# Biostat 203B Homework 3

**Due February 21st, 2025 @ 11:59PM**

Luke Hodges 906182810

```
sessionInfo()
```

```
R version 4.4.2 (2024-10-31)
Platform: x86_64-apple-darwin20
Running under: macOS Sequoia 15.0

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_4.4.2    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
```

Load necessary libraries

```
library(arrow)
```

```
Attaching package: 'arrow'

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

    timestamp
```

```
library(memuse)
library(pryr)
library(R.utils)
```

```
Loading required package: R.oo

Loading required package: R.methodsS3

R.methodsS3 v1.8.2 (2022-06-13 22:00:14 UTC) successfully loaded. See ?R.methodsS3 for help.

R.oo v1.27.0 (2024-11-01 18:00:02 UTC) successfully loaded. See ?R.oo for help.


Attaching package: 'R.oo'

The following object is masked from 'package:R.methodsS3':

    throw

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

    getClasses, getMethods

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

    attach, detach, load, save

R.utils v2.12.3 (2023-11-18 01:00:02 UTC) successfully loaded. See ?R.utils for help.


Attaching package: 'R.utils'
```

```
The following object is masked from 'package:arrow':

    timestamp

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

    timestamp

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

    cat, commandArgs, getOption, isOpen, nullfile, parse, use, warnings
```

```
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 purrr::compose()      masks pryr::compose()
x lubridate::duration() masks arrow::duration()
x tidyr::extract()      masks R.utils::extract()
x dplyr::filter()       masks stats::filter()
x dplyr::lag()          masks stats::lag()
x purrr::partial()      masks pryr::partial()
x dplyr::where()        masks pryr::where()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

Display your machine memory.

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

```
Totalram:  32.000 GiB
Freeram:   19.159 GiB
```

In this exercise, we use tidyverse (ggplot2, dplyr, etc) to explore the MIMIC-IV data introduced in homework 1 and to build a cohort of ICU stays.

## Question 1:

Q1. Visualizing patient trajectory Visualizing a patient's encounters in a health care system is a common task in clinical data analysis. In this question, we will visualize a patient's ADT (admission-discharge-transfer) history and ICU vitals in the MIMIC-IV data.

## Question 1.1: Graph Duplication

### Reading in the files for Q1

```
patients <- arrow::open_dataset("~/mimic/hosp/patients.csv.gz",
                                format = "csv")

admissions <- arrow::open_dataset("~/mimic/hosp/admissions.csv.gz",
                                  format = "csv")

transfers <- arrow::open_dataset("~/mimic/hosp/transfers.csv.gz",
                                 format = "csv")

procedures_icd <- arrow::open_dataset("~/mimic/hosp/procedures_icd.csv.gz",
                                      format = "csv")

diagnoses_icd <- arrow::open_dataset("~/mimic/hosp/diagnoses_icd.csv.gz",
                                     format = "csv")

d_icd_procedures <- arrow::open_dataset("~/mimic/hosp/d_icd_procedures.csv.gz",
                                        format = "csv")

d_icd_diagnoses <- arrow::open_dataset("~/mimic/hosp/d_icd_diagnoses.csv.gz",
                                       format = "csv")
```

**First, I am going to designate the patient ID so that the TA can change it later and it can be easily filtered. Afterwards, I will then use the subject_id to filter the data sets**

```
subject_id <- 10063848

patients.filter <- patients %>%
  collect() %>%
  dplyr::filter(subject_id == !!subject_id)
```

```r
admissions.filter <- admissions %>%
  dplyr::filter(subject_id == !!subject_id)

transfers.filter <- transfers %>%
  dplyr::filter(subject_id == !!subject_id)

procedures_icd.filter <- procedures_icd %>%
  dplyr::filter(subject_id == !!subject_id)

diagnoses_icd.filter <- diagnoses_icd %>%
  dplyr::filter(subject_id == !!subject_id)

d_icd_procedures.filter <- d_icd_procedures %>%
  dplyr::filter(subject_id == !!subject_id)

d_icd_diagnoses.filter <- d_icd_diagnoses %>%
  dplyr::filter(subject_id == !!subject_id)
```

**Now, I want to make sure that this was correct by looking at the first ten lines of each the files**

**For Patients**

```r
patients10 <- patients.filter %>%
  head(10) %>%
  collect()

print(patients10)
```

```
# A tibble: 1 x 6
  subject_id gender anchor_age anchor_year anchor_year_group dod
       <int> <chr>       <int>       <int> <chr>              <date>
1   10063848 F              75        2177 2017 - 2019       NA
```

**For Admissions**

```r
admissions10 <- admissions.filter %>%
  head(10) %>%
  collect()

print(admissions10)
```

```
# A tibble: 3 x 16
  subject_id hadm_id admittime           dischtime           deathtime
       <int>   <int> <dttm>              <dttm>              <dttm>
1   10063848  2.13e7 2177-07-24 21:29:00 2177-08-06 06:20:00 NA
2   10063848  2.41e7 2177-08-27 19:58:00 2177-09-09 08:00:00 NA
3   10063848  2.69e7 2177-08-17 14:59:00 2177-08-19 10:25:00 NA
# i 11 more variables: 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>, hospital_expire_flag <int>
```

**For Transfers**

```
transfers10 <- transfers.filter %>%
  head(10) %>%
  collect()

print(transfers10)
```

```
# A tibble: 10 x 7
   subject_id  hadm_id transfer_id eventtype careunit        intime
        <int>    <int>       <int> <chr>     <chr>           <dttm>
 1   10063848 21345067    30123135 ED        Emergency Depa~ 2177-07-24 14:40:00
 2   10063848 21345067    31332266 transfer  Surgical Inten~ 2177-07-27 11:24:10
 3   10063848 21345067    32562188 transfer  Med/Surg/Trauma 2177-07-26 13:45:22
 4   10063848 21345067    33836703 transfer  Surgical Inten~ 2177-07-30 03:45:32
 5   10063848 21345067    36311072 admit     Med/Surg/Trauma 2177-07-24 22:16:00
 6   10063848 21345067    37426639 transfer  Med/Surg/Trauma 2177-07-30 03:41:48
 7   10063848 21345067    38029409 transfer  Med/Surg/Trauma 2177-07-30 06:49:58
 8   10063848 21345067    39239596 discharge UNKNOWN         2177-08-06 06:21:01
 9   10063848 24092966    30673893 admit     Medicine        2177-08-27 21:11:00
10   10063848 24092966    32845759 discharge UNKNOWN         2177-09-09 11:12:32
# i 1 more variable: outtime <dttm>
```

**For Procedures_icd**

```
procedures_icd10 <- procedures_icd.filter %>%
  head(10) %>%
  collect()

print(procedures_icd10)
```

```
# A tibble: 6 x 6
  subject_id  hadm_id seq_num chartdate  icd_code icd_version
       <int>    <int>   <int> <date>     <chr>          <int>
1   10063848 21345067       1 2177-07-25 0DB80ZZ           10
2   10063848 21345067       2 2177-07-25 0DN80ZZ           10
3   10063848 21345067       3 2177-08-03 4A023N6           10
4   10063848 21345067       4 2177-07-28 02HV33Z           10
5   10063848 24092966       1 2177-08-29 0W9G30Z           10
6   10063848 24092966       2 2177-09-04 0W9G30Z           10
```

**For Diagnoses_icd**

```
diagnoses_icd10 <- diagnoses_icd.filter %>%
  head(10) %>%
  collect()

print(diagnoses_icd10)
```

```
# A tibble: 10 x 5
   subject_id  hadm_id seq_num icd_code icd_version
        <int>    <int>   <int> <chr>          <int>
 1   10063848 21345067       1 K565              10
 2   10063848 21345067       2 J9601             10
 3   10063848 21345067       3 D680              10
 4   10063848 21345067       4 I272              10
 5   10063848 21345067       5 D6959             10
 6   10063848 21345067       6 K521              10
 7   10063848 21345067       7 I471              10
 8   10063848 21345067       8 N390              10
 9   10063848 21345067       9 D62               10
10   10063848 21345067      10 K567              10
```

**For d_icd_procedures**

```
d_icd_procedures10 <- d_icd_procedures.filter %>%
  head(10) %>%
  collect()

print(d_icd_procedures10)
```

```
# A tibble: 10 x 3
   icd_code icd_version long_title
   <chr>          <int> <chr>
 1 0001               9 Therapeutic ultrasound of vessels of head and neck
 2 0002               9 Therapeutic ultrasound of heart
 3 0003               9 Therapeutic ultrasound of peripheral vascular vessels
 4 0009               9 Other therapeutic ultrasound
 5 001               10 Central Nervous System and Cranial Nerves, Bypass
 6 0010               9 Implantation of chemotherapeutic agent
 7 0011               9 Infusion of drotrecogin alfa (activated)
 8 0012               9 Administration of inhaled nitric oxide
 9 0013               9 Injection or infusion of nesiritide
10 0014               9 Injection or infusion of oxazolidinone class of antibio~
```

**For d_icd_diagnoses**

```
d_icd_diagnoses10 <- d_icd_diagnoses.filter %>%
  head(10) %>%
  collect()

print(d_icd_diagnoses10)
```

```
# A tibble: 10 x 3
   icd_code icd_version long_title
   <chr>          <int> <chr>
 1 0010               9 Cholera due to vibrio cholerae
 2 0011               9 Cholera due to vibrio cholerae el tor
 3 0019               9 Cholera, unspecified
 4 0020               9 Typhoid fever
 5 0021               9 Paratyphoid fever A
 6 0022               9 Paratyphoid fever B
 7 0023               9 Paratyphoid fever C
 8 0029               9 Paratyphoid fever, unspecified
 9 0030               9 Salmonella gastroenteritis
10 0031               9 Salmonella septicemia
```

**All of the code above addresses the other data sets in the mimic folder. However, we still have to look into the labevents folder as well**

**This is specifically for the labevents**

**The first goal, we want to see what parquet we want to use, using BASH, by looking at the first 10 lines**

```
zcat < ~/mimic/hosp/labevents_filtered.csv.gz | head -10
```

```
subject_id,itemid,charttime,valuenum
10000032,50931,2180-03-23 11:51:00,95
10000032,50882,2180-03-23 11:51:00,27
10000032,50902,2180-03-23 11:51:00,101
10000032,50912,2180-03-23 11:51:00,0.4
10000032,50971,2180-03-23 11:51:00,3.7
10000032,50983,2180-03-23 11:51:00,136
10000032,51221,2180-03-23 11:51:00,45.4
10000032,51301,2180-03-23 11:51:00,3
10000032,51221,2180-05-06 22:25:00,42.6
```

**Looking at this, this data is filtered for the adequate columns we need. However, we already created a parquet of the data as well. Let us see if we can look ten lines into the parquet to see which one we should use**

```
file.info("part-0.parquet")$size
```

```
[1] 152917918
```

**This is 152 MB, so this is the parquet used in labevents.filtered folder in the hosp filter within the mimic folder**

```
labevents_pq <- read_parquet("~/mimic/hosp/part-0.parquet")

labevents_pq10 <- labevents_pq %>%
  head(10) %>%
  collect()

print(labevents_pq10)
```

```
# A tibble: 10 x 4
   subject_id itemid charttime           valuenum
        <int>  <int> <dttm>                 <dbl>
 1   10000032  50931 2180-03-23 04:51:00     95
 2   10000032  50882 2180-03-23 04:51:00     27
 3   10000032  50902 2180-03-23 04:51:00    101
 4   10000032  50912 2180-03-23 04:51:00      0.4
 5   10000032  50971 2180-03-23 04:51:00      3.7
```

```
 6   10000032  50983 2180-03-23 04:51:00    136
 7   10000032  51221 2180-03-23 04:51:00     45.4
 8   10000032  51301 2180-03-23 04:51:00      3
 9   10000032  51221 2180-05-06 15:25:00     42.6
10   10000032  51301 2180-05-06 15:25:00      5
```

**Now, let us filter the parquet to go with what we are looking for: subject ID 10001217**

```
labevents_pq.filter <- labevents_pq %>%
  dplyr::filter(subject_id == !!subject_id)
```

**Now, let us look at the first ten lines of the filtered parquet**

```
labevents_pq10.filter <- labevents_pq.filter %>%
  head(10) %>%
  collect()

print(labevents_pq10.filter)
```

```
# A tibble: 10 x 4
   subject_id itemid charttime           valuenum
        <int>  <int> <dttm>                 <dbl>
 1   10063848  51221 2177-07-24 16:45:00     44.2
 2   10063848  51301 2177-07-24 16:45:00     12.2
 3   10063848  50882 2177-07-24 16:45:00     26
 4   10063848  50902 2177-07-24 16:45:00     99
 5   10063848  50912 2177-07-24 16:45:00      0.9
 6   10063848  50931 2177-07-24 16:45:00    130
 7   10063848  50971 2177-07-24 16:45:00      4.1
 8   10063848  50983 2177-07-24 16:45:00    142
 9   10063848  51221 2177-07-25 00:30:00     41.6
10   10063848  51301 2177-07-25 00:30:00      8.6
```

With this in mind, we now want to create a symbolic link to the parquet we created in HW2, part.0.parquet, so that we can use it in this homework.

**Create a symbolic Link for labevents_pq**

cd ~/Desktop/203b-hw/hw3

ln -s ~/mimic/hosp/labevents_pq labevents_pq

ls -l labevents_pq**

```
ls -l ~/Desktop/203b-hw/hw3
```

```
total 13752
-rw-r--r--@ 1 lukehodges  staff  5472513 Feb 20 18:46 HW3.html
-rw-r--r--@ 1 lukehodges  staff    44097 Feb 20 18:46 HW3.qmd
-rw-r--r--@ 1 lukehodges  staff    44527 Feb 20 18:46 HW3.rmarkdown
-rw-r--r--@ 1 lukehodges  staff   635346 Feb 18 13:23 Patient_Vitals_Plot.png
lrwxr-xr-x@ 1 lukehodges  staff       58 Feb 18 18:37 chartevents_pq -> /Users/lukehodges/mi
lrwxr-xr-x@ 1 lukehodges  staff       56 Feb 18 13:44 labevents_pq -> /Users/lukehodges/mimi
lrwxr-xr-x@ 1 lukehodges  staff       43 Feb 20 14:30 part-0.parquet -> /Users/lukehodges/mi
```

**Now, we have to conjoin the necessary tables for the data**

**Since we have the dictionary for the diagnoses and the diagnoses themselves, we can merge them together to figure out what actually happened**

```
LJDiagnoses <- diagnoses_icd.filter %>%
  left_join(d_icd_diagnoses.filter, by = c("icd_code", "icd_version"))

LJDiagnoses10 <- LJDiagnoses %>%
  head(10) %>%
  collect()

print(LJDiagnoses10)
```

```
# A tibble: 10 x 6
   subject_id  hadm_id seq_num icd_code icd_version long_title
        <int>    <int>   <int> <chr>          <int> <chr>
 1   10063848 21345067       1 K565              10 Intestinal adhesions [bands~
 2   10063848 21345067       2 J9601             10 Acute respiratory failure w~
 3   10063848 21345067       3 D680              10 Von Willebrand disease
 4   10063848 21345067       4 I272              10 Other secondary pulmonary h~
 5   10063848 21345067       5 D6959             10 Other secondary thrombocyto~
 6   10063848 21345067       6 K521              10 Toxic gastroenteritis and c~
 7   10063848 21345067       7 I471              10 Supraventricular tachycardia
 8   10063848 21345067       8 N390              10 Urinary tract infection, si~
 9   10063848 21345067       9 D62               10 Acute posthemorrhagic anemia
10   10063848 21345067      10 K567              10 Ileus, unspecified
```

**The top diagnoses were intestinal adhesions with obstruction, acurate respiratory failure with hypoxia, and von willebrand disease for the patient of interest. We**

**have to make sure we count each of the different long_titles and ensure they are able to put into the ggplot and are correct. I received an error regarding an as_vector, so let us set this to true to prevent this happening as well.**

```
LJDiagnosesT3 <- LJDiagnoses %>%
  group_by(long_title) %>%
  summarise(n = n()) %>%
  arrange(desc(n), long_title) %>%
  slice_head(n = 3) %>%
  collect() %>%
  pull(long_title)

LJDiagnosesT3
```

```
[1] "Fistula of intestine"
[2] "Other secondary pulmonary hypertension"
[3] "Unspecified Escherichia coli [E. coli] as the cause of diseases classified elsewhere"
```

**this code now displays the three most common diagnoses, which are Von Willebrand Disease, other secondary pulmonary hypertension and E. Coli**

**Let us now left join the procedures based on the ICD_Code and ICD_version**

```
LJProcedures <- procedures_icd.filter %>%
  left_join(d_icd_procedures, by = c("icd_code", "icd_version"))

LJProcedures10 <- LJProcedures %>%
  head(10) %>%
  collect()

print(LJProcedures10)
```

```
# A tibble: 6 x 7
  subject_id  hadm_id seq_num chartdate  icd_code icd_version long_title
       <int>    <int>   <int> <date>     <chr>          <int> <chr>
1   10063848 21345067       1 2177-07-25 0DB80ZZ           10 Excision of Small~
2   10063848 21345067       2 2177-07-25 0DN80ZZ           10 Release Small Int~
3   10063848 21345067       3 2177-08-03 4A023N6           10 Measurement of Ca~
4   10063848 21345067       4 2177-07-28 02HV33Z           10 Insertion of Infu~
5   10063848 24092966       1 2177-08-29 0W9G30Z           10 Drainage of Perit~
6   10063848 24092966       2 2177-09-04 0W9G30Z           10 Drainage of Perit~
```

**I received an error prior about how this data is not a data.frame. Let us convert these to a data frame to prevent this from happening**

**Created data.frame so that we can use it in ggplot**

```
transfer.filter2 <- as.data.frame(transfers.filter)
LJProcedures <- as.data.frame(LJProcedures)
```

**Changing it so that GGplot can read date and time better**

```
transfer.filter2 <- transfer.filter2 %>%
  mutate(intime = as.POSIXct(intime, format = "%Y-%m-%d %H:%M:%S"),
         outtime = as.POSIXct(outtime, format = "%Y-%m-%d %H:%M:%S"))

LJProcedures <- LJProcedures %>%
  mutate(chartdate = as.POSIXct(chartdate, format = "%Y-%m-%d"))

labevents.filter <- labevents_pq.filter %>%
  collect() %>%
  mutate(charttime = as.POSIXct(charttime, format = "%Y-%m-%d %H:%M:%S"))
```

**We have to make sure we get the title of the graph as well**

**Let us get the patient's info**

```
patient_info <- paste0(
  "Patient ", subject_id, ", ",
  patients.filter$gender, ", ",
  patients.filter$anchor_age, " years old"
)

print(patient_info)
```

```
[1] "Patient 10063848, F, 75 years old"
```

**When we do the regular ggplot, the legend has text that is way too long. Let us wrap this**

**Now let us make each procedure a unique factor so that it can be recognized in ggplot**

