Vehicle Report

Problem

- We have a data set which evaluates different cars and we need to find the best classifier to classify and predict the outputs efficiently

Dataset

- The dataset is based on a car evaluation database which has seven attributes:
  ---buying (the price of the car),
  -maint (price of maintenance),
  -doors (number of doors),
  -persons,
  -lug_boot (size of the luggage boot),
  -safety (estimated safety of the car)
  -class (reperesents wheter a class is un/acceptable, good or very good)

Data cleaning

1. Renaming of attributes to make more sense visually
2. Used the summary function to view a visual summary of what the data consists of
3. Checked for any NA's by using the is.na function
4. Converted all to numeric for corrplot usage
5. Changed the class attribute back to a factor

Metrics for both classifiers

- I created a method to output the Accuracy, Precision, Recall, FAR and FRR metrics by simply passing the confusion matrix result.
- The function and the metrics for each classifier can see below. (For the purpose of this PowerPoint I am simply showing one configuration metric results of each classifier, however in R you may find that I calculated the metrics for each configuration of each classifier.)

```r
CreateMetrics <- function(table_matrix, confMat) {
  #accuracy
  accuracyClass1 <- confMat$byClass[1,11]
  accuracyClass2 <- confMat$byClass[2,11]
  accuracyClass3 <- confMat$byClass[3,11]
  accuracyClass4 <- confMat$byClass[4,11]

  #FAR: FP/(FP + TN) = 1 - specificity
  #so im getting the specifity from conf matrix and doing 1-specifiy for each class
  farClass1 <- 1 - confMat$byClass[1,2]
  farClass2 <- 1 - confMat$byClass[2,2]
  farClass3 <- 1 - confMat$byClass[3,2]
  farClass4 <- 1 - confMat$byClass[4,2]

  #FRR: FRR = FNR = FN/(TP+FN) = 1 - recall
  #same as above explanation
  frrClass1 <- 1 - confMat$byClass[1,6] #
  frrClass2 <- 1 - confMat$byClass[2,6] #
  frrClass3 <- 1 - confMat$byClass[3,6] #
  frrClass4 <- 1 - confMat$byClass[4,6] #

  #precision
  precisionClass1 <- confMat$byClass[1,5] #
  precisionClass2 <- confMat$byClass[2,5] #
  precisionClass3 <- confMat$byClass[3,5] #
  precisionClass4 <- confMat$byClass[4,5] #

  #recal
  recallClass1 <- confMat$byClass[1,6] #
  recallClass2 <- confMat$byClass[2,6] #
  recallClass3 <- confMat$byClass[3,6] #
  recallClass4 <- confMat$byClass[4,6] #

  resultClass1 <- data.frame(accuracyClass1,farClass1,frrClass1, precisionClass1, recallClass1)
  resultClass2 <- data.frame(accuracyClass2, farClass2,frrClass2, precisionClass2, recallClass2)
  resultClass3 <- data.frame(accuracyClass3, farClass3,frrClass3, precisionClass3, recallClass3)
  resultClass4 <- data.frame(accuracyClass4,farClass4,frrClass4, precisionClass4, recallClass4)
  my_list <- list("class 1 metrics " = resultClass1,
                  "class 2 metrics " = resultClass2,
                  "class 3 metrics " = resultClass3,
                  "class 4 metrics " = resultClass4)
  return(my_list)

}
```

```
> best_model_result <- CreateMetrics(bestCfMatrix_dt)
> best_model_result
$`class 1 metrics `
  accuracyClass1 farClass1     frrClass1 precisionClass1 recallClass1
1   0.9747474747         0 0.05050505051                1 0.9494949495

$`class 2 metrics `
  accuracyClass2      farClass2 frrClass2 precisionClass2 recallClass2
1   0.9987922705 0.002415458937         0    0.9473684211            1

$`class 3 metrics `
  accuracyClass3     farClass3 frrClass3 precisionClass3 recallClass3
1    0.984962406 0.03007518797         0    0.9867986799            1

$`class 4 metrics `
  accuracyClass4 farClass4 frrClass4 precisionClass4 recallClass4
1              1         0         0                1            1
```

```
> model2_result <- CreateMetrics(nnc)
> model2_result
$`class 1 metrics `
  accuracyClass1     farClass1    frrClass1 precisionClass1 recallClass1
1   0.9337977615 0.02782931354 0.1045751634    0.9013157895 0.8954248366

$`class 2 metrics `
  accuracyClass2     farClass2     frrClass2 precisionClass2 recallClass2
1   0.9750339213 0.00447761194 0.04545454545           0.875 0.9545454545

$`class 3 metrics `
  accuracyClass3     farClass3     frrClass3 precisionClass3 recallClass3
1   0.9585914755 0.05418719212 0.02862985685    0.9773662551 0.9713701431

$`class 4 metrics `
  accuracyClass4      farClass4     frrClass4 precisionClass4 recallClass4
1    0.979883821 0.004518072289 0.03571428571             0.9 0.9642857143

>
```

Classifier 1:Decision tree

- Decision trees help to handle discrete and numerical attributes by using splitting conditions.
- An advantage of a decision tree model, is the human readability. They provide high accuracy and interpretability.

Classifier 1 : Decision tree

Different classifiers with different configurations (trials, hidden layers/nodes)

- 1. I first sampled the data – 70%,75%,80% and split it accordingly for each sample
- 2. Used the c5.0 function to create a decision tree for each sample, setting trails as 10 each time and displayed the summary of each
- 3. Performed prediction by using the predict function, using each c5.0 model and the test data.
- 4. I used the trainControl and train functions to find the best model to use to obtain the best accuracy -> trails 20, model = rules and winnow = false

How I found the best model and it's result + metric results

```
control <- trainControl(method="repeatedcv", number=5, repeats=5)
set.seed(123)
fit.c50 <- caret::train(class~., data=cars, method="C5.0", metric='Accuracy', trControl=control)
fit.c50 #The final values used for the model were trials = 20, model = rules and winnow = FALSE
```

```
   (a)   (b)   (c)   (d)    <-classified as
  ----  ----  ----  ----
   290                      (a): class 1
         50                 (b): class 2
               907          (c): class 3
                     49     (d): class 4


Attribute usage:

100.00% BuyingPrice
100.00% PriceOfMaintenance
100.00% Capacity
100.00% EstimatedSafety
 85.80% SizeOfBoot
 68.21% Doors
```

```
> best_model_result
$`class 1 metrics `
  accuracyClass1 farClass1     frrClass1 precisionClass1 recallClass1
1   0.9747474747         0 0.05050505051                1 0.9494949495

$`class 2 metrics `
  accuracyClass2      farClass2 frrClass2 precisionClass2 recallClass2
1   0.9987922705 0.002415458937         0    0.9473684211            1

$`class 3 metrics `
  accuracyClass3     farClass3 frrClass3 precisionClass3 recallClass3
1    0.984962406 0.03007518797         0    0.9867986799            1

$`class 4 metrics `
  accuracyClass4 farClass4 frrClass4 precisionClass4 recallClass4
1              1         0         0                1            1

>
```

Classifier 1 : Decision tree

Evaluation of different classifiers

- I used the table function as well as the CrossTable function to evaluate my decision tree.

	cars.predict_dt75				
cars.test_dt75$class	1	2	3	4	Row Total
1	94	0	0	0	94
2	1	18	0	0	19
3	3	0	300	0	303
4	0	0	0	16	16
Column Total	98	18	300	16	432

Classifier 1 : Neural Network

Different classifiers with different configurations (trials, hidden layers/nodes)

- I created dummy columns using the dummy_cols function, which outputted 4 new columns displaying whether the entry was un acceptable, acceptable, good or very good
- I normalized the data by using a function I created which uses the min-max approach and scaling of the data
- I then added the new columns to the data and disregarded the old class attribute.

Classifier 1 : Neural Network

Different classifiers with different configurations (trials, hidden layers/nodes)

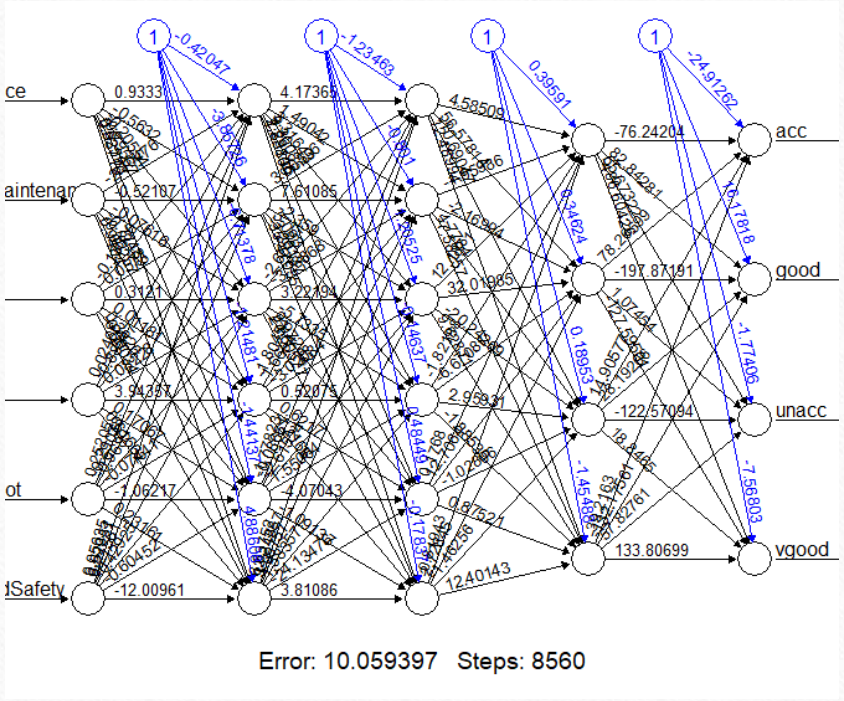
- I split the data into two samples (70% and 60%).
- I created three models using neuralnet function, one wit 2 hidden layers of 2 and 4, and two with three hidden layers of 6, 6, 4.

Classifier 1 : Neural Network

Evaluation of different classifiers

- I used the compute, max.col functions to obtain the computed result of each neural network and max column.
- I checked where the predicted values where equal to the original and set them accordingly (true or false)
- I used the table and CrossTable and confusion matrix functions to output the accuracy.
- The highest accuracy obtained for this classifier is 95% using a sample size of 60-40.

Classifier 1: Neural Network



Best Classifier: Decision tree

- The decision tree was chosen as it had the highest accuracy of 99% using a sample size of 75-25.

```
> C5imp(cars.best_model_dt,metric='usage')
                  Overall
BuyingPrice        100.00
PriceOfMaintenance 100.00
Capacity           100.00
EstimatedSafety    100.00
SizeOfBoot          85.80
Doors               68.21
```

```
> best_model_result
     accuracy    precision       recall           f1
1 0.9884259259 0.9494949495 1.0000000000 0.9740932642
2 0.9884259259 1.0000000000 0.9473684211 0.9729729730
3 0.9884259259 1.0000000000 0.9867986799 0.9933554817
4 0.9884259259 1.0000000000 1.0000000000 1.0000000000
>
```

C02 Emissions Report

Problem

- CO2 emissions have been rising, and decreasing and we want to know what effects it, be it income group, country, region etc.
- As well as increasing population growth in certain areas.

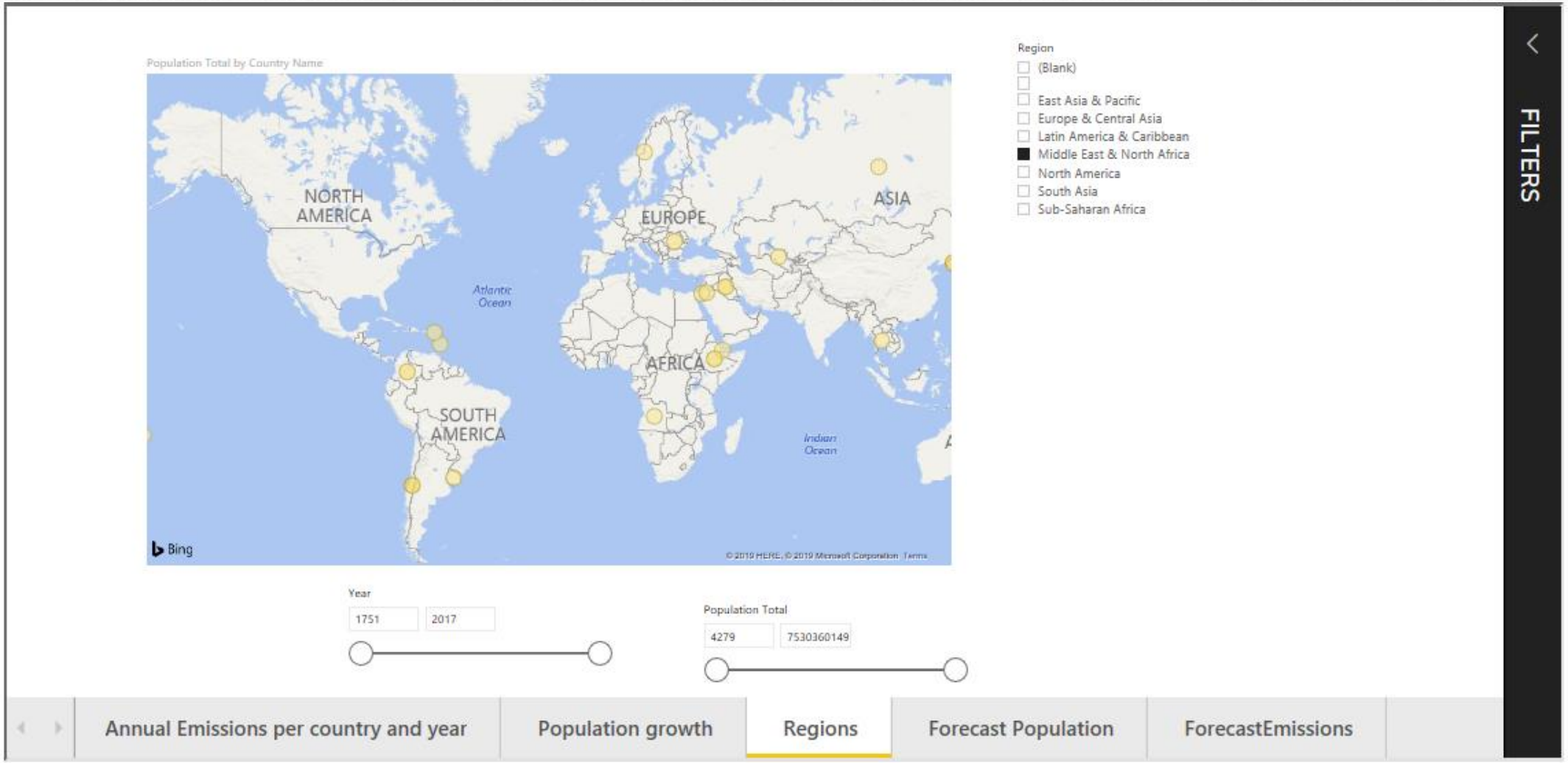
Dataset

- I have three excels which I made use of in PowerBI and also made sure there were incorrect values or symbols in the data.
- I made an extra table called Years to link all the years together, as this allows me to use just one attribute for each excel file. Without it, I would need to use the year attribute from each table, which is possible but I felt it was easier to simply use a common year attribute.
- I linked each excel file together on PowerBI by using the relationship attribute and linking them using the country code.

Data cleaning

1. Checked for any symbols or invalid data
2. Renamed some columns for better readability.

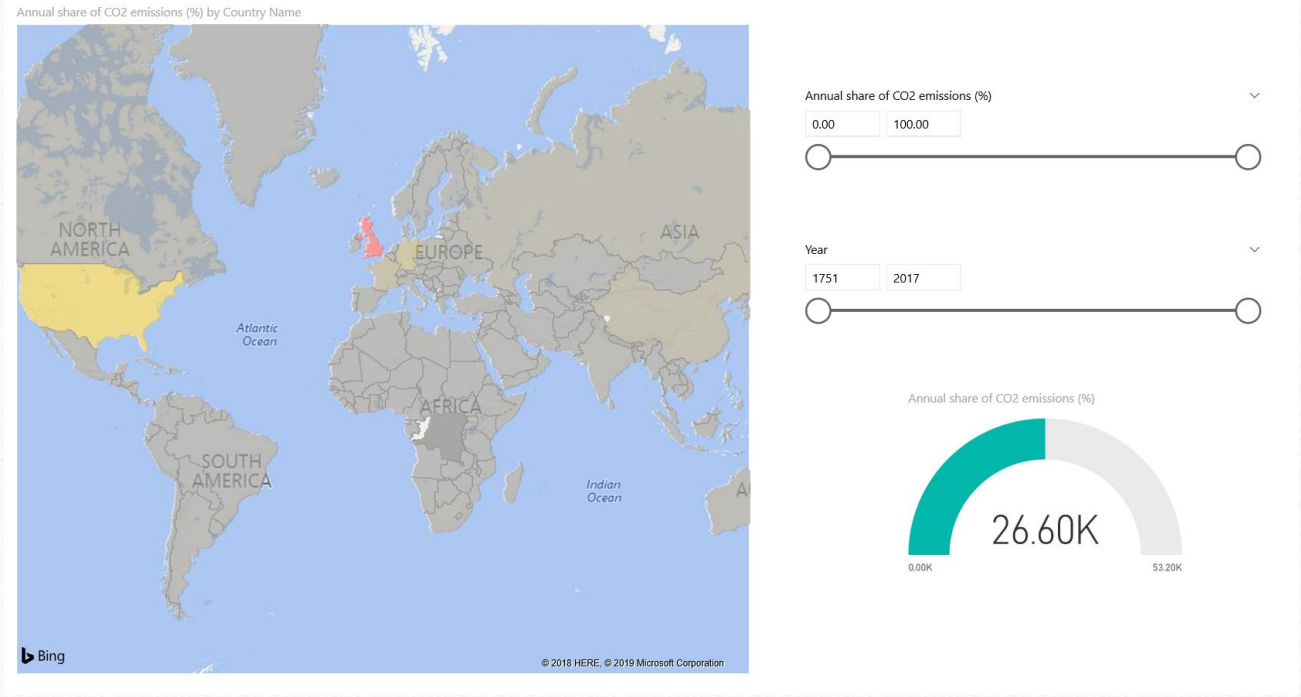
My Report


Population Total by Country Name
NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
Atlantic Ocean
Indian Ocean
Bing
Region
(Blank)
East Asia & Pacific
Europe & Central Asia
Latin America & Caribbean
Middle East & North Africa
North America
South Asia
Sub-Saharan Africa
FILTERS
Year
1751
2017
Population Total
4279
7530360149
Annual Emissions per country and year
Population growth
Regions
Forecast Population
ForecastEmissions

Basic Interpretation

- I created 3 reports in power BI with my basic interpretation
- This includes exploring the data by year and country, region. As well as exploring different income groups and population growth throughout the years and allowing the user to pick the years or population number through the user of a slider.
- All this can be seen in the slides below.

Annual Emissions per country per year



This page displays the world map which changes when using the two sliders on the right. The first slider represents the annual share of C02 %, and the second slider represents the years. As well as the gauge on the bottom right, which represents the same data as the first slider.

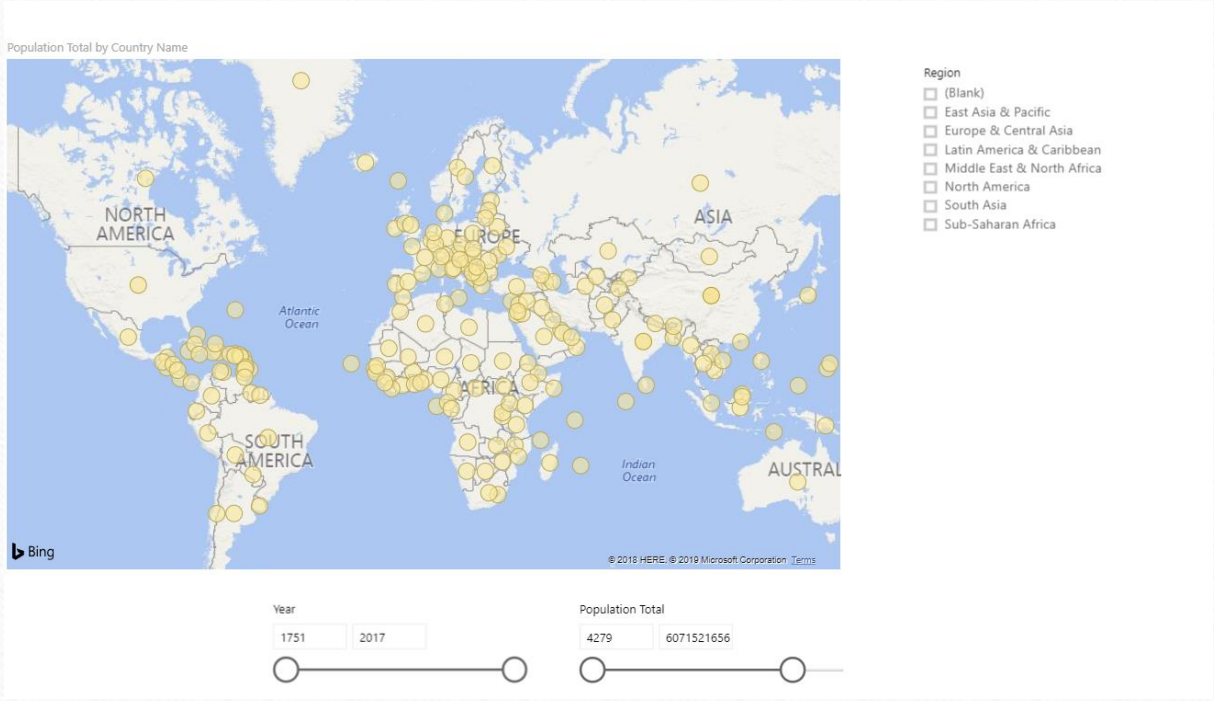
Population Growth

Population Total	Year
11192	2017
13649	2017
21729	2017
31196	2017
32125	2017
33400	2017
34571	2017
35446	2017
37922	2017
38695	2017
41109	2017
49290	2017
53127	2017
55144	2017
55345	2017
55641	2017
56171	2017
61559	2017
65441	2017
73925	2017
76965	2017
84287	2017
95843	2017

IncomeGroup
High income
Low income
Lower middle income
Upper middle income

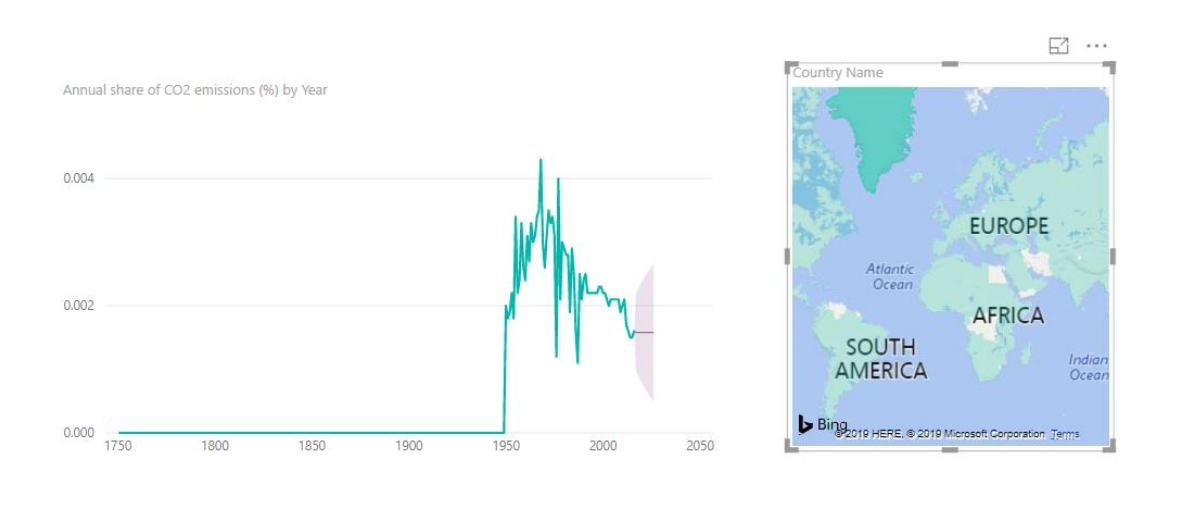
This page displays the world map on the left, and each bubble represents a country. The light green bubbles represent less population than the yellow/red ones. On the right is a table with the population total per country per year, and at the bottom is a table displaying the income group. If any item in the tables are clicked, the map will change.

Regions



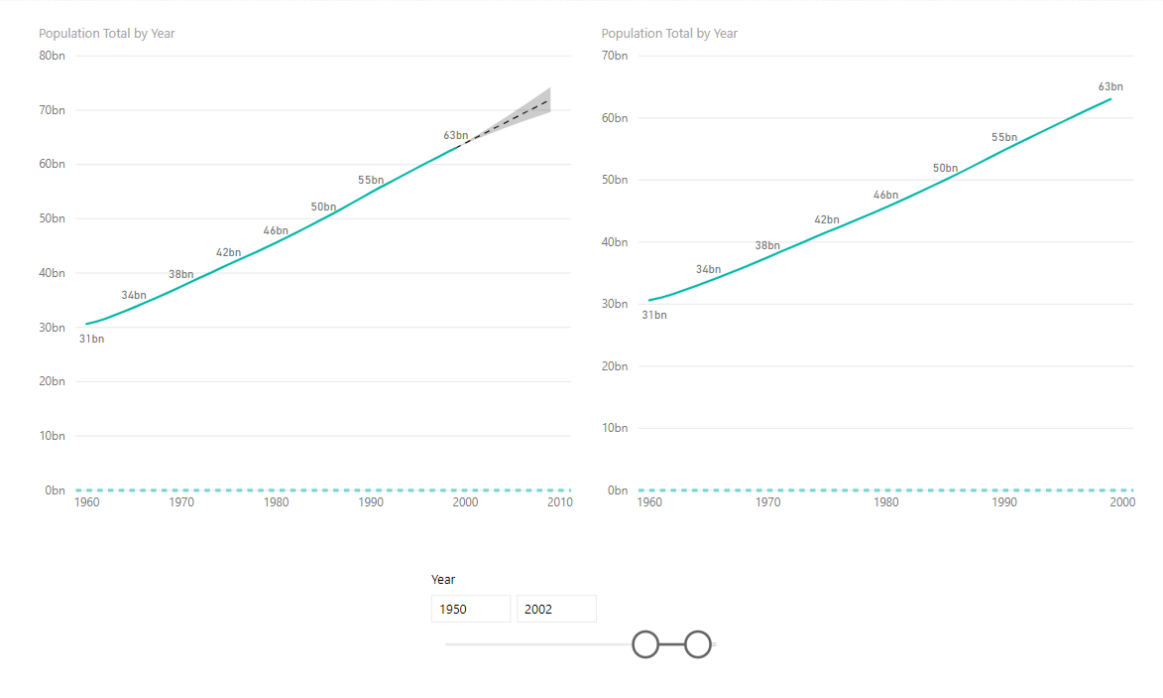
In this page, there is a slicer with the different regions and when one is chosen, the map will change to represent the chosen region. As well as the two sliders at the bottom which are the different years and population totals.

Forecast of future Population



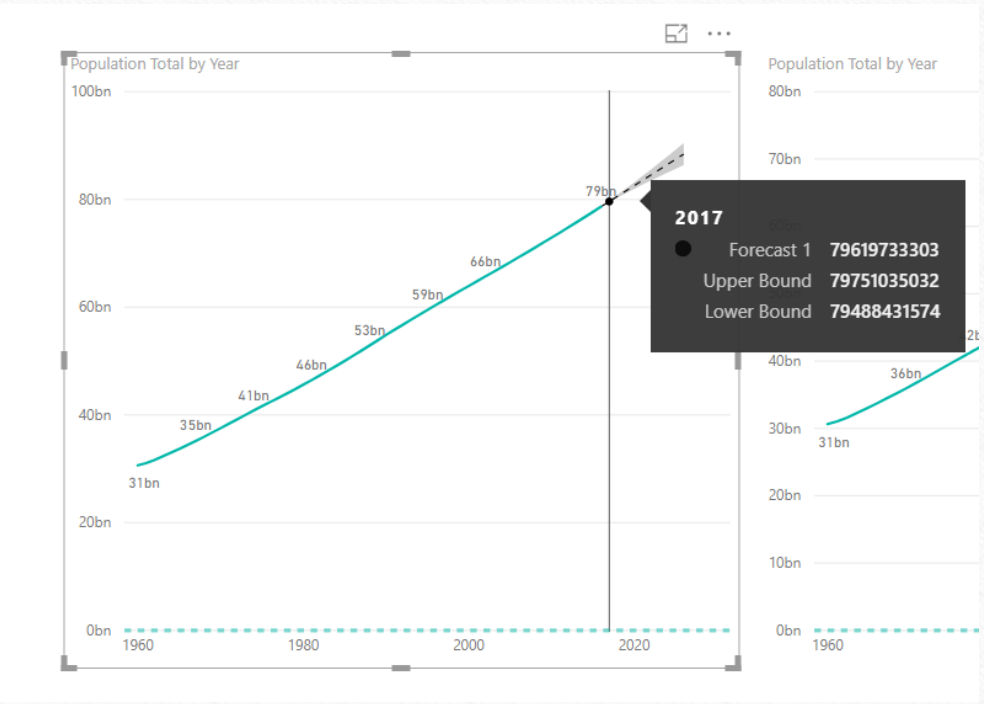
This line chart represents that population total by year and the map represents the amount of population per country (light green is lower population whilst yellow/red is high). The top grey part in the line chart is the forecast of how much the population will change in the coming years. You can click on a country on the map and the forecast will change per country.

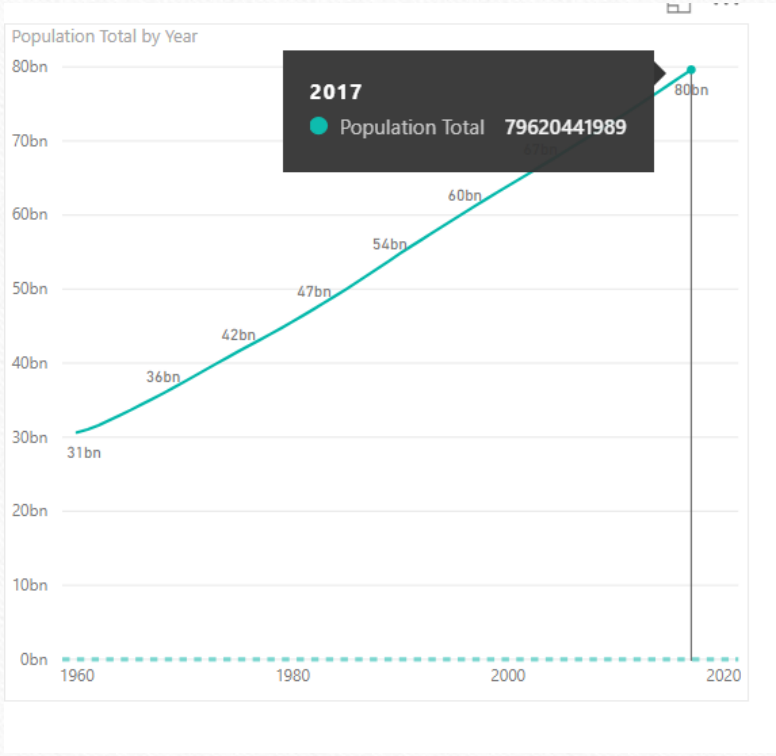
Forecast of Population



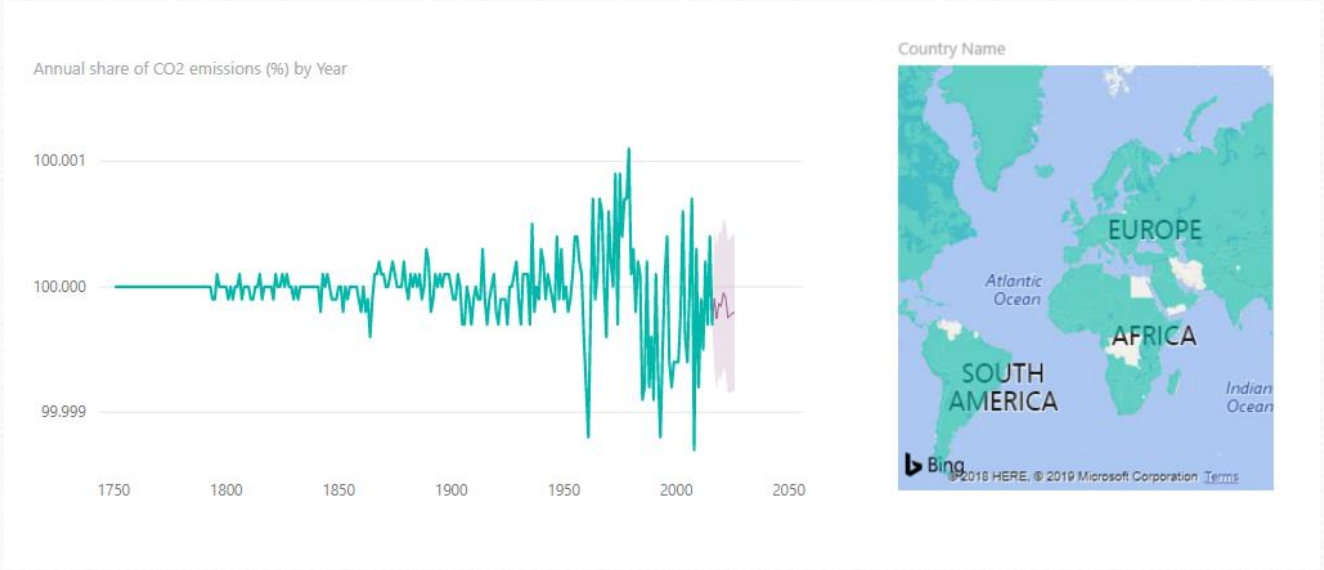
The left line chart is contains the forecasted population growth, and the one beside it contains the actual population growth. As you can see in 1999 it is forecasted as 63billion , and on the right it is actually 63 billion.

Forecast of Population 2017 comparison



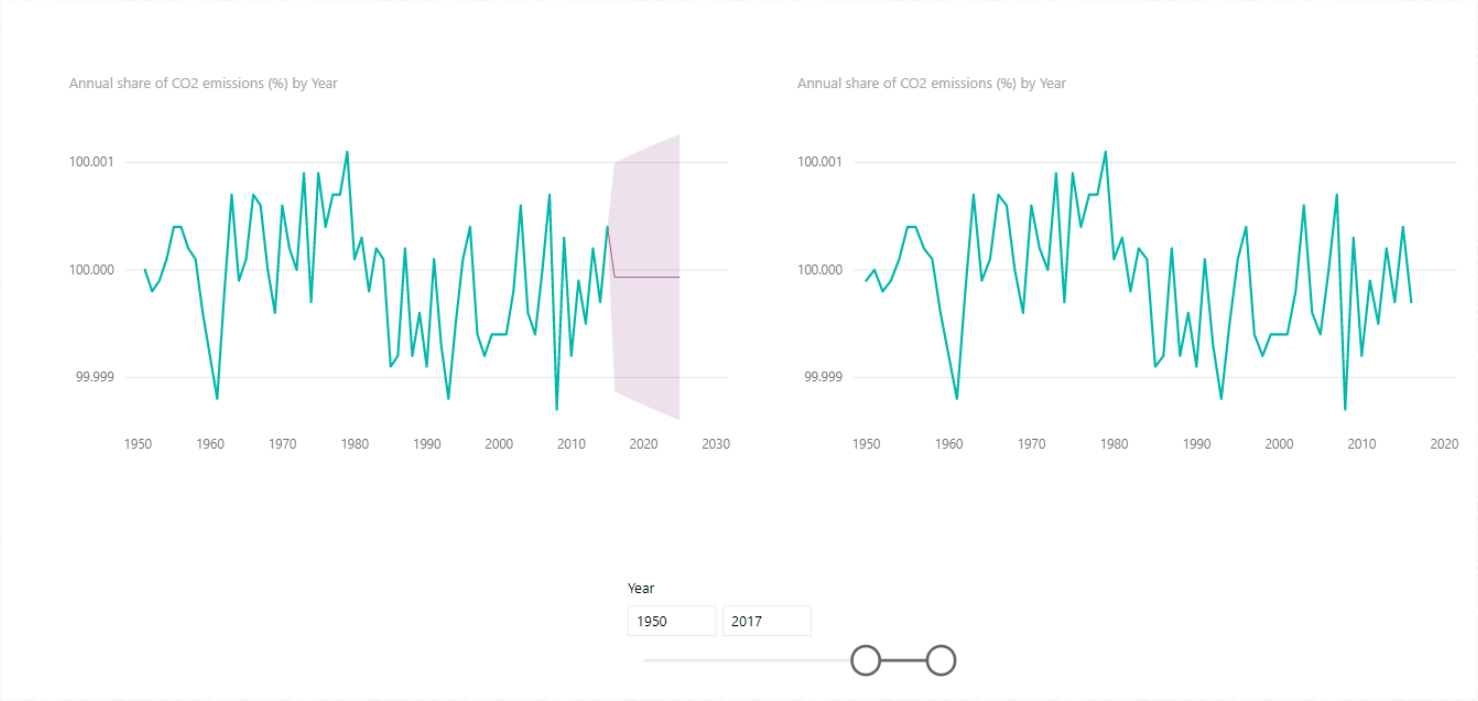


Forecast of future Emissions



The line chart represents the annual share of C02 emissions per year. The last piece of the line (in red) is the forecast of how much the C02 emissions will change in the coming years. You can click on a country on the map and the forecast will change per country.

Forecast Emission



The left line chart is contains the forecasted emissions, and the one beside it contains the actual emissions. As you can see the emissions were forecasted properly as they match the actual

Advanced Interpretation

- Between 1757 up until approximately the 1880's, the UK was the country who emitted most CO2
- After 1890's we can see a drop in CO2 emissions resulting in only 54.88% whereas before it was over 6k.
- Nowadays, from 2013 onwards, the emissions are dropping a little every year due to coal only accounting for 5.3% of total energy consumed in the UK, down from 22% in 1995.
- These visualizations can be seen in the following slides.
- Article backing up my interpretation: https://www.ft.com/content/47563b2a-21f6-11e8-9a70-08f715791301

Annual share of CO2 emissions (%) by Country Name

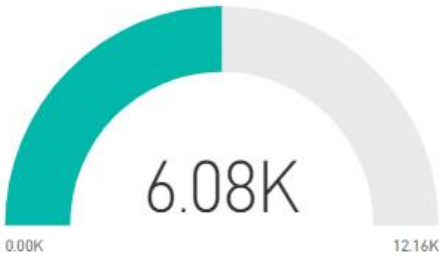
Annual share of CO2 emissions (%)

26.69 100.00

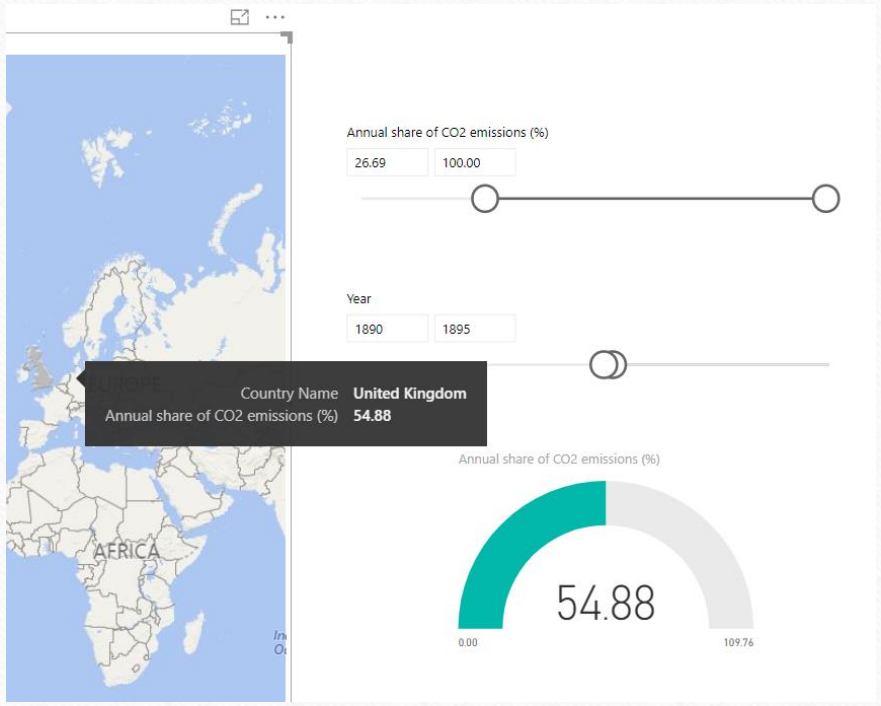
Year

1757 1819

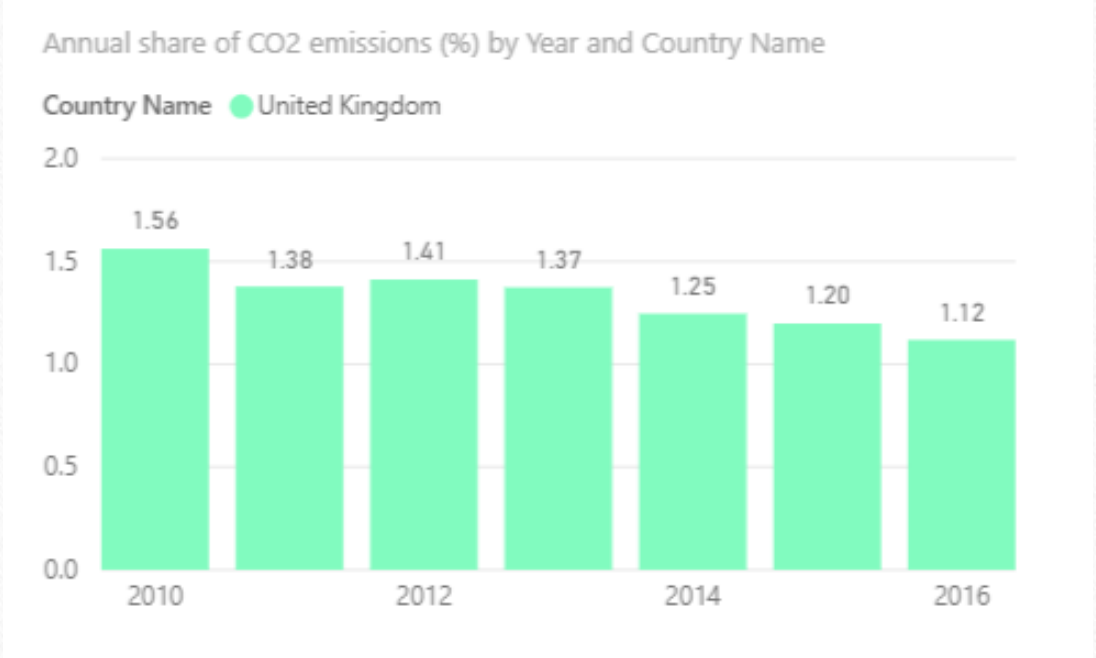
Annual share of CO2 emissions (%)



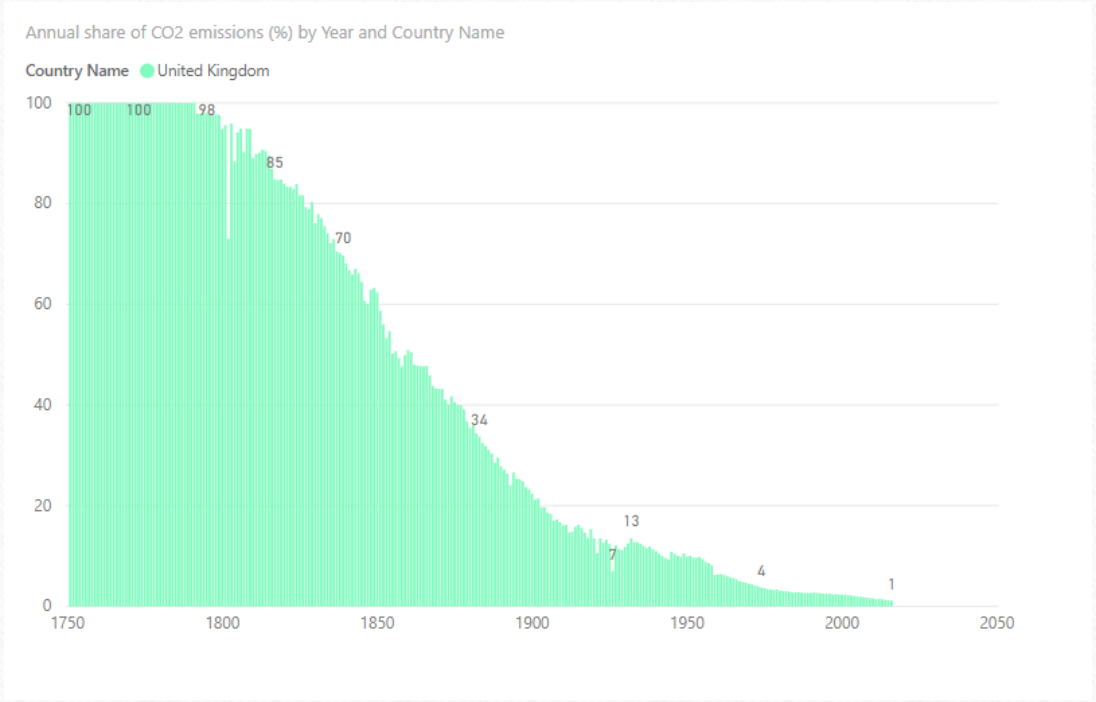
UK drop in 1890



UK dropping from 2013



UK emissions throughout the years



Proposed Arguments

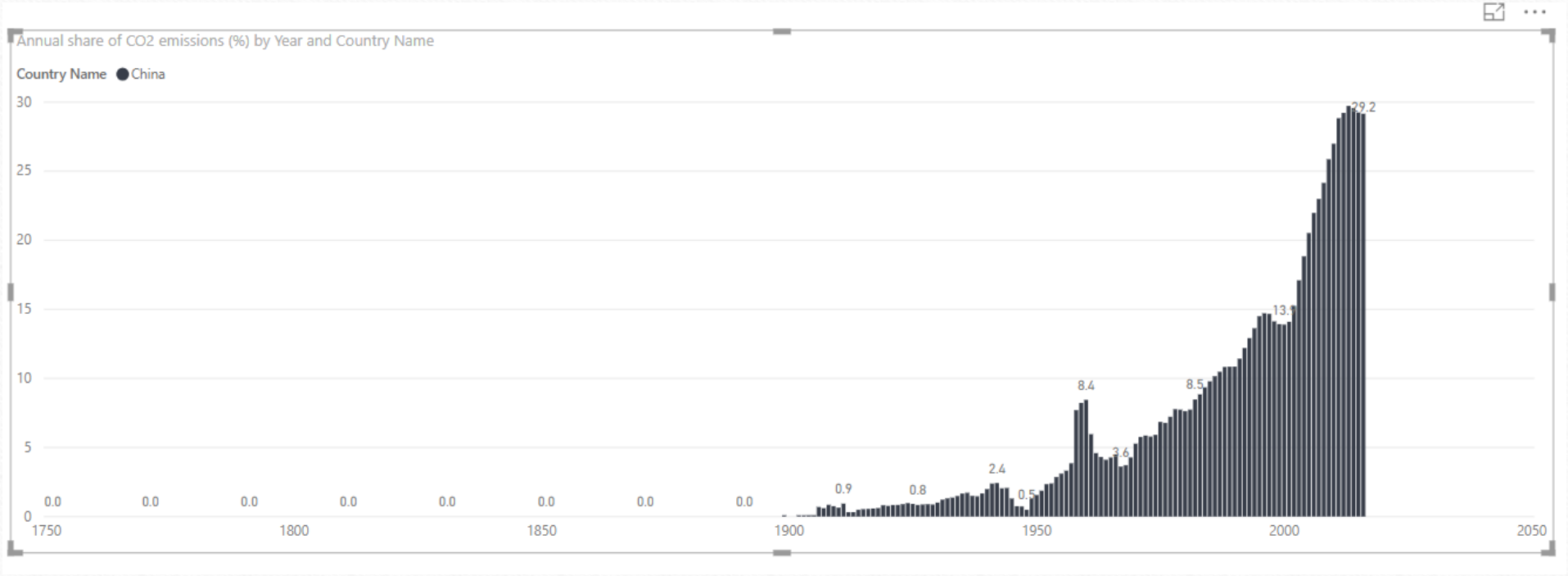
- In 1978 China had a GDP of only 200 billion, only about 4% of the worlds GDP
- Nowadays it has risen to 11 trillion and accounts for 15% of all economic activity in the world.
- This activity is the manufacturing industry thanks to cheap world wide shipping.

Proposed Arguments

- China became the manufacturing giant because of the rise of Shenzhen
- This created a huge impact on the CO2 emission of china, and this is why China emits significantly more then its population share (29% emissions, 19% population)
- The articles backing up my argument can be found here :
  https://knowlo.in/why-chinese-manufacturing-wins/

  https://www.nature.com/articles/sdata2017201

China throughout the years



Further explanation of argument

- The co2 emissions of China increased by 13% in the span of 15 years (from 2002-2017).
- This is due to the fact that they joined the world trade orgranisation in 2002 and that's when the huge impact happened, as I explained before, due to the Manufacturing industry.
- This representation can be seen in the next slide

China emissions spike between 2002 and 2017