Stat 480 - Midterm

Vahid Azizi

Academic Honesty Statement

THIS IS AN INDIVIDUAL ASSESSMENT, THIS DOCUMENT AND YOUR ANSWERS ARE FOR YOUR EYES ONLY. ANY VIOLATION OF THIS POLICY WILL BE IMMEDIATELY REPORTED.

Replace the underscores below with your name acknowledging that you have read and understood your institution's academic misconduct policy.

I, Vahid Azizi, hereby state that I have not communicated with or gained information in any way from my classmates or anyone other than the Professor or TA during this exam, and that all work is my own.

Tracking the Global Outbreak of COVID-19

The coronavirus pandemic has sickened more than 1.4 million people, according to official counts. Here, we will explore both the global and local growth of COVID-19 using data sourced on April 8th, 2020.

Part I: Recovery data

This data set contains information on some of the first fully recovered cases of COVID-19. We will look at the time it took these patients to recover, defined as the number of days between a confirmed test and an official discharge date. The data is available at https://raw.githubusercontent.com/Stat480-at-ISU/Stat480-at-ISU.github.io/master/exams/data/covid19-recovered.csv (https://raw.githubusercontent.com/Stat480-at-ISU/Stat480-at-ISU.github.io/master/exams/data/covid19-recovered.csv)

Question #1: An overview

i. Read the data without downloading the file locally.

```
recovery_data <- readr::read_csv("https://raw.githubusercontent.com/Stat480-at-ISU/Stat480-at-ISU.github.io/master/exams/data/covid19-recovered.csv")
```

ii. A first look:

- What are the dimensions of the data?
- What variables are included and what are their types?

```
dim(recovery_data)
```

```
## [1] 100   6
```

```
str(recovery_data)
```

```
## tibble [100 x 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ confirmed : chr [1:100] "1/23/2020" "1/24/2020" "1/24/2020" "1/25/2020" ...
##  $ discharged: chr [1:100] "2020/2/19" "2020/2/7" "2020/2/21" "2020/2/12" ...
##  $ recovery  : chr [1:100] "27 days" "14 days" "28 days" "18 days" ...
##  $ category  : chr [1:100] "Imported" "Imported" "Imported" "Imported" ...
##  $ age       : num [1:100] 66 53 37 36 56 56 35 56 56 56 ...
##  $ gender    : num [1:100] 1 0 1 1 0 1 1 0 1 1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   confirmed = col_character(),
##   ..   discharged = col_character(),
##   ..   recovery = col_character(),
##   ..   category = col_character(),
##   ..   age = col_double(),
##   ..   gender = col_double()
##   .. )
```

Variables:confirmed, discharged, recovery, category, age, gender Variables type:chr,chr,chr,chr,num,num

Question #2: Some wrangling

In order to continue with an analysis of this data, we should make some modifications to it.

i. Use functions from the `tidyverse` package to make the following modifications:
   - Convert the variables `confirmed` and `discharged` into variables of type “date”.

```
library(tidyverse)
library(lubridate)
library(readr)

recovery_data <- recovery_data %>% mutate(confirmed= mdy(confirmed),discharged= ymd(discharged))
str(recovery_data)
```

```
## tibble [100 x 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ confirmed : Date[1:100], format: "2020-01-23" "2020-01-24" ...
##  $ discharged: Date[1:100], format: "2020-02-19" "2020-02-07" ...
##  $ recovery  : chr [1:100] "27 days" "14 days" "28 days" "18 days" ...
##  $ category  : chr [1:100] "Imported" "Imported" "Imported" "Imported" ...
##  $ age       : num [1:100] 66 53 37 36 56 56 35 56 56 56 ...
##  $ gender    : num [1:100] 1 0 1 1 0 1 1 0 1 1 ...
```

```
- Extract the numeric value from the variable `recovery`.
```

```
recovery_data <- recovery_data %>% mutate(recovery= parse_number(recovery))
head(recovery_data)
```

```
## # A tibble: 6 x 6
##   confirmed  discharged recovery category   age gender
##   <date>     <date>        <dbl> <chr>    <dbl>  <dbl>
## 1 2020-01-23 2020-02-19       27 Imported    66      1
## 2 2020-01-24 2020-02-07       14 Imported    53      0
## 3 2020-01-24 2020-02-21       28 Imported    37      1
## 4 2020-01-25 2020-02-12       18 Imported    36      1
## 5 2020-01-27 2020-02-18       22 Imported    56      0
## 6 2020-01-27 2020-02-20       24 Imported    56      1
```

```
- Re-derive the variable `recovery` as the number of days between `confirmed` and `discharged` a
nd save as `recovery_days`.
```

```
recovery_data <- recovery_data %>% mutate(recovery_days= discharged-confirmed)
head(recovery_data)
```

```
## # A tibble: 6 x 7
##   confirmed  discharged recovery category   age gender recovery_days
##   <date>     <date>        <dbl> <chr>    <dbl>  <dbl> <drtn>
## 1 2020-01-23 2020-02-19       27 Imported    66      1 27 days
## 2 2020-01-24 2020-02-07       14 Imported    53      0 14 days
## 3 2020-01-24 2020-02-21       28 Imported    37      1 28 days
## 4 2020-01-25 2020-02-12       18 Imported    36      1 18 days
## 5 2020-01-27 2020-02-18       22 Imported    56      0 22 days
## 6 2020-01-27 2020-02-20       24 Imported    56      1 24 days
```

```
- Convert the variable `category` from type `character` to type `factor`.
- Save this data as `recovered` and use this data for the remaining questions in part I.
```

```
recovered <- recovery_data %>% mutate(category= as.factor(category))
str(recovered)
```

```
## tibble [100 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ confirmed    : Date[1:100], format: "2020-01-23" "2020-01-24" ...
##  $ discharged   : Date[1:100], format: "2020-02-19" "2020-02-07" ...
##  $ recovery     : num [1:100] 27 14 28 18 22 24 8 21 25 11 ...
##  $ category     : Factor w/ 2 levels "Imported","Local": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age          : num [1:100] 66 53 37 36 56 56 35 56 56 56 ...
##  $ gender       : num [1:100] 1 0 1 1 0 1 1 0 1 1 ...
##  $ recovery_days: 'difftime' num [1:100] 27 14 28 18 ...
##   ..- attr(*, "units")= chr "days"
```

