Impact of Natural Events on Public Health and the Economy from 1950 to 2011, using Data from NOAA

hhp2125

6/7/2020

Contents

Loaded packages:
1. “tidyverse”
2. “data.table”
3. “lubridate”
4. “janitor”
5. “gridExtra”

Chapter 1

Synopsis:

The “Storm” dataset gathers natural events data from 1950 to 2011. We used the storm data set to answer two questions: which natural event has the highest impact on 1) public health, and 2) the economy. The main variables that we are working with from this data set are “evtype” or “event types”, “bgn_date” or “begin date”, “fatalities,” “injuries”, “PROPDMG” or “properties damage” and “CROPDMG” or "crop damage. From our analysis, in terms of public health, **“Tornado”** has the highest impact both in terms of average annual cases of injuries and fatalities, In terms of economic damages, **“Flood”** has the highest economic cost at 4.079 billion dollars annual on average. **Figure 1**, and **Figure 2** shows the results of our analysis.

Chapter 2

Sourcing the data:

2.1 Download the file:

```
if(!file.exists("./storm.csv.bz2")){
url<-"https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2FStormData.csv.bz2"
download.file(url, "./storm.csv.bz2", method = "curl")
}
```

2.2 Read the file:

```
if(!("data" %in% ls())){
data<-fread(file = "./storm.csv.bz2") %>%                    #Read using fread() function
        as_tibble %>%                                        #Convert to tibbles format
        clean_names                                          #Clean names using clean_names function
}
```

The clean_names function did the following:
1. Removed the special characters at the end of the "STATE" variable
2. Lower cases for all variables 3. Make variable names unique, by adding "[a number]" at the end. For example, we have "state" and "state_2" variables corresponding to the state numbers and two-character abbreviations of that state

Chapter 3

Methods:

3.1 Public Health:

3.1.1 Data Analysis Strategy:

Step 1: Find the total number of fatalities and casualties for all the years, stratified by event types
Step 2: Divide the total numbers by the total number of years, which is 62, to get the average annual number.
Step 3: Plot the top 10 highest average annual number and see which event type(s) have or has the highest impact on public health
Step 4: Conclude the result

3.1.2 Data Processing Strategy:

Step 1: Create new data frame only with the column: "bgn_date", "evtype", "injuries", and "fatalities" where only the number of cases is > 0.
Step 2: Convert all the strings in the "evtype" column to lower case
Step 3: Summarize the total sum of cases grouped by the variable "evtype"
Step 4: Sum the results of all the rows that is related to "heat", "tornado", and "tstm" or "thunderstorm", such as "extreme heat" or "tornado f3". Step 5: Replace all the related rows with the summarized row in two new datasets
Step 6: Divide the total number of cases by 62 (years) into a new column for average annual cases values
Step 7: Construct the plots with descending average annual number of cases by type of event

3.2 Economic Damage:

3.2.1 Data Analysis Strategy:

Step 1: Find the total economic cost by combining properties damage and crop damage, stratified by event types. Since we are only focusing on the maximum value, the mismatch of factors such as "+", "-" or "2" in the "exp" columns are insignificant comparing to billions and millions dollars amounts.
Step 2: Divide the total numbers by the total number of years, which is 62, to get the average annual cost
Step 3: Plot the top 10 highest average annual number and see which event type(s) have or has the highest economic impact
Step 4: Conclude the result

3.2.2 Data Processing Strategy:

Step 1: Create new data frame only with the column: "bgn_date", "evtype", "propdmg", "propdmgexp", "cropdmg", and "cropdmgexp" where only the number of dollars is > 0.

Step 2: Convert all the strings in the “evtype”, “propdmgexp” and “cropdmgexp” columns to lower case
Step 3: Rename the factors appropriately: “h” into 100, “k” into 10,000, “m” into 1,000,000, and “b” into 1,000,000,000, and other factors into “1”, and create a new column that sums the economic costs of both crops and properties
Step 4: Sum the costs of similar rows of “flood,” “hurricane|typhoon”, “tornado”, “tstm|thunderstorm” and “drought” such as “flood”" and “flooding”, “tornado” and “tornado f3”, etc.
Step 5: Divide the total number of cases by 62 (years) into a new column for average annual cost types
Step 6: Construct the plot by the descending average annual number cost by types of event

