Homework 04

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 [2]:
```
library(tidyverse)

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)
)
table_a
table_b
```

A tibble: 5 × 5

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

A tibble: 5 × 5

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

In [9]:

```r
# Left Join
library(tidyverse)
left_join <- table_a %>% left_join(table_b, by = c("SKU", "Color", "Fruit"))
left_join
```

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

In [10]:

```r
# Right Join
library(tidyverse)
right_join <- table_a %>% right_join(table_b, by = c("SKU", "Color", "Fruit"))
right_join
```

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

In [11]:

```
# Inner Join
library(tidyverse)
inner_join <- table_a %>% inner_join(table_b, by = c("SKU", "Fruit", "Color"))
inner_join
```

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

In [12]:

```
# Full Join
library(tidyverse)
full_join <- table_a %>% full_join(table_b, by = c("SKU", "Color", "Fruit"))
full_join
```

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

In [13]:

```
# Semi Join
library(tidyverse)
```

```
semi_join <- table_a %>% semi_join(table_b, by = c("SKU", "Color", "Fruit"))
semi_join
```

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

In [14]:

```
# Anti Join
library(tidyverse)
anti_join <- table_a %>% anti_join(table_b, by = c("SKU", "Color", "Fruit"))
anti_join
```

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 [3]:

```
library(tidyverse)
demo <- read_csv("demographics.csv")
full <- read_csv("full.csv")
hospitals <- read_csv("hospitals.csv")
names <- read_csv("patient_names.csv")
treatment <- read_csv("treatment_info.csv")

glimpse(demo)
glimpse(full)
glimpse(hospitals)
glimpse(names)
glimpse(treatment)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   4.0.0     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
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.
```

```
Rows: 35
Columns: 5
$ patient_id <chr> "P001", "P002", "P003", "P004", "P005", "P006", "P007", "P0…
$ age        <dbl> 51, 73, 49, 6, 64, 38, 36, 22, 20, 85, 61, 23, 54, 22, 29, …
$ gender     <chr> "Male", "Male", NA, "Other", "Other", "Other", "Female", "O…
$ race       <chr> "Hispanic", "Hispanic", "White", "White", "White", "Hispani…
$ ethnicity  <chr> "Non-Hispanic", "Non-Hispanic", "Non-Hispanic", "Non-Hispan…
Rows: 35
Columns: 16
$ patient_id      <chr> "P001", "P002", "P003", "P004", "P005", "P006", "P007"…
$ name            <chr> "Mary Hicks", "Matthew Christensen", "Lisa Graham", "G…
$ age             <dbl> 51, 73, 49, 6, 64, 38, 36, 22, 20, 85, 61, 23, 54, 22,…
$ gender          <chr> "Male", "Male", NA, "Other", "Other", "Other", "Female…
$ race            <chr> "Hispanic", "Hispanic", "White", "White", "White", "Hi…
$ ethnicity       <chr> "Non-Hispanic", "Non-Hispanic", "Non-Hispanic", "Non-H…
$ condition       <chr> "Cancer", "Heart Disease", "Asthma", "Heart Disease", …
$ treatment       <chr> "Chemotherapy", "Bypass Surgery", "Inhaler Therapy", "…
$ department      <chr> "Oncology", "Cardiology", "Pediatrics", "Cardiology", …
$ hospital        <chr> "H1", "H5", "H5", "H3", "H1", "H3", "H5", "H3", "H1", …
$ admission_date  <date> 2024-09-30, 2025-06-09, 2025-09-08, 2025-09-02, 2025-…
$ release_date    <date> 2025-04-24, 2025-09-04, 2025-09-08, 2025-09-06, 2025-…
$ patient_address <chr> NA, "762 Hatfield Lights Apt. 887", "25592 Foley Forge…
$ patient_city    <chr> NA, "North Thomasbury", "New Tiffany", "Underwoodburgh…
$ patient_state   <chr> NA, "WI", "IN", "NV", "HI", "MS", "NV", "WA", "DE", "N…
$ patient_zipcode <dbl> NA, 96149, 33286, 9762, 99546, 87095, 4548, 29439, 357…
Rows: 5
Columns: 6
$ hospital_id       <chr> "H1", "H2", "H3", "H4", "H5"
$ hospital_name     <chr> "Greenwood Medical Center", "Lakeside Hospital", "Su…
$ hospital_address  <chr> "123 Maple St", "456 Elm St", "789 Oak Ave", "321 Pi…
$ hospital_city     <chr> "Springfield", "Madison", "Los Angeles", "Denver", "…
$ hospital_state    <chr> "IL", "WI", "CA", "CO", "CO"
$ hospital_zip_code <dbl> 62701, 53703, 90012, 80203, 80302
Rows: 35
Columns: 4
$ patient_id   <chr> "P001", "P002", "P003", "P004", "P005", "P006", "P007", "…
$ name         <chr> "Mary Hicks", "Matthew Christensen", "Lisa Graham", "Greg…
$ hospital_id  <chr> "H1", "H5", "H5", "H3", "H1", "H3", "H5", "H3", "H1", "H5…
$ condition_id <chr> "C", "HD", "A", "HD", "HD", "A", "A", "S", "A", "F", "A",…
Rows: 5
Columns: 4
$ condition_id <chr> "HD", "S", "C", "F", "A"
$ condition    <chr> "Heart Disease", "Stroke", "Cancer", "Fracture", "Asthma"
$ treatment    <chr> "Bypass Surgery", "Rehabilitation Therapy", "Chemotherapy…
$ department   <chr> "Cardiology", "Neurology", "Oncology", "Orthopedics", "Pe…
```

