858 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 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.1     ✔ tibble    3.2.1
✔ lubridate 1.9.4     ✔ 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
```

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** 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"        "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"         
```

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>                                            <dttm>             
 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
   <dttm>              <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
```

Q1.3 admissions data

Connect to the admissions table.

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

```
# Source:     SQL [?? x 16]
# Database:   BigQueryConnection
# Ordered by: subject_id, hadm_id
   subject_id  hadm_id admittime           dischtime           deathtime
        <int>    <int> <dttm>              <dttm>              <dttm>
 1   10000032 22595853 2180-05-06 22:23:00 2180-05-07 17:15:00 NA
 2   10000032 22841357 2180-06-26 18:27:00 2180-06-27 18:49:00 NA
 3   10000032 25742920 2180-08-05 23:44:00 2180-08-07 17:50:00 NA
 4   10000032 29079034 2180-07-23 12:35:00 2180-07-25 17:55:00 NA
 5   10000068 25022803 2160-03-03 23:16:00 2160-03-04 06:26:00 NA
 6   10000084 23052089 2160-11-21 01:56:00 2160-11-25 14:52:00 NA
 7   10000084 29888819 2160-12-28 05:11:00 2160-12-28 16:07:00 NA
 8   10000108 27250926 2163-09-27 23:17:00 2163-09-28 09:04:00 NA
 9   10000117 22927623 2181-11-15 02:05:00 2181-11-15 14:52:00 NA
10   10000117 27988844 2183-09-18 18:10:00 2183-09-21 16:30:00 NA
   admission_type    admit_provider_id admission_location     discharge_location
   <chr>             <chr>             <chr>                  <chr>
 1 URGENT            P49AFC            TRANSFER FROM HOSPITAL HOME
 2 EW EMER.          P784FA            EMERGENCY ROOM         HOME
 3 EW EMER.          P19UTS            EMERGENCY ROOM         HOSPICE
 4 EW EMER.          P06OTX            EMERGENCY ROOM         HOME
 5 EU OBSERVATION    P39NWO            EMERGENCY ROOM         <NA>            
 6 EW EMER.          P42H7G            WALK-IN/SELF REFERRAL  HOME HEALTH CARE
 7 EU OBSERVATION    P35NE4            PHYSICIAN REFERRAL     <NA>            
 8 EU OBSERVATION    P40JML            EMERGENCY ROOM         <NA>            
 9 EU OBSERVATION    P47EY8            EMERGENCY ROOM         <NA>            
10 OBSERVATION ADMIT P13ACE            WALK-IN/SELF REFERRAL  HOME HEALTH CARE
   insurance language marital_status race  edregtime          
   <chr>     <chr>    <chr>          <chr> <dttm>             
 1 Medicaid  English  WIDOWED        WHITE 2180-05-06 19:17:00
 2 Medicaid  English  WIDOWED        WHITE 2180-06-26 15:54:00
 3 Medicaid  English  WIDOWED        WHITE 2180-08-05 20:58:00
 4 Medicaid  English  WIDOWED        WHITE 2180-07-23 05:54:00
 5 <NA>      English  SINGLE         WHITE 2160-03-03 21:55:00
 6 Medicare  English  MARRIED        WHITE 2160-11-20 20:36:00
 7 Medicare  English  MARRIED        WHITE 2160-12-27 18:32:00
 8 <NA>      English  SINGLE         WHITE 2163-09-27 16:18:00
 9 Medicaid  English  DIVORCED       WHITE 2181-11-14 21:51:00
10 Medicaid  English  DIVORCED       WHITE 2183-09-18 08:41:00
   edouttime           hospital_expire_flag
   <dttm>                             <int>
 1 2180-05-06 23:30:00                    0
 2 2180-06-26 21:31:00                    0
 3 2180-08-06 01:44:00                    0
 4 2180-07-23 14:00:00                    0
 5 2160-03-04 06:26:00                    0
 6 2160-11-21 03:20:00                    0
 7 2160-12-28 16:07:00                    0
 8 2163-09-28 09:04:00                    0
 9 2181-11-15 09:57:00                    0
10 2183-09-18 20:20:00                    0
# ℹ more rows
```

Q1.4 `patients` data

Connect to the `patients` table.

