Biostat 203B Homework 4

Due Mar 9 @ 11:59PM

Yanzi Sun 106183069

Table of contents

Display machine information:

```r
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: aarch64-apple-darwin20
Running under: macOS Sonoma 14.7.4

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_1.1.4       evaluate_1.0.1
```

Display my machine memory.

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

```
Totalram:   16.000 GiB
Freeram:   144.875 MiB
```

Load database libraries and the tidyverse frontend:

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

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.2
-- 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 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.

```r
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.

```
dlabitems_tble <- tbl(con_bq, "d_labitems") |>
  filter(itemid %in%
           c(50912, 50971, 50983, 50902, 50882, 51221, 51301, 50931)) |>
  collect()

labevents_tble <- tbl(con_bq, "labevents") |>
  filter(itemid %in% dlabitems_tble$itemid) |>
  left_join(
    select(icustays_tble, subject_id, stay_id, intime),
    by = "subject_id"
  ) |>
  filter(storetime < intime) |>
  group_by(subject_id, stay_id, itemid) |>
  slice_max(storetime, n = 1) |>
  ungroup() |>
  select(subject_id, stay_id, itemid, valuenum) |>
  pivot_wider(names_from = itemid, values_from = valuenum) |>
  rename_at(
    vars(as.character(dlabitems_tble$itemid)),
    ~ str_to_lower(dlabitems_tble$label)
  ) |>
  rename(wbc = `white blood cells`) |>
  arrange(subject_id, stay_id) |>
  relocate(subject_id, stay_id, bicarbonate, chloride, creatinine, glucose, potassium, sodiu
  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 bicarbonate chloride creatinine glucose potassium sodium
        <int>    <int>       <dbl>    <dbl>      <dbl>   <dbl>     <dbl>  <dbl>
 1   10000032 39553978          25       95        0.7     102       6.7    126
 2   10000690 37081114          26      100        1        85       4.8    137
 3   10000980 39765666          21      109        2.3      89       3.9    144
 4   10001217 34592300          30      104        0.5      87       4.1    142
 5   10001217 37067082          22      108        0.6     112       4.2    142
 6   10001725 31205490          NA       98       NA        NA       4.1    139
 7   10001843 39698942          28       97        1.3     131       3.9    138
 8   10001884 37510196          30       88        1.1     141       4.5    130
 9   10002013 39060235          24      102        0.9     288       3.5    137
10   10002114 34672098          18       NA        3.1      95       6.5    125
   hematocrit   wbc
        <dbl> <dbl>
 1       41.1   6.9
 2       36.1   7.1
 3       27.3   5.3
 4       37.4   5.4
 5       38.1  15.7
 6       NA    NA
 7       31.4  10.4
 8       39.7  12.2
 9       34.9   7.2
10       34.3  16.8
# i more rows
```

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.

```
# # TODO
# chartevents_tble <- '

dchartitems_tble <-  tbl(con_bq, "d_items")|>
  filter(itemid %in% c(
    220045, 220179, 220180, 223761, 220210)) |>
  mutate(itemid = as.integer(itemid)) |>
  collect()

chartevents_tble <- tbl(con_bq, "chartevents") |>
  select(subject_id, itemid, storetime, valuenum) |>
  filter(itemid %in% dchartitems_tble$itemid) |>
  left_join(
    select(icustays_tble, subject_id, stay_id),
    by=c("subject_id"),
    #copy=TRUE copies the r table into duckdb table to make them mergeable
    copy = TRUE )|>
  group_by(subject_id, stay_id, itemid) |>
  # i forgot if Dr. Zhou want us to take average of the mean value or use the first stored va
  #summarise(mean_valuenum = mean(valuenum, na.rm = TRUE), .groups = "drop") |>
  slice_min(storetime, n = 1) |>
  select(-storetime) |>
  ungroup() |>
  pivot_wider(names_from = itemid, values_from = valuenum) |>
  rename_at (
    vars(as.character(dchartitems_tble$itemid)),
    ~str_to_lower(dchartitems_tble$label)
  ) |>
  # # # show_query() |>
  # collect() |>
  arrange(subject_id, stay_id) |>
  relocate(subject_id, stay_id, `heart rate`,
           `non invasive blood pressure systolic`,
           `non invasive blood pressure diastolic`,
           `respiratory rate`, `temperature fahrenheit`) |>
  print(width = Inf)
```

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

