# Homework 4

## Homework 04

Name: Jama Brookes

For questions 2-6, please use hw4.zip, which contains a data base of patient/hopsital data.

```
#convert_ipynb("./BIOS512_HW4.ipynb", output = xfun::with_ext("./BIOS512_HW4.ipynb", "Rmd"))
```

### Question 1

*For this question, you can either import these tables into R and do each join, or create the tables we expect to see in a Markdown cell.*
Please see the tables below.

```
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   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
table_a <- tibble(
  SKU = c(102345, 104567, 108912, 109876, 112233),
  Fruit = c("Apple", "Orange", "Mango", "Blueberry", "Watermelon"),
  Color = c("Red", "Orange", "Yellow", "Blue", "Green"),
  Price = c(1.20, 1.40, 1.70, 3.50, 4.40),
  In_Stock = c("Yes", "Yes", "No", "Yes", "No")
)

table_b <- tibble(
  SKU = c(102345, 105432, 106789, 104567, 107654),
  Fruit = c("Apple", "Banana", "Grape", "Orange", "Pear"),
  Color = c("Red", "Yellow", "Purple", "Orange", "Green"),
  Sale_Price = c(1.00, 0.50, 2.00, 1.20, 1.10),
  Number_in_Stock = c(50, 120, 0, 75, 0)
)
```

What would the result be if you did...
a) Left join
b) Right join
c) Inner join
d) Full join
e) Semi join
f) Anti join

```
#a) left join:
table_a %>% left_join(table_b, by = "SKU")
```

```
## # A tibble: 5 x 9
##      SKU Fruit.x    Color.x Price In_Stock Fruit.y Color.y Sale_Price
##    <dbl> <chr>      <chr>   <dbl> <chr>    <chr>   <chr>        <dbl>
## 1 102345 Apple      Red       1.2 Yes      Apple   Red            1
## 2 104567 Orange     Orange    1.4 Yes      Orange  Orange         1.2
## 3 108912 Mango      Yellow    1.7 No       <NA>    <NA>          NA
## 4 109876 Blueberry  Blue      3.5 Yes      <NA>    <NA>          NA
## 5 112233 Watermelon Green     4.4 No       <NA>    <NA>          NA
## # i 1 more variable: Number_in_Stock <dbl>
```

```
#all rows in table_a will be kept according to SKU and table_b rows will be dropped that do not match S
```

```
#b) right join
table_a %>% right_join(table_b, by = "SKU")
```

```
## # A tibble: 5 x 9
##      SKU Fruit.x Color.x Price In_Stock Fruit.y Color.y Sale_Price
##    <dbl> <chr>   <chr>   <dbl> <chr>    <chr>   <chr>        <dbl>
## 1 102345 Apple   Red       1.2 Yes      Apple   Red            1
## 2 104567 Orange  Orange    1.4 Yes      Orange  Orange         1.2
## 3 105432 <NA>    <NA>     NA   <NA>     Banana  Yellow         0.5
## 4 106789 <NA>    <NA>     NA   <NA>     Grape   Purple         2
## 5 107654 <NA>    <NA>     NA   <NA>     Pear    Green          1.1
## # i 1 more variable: Number_in_Stock <dbl>
```

