

Biostat 203B Homework 4

Due Mar 9 @ 11:59PM

AUTHOR

Zhiyuan Yu 906405523

Display machine information:

```
sessionInfo()
```

R version 4.4.2 (2024-10-31)

Platform: x86_64-pc-linux-gnu

Running under: Ubuntu 24.04.1 LTS

Matrix products: default

BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.12.0

LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.12.0

locale:

```
[1] LC_CTYPE=C.UTF-8      LC_NUMERIC=C          LC_TIME=C.UTF-8
[4] LC_COLLATE=C.UTF-8    LC_MONETARY=C.UTF-8   LC_MESSAGES=C.UTF-8
[7] LC_PAPER=C.UTF-8      LC_NAME=C             LC_ADDRESS=C
[10] LC_TELEPHONE=C        LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
```

time zone: America/Los_Angeles

tzcode source: system (glibc)

attached base packages:

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

loaded via a namespace (and not attached):

```
[1] htmlwidgets_1.6.4 compiler_4.4.2 fastmap_1.2.0 cli_3.6.3
[5] tools_4.4.2      htmltools_0.5.8.1 rstudioapi_0.17.1 yaml_2.3.10
[9] rmarkdown_2.29   knitr_1.49       jsonlite_1.8.9   xfun_0.50
[13] digest_0.6.37    rlang_1.1.4      evaluate_1.0.3
```

Display my machine memory.

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

Totalram: 15.463 GiB

Freeram: 14.070 GiB

Load database libraries and the tidyverse frontend:

```
library(bigrquery)
library(dbplyr)
```

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

— Attaching core tidyverse packages — tidyverse 2.0.0 —

```
✓ dplyr      1.1.4    ✓ readr      2.1.5
✓ forcats    1.0.0    ✓ stringr    1.5.1
✓ ggplot2    3.5.1    ✓ tibble     3.2.1
✓ lubridate  1.9.4    ✓ tidyr      1.3.1
✓ purrr      1.0.2
```

— Conflicts — tidyverse_conflicts() —

```
✗ dplyr::filter() masks stats::filter()
✗ dplyr::ident()  masks dbplyr::ident()
✗ dplyr::lag()    masks stats::lag()
✗ dplyr::sql()    masks dbplyr::sql()
```

ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

```
library(tidyr)
library(dplyr)
library(readr)
library(forcats)
library(ggplot2)
library(shiny)
library(DT)
```

Attaching package: 'DT'

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

```
dataTableOutput, renderDataTable
```

```
library(lubridate)
library(arrow)
```

Attaching package: 'arrow'

The following object is masked from 'package:lubridate':

```
duration
```

The following object is masked from 'package:utils':

```
timestamp
```

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"	"hpcsevents"	"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>                                       <dtm>
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
    <dtm>                                <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>	<dtm>		<dtm>		<dtm>
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>	<dtm>		
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
  <dtm>          <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") |>
# show_query() |>
print(width = Inf)

```

```

# Source:   table<`patients`> [?? x 6]
# Database: BigQueryConnection
  subject_id gender anchor_age anchor_year anchor_year_group dod
      <int> <chr>      <int>      <int> <chr>              <date>
1    10078138 F           18        2110 2017 - 2019        NA
2    10180372 M           18        2110 2008 - 2010        NA
3    10686175 M           18        2110 2011 - 2013        NA
4    10851602 F           18        2110 2014 - 2016        NA
5    10902424 F           18        2110 2017 - 2019        NA
6    11092326 M           18        2110 2008 - 2010        NA
7    11289691 F           18        2110 2017 - 2019        NA
8    11595073 M           18        2110 2011 - 2013        NA
9    11739764 F           18        2110 2017 - 2019        NA
10   11776346 F           18        2110 2008 - 2010        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. **Solution:**

```

labevents_tble <- tbl(con_bq, "labevents") |>
  select(subject_id, itemid, storetime, valuenum) |>