ii. Look at a summary of the variables:

```
summary(recovered)
```

```
##    confirmed            discharged           recovery         category
##  Min.   :2020-01-23   Min.   :2020-02-04   Min.   : 1.00   Imported:23
##  1st Qu.:2020-02-04   1st Qu.:2020-02-18   1st Qu.: 7.00   Local   :77
##  Median :2020-02-12   Median :2020-02-23   Median :11.00
##  Mean   :2020-02-11   Mean   :2020-02-24   Mean   :12.29
##  3rd Qu.:2020-02-17   3rd Qu.:2020-03-02   3rd Qu.:16.25
##  Max.   :2020-03-08   Max.   :2020-03-14   Max.   :31.00
##       age            gender    recovery_days
##  Min.   : 0.50   Min.   :0.0   Length:100
##  1st Qu.:34.75   1st Qu.:0.0   Class :difftime
##  Median :41.50   Median :1.0   Mode  :numeric
##  Mean   :42.53   Mean   :0.6
##  3rd Qu.:54.00   3rd Qu.:1.0
##  Max.   :79.00   Max.   :1.0
```

iii. What was the longest amount of time someone represented in this data took to recover from COVID-19? Which observation was this? Use indexing to print this row of the data frame.

```
#first method
max(recovered$recovery)
```

```
## [1] 31
```

```
which.max(recovered$recovery)
```

```
## [1] 50
```

```
recovered[which.max(recovered$recovery),]
```

```
## # A tibble: 1 x 7
##   confirmed  discharged recovery category   age gender recovery_days
##   <date>     <date>        <dbl> <fct>    <dbl>  <dbl> <drtn>
## 1 2020-02-12 2020-03-14       31 Local       54      1 31 days
```

```
#second method
recovered %>% slice(which.max(recovery))
```

```
## # A tibble: 1 x 7
##   confirmed  discharged recovery category   age gender recovery_days
##   <date>     <date>        <dbl> <fct>    <dbl>  <dbl> <drtn>
## 1 2020-02-12 2020-03-14       31 Local       54      1 31 days
```

31 days. observation 50.

iv. When was the first confirmed case in this data? Which observation is this? Use indexing to print this row of the data frame.

```
#first method
min(recovered$confirmed)
```

```
## [1] "2020-01-23"
```

```
which.min(recovered$confirmed)
```

```
## [1] 1
```

```
recovered[which.min(recovered$confirmed),]
```

```
## # A tibble: 1 x 7
##   confirmed  discharged recovery category   age gender recovery_days
##   <date>     <date>        <dbl> <fct>    <dbl>  <dbl> <drtn>
## 1 2020-01-23 2020-02-19       27 Imported    66      1 27 days
```

```
#second method
recovered %>% slice(which.min(confirmed))
```

```
## # A tibble: 1 x 7
##   confirmed  discharged recovery category   age gender recovery_days
##   <date>     <date>        <dbl> <fct>    <dbl>  <dbl> <drtn>
## 1 2020-01-23 2020-02-19       27 Imported    66      1 27 days
```