```r
library(stringr)

# Apply str_wrap() to wrap text for better legend display
LJProcedures$long_title_wrapped <- str_wrap(LJProcedures$long_title,
                                            width = 17)

# Check result
print(LJProcedures)
```

```
  subject_id  hadm_id seq_num  chartdate icd_code icd_version
1   10063848 21345067       1 2177-07-25  0DB80ZZ          10
2   10063848 21345067       2 2177-07-25  0DN80ZZ          10
3   10063848 21345067       3 2177-08-03  4A023N6          10
4   10063848 21345067       4 2177-07-28  02HV33Z          10
5   10063848 24092966       1 2177-08-29  0W9G30Z          10
6   10063848 24092966       2 2177-09-04  0W9G30Z          10
                                                                  long_title
1                                      Excision of Small Intestine, Open Approach
2                                          Release Small Intestine, Open Approach
3 Measurement of Cardiac Sampling and Pressure, Right Heart, Percutaneous Approach
4      Insertion of Infusion Device into Superior Vena Cava, Percutaneous Approach
5        Drainage of Peritoneal Cavity with Drainage Device, Percutaneous Approach
6        Drainage of Peritoneal Cavity with Drainage Device, Percutaneous Approach
                                                               long_title_wrapped
1                                        Excision of Small\nIntestine, Open\nApproach
2                                            Release Small\nIntestine, Open\nApproach
3 Measurement of\nCardiac Sampling\nand Pressure,\nRight Heart,\nPercutaneous\nApproach
4      Insertion of\nInfusion Device\ninto Superior\nVena Cava,\nPercutaneous\nApproach
5        Drainage of\nPeritoneal\nCavity with\nDrainage Device,\nPercutaneous\nApproach
6        Drainage of\nPeritoneal\nCavity with\nDrainage Device,\nPercutaneous\nApproach
```

**Now let us make each procedure a unique factor so that it can be recognized in ggplot. By setting the names we are making sure that each of them are made into different shapes and colors**

```r
ProceduresUnique <- unique(LJProcedures$long_title_wrapped)
Shapemanual <- setNames((seq_along(ProceduresUnique) %%
                         5) + 16, ProceduresUnique)
Colormanual <- setNames(seq_along(transfer.filter2$careunit) %%
                         25 + 1, transfer.filter2$careunit)
```

**Now, let us make the ggplot**

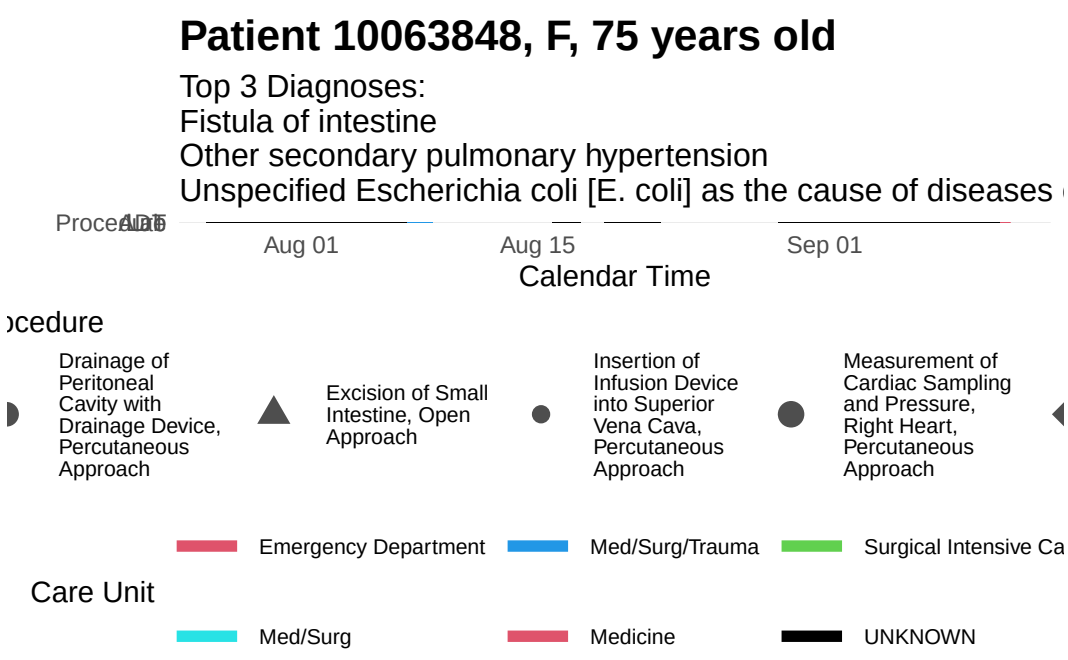
```
ggplot() +
 # Procedures (Different Shapes)
  geom_point(data = LJProcedures,
             aes(x = chartdate, y = "Procedure",
                 shape = long_title_wrapped),
             color = "black",
             size = 4,
             alpha = 0.7,
             position = position_jitter(width = 0, height = -0.5)) +

  # ADT Events (Colored by Care Unit)
  geom_segment(data = transfer.filter2,
               aes(x = intime, xend = outtime, y = "ADT",
                   color = careunit),
               size = 2)  +

  # Lab Events (Black crosses `+`)
  geom_point(data = labevents.filter,
             aes(x = charttime, y = "Lab"),
             shape = 3, size = 3, color = "black") +
  labs(
    y = "",
    x = "Calendar Time",
    title = patient_info,
    subtitle = paste0("Top 3 Diagnoses:\n",
                      paste(LJDiagnosesT3, collapse = "\n")),
    color = "Care Unit",
    shape = "Procedure"
  ) +
guides(
    color = guide_legend(title = "Care Unit", nrow = 2,
                         title.position = "left"),
    shape = guide_legend(title = "Procedure", nrow = 1,
                         title.position = "top"),
    override.aes = list(size=3))+
 scale_color_manual(values = Colormanual) +
  scale_shape_manual(values = Shapemanual) +
  scale_fill_manual(values = Shapemanual) +
  scale_y_discrete(limits = rev) +

  theme_minimal() +
  theme(
```

```
    plot.title = element_text(size = 16, face = "bold"),
    plot.subtitle = element_text(size = 12),
    axis.text.x = element_text(angle = 0, hjust = 1),
    legend.position = "bottom",
    legend.box = "vertical",
    legend.key.size = unit(1, "cm"),
    legend.text = element_text(size = 7.5),
    plot.margin = margin(5, 5, 5, 5),
    legend.spacing.y = unit(0.01, "cm"),
    legend.box.spacing = unit(0.01, "cm")
  )
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

```
Warning: No shared levels found between `names(values)` of the manual scale and the
data's fill values.
```

```
Warning: Removed 3 rows containing missing values or values outside the scale range
(`geom_segment()`).
```



### 1.2 Question Graph Duplication

**First, let us make sure that the chartevents.csv is read into the system and confirm that we have the right directory for it**

```
#| eval: false

zcat < ~/mimic/icu/chartevents.csv.gz > ~/mimic/icu/chartevents.csv
```

```
ls -l ~/mimic/icu/chartevents.csv
```

```
-rw-r--r--@ 1 lukehodges  staff  41935806083 Feb 20 18:47 /Users/lukehodges/mimic/icu/charte
```

**We can now open the data set**

```
chartevents.arrow <- arrow::open_dataset("~/mimic/icu/chartevents.csv",
                                         format = "csv")
```

**And filter it based on the information we would like: 220045, 220181, 220179, 223761, and 22010**

```
library(dplyr
        )
chartevents.filtered.arrow <- chartevents.arrow %>%
  dplyr::select(subject_id, itemid, charttime, valuenum) %>%
  dplyr::filter(itemid %in% c(220045, 220181, 220179, 223761, 220210)) %>%
  dplyr::filter(subject_id == !!subject_id)
```

**Let us make sure that everything looks correct**

```
charteventsprint <- chartevents.filtered.arrow %>%
head(50) %>%
collect()

print(charteventsprint)
```

```
# A tibble: 50 x 4
   subject_id itemid charttime           valuenum
        <int>  <int> <dttm>                 <dbl>
```

```
 1   10063848 220045 2177-07-29 14:00:00       93
 2   10063848 220210 2177-07-29 14:00:00       23
 3   10063848 220179 2177-07-29 14:02:00       97
 4   10063848 220181 2177-07-29 14:02:00       65
 5   10063848 220045 2177-07-27 12:00:00      153
 6   10063848 220210 2177-07-27 12:00:00       25
 7   10063848 220179 2177-07-27 12:02:00      129
 8   10063848 220181 2177-07-27 12:02:00       89
 9   10063848 220045 2177-07-27 13:00:00       97
10   10063848 220210 2177-07-27 13:00:00       28
# i 40 more rows
```

**Like before, we need to make sure that we make the data frame so that ggplot can read it and that the dates and time are consistent**

```
chartevents.filtered.arrow <- as.data.frame(chartevents.filtered.arrow)
chartevents.filtered.arrow$charttime <- as.POSIXct(
  chartevents.filtered.arrow$charttime, format = "%Y-%m-%d %H:%M:%S")
```

**Received an error saying that the values are not a factor. So let us make it one**

```
chartevents.filtered.arrow$subject_id <- as.factor(
  chartevents.filtered.arrow$subject_id)
```

**Let us make sure that the above command worked**

```
colnames(chartevents.filtered.arrow)
```

```
[1] "subject_id" "itemid"     "charttime"  "valuenum"
```

**Reading in the ICUStays and then filtering it for the ID given**

```
zcat < ~/mimic/icu/icustays.csv.gz > ~/mimic/icu/icustays.csv
```

```
icustays <- arrow::open_dataset("~/mimic/icu/icustays.csv", format = "csv")
```

**I want to make sure that all the data sets are read in correctly after filtering**

**We can now do the same thing but this time filter the icustays with the correct subject_id**

```
icustays.filtered <- icustays %>%
  collect() %>%
  dplyr::filter(subject_id == !!subject_id)
```

**Now we have to ensure that both of the factors are the same so that they can be graphed**

```
icustays.filtered10 <- icustays.filtered %>%
  head(10) %>%
  collect()

print(icustays.filtered10)
```

```
# A tibble: 2 x 8
  subject_id  hadm_id  stay_id first_careunit  last_careunit intime
       <int>    <int>    <int> <chr>           <chr>         <dttm>
1   10063848 21345067 31332266 Surgical Inten~ Surgical Int~ 2177-07-27 11:24:10
2   10063848 21345067 33836703 Surgical Inten~ Surgical Int~ 2177-07-30 03:45:32
# i 2 more variables: outtime <dttm>, los <dbl>
```

```
chartevents.filtered.arrow$subject_id <- as.integer(
  as.character(chartevents.filtered.arrow$subject_id))

icustays.filtered$subject_id <- as.integer(
  icustays.filtered$subject_id)
```

**To make it universal, we will have to figure out what the minimum and maximum charttimes are for both the stay_ids. We do the as.POSIXct as that is the error I got that I needed to fix**

```
min_datetime1 <- as.POSIXct(icustays.filtered$intime[1])
max_datetime1 <- as.POSIXct(icustays.filtered$outtime[1])

min_datetime2 <- as.POSIXct(icustays.filtered$intime[2])
max_datetime2 <- as.POSIXct(icustays.filtered$outtime[2])

min_datetime1
```

```
[1] "2177-07-27 11:24:10 PDT"
```

```
max_datetime1
```

```
[1] "2177-07-30 03:41:48 PDT"
```

```
min_datetime2
```

```
[1] "2177-07-30 03:45:32 PDT"
```

```
max_datetime2
```

```
[1] "2177-07-30 06:49:58 PDT"
```

**We are making it so that if the time falls between min and max 1, it is assigned the first stay_id. If it falls between min 2 and max 2, then it is assigned the second**

```
chartevents.filtered.arrow <- chartevents.filtered.arrow %>%
  mutate(stay_id = case_when(
    charttime >= min_datetime1 &
      charttime <= max_datetime1 ~ icustays.filtered$stay_id[1],
    charttime >= min_datetime2 &
      charttime <= max_datetime2 ~ icustays.filtered$stay_id[2],
    TRUE ~ NA_real_
  ))

chartevents.filtered.arrow10 <- chartevents.filtered.arrow %>%
  head(10) %>%
  collect()

print(chartevents.filtered.arrow10)
```

```
   subject_id itemid           charttime valuenum  stay_id
1    10063848 220045 2177-07-29 14:00:00       93 31332266
2    10063848 220210 2177-07-29 14:00:00       23 31332266
3    10063848 220179 2177-07-29 14:02:00       97 31332266
4    10063848 220181 2177-07-29 14:02:00       65 31332266
5    10063848 220045 2177-07-27 12:00:00      153 31332266
6    10063848 220210 2177-07-27 12:00:00       25 31332266
7    10063848 220179 2177-07-27 12:02:00      129 31332266
```

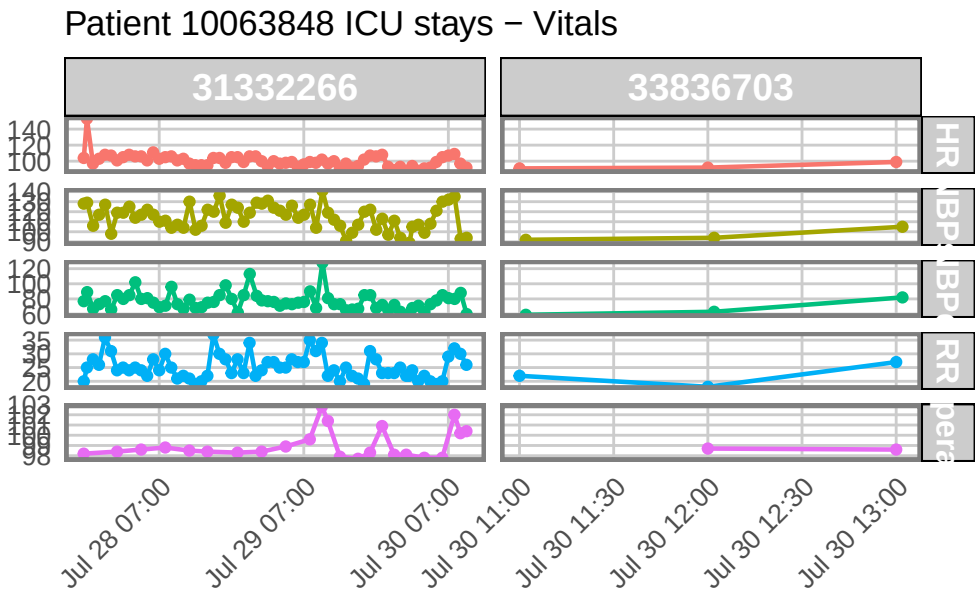
```
8    10063848 220181 2177-07-27 12:02:00       89 31332266
9    10063848 220045 2177-07-27 13:00:00       97 31332266
10   10063848 220210 2177-07-27 13:00:00       28 31332266
```

```
stay_ranges <- chartevents.filtered.arrow %>%
  group_by(stay_id) %>%
  summarise(min_time = min(charttime), max_time = max(charttime)) %>%
  ungroup()

full_min_time <- min(stay_ranges$min_time)
full_max_time <- max(stay_ranges$max_time)
```

**With all the data in one table, we can not make the ggplot**

```
ggplot(chartevents.filtered.arrow,
       aes(x = charttime, y = valuenum, color = factor(itemid))) +
  geom_line(size = 0.8) +
  geom_point(size = 1.5) +
  facet_grid(itemid ~ stay_id, scales = "free", space = "fixed",
             labeller = labeller(itemid = c(
    "220045" = "HR",
    "220181" = "NBPd",
    "220179" = "NBPs",
    "220210" = "RR",
    "223761" = "Temperature"
  ))) +
  labs(
    title = paste("Patient", unique(chartevents.filtered.arrow$subject_id),
                  "ICU stays - Vitals"),
    x = "",
    y = "",
    color = "Vital Type"
  ) +
  scale_x_datetime(date_labels = "%b %d %H:%M") +
  theme_minimal() +
  theme(
    strip.text.x = element_text(size = 14, face = "bold", color = "white"),
    strip.text.y = element_text(size = 12, face = "bold", color = "white"),
    strip.background = element_rect(fill = "grey80"),
    panel.grid.major = element_line(color = "gray80"),
    panel.grid.minor = element_blank(),
    axis.text.x = element_text(angle = 45, hjust = 1, size = 10),
```

```
    axis.text.y = element_text(size = 10),
    legend.position = "none",
    plot.margin = margin(10, 10, 10, 10),
    panel.border = element_rect(color = "grey50", fill = NA, linewidth = 1.5)
  )
```




## Q2. ICU stays

icustays.csv.gz (https://mimic.mit.edu/docs/iv/modules/icu/icustays/) contains data about Intensive Care Units (ICU) stays. The first 10 lines are

```
zcat < ~/mimic/icu/icustays.csv.gz | head
```

```
subject_id,hadm_id,stay_id,first_careunit,last_careunit,intime,outtime,los
10000032,29079034,39553978,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (MI
10000690,25860671,37081114,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (MI
10000980,26913865,39765666,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (MI
10001217,24597018,37067082,Surgical Intensive Care Unit (SICU),Surgical Intensive Care Unit
10001217,27703517,34592300,Surgical Intensive Care Unit (SICU),Surgical Intensive Care Unit
10001725,25563031,31205490,Medical/Surgical Intensive Care Unit (MICU/SICU),Medical/Surgical
10001843,26133978,39698942,Medical/Surgical Intensive Care Unit (MICU/SICU),Medical/Surgical
```

```
10001884,26184834,37510196,Medical Intensive Care Unit (MICU),Medical Intensive Care Unit (MI
10002013,23581541,39060235,Cardiac Vascular Intensive Care Unit (CVICU),Cardiac Vascular Inte
```

### Q2.1 Ingestion

Import icustays.csv.gz as a tibble icustays_tble.

```
icustays_arrow <- arrow::open_dataset("~/mimic/icu/icustays.csv",
                                      format = "csv")

icustays_tble <- icustays_arrow %>%
  collect() %>%  # Pulls data into memory
  as_tibble()

glimpse(icustays_tble)
```

```
Rows: 94,458
Columns: 8
$ subject_id     <int> 10000032, 10000690, 10000980, 10001217, 10001217, 10001~
$ hadm_id        <int> 29079034, 25860671, 26913865, 24597018, 27703517, 25563~
$ stay_id        <int> 39553978, 37081114, 39765666, 37067082, 34592300, 31205~
$ first_careunit <chr> "Medical Intensive Care Unit (MICU)", "Medical Intensiv~
$ last_careunit  <chr> "Medical Intensive Care Unit (MICU)", "Medical Intensiv~
$ intime         <dttm> 2180-07-23 07:00:00, 2150-11-02 11:37:00, 2189-06-27 0~
$ outtime        <dttm> 2180-07-23 16:50:47, 2150-11-06 09:03:17, 2189-06-27 1~
$ los            <dbl> 0.4102662, 3.8932523, 0.4975347, 1.1180324, 0.9481134, ~
```

### Q2.2 Summary and visualization

How many unique subject_id? Can a subject_id have multiple ICU stays? Summarize the number of ICU stays per subject_id by graphs.

