Biostat 203B Homework 4

Due Mar 9 @ 11:59PM

Emma Mo 906542365

Display machine information:

```r
sessionInfo()
```

```
R version 4.4.1 (2024-06-14)
Platform: x86_64-apple-darwin20
Running under: macOS Monterey 12.4

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.4-x86_64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.4-x86_64/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_4.4.1    fastmap_1.2.0     cli_3.6.4         tools_4.4.1
 [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:   881.039 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     1.0.4
-- 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
```

```
library(dplyr)
```

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.

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

```
<SQL>
SELECT `icustays`.*
FROM `icustays`
ORDER BY `subject_id`, `hadm_id`, `stay_id`
# 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
# # TODO
admissions_tble <- tbl(con_bq, "admissions") |>
  arrange(subject_id, hadm_id) |>
  show_query() |>
  print(width = Inf)
```

```
<SQL>
SELECT `admissions`.*
FROM `admissions`
ORDER BY `subject_id`, `hadm_id`
# Source:     SQL [?? x 16]
# Database:   BigQueryConnection
# Ordered by: subject_id, hadm_id
   subject_id  hadm_id admittime           dischtime           deathtime
        <int>    <int> <dttm>              <dttm>              <dttm>   
 1   10000032 22595853 2180-05-06 22:23:00 2180-05-07 17:15:00 NA       
 2   10000032 22841357 2180-06-26 18:27:00 2180-06-27 18:49:00 NA       
 3   10000032 25742920 2180-08-05 23:44:00 2180-08-07 17:50:00 NA       
 4   10000032 29079034 2180-07-23 12:35:00 2180-07-25 17:55:00 NA       
 5   10000068 25022803 2160-03-03 23:16:00 2160-03-04 06:26:00 NA       
 6   10000084 23052089 2160-11-21 01:56:00 2160-11-25 14:52:00 NA       
 7   10000084 29888819 2160-12-28 05:11:00 2160-12-28 16:07:00 NA       
 8   10000108 27250926 2163-09-27 23:17:00 2163-09-28 09:04:00 NA       
 9   10000117 22927623 2181-11-15 02:05:00 2181-11-15 14:52:00 NA       
```

```
10   10000117 27988844 2183-09-18 18:10:00 2183-09-21 16:30:00 NA
   admission_type    admit_provider_id admission_location     discharge_location
   <chr>             <chr>             <chr>                  <chr>
 1 URGENT            P49AFC            TRANSFER FROM HOSPITAL HOME
 2 EW EMER.          P784FA            EMERGENCY ROOM         HOME
 3 EW EMER.          P19UTS            EMERGENCY ROOM         HOSPICE
 4 EW EMER.          P06OTX            EMERGENCY ROOM         HOME
 5 EU OBSERVATION    P39NWO            EMERGENCY ROOM         <NA>
 6 EW EMER.          P42H7G            WALK-IN/SELF REFERRAL  HOME HEALTH CARE
 7 EU OBSERVATION    P35NE4            PHYSICIAN REFERRAL     <NA>
 8 EU OBSERVATION    P40JML            EMERGENCY ROOM         <NA>
 9 EU OBSERVATION    P47EY8            EMERGENCY ROOM         <NA>
10 OBSERVATION ADMIT P13ACE            WALK-IN/SELF REFERRAL  HOME HEALTH CARE
   insurance language marital_status race  edregtime
   <chr>     <chr>    <chr>          <chr> <dttm>
 1 Medicaid  English  WIDOWED        WHITE 2180-05-06 19:17:00
 2 Medicaid  English  WIDOWED        WHITE 2180-06-26 15:54:00
 3 Medicaid  English  WIDOWED        WHITE 2180-08-05 20:58:00
 4 Medicaid  English  WIDOWED        WHITE 2180-07-23 05:54:00
 5 <NA>      English  SINGLE         WHITE 2160-03-03 21:55:00
 6 Medicare  English  MARRIED        WHITE 2160-11-20 20:36:00
 7 Medicare  English  MARRIED        WHITE 2160-12-27 18:32:00
 8 <NA>      English  SINGLE         WHITE 2163-09-27 16:18:00
 9 Medicaid  English  DIVORCED       WHITE 2181-11-14 21:51:00
10 Medicaid  English  DIVORCED       WHITE 2183-09-18 08:41:00
   edouttime           hospital_expire_flag
   <dttm>                             <int>
 1 2180-05-06 23:30:00                    0
 2 2180-06-26 21:31:00                    0
 3 2180-08-06 01:44:00                    0
 4 2180-07-23 14:00:00                    0
 5 2160-03-04 06:26:00                    0
 6 2160-11-21 03:20:00                    0
 7 2160-12-28 16:07:00                    0
 8 2163-09-28 09:04:00                    0
 9 2181-11-15 09:57:00                    0
10 2183-09-18 20:20:00                    0
# i more rows
```

Q1.4 `patients` data

Connect to the `patients` table.

