Homework 04

Collin Stewart

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

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.

In [4]:
```
library(tidyverse)
library(tidyr)
library(dplyr)

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

Question 1a)

In [6]:
```
table_1a <- left_join(table_a, table_b, c("SKU", "Fruit", "Color"))
table_1a
```

A tibble: 5 × 7

SKU	Fruit	Color	Price	In_Stock	Sale_Price	Number_in_Stock
<dbl>	<chr>	<chr>	<dbl>	<chr>	<dbl>	<dbl>
102345	Apple	Red	1.2	Yes	1.0	50
104567	Orange	Orange	1.4	Yes	1.2	75
108912	Mango	Yellow	1.7	No	NA	NA
109876	Blueberry	Blue	3.5	Yes	NA	NA
112233	Watermelon	Green	4.4	No	NA	NA

Question 1b)

In [7]:
```
table_1b <- right_join(table_a, table_b, by = c("SKU", "Fruit", "Color"))
table_1b
```

A tibble: 5 × 7

SKU	Fruit	Color	Price	In_Stock	Sale_Price	Number_in_Stock
<dbl>	<chr>	<chr>	<dbl>	<chr>	<dbl>	<dbl>
102345	Apple	Red	1.2	Yes	1.0	50
104567	Orange	Orange	1.4	Yes	1.2	75
105432	Banana	Yellow	NA	NA	0.5	120
106789	Grape	Purple	NA	NA	2.0	0
107654	Pear	Green	NA	NA	1.1	0

Question 1c)

In [8]:
```
table_1c <- inner_join(table_a, table_b, by = c("SKU", "Fruit", "Color"))
table_1c
```

A tibble: 2 × 7

SKU	Fruit	Color	Price	In_Stock	Sale_Price	Number_in_Stock
<dbl>	<chr>	<chr>	<dbl>	<chr>	<dbl>	<dbl>
102345	Apple	Red	1.2	Yes	1.0	50
104567	Orange	Orange	1.4	Yes	1.2	75

Question 1d)

In [9]:
```
table_1d <- full_join(table_a, table_b, by = c("SKU", "Fruit", "Color"))
table_1d
```

A tibble: 8 × 7

SKU	Fruit	Color	Price	In_Stock	Sale_Price	Number_in_Stock
<dbl>	<chr>	<chr>	<dbl>	<chr>	<dbl>	<dbl>
102345	Apple	Red	1.2	Yes	1.0	50
104567	Orange	Orange	1.4	Yes	1.2	75
108912	Mango	Yellow	1.7	No	NA	NA
109876	Blueberry	Blue	3.5	Yes	NA	NA
112233	Watermelon	Green	4.4	No	NA	NA
105432	Banana	Yellow	NA	NA	0.5	120
106789	Grape	Purple	NA	NA	2.0	0
107654	Pear	Green	NA	NA	1.1	0

Question 1e)

In [12]:
```
table_1e <- semi_join(table_a, table_b, by = c("SKU", "Fruit", "Color"))
table_1e
```

A tibble: 2 × 5

SKU	Fruit	Color	Price	In_Stock
<dbl>	<chr>	<chr>	<dbl>	<chr>
102345	Apple	Red	1.2	Yes
104567	Orange	Orange	1.4	Yes

Question 1f)

In [14]:
```
table_1f <- anti_join(table_a, table_b, by = c("SKU", "Fruit", "Color"))
table_1f
```

A tibble: 3 × 5

SKU	Fruit	Color	Price	In_Stock
<dbl>	<chr>	<chr>	<dbl>	<chr>
108912	Mango	Yellow	1.7	No
109876	Blueberry	Blue	3.5	Yes
112233	Watermelon	Green	4.4	No

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.

In [15]:
```
demo <- read_csv("demographics.csv")
full <- read_csv("full.csv")
hosp <- read_csv("hospitals.csv")
pat_names <- read_csv("patient_names.csv")
trt_info <- read_csv("treatment_info.csv")
```

```
Rows: 35 Columns: 5
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (4): patient_id, gender, race, ethnicity
dbl (1): age

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
Rows: 35 Columns: 16
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (12): patient_id, name, gender, race, ethnicity, condition, treatment, ...
dbl   (2): age, patient_zipcode
date  (2): admission_date, release_date

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
Rows: 5 Columns: 6
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (5): hospital_id, hospital_name, hospital_address, hospital_city, hospit...
dbl (1): hospital_zip_code

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
Rows: 35 Columns: 4
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (4): patient_id, name, hospital_id, condition_id

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
Rows: 5 Columns: 4
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (4): condition_id, condition, treatment, department

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

In [16]:
```
summary(demo)
```

```
 patient_id            age          gender             race
Length:35          Min.   : 1.00   Length:35          Length:35
Class :character   1st Qu.:22.00   Class :character   Class :character
Mode  :character   Median :50.00   Mode  :character   Mode  :character
                   Mean   :45.18
                   3rd Qu.:69.25
                   Max.   :87.00
                   NA's   :1
 ethnicity
Length:35
Class :character
Mode  :character
```

