Biostat 203B Homework 4

Due Mar 9 @ 11:59PM

Jiaye Tian UID: 306541095

Table of contents

Display machine information:

```
sessionInfo()
```

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

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

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

time zone: America/Los_Angeles
```

```
tzcode source: internal

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

loaded via a namespace (and not attached):
 [1] compiler_4.4.2    fastmap_1.2.0     cli_3.6.3         tools_4.4.2
 [5] htmltools_0.5.8.1 rstudioapi_0.17.1 yaml_2.3.10       rmarkdown_2.29
 [9] knitr_1.49        jsonlite_1.8.9    xfun_0.50         digest_0.6.37
[13] rlang_1.1.5       evaluate_1.0.3
```

Display my machine memory.

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

```
Totalram:  16.000 GiB
Freeram:   63.812 MiB
```

Load database libraries and the tidyverse frontend:

```
library(bigrquery)
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(dbplyr)
```

```

Attaching package: 'dbplyr'
```

```
The following objects are masked from 'package:dplyr':

    ident, sql
```

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

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v readr     2.1.5
v ggplot2   3.5.1     v stringr   1.5.1
v lubridate 1.9.4     v tibble    3.2.1
v purrr     1.0.4     v tidyr     1.3.1
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dbplyr::ident() masks dplyr::ident()
x dplyr::lag()    masks stats::lag()
x dbplyr::sql()   masks dplyr::sql()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

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.

```
# # TODO
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.           P060TX            EMERGENCY ROOM         HOME
 5 EU OBSERVATION     P39NWO            EMERGENCY ROOM         <NA>
 6 EW EMER.           P42H7G            WALK-IN/SELF REFERRAL  HOME HEALTH CARE
 7 EU OBSERVATION     P35NE4            PHYSICIAN REFERRAL     <NA>
 8 EU OBSERVATION     P40JML            EMERGENCY ROOM         <NA>
 9 EU OBSERVATION     P47EY8            EMERGENCY ROOM         <NA>
10 OBSERVATION ADMIT P13ACE            WALK-IN/SELF REFERRAL  HOME HEALTH CARE
   insurance language marital_status race  edregtime
   <chr>     <chr>    <chr>          <chr> <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.

```
# # TODO
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.

```
# # TODO
labevents_tble <- tbl(con_bq, "labevents") %>%
  select(subject_id, itemid, storetime, valuenum) %>%
  filter(itemid %in% c(50882, 50902, 50912, 50931,
                       50971, 50983, 51221, 51301)) %>%
  left_join(icustays_tble, by = "subject_id") %>%
  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(
    bicarbonate           = `50882`,
    chloride              = `50902`,
    creatinine            = `50912`,
    glucose               = `50931`,
    potassium             = `50971`,
    sodium                = `50983`,
    hematocrit            = `51221`,
    `white blood cells`   = `51301`
  ) %>%
  rename(wbc = `white blood cells`) %>%
  arrange(subject_id, stay_id) %>%
  select(
    subject_id,
    stay_id,
    bicarbonate,
    chloride,
    creatinine,
    glucose,
    hematocrit,
    intime,
    potassium,
    sodium,
    wbc
  ) %>%
  show_query() %>%
  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?
```

