R Notebook - COVID Cases in Northeast Colorado

Maya Reese Farmer

September 28, 2021

Introduction

I will be demonstrating some introductory modeling and data visualization techniques using the COVID-19 data for the northeast district in Colorado. This includes Logan, Morgan, Washington, Yuma, Phillips, and Sedgwick county.

The 'COVID19CaseData' dataset contains over 10,000 COVID case observations and has 19 columns. The most notable variables include:

- Case Status

- Date Reported
- County
- Gender
- Age
- Race and Ethnicity

- Outcome (alive or deceased)
- Hospitalized (yes or no)
- Spec1 Test Result (positive or negative)
- Reinfection (yes or no)

```r
## importing dataset into r
Covid19CaseData <- read.csv("~/Downloads/Covid19CaseData.csv")
attach(Covid19CaseData)

## view first 6 rows of data
head(Covid19CaseData)
```

```
##   Disease.Name Case.Status Reported.Date County Gender Age.yrs     Race.1
## 1     COVID-19   Confirmed     3/18/2020   Yuma Female      63      White
## 2     COVID-19   Confirmed     3/18/2020 Morgan   Male      65      White
## 3     COVID-19    Probable     3/19/2020 Morgan Female      63      White
## 4     COVID-19   Confirmed     3/22/2020  Logan   Male      84      White
## 5     COVID-19   Confirmed     3/22/2020 Morgan Female      71 Other Race
## 6     COVID-19   Confirmed     3/22/2020  Logan   Male      63      White
##                Ethnicity     City    State Zip.Code Onset.Date
## 1 Not Hispanic or Latino     YUMA Colorado    80759  3/11/2020
## 2 Not Hispanic or Latino  Weldona Colorado    80653   3/3/2020
## 3 Not Hispanic or Latino  Weldona Colorado    80653  3/14/2020
## 4 Not Hispanic or Latino Sterling Colorado    80751  3/15/2020
## 5     Hispanic or Latino    Brush Colorado    80723  3/15/2020
## 6 Not Hispanic or Latino Sterling Colorado    80751  3/12/2020
##   Onset.Date.Unavailable                Outcome Hospitalized Spec1.Test1.Name
## 1                     No                  Alive           No           RT-PCR
## 2                     No                  Alive           No           RT-PCR
## 3                     No                  Alive           No
## 4                     No Patient died (finding)          Yes           RT-PCR
## 5                    Yes                  Alive          Yes           RT-PCR
## 6                     No                  Alive           No           RT-PCR
##   Spec1.Test1.Result Spec1.Test1.Result.Date Re.Infection
## 1           Positive               3/17/2020           No
## 2           Positive               3/18/2020           No
## 3                                                      No
## 4           Positive               3/22/2020           No
## 5           Positive               3/22/2020           No
## 6           Positive               3/22/2020           No
```

Queries and Data Visualization

I will be querying the data to answer a series of questions using R programming language. I'll present my findings using either tables or ggplot2 for data visualization.

Total COVID Cases The first thing I want to do is determine the total number of cases in the northeast district. This should include only cases that have a 'confirmed' or 'probable' case status.

```
library(plyr)
## count the number of individuals in each Case Status category
count(Case.Status)
```

```
##           x freq
## 1              6
## 2 Confirmed 8741
## 3  Probable  884
## 4   Suspect  856
## 5   Unknown    1
```

The output shows that there are 8741 Confirmed cases and 884 Probable cases. This means there have been a total of 9625 cases in the northeast district to date.

Next, I want to extract these confirmed and probable cases into a new data set for additional querying.

```
COVIDData.confirmed <- subset(Covid19CaseData, Case.Status ==
    "Confirmed" | Case.Status == "Probable")
attach(COVIDData.confirmed)
```

```
## The following objects are masked from Covid19CaseData:
##
##     Age.yrs, Case.Status, City, County, Disease.Name, Ethnicity,
##     Gender, Hospitalized, Onset.Date, Onset.Date.Unavailable, Outcome,
##     Race.1, Re.Infection, Reported.Date, Spec1.Test1.Name,
##     Spec1.Test1.Result, Spec1.Test1.Result.Date, State, Zip.Code
```

Now the COVIDDate.confirmed data table only includes those 9625 COVID cases.