In [17]:
```
summary(full)
```

```
 patient_id            name                age           gender
Length:35          Length:35          Min.   : 1.00   Length:35
Class :character   Class :character   1st Qu.:22.00   Class :character
Mode  :character   Mode  :character   Median :50.00   Mode  :character
                                      Mean   :45.18
                                      3rd Qu.:69.25
                                      Max.   :87.00
                                      NA's   :1
    race            ethnicity          condition          treatment
Length:35          Length:35          Length:35          Length:35
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character



 department          hospital         admission_date
Length:35          Length:35          Min.   :2024-09-13
Class :character   Class :character   1st Qu.:2024-12-20
Mode  :character   Mode  :character   Median :2025-02-20
                                      Mean   :2025-03-05
                                      3rd Qu.:2025-05-08
                                      Max.   :2025-09-08

 release_date        patient_address    patient_city       patient_state
Min.   :2024-12-06   Length:35          Length:35          Length:35
1st Qu.:2025-04-27   Class :character   Class :character   Class :character
Median :2025-06-05   Mode  :character   Mode  :character   Mode  :character
Mean   :2025-06-02
3rd Qu.:2025-08-01
Max.   :2025-09-08

patient_zipcode
Min.   : 3168
1st Qu.:33286
Median :68474
Mean   :58863
3rd Qu.:80463
Max.   :99546
NA's   :2
```

In [9]:
```
summary(hosp)
```

```
 hospital_id        hospital_name      hospital_address   hospital_city
 Length:5           Length:5           Length:5           Length:5
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character


 hospital_state     hospital_zip_code
 Length:5           Min.   :53703
 Class :character   1st Qu.:62701
 Mode  :character   Median :80203
                    Mean   :73384
                    3rd Qu.:80302
                    Max.   :90012
```

In [18]:
```
summary(pat_names)
```

```
 patient_id           name           hospital_id        condition_id
 Length:35          Length:35          Length:35          Length:35
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

In [19]:
```
summary(trt_info)
```

```
 condition_id        condition          treatment          department
 Length:5           Length:5           Length:5           Length:5
 Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

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!*)

In [20]:
```
full_longer <- pivot_longer(full,
        cols= -c(patient_id,name),
        names_to = "variable",
        values_to = "value",
        values_transform = function(x) ifelse(is.na(x), NA, as.character(x)))

full_longer %>%
    count(patient_id, name)
```

A tibble: 35 × 3

patient_id	name	n
<chr>	<chr>	<int>
P001	Mary Hicks	14
P002	Matthew Christensen	14
P003	Lisa Graham	14
P004	Greg Brown	14
P005	Joshua Baker	14
P006	Wendy Richardson	14
P007	April Sanchez	14
P008	Melinda Moody	14
P009	Dylan Lopez DVM	14
P010	Maria Bruce	14
P011	Kristine Lewis	14
P012	Jessica Ibarra	14
P013	Matthew Rogers	14
P014	Joseph Thompson	14
P015	Holly Contreras	14
P016	Heather Chandler	14
P017	John Brown	14
P018	Nathan Chase	14
P019	Casey Norman	14
P020	Nicholas Smith MD	14
P021	Mary Cobb	14
P022	Thomas Logan	14
P023	Anthony Anderson	14
P024	Matthew Jones	14
P025	Kathryn Harrison	14
P026	Jose Young	14
P027	Samuel Herrera	14
P028	Wanda Simmons	14
P029	Whitney Fuller	14

patient_id	name	n
<chr>	<chr>	<int>
P030	John Rodriguez	14
P031	John Ibarra	14
P032	Erica Foley	14
P033	Spencer Wells	14
P034	Holly Mclaughlin	14
P035	Ashley Johnson	14

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 greate 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"
```

In [21]:
```
full_chrcols <- pivot_longer(full,
        cols = c(gender:hospital, patient_address:patient_state),
        names_to = "variable",
        values_to = "value_chr",
        values_transform = list(value_chr = as.character)
    ) %>%
group_by(patient_id, name) %>%
    mutate(row = row_number()) %>%
    ungroup()
```

```
full_numcols <- pivot_longer(full,
        cols = c(age, patient_zipcode),
        names_to = "variable",
        values_to = "value_num",
        values_transform = list(value_num = as.numeric)
    ) %>%
group_by(patient_id, name) %>%
    mutate(row = row_number()) %>%
    ungroup()

full_datecols <- pivot_longer(full,
        cols = c(admission_date, release_date),
        names_to = "variable",
        values_to = "value_date",
        values_transform = list(value_date = as.Date)
    ) %>%
group_by(patient_id, name) %>%
    mutate(row = row_number()) %>%
    ungroup()



full_typecols <- full_chrcols %>%
    left_join(full_numcols, by = c("patient_id", "name", "row")) %>%
    left_join(full_datecols, by = c("patient_id", "name", "row"))


full_typecols %>%
    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 × 5

patient_id	name	n_chr	n_num	n_date
<chr>	<chr>	<int>	<int>	<int>
P001	Mary Hicks	7	1	2
P002	Matthew Christensen	10	2	2
P003	Lisa Graham	9	2	2
P004	Greg Brown	10	2	2
P005	Joshua Baker	10	2	2
P006	Wendy Richardson	10	2	2
P007	April Sanchez	10	2	2
P008	Melinda Moody	10	2	2
P009	Dylan Lopez DVM	10	2	2
P010	Maria Bruce	10	2	2
P011	Kristine Lewis	10	2	2
P012	Jessica Ibarra	10	2	2
P013	Matthew Rogers	10	2	2
P014	Joseph Thompson	10	2	2
P015	Holly Contreras	10	2	2
P016	Heather Chandler	8	2	2
P017	John Brown	10	2	2
P018	Nathan Chase	10	2	2
P019	Casey Norman	7	1	2
P020	Nicholas Smith MD	10	2	2
P021	Mary Cobb	10	2	2
P022	Thomas Logan	10	2	2
P023	Anthony Anderson	10	2	2
P024	Matthew Jones	10	2	2
P025	Kathryn Harrison	10	2	2
P026	Jose Young	10	2	2
P027	Samuel Herrera	10	2	2
P028	Wanda Simmons	10	2	2
P029	Whitney Fuller	10	2	2

patient_id	name	n_chr	n_num	n_date
<chr>	<chr>	<int>	<int>	<int>
P030	John Rodriguez	10	1	2
P031	John Ibarra	10	2	2
P032	Erica Foley	10	2	2
P033	Spencer Wells	10	2	2
P034	Holly Mclaughlin	10	2	2
P035	Ashley Johnson	10	2	2

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`.

