

Computer Lab 2 - Gun Laws

Ralf Becker

Introduction

In this computer lab you will be practicing the following

- Creating time series plots with ggplot
- Merge datafiles
- Performing hypothesis tests to test the equality of means
- Estimate and interpret regressions

Let's start by loading some useful packages

```
library(readxl)      # enable the read_excel function
library(tidyverse)   # for almost all data handling tasks
library(ggplot2)     # plotting toolbox
library(stargazer)   # for nice regression output
```

Context

Here we are looking at replicating aspects of this paper:

Siegel et al (2019) The Impact of State Firearm Laws on Homicide and Suicide Deaths in the USA, 1991–2016: a Panel Study, J Gen Intern Med 34(10):2021–8.

this is the paper which we will replicate throughout the unit in order to demonstrate some Diff-in-Diff techniques (not in this session but later).

Data Import

Import the data from the “US_Gun_example.csv” file. Recall, make sure the file (from the [Week 2 BB page](#)) is saved in your working directory, that you set the working directory correctly and that you set the the `na=` option in the `read.csv` function to the value in which missing values are coded in the csv file. To do this correctly you will have to open the csv file (with your spreadsheet software, e.g. Excel) and check for instance cell G9. The `stringsAsFactors = TRUE` option in `read.csv` automatically converts character variables into factor (categorical) variables. This is useful when you know that these variables represent categories (like here states).

```
setwd("YOUR WORKING DIRECTORY")
data <- read.csv(XXXX,na="XXXX", stringsAsFactors = TRUE)
str(data)
```

```
## 'data.frame':   1071 obs. of  19 variables:
## $ X             : int  1 2 3 4 5 6 7 8 9 10 ...
## $ Year           : int  2001 2001 2001 2001 2001 2001 2001 2001 2001 2001 ...
## $ State_code     : Factor w/ 51 levels "AK","AL","AR",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ State          : Factor w/ 51 levels "Alabama","Alaska",...: 2 1 4 3 5 6 7 9 8 10 ...
## $ Population     : int  633687 4467634 2691571 5273477 34479458 4425687 3432835 574504 795699 163...
```

```
## $ Mechanism      : Factor w/ 1 level "Firearm": 1 1 1 1 1 1 1 1 1 1 ...
## $ pol_ubic       : int   0 0 0 0 1 0 1 NA 0 0 ...
## $ pol_vmisd      : int   0 0 0 0 1 0 1 NA 1 0 ...
## $ pol_mayissue    : int   0 1 0 0 1 1 1 NA 1 0 ...
## $ Age.Adjusted.Rate: num  14.83 16.41 15.27 15.92 9.32 ...
## $ logy           : num   2.7 2.8 2.73 2.77 2.23 ...
## $ ur             : num   6.28 5.18 4.76 4.72 5.47 ...
## $ law.officers    : int  1821 15303 7538 18548 106244 15172 9825 4716 2964 63041 ...
## $ law.officers.pc : num   287 343 280 352 308 ...
## $ vcrime          : int  3696 19203 12042 28275 210661 15334 11387 4845 4845 129839 ...
## $ vcrime.pc       : num   583 430 447 536 611 ...
## $ alcc.pc         : num   2.67 1.86 1.74 2.5 2.2 ...
## $ incarceration   : int  3033 24741 11489 27710 157142 14888 17507 NA 6841 72404 ...
## $ incarceration.pc : num   479 554 427 525 456 ...
```

You got it right if the output from `str(data)` looks like the above.

Importing and Merging additional datasets

Age proportion

A variable that is used in the paper but not yet included in the “US_Gun_example.csv” dataset is the age structure of a state’s population. In the Siegel et al. (2019) paper you will find that they use a variable called “Percent male among population ages 15-29”. We shall attempt to use a different variable “Proportion of 18-24 year olds in the population”. You will see below that adding this data to our datasets require a bit of work. With enough work we could add the data used in the paper, but for today’s exercise we will make our life a little easier. But the work done in what follows is quite typical of the work that needs doing when you merge data.