**First let us find out how many unique subject_id there are**

```
uniqueicu <- icustays_tble %>%
  distinct(subject_id) %>%
  nrow()

print(uniqueicu)
```

```
[1] 65366
```

**So there are 65366 unique subject_ids**

**Now let us check to see if a subject_id have multiple stays in the intensive care unit**

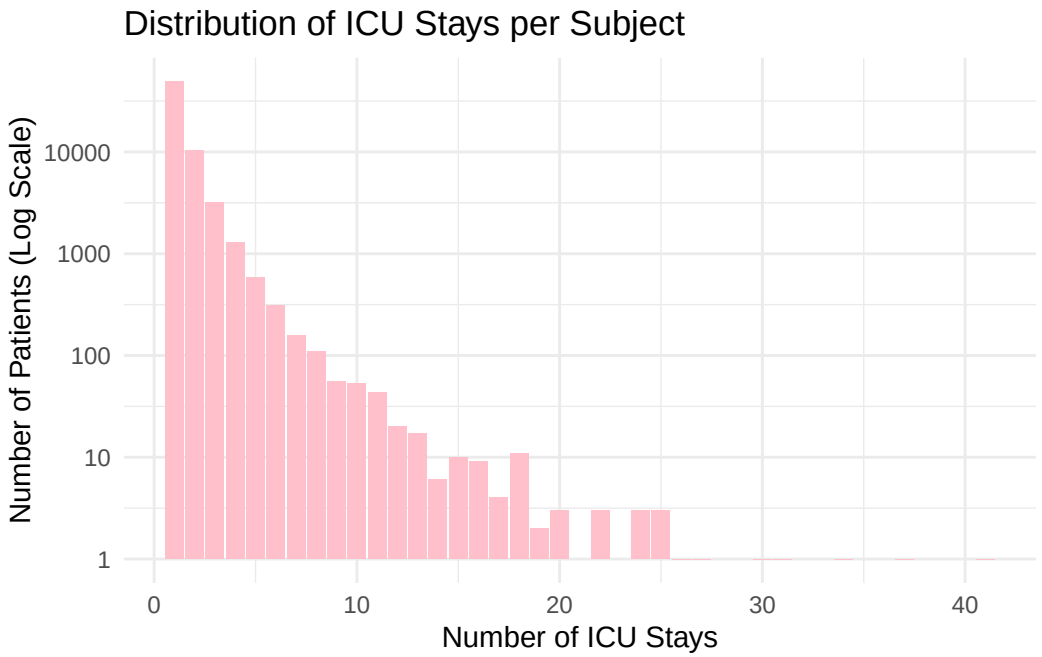
```
icustayscount <- icustays_tble %>%
  group_by(subject_id) %>%
  summarise(n = n()) %>%
  ungroup()

maxstays <- max(icustayscount$n)
print(maxstays)
```

```
[1] 41
```

**The answer is 41, so some subjects DO have multiple stays in the intensive care unit. The most being one subject having 41 stays**

**We can make a bar chart to illustrate the amount of icu stays per subject. I chose to do a log-scale because the bart chart makes the data hard to see since it is highly skewed**

```
ggplot(icustayscount, aes(x = n)) +
  geom_bar(fill = "pink") +
  scale_y_log10() +
  labs(
    title = "Distribution of ICU Stays per Subject",
    x = "Number of ICU Stays",
    y = "Number of Patients (Log Scale)"
  ) +
  theme_minimal()
```




**This shows that the main frequency of subjects only have one stay in the intensive care unit**

## Q3 Admissions Data

Information of the patients admitted into hospital is available in admissions.csv.gz. See https://mimic.mit.edu/docs/iv/modules/hosp/admissions/ for details of each field in this file. The first 10 lines are

```
zcat < ~/mimic/hosp/admissions.csv.gz | head
```

```
subject_id,hadm_id,admittime,dischtime,deathtime,admission_type,admit_provider_id,admission_l
10000032,22595853,2180-05-06 22:23:00,2180-05-07 17:15:00,,URGENT,P49AFC,TRANSFER FROM HOSPI
10000032,22841357,2180-06-26 18:27:00,2180-06-27 18:49:00,,EW EMER.,P784FA,EMERGENCY ROOM,HO
10000032,25742920,2180-08-05 23:44:00,2180-08-07 17:50:00,,EW EMER.,P19UTS,EMERGENCY ROOM,HOS
10000032,29079034,2180-07-23 12:35:00,2180-07-25 17:55:00,,EW EMER.,P06OTX,EMERGENCY ROOM,HO
10000068,25022803,2160-03-03 23:16:00,2160-03-04 06:26:00,,EU OBSERVATION,P39NWO,EMERGENCY RO
10000084,23052089,2160-11-21 01:56:00,2160-11-25 14:52:00,,EW EMER.,P42H7G,WALK-IN/SELF REFE
10000084,29888819,2160-12-28 05:11:00,2160-12-28 16:07:00,,EU OBSERVATION,P35NE4,PHYSICIAN R
10000108,27250926,2163-09-27 23:17:00,2163-09-28 09:04:00,,EU OBSERVATION,P40JML,EMERGENCY RO
10000117,22927623,2181-11-15 02:05:00,2181-11-15 14:52:00,,EU OBSERVATION,P47EY8,EMERGENCY RO
```

## Q3.1 Ingestion

Import admissions.csv.gz as a tibble admissions_tble.

```
admissions_tble <- admissions %>%
  collect() %>%
  as_tibble()

glimpse(admissions_tble)
```

```
Rows: 546,028
Columns: 16
$ subject_id          <int> 10000032, 10000032, 10000032, 10000032, 10000068,~
$ hadm_id             <int> 22595853, 22841357, 25742920, 29079034, 25022803,~
$ admittime           <dttm> 2180-05-06 15:23:00, 2180-06-26 11:27:00, 2180-0~
$ dischtime           <dttm> 2180-05-07 10:15:00, 2180-06-27 11:49:00, 2180-0~
$ deathtime           <dttm> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
$ admission_type      <chr> "URGENT", "EW EMER.", "EW EMER.", "EW EMER.", "EU~
$ admit_provider_id   <chr> "P49AFC", "P784FA", "P19UTS", "P06OTX", "P39NWO",~
$ admission_location  <chr> "TRANSFER FROM HOSPITAL", "EMERGENCY ROOM", "EMER~
$ discharge_location  <chr> "HOME", "HOME", "HOSPICE", "HOME", "", "HOME HEAL~
$ insurance           <chr> "Medicaid", "Medicaid", "Medicaid", "Medicaid", "~
$ language            <chr> "English", "English", "English", "English", "Engl~
$ marital_status      <chr> "WIDOWED", "WIDOWED", "WIDOWED", "WIDOWED", "SING~
$ race                <chr> "WHITE", "WHITE", "WHITE", "WHITE", "WHITE", "WHI~
$ edregtime           <dttm> 2180-05-06 12:17:00, 2180-06-26 08:54:00, 2180-0~
$ edouttime           <dttm> 2180-05-06 16:30:00, 2180-06-26 14:31:00, 2180-0~
$ hospital_expire_flag <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

## Q3.2 Summary and visualization

Summarize the following information by graphics and explain any patterns you see.

number of admissions per patient admission hour (anything unusual?) admission minute (anything unusual?) length of hospital stay (from admission to discharge) (anything unusual?)

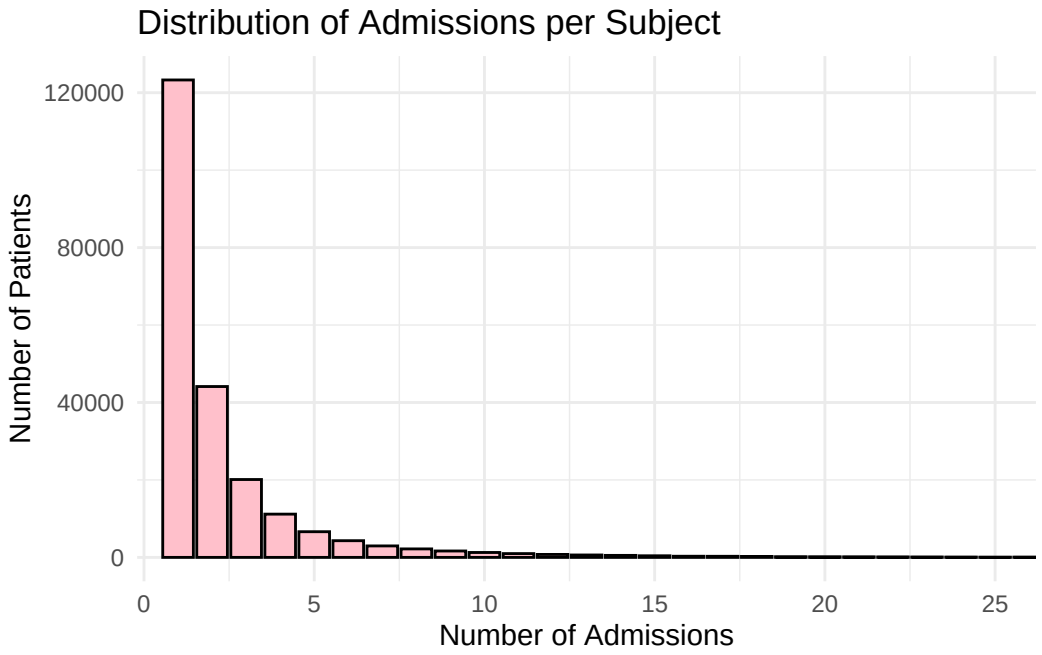
**let us start by making a graph of the number of admissions per patient**

**Before we make the graph, we need to summarize the data for the subject_ids and group them**

```r
admissionscount <- admissions_tble %>%
  group_by(subject_id) %>%
  summarise(n = n()) %>%
  ungroup()
```

**Now we can make the graph**

```r
ggplot(admissionscount, aes(x = n)) +
  geom_bar(fill = "pink", color = "black") +
  labs(
    title = "Distribution of Admissions per Subject",
    x = "Number of Admissions",
    y = "Number of Patients"
  ) +
  coord_cartesian(xlim = c(1, 25)) +
  theme_minimal()
```

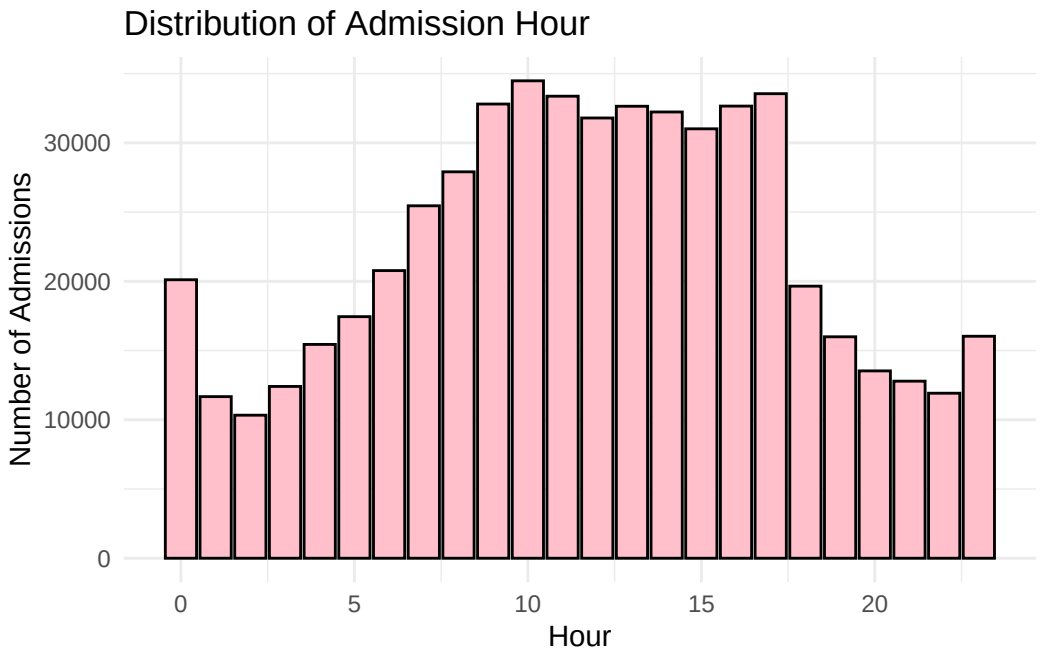



**I do not see any patterns that occur. If anything, the only pattern here is that many of the subjects only are admitted once, while some of admitted more than once, however, this becomes increasingly rare. The reason why this may occur is because the subjects are coming in for more curable/fixable issues and are leaving with a solution. Unless the patient has an ongoing disease that requires**

**consistent maintenence into the hospital, this would be why most of them are admitted once.**

**Now let us make a graph summarizing the admission hour (anything unusual?)**

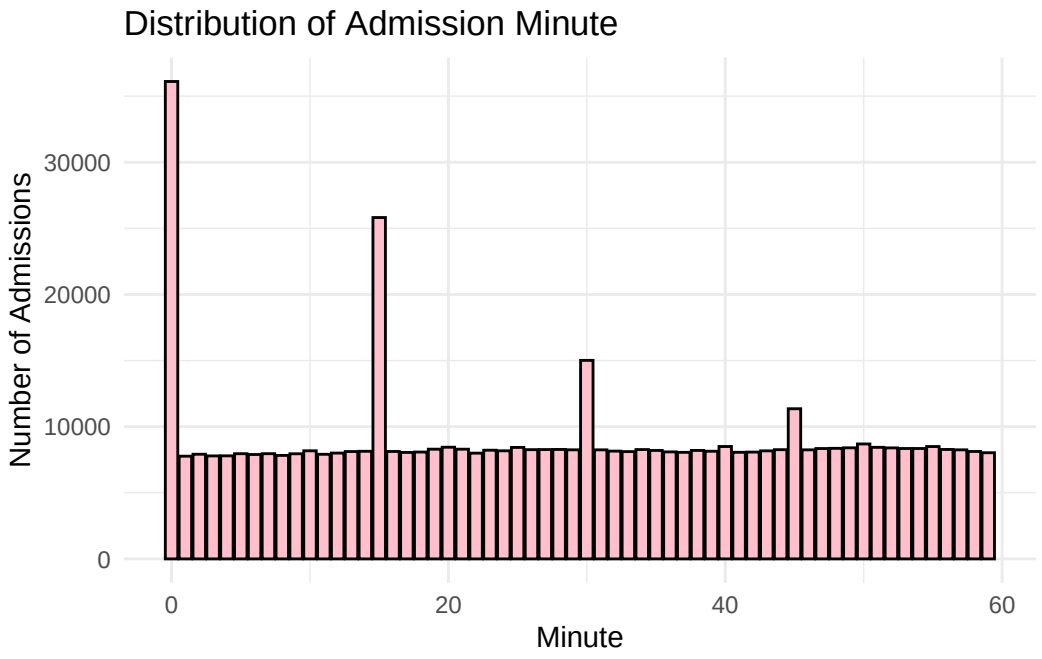
```
ggplot(admissions_tble, aes(x = hour(admittime))) +
  geom_bar(fill = "pink", color = "black") +
  labs(
    title = "Distribution of Admission Hour",
    x = "Hour",
    y = "Number of Admissions"
  ) +
  theme_minimal()
```




**This graph looks at specifically the hour at which the admissions occur. What is interesting about this graph is that most of the admissions occur at the 12th hour, which is noon. This is interesting as most people would think that admissions would occur in the morning, but this is not the case. On top of this, the admissions in the 23rd and 0th hour are pretty high compared to their neighboring hours. This would make sense since some illnesses, like asthma, a disease that I suffer from, are more likely to get worse at night. Individuals may also try and manage their symptoms throughout the day, until they realize they actually cannot, which they would make this decision before going to bed (11 pm to 12 am)**

**Now let us make a graph summarizing the admission minute (anything unusual?)**

```
ggplot(admissions_tble, aes(x = minute(admittime))) +
  geom_bar(fill = "pink", color = "black") +
  labs(
    title = "Distribution of Admission Minute",
    x = "Minute",
    y = "Number of Admissions"
  ) +
  theme_minimal()
```




**What is unusual about this is that most of the admissions occurred at each quarter of an hour (0 minutes, 15 minutes, 30 minutes, and 45 minutes). On top of this minute 60 has no admissions in them, which would make sense as when minute sixty hits, it is minute zero of the new hour. The reason why this occurs is because nurses or doctors may be rounding their admission minute to the nearest quarter hour to make it easier to keep track of the patients and because of convenience**

**Now let us make a graph summarizing the length of hospital stay (from admission to discharge) (anything unusual?)**

**Making sure the admittime can be read in correct format**

```
admissions_tble <- admissions_tble %>%
  mutate(admittime = as.POSIXct(admittime, format = "%Y-%m-%d %H:%M:%S"),
         dischtime = as.POSIXct(dischtime, format = "%Y-%m-%d %H:%M:%S"))
```

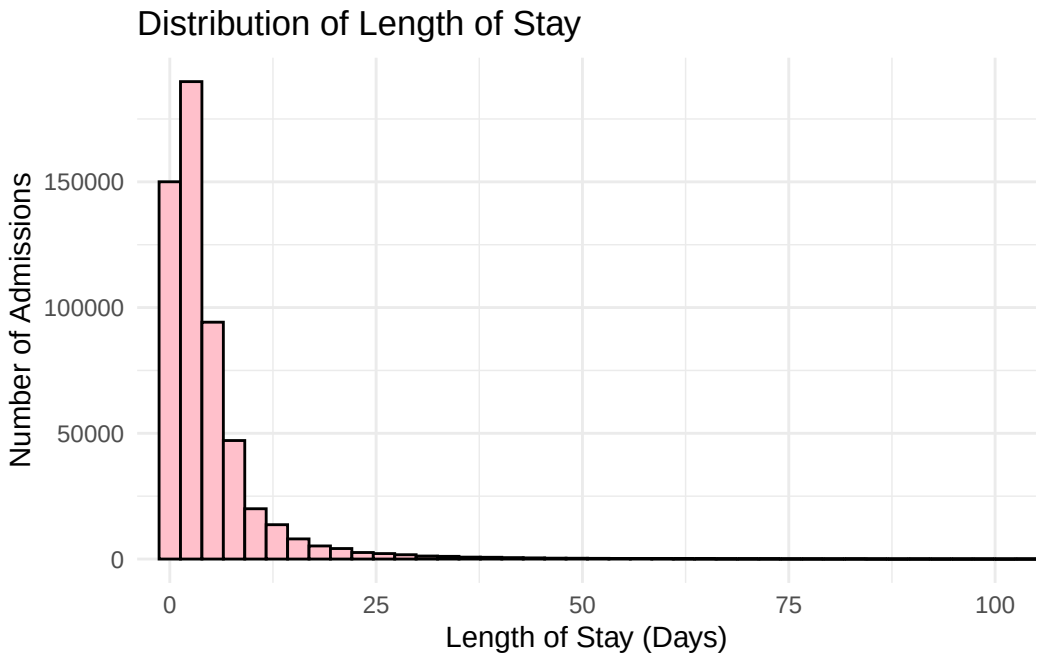
**Mutating the data information so that length_of_stay is in the units of days for both the admittime and the discharge time**

```
admissions_tble <- admissions_tble %>%
  mutate(length_of_stay = difftime(dischtime, admittime, units = "days"))
```

**Graphing the data now**

```
ggplot(admissions_tble, aes(x = length_of_stay)) +
  geom_histogram(fill = "pink", color = "black", bins = 200) +
  labs(
    title = "Distribution of Length of Stay",
    x = "Length of Stay (Days)",
    y = "Number of Admissions"
  ) +
  coord_cartesian(xlim = c(1, 100)) +
  theme_minimal()
```

```
Don't know how to automatically pick scale for object of type <difftime>.
Defaulting to continuous.
```




**Looking at this chart, there really is nothing unusual that happens. I think the one thing here that stands out is the fact that individuals admitted usually spend more than one day rather than being discharged same day. However, this does make sense as many hospitals want to make sure that you are okay and healthy before you are discharged. In other words, staying over night for observation and to make sure there are no complications would explain why individuals stay for longer days rather than being discharged same day**

### Q4.1 Ingestion

Import patients.csv.gz (https://mimic.mit.edu/docs/iv/modules/hosp/patients/) as a tibble patients_tble.