```r
patients_tble <- tbl(con_bq, "patients") |>
  arrange(subject_id) |>
  print(width = Inf)
```

```
# Source:     SQL [?? x 6]
# Database:   BigQueryConnection
# Ordered by: subject_id
   subject_id gender anchor_age anchor_year anchor_year_group dod       
        <int> <chr>       <int>       <int> <chr>             <date>    
 1   10000032 F              52        2180 2014 - 2016       2180-09-09
 2   10000048 F              23        2126 2008 - 2010       NA        
 3   10000058 F              33        2168 2020 - 2022       NA        
 4   10000068 F              19        2160 2008 - 2010       NA        
 5   10000084 M              72        2160 2017 - 2019       2161-02-13
 6   10000102 F              27        2136 2008 - 2010      NA
 7   10000108 M              25        2163 2014 - 2016      NA
 8   10000115 M              24        2154 2017 - 2019      NA
 9   10000117 F              48        2174 2008 - 2010      NA
10   10000161 M              60        2163 2020 - 2022      NA
# ℹ 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 (by `storetime`) before the ICU stay and pivot lab items to become variables/columns. Write all steps in *one* chain of pipes.

```
itemids <- c(50912, 50971, 50983, 50902, 50882, 51221, 51301, 50931)

dlabitems_tble <- tbl(con_bq, "d_labitems") |>
  filter(itemid %in% itemids) |>
  select(itemid, label) |>
  print(width = Inf)
```

```
# Source:   SQL [?? x 2]
# Database: BigQueryConnection
  itemid label
   <int> <chr>
1  50882 Bicarbonate
2  50902 Chloride
3  50912 Creatinine
4  50931 Glucose
5  50971 Potassium
6  50983 Sodium
7  51221 Hematocrit
8  51301 White Blood Cells
```

```
labevents_tble <- tbl(con_bq, "labevents") |>
  # only keep information needed from labevents
  select(subject_id, itemid, storetime, valuenum) |>
  # filter subjects who appear in `icustays_tble`
  semi_join(icustays_tble, by = "subject_id") |>
  # filter specific lab items
  filter(itemid %in% itemids) |>
  # merge with icustays to get stay info
  left_join(
    select(icustays_tble, subject_id, stay_id, intime),
    by = c("subject_id"),
    copy = TRUE
  ) |>
  # keep labevents before icustay
  filter(storetime < intime) |>
  # for each patient, icu stay, itemid combination
  group_by(subject_id, stay_id, itemid) |>
  # only keep the most recent labevent
  slice_max(storetime, n = 1) |>
  # discard storetime and intime
  select(-storetime, -intime) |>
  ungroup() |>
  # pivot lab items to become columns
  pivot_wider(names_from = itemid, values_from = valuenum) |>
  # change itemid to label
  rename_at(
    vars(as.character(pull(dlabitems_tble, itemid))),
    ~str_to_lower(pull(dlabitems_tble, label))
  ) |>
  # reorder columns
  select(
    subject_id, stay_id, str_to_lower(sort(pull(dlabitems_tble, label)))) |>
  rename(wbc = `white blood cells`) |>
  # sort table by `subject_id` and `stay_id`
  arrange(subject_id, stay_id) |>
  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?

# Source:     SQL [?? x 10]
# Database:   BigQueryConnection
# Ordered by: subject_id, stay_id
   subject_id  stay_id bicarbonate chloride creatinine glucose hematocrit
        <int>    <int>       <dbl>    <dbl>      <dbl>   <dbl>      <dbl>
 1   10000032 39553978          25       95        0.7     102       41.1
 2   10000690 37081114          26      100        1        85       36.1
 3   10000980 39765666          21      109        2.3      89       27.3
 4   10001217 34592300          30      104        0.5      87       37.4
 5   10001217 37067082          22      108        0.6     112       38.1
 6   10001725 31205490          NA       98       NA        NA        NA
 7   10001843 39698942          28       97        1.3     131       31.4
 8   10001884 37510196          30       88        1.1     141       39.7
 9   10002013 39060235          24      102        0.9     288       34.9
10   10002114 34672098          18       NA        3.1      95       34.3
   potassium sodium   wbc
       <dbl>  <dbl> <dbl>
 1       6.7    126   6.9
 2       4.8    137   7.1
 3       3.9    144   5.3
 4       4.1    142   5.4
 5       4.2    142  15.7
 6       4.1    139  NA
 7       3.9    138  10.4
 8       4.5    130  12.2
 9       3.5    137   7.2
10       6.5    125  16.8
# 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 (by storetime) during ICU stay and pivot chart events to become variables/columns. Write all steps in *one* chain of pipes. Similary to HW3, if a vital has multiple measurements at the first storetime , average them.