Percent COVID Cases by Age Group Next I want to look at the percentage of cases by age group. The age groups include <18 yo, 18-35 yo, 36-55 yo, 56-75 yo, >75 yo. By doing this, we can observe whether individuals in different age groups have been influenced differently by COVID-19.

```
## count the number of cases if Age < 18yo
count(Age.yrs < 18)
```

```
##       x freq
## 1 FALSE 9006
## 2  TRUE  618
## 3    NA    1
```

```
## count # of cases when Age is 18-35
count(Age.yrs >= 18 & Age.yrs <= 35)
```

```
##       x freq
## 1 FALSE 6601
## 2  TRUE 3023
## 3    NA    1
```

```
## count # of cases when Age is 36-55
count(Age.yrs >= 36 & Age.yrs <= 55)
```

```
##       x freq
## 1 FALSE 6624
## 2  TRUE 3000
## 3    NA    1
```

```
## count # of cases when Age is 56-75
count(Age.yrs >= 56 & Age.yrs <= 75)
```

```
##       x freq
## 1 FALSE 7417
## 2  TRUE 2207
## 3    NA    1
```

```
## count # of cases when Age is >75
count(Age.yrs > 75)
```

```
##       x freq
## 1 FALSE 8848
## 2  TRUE  776
## 3    NA    1
```

We can now create a new data table using these outputs to show how cases are distributed across age groups.

```
age.group = c("< 18 yrs", "18-35 yrs", "36-55 yrs", "56-75 yrs",
    "> 75 yrs", "Not Specified")
number.cases = c(618, 3023, 3000, 2207, 776, 1)
percent.agedata = data.frame(age.group, number.cases)
percent.agedata
```

```
##       age.group number.cases
## 1      < 18 yrs          618
## 2     18-35 yrs         3023
## 3     36-55 yrs         3000
## 4     56-75 yrs         2207
## 5      > 75 yrs          776
## 6 Not Specified            1
```

```r
## calculate percent cases by age
percent.age = (number.cases/sum(number.cases))*100
percent.age
```

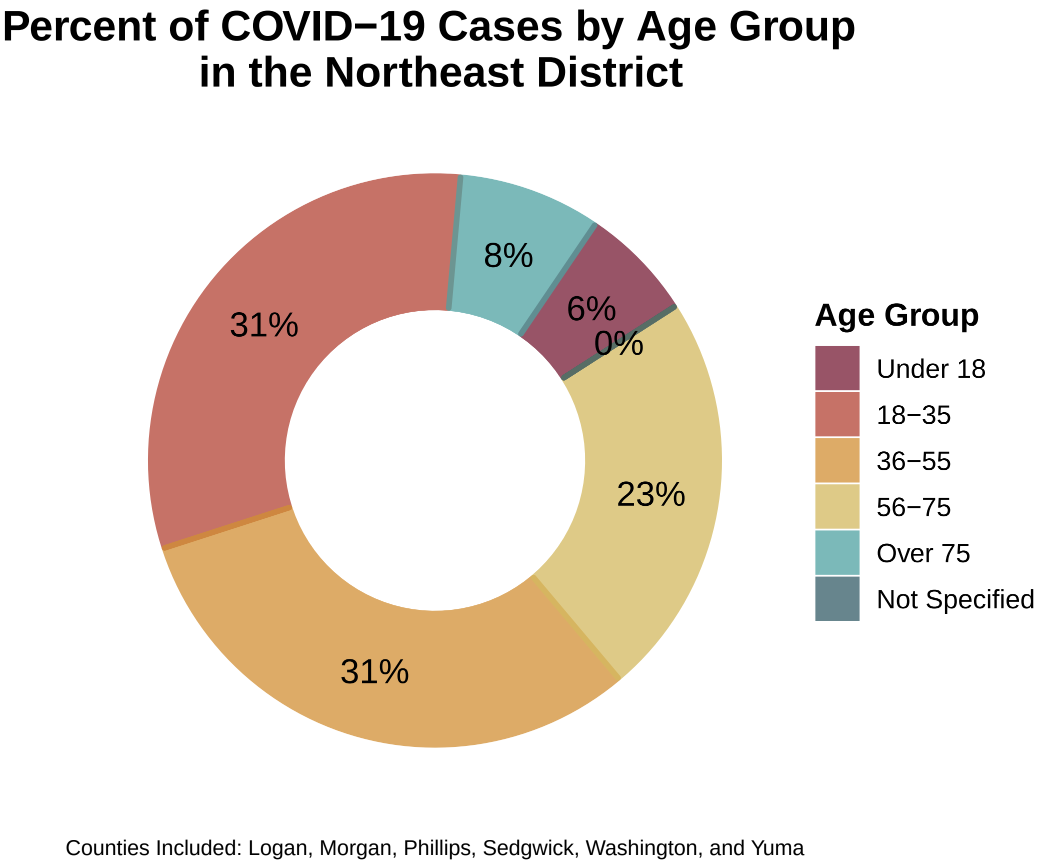
```
## [1]  6.42077922 31.40779221 31.16883117 22.92987013  8.06233766  0.01038961
```

```r
## attach data string to percent.agedata dataframe
percent.agedata$percent.age = percent.age
attach(percent.agedata)
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##     age.group, number.cases, percent.age
```