```
patients_tble <- patients %>%
  collect() %>%
  as_tibble()

glimpse(patients_tble)
```

```
Rows: 364,627
Columns: 6
$ subject_id        <int> 10000032, 10000048, 10000058, 10000068, 10000084, 10~
```

```
$ gender            <chr> "F", "F", "F", "F", "M", "F", "M", "M", "F", "M", "F~
$ anchor_age        <int> 52, 23, 33, 19, 72, 27, 25, 24, 48, 60, 59, 34, 20, ~
$ anchor_year       <int> 2180, 2126, 2168, 2160, 2160, 2136, 2163, 2154, 2174~
$ anchor_year_group <chr> "2014 - 2016", "2008 - 2010", "2020 - 2022", "2008 -~
$ dod               <date> 2180-09-09, NA, NA, NA, 2161-02-13, NA, NA, NA, NA,~
```

### Q4.2 Summary and visualization

Summarize variables gender and anchor_age by graphics, and explain any patterns you see.

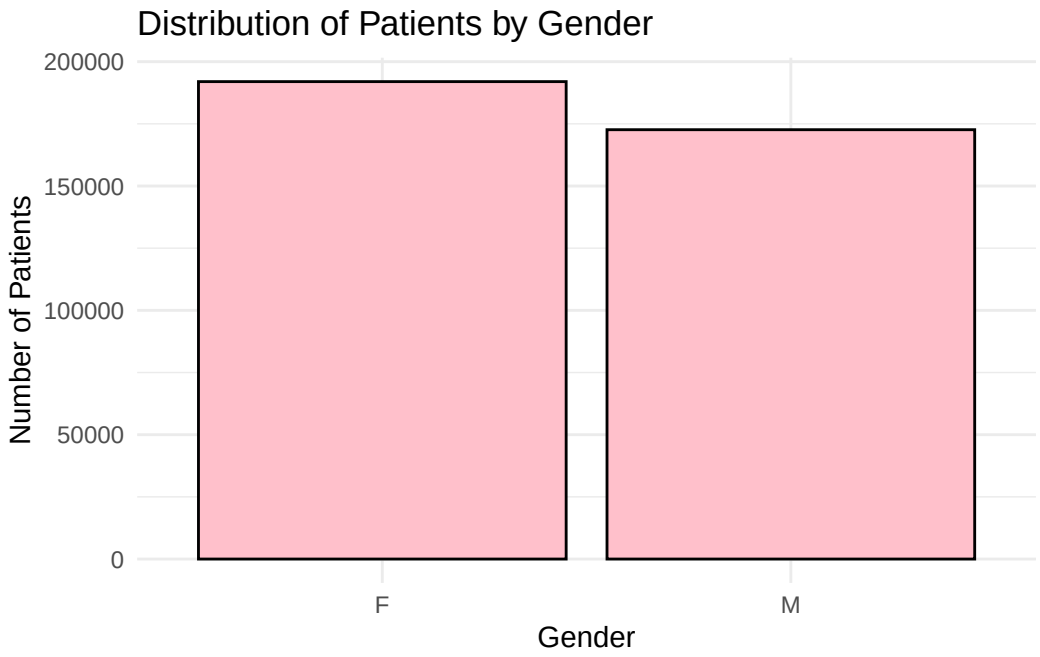
**Let us start by making a graph of of the variable gender first**

**We have to make gender a factor since I get an error when I read it into ggplot**

```
patients_tble <- patients_tble %>%
  mutate(gender = as.factor(gender))
```

**this will be making the graph**

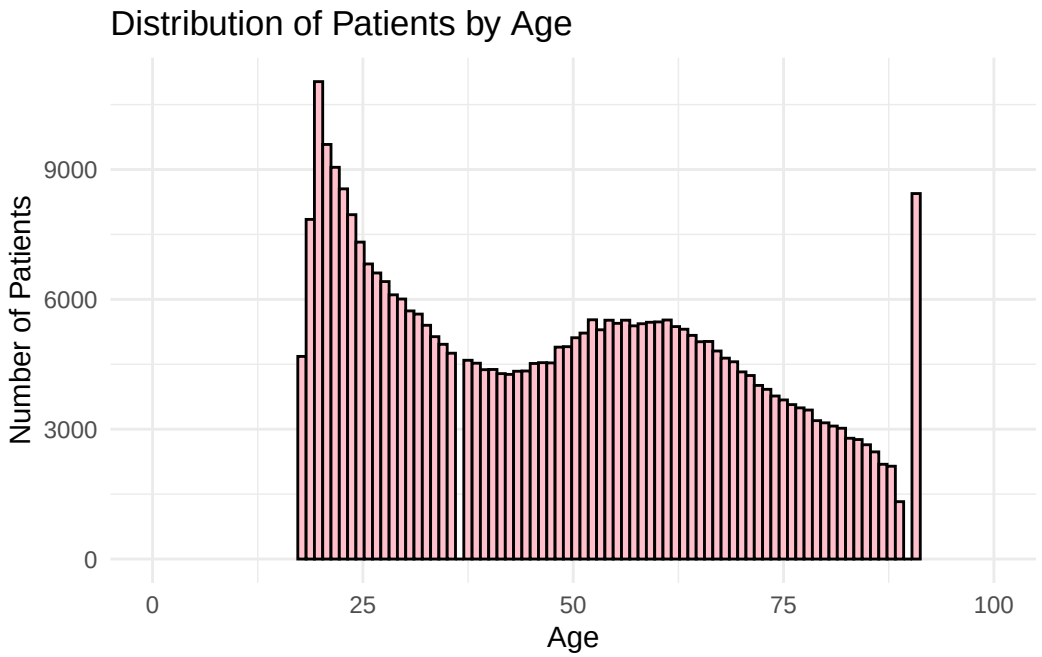
```
ggplot(patients_tble, aes(x = gender)) +
  geom_bar(fill = "pink", color = "black") +
  labs(
    title = "Distribution of Patients by Gender",
    x = "Gender",
    y = "Number of Patients"
  ) +
  theme_minimal()
```




**It looks like most of the patients that show up identify as female for their gender. This makes sense considering that women have to go more to the doctors to check up on reproductive health and yearly screenings for cancer depending on their age**

**Now we have to do it by age**

```
ggplot(patients_tble, aes(x = anchor_age)) +
  geom_histogram(fill = "pink", color = "black", bins = 75) +
  labs(
    title = "Distribution of Patients by Age",
    x = "Age",
    y = "Number of Patients"
  ) +
  coord_cartesian(xlim = c(0, 100))+
  theme_minimal()
```




**This data is indicative of age-rounding, where either the patient or the practitioner rounds the age of the patient either up or down. There is also no data before about 20 years of age, and past 87.5 years of age. As we know, these ages or possible so these are not being reported. We also see that at one point there is data missing completely, which could be more indicative of age rounding as well or not taking into consideration age for the patient when examining them. The large amount of young adult admissions could be because of the onset of diseases that do not show up until young adulthood, like diabetes, MS, or schizophrenia. Individuals of this age are also still on their parent's health insurance, so they may be more likely to go visit the hospital since it would not be on their dime. This then explains the dip in the 30s: individuals are on their own health insurance now and have to be more financially conscientious since their insurance is likely expensive. These adults are also more likely to be healthy compared to other age ranges. The last peak is quiet indicative of a cutoff or even an issue with data entry.**

### Q5 Lab results

labevents.csv.gz (https://mimic.mit.edu/docs/iv/modules/hosp/labevents/) contains all laboratory measurements for patients. The first 10 lines are

```
zcat < ~/mimic/hosp/labevents.csv.gz | head
```

```
labevent_id,subject_id,hadm_id,specimen_id,itemid,order_provider_id,charttime,storetime,value
1,10000032,,2704548,50931,P69FQC,2180-03-23 11:51:00,2180-03-23 15:56:00,___,95,mg/dL,70,100
2,10000032,,36092842,51071,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,
3,10000032,,36092842,51074,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,
4,10000032,,36092842,51075,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,"
5,10000032,,36092842,51079,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,
6,10000032,,36092842,51087,P69FQC,2180-03-23 11:51:00,,,,,,,ROUTINE,RANDOM.
7,10000032,,36092842,51089,P69FQC,2180-03-23 11:51:00,2180-03-23 16:15:00,,,,,,ROUTINE,PRESU
8,10000032,,36092842,51090,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,MI
9,10000032,,36092842,51092,P69FQC,2180-03-23 11:51:00,2180-03-23 16:00:00,NEG,,,,,ROUTINE,"C
```

d_labitems.csv.gz (https://mimic.mit.edu/docs/iv/modules/hosp/d_labitems/) is the dictionary of lab measurements.

```
zcat < ~/mimic/hosp/d_labitems.csv.gz | head
```

```
itemid,label,fluid,category
50801,Alveolar-arterial Gradient,Blood,Blood Gas
50802,Base Excess,Blood,Blood Gas
50803,"Calculated Bicarbonate, Whole Blood",Blood,Blood Gas
50804,Calculated Total CO2,Blood,Blood Gas
50805,Carboxyhemoglobin,Blood,Blood Gas
50806,"Chloride, Whole Blood",Blood,Blood Gas
50808,Free Calcium,Blood,Blood Gas
50809,Glucose,Blood,Blood Gas
50810,"Hematocrit, Calculated",Blood,Blood Gas
```

We are interested in the lab measurements of creatinine (50912), potassium (50971), sodium (50983), chloride (50902), bicarbonate (50882), hematocrit (51221), white blood cell count (51301), and glucose (50931). Retrieve a subset of labevents.csv.gz that only containing these items for the patients in icustays_tble. Further restrict to the last available measurement (by storetime) before the ICU stay. The final labevents_tble should have one row per ICU stay and columns for each lab measurement.

**First, let us create the labevents file**

```
labevents <- arrow::open_dataset("~/mimic/hosp/labevents.csv", format = "csv")
```

**Now let us make it a parquet so we can make the directory after**

```
arrow::write_dataset(labevents, path = "~/mimic/hosp/labevents_pq", format = "parquet")
```

**The symbolic link was already made, but let us make sure it is still there**

```
ls -l labevents_pq
```

```
lrwxr-xr-x@ 1 lukehodges  staff  56 Feb 18 13:44 labevents_pq -> /Users/lukehodges/mimic/hosp
```

```
labevents_pq <- arrow::open_dataset("~/mimic/hosp/labevents_pq", format = "parquet")
```

**Now let us filter for what we need and check to see if we did is correct**

```
labevents_pq.filtered <- labevents_pq %>%
  dplyr::select(subject_id, storetime, itemid, charttime, valuenum) %>%
  dplyr::filter(itemid %in% c(50912, 50971, 50983, 50902, 50882, 51221, 51301, 50931))

labevents_pq.filtered10 <- labevents_pq.filtered %>%
  head(10) %>%
  collect()

print(labevents_pq.filtered10)
```

```
# A tibble: 10 x 5
   subject_id storetime           itemid charttime           valuenum
        <int> <dttm>               <int> <dttm>                 <dbl>
 1   10000032 2180-03-23 08:56:00  50931 2180-03-23 04:51:00     95
 2   10000032 2180-03-23 09:40:00  50882 2180-03-23 04:51:00     27
 3   10000032 2180-03-23 09:40:00  50902 2180-03-23 04:51:00    101
 4   10000032 2180-03-23 09:40:00  50912 2180-03-23 04:51:00      0.4
 5   10000032 2180-03-23 09:40:00  50971 2180-03-23 04:51:00      3.7
 6   10000032 2180-03-23 09:40:00  50983 2180-03-23 04:51:00    136
 7   10000032 2180-03-23 08:19:00  51221 2180-03-23 04:51:00     45.4
 8   10000032 2180-03-23 08:19:00  51301 2180-03-23 04:51:00      3
 9   10000032 2180-05-06 15:42:00  51221 2180-05-06 15:25:00     42.6
10   10000032 2180-05-06 15:42:00  51301 2180-05-06 15:25:00      5
```

**Now we have to change the subject_id to an integer so that we can join it with the icustays_tble**

```
icustays_tble <- icustays_tble %>%
  mutate(subject_id = as.integer(subject_id))

icustays_tble10 <- icustays_tble %>%
  head(10) %>%
  collect()

print(icustays_tble10)
```

```
# A tibble: 10 x 8
   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 07:00:00
 2   10000690 25860671 37081114 Medical Inten~ Medical Inte~ 2150-11-02 11:37:00
 3   10000980 26913865 39765666 Medical Inten~ Medical Inte~ 2189-06-27 01:42:00
 4   10001217 24597018 37067082 Surgical Inte~ Surgical Int~ 2157-11-20 11:18:02
 5   10001217 27703517 34592300 Surgical Inte~ Surgical Int~ 2157-12-19 07:42:24
 6   10001725 25563031 31205490 Medical/Surgi~ Medical/Surg~ 2110-04-11 08:52:22
 7   10001843 26133978 39698942 Medical/Surgi~ Medical/Surg~ 2134-12-05 10:50:03
 8   10001884 26184834 37510196 Medical Inten~ Medical Inte~ 2131-01-10 20:20:05
 9   10002013 23581541 39060235 Cardiac Vascu~ Cardiac Vasc~ 2160-05-18 03:00:53
10   10002114 27793700 34672098 Coronary Care~ Coronary Car~ 2162-02-17 15:30:00
# i 2 more variables: outtime <dttm>, los <dbl>
```

**We now have to load the data into R and the do an inner_join. Note that an inner_join was used because a left_join took about 30 minutes to do**

```
labevents_pq.filtered.icu <- labevents_pq.filtered %>%
  collect() %>%
  left_join(icustays_tble, by = c("subject_id"))
```

```
Warning in left_join(., icustays_tble, by = c("subject_id")): Detected an unexpected many-to-
i Row 3958 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

**Let us view the table to see if we are all set so far**

```r
labevents_pq.filtered.icu10 <- labevents_pq.filtered.icu %>%
  head(10) %>%
  collect()

print(labevents_pq.filtered.icu10)
```

```
# A tibble: 10 x 12
   subject_id storetime           itemid charttime           valuenum  hadm_id
        <int> <dttm>               <int> <dttm>                 <dbl>    <int>
 1   10000032 2180-03-23 08:56:00  50931 2180-03-23 04:51:00     95   29079034
 2   10000032 2180-03-23 09:40:00  50882 2180-03-23 04:51:00     27   29079034
 3   10000032 2180-03-23 09:40:00  50902 2180-03-23 04:51:00    101   29079034
 4   10000032 2180-03-23 09:40:00  50912 2180-03-23 04:51:00      0.4 29079034
 5   10000032 2180-03-23 09:40:00  50971 2180-03-23 04:51:00      3.7 29079034
 6   10000032 2180-03-23 09:40:00  50983 2180-03-23 04:51:00    136   29079034
 7   10000032 2180-03-23 08:19:00  51221 2180-03-23 04:51:00     45.4 29079034
 8   10000032 2180-03-23 08:19:00  51301 2180-03-23 04:51:00      3   29079034
 9   10000032 2180-05-06 15:42:00  51221 2180-05-06 15:25:00     42.6 29079034
10   10000032 2180-05-06 15:42:00  51301 2180-05-06 15:25:00      5   29079034
# i 6 more variables: stay_id <int>, first_careunit <chr>, last_careunit <chr>,
#   intime <dttm>, outtime <dttm>, los <dbl>
```

**We need to filter it for the charttime less than the intime and then group it by the three important variables: subject_id, stay_id, and itemid. We then have to slice it per the instructions of Dr. Zhou in the slack and from what we saw in class, and then ungroup it after**

```r
labevents_pq.filtered.icu <- labevents_pq.filtered.icu %>%
  filter(storetime < intime) %>%
  group_by(subject_id, stay_id, itemid) %>%
  slice_max(order_by = storetime, n = 1) %>%
  ungroup()
```

**Making the column names wide version**

```r
labevents_tble <- labevents_pq.filtered.icu %>%
  select(subject_id, stay_id, itemid, valuenum) %>%
  pivot_wider(names_from = itemid, values_from = valuenum)
```

```
Warning: Values from `valuenum` are not uniquely identified; output will contain
```

```
list-cols.
* Use `values_fn = list` to suppress this warning.
* Use `values_fn = {summary_fun}` to summarise duplicates.
* Use the following dplyr code to identify duplicates.
  {data} |>
  dplyr::summarise(n = dplyr::n(), .by = c(subject_id, stay_id, itemid)) |>
  dplyr::filter(n > 1L)
```

```
labevents_tble
```

```
# A tibble: 88,086 x 10
   subject_id  stay_id `50882`   `50902` `50912` `50931` `50971` `50983` `51221`
        <int>    <int> <list>    <list>  <list>  <list>  <list>  <list>  <list>
 1   10000032 39553978 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 2   10000690 37081114 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 3   10000980 39765666 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 4   10001217 34592300 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 5   10001217 37067082 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 6   10001725 31205490 <NULL>    <dbl>   <NULL>  <NULL>  <dbl>   <dbl>   <NULL>
 7   10001843 39698942 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 8   10001884 37510196 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 9   10002013 39060235 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
10   10002114 34672098 <dbl [1]> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
# i 88,076 more rows
# i 1 more variable: `51301` <list>
```