In [28]:
```
patnames_hosp <- pat_names %>%
    left_join(hosp, by = "hospital_id")


patnames_hosp <- patnames_hosp %>%
select (-hospital_id)
patnames_hosp
```

A tibble: 35 × 8

patient_id	name	condition_id	hospital_name	hospital_address	hospital_city	hospital_s
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<
P001	Mary Hicks	C	Greenwood Medical Center	123 Maple St	Springfield	
P002	Matthew Christensen	HD	Mountainview Clinic	654 Birch Blvd	Boulder	
P003	Lisa Graham	A	Mountainview Clinic	654 Birch Blvd	Boulder	
P004	Greg Brown	HD	Sunrise Health	789 Oak Ave	Los Angeles	
P005	Joshua Baker	HD	Greenwood Medical Center	123 Maple St	Springfield	
P006	Wendy Richardson	A	Sunrise Health	789 Oak Ave	Los Angeles	
P007	April Sanchez	A	Mountainview Clinic	654 Birch Blvd	Boulder	
P008	Melinda Moody	S	Sunrise Health	789 Oak Ave	Los Angeles	
P009	Dylan Lopez DVM	A	Greenwood Medical Center	123 Maple St	Springfield	
P010	Maria Bruce	F	Mountainview Clinic	654 Birch Blvd	Boulder	
P011	Kristine Lewis	A	Valley General Hospital	321 Pine Rd	Denver	
P012	Jessica Ibarra	F	Lakeside Hospital	456 Elm St	Madison	
P013	Matthew Rogers	F	Valley General Hospital	321 Pine Rd	Denver	
P014	Joseph Thompson	F	Sunrise Health	789 Oak Ave	Los Angeles	
P015	Holly Contreras	HD	Greenwood Medical Center	123 Maple St	Springfield	
P016	Heather Chandler	A	Greenwood Medical Center	123 Maple St	Springfield	
P017	John Brown	A	Greenwood Medical Center	123 Maple St	Springfield	
P018	Nathan Chase	HD	Lakeside Hospital	456 Elm St	Madison	

patient_id	name	condition_id	hospital_name	hospital_address	hospital_city	hospital_s
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<
P019	Casey Norman	A	Greenwood Medical Center	123 Maple St	Springfield	
P020	Nicholas Smith MD	C	Greenwood Medical Center	123 Maple St	Springfield	
P021	Mary Cobb	S	Mountainview Clinic	654 Birch Blvd	Boulder	
P022	Thomas Logan	C	Valley General Hospital	321 Pine Rd	Denver	
P023	Anthony Anderson	F	Valley General Hospital	321 Pine Rd	Denver	
P024	Matthew Jones	A	Sunrise Health	789 Oak Ave	Los Angeles	
P025	Kathryn Harrison	F	Mountainview Clinic	654 Birch Blvd	Boulder	
P026	Jose Young	C	Mountainview Clinic	654 Birch Blvd	Boulder	
P027	Samuel Herrera	C	Lakeside Hospital	456 Elm St	Madison	
P028	Wanda Simmons	F	Mountainview Clinic	654 Birch Blvd	Boulder	
P029	Whitney Fuller	C	Sunrise Health	789 Oak Ave	Los Angeles	
P030	John Rodriguez	C	Valley General Hospital	321 Pine Rd	Denver	
P031	John Ibarra	C	Greenwood Medical Center	123 Maple St	Springfield	
P032	Erica Foley	C	Greenwood Medical Center	123 Maple St	Springfield	
P033	Spencer Wells	S	Mountainview Clinic	654 Birch Blvd	Boulder	
P034	Holly Mclaughlin	HD	Sunrise Health	789 Oak Ave	Los Angeles	
P035	Ashley Johnson	HD	Greenwood Medical Center	123 Maple St	Springfield	

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`.

In [23]:
```
full_demo <- pat_names %>%
    left_join(demo, by = "patient_id")

full_q6 <- full_demo %>%
    left_join(trt_info, by = "condition_id")

full_q6 %>%
    select(patient_id, name, age, gender, condition, treatment)
```

A tibble: 35 × 6

patient_id	name	age	gender	condition	treatment
<chr>	<chr>	<dbl>	<chr>	<chr>	<chr>
P001	Mary Hicks	51	Male	Cancer	Chemotherapy
P002	Matthew Christensen	73	Male	Heart Disease	Bypass Surgery
P003	Lisa Graham	49	NA	Asthma	Inhaler Therapy
P004	Greg Brown	6	Other	Heart Disease	Bypass Surgery
P005	Joshua Baker	64	Other	Heart Disease	Bypass Surgery
P006	Wendy Richardson	38	Other	Asthma	Inhaler Therapy
P007	April Sanchez	36	Female	Asthma	Inhaler Therapy
P008	Melinda Moody	22	Other	Stroke	Rehabilitation Therapy
P009	Dylan Lopez DVM	20	Male	Asthma	Inhaler Therapy
P010	Maria Bruce	85	Other	Fracture	Surgery
P011	Kristine Lewis	61	Female	Asthma	Inhaler Therapy
P012	Jessica Ibarra	23	Other	Fracture	Surgery
P013	Matthew Rogers	54	Female	Fracture	Surgery
P014	Joseph Thompson	22	Other	Fracture	Surgery
P015	Holly Contreras	29	Male	Heart Disease	Bypass Surgery
P016	Heather Chandler	74	Female	Asthma	Inhaler Therapy
P017	John Brown	81	Female	Asthma	Inhaler Therapy
P018	Nathan Chase	7	Other	Heart Disease	Bypass Surgery
P019	Casey Norman	28	Male	Asthma	Inhaler Therapy
P020	Nicholas Smith MD	67	Male	Cancer	Chemotherapy
P021	Mary Cobb	87	Female	Stroke	Rehabilitation Therapy
P022	Thomas Logan	1	Male	Cancer	Chemotherapy
P023	Anthony Anderson	70	Male	Fracture	Surgery
P024	Matthew Jones	75	Male	Asthma	Inhaler Therapy
P025	Kathryn Harrison	51	Male	Fracture	Surgery
P026	Jose Young	76	Other	Cancer	Chemotherapy
P027	Samuel Herrera	10	Female	Cancer	Chemotherapy
P028	Wanda Simmons	8	Female	Fracture	Surgery
P029	Whitney Fuller	2	Male	Cancer	Chemotherapy