We shall import a new datafile that contains some of that information. The data are sourced from the [StatsAmerica website](#). Download the “US states population age and sex.csv” file from the [Week 2 BB page](#) and save it into your working folder.

```
data_pop <- read.csv(XXXX,na="XXX", stringsAsFactors = TRUE)
str(data_pop)
```

```
## 'data.frame':   63882 obs. of  17 variables:
## $ IBRC_Geo_ID    : int   0 0 0 0 0 0 0 0 0 0 ...
## $ Statefips      : int   0 0 0 0 0 0 0 0 0 0 ...
## $ Countyfips     : int   0 0 0 0 0 0 0 0 0 0 ...
## $ Description    : Factor w/ 3197 levels "Abbeville County, SC",...: 2892 2892 2892 2892 2892 2892 ...
## $ Year           : int  2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ...
## $ Total.Population : int  282162411 284968955 287625193 290107933 292805298 295516599 298379912 30125962 ...
## $ Population.0.4  : int  19178293 19298217 19429192 19592446 19785885 19917400 19938883 20125962 ...
## $ Population.5.17 : int  53197896 53372958 53507265 53508312 53511850 53606269 53818831 53893443 ...
## $ Population.18.24 : int  27315274 27992652 28480708 28916746 29302179 29441546 29602839 29808025 ...
## $ Population.25.44 : int  84973340 84523274 83990295 83398001 83066831 82764185 82638980 82509693 ...
## $ Population.45.64 : int  62428040 64491563 66695526 68828899 70935234 73137401 75216272 77068373 ...
## $ Population.65.   : int  35069568 35290291 35522207 35863529 36203319 36649798 37164107 37825711 ...
## $ Population.Under.18: int  72376189 72671175 72936457 73100758 73297735 73523669 73757714 74019405 ...
## $ Population.18.54 : int  150287588 151902194 152463197 153134701 153998940 154701635 155527978 15635211 ...
## $ Population.55.   : int  59498634 60395586 62225539 63872474 65508623 67291295 69094220 70954145 ...
## $ Male.Population  : int  138443407 139891492 141230559 142428897 143828012 145197078 146647265 14811111 ...
## $ Female.Population : int  143719004 145077463 146394634 147679036 148977286 150319521 151732647 15324848 ...
```

This file has 63882 rows. How many would you have expected if there were 51 states and 21 years for each

state? Exactly, 1,071, which is the number of rows in the `data` object. You need to figure out why there are so many rows before we can merge data into the `data` dataframe. Have a look at the spreadsheet. Can you see the problem/issue?

The spreadsheet includes data for every county and not only for the whole state of Alabama. For instance you see that some rows have in the description column only the name of a state and others have the name of a county. To illustrate this look at the following snippet from the data table:

```
data_pop[1979:1982,]

##      IBRC_Geo_ID Statefips Countyfips      Description Year Total.Population
## 1979          4000         4         0      Arizona 2018      7158024
## 1980          4000         4         0      Arizona 2019      7278717
## 1981          4001         4         1 Apache County, AZ 2000        69507
## 1982          4001         4         1 Apache County, AZ 2001        67863
##      Population.0.4 Population.5.17 Population.18.24 Population.25.44
## 1979          432798          1204802          686858          1863583
## 1980          429788          1210448          693844          1903120
## 1981           6281           20398           6601           17384
## 1982           5821           19498           6530           16785
##      Population.45.64 Population.65. Population.Under.18 Population.18.54
## 1979          1713811          1256172          1637600          3399827
## 1980          1732884          1308633          1640236          3449054
## 1981           13070           5773           26679           31663
## 1982           13312           5917           25319           31174
##      Population.55. Male.Population Female.Population
## 1979          2120597          3560169          3604059
## 1980          2189427          3622802          3669041
## 1981           11165           34441           35066
## 1982           11370           33725           34138
```

Note that in lines 1979 and 1980 we are having data for the whole state of Arizona and in lines 1981 and 1982 you find data for Apache County (in Arizona). We only want statewide data. In the above snippet you can see that there is a variable called `Countyfips` which is a numerical code for the different counties. The statewide data have a value of 0 in the `Countyfips` variable. You should confirm (by looking at the data) that this is true for the other states as well.

One additional aspect of the data is that you will see that population data are only available from 2000 to 2019. This is not aligned with the 2001 to 2021 date range in `data`. The common years are 2001 to 2019 and therefore we should expect to get 969 (=51*19) observations which we can match.

Let us first filter out the statewide data and remove the county level data.

```
data_pop <- data_pop %>% filter(Countyfips == 0) # we only keep data with Countyfips equal to 0
```

You will notice that this dataframe now has `nrow(data_pop2)` rows of data. This is still too many rows. Let's look at the different geographies in our dataset.

```
unique(data_pop$Description)

## [1] U.S.      Alabama   Alaska
## [4] Arizona   Arkansas  California
## [7] Colorado  Connecticut Delaware
## [10] District of Columbia Florida    Georgia
## [13] Hawaii    Idaho     Illinois
## [16] Indiana   Iowa      Kansas
## [19] Kentucky  Louisiana Maine
## [22] Maryland  Massachusetts Michigan
```

```
## [25] Minnesota      Mississippi      Missouri
## [28] Montana         Nebraska         Nevada
## [31] New Hampshire   New Jersey       New Mexico
## [34] New York        North Carolina   North Dakota
## [37] Ohio            Oklahoma         Oregon
## [40] Pennsylvania    Rhode Island     South Carolina
## [43] South Dakota    Tennessee        Texas
## [46] Utah            Vermont          Virginia
## [49] Washington      West Virginia    Wisconsin
## [52] Wyoming         Puerto Rico
## 3197 Levels: Abbeville County, SC Acadia Parish, LA ... Ziebach County, SD
```

You will immediately see that there are also observations for the entire U.S.. So, let's extract the data that are from states and years which are also represented in `data`, our original dataset. Complete the following code for this task.

```
state_list <- unique(data$XXXX) # creates a list with state names in data
year_list <- XXXX(XXXX$Year)    # creates a list of years in data
data_pop <- data_pop %>% filter(Description %in% XXXX) %>%
  filter(XXXX XXXX year_list)
```