```
#all rows in table_b will be kept according to SKU and table_a rows will be dropped that do not match S
```

```
#c) Inner join
table_a %>% inner_join(table_b, by = "SKU")
```

```
## # A tibble: 2 x 9
##      SKU Fruit.x Color.x Price In_Stock Fruit.y Color.y Sale_Price
##    <dbl> <chr>   <chr>   <dbl> <chr>    <chr>   <chr>        <dbl>
## 1 102345 Apple   Red       1.2 Yes      Apple   Red            1
## 2 104567 Orange  Orange    1.4 Yes      Orange  Orange         1.2
## # i 1 more variable: Number_in_Stock <dbl>
```

```
#only rows with matching information in x and y will be kept by SKU
```

```r
#d) Full join
table_a %>% full_join(table_b, by = "SKU")
```

```
## # A tibble: 8 x 9
##      SKU Fruit.x    Color.x Price In_Stock Fruit.y Color.y Sale_Price
##    <dbl> <chr>      <chr>   <dbl> <chr>    <chr>   <chr>        <dbl>
## 1 102345 Apple      Red       1.2 Yes      Apple   Red            1
## 2 104567 Orange     Orange    1.4 Yes      Orange  Orange         1.2
## 3 108912 Mango      Yellow    1.7 No       <NA>    <NA>          NA
## 4 109876 Blueberry  Blue      3.5 Yes      <NA>    <NA>          NA
## 5 112233 Watermelon Green     4.4 No       <NA>    <NA>          NA
## 6 105432 <NA>       <NA>     NA   <NA>     Banana  Yellow         0.5
## 7 106789 <NA>       <NA>     NA   <NA>     Grape   Purple         2
## 8 107654 <NA>       <NA>     NA   <NA>     Pear    Green          1.1
## # i 1 more variable: Number_in_Stock <dbl>
```

```r
#every row and column will be kept and missing values will be NA
```

```r
#e) Semi join
table_a %>% semi_join(table_b, by = "SKU")
```

```
## # A tibble: 2 x 5
##      SKU Fruit  Color  Price In_Stock
##    <dbl> <chr>  <chr>  <dbl> <chr>
## 1 102345 Apple  Red      1.2 Yes
## 2 104567 Orange Orange   1.4 Yes
```

```r
#all rows in table_a will be returned that have a match with table_b according to SKU
```

```r
#f) Anti join
table_a %>% anti_join(table_b, by = "SKU")
```

```
## # A tibble: 3 x 5
##      SKU Fruit      Color  Price In_Stock
##    <dbl> <chr>      <chr>  <dbl> <chr>
## 1 108912 Mango      Yellow   1.7 No
## 2 109876 Blueberry  Blue     3.5 Yes
## 3 112233 Watermelon Green    4.4 No
```

```r
#all rows in table_a will be returned that do not have a match with table_b according to SKU
```