**Turn the data into the correct format**

```
labevents_tble <- labevents_tble %>%
  mutate(across(where(is.list), ~ map_dbl(.x, ~ ifelse(is.null(.x), NA, .x))))
```

**renaming the variables**

```
column_names <- c(
  "50882" = "Bicarbonate",
  "50902" = "Chloride",
  "50912" = "Creatinine",
  "50931" = "Glucose",
  "50971" = "Potassium",
  "50983" = "Sodium",
  "51221" = "Hematocrit",
```

```
  "51301" = "WBC"
)
```

```
labevents_tble <- labevents_tble %>%
  rename_with(~ column_names[.x], .cols = names(column_names))
```

**Viewing the table**

```
labevents_tble
```

```
# A tibble: 88,086 x 10
   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 88,076 more rows
# i 2 more variables: Hematocrit <dbl>, WBC <dbl>
```

**This is the exact table as seen in Question 5!**

## Q6 Vitals from charted events

chartevents.csv.gz (https://mimic.mit.edu/docs/iv/modules/icu/chartevents/) contains all the charted data available for a patient. During their ICU stay, the primary repository of a patient's information is their electronic chart. The itemid variable indicates a single measurement type in the database. The value variable is the value measured for itemid. The first 10 lines of chartevents.csv.gz are

**Looking at the first few lines of the chartevents.csv.gz**

```
zcat < ~/mimic/icu/chartevents.csv.gz | head
```

```
subject_id,hadm_id,stay_id,caregiver_id,charttime,storetime,itemid,value,valuenum,valueuom,wa
10000032,29079034,39553978,18704,2180-07-23 12:36:00,2180-07-23 14:45:00,226512,39.4,39.4,kg
10000032,29079034,39553978,18704,2180-07-23 12:36:00,2180-07-23 14:45:00,226707,60,60,Inch,0
10000032,29079034,39553978,18704,2180-07-23 12:36:00,2180-07-23 14:45:00,226730,152,152,cm,0
10000032,29079034,39553978,18704,2180-07-23 14:00:00,2180-07-23 14:18:00,220048,SR (Sinus Rhy
10000032,29079034,39553978,18704,2180-07-23 14:00:00,2180-07-23 14:18:00,224642,Oral,,,0
10000032,29079034,39553978,18704,2180-07-23 14:00:00,2180-07-23 14:18:00,224650,None,,,0
10000032,29079034,39553978,18704,2180-07-23 14:00:00,2180-07-23 14:20:00,223761,98.7,98.7,°F
10000032,29079034,39553978,18704,2180-07-23 14:11:00,2180-07-23 14:17:00,220179,84,84,mmHg,0
10000032,29079034,39553978,18704,2180-07-23 14:11:00,2180-07-23 14:17:00,220180,48,48,mmHg,0
```

d_items.csv.gz (https://mimic.mit.edu/docs/iv/modules/icu/d_items/) is the dictionary for the itemid in chartevents.csv.gz.

**Looking into the first few lines of the d_items.csv.gz file**

```
zcat < ~/mimic/icu/d_items.csv.gz | head
```

```
itemid,label,abbreviation,linksto,category,unitname,param_type,lownormalvalue,highnormalvalue
220001,Problem List,Problem List,chartevents,General,,Text,,
220003,ICU Admission date,ICU Admission date,datetimeevents,ADT,,Date and time,,
220045,Heart Rate,HR,chartevents,Routine Vital Signs,bpm,Numeric,,
220046,Heart rate Alarm - High,HR Alarm - High,chartevents,Alarms,bpm,Numeric,,
220047,Heart Rate Alarm - Low,HR Alarm - Low,chartevents,Alarms,bpm,Numeric,,
220048,Heart Rhythm,Heart Rhythm,chartevents,Routine Vital Signs,,Text,,
220050,Arterial Blood Pressure systolic,ABPs,chartevents,Routine Vital Signs,mmHg,Numeric,90
220051,Arterial Blood Pressure diastolic,ABPd,chartevents,Routine Vital Signs,mmHg,Numeric,60
220052,Arterial Blood Pressure mean,ABPm,chartevents,Routine Vital Signs,mmHg,Numeric,,
```

We are interested in the vitals for ICU patients: heart rate (220045), systolic non-invasive blood pressure (220179), diastolic non-invasive blood pressure (220180), body temperature in Fahrenheit (223761), and respiratory rate (220210). Retrieve a subset of chartevents.csv.gz only containing these items for the patients in icustays_tble. Further restrict to the first vital measurement within the ICU stay. The final chartevents_tble should have one row per ICU stay and columns for each vital measurement.

**Let us create the chartevents file, write it as a parquet, and then create a symbolic link to it**

```
chartevents6 <- arrow::open_dataset("~/mimic/icu/chartevents.csv",
                                    format = "csv")
```

```
arrow::write_dataset(chartevents6, path = "~/mimic/hosp/chartevents_pq",
                     format = "parquet")
```

**Now I have to make a symbolic link to the chartevents_pq by running the following commands in the terminal**

cd ~/Desktop/203b-hw/hw3

and then

ln -s /Users/lukehodges/Desktop/203b-hw/hw3/chartevents_pq chartevents_pq

and then

```
ls -l ~/Desktop/203b-hw/hw3
```

```
total 13752
-rw-r--r--@ 1 lukehodges  staff  5472513 Feb 20 18:46 HW3.html
-rw-r--r--@ 1 lukehodges  staff    44097 Feb 20 18:46 HW3.qmd
-rw-r--r--@ 1 lukehodges  staff    44527 Feb 20 18:46 HW3.rmarkdown
drwxr-xr-x@ 3 lukehodges  staff       96 Feb 20 18:46 HW3_files
-rw-r--r--@ 1 lukehodges  staff   635346 Feb 18 13:23 Patient_Vitals_Plot.png
lrwxr-xr-x@ 1 lukehodges  staff       58 Feb 18 18:37 chartevents_pq -> /Users/lukehodges/mi
lrwxr-xr-x@ 1 lukehodges  staff       56 Feb 18 13:44 labevents_pq -> /Users/lukehodges/mimi
lrwxr-xr-x@ 1 lukehodges  staff       43 Feb 20 14:30 part-0.parquet -> /Users/lukehodges/mi
```

**Turning chartevents into a parquet**

```
chartevents6 <- arrow::open_dataset("~/mimic/hosp/chartevents_pq",
                                    format = "parquet")
```

**Now, let us filter for the required itemid values and then display the first ten lines of the needed tables**

```
chartevents_filtered <- chartevents6 %>%
  dplyr::select(subject_id, itemid, charttime, valuenum, stay_id,
                storetime) %>%
  dplyr::filter(itemid %in% c(220045, 220179, 220180, 223761, 220210))

chartevents_filtered10 <- chartevents_filtered %>%
  head(10) %>%
  collect()
```

```
print(chartevents_filtered10)
```

```
# A tibble: 10 x 6
   subject_id itemid charttime           valuenum  stay_id storetime
        <int>  <int> <dttm>                 <dbl>    <int> <dttm>
 1   10000032 223761 2180-07-23 07:00:00     98.7 39553978 2180-07-23 07:20:00
 2   10000032 220179 2180-07-23 07:11:00     84   39553978 2180-07-23 07:17:00
 3   10000032 220180 2180-07-23 07:11:00     48   39553978 2180-07-23 07:17:00
 4   10000032 220045 2180-07-23 07:12:00     91   39553978 2180-07-23 07:17:00
 5   10000032 220210 2180-07-23 07:12:00     24   39553978 2180-07-23 07:17:00
 6   10000032 220045 2180-07-23 07:30:00     93   39553978 2180-07-23 07:43:00
 7   10000032 220179 2180-07-23 07:30:00     95   39553978 2180-07-23 07:43:00
 8   10000032 220180 2180-07-23 07:30:00     59   39553978 2180-07-23 07:43:00
 9   10000032 220210 2180-07-23 07:30:00     21   39553978 2180-07-23 07:43:00
10   10000032 220045 2180-07-23 08:00:00     94   39553978 2180-07-23 08:34:00
```

```
icustays_tble10 <- icustays_tble %>%
  head(10) %>%
  collect()

print(icustays_tble10)
```

```
# A tibble: 10 x 8
   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 07:00:00
 2   10000690 25860671 37081114 Medical Inten~ Medical Inte~ 2150-11-02 11:37:00
 3   10000980 26913865 39765666 Medical Inten~ Medical Inte~ 2189-06-27 01:42:00
 4   10001217 24597018 37067082 Surgical Inte~ Surgical Int~ 2157-11-20 11:18:02
 5   10001217 27703517 34592300 Surgical Inte~ Surgical Int~ 2157-12-19 07:42:24
 6   10001725 25563031 31205490 Medical/Surgi~ Medical/Surg~ 2110-04-11 08:52:22
 7   10001843 26133978 39698942 Medical/Surgi~ Medical/Surg~ 2134-12-05 10:50:03
 8   10001884 26184834 37510196 Medical Inten~ Medical Inte~ 2131-01-10 20:20:05
 9   10002013 23581541 39060235 Cardiac Vascu~ Cardiac Vasc~ 2160-05-18 03:00:53
10   10002114 27793700 34672098 Coronary Care~ Coronary Car~ 2162-02-17 15:30:00
# i 2 more variables: outtime <dttm>, los <dbl>
```

**Make the subject_ids in the icustays_tble into an integer as I did in Q5 and then make sure it worked**

```r
icustays_tble <- icustays_tble %>%
  mutate(subject_id = as.integer(subject_id))

icustays_tble10 <- icustays_tble %>%
  head(10) %>%
  collect()

print(icustays_tble10)
```

```
# A tibble: 10 x 8
   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 07:00:00
 2   10000690 25860671 37081114 Medical Inten~ Medical Inte~ 2150-11-02 11:37:00
 3   10000980 26913865 39765666 Medical Inten~ Medical Inte~ 2189-06-27 01:42:00
 4   10001217 24597018 37067082 Surgical Inte~ Surgical Int~ 2157-11-20 11:18:02
 5   10001217 27703517 34592300 Surgical Inte~ Surgical Int~ 2157-12-19 07:42:24
 6   10001725 25563031 31205490 Medical/Surgi~ Medical/Surg~ 2110-04-11 08:52:22
 7   10001843 26133978 39698942 Medical/Surgi~ Medical/Surg~ 2134-12-05 10:50:03
 8   10001884 26184834 37510196 Medical Inten~ Medical Inte~ 2131-01-10 20:20:05
 9   10002013 23581541 39060235 Cardiac Vascu~ Cardiac Vasc~ 2160-05-18 03:00:53
10   10002114 27793700 34672098 Coronary Care~ Coronary Car~ 2162-02-17 15:30:00
# i 2 more variables: outtime <dttm>, los <dbl>
```

**Conduct an inner_join of the values from the "subject_id", "stay_id", and the "hadm_id". Hadm_id was used to avoid two**

```r
chartevents_icu <- chartevents_filtered %>%
  collect() %>%
  left_join(icustays_tble, by = c("subject_id", "stay_id")) %>%
  filter(storetime >= intime & storetime <= outtime)
```

**View the data table created**

```r
chartevents_icu
```

```
# A tibble: 30,118,451 x 12
   subject_id itemid charttime           valuenum  stay_id storetime
        <int>  <int> <dttm>                 <dbl>    <int> <dttm>
 1   10001884 220045 2131-01-14 23:00:00     74   37510196 2131-01-15 00:32:00
```

```
 2   10001884 220210 2131-01-14 23:00:00     26   37510196 2131-01-15 00:32:00
 3   10001884 220179 2131-01-14 23:01:00    122   37510196 2131-01-15 00:32:00
 4   10001884 220180 2131-01-14 23:01:00     79   37510196 2131-01-15 00:32:00
 5   10001884 220045 2131-01-15 00:00:00     74   37510196 2131-01-15 00:32:00
 6   10001884 220210 2131-01-15 00:00:00     25   37510196 2131-01-15 00:32:00
 7   10001884 223761 2131-01-15 00:00:00     99.5 37510196 2131-01-15 00:32:00
 8   10001884 220179 2131-01-15 00:01:00    121   37510196 2131-01-15 00:32:00
 9   10001884 220180 2131-01-15 00:01:00     74   37510196 2131-01-15 00:32:00
10   10001884 220045 2131-01-15 01:00:00     70   37510196 2131-01-15 01:10:00
# i 30,118,441 more rows
# i 6 more variables: hadm_id <int>, first_careunit <chr>, last_careunit <chr>,
#   intime <dttm>, outtime <dttm>, los <dbl>
```

**We need to arrange the values, then group them, and then get the first vital measurement (by storetime) within the ICU stay.**

```
chartevents_icu_first <- chartevents_icu %>%
  arrange(subject_id, stay_id, itemid, storetime)

chartevents_icu_second <- chartevents_icu_first %>%
  group_by(subject_id, stay_id, itemid, storetime) %>%
  summarise(valuenum = mean(valuenum, na.rm = TRUE), .groups = "drop") %>%
  group_by(subject_id, stay_id, itemid) %>%
  slice(1) %>%
  ungroup()

chartevents_icu_second
```

```
# A tibble: 467,516 x 5
   subject_id  stay_id itemid storetime           valuenum
        <int>    <int>  <int> <dttm>                 <dbl>
 1   10000032 39553978 220045 2180-07-23 07:17:00     91
 2   10000032 39553978 220179 2180-07-23 07:17:00     84
 3   10000032 39553978 220180 2180-07-23 07:17:00     48
 4   10000032 39553978 220210 2180-07-23 07:17:00     24
 5   10000032 39553978 223761 2180-07-23 07:20:00     98.7
 6   10000690 37081114 220045 2150-11-02 12:12:00     78
 7   10000690 37081114 220179 2150-11-02 12:12:00    106
 8   10000690 37081114 220180 2150-11-02 12:12:00     56.5
 9   10000690 37081114 220210 2150-11-02 12:12:00     24.3
10   10000690 37081114 223761 2150-11-02 12:12:00     97.7
# i 467,506 more rows
```

**This is changing the format to be more representative of what is seen in Q6**

```
chartevents_pivot <- chartevents_icu_second %>%
  select(subject_id, stay_id, itemid, valuenum) %>%
  pivot_wider(names_from = itemid, values_from = valuenum)

chartevents_pivot
```

```
# A tibble: 94,363 x 7
   subject_id  stay_id `220045` `220179` `220180` `220210` `223761`
        <int>    <int>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
 1   10000032 39553978     91       84       48       24       98.7
 2   10000690 37081114     78      106       56.5     24.3     97.7
 3   10000980 39765666     76      154      102       23.5     98
 4   10001217 34592300     79.3    156       93.3     14       97.6
 5   10001217 37067082     86      151       90       18       98.5
 6   10001725 31205490     86       73       56       19       97.7
 7   10001843 39698942    124.     110       78       16.5     97.9
 8   10001884 37510196     49      174.      30.5     13       98.1
 9   10002013 39060235     80       98.5     62       14       97.2
10   10002114 34672098    110.     112       80       21       97.9
# i 94,353 more rows
```

**Changing the column names to be more exact**

```
chartevents_pivot <- chartevents_pivot %>%
  rename(
    Heart_Rate = "220045",
    SysBP = "220179",
    DiaBP = "220180",
    Temp = "223761",
    Respiratory_Rate = "220210"
  )

chartevents_pivot
```

```
# A tibble: 94,363 x 7
   subject_id  stay_id Heart_Rate SysBP DiaBP Respiratory_Rate  Temp
        <int>    <int>      <dbl> <dbl> <dbl>            <dbl> <dbl>
 1   10000032 39553978       91    84    48               24    98.7
 2   10000690 37081114       78   106    56.5             24.3  97.7
```

```
 3   10000980 39765666       76   154   102              23.5  98
 4   10001217 34592300       79.3 156    93.3            14    97.6
 5   10001217 37067082       86   151    90              18    98.5
 6   10001725 31205490       86    73    56              19    97.7
 7   10001843 39698942      124.  110    78              16.5  97.9
 8   10001884 37510196       49   174.   30.5            13    98.1
 9   10002013 39060235       80    98.5  62              14    97.2
10   10002114 34672098      110.  112    80              21    97.9
# i 94,353 more rows
```

**This is the exact chart seen in Q6**

## Q7 Putting things together

**Everything is read in. We now have to make the table**

```
icu_adults <- icustays_tble %>%
  inner_join(admissions_tble, by = c("subject_id", "hadm_id")) %>%
  inner_join(patients_tble, by = "subject_id") %>%
  filter(anchor_age >= 18)  # Use anchor_age instead of calculating from dob

names(icu_adults)
```

```
 [1] "subject_id"           "hadm_id"              "stay_id"
 [4] "first_careunit"       "last_careunit"        "intime"
 [7] "outtime"              "los"                  "admittime"
[10] "dischtime"            "deathtime"            "admission_type"
[13] "admit_provider_id"    "admission_location"   "discharge_location"
[16] "insurance"            "language"             "marital_status"
[19] "race"                 "edregtime"            "edouttime"
[22] "hospital_expire_flag" "length_of_stay"       "gender"
[25] "anchor_age"           "anchor_year"          "anchor_year_group"
[28] "dod"
```

