

Global CO2 Emissions Data Analysis

Jennifer Liu, David Kook, Xinwen Xu, Yuxuan Peng, William Escobar

2022-11-10

Total CO2 Emission Analysis using data from 1750-2020

The code below analyzes CO2 emission data from Kaggle. The three main goals are: 1. to understand the historical cumulative emissions,
2. Present day emission analysis,
3. Using this data for future emission predictions.

Libraries needed for this analysis:

Data Cleaning

```
data <- getURL("https://raw.githubusercontent.com/jliuob/QBSsquad/main/CO2_emission_by_countries.csv")
rawDat <- read.csv(text = data)
```

```
rawCO2<-rawDat
rawCO2%>% summarise_all(~ sum(is.na(.)))
```

```
##   Country Code Calling.Code Year CO2.emission..Tons. Population.2022. Area
## 1      0 271           0    0              0          6504 4336
##   X..of.World Density.km2.
## 1      0           0
```

```
#Find all the countries that has NA in each column
naCode <- rawCO2 %>% filter(is.na(Code)) %>% distinct(Country)
naCall <- rawCO2 %>% filter(is.na(Calling.Code)) %>% distinct(Country)
naPopu <- rawCO2 %>% filter(is.na(Population.2022.)) %>% distinct(Country)
naArea <- rawCO2 %>% filter(is.na(Area)) %>% distinct(Country)
naXPre <- rawCO2 %>% filter(is.na(X..of.World)) %>% distinct(Country)
naDens <- rawCO2 %>% filter(is.na(Density.km2.)) %>% distinct(Country)
allNA <- bind_rows(naCode,naCall,naPopu,naArea,naXPre,naDens)
allNA <- distinct(allNA) %>% arrange(Country)
head(allNA,10) #In total of 41 countries have different missing values
```

```
##           Country
## 1      Anguilla
## 2      Antarctica
## 3        Aruba
```

```
## 4          Bermuda
## 5          Brunei
## 6      Cape Verde
## 7 Christmas Island
## 8          Congo
## 9      Cook Islands
## 10     Cote d'Ivoire
```

Combine all the table above to better see which values are missing.

```
nacheck<-function(str){
  #This function get the individual column data above,
  df<- get(str)
  df$UniC <- df$Country
  #create a second column to later join all the table together
  colnames(df)[1] = str
  #rename the country column as the abbreviation of the initial data frame
  return(df)
}
#run the function for all the column that had missing data
naCode<- nacheck("naCode")
naCall<- nacheck("naCall")
naPopu<- nacheck("naPopu")
naArea<- nacheck("naArea")
naXPre<- nacheck("naXPre")
naDens<- nacheck("naDens")
#Join all the column together
allNATable<-allNA %>% full_join(naCode,by = c("Country" = "UniC")) %>%
  full_join(naCall,by = c("Country" = "UniC")) %>%
  full_join(naPopu,by = c("Country" = "UniC")) %>%
  full_join(naArea,by = c("Country" = "UniC")) %>%
  full_join(naXPre,by = c("Country" = "UniC")) %>%
  full_join(naDens,by = c("Country" = "UniC"))
view(allNATable)
```

```
write.csv(allNATable, file = "OriginalNAs.csv",row.names = TRUE)
#Write out the table for future reference, can also see this data in Excel
```

```
dat20 <- rawCO2 %>% filter(Year == 2020)
summary(dat20$CO2.emission..Tons.)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## 1.539e+05 4.506e+07 3.011e+08 7.519e+09 2.770e+09 4.170e+11
```

```
#decided on the first quartile
```

```
#Read in the cleaned version from excel
y <- getURL("https://raw.githubusercontent.com/jliuob/QBSsquad/main/CO2_Cleaned_V2.csv")
CO2 <- read.csv(text = y)
colnames(CO2)[1] = "Country"
#again, check for all the NA entries
CO2 %>% summarise_all(~ sum(is.na(.)))
```

```
## Country Code Calling.Code Year CO2.emission..Tons. Population.2022. Area
## 1 0 0 0 0 0 4065 1355
## X..of.World Density.km2.
## 1 0 0
```

```
C02 <- C02 %>% dplyr::select(-c(8,9))
#omitted the last two column
head(C02)
```

```
## Country Code Calling.Code Year CO2.emission..Tons. Population.2022.
## 1 Afghanistan AF 93 1750 0 41128771
## 2 Afghanistan AF 93 1751 0 41128771
## 3 Afghanistan AF 93 1752 0 41128771
## 4 Afghanistan AF 93 1753 0 41128771
## 5 Afghanistan AF 93 1754 0 41128771
## 6 Afghanistan AF 93 1755 0 41128771
## Area
## 1 652230
## 2 652230
## 3 652230
## 4 652230
## 5 652230
## 6 652230
```

```
#check the dimension of data
dim(C02)
```

```
## [1] 59620 7
```

Please proceed with this C02 (don't forget to omit.na)

Historical CO2 Data analysis and visualization

Aim 1.1 General overview of data, any visual or numerical differences.

Finding the first year in history that countries began emitting CO2

```
#finding and removing NAs
C02 <- na.omit(C02)
View(C02)

#filter emissions 0 and use tidyverse groupby country minimum year
C02_no0<-C02 %>% group_by(min(Year)) %>% filter(CO2.emission..Tons.!=0)

C02_no0 %>%
  group_by(Country) %>%
  slice(which.min(Year))
```

```
## # A tibble: 204 x 8
## # Groups: Country [204]
## Country Code Calling.Code Year CO2.emi~1 Popul~2 Area min(Y~3
```

```
##      <chr>          <chr> <chr>          <int>      <dbl>      <int> <int>      <int>
## 1 Afghanistan    AF      93          1949      14656  4.11e7  6.52e5  1750
## 2 Albania         AL     355         1933       7328  2.84e6  2.87e4  1750
## 3 Algeria         DZ     213         1916       3664  4.49e7  2.38e6  1750
## 4 Andorra         AD     376         1990     406704  7.98e4  4.68e2  1750
## 5 Angola          AO     244         1950     186864  3.56e7  1.25e6  1750
## 6 Antigua and Barbuda AG  1-268      1957       21984  9.38e4  4.42e2  1750
## 7 Argentina       AR     54          1887    1084544  4.55e7  2.78e6  1750
## 8 Armenia         AM     374         1830        42  2.78e6  2.97e4  1750
## 9 Aruba           AW     297         1926     33394  1.08e5  1.8 e2  1750
## 10 Australia      AU     61          1860     278464  2.62e7  7.69e6  1750
## # ... with 194 more rows, and abbreviated variable names
## # 1: CO2.emission..Tons., 2: Population.2022., 3: 'min(Year)'
```

```
CO2_no0 <- data.table(CO2_no0)
```

```
#removing duplicates country names
```

```
CO2_noduplicates <- CO2_no0[!duplicated(CO2_no0$Country),]
CO2_noduplicates
```

```
##      Country Code Calling.Code Year CO2.emission..Tons. Population.2022.
## 1: Afghanistan AF          93 1949          14656          41128771
## 2:  Albania    AL         355 1933           7328          2842321
## 3:  Algeria    DZ         213 1916           3664          44903225
## 4:  Andorra    AD         376 1990         406704           79824
## 5:  Angola     AO         244 1950         186864         35588987
## ---
## 200: Venezuela VE          58 1904           3664         28301696
## 201:  Vietnam  VN          84 1892         212512         98186856
## 202:   Yemen   YE         967 1950         58624         33696614
## 203:   Zambia  ZM         260 1950        2330187         20017675
## 204: Zimbabwe ZW         263 1903        113584         16320537
##      Area min(Year)
## 1: 652230      1750
## 2: 28748      1750
## 3: 2381741     1750
## 4: 468      1750
## 5: 1246700     1750
## ---
## 200: 916445     1750
## 201: 331212     1750
## 202: 527968     1750
## 203: 752612     1750
## 204: 390757     1750
```

```
CO2_no_year_min <- CO2_no0[, list(Year = min(Year)), by = Country]
#new data table to identify first year with non-zero CO2 emission value
CO2_no_year_min[]
```

```
##      Country Year
## 1: Afghanistan 1949
## 2:  Albania    1933
## 3:  Algeria    1916
```

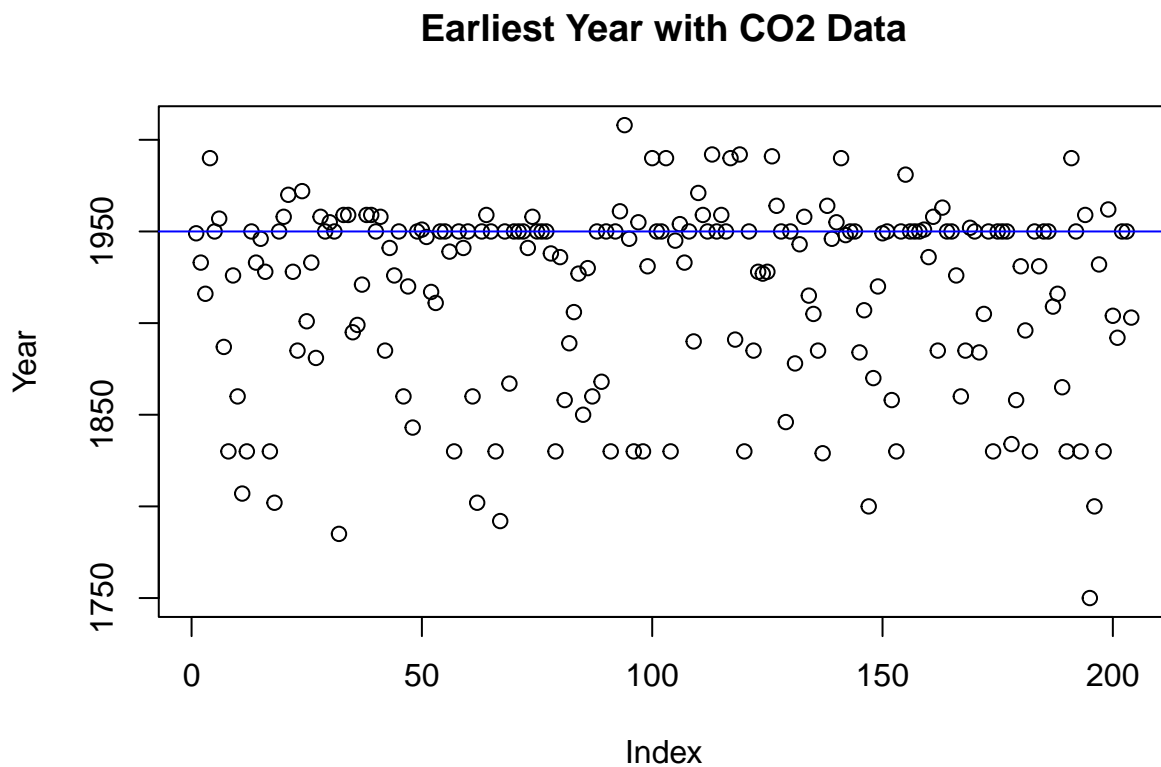
```
## 4: Andorra 1990
## 5: Angola 1950
## ---
## 200: Venezuela 1904
## 201: Vietnam 1892
## 202: Yemen 1950
## 203: Zambia 1950
## 204: Zimbabwe 1903
```

```
summary(CO2_no_year_min$Year)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1750   1885   1944   1919   1950   2008
```

```
#scatter plot of the years with non-zero CO2 value for different countries, with line at 1950
```

```
plot(CO2_no_year_min$Year,main="Earliest Year with CO2 Data",xlab="Index",ylab="Year")+abline(h=1950,col="blue")
```



```
## integer(0)
```

Based on the scatter plot above, in our data, most countries have their first non-zero CO2 value at 1950.

Finding total CO2 emissions per country & CO2 emissions per country per person

```
#finding CO2 emission/population
```

```
CO2total_byCountry <- aggregate(CO2_no0$CO2.emission..Tons., by=list(Category=CO2_no0$Country), FUN=sum)
```

```

#renaming columns
colnames(CO2total_byCountry)[1] <- "Country"
colnames(CO2total_byCountry)[2] <- "CO2.Total"
CO2total_byCountry #checking that the column renaming worked

```

```

##          Country      CO2.Total
## 1      Afghanistan 3.754106e+09
## 2          Albania 9.250246e+09
## 3          Algeria 9.582024e+10
## 4          Andorra 2.360722e+08
## 5          Angola 1.143158e+10
## 6  Antigua and Barbuda 6.376165e+08
## 7          Argentina 2.632189e+11
## 8          Armenia 1.430517e+10
## 9          Aruba 2.357567e+09
## 10         Australia 5.693351e+11
## 11          Austria 3.074275e+11
## 12         Azerbaijan 8.853483e+10
## 13          Bahamas 5.773362e+09
## 14          Bahrain 2.037107e+10
## 15         Bangladesh 2.390055e+10
## 16          Barbados 1.337450e+09
## 17          Belarus 1.809370e+11
## 18          Belgium 7.903086e+11
## 19          Belize 4.207057e+08
## 20          Benin 1.485290e+09
## 21          Bhutan 2.066234e+08
## 22          Bolivia 1.019115e+10
## 23  Bosnia and Herzegovina 2.886810e+10
## 24          Botswana 2.362863e+09
## 25          Brazil 3.713855e+11
## 26          Brunei 1.074748e+10
## 27          Bulgaria 1.331526e+11
## 28         Burkina Faso 9.303006e+08
## 29          Burundi 2.720993e+08
## 30          Cambodia 1.866981e+09
## 31          Cameroon 4.568516e+09
## 32          Canada 1.255512e+12
## 33  Central African Republic 2.957916e+08
## 34          Chad 4.866396e+08
## 35          Chile 8.188228e+10
## 36          China 4.162896e+12
## 37         Colombia 9.086583e+10
## 38          Comoros 9.363528e+07
## 39          Congo 1.625597e+09
## 40         Costa Rica 5.276408e+09
## 41         Cote d'Ivoire 7.587251e+09
## 42          Croatia 3.273520e+10
## 43          Cuba 5.034312e+10
## 44          Curacao 2.561368e+10
## 45          Cyprus 6.666480e+09
## 46          Czechia 6.045514e+11
## 47  Democratic Republic of Congo 7.464739e+09

```

## 48	Denmark	1.823616e+11
## 49	Djibouti	4.713698e+08
## 50	Dominica	9.199167e+07
## 51	Dominican Republic	1.494841e+10
## 52	Ecuador	2.562314e+10
## 53	Egypt	1.264498e+11
## 54	El Salvador	5.108406e+09
## 55	Equatorial Guinea	1.776225e+09
## 56	Eritrea	2.585497e+08
## 57	Estonia	6.332375e+10
## 58	Eswatini	8.891676e+08
## 59	Ethiopia	4.277616e+09
## 60	Fiji	1.275528e+09
## 61	Finland	1.001889e+11
## 62	France	2.238247e+12
## 63	French Guiana	6.071543e+08
## 64	Gabon	6.993919e+09
## 65	Gambia	2.716380e+08
## 66	Georgia	2.531358e+10
## 67	Germany	5.300531e+12
## 68	Ghana	7.399346e+09
## 69	Greece	1.036874e+11
## 70	Greenland	9.928700e+08
## 71	Grenada	1.607098e+08
## 72	Guadeloupe	1.672795e+09
## 73	Guatemala	8.902918e+09
## 74	Guinea	1.924528e+09
## 75	Guyana	2.930467e+09
## 76	Haiti	1.564480e+09
## 77	Honduras	4.863815e+09
## 78	Hong Kong	3.593146e+10
## 79	Hungary	2.273008e+11
## 80	Iceland	4.174625e+09
## 81	India	1.089952e+12
## 82	Indonesia	2.936194e+11
## 83	Iran	4.173757e+11
## 84	Iraq	9.716365e+10
## 85	Ireland	7.438604e+10
## 86	Israel	5.355184e+10
## 87	Italy	8.342234e+11
## 88	Jamaica	1.248806e+10
## 89	Japan	2.051064e+12
## 90	Jordan	1.281551e+10
## 91	Kazakhstan	4.539328e+11
## 92	Kenya	1.081293e+10
## 93	Kiribati	4.856884e+07
## 94	Kosovo	7.308271e+08
## 95	Kuwait	6.284507e+10
## 96	Kyrgyzstan	3.493852e+10
## 97	Laos	1.297428e+09
## 98	Latvia	3.944943e+10
## 99	Lebanon	1.560259e+10
## 100	Lesotho	9.357852e+08
## 101	Liberia	1.693776e+09