You got it right if `data_pop` has 969 observations and you can replicate the following table:

```
summary(data_pop$Total.Population)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  494657 1651059 4219239 6041273 6859789 39512223
```

Let's look at the variables that are contained in this datafile.

```
names(data_pop)
```

```
## [1] "IBRC_Geo_ID"      "Statefips"        "Countyfips"
## [4] "Description"      "Year"             "Total.Population"
## [7] "Population.0.4"    "Population.5.17"  "Population.18.24"
## [10] "Population.25.44"  "Population.45.64" "Population.65."
## [13] "Population.Under.18" "Population.18.54" "Population.55."
## [16] "Male.Population"   "Female.Population"
```

We shall not merge all of these variables into `data` but only what we want, namely the “Proportion of 18-24 year olds in the population”. That is actually not one of the variables in the list. There is the population between 18 and 24 (`Population.18.24`) and the overall population (`Total.Population`) and we can calculate the proportion we need as a new variable, `prop18.24`. Complete the following code:

```
data_pop$prop18.24 <- 100*XXXX$Population.18.24/data_pop$XXXX
```

You get it right if you can replicate these summary statistics for the new variable.

```
summary(data_pop$prop18.24)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   7.955   9.463   9.867   9.944  10.282  14.385
```

Now we select only the variables we wish to merge into `data`, namely only `prop18.24`. However, in order to merge the data into `data` we also need the year (`Year`) and state name (`Description`).

```
data_pop <- data_pop %>% select(Year, Description, prop18.24)
```

It is easiest to merge datafiles if the variables on which we want to match (state name and Year) are called the same in both datasets (`data` and `data_pop`). This is true for the `Year` variable, but not for the state

name (State in data and Description in data_pop). Let's fix that and change the state variable name in data_pop to State.

```
names(data_pop)[names(data_pop)=="Description"] <- "State"
```

Then look at `names(data_pop)` and see whether you achieved what you wanted ... no, you didn't? The name has not changed? Sometimes you make a mistake but there is no error message. Look at the previous line again and try and figure out what the problem is, correct it and rename the variable. But the message here is an important one. Don't assume that just because R ran your line and didn't spit out an error message that everything you wanted to happen did happen. You should always check whether the result is as expected.

```
names(data_pop)[names(data_pop)=="Description"] <- "State"
```

Now we are in a position to merge the two datafiles.

```
data2 <- merge(data, data_pop)
```

As result your datafile has gained one variable, `prop18.24`, but lost a few rows. By default, the merge function deletes rows for which it did not have matching rows in both datafiles and therefore all 2020 and 2021 observations have gone. Look at the help for merge (by typig `?merge` into the console) and find the change you need in the above line to make sure that we keep all 1071 observations from the `data` dataframe. Then re-run the above line.

```
data2 <- merge(data, data_pop, all.x = TRUE)
```

Your `data2` dataframe should end up with 1071 rows and 20 variables.

Region information

Merging datasets is a super important skill, so let's practice this here again. We wish to differentiate between different regions in the U.S. In your `data2` dataframe one of the information is the state, coded by both `State` and `State_code` variables. What we need is an additional variable that tells you which region the state is in.

So, for instance:

State	State code	Region
Alabama	AL	South
Alaska	AK	West
Arizona	AZ	West

You will first have to find a dataset on the internet that maps states to regions. Go to your favorite search engine and search for something like "csv us States and regions". Alternatively you could enlist the help of an AI. For instance you could go to Google Bart and ask something like "create a csv file that maps US states to regions". Then save that file into your working directory and merge the `Region` variable into your `data2` file.

```
states_info <- read_xlsx("states.xlsx")
states_info <- states_info %>% select(STATEAB, Region)
names(states_info)[names(states_info)=="STATEAB"] <- "State_code"
data2 <- merge(data2, states_info)
```

Make sure that the region variable is called `Region`. If you got it right the following code should give you the same result

```
tab_regions <- data2 %>% select(State_code, Region) %>%
  unique() %>%
```

```
group_by(Region) %>%
summarise(n = n()) %>%
print()
```

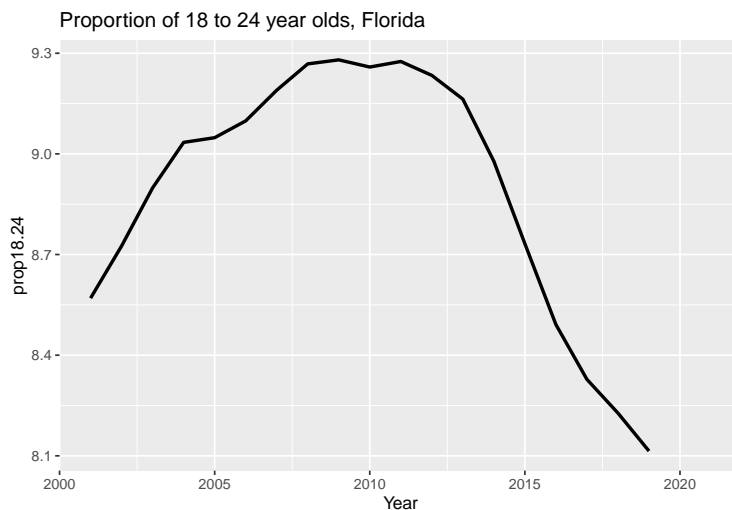
```
## # A tibble: 4 x 2
##   Region      n
##   <chr>    <int>
## 1 Midwest     12
## 2 Northeast     9
## 3 South       17
## 4 West        13
```

