Biostat 203B Homework 4

Due Mar 9 @ 11:59PM

Amaan Jogia-Sattar, 206324648

Display machine information:

```
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: aarch64-apple-darwin20
Running under: macOS Sequoia 15.3.1

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  L

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: America/Los_Angeles
tzcode source: internal

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

loaded via a namespace (and not attached):
 [1] compiler_4.4.2    fastmap_1.2.0     cli_3.6.3         tools_4.4.2
 [5] htmltools_0.5.8.1 rstudioapi_0.17.1 yaml_2.3.10       rmarkdown_2.28
 [9] knitr_1.48        jsonlite_1.8.9    xfun_0.48         digest_0.6.37
[13] rlang_1.1.4       evaluate_1.0.1
```

Display my machine memory.

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

```
Totalram:   36.000 GiB
Freeram:   221.766 MiB
```

Load database libraries and the tidyverse frontend:

```
library(bigrquery)
library(dbplyr)
library(DBI)
library(gt)
library(gtsummary)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::ident()  masks dbplyr::ident()
x dplyr::lag()    masks stats::lag()
x dplyr::sql()    masks dbplyr::sql()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

```
library(forcats)
```

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) |>
  # show_query() |>
  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.

```
patients_tble <- tbl(con_bq,
                     "patients") |>
  arrange(subject_id) |>
```

```
  # show_query() |>
  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.

```
lab_items <- c(50912,
               50971,
               50983,
               50902,
               50882,
               51221,
               51301,
               50931)
labevents_tble <- tbl(con_bq,
                      "labevents") |>
  select(subject_id,
         itemid,
         storetime,
         valuenum) |>
```

```
  filter(itemid %in% lab_items) |>
  left_join(
    select(icustays_tble,
           subject_id,
           stay_id,
           intime),
    by = c("subject_id"),
    copy = TRUE
  ) |>
  filter(storetime < intime) |>
  group_by(subject_id,
           stay_id,
           itemid) |>
  slice_max(storetime,
            n = 1) |>
  select(-storetime,
         -intime) |>
  ungroup() |>
  pivot_wider(names_from = itemid,
              values_from = valuenum) |>
  rename(
    creatinine = `50912`,
    potassium = `50971`,
    sodium = `50983`,
    chloride = `50902`,
    bicarbonate = `50882`,
    hematocrit = `51221`,
    wbc_count = `51301`,
    glucose = `50931`
  ) |>
  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?

# Source:     SQL [?? x 10]
# Database:   BigQueryConnection
# Ordered by: subject_id, stay_id
```

```
   subject_id  stay_id glucose potassium sodium chloride creatinine wbc_count
        <int>    <int>   <dbl>     <dbl>  <dbl>    <dbl>      <dbl>     <dbl>
 1   10000032 39553978     102       6.7    126       95        0.7       6.9
 2   10000690 37081114      85       4.8    137      100        1         7.1
 3   10000980 39765666      89       3.9    144      109        2.3       5.3
 4   10001217 34592300      87       4.1    142      104        0.5       5.4
 5   10001217 37067082     112       4.2    142      108        0.6      15.7
 6   10001725 31205490      NA       4.1    139       98       NA        NA
 7   10001843 39698942     131       3.9    138       97        1.3      10.4
 8   10001884 37510196     141       4.5    130       88        1.1      12.2
 9   10002013 39060235     288       3.5    137      102        0.9       7.2
10   10002114 34672098      95       6.5    125       NA        3.1      16.8
   bicarbonate hematocrit
         <dbl>      <dbl>
 1          25       41.1
 2          26       36.1
 3          21       27.3
 4          30       37.4
 5          22       38.1
 6          NA       NA
 7          28       31.4
 8          30       39.7
 9          24       34.9
10          18       34.3
# i more rows
```

We can verify the number of rows in our table using the tally function:

```
labevents_tble %>% tally()
```

```
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 [1 x 1]
# Database:   BigQueryConnection
# Ordered by: subject_id, stay_id
      n
  <int>
1 88086
```