## 102	Libya	4.855690e+10
## 103	Liechtenstein	1.034699e+08
## 104	Lithuania	6.299174e+10
## 105	Luxembourg	2.758179e+10
## 106	Macao	1.169537e+09
## 107	Madagascar	2.256616e+09
## 108	Malawi	1.385338e+09
## 109	Malaysia	1.030422e+11
## 110	Maldives	2.781142e+08
## 111	Mali	9.651067e+08
## 112	Malta	2.596434e+09
## 113	Marshall Islands	4.552062e+07
## 114	Martinique	1.971384e+09
## 115	Mauritania	1.457707e+09
## 116	Mauritius	2.135723e+09
## 117	Mayotte	8.141350e+07
## 118	Mexico	5.635544e+11
## 119	Micronesia	5.337956e+07
## 120	Moldova	6.240856e+10
## 121	Mongolia	1.241428e+10
## 122	Montenegro	3.032392e+09
## 123	Morocco	3.494891e+10
## 124	Mozambique	5.181102e+09
## 125	Myanmar	1.545320e+10
## 126	Namibia	9.355316e+08
## 127	Nauru	1.510645e+08
## 128	Nepal	1.965991e+09
## 129	Netherlands	4.948726e+11
## 130	New Caledonia	3.816046e+09
## 131	New Zealand	6.918458e+10
## 132	Nicaragua	4.095118e+09
## 133	Niger	9.251186e+08
## 134	Nigeria	8.799022e+10
## 135	North Korea	1.787941e+11
## 136	North Macedonia	2.145468e+10
## 137	Norway	1.040229e+11
## 138	Oman	1.781582e+10
## 139	Pakistan	9.551340e+10
## 140	Palau	2.788467e+08
## 141	Palestine	7.420809e+08
## 142	Panama	6.685614e+09
## 143	Papua New Guinea	3.285263e+09
## 144	Paraguay	3.313264e+09
## 145	Peru	6.016985e+10
## 146	Philippines	7.225130e+10
## 147	Poland	1.311554e+12
## 148	Portugal	7.955163e+10
## 149	Puerto Rico	2.109365e+07
## 150	Qatar	3.334660e+10
## 151	Reunion	2.138660e+09
## 152	Romania	3.294970e+11
## 153	Russia	4.080739e+12
## 154	Rwanda	6.174393e+08
## 155	Saint Kitts and Nevis	9.972552e+07


```

## 156                Saint Lucia 2.700456e+08
## 157 Saint Vincent and the Grenadines 1.302473e+08
## 158                Samoa 1.412329e+08
## 159                Sao Tome and Principe 6.885642e+07
## 160                Saudi Arabia 2.933864e+11
## 161                Senegal 4.788479e+09
## 162                Serbia 8.595910e+10
## 163                Seychelles 2.295101e+08
## 164                Sierra Leone 1.098976e+09
## 165                Singapore 5.106388e+10
## 166                Sint Maarten (Dutch part) 3.057584e+09
## 167                Slovakia 1.946301e+11
## 168                Slovenia 2.588703e+10
## 169                Solomon Islands 2.125868e+08
## 170                Somalia 8.926861e+08
## 171                South Africa 6.347067e+11
## 172                South Korea 3.337672e+11
## 173                South Sudan 7.882633e+08
## 174                Spain 4.764031e+11
## 175                Sri Lanka 1.028981e+10
## 176                Sudan 8.726657e+09
## 177                Suriname 3.242840e+09
## 178                Sweden 2.394784e+11
## 179                Switzerland 1.281434e+11
## 180                Syria 4.260710e+10
## 181                Taiwan 1.862882e+11
## 182                Tajikistan 2.000847e+10
## 183                Tanzania 4.903444e+09
## 184                Thailand 1.238806e+11
## 185                Togo 1.314020e+09
## 186                Tonga 9.236051e+07
## 187                Trinidad and Tobago 4.497699e+10
## 188                Tunisia 1.884400e+10
## 189                Turkey 2.201051e+11
## 190                Turkmenistan 6.537110e+10
## 191                Tuvalu 4.456110e+06
## 192                Uganda 2.101097e+09
## 193                Ukraine 1.213335e+12
## 194                United Arab Emirates 8.106576e+10
## 195                United Kingdom 6.162082e+12
## 196                United States 1.906877e+13
## 197                Uruguay 1.221006e+10
## 198                Uzbekistan 1.978867e+11
## 199                Vanuatu 1.142344e+08
## 200                Venezuela 2.335395e+11
## 201                Vietnam 7.262503e+10
## 202                Yemen 1.421732e+10
## 203                Zambia 8.811259e+09
## 204                Zimbabwe 2.936469e+10

```

```
#joining columns together to conduct calculations
```

```
C02total_byCountry <- C02total_byCountry %>% inner_join(C02_noduplicates) %>% select("Country", "C02.Total")
```

```
## Joining, by = "Country"
```

#dividing total CO2 emissions per country by the country's population

```
CO2total_byCountry$CO2perPop <- NULL
CO2total_byCountry$CO2perPop <- CO2total_byCountry$CO2.Total/CO2total_byCountry$Population.2022.
CO2total_byCountry
```

##	Country	CO2.Total	Population.2022.	CO2perPop
## 1	Afghanistan	3.754106e+09	41128771	91.27688
## 2	Albania	9.250246e+09	2842321	3254.46901
## 3	Algeria	9.582024e+10	44903225	2133.92786
## 4	Andorra	2.360722e+08	79824	2957.40938
## 5	Angola	1.143158e+10	35588987	321.21110
## 6	Antigua and Barbuda	6.376165e+08	93763	6800.29918
## 7	Argentina	2.632189e+11	45510318	5783.71851
## 8	Armenia	1.430517e+10	2780469	5144.87717
## 9	Aruba	2.357567e+09	107825	21864.75420
## 10	Australia	5.693351e+11	26177413	21749.09751
## 11	Austria	3.074275e+11	8939617	34389.34208
## 12	Azerbaijan	8.853483e+10	10358074	8547.42184
## 13	Bahamas	5.773362e+09	409984	14081.92069
## 14	Bahrain	2.037107e+10	1472233	13836.85123
## 15	Bangladesh	2.390055e+10	171186372	139.61713
## 16	Barbados	1.337450e+09	281635	4748.87638
## 17	Belarus	1.809370e+11	9534954	18976.18109
## 18	Belgium	7.903086e+11	11655930	67803.13847
## 19	Belize	4.207057e+08	405272	1038.08245
## 20	Benin	1.485290e+09	13352864	111.23381
## 21	Bhutan	2.066234e+08	782455	264.07067
## 22	Bolivia	1.019115e+10	12224110	833.69270
## 23	Bosnia and Herzegovina	2.886810e+10	3233526	8927.74655
## 24	Botswana	2.362863e+09	2630296	898.32601
## 25	Brazil	3.713855e+11	215313498	1724.85931
## 26	Brunei	1.074748e+10	449002	23936.38157
## 27	Bulgaria	1.331526e+11	6781953	19633.37493
## 28	Burkina Faso	9.303006e+08	22673762	41.02983
## 29	Burundi	2.720993e+08	12889576	21.11003
## 30	Cambodia	1.866981e+09	16767842	111.34297
## 31	Cameroon	4.568516e+09	27914536	163.66082
## 32	Canada	1.255512e+12	38454327	32649.42101
## 33	Central African Republic	2.957916e+08	5579144	53.01738
## 34	Chad	4.866396e+08	17723315	27.45759
## 35	Chile	8.188228e+10	19603733	4176.87177
## 36	China	4.162896e+12	1425887337	2919.51278
## 37	Colombia	9.086583e+10	51874024	1751.66338
## 38	Comoros	9.363528e+07	836774	111.90033
## 39	Congo	1.625597e+09	5835340	278.57802
## 40	Costa Rica	5.276408e+09	5180829	1018.44851
## 41	Cote d'Ivoire	7.587251e+09	95930038	79.09150
## 42	Croatia	3.273520e+10	4030358	8122.15727
## 43	Cuba	5.034312e+10	11212191	4490.03454
## 44	Curacao	2.561368e+10	165663	154613.13722
## 45	Cyprus	6.666480e+09	1251488	5326.84308
## 46	Czechia	6.045514e+11	10755398	56209.11827
## 47	Democratic Republic of Congo	7.464739e+09	95930038	77.81441

## 48	Denmark	1.823616e+11	5882261	31001.94924
## 49	Djibouti	4.713698e+08	1120849	420.54710
## 50	Dominica	9.199167e+07	72737	1264.71636
## 51	Dominican Republic	1.494841e+10	11228821	1331.25350
## 52	Ecuador	2.562314e+10	18001000	1423.42850
## 53	Egypt	1.264498e+11	110990103	1139.28898
## 54	El Salvador	5.108406e+09	6336392	806.20113
## 55	Equatorial Guinea	1.776225e+09	1674908	1060.49079
## 56	Eritrea	2.585497e+08	3684032	70.18116
## 57	Estonia	6.332375e+10	1326062	47753.23364
## 58	Eswatini	8.891676e+08	1201670	739.94322
## 59	Ethiopia	4.277616e+09	123379924	34.67027
## 60	Fiji	1.275528e+09	929766	1371.88029
## 61	Finland	1.001889e+11	5540745	18082.21241
## 62	France	2.238247e+12	64626628	34633.51398
## 63	French Guiana	6.071543e+08	304557	1993.56544
## 64	Gabon	6.993919e+09	2388992	2927.56070
## 65	Gambia	2.716380e+08	2705992	100.38387
## 66	Georgia	2.531358e+10	3744385	6760.41120
## 67	Germany	5.300531e+12	83369843	63578.51556
## 68	Ghana	7.399346e+09	33475870	221.03522
## 69	Greece	1.036874e+11	10384971	9984.37353
## 70	Greenland	9.928700e+08	56466	17583.50129
## 71	Grenada	1.607098e+08	125438	1281.18892
## 72	Guadeloupe	1.672795e+09	395752	4226.87802
## 73	Guatemala	8.902918e+09	17843908	498.93322
## 74	Guinea	1.924528e+09	13859341	138.86142
## 75	Guyana	2.930467e+09	808726	3623.56028
## 76	Haiti	1.564480e+09	11584996	135.04364
## 77	Honduras	4.863815e+09	10432860	466.20154
## 78	Hong Kong	3.593146e+10	7488865	4797.98451
## 79	Hungary	2.273008e+11	9967308	22804.63767
## 80	Iceland	4.174625e+09	372899	11195.05680
## 81	India	1.089952e+12	1417173173	769.10266
## 82	Indonesia	2.936194e+11	275501339	1065.76379
## 83	Iran	4.173757e+11	88550570	4713.41577
## 84	Iraq	9.716365e+10	44496122	2183.64319
## 85	Ireland	7.438604e+10	5023109	14808.76544
## 86	Israel	5.355184e+10	9038309	5924.98426
## 87	Italy	8.342234e+11	59037474	14130.40539
## 88	Jamaica	1.248806e+10	2827377	4416.83733
## 89	Japan	2.051064e+12	123951692	16547.28791
## 90	Jordan	1.281551e+10	11285869	1135.53615
## 91	Kazakhstan	4.539328e+11	19397998	23401.01517
## 92	Kenya	1.081293e+10	54027487	200.13763
## 93	Kiribati	4.856884e+07	131232	370.09905
## 94	Kosovo	7.308271e+08	1802250	405.50819
## 95	Kuwait	6.284507e+10	4268873	14721.70122
## 96	Kyrgyzstan	3.493852e+10	6630623	5269.26585
## 97	Laos	1.297428e+09	7529475	172.31326
## 98	Latvia	3.944943e+10	1850651	21316.51367
## 99	Lebanon	1.560259e+10	5489739	2842.13707
## 100	Lesotho	9.357852e+08	2305825	405.83532
## 101	Liberia	1.693776e+09	5302681	319.41874

## 102	Libya	4.855690e+10	6812341	7127.78454
## 103	Liechtenstein	1.034699e+08	39327	2631.01454
## 104	Lithuania	6.299174e+10	2750055	22905.63096
## 105	Luxembourg	2.758179e+10	647599	42590.84014
## 106	Macao	1.169537e+09	670052	1745.44266
## 107	Madagascar	2.256616e+09	29611714	76.20688
## 108	Malawi	1.385338e+09	20405317	67.89103
## 109	Malaysia	1.030422e+11	33938221	3036.17044
## 110	Maldives	2.781142e+08	523787	530.96816
## 111	Mali	9.651067e+08	22593590	42.71595
## 112	Malta	2.596434e+09	533286	4868.74661
## 113	Marshall Islands	4.552062e+07	41569	1095.06175
## 114	Martinique	1.971384e+09	367507	5364.20751
## 115	Mauritania	1.457707e+09	4736139	307.78380
## 116	Mauritius	2.135723e+09	1299469	1643.53507
## 117	Mayotte	8.141350e+07	326101	249.65732
## 118	Mexico	5.635544e+11	127504125	4419.89154
## 119	Micronesia	5.337956e+07	114164	467.56909
## 120	Moldova	6.240856e+10	3272996	19067.71588
## 121	Mongolia	1.241428e+10	3398366	3653.01578
## 122	Montenegro	3.032392e+09	627082	4835.71903
## 123	Morocco	3.494891e+10	37457971	933.01664
## 124	Mozambique	5.181102e+09	32969518	157.14825
## 125	Myanmar	1.545320e+10	54179306	285.22329
## 126	Namibia	9.355316e+08	2567012	364.44378
## 127	Nauru	1.510645e+08	12668	11924.89304
## 128	Nepal	1.965991e+09	30547580	64.35831
## 129	Netherlands	4.948726e+11	17564014	28175.37243
## 130	New Caledonia	3.816046e+09	291913	13072.54726
## 131	New Zealand	6.918458e+10	5185288	13342.47511
## 132	Nicaragua	4.095118e+09	6948392	589.36198
## 133	Niger	9.251186e+08	26207977	35.29912
## 134	Nigeria	8.799022e+10	218541212	402.62529
## 135	North Korea	1.787941e+11	26069416	6858.38430
## 136	North Macedonia	2.145468e+10	2093599	10247.75238
## 137	Norway	1.040229e+11	5434319	19141.84620
## 138	Oman	1.781582e+10	4576298	3893.06290
## 139	Pakistan	9.551340e+10	235824862	405.01837
## 140	Palau	2.788467e+08	18055	15444.29022
## 141	Palestine	7.420809e+08	5250072	141.34681
## 142	Panama	6.685614e+09	4408581	1516.50029
## 143	Papua New Guinea	3.285263e+09	10142619	323.90677
## 144	Paraguay	3.313264e+09	6780744	488.62835
## 145	Peru	6.016985e+10	34049588	1767.12425
## 146	Philippines	7.225130e+10	115559009	625.23295
## 147	Poland	1.311554e+12	39857145	32906.36461
## 148	Portugal	7.955163e+10	10270865	7745.36844
## 149	Puerto Rico	2.109365e+07	3252407	6.48555
## 150	Qatar	3.334660e+10	2695122	12372.94745
## 151	Reunion	2.138660e+09	974052	2195.63209
## 152	Romania	3.294970e+11	19659267	16760.39336
## 153	Russia	4.080739e+12	144713314	28198.77848
## 154	Rwanda	6.174393e+08	13776698	44.81765
## 155	Saint Kitts and Nevis	9.972552e+07	47657	2092.56806

## 156	Saint Lucia	2.700456e+08	179857	1501.44622
## 157	Saint Vincent and the Grenadines	1.302473e+08	103948	1253.00450
## 158	Samoa	1.412329e+08	222382	635.09138
## 159	Sao Tome and Principe	6.885642e+07	227380	302.82532
## 160	Saudi Arabia	2.933864e+11	36408820	8058.11261
## 161	Senegal	4.788479e+09	17316449	276.52777
## 162	Serbia	8.595910e+10	7221365	11903.44211
## 163	Seychelles	2.295101e+08	107118	2142.59121
## 164	Sierra Leone	1.098976e+09	8605718	127.70296
## 165	Singapore	5.106388e+10	5975689	8545.27034
## 166	Sint Maarten (Dutch part)	3.057584e+09	44015	69466.86402
## 167	Slovakia	1.946301e+11	5643453	34487.76021
## 168	Slovenia	2.588703e+10	2119844	12211.76174
## 169	Solomon Islands	2.125868e+08	724273	293.51746
## 170	Somalia	8.926861e+08	17597511	50.72797
## 171	South Africa	6.347067e+11	59893885	10597.18638
## 172	South Korea	3.337672e+11	51815810	6441.41619
## 173	South Sudan	7.882633e+08	10913164	72.23050
## 174	Spain	4.764031e+11	47558630	10017.17504
## 175	Sri Lanka	1.028981e+10	21832143	471.31478
## 176	Sudan	8.726657e+09	46874204	186.17185
## 177	Suriname	3.242840e+09	618040	5246.97468
## 178	Sweden	2.394784e+11	10549347	22700.78219
## 179	Switzerland	1.281434e+11	8740472	14660.92604
## 180	Syria	4.260710e+10	22125249	1925.72280
## 181	Taiwan	1.862882e+11	23893394	7796.63988
## 182	Tajikistan	2.000847e+10	9952787	2010.33831
## 183	Tanzania	4.903444e+09	65497748	74.86432
## 184	Thailand	1.238806e+11	71697030	1727.83503
## 185	Togo	1.314020e+09	8848699	148.49862
## 186	Tonga	9.236051e+07	106858	864.32939
## 187	Trinidad and Tobago	4.497699e+10	1531044	29376.67870
## 188	Tunisia	1.884400e+10	12356117	1525.07447
## 189	Turkey	2.201051e+11	85341241	2579.11750
## 190	Turkmenistan	6.537110e+10	6430770	10165.36126
## 191	Tuvalu	4.456110e+06	11312	393.92769
## 192	Uganda	2.101097e+09	47249585	44.46804
## 193	Ukraine	1.213335e+12	39701739	30561.26580
## 194	United Arab Emirates	8.106576e+10	9441129	8586.44735
## 195	United Kingdom	6.162082e+12	67508936	91278.01797
## 196	United States	1.906877e+13	338289857	56368.13240
## 197	Uruguay	1.221006e+10	3422794	3567.27912
## 198	Uzbekistan	1.978867e+11	34627652	5714.70143
## 199	Vanuatu	1.142344e+08	326740	349.61854
## 200	Venezuela	2.335395e+11	28301696	8251.78637
## 201	Vietnam	7.262503e+10	98186856	739.66144
## 202	Yemen	1.421732e+10	33696614	421.92130
## 203	Zambia	8.811259e+09	20017675	440.17392
## 204	Zimbabwe	2.936469e+10	16320537	1799.24804

#renaming column names

```
colnames(CO2total_byCountry)[2] <- "CO2.Total.Per.Country"
colnames(CO2total_byCountry)[4] <- "CO2.per.Person.per.Country"
```

```
summary(CO2total_byCountry)
```

```
##      Country      CO2.Total.Per.Country Population.2022.
## Length:204      Min.      :4.456e+06      Min.      :1.131e+04
## Class :character 1st Qu.:1.373e+09      1st Qu.:1.516e+06
## Mode  :character Median :1.078e+10      Median :8.068e+06
##                      Mean  :3.024e+11      Mean  :3.938e+07
##                      3rd Qu.:8.813e+10      3rd Qu.:2.863e+07
##                      Max.   :1.907e+13      Max.   :1.426e+09
## CO2.per.Person.per.Country
## Min.      :      6.49
## 1st Qu.:    400.45
## Median :   2113.25
## Mean  :   9011.85
## 3rd Qu.:  10185.96
## Max.    : 154613.14
```

```
CO2total_byCountry
```

```
##      Country CO2.Total.Per.Country Population.2022.
## 1      Afghanistan      3.754106e+09      41128771
## 2      Albania      9.250246e+09      2842321
## 3      Algeria      9.582024e+10      44903225
## 4      Andorra      2.360722e+08      79824
## 5      Angola      1.143158e+10      35588987
## 6      Antigua and Barbuda      6.376165e+08      93763
## 7      Argentina      2.632189e+11      45510318
## 8      Armenia      1.430517e+10      2780469
## 9      Aruba      2.357567e+09      107825
## 10     Australia      5.693351e+11      26177413
## 11     Austria      3.074275e+11      8939617
## 12     Azerbaijan      8.853483e+10      10358074
## 13     Bahamas      5.773362e+09      409984
## 14     Bahrain      2.037107e+10      1472233
## 15     Bangladesh      2.390055e+10      171186372
## 16     Barbados      1.337450e+09      281635
## 17     Belarus      1.809370e+11      9534954
## 18     Belgium      7.903086e+11      11655930
## 19     Belize      4.207057e+08      405272
## 20     Benin      1.485290e+09      13352864
## 21     Bhutan      2.066234e+08      782455
## 22     Bolivia      1.019115e+10      12224110
## 23     Bosnia and Herzegovina      2.886810e+10      3233526
## 24     Botswana      2.362863e+09      2630296
## 25     Brazil      3.713855e+11      215313498
## 26     Brunei      1.074748e+10      449002
## 27     Bulgaria      1.331526e+11      6781953
## 28     Burkina Faso      9.303006e+08      22673762
## 29     Burundi      2.720993e+08      12889576
## 30     Cambodia      1.866981e+09      16767842
## 31     Cameroon      4.568516e+09      27914536
## 32     Canada      1.255512e+12      38454327
```