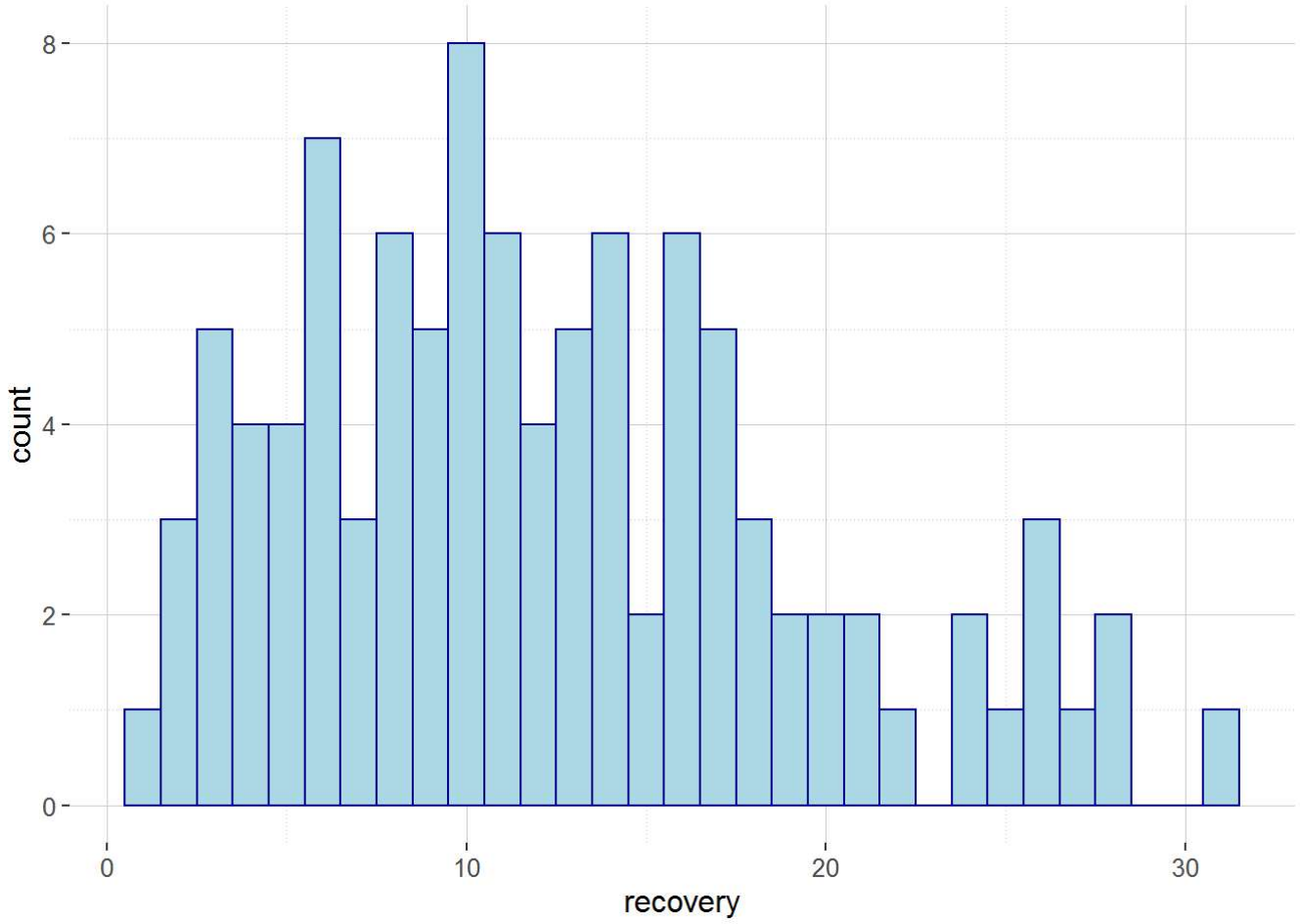
date “2020-01-23” observation 1

Question #4: Time to recovery

If indeed infected, how long would it take for you to be free of the novel coronavirus?

i. Use `ggplot2` to look at the distribution of the variable `recovery` (you may need to adjust the size of the bins).

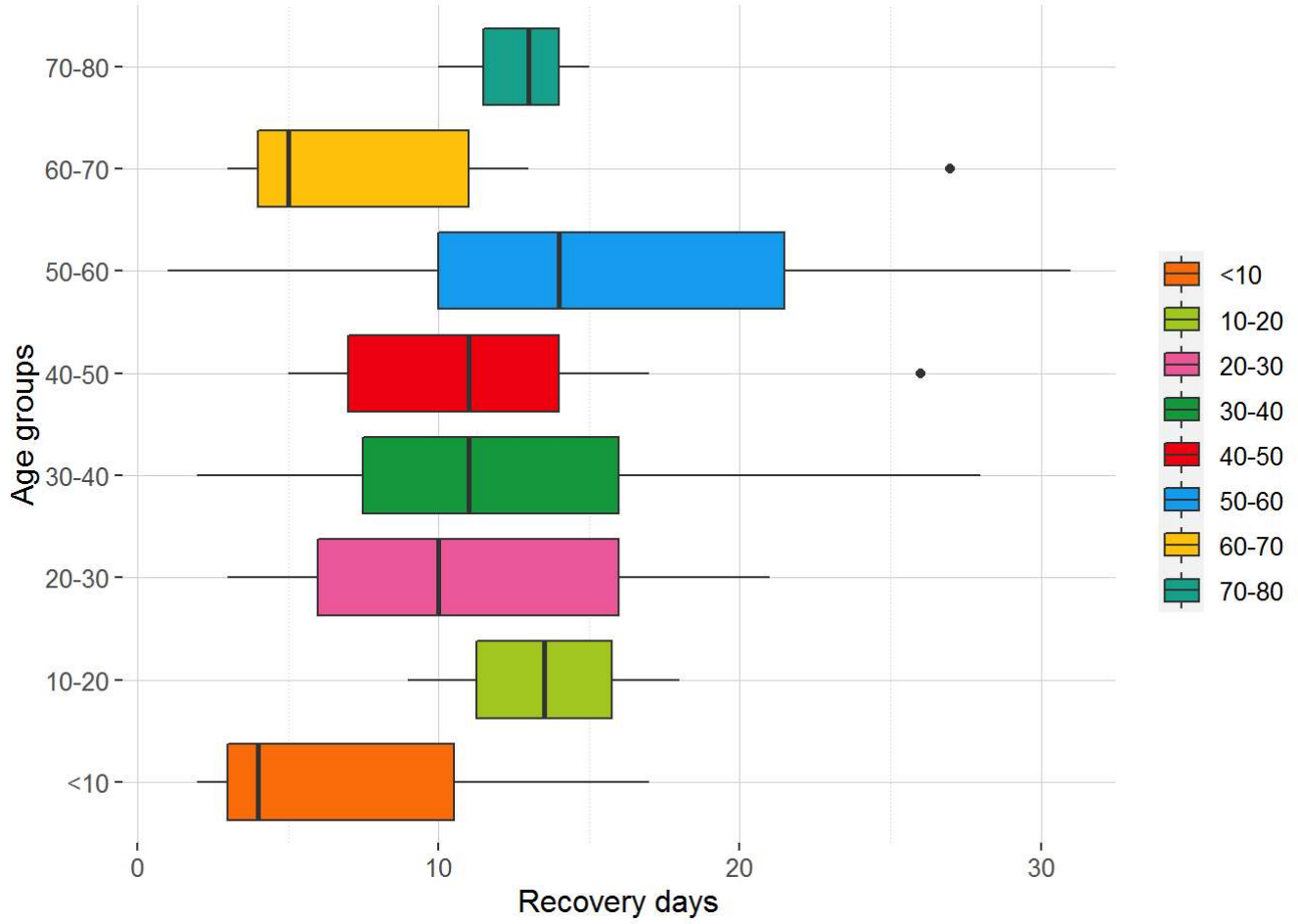
```
recovered %>% ggplot(aes(x = recovery)) +
  geom_histogram(binwidth = 1,color="darkblue", fill="lightblue")
```



ii. Is there a difference in the time it took to recover for different ages?

