Computer Lab 2 Covid

Ralf Becker

Introduction

In this computer lab you will be practicing the following

- Creating time series plots with ggplot
- Performing hypothesis tests to test the equality of means
- Estimate regressions
- Perform inference on regression coefficients

```
library(sets)         # used for some set operations
library(readxl)       # enable the read_excel function
library(tidyverse)    # for almost all data handling tasks
library(ggplot2)      # plotting toolbox
library(utils)        # for reading data into R # for reading data into R
library(httr)         # for downloading data from a URL
library(stargazer)    # for nice regression output
```

Data Import

Import the data from the "StaticECDCdata_8Feb21.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 F61.

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

```
## 'data.frame':    11157 obs. of  9 variables:
##  $ dateRep                : Factor w/ 59 levels "01/02/2021","01/06/2020",..: 58 10 24 37 50 5 19 32
##  $ year_week              : Factor w/ 59 levels "2020-01","2020-02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ cases_weekly           : int  0 0 0 0 0 0 0 0 1 3 ...
##  $ deaths_weekly          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ countriesAndTerritories: Factor w/ 219 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ geoId                  : Factor w/ 213 levels "AD","AE","AF",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ countryterritoryCode   : Factor w/ 214 levels "","ABW","AFG",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ popData2019            : num  38928341 38928341 38928341 38928341 38928341 ...
##  $ continentExp           : Factor w/ 5 levels "Africa","America",..: 3 3 3 3 3 3 3 3 3 3 ...
```

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

Now we need to change some variable names and set the dates up as dates

```r
names(data)[names(data) == "countriesAndTerritories"] <- "country"
names(data)[names(data) == "countryterritoryCode"] <- "countryCode"
names(data)[names(data) == "dateRep"] <- "dates"

data$dates <- as.Date(as.character(data$dates),format = "%d/%m/%Y")
```

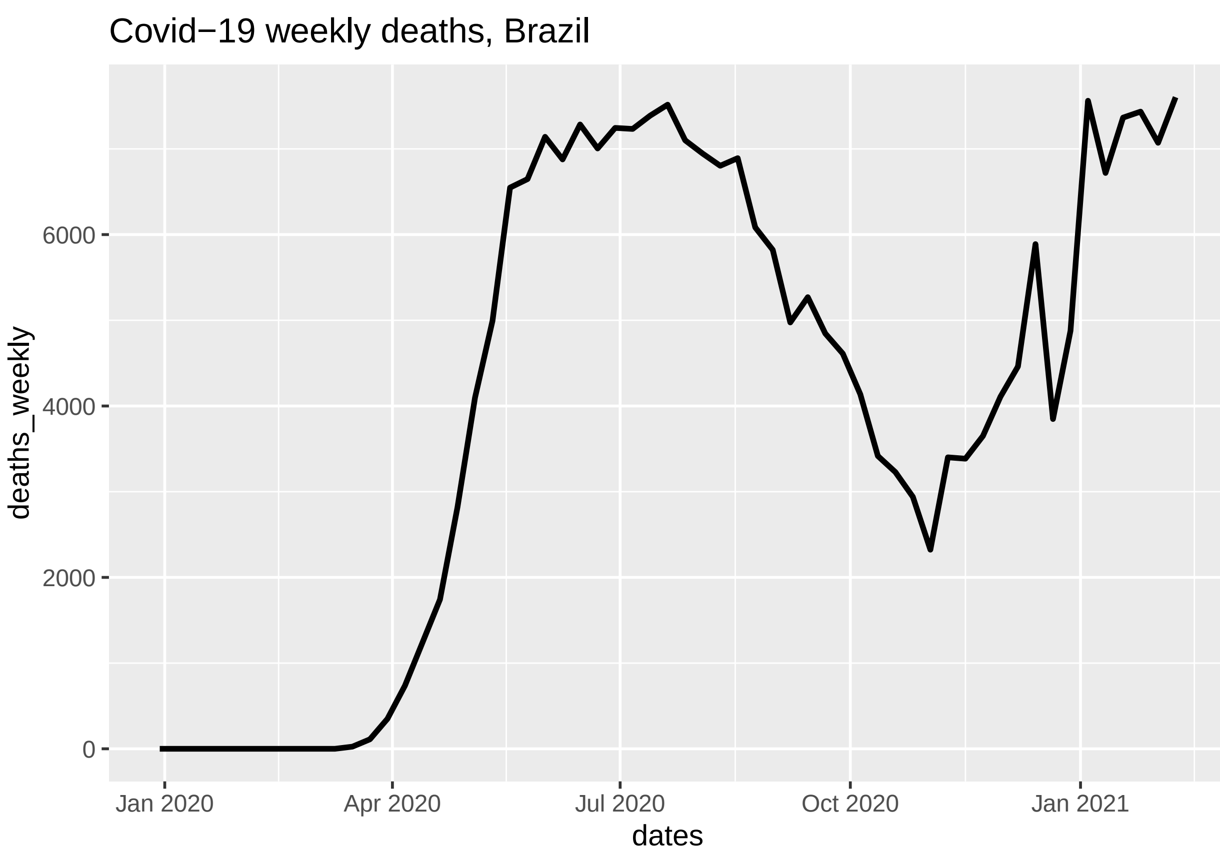
Let's also calculate the per-capita data to ensure that we can compare countries of different sizes.

```r
data <- data %>%
          mutate(pc_cases = (cases_weekly/popData2019)*100000,
                 pc_deaths = (deaths_weekly/popData2019)*100000)
```

Plotting data as time-series

Here we will practice some time-series plotting. Let's start with a simple plot for Brazil.

```r
g1 <- XXXX(subset(XXXX, XXXX == "Brazil"), aes(x=dates,y=deaths_weekly)) +
      geom_XXXX() +
      ggtitle("Covid-19 weekly cases, Brazil")
g1
```