Chapter 4

Data Processing:

4.1 Public Health:

The storm dataset documents the number of cases of fatality and injuries for all types of events. The general strategy is to calculate the mean of total annual fatalities or injuries by different events. We create a new data frame for public health impact analysis

```
#Step 1: Drop all columns unrelated to date, fatalities, injuries, and event type
#Step 2: Only select observations where the number of cases is > 0
#Step 3: Add a column that extracts the year of bgn_date for annual summary.
data_ph<- data %>% select(bgn_date, evtype, fatalities, injuries) %>%
        filter((fatalities > 0 | injuries > 0)) %>%
        mutate(year = year(mdy_hms(.$bgn_date)))
```

We then check the number of unique values of the variable "evtype"

```
check<-unique(data_ph$evtype)
length(check)
```

```
## [1] 220
```

Taking a cursory look at the unique values of the variable "evtype" of the "data_ph" dataset, we can see that there are potential issues. The string "AVALANCE" for example, is obviously a typo of the string "AVALANCHE." Other problems include: separate categories for different forms of the same noun ("FLOOD" vs. "FLOODS"), upper cases vs lower cases ("COLd" vs. "COLD"), or closely related event types ("EXTREME HEAT" vs. "EXCESSIVE HEAT"). To deal with these issues, we shall convert every strings to lower cases.

```
data_ph<- data_ph %>%
        mutate(evtype = str_to_lower(evtype, locale = "en")) %>%
        arrange(evtype)
check<-unique(data_ph$evtype)
length(check)
```

```
## [1] 205
```

We can see that lowering the cases decreases the unique "evtype" to 205 from 220.

Since typos are uncommon, and we are only concerned about the most impactful events, we shall calculate the total number of cases from every causes and see if the typos/similar cases would affect the top 10 highest results.

We make the data frame containing the total injuries from each type of events.

```
# total injuries of each event
total_injuries <- data_ph %>%
        group_by(evtype) %>%
        filter(injuries > 0) %>%
        summarize(sum(injuries)) %>%
        mutate(type = rep("injuries",length(.$evtype)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
names(total_injuries) <- c("evtype","total", "type")
total_injuries<-arrange(total_injuries, desc(total))
```

We make a data frame containing the total fatalities from each type of events.

```
# total fatalities of each event
total_fatalities <- data_ph %>%
        group_by(evtype) %>%
        filter(fatalities > 0) %>%
        summarize(sum(fatalities)) %>%
        mutate(type = rep("fatalities",length(.$evtype)))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
names(total_fatalities) <- c("evtype","total", "type")
total_fatalities<-arrange(total_fatalities, desc(total))
```

Looking at the top 10 observations from both data frame, we can see that "tornado", "heat", and "thunderstorm wind" needs to be further processed by summing the number of cases with that of rows with similar names. For example: "tstm wind" and "thunderstorm winds" are the same, as well as "extreme heat" and "excessive heat."

```
#dealing with similar cases
str_process<-function(x,y){
        if((x == "fatalities" & y == "thunderstorm")){
                sum <- total_fatalities %>%
                        filter(str_detect(evtype, "thunderstorm|tstm")) %>%
                        summarize(sum(total))
        }
        if((x == "fatalities" & y == "heat")){
                sum <- total_fatalities %>%
                        filter(str_detect(evtype, "heat")) %>%
                        summarize(sum(total))
        }
        if((x == "fatalities" & y == "tornado")){
                sum <- total_fatalities %>%
                        filter(str_detect(evtype, "tornado")) %>%
                        summarize(sum(total))
        }
        if((x == "injuries" & y == "thunderstorm")){
                sum <- total_injuries %>%
                        filter(str_detect(evtype, "thunderstorm|tstm")) %>%
                        summarize(sum(total))
        }
        if((x == "injuries" & y == "heat")){
                sum <- total_injuries %>%
                        filter(str_detect(evtype, "heat")) %>%
```

```
                        summarize(sum(total))
        }
        if((x == "injuries" & y == "tornado")){
                sum <- total_injuries %>%
                        filter(str_detect(evtype, "tornado")) %>%
                        summarize(sum(total))
        }
        df<- bind_cols(evtype = y, type = x, total = sum[[1]])
}

fin_mat<-function(x,y){
        if((length(x) == 2 & length(y) == 3)){
        a<-str_process(x[1], y[1])
        b<-str_process(x[1], y[2])
        c<-str_process(x[1], y[3])
        d<-str_process(x[2], y[1])
        e<-str_process(x[2], y[2])
        f<-str_process(x[2], y[3])
        df<- a %>% full_join(b) %>%
                full_join(c) %>%
                full_join(d) %>%
                full_join(e) %>%
                full_join(f)
        }
}
merged_mat<-fin_mat(c("fatalities","injuries"),c("thunderstorm","heat","tornado"))
```

```
## Joining, by = c("evtype", "type", "total")
## Joining, by = c("evtype", "type", "total")
## Joining, by = c("evtype", "type", "total")
## Joining, by = c("evtype", "type", "total")
## Joining, by = c("evtype", "type", "total")
```