## 33	Central African Republic	2.957916e+08	5579144
## 34	Chad	4.866396e+08	17723315
## 35	Chile	8.188228e+10	19603733
## 36	China	4.162896e+12	1425887337
## 37	Colombia	9.086583e+10	51874024
## 38	Comoros	9.363528e+07	836774
## 39	Congo	1.625597e+09	5835340
## 40	Costa Rica	5.276408e+09	5180829
## 41	Cote d'Ivoire	7.587251e+09	95930038
## 42	Croatia	3.273520e+10	4030358
## 43	Cuba	5.034312e+10	11212191
## 44	Curacao	2.561368e+10	165663
## 45	Cyprus	6.666480e+09	1251488
## 46	Czechia	6.045514e+11	10755398
## 47	Democratic Republic of Congo	7.464739e+09	95930038
## 48	Denmark	1.823616e+11	5882261
## 49	Djibouti	4.713698e+08	1120849
## 50	Dominica	9.199167e+07	72737
## 51	Dominican Republic	1.494841e+10	11228821
## 52	Ecuador	2.562314e+10	18001000
## 53	Egypt	1.264498e+11	110990103
## 54	El Salvador	5.108406e+09	6336392
## 55	Equatorial Guinea	1.776225e+09	1674908
## 56	Eritrea	2.585497e+08	3684032
## 57	Estonia	6.332375e+10	1326062
## 58	Eswatini	8.891676e+08	1201670
## 59	Ethiopia	4.277616e+09	123379924
## 60	Fiji	1.275528e+09	929766
## 61	Finland	1.001889e+11	5540745
## 62	France	2.238247e+12	64626628
## 63	French Guiana	6.071543e+08	304557
## 64	Gabon	6.993919e+09	2388992
## 65	Gambia	2.716380e+08	2705992
## 66	Georgia	2.531358e+10	3744385
## 67	Germany	5.300531e+12	83369843
## 68	Ghana	7.399346e+09	33475870
## 69	Greece	1.036874e+11	10384971
## 70	Greenland	9.928700e+08	56466
## 71	Grenada	1.607098e+08	125438
## 72	Guadeloupe	1.672795e+09	395752
## 73	Guatemala	8.902918e+09	17843908
## 74	Guinea	1.924528e+09	13859341
## 75	Guyana	2.930467e+09	808726
## 76	Haiti	1.564480e+09	11584996
## 77	Honduras	4.863815e+09	10432860
## 78	Hong Kong	3.593146e+10	7488865
## 79	Hungary	2.273008e+11	9967308
## 80	Iceland	4.174625e+09	372899
## 81	India	1.089952e+12	1417173173
## 82	Indonesia	2.936194e+11	275501339
## 83	Iran	4.173757e+11	88550570
## 84	Iraq	9.716365e+10	44496122
## 85	Ireland	7.438604e+10	5023109
## 86	Israel	5.355184e+10	9038309

## 87	Italy	8.342234e+11	59037474
## 88	Jamaica	1.248806e+10	2827377
## 89	Japan	2.051064e+12	123951692
## 90	Jordan	1.281551e+10	11285869
## 91	Kazakhstan	4.539328e+11	19397998
## 92	Kenya	1.081293e+10	54027487
## 93	Kiribati	4.856884e+07	131232
## 94	Kosovo	7.308271e+08	1802250
## 95	Kuwait	6.284507e+10	4268873
## 96	Kyrgyzstan	3.493852e+10	6630623
## 97	Laos	1.297428e+09	7529475
## 98	Latvia	3.944943e+10	1850651
## 99	Lebanon	1.560259e+10	5489739
## 100	Lesotho	9.357852e+08	2305825
## 101	Liberia	1.693776e+09	5302681
## 102	Libya	4.855690e+10	6812341
## 103	Liechtenstein	1.034699e+08	39327
## 104	Lithuania	6.299174e+10	2750055
## 105	Luxembourg	2.758179e+10	647599
## 106	Macao	1.169537e+09	670052
## 107	Madagascar	2.256616e+09	29611714
## 108	Malawi	1.385338e+09	20405317
## 109	Malaysia	1.030422e+11	33938221
## 110	Maldives	2.781142e+08	523787
## 111	Mali	9.651067e+08	22593590
## 112	Malta	2.596434e+09	533286
## 113	Marshall Islands	4.552062e+07	41569
## 114	Martinique	1.971384e+09	367507
## 115	Mauritania	1.457707e+09	4736139
## 116	Mauritius	2.135723e+09	1299469
## 117	Mayotte	8.141350e+07	326101
## 118	Mexico	5.635544e+11	127504125
## 119	Micronesia	5.337956e+07	114164
## 120	Moldova	6.240856e+10	3272996
## 121	Mongolia	1.241428e+10	3398366
## 122	Montenegro	3.032392e+09	627082
## 123	Morocco	3.494891e+10	37457971
## 124	Mozambique	5.181102e+09	32969518
## 125	Myanmar	1.545320e+10	54179306
## 126	Namibia	9.355316e+08	2567012
## 127	Nauru	1.510645e+08	12668
## 128	Nepal	1.965991e+09	30547580
## 129	Netherlands	4.948726e+11	17564014
## 130	New Caledonia	3.816046e+09	291913
## 131	New Zealand	6.918458e+10	5185288
## 132	Nicaragua	4.095118e+09	6948392
## 133	Niger	9.251186e+08	26207977
## 134	Nigeria	8.799022e+10	218541212
## 135	North Korea	1.787941e+11	26069416
## 136	North Macedonia	2.145468e+10	2093599
## 137	Norway	1.040229e+11	5434319
## 138	Oman	1.781582e+10	4576298
## 139	Pakistan	9.551340e+10	235824862
## 140	Palau	2.788467e+08	18055

## 141	Palestine	7.420809e+08	5250072
## 142	Panama	6.685614e+09	4408581
## 143	Papua New Guinea	3.285263e+09	10142619
## 144	Paraguay	3.313264e+09	6780744
## 145	Peru	6.016985e+10	34049588
## 146	Philippines	7.225130e+10	115559009
## 147	Poland	1.311554e+12	39857145
## 148	Portugal	7.955163e+10	10270865
## 149	Puerto Rico	2.109365e+07	3252407
## 150	Qatar	3.334660e+10	2695122
## 151	Reunion	2.138660e+09	974052
## 152	Romania	3.294970e+11	19659267
## 153	Russia	4.080739e+12	144713314
## 154	Rwanda	6.174393e+08	13776698
## 155	Saint Kitts and Nevis	9.972552e+07	47657
## 156	Saint Lucia	2.700456e+08	179857
## 157	Saint Vincent and the Grenadines	1.302473e+08	103948
## 158	Samoa	1.412329e+08	222382
## 159	Sao Tome and Principe	6.885642e+07	227380
## 160	Saudi Arabia	2.933864e+11	36408820
## 161	Senegal	4.788479e+09	17316449
## 162	Serbia	8.595910e+10	7221365
## 163	Seychelles	2.295101e+08	107118
## 164	Sierra Leone	1.098976e+09	8605718
## 165	Singapore	5.106388e+10	5975689
## 166	Sint Maarten (Dutch part)	3.057584e+09	44015
## 167	Slovakia	1.946301e+11	5643453
## 168	Slovenia	2.588703e+10	2119844
## 169	Solomon Islands	2.125868e+08	724273
## 170	Somalia	8.926861e+08	17597511
## 171	South Africa	6.347067e+11	59893885
## 172	South Korea	3.337672e+11	51815810
## 173	South Sudan	7.882633e+08	10913164
## 174	Spain	4.764031e+11	47558630
## 175	Sri Lanka	1.028981e+10	21832143
## 176	Sudan	8.726657e+09	46874204
## 177	Suriname	3.242840e+09	618040
## 178	Sweden	2.394784e+11	10549347
## 179	Switzerland	1.281434e+11	8740472
## 180	Syria	4.260710e+10	22125249
## 181	Taiwan	1.862882e+11	23893394
## 182	Tajikistan	2.000847e+10	9952787
## 183	Tanzania	4.903444e+09	65497748
## 184	Thailand	1.238806e+11	71697030
## 185	Togo	1.314020e+09	8848699
## 186	Tonga	9.236051e+07	106858
## 187	Trinidad and Tobago	4.497699e+10	1531044
## 188	Tunisia	1.884400e+10	12356117
## 189	Turkey	2.201051e+11	85341241
## 190	Turkmenistan	6.537110e+10	6430770
## 191	Tuvalu	4.456110e+06	11312
## 192	Uganda	2.101097e+09	47249585
## 193	Ukraine	1.213335e+12	39701739
## 194	United Arab Emirates	8.106576e+10	9441129

## 195	United Kingdom	6.162082e+12	67508936
## 196	United States	1.906877e+13	338289857
## 197	Uruguay	1.221006e+10	3422794
## 198	Uzbekistan	1.978867e+11	34627652
## 199	Vanuatu	1.142344e+08	326740
## 200	Venezuela	2.335395e+11	28301696
## 201	Vietnam	7.262503e+10	98186856
## 202	Yemen	1.421732e+10	33696614
## 203	Zambia	8.811259e+09	20017675
## 204	Zimbabwe	2.936469e+10	16320537
##	C02.per.Person.per.Country		
## 1	91.27688		
## 2	3254.46901		
## 3	2133.92786		
## 4	2957.40938		
## 5	321.21110		
## 6	6800.29918		
## 7	5783.71851		
## 8	5144.87717		
## 9	21864.75420		
## 10	21749.09751		
## 11	34389.34208		
## 12	8547.42184		
## 13	14081.92069		
## 14	13836.85123		
## 15	139.61713		
## 16	4748.87638		
## 17	18976.18109		
## 18	67803.13847		
## 19	1038.08245		
## 20	111.23381		
## 21	264.07067		
## 22	833.69270		
## 23	8927.74655		
## 24	898.32601		
## 25	1724.85931		
## 26	23936.38157		
## 27	19633.37493		
## 28	41.02983		
## 29	21.11003		
## 30	111.34297		
## 31	163.66082		
## 32	32649.42101		
## 33	53.01738		
## 34	27.45759		
## 35	4176.87177		
## 36	2919.51278		
## 37	1751.66338		
## 38	111.90033		
## 39	278.57802		
## 40	1018.44851		
## 41	79.09150		
## 42	8122.15727		
## 43	4490.03454		

## 44	154613.13722
## 45	5326.84308
## 46	56209.11827
## 47	77.81441
## 48	31001.94924
## 49	420.54710
## 50	1264.71636
## 51	1331.25350
## 52	1423.42850
## 53	1139.28898
## 54	806.20113
## 55	1060.49079
## 56	70.18116
## 57	47753.23364
## 58	739.94322
## 59	34.67027
## 60	1371.88029
## 61	18082.21241
## 62	34633.51398
## 63	1993.56544
## 64	2927.56070
## 65	100.38387
## 66	6760.41120
## 67	63578.51556
## 68	221.03522
## 69	9984.37353
## 70	17583.50129
## 71	1281.18892
## 72	4226.87802
## 73	498.93322
## 74	138.86142
## 75	3623.56028
## 76	135.04364
## 77	466.20154
## 78	4797.98451
## 79	22804.63767
## 80	11195.05680
## 81	769.10266
## 82	1065.76379
## 83	4713.41577
## 84	2183.64319
## 85	14808.76544
## 86	5924.98426
## 87	14130.40539
## 88	4416.83733
## 89	16547.28791
## 90	1135.53615
## 91	23401.01517
## 92	200.13763
## 93	370.09905
## 94	405.50819
## 95	14721.70122
## 96	5269.26585
## 97	172.31326

## 98	21316.51367
## 99	2842.13707
## 100	405.83532
## 101	319.41874
## 102	7127.78454
## 103	2631.01454
## 104	22905.63096
## 105	42590.84014
## 106	1745.44266
## 107	76.20688
## 108	67.89103
## 109	3036.17044
## 110	530.96816
## 111	42.71595
## 112	4868.74661
## 113	1095.06175
## 114	5364.20751
## 115	307.78380
## 116	1643.53507
## 117	249.65732
## 118	4419.89154
## 119	467.56909
## 120	19067.71588
## 121	3653.01578
## 122	4835.71903
## 123	933.01664
## 124	157.14825
## 125	285.22329
## 126	364.44378
## 127	11924.89304
## 128	64.35831
## 129	28175.37243
## 130	13072.54726
## 131	13342.47511
## 132	589.36198
## 133	35.29912
## 134	402.62529
## 135	6858.38430
## 136	10247.75238
## 137	19141.84620
## 138	3893.06290
## 139	405.01837
## 140	15444.29022
## 141	141.34681
## 142	1516.50029
## 143	323.90677
## 144	488.62835
## 145	1767.12425
## 146	625.23295
## 147	32906.36461
## 148	7745.36844
## 149	6.48555
## 150	12372.94745
## 151	2195.63209

## 152	16760.39336
## 153	28198.77848
## 154	44.81765
## 155	2092.56806
## 156	1501.44622
## 157	1253.00450
## 158	635.09138
## 159	302.82532
## 160	8058.11261
## 161	276.52777
## 162	11903.44211
## 163	2142.59121
## 164	127.70296
## 165	8545.27034
## 166	69466.86402
## 167	34487.76021
## 168	12211.76174
## 169	293.51746
## 170	50.72797
## 171	10597.18638
## 172	6441.41619
## 173	72.23050
## 174	10017.17504
## 175	471.31478
## 176	186.17185
## 177	5246.97468
## 178	22700.78219
## 179	14660.92604
## 180	1925.72280
## 181	7796.63988
## 182	2010.33831
## 183	74.86432
## 184	1727.83503
## 185	148.49862
## 186	864.32939
## 187	29376.67870
## 188	1525.07447
## 189	2579.11750
## 190	10165.36126
## 191	393.92769
## 192	44.46804
## 193	30561.26580
## 194	8586.44735
## 195	91278.01797
## 196	56368.13240
## 197	3567.27912
## 198	5714.70143
## 199	349.61854
## 200	8251.78637
## 201	739.66144
## 202	421.92130
## 203	440.17392
## 204	1799.24804

The results of `summary()` on the `CO2total_byCountry` data set indicate that the average total CO2 emissions in the data set is $3.024e+11$ tons.

Aim 1.2 A: To visualize the cumulative CO2 emissions by country and continent

```
CO2total_byCountry$continent <- countrycode(sourcevar = CO2total_byCountry[, "Country"],
                                             origin = "country.name",
                                             destination = "continent")
```

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some va
```

```
#summing the total CO2 emissions by geographic region: Oceania, Africa, Asia, Americas, and Europe (Ant
CO2total_byContinent <- aggregate(CO2total_byCountry$CO2.Total.Per.Country, by=list(Category=CO2total_by
CO2total_byContinent
```

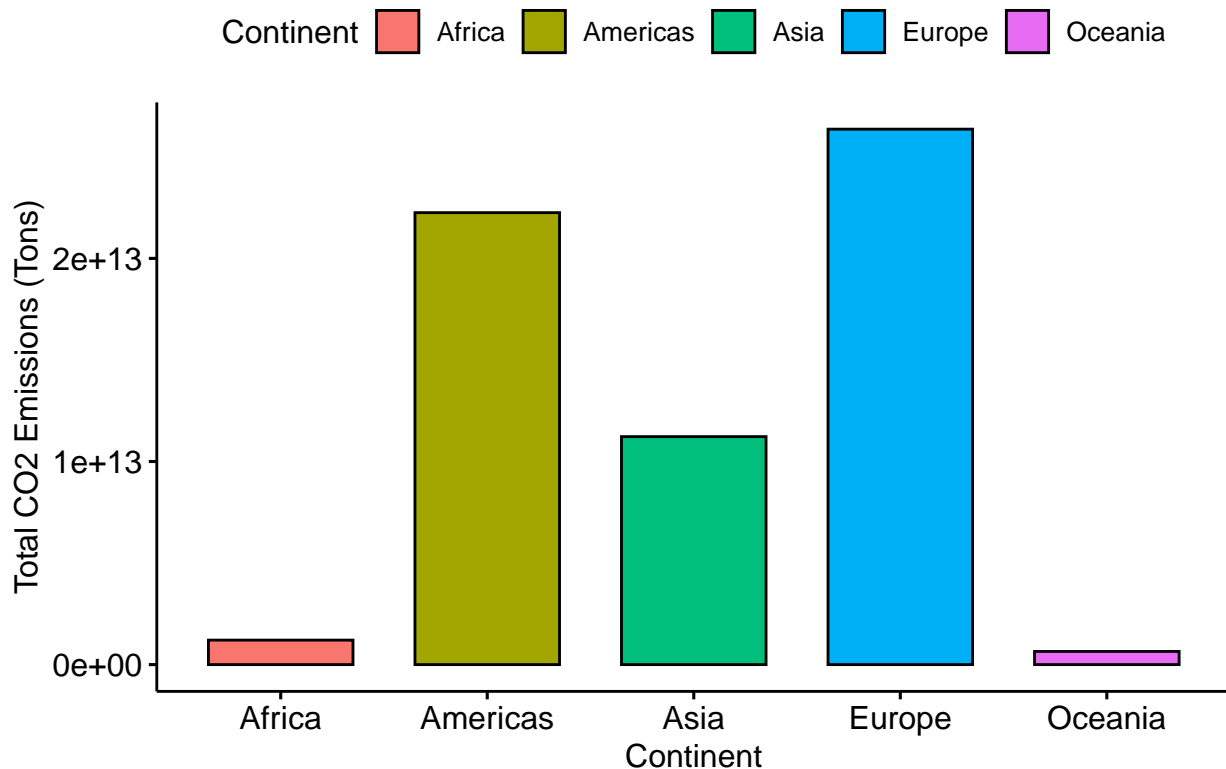
```
##   Category          x
## 1  Africa 1.204794e+12
## 2 Americas 2.225088e+13
## 3   Asia 1.122400e+13
## 4  Europe 2.636167e+13
## 5 Oceania 6.479854e+11
```

```
colnames(CO2total_byContinent) <- c("Geographic_Region", "Total_CO2_Emissions")
```

```
CO2total_byContinent <- as.data.frame(CO2total_byContinent)
```

```
continent_barplot <- ggbarplot(data=CO2total_byContinent, x="Geographic_Region", y="Total_CO2_Emissions",
continent_barplot <- continent_barplot + labs(x="Continent",
y="Total CO2 Emissions (Tons)", title="CO2 Totals by Continent", fill="Continent")
continent_barplot
```

CO2 Totals by Continent



Create a map visualizing the amount of CO2 emission by country

To visualize the CO2 emissions by country using a world map, load in ggplot2's map data via the code below. This data set contains longitude, latitude, and country codes to plot the world map. The cleaned CO2 data set will merge with the world data set by country name.