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.

```
chartevents_tble <-
  tbl(con_bq,
      "chartevents") |>
  filter(itemid %in% c(220045,
                       220179,
                       220180,
                       223761,
                       220210)) |>
  inner_join(tbl(con_bq,
                 "icustays"),
             by = c("subject_id",
                    "hadm_id",
                    "stay_id")) |>
  filter(storetime >= intime,
         storetime < outtime) |>
  group_by(subject_id,
           stay_id,
           itemid) |>
  slice_min(order_by = storetime,
            n = 1) |>
  ungroup() |>
  select(subject_id,
         stay_id,
         hadm_id,
         itemid,
         valuenum) |>
  group_by(subject_id,
           stay_id,
           hadm_id,
           itemid) |>
  summarise(valuenum = mean(valuenum,
                            na.rm = TRUE)) |>
  ungroup() |>
  pivot_wider(names_from = itemid,
              values_from = valuenum) |>
```

```
  rename(
    `Heart Rate` = `220045`,
    `Noninvasive BP Systolic` = `220179`,
    `Noninvasive BP Diastolic` = `220180`,
    `Temperature_F` = `223761`,
    `Respiratory Rate` = `220210`
  ) |>
  arrange(subject_id) |>
  print(width = Inf)
```

```
`summarise()` has grouped output by "subject_id", "stay_id", and "hadm_id". You
can override using the `.groups` argument.
`summarise()` has grouped output by "subject_id", "stay_id", and "hadm_id". You
can override using the `.groups` argument.
```

```
# Source:     SQL [?? x 8]
# Database:   BigQueryConnection
# Ordered by: subject_id
   subject_id  stay_id  hadm_id `Noninvasive BP Diastolic` `Respiratory Rate`
        <int>    <int>    <int>                     <dbl>              <dbl>
 1   10000032 39553978 29079034                      48                 24
 2   10000690 37081114 25860671                      56.5               24.3
 3   10000980 39765666 26913865                     102                 23.5
 4   10001217 37067082 24597018                      90                 18
 5   10001217 34592300 27703517                      93.3               14
 6   10001725 31205490 25563031                      56                 19
 7   10001843 39698942 26133978                      78                 16.5
 8   10001884 37510196 26184834                      30.5               13
 9   10002013 39060235 23581541                      62                 14
10   10002114 34672098 27793700                      80                 21
   `Noninvasive BP Systolic` `Heart Rate` Temperature_F
                       <dbl>        <dbl>         <dbl>
 1                        84           91          98.7
 2                       106           78          97.7
 3                       154           76          98
 4                       151           86          98.5
 5                       156           79.3        97.6
 6                        73           86          97.7
 7                       110          124.         97.9
 8                       174.          49          98.1
 9                        98.5         80          97.2
```

```
10                    112          110.           97.9
# i more rows
```

We can verify the number of rows in our table using the tally function:

```
chartevents_tble %>% tally()
```

```
`summarise()` has grouped output by "subject_id", "stay_id", and "hadm_id". You
can override using the `.groups` argument.
```

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

```
# Source:     SQL [1 x 1]
# Database:   BigQueryConnection
# Ordered by: subject_id
      n
  <int>
1 94363
```

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)`.

```
# # TODO
mimic_icu_cohort <-
  icustays_tble %>%
  left_join(admissions_tble,
            by = c("subject_id",
                   "hadm_id")) %>%
  left_join(patients_tble,
            by = "subject_id") %>%
  left_join(labevents_tble,
            by = c("subject_id",
                   "stay_id")) %>%
  left_join(chartevents_tble,
```

```
            by = c("subject_id",
                   "hadm_id",
                   "stay_id")) %>%
  mutate(age_intime = year(intime) - anchor_year + anchor_age) %>%
  filter(age_intime >= 18) %>%
  arrange(subject_id,
          hadm_id,
          stay_id) %>%
  collect() %>%
  print(width = Inf)
```

```
`summarise()` has grouped output by "subject_id", "stay_id", and "hadm_id". You
can override using the `.groups` argument.
```

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

```
# A tibble: 94,458 x 41
   subject_id  hadm_id  stay_id first_careunit
        <int>    <int>    <int> <chr>
 1   10000032 29079034 39553978 Medical Intensive Care Unit (MICU)
 2   10000690 25860671 37081114 Medical Intensive Care Unit (MICU)
 3   10000980 26913865 39765666 Medical Intensive Care Unit (MICU)
 4   10001217 24597018 37067082 Surgical Intensive Care Unit (SICU)
 5   10001217 27703517 34592300 Surgical Intensive Care Unit (SICU)
 6   10001725 25563031 31205490 Medical/Surgical Intensive Care Unit (MICU/SICU)
 7   10001843 26133978 39698942 Medical/Surgical Intensive Care Unit (MICU/SICU)
 8   10001884 26184834 37510196 Medical Intensive Care Unit (MICU)
 9   10002013 23581541 39060235 Cardiac Vascular Intensive Care Unit (CVICU)
10   10002114 27793700 34672098 Coronary Care Unit (CCU)
   last_careunit                                       intime
```

```
   <chr>                                          <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        glucose potassium sodium chloride creatinine wbc_count bicarbonate
   <date>       <dbl>     <dbl>  <dbl>    <dbl>      <dbl>     <dbl>       <dbl>
 1 2180-09-09     102       6.7    126       95        0.7       6.9          25
 2 2152-01-30      85       4.8    137      100        1         7.1          26
 3 2193-08-26      89       3.9    144      109        2.3       5.3          21
 4 NA             112       4.2    142      108        0.6      15.7          22
 5 NA              87       4.1    142      104        0.5       5.4          30
 6 NA              NA       4.1    139       98       NA        NA            NA
 7 2134-12-06     131       3.9    138       97        1.3      10.4          28
 8 2131-01-20     141       4.5    130       88        1.1      12.2          30
 9 NA             288       3.5    137      102        0.9       7.2          24
10 2162-12-11      95       6.5    125       NA        3.1      16.8          18
   hematocrit `Noninvasive BP Diastolic` `Respiratory Rate`
        <dbl>                      <dbl>              <dbl>
 1       41.1                         48                 24
```

```
 2       36.1                        56.5               24.3
 3       27.3                       102                 23.5
 4       38.1                        90                 18
 5       37.4                        93.3               14
 6       NA                          56                 19
 7       31.4                        78                 16.5
 8       39.7                        30.5               13
 9       34.9                        62                 14
10       34.3                        80                 21
   `Noninvasive BP Systolic` `Heart Rate` Temperature_F age_intime
                       <dbl>        <dbl>         <dbl>      <int>
 1                      84           91            98.7         52
 2                     106           78            97.7         86
 3                     154           76            98           76
 4                     151           86            98.5         55
 5                     156           79.3          97.6         55
 6                      73           86            97.7         46
 7                     110          124.           97.9         76
 8                     174.          49            98.1         77
 9                      98.5         80            97.2         57
10                     112          110.           97.9         56
# 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.

Solution: We proceed to preprocess the data. We examine Dr. Zhou's numerical summary of his preprocessing tibble, as well as frequency tables for our categorical variables. We decide to lump infrequent levels into "Other" such that we end up with a maximum of 5 levels for each categorical variable.