At the end of the process, we have a data frame containing the total number of cases for "tornado," "thunderstorm," and "heat." Now we just have to replace the un-merged rows with the merged rows in this data frame.

```
total_injuries_1<- total_injuries %>%
        filter(!str_detect(evtype,"tornado")) %>%
        filter(!str_detect(evtype,"thunderstorm|tstm")) %>%
        filter(!str_detect(evtype, "heat"))
total_fatalities_1<- total_fatalities %>%
        filter(!str_detect(evtype,"tornado")) %>%
        filter(!str_detect(evtype,"thunderstorm|tstm")) %>%
        filter(!str_detect(evtype, "heat"))


mean_tot_inj<- merged_mat %>%
        filter(type == "injuries") %>%
        full_join(total_injuries_1) %>%
        mutate(evtype = str_to_title(evtype)) %>%
        mutate(total = total/length(unique(data_ph$year))) %>%
        arrange(desc(total))
```

```
## Joining, by = c("evtype", "type", "total")
```

```
mean_tot_fat<- merged_mat %>%
        filter(type == "fatalities") %>%
        full_join(total_fatalities_1) %>%
        mutate(evtype = str_to_title(evtype)) %>%
        mutate(total = total/length(unique(data_ph$year))) %>%
        arrange(desc(total))
```

```
## Joining, by = c("evtype", "type", "total")
```

Now we can use these datasets to plot the top highest causes of fatalities and injuries by types.

4.2 Economic Damage

Create a new dataset with selected columns

```
data_econ<- data %>%
        select(bgn_date, evtype, propdmg:cropdmgexp) %>%
        filter((propdmg > 0 | cropdmg > 0)) %>%
        mutate(year = year(mdy_hms(.$bgn_date)))
```

Lower cases for “propdmgexp” and “cropdmgexp” and “evtype”

```
data_econ<- data_econ %>%
        mutate(evtype = str_to_lower(evtype, locale = "en")) %>%
        mutate(propdmgexp = str_to_lower(propdmgexp, locale = "en")) %>%
        mutate(cropdmgexp = str_to_lower(cropdmgexp, locale = "en"))
```

Change factors: “h” into 100, “k” into 10,000, “m” into 1,000,000, and “b” into 1,000,000,000, and other characters into “1”

```
data_econ_1 <- data_econ %>%
        mutate(propdmgexp = as.factor(propdmgexp)) %>%
        mutate(cropdmgexp = as.factor(cropdmgexp)) %>%
        mutate(propdmgexp = fct_recode(.$propdmgexp,"100" = "h","10000" = "k","1000000" = "m","10000000
        mutate(propdmgexp = fct_other(.$propdmgexp,keep = c("100","10000","1000000","1000000000"))) %>%
        mutate(cropdmgexp = fct_recode(.$cropdmgexp,"10000" = "k","1000000" = "m","1000000000" = "b")) %
        mutate(cropdmgexp = fct_other(.$cropdmgexp,keep = c("10000","1000000","1000000000"))) %>%
        mutate(propdmgexp = fct_recode(.$propdmgexp,"1" = "Other")) %>%
        mutate(cropdmgexp = fct_recode(.$cropdmgexp,"1" = "Other"))
```

Create a new column that contains the sum of properties damage and crop damage in millions of dollars

```
data_econ_2 <- data_econ_1 %>%
        mutate(total = ((propdmg*parse_number(as.character(propdmgexp)))/1e6+(cropdmg*parse_number(as.cl
```