```r
itemids <- c(220045, 220179, 220180, 223761, 220210)

ditems_tbl <- tbl(con_bq, "d_items") |>
  filter(itemid %in% itemids) |>
  select(itemid, label) |>
  mutate(label = tolower(gsub(" ", "_", label))) |>
  print(width = Inf)
```

```
# Source:   SQL [?? x 2]
# Database: BigQueryConnection
  itemid label
   <int> <chr>
1 220210 respiratory_rate
2 220045 heart_rate
3 220179 non_invasive_blood_pressure_systolic
4 220180 non_invasive_blood_pressure_diastolic
5 223761 temperature_fahrenheit
```

```r
chartevents_tble <- tbl(con_bq, "chartevents") |>
  # only keep information needed for chartevents
  select(subject_id, itemid, storetime, valuenum) |>
  # filter itemid
  filter(itemid %in% itemids) |>
  # merge with icustays to get stay information
  left_join(
    select(icustays_tble, subject_id, stay_id, intime, outtime),
    by = c("subject_id"),
    copy = TRUE
  ) |>
  # only keep chartevents within the ICU stay
  filter(storetime >= intime & storetime <= outtime) |>
  # group by each patient, icu stay, itemid combination
  group_by(subject_id, stay_id, itemid, storetime) |>
  # get the average for measurement with same storetime
  mutate(valuenum = mean(valuenum, na.rm = TRUE)) |>
  ungroup() |>
  group_by(subject_id, stay_id, itemid) |>
  # only keep the first vital measurement
  slice_min(storetime, n = 1) |>
  # discard storetime, intime, and outtime
  select(-storetime, -intime, -outtime) |>
  ungroup() |>
  # pivot chart events to columns
  pivot_wider(names_from = itemid, values_from = valuenum) |>
  # rename columns name
  rename_at(
    vars(as.character(pull(ditems_tbl, itemid))),
    ~str_to_lower(pull(ditems_tbl, label))
  ) |>
  arrange(subject_id, stay_id) |>
  select(subject_id, stay_id,
         str_to_lower(sort(pull(ditems_tbl, label)))) |>
  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_diastolic
        <int>    <int>      <dbl>                                 <dbl>
 1   10000032 39553978       91                                    48  
 2   10000690 37081114       78                                    56.5
 3   10000980 39765666       76                                   102  
 4   10001217 34592300       79.3                                  93.3
 5   10001217 37067082       86                                    90  
 6   10001725 31205490       86                                    56  
 7   10001843 39698942      124.                                   78  
 8   10001884 37510196       49                                    30.5
 9   10002013 39060235       80                                    62  
10   10002114 34672098      110.                                   80  
   non_invasive_blood_pressure_systolic respiratory_rate temperature_fahrenheit
                                  <dbl>            <dbl>                  <dbl>
 1                                   84             24                     98.7
 2                                  106             24.3                   97.7
 3                                  154             23.5                   98  
 4                                  156             14                     97.6
 5                                  151             18                     98.5
 6                                   73             19                     97.7
 7                                  110             16.5                   97.9
 8                                174.            13                   98.1
 9                                 98.5           14                   97.2
10                                112             21                   97.9
# 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, (vi) sort `subject_id`, `hadm_id`, `stay_id` and `print(width = Inf)`.

```r
mimic_icu_cohort <- icustays_tble |>
  # merge in admissions and patients
  left_join(admissions_tble, by = c("subject_id", "hadm_id")) |>
  left_join(patients_tble, by = "subject_id") |>
  # keep adults only
  mutate(age_intime = year(intime) - (anchor_year - anchor_age)) |>
  filter(age_intime >= 18) |>
  # merge in labevents and chartevents
  left_join(labevents_tble, by = c("subject_id", "stay_id")) |>
  left_join(chartevents_tble, by = c("subject_id", "stay_id")) |>
  # collect the tibble
  collect() |>
  as_tibble() |>
  # sort `subject_id`, `hadm_id`, `stay_id`
  arrange(subject_id, hadm_id, stay_id) |>
  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?
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 × 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>                                            <dttm>             
 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           dischtime          
   <dttm>              <dbl> <dttm>              <dttm>             
 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
   <dttm>              <chr>                       <chr>            
 1 NA                  EW EMER.                    P06OTX           
 2 NA                  EW EMER.                    P26QQ4           
 3 NA                  EW EMER.                    P06OTX           
 4 NA                  EW EMER.                    P3610N           
 5 NA                  DIRECT EMER.                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
   <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>                  <dttm>              <dttm>
 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_intime bicarbonate chloride creatinine glucose hematocrit
   <date>          <int>       <dbl>    <dbl>      <dbl>   <dbl>      <dbl>
 1 2180-09-09         52          25       95        0.7     102       41.1
 2 2152-01-30         86          26      100        1        85       36.1
 3 2193-08-26         76          21      109        2.3      89       27.3
 4 NA                 55          22      108        0.6     112       38.1
 5 NA                 55          30      104        0.5      87       37.4
 6 NA                 46          NA       98       NA        NA       NA
 7 2134-12-06         76          28       97        1.3     131       31.4
 8 2131-01-20         77          30       88        1.1     141       39.7
 9 NA                 57          24      102        0.9     288       34.9
10 2162-12-11         56          18       NA        3.1      95       34.3
   potassium sodium   wbc heart_rate non_invasive_blood_pressure_diastolic
       <dbl>  <dbl> <dbl>      <dbl>                                 <dbl>
 1       6.7    126   6.9       91                                    48
 2       4.8    137   7.1       78                                    56.5
 3       3.9    144   5.3       76                                   102
 4       4.2    142  15.7       86                                    90
 5       4.1    142   5.4       79.3                                  93.3
 6       4.1    139  NA         86                                    56
 7       3.9    138  10.4      124.                                   78
 8       4.5    130  12.2       49                                    30.5
 9       3.5    137   7.2       80                                    62
10       6.5    125  16.8      110.                                   80
   non_invasive_blood_pressure_systolic respiratory_rate temperature_fahrenheit
                                  <dbl>            <dbl>                  <dbl>
 1                                 84               24                     98.7
 2                                106               24.3                   97.7
 3                                154               23.5                   98
 4                                151               18                     98.5
 5                                156               14                     97.6
 6                                 73               19                     97.7
 7                                110               16.5                   97.9
 8                                174.              13                     98.1
 9                                 98.5             14                     97.2
10                                112               21                     97.9
# ℹ 94,448 more rows
```