```
mimic_icu_cohort_preprocessed <- mimic_icu_cohort %>%
  mutate(
    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")
  ) %>%
  # Ensure race is a factor so we can work with its levels
  mutate(race = factor(race)) %>%
  {
    # Capture the current levels of race
    race_levels <- levels(.$race)
    mutate(., race = fct_collapse(race,
      ASIAN    = race_levels[grep("ASIAN",
                                  race_levels)],
      BLACK    = race_levels[grep("BLACK",
                                  race_levels)],
      HISPANIC = race_levels[grep("HISPANIC",
                                  race_levels)],
      WHITE    = race_levels[grep("WHITE",
                                  race_levels)],
      OTHER    = setdiff(race_levels,
                         c(race_levels[grep("ASIAN",
                                            race_levels)],
                           race_levels[grep("BLACK",
                                            race_levels)],
                           race_levels[grep("HISPANIC",
                                            race_levels)],
                           race_levels[grep("WHITE",
                                            race_levels)]))
    ))
  }
mimic_icu_cohort_preprocessed <- mimic_icu_cohort_preprocessed %>%
  mutate(los_long = los >= 2)

mimic_icu_cohort_preprocessed <- mimic_icu_cohort_preprocessed %>%
```

```
  select(
    first_careunit,
    last_careunit,
    los_long,
    admission_type,
    admission_location,
    discharge_location,
    insurance,
    language,
    marital_status,
    race,
    hospital_expire_flag,
    gender,
    dod,
    chloride,
    creatinine,
    sodium,
    potassium,
    glucose,
    hematocrit,
    wbc_count,
    bicarbonate,
    `Noninvasive BP Systolic`,
    `Noninvasive BP Diastolic`,
    `Respiratory Rate`,
    `Temperature_F`,
    `Heart Rate`,
    age_intime
  ) %>%
  tbl_summary(by = los_long)
```

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

```
mimic_icu_cohort_preprocessed
```

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

Characteristic	**TRUE** N = 46,337[1]	**F**
first_careunit		
Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)	
Medical Intensive Care Unit (MICU)	9,837 (21%)	
Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)	
Surgical Intensive Care Unit (SICU)	6,434 (14%)	
Other	16,046 (35%)	
last_careunit		
Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)	
Medical Intensive Care Unit (MICU)	9,837 (21%)	
Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)	
Surgical Intensive Care Unit (SICU)	6,434 (14%)	
Other	16,046 (35%)	
admission_type		
EW EMER.	23,012 (50%)	
OBSERVATION ADMIT	7,393 (16%)	
SURGICAL SAME DAY ADMISSION	4,001 (8.6%)	
URGENT	8,691 (19%)	
Other	3,240 (7.0%)	
admission_location		
EMERGENCY ROOM	17,058 (37%)	
PHYSICIAN REFERRAL	11,013 (24%)	
TRANSFER FROM HOSPITAL	13,904 (30%)	
Other	4,362 (9.4%)	
discharge_location		
DIED	6,884 (15%)	
HOME	6,879 (15%)	
HOME HEALTH CARE	10,620 (23%)	
SKILLED NURSING FACILITY	8,785 (19%)	
Other	13,092 (28%)	
Unknown	77	
insurance		
Medicaid	6,768 (15%)	
Medicare	26,330 (58%)	
No charge	5 (<0.1%)	
Other	1,091 (2.4%)	
Private	11,515 (25%)	
Unknown	628	
language		
American Sign Language	29 (<0.1%)	
Amharic	14 (<0.1%)	
Arabic	87 (0.2%)	
Armenian	12 (<0.1%)	
Bengali	22 (<0.1%)	
Chinese	550 (1.2%)	
English	41,563 (90%)	
French	18 (<0.1%)	
Haitian	375 (0.8%)	
Hindi	24 (<0.1%)	
Italian	101 (0.2%)	

Q1.9 Save the final tibble

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

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

Close database connection and clear workspace.

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

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

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.

Our Shiny app is located in the 'mimiciv_shiny folder'.