```
world <- map_data("world")  
#table(world) #world data codes country names differently. Change prior to joining  
#Add the data you want to map countries by to world  
#In this example, I add lengths of country names plus some offset
```

Creating a barplot & world map to display CO2 emissions per country

The world data set codes some country names differently. So, prior to joining the CO2 and world datasets, match the country names for "United States" and "Democratic Republic of the Congo" to the way the world dataset codes these countries.

```
which(CO2total_byCountry$Country=="United States")
```

```
## [1] 196
```

```
which(CO2total_byCountry$Country=="Democratic Republic of Congo")
```

```
## [1] 47
```

```
CO2total_byCountry$Country[196] <- "USA" #renaming "United States" to "USA", to ensure uniformity in th
CO2total_byCountry$Country[47] <- "Democratic Republic of the Congo"
CO2total_byCountry[196,]
```

```
##      Country CO2.Total.Per.Country Population.2022. CO2.per.Person.per.Country
## 196      USA      1.906877e+13      338289857      56368.13
##      continent
## 196 Americas
```

```
#renaming "region" in the data we are importing to "country"
world <- inner_join(world,CO2total_byCountry,by=c("region"="Country"))
#remove duplicates
world <- world[!duplicated(world),]
world$CO2.Total.Per.Country[world$region=="USA"]
```

```
##      [1] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##      [6] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [11] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [16] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [21] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [26] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [31] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [36] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [41] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [46] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [51] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [56] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [61] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [66] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [71] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [76] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [81] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [86] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [91] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##     [96] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [101] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [106] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [111] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [116] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [121] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [126] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [131] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [136] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [141] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [146] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [151] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [156] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [161] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [166] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [171] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [176] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
##    [181] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

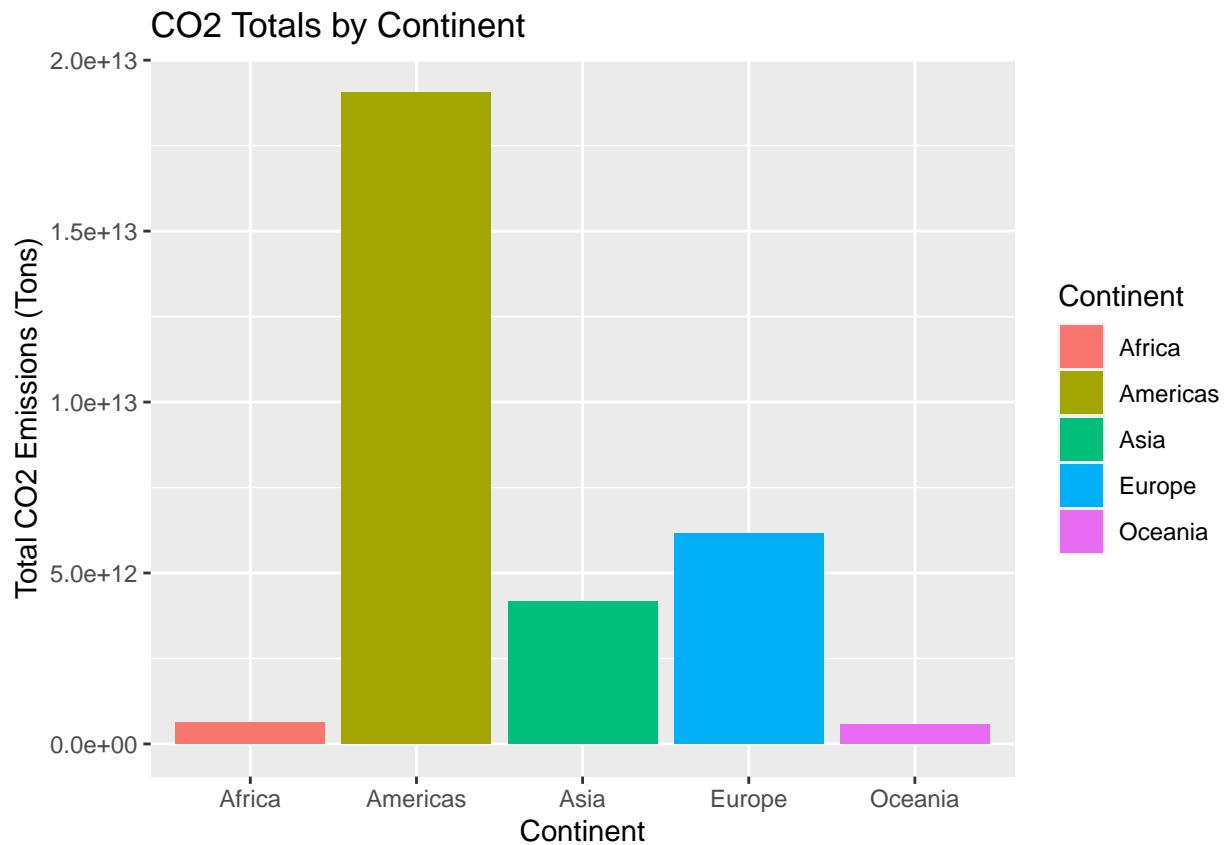
[illegible]

[illegible]


```
## [5586] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5591] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5596] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5601] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5606] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5611] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5616] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5621] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5626] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5631] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5636] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5641] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5646] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5651] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5656] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5661] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5666] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5671] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5676] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5681] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5686] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5691] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5696] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5701] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5706] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5711] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5716] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5721] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5726] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5731] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5736] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5741] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5746] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
## [5751] 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13 1.906877e+13
```

```
#barplot of CO2 total emissions in world map
```

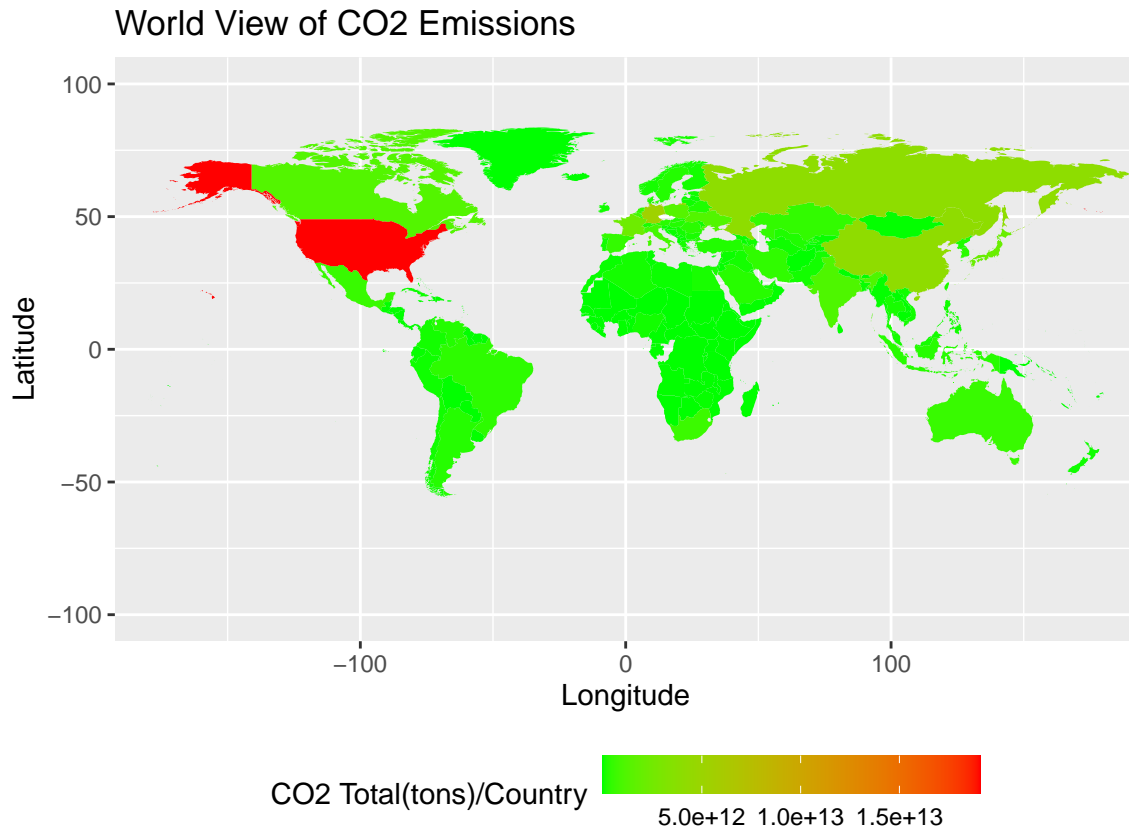
```
continent_barplot <- ggplot(data=subset(CO2total_byCountry,!is.na(continent)),aes(x=continent,y=CO2.Tot
continent_barplot <- continent_barplot + labs(x="Continent",y="Total CO2 Emissions (Tons)",title="CO2 T
continent_barplot
```



```
#graphically presenting CO2 emission in a world map
gg <- ggplot() + theme(legend.position="bottom",legend.key.size = unit(0.5, "cm"),legend.key.width = unit(1, "cm"))
gg <- gg + geom_map(data=world, map=world, aes(map_id=region,x=long, y=lat, fill=CO2.Total.Per.Country))
```

```
## Warning in geom_map(data = world, map = world, aes(map_id = region, x = long, y = lat, fill = CO2.Total.Per.Country)) :
## Ignoring unknown aesthetics: x and y
```

```
#changing features of world map (color, axes scales, adding a title, etc)
gg <- gg + scale_fill_gradient(low = "green", high = "red", guide = "colourbar",aesthetics = "fill") +
gg <- gg + coord_equal() + ggtitle("World View of CO2 Emissions")
gg
```



Making a world map just for 2019 data:

To see if there's a difference between historical cumulative CO2 and 2019 CO2 emissions, we made a world map just for 2019 emissions data.

```
cleanedC02 <- C02
cleanedC02_2019<-cleanedC02[cleanedC02$Year == "2019",] #below code is same as cumulative analysis
C02total_byCountry_2019 <- aggregate(cleanedC02_2019$C02.emission..Tons., by=list(Category=cleanedC02_2019$Country), FUN=sum)
colnames(C02total_byCountry_2019)[1] <- "Country"
colnames(C02total_byCountry_2019)[2] <- "C02.Total"
C02total_byCountry_2019<- C02total_byCountry_2019 %>%
  inner_join(cleanedC02_2019) %>%
  select("Country", "C02.Total", "Population.2022.")
which(C02total_byCountry_2019$Country=="United States")
```

```
## [1] 196
```

```
which(C02total_byCountry_2019$Country=="Democratic Republic of Congo")
```

```
## [1] 47
```

```
C02total_byCountry_2019$Country[196] <- "USA"#coding USA differently #because world data has it as USA,
C02total_byCountry_2019$Country[47] <- "Democratic Republic of the Congo"
world2 <- map_data("world")
world2 <- inner_join(world2,C02total_byCountry_2019,by=c("region"="Country"))
```

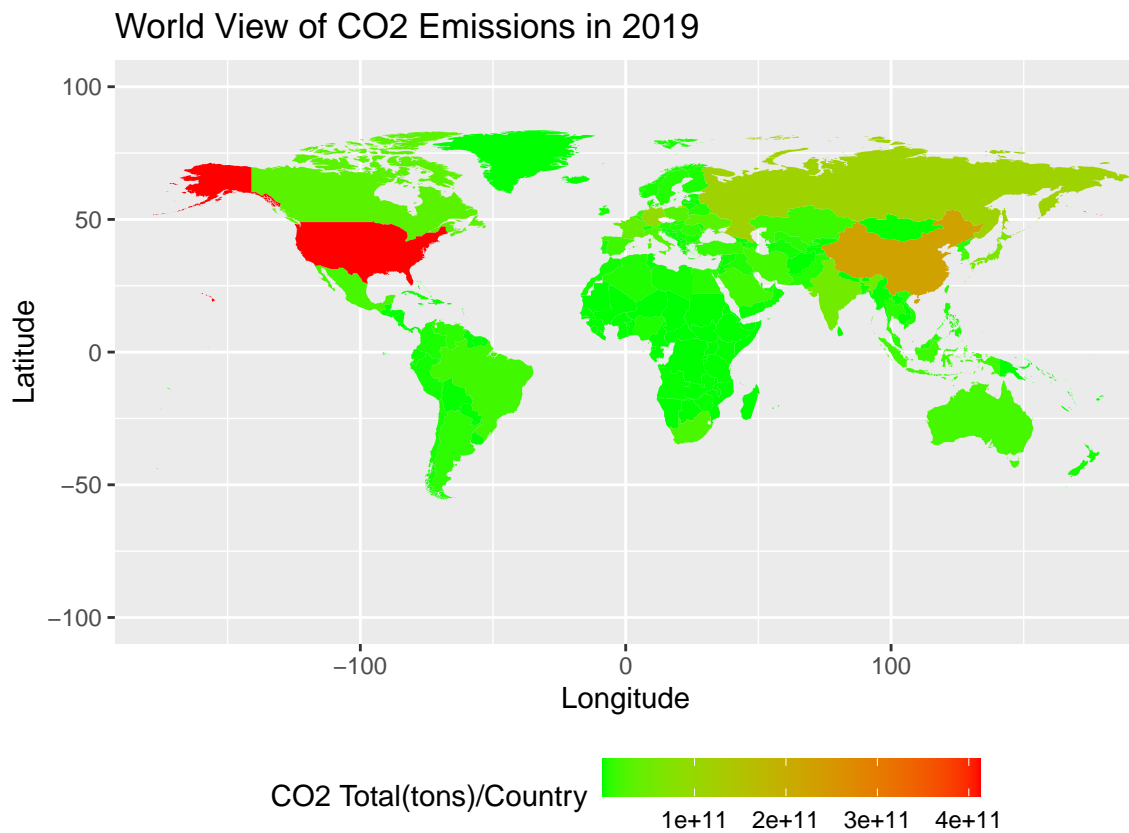
```

gg2 <- ggplot() + theme(legend.position="bottom",legend.key.size = unit(0.5, "cm"),legend.key.width = unit(1, "cm"))
gg2 <- gg2 + geom_map(data=world2, map=world2, aes(map_id=region,x=long, y=lat, fill=CO2.Total))

## Warning in geom_map(data = world2, map = world2, aes(map_id = region, x =
## long, : Ignoring unknown aesthetics: x and y

#changing features of world map (color, axes scales, adding a title, etc)
gg2 <- gg2 + scale_fill_gradient(low = "green", high = "red", guide = "colourbar",aesthetics = "fill")
gg2 <- gg2 + coord_equal() + ggtitle("World View of CO2 Emissions in 2019")
gg2

```



Analyze the top 3 countries that emit the most CO2 by geographic region

```

#finding and plotting the top 3 countries in each geographic region that emit the most CO2
topCountries <- CO2total_byCountry[order(-CO2total_byCountry$CO2.Total.Per.Country),] #get CO2 emission.

#finding the top 3 countries in each geographic region that emit the most CO2
top3Africa<-head(topCountries[topCountries$continent=="Africa",],3)
top3Americas<-head(topCountries[topCountries$continent=="Americas",],3)
top3Asia<-head(topCountries[topCountries$continent=="Asia",],3)
top3Europe<-head(topCountries[topCountries$continent=="Europe",],3)
top3Oceania<-head(topCountries[topCountries$continent=="Oceania",],3)

