

# 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
library(ISOweek)   # Weeks are provided in the ISO weeks format
library(countrycode) # to translate country codes
```

## Data Import

Import the data from the “StaticECDCdata\_18Feb22.csv” file. Make sure the file 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 K2.

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

```
## 'data.frame':  22679 obs. of  14 variables:
## $ X                : int  1 2 3 4 5 6 7 8 9 10 ...
## $ country          : Factor w/ 226 levels "Afghanistan",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ country_code      : Factor w/ 220 levels "ABW","AFG","AGO",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ continent         : Factor w/ 5 levels "Africa","America",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ popData2019       : num  38928341 38928341 38928341 38928341 38928341 ...
## $ dates             : Factor w/ 111 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 0 1 3 ...
## $ deaths_weekly     : int  0 0 0 0 0 0 0 0 0 0 0 ...
## $ cases_weekly_cumulative : int  0 0 0 0 0 0 0 0 0 1 4 ...
## $ deaths_weekly_cumulative: int  0 0 0 0 0 0 0 0 0 0 0 ...
## $ cases_14_day       : num  NA 0 0 0 0 0 ...
## $ deaths_14_day      : num  NA 0 0 0 0 0 0 0 0 0 0 ...
```

```
## $ source_cases      : Factor w/ 3 levels "Epidemic intelligence national data",...: 1 1 1 1 1
## $ source_deaths     : Factor w/ 3 levels "Epidemic intelligence national data",...: 1 1 1 1 1
```

The addition of `stringsAsFactors = TRUE` ensures that character variables are automatically translated into categorical (factor) variables. This will save us some work.

You got it right if the output from `str(data)` looks like the above. The dataset also contains data on entire continents but we will remove these datalies.

```
# remove continent data
data <- data %>% filter(!is.na(country_code))
```

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

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

The date information is currently captured in the `dates` variable. Variable `dates` is currently a factor (factor) variable (originally a `chr` but as we imported we converted these all to factor), but we want R to know that each string represents a date. What we do here was described in detail in the code to Lecture 2.

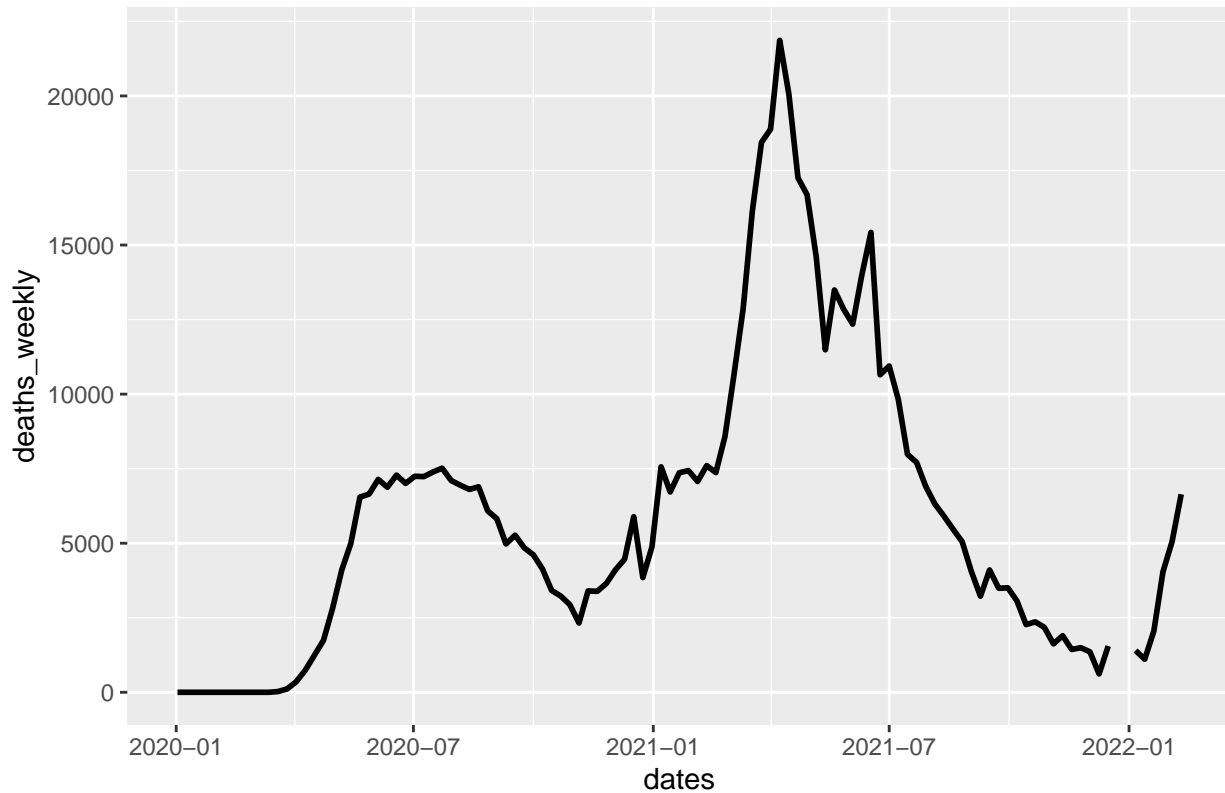
```
data <- data %>%
  separate(dates, c("year", "week"), "-") %>%
  mutate(dates = ISOweek2date(paste0(year, "-W", week, "-4")))
```

## Plotting data as time-series

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

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

Covid-19 weekly deaths, Brazil



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.

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

##	country	popData2019
## 1	China	1439323774
## 2	India	1380004385
## 3	United States Of America	331002647
## 4	Indonesia	273523621
## 5	Pakistan	220892331
## 6	Brazil	212559409
## 7	Nigeria	206139587
## 8	Bangladesh	164689383
## 9	Russia	145934460
## 10	Mexico	128932753
## 11	Japan	126476458
## 12	Ethiopia	114963583
## 13	Philippines	109581085
## 14	Egypt	102334403

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?

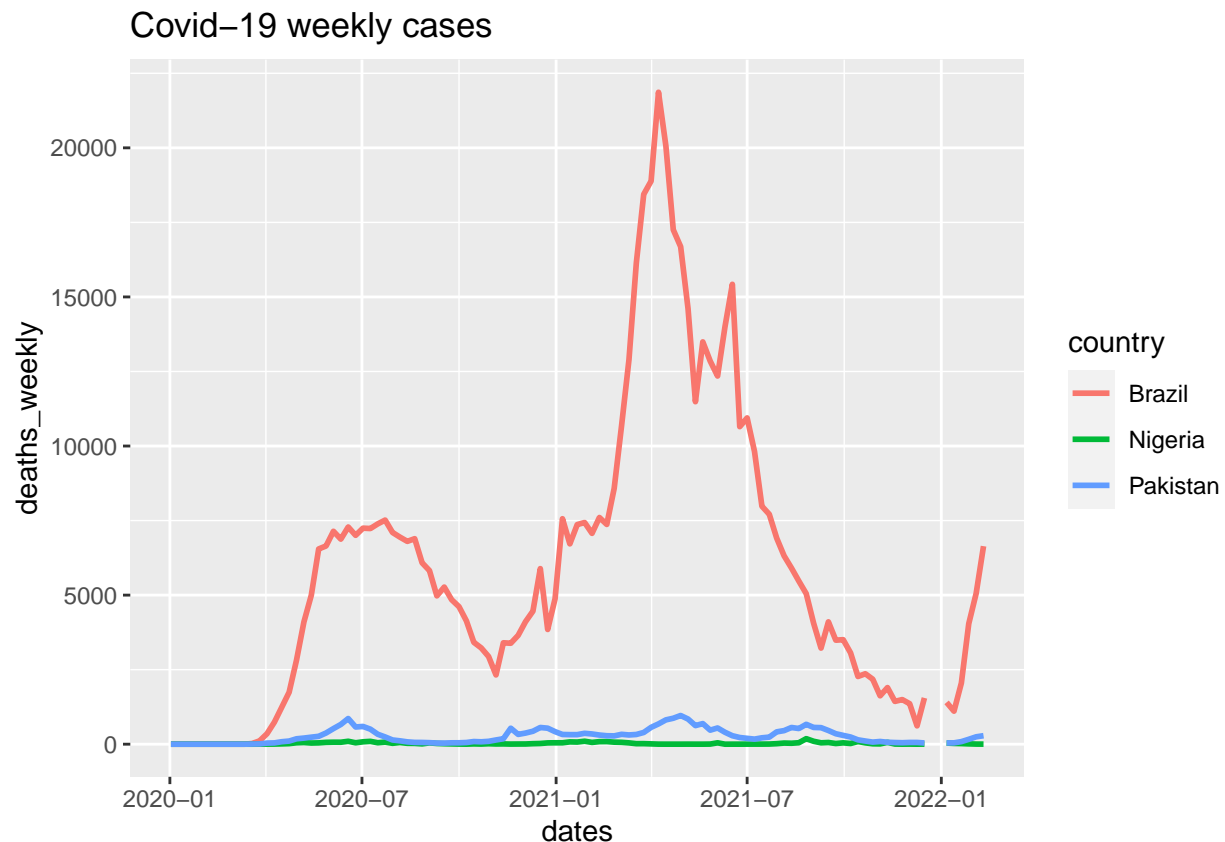
For instance, what does the following do?

```
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. Create a new variable `sel_countries` which you use to select data for these three countries and then plot a line graph with weekly case numbers for these three countries. You may want to check out previous code (Week 2 Empirical Work Script) to see what needs doing.

```
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 (`country_code`) 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.

```
countryInd <- read_csv("CountryIndicators.csv", na = "#N/A")
countryInd <- countryInd %>% select(-country)
obesity <- read_csv("Obesity.csv") # Adds obesity and diabetis country
obesity <- obesity %>% select(-country)
over65p <- read_excel("Over 65s 2.xlsx")
```

In “CountryIndicators.csv” and “Obesity.csv” we can find a 2-digit `geoID` 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. We use country codes to match the data precisely to avoid any problems from slightly different spellings.

The data coming from “Over 65s 2.xlsx” contain a three digit country code, as does the `data` file at this stage. For instance the 3 digit code for Afghanistan is AFG and the two digit code is AF. So, we have a mixture of codes and we need something that translates one into the other so that we can merge these data all into one dataframe.

As discussed in the Week 2 empirical workscript there is a tool just for that job, the `countrycode` library. We add a two digit country code to `data` and call it `geoID`.

```
data$geoID <- countrycode(data$country_code, origin = "iso3c", destination = "iso2c")
```

Now we are in a position to merge the datasets.

```
# 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)
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 (`country_code`). 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.

```
data <- merge(data, over65p, by.x = "country_code", by.y = "countryCode", 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':   22017 obs. of  25 variables:
## $ country_code      : Factor w/ 220 levels "ABW","AFG","AGO",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ geoID              : chr  "AW" "AW" "AW" "AW" ...
## $ X                  : int  1245 1246 1263 1261 1262 1244 1264 1186 1242 1256 ...
## $ country            : Factor w/ 226 levels "Afghanistan",...: 13 13 13 13 13 13 13 13 13 13 ..
## $ continent          : Factor w/ 5 levels "Africa","America",...: 2 2 2 2 2 2 2 2 2 2 ...
## $ popData2019        : num  106766 106766 106766 106766 106766 ...
## $ year                : chr  "2021" "2021" "2022" "2021" ...
## $ week                : chr  "35" "36" "02" "52" ...
## $ cases_weekly       : int  353 231 3285 NA 5372 448 1728 8 877 121 ...
```

```
## $ deaths_weekly      : int  12 4 2 NA 1 11 9 0 10 1 ...
## $ cases_weekly_cumulative : int 14861 15092 30677 22020 27392 14508 32405 113 13406 16210 ...
## $ deaths_weekly_cumulative: int 151 155 184 181 182 139 193 3 121 173 ...
## $ cases_14_day       : num  750 547 8108 4844 NA ...
## $ deaths_14_day      : num  215.42 149.86 28.1 9.37 NA ...
## $ source_cases       : Factor w/ 3 levels "Epidemic intelligence national data",...: 1 1 1 1 1
## $ source_deaths      : Factor w/ 3 levels "Epidemic intelligence national data",...: 1 1 1 1 1
## $ pc_cases           : num  331 216 3077 NA 5032 ...
## $ pc_deaths          : num  11.24 3.747 1.873 NA 0.937 ...
## $ dates              : Date, format: "2021-09-02" "2021-09-09" ...
## $ 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  NA NA NA NA NA NA NA NA NA NA ...
## $ Over_65s           : num  14.1 14.1 14.1 14.1 14.1 ...
## $ Diabetis           : num  11.6 11.6 11.6 11.6 11.6 11.6 11.6 11.6 11.6 11.6 ...
```

```
summary(data)
```

```
## country_code      geoID      X      country
## AUT : 111 Length:22017 Min. : 1 Austria : 111
## BEL : 111 Class :character 1st Qu.: 5835 Belgium : 111
## BGR : 111 Mode :character Median :11561 Bulgaria: 111
## CYP : 111 Mean :11520 Croatia : 111
## CZE : 111 3rd Qu.:17175 Cyprus : 111
## DEU : 111 Max. :22679 Czechia : 111
## (Other):21351 (Other) :21351
## continent      popData2019      year      week
## Africa :5467 Min. :8.090e+02 Length:22017 Length:22017
## America:4922 1st Qu.:8.880e+05 Class :character Class :character
## Asia :4433 Median :6.871e+06 Mode :character Mode :character
## Europe :5928 Mean :3.790e+07
## Oceania:1267 3rd Qu.:2.638e+07
## Max. :1.439e+09
##
## cases_weekly      deaths_weekly      cases_weekly_cumulative
## Min. : -3864 Min. : -439.0 Min. : 0
## 1st Qu.: 30 1st Qu.: 0.0 1st Qu.: 1234
## Median : 489 Median : 6.0 Median : 16839
## Mean : 18417 Mean : 264.1 Mean : 584670
## 3rd Qu.: 5027 3rd Qu.: 69.0 3rd Qu.: 199083
## Max. :6178670 Max. :29330.0 Max. :77739880
## NA's :188 NA's :188
## deaths_weekly_cumulative cases_14_day      deaths_14_day
## Min. : 0 Min. : -1.056 Min. : -67.135
## 1st Qu.: 18 1st Qu.: 2.229 1st Qu.: 0.000
## Median : 255 Median : 24.120 Median : 2.431
## Mean : 12193 Mean : 212.259 Mean : 20.852
## 3rd Qu.: 3452 3rd Qu.: 161.606 3rd Qu.: 19.898
## Max. :919696 Max. :20394.966 Max. :1395.031
## NA's :409 NA's :409
##
## source_cases
## Epidemic intelligence national data :19020
## Epidemic intelligence national data and TESSy COVID-19: 0
## TESSy COVID-19 : 2997
```

```
##
##
##
##
## source_deaths
## Epidemic intelligence national data :19575
## Epidemic intelligence national data and TESSy COVID-19: 0
## TESSy COVID-19 : 2442
##
##
##
##
## pc_cases pc_deaths dates Land_Area_sqkm
## Min. : -21.901 Min. : -6.7288 Min. : 2020-01-02 Min. : 60
## 1st Qu.: 0.918 1st Qu.: 0.0000 1st Qu.: 2020-08-20 1st Qu.: 28100
## Median : 11.317 Median : 0.0986 Median : 2021-02-18 Median : 130000
## Mean : 109.218 Mean : 1.0413 Mean : 2021-02-14 Mean : 717185
## 3rd Qu.: 79.563 3rd Qu.: 0.9462 3rd Qu.: 2021-08-12 3rd Qu.: 548000
## Max. : 10353.013 Max. : 80.1401 Max. : 2022-02-10 Max. : 16400000
## NA's :188 NA's :188 NA's :3272
## HealthExp GDPpc Obese_Pcent Over_65s
## Min. : 1.597 Min. : 310.3 Min. : 2.10 Min. : 1.157
## 1st Qu.: 4.401 1st Qu.: 2201.6 1st Qu.: 8.90 1st Qu.: 3.509
## Median : 6.301 Median : 6372.6 Median : 20.60 Median : 7.273
## Mean : 6.478 Mean : 16783.1 Mean : 18.81 Mean : 9.319
## 3rd Qu.: 8.136 3rd Qu.: 19441.1 3rd Qu.: 25.00 3rd Qu.: 15.061
## Max. : 17.553 Max. : 185835.0 Max. : 55.90 Max. : 28.002
## NA's :3865 NA's :3865 NA's :3384 NA's :2890
## Diabetis
## Min. : 0.000
## 1st Qu.: 5.100
## Median : 6.800
## Mean : 7.917
## 3rd Qu.: 10.200
## Max. : 30.500
## NA's :963
```

```
names(data)
```

```
## [1] "country_code" "geoID"
## [3] "X" "country"
## [5] "continent" "popData2019"
## [7] "year" "week"
## [9] "cases_weekly" "deaths_weekly"
## [11] "cases_weekly_cumulative" "deaths_weekly_cumulative"
## [13] "cases_14_day" "deaths_14_day"
## [15] "source_cases" "source_deaths"
## [17] "pc_cases" "pc_deaths"
## [19] "dates" "Land_Area_sqkm"
## [21] "HealthExp" "GDPpc"
## [23] "Obese_Pcent" "Over_65s"
## [25] "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 219.1622949.

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.    NA's
##      2.115   36.175   86.756   219.162  213.783  8251.542   3272
```

## 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 emanating 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.

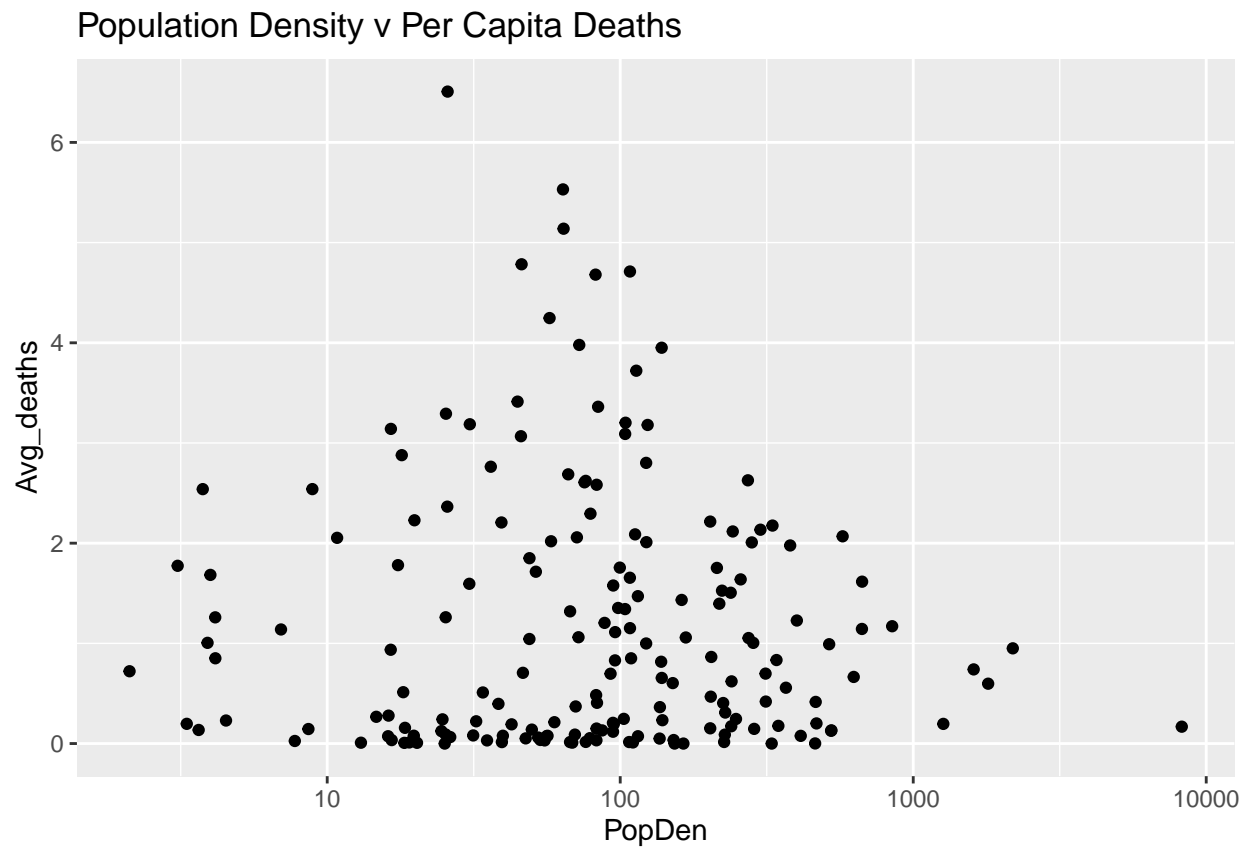
```
## # A tibble: 6 x 10
##   country Avg_cases Avg_deaths PopDen Obese Diabetis Over_65s GDPpc HealthExp
##   <fct>      <dbl>      <dbl> <dbl> <dbl>      <dbl>      <dbl> <dbl>      <dbl>
## 1 Afghanis~    4.59      0.213    59.6   5.5        9.2        2.62  0.530    9.40
```



```
## 2 Albania      106.      1.34    104.    21.7      9      14.2    5.22      5.26
## 3 Algeria       6.52     0.157    18.4    27.4      6.7      6.55    4.11      6.22
## 4 American~     5.07      0       NA     NA       0       NA     NA       NA
## 5 Andorra      499.      1.43    162.    25.6      7.7     NA     42.1     6.71
## 6 Angola        2.85     0.0650   26.3    8.2      4.5      2.20    3.44     2.55
## # ... with 1 more variable: Continent <fct>
```

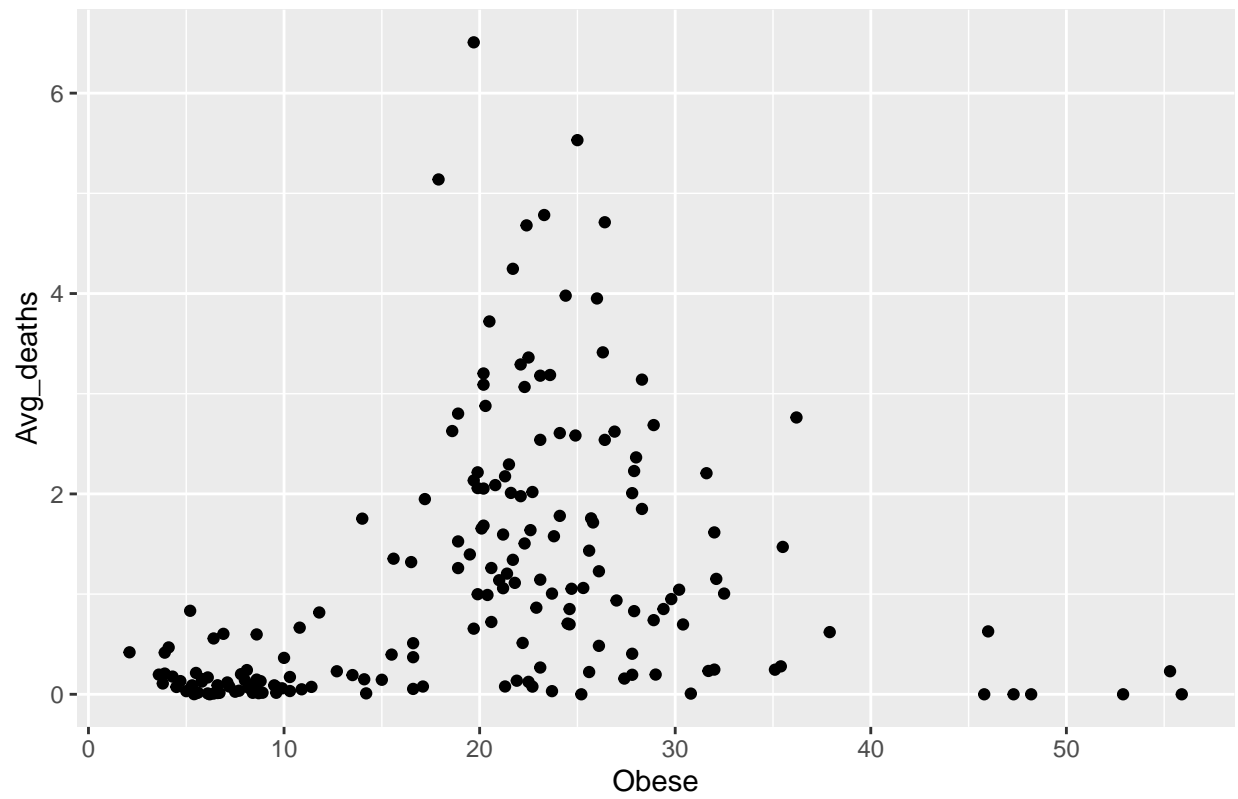
Let's create a few plots which show the average death numbers against some of our country specific information.

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

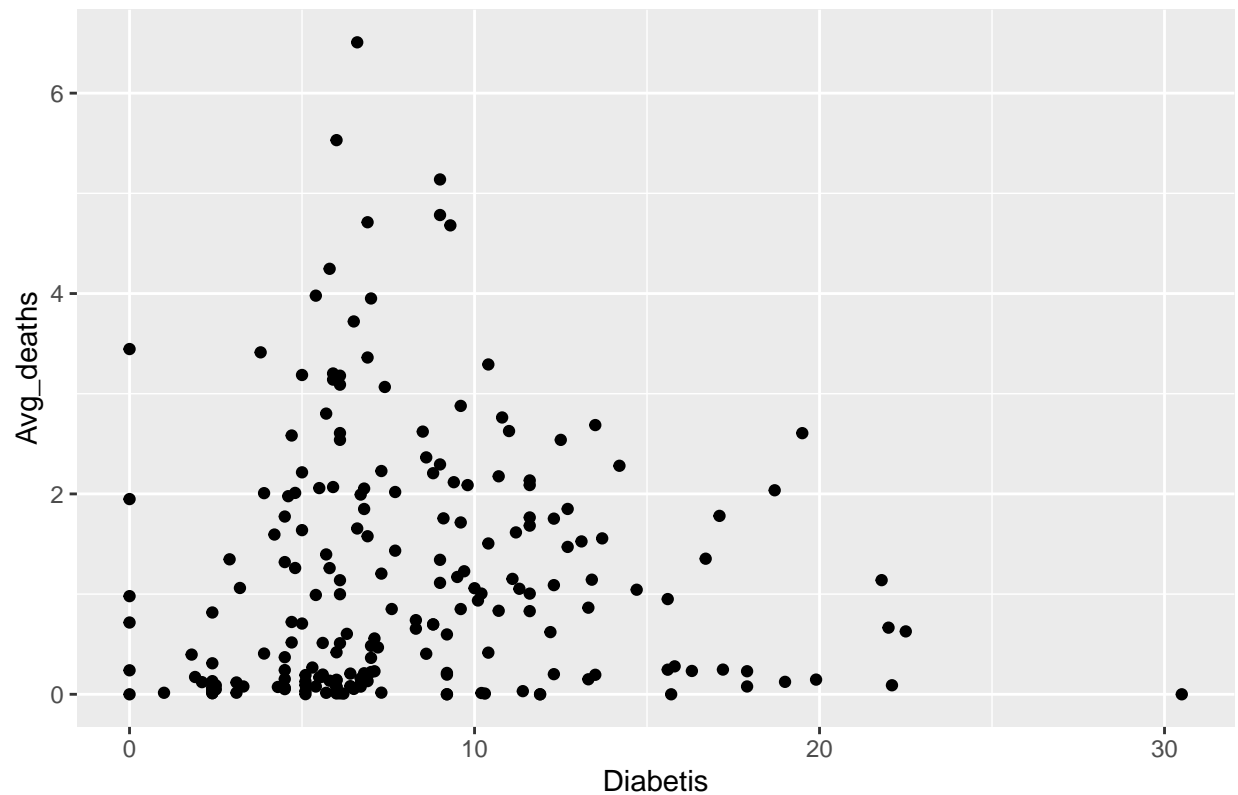


Now replicate the following graphs.

Percentage of Obese v Per Capita Deaths



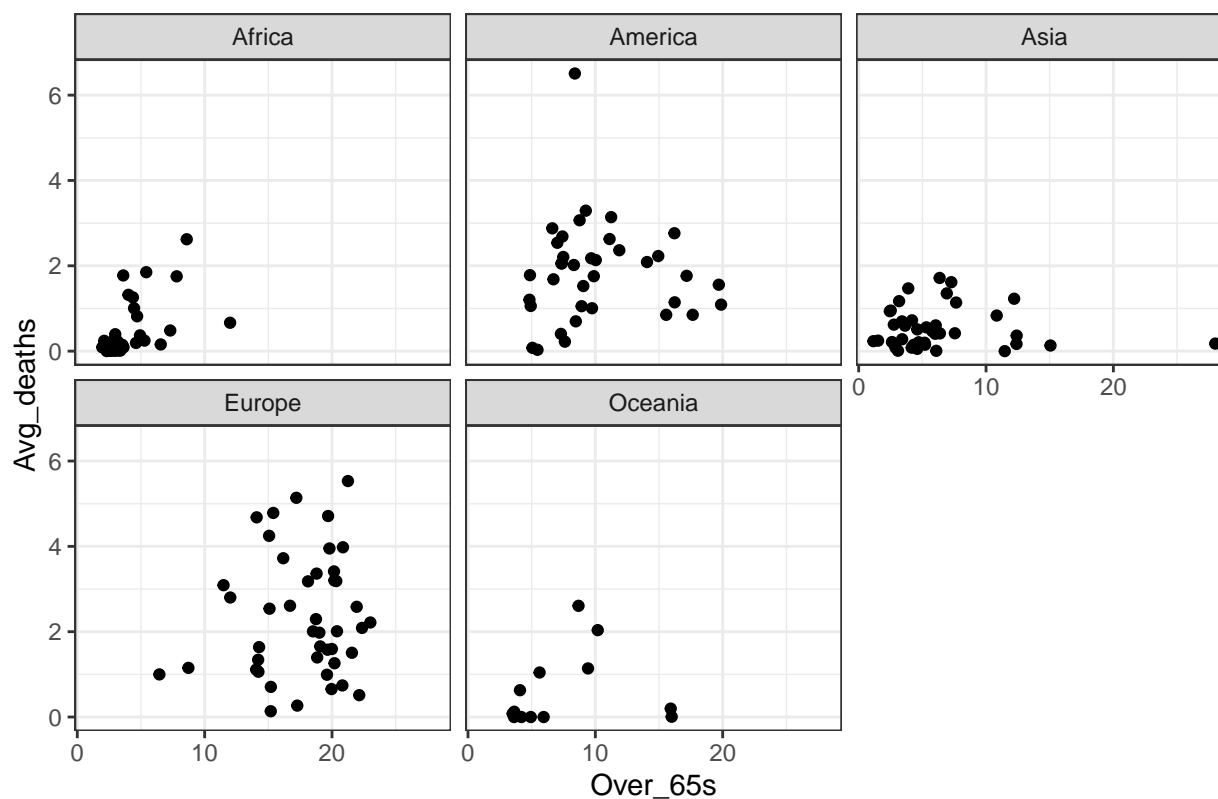
Prevalence of Diabetes v Per Capita Deaths



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")
```

## 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      25.2        0.329    55
## 2 America    133.         1.63     49
## 3 Asia       77.3         0.512    42
## 4 Europe    263.         2.15     55
## 5 Oceania   122.         0.487     19
```

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  $\mu$  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 = -6.1998, df = 65.827, p-value = 4.202e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -1.4778610 -0.7578499
## sample estimates:
## mean of x mean of y
## 0.5122709 1.6301263
```

The difference in the averages is  $0.5122709 - 1.6301263 = -1.1179$  (a bit more than 1 in 100,000 population). We get a t-test statistic of around -6. If in truth the two means were the same then we should expect the test statistic to be around 0. Is -6 far enough away from 0 for us to conclude that we should stop supporting the null hypothesis? The value of the t-test is approximately -6.2. Is that big. If  $H_0$  was correct (same average death rates in America and Asia) then we should on average expect the t-test to come out around a value of 0. So -6.2 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  $4.202e-08$  or  $0.00000004202$  or  $0.000004\%$ . This means that if the  $H_0$  was correct, the probability of getting a difference of -1.1179 (per 100,000 population) or a more extreme difference is  $0.000004\%$ . 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 case rate in Europe is larger by more than 100 per 100,000 population than the case rate in America.

Here our  $H_0$  is  $H_0 : \mu_{EU} = \mu_{AM} + 100$  (or  $\mu_{EU} - \mu_{AM} = 100$ ) and we would test this against an alternative hypothesis of  $H_A : \mu_{EU} > \mu_{AM} + 100$  (or  $H_A : \mu_{EU} - \mu_{AM} > 100$ ). 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=100, alternative = "greater")

##
## Welch Two Sample t-test
##
## data: test_data_EU$Avg_cases and test_data_AM$Avg_cases
## t = 1.4819, df = 93.809, p-value = 0.07086
```

```
## alternative hypothesis: true difference in means is greater than 100
## 95 percent confidence interval:
##  96.37232      Inf
## sample estimates:
## mean of x mean of y
##  263.1109  133.1460
```

Note the following. The parameter `mu` now takes the value 100 as we are hypothesising that the difference in the means is 100 (or larger than that in the  $H_A$ ). Also, in contrast to the previous test we now care whether the deviation is less or greater than 100. 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 ( $263.1109 - 133.1460 = 129.9649$ ) is consistent with the null hypothesis (of the population difference being 100 - or smaller). Here the answer is subtle. The p-value is 0.07086, so the probability of obtaining a sample difference as big as 129.9649 (or bigger) is just a little over 5%, if in truth  $H_0$  was true. 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 politician is lamenting that the European case rate is more than 200 (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 = -0.62074, df = 94.421, p-value = 0.7319
## alternative hypothesis: true difference in means is greater than 200
## 95 percent confidence interval:
##  147.836      Inf
## sample estimates:
## mean of x mean of y
##  263.11095  77.30111
```

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. (If you are working on your own computer you will first have to install this package. On university computers it should be pre-installed.)

```
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 2020 onwards only (hence the results here will be somewhat different to those in the lecture).

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

```
##
## =====
##                               Dependent variable:
##                               -----
##                               Avg_deaths
## -----
## GDPpc                        0.004
##                               (0.004)
##
## HealthExp                    0.086**
##                               (0.036)
##
## Constant                    0.579**
##                               (0.247)
##
## -----
## Observations                 176
## R2                          0.044
## Adjusted R2                 0.032
## Residual Std. Error        1.280 (df = 173)
## F Statistic                 3.938** (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 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.

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

```
##
## =====
##                               Dependent variable:
##                               -----
##                               Avg_deaths
##                               (1)                (2)
## -----
## GDPpc                        0.004                -0.024***
##                               (0.004)                (0.005)
##
## HealthExp                    0.086**                -0.040
##                               (0.036)                (0.035)
##
## Obese                       0.051***
##                               (0.011)
##
## Over_65s                     0.138***
```

```
## (0.016)
##
## Diabetis -0.011
## (0.022)
##
## Constant 0.579** -0.315
## (0.247) (0.273)
##
## -----
## Observations 176 166
## R2 0.044 0.458
## Adjusted R2 0.032 0.441
## Residual Std. Error 1.280 (df = 173) 0.992 (df = 160)
## F Statistic 3.938** (df = 2; 173) 27.000*** (df = 5; 160)
## =====
## Note: *p<0.1; **p<0.05; ***p<0.01
```

If you want to perform a hypothesis test say on  $\beta_3$  (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 0.051 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     161 180.41
## 2     160 157.55  1    22.863 23.219 3.333e-06 ***
## ---
## 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, 0.00000333 (=3.333e-06), and as predicted < 0.01. In fact it is even smaller than 0.001.

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     161 158.86
```



```
## 2      160 157.55  1      1.3096 1.33 0.2505
```

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`).

```
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     161 163.46
## 2     160 157.55  1      5.9163 6.0084 0.01531 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, that null hypothesis can be rejected at a 5% but not at a 1% level.

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     163 275.47
## 2     160 157.55  3     117.92 39.92 < 2.2e-16 ***
## ---
## 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 absolutely fundamental to the remainder of this unit, so please ensure that you have not rushed over the material.