patient_id	name	age	gender	condition	treatment
<chr>	<chr>	<dbl>	<chr>	<chr>	<chr>
P030	John Rodriguez	NA	Male	Cancer	Chemotherapy
P031	John Ibarra	75	Female	Cancer	Chemotherapy
P032	Erica Foley	47	Male	Cancer	Chemotherapy
P033	Spencer Wells	66	Male	Stroke	Rehabilitation Therapy
P034	Holly Mclaughlin	56	Other	Heart Disease	Bypass Surgery
P035	Ashley Johnson	22	Other	Heart Disease	Bypass Surgery

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.

In [29]:
```
# Load UFO sightings data from a GitHub CSV
df <- read_csv("https://raw.githubusercontent.com/Vincent-Toups/bios512/refs/heads/

# Read column names
names(df)

# Count the occurrences of each unique 'shape' value
unique_vals <- df$shape %>% table()

# Sort the counts of shapes in descending order and get the names
unique_vals %>% sort(decreasing = T) %>% names()

# Store column names in a vector
column_names <- names(df)

# Total number of rows in the dataset
n_total <- nrow(df)

# Loop over each column to get basic summary stats
for(col in column_names) {
  values <- df[[col]];        # Extract column
  n_na <- sum(is.na(values))  # Count number of NA values

  unique_vals <- values %>% table() %>% sort(decreasing = T)  # Count unique values
  n_unique <- length(unique_vals)

  cat(sprintf("%s:\n", col))  # Print column name
  cat(sprintf("\tnumber of NA values %d (%0.2f %%)\n", n_na, 100*n_na/n_total)) # P
  if(n_unique < 150) cat(sprintf("\t\t%s\n", names(unique_vals) %>% paste(collapse=
  cat(sprintf("\tnumber of unique values %d (%0.2f %%)\n", length(unique_vals), # P
```

```
    100*length(unique_vals)/n_total))
}

# Count number of reports per state and sort ascending
df %>% group_by(state) %>% tally() %>% arrange(n)

# Extract the 'occurred' column as a vector
df %>% pull(occurred)

# Helper function: nth(n) returns a function that extracts the nth element of a vec
nth <- function(n) function(a) a[n]

# Custom function to parse date strings by splitting on - / space : characters
parse_date <- function(s){
                          space_split <- s %>% str_split("[-/ :]")
                          tibble(d1 = Map(nth(1), space_split) %>% as.character(),
                                     d2 = Map(nth(2), space_split) %>% as.characte
                                     d3 = Map(nth(3), space_split) %>% as.characte
                                     d4 = Map(nth(4), space_split) %>% as.characte
                                     d5 = Map(nth(5), space_split) %>% as.characte
                          }

# Apply the parsing function to the 'occurred' column
date_stuff <- parse_date(df %>% pull(occurred))
head(date_stuff, 10)

# Histogram of the second component of the split date (likely month)
ggplot (date_stuff, aes(d2))+ geom_bar() + labs(x = "Month", y = "Count")

# Install and load the skimr package for a nicer summary
library(skimr)

# Quick summary of the dataset
skim_output <- skimr::skim(df)

# Count occurrences for categorical columns
df %>% count(country, sort = TRUE)
df %>% count(state, sort = TRUE)
df %>% count(shape, sort = TRUE)

# Convert 'occurred' and 'reported' to proper date-time format using lubridate
df <- df %>%
  mutate(
  occurred = lubridate::mdy_hm(occurred, quiet = TRUE),
  reported = lubridate::mdy_hm(reported, quiet = TRUE)
  )

# Plot UFO sightings per year
df %>%
  filter(!is.na(occurred)) %>%
  count(year = lubridate::year(occurred)) %>%
  ggplot(aes(year, n)) +
  geom_line() +
    labs(title = "UFO Sightings per Year", x = "Year", y = "Number of Reports")
```

```
Rows: 156711 Columns: 11
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (10): link_url, occurred, city, state, country, shape, summary, reported...
dbl  (1): id

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

'id' · 'link_url' · 'occurred' · 'city' · 'state' · 'country' · 'shape' · 'summary' · 'reported' · 'has_image' · 'explanation'

'Light' · 'Circle' · 'Triangle' · 'Unknown' · 'Other' · 'Fireball' · 'Disk' · 'Sphere' · 'Orb' · 'Oval' · 'Formation' · 'Changing' · 'Cigar' · 'Rectangle' · 'Cylinder' · 'Flash' · 'Diamond' · 'Chevron' · 'Egg' · 'Teardrop' · 'Cone' · 'Cross' · 'Star' · 'Cube' · 'light' · 'other' · 'triangle' · 'circle' · 'sphere' · 'cylinder' · 'rectangle' · 'cigar' · 'diamond' · 'fireball' · 'oval' · 'changing' · 'egg' · 'flash' · 'unknown'