This table shows that there are 12 states in the Midwest region and 9 in the Northeast. Altogether the U.S. is divided into four regions.

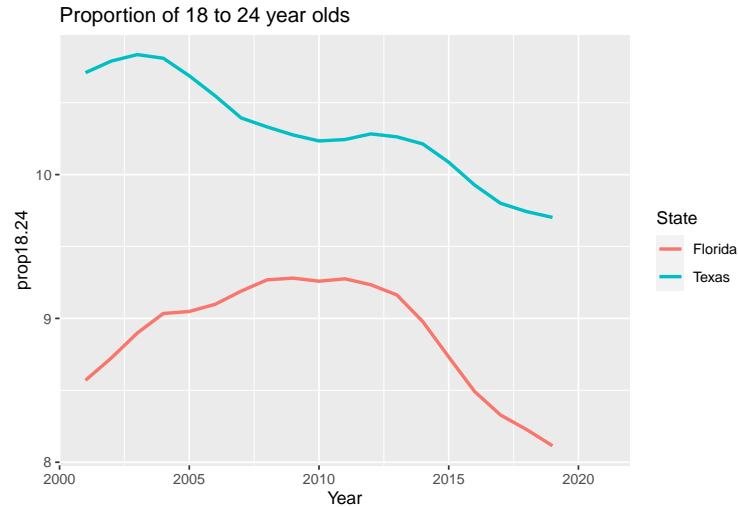
Plotting data as time-series

Here we will practice some time-series plotting. Let's start with a simple plot for `prop18.24` for Florida. You googled the internet to find an example for how to create a time-series plot using `ggplot` and found the following lines which seem relevant. In the bit of code you did find on the internet you find the element `subset(dataset, country == "Brazil")`. This takes a dataframe called `dataset` and extracts all rows for which the variable `country` takes the value "Brazil". Adjust this code to create the following plot from the data in your `data2`.

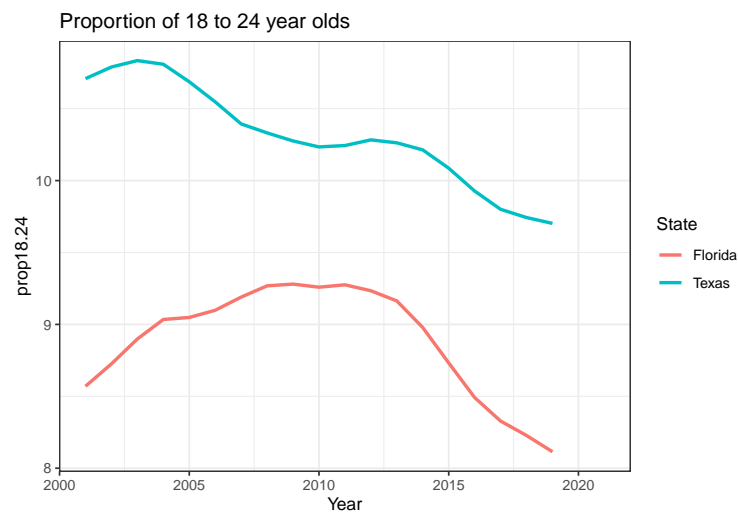
```
g1 <- ggplot(subset(dataset, country == "Brazil"), aes(x=dates,y=deaths_weekly)) +
  geom_line(size=1) +
  ggtitle("Covid-19 weekly deaths, Brazil")
g1
```



Now you want to compare this proportion between Florida and Texas. As you create your graph you should be able to select the data from these two states by using `subset(data2, State %in% c("Florida", "Texas"))`. How to create separate lines for the two states you learned in the Week 2 lecture (or better the accompanying code).



If you wish you could experiment with the **theme** options in ggplot. Why not try the following code:



Check out the [GGplot cheat sheet](#) for more tricks and illustrations of the ggplot packages' capabilities and other themes available.