```
<SQL>
SELECT
  `subject_id`,
  `stay_id`,
  MAX(IF(`itemid` = 50882, `valuenum`, NULL)) AS `bicarbonate`,
  MAX(IF(`itemid` = 50902, `valuenum`, NULL)) AS `chloride`,
  MAX(IF(`itemid` = 50912, `valuenum`, NULL)) AS `creatinine`,
  MAX(IF(`itemid` = 50931, `valuenum`, NULL)) AS `glucose`,
  MAX(IF(`itemid` = 51221, `valuenum`, NULL)) AS `hematocrit`,
  `intime`,
  MAX(IF(`itemid` = 50971, `valuenum`, NULL)) AS `potassium`,
  MAX(IF(`itemid` = 50983, `valuenum`, NULL)) AS `sodium`,
  MAX(IF(`itemid` = 51301, `valuenum`, NULL)) AS `wbc`
FROM (
  SELECT
    `subject_id`,
    `itemid`,
    `valuenum`,
    `hadm_id`,
    `stay_id`,
    `first_careunit`,
    `last_careunit`,
    `intime`,
    `outtime`,
    `los`
  FROM (
    SELECT
      `q01`.*,
      RANK() OVER (PARTITION BY `subject_id`, `stay_id`, `itemid` ORDER BY `storetime` DESC)
    FROM (
      SELECT
        `LHS`.*,
        `hadm_id`,
        `stay_id`,
        `first_careunit`,
        `last_careunit`,
        `intime`,
        `outtime`,
        `los`
      FROM (
        SELECT `subject_id`, `itemid`, `storetime`, `valuenum`
        FROM `labevents`
        WHERE (`itemid` IN (50882.0, 50902.0, 50912.0, 50931.0, 50971.0, 50983.0, 51221.0, 5
      ) `LHS`
      LEFT JOIN (
        SELECT `icustays`.*
        FROM `icustays`
      ) `RHS`
        ON (`LHS`.`subject_id` = `RHS`.`subject_id`)
    ) `q01`
    WHERE (`storetime` < `intime`)
  ) `q01`
  WHERE (`col01` <= 1)
) `q01`
GROUP BY
  `subject_id`,
  `hadm_id`,
  `stay_id`,
  `first_careunit`,
  `last_careunit`,
  `intime`,
  `outtime`,
  `los`
ORDER BY `subject_id`, `stay_id`

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

```
# Source:     SQL [?? x 11]
# 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
   intime              potassium sodium   wbc
   <dttm>                  <dbl>  <dbl> <dbl>
 1 2180-07-23 14:00:00       6.7    126   6.9
 2 2150-11-02 19:37:00       4.8    137   7.1
 3 2189-06-27 08:42:00       3.9    144   5.3
 4 2157-12-19 15:42:24       4.1    142   5.4
 5 2157-11-20 19:18:02       4.2    142  15.7
 6 2110-04-11 15:52:22       4.1    139  NA
 7 2134-12-05 18:50:03       3.9    138  10.4
 8 2131-01-11 04:20:05       4.5    130  12.2
 9 2160-05-18 10:00:53       3.5    137   7.2
10 2162-02-17 23:30:00       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
# # TODO
chartevents_tble <- tbl(con_bq, "chartevents") %>%
  select(subject_id, stay_id, itemid, storetime, value) %>%
  mutate(
    value = as.numeric(value)) %>%
  semi_join(
    tbl(con_bq, "d_items") %>%
      filter(itemid %in% c(220045, 220179, 220180, 223761, 220210)) %>%
      mutate(itemid = as.integer(itemid)) %>%
      select(itemid, label),
    by = "itemid"
  ) %>%
  left_join(
    icustays_tble %>% select(subject_id, stay_id, intime, outtime),
    by = c("subject_id", "stay_id")
  ) %>%
  filter(storetime >= intime, storetime <= outtime) %>%
  group_by(subject_id, stay_id, itemid) %>%
  slice_min(order_by = storetime, with_ties = TRUE) %>%
  select(-storetime, -intime, -outtime) %>%
  ungroup() %>%
  pivot_wider(names_from = itemid, values_from = value, values_fn = mean) %>%
  rename(
    `heart rate` = `220045`,
    `non invasive blood pressure systolic` = `220179`,
    `non invasive blood pressure diastolic` = `220180`,
    `respiratory rate` = `223761`,
    `temperature fahrenheit` = `220210`
  ) %>%
  arrange(subject_id, stay_id) %>%
  select(
    subject_id,
    stay_id,
    `heart rate`,
    `non invasive blood pressure systolic`,
    `non invasive blood pressure diastolic`,
    `respiratory rate`,
    `temperature fahrenheit`
  ) %>%
  mutate(
  `heart rate` = round(`heart rate`, 1),
  `non invasive blood pressure systolic` = round(`non invasive blood pressure systolic`, 1),
  `non invasive blood pressure diastolic` = round(`non invasive blood pressure diastolic`, 1
  `respiratory rate` = round(`respiratory rate`, 1)) %>%
  show_query() %>%
  print(width = Inf)
```