```
id:
        number of NA values 0 (0.00 %)
        number of unique values 156711 (100.00 %)
link_url:
        number of NA values 0 (0.00 %)
        number of unique values 156711 (100.00 %)
occurred:
        number of NA values 299 (0.19 %)
        number of unique values 134472 (85.81 %)
city:
        number of NA values 823 (0.53 %)
        number of unique values 31884 (20.35 %)
state:
        number of NA values 9105 (5.81 %)
        number of unique values 975 (0.62 %)
country:
        number of NA values 0 (0.00 %)
        number of unique values 406 (0.26 %)
shape:
        number of NA values 6343 (4.05 %)
                Light, Circle, Triangle, Unknown, Other, Fireball, Disk, Sphere, Or
b, Oval, Formation, Changing, Cigar, Rectangle, Cylinder, Flash, Diamond, Chevron, E
gg, Teardrop, Cone, Cross, Star, Cube, light, other, triangle, circle, sphere, cylin
der, rectangle, cigar, diamond, fireball, oval, changing, egg, flash, unknown
        number of unique values 39 (0.02 %)
summary:
        number of NA values 74 (0.05 %)
        number of unique values 153832 (98.16 %)
reported:
        number of NA values 0 (0.00 %)
        number of unique values 10759 (6.87 %)
has_image:
        number of NA values 149133 (95.16 %)
                Y
        number of unique values 1 (0.00 %)
explanation:
        number of NA values 153546 (97.98 %)
                Drone?, Rocket, Starlink, Balloon?, Aircraft?, Planet/Star, Aircraf
t, Balloon, Chinese Lantern?, Chinese Lantern, Planet/Star?, Starlink?, Camera Anoma
ly, Searchlight, Meteor?, Satellite?, Rocket?, Bird?, Drone, Meteor, Contrail, Satel
lite, Camera Anomaly?, Birds?, Bird, Insect?, Contrail?, Insect, Searchlight?, Ballo
ons, Starlink (Racetrack), Starlink (Racetrack)?, Flares?, Reflection, Blimp, Cloud,
Cloud?, Birds, Satellites?, Unexplained, Hoax?, Chinese Lanterns, Hoax, ISS, Moon, C
hinese Lanterns?, Fireworks?, ISS?, Laser, Reflection?, Space Junk, Balloons?, Blim
p?, Drones?, Flares, Kite, Kite?, Laser?, Lightning, Satellites, Animal?, Aurora Bor
ealis?, Aurora?, Ball Lightning?, Bat?, birds?, Boat?, Boats, Boats?, Comet, Debri
s?, Dream?, Fireworks, Flare?, Green fishing lights, Headlights?, Helicopter?, Insec
t web?, Insects?, Lightning?, Moon?, shock cone???, Smoke, Smoke ring, Space Junk?,
Spiderweb, Starlink-Racetrack, Sundog?, Truck
        number of unique values 89 (0.06 %)
```

A tibble: 976 × 2

state	n
<chr>	<int>
0	1
Abu Dhabi	1
Adana Province	1
Addis Ababa	1
Adjara	1
Administrative-Territorial Units of the Left Bank	1
Afyonkarahisar	1
Agder	1
Akita	1
Al Ahmadi Governorate	1
Al Anbar Governorate	1
Al Farwaniyah	1
Alagoas	1
Alicante	1
Almería Province	1
Alytaus apskritis	1
Alytus County	1
Amhara	1
Andreas	1
Antrim	1
Antrim and Newtownabbey	1
Aosta Valley	1
Appenzell Ausserrhoden	1
Apulia	1
Armagh City and District Council	1
Astana	1
Asunción	1
Asyut	1
Atlántico Department	1

state	n
<chr>	<int>
Auvergne-Rhône-Alpes	1
⋮	⋮
NM	1758
NV	1785
KY	1793
MD	1954
CT	2111
MN	2229
SC	2347
TN	2439
WI	2566
ON	2660
VA	2838
IN	2839
MA	2841
GA	2889
MO	2908
NJ	3036
CO	3489
OR	3732
MI	3834
NC	3852
IL	4446
OH	4650
AZ	5267
PA	5292
NY	6224
TX	6548
WA	7510

state	n
<chr>	<int>
FL	8717
NA	9105
CA	16913

'08/31/2025 21:00' · '08/31/2025 02:30' · '08/30/2025 11:30' · '08/30/2025 02:30' ·
'08/19/2025 19:00' · '08/13/2025 19:40' · '08/13/2025 16:22' · '08/13/2025 04:40' ·
'08/13/2025 04:30' · '08/13/2025 03:00' · '08/13/2025 01:58' · '08/13/2025 00:48' ·
'08/12/2025 23:28' · '08/12/2025 22:50' · '08/12/2025 22:45' · '08/12/2025 22:35' ·
'08/12/2025 22:34' · '08/12/2025 22:33' · '08/12/2025 22:30' · '08/12/2025 22:30' ·
'08/12/2025 21:40' · '08/12/2025 21:40' · '08/12/2025 21:38' · '08/12/2025 20:35' ·
'08/12/2025 15:30' · '08/12/2025 09:25' · '08/12/2025 04:34' · '08/12/2025 02:30' ·
'08/12/2025 01:30' · '08/12/2025 00:00' · '08/11/2025 23:45' · '08/11/2025 23:30' ·