## Question 2

Inspect the data sets in our database!
a) Import them.
b) Check out the columns and their variable types using one of R's tibble summary functions.

```r
#a) import
demographics <- read.csv("./hw4/demographics.csv")
full <- read.csv("./hw4/full.csv")
hospitals <- read.csv("./hw4/hospitals.csv")
patient_names <- read.csv("./hw4/patient_names.csv")
treatment_info <- read.csv("./hw4/treatment_info.csv")

#b) check out data
#demographics
head(demographics)
```

```
##   patient_id age gender     race    ethnicity
## 1       P001  51   Male Hispanic Non-Hispanic
## 2       P002  73   Male Hispanic Non-Hispanic
## 3       P003  49          White Non-Hispanic
## 4       P004   6  Other    White Non-Hispanic
## 5       P005  64  Other    White Non-Hispanic
## 6       P006  38  Other Hispanic Non-Hispanic
```

```r
#full
head(full)
```

```
##   patient_id                name age gender     race    ethnicity     condition
## 1       P001           Mary Hicks  51   Male Hispanic Non-Hispanic        Cancer
## 2       P002 Matthew Christensen  73   Male Hispanic Non-Hispanic Heart Disease
## 3       P003         Lisa Graham  49   <NA>    White Non-Hispanic        Asthma
## 4       P004          Greg Brown   6  Other    White Non-Hispanic Heart Disease
## 5       P005        Joshua Baker  64  Other    White Non-Hispanic Heart Disease
## 6       P006    Wendy Richardson  38  Other Hispanic Non-Hispanic        Asthma
##         treatment department hospital admission_date release_date
## 1    Chemotherapy   Oncology       H1     2024-09-30   2025-04-24
## 2  Bypass Surgery Cardiology       H5     2025-06-09   2025-09-04
## 3 Inhaler Therapy Pediatrics       H5     2025-09-08   2025-09-08
## 4  Bypass Surgery Cardiology       H3     2025-09-02   2025-09-06
## 5  Bypass Surgery Cardiology       H1     2025-02-23   2025-06-24
## 6 Inhaler Therapy Pediatrics       H3     2025-01-06   2025-05-14
##                 patient_address     patient_city patient_state patient_zipcode
## 1                          <NA>             <NA>          <NA>              NA
## 2  762 Hatfield Lights Apt. 887 North Thomasbury            WI           96149
## 3   25592 Foley Forge Suite 365      New Tiffany            IN           33286
## 4    1189 Swanson Pike Apt. 921   Underwoodburgh            NV            9762
## 5 81598 Chambers Mall Suite 136      Timothyfurt            HI           99546
## 6            1890 Norman Fields       Davidhaven            MS           87095
```

```r
str(full)
```

```
## 'data.frame':    35 obs. of  16 variables:
##  $ patient_id     : chr  "P001" "P002" "P003" "P004" ...
##  $ name           : chr  "Mary Hicks" "Matthew Christensen" "Lisa Graham" "Greg Brown" ...
##  $ age            : int  51 73 49 6 64 38 36 22 20 85 ...
##  $ gender         : chr  "Male" "Male" NA "Other" ...
##  $ race           : chr  "Hispanic" "Hispanic" "White" "White" ...
```

```
##  $ ethnicity      : chr  "Non-Hispanic" "Non-Hispanic" "Non-Hispanic" "Non-Hispanic" ...
##  $ condition      : chr  "Cancer" "Heart Disease" "Asthma" "Heart Disease" ...
##  $ treatment      : chr  "Chemotherapy" "Bypass Surgery" "Inhaler Therapy" "Bypass Surgery" ...
##  $ department     : chr  "Oncology" "Cardiology" "Pediatrics" "Cardiology" ...
##  $ hospital       : chr  "H1" "H5" "H5" "H3" ...
##  $ admission_date : chr  "2024-09-30" "2025-06-09" "2025-09-08" "2025-09-02" ...
##  $ release_date   : chr  "2025-04-24" "2025-09-04" "2025-09-08" "2025-09-06" ...
##  $ patient_address: chr  NA "762 Hatfield Lights Apt. 887" "25592 Foley Forge Suite 365" "1189 Swans
##  $ patient_city   : chr  NA "North Thomasbury" "New Tiffany" "Underwoodburgh" ...
##  $ patient_state  : chr  NA "WI" "IN" "NV" ...
##  $ patient_zipcode: int  NA 96149 33286 9762 99546 87095 4548 29439 35771 3346 ...
```

```
#patient_names
head(patient_names)
```

```
##   patient_id                name hospital_id condition_id
## 1       P001           Mary Hicks          H1            C
## 2       P002 Matthew Christensen          H5           HD
## 3       P003         Lisa Graham          H5            A
## 4       P004           Greg Brown          H3           HD
## 5       P005        Joshua Baker          H1           HD
## 6       P006    Wendy Richardson          H3            A
```

```
#hospitals
head(hospitals)
```

```
##   hospital_id            hospital_name hospital_address hospital_city
## 1          H1 Greenwood Medical Center     123 Maple St   Springfield
## 2          H2        Lakeside Hospital       456 Elm St       Madison
## 3          H3           Sunrise Health      789 Oak Ave   Los Angeles
## 4          H4  Valley General Hospital      321 Pine Rd        Denver
## 5          H5      Mountainview Clinic   654 Birch Blvd       Boulder
##   hospital_state hospital_zip_code
## 1             IL             62701
## 2             WI             53703
## 3             CA             90012
## 4             CO             80203
## 5             CO             80302
```

```
#treatment_info
head(treatment_info)
```

```
##   condition_id     condition              treatment  department
## 1           HD Heart Disease         Bypass Surgery  Cardiology
## 2            S        Stroke Rehabilitation Therapy   Neurology
## 3            C        Cancer           Chemotherapy    Oncology
## 4            F      Fracture                Surgery Orthopedics
## 5            A        Asthma        Inhaler Therapy  Pediatrics
```