```
# Source:     SQL [?? x 7]
# Database:   BigQueryConnection
# Ordered by: subject_id, stay_id
   subject_id  stay_id `heart rate` `non invasive blood pressure systolic`
        <int>    <int>        <dbl>                                  <dbl>
 1   10000032 39553978           91                                     84
 2   10000690 37081114           80                                    107
 3   10000980 39765666           77                                    158
 4   10001217 34592300           86                                    151
 5   10001217 37067082           86                                    151
 6   10001725 31205490           86                                     73
 7   10001843 39698942          131                                    112
 8   10001884 37510196           60                                    180
 9   10002013 39060235           80                                    104
10   10002114 34672098          111                                    112
   `non invasive blood pressure diastolic` `respiratory rate`
                                     <dbl>              <dbl>
 1                                      48                 24
 2                                      63                 27
 3                                     127                 24
 4                                      90                 18
 5                                      90                 18
 6                                      56                  19
 7                                      85                  17
 8                                      49                  16
 9                                      70                  14
10                                      80                  22
   `temperature fahrenheit`
                      <dbl>
 1                     98.7
 2                     97.7
 3                     98
 4                     98.5
 5                     98.5
 6                     97.7
 7                     97.9
 8                     98.1
 9                     97.2
10                     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 |>
  left_join(patients_tble, by = c("subject_id")) |>
  mutate(age_at_intime = year(intime) - anchor_year + anchor_age) |>
  filter(age_at_intime >= 18) |>
  left_join(admissions_tble, by = c("subject_id", "hadm_id"))

# Ensure `stay_id` is an integer to prevent partitioning issues
first_vitals <- chartevents_tble |>
  mutate(stay_id = as.integer(stay_id)) |>  # Convert stay_id to INT64
  group_by(subject_id, stay_id) |>
  slice_min(stay_id, n = 1) |>
  ungroup()

last_labs <- labevents_tble |>
  mutate(stay_id = as.integer(stay_id)) |>  # Convert stay_id to INT64
  group_by(subject_id, stay_id) |>
  slice_max(stay_id, n = 1) |>
  ungroup()

# JOIN summarized tables to mimic_icu_cohort
mimic_icu_cohort <- mimic_icu_cohort |>
  left_join(first_vitals, by = c("subject_id", "stay_id")) |>
  left_join(last_labs, by = c("subject_id", "stay_id")) |>
  collect() |>  # Collect data BEFORE using arrange()
  mutate(stay_id = as.integer(stay_id)) |>  # Convert to INT after collecting
  arrange(subject_id, hadm_id, stay_id)  # Arrange AFTER collecting
```

```
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?
```

```
print(mimic_icu_cohort)
```

```
# A tibble: 94,458 x 41
   subject_id  hadm_id  stay_id first_careunit last_careunit intime
        <int>    <int>    <int> <chr>          <chr>         <dttm>
 1   10000032 29079034 39553978 Medical Inten~ Medical Inte~ 2180-07-23 14:00:00
 2   10000690 25860671 37081114 Medical Inten~ Medical Inte~ 2150-11-02 19:37:00
 3   10000980 26913865 39765666 Medical Inten~ Medical Inte~ 2189-06-27 08:42:00
 4   10001217 24597018 37067082 Surgical Inte~ Surgical Int~ 2157-11-20 19:18:02
 5   10001217 27703517 34592300 Surgical Inte~ Surgical Int~ 2157-12-19 15:42:24
 6   10001725 25563031 31205490 Medical/Surgi~ Medical/Surg~ 2110-04-11 15:52:22
 7   10001843 26133978 39698942 Medical/Surgi~ Medical/Surg~ 2134-12-05 18:50:03
 8   10001884 26184834 37510196 Medical Inten~ Medical Inte~ 2131-01-11 04:20:05
 9   10002013 23581541 39060235 Cardiac Vascu~ Cardiac Vasc~ 2160-05-18 10:00:53
```