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:

Characteristic	TRUE N = 46,337[1]	FALSE N = 48,107[1]
first_careunit		
Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)	7,416 (15%)
Medical Intensive Care Unit (MICU)	9,837 (21%)	10,862 (23%)
Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)	8,780 (18%)
Surgical Intensive Care Unit (SICU)	6,434 (14%)	6,574 (14%)
Other	16,046 (35%)	14,475 (30%)
last_careunit		
Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)	7,416 (15%)
Medical Intensive Care Unit (MICU)	9,837 (21%)	10,862 (23%)
Medical/Surgical Intensive Care Unit		

```r
mimic_icu_cohort <- mimic_icu_cohort |>
  # lump infrequent levels into "Other"
  mutate(first_careunit = fct_lump_n(first_careunit, n = 4),
         last_careunit = fct_lump_n(last_careunit, n = 4),
         admission_type = fct_lump_n(admission_type, n = 4),
         admission_location = fct_lump_n(admission_location, n = 3),
         discharge_location = fct_lump_n(discharge_location, n = 4)) |>
  # collapse levels of `race`
  mutate(
    race = fct_collapse(
      race,
      ASIAN = c("ASIAN", "ASIAN - SOUTH EAST ASIAN", "ASIAN - CHINESE",
                "ASIAN - KOREAN", "ASIAN - ASIAN INDIAN"),
      BLACK = c("BLACK/AFRICAN AMERICAN", "BLACK/AFRICAN",
                  "BLACK/CAPE VERDEAN", "BLACK/CARIBBEAN ISLAND"),
      HISPANIC = c("HISPANIC OR LATINO", "HISPANIC/LATINO - SALVADORAN",
                     "HISPANIC/LATINO - PUERTO RICAN",
                     "HISPANIC/LATINO - GUATEMALAN",
                     "HISPANIC/LATINO - CUBAN", "HISPANIC/LATINO - DOMINICAN",
                     "HISPANIC/LATINO - CENTRAL AMERICAN",
                     "HISPANIC/LATINO - HONDURAN",
                     "HISPANIC/LATINO - COLUMBIAN", "HISPANIC/LATINO - MEXICAN"),
      WHITE = c("WHITE", "WHITE - RUSSIAN", "WHITE - OTHER EUROPEAN",
                "WHITE - BRAZILIAN", "WHITE - EASTERN EUROPEAN"),
      Other = c("OTHER", "UNKNOWN", "UNABLE TO OBTAIN",
                "PATIENT DECLINED TO ANSWER", "AMERICAN INDIAN/ALASKA NATIVE",
                "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER",
                "MULTIPLE RACE/ETHNICITY", "PORTUGUESE", "SOUTH AMERICAN")
      ),
    race = factor(race, levels = c("ASIAN", "BLACK", "HISPANIC",
                                   "WHITE", "Other"))) |>
  # create `los_long` that is `TRUE` when `los` >= 2
  mutate(los_long = los >= 2,
         los_long = factor(los_long, levels = c(TRUE, FALSE)))


# summarize data, stratified by `los_long`
mimic_icu_cohort |>
  tbl_summary(
    by = los_long,
    include = c(-subject_id, -hadm_id, -stay_id, -intime, -outtime,
                -admittime, -dischtime, -deathtime, -admit_provider_id,
                -edregtime, -edouttime, -anchor_age, -anchor_year,
                -anchor_year_group))
```

```
14 missing rows in the "los_long" column have been removed.
The following errors were returned during `tbl_summary()`:
✖ For variable `dod` (`los_long = "FALSE"`) and "p75" statistic: * not defined
  for "Date" objects
```