Summarize the total damage caused by each event types

```
total_econdmg<- data_econ_2 %>%
        select(evtype, total) %>%
        group_by(evtype) %>%
        summarize(sum(total))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
names(total_econdmg)<-c("evtype","total")

total_econdmg<- total_econdmg %>%
        arrange(desc(total))
```

From the data, we sum the costs of similar rows of "flood," "hurricane|typhoon", "tornado", "tstm|thunderstorm" and "drought"

```
#dealing with similar cases for econ
str_process_1<-function(y){
        if(y == "thunderstorm"){
                sum <- total_econdmg %>%
                        filter(str_detect(evtype, "thunderstorm|tstm")) %>%
                        summarize(sum(total))
        }
        if(y == "flood"){
                sum <- total_econdmg %>%
                        filter(str_detect(evtype, "flood|surge")) %>%
                        summarize(sum(total))
        }
        if(y == "hurricane"){
                sum <- total_econdmg %>%
                        filter(str_detect(evtype, "hurricane|typhoon")) %>%
                        summarize(sum(total))
        }
        if(y == "tornado"){
                sum <- total_econdmg %>%
                        filter(str_detect(evtype, "tornado")) %>%
                        summarize(sum(total))
        }
        if(y == "drought"){
                sum <- total_econdmg %>%
                        filter(str_detect(evtype, "drought")) %>%
                        summarize(sum(total))
        }
        df<- bind_cols(evtype = y, total = sum[[1]])
}

fin_mat_econ<-function(y){
        if(length(y) == 5){
        a<-str_process_1(y[1])
        b<-str_process_1(y[2])
        c<-str_process_1(y[3])
        d<-str_process_1(y[4])
        e<-str_process_1(y[5])
        df<- a %>% full_join(b) %>%
                full_join(c) %>%
                full_join(d) %>%
                full_join(e)
        }
}
merged_mat_econ<-fin_mat_econ(c("thunderstorm","flood","hurricane", "tornado","drought"))
```

```
## Joining, by = c("evtype", "total")
## Joining, by = c("evtype", "total")
## Joining, by = c("evtype", "total")
## Joining, by = c("evtype", "total")
```

Similar to the public health data frame. Now we just have to replace the unm-erged rows with the merged rows in this data frame.

```
total_econdmg_1<- total_econdmg %>%
        filter(!str_detect(evtype,"thunderstorm|tstm")) %>%
        filter(!str_detect(evtype,"flood|surge")) %>%
        filter(!str_detect(evtype, "hurricane|typhoon")) %>%
        filter(!str_detect(evtype,"tornado")) %>%
        filter(!str_detect(evtype, "drought"))

mean_tot_econ<- merged_mat_econ %>%
        full_join(total_econdmg_1) %>%
        mutate(evtype = str_to_title(evtype)) %>%
        mutate(avg = total/length(unique(data_ph$year))) %>%
        arrange(desc(total))
```

```
## Joining, by = c("evtype", "total")
```

With the mean_tot_econ data frame, we can plot the top 10 annual costs by events.

Chapter 5

Results:

```
plot_fat<-ggplot(head(mean_tot_fat, n = 10), aes(x = reorder(evtype, -total), y = total, label = round(
        labs(title = "Top 10 Highest Average Annual Fatalities by Event Types", x = "Event Type", y = "
        geom_label(fill = "white", size = 4)

plot_inj<-ggplot(head(mean_tot_inj, n = 10), aes(x = reorder(evtype, -total), y = total, label = round(
        labs(title = "Figure 1: Top 10 Highest Average Annual Injuries by Event Types", x = "Event Type
        geom_label(fill = "white", size = 4)

grid.arrange(plot_fat,plot_inj, nrow = 2)
```