## Question 3

Using the `full.csv` data set from our database, **pivot longer** by making all of the variables the same type. Use both `patient_ID` and `name` as ID variables. After pivoting, get a `tally` for number of observations per

patient ID/name. (*Hint: We did this in lecture 5!*)

```
full_long <- pivot_longer(full, age:patient_zipcode,
                          names_to = "property",
                          values_to = "observation",
                          values_transform = function(x)
                            ifelse(is.na(x), NA, as.character(x)))
head(full_long)
```

```
## # A tibble: 6 x 4
##   patient_id name       property  observation
##   <chr>      <chr>      <chr>     <chr>
## 1 P001       Mary Hicks age       51
## 2 P001       Mary Hicks gender    Male
## 3 P001       Mary Hicks race      Hispanic
## 4 P001       Mary Hicks ethnicity Non-Hispanic
## 5 P001       Mary Hicks condition Cancer
## 6 P001       Mary Hicks treatment Chemotherapy
```

```
#tally number of observations
full_long %>%
  group_by(name,patient_id) %>%
    tally() %>%
      arrange()
```

```
## # A tibble: 35 x 3
## # Groups:   name [35]
##    name             patient_id     n
##    <chr>            <chr>      <int>
##  1 Anthony Anderson P023          14
##  2 April Sanchez    P007          14
##  3 Ashley Johnson   P035          14
##  4 Casey Norman     P019          14
##  5 Dylan Lopez DVM  P009          14
##  6 Erica Foley      P032          14
##  7 Greg Brown       P004          14
##  8 Heather Chandler P016          14
##  9 Holly Contreras  P015          14
## 10 Holly Mclaughlin P034          14
## # i 25 more rows
```

## Question 4

Pivot longer by making one column per data type. Use both `patient_ID` and `name` as ID variables. After pivoting, get a `tally` for number of each type of observation per `patient ID/name`.

**Helpful Hints:**
1. You're performing 3 seperate pivots with careful column selection then joining them after!

2. After each pivot, add the code below to create a unique row number:

```
%>%
group_by(patient_id, name) %>%
  mutate(row = row_number()) %>%
  ungroup()
```