- Create a new variable `age_blks` from `age` that introduces age categories that groups the ages of the patients into intervals: < 10, 10-20, 20-30, 30-40, 40-50, 50-60, 60-70, 70-80, and >80. (see ?cut).
- Create side-by-side boxplots of the number of days to recovery for the different age groups.
- Flip the coordinates and map the variable `age_blks` to the fill aesthetic.

```
recovered%>% mutate(age_blks=cut(age, c(0,10,20,30,40,50,60,70,80,200),
              labels = c('<10','10-20','20-30','30-40','40-50','50-60','60-70','70-80','>80')))
 %>% group_by(age_blks) %>% ggplot(aes(x=as.factor(age_blks),y=recovery,fill=as.factor(age_blk
s))) +
  geom_boxplot() +
  coord_flip()+
  labs(x ="Age groups", y = "Recovery days")
```

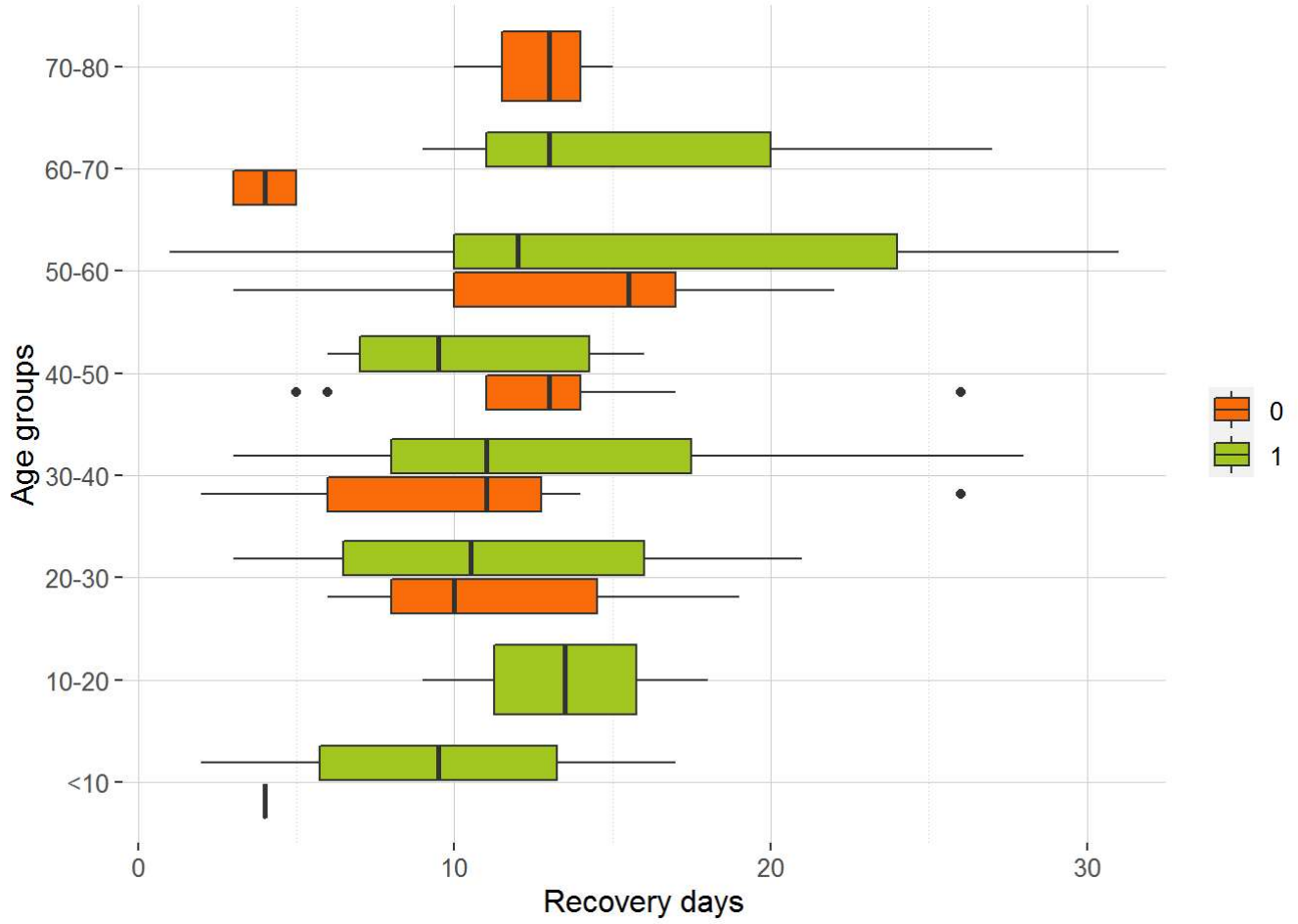


It seems age group <10 recover sooner but we can not conclud it certainly because there is a a small number of data points in this group. Also, for other age group there is no spesific pattern to makea conclusion.

iii. Is there a difference between the genders in the time it took to recover for any of the groups?