Now that the percent cases by age data table has been created, we can visualize the data using a donut chart.

```r
library(ggplot2)
library(cowplot)

ggplot(percent.agedata, aes(x = 2, y = percent.age, fill = age.group,
    color = age.group)) + geom_col(size = 1) + coord_polar(theta = "y",
    start = 1) + xlim(c(0.5, 2.5)) + theme_void() + geom_text(aes(label = paste0(round(percent.age),
    "%")), color = "black", position = position_stack(vjust = 0.5),
    check_overlap = T, size = 4.5, show.legend = FALSE) + scale_fill_manual(values = alpha(c("#751a33",
    "#b34233", "#d28f33", "#d4b95e", "#4ea2a2", "#335c67"), 0.75),
    name = "Age Group", breaks = c("< 18 yrs", "18-35 yrs", "36-55 yrs",
        "56-75 yrs", "> 75 yrs", "Not Specified"), labels = c("Under 18",
        "18-35", "36-55", "56-75", "Over 75", "Not Specified")) +
    scale_color_manual(values = alpha(c("#751a33", "#b34233",
        "#d28f33", "#d4b95e", "#4ea2a2", "#335c67"), 0.75), name = "Age Group",
        breaks = c("< 18 yrs", "18-35 yrs", "36-55 yrs", "56-75 yrs",
            "> 75 yrs", "Not Specified"), labels = c("Under 18",
            "18-35", "36-55", "56-75", "Over 75", "Not Specified")) +
    ggtitle("Percent of COVID-19 Cases by Age Group \n in the Northeast District") +
    theme(plot.title = element_text(size = 16, face = "bold",
        hjust = 0.5), plot.caption = element_text(size = 8, hjust = 0.5),
        legend.title = element_text(size = 12, face = "bold"),
        legend.text = element_text(size = 10)) + labs(caption = "Counties Included: Logan, Morgan, Phil
```


Percent of COVID−19 Cases by Age Group in the Northeast District

Counties Included: Logan, Morgan, Phillips, Sedgwick, Washington, and Yuma

---

Percent COVID Cases by Race Next, we can look at a similar breakdown of percent COVID cases based on Race.

```
count(Race.1)
```

```
##                                              x freq
## 1                                           3483
## 2          American Indian or Alaska Native   26
## 3                                     Asian   22
## 4                 Black or African American  324
## 5 Native Hawaiian or Other Pacific Islander   16
## 6                                Other Race  532
## 7                                   Refused   54
## 8                                   Unknown  436
## 9                                     White 4732
```

We can see from the output that Race is missing for 3483 individuals. Also, there are 2 redundant race categories: Unknown abd Refused. To make the calculations and visualization more concise, we should change the blank, Refused, and Unknown characters to a single "Not Specified" character.

```
COVIDData.confirmed$Race.1 <- sub("^$", "Not Specified", COVIDData.confirmed$Race.1)
COVIDData.confirmed$Race.1 <- sub("Unknown", "Not Specified", COVIDData.confirmed$Race.1)
```

```r
COVIDData.confirmed$Race.1 <- sub("Refused", "Not Specified", COVIDData.confirmed$Race.1)

## recount cases based on Race
count(COVIDData.confirmed$Race.1)
```

```
##                                           x freq
## 1          American Indian or Alaska Native   26
## 2                                     Asian   22
## 3                 Black or African American  324
## 4 Native Hawaiian or Other Pacific Islander   16
## 5                             Not Specified 3973
## 6                                Other Race  532
## 7                                     White 4732
```