'08/11/2025 23:00' · '08/11/2025 22:00' · '08/11/2025 21:10' · '08/11/2025 20:47' ·
'08/11/2025 13:00' · '08/11/2025 12:00' · '08/11/2025 11:14' · '08/11/2025 07:40' ·
'08/11/2025 07:00' · '08/11/2025 04:30' · '08/11/2025 03:49' · '08/11/2025 03:00' ·
'08/11/2025 01:35' · '08/10/2025 23:45' · '08/10/2025 23:45' · '08/10/2025 21:45' ·
'08/10/2025 21:37' · '08/10/2025 21:30' · '08/10/2025 21:30' · '08/10/2025 21:20' ·
'08/10/2025 20:56' · '08/10/2025 19:50' · '08/10/2025 11:15' · '08/10/2025 03:45' ·
'08/09/2025 23:00' · '08/09/2025 21:57' · '08/09/2025 21:31' · '08/09/2025 21:05' ·
'08/09/2025 21:00' · '08/09/2025 15:07' · '08/09/2025 12:00' · '08/09/2025 11:42' ·
'08/09/2025 05:50' · '08/09/2025 04:02' · '08/09/2025 02:00' · '08/09/2025 01:20' ·
'08/08/2025 21:30' · '08/08/2025 20:45' · '08/08/2025 18:15' · '08/08/2025 10:28' ·
'08/07/2025 22:30' · '08/07/2025 22:21' · '08/07/2025 21:55' · '08/07/2025 20:53' ·
'08/07/2025 04:00' · '08/07/2025 03:53' · '08/06/2025 23:34' · '08/06/2025 22:30' ·
'08/06/2025 14:50' · '08/06/2025 02:40' · '08/05/2025 22:09' · '08/05/2025 21:55' ·
'08/05/2025 17:00' · '08/05/2025 11:38' · '08/05/2025 08:35' · '08/05/2025 05:15' ·
'08/04/2025 23:57' · '08/04/2025 23:10' · '08/04/2025 22:54' · '08/04/2025 22:30' ·
'08/04/2025 22:24' · '08/04/2025 22:00' · '08/04/2025 21:45' · '08/04/2025 21:30' ·
'08/04/2025 20:35' · '08/04/2025 20:30' · '08/04/2025 05:07' · '08/04/2025 05:06' ·
'08/04/2025 04:30' · '08/04/2025 02:30' · '08/04/2025 02:30' · '08/04/2025 00:00' ·
'08/03/2025 23:46' · '08/03/2025 20:37' · '08/03/2025 16:19' · '08/03/2025 13:15' ·
'08/03/2025 10:30' · '08/03/2025 09:45' · '08/03/2025 04:30' · '08/03/2025 04:17' ·
'08/03/2025 03:55' · '08/03/2025 02:33' · '08/02/2025 23:50' · '08/02/2025 23:29' ·
'08/02/2025 22:50' · '08/02/2025 22:30' · '08/02/2025 22:00' · '08/02/2025 21:18' ·
'08/02/2025 21:02' · '08/02/2025 20:50' · '08/02/2025 10:50' · '08/02/2025 01:17' ·
'08/01/2025 22:51' · '08/01/2025 22:10' · '08/01/2025 21:00' · '08/01/2025 21:00' ·
'08/01/2025 20:28' · '08/01/2025 20:06' · '08/01/2025 15:33' · '08/01/2025 06:35' ·
'08/01/2025 04:30' · '08/01/2025 01:20' · '07/31/2025 22:40' · '07/31/2025 18:00' ·
'07/31/2025 05:07' · '07/31/2025 03:00' · '07/31/2025 00:15' · '07/31/2025 00:05' ·
'07/30/2025 22:30' · '07/30/2025 22:30' · '07/30/2025 22:26' · '07/30/2025 22:10' ·
'07/30/2025 21:09' · '07/30/2025 18:43' · '07/30/2025 18:12' · '07/30/2025 14:30' ·
'07/30/2025 05:40' · '07/30/2025 05:20' · '07/30/2025 04:02' · '07/30/2025 02:11' ·
'07/30/2025 02:00' · '07/30/2025 00:30' · '07/29/2025 23:46' · '07/29/2025 21:45' ·
'07/29/2025 21:30' · '07/29/2025 15:00' · '07/29/2025 11:40' · '07/28/2025 23:30' ·
'07/28/2025 22:39' · '07/28/2025 22:33' · '07/28/2025 22:20' · '07/28/2025 22:00' ·
'07/28/2025 20:39' · '07/28/2025 12:45' · '07/28/2025 04:19' · '07/28/2025 02:30' ·

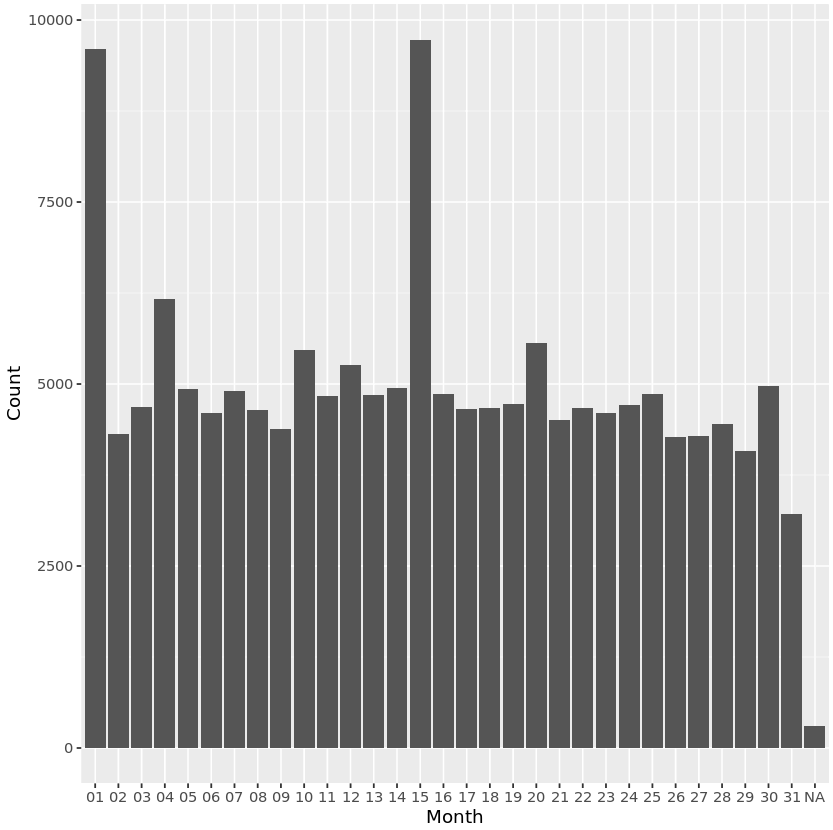
'07/27/2025 23:30' · '07/27/2025 22:30' · '07/27/2025 22:22' · '07/27/2025 22:15' · '07/27/2025 21:00' · '07/27/2025 19:35' · '07/27/2025 04:50' · '07/26/2025 23:40' · '07/26/2025 19:30' · '07/26/2025 15:40' · '07/26/2025 12:57' · '07/26/2025 11:00' · '07/26/2025 06:00' · '07/26/2025 05:00' · '07/26/2025 04:00' · '07/26/2025 02:30' · '07/25/2025 23:44' · '07/25/2025 23:30' · '07/25/2025 23:27' · '07/25/2025 23:06' · '07/25/2025 22:15' · '07/25/2025 22:00' · '07/25/2025 21:53' · '07/25/2025 21:52' · '07/25/2025 20:55' · '07/25/2025 13:02' · '07/25/2025 12:05' · '07/25/2025 12:00' · '07/25/2025 11:00' · '07/25/2025 04:00' · '07/25/2025 03:30' · '07/25/2025 01:30' · ⋯ · NA

A tibble: 10 × 5

d1	d2	d3	d4	d5
<chr>	<chr>	<chr>	<chr>	<chr>
08	31	2025	21	00
08	31	2025	02	30
08	30	2025	11	30
08	30	2025	02	30
08	19	2025	19	00
08	13	2025	19	40
08	13	2025	16	22
08	13	2025	04	40
08	13	2025	04	30
08	13	2025	03	00