- Use the age blocks created in the last question.
- Create side-by-side boxplots for males and females (1's and 0's, respectively) for each of the age groups.
- Fill your boxplots by mapping the variable `gender` to the aesthetic `fill` .

```
recovered%>% mutate(age_blks=cut(age, c(0,10,20,30,40,50,60,70,80,200),
              labels = c('<10','10-20','20-30','30-40','40-50','50-60','60-70','70-80','>80')))
 %>% group_by(age_blks,gender) %>% ggplot(aes(x=as.factor(age_blks),y=recovery,fill=as.factor(ge
nder))) +
  geom_boxplot() +
  coord_flip()+
  labs(x ="Age groups", y = "Recovery days")
```



It seems that women recover sooner than men.

Part II: Global Data

Question #1: First Overview

i. Read the data from https://raw.githubusercontent.com/Stat480-at-ISU/Stat480-at-ISU.github.io/master/exams/data/covid19-global.csv (https://raw.githubusercontent.com/Stat480-at-ISU/Stat480-at-ISU.github.io/master/exams/data/covid19-global.csv) without downloading the file locally. Each line of the file contains daily counts for Province/State-County/Region pair.

```
global_data <- readr::read_csv("https://raw.githubusercontent.com/Stat480-at-ISU/Stat480-at-ISU.github.io/master/exams/data/covid19-global.csv")
```

ii. How many rows and columns does the data have?

```
dim(global_data)
```

```
## [1] 263  81
```

rows=263 cols=81

iii. What are the variables called?

```
colnames(global_data)
```

```
##  [1] "Province/State" "Country/Region" "Lat"           "Long"
##  [5] "1/22/20"        "1/23/20"        "1/24/20"        "1/25/20"
##  [9] "1/26/20"        "1/27/20"        "1/28/20"        "1/29/20"
## [13] "1/30/20"        "1/31/20"        "2/1/20"         "2/2/20"
## [17] "2/3/20"         "2/4/20"         "2/5/20"         "2/6/20"
## [21] "2/7/20"         "2/8/20"         "2/9/20"         "2/10/20"
## [25] "2/11/20"        "2/12/20"        "2/13/20"        "2/14/20"
## [29] "2/15/20"        "2/16/20"        "2/17/20"        "2/18/20"
## [33] "2/19/20"        "2/20/20"        "2/21/20"        "2/22/20"
## [37] "2/23/20"        "2/24/20"        "2/25/20"        "2/26/20"
## [41] "2/27/20"        "2/28/20"        "2/29/20"        "3/1/20"
## [45] "3/2/20"         "3/3/20"         "3/4/20"         "3/5/20"
## [49] "3/6/20"         "3/7/20"         "3/8/20"         "3/9/20"
## [53] "3/10/20"        "3/11/20"        "3/12/20"        "3/13/20"
## [57] "3/14/20"        "3/15/20"        "3/16/20"        "3/17/20"
## [61] "3/18/20"        "3/19/20"        "3/20/20"        "3/21/20"
## [65] "3/22/20"        "3/23/20"        "3/24/20"        "3/25/20"
## [69] "3/26/20"        "3/27/20"        "3/28/20"        "3/29/20"
## [73] "3/30/20"        "3/31/20"        "4/1/20"         "4/2/20"
## [77] "4/3/20"         "4/4/20"         "4/5/20"         "4/6/20"
## [81] "4/7/20"
```

iv. Rename the variables `Province/State`, `Country/Region`, `Lat`, and `Long` to be `province`, `country`, `lat`, and `long`, respectively.

```
colnames(global_data)[1:4] <-c('province','country','lat','long')
colnames(global_data)
```

```
##  [1] "province" "country"  "lat"      "long"     "1/22/20"  "1/23/20"
##  [7] "1/24/20"  "1/25/20"  "1/26/20"  "1/27/20"  "1/28/20"  "1/29/20"
## [13] "1/30/20"  "1/31/20"  "2/1/20"   "2/2/20"   "2/3/20"   "2/4/20"
## [19] "2/5/20"   "2/6/20"   "2/7/20"   "2/8/20"   "2/9/20"   "2/10/20"
## [25] "2/11/20"  "2/12/20"  "2/13/20"  "2/14/20"  "2/15/20"  "2/16/20"
## [31] "2/17/20"  "2/18/20"  "2/19/20"  "2/20/20"  "2/21/20"  "2/22/20"
## [37] "2/23/20"  "2/24/20"  "2/25/20"  "2/26/20"  "2/27/20"  "2/28/20"
## [43] "2/29/20"  "3/1/20"   "3/2/20"   "3/3/20"   "3/4/20"   "3/5/20"
## [49] "3/6/20"   "3/7/20"   "3/8/20"   "3/9/20"   "3/10/20"  "3/11/20"
## [55] "3/12/20"  "3/13/20"  "3/14/20"  "3/15/20"  "3/16/20"  "3/17/20"
## [61] "3/18/20"  "3/19/20"  "3/20/20"  "3/21/20"  "3/22/20"  "3/23/20"
## [67] "3/24/20"  "3/25/20"  "3/26/20"  "3/27/20"  "3/28/20"  "3/29/20"
## [73] "3/30/20"  "3/31/20"  "4/1/20"   "4/2/20"   "4/3/20"   "4/4/20"
## [79] "4/5/20"   "4/6/20"   "4/7/20"
```

v. Each row contains data for one province-country pair. How many countries are represented in this data set?

```
length(unique(global_data$country))
```

```
## [1] 184
```

vi. For each country represented, how many provinces are recorded? Print a table for the five countries with the largest number of provinces recorded.

```
global_data %>% group_by(country) %>% summarise(n_province=length(unique(province[!is.na(provinc
e)]))) %>% arrange(desc(n_province)) %>% top_n(n_province,n=5)
```

```
## # A tibble: 5 x 2
##   country        n_province
##   <chr>               <int>
## 1 China                  33
## 2 Canada                 15
## 3 France                 10
## 4 United Kingdom         10
## 5 Australia               8
```

vii. How many countries do not have any provinces recorded in this data?

```
countries<-global_data %>% group_by(country) %>% summarise(n_province=length(unique(province[!i
s.na(province)])))

n_zero_province<-sum(countries$n_province==0)
n_zero_province
```

```
## [1] 177
```