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 [25]:

```
full_longer <- pivot_longer(full, age:patient_zipcode, names_to = "patient_property
    values_to = "record",
    values_transform = function(x) ifelse(is.na(x), NA, as.character(x)))
full_longer
full_longer %>%
    group_by(name) %>%
    tally() %>%
    arrange(desc(n))
```

A tibble: 490 × 4

patient_id	name	patient_property	record
<chr>	<chr>	<chr>	<chr>
P001	Mary Hicks	age	51
P001	Mary Hicks	gender	Male
P001	Mary Hicks	race	Hispanic
P001	Mary Hicks	ethnicity	Non-Hispanic
P001	Mary Hicks	condition	Cancer
P001	Mary Hicks	treatment	Chemotherapy
P001	Mary Hicks	department	Oncology
P001	Mary Hicks	hospital	H1
P001	Mary Hicks	admission_date	2024-09-30
P001	Mary Hicks	release_date	2025-04-24
P001	Mary Hicks	patient_address	NA
P001	Mary Hicks	patient_city	NA
P001	Mary Hicks	patient_state	NA
P001	Mary Hicks	patient_zipcode	NA
P002	Matthew Christensen	age	73
P002	Matthew Christensen	gender	Male
P002	Matthew Christensen	race	Hispanic
P002	Matthew Christensen	ethnicity	Non-Hispanic
P002	Matthew Christensen	condition	Heart Disease
P002	Matthew Christensen	treatment	Bypass Surgery
P002	Matthew Christensen	department	Cardiology
P002	Matthew Christensen	hospital	H5
P002	Matthew Christensen	admission_date	2025-06-09
P002	Matthew Christensen	release_date	2025-09-04
P002	Matthew Christensen	patient_address	762 Hatfield Lights Apt. 887
P002	Matthew Christensen	patient_city	North Thomasbury
P002	Matthew Christensen	patient_state	WI
P002	Matthew Christensen	patient_zipcode	96149
P003	Lisa Graham	age	49

patient_id	name	patient_property	record
<chr>	<chr>	<chr>	<chr>
P003	Lisa Graham	gender	NA
⋮	⋮	⋮	⋮
P033	Spencer Wells	patient_state	CA
P033	Spencer Wells	patient_zipcode	7129
P034	Holly Mclaughlin	age	56
P034	Holly Mclaughlin	gender	Other
P034	Holly Mclaughlin	race	Asian
P034	Holly Mclaughlin	ethnicity	Non-Hispanic
P034	Holly Mclaughlin	condition	Heart Disease
P034	Holly Mclaughlin	treatment	Bypass Surgery
P034	Holly Mclaughlin	department	Cardiology
P034	Holly Mclaughlin	hospital	H3
P034	Holly Mclaughlin	admission_date	2025-04-18
P034	Holly Mclaughlin	release_date	2025-07-01
P034	Holly Mclaughlin	patient_address	06756 Mcclure Forks Apt. 108
P034	Holly Mclaughlin	patient_city	Williamshaven
P034	Holly Mclaughlin	patient_state	CO
P034	Holly Mclaughlin	patient_zipcode	65766
P035	Ashley Johnson	age	22
P035	Ashley Johnson	gender	Other
P035	Ashley Johnson	race	White
P035	Ashley Johnson	ethnicity	Non-Hispanic
P035	Ashley Johnson	condition	Heart Disease
P035	Ashley Johnson	treatment	Bypass Surgery
P035	Ashley Johnson	department	Cardiology
P035	Ashley Johnson	hospital	H1
P035	Ashley Johnson	admission_date	2025-02-20
P035	Ashley Johnson	release_date	2025-05-10
P035	Ashley Johnson	patient_address	79347 Freeman Mount