**We have successfully merge it with admissions_tble, patients_tble, but now we need to do so by labevents_tble and chartevents_pivot**

```
labevents_tble <- labevents_tble %>%
  select(subject_id, stay_id, Creatinine, Potassium,
         Sodium, Chloride, Bicarbonate, Hematocrit, WBC, Glucose)

glimpse(labevents_tble)
```

```
Rows: 88,086
Columns: 10
$ subject_id  <int> 10000032, 10000690, 10000980, 10001217, 10001217, 10001725~
$ stay_id     <int> 39553978, 37081114, 39765666, 34592300, 37067082, 31205490~
$ Creatinine  <dbl> 0.7, 1.0, 2.3, 0.5, 0.6, NA, 1.3, 1.1, 0.9, 3.1, 2.8, 1.4,~
$ Potassium   <dbl> 6.7, 4.8, 3.9, 4.1, 4.2, 4.1, 3.9, 4.5, 3.5, 6.5, 4.9, 5.7~
$ Sodium      <dbl> 126, 137, 144, 142, 142, 139, 138, 130, 137, 125, 135, 120~
$ Chloride    <dbl> 95, 100, 109, 104, 108, 98, 97, 88, 102, NA, 98, 85, 105, ~
$ Bicarbonate <dbl> 25, 26, 21, 30, 22, NA, 28, 30, 24, 18, 23, 26, 24, 22, 25~
$ Hematocrit  <dbl> 41.1, 36.1, 27.3, 37.4, 38.1, NA, 31.4, 39.7, 34.9, 34.3, ~
$ WBC         <dbl> 6.9, 7.1, 5.3, 5.4, 15.7, NA, 10.4, 12.2, 7.2, 16.8, 17.9,~
$ Glucose     <dbl> 102, 85, 89, 87, 112, NA, 131, 141, 288, 95, 117, 133, 138~
```

```r
chartevents_pivot <- chartevents_pivot %>%
  select(subject_id, stay_id, Heart_Rate, SysBP,
         DiaBP, Respiratory_Rate, Temp)

glimpse(chartevents_pivot)
```

```
Rows: 94,363
Columns: 7
$ subject_id       <int> 10000032, 10000690, 10000980, 10001217, 10001217, 100~
$ stay_id          <int> 39553978, 37081114, 39765666, 34592300, 37067082, 312~
$ Heart_Rate       <dbl> 91.00000, 78.00000, 76.00000, 79.33333, 86.00000, 86.~
$ SysBP            <dbl> 84.0, 106.0, 154.0, 156.0, 151.0, 73.0, 110.0, 173.5,~
$ DiaBP            <dbl> 48.00000, 56.50000, 102.00000, 93.33333, 90.00000, 56~
$ Respiratory_Rate <dbl> 24.00000, 24.33333, 23.50000, 14.00000, 18.00000, 19.~
$ Temp             <dbl> 98.7, 97.7, 98.0, 97.6, 98.5, 97.7, 97.9, 98.1, 97.2,~
```

**Now that we have modified the tables, we need to merge them**

```r
mimic_icu_cohort <- icu_adults %>%
  left_join(labevents_tble, by = c("subject_id", "stay_id")) %>%
  left_join(chartevents_pivot, by = c("subject_id", "stay_id"))

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 07:00:00
```

```
 2   10000690 25860671 37081114 Medical Inten~ Medical Inte~ 2150-11-02 11:37:00
 3   10000980 26913865 39765666 Medical Inten~ Medical Inte~ 2189-06-27 01:42:00
 4   10001217 24597018 37067082 Surgical Inte~ Surgical Int~ 2157-11-20 11:18:02
 5   10001217 27703517 34592300 Surgical Inte~ Surgical Int~ 2157-12-19 07:42:24
 6   10001725 25563031 31205490 Medical/Surgi~ Medical/Surg~ 2110-04-11 08:52:22
 7   10001843 26133978 39698942 Medical/Surgi~ Medical/Surg~ 2134-12-05 10:50:03
 8   10001884 26184834 37510196 Medical Inten~ Medical Inte~ 2131-01-10 20:20:05
 9   10002013 23581541 39060235 Cardiac Vascu~ Cardiac Vasc~ 2160-05-18 03:00:53
10   10002114 27793700 34672098 Coronary Care~ Coronary Car~ 2162-02-17 15:30:00
# i 94,448 more rows
# i 35 more variables: outtime <dttm>, los <dbl>, 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>,
#   hospital_expire_flag <int>, length_of_stay <drtn>, gender <fct>, ...
```

**This is the table as seen in the diagram**

## Q8 Exploratory data analysis (EDA)

Summarize the following information about the ICU stay cohort mimic_icu_cohort using appropriate numerics or graphs:

Length of ICU stay los vs demographic variables (race, insurance, marital_status, gender, age at intime)

**First, let us make the length of stay a numeric value**

```
mimic_icu_cohort <- mimic_icu_cohort %>%
  mutate(los = as.numeric(los))

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 07:00:00
 2   10000690 25860671 37081114 Medical Inten~ Medical Inte~ 2150-11-02 11:37:00
 3   10000980 26913865 39765666 Medical Inten~ Medical Inte~ 2189-06-27 01:42:00
 4   10001217 24597018 37067082 Surgical Inte~ Surgical Int~ 2157-11-20 11:18:02
 5   10001217 27703517 34592300 Surgical Inte~ Surgical Int~ 2157-12-19 07:42:24
```

```
 6   10001725 25563031 31205490 Medical/Surgi~ Medical/Surg~ 2110-04-11 08:52:22
 7   10001843 26133978 39698942 Medical/Surgi~ Medical/Surg~ 2134-12-05 10:50:03
 8   10001884 26184834 37510196 Medical Inten~ Medical Inte~ 2131-01-10 20:20:05
 9   10002013 23581541 39060235 Cardiac Vascu~ Cardiac Vasc~ 2160-05-18 03:00:53
10   10002114 27793700 34672098 Coronary Care~ Coronary Car~ 2162-02-17 15:30:00
# i 94,448 more rows
# i 35 more variables: outtime <dttm>, los <dbl>, 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>,
#   hospital_expire_flag <int>, length_of_stay <drtn>, gender <fct>, ...
```

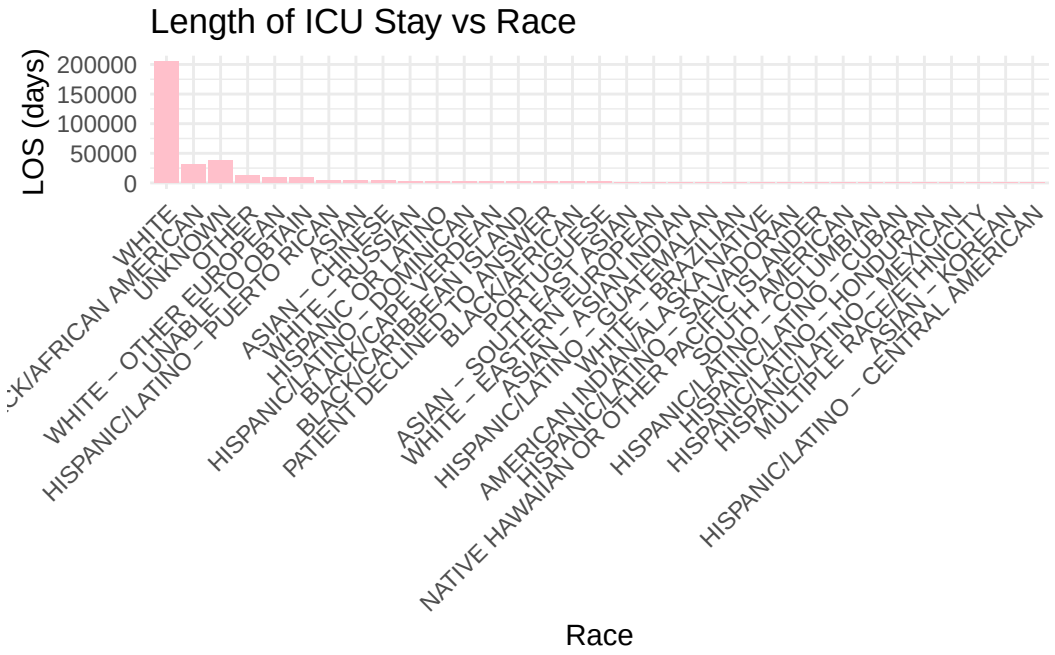
```r
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(race) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))
```

**Now let us make the graph**

**LOS vs. Race**

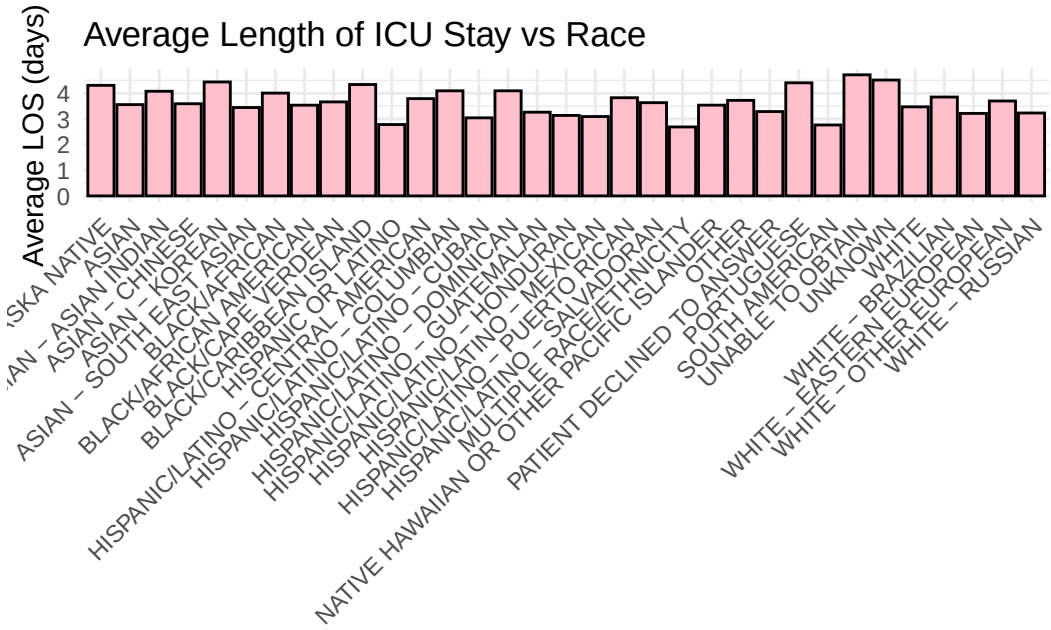
```r
ggplot(mimic_icu_cohort, aes(x = fct_infreq(race), y = los)) +
  geom_col(fill = "pink") +
  labs(title = "Length of ICU Stay vs Race", x = "Race", y = "LOS (days)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
Warning: Removed 14 rows containing missing values or values outside the scale range
(`geom_col()`).
```



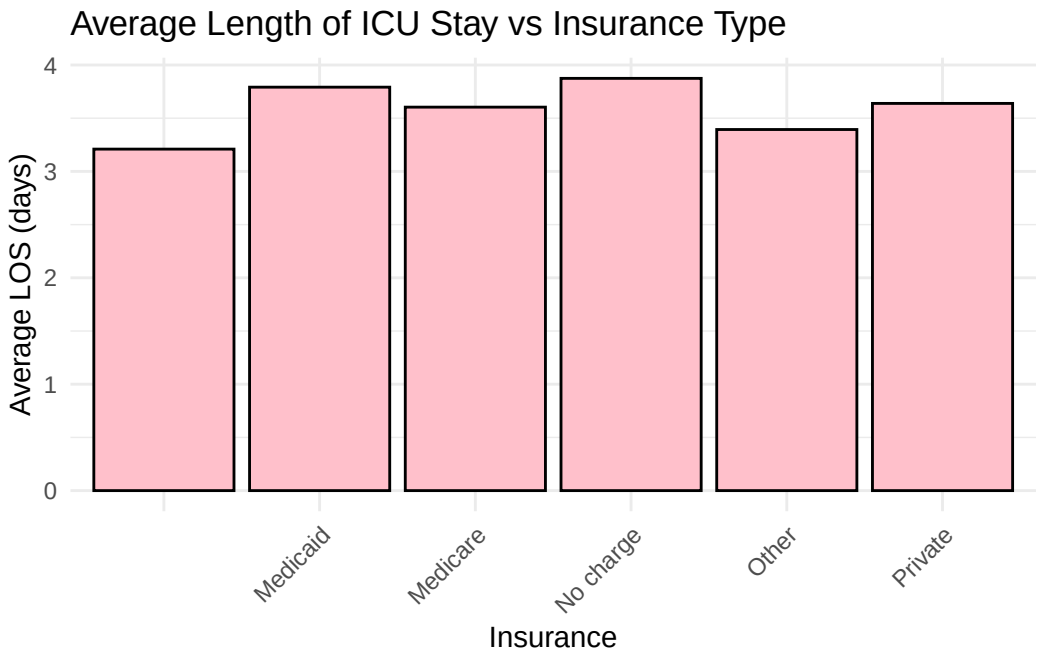

**Although this is a nice graph, I would much rather have it be average length of stay**

```
ggplot(mimic_icu_summary, aes(x = fct_infreq(race), y = mean_los)) +
  geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Race", x = "Race", y = "Average LOS (days)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




**LOS vs. Insurance**

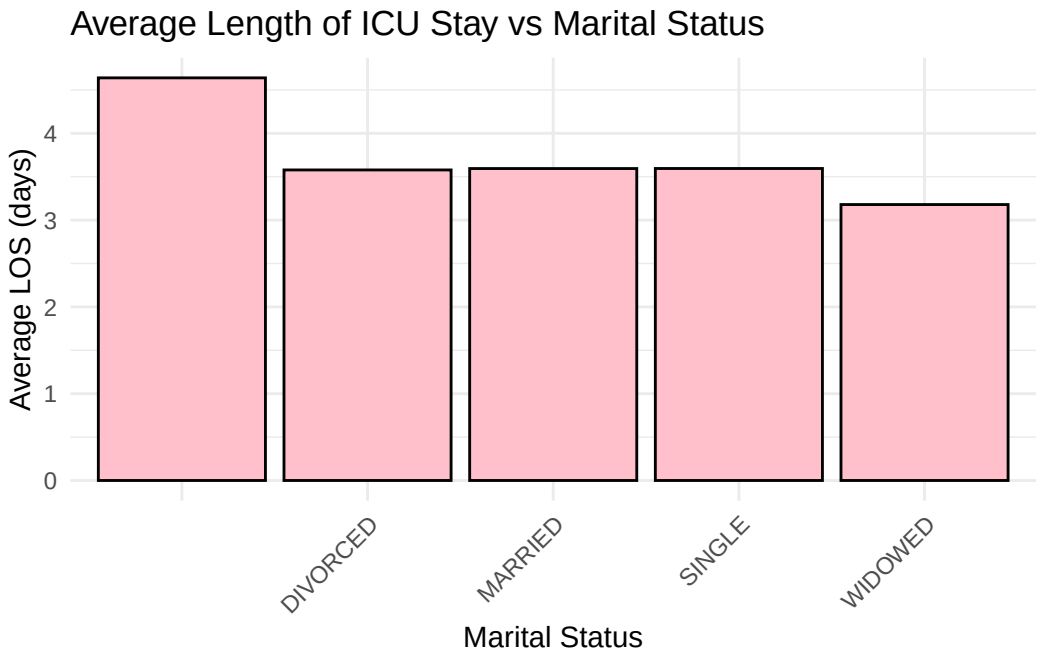
```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(insurance) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = fct_infreq(insurance), y = mean_los)) +
  geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Insurance Type", x = "Insurance",
       y = "Average LOS (days)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




**When I do unique(mimic_icu_cohort$insurance), there is an insurance that is " " or in other words, blank**

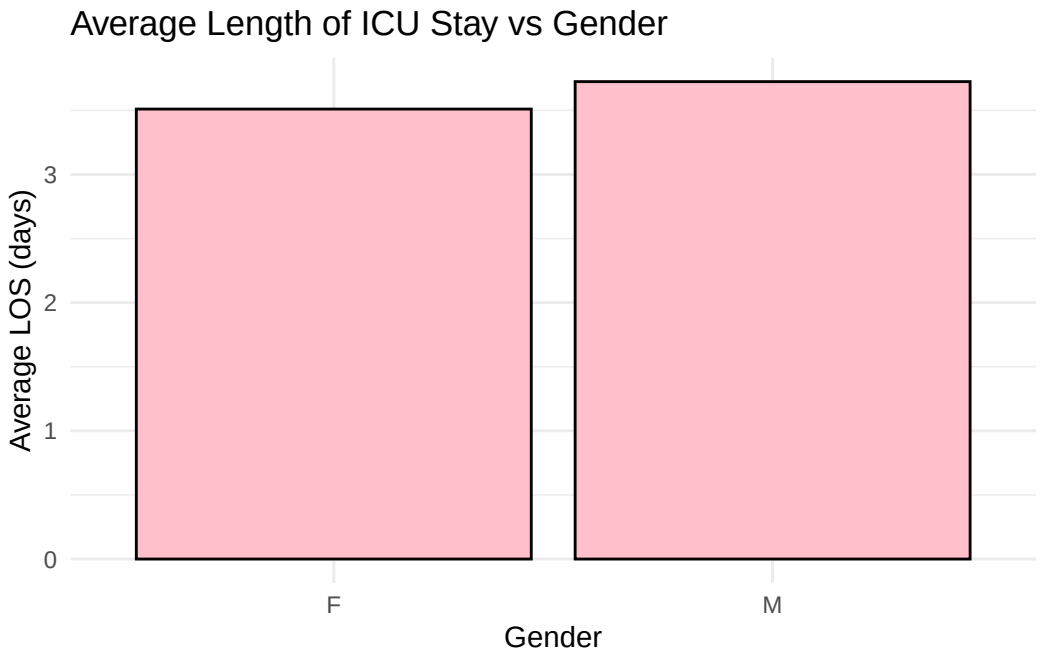
**LOS vs. Marital Status**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(marital_status) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = fct_infreq(marital_status), y = mean_los)) +
  geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Marital Status",
       x = "Marital Status", y = "Average LOS (days)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



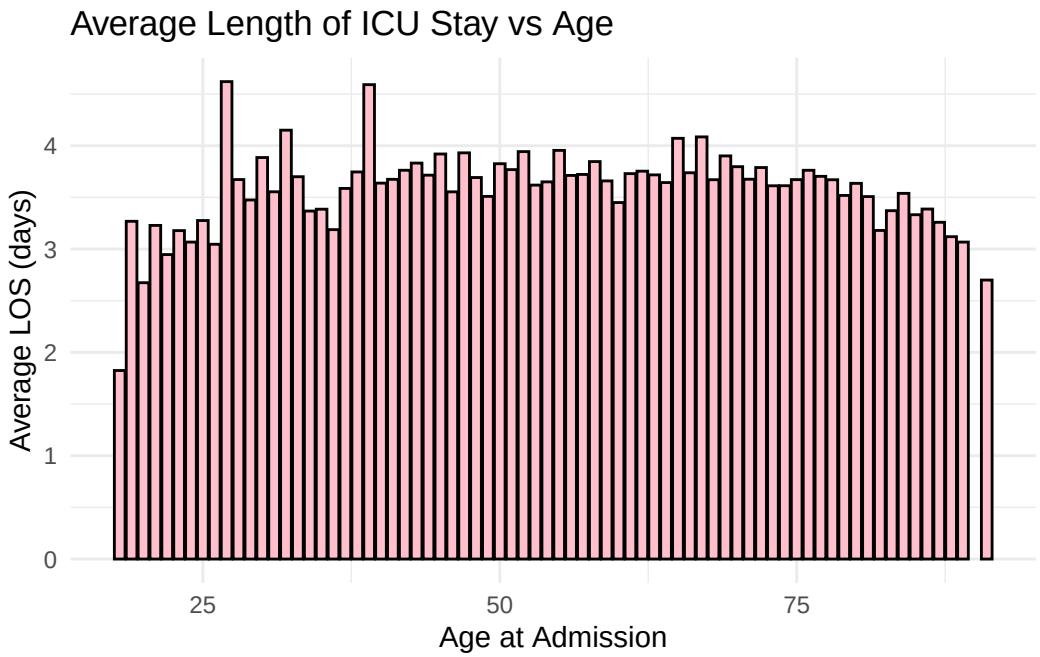

**LOS vs. Gender**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(gender) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = gender, y = mean_los)) +
  geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Gender",
       x = "Gender", y = "Average LOS (days)") +
  theme_minimal()
```




**LOS vs. Anchor_age**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(anchor_age) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = anchor_age, y = mean_los)) +
  geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Age",
       x = "Age at Admission", y = "Average LOS (days)") +
  theme_minimal()
```