Average data over the sample period

What we now do is to aggregate or average data across the sample period. We shall use the awesome power of the tidyverse language to do this. We want to calculate the average `prop18.24`, the average rate of firearm deaths (`Age.Adjusted.Rate`), the average for `law.officers.pc` and the average of `ur` for each state.

```
tab1 <- data2 %>% group_by(State) %>%
  summarise(avg_prop18.24 = mean(prop18.24),
            avg_fad.rate = mean(Age.Adjusted.Rate),
            avg_law.officers.pc = mean(law.officers.pc),
            avg_ur = mean(ur)) %>%
  print()
```

```
## # A tibble: 51 x 5
##   State          avg_prop18.24 avg_fad.rate avg_law.officers.pc avg_ur
##   <fct>              <dbl>         <dbl>              <dbl> <dbl>
```

```
## 1 Alabama NA 18.5 310. 6.04
## 2 Alaska NA 19.9 274. 6.93
## 3 Arizona NA 15.3 338. 6.26
## 4 Arkansas NA 17.0 316. 5.55
## 5 California NA 8.40 314. 7.32
## 6 Colorado NA 12.4 336. 5.33
## 7 Connecticut NA 5.22 275. 5.89
## 8 Delaware NA 10.4 355. 5.32
## 9 District of Columbia NA 18.1 791. 7.40
## 10 Florida NA 11.9 371. 5.69
## # i 41 more rows
```

As you can see the code calculated the average values for the firearm death rate (for instance, on average there were 18.5 firearm deaths per 100,000 in Alabama per year), but the code did not calculate the average proportion of 18 to 24 year olds. We get “NA” for all states. The reason for that is that some of the `prop18.24` observations are not available, in particular the data for years 2020 and 2021. The `mean` function by default refuses to calculate the mean value if any of the data are “NA”s. However, there is a way to instruct the `mean` function to ignore missing values and calculate the mean value on the basis of the available data. Check the help for the mean function (type `?mean` into the console) to find the option you should add to the mean function to achieve this.

If you get it right you should be able to replicate the following table.

```
## # A tibble: 51 x 5
##   State      avg_prop18.24 avg_fad.rate avg_law.officers.pc avg_ur
##   <fct>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Alabama      9.86      18.5      310.      6.04
## 2 Alaska     10.4      19.9      274.      6.93
## 3 Arizona      9.93      15.3      338.      6.26
## 4 Arkansas      9.73      17.0      316.      5.55
## 5 California    10.2      8.40      314.      7.32
## 6 Colorado      9.79      12.4      336.      5.33
## 7 Connecticut    9.17      5.22      275.      5.89
## 8 Delaware      9.67      10.4      355.      5.32
## 9 District of Columbia 12.5      18.1      791.      7.40
## 10 Florida      8.89      11.9      371.      5.69
## # i 41 more rows
```

It is possibly not good practice to calculate averages over different sample sizes (over 2001 to 2019 for `prop18.24` and over 2001 to 2021 for `Age.Adjusted.Rate`). We therefore repeat the calculation but only for the years up to and including 2019.

There is one mistake in the code and you should get an error message.

```
tab1 <- data2 %>% filter(year <= 2019) %>%
  group_by(State) %>%
  summarise(avg_prop18.24 = mean(prop18.24, na.rm = TRUE),
            avg_fad.rate = mean(Age.Adjusted.Rate),
            avg_law.officers.pc = mean(law.officers.pc),
            avg_ur = mean(ur)) %>%
  print()
```

Fix the error to obtain the following table.

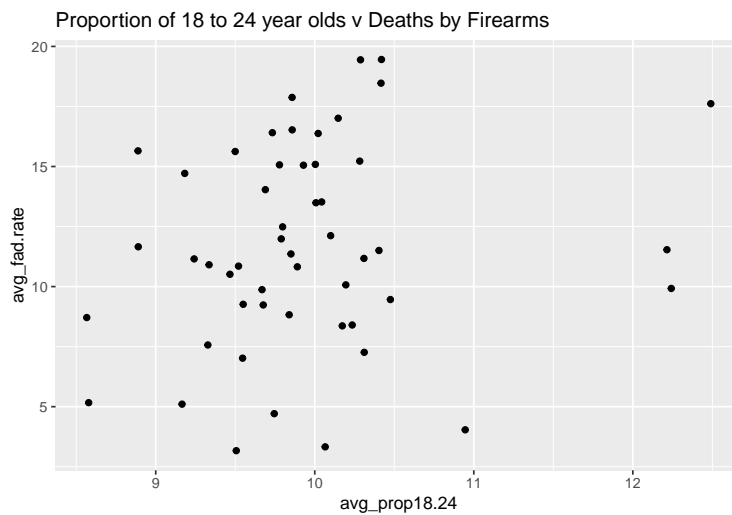
```
## # A tibble: 51 x 6
##   State      Region avg_prop18.24 avg_fad.rate avg_law.officers.pc avg_ur
##   <fct>      <chr>      <dbl>      <dbl>      <dbl>      <dbl>
## 1 Alabama    South      9.86      17.9      305.      6.16
```



```
## 2 Alaska      West      10.4      19.5      275.    6.88
## 3 Arizona     West      9.93     15.1     341.    6.24
## 4 Arkansas    South     9.73     16.4     313.    5.59
## 5 California  West     10.2     8.36     315.    7.17
## 6 Colorado    West     9.79     12.0     336.    5.25
## 7 Connecticut North~    9.17     5.11     278.    5.76
## 8 Delaware    South     9.67     9.87     354.    5.19
## 9 District of Col~ South    12.5     17.6     797.    7.40
## 10 Florida    South     8.89     11.7     374.    5.61
## # i 41 more rows
```