3. To create the tally, add what is below after your grouping statement:

```
%>%
summarise(
    n_chr  = sum(!is.na(value_chr)),
    n_num  = sum(!is.na(value_num)),
    n_date = sum(!is.na(value_date)),
    .groups = "drop"
```

```
chr_col <- c('gender', 'race', 'ethnicity', 'condition', 'treatment', 'department', 'hospital',
             'patient_address', 'patient_city', 'patient_state')
int_col <- c('age', 'patient_zipcode')
date_col <- c('admission_date', 'release_date')

#updating date columns to be date variable types
full$admission_date <- as.Date(ifelse(full$admission_date  > Sys.Date(),
  format(full$admission_date , "19%y-%m-%d"),
  format(full$admission_date )))
full$release_date <- as.Date(ifelse(full$release_date > Sys.Date(),
  format(full$release_date, "19%y-%m-%d"),
  format(full$release_date)))

#pivoting character columns
full_chr_long <- full %>% pivot_longer(
                          cols = all_of(chr_col),
                          names_to = "chr_col",
                          values_to = "value_chr")
full_chr_long <- full_chr_long %>%
group_by(patient_id, name) %>%
  mutate(row = row_number()) %>%
  ungroup()

full_chr_long <- full_chr_long %>%
                    select(!(age:patient_zipcode)) #removing unwanted columns

#pivoting integer columns
full_int_long <- full %>% pivot_longer(
                          cols = all_of(int_col),
                          names_to = "int_col",
                          values_to = "value_num")
full_int_long <- full_int_long %>%
group_by(patient_id, name) %>%
  mutate(row = row_number()) %>%
  ungroup()

full_int_long <- full_int_long %>%
                    select(!(gender:patient_state)) #removing unwanted columns
```

```
#pivoting date columns
full_date_long <- full %>% pivot_longer(
                          cols = all_of(date_col),
                          names_to = "dat_col",
                          values_to = "value_date")
full_date_long <- full_date_long %>%
group_by(patient_id) %>%
  mutate(row = row_number()) %>%
  ungroup()

full_date_long <- full_date_long %>%
                    select(!(age:patient_zipcode)) #removing unwanted columns


full_long_type <- full_join(full_chr_long, full_int_long, full_date_long,
                            by = c("patient_id", "name", "row")) %>%
                      full_join(full_date_long, by = c("patient_id", "name", "row"))

full_long_type %>%
    group_by(patient_id, name) %>%
    summarise(
        n_chr  = sum(!is.na(value_chr)),
        n_num  = sum(!is.na(value_num)),
        n_date = sum(!is.na(value_date)),
        .groups = "drop")
```

```
## # A tibble: 35 x 5
##    patient_id name                n_chr n_num n_date
##    <chr>      <chr>               <int> <int>  <int>
##  1 P001       Mary Hicks              7     1      2
##  2 P002       Matthew Christensen    10     2      2
##  3 P003       Lisa Graham             9     2      2
##  4 P004       Greg Brown             10     2      2
##  5 P005       Joshua Baker           10     2      2
##  6 P006       Wendy Richardson       10     2      2
##  7 P007       April Sanchez          10     2      2
##  8 P008       Melinda Moody          10     2      2
##  9 P009       Dylan Lopez DVM        10     2      2
## 10 P010       Maria Bruce            10     2      2
## # i 25 more rows
```

## Question 5

Match patient names to the name of the hospital they were treated at.
*Hint: You'll need* `patient_names.csv` *and* `hospitals.csv`.

```
patient_hospitals <- patient_names %>%
                                    left_join(hospitals,
                                              by = "hospital_id") %>%
                                       select(c("name", "hospital_name"))
as_tibble(patient_hospitals)
```

```
## # A tibble: 35 x 2
##    name                hospital_name
##    <chr>               <chr>
##  1 Mary Hicks          Greenwood Medical Center
##  2 Matthew Christensen Mountainview Clinic
##  3 Lisa Graham         Mountainview Clinic
##  4 Greg Brown          Sunrise Health
##  5 Joshua Baker        Greenwood Medical Center
##  6 Wendy Richardson    Sunrise Health
##  7 April Sanchez       Mountainview Clinic
##  8 Melinda Moody       Sunrise Health
##  9 Dylan Lopez DVM     Greenwood Medical Center
## 10 Maria Bruce         Mountainview Clinic
## # i 25 more rows
```

## Question 6

Using joins, create a table that shows `patient_id`, `name`, `age`, `gender`, `condition`, and `treatment`.
*Hint: You'll need `patient_names.csv`, `demographics.csv`, and `treatment_info.csv`.*

```
patient_demo_dx <- full_join(patient_names, demographics, by = "patient_id") %>%
                        left_join(treatment_info, by = "condition_id") %>%
                            select(c('patient_id', 'name',
                                     'age', 'gender',
                                     'condition', 'treatment'))
as_tibble(patient_demo_dx)
```

```
## # A tibble: 35 x 6
##    patient_id name                  age gender   condition     treatment
##    <chr>      <chr>               <int> <chr>    <chr>         <chr>
##  1 P001       Mary Hicks             51 "Male"   Cancer        Chemotherapy
##  2 P002       Matthew Christensen    73 "Male"   Heart Disease Bypass Surgery
##  3 P003       Lisa Graham            49 ""       Asthma        Inhaler Therapy
##  4 P004       Greg Brown              6 "Other"  Heart Disease Bypass Surgery
##  5 P005       Joshua Baker           64 "Other"  Heart Disease Bypass Surgery
##  6 P006       Wendy Richardson       38 "Other"  Asthma        Inhaler Therapy
##  7 P007       April Sanchez          36 "Female" Asthma        Inhaler Therapy
##  8 P008       Melinda Moody          22 "Other"  Stroke        Rehabilitation T~
##  9 P009       Dylan Lopez DVM        20 "Male"   Asthma        Inhaler Therapy
## 10 P010       Maria Bruce            85 "Other"  Fracture      Surgery
## # i 25 more rows
```