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

```
<SQL>
SELECT `patients`.*
FROM `patients`
ORDER BY `subject_id`
# 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.

```r
# TODO
itemid_label_lab <- c(
  "50912" = "creatinine",
  "50971" = "potassium",
  "50983" = "sodium",
  "50902" = "chloride",
  "50882" = "bicarbonate",
  "51221" = "hematocrit",
  "51301" = "wbc",
  "50931" = "glucose"
)
```

```
labevents_tble <- tbl(con_bq, "labevents") |>
  select(subject_id, itemid, storetime, valuenum) |>
  filter(
    itemid %in% c(50912, 50971, 50983, 50902, 50882, 51221, 51301, 50931)) |>
  left_join(
    select(icustays_tble, subject_id, stay_id, intime),
    by = c("subject_id"),
  ) |>
  filter(storetime < intime) |>
  group_by(subject_id, stay_id, itemid) |>
  slice_max(storetime) |>
  select(-storetime, -intime) |>
  ungroup() |>
  pivot_wider(names_from = itemid, values_from = valuenum) |>
  rename_at(
    vars(names(itemid_label_lab)),
    ~ itemid_label_lab[.]
    ) |>
  # show_query() |>
  arrange(subject_id, stay_id) |>
  # relocate(subject_id, stay_id, sort(names(.))) |>
  print()
```

```
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
        <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
# i more rows
# i 2 more variables: bicarbonate <dbl>, hematocrit <dbl>
```

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
itemid_label_chart <- c(
  "220045" = "heart_rate",
  "220179" = "non_invasive_blood_pressure_systolic",
  "220180" = "non_invasive_blood_pressure_diastolic",
  "223761" = "temperature_fahrenheit",
  "220210" = "respiratory_rate"
)
```

```
chartevents_tble <- tbl(con_bq, "chartevents") |>
  select(subject_id, storetime, valuenum, itemid) |>
  filter(itemid %in% c(220045, 220179, 220180, 223761, 220210)) |>
  left_join(
    select(icustays_tble, subject_id, stay_id, intime, outtime),
    by = "subject_id",
    ) |>
  filter(storetime >= intime, storetime <= outtime) |>
  group_by(subject_id, stay_id, itemid) |>
  slice_min(storetime) |>
  summarise(valuenum = mean(as.numeric(valuenum), na.rm = TRUE),
            .groups = "drop") |>
  ungroup() |>
  pivot_wider(names_from = itemid, values_from = valuenum) |>
  rename_at(
    vars(names(itemid_label_chart)),
    ~ itemid_label_chart[.]
  ) |>
  # show_query() |>
  arrange(subject_id, stay_id) |>
  # relocate(subject_id, stay_id, sort(names(.))) |>
  print()
```

```
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 temperature_fahrenheit non_invasive_blood_pressure_syst~1
        <int>    <int>                  <dbl>                              <dbl>
 1   10000032 39553978                   98.7                                84
 2   10000690 37081114                   97.7                               106
 3   10000980 39765666                   98                                 154
 4   10001217 34592300                   97.6                               156
 5   10001217 37067082                   98.5                               151
 6   10001725 31205490                   97.7                                73
 7   10001843 39698942                   97.9                               110
 8   10001884 37510196                   98.1                               174.
 9   10002013 39060235                   97.2                                98.5
10   10002114 34672098                   97.9                               112
# i more rows
# i abbreviated name: 1: non_invasive_blood_pressure_systolic
# i 3 more variables: respiratory_rate <dbl>,
#   non_invasive_blood_pressure_diastolic <dbl>, heart_rate <dbl>
```

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
# TODO
mimic_icu_cohort <- icustays_tble |>
  left_join(admissions_tble, by = c("subject_id", "hadm_id")) |>
  left_join(patients_tble, by = "subject_id") |>
  mutate(age_intime = anchor_age + (year(intime) - anchor_year)) |>
  filter(age_intime >= 18) |>
  left_join(labevents_tble, by = c("subject_id", "stay_id")) |>
  left_join(chartevents_tble, by = c("subject_id", "stay_id")) |>

  collect() |>
  arrange(subject_id, hadm_id, stay_id) |>
  print(width = Inf)
```

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