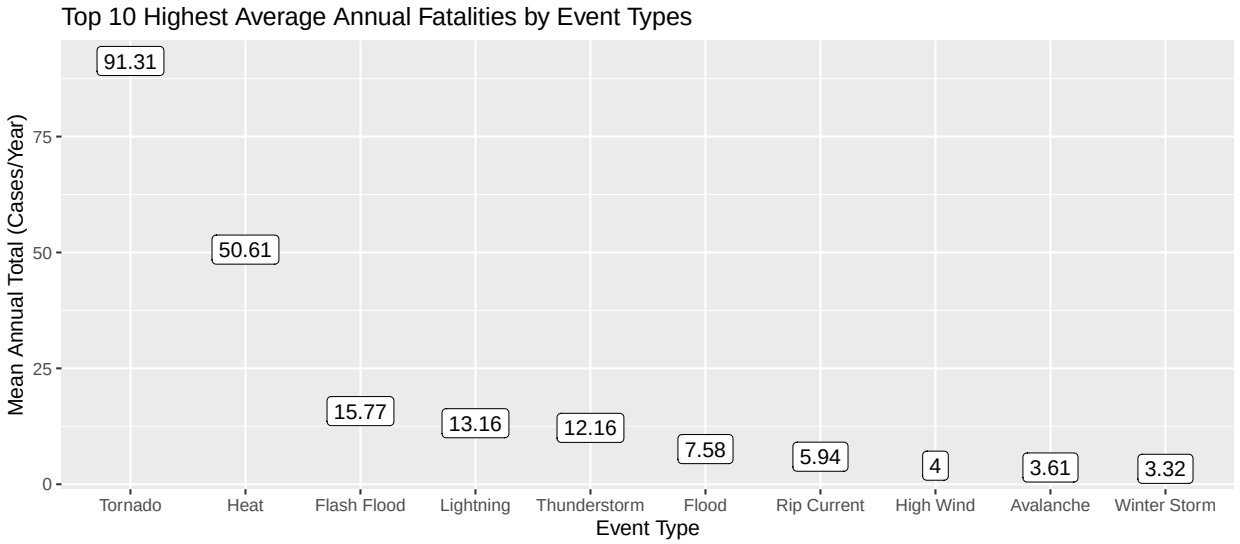
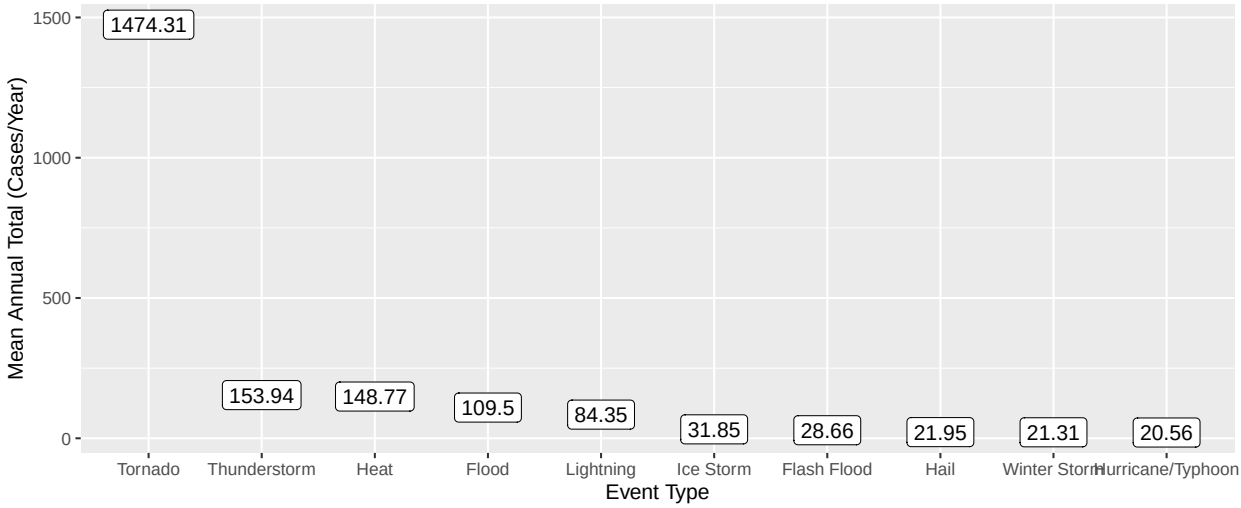