```

```

inner_join(select(tbl(con_bq, "d_labitems") %>%
  filter(itemid %in% c(50912, 50971, 50983, 50902,
    50882, 51221, 51301, 50931)) %>%
  mutate(itemid = as.integer(itemid)), itemid), by = "itemid") |>
left_join(
  select(tbl(con_bq, "icustays") |>
  arrange(subject_id, hadm_id, stay_id),
  subject_id, stay_id, intime),
  by = c("subject_id"),
  copy = TRUE) |>

# Keep only lab items before ICU stay
filter(storetime < intime) |>

# Group by subject_id, stay_id, and itemid
group_by(subject_id, stay_id, itemid) |>

# Keep only the last lab value before ICU stay
slice_max(order_by = storetime, n = 1) |>

# Remove unnecessary columns
select(-storetime, -intime) |>

ungroup() |>

# Pivot wider to reshape data
pivot_wider(names_from = itemid, values_from = valuenum) |>

# Rename columns
mutate(
  Bicarbonate = `50882`,
  Chloride = `50902`,
  Creatinine = `50912`,
  Glucose = `50931`,
  Potassium = `50971`,
  Sodium = `50983`,
  Hematocrit = `51221`,
  White_Blood_Cells = `51301`
) |>

select(-`50882`, -`50902`, -`50912`, -`50931`, -`50971`, -`50983`, -`51221`,
  -`51301`) |>

arrange(subject_id, stay_id) |>

show_query(labevents_tble) |> print(labevents_tble)

```

Warning: ORDER BY is ignored in subqueries without LIMIT

❗ 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` = 50971, `valuenum`, NULL)) AS `Potassium`,
    MAX(IF(`itemid` = 50983, `valuenum`, NULL)) AS `Sodium`,
    MAX(IF(`itemid` = 51221, `valuenum`, NULL)) AS `Hematocrit`,
    MAX(IF(`itemid` = 51301, `valuenum`, NULL)) AS `White_Blood_Cells`
FROM (
    SELECT `subject_id`, `itemid`, `valuenum`, `stay_id`
    FROM (
        SELECT
            `q01`.*,
            RANK() OVER (PARTITION BY `subject_id`, `stay_id`, `itemid` ORDER BY `storetime` DESC) AS
`col01`
        FROM (
            SELECT
                `labevents`.`subject_id` AS `subject_id`,
                `labevents`.`itemid` AS `itemid`,
                `storetime`,
                `valuenum`,
                `stay_id`,
                `intime`
            FROM `labevents`
            INNER JOIN (
                SELECT SAFE_CAST(`itemid` AS INT64) AS `itemid`
                FROM `d_labitems`
                WHERE (`itemid` IN (50912.0, 50971.0, 50983.0, 50902.0, 50882.0, 51221.0, 51301.0,
50931.0))
            ) `...2`
            ON (`labevents`.`itemid` = `...2`.`itemid`)
            LEFT JOIN (
                SELECT `subject_id`, `stay_id`, `intime`
                FROM `icustays`
            ) `...3`
            ON (`labevents`.`subject_id` = `...3`.`subject_id`)
        ) `q01`
        WHERE (`storetime` < `intime`)
    ) `q01`
    WHERE (`col01` <= 1)
) `q01`
GROUP BY `subject_id`, `stay_id`
ORDER BY `subject_id`, `stay_id`

```


Warning: ``...`` must be empty in `format.tbl()`

ORDER BY is ignored in subqueries without LIMIT

i Do you need to move arrange() later in the pipeline or use window_order() instead?

Caused by error in `format.tbl()`:

! ``...`` must be empty.

X Problematic argument:

- ...1 = labevents_tble

i Did you forget to name an argument?

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

i more rows

i 2 more variables: Hematocrit <dbl>, White_Blood_Cells <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. **Soulution:**

```
chartevents_tble <- tbl(con_bq, "chartevents") |>

select(subject_id, stay_id, itemid, storetime, valuenum) |>
filter(itemid %in% c(220045, 220179, 220180, 223761, 220210)) |>
left_join(
  select(tbl(con_bq, "icustays"), subject_id, stay_id, intime, outtime),
  by = c("subject_id", "stay_id"),
  copy = TRUE
) |>
filter(storetime >= intime & storetime <= outtime) |>
group_by(subject_id, stay_id, itemid, storetime) |>

# Compute the average vital measurement per storetime
mutate(valuenum = mean(valuenum, na.rm = TRUE)) |>
ungroup() |>
```

```

# Group again to get the first recorded vital per subject, stay, item
group_by(subject_id, stay_id, itemid) |>
slice_min(order_by = storetime, n = 1) |>