```



```

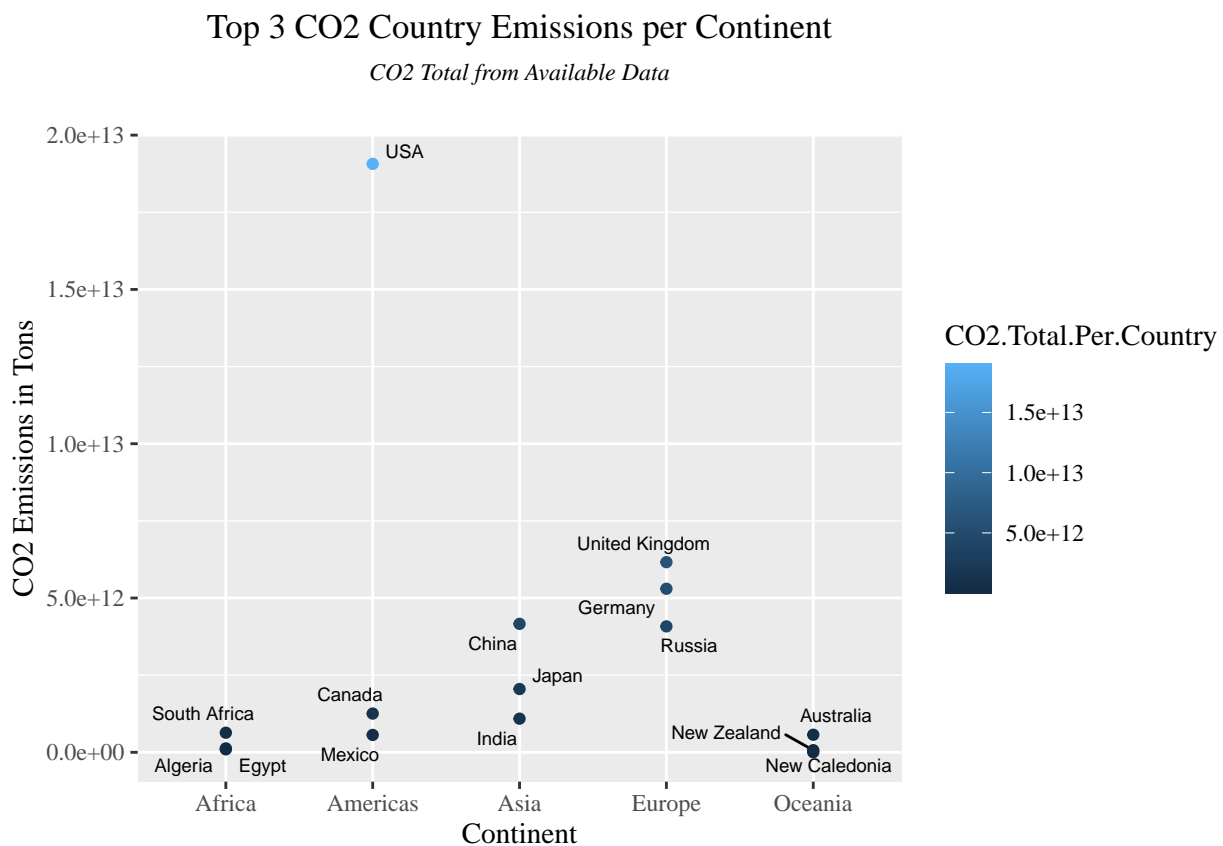
top3pContinent <- rbind(top3Africa,top3Americas,top3Asia,top3Europe,top3Oceania)

#soring the data into a variable for plotting
mostEmissions <- ggplot(top3pContinent,aes(x=continent,y=C02.Total.Per.Country))

#plotting the data
mostEmissions<- mostEmissions +
  geom_text_repel(aes(label = Country), size = 2.5)+
  geom_point(aes(colour=C02.Total.Per.Country))+
  ggtitle(expression(atop("Top 3 CO2 Country Emissions per Continent",
                           atop(italic("CO2 Total from Available Data"),""))))+
  xlab("Continent") +
  ylab("CO2 Emissions in Tons")+
  theme(text = element_text(family = 'serif'))+
  theme(plot.title = element_text(hjust=0.5))

mostEmissions

```



1.2B: CO2 Emissions by Area:

```

cleanedC02 <- C02
#identified countries with NA values, then excluded the lower 50% of the countries who contributed the

```

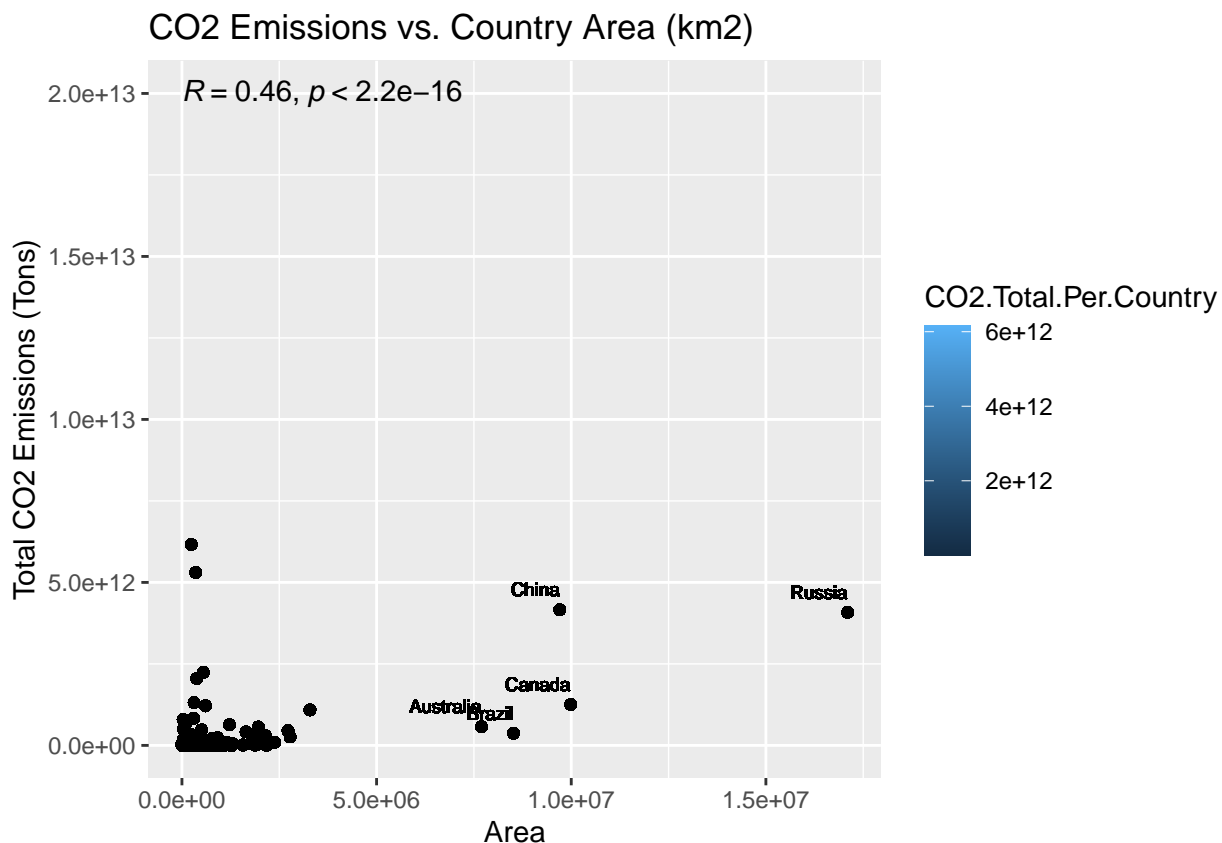
```

#joining country, total area of each country, percent of global land mass that the country takes up, and
cleanedCO2 <- inner_join(cleanedCO2,CO2total_byCountry) %>% select("Country","Area","CO2.Total.Per.Country")

#plotting a scatter plot for country land mass vs. total CO2 emissions per, and finding the correlation
areavsco2plot <- ggplot(data=cleanedCO2,aes(x=Area,y=CO2.Total.Per.Country))+
  geom_point(aes(fill=CO2.Total.Per.Country)) +
  ggtitle("Country Area vs. Total CO2 Emission") +
  labs(x="Area",y="Total CO2 Emissions (Tons)",
       title="CO2 Emissions vs. Country Area (km2)") +
  geom_text(aes(label=ifelse(Area>5e6,as.character(Country),'')),size=2.5,hjust=1,vjust=-1) +
  stat_cor(method = "pearson",label.x = 0e0, label.y = 2e13)

areavsco2plot

```



CO2 Emissions in Developed vs. Non-developed:

```

x <- getURL("https://raw.githubusercontent.com/owid/co2-data/master/owid-co2-data.csv")
dat2 <- read.csv(text = x)
y <- getURL("https://raw.githubusercontent.com/jliuob/QBSsquad/main/CO2_Cleaned_V2.csv")
dat1 <- read.csv(text = y)
dat1 <- dat1[,1:7]
class(dat1)

```

```
## [1] "data.frame"
```

```
class(dat2)
```

```
## [1] "data.frame"
```

```
colnames(dat1)[1] <- "Country"
```

```
merged.dat <- sqldf("SELECT d1.Country, d1.year, d1.'CO2.emission..Tons.',  
d2.gdp, d2.co2_per_gdp, d2.co2_per_capita, d2.coal_co2, d2.coal_co2_per_capita  
FROM 'dat1' AS d1  
JOIN 'dat2' AS d2  
ON d1.Country=d2.country  
AND d1.Year=d2.year")  
dim(merged.dat)
```

```
## [1] 21731      8
```

```
merged.dat <- na.omit(merged.dat)  
View(merged.dat)
```

Finding CO2 per capita per country:

```
#finding the average CO2 emissions per country  
avgCO2byCountry <- aggregate(merged.dat$CO2.emission..Tons.,by=list(Category=merged.dat$Country), FUN=mean)  
  
#finding the average gdp per country  
#recode NAs to zero to run mean functions  
merged.dat$gdp[is.na(merged.dat$gdp)] <- 0  
avgGDP <- aggregate(merged.dat$gdp,by=list(Category=merged.dat$Country), FUN=mean)  
  
merged.dat$gdp[merged.dat$Country=="Aruba"]
```

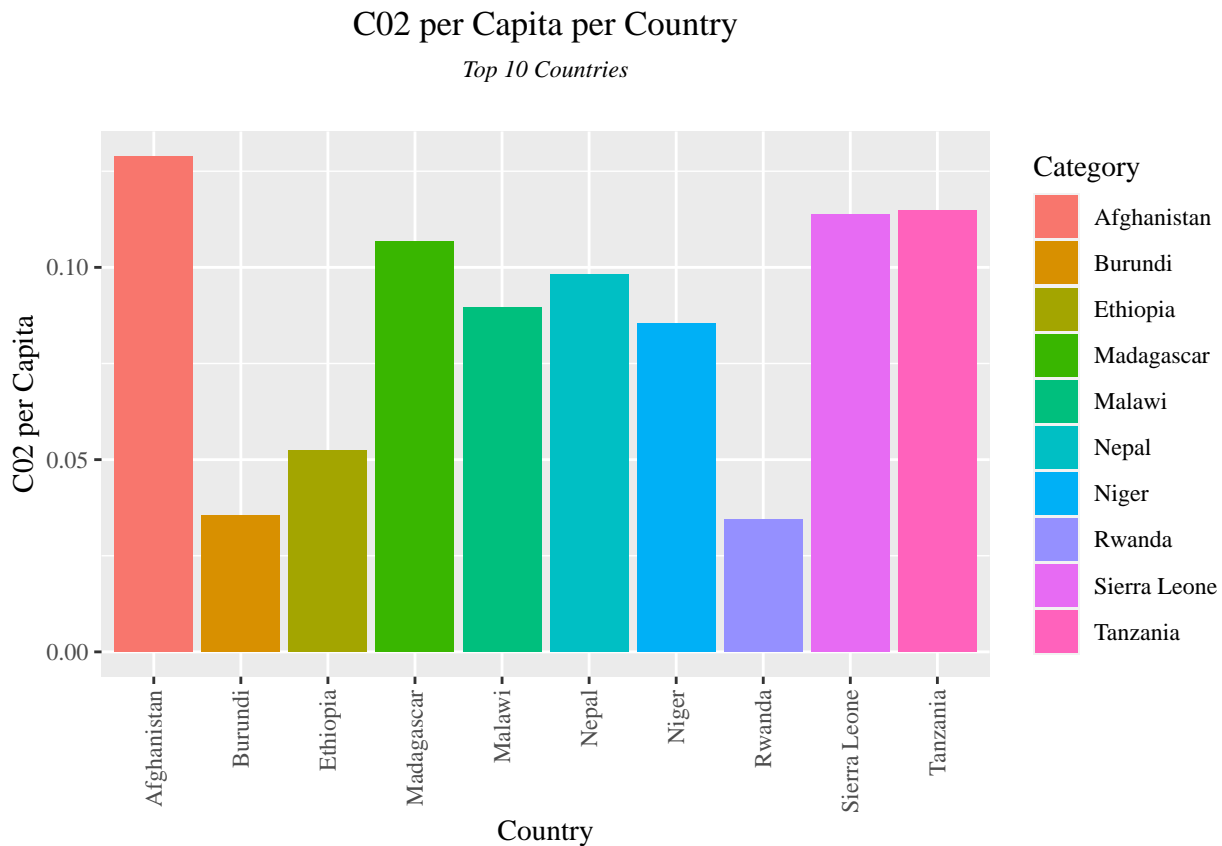
```
## numeric(0)
```

```
#calculating the gdp per capita for each country  
avgCO2perCapita <- aggregate(merged.dat$co2_per_capita,by=list(Category=merged.dat$Country), FUN=mean)  
  
#omitting na values and finding the top 10 CO2 emissions per capita per country  
avgCO2perCapita <- na.omit(avgCO2perCapita)  
avgCO2perCapita_10 <- avgCO2perCapita[order(avgCO2perCapita$x),]  
avgCO2perCapita_10 <- head(avgCO2perCapita_10,10)  
  
#plotting top 10 CO2 emissions per capita per country  
capitaplot <- ggplot(avgCO2perCapita_10,  
aes(Category,x,fill=Category))+  
geom_bar(stat="identity")+  
scale_x_discrete(guide=guide_axis(angle = 90))+  
xlab("Country")+
```

```

ylab("C02 per Capita")+
ggtitle(expression(atop("C02 per Capita per Country",
                        atop(italic("Top 10 Countries"),"")))) +
theme(plot.title = element_text(hjust=0.5))+
theme(text = element_text(family = 'serif'))
capitaplot

```



C02 by GDP:

```

colnames(avgGDP) <- c("Country","GDP_Average") #renaming column names

#merging country, GDP_Average, and Population 2022 columns into one table
avgGDP <- avgGDP %>% inner_join(cleanedC02) %>% select("Country","GDP_Average","Population.2022.")

## Joining, by = "Country"

avgGDP <- avgGDP %>% mutate(GDPperCapita = GDP_Average/Population.2022.)

#creating ordinal data for developed vs. developing
#changed developed threshold to $9000 per capita because original GDP threshold was too conservative
develop_ord <- ifelse(avgGDP$GDPperCapita>=9000,"developed",
                      ifelse(avgGDP$GDPperCapita<9000,"developing",0))
avgGDP$DevelopStatus <- develop_ord

```

```
table(avgGDP$Country[avgGDP$GDPperCapita>=9000])
```

```
##
##      Armenia      Austria      Belarus
##      271          271          271
##      Belgium      Canada      Costa Rica
##      271          271          271
##      Croatia      Cyprus      Czechia
##      271          271          271
##      Denmark      Estonia      Finland
##      271          271          271
##      France      Georgia      Germany
##      271          271          271
##      Hong Kong      Hungary      Iceland
##      271          271          271
##      Ireland      Israel      Italy
##      271          271          271
##      Japan      Kazakhstan      Latvia
##      271          271          271
##      Lithuania      Luxembourg      Malta
##      271          271          271
##      Netherlands      Norway      Russia
##      271          271          271
##      Singapore      Slovakia      Slovenia
##      271          271          271
##      Sweden      Switzerland      Taiwan
##      271          271          271
##      Ukraine United Arab Emirates      United Kingdom
##      271          271          271
##      Uruguay
##      271
```

```
table(avgGDP$Country[avgGDP$GDPperCapita<9000])
```

```
##
##      Afghanistan      Albania      Algeria
##      271          271          271
##      Angola      Argentina      Australia
##      271          271          271
##      Azerbaijan      Bahrain      Bangladesh
##      271          271          271
##      Barbados      Benin      Bolivia
##      271          271          271
##      Bosnia and Herzegovina      Botswana      Brazil
##      271          271          271
##      Bulgaria      Burundi      Cambodia
##      271          271          271
##      Cameroon      Chile      China
##      271          271          271
##      Colombia      Cuba      Dominican Republic
##      271          271          271
##      Egypt      El Salvador      Eswatini
```

##	271	271	271
##	Ethiopia	Ghana	Greece
##	271	271	271
##	Guatemala	Haiti	Honduras
##	271	271	271
##	India	Indonesia	Iran
##	271	271	271
##	Iraq	Jamaica	Jordan
##	271	271	271
##	Kenya	Kyrgyzstan	Laos
##	271	271	271
##	Lebanon	Lesotho	Libya
##	271	271	271
##	Madagascar	Malawi	Malaysia
##	271	271	271
##	Mauritania	Mauritius	Mexico
##	271	271	271
##	Moldova	Mongolia	Montenegro
##	271	271	271
##	Morocco	Mozambique	Myanmar
##	271	271	271
##	Namibia	Nepal	New Zealand
##	271	271	271
##	Niger	Nigeria	North Korea
##	271	271	271
##	North Macedonia	Pakistan	Panama
##	271	271	271
##	Paraguay	Peru	Philippines
##	271	271	271
##	Poland	Portugal	Romania
##	271	271	271
##	Rwanda	Senegal	Serbia
##	271	271	271
##	Sierra Leone	South Africa	South Korea
##	271	271	271
##	Spain	Sri Lanka	Syria
##	271	271	271
##	Tajikistan	Tanzania	Thailand
##	271	271	271
##	Trinidad and Tobago	Tunisia	Turkey
##	271	271	271
##	Turkmenistan	Uzbekistan	Venezuela
##	271	271	271
##	Vietnam	Yemen	Zambia
##	271	271	271
##	Zimbabwe		
##	271		

avgGDP\$GDP_Average

##	[1]	2.146437e+10	2.146437e+10	2.146437e+10	2.146437e+10	2.146437e+10
##	[6]	2.146437e+10	2.146437e+10	2.146437e+10	2.146437e+10	2.146437e+10
##	[11]	2.146437e+10	2.146437e+10	2.146437e+10	2.146437e+10	2.146437e+10
##	[16]	2.146437e+10	2.146437e+10	2.146437e+10	2.146437e+10	2.146437e+10

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

177

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## [36201] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36206] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36211] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36216] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36221] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36226] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36231] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36236] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36241] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36246] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36251] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36256] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36261] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36266] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36271] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36276] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36281] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36286] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36291] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36296] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36301] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36306] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
## [36311] 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10 1.542784e+10
```

```
#merging CO2 Data
```

```
avgGDP <- avgGDP %>% inner_join(CO2_noduplicates) %>% select("Country","GDP_Average","Population.2022."
```