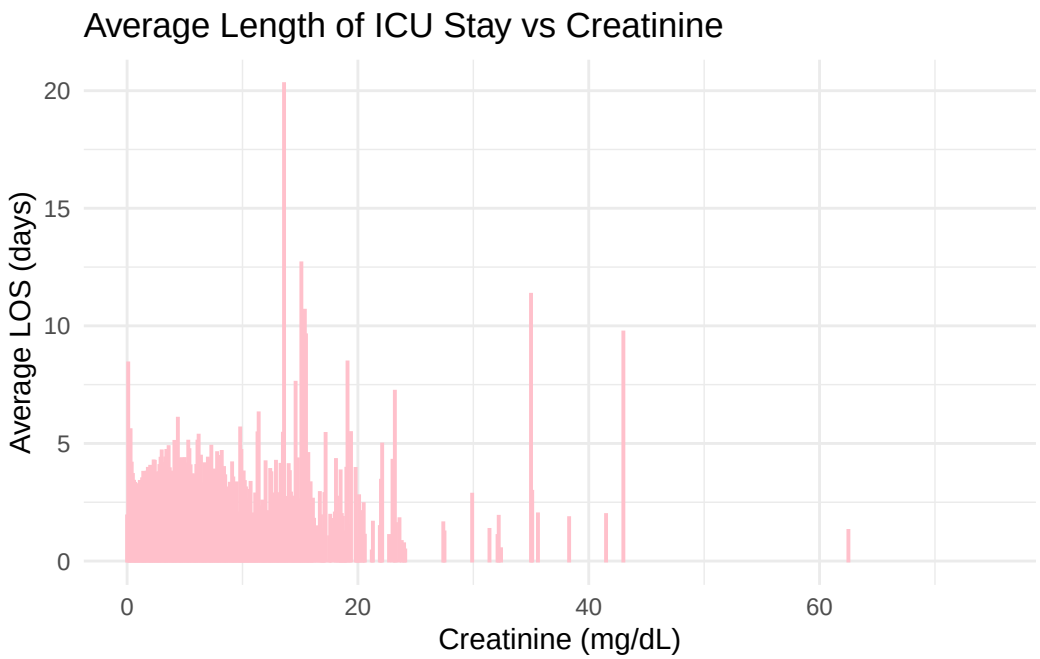

**LOS vs. Last Available Lab Measurements Before ICU Stay**

**LOS vs. Lab Creatinine**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Creatinine) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Creatinine, y = mean_los)) +
  geom_col(color = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Creatinine",
       x = "Creatinine (mg/dL)",
       y = "Average LOS (days)") +
  theme_minimal() +
  coord_cartesian(xlim = c(0, 75))
```

```
Warning: Duplicated aesthetics after name standardisation: colour
```

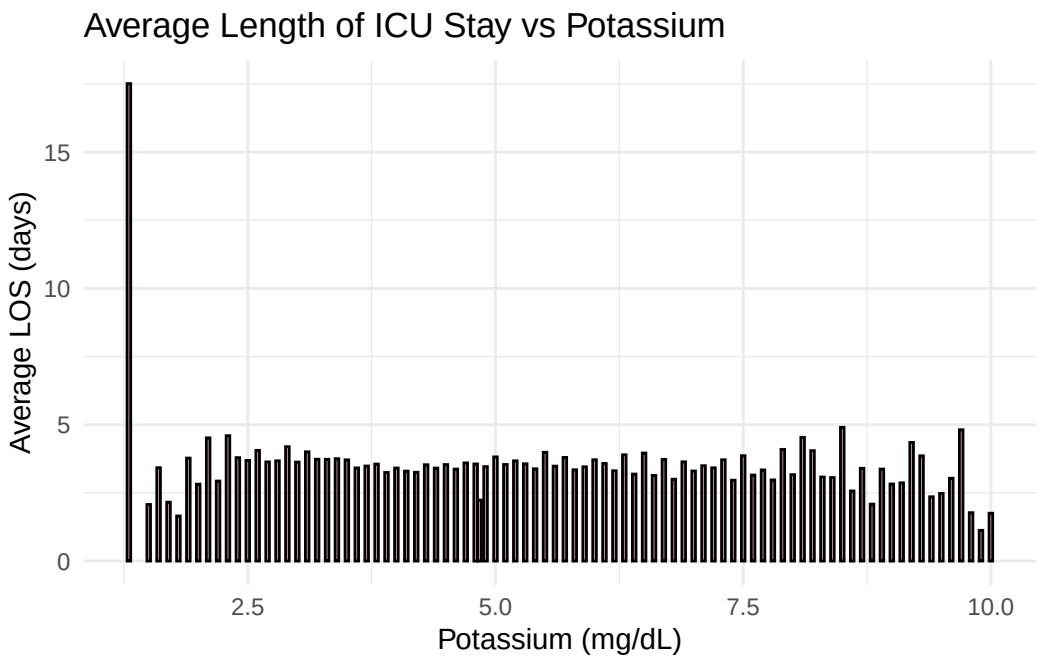
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




### LOS vs. Lab Potassium

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Potassium) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Potassium, y = mean_los)) +
  geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Potassium",
       x = "Potassium (mg/dL)", y = "Average LOS (days)") +
  theme_minimal()
```

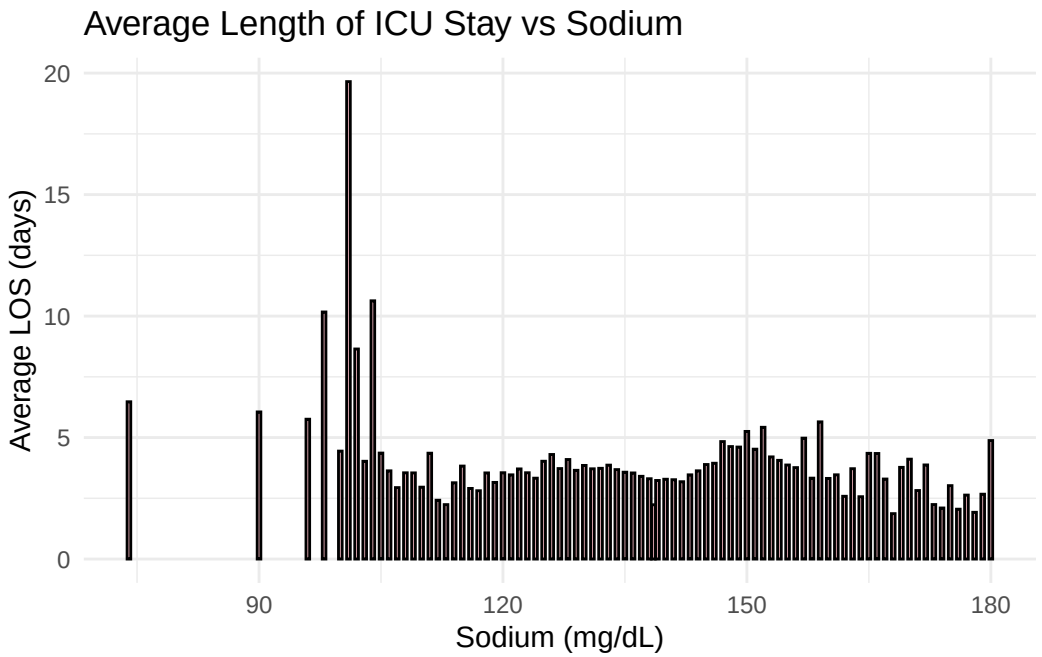
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




### LOS vs. Lab Sodium

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Sodium) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Sodium, y = mean_los )) +
  geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Sodium",
       x = "Sodium (mg/dL)", y = "Average LOS (days)") +
  theme_minimal()
```

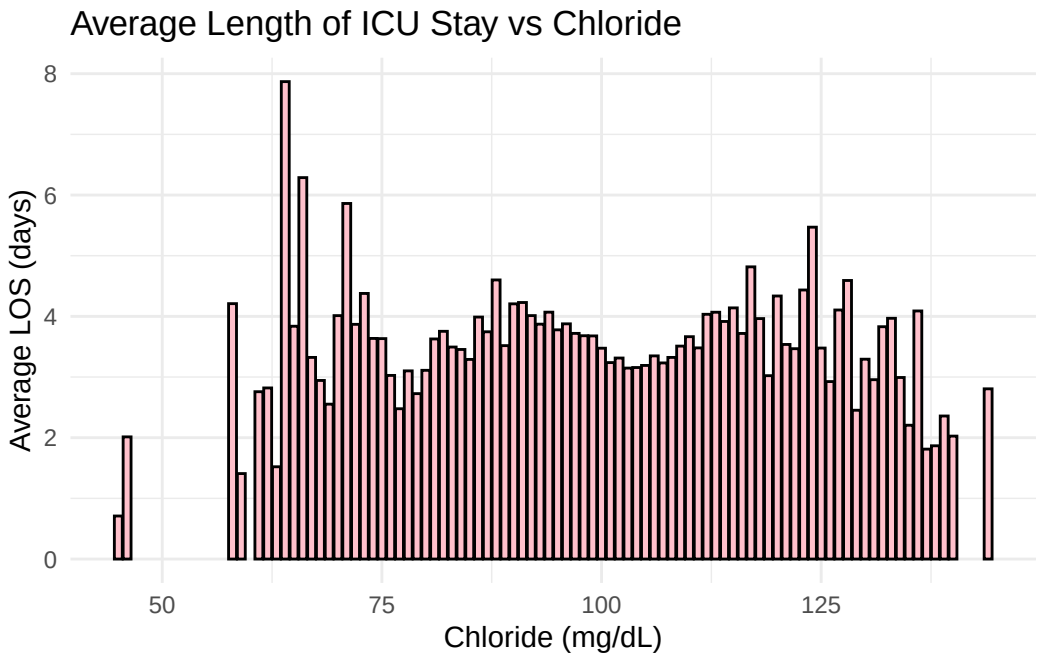
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




**LOS vs. Lab Chloride**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Chloride) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Chloride, y = mean_los )) +
   geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs Chloride",
       x = "Chloride (mg/dL)", y = "Average LOS (days)") +
  theme_minimal()
```

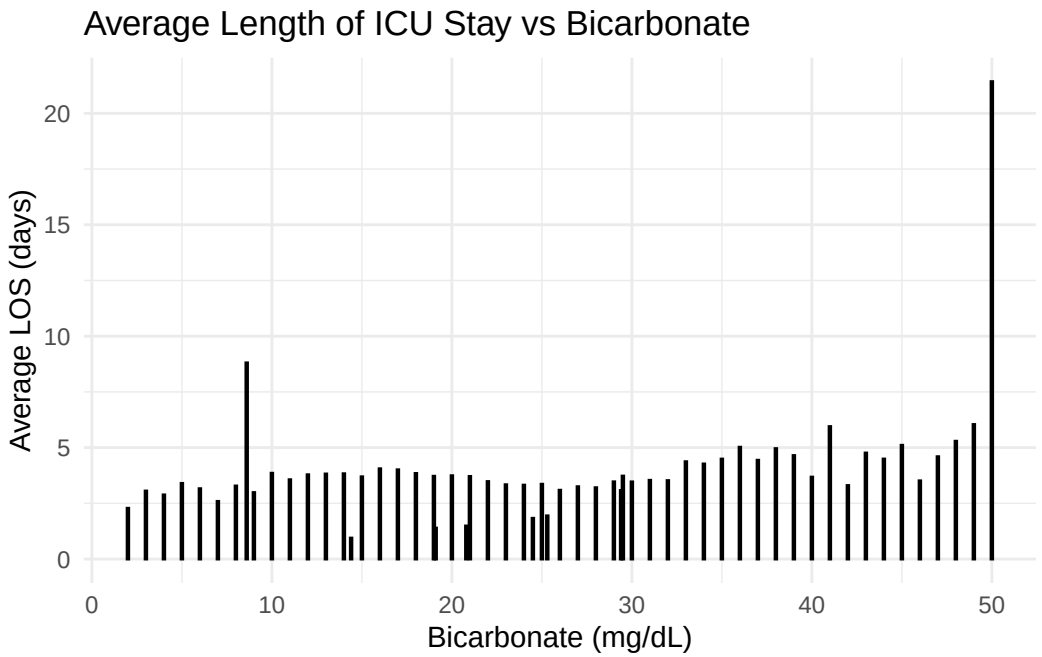
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




**LOS vs. Lab Bicarbonate**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Bicarbonate) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Bicarbonate, y = mean_los )) +
   geom_col(color = "black") +
  labs(title = "Average Length of ICU Stay vs Bicarbonate",
       x = "Bicarbonate (mg/dL)", y = "Average LOS (days)") +
  theme_minimal()
```

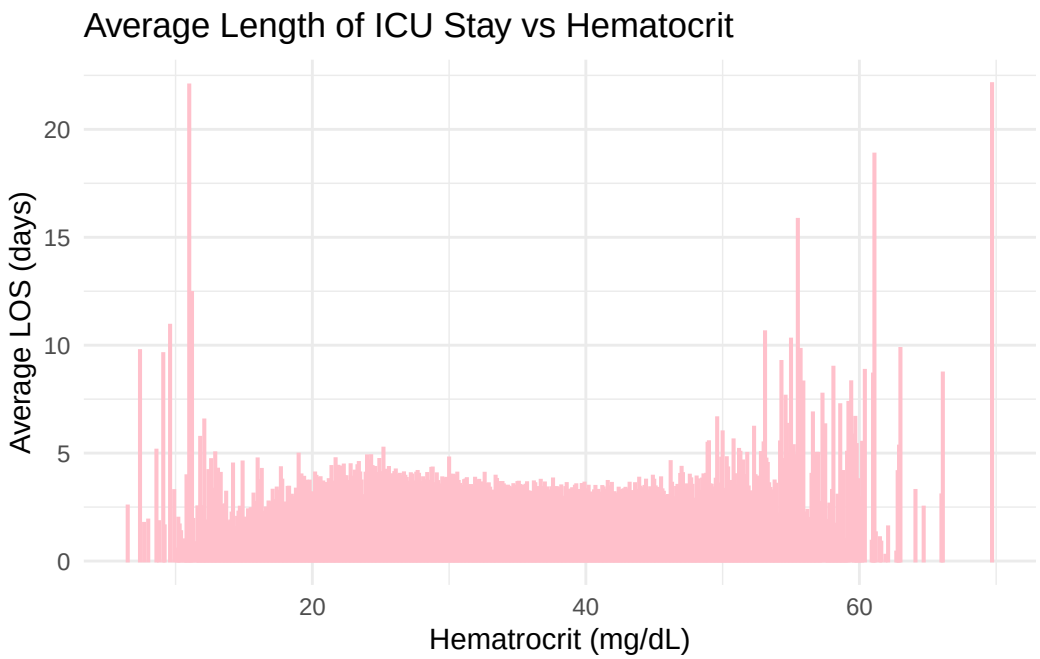
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




**LOS vs. Lab Hematocrit**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Hematocrit) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Hematocrit, y = mean_los )) +
   geom_col(color = "pink") +
  labs(title = "Average Length of ICU Stay vs Hematocrit",
       x = "Hematrocrit (mg/dL)", y = "Average LOS (days)") +
  theme_minimal()
```

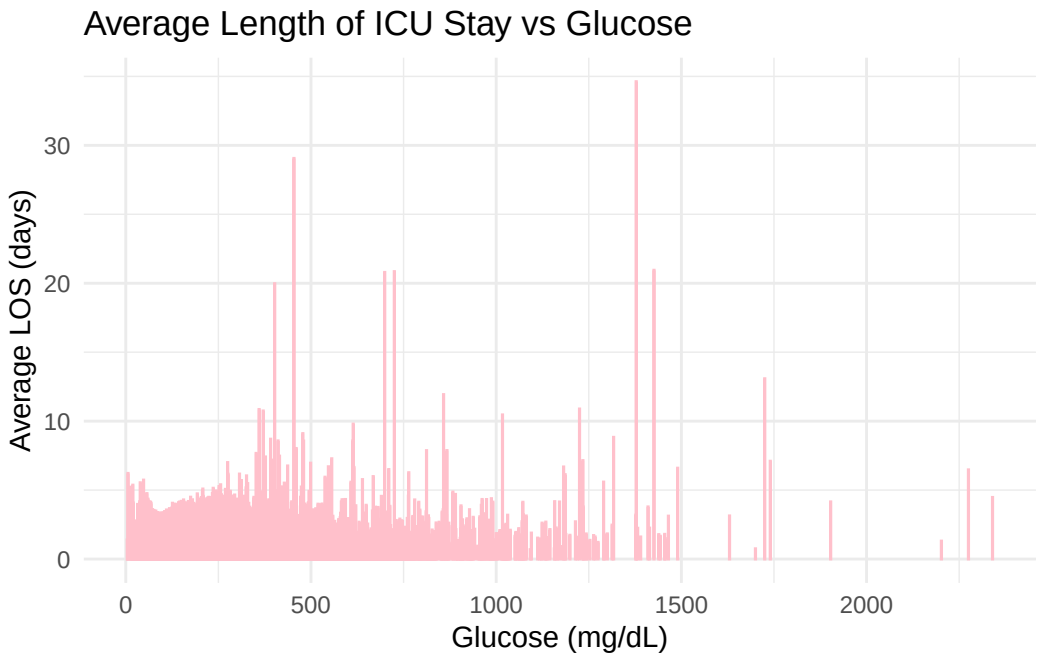
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