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

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

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

```
<SQL>
SELECT
  `subject_id`,
  `stay_id`,
  ROUND(`heart rate`, 1) AS `heart rate`,
  ROUND(`non invasive blood pressure systolic`, 1) AS `non invasive blood pressure systolic`
  ROUND(`non invasive blood pressure diastolic`, 1) AS `non invasive blood pressure diastoli
  ROUND(`respiratory rate`, 1) AS `respiratory rate`,
  `temperature fahrenheit`
FROM (
  SELECT
    `subject_id`,
    `stay_id`,
    AVG(IF(`itemid` = 220045, `value`, NULL)) AS `heart rate`,
    AVG(IF(`itemid` = 220179, `value`, NULL)) AS `non invasive blood pressure systolic`,
    AVG(IF(`itemid` = 220180, `value`, NULL)) AS `non invasive blood pressure diastolic`,
    AVG(IF(`itemid` = 223761, `value`, NULL)) AS `respiratory rate`,
    AVG(IF(`itemid` = 220210, `value`, NULL)) AS `temperature fahrenheit`
  FROM (
    SELECT `subject_id`, `stay_id`, `itemid`, `value`
    FROM (
      SELECT
        `q01`.*,
        RANK() OVER (PARTITION BY `subject_id`, `stay_id`, `itemid` ORDER BY `storetime`) AS
      FROM (
        SELECT `LHS`.*, `intime`, `outtime`
        FROM (
          SELECT `LHS`.*
          FROM (
            SELECT
```

```
              `subject_id`,
              `stay_id`,
              `itemid`,
              `storetime`,
              SAFE_CAST(`value` AS FLOAT64) AS `value`
            FROM `chartevents`
) `LHS`
          WHERE EXISTS (
            SELECT 1 FROM (
            SELECT SAFE_CAST(`itemid` AS INT64) AS `itemid`, `label`
            FROM `d_items`
            WHERE (`itemid` IN (220045.0, 220179.0, 220180.0, 223761.0, 220210.0))
) `RHS`
            WHERE (`LHS`.`itemid` = `RHS`.`itemid`)
          )
        ) `LHS`
        LEFT JOIN (
          SELECT `subject_id`, `stay_id`, `intime`, `outtime`
          FROM `icustays`
        ) `RHS`
          ON (
            `LHS`.`subject_id` = `RHS`.`subject_id` AND
            `LHS`.`stay_id` = `RHS`.`stay_id`
          )
      ) `q01`
      WHERE (`storetime` >= `intime`) AND (`storetime` <= `outtime`)
    ) `q01`
    WHERE (`col01` <= 1)
  ) `q01`
  GROUP BY `subject_id`, `stay_id`
) `q01`
```

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

```
# Source:     SQL [?? x 7]
# Database:   BigQueryConnection
# Ordered by: subject_id, stay_id
   subject_id  stay_id `heart rate` `non invasive blood pressure systolic`
        <int>    <int>        <dbl>                                  <dbl>
```

```
 1   10023994 37135700         96                                     108.
 2   10056539 31185929         81.3                                   117.
 3   10106899 30008792         87                                     107
 4   10181514 33674470         96.5                                   126.
 5   10209126 39390511         98                                     113
 6   10259372 35622225         82                                     102
 7   10290183 39061542         88                                     100
 8   10295020 36479916         67                                      94
 9   10303799 38159761         73                                     108
10   10345936 35615744         76                                     112
   `non invasive blood pressure diastolic` `respiratory rate`
                                     <dbl>              <dbl>
 1                                    71                 98.3
 2                                    67.3               98.2
 3                                    43                 97
 4                                    85.5               98.2
 5                                    65                 99.7
 6                                    52                 98.3
 7                                    48                 99
 8                                    49.5               96.2
 9                                    54                 97
10                                    71                 99.2
   `temperature fahrenheit`
                      <dbl>
 1                     15.5
 2                     14.7
 3                     16
 4                     18.5
 5                     15
 6                     14
 7                     21
 8                     19
 9                     15
10                     11
# 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)`.