Characteristic	**TRUE** N = 46,337[1]	**FALSE** N = 48,107[1]
first_careunit		
Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)	7,416 (15%)
Medical Intensive Care Unit (MICU)	9,837 (21%)	10,862 (23%)
Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)	8,780 (18%)
Surgical Intensive Care Unit (SICU)	6,434 (14%)	6,574 (14%)
Other	16,046 (35%)	14,475 (30%)
last_careunit		

[1] n (%); Median (Q1, Q3)

Characteristic	**TRUE** N = 46,337[1]	**FALSE** N = 48,107[1]
Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)	7,416 (15%)
Medical Intensive Care Unit (MICU)	9,837 (21%)	10,862 (23%)
Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)	8,780 (18%)
Surgical Intensive Care Unit (SICU)	6,434 (14%)	6,574 (14%)
Other	16,046 (35%)	14,475 (30%)
los	3.9 (2.7, 6.8)	1.1 (0.8, 1.5)
admission_type		
EW EMER.	23,012 (50%)	25,337 (53%)
OBSERVATION ADMIT	7,393 (16%)	6,638 (14%)
SURGICAL SAME DAY ADMISSION	4,001 (8.6%)	5,543 (12%)
URGENT	8,691 (19%)	6,683 (14%)
Other	3,240 (7.0%)	3,906 (8.1%)
admission_location		
EMERGENCY ROOM	17,058 (37%)	20,443 (42%)
PHYSICIAN REFERRAL	11,013 (24%)	12,684 (26%)
TRANSFER FROM HOSPITAL	13,904 (30%)	10,400 (22%)
Other	4,362 (9.4%)	4,580 (9.5%)
discharge_location		
DIED	6,884 (15%)	4,436 (9.4%)
HOME	6,879 (15%)	15,210 (32%)
HOME HEALTH CARE	10,620 (23%)	13,422 (28%)
SKILLED NURSING FACILITY	8,785 (19%)	7,489 (16%)

[1] n (%); Median (Q1, Q3)

Characteristic	TRUE N = 46,337[1]	FALSE N = 48,107[1]
Other	13,092 (28%)	6,779 (14%)
Unknown	77	771
insurance		
Medicaid	6,768 (15%)	7,469 (16%)
Medicare	26,330 (58%)	25,485 (54%)
No charge	5 (<0.1%)	3 (<0.1%)
Other	1,091 (2.4%)	1,237 (2.6%)
Private	11,515 (25%)	13,018 (28%)
Unknown	628	895
language		
American Sign Language	29 (<0.1%)	34 (<0.1%)
Amharic	14 (<0.1%)	9 (<0.1%)
Arabic	87 (0.2%)	62 (0.1%)
Armenian	12 (<0.1%)	13 (<0.1%)
Bengali	22 (<0.1%)	12 (<0.1%)
Chinese	550 (1.2%)	611 (1.3%)
English	41,563 (90%)	43,483 (91%)
French	18 (<0.1%)	14 (<0.1%)
Haitian	375 (0.8%)	252 (0.5%)
Hindi	24 (<0.1%)	21 (<0.1%)
Italian	101 (0.2%)	107 (0.2%)
Japanese	5 (<0.1%)	7 (<0.1%)
Kabuverdianu	301 (0.7%)	345 (0.7%)