Question #2: Data wrangling

In order to continue with an analysis of this data, we should reshape it.

i. Use functions from the `tidyverse` package to modify the shape and form of the data:
   - Use a function from `dplyr` to remove the `lat` and `long` variables from the `cases` data.
   - Then use a function from the `tidyr` package to move from wide format into long format where each row represents the number of confirmed cases on a particular date for each country-province pair.
   - Lastly, use a function from `lubridate` to convert the variable `date` from a string into an object of type `date`.
   - Save the resulting data frame as `cases_long`.

```
library(tidyr)
library(lubridate)

cases_long <- global_data %>% select (-c(lat, long)) %>%
pivot_longer(cols = -c('country','province'),
             names_to = "date_",
             values_to = "confirmed")  %>% mutate(date=mdy(date_)) %>% select(c('country','provi
nce','date','confirmed'))
```

ii. Identify the nine countries with the largest number of confirmed cases and save these in a data frame named `cases_by_country`. Plan of attack:
   - Begin with the data frame `cases_long`.
   - Calculate the number of confirmed cases for each country on each date.
   - Find the rank of the countries by current number of confirmed cases for each country.
   - Filter the top nine countries.
   - Save this data frame as `cases_by_country`.

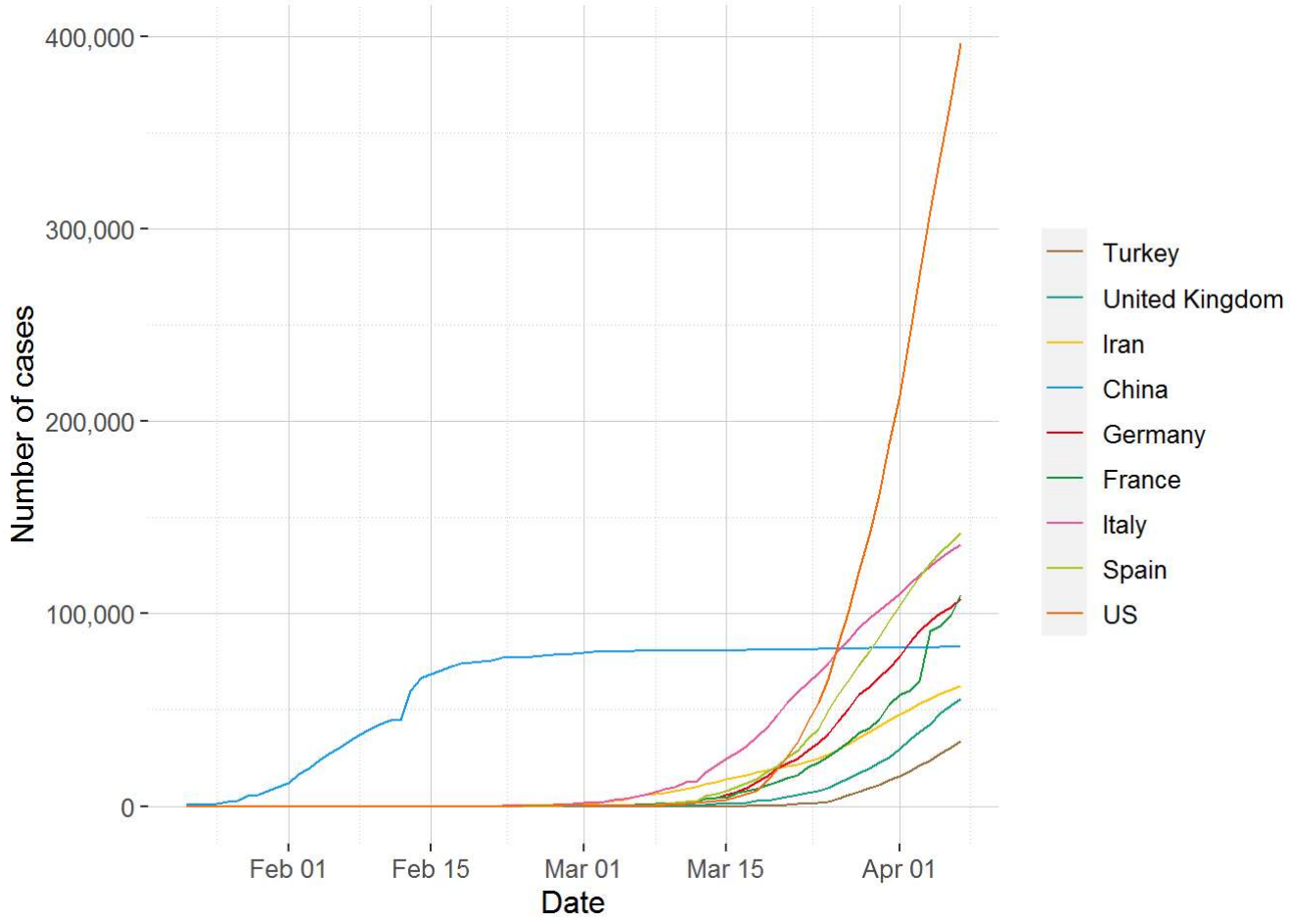
```
temp_data <-cases_long %>% group_by(country,date) %>% summarise(n_cases=sum(confirmed)) %>% ungr
oup()
top_nine_countries<-temp_data %>% filter(date=='2020-04-07') %>% arrange(desc(n_cases)) %>% slic
e(1:9)
names<-top_nine_countries$country
cases_by_country<- temp_data %>% filter(country %in% names)
```

Question #3: Growth over time

i. Let's look at how the number confirmed cases for these nine countries grew over time.