patient_id	name	patient_property	record
<chr>	<chr>	<chr>	<chr>
P035	Ashley Johnson	patient_city	Port Austin
P035	Ashley Johnson	patient_state	KY
P035	Ashley Johnson	patient_zipcode	73451

A tibble: 35 × 2

name	n
<chr>	<int>
Anthony Anderson	14
April Sanchez	14
Ashley Johnson	14
Casey Norman	14
Dylan Lopez DVM	14
Erica Foley	14
Greg Brown	14
Heather Chandler	14
Holly Contreras	14
Holly Mclaughlin	14
Jessica Ibarra	14
John Brown	14
John Ibarra	14
John Rodriguez	14
Jose Young	14
Joseph Thompson	14
Joshua Baker	14
Kathryn Harrison	14
Kristine Lewis	14
Lisa Graham	14
Maria Bruce	14
Mary Cobb	14
Mary Hicks	14
Matthew Christensen	14
Matthew Jones	14
Matthew Rogers	14
Melinda Moody	14
Nathan Chase	14
Nicholas Smith MD	14

name	n
<chr>	<int>
Samuel Herrera	14
Spencer Wells	14
Thomas Logan	14
Wanda Simmons	14
Wendy Richardson	14
Whitney Fuller	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 [69]:
```
value_num <- pivot_longer(full %>% select(-c(gender:patient_state)), c(age, patient
    values_to = "num_obs",
    values_transform = function(x) ifelse(is.na(x), NA, as.double(x))) %>%
group_by(patient_id, name) %>%
  mutate(row = row_number()) %>%
  ungroup()

value_chr <- pivot_longer(full %>% select(-admission_date, -release_date, -age, -pa
                          c(gender:hospital, patient_address:patient_state), names_
    values_to = "chr_obs",
    values_transform = function(x) ifelse(is.na(x), NA, as.character(x))) %>%
group_by(patient_id, name) %>%
  mutate(row = row_number()) %>%
  ungroup()

value_date <- pivot_longer(full %>% select(-c(age:hospital), -c(patient_address:pat
                          c(admission_date, release_date), names_to = "patient_prop
    values_to = "date_obs") %>%
group_by(patient_id, name) %>%
  mutate(row = row_number()) %>%
  ungroup()

joined <- full_join(value_num, value_date, by = c('patient_id', 'name', 'row'))
joined <- full_join(joined, value_chr, by = c('patient_id', 'name', 'row'))
joined %>%
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: 1 × 3

n_chr	n_num	n_date
<int>	<int>	<int>
1741	347	350

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 [71]:

```
joined_names <- names %>% left_join(hospitals, by = "hospital_id")
joined_names
```

A spec_tbl_df: 35 × 9

patient_id	name	hospital_id	condition_id	hospital_name	hospital_address	hospital_cit
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr
P001	Mary Hicks	H1	C	Greenwood Medical Center	123 Maple St	Springfiel
P002	Matthew Christensen	H5	HD	Mountainview Clinic	654 Birch Blvd	Boulde
P003	Lisa Graham	H5	A	Mountainview Clinic	654 Birch Blvd	Boulde
P004	Greg Brown	H3	HD	Sunrise Health	789 Oak Ave	Los Angele
P005	Joshua Baker	H1	HD	Greenwood Medical Center	123 Maple St	Springfiel
P006	Wendy Richardson	H3	A	Sunrise Health	789 Oak Ave	Los Angele
P007	April Sanchez	H5	A	Mountainview Clinic	654 Birch Blvd	Boulde
P008	Melinda Moody	H3	S	Sunrise Health	789 Oak Ave	Los Angele
P009	Dylan Lopez DVM	H1	A	Greenwood Medical Center	123 Maple St	Springfiel
P010	Maria Bruce	H5	F	Mountainview Clinic	654 Birch Blvd	Boulde
P011	Kristine Lewis	H4	A	Valley General Hospital	321 Pine Rd	Denve
P012	Jessica Ibarra	H2	F	Lakeside Hospital	456 Elm St	Madiso
P013	Matthew Rogers	H4	F	Valley General Hospital	321 Pine Rd	Denve
P014	Joseph Thompson	H3	F	Sunrise Health	789 Oak Ave	Los Angele
P015	Holly Contreras	H1	HD	Greenwood Medical Center	123 Maple St	Springfiel
P016	Heather Chandler	H1	A	Greenwood Medical Center	123 Maple St	Springfiel
P017	John Brown	H1	A	Greenwood Medical Center	123 Maple St	Springfiel
P018	Nathan Chase	H2	HD	Lakeside Hospital	456 Elm St	Madiso

patient_id	name	hospital_id	condition_id	hospital_name	hospital_address	hospital_cit
<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr
P019	Casey Norman	H1	A	Greenwood Medical Center	123 Maple St	Springfiel
P020	Nicholas Smith MD	H1	C	Greenwood Medical Center	123 Maple St	Springfiel
P021	Mary Cobb	H5	S	Mountainview Clinic	654 Birch Blvd	Boulde
P022	Thomas Logan	H4	C	Valley General Hospital	321 Pine Rd	Denve
P023	Anthony Anderson	H4	F	Valley General Hospital	321 Pine Rd	Denve
P024	Matthew Jones	H3	A	Sunrise Health	789 Oak Ave	Los Angele
P025	Kathryn Harrison	H5	F	Mountainview Clinic	654 Birch Blvd	Boulde
P026	Jose Young	H5	C	Mountainview Clinic	654 Birch Blvd	Boulde
P027	Samuel Herrera	H2	C	Lakeside Hospital	456 Elm St	Madiso
P028	Wanda Simmons	H5	F	Mountainview Clinic	654 Birch Blvd	Boulde
P029	Whitney Fuller	H3	C	Sunrise Health	789 Oak Ave	Los Angele
P030	John Rodriguez	H4	C	Valley General Hospital	321 Pine Rd	Denve
P031	John Ibarra	H1	C	Greenwood Medical Center	123 Maple St	Springfiel
P032	Erica Foley	H1	C	Greenwood Medical Center	123 Maple St	Springfiel
P033	Spencer Wells	H5	S	Mountainview Clinic	654 Birch Blvd	Boulde
P034	Holly Mclaughlin	H3	HD	Sunrise Health	789 Oak Ave	Los Angele
P035	Ashley Johnson	H1	HD	Greenwood Medical Center	123 Maple St	Springfiel

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 [75]:
```
init_join <- names %>% left_join(treatment, by = "condition_id")
join_2 <- init_join %>% left_join(demo, by = "patient_id")
join_2 <- join_2 %>% select(-hospital_id, -condition_id, -department, -ethnicity, -
join_2
```

A tibble: 35 × 6

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

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

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 [11]:

```
# Load UFO sightings data from a GitHub CSV
library(tidyverse)
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")
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   4.0.0     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
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 · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA · NA ·
NA · NA · NA · NA · NA · NA · NA · NA
```

A tibble: 10 × 5

d1 <chr>	d2 <chr>	d3 <chr>	d4 <chr>	d5 <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

A spec_tbl_df: 406 × 2

country	n
<chr>	<int>
USA	138705
Canada	6216
United Kingdom	3805
Australia	1060
India	571
Mexico	542
Brazil	267
Germany	254
South Africa	244
New Zealand	230
Ireland	229
Spain	177
Netherlands	174
Unspecified	139
Philippines	130
France	129
Italy	112
Turkey	107
Portugal	100
Greece	97
Sweden	95
Japan	93
Belgium	81
Norway	81
Malaysia	77
Iran	76
China	75
Israel	74
Poland	74

country	n
<chr>	<int>
Argentina	69
⋮	⋮
Unioted Kingdom	1
United Arad Emirates	1
United Kingdon	1
United kingdom	1
Unknown	1
Unuted Kingdom	1
Vanuatu	1
Vatican City	1
Western Australia	1
Yemen	1
Yup	1
finland	1
france	1
great britain	1
hatton city, Sri Lanka	1
india	1
italy	1
lat 2 deg 48 min N 124 deg W	1
mediterranean sea	1
mexico	1
mid-Atlantic Ocean	1
non applicable	1
over New Brunswick	1
saipan	1
slovakia	1
south africa	1
sri lanka	1

country	n
<chr>	<int>
turkey	1
united kingdom	1
unknown/at sea	1

A spec_tbl_df: 976 × 2

state	n
<chr>	<int>
CA	16913
NA	9105
FL	8717