## Question 7

Let's revisit the NOFORC workshop.
Below is what we completed in class on 9/9.
**Please note: This contains the skimr library. Make sure you install that package! See the link for instructions: https://github.com/rjenki/BIOS512#adding-packages-to-installr-later.**

For the columns that have a low (relative to this dataset, which has ~150,000 observation) number of unique values, create a table that lists these unique values in ascending order.

```r
#Answer to Question 7:

#arranging low count unique values in ascending order by count
df_unique <- as.data.frame(unique_vals)
colnames(df_unique) <- c("unique_value", "count")

filtered_unique_vals <- df_unique %>%
                          filter(count < 150) %>%
                            group_by(count) %>%
                             arrange()

as_tibble(filtered_unique_vals)
```

```
## # A tibble: 83 x 2
##    unique_value     count
##    <fct>            <int>
##  1 Aircraft           148
##  2 Balloon            130
##  3 Chinese Lantern?   100
##  4 Chinese Lantern     85
##  5 Planet/Star?        84
##  6 Starlink?           82
##  7 Camera Anomaly      78
##  8 Searchlight         65
##  9 Meteor?             63
## 10 Satellite?          46
## # i 73 more rows
```

```r
#arranging ascending by name of unique_value
low_unique <- function(x, threshold) {
    unique_vals2 <- table(df[[col]])
  filtered_vals <- unique_vals2[unique_vals2 < threshold]
  cat("\n--- Unique values in Ascending Order ---\n")
    print(sort(names(filtered_vals)))
}

low_unique(unique_vals, 150)
```

```
##
## --- Unique values in Ascending Order ---
##  [1] "Aircraft"              "Animal?"              "Aurora Borealis?"
##  [4] "Aurora?"               "Ball Lightning?"      "Balloon"
##  [7] "Balloons"              "Balloons?"            "Bat?"
## [10] "Bird"                  "Bird?"                "Birds"
## [13] "birds?"                "Birds?"               "Blimp"
## [16] "Blimp?"                "Boat?"                "Boats"
## [19] "Boats?"                "Camera Anomaly"       "Camera Anomaly?"
## [22] "Chinese Lantern"       "Chinese Lantern?"     "Chinese Lanterns"
## [25] "Chinese Lanterns?"     "Cloud"                "Cloud?"
## [28] "Comet"                 "Contrail"             "Contrail?"
## [31] "Debris?"               "Dream?"               "Drone"
## [34] "Drones?"               "Fireworks"            "Fireworks?"
## [37] "Flare?"                "Flares"               "Flares?"
```

```
## [40] "Green fishing lights"  "Headlights?"           "Helicopter?"
## [43] "Hoax"                  "Hoax?"                 "Insect"
## [46] "Insect web?"           "Insect?"               "Insects?"
## [49] "ISS"                   "ISS?"                  "Kite"
## [52] "Kite?"                 "Laser"                 "Laser?"
## [55] "Lightning"             "Lightning?"            "Meteor"
## [58] "Meteor?"               "Moon"                  "Moon?"
## [61] "Planet/Star?"          "Reflection"            "Reflection?"
## [64] "Rocket?"               "Satellite"             "Satellite?"
## [67] "Satellites"            "Satellites?"           "Searchlight"
## [70] "Searchlight?"          "shock cone???"         "Smoke"
## [73] "Smoke ring"            "Space Junk"            "Space Junk?"
## [76] "Spiderweb"             "Starlink (Racetrack)"  "Starlink (Racetrack)?"
## [79] "Starlink-Racetrack"    "Starlink?"             "Sundog?"
## [82] "Truck"                 "Unexplained"
```