Now that the data are cleaned up, we can create a data frame from this output which we'll use for the next visualization.

```r
race = c("American Indian or Alaska Native", "Asian", "Black or African American",
    "Native Hawaiian or Other Pacific Islander", "Not Specified",
    "Other Race", "White")
cases = c(26, 22, 324, 16, 3973, 532, 4732)
percent.racedata = data.frame(race, cases)
percent.racedata
```

```
##                                        race cases
## 1          American Indian or Alaska Native    26
## 2                                     Asian    22
## 3                 Black or African American   324
## 4 Native Hawaiian or Other Pacific Islander    16
## 5                             Not Specified  3973
## 6                                Other Race   532
## 7                                     White  4732
```

```r
## calculate percent cases by race
percent.race = (cases/sum(cases))
percent.race
```

```
## [1] 0.002701299 0.002285714 0.033662338 0.001662338 0.412779221 0.055272727
## [7] 0.491636364
```

```r
## attach data string to percent.agedata dataframe
percent.racedata$percent.race = percent.race
attach(percent.racedata)
```

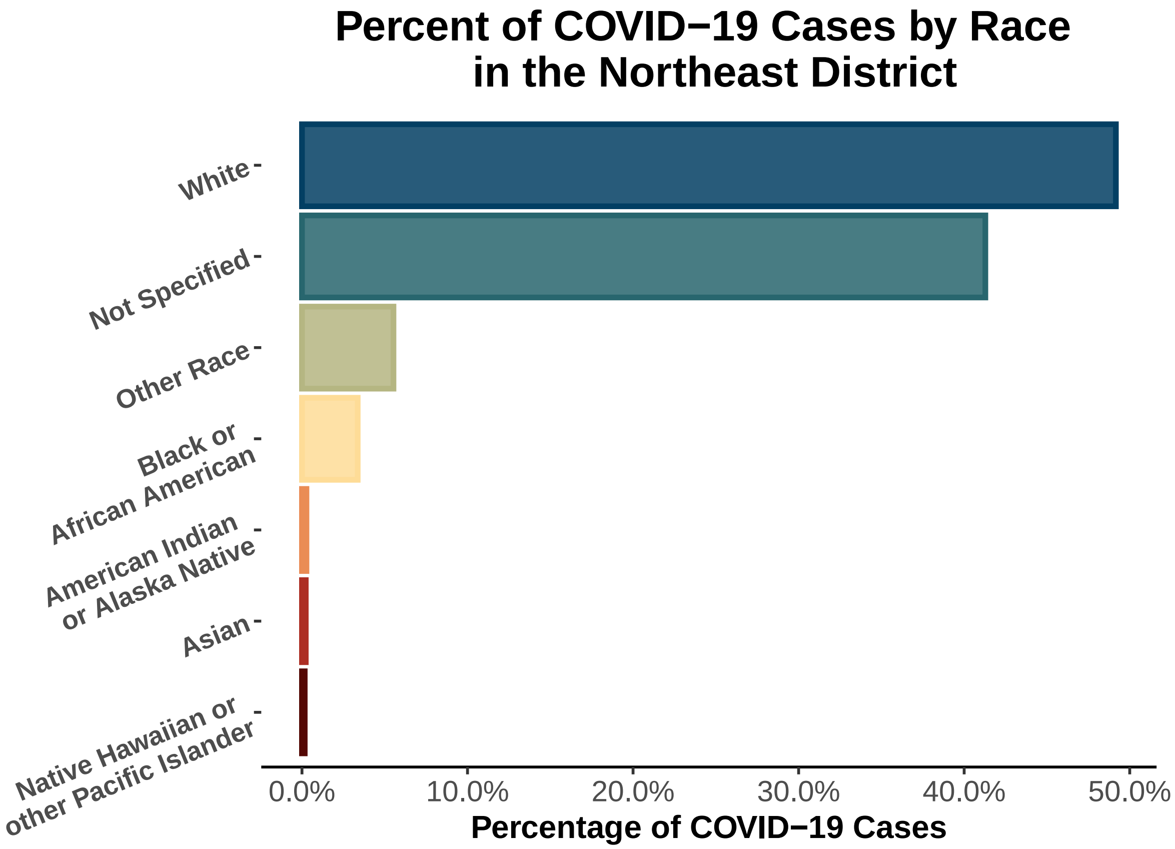
```
## The following objects are masked _by_ .GlobalEnv:
##
##     cases, percent.race, race
```

```r
percent.racedata
```

```
##                                         race cases percent.race
## 1          American Indian or Alaska Native    26  0.002701299
## 2                                     Asian    22  0.002285714
## 3                 Black or African American   324  0.033662338
## 4 Native Hawaiian or Other Pacific Islander    16  0.001662338
## 5                             Not Specified  3973  0.412779221
## 6                                Other Race   532  0.055272727
## 7                                     White  4732  0.491636364
```