# Remove unneeded columns
select(-storetime, -intime) |>

ungroup() |>
pivot_wider(names_from = itemid, values_from = valuenum) |>

# Rename columns dynamically inside BigQuery
mutate(
  heart_rate = `220045`,
  systolic_non_invasive_blood_pressure = `220179`,
  diastolic_non_invasive_blood_pressure = `220180`,
  temperature_fahrenheit = `223761`,
  respiratory_rate = `220210`
) |>

# Remove old numeric columns
select(-`220045`, -`220179`, -`220180`, -`223761`, -`220210`) |>
arrange(subject_id, stay_id)

show_query(chartevents_tble) |> print(chartevents_tble)

```

<SQL>

SELECT

```

`subject_id`,
`stay_id`,
`outtime`,
MAX(IF(`itemid` = 220045, `valuenum`, NULL)) AS `heart_rate`,
MAX(IF(`itemid` = 220179, `valuenum`, NULL)) AS `systolic_non_invasive_blood_pressure`,
MAX(IF(`itemid` = 220180, `valuenum`, NULL)) AS `diastolic_non_invasive_blood_pressure`,
MAX(IF(`itemid` = 223761, `valuenum`, NULL)) AS `temperature_fahrenheit`,
MAX(IF(`itemid` = 220210, `valuenum`, NULL)) AS `respiratory_rate`
FROM (
  SELECT `subject_id`, `stay_id`, `itemid`, `valuenum`, `outtime`
  FROM (
    SELECT
      `q01`.*,
      RANK() OVER (PARTITION BY `subject_id`, `stay_id`, `itemid` ORDER BY `storetime`) AS
`col01`
    FROM (
      SELECT
        `subject_id`,
        `stay_id`,
        `itemid`,
        `storetime`,
        AVG(`valuenum`) OVER (PARTITION BY `subject_id`, `stay_id`, `itemid`, `storetime`) AS
`valuenum`,

```