```
# 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.                    P361ON           
 5 NA                  DIRECT EMER.                P276OU           
 6 NA                  EW EMER.                    P32W56           
 7 2134-12-06 12:54:00 URGENT                      P67ATB           
 8 2131-01-20 05:15:00 OBSERVATION ADMIT           P49AFC           
 9 NA                  SURGICAL SAME DAY ADMISSION P8286C           
10 NA                  OBSERVATION ADMIT           P46834           
   admission_location     discharge_location insurance language marital_status
   <chr>                  <chr>              <chr>     <chr>    <chr>         
 1 EMERGENCY ROOM         HOME               Medicaid  English  WIDOWED       
 2 EMERGENCY ROOM         REHAB              Medicare  English  WIDOWED       
 3 EMERGENCY ROOM         HOME HEALTH CARE   Medicare  English  MARRIED       
 4 EMERGENCY ROOM         HOME HEALTH CARE   Private   Other    MARRIED       
 5 PHYSICIAN REFERRAL     HOME HEALTH CARE   Private   Other    MARRIED       
 6 PACU                   HOME                Private   English  MARRIED
 7 TRANSFER FROM HOSPITAL DIED                Medicare  English  SINGLE
 8 EMERGENCY ROOM         DIED                Medicare  English  MARRIED
 9 PHYSICIAN REFERRAL     HOME HEALTH CARE   Medicare  English  SINGLE
10 PHYSICIAN REFERRAL     HOME HEALTH CARE   Medicaid  English  <NA>
   race                   edregtime           edouttime
   <chr>                  <dttm>              <dttm>
 1 WHITE                  2180-07-23 05:54:00 2180-07-23 14:00:00
 2 WHITE                  2150-11-02 11:41:00 2150-11-02 19:37:00
 3 BLACK/AFRICAN AMERICAN 2189-06-27 06:25:00 2189-06-27 08:42:00
 4 WHITE                  2157-11-18 17:38:00 2157-11-19 01:24:00
 5 WHITE                  NA                  NA
 6 WHITE                  NA                  NA
 7 WHITE                  NA                  NA
 8 BLACK/AFRICAN AMERICAN 2131-01-07 13:36:00 2131-01-07 22:13:00
 9 OTHER                  NA                  NA
10 UNKNOWN                2162-02-17 19:35:00 2162-02-17 23:30:00
   hospital_expire_flag gender anchor_age anchor_year anchor_year_group
                  <int> <chr>       <int>       <int> <chr>
 1                    0 F              52        2180 2014 - 2016
 2                    0 F              86        2150 2008 - 2010
 3                    0 F              73        2186 2008 - 2010
 4                    0 F              55        2157 2011 - 2013
 5                    0 F              55        2157 2011 - 2013
 6                    0 F              46        2110 2011 - 2013
 7                    1 M              73        2131 2017 - 2019
 8                    1 F              68        2122 2008 - 2010
 9                    0 F              53        2156 2008 - 2010
10                    0 M              56        2162 2020 - 2022
   dod        age_intime glucose potassium sodium chloride creatinine   wbc
   <date>          <int>   <dbl>     <dbl>  <dbl>    <dbl>      <dbl> <dbl>
 1 2180-09-09         52     102       6.7    126       95        0.7   6.9
 2 2152-01-30         86      85       4.8    137      100        1     7.1
 3 2193-08-26         76      89       3.9    144      109        2.3   5.3
 4 NA                 55     112       4.2    142      108        0.6  15.7
 5 NA                 55      87       4.1    142      104        0.5   5.4
 6 NA                 46      NA       4.1    139       98       NA    NA
 7 2134-12-06         76     131       3.9    138       97        1.3  10.4
 8 2131-01-20         77     141       4.5    130       88        1.1  12.2
 9 NA                 57     288       3.5    137      102        0.9   7.2
10 2162-12-11         56      95       6.5    125       NA        3.1  16.8
   bicarbonate hematocrit temperature_fahrenheit
         <dbl>      <dbl>                  <dbl>
 1          25       41.1                   98.7
 2          26       36.1                   97.7
 3          21       27.3                   98
 4          22       38.1                   98.5
 5          30       37.4                   97.6
 6          NA       NA                     97.7
 7          28       31.4                   97.9
 8          30       39.7                   98.1
 9          24       34.9                   97.2
10          18       34.3                   97.9
   non_invasive_blood_pressure_systolic respiratory_rate
                                  <dbl>            <dbl>
 1                                 84               24
 2                                106               24.3
 3                                154               23.5
 4                                151               18
 5                                156               14
 6                                 73               19
 7                                110               16.5
 8                                174.              13
 9                                 98.5             14
10                                112               21
   non_invasive_blood_pressure_diastolic heart_rate
                                   <dbl>      <dbl>
 1                                  48         91
 2                                  56.5       78
 3                                 102         76
 4                                  90         86
 5                                  93.3       79.3
 6                                  56         86
 7                                  78        124.
 8                                  30.5       49
 9                                  62         80
10                                  80        110.
# 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