Next we want to compare this development to the similar time line for the two countries which have population size close to Brazil. For that purpose we want to see a Table of Data with merely country names and populations ordered by population size. Then we pick the country with the next smaller and next larger population compared to Brazil.

```r
temp <- data %>% select(country,popData2019) %>%
          unique() %>%
          arrange(desc(popData2019))
head(temp,14)
```

```
##             country popData2019
## 1      Asia (total)  4498460442
## 2             China  1439323774
## 3             India  1380004385
## 4    Africa (total)  1339423921
## 5   America (total)  1021703563
## 6    Europe (total)   851186002
## 7    EU/EEA (total)   453090377
## 8     United States   331002647
## 9         Indonesia   273523621
## 10         Pakistan   220892331
## 11           Brazil   212559409
## 12          Nigeria   206139587
## 13       Bangladesh   164689383
## 14           Russia   145934460
```

Try and figure out what the above does. What do `select`, `unique` and `arrange` do? Could you change the order in which you call these actions?

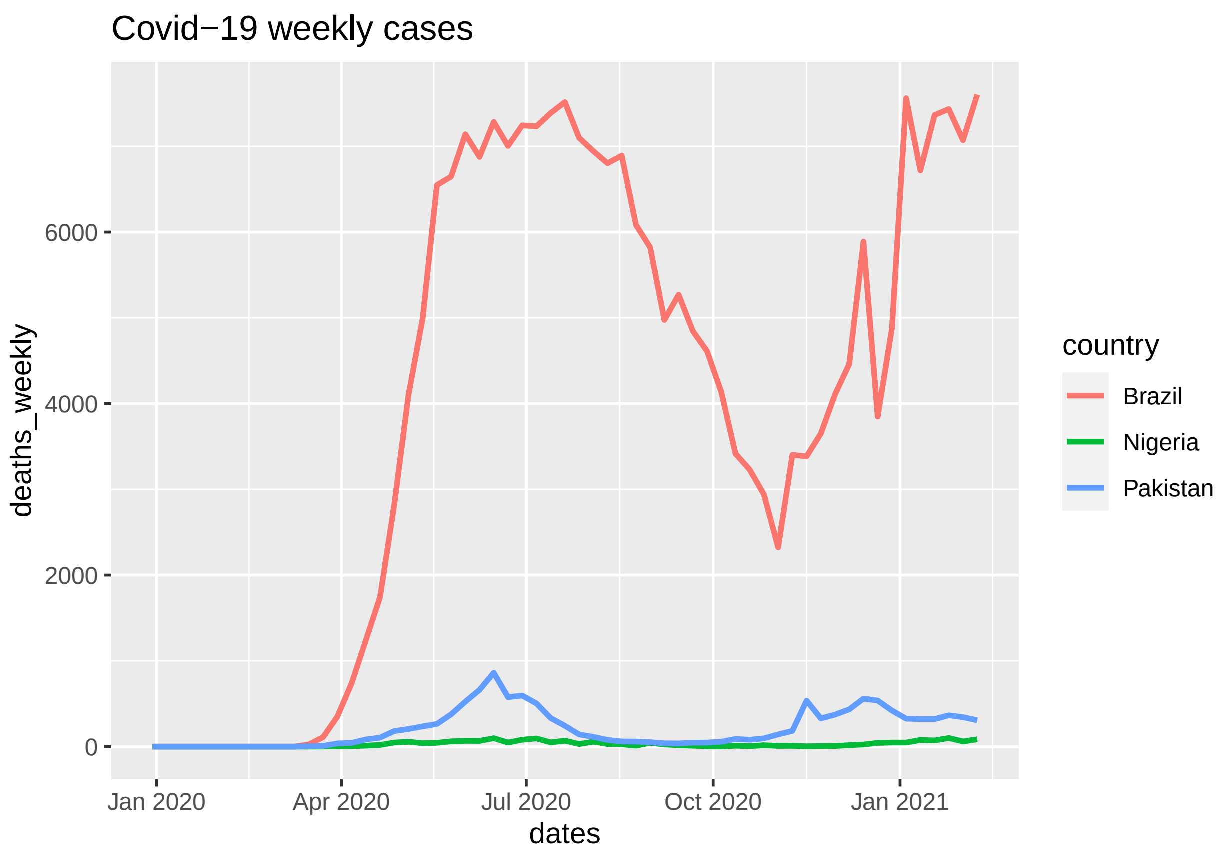
For instance, what does the following do?

```r
temp2 <- data %>% arrange(desc(popData2019)) %>%
          unique() %>%
          select(country,popData2019)
```

You should find that table to be a lot less useful than `temp`.

From Table `temp` you should be able to identify that Pakistan and Nigeria are the next larger and next smaller country.

```r
sel_countries <- c("Brazil", "XXXX", "XXXX")
g2 <- ggplot(subset(XXXX, country %in% XXXX),
             XXXX(x=dates,XXXX=XXXX, color = XXXX)) +
      geom_line(XXXX=1) +
      XXXX("Covid-19 weekly cases")
g2
```



Import additional country indicators

The following three files will add the following variables to your dataframe

- Land_Area_sqkm
- HealthExp
- GDPpc
- Obese_Pcent
- Over_65s
- Diabetis

Make sure that these files are saved in your working directory. In our dataframe `data` we have 2-digit (`geoId`) and 3-digit (`countryCode`) country codes. If at all possible you should merge data on the basis of such codes. Often different organisations name countries slightly differently (e.g. Ivory Coast or Cote d'Ivoire) and only the slightest difference will prevent any matching.