```

    `intime`,
    `outtime`
FROM (
  SELECT `LHS`.*, `intime`, `outtime`
  FROM (
    SELECT `subject_id`, `stay_id`, `itemid`, `storetime`, `valuenum`
    FROM `chartevents`
    WHERE (`itemid` IN (220045.0, 220179.0, 220180.0, 223761.0, 220210.0))
  ) `LHS`
  LEFT JOIN `icustays`
  ON (
    `LHS`.`subject_id` = `icustays`.`subject_id` AND
    `LHS`.`stay_id` = `icustays`.`stay_id`
  )
) `q01`
WHERE (`storetime` >= `intime` AND `storetime` <= `outtime`)
) `q01`
) `q01`
WHERE (`col01` <= 1)
) `q01`
GROUP BY `subject_id`, `stay_id`, `outtime`
ORDER BY `subject_id`, `stay_id`

```

Warning: `...` must be empty in `format.tbl()`

Caused by error in `format_tbl()`:

! `...` must be empty.

✖ Problematic argument:

- ..1 = chartevents_tble

❖ Did you forget to name an argument?

Source: SQL [?? x 8]

Database: BigQueryConnection

Ordered by: subject_id, stay_id

	subject_id	stay_id	outtime	heart_rate	systolic_non_invasive_blood...
	<int>	<int>	<dtm>	<dbl>	<dbl>
1	10000032	39553978	2180-07-23 23:50:47	91	84
2	10000690	37081114	2150-11-06 17:03:17	78	106
3	10000980	39765666	2189-06-27 20:38:27	76	154
4	10001217	34592300	2157-12-20 14:27:41	79.3	156
5	10001217	37067082	2157-11-21 22:08:00	86	151
6	10001725	31205490	2110-04-12 23:59:56	86	73
7	10001843	39698942	2134-12-06 14:38:26	124.	110
8	10001884	37510196	2131-01-20 08:27:30	49	174.
9	10002013	39060235	2160-05-19 17:33:33	80	98.5
10	10002114	34672098	2162-02-20 21:16:27	110.	112

i more rows

i abbreviated name: ¹systolic_non_invasive_blood_pressure

i 3 more variables: diastolic_non_invasive_blood_pressure <dbl>,

temperature_fahrenheit <dbl>, respiratory_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 \geq 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)`.

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

Warning: `...` must be empty in `format.tbl()`

Caused by error in `format_tbl()`:

! `...` must be empty.

✗ Problematic argument:

- ..1 = mimic_icu_cohort

i Did you forget to name an argument?

A tibble: 94,458 × 42

	subject_id	hadm_id	stay_id	first_careunit	last_careunit	intime
	<int>	<int>	<int>	<chr>	<chr>	<dtm>
1	10270110	20171261	35854639	PACU	PACU	2134-03-25 03:32:02
2	10270110	20171261	36372959	PACU	PACU	2134-03-24 01:31:39
3	10270644	20019675	35548343	PACU	PACU	2159-12-03 16:20:31
4	10368426	21588639	39194905	PACU	PACU	2164-12-30 13:29:21
5	10464753	28216499	32421516	PACU	PACU	2183-01-10 20:51:04
6	10640410	25898987	34344828	PACU	PACU	2112-02-03 12:55:23
7	10691194	24438843	37799251	PACU	PACU	2147-06-01 17:38:48
8	10710188	21362776	34067486	PACU	PACU	2147-06-22 11:48:40
9	10710188	21362776	36638120	PACU	PACU	2147-05-28 16:18:40
10	10826759	28468289	37075137	PACU	PACU	2121-05-19 18:07:00
	outtime_x	los	admittime	disctime		
	<dtm>	<dbl>	<dtm>	<dtm>		
1	2134-03-25 14:20:42	0.450	2134-03-22 04:57:00	2134-04-26 14:17:00		
2	2134-03-25 03:31:52	1.08	2134-03-22 04:57:00	2134-04-26 14:17:00		

	deathtime <dtm>	admission_type <chr>	admit_provider_id <chr>	admission_location <chr>	discharge_location <chr>	insurance <chr>	language <chr>	marital_status <chr>	race <chr>	edregtime <dtm>	edouttime <dtm>	hospital_expire_flag <int>	gender <chr>	anchor_age <int>	anchor_year <int>
3	2159-12-08 17:28:42	5.05	2159-12-03 01:17:00	2159-12-28 17:30:00											
4	2164-12-30 14:00:38	0.0217	2164-12-26 15:39:00	2165-01-03 16:30:00											
5	2183-01-11 22:58:45	1.09	2182-12-27 19:24:00	2183-01-27 17:39:00											
6	2112-02-08 15:14:54	5.10	2112-02-03 12:54:00	2112-02-19 18:00:00											
7	2147-06-01 17:58:44	0.0138	2147-04-25 08:30:00	2147-06-11 15:22:00											