## Question 8

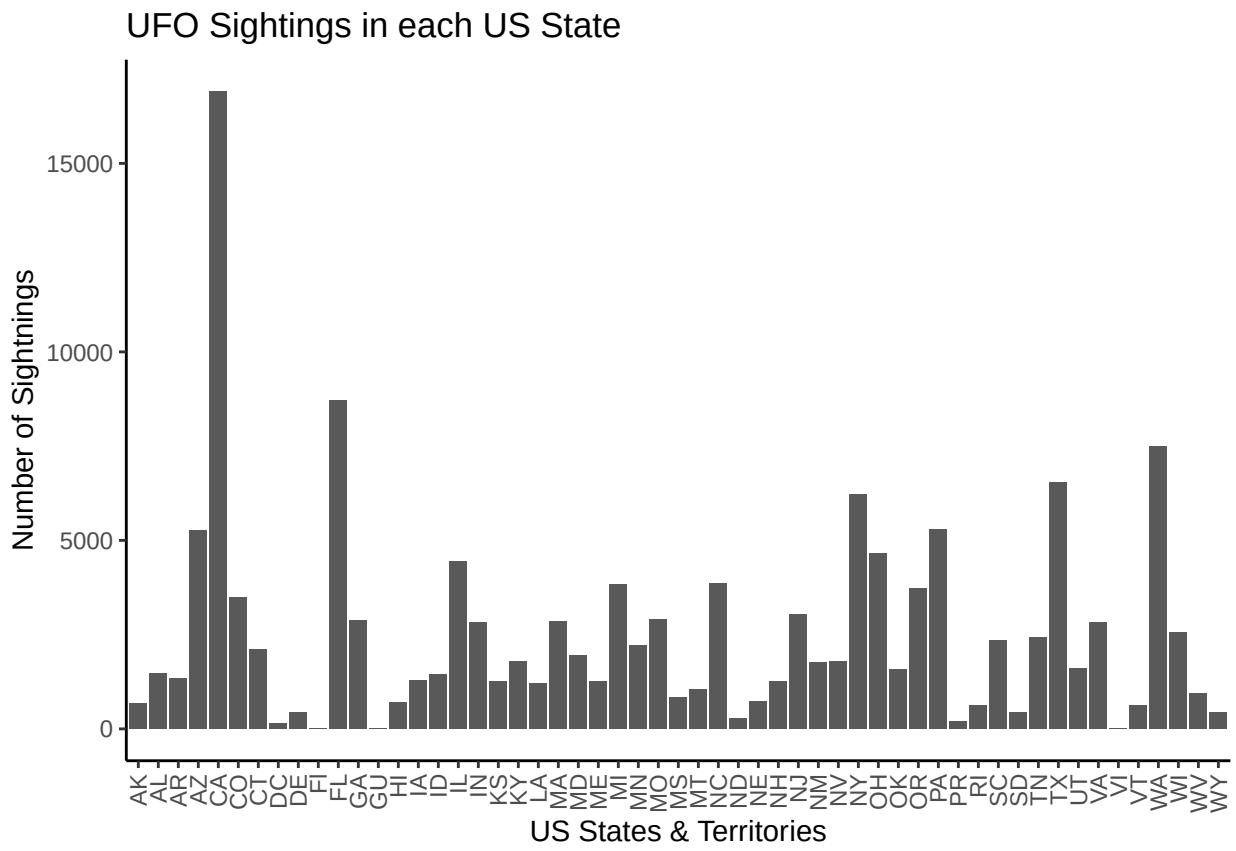
Make a plot of number of UFO sightings by state (United States only). You can filter out states that only have one observation.

```r
#filtering data to only USA
UFO_USA <- df %>%
            filter(country == "USA") %>%
            filter(state != "-") #filtering out state "-"

#counting state sightings and limiting the data to each observation > 1
UFO_state_count <- UFO_USA %>% count(state, sort= T) %>% filter(n > 1) %>% as.data.frame()

colnames(UFO_state_count) <- c("state", "count")

ggplot(UFO_state_count, aes(state, count)) +
  geom_bar(stat = "identity") +
  theme_classic() +
  theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5)) +
  labs(title = "UFO Sightings in each US State", y= "Number of Sightnings", x = "US States & Territorie
```