### LOS vs. Lab Glucose

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Glucose) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Glucose, y = mean_los )) +
   geom_col(color = "pink") +
  labs(title = "Average Length of ICU Stay vs Glucose",
       x = "Glucose (mg/dL)", y = "Average LOS (days)") +
  theme_minimal()
```

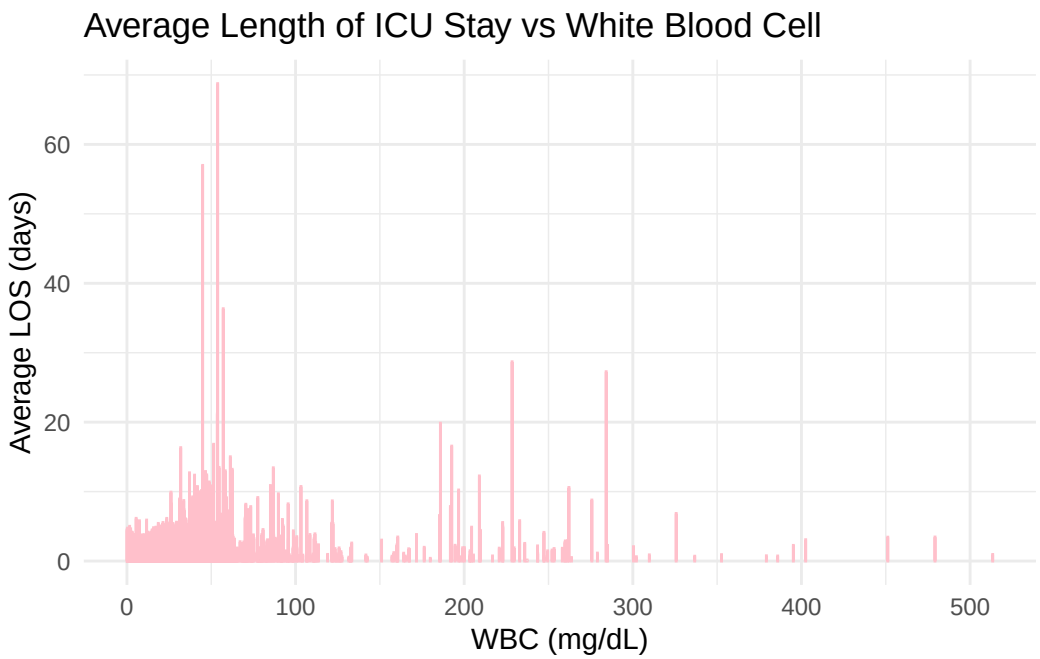
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




**LOS vs. Lab WBC**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(WBC) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = WBC, y = mean_los )) +
   geom_col(color = "pink") +
  labs(title = "Average Length of ICU Stay vs White Blood Cell",
       x = "WBC (mg/dL)", y = "Average LOS (days)") +
  theme_minimal()
```

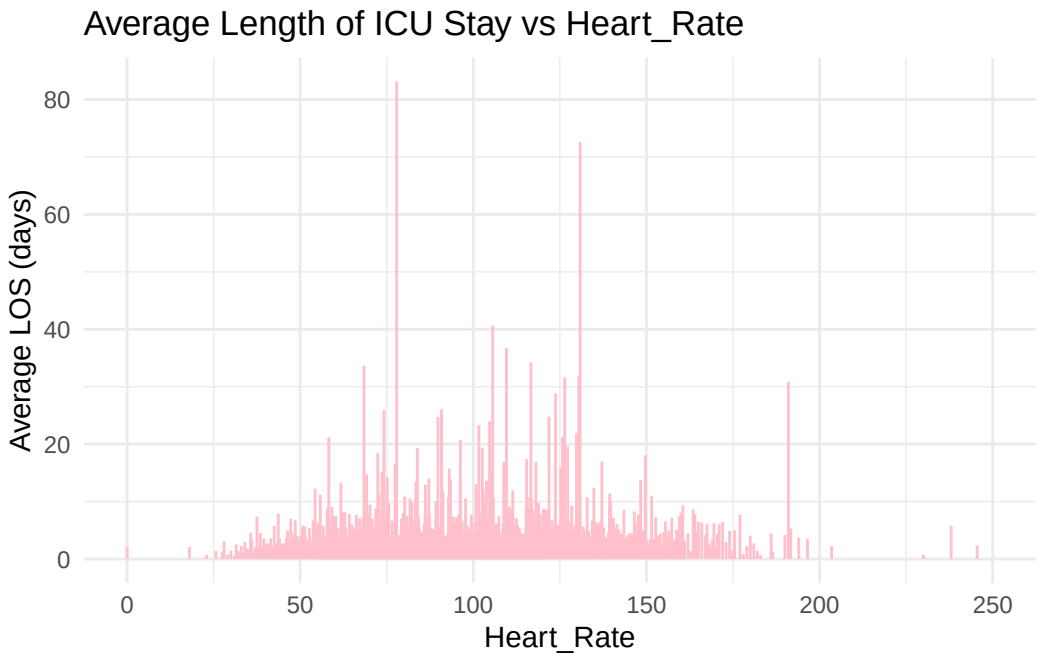
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




**LOS vs. Vital Heart Rate**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Heart_Rate) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Heart_Rate, y = mean_los )) +
   geom_col(color = "pink") +
  labs(title = "Average Length of ICU Stay vs Heart_Rate",
       x = "Heart_Rate", y = "Average LOS (days)") +
  theme_minimal() +
  coord_cartesian(xlim = c(0, 250)
  )
```

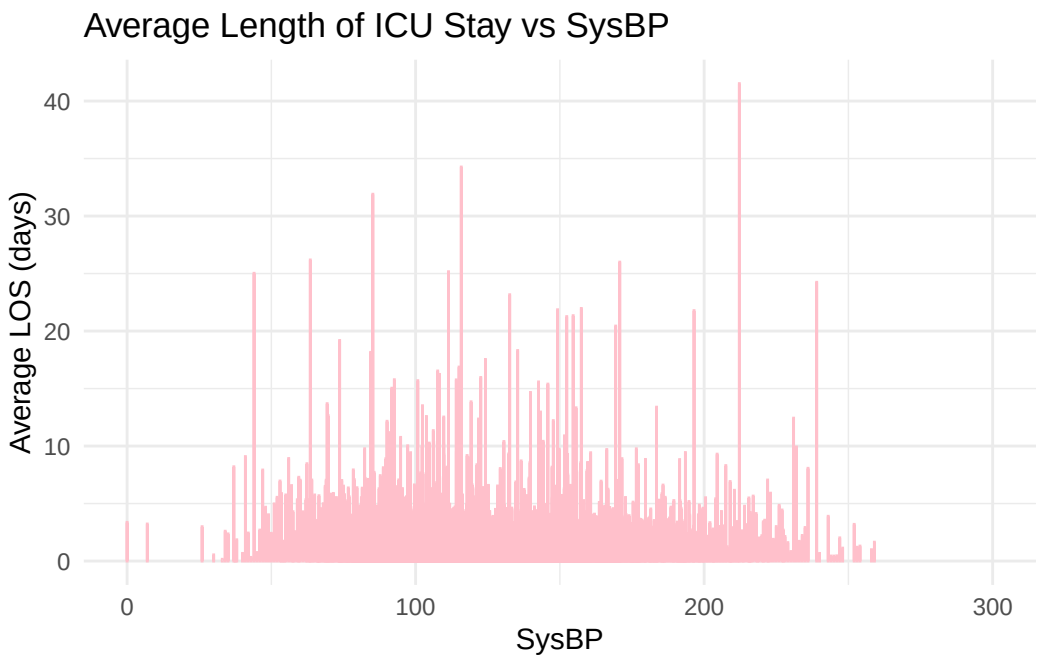
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




**LOS vs. Vital SysBP**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(SysBP) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = SysBP, y = mean_los )) +
   geom_col(color = "pink") +
  labs(title = "Average Length of ICU Stay vs SysBP",
       x = "SysBP", y = "Average LOS (days)") +
  theme_minimal() +
  coord_cartesian(xlim = c(0, 300)
  )
```

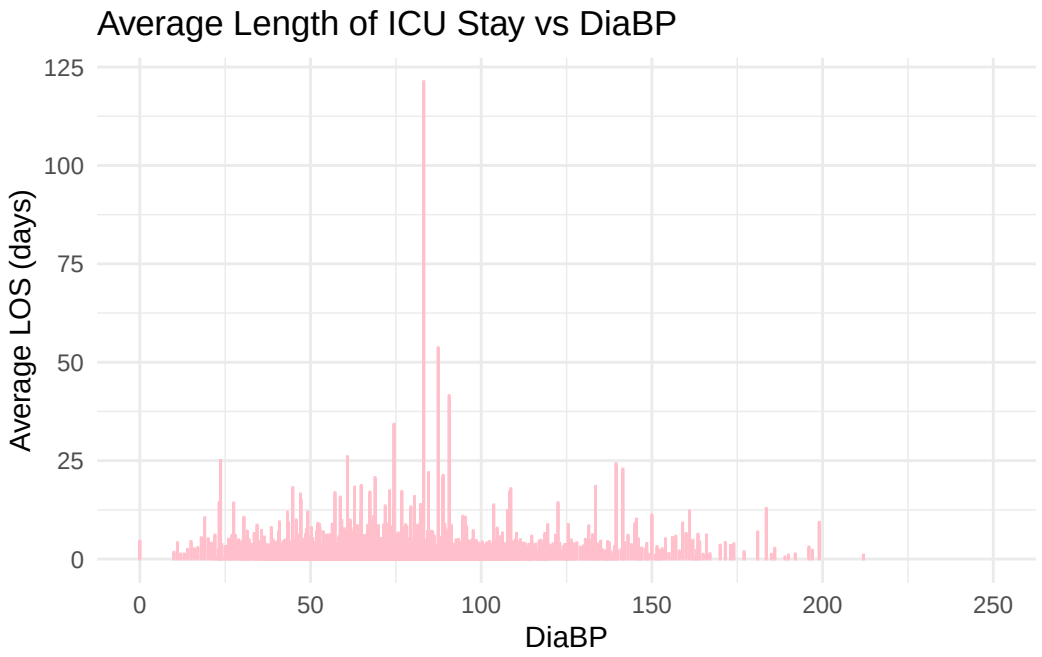
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




### LOS vs. Vital DiaBP

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(DiaBP) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = DiaBP, y = mean_los )) +
   geom_col(color = "pink") +
  labs(title = "Average Length of ICU Stay vs DiaBP",
       x = "DiaBP", y = "Average LOS (days)") +
  theme_minimal() +
  coord_cartesian(xlim = c(0, 250)
  )
```

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```

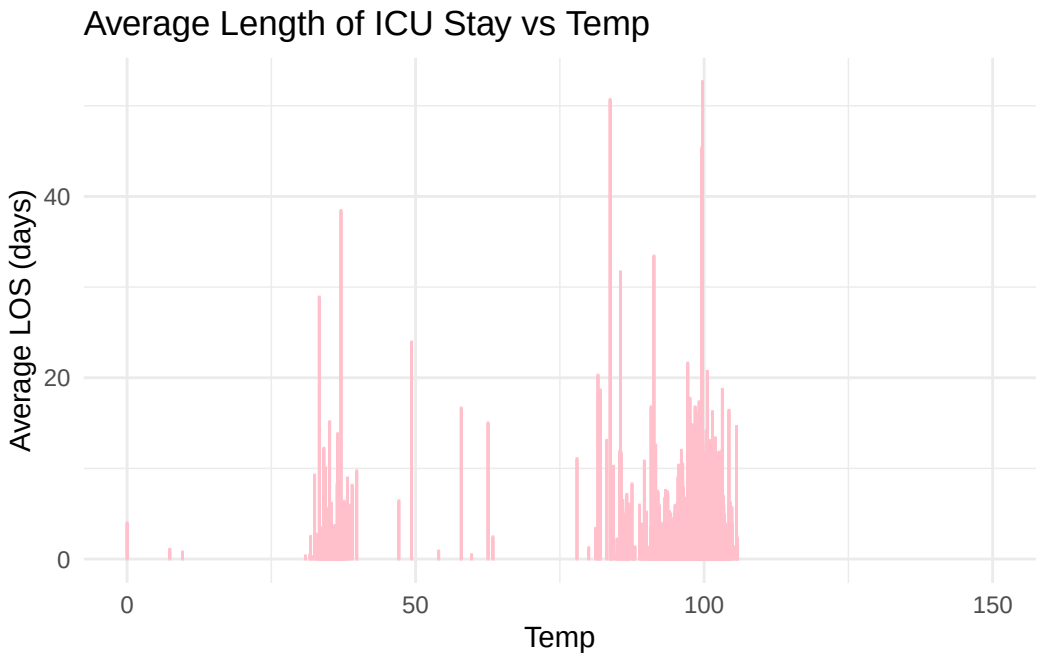



**LOS vs. Vital Temp**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Temp) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Temp, y = mean_los )) +
   geom_col(color = "pink") +
  labs(title = "Average Length of ICU Stay vs Temp",
       x = "Temp", y = "Average LOS (days)") +
  theme_minimal() +
  coord_cartesian(xlim = c(0, 150)
  )
```

```
Warning: `position_stack()` requires non-overlapping x intervals.
```

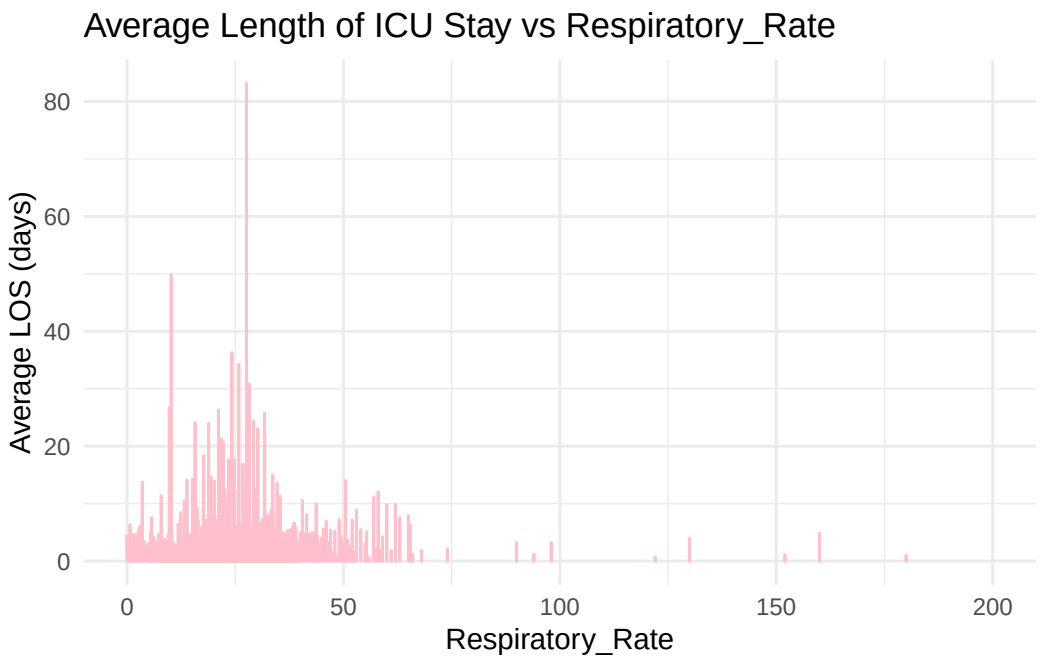
```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




**LOS vs. Vital Respiratory_Rate**

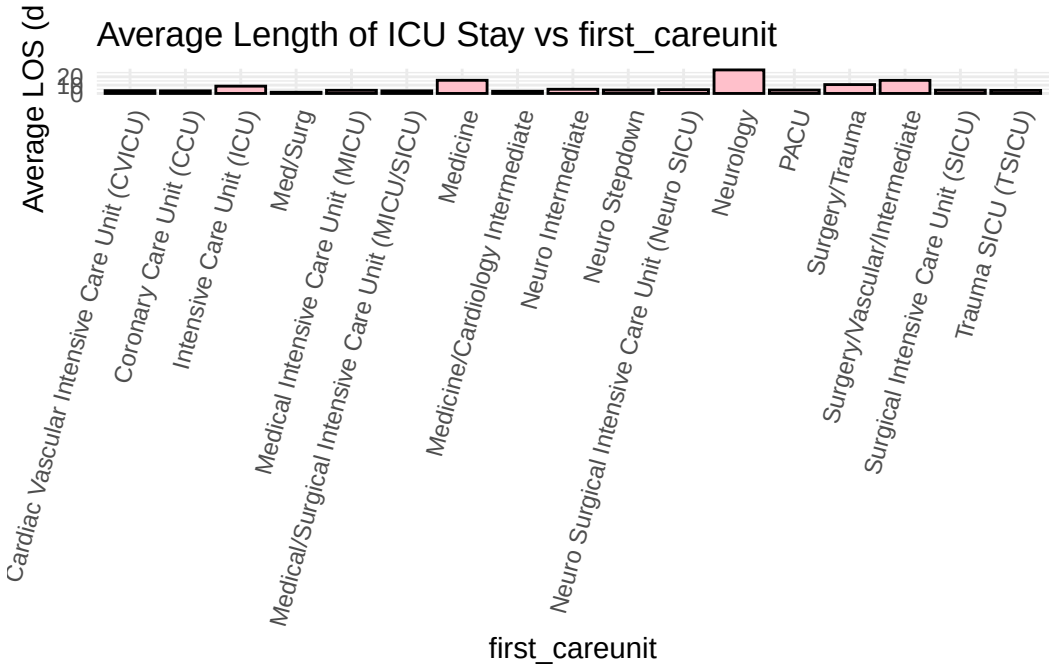
```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(Respiratory_Rate) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = Respiratory_Rate, y = mean_los)) +
   geom_col(color = "pink") +
  labs(title = "Average Length of ICU Stay vs Respiratory_Rate",
       x = "Respiratory_Rate", y = "Average LOS (days)") +
  theme_minimal() +
  coord_cartesian(xlim = c(0, 200)
  )
```

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```




**LOS vs. First ICU_Unit**

```
mimic_icu_summary <- mimic_icu_cohort %>%
  group_by(first_careunit) %>%
  summarise(mean_los = mean(los, na.rm = TRUE))

ggplot(mimic_icu_summary, aes(x = fct_infreq(first_careunit), y = mean_los )) +
   geom_col(fill = "pink", color = "black") +
  labs(title = "Average Length of ICU Stay vs first_careunit",
       x = "first_careunit", y = "Average LOS (days)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 75, hjust = 1))
```


Average Length of ICU Stay vs first_careunit
Average LOS (d
20
0
Cardiac Vascular Intensive Care Unit (CVICU)
Coronary Care Unit (CCU)
Intensive Care Unit (ICU)
Med/Surg
Medical Intensive Care Unit (MICU)
Medical/Surgical Intensive Care Unit (MICU/SICU)
Medicine
Medicine/Cardiology Intermediate
Neuro Intermediate
Neuro Stepdown
Neuro Surgical Intensive Care Unit (Neuro SICU)
Neurology
PACU
Surgery/Trauma
Surgery/Vascular/Intermediate
Surgical Intensive Care Unit (SICU)
Trauma SICU (TSICU)
first_careunit