8	2147-06-23 11:35:59	0.991	2147-05-28 16:17:00	2147-06-23 14:21:00											
9	2147-06-22 11:48:30	24.8	2147-05-28 16:17:00	2147-06-23 14:21:00											
10	2121-05-20 16:32:39	0.934	2121-05-19 17:00:00	2121-05-24 12:30:00											
1	NA	EW EMER.	P44KDZ												
2	NA	EW EMER.	P44KDZ												
3	NA	EW EMER.	P68D28												
4	NA	EW EMER.	P46834												
5	NA	OBSERVATION ADMIT	P411FD												
6	NA	OBSERVATION ADMIT	P55X3P												
7	NA	ELECTIVE	P93BYT												
8	2147-06-23 14:21:00	EW EMER.	P502T3												
9	2147-06-23 14:21:00	EW EMER.	P502T3												
10	NA	EW EMER.	P20PIB												
1				TRANSFER FROM HOSPITAL	HOSPICE	Medicaid									
2				TRANSFER FROM HOSPITAL	HOSPICE	Medicaid									
3				PHYSICIAN REFERRAL	SKILLED NURSING FACILITY	Medicare									
4				WALK-IN/SELF REFERRAL	SKILLED NURSING FACILITY	Medicare									
5				TRANSFER FROM HOSPITAL	HOSPICE	Medicare									
6				CLINIC REFERRAL	HOME HEALTH CARE	Private									
7				PHYSICIAN REFERRAL	SKILLED NURSING FACILITY	Medicare									
8				TRANSFER FROM SKILLED NURSING FACILITY	DIED	Medicare									
9				TRANSFER FROM SKILLED NURSING FACILITY	DIED	Medicare									
10				TRANSFER FROM HOSPITAL	REHAB	Medicare									
1	English	MARRIED	WHITE							2134-03-22 01:01:00					
2	English	MARRIED	WHITE							2134-03-22 01:01:00					
3	English	DIVORCED	WHITE							2159-12-02 19:45:00					
4	English	WIDOWED	WHITE							2164-12-26 08:22:00					
5	English	MARRIED	UNABLE TO OBTAIN							2182-12-27 18:59:00					
6	English	MARRIED	BLACK/AFRICAN							2112-02-03 08:05:00					
7	English	WIDOWED	WHITE							NA					
8	English	MARRIED	WHITE - OTHER EUROPEAN							2147-05-28 11:58:00					
9	English	MARRIED	WHITE - OTHER EUROPEAN							2147-05-28 11:58:00					
10	English	SINGLE	WHITE - BRAZILIAN							2121-05-19 08:03:00					
1	2134-03-22 07:40:00		0 M	78	2134										
2	2134-03-22 07:40:00		0 M	78	2134										
3	2159-12-03 02:51:00		0 F	84	2152										
4	2164-12-26 21:43:00		0 M	80	2154										
5	2182-12-27 21:24:00		0 M	86	2182										

6	2112-02-03 14:15:00	0 F	44	2112
7	NA	0 F	74	2144
8	2147-05-28 18:23:00	1 M	86	2147
9	2147-05-28 18:23:00	1 M	86	2147
10	2121-05-19 18:07:00	0 F	77	2121

	anchor_year_group	dod	Bicarbonate	Chloride	Creatinine	Glucose
	<chr>	<date>	<dbl>	<dbl>	<dbl>	<dbl>
1	2020 - 2022	2134-04-30	23	105	1	178
2	2020 - 2022	2134-04-30	24	104	0.8	98
3	2014 - 2016	2160-06-25	20	108	0.5	75
4	2011 - 2013	2165-03-18	24	109	0.8	131
5	2020 - 2022	2183-01-28	22	107	1.2	111
6	2017 - 2019	NA	21	96	10.3	102
7	2017 - 2019	2147-09-16	24	94	5	95
8	2020 - 2022	2147-06-23	31	107	0.6	173
9	2020 - 2022	2147-06-23	20	110	4.1	120
10	2020 - 2022	NA	NA	NA	NA	NA

	Potassium	Sodium	Hematocrit	White_Blood_Cells	outtime_y	heart_rate
	<dbl>	<dbl>	<dbl>	<dbl>	<dtm>	<dbl>
1	3.8	136	19.6	13.3	2134-03-25 14:20:42	86
2	4	137	24.8	42.1	2134-03-25 03:31:52	101
3	3.8	145	31.4	9.2	2159-12-08 17:28:42	55
4	4	138	29.4	4.8	NA	NA
5	3.6	140	31	12.8	2183-01-11 22:58:45	96
6	5.4	137	30.5	11.2	2112-02-08 15:14:54	97
7	4	135	26.2	8.7	NA	NA
8	5	144	29.6	10.8	2147-06-23 11:35:59	92
9	5.2	150	47.5	17.8	2147-06-22 11:48:30	62
10	NA	NA	37.4	8.3	2121-05-20 16:32:39	77

	systolic_non_invasive_blood_pressure	diastolic_non_invasive_blood_pressure
	<dbl>	<dbl>
1	NA	NA
2	NA	NA
3	92	62
4	NA	NA
5	106	63
6	173	107
7	NA	NA
8	89	61
9	104	61
10	116	58

	temperature_fahrenheit	respiratory_rate	age_intime
	<dbl>	<dbl>	<int>
1	97.7	14	78
2	96.7	15	78
3	97.9	16	91
4	NA	NA	90
5	97.3	20	87
6	97.8	22	44
7	NA	NA	77
8	101.	23	86