Now that the percent cases by Race data table has been created, we can visualize the data using a bar chart.

```r
library(scales)

ggplot(percent.racedata, aes(x = reorder(race, percent.race),
    y = percent.race, fill = reorder(race, percent.race), color = reorder(race,
        percent.race))) + geom_bar(stat = "identity", size = 1) +
    coord_flip() + labs(x = "", y = "Percentage of COVID-19 Cases") +
    scale_fill_manual(values = alpha(c("#033f63", "#28666e",
        "#b5b682", "#fedc97", "#ea8c55", "#ad2e24", "#540804"),
        0.85), breaks = c("White", "Not Specified", "Other Race",
        "Black or African American", "American Indian or Alaska Native",
        "Asian", "Native Hawaiian or Other Pacific Islander")) +
    scale_color_manual(values = c("#033f63", "#28666e", "#b5b682",
        "#fedc97", "#ea8c55", "#ad2e24", "#540804"), breaks = c("White",
        "Not Specified", "Other Race", "Black or African American",
        "American Indian or Alaska Native", "Asian", "Native Hawaiian or Other Pacific Islander")) +
    scale_y_continuous(labels = percent) + ggtitle("Percent of COVID-19 Cases by Race \n in the Northeas
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), plot.title = element_text(size = 16,
            face = "bold", hjust = 0.5), legend.position = "none",
        axis.text.y = element_text(face = "bold", size = 10,
            angle = 22.5), axis.title.x = element_text(face = "bold",
            size = 12), axis.line.x = element_line(color = "black"),
        axis.text.x = element_text(size = 11)) + scale_x_discrete(labels = c("Native Hawaiian or \n othe
    "Asian", "American Indian \n or Alaska Native", "Black or \n African American",
    "Other Race", "Not Specified", "White"))
```



Incidence Rate per 100K by County Finally, I want to see how the COVID incidence rate per 100K differs by county.

```
## count the number of confirmed cases for each county
count(COVIDData.confirmed$County)
```

```
##            x freq
## 1      Logan 4311
## 2     Morgan 3046
## 3   Phillips  451
## 4   Sedgwick  269
## 5 Washington  508
## 6       Yuma 1040
```

Now we can see the total number of confirmed cases per county.

In order to calculate incidence rate, we need to create a new data frame that includes the population for each county.

```
county = c("Logan","Morgan","Phillips","Sedgwick","Washington","Yuma")
population = c(21914,28984,4278,2229,4742,10063)
incid.county = c(4311,3046,451,269,508,1040)
incidence = data.frame(county,population,incid.county)
attach(incidence)
```

```
## The following objects are masked _by_ .GlobalEnv:
##
##     county, incid.county, population
```

```
head(incidence)
```

```
##       county population incid.county
## 1      Logan      21914         4311
## 2     Morgan      28984         3046
## 3   Phillips       4278          451
## 4   Sedgwick       2229          269
## 5 Washington       4742          508
## 6       Yuma      10063         1040
```

I can now use the number of cases for each county and county population to calculate incidence rate.

```
incidence.rate = (incid.county/population)*100000

## attach incidence rate to incidence data frame
incidence$incidence.rate = incidence.rate
attach(incidence)
```