```
# # TODO
mimic_icu_cohort <- icustays_tble %>%
  left_join(admissions_tble, by = c("subject_id", "hadm_id")) %>%
  left_join(patients_tble, by = "subject_id") %>%
  # keep adults only (age >= 18), using MIMIC-IV's anchor_age
  mutate(ageintime = anchor_age + (year(intime) - anchor_year))%>%
  filter(ageintime >= 18) %>%
  left_join(labevents_tble, by = c("subject_id", "stay_id")) %>%
  left_join(chartevents_tble, by = c("subject_id", "stay_id")) %>%
  select(-intime_x) %>%
  rename(intime = intime_y) %>%
  collect() %>%
  arrange(subject_id, hadm_id, stay_id) %>%
  print(mimic_icu_cohort, 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?
```

```
Warning: `...` must be empty in `format.tbl()`
Caused by error in `format_tbl()`:
! `...` must be empty.
x Problematic argument:
* ..1 = mimic_icu_cohort
i Did you forget to name an argument?
```

```
# 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                                    outtime               los
   <chr>                                            <dttm>              <dbl>
 1 Medical Intensive Care Unit (MICU)               2180-07-23 23:50:47 0.410
 2 Medical Intensive Care Unit (MICU)               2150-11-06 17:03:17 3.89
 3 Medical Intensive Care Unit (MICU)               2189-06-27 20:38:27 0.498
 4 Surgical Intensive Care Unit (SICU)              2157-11-21 22:08:00 1.12
 5 Surgical Intensive Care Unit (SICU)              2157-12-20 14:27:41 0.948
 6 Medical/Surgical Intensive Care Unit (MICU/SICU) 2110-04-12 23:59:56 1.34
 7 Medical/Surgical Intensive Care Unit (MICU/SICU) 2134-12-06 14:38:26 0.825
 8 Medical Intensive Care Unit (MICU)               2131-01-20 08:27:30 9.17
 9 Cardiac Vascular Intensive Care Unit (CVICU)     2160-05-19 17:33:33 1.31
10 Coronary Care Unit (CCU)                         2162-02-20 21:16:27 2.91
   admittime           dischtime           deathtime
   <dttm>              <dttm>              <dttm>
 1 2180-07-23 12:35:00 2180-07-25 17:55:00 NA
 2 2150-11-02 18:02:00 2150-11-12 13:45:00 NA
 3 2189-06-27 07:38:00 2189-07-03 03:00:00 NA
 4 2157-11-18 22:56:00 2157-11-25 18:00:00 NA
 5 2157-12-18 16:58:00 2157-12-24 14:55:00 NA
 6 2110-04-11 15:08:00 2110-04-14 15:00:00 NA
 7 2134-12-05 00:10:00 2134-12-06 12:54:00 2134-12-06 12:54:00
 8 2131-01-07 20:39:00 2131-01-20 05:15:00 2131-01-20 05:15:00
 9 2160-05-18 07:45:00 2160-05-23 13:30:00 NA
10 2162-02-17 22:32:00 2162-03-04 15:16:00 NA
   admission_type              admit_provider_id admission_location
   <chr>                       <chr>             <chr>
 1 EW EMER.                    P06OTX            EMERGENCY ROOM
 2 EW EMER.                    P26QQ4            EMERGENCY ROOM
 3 EW EMER.                    P06OTX            EMERGENCY ROOM
 4 EW EMER.                    P361ON            EMERGENCY ROOM
 5 DIRECT EMER.                P276OU            PHYSICIAN REFERRAL
 6 EW EMER.                   P32W56            PACU
 7 URGENT                     P67ATB            TRANSFER FROM HOSPITAL
 8 OBSERVATION ADMIT          P49AFC            EMERGENCY ROOM
 9 SURGICAL SAME DAY ADMISSION P8286C           PHYSICIAN REFERRAL
10 OBSERVATION ADMIT          P46834            PHYSICIAN REFERRAL
   discharge_location insurance language marital_status race
   <chr>              <chr>     <chr>    <chr>          <chr>
 1 HOME               Medicaid  English  WIDOWED        WHITE
 2 REHAB              Medicare  English  WIDOWED        WHITE
 3 HOME HEALTH CARE   Medicare  English  MARRIED        BLACK/AFRICAN AMERICAN
 4 HOME HEALTH CARE   Private   Other    MARRIED        WHITE
 5 HOME HEALTH CARE   Private   Other    MARRIED        WHITE
 6 HOME               Private   English  MARRIED        WHITE
 7 DIED               Medicare  English  SINGLE         WHITE
 8 DIED               Medicare  English  MARRIED        BLACK/AFRICAN AMERICAN
 9 HOME HEALTH CARE   Medicare  English  SINGLE         OTHER
10 HOME HEALTH CARE   Medicaid  English  <NA>           UNKNOWN
   edregtime           edouttime           hospital_expire_flag gender
   <dttm>              <dttm>                             <int> <chr>
 1 2180-07-23 05:54:00 2180-07-23 14:00:00                    0 F
 2 2150-11-02 11:41:00 2150-11-02 19:37:00                    0 F