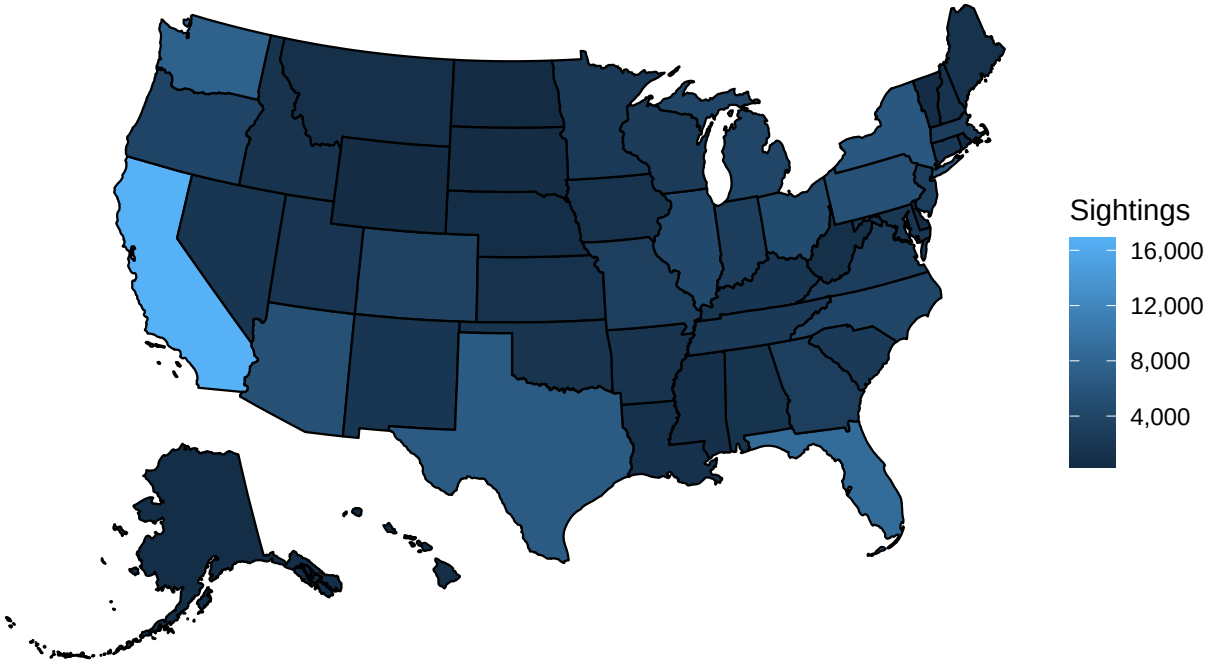



```
#plotting data on US map, excluding DC, PR, & US Territories

library(usmap)

UFO_state_count2 <- UFO_state_count %>%
  mutate(state = state.name[match(state, state.abb)])  #Converting "CA" to "California" for map data

#plotting US sightings on map
plot_usmap(data = UFO_state_count2, regions = "states", values = "count") +
  scale_fill_continuous(name = "Sightings", label = scales::comma) +
  theme_void() +
  labs(title = "UFO Sightings in US State", ) +
  theme(plot.title = element_text(hjust = 0.5))
```


UFO Sightings in US State
Sightings
16,000
12,000
8,000
4,000