```
## Joining, by = c("Country", "Population.2022.")
```

```
#categorizing developed nations into a developed variable
```

```
#categorizing developing nations into a developing variable
```

```
developed <- avgGDP[avgGDP$DevelopStatus=="developed",]
```

```
developing <- avgGDP[avgGDP$DevelopStatus=="developing",]
```

```
sumCO2Developed <- sum(developed$CO2.emission..Tons.)
```

```
sumCO2Developing <- sum(developing$CO2.emission..Tons.)
```

```
#creating a table and a barplot comparing developed vs developing nations total CO2 emissions
```

```
DevelopvsDeveloping <- matrix(ncol=2,nrow=2)
```

```
DevelopvsDeveloping <- as.data.frame(DevelopvsDeveloping)
```

```
colnames(DevelopvsDeveloping)<-c("CO2_Total","Status")
```

```
DevelopvsDeveloping[1,] <- c(as.numeric(sumCO2Developed),"Developed")
```

```
DevelopvsDeveloping[2,] <- c(as.numeric(sumCO2Developing),"Developing")
```

```
DevelopvDevelopingplot <- ggplot(DevelopvsDeveloping,
                                aes(x=Status,y=CO2_Total,fill=Status))+
```

```
  geom_bar(stat="identity") +
```

```
  theme(axis.text.x = element_text(size = 11)) +
```

```
  scale_x_discrete(guide = guide_axis(n.dodge = 3))+
```

```
  xlab(c("Development Status"))+
```

```
  ylab("CO2 Total Tons")+
```

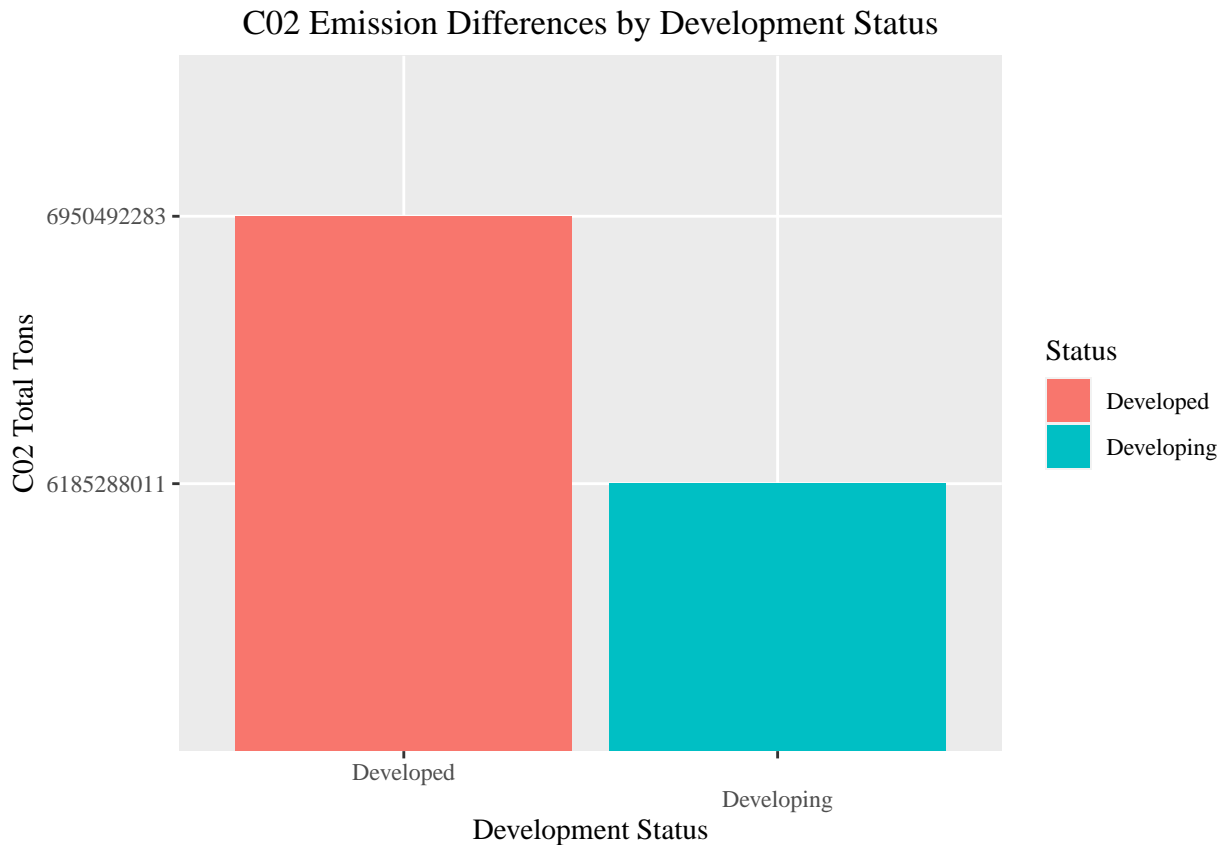
```
  ggtitle("CO2 Emission Differences by Development Status")+
```

```
  theme_gray()+
```

```
  theme(plot.title = element_text(hjust=0.5))+
```

```
theme(text = element_text(family = 'serif'))+
stat_cor(method = "pearson", label.x = 0, label.y = 0)
```

DevelopvDevelopingplot



Aim 2.1 COVID CO2 emission analysis

First, to examine global CO2 emission in pre-pandemic and pandemic years, a subset of data about 2018, 2019, and 2020 CO2 emission in all countries is created.

```
pandemic.CO2 <- CO2 %>%
  filter(Year >= 2018, na.rm = TRUE)
head(pandemic.CO2)
```

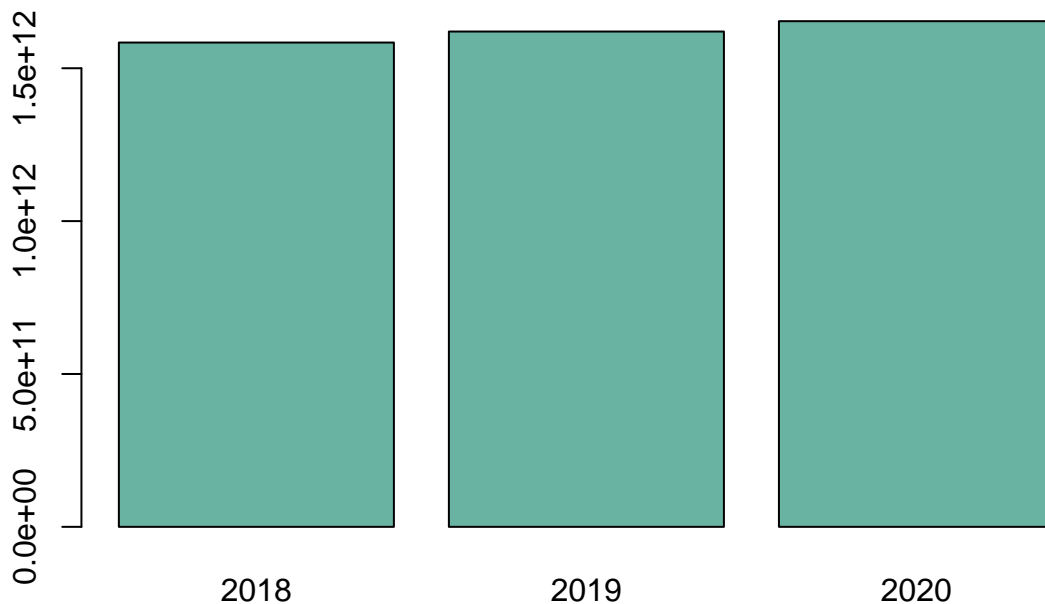
```
##      Country Code Calling.Code Year CO2.emission..Tons. Population.2022.
## 1 Afghanistan  AF           93 2018      168541787      41128771
## 2 Afghanistan  AF           93 2019      180688461      41128771
## 3 Afghanistan  AF           93 2020      192848747      41128771
## 4   Albania    AL          355 2018      276104384      2842321
## 5   Albania    AL          355 2019      280967834      2842321
## 6   Albania    AL          355 2020      285502507      2842321
##      Area
## 1 652230
## 2 652230
```

```
## 3 652230
## 4 28748
## 5 28748
## 6 28748
```

Then, we want to explore the general trend of changes in CO2 emission in all countries due to the pandemic. Thus, we presented the global cumulative CO2 emission during these three years.

```
cum.2018 <- sum(pandemic.CO2[which(pandemic.CO2$Year=='2018'), 5])
cum.2019 <- sum(pandemic.CO2[which(pandemic.CO2$Year=='2019'), 5])
cum.2020 <- sum(pandemic.CO2[which(pandemic.CO2$Year=='2020'), 5])

barplot(tapply(pandemic.CO2$CO2.emission..Tons., format(pandemic.CO2$Year), FUN=sum), col="#69b3a2")
```



We could see from the graph that there's not much change in global cumulative CO2 emission during the pandemic years. Here we want to separately analyze the top5 emission countries and see if there's any difference.

```
# dataframe for top 5 co2 emission countries each year
CO2.a <- pandemic.CO2 %>%
  filter(Year == "2018") %>%
  arrange(desc(CO2.emission..Tons.)) %>%
  slice_head(n = 5)
CO2.a
```

```
##      Country Code Calling.Code Year CO2.emission..Tons. Population.2022.
## 1  United States  US           1 2018      407000000000      338289857
## 2    China      CN           86 2018      214000000000      1425887337
## 3   Russia     XX            7 2018      112000000000      144713314
## 4   Germany    DE           49 2018       91279876936       83369843
## 5 United Kingdom GB           44 2018       77462557032       67508936
##      Area
## 1 9372610
## 2 9706961
```

```
## 3 17098242
## 4 357114
## 5 242900
```

```
C02.b <- pandemic.CO2 %>%
  filter(Year == "2019") %>%
  arrange(desc(C02.emission..Tons.)) %>%
  slice_head(n = 5)
C02.b
```

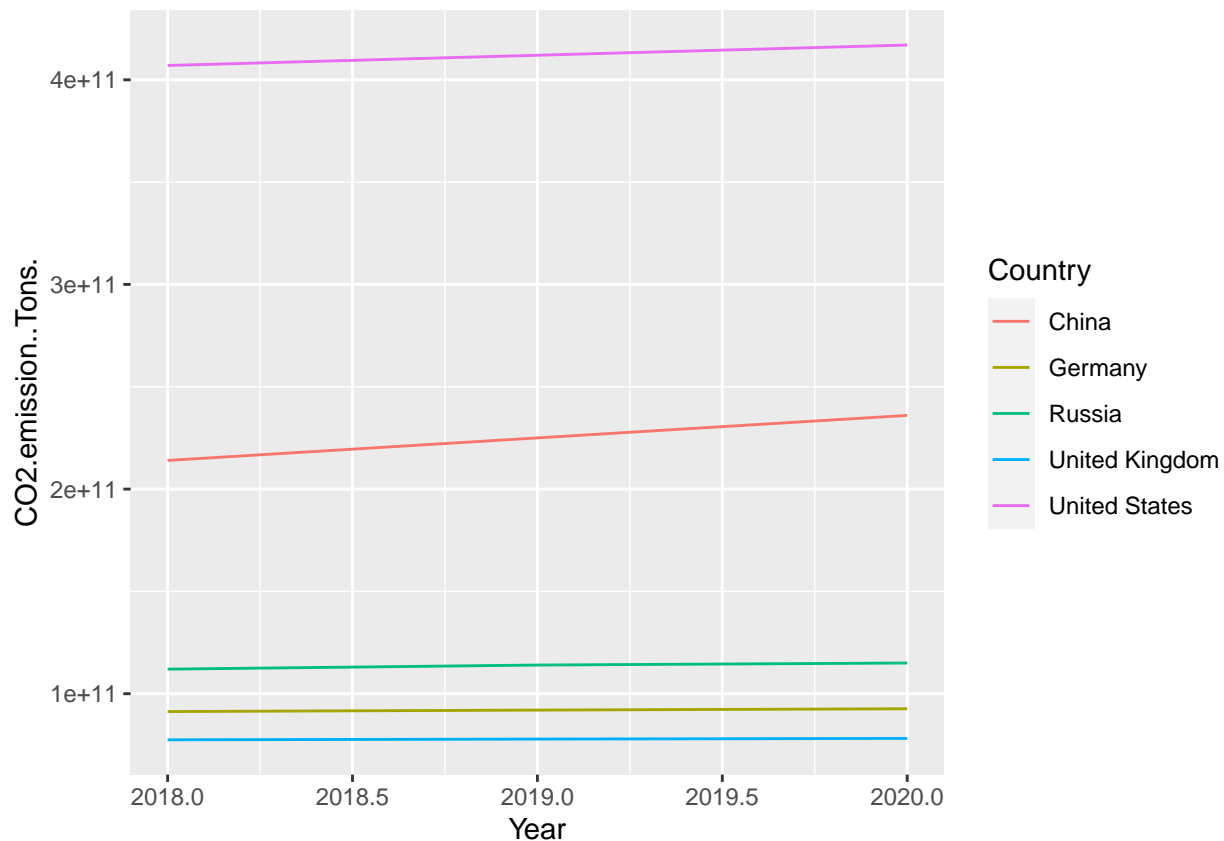
```
##      Country Code Calling.Code Year C02.emission..Tons. Population.2022.
## 1 United States US          1 2019      412000000000      338289857
## 2 China CN              86 2019      225000000000      1425887337
## 3 Russia XX             7 2019      114000000000      144713314
## 4 Germany DE           49 2019       91991304745      83369843
## 5 United Kingdom GB      44 2019       77831566725      67508936
##      Area
## 1 9372610
## 2 9706961
## 3 17098242
## 4 357114
## 5 242900
```

```
C02.c <- pandemic.CO2 %>%
  filter(Year == "2020") %>%
  arrange(desc(C02.emission..Tons.)) %>%
  slice_head(n = 5)
C02.c
```

```
##      Country Code Calling.Code Year C02.emission..Tons. Population.2022.
## 1 United States US          1 2020      417000000000      338289857
## 2 China CN              86 2020      236000000000      1425887337
## 3 Russia XX             7 2020      115000000000      144713314
## 4 Germany DE           49 2020       92635615097      83369843
## 5 United Kingdom GB      44 2020       78161145636      67508936
##      Area
## 1 9372610
## 2 9706961
## 3 17098242
## 4 357114
## 5 242900
```

```
df.top5 = rbind(C02.a,C02.b,C02.c)
```

```
# plot general trend
ggplot(df.top5, aes(x = Year, y = C02.emission..Tons., color = Country)) +
  geom_line()
```

As shown in this graph, although the overall CO2 emission in these 5 top emission countries slightly increased during pandemic, there's still not much difference observed. This is not what we've expected before since we expect the emission rate to decrease during the pandemic.

Now we want to join another dataset with information about absolute growth of CO2 and other information about CO2 per capita/gdp/unit energy to explore further on pandemic-CO2 emission relationship.

First, we read our new dataset about owid-co2-data from github and filtered the wanted period.

```
res = GET("https://github.com/owid/co2-data")

x <- getURL("https://raw.githubusercontent.com/owid/co2-data/master/owid-co2-data.csv")
world.co2 <- read.csv(text = x)
class(world.co2)
```

```
## [1] "data.frame"
```

```
world.pandemic <- world.co2 %>%
  filter(year >= 1990, na.rm = TRUE)
```

Then we joined the two datasets based on country names. Here we used SQL to make the join.

```
# based on both country names and year
tot.dat <- sqldf("SELECT p.Country, p.year, p.'CO2.emission..Tons.' AS 'co2 emission(tons)', p.'Populat
FROM 'pandemic.CO2' AS p
JOIN 'world.pandemic' AS w
```

```
ON p.Country=w.country
AND p.Year=w.year")
```

Now we wanted to plot the absolute growth of co2 emission during the pandemic in a global perspective. First, let's check if there are any missing values in our interested data.

```
sum(is.na(tot.dat$co2_growth_abs))
```

```
## [1] 3
```

```
which(is.na(tot.dat$co2_growth_abs))
```

```
## [1] 442 443 444
```

```
# NAs are located at Antarctica, Christmas Island, and Puerto Rico. We noticed that there might be no d
tot.dat$co2_growth_abs[is.na(tot.dat$co2_growth_abs)] = 0
```

```
#check again
```

```
sum(is.na(tot.dat$co2_growth_abs))
```

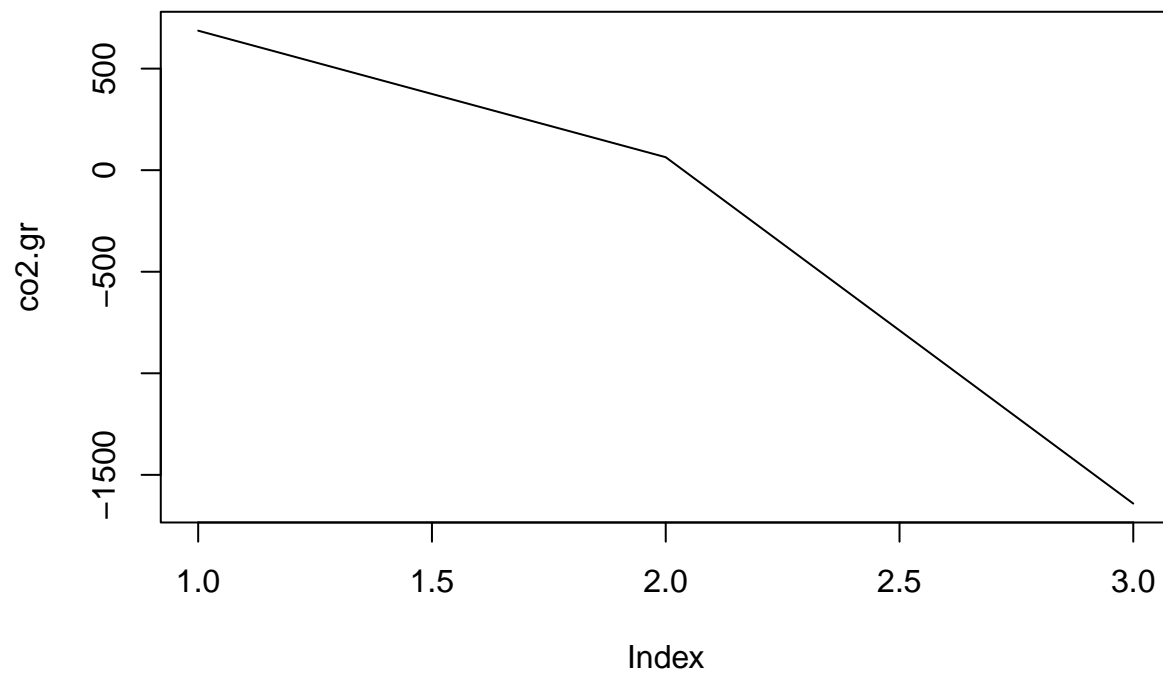
```
## [1] 0
```

Now we can visualize our results regarding absolute growth of CO2 emission. First, let's check if there's difference in the absolute CO2 growth during the pandemic.

Then we plotted the absolute co2 growth and co2 emission per-capita during the pandemic in countries with the highest overall co2 emission these years.