WA	7510
TX	6548
NY	6224
PA	5292
AZ	5267
OH	4650
IL	4446
NC	3852
MI	3834
OR	3732
CO	3489
NJ	3036
MO	2908
GA	2889
MA	2841
IN	2839
VA	2838
ON	2660
WI	2566
TN	2439
SC	2347
MN	2229
CT	2111
MD	1954
KY	1793
NV	1785

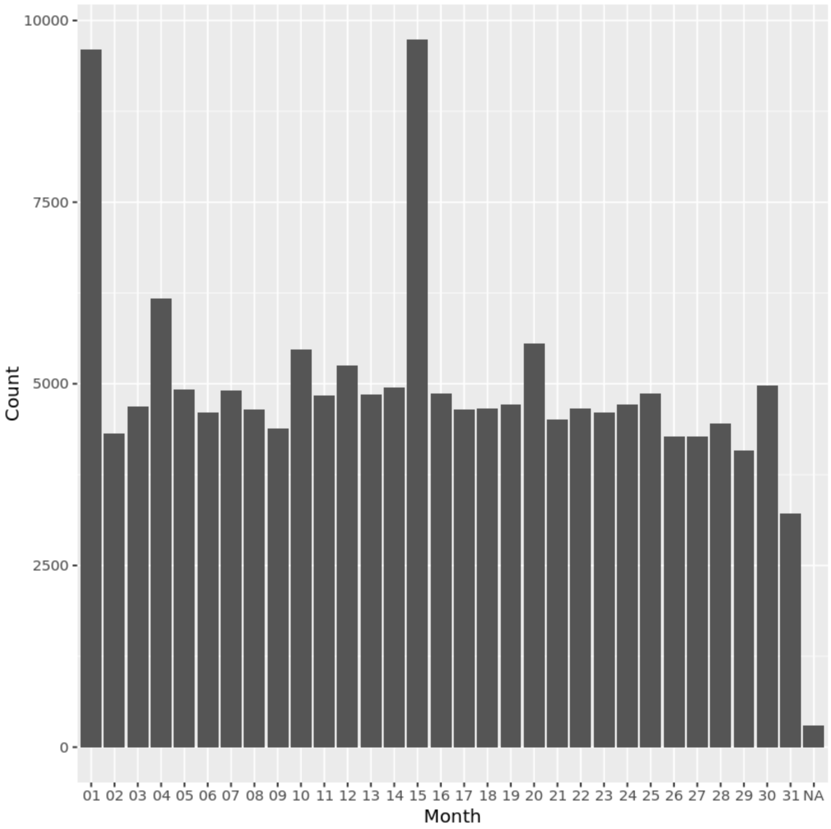
state	n
<chr>	<int>
NM	1758
⋮	⋮
West Virginia	1
Western	1
Western Division	1
Westmoreland Parish	1
Wicklow	1
Windsor and Maidenhead	1
Wisconsin	1
Województwo lubelskie	1
Województwo małopolskie	1
Województwo pomorskie	1
Województwo warmińsko-mazurskie	1
Województwo wielkopolskie	1
Województwo łódzkie	1
Województwo śląskie	1
Wokingham	1
Yangon Region	1
Zacapa Department	1
Zagreb County	1
Zagrebačka županija	1
Zaječar District	1
Zaporiz'ka oblast	1
Zaporizhia Oblast	1
Zulia	1
Évora District	1
Örebro County	1
Łódzkie	1
Łódź Voivodeship	1

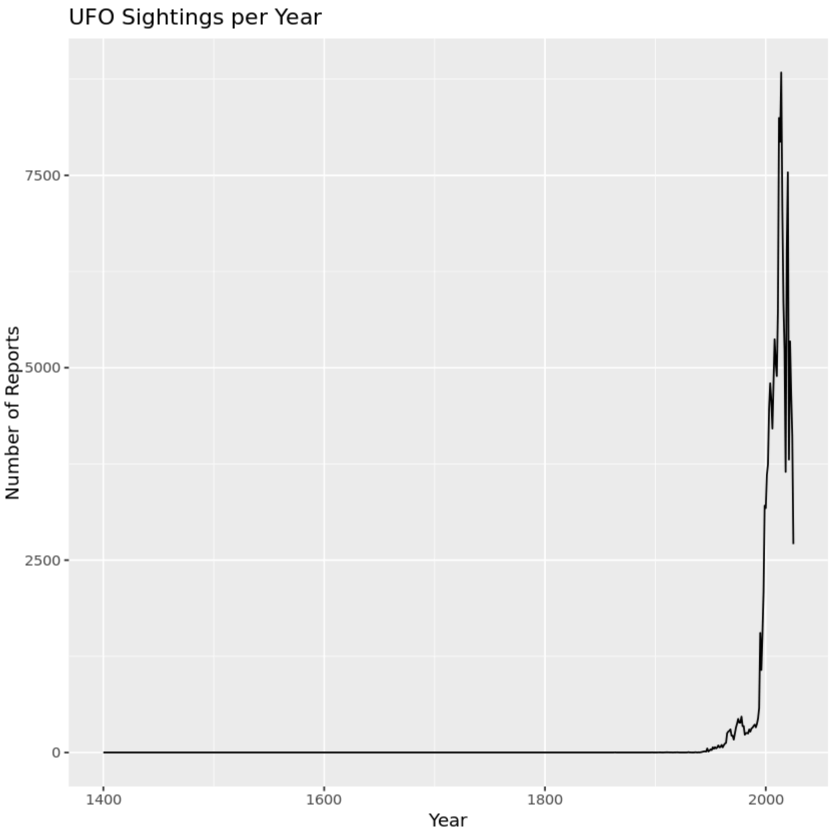
state	n
<chr>	<int>
Šiauliai District Municipality	1
Šiaulių apskritis	1
Żebbuġ Malta	1

A spec_tbl_df: 40 × 2

shape	n
<chr>	<int>
Light	28571
Circle	15403
Triangle	13823
Unknown	10543
Other	10519
Fireball	10069
Disk	9216
Sphere	8033
Orb	7364
Oval	6691
NA	6343
Formation	5080
Changing	4413
Cigar	4031
Rectangle	2829
Cylinder	2703
Flash	2527
Diamond	2251
Chevron	1857
Egg	1362
Teardrop	1291
Cone	656
Cross	545
Star	347
Cube	115
light	55
other	19
triangle	18
circle	8

shape	n
<chr>	<int>
sphere	7
cylinder	5
rectangle	4
cigar	3
diamond	2
fireball	2
oval	2
changing	1
egg	1
flash	1
unknown	1





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 [20]:

```r
df$state[df$state == '0'] <- NA
df %>% group_by(state) %>% tally() %>% arrange(n)

df$country[df$country == 'Above the pacific ocean'] <- 'Pacific Ocean'
df$country[df$country == 'Bahamas The'] <- 'Bahamas/USA'
df$country[df$country == 'Unspecified'] <- NA

df %>% group_by(country) %>% tally() %>% arrange(n)

df <- df %>%
  mutate(shape = str_to_lower(shape))
df$shape[df$shape == 'NA'] <- NA
df %>% group_by(shape) %>% tally() %>% arrange(n)

df %>% group_by(shape) %>% tally() %>% arrange(n)
df %>% group_by(explanation) %>% tally() %>% arrange(n)
```

A tibble: 973 × 2

state <chr>	n <int>
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
Auvergne-Rhône-Alpes	1

state	n
<chr>	<int>
Azores	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	9231
CA	16913

A tibble: 404 × 2

country	n
<chr>	<int>
Aegean Sea	1
Andaman Islands	1
Angola	1
Anguilla	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
El Cobre	1
Euleuthera	1
Far East	1

country	n
<chr>	<int>
Faroe Islands	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
NA	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

A tibble: 25 × 2

shape	n
<chr>	<int>
cube	115
star	347
cross	545
cone	656
teardrop	1291
egg	1363
chevron	1857
diamond	2253
flash	2528
cylinder	2708
rectangle	2833
cigar	4034
changing	4414
formation	5080
NA	6343
oval	6693
orb	7364
sphere	8040
disk	9216
fireball	10071
other	10538
unknown	10544
triangle	13841
circle	15411
light	28626

A tibble: 25 × 2

shape	n
<chr>	<int>
cube	115
star	347
cross	545
cone	656
teardrop	1291
egg	1363
chevron	1857
diamond	2253
flash	2528
cylinder	2708
rectangle	2833
cigar	4034
changing	4414
formation	5080
NA	6343
oval	6693
orb	7364
sphere	8040
disk	9216
fireball	10071
other	10538
unknown	10544
triangle	13841
circle	15411
light	28626

A tibble: 90 × 2

explanation	n
<chr>	<int>
Animal?	1
Aurora Borealis?	1
Aurora?	1
Ball Lightning?	1
Bat?	1
Boat?	1
Boats	1
Boats?	1
Comet	1
Debris?	1
Dream?	1
Fireworks	1
Flare?	1
Green fishing lights	1
Headlights?	1
Helicopter?	1
Insect web?	1
Insects?	1
Lightning?	1
Moon?	1
Smoke	1
Smoke ring	1
Space Junk?	1
Spiderweb	1
Starlink-Racetrack	1
Sundog?	1
Truck	1
birds?	1
shock cone???	1

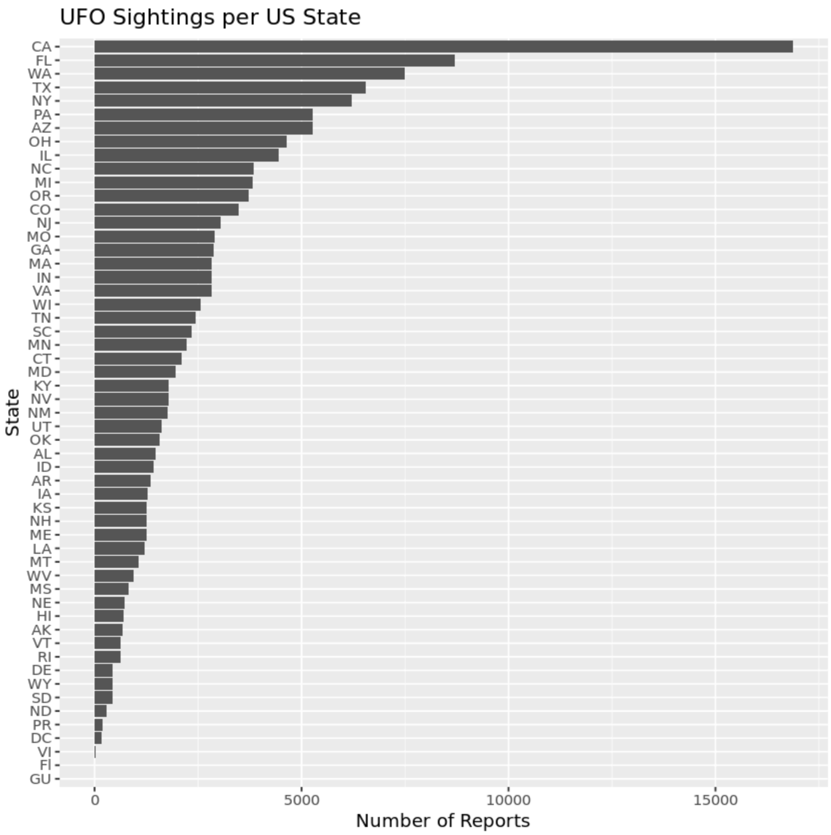
explanation	n
<chr>	<int>
Kite	2
⋮	⋮
Searchlight?	19
Insect	22
Contrail?	24
Insect?	25
Bird	27
Birds?	29
Camera Anomaly?	31
Contrail	38
Satellite	38
Drone	40
Meteor	40
Bird?	41
Rocket?	43
Satellite?	46
Meteor?	63
Searchlight	65
Camera Anomaly	78
Starlink?	82
Planet/Star?	84
Chinese Lantern	85
Chinese Lantern?	100
Balloon	130
Aircraft	148
Planet/Star	153
Aircraft?	181
Balloon?	218
Starlink	240

explanation	n
<chr>	<int>
Rocket	416
Drone?	424
NA	153546

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 [49]: state_counts <- table(df$state[df$country == "USA"])
state_filter <- names(state_counts[state_counts > 1])
df_filtered <- df %>% filter(country == "USA") %>% filter(state %in% state_filter)
state_factor <- df_filtered %>% group_by(state) %>% tally() %>% arrange(n)
df_filtered %>%
  filter(!is.na(occurred)) %>%
  ggplot(aes(y = factor(state, state_factor$state))) +
  geom_bar(stat = 'count') +
    labs(title = "UFO Sightings per US State", x = "Number of Reports", y = "State"
```


UFO Sightings per 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
Number of Reports