[1] n (%); Median (Q1, Q3)

Characteristic	TRUE N = 46,337[1]	FALSE N = 48,107[1]
Khmer	50 (0.1%)	37 (<0.1%)
Korean	40 (<0.1%)	32 (<0.1%)
Modern Greek (1453-)	102 (0.2%)	88 (0.2%)
Other	152 (0.3%)	153 (0.3%)
Persian	42 (<0.1%)	35 (<0.1%)
Polish	36 (<0.1%)	38 (<0.1%)
Portuguese	351 (0.8%)	314 (0.7%)
Russian	601 (1.3%)	659 (1.4%)
Somali	8 (<0.1%)	15 (<0.1%)
Spanish	1,472 (3.2%)	1,429 (3.0%)
Thai	21 (<0.1%)	22 (<0.1%)
Vietnamese	151 (0.3%)	129 (0.3%)
Unknown	210	186
marital_status		
DIVORCED	3,377 (8.0%)	3,555 (8.0%)
MARRIED	20,557 (49%)	21,344 (48%)
SINGLE	12,745 (30%)	14,039 (31%)
WIDOWED	5,319 (13%)	5,752 (13%)
Unknown	4,339	3,417
race		
ASIAN	1,369 (3.0%)	1,516 (3.2%)
BLACK	4,933 (11%)	5,452 (11%)
HISPANIC	1,687 (3.6%)	1,908 (4.0%)

[1] n (%); Median (Q1, Q3)

Characteristic	**TRUE** N = 46,337[1]	**FALSE** N = 48,107[1]
WHITE	30,312 (65%)	32,351 (67%)
Other	8,036 (17%)	6,880 (14%)
hospital_expire_flag	6,831 (15%)	4,512 (9.4%)
gender		
F	20,106 (43%)	21,471 (45%)
M	26,231 (57%)	26,636 (55%)
dod	2155-09-06 (2135-07-16, 2175-10-08)	2155-12-18 (2136-04-26, NA)
Unknown	25,846	30,639
age_intime	67 (56, 77)	66 (54, 77)
bicarbonate	24.0 (21.0, 27.0)	24.0 (21.0, 27.0)
Unknown	6,272	5,277
chloride	102 (98, 105)	102 (98, 105)
Unknown	6,184	5,167
creatinine	1.00 (0.80, 1.60)	1.00 (0.80, 1.40)
Unknown	4,541	3,486
glucose	122 (100, 159)	118 (98, 154)
Unknown	6,340	5,314
hematocrit	35 (29, 40)	36 (30, 41)
Unknown	3,857	2,894
potassium	4.20 (3.90, 4.70)	4.20 (3.90, 4.60)
Unknown	6,200	5,187
sodium	138.0 (135.0, 141.0)	139.0 (136.0, 141.0)

[1] n (%); Median (Q1, Q3)

Characteristic	TRUE N = 46,337[1]	FALSE N = 48,107[1]
Unknown	6,167	5,163
wbc	9.7 (7.0, 13.8)	9.0 (6.6, 12.6)
Unknown	3,906	2,944
heart_rate	87 (75, 102)	84 (73, 99)
Unknown	1	84
non_invasive_blood_pressure_diastolic	67 (57, 79)	68 (58, 80)
Unknown	350	1,015
non_invasive_blood_pressure_systolic	120 (104, 137)	122 (107, 138)
Unknown	347	1,013
respiratory_rate	19.0 (16.0, 23.0)	18.0 (15.0, 22.0)
Unknown	14	181
temperature_fahrenheit	98.20 (97.70, 98.80)	98.10 (97.60, 98.60)
Unknown	230	1,386

[1] n (%); Median (Q1, Q3)

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")
}
# save the final tibble
mimic_icu_cohort |>
  write_rds("mimiciv_shiny/mimic_icu_cohort.rds", compress = "gz")
```

Close database connection and clear workspace.

```r
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.

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 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.