```
# global trend in co2 absolute growth
a <- sum(tot.dat[which(tot.dat$Year=='2018'), 6])
b <- sum(tot.dat[which(tot.dat$Year=='2019'), 6])
c <- sum(tot.dat[which(tot.dat$Year=='2020'), 6])
co2.gr <- c(a,b,c)
plot(co2.gr, type = "line")
```

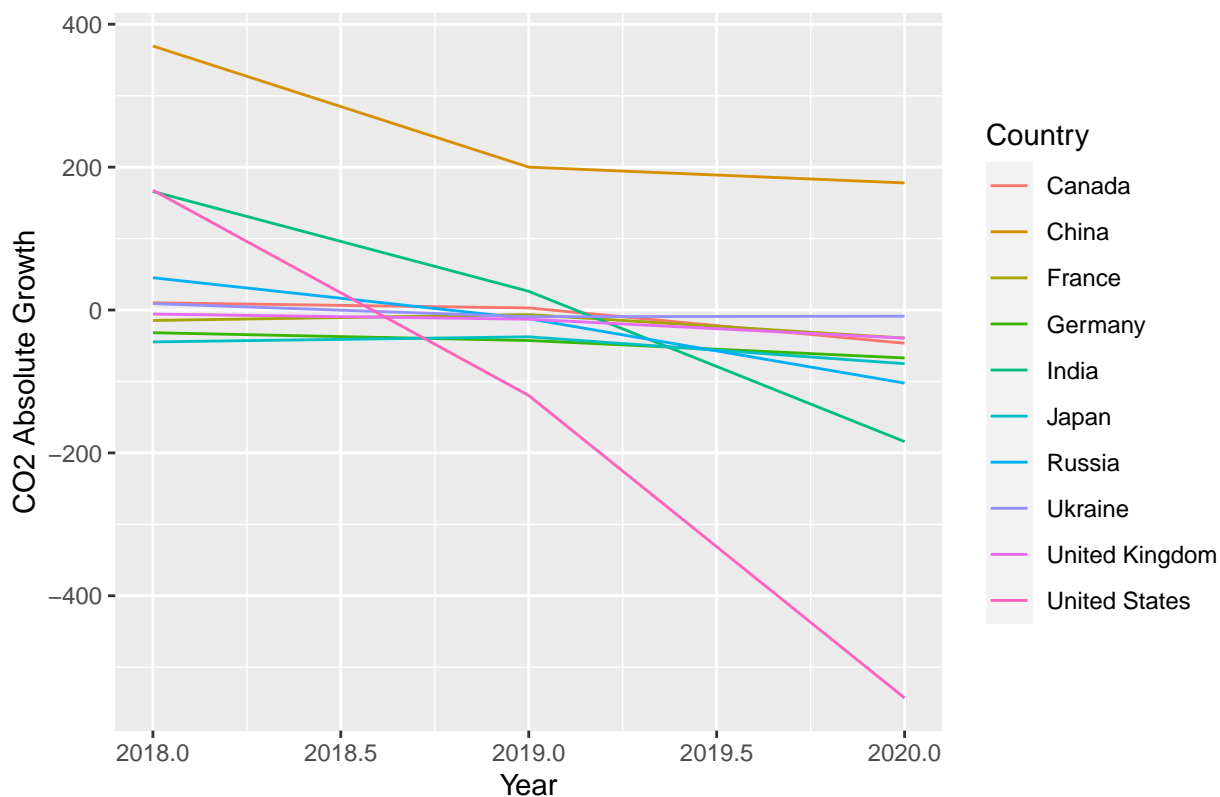
```
## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to first
## character
```



```
# trend in top 10 co2 emission countries
tot.dat <- tot.dat %>%
  arrange(desc(`co2 emission(tons)`) )
top.co2 <- tot.dat[1:30,]

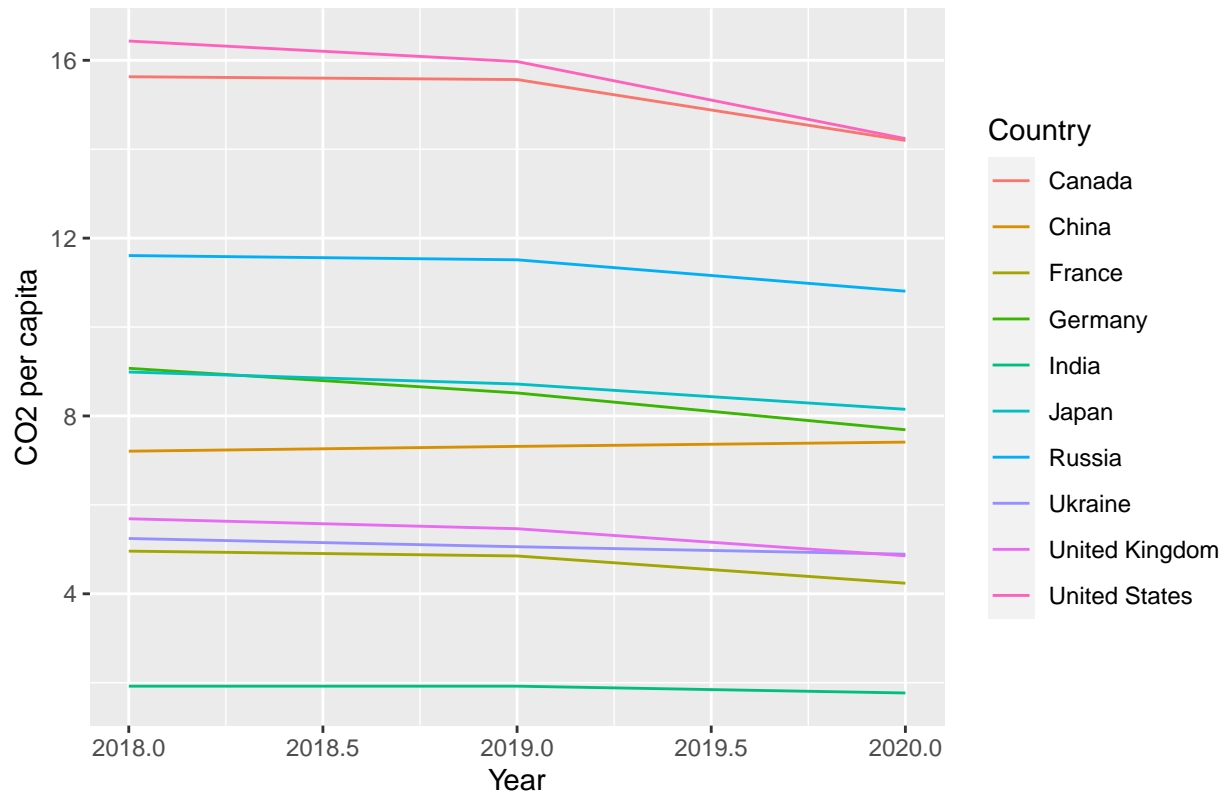
ggplot(top.co2, aes(x = Year, y = co2_growth_abs, color = Country)) +
  xlab("Year")+
  ylab("C02 Absolute Growth")+
  ggtitle("Absolute C02 Growth During Pandemic Years in top10 C02 Emission Countries")+
  geom_line()
```

Absolute CO2 Growth During Pandemic Years in top10 CO2 Emission Coi



```
ggplot(top.co2, aes(x = Year, y = co2_per_capita, color = Country)) +
  xlab("Year")+
  ylab("CO2 per capita")+
  ggtitle("CO2 Emission per capita During Pandemic Years in top10 CO2 Emission Countries")+
  geom_line()
```

CO2 Emission per capita During Pandemic Years in top10 CO2 Emission Countries



To further explore the relationship and intensity of different features in our dataset, we filtered out top 10 CO2 emission countries each year and presented our observations in a heatmap.

```
# filtered out top 10 emission countries
C02.1 <- tot.dat %>%
  filter(Year == "2018") %>%
  arrange(desc(`co2 emission(tons)`) ) %>%
  slice_head(n = 10)

C02.2 <- tot.dat %>%
  filter(Year == "2019") %>%
  arrange(desc(`co2 emission(tons)`) ) %>%
  slice_head(n = 10)

C02.3 <- tot.dat %>%
  filter(Year == "2020") %>%
  arrange(desc(`co2 emission(tons)`) ) %>%
  slice_head(n = 10)

co2.top10 = rbind(C02.1,C02.2,C02.3)
co2.top10$Country_Year <- paste(co2.top10$Country, co2.top10$Year)
rownames(co2.top10) <- co2.top10[, 9]

# each country along with the year should be distinctly represented
co2.top10 %<>% arrange(Country_Year)
```

```
# interactive heatmap
dir.create("folder")
```

```
## Warning in dir.create("folder"): 'folder' already exists
```

```
heatmaply(percentize(co2.top10)[- (1:2)],
  xlab = "Features",
  ylab = "Country Year",
  main = "Features related to CO2 emission during pandemic in top10 CO2 emission countries",
  margins = c(60,100,40,20),
  colors = colorRampPalette(brewer.pal(3, "RdBu"))(256),
  seriate = "OLO",
  heatmap_layers = theme(axis.line=element_blank()),
  file = "folder/CO2_plot.html")
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, please
```

```
# browseURL("folder/CO2_plot.html")
```

We did the same thing by filtered out top 10 populated countries each year and presented our observations in a heatmap.

```
# filtered out top 10 populated countries
pop1 <- tot.dat %>%
  filter(Year == "2018") %>%
  arrange(desc(population)) %>%
  slice_head(n = 10)

pop2 <- tot.dat %>%
  filter(Year == "2019") %>%
  arrange(desc(population)) %>%
  slice_head(n = 10)

pop3 <- tot.dat %>%
  filter(Year == "2020") %>%
  arrange(desc(population)) %>%
  slice_head(n = 10)

pop.top10 = rbind(pop1,pop2,pop3)
pop.top10$Country_Year <- paste(pop.top10$Country, pop.top10$Year)
rownames(pop.top10) <- pop.top10[, 9]

# each country along with the year should be distinctly represented
pop.top10 %<>% arrange(Country_Year)

# interactive heatmap
dir.create("folder")
```

```
## Warning in dir.create("folder"): 'folder' already exists
```

```
heatmaply(percentize(pop.top10)[-1:2]),
          xlab = "Features",
          ylab = "Country Year",
          main = "Features related to CO2 emission during pandemic in top10 populated countries",
          margins = c(60,100,40,20),
          colors = colorRampPalette(brewer.pal(3, "RdBu"))(256),
          seriate = "OLO",
          heatmap_layers = theme(axis.line=element_blank()),
          file = "folder/pop_plot.html")
```

```
# browseURL("folder/pop_plot.html")
```

From the analysis of co2 emissions during COVID, although we cannot conclude an association between changes in co2 emissions and the pandemic because we ignored other variables, we thought it is interesting to display the result. The result might be more significant if we have updated data including 2021 and 2022.

Aim 2.2 Forecasting

We decided to predict the CO2 emissions in the next 30 years by using the data from 1992 to 2019, and fit in the proper ARIMA models to do the prediction. We choose six countries and five continents to do the prediction.

```
# select three columns to do forecasting
CO2_forecasting<-CO2 %>% dplyr::select(Country, Year, CO2.emission..Tons.)
# make sure there is no NA
dim(CO2_forecasting)
```

```
## [1] 55284      3
```

```
dim(na.omit(CO2_forecasting))
```

```
## [1] 55284      3
```

```
# see which year should we start for forecasting
CO2_no_year_min[which(CO2_no_year_min$Year==max(CO2_no_year_min[]$Year)),]
```

```
##      Country Year
## 1:  Kosovo 2008
```

```
# decided to use 1992-2019 data to do the forecasting
CO2_forecasting<-CO2_forecasting %>%
  filter(Year>1991&Year<2020) %>%
  group_by(Country)
# View(CO2_forecasting)
```

Predictions on selected countries

```

# extract the six countries' data and fit in time series models separately
# US
CO2_US<-CO2_forecasting %>%
  filter(Country=="United States")
pred.us<-ts(CO2_US$CO2.emission..Tons., start=1992,frequency = 1) %>%
  auto.arima() %>%
  forecast(h=30)
# United Kingdom
CO2_UK<-CO2_forecasting %>%
  filter(Country=="United Kingdom")
pred.uk<-ts(CO2_UK$CO2.emission..Tons.,start=1992,frequency = 1) %>%
  auto.arima() %>%
  forecast(h=30)
# Germany
CO2_GE<-CO2_forecasting %>%
  filter(Country=="Germany")
pred.ge<-ts(CO2_GE$CO2.emission..Tons.,start=1992,frequency = 1) %>%
  auto.arima() %>%
  forecast(h=30)
# China
CO2_CH<-CO2_forecasting %>%
  filter(Country=="China")
pred.ch<-ts(CO2_CH$CO2.emission..Tons.,start=1992,frequency = 1) %>%
  auto.arima() %>%
  forecast(h=30)
# Russia
CO2_RU<-CO2_forecasting %>%
  filter(Country=="Russia")
pred.ru<-ts(CO2_RU$CO2.emission..Tons.,start=1992,frequency = 1) %>%
  auto.arima() %>%
  forecast(h=30)
# France
CO2_FR<-CO2_forecasting %>%
  filter(Country=="France")
pred.fr<-ts(CO2_FR$CO2.emission..Tons.,start=1992,frequency = 1) %>%
  auto.arima() %>%
  forecast(h=30)

```

Time series analysis based on six selected countries

Plot the predicted CO2 emissions for the six countries We expect to see what's the increasing or decreasing trend of the CO2 emissions in the next 30 years.

```

# devtools::install_github('cttobin/ggthemr')
library(ggthemr)

```

```

##
## Attaching package: 'ggthemr'

## The following objects are masked from 'package:ggpubr':

```



```
##
##      rotate_x_text, rotate_y_text

ggthemr("dust")
US.plot<-autoplot(pred.us)+
  ggtitle("United States")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

UK.plot<-autoplot(pred.uk)+
  ggtitle("United Kingdom")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

GE.plot<-autoplot(pred.ge)+
  ggtitle("Germany")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

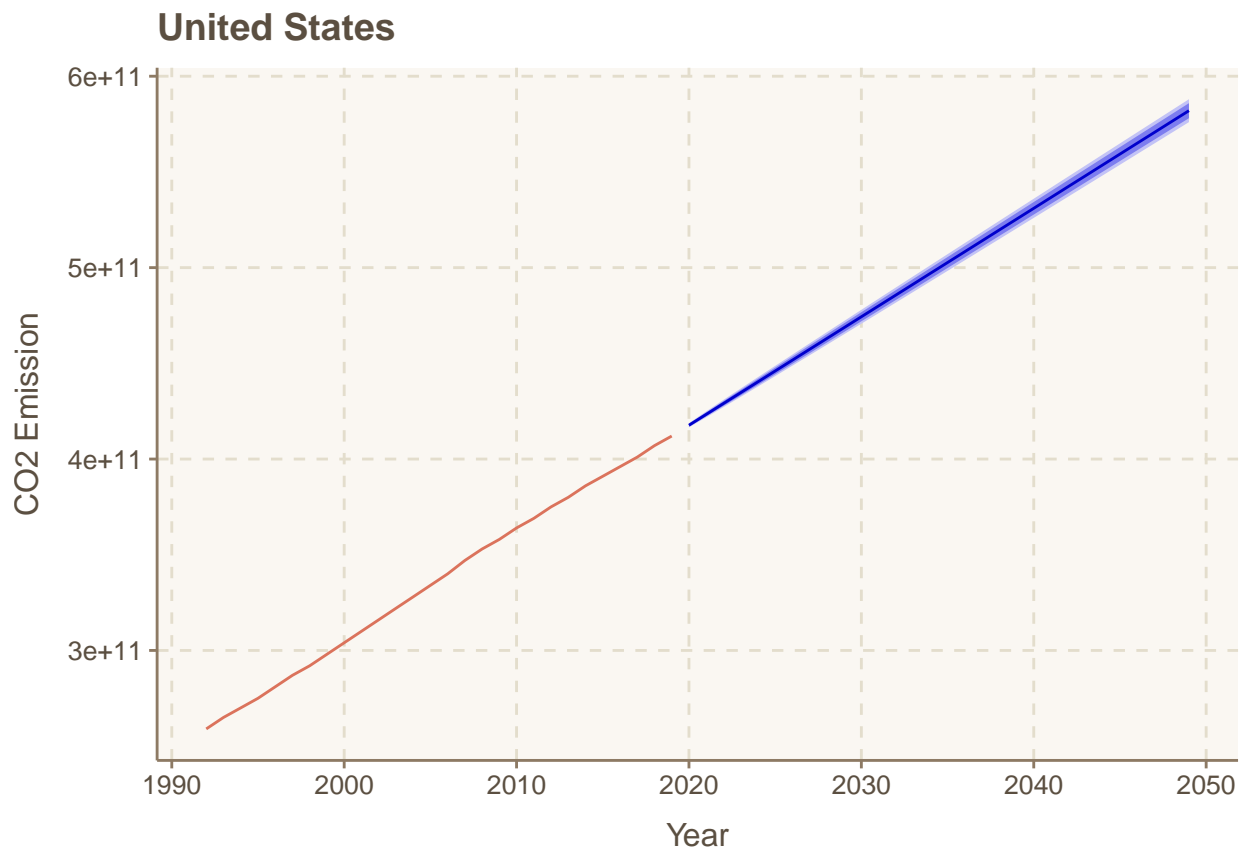
CH.plot<-autoplot(pred.ch)+
  ggtitle("China")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

RU.plot<-autoplot(pred.ru)+
  ggtitle("Russia")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