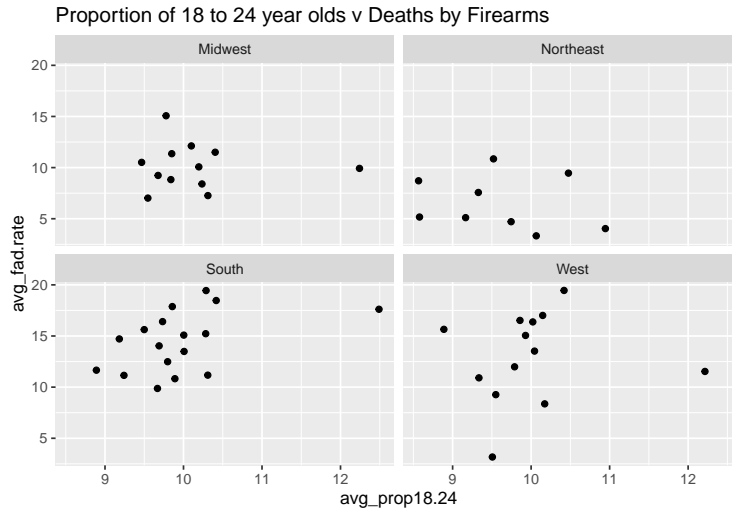
Let's create a few plots which show the average death numbers against some of our country specific information.

```
ggplot(tab1,aes(avg_prop18.24,avg_fad.rate)) +
  geom_point() +
  ggtitle("Proportion of 18 to 24 year olds v Deaths by Firearms")
```



In order to demonstrate another trick in ggplot box of tricks we will also use the `Region` information.

```
ggplot(tab1,aes(avg_prop18.24,avg_fad.rate)) +
  geom_point() +
  facet_wrap(vars(Region)) +
  ggtitle("Proportion of 18 to 24 year olds v Deaths by Firearms")
```



Very neat indeed. What we learn from these scatterplots is that there is no obvious correlation between the average proportions of 18 to 24 year olds and the rate of firearm deaths.

Testing for equality of means

Let's perform some hypothesis tests to check whether there are significant differences between the average rates of cases and deaths since June 2020 between continents.

We therefore continue to work with the data in `table3`. In `table4` we calculate continental averages.

```
tab2 <- tab1 %>%
  group_by(Region) %>%
  summarise(RAvg_cases = mean(avg_fad.rate),
            n = n()) %>% print()
```

```
## # A tibble: 4 x 3
##   Region   RAvG_cases    n
##   <chr>     <dbl> <int>
## 1 Midwest     10.1     12
## 2 Northeast     6.55     9
## 3 South       14.4     17
## 4 West        13.0     13
```

Let's see whether we find the regional averages to be statistically significantly different. Say we compare the `avg_fad.rate` in the Northeast to that in the Midwest. So test the null hypothesis that $H_0 : \mu_{NE} = \mu_{MW}$ (or $H_0 : \mu_{NE} - \mu_{MW} = 0$) against the alternative hypothesis that $H_A : \mu_{NE} \neq \mu_{MW}$, where μ represents the average firearm death rate (per 100,000 population) in states in the respective region over the sample period.

```
test_data_NE <- tab1 %>%
  filter(Region == "Northeast")      # pick Northeast states

test_data_MW <- tab1 %>%
  filter(Region == "Midwest")        # pick Midwest states

t.test(test_data_NE$avg_fad.rate, test_data_MW$avg_fad.rate, mu=0) # testing that mu = 0
```

```
##
## Welch Two Sample t-test
##
```

```
## data: test_data_NE$avg_fad.rate and test_data_MW$avg_fad.rate
## t = -3.2399, df = 15.565, p-value = 0.005282
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -5.895230 -1.225443
## sample estimates:
## mean of x mean of y
## 6.548129 10.108465
```

The difference in the averages is $6.548 - 10.108 = -3.56$ (more than 3 in 100,000 population). We get a t-test statistic of about -3.2. If in truth the two means were the same (H_0 was correct) then we should expect the test statistic to be around 0. Is -3.2 far enough away from 0 for us to conclude that we should stop supporting the null hypothesis? Is -3.2 large (in absolute terms) enough?

The answer is yes and the p-value does tell us that it is. The p-value is 0.005282 0.53%. This means that if the H_0 was correct, the probability of getting a difference of -3.56 (per 100,000 population) or a more extreme difference is 0.53%. We judge this probability to be too small for us to continue to support the H_0 and we reject the H_0 . We do so as the p-value is smaller than any of the usual significance levels (10%, 5% or 1%).

We are not restricted to testing whether two population means are the same. You could also test whether the difference in the population is anything different but 0. Say a politician claims that evidently the firearm death rate in the Northeast is smaller by more than 3 per 100,000 population than the firearm death rate in the Midwest.