In "CountryIndicators.csv" and "Obesity.csv" we can find a 2-digit `geoID` (note the very slight difference in spelling!) and hence we will match on the basis of this variable. As both these files also contain a variable `country` (with potentially different spellings to those in `data`) we remove these variables before we merge.

```
countryInd <- read_csv("CountryIndicators.csv",na = "#N/A")
countryInd <- countryInd %>% select(-country)
# by.x and by.y specify the matching variables of x (data) and y (countryInd)
data<- merge(data,countryInd,by.x="geoId", by.y="geoID",all.x=TRUE)

obesity <- read_csv("Obesity.csv")   # Adds obesity and diabetis country
```

```
obesity <- obesity %>% select(-country)
data <- merge(data,obesity,by.x="geoId", by.y="geoID",all.x=TRUE)
```

In “Over 65s 2.xlsx” you will find a 3-digit country code (`countryCode`). This is spelled exactly as in `data` and hence we do not need to specify `by.x` and `by.y`. The merge function will, if not advised otherwise by `by.x` and `by.y` match on variables which have the same name in both dataframes.

```
over65p <- read_excel("Over 65s 2.xlsx")
data <- merge(data,over65p,all.x=TRUE)
```

Check whether `data` indeed contains these variables. Which of the following commands is useful for this?

```
view(data)
str(data)
```

```
## 'data.frame':    11157 obs. of  17 variables:
##  $ countryCode   : Factor w/ 214 levels "","ABW","AFG",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ geoId         : Factor w/ 213 levels "AD","AE","AF",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ dates         : Date, format: "2020-10-12" "2020-10-19" ...
##  $ year_week     : Factor w/ 59 levels "2020-01","2020-02",..: 42 43 44 45 46 47 48 49 37 38 ...
##  $ cases_weekly  : int  689552 1039692 1317705 1513773 1290314 1169669 959936 850042 353980 418848 .
##  $ deaths_weekly : int  1666 1518 1943 2153 2005 2295 2071 2133 1338 1330 ...
##  $ country       : Factor w/ 219 levels "Afghanistan",..: 69 69 69 69 69 69 69 69 70 70 ...
##  $ popData2019   : num  4.53e+08 4.53e+08 4.53e+08 4.53e+08 4.53e+08 ...
##  $ continentExp  : Factor w/ 5 levels "Africa","America",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ pc_cases      : num  152 229 291 334 285 ...
##  $ pc_deaths     : num  0.368 0.335 0.429 0.475 0.443 ...
##  $ Land_Area_sqkm: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ HealthExp     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ GDPpc         : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Obese_Pcent   : num  17.2 17.2 17.2 17.2 17.2 17.2 17.2 17.2 17.2 17.2 ...
##  $ Over_65s      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Diabetis      : num  NA NA NA NA NA NA NA NA NA NA ...
```

```
summary(data)
```

```
##   countryCode        geoId           dates              year_week
##         :  354   AE     :   59   Min.   :2019-12-30   2021-01: 219
##  AFG    :   59   AF     :   59   1st Qu.:2020-05-18   2021-02: 219
##  ARE    :   59   AM     :   59   Median :2020-08-17   2021-03: 219
##  ARM    :   59   AT     :   59   Mean   :2020-08-13   2021-04: 219
##  AUS    :   59   AU     :   59   3rd Qu.:2020-11-16   2021-05: 219
##  AUT    :   59   (Other):10508   Max.   :2021-02-08   2021-06: 219
##  (Other):10508   NA's   :  354                        (Other):9843
##   cases_weekly     deaths_weekly              country
##  Min.   : -17182   Min.   : -875.0   Afghanistan    :   59
##  1st Qu.:     10   1st Qu.:    0.0   Africa (total) :   59
##  Median :    231   Median :    3.0   Algeria        :   59
##  Mean   :  21463   Mean   :  268.6   America (total):   59
##  3rd Qu.:   3240   3rd Qu.:   53.0   Armenia        :   59
##  Max.   :2699838   Max.   :23518.0   Asia (total)   :   59
##                                      (Other)        :10803
##   popData2019         continentExp     pc_cases           pc_deaths
##  Min.   :8.090e+02   Africa :2721   Min.   :-181.833   Min.   :-6.72880
##  1st Qu.:1.318e+06   America:2482   1st Qu.:   0.340   1st Qu.: 0.00000
##  Median :8.606e+06   Asia   :2350   Median :   4.268   Median : 0.03471
```

```
##  Mean   :8.263e+07   Europe :3069   Mean   :  43.343   Mean   : 0.76442
##  3rd Qu.:3.237e+07   Oceania: 535   3rd Qu.:  36.483   3rd Qu.: 0.48241
##  Max.   :4.498e+09                  Max.   :2267.668   Max.   :80.14010
##
##  Land_Area_sqkm       HealthExp          GDPpc            Obese_Pcent
##  Min.   :       60   Min.   : 1.597   Min.   :   310.3   Min.   : 2.10
##  1st Qu.:    28500   1st Qu.: 4.401   1st Qu.:  2222.0   1st Qu.: 9.50
##  Median :   143000   Median : 6.301   Median :  7046.2   Median :20.20
##  Mean   :   748649   Mean   : 6.468   Mean   : 17428.1   Mean   :18.51
##  3rd Qu.:   567000   3rd Qu.: 8.136   3rd Qu.: 23090.1   3rd Qu.:24.70
##  Max.   :16400000   Max.   :17.553   Max.   :185835.0   Max.   :52.90
##  NA's   :1817       NA's   :2093     NA's   :2093       NA's   :1618
##     Over_65s          Diabetis
##  Min.   : 1.157   Min.   : 0.000
##  1st Qu.: 3.509   1st Qu.: 5.100
##  Median : 7.301   Median : 6.800
##  Mean   : 9.447   Mean   : 7.875
##  3rd Qu.:15.094   3rd Qu.:10.200
##  Max.   :28.002   Max.   :30.500
##  NA's   :1658     NA's   :806
```

```
names(data)
```

```
##  [1] "countryCode"    "geoId"          "dates"          "year_week"
##  [5] "cases_weekly"   "deaths_weekly"  "country"        "popData2019"
##  [9] "continentExp"   "pc_cases"       "pc_deaths"      "Land_Area_sqkm"
## [13] "HealthExp"      "GDPpc"          "Obese_Pcent"    "Over_65s"
## [17] "Diabetis"
```

Now we need to calculate the Population density.

```
# calculate population density
data <- data %>% XXXX(popdens = XXXX/XXXX)
```

Confirm that the average population density in your dataset is 223.838.