- Start with the data frame `cases_by_country`.
- Use `ggplot2` to plot the number of confirmed cases for each of the nine countries over time.
- Map the variable `country` to color and use the function `fct_reorder2()` from the `forcats` package to align the colors of the lines with the colors in the legend.
- Optional: to make the y-axis labels more readable, add the layer `+ scale_y_continuous(labels = scales::comma)`.

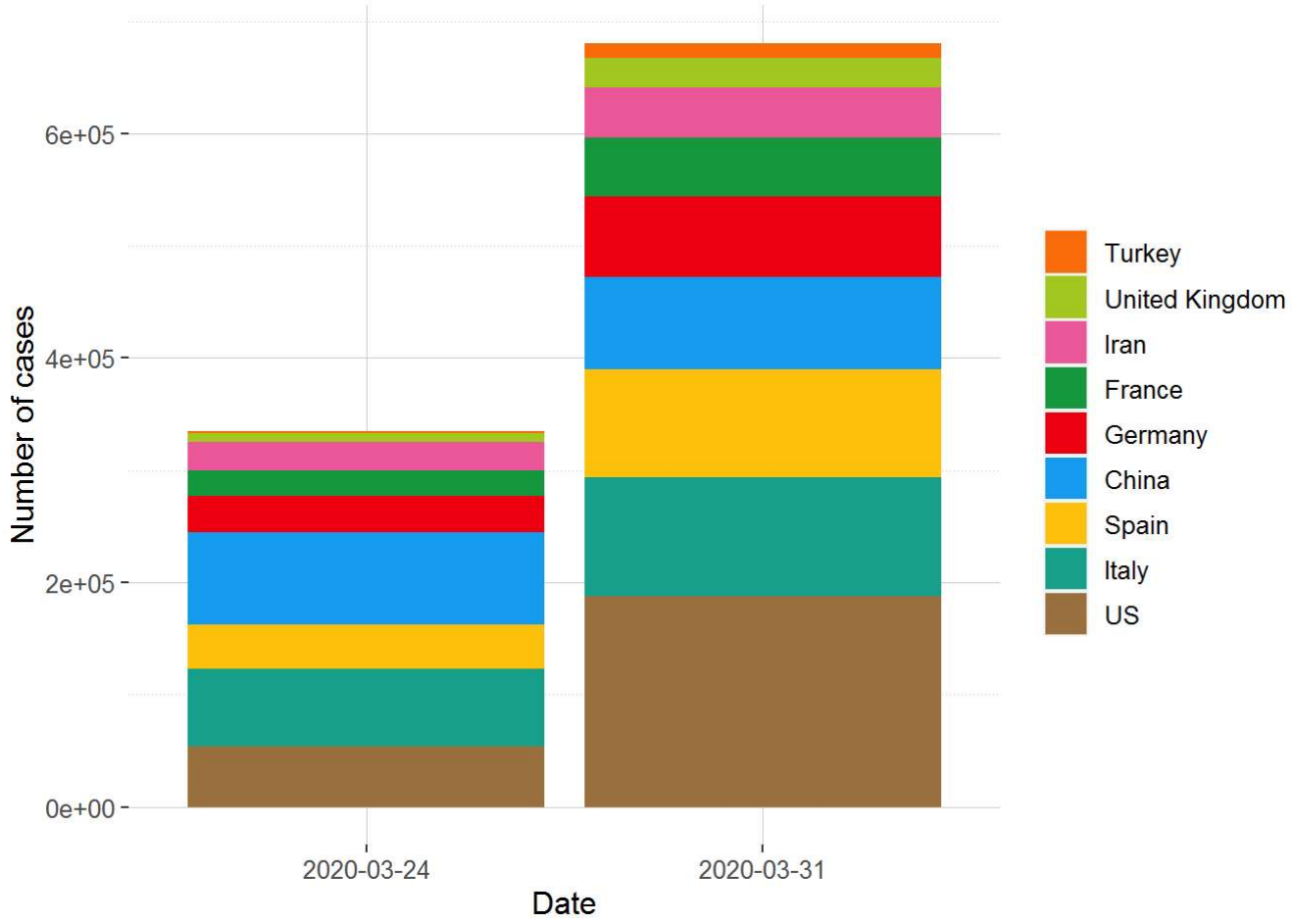
```
library(forcats)
cases_by_country %>% ggplot(aes(x=date,y=n_cases,color=fct_reorder(country,n_cases,.fun="max")))
+
  geom_line() +
  scale_y_continuous(labels = scales::comma)+
  labs(y="Number of cases",x="Date")
```



ii. Let's next look at the difference the last week of March made (Mar 24 vs. Mar 31).