```
Error in library(skimr): there is no package called ‘skimr’
Traceback:

1. library(skimr)
```



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.

In [30]:
```
print("Unique City Entries")
df %>% group_by(city) %>% tally() %>% arrange(n)

print("Unique State Entries")
df %>% group_by(state) %>% tally() %>% arrange(n)

print("Unique Country Entries")
df %>% group_by(country) %>% tally() %>% arrange(n)

print("Unique Shape Entries")
df %>% group_by(shape) %>% tally() %>% arrange(n)
```

```
[1] "Unique City Entries"
```

A tibble: 31885 × 2

city	n
<chr>	<int>
Moundville	1
((HOAX??))	1
((Location no revealed by witness))	1
((Location unspecified)) (UK/England)	1
((Location unspecified; rural area))	1
((Unknown))	1
((Unspecified by witness))	1
((Unspecified location))	1
((name of town deleted))	1
((town name temporarily deleted))	1
((unspecified by witness))	1
((unspecified))	1
(City not specified)	1
(City unknown)	1
(Norway)	1
(S. of) Bradford VT. -- milepost 93.0 on I-91	1
(Switzerland)	1
(Unspecified by witness)	1
(Unspecified location)	1
(Unspecified)	1
(above mountains in airplane)	1
(observed from airplane)	1
(unknown)	1
, Florissant, MO 63033	1
,stocton,on,tees (UK/ngland)	1
-	1
1-25 corridor (southbound, 65 miles north NM border)	1
100 Mile (Canada)	1
100 Mile House (Canada)	1

city	n
<chr>	<int>
12 miles east of Culbertson,Mt.	1
⋮	⋮
Charlotte	267
Louisville	275
Dallas	276
Indianapolis	281
Spokane	287
Colorado Springs	293
Salem	300
San Jose	301
San Antonio	312
Myrtle Beach	324
Sacramento	336
Boise	350
Jacksonville	364
Denver	368
Columbus	377
Austin	386
Miami	387
Springfield	393
Orlando	402
Albuquerque	413
Houston	457
Chicago	475
Tucson	507
San Diego	584
Los Angeles	608
Las Vegas	685
Portland	686

city	n
<chr>	<int>
Seattle	755
Phoenix	809
NA	823

```
[1] "Unique State Entries"
```

A tibble: 976 × 2

state	n
<chr>	<int>
0	1
Abu Dhabi	1
Adana Province	1
Addis Ababa	1
Adjara	1
Administrative-Territorial Units of the Left Bank	1
Afyonkarahisar	1
Agder	1
Akita	1
Al Ahmadi Governorate	1
Al Anbar Governorate	1
Al Farwaniyah	1
Alagoas	1
Alicante	1
Almería Province	1
Alytaus apskritis	1
Alytus County	1
Amhara	1
Andreas	1
Antrim	1
Antrim and Newtownabbey	1
Aosta Valley	1
Appenzell Ausserrhoden	1
Apulia	1
Armagh City and District Council	1
Astana	1
Asunción	1
Asyut	1
Atlántico Department	1

state	n
<chr>	<int>
Auvergne-Rhône-Alpes	1
⋮	⋮
NM	1758
NV	1785
KY	1793
MD	1954
CT	2111
MN	2229
SC	2347
TN	2439
WI	2566
ON	2660
VA	2838
IN	2839
MA	2841
GA	2889
MO	2908
NJ	3036
CO	3489
OR	3732
MI	3834
NC	3852
IL	4446
OH	4650
AZ	5267
PA	5292
NY	6224
TX	6548
WA	7510

state	n
<chr>	<int>
FL	8717
NA	9105
CA	16913