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##    2.115   36.175   86.756  223.838  213.783 8251.542     1817
```

Average data over the sample period

What we now do is to aggregate the weekly cases and deaths data. In the Lecture and the Review and Q&A session we did this over the entire available sample period. Could there be reasons why we may not want to do this over the entire period?

It is in the nature of such a pandemic that it starts in one location and then, initially slowly, spreads through different geographies. The initial spread may well be determined by travel patterns eminating from the country initial effected (here China). In order to reduce the influence of this initial geographic pattern we now decide to aggregate only for data from June 2021 onwards ("2020-06-01" and later).

This was the code we used in the Week2 material to calculate these averages (including all available data).

```
table3 <- data %>% group_by(country) %>% # groups by Country
            summarise(Avg_cases = mean(pc_cases,na.rm = TRUE),
                      Avg_deaths = mean(pc_deaths,na.rm = TRUE),
                      PopDen = mean(popdens))
```

Find a way to adjust this bit of code such that the average calculations are only based on data from "2020-06-01" onwards. What operation should you use in place of `XXXX`? Here is a link to a one page tidyverse cheat sheet. There are 4 major type of operations you can perform in a pipe (`%>%`), `filter`, `arrange`, `mutate`, `summarise\summarize` and (although not on the cheat sheet) `select`. Which one is the one to use?

Also note the following. The `summarise` function is designed to summarise information, e.g. for a particular country, which varies in the country specific sample. However, we not only want to summarise the number of weekly cases and deaths, we also want to have the country information for population density, obesity, diabetis, Over 65s, GDPpc, HealthExp and the countries continent. Below you see, inside the summarise function terms like `PopDen = first(popdens)`. This selects the first `popdens` observation for a particular country. As all these variables do not vary through our sample this little trick delivers exactly what we want.