Here our H_0 is $H_0 : \mu_{NE} = \mu_{MW} - 3$ (or $\mu_{NE} - \mu_{MW} = -3$) and we would test this against an alternative hypothesis of $H_0 : \mu_{NE} < \mu_{MW} - 3$ (or $H_0 : \mu_{NE} - \mu_{MW} < -3$). Here the statement of the politician is represented in the H_A .

```
# testing that mu = -3
t.test(test_data_NE$avg_fad.rate, test_data_MW$avg_fad.rate, mu=-3, alternative = "less")
```

```
##
## Welch Two Sample t-test
##
## data: test_data_NE$avg_fad.rate and test_data_MW$avg_fad.rate
## t = -0.5099, df = 15.565, p-value = 0.3086
## alternative hypothesis: true difference in means is less than -3
## 95 percent confidence interval:
## -Inf -1.638469
## sample estimates:
## mean of x mean of y
## 6.548129 10.108465
```

Note the following. The parameter mu now takes the value -3 as we are hypothesising that the difference in the means is -3 (or smaller than that in the H_A). Also, in contrast to the previous test we now care whether the deviation is less than -3. In this case we wonder whether it is really smaller. Hence we use the additional input into the test function, `alternative = "less"`. (The default for this input is `alternative = "two.sided"` and that is what is used, as in the previous case, if you don't add it to the `t.test` function). Also check `?t.test` for an explanation of these optional input parameters.

Again we find ourselves asking whether the sample difference we obtained (-3.56) is consistent with the null hypothesis (of the population difference being -3). The p-value is 0.3086, so the probability of obtaining a sample difference as big as -3.56 (or smaller) is just a little over 30%. Say we set out to perform a test at a 10% significance level, then we would judge that a probability of just above 30% is larger than that p-value and we would fail to reject the null hypothesis.

So let's perform another test. A Republican governor of a Southern state in the U.S. claims that the average firearm death rate in the South is just as big as the one in the West. Perform the appropriate hypothesis test.

```

test_data_SO XXXX tab1 %>%
  filter(XXXX == "South")      # pick Southern states

XXXX <- tab1 XXXX
  XXXX(XXXX == "West")        # pick Western states

XXXX(XXXX$avg_fad.rate,XXXX$XXXX, XXXX=0) # testing that mu = 0

##
##  Welch Two Sample t-test
##
## data:  test_data_SO$avg_fad.rate and test_data_WE$avg_fad.rate
## t = 1.0149, df = 19.808, p-value = 0.3224
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.515748  4.384902
## sample estimates:
## mean of x mean of y
##  14.42065  12.98607

```

The p-value is certainly larger than any of the usual significance levels and we fail to reject H_0 . This means that the opposition governor's statement is supported by the data or at least the data do not contradict it.

Regression

To perform inference in the context of regressions it pays to use an additional package, the `car` package. So please load this package.

```
library(car)
```

If you get an error message it is likely that you first have to install that package.

Estimating and interpreting a regression model

In the lecture we talked about the following regression (Lecture Week 2 - Regression Analysis - Example 3)

$$vcrime.pc_i = \alpha + \beta law.officer.pc_i + u_i$$

Let us estimate this again, using the subset function to filter the 2021 data (as in the lecture) from `data2`.

```

mod1 <- lm(vcrime.pc~law.officers.pc,data=subset(data2, Year == 2021))
stargazer(mod1,type = "text")

```

```

##
## =====
##                               Dependent variable:
##                               -----
##                               vcrime.pc
## -----
## law.officers.pc              0.204
##                               (0.193)
##
## Constant                     311.421***
##                               (61.296)
## -----
##

```

```
## Observations          50
## R2                    0.023
## Adjusted R2           0.002
## Residual Std. Error   156.747 (df = 48)
## F Statistic           1.114 (df = 1; 48)
## =====
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

Let's change the dependent variable to the rate of firearm deaths (`Age.Adjusted.rate` or `AAR` for short) and use several explanatory variables, the number of law officers, the unemployment rate and the amount of per capita alcohol consumption. Also, let's use all the years of data in our dataset.

$$AAR_i = \alpha + \beta_1 \text{law.officer.pc}_i + \beta_2 \text{ur}_i + \beta_3 \text{alcc.pc}_i + u_i$$

We will estimate two models, one with only `law.officers.pc` as the explanatory variable and one with all three explanatory variables.