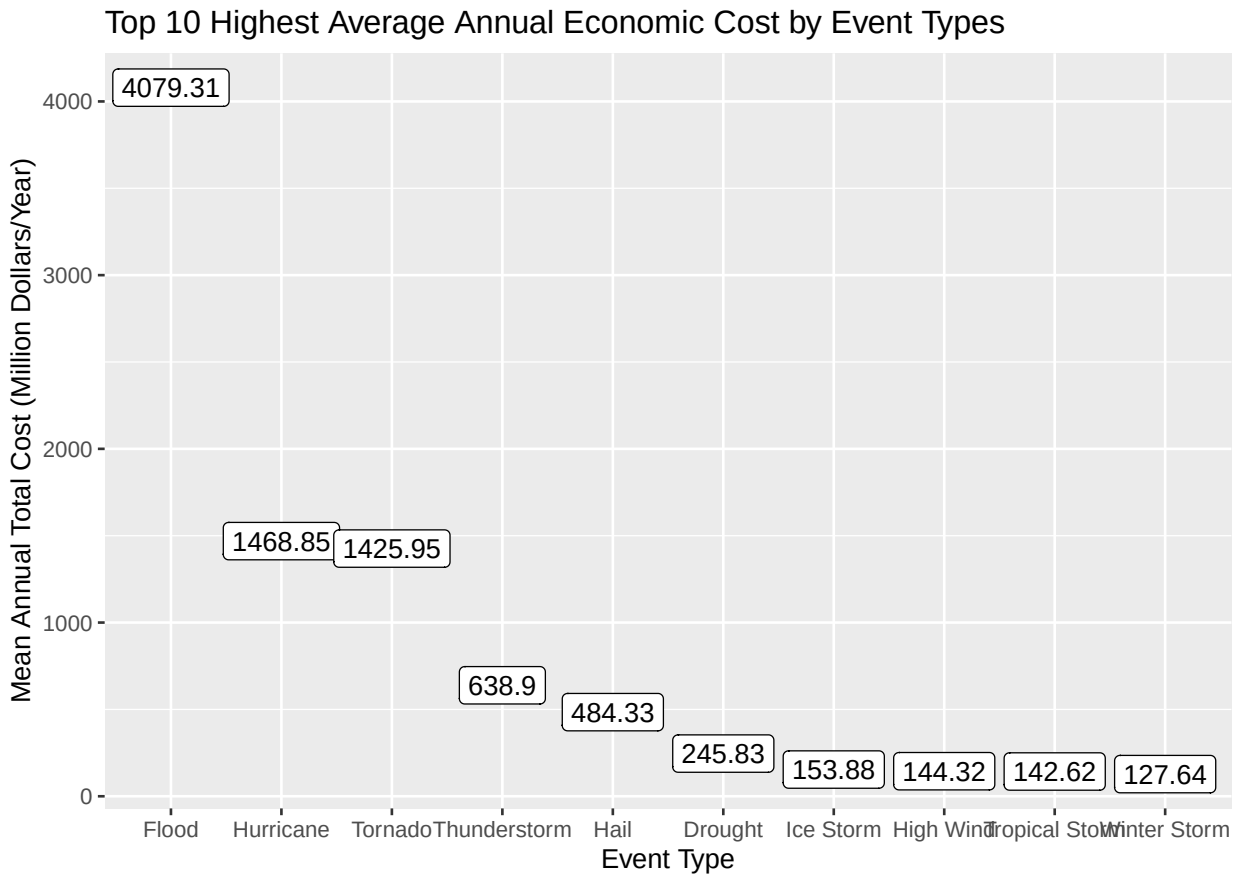
Figure 1: Top 10 Highest Average Annual Injuries by Event Types



From the Figure, we can say that "Tornado" has the most devastating impact to public health based on both the average annual fatalities and injuries.

5.2 Economic Damage

```
plot_econ<-ggplot(head(mean_tot_econ, n = 10), aes(x = reorder(evtype, -avg), y = avg, label = round(avg
        labs(title = "Top 10 Highest Average Annual Economic Cost by Event Types", x = "Event Type", y =
        geom_label(fill = "white", size = 4)
plot_econ
```



From the Figure, we can say that "Flooding" has the most economic impact annually.

5.3 Write to Files:

```
png(file = "Figure 1.png", width = 640, height = 480)
grid.arrange(plot_fat,plot_inj, nrow = 2)
dev.off()
```

```
## pdf
##   2
```

```
png(file = "Figure 2.png", width = 640, height = 480)
plot_econ
dev.off()
```

```
## pdf
##   2
```