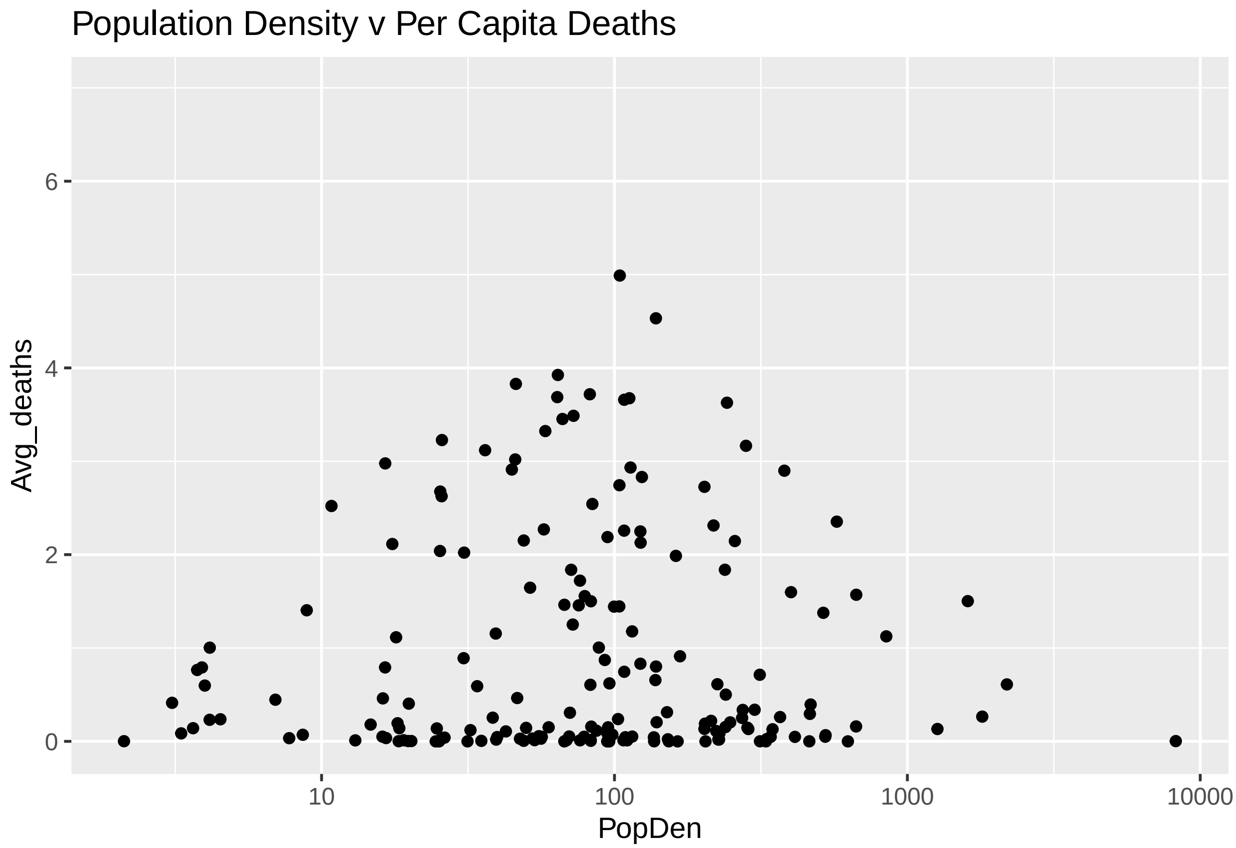
```
table3 <- data %>% XXXX %>%
            group_by(country) %>% # groups by Country
            summarise(Avg_cases = mean(pc_cases,na.rm = TRUE),
                      Avg_deaths = mean(pc_deaths,na.rm = TRUE),
                      PopDen = first(popdens),
                      Obese = first(Obese_Pcent),
                      Diabetis = first(Diabetis),
                      Over_65s = first(Over_65s),
                      GDPpc = first(GDPpc)/1000,  # calculate GDPpc in $1000s
                      HealthExp = first(HealthExp),
                      Continent = first(continentExp))
```

After you selected, check `head(table3)` to confirm that you got the same result.

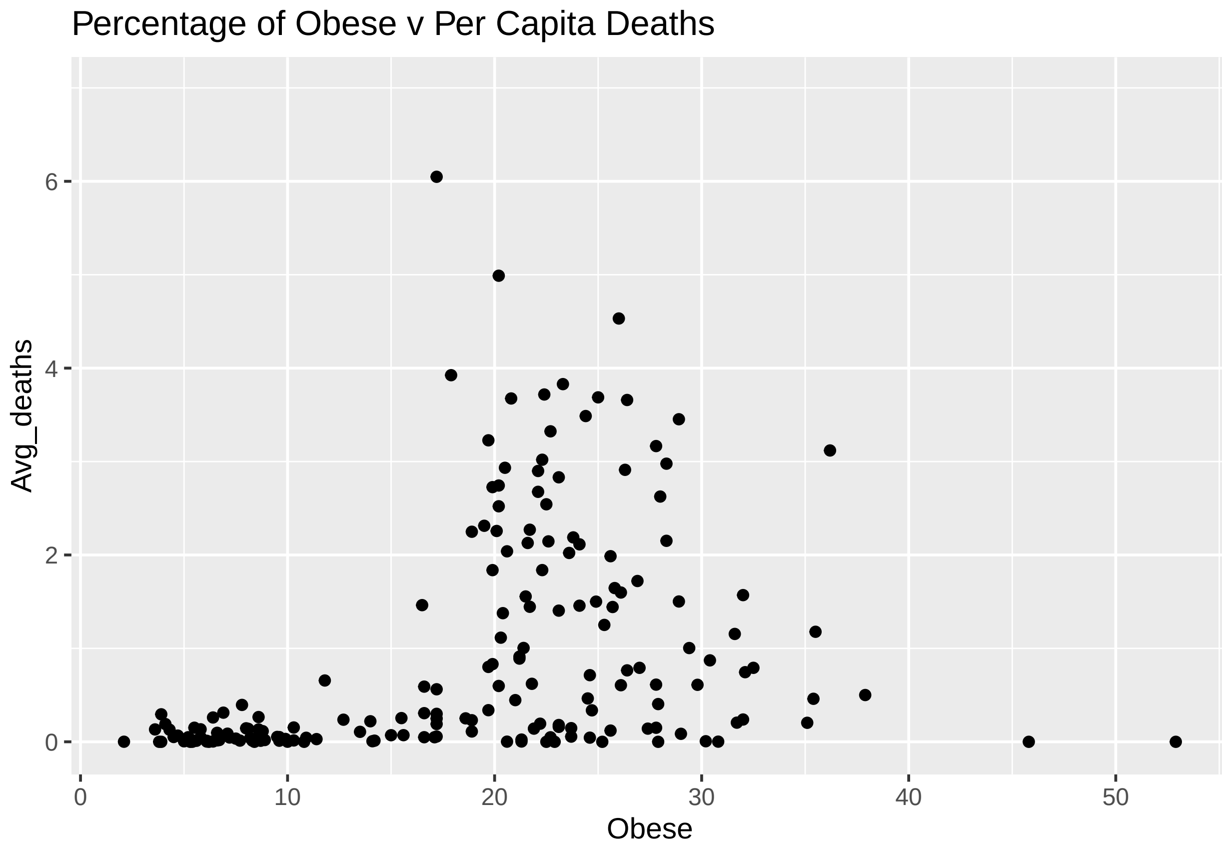
```
## # A tibble: 6 x 10
##   country   Avg_cases Avg_deaths PopDen Obese Diabetis Over_65s  GDPpc HealthExp
##   <fct>         <dbl>      <dbl>  <dbl> <dbl>    <dbl>    <dbl>  <dbl>     <dbl>
## 1 Afghanis~      2.80      0.151   59.6   5.5      9.2     2.62  0.530      9.40
## 2 Africa (~      7.28      0.190   NA    17.2     NA      NA    NA         NA
## 3 Albania       87.3       1.45   104.   21.7      9      14.2   5.22       5.26
## 4 Algeria        6.24      0.141   18.4  27.4      6.7     6.55  4.11       6.22
## 5 America ~    122.        0.249   NA    17.2     NA      NA    NA         NA
## 6 Andorra      347.        1.99   162.   25.6      7.7    NA    42.1        6.71
## # ... with 1 more variable: Continent <fct>
```

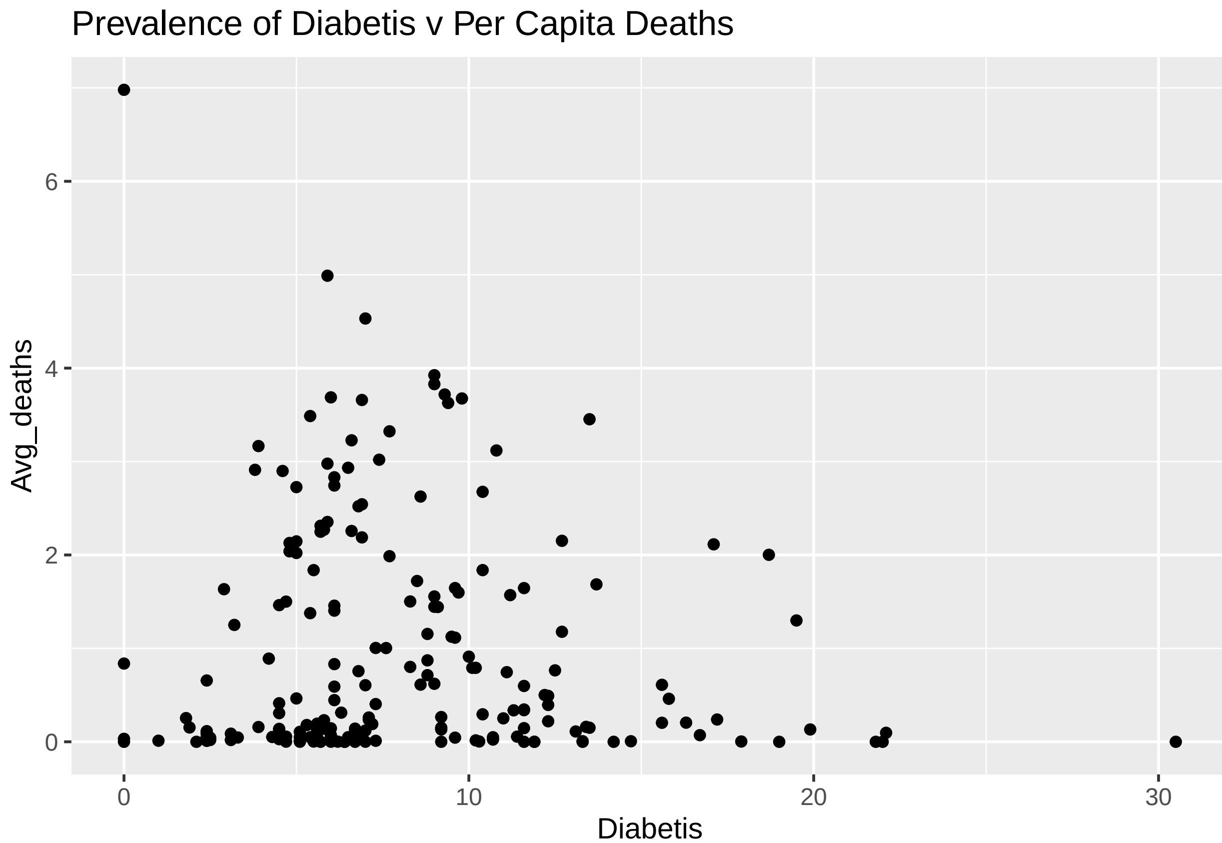
Let's create a few plots which show the average death numbers against some of our country specific information.

```
ggplot(table3,aes(PopDen,Avg_deaths)) +
  geom_point() +
  scale_x_log10() +
  ggtitle("Population Density v Per Capita Deaths")
```



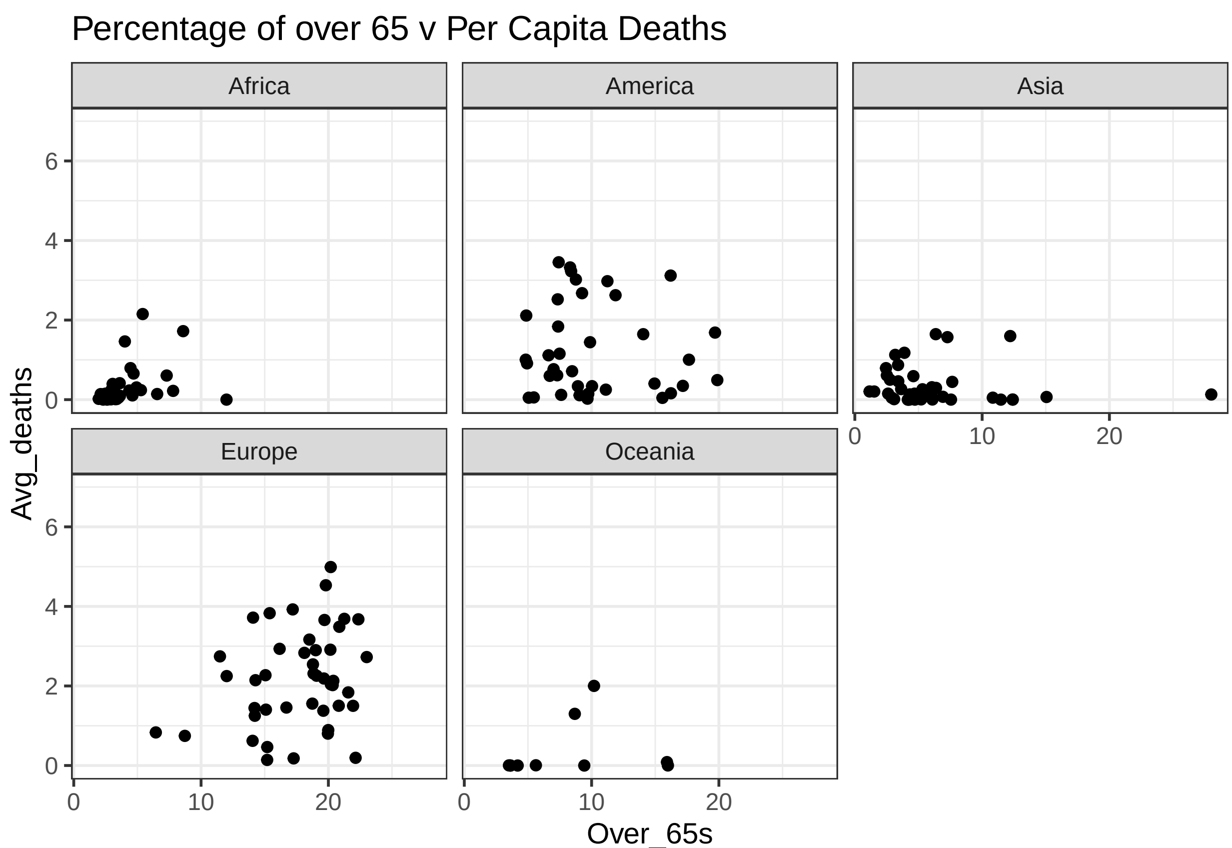
Now replicate the following graphs.


Percentage of Obese v Per Capita Deaths
Avg_deaths
0
2
4
6
0
10
20
30
40
50
Obese



Let's also create plots of deaths against the proportion of over 65s, but this time we want to split the graph according to continents.

```
ggplot(table3,aes(Over_65s,Avg_deaths)) +
  geom_point() +
  facet_wrap(~ Continent) +  # this is where the magic happens!
  theme_bw() +
  ggtitle("Percentage of over 65 v Per Capita Deaths")
```



Nice, right?! Check out the GGplot cheat sheet for more tricks and illustrations of this packages' capabilities.

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.

```
table4 <- table3 %>%
              group_by(Continent) %>%
              summarise(CAvg_cases = mean(Avg_cases, na.rm = TRUE),
                        CAvg_deaths = mean(Avg_deaths, na.rm = TRUE),
                        n = n()) %>% print()
```

```
## # A tibble: 5 x 4
##   Continent CAvg_cases CAvg_deaths     n
##   <fct>          <dbl>       <dbl> <int>
## 1 Africa          10.9       0.211    56
## 2 America         56.7       0.977    50
## 3 Asia            34.6       0.334    43
## 4 Europe         126.        2.05     56
## 5 Oceania         22.3       0.675    14
```

Let's see whether we find the continental averages to be statistically significantly different. Say we compare the `avg_deaths` in America and Asia. So test the null hypothesis that $H_0 : \mu_{AS} = \mu_{AM}$ (or $H_0 : \mu_{AS} - \mu_{AM} = 0$)

against the alternative hypothesis that $H_A : \mu_{AS} \neq \mu_{AM}$, where μ represents the average death rate of countries in the respective continent over the sample period (here June onwards).

```
test_data_AS <- table3 %>%
  filter(Continent == "Asia")      # pick Asian data

test_data_AM <- table3 %>%
  filter(Continent == "America")      # pick European data

t.test(test_data_AS$Avg_deaths,test_data_AM$Avg_deaths, mu=0)  # testing that mu = 0
```

```
##
##  Welch Two Sample t-test
##
## data:  test_data_AS$Avg_deaths and test_data_AM$Avg_deaths
## t = -3.778, df = 67.911, p-value = 0.0003354
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.9833444 -0.3035944
## sample estimates:
## mean of x mean of y
## 0.3335022 0.9769716
```

The difference in the averages is 0.3335 - 0.9770 = -0.6435 (less than 1 in 100,000 population). We get a t-test ststiatic of almost -4. If in truth the two means were the same then we should expect the test statistic to be around 0. Is -4 far enough away from 0 for us to conclude that we should stop supporting the null hypothesis? The value of the t-test is almost -4. Is that big. If H_0 was correct (same average death rates in Ameraica and Asia) then we should on average expect the t-test to come out around a value of 0. So -4 is clearly not 0, but is it so far away from 0 that we should reject H_0?

The answer is yes and the p-value does tell us that it is. The p-value is 0.00034 or 0.034%. This means that if the H_0 was correct, the probability of getting a difference of -0.6435 (per 100,000 population) or a more extreme difference is 0.034%. We judge this probability to be too small for us to coninue 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 case rate in Europe is larger by more than 50 per 100,000 population than the case rate in America.

Here our H_0 is $H_0 : \mu_{EU} = \mu_{AM} + 50$ (or $\mu_{EU} - \mu_{AM} = 50$) and we would test this against an alternative hypothesis of $H_0 : \mu_{EU} > \mu_{AM} + 50$ (or $H_0 : \mu_{EU} - \mu_{AM} > 50$). Here the statement of the politician is represented in the H_A.

```
test_data_EU <- table3 %>%
  filter(Continent == "Europe")      # pick European data

test_data_AM <- table3 %>%
  filter(Continent == "America")      # pick American data

t.test(test_data_EU$Avg_cases,test_data_AM$Avg_cases, mu=50, alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  test_data_EU$Avg_cases and test_data_AM$Avg_cases
## t = 1.5202, df = 101.15, p-value = 0.06579
```

```
## alternative hypothesis: true difference in means is greater than 50
## 95 percent confidence interval:
##  48.24822      Inf
## sample estimates:
## mean of x mean of y
## 125.70717  56.66677
```

Note the following. The parameter `mu` now takes the value 50 as we are hypothesising that the difference in the means in 50 (or larger than that in the H_A). Also, in contrast to the pevious test we now care whether the deviation is less or greater than 50. In this case we wonder whether it is really greater. Hence we use the additional input into the test function, `alternative = "greater"`. (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 (125.70717-56.66677=69.0404) is consistent with the null hypothesis (of the population difference being 50). Here the answer is subtle. The p-value is 0.0658, so the probability of optaining a sample difference as big as 69.0404 (or bigger) is just a little over 5%. Say we set out to perform a test at a 10% significance level, then we would judge a probability of just above 5% to be too small and hence we would reject the null hypothesis. If however we set out to perform a test at a 1% significance level then we would not reject the null hypothesis.

So let's perform another test. An European opposition politicial is lamenting that the European case rate is more than 100 (per 100,000 population) larger than that in Asia. Perform the appropriate hypothesis test.

```
t.test(test_data_XXXX$Avg_cases,test_data_XXXX$Avg_cases, mu=XXXX, alternative = XXXX)
```

```
##
##  Welch Two Sample t-test
##
## data:  test_data_EU$Avg_cases and test_data_AS$Avg_cases
## t = -0.71589, df = 95.807, p-value = 0.7621
## alternative hypothesis: true difference in means is greater than 100
## 95 percent confidence interval:
##  70.46303      Inf
## sample estimates:
## mean of x mean of y
## 125.70717  34.60362
```

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 politician's statement is not supported by the data.

Regression and inference

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.

In the lecture we talked about a base case regression

$Avg\_deaths_i = \alpha + \beta_1 \ GDPpc_i + \beta_2 \ HealthExp_i + u_i$

Let us estimate this again using the average rates calculated on data from June onwards only (hence the results here will be somewhat different to those in the lecture).

```r
mod3 <- lm(Avg_deaths~GDPpc+HealthExp,data=table3)
stargazer(mod3,type = "text")
```

```
##
## ===============================================
##                         Dependent variable:
##                     ---------------------------
##                             Avg_deaths
## -----------------------------------------------
## GDPpc                        0.010***
##                               (0.004)
##
## HealthExp                    0.117***
##                               (0.031)
##
## Constant                      0.040
##                               (0.215)
##
## -----------------------------------------------
## Observations                   176
## R2                            0.136
## Adjusted R2                   0.126
## Residual Std. Error      1.113 (df = 173)
## F Statistic          13.583*** (df = 2; 173)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

We see that, for these data, the `HealthExp` variable remains statistically significant although the `GDPpc` variable is now not statistically significant.

Now add the `Obese`, `Diabetis` and `Over_65s` variables to the regression in order to evaluate whether their inclusion change the implausible negative sign on `HealthExp`.

```r
mod4 <- lm(Avg_deaths~GDPpc+XXXX,data=table3)
stargazer(mod3,mod4,type = "text")
```

```
##
## ===================================================================
##                                   Dependent variable:
##                     -----------------------------------------------
##                                       Avg_deaths
##                               (1)                     (2)
## -------------------------------------------------------------------
## GDPpc                      0.010***                -0.012***
##                             (0.004)                 (0.004)
##
## HealthExp                  0.117***                  0.018
##                             (0.031)                 (0.032)
##
## Obese                                              0.044***
##                                                     (0.010)
##
## Over_65s                                           0.106***
##                                                     (0.014)
##
```

```
## Diabetis                                         -0.033
##                                                  (0.021)
##
## Constant                      0.040             -0.507**
##                              (0.215)            (0.250)
##
## -------------------------------------------------------------------
## Observations                   176                 166
## R2                            0.136               0.446
## Adjusted R2                   0.126               0.429
## Residual Std. Error    1.113 (df = 173)      0.910 (df = 160)
## F Statistic         13.583*** (df = 2; 173) 25.747*** (df = 5; 160)
## ===================================================================
## Note:                                   *p<0.1; **p<0.05; ***p<0.01
```

If you want to perform a hypothesis test say on β_3 (the coefficient on the `Obese` variable), then the usual hypothesis to pose is $H_0 : \beta_3 = 0$ versus $H_A : \beta_3 \neq 0$. It is the p-value to that hypothesis test which is represented by the asteriks next to the estimated coefficient. Let's confirm that. The estimated coefficient to the `Obese` variable is 0.047 and the (***) indicate that the p-value to that test should be less than 0.01.

Here is how you can perform this test manually using the `lht` (stands for Linear Hypothesis Test) function which is written to use regression output (here saved in `mod4`) for hypothesis testing.

```
lht(mod4,"Obese=0")
```

```
## Linear hypothesis test
##
## Hypothesis:
## Obese = 0
##
## Model 1: restricted model
## Model 2: Avg_deaths ~ GDPpc + HealthExp + Obese + Over_65s + Diabetis
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    161 149.25
## 2    160 132.40  1    16.852 20.365 1.234e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There is a lot of information, but the important one is the value displayed under ("Pr(>F)"), that is the p-value. Here it is very small, 0.0000219 (=2.19e-05), and as predicted < 0.01.

Confirm that p-value for $H_0 : \beta_2 = 0$ versus $H_A : \beta_2 \neq 0$ (coefficient on `HealthExp`) is larger than 0.1.

```
## Linear hypothesis test
##
## Hypothesis:
## HealthExp = 0
##
## Model 1: restricted model
## Model 2: Avg_deaths ~ GDPpc + HealthExp + Obese + Over_65s + Diabetis
##
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1    161 132.68
## 2    160 132.40  1   0.27638 0.334 0.5641
```

The use of the `lht` function is that you can test different hypothesis. Say $H_0 : \beta_4 = 0.1$ versus $H_A : \beta_4 \neq 0.1$ (coefficient on `Over_65s`).

```r
lht(mod4,"Over_65s=0.1")
```

```
## Linear hypothesis test
##
## Hypothesis:
## Over_65s = 0.1
##
## Model 1: restricted model
## Model 2: Avg_deaths ~ GDPpc + HealthExp + Obese + Over_65s + Diabetis
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    161 132.53
## 2    160 132.40  1   0.12563 0.1518 0.6973
```

So, that null hypothesis cannot be rejected.

Even more so, you can use this function to test multiple hypotheses. Say you want to test whether the inclusion of the additional three variables (in `mod4` as opposed to `mod3`) is relevant. If it wasn't then the following null hypothesis should be correct: $H_0 : \beta_3 = \beta_4 = \beta_5 = 0$. We call this a multiple hypothesis.

Use the help function (`?lht`) or search for advice () on how to use the `lht` function to test this hypothesis.

```
## Linear hypothesis test
##
## Hypothesis:
## Obese = 0
## Diabetis = 0
## Over_65s = 0
##
## Model 1: restricted model
## Model 2: Avg_deaths ~ GDPpc + HealthExp + Obese + Over_65s + Diabetis
##
##   Res.Df    RSS Df Sum of Sq  F    Pr(>F)
## 1    163 204.39
## 2    160 132.40  3    71.992 29 4.993e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The hypothesis that none of the three variables is relevant is clearly rejected.

The techniques you covered in this computer lab are absolutly fundamental to the remainder of this unit, so please ensure that you have not rushed over the material.