```
mod2 <- lm(Age.Adjusted.Rate~law.officers.pc,data=data2)
mod3 <- lm(Age.Adjusted.Rate~law.officers.pc+ur+alcc.pc,data=data2)
stargazer(mod2,mod3,type = "text")
```

```
##
## =====
##                               Dependent variable:
##                               -----
##                               Age.Adjusted.Rate
##                               (1)                (2)
## -----
## law.officers.pc              0.006***          0.007***
##                               (0.002)          (0.002)
##
## ur                           0.039
##                               (0.076)
##
## alcc.pc                      -0.528*
##                               (0.286)
##
## Constant                    10.301***          11.151***
##                               (0.501)          (0.865)
## -----
## Observations                 1,048             1,048
## R2                           0.014             0.017
## Adjusted R2                  0.013             0.014
## Residual Std. Error          4.907 (df = 1046)    4.902 (df = 1044)
## F Statistic                   14.461*** (df = 1; 1046) 6.118*** (df = 3; 1044)
## =====
## Note:                        *p<0.1; **p<0.05; ***p<0.01
```

How would we interpret the value of $\hat{\beta}_1 = 0.007$? For a one unit increase in the explanatory variable (law officers per 100,000) we would expect the number of firearm deaths to increase by 0.007. Is that a lot or is that economically significant? There are two aspects to that. Is an increase of 1 officer per 100,000 a lot? To answer that we need to know how many officers there typically are. And to know whether 0.007 is a large increase we need to know how many firearm deaths there typically are.

Let's look at the summary stats to help with this judgement.

```
summary(data2[c("Age.Adjusted.Rate", "law.officers.pc", "ur", "alcc.pc")])
```

```
## Age.Adjusted.Rate law.officers.pc      ur      alcc.pc
## Min.   : 2.14      Min.   : 37.08    Min.   : 2.100    Min.   :1.271
## 1st Qu.: 8.84      1st Qu.:255.98    1st Qu.: 4.204    1st Qu.:2.110
## Median :11.72      Median :296.35    Median : 5.283    Median :2.344
## Mean   :12.05      Mean   :310.21    Mean   : 5.648    Mean   :2.441
## 3rd Qu.:15.05      3rd Qu.:343.69    3rd Qu.: 6.737    3rd Qu.:2.657
## Max.   :33.82      Max.   :894.46    Max.   :13.733    Max.   :4.799
##                                     NA's   :23
```

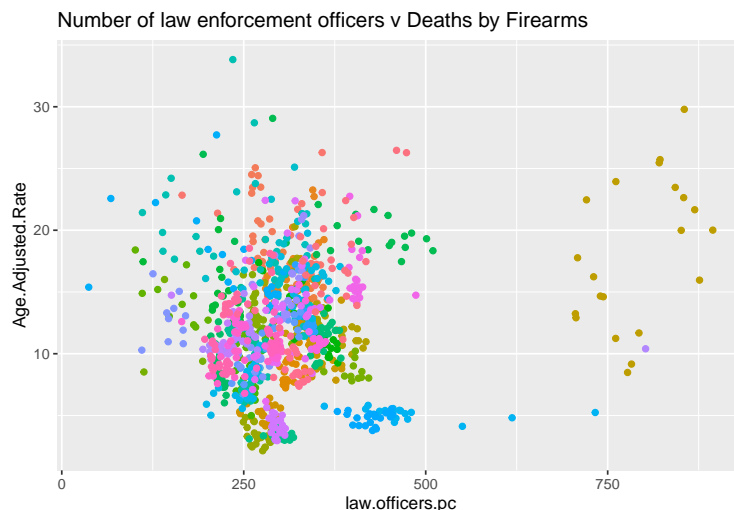
Now we can judge the economic importance of this effect. One officer extra per 100,000 population is not a large change when on average, across all states and years, the average number of officers is 310. So let's consider a 5% increase in the number of officers as this seems like a perhaps feasible but significant policy. That would be around 15 officers per 100,000. This means the effect on the rate of firearm deaths would be $15 \times 0.007 = 0.105$. Now the question is whether this would be a sizable effect on the outcome variable (Age.Adjusted.Rate)? The average rate is 12.05, which implies that the effect of increasing the number of law enforcement officers by 5% would be to increase the number of firearm deaths by less than 1%. This certainly does not imply a very large effect.

You can see that on the face of it, higher numbers of law enforcement officers seem to suggest a higher rate of firearm deaths. This initially certainly seems counter-intuitive until you realise that it is quite likely that police will have higher numbers in states in which crime is a bigger problem. This is a classic example of simultaneity. Crime impacts the numbers of police and the numbers of police may impact crime. So this is an excellent example to understand that just looking at regression results you cannot make causal statements, here you would not be justified in arguing that higher numbers of law enforcement officers **cause** more crime.

We may also want to look at the R^2 of the above regression. You can see that they are both very small 0.014 and 0.017, meaning that both regressions explain less than 2% of the variation in the dependent variable.

Let us investigate a little further why this regression explains so little variation. We plot a scatter graph where different colors represent different states.

```
ggplot(data2, aes(law.officers.pc, Age.Adjusted.Rate, color = State)) +
  geom_point() +
  guides(color = "none") + # removes the legend
  ggtitle("Number of law enforcement officers v Deaths by Firearms")
```



What you can see from here is that observations for the same state cluster together and that different states

seem to differ significantly between each other. This variation is not reflected in the above estimation. In next week's computer lab you will see how this issue can be tackled.

Summary

In today's computer lab you achieved a lot. You produced various plots and started to explore the amazing power of the ggplot package. You merged data, performed hypothesis tests and learned how to run and display multiple regressions.

TAKE A BREAK!