```

9          98.3          23          86
10         98.5         16.5          77
# i 94,448 more rows

```

Q1.8 Preprocessing [🔗](#)

Perform the following preprocessing steps. (i) Lump infrequent levels into “Other” level for `first_careunit`, `last_careunit`, `admission_type`, `admission_location`, and `discharge_location`. (ii) Collapse the levels of `race` into `ASIAN`, `BLACK`, `HISPANIC`, `WHITE`, and `Other`. (iii) Create a new variable `los_long` that is `TRUE` when `los` is greater than or equal to 2 days. (iv) Summarize the data using `tbl_summary()`, stratified by `los_long`. Hint: `fct_lump_n` and `fct_collapse` from the `forcats` package are useful.

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

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

Solution:*

```

# Inspect unique levels of the 'race' variable
unique_race_levels <- mimic_icu_cohort %>%
  pull(race) %>%
  unique()

print(unique_race_levels)

```

```

[1] "WHITE"
[2] "UNABLE TO OBTAIN"
[3] "BLACK/AFRICAN"
[4] "WHITE - OTHER EUROPEAN"
[5] "WHITE - BRAZILIAN"
[6] "UNKNOWN"
[7] "BLACK/AFRICAN AMERICAN"
[8] "PORTUGUESE"
[9] "OTHER"
[10] "ASIAN"
[11] "WHITE - RUSSIAN"
[12] "HISPANIC/LATINO - DOMINICAN"
[13] "BLACK/CARIBBEAN ISLAND"
[14] "HISPANIC/LATINO - PUERTO RICAN"
[15] "ASIAN - ASIAN INDIAN"
[16] "PATIENT DECLINED TO ANSWER"
[17] "AMERICAN INDIAN/ALASKA NATIVE"
[18] "HISPANIC/LATINO - SALVADORAN"
[19] "HISPANIC/LATINO - CENTRAL AMERICAN"
[20] "WHITE - EASTERN EUROPEAN"
[21] "ASIAN - SOUTH EAST ASIAN"
[22] "ASIAN - CHINESE"
[23] "HISPANIC/LATINO - GUATEMALAN"
[24] "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER"
[25] "BLACK/CAPE VERDEAN"
[26] "HISPANIC/LATINO - MEXICAN"
[27] "HISPANIC OR LATINO"
[28] "HISPANIC/LATINO - HONDURAN"
[29] "SOUTH AMERICAN"
[30] "HISPANIC/LATINO - COLUMBIAN"
[31] "ASIAN - KOREAN"
[32] "HISPANIC/LATINO - CUBAN"
[33] "MULTIPLE RACE/ETHNICITY"

```

```

mimic_icu_cohort <- mimic_icu_cohort %>%
  mutate(race = toupper(trimws(race)))

mimic_icu_cohort <- mimic_icu_cohort %>%
  mutate(
    race = fct_collapse(race,
      ASIAN = c("ASIAN", "ASIAN - ASIAN INDIAN", "ASIAN - SOUTH EAST ASIAN",
        "ASIAN - CHINESE", "ASIAN - KOREAN"),

      BLACK = c("BLACK/AFRICAN", "BLACK/AFRICAN AMERICAN",
        "BLACK/CARIBBEAN ISLAND", "BLACK/CAPE VERDEAN"),

      HISPANIC = c("HISPANIC OR LATINO", "HISPANIC/LATINO - DOMINICAN",
        "HISPANIC/LATINO - PUERTO RICAN",
        "HISPANIC/LATINO - SALVADORAN",
        "HISPANIC/LATINO - CENTRAL AMERICAN",

```



```

    "HISPANIC/LATINO - GUATEMALAN",
    "HISPANIC/LATINO - MEXICAN", "HISPANIC/LATINO - HONDURAN",
    "HISPANIC/LATINO - COLUMBIAN", "HISPANIC/LATINO - CUBAN"),

  WHITE = c("WHITE", "WHITE - OTHER EUROPEAN", "WHITE - BRAZILIAN",
    "WHITE - RUSSIAN", "WHITE - EASTERN EUROPEAN", "PORTUGUESE"),

  Other = c("UNKNOWN", "OTHER", "UNABLE TO OBTAIN",
    "PATIENT DECLINED TO ANSWER", "AMERICAN INDIAN/ALASKA NATIVE",
    "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER",
    "SOUTH AMERICAN", "MULTIPLE RACE/ETHNICITY")
)
) %>%

# Lump infrequent levels for specified categorical variables
mutate(across(c(first_careunit, last_careunit, admission_type, admission_location, discharge_lo
  ~ fct_lump_n(.x, n = 5, other_level = "Other")))) %>%