- Use `ggplot2` to create a barchart of the number of cases for the top nine countries for the two dates, sorted according to the total number of cases in that country.
- Make sure the labels of the bars are readable and fill by country.

```
cases_by_country %>% filter((date == "2020-03-24") | (date == "2020-03-31")) %>%
   ggplot(aes(x=as.factor(date),y=n_cases,fill=fct_reorder(country,n_cases,.fun="max"))) +geom_b
ar(stat="identity")+
    labs(y="Number of cases",x="Date")
```



Question #4: Some summaries

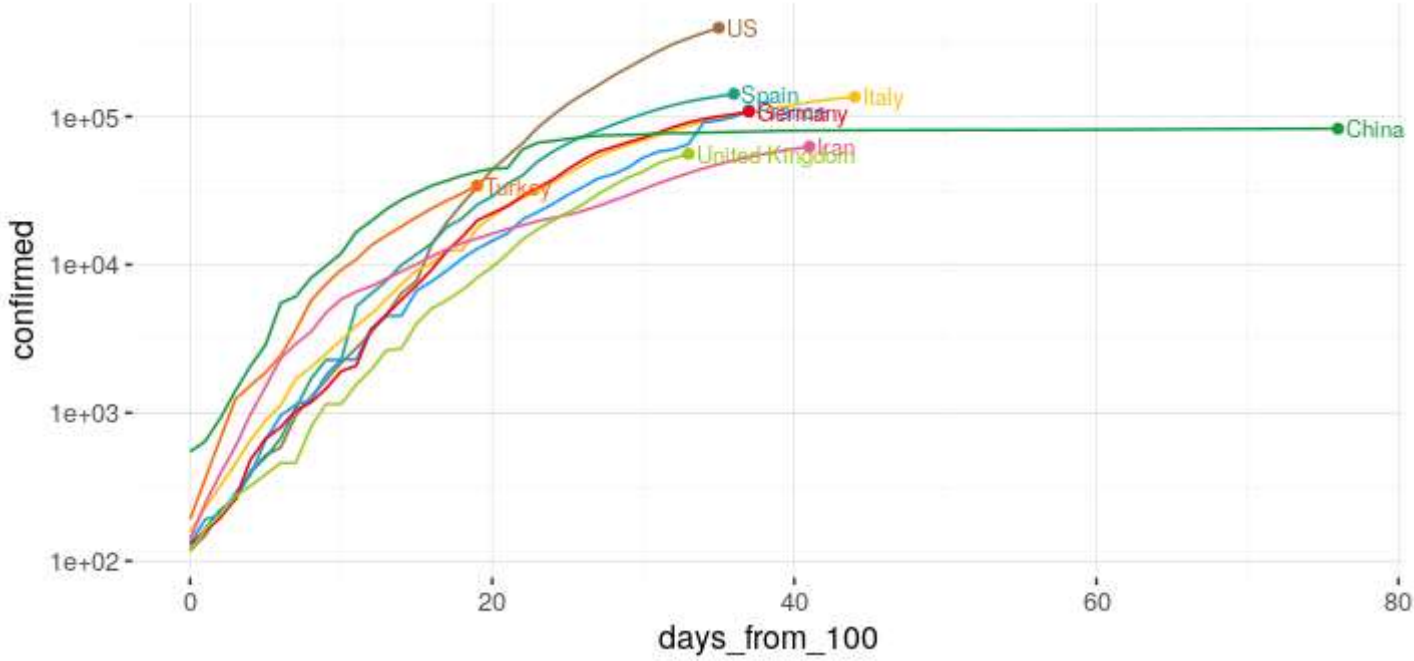
i. How many days did it take for each of the nine countries to go from their 500th case to their 20,000th case?

```
cases_by_country %>% mutate(case_blks=cut(n_cases, c(-1,500,20000,400000), labels = c('<500','50
0-20k','>20k'))) %>% group_by(case_blks) %>% filter(case_blks=="500-20k") %>% group_by(country)
 %>%
  summarise(date_of_500=min(date),date_of_20k=max(date),duration=max(date)-min(date))
```

```
## # A tibble: 9 x 4
##   country        date_of_500 date_of_20k duration
##   <chr>          <date>      <date>      <drtn>
## 1 China          2020-01-22  2020-02-03  12 days
## 2 France         2020-03-06  2020-03-22  16 days
## 3 Germany        2020-03-06  2020-03-20  14 days
## 4 Iran           2020-02-29  2020-03-20  20 days
## 5 Italy          2020-02-27  2020-03-13  15 days
## 6 Spain          2020-03-08  2020-03-19  11 days
## 7 Turkey         2020-03-21  2020-04-02  12 days
## 8 United Kingdom 2020-03-13  2020-03-29  16 days
## 9 US             2020-03-08  2020-03-20  12 days
```

ii. Let's take another look at how the number of cases has grown. This time, though, let's look at the growth for each country starting at their 100th case.

- For each country, calculate the first date that the country had 100 or more cases.
- Introduce a new variable that transforms the date variable into the number of days since the 100th case.
- Save this data frame as `cases100`.
- Create a subset of the `cases100` that contains only the last date and save as `cases100_last`.
- Extra credit: Using `cases100` and `cases100_last`, recreate the visualization below.




```
cases100 <- cases_by_country %>% mutate(case_blks=cut(n_cases, c(-1,100,400000), labels = c('<10
0','>100'))) %>% group_by(case_blks) %>% filter(case_blks==">100") %>% group_by(country) %>% sum
marise(date_of_100=min(date),last_date=max(date),days_Since_100=max(date)-min(date))

cases100_last<- cases100 %>% select(last_date)
```

```
library(directlabels)


cases_by_country %>% left_join(cases100,by="country") %>% mutate(date_diff=date-date_of_100) %>%
filter(date_diff>0) %>% ggplot(aes(x=date_diff,y=n_cases,color=fct_reorder(country,n_cases,.fun=
"max")))+
  geom_line()+
  theme(legend.position="none")+
  geom_dl(aes(label = fct_reorder(country,n_cases,.fun="max")), method = list(dl.combine("last.p
oints"), cex = 0.6)) +
    labs(y="confirmed",x="days_from_100")
```