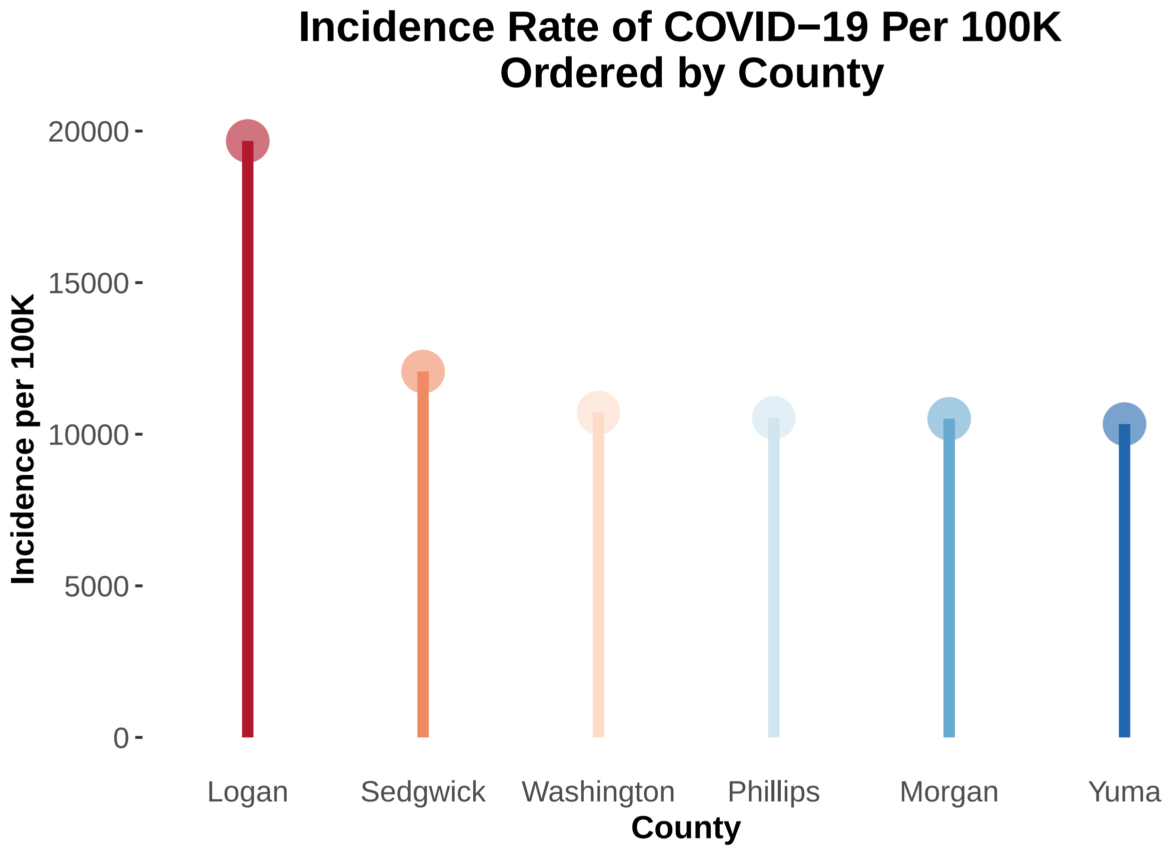
```
## The following objects are masked _by_ .GlobalEnv:
##
##     county, incid.county, incidence.rate, population

## The following objects are masked from incidence (pos = 3):
##
##     county, incid.county, population
```

```
head(incidence)
```

```
##       county population incid.county incidence.rate
## 1      Logan      21914         4311       19672.36
## 2     Morgan      28984         3046       10509.25
## 3   Phillips       4278          451       10542.31
## 4   Sedgwick       2229          269       12068.19
## 5 Washington       4742          508       10712.78
## 6       Yuma      10063         1040       10334.89
```

Finally, we can visualize this data using a segment plot.

```
ggplot(incidence, aes(x = reorder(county, -incidence.rate), y = incidence.rate,
    color = reorder(county, -incidence.rate), fill = reorder(county,
        -incidence.rate))) + geom_segment(aes(x = reorder(county,
    -incidence.rate), xend = reorder(county, -incidence.rate),
    y = 0, yend = incidence.rate), size = 2) + geom_point(size = 7,
    shape = 21, alpha = 0.6) + labs(x = "County", y = "Incidence per 100K") +
    scale_fill_brewer(palette = "RdBu") + scale_color_brewer(palette = "RdBu") +
    ggtitle("Incidence Rate of COVID-19 Per 100K \n Ordered by County") +
    theme(plot.title = element_text(size = 16, face = "bold",
        hjust = 0.5), axis.ticks.x = element_blank(), axis.text.x = element_text(size = 11),
        axis.title.x = element_text(face = "bold", size = 12),
        axis.title.y = element_text(face = "bold", size = 12),
        axis.text.y = element_text(size = 11), legend.position = "none",
        panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank())
```


Incidence Rate of COVID−19 Per 100K
Ordered by County
Incidence per 100K
Logan
Sedgwick
Washington
Phillips
Morgan
Yuma
County