# Create los_long variable
mutate(los_long = los >= 2)

# Generate summary table stratified by 'los_long'
summary_table <- 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, White_Blood_Cells, Bicarbonate,
    systolic_non_invasive_blood_pressure,
    diastolic_non_invasive_blood_pressure, respiratory_rate,
    temperature_fahrenheit, heart_rate, age_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()`:

✗ For variable `dod` (`los_long = FALSE`) and "p75" statistic: * not defined for "Date" objects

```
summary_table %>% as_gt()
```

	TRUE	FALSE
Characteristic	N = 46,337 ¹	N = 48,107 ¹
first_careunit		

¹ n (%); Median (Q1, Q3)

Characteristic	TRUE N = 46,337 ¹	FALSE N = 48,107 ¹
Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)	7,416 (15%)
Coronary Care Unit (CCU)	5,433 (12%)	5,338 (11%)
Medical Intensive Care Unit (MICU)	9,837 (21%)	10,862 (23%)
Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)	8,780 (18%)
Surgical Intensive Care Unit (SICU)	6,434 (14%)	6,574 (14%)
Other	10,613 (23%)	9,137 (19%)
last_careunit		
Cardiac Vascular Intensive Care Unit (CVICU)	7,353 (16%)	7,416 (15%)
Coronary Care Unit (CCU)	5,433 (12%)	5,338 (11%)
Medical Intensive Care Unit (MICU)	9,837 (21%)	10,862 (23%)
Medical/Surgical Intensive Care Unit (MICU/SICU)	6,667 (14%)	8,780 (18%)
Surgical Intensive Care Unit (SICU)	6,434 (14%)	6,574 (14%)
Other	10,613 (23%)	9,137 (19%)
los	3.9 (2.7, 6.8)	1.1 (0.8, 1.5)
admission_type		
DIRECT EMER.	1,726 (3.7%)	1,590 (3.3%)
EW EMER.	23,012 (50%)	25,337 (53%)
OBSERVATION ADMIT	7,393 (16%)	6,638 (14%)
SURGICAL SAME DAY ADMISSION	4,001 (8.6%)	5,543 (12%)
URGENT	8,691 (19%)	6,683 (14%)
Other	1,514 (3.3%)	2,316 (4.8%)
¹ n (%); Median (Q1, Q3)		

Characteristic	TRUE N = 46,337 ¹	FALSE N = 48,107 ¹
admission_location		
EMERGENCY ROOM	17,058 (37%)	20,443 (42%)
PHYSICIAN REFERRAL	11,013 (24%)	12,684 (26%)
TRANSFER FROM HOSPITAL	13,904 (30%)	10,400 (22%)
TRANSFER FROM SKILLED NURSING FACILITY	803 (1.7%)	713 (1.5%)
WALK-IN/SELF REFERRAL	2,169 (4.7%)	2,308 (4.8%)
Other	1,390 (3.0%)	1,559 (3.2%)
discharge_location		
DIED	6,884 (15%)	4,436 (9.4%)
HOME	6,879 (15%)	15,210 (32%)
HOME HEALTH CARE	10,620 (23%)	13,422 (28%)
REHAB	5,574 (12%)	2,445 (5.2%)
SKILLED NURSING FACILITY	8,785 (19%)	7,489 (16%)
Other	7,518 (16%)	4,334 (9.2%)
Unknown	77	771
insurance		
Medicaid	6,768 (15%)	7,469 (16%)
Medicare	26,330 (58%)	25,485 (54%)
No charge	5 (<0.1%)	3 (<0.1%)
Other	1,091 (2.4%)	1,237 (2.6%)
Private	11,515 (25%)	13,018 (28%)
Unknown	628	895
¹ n (%); Median (Q1, Q3)		