```
[1] "Unique Country Entries"
```

A tibble: 406 × 2

country	n
<chr>	<int>
Above the pacific ocean	1
Aegean Sea	1
Andaman Islands	1
Angola	1
Anguilla	1
Bahamas The	1
Bahamas/USA	1
Bosnia and herzegovina	1
Burkina Faso	1
CZECH republic	1
Caicos Islands	1
Cape Verde Island	1
Caribbean (Grand Turk)	1
Chad	1
Channel Islands	1
Chennai. Tamil Nadu	1
Corsica	1
Corsica (France)	1
Crete (Greece)	1
Cruise ship	1
Cuba/Florida (between)	1
Czech republic	1
Djibouti	1
Dominica, West Indies	1
Dominican republic	1
Dublin Ireland	1
East Atlantic Ocean	1
East China Sea	1
East Timor	1

country	n
<chr>	<int>
El Cobre	1
⋮	⋮
Argentina	69
Israel	74
Poland	74
China	75
Iran	76
Malaysia	77
Belgium	81
Norway	81
Japan	93
Sweden	95
Greece	97
Portugal	100
Turkey	107
Italy	112
France	129
Philippines	130
Unspecified	139
Netherlands	174
Spain	177
Ireland	229
New Zealand	230
South Africa	244
Germany	254
Brazil	267
Mexico	542
India	571
Australia	1060

country	n
<chr>	<int>
United Kingdom	3805
Canada	6216
USA	138705

```
[1] "Unique Shape Entries"
```

A tibble: 40 × 2

shape	n
<chr>	<int>
changing	1
egg	1
flash	1
unknown	1
diamond	2
fireball	2
oval	2
cigar	3
rectangle	4
cylinder	5
sphere	7
circle	8
triangle	18
other	19
light	55
Cube	115
Star	347
Cross	545
Cone	656
Teardrop	1291
Egg	1362
Chevron	1857
Diamond	2251
Flash	2527
Cylinder	2703
Rectangle	2829
Cigar	4031
Changing	4413
Formation	5080

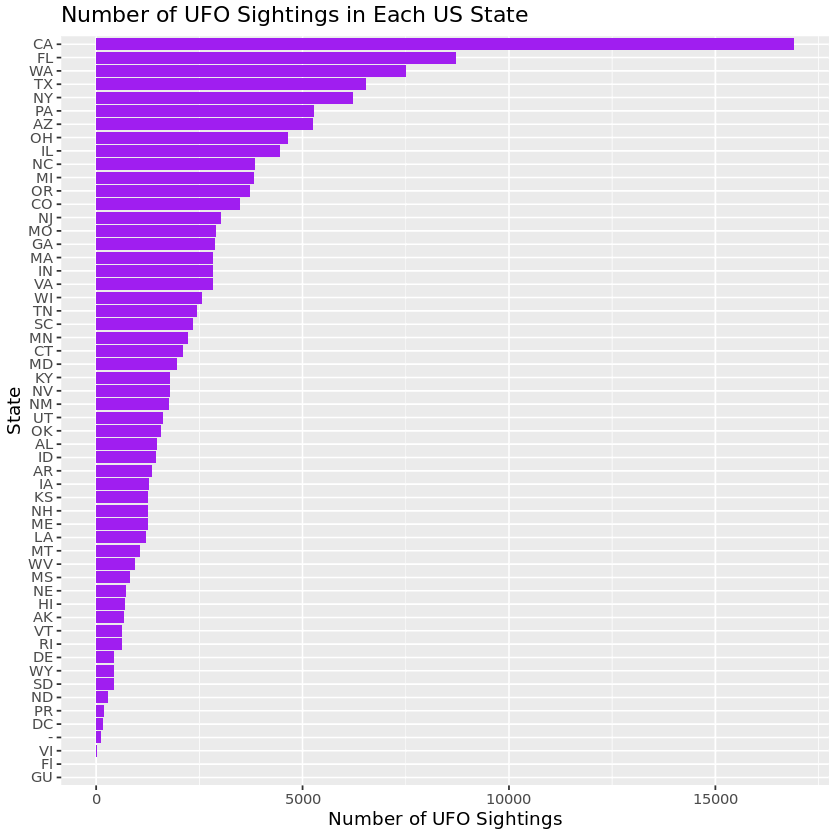
shape	n
<chr>	<int>
NA	6343
Oval	6691
Orb	7364
Sphere	8033
Disk	9216
Fireball	10069
Other	10519
Unknown	10543
Triangle	13823
Circle	15403
Light	28571

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.

In [34]:
```
df %>%
    filter(country == "USA") %>%
    count(state, name = "n") %>%
    filter(n > 1) %>%

ggplot(aes(x = reorder(state,n), y = n)) +
    geom_col(fill = "purple") +
    coord_flip() +
    labs(
        title = "Number of UFO Sightings in Each US State",
        x = "State",
        y = "Number of UFO Sightings"
    )
```


Number of UFO Sightings in Each US State
State
CA
FL
WA
TX
NY
PA
AZ
OH
IL
NC
MI
OR
CO
NJ
MO
GA
MA
IN
VA
WI
TN
SC
MN
CT
MD
KY
NV
NM
UT
OK
AL
ID
AR
IA
KS
NH
ME
LA
MT
WV
MS
NE
HI
AK
VT
RI
DE
WY
SD
ND
PR
DC
-
VI
FI
GU
0
5000
10000
15000
Number of UFO Sightings

In [ ]: