Exploring with R - Natural Resources

G.Tsolov

2022-03-29

Source

https://ourworldindata.org/fossil-fuels
further sources cited within each dataset.

Environment setup

Loading and preparing the usual.

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(tidyr)
library(readr)
library(ggplot2)
install.packages("corrplot")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
install.packages("here")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(here)
```

```
## here() starts at /cloud/project
```

```
install.packages("skimr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(skimr)
install.packages("janitor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

Loading and viewing the data

```
resources_final <- readRDS("resources_final.RDS")

str(resources_final)
```

```
## spec_tbl_df [8,972 x 23] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1               : num [1:8972] 1 2 3 4 5 6 7 8 9 10 ...
##  $ record_id          : num [1:8972] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Country            : chr [1:8972] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ Population         : num [1:8972] 13360000 13170000 12880000 12540000 12200000 ...
##  $ Year               : num [1:8972] 1980 1981 1982 1983 1984 ...
##  $ gas_production     : num [1:8972] 1.70e+09 2.24e+09 2.29e+09 2.41e+09 2.41e+09 ...
##  $ gas_consumption    : num [1:8972] 5.66e+07 8.50e+07 1.42e+08 1.42e+08 1.42e+08 ...
##  $ gas_exports        : num [1:8972] 0 0 0 0 0 0 0 0 0 0 ...
##  $ coal_production    : num [1:8972] 119000 125000 145000 145000 148000 151000 160000 167000 138000 1
##  $ coal_consumption   : num [1:8972] 119000 125000 145000 145000 148000 151000 160000 167000 138000 1
##  $ coal_exports       : num [1:8972] 0 0 0 0 0 0 0 0 0 0 ...
##  $ oil_production     : num [1:8972] 0 0 0 0 0 0 0 0 0 0 ...
##  $ oil_consumption    : num [1:8972] 406500 464600 452900 638800 638800 ...
##  $ oil_exports        : num [1:8972] 0 0 0 0 0 0 0 0 0 0 ...
##  $ gas_production_pc  : num [1:8972] 127 170 178 192 197 ...
##  $ gas_consumption_pc : num [1:8972] 4.24 6.45 10.99 11.29 11.6 ...
##  $ gas_exports_pc     : num [1:8972] 0 0 0 0 0 0 0 0 0 0 ...
##  $ coal_production_pc : num [1:8972] 0.00891 0.00949 0.01126 0.01157 0.01213 ...
##  $ coal_consumption_pc: num [1:8972] 0.00891 0.00949 0.01126 0.01157 0.01213 ...
##  $ coal_exports_pc    : num [1:8972] 0 0 0 0 0 0 0 0 0 0 ...
##  $ oil_production_pc  : num [1:8972] 0 0 0 0 0 0 0 0 0 0 ...
##  $ oil_consumption_pc : num [1:8972] 0.0304 0.0353 0.0352 0.051 0.0523 ...
##  $ oil_exports_pc     : num [1:8972] 0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   record_id = col_double(),
##   ..   Country = col_character(),
##   ..   Population = col_double(),
```

```
##   ..   Year = col_double(),
##   ..   gas_production = col_double(),
##   ..   gas_consumption = col_double(),
##   ..   gas_exports = col_double(),
##   ..   coal_production = col_double(),
##   ..   coal_consumption = col_double(),
##   ..   coal_exports = col_double(),
##   ..   oil_production = col_double(),
##   ..   oil_consumption = col_double(),
##   ..   oil_exports = col_double(),
##   ..   gas_production_pc = col_double(),
##   ..   gas_consumption_pc = col_double(),
##   ..   gas_exports_pc = col_double(),
##   ..   coal_production_pc = col_double(),
##   ..   coal_consumption_pc = col_double(),
##   ..   coal_exports_pc = col_double(),
##   ..   oil_production_pc = col_double(),
##   ..   oil_consumption_pc = col_double(),
##   ..   oil_exports_pc = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Seems like the first three gas columns in **View** and **str** can't fit the data. Tried changing the data type when loading. Originally as double, change to integer or numeric - no success. Values are readable when viewed with **head**.

Need to check if the actual values are being read. Let's see how much is the total gas production:

```
sum(resources_final$gas_production, na.rm=TRUE)
```

```
## [1] 1.048017e+14
```

It seems it's performing calculations but the result is not readable.
Will try and scale down the numbers by creating a calculated field dividing by 1 000 000. Removing 6 zeroes might do the trick:

```
test_new_field <- resources_final %>%
  mutate(new_gas_prod = gas_production / 1000000)

sum(test_new_field$new_gas_prod, na.rm = TRUE)
```

```
## [1] 104801685
```

Result is now readable. Must keep track of this and name fields accordingly!!!
Reproduce for all three gas columns:

```
resources_1 <- resources_final %>%
  mutate(gas_production_mill = gas_production / 1000000) %>%
  mutate(gas_consumption_mill = gas_consumption / 1000000) %>%
  mutate(gas_exports_mill = gas_exports / 1000000)
```

Tested summing up all new columns. Results are readable. Creating the final work table while excluding the original fields:

```
work_df <- resources_1 %>%
  select(-gas_production, -gas_consumption, -gas_exports)

glimpse(work_df)
```

```
## Rows: 8,972
## Columns: 23
## $ ...1                 <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15~
## $ record_id            <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15~
## $ Country              <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afg~
## $ Population           <dbl> 13360000, 13170000, 12880000, 12540000, 12200000,~
## $ Year                 <dbl> 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1~
## $ coal_production      <dbl> 119000, 125000, 145000, 145000, 148000, 151000, 1~
## $ coal_consumption     <dbl> 119000, 125000, 145000, 145000, 148000, 151000, 1~
## $ coal_exports         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ oil_production       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ oil_consumption      <dbl> 406500, 464600, 452900, 638800, 638800, 754900, 7~
## $ oil_exports          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1~
## $ gas_production_pc    <dbl> 127.20, 169.90, 178.10, 192.00, 197.20, 249.10, 2~
## $ gas_consumption_pc   <dbl> 4.24, 6.45, 10.99, 11.29, 11.60, 0.00, 0.00, 53.6~
## $ gas_exports_pc       <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
## $ coal_production_pc   <dbl> 0.008910, 0.009490, 0.011260, 0.011570, 0.012130,~
## $ coal_consumption_pc  <dbl> 0.008910, 0.009490, 0.011260, 0.011570, 0.012130,~
## $ coal_exports_pc      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ oil_production_pc    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ oil_consumption_pc   <dbl> 0.030430, 0.035270, 0.035160, 0.050950, 0.052340,~
## $ oil_exports_pc       <dbl> 0.000000, 0.000000, 0.000000, 0.000000, 0.000000,~
## $ gas_production_mill  <dbl> 1699.00, 2237.00, 2294.00, 2407.00, 2407.00, 2974~
## $ gas_consumption_mill <dbl> 56.64, 84.96, 141.60, 141.60, 141.60, 0.00, 0.00,~
## $ gas_exports_mill     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1025, 0, 0, 0, 0, 0~
```

- Hours later… After much trouble with readability (also in visualizations) I decided to divide all relevant fields by 1 million. Note to self - always scale down your values to avoid such complications. I think that's also the right way to keep data (thousands/millions in field name, not piling on mad zeroes).

```
work_df_1 <- work_df %>%
  mutate(coal_production_mill = coal_production / 1000000) %>%
  mutate(coal_consumption_mill = coal_consumption / 1000000) %>%
  mutate(coal_exports_mill = coal_exports / 1000000) %>%
  mutate(oil_production_mill = oil_production / 1000000) %>%
  mutate(oil_consumption_mill = oil_consumption / 1000000) %>%
  mutate(oil_exports_mill = oil_exports / 1000000)
```

- New work df (hopefully final)

```
work_df_final <- work_df_1 %>%
  select(-coal_production, -coal_consumption, -coal_exports, -oil_production, -oil_consumption, -oil_ex

glimpse(work_df_final)
```

Trying out a matrix plot:

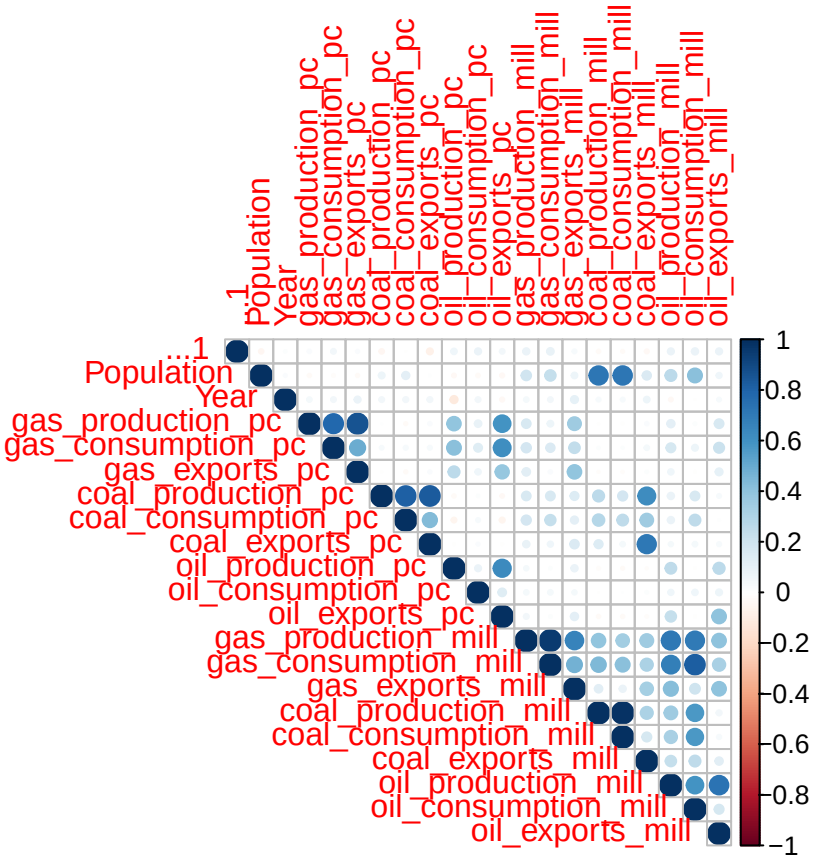
```
test_corr <- work_df_final %>%
  select(-Country, -record_id)

corr_main <- cor(test_corr)

round(corr_main, digits = 2)
```

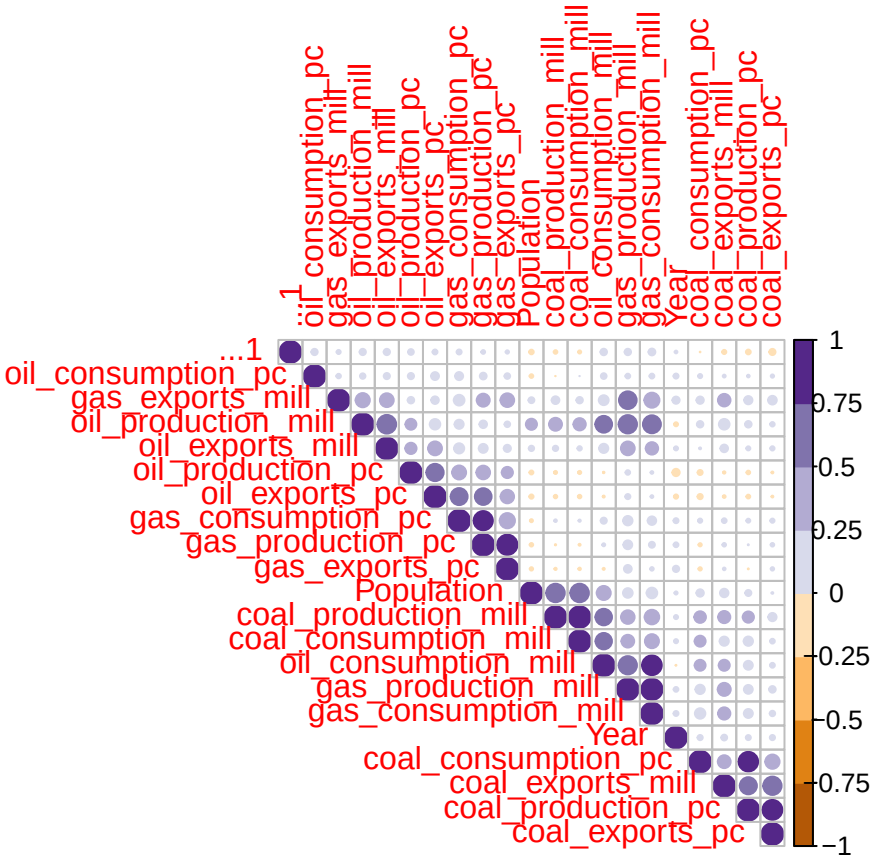
- Visualizing correlation matrix:

```
corrplot(corr_main, type = "upper")
```



- Reordering correlation matrix + color:

```
library(RColorBrewer)
corrplot(corr_main, type = "upper", order = "hclust", col = brewer.pal(n=8, name = "PuOr"))
```



I know this dataframe is not the best for testing correlation but I wanted to try it out. Result - a few interesting relationships to investigate and no inverse relationships.

Back on it

As most correlations are obvious - like production to exports, and production to consumption - there are a few interesting ones which I will preview in a bit. Obviously, some of the biggest producers of fossil fuels in the world are also one of the biggest consumers and exporters.

Basic info first

10 year period - 2011 to 2020!!! TOP 10 countries.

- COAL - biggest producers

```
coal_prod_summary <- work_df_final %>%
  select(Country, Year, coal_production_mill) %>%
  filter(Year >= 2011, Year <= 2020) %>%
  group_by(Country) %>%
  summarize(total_period = sum(coal_production_mill))

coal_prod_top10 <- coal_prod_summary %>%
  arrange(-total_period) %>%
  top_n(10)
```

```
## Selecting by total_period
```

- COAL - biggest consumers

```
coal_consump_summary <- work_df_final %>%
  select(Country, Year, coal_consumption_mill) %>%
  filter(Year >= 2011, Year <= 2020) %>%
  group_by(Country) %>%
  summarize(total_period = sum(coal_consumption_mill))

coal_consump_top10 <- coal_consump_summary %>%
  arrange(-total_period) %>%
  top_n(10)
```

```
## Selecting by total_period
```

- COAL - biggest exporters

```
coal_exports_summary <- work_df_final %>%
  select(Country, Year, coal_exports_mill) %>%
  filter(Year >= 2011, Year <= 2020) %>%
  group_by(Country) %>%
  summarize(total_period = sum(coal_exports_mill))

coal_exports_top10 <- coal_exports_summary %>%
  arrange(-total_period) %>%
  top_n(10)
```

```
## Selecting by total_period
```

- A visual for Coal:

Top 10 countries for coal production, consumption, and exports.
Aggregate for the last ten years (2011 - 2020) in million metric tonnes

Coal production

	Country	total_period
1	China	37627.00
2	United States	7706.40
3	India	6168.30
4	Australia	4868.50
5	Indonesia	4788.70
6	Russia	3766.70
7	South Africa	2561.60
8	Germany	1707.10
9	Poland	1287.60
10	Kazakhstan	1094.24

Coal consumption

	Country	total_period
1	China	40509.00
2	India	7949.10
3	United States	7016.50
4	Germany	2187.90
5	Russia	2114.20
6	Japan	1903.50
7	South Africa	1799.00
8	Poland	1333.80
9	South Korea	1318.50
10	Australia	1140.19

Coal exports

	Country	total_period
1	Indonesia	3818.600
2	Australia	3661.600
3	Russia	1730.000
4	United States	863.360
5	Colombia	780.950
6	South Africa	767.510
7	Canada	332.900
8	Netherlands	326.530
9	Mongolia	247.780
10	North Korea	100.621

- GAS - repeat everything.

Top 10 countries for natural gas production, consumption, and exports.
Aggregate for the last ten years (2011 - 2020) in million cubic meters

Gas production

	Country	total_period_mill
1	United States	7821600
2	Russia	6324100
3	Iran	1715600
4	Canada	1555400
5	Qatar	1456300
6	China	1220400
7	Norway	1140200
8	Saudi Arabia	1043610
9	Algeria	784870
10	Australia	762710

Gas consumption

	Country	total_period_mill
1	United States	7825700
2	Russia	4645300
3	China	1842300
4	Iran	1669300
5	Saudi Arabia	1043610
6	Japan	1037100
7	Canada	1026200
8	Germany	885120
9	United Kingdom	700240
10	Mexico	689020

Gas exports

	Country	total_period_mill
1	Russia	1811100
2	Qatar	1111400
3	Norway	1092310
4	United States	765570
5	Canada	742030
6	Netherlands	491910
7	Australia	474520
8	Algeria	436260
9	Turkmenistan	352430
10	Malaysia	348200

- Oil - repeat again.

Top 10 countries for oil production, consumption, and exports.
Aggregate for the last ten years (2011 - 2020) in million cubic meters

Oil production

	Country	total_period
1	Russia	5965.7
2	Saudi Arabia	5784.0
3	United States	5264.7
4	China	2327.5
5	Iraq	2234.0
6	Canada	2163.3
7	Iran	2060.3
8	United Arab Emirates	1811.7
9	Kuwait	1593.9
10	Brazil	1412.7

Oil consumption

	Country	total_period
1	United States	11257.0
2	China	6235.8
3	Japan	2378.7
4	India	2084.0
5	Russia	2001.3
6	Brazil	1788.9
7	Saudi Arabia	1617.8
8	South Korea	1426.5
9	Canada	1406.6
10	Germany	1371.3

Oil exports

	Country	total_period
1	Saudi Arabia	3394.00
2	Russia	2273.70
3	Canada	1512.40
4	Iraq	1408.50
5	United Arab Emirates	1177.00
6	Nigeria	953.20
7	Kuwait	890.40
8	Iran	821.19
9	Norway	774.79
10	Venezuela	763.13

Interesting correlations

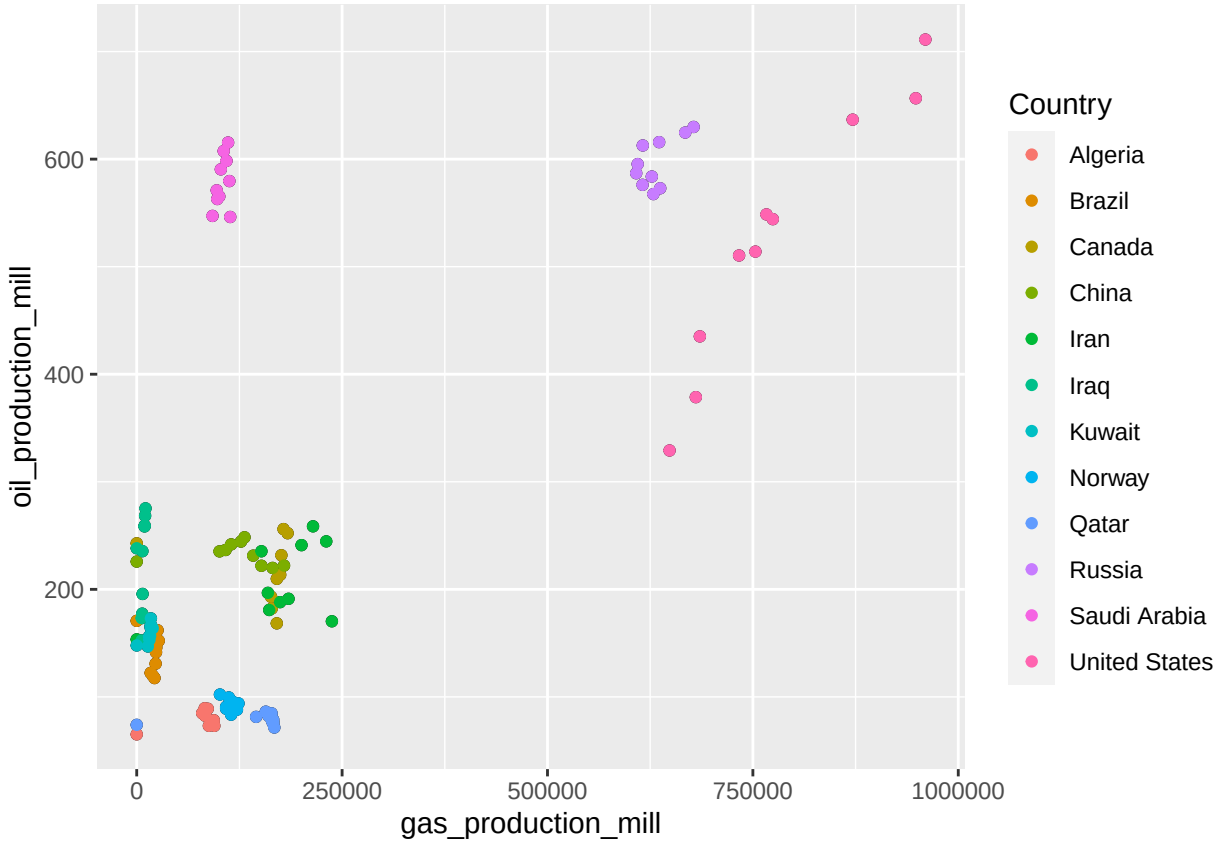
In order to keep things cleaner I will focus on some of the countries from the top_10 results above.

- Gas production to Oil production

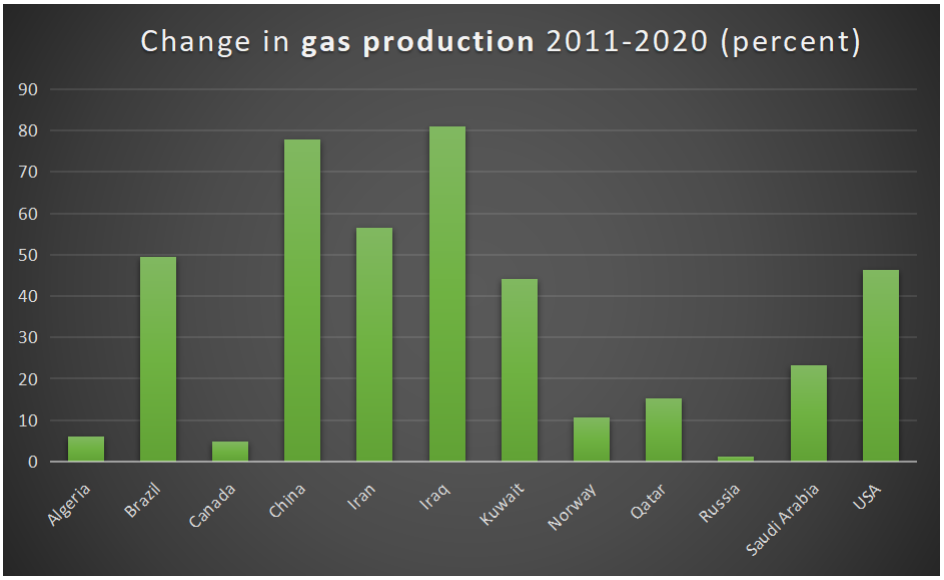
```
new_a1 <- work_df_final %>%
  filter(Year >= 2011, Year <= 2020) %>%
```

```
  filter(Country %in% c("United States", "Norway", "Iran", "China", "Iraq", "Canada", "Saudi Arabia", "
```

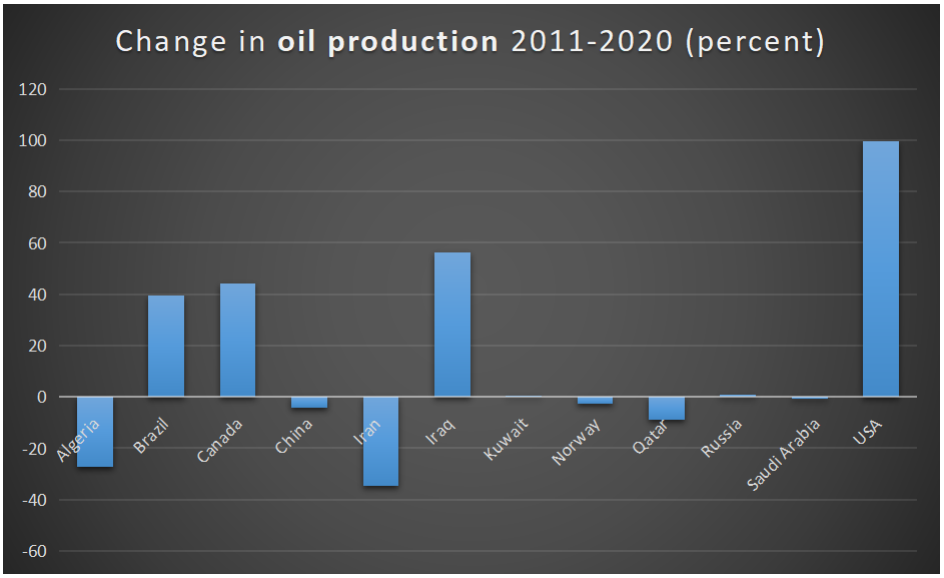
```
ggplot(new_a1, aes(gas_production_mill, oil_production_mill)) +
  geom_jitter(aes(group = Country)) +
  geom_point(aes(colour = Country))
```



- Change in gas production.
  China, Iraq, Iran, and Brazil have achieved significant increase in gas production. Followed closely by Kuwait and the US. Notable here is Russia with practically no change.



- Change in oil production.
  The US have doubled their oil production over the past ten years surpassing even Russia and Saudi Arabia. This now grants them the number one spot in the world. Other countries to note are Iraq, Canada, and Brazil, who also see a significant increase. Russia and Saudi Arabia, as former leaders in oil production see little-to-no change in their production.



- Coal production & consumption compared to Population

```
new_a2 <- work_df_final %>%
  select(Country, Year, Population, coal_production_mill, coal_consumption_mill) %>%
  filter(Year >= 2011, Year <= 2020) %>%
  filter(Country %in% c("United States", "Australia", "Germany", "Russia", "Indonesia", "South Africa",
```

I'm excluding China and India here for a better visualization later.

Again having trouble with readability on visualizations.

Tried adding a calculated field but it reads the data as string, despite the fact that Population data type is listed as double... Exporting a few filtered tables to spreadsheets so I can divide Population by 1 mill.

Added new field and imported table. Let's try this again:

```
install.packages("readxl")
```

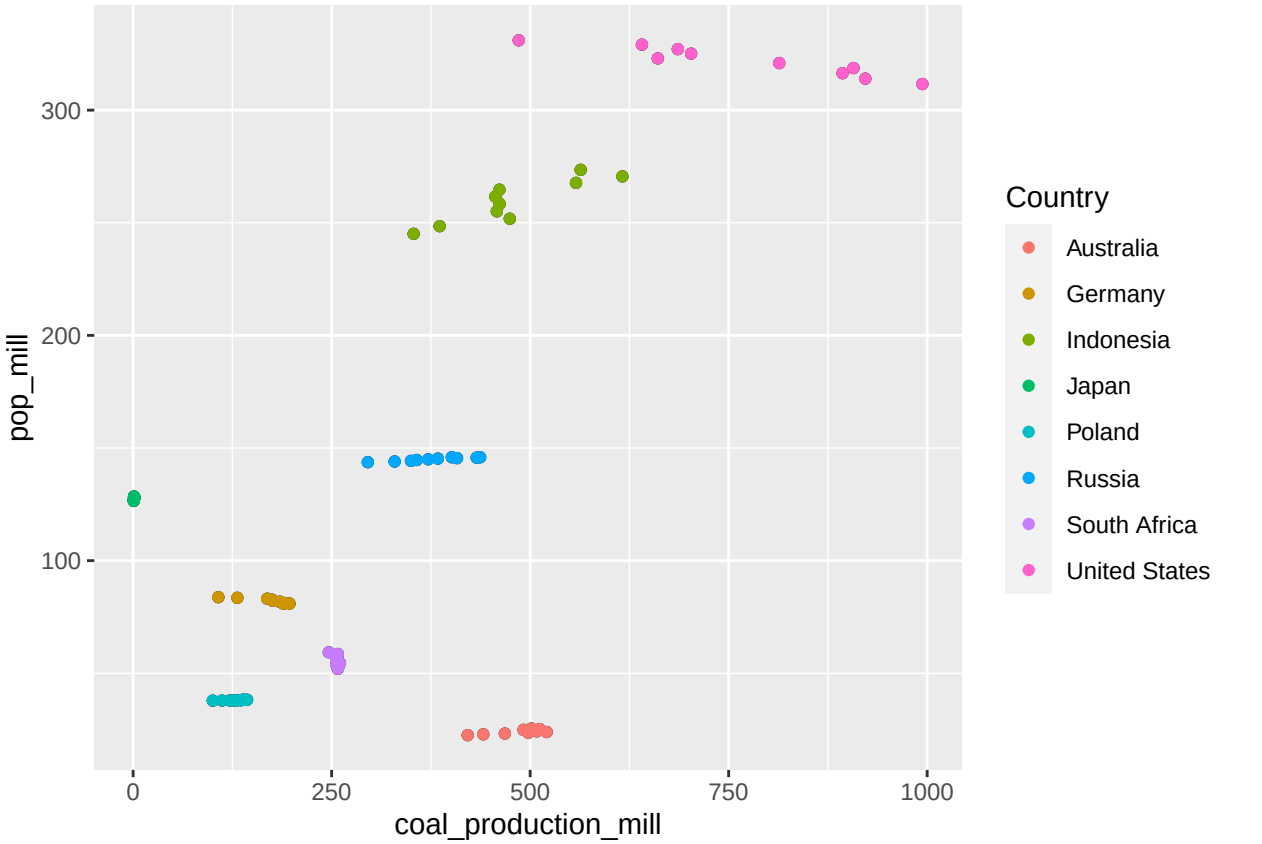
```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(readxl)

new_a3 <- read_xlsx("new_a3.xlsx")
str(new_a3)
```

```
## tibble [80 x 6] (S3: tbl_df/tbl/data.frame)
##  $ Country              : chr [1:80] "Australia" "Australia" "Australia" "Australia" ...
##  $ Year                 : num [1:80] 2011 2012 2013 2014 2015 ...
##  $ Population           : num [1:80] 22540000 22900000 23250000 23600000 23930000 ...
##  $ coal_production_mill : num [1:80] 421 441 468 498 521 ...
##  $ coal_consumption_mill: num [1:80] 129 128 117 112 117 ...
##  $ pop_mill             : num [1:80] 22.5 22.9 23.2 23.6 23.9 ...
```

```
ggplot(new_a3, aes(coal_production_mill, pop_mill)) +
  geom_jitter(aes(group = Country)) +
  geom_point(aes(colour = Country))
```



Interesting here is Australia with it's small population but very high coal production. They must have one of the highest coal production per capita - let's check:

```
a3_pc_top <- work_df_final %>%
  select(Country, Year, coal_production_pc) %>%
  filter(Year >= 2011, Year <= 2020) %>%
  group_by(Country) %>%
  summarize(avg_coal_pc = mean(coal_production_pc)) %>%
  arrange(-avg_coal_pc) %>%
  top_n(10)
```

```
## Selecting by avg_coal_pc
```

```
head(a3_pc_top)
```

```
## # A tibble: 6 x 2
##   Country      avg_coal_pc
##   <chr>              <dbl>
## 1 Australia          20.2
## 2 Mongolia           12.3
## 3 Kazakhstan          6.22
## 4 South Africa        4.61
## 5 Czechia             4.35
## 6 Bulgaria            4.35
```

We can confirm it. Australia has BY FAR the largest per capita production of coal in the world. Otherwise the biggest producer of coal by sheer volume is undisputedly China.

- Now let's see consumption (including China & India):

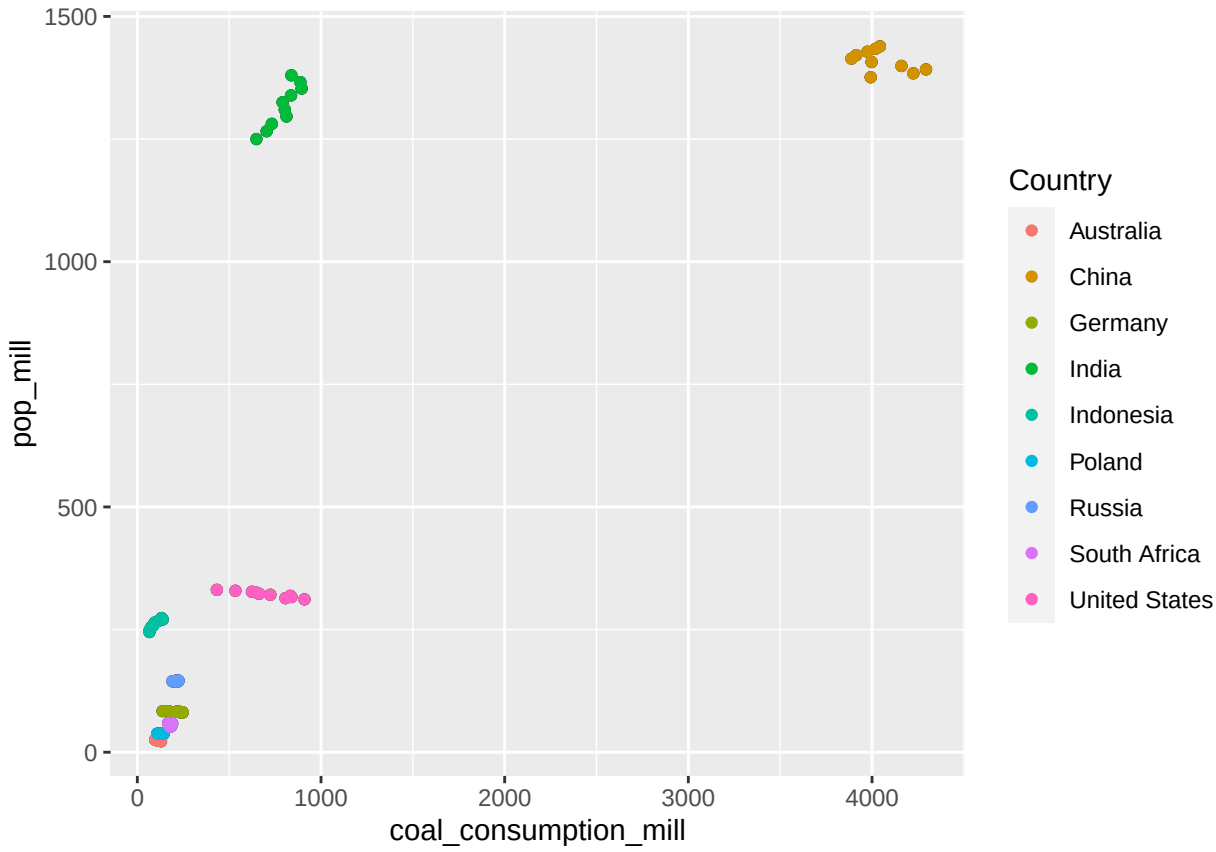
```
new_a4 <- work_df_final %>%
  select(Country, Year, Population, coal_production_mill, coal_consumption_mill) %>%
  filter(Year >= 2011, Year <= 2020) %>%
  filter(Country %in% c("United States", "Australia", "Germany", "Russia", "Indonesia", "South Africa",
```

Again had to export to spreadsheets for the calculation. Note to self - I should really make a better plan at the start...

```
new_a5 <- read_xlsx("new_a4.xlsx")
str(new_a5)
```

```
## tibble [90 x 6] (S3: tbl_df/tbl/data.frame)
##  $ Country              : chr [1:90] "Australia" "Australia" "Australia" "Australia" ...
##  $ Year                 : num [1:90] 2011 2012 2013 2014 2015 ...
##  $ Population           : num [1:90] 22540000 22900000 23250000 23600000 23930000 ...
##  $ coal_production_mill : num [1:90] 421 441 468 498 521 ...
##  $ coal_consumption_mill: num [1:90] 129 128 117 112 117 ...
##  $ pop_mill             : num [1:90] 22.5 22.9 23.2 23.6 23.9 ...
```

```
ggplot(new_a5, aes(coal_consumption_mill, pop_mill)) +
  geom_jitter(aes(group = Country)) +
  geom_point(aes(colour = Country))
```



- Finally let's see countries with the highest consumption per capita:

```
a5_pc_top <- work_df_final %>%
  select(Country, Year, coal_consumption_pc) %>%
  filter(Year >= 2011, Year <= 2020) %>%
  group_by(Country) %>%
  summarize(avg_coal_pc = mean(coal_consumption_pc)) %>%
  arrange(-avg_coal_pc) %>%
  top_n(10)
```

```
## Selecting by avg_coal_pc
```

```
head(a5_pc_top)
```

```
## # A tibble: 6 x 2
##   Country    avg_coal_pc
##   <chr>            <dbl>
## 1 Australia         4.76
## 2 Kazakhstan        4.54
## 3 Bulgaria          4.53
## 4 Serbia            4.39
## 5 Czechia           4.27
## 6 Greece            3.99
```