Characteristic	TRUE N = 46,337 ¹	FALSE N = 48,107 ¹
language		
American Sign Language	29 (<0.1%)	34 (<0.1%)
Amharic	14 (<0.1%)	9 (<0.1%)
Arabic	87 (0.2%)	62 (0.1%)
Armenian	12 (<0.1%)	13 (<0.1%)
Bengali	22 (<0.1%)	12 (<0.1%)
Chinese	550 (1.2%)	611 (1.3%)
English	41,563 (90%)	43,483 (91%)
French	18 (<0.1%)	14 (<0.1%)
Haitian	375 (0.8%)	252 (0.5%)
Hindi	24 (<0.1%)	21 (<0.1%)
Italian	101 (0.2%)	107 (0.2%)
Japanese	5 (<0.1%)	7 (<0.1%)
Kabuverdianu	301 (0.7%)	345 (0.7%)
Khmer	50 (0.1%)	37 (<0.1%)
Korean	40 (<0.1%)	32 (<0.1%)
Modern Greek (1453-)	102 (0.2%)	88 (0.2%)
Other	152 (0.3%)	153 (0.3%)
Persian	42 (<0.1%)	35 (<0.1%)
Polish	36 (<0.1%)	38 (<0.1%)
Portuguese	351 (0.8%)	314 (0.7%)
Russian	601 (1.3%)	659 (1.4%)
Somali	8 (<0.1%)	15 (<0.1%)

¹ n (%); Median (Q1, Q3)

Characteristic	TRUE N = 46,337 ¹	FALSE N = 48,107 ¹
Spanish	1,472 (3.2%)	1,429 (3.0%)
Thai	21 (<0.1%)	22 (<0.1%)
Vietnamese	151 (0.3%)	129 (0.3%)
Unknown	210	186
marital_status		
DIVORCED	3,377 (8.0%)	3,555 (8.0%)
MARRIED	20,557 (49%)	21,344 (48%)
SINGLE	12,745 (30%)	14,039 (31%)
WIDOWED	5,319 (13%)	5,752 (13%)
Unknown	4,339	3,417
race		
Other	7,802 (17%)	6,689 (14%)
ASIAN	1,369 (3.0%)	1,516 (3.2%)
BLACK	4,933 (11%)	5,452 (11%)
HISPANIC	1,687 (3.6%)	1,908 (4.0%)
WHITE	30,546 (66%)	32,542 (68%)
hospital_expire_flag	6,831 (15%)	4,512 (9.4%)
gender		
F	20,106 (43%)	21,471 (45%)
M	26,231 (57%)	26,636 (55%)
dod	2155-09-06 (2135-07-16, 2175-10-08)	2155-12-18 (2136-04-26, NA)
Unknown	25,846	30,639

¹ n (%); Median (Q1, Q3)

Characteristic	TRUE N = 46,337 ¹	FALSE N = 48,107 ¹
Chloride	102 (98, 105)	102 (98, 105)
Unknown	6,184	5,167
Creatinine	1.00 (0.80, 1.60)	1.00 (0.80, 1.40)
Unknown	4,541	3,486
Sodium	138.0 (135.0, 141.0)	139.0 (136.0, 141.0)
Unknown	6,167	5,163
Potassium	4.20 (3.90, 4.70)	4.20 (3.90, 4.60)
Unknown	6,200	5,187
Glucose	122 (100, 159)	118 (98, 154)
Unknown	6,340	5,314
Hematocrit	35 (29, 40)	36 (30, 41)
Unknown	3,857	2,894
White_Blood_Cells	9.7 (7.0, 13.8)	9.0 (6.6, 12.6)
Unknown	3,906	2,944
Bicarbonate	24.0 (21.0, 27.0)	24.0 (21.0, 27.0)
Unknown	6,272	5,277
systolic_non_invasive_blood_pressure	119 (104, 137)	122 (107, 138)
Unknown	348	1,022
diastolic_non_invasive_blood_pressure	67 (57, 79)	68 (58, 80)
Unknown	351	1,024
respiratory_rate	19.0 (16.0, 23.0)	18.0 (15.0, 22.0)
Unknown	15	183
temperature_fahrenheit	98.20 (97.70, 98.80)	98.10 (97.60, 98.60)

¹ n (%); Median (Q1, Q3)

Characteristic	TRUE N = 46,337 ¹	FALSE N = 48,107 ¹
Unknown	231	1,444
heart_rate	87 (75, 102)	84 (73, 99)
Unknown	1	85
age_intime	67 (56, 77)	66 (54, 77)

¹ n (%); Median (Q1, Q3)

Q1.9 Save the final tibble

Solution: 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")
```

Done.

Close database connection and clear workspace.

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

Done.

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.

Solution: The detailed code for shiny app can be found in the file app.R.