```
10   10002114 27793700 34672098 Coronary Care~ Coronary Car~ 2162-02-17 23:30:00
# i 94,448 more rows
# i 35 more variables: outtime <dttm>, los <dbl>, gender <chr>,
#   anchor_age <int>, anchor_year <int>, anchor_year_group <chr>, dod <date>,
#   age_at_intime <int>, admittime <dttm>, dischtime <dttm>, deathtime <dttm>,
#   admission_type <chr>, admit_provider_id <chr>, admission_location <chr>,
#   discharge_location <chr>, insurance <chr>, language <chr>,
#   marital_status <chr>, race <chr>, edregtime <dttm>, edouttime <dttm>, ...
```

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
unique(mimic_icu_cohort$race)
```

```
 [1] "WHITE"
 [2] "BLACK/AFRICAN AMERICAN"
 [3] "OTHER"
 [4] "UNKNOWN"
 [5] "UNABLE TO OBTAIN"
 [6] "WHITE - RUSSIAN"
 [7] "PORTUGUESE"
 [8] "BLACK/CAPE VERDEAN"
 [9] "HISPANIC/LATINO - SALVADORAN"
[10] "HISPANIC/LATINO - PUERTO RICAN"
[11] "ASIAN - SOUTH EAST ASIAN"
[12] "WHITE - OTHER EUROPEAN"
[13] "WHITE - BRAZILIAN"
[14] "HISPANIC OR LATINO"
[15] "BLACK/AFRICAN"
[16] "PATIENT DECLINED TO ANSWER"
[17] "HISPANIC/LATINO - GUATEMALAN"
[18] "ASIAN"
[19] "BLACK/CARIBBEAN ISLAND"
[20] "HISPANIC/LATINO - CUBAN"
```

```
[21] "ASIAN - CHINESE"
[22] "HISPANIC/LATINO - DOMINICAN"
[23] "ASIAN - KOREAN"
[24] "ASIAN - ASIAN INDIAN"
[25] "AMERICAN INDIAN/ALASKA NATIVE"
[26] "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER"
[27] "WHITE - EASTERN EUROPEAN"
[28] "HISPANIC/LATINO - CENTRAL AMERICAN"
[29] "HISPANIC/LATINO - HONDURAN"
[30] "HISPANIC/LATINO - COLUMBIAN"
[31] "SOUTH AMERICAN"
[32] "HISPANIC/LATINO - MEXICAN"
[33] "MULTIPLE RACE/ETHNICITY"
```

```r
library(forcats)
mimic_icu_cohort <- 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"),

    race = fct_collapse(race,
      ASIAN = c("ASIAN", "ASIAN - CHINESE", "ASIAN - KOREAN",
                "ASIAN - ASIAN INDIAN", "ASIAN - SOUTH EAST ASIAN",
                "ASIAN - VIETNAMESE", "ASIAN - FILIPINO", "ASIAN - CAMBODIAN",
                "ASIAN - OTHER", "ASIAN - JAPANESE", "ASIAN - THAI"),
      BLACK = c("BLACK/AFRICAN AMERICAN", "BLACK/CAPE VERDEAN",
                "BLACK/HAITIAN", "BLACK/AFRICAN", "BLACK/CARIBBEAN ISLAND"),
      HISPANIC = c("HISPANIC OR LATINO", "HISPANIC/LATINO - PUERTO RICAN",
                   "HISPANIC/LATINO - DOMINICAN", "HISPANIC/LATINO - GUATEMALAN",
                   "HISPANIC/LATINO - CUBAN", "HISPANIC/LATINO - SALVADORAN",
                   "HISPANIC/LATINO - MEXICAN", "HISPANIC/LATINO - COLUMBIAN",
                   "HISPANIC/LATINO - HONDURAN", "HISPANIC/LATINO - CENTRAL AMERICAN"),
      WHITE = c("WHITE", "WHITE - OTHER EUROPEAN", "WHITE - EASTERN EUROPEAN",
                "WHITE - BRAZILIAN", "WHITE - RUSSIAN", "PORTUGUESE"),
      Other = c("UNKNOWN", "OTHER", "PATIENT DECLINED TO ANSWER",
                "UNABLE TO OBTAIN", "MULTIPLE RACE/ETHNICITY",
                "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER",
                "SOUTH AMERICAN", "AMERICAN INDIAN/ALASKA NATIVE")
    ),    los_long = los >= 2)
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `race = fct_collapse(...)`.
Caused by warning:
! Unknown levels in `f`: ASIAN - VIETNAMESE, ASIAN - FILIPINO, ASIAN - CAMBODIAN, ASIAN - OT
```

```r
mimic_icu_cohort|>
  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', age_at_intime, los_long) |>
  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
```

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.

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%)	

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.