 3 2189-06-27 06:25:00 2189-06-27 08:42:00                    0 F
 4 2157-11-18 17:38:00 2157-11-19 01:24:00                    0 F
 5 NA                  NA                                     0 F
 6 NA                  NA                                     0 F
 7 NA                  NA                                     1 M
 8 2131-01-07 13:36:00 2131-01-07 22:13:00                    1 F
 9 NA                  NA                                     0 F
10 2162-02-17 19:35:00 2162-02-17 23:30:00                    0 M
   anchor_age anchor_year anchor_year_group dod        ageintime bicarbonate
        <int>       <int> <chr>             <date>         <int>       <dbl>
 1         52        2180 2014 - 2016       2180-09-09        52          25
 2         86        2150 2008 - 2010       2152-01-30        86          26
 3         73        2186 2008 - 2010       2193-08-26        76          21
 4         55        2157 2011 - 2013       NA                55          22
 5         55        2157 2011 - 2013       NA                55          30
 6         46        2110 2011 - 2013       NA                46          NA
 7         73        2131 2017 - 2019       2134-12-06        76          28
 8         68        2122 2008 - 2010       2131-01-20        77          30
 9         53        2156 2008 - 2010       NA                57          24
10         56        2162 2020 - 2022       2162-12-11        56          18
   chloride creatinine glucose hematocrit intime              potassium sodium
      <dbl>      <dbl>   <dbl>      <dbl> <dttm>                  <dbl>  <dbl>
 1       95       0.7     102       41.1 2180-07-23 14:00:00       6.7    126
 2      100       1        85       36.1 2150-11-02 19:37:00       4.8    137
 3      109       2.3      89       27.3 2189-06-27 08:42:00       3.9    144
 4      108       0.6     112       38.1 2157-11-20 19:18:02       4.2    142
 5      104       0.5      87       37.4 2157-12-19 15:42:24       4.1    142
 6       98      NA        NA       NA   2110-04-11 15:52:22       4.1    139
 7       97       1.3     131       31.4 2134-12-05 18:50:03       3.9    138
 8       88       1.1     141       39.7 2131-01-11 04:20:05       4.5    130
 9      102       0.9     288       34.9 2160-05-18 10:00:53       3.5    137
10       NA       3.1      95       34.3 2162-02-17 23:30:00       6.5    125
     wbc `heart rate` `non invasive blood pressure systolic`
   <dbl>        <dbl>                                  <dbl>
 1   6.9         91                                     84
 2   7.1         78                                    106
 3   5.3         76                                    154
 4  15.7         86                                    151
 5   5.4         79.3                                  156
 6  NA           86                                     73
 7  10.4        124.                                   110
 8  12.2         49                                    174.
 9   7.2         80                                     98.5
10  16.8        110.                                   112
   `non invasive blood pressure diastolic` `respiratory rate`
                                     <dbl>              <dbl>
 1                                    48                 98.7
 2                                    56.5               97.7
 3                                   102                 98
 4                                    90                 98.5
 5                                    93.3               97.6
 6                                    56                 97.7
 7                                    78                 97.9
 8                                    30.5               98.1
 9                                    62                 97.2
10                                    80                 97.9
   `temperature fahrenheit`
                      <dbl>
 1                     24
 2                     24.3
 3                     23.5
 4                     18
 5                     14
 6                     19
 7                     16.5
 8                      13
 9                      14
10                      21
# i 94,448 more rows
```

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:

```r
mimic_icu_cohort_gtsummary <- 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 = 4, other_level = "Other"),
    discharge_location  = fct_lump_n(discharge_location, n = 4, other_level = "Other"),
    language = language,
    race = case_when(
      str_detect(race, "ASIAN") ~ "ASIAN",
      str_detect(race, "BLACK") ~ "BLACK",
      str_detect(race, "HISPANIC") ~ "HISPANIC",
      str_detect(race, "WHITE") ~ "WHITE",
      TRUE ~ "Other"
    ) %>%
      factor(levels = c("ASIAN", "BLACK", "HISPANIC", "WHITE", "Other")),

    los_long = (los >= 2),

    `non invasive blood pressure systolic`  = as.numeric(`non invasive blood pressure systoli
    `non invasive blood pressure diastolic` = as.numeric(`non invasive blood pressure diastol
    `respiratory rate` = as.numeric(`respiratory rate`),
    `temperature fahrenheit` = as.numeric(`temperature fahrenheit`),
    `heart rate` = as.numeric(`heart rate`)
  ) %>%
  select(
    first_careunit,
    last_careunit,
    los,
    admission_type,
    admission_location,
    discharge_location,
    insurance,
    language,
    marital_status,
    race,
    hospital_expire_flag,
    gender,
    dod,
    chloride,
    creatinine,
    sodium,
    potassium,
    glucose,
    hematocrit,
    wbc,
    bicarbonate,
    `non invasive blood pressure systolic`,
    `non invasive blood pressure diastolic`,
    `respiratory rate`,
    `temperature fahrenheit`,
    `heart rate`,
    anchor_age,
    los_long
  )

final_tbl <- mimic_icu_cohort_gtsummary %>%
  tbl_summary(
    by = los_long,
    missing = "ifany",
    missing_text = "Unknown",
    statistic = list(
      all_continuous() ~ "{median} ({p25}, {p75})",
      all_categorical() ~ "{n} ({p}%)"
    )
  ) %>%
  modify_header(label = "**Characteristic**") %>%
  bold_labels()
```

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

```
final_tbl
```

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.

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%)	
los	3.9 (2.7, 6.8)	
admission_type		
EW EMER.	23,012 (50%)	
OBSERVATION ADMIT	7,393 (16%)	
SURGICAL SAME DAY ADMISSION	4,001 (8.6%)	
URGENT	8,691 (19%)	
Other	3,240 (7.0%)	
admission_location		
EMERGENCY ROOM	17,058 (37%)	
PHYSICIAN REFERRAL	11,013 (24%)	
TRANSFER FROM HOSPITAL	13,904 (30%)	
WALK-IN/SELF REFERRAL	2,169 (4.7%)	
Other	2,193 (4.7%)	
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%)	