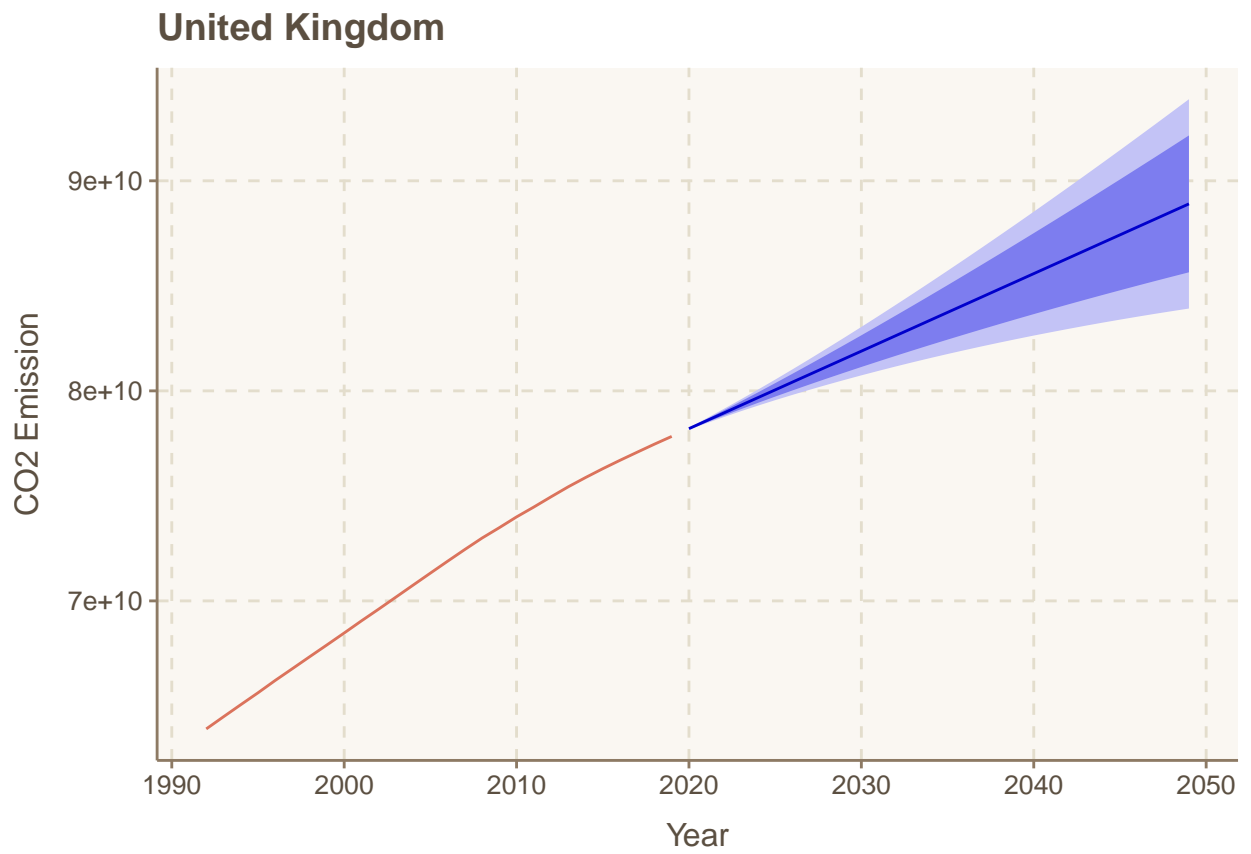
FR.plot<-autoplot(pred.fr)+
  ggtitle("France")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

# Save the plots
ggsave("US.png",US.plot,width = 10,height = 7.5)
ggsave("UK.png",UK.plot,width = 10,height = 7.5)
ggsave("GE.png",GE.plot,width = 10,height = 7.5)
ggsave("CH.png",CH.plot,width = 10,height = 7.5)
ggsave("RU.png",RU.plot,width = 10,height = 7.5)
ggsave("FR.png",FR.plot,width = 10,height = 7.5)

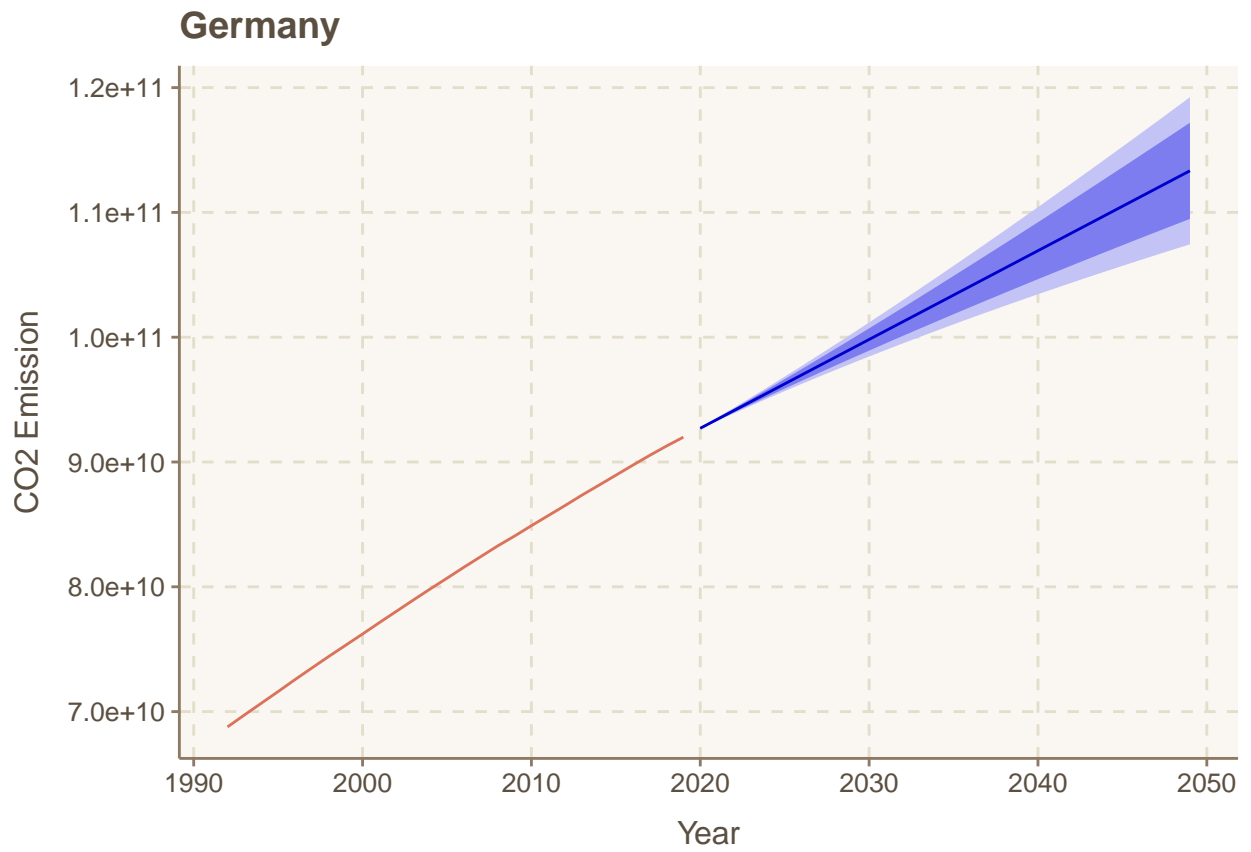
# view the plots
US.plot
```



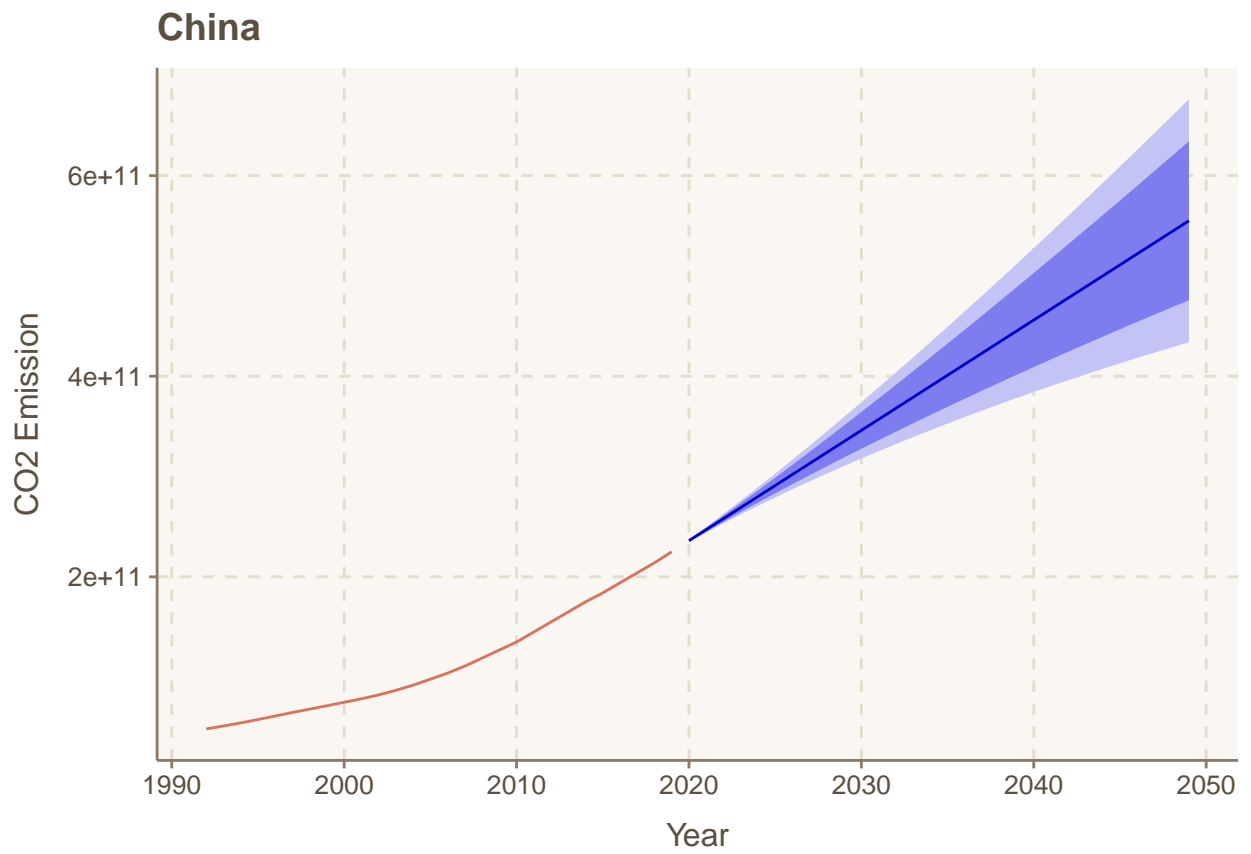
UK.plot



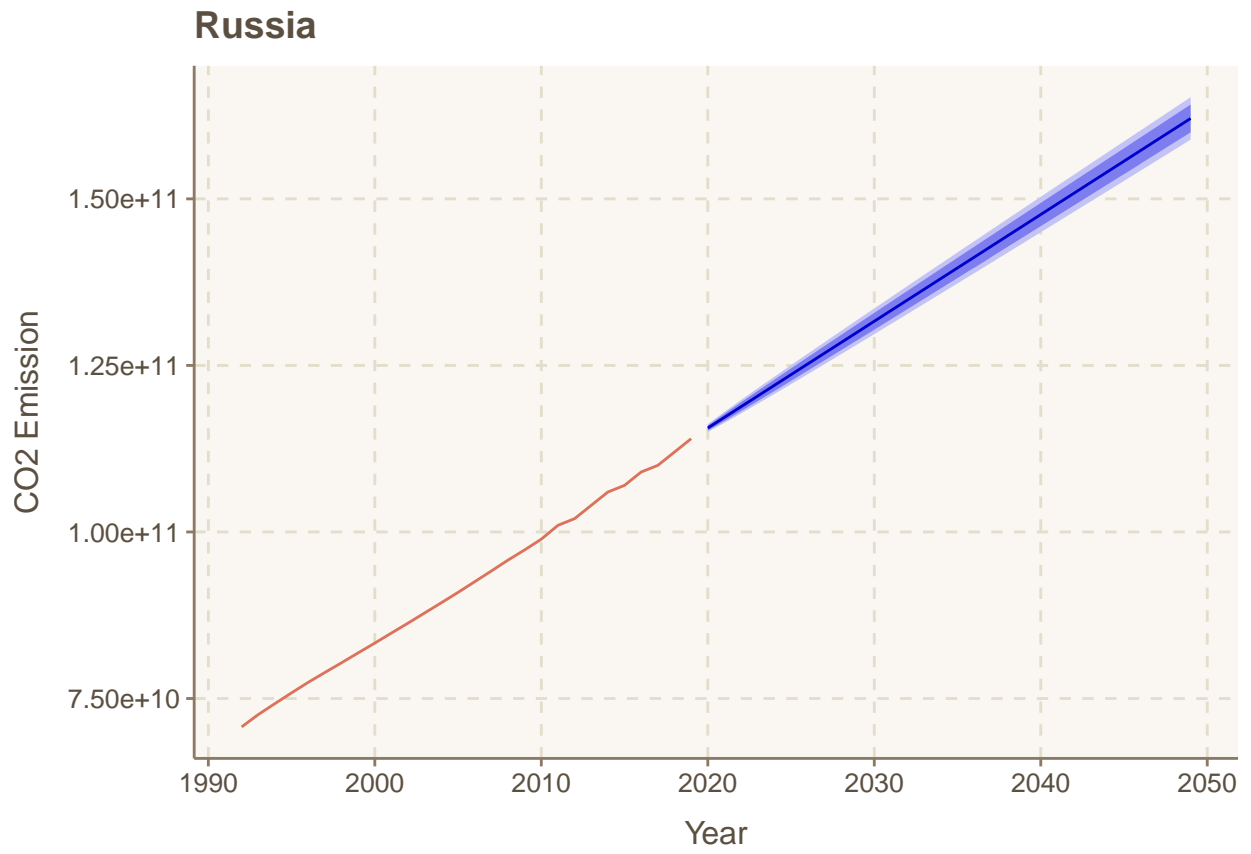
GE.plot



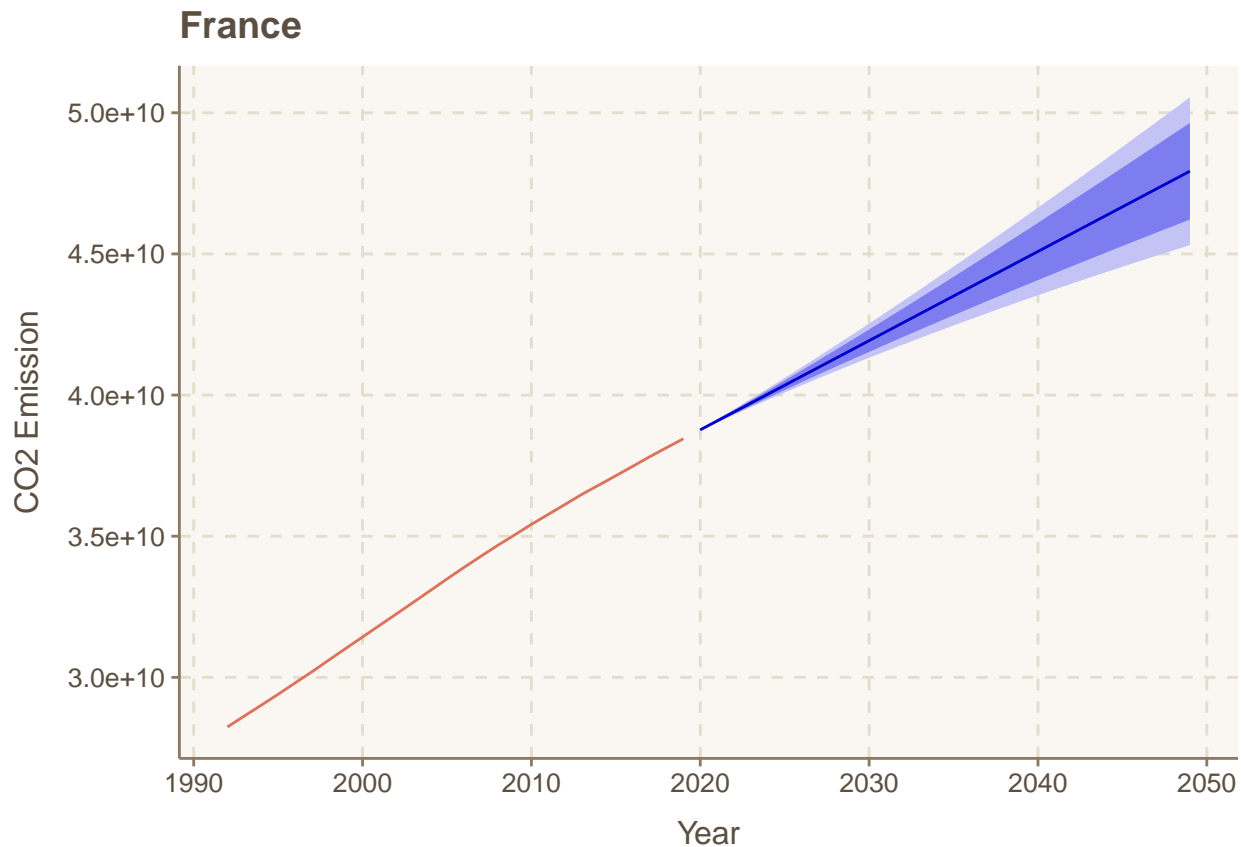
CH.plot



RU.plot



FR.plot



Next, plot the predictions on the same scales, from 2.75×10^{10} to 6.5×10^{11} , to compare the predicted CO2 emissions among the six countries.

```
# devtools::install_github('cttobin/ggthemr')
US.scaled.plot<-autoplot(pred.us)+
  ggtitle("United States")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))+
  ylim(c(2.75e+10, 6.5e+11))

UK.scaled.plot<-autoplot(pred.uk)+
  ggtitle("United Kingdom")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))+
  ylim(c(2.75e+10, 6.5e+11))

GE.scaled.plot<-autoplot(pred.ge)+
  ggtitle("Germany")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))+
  ylim(c(2.75e+10, 6.5e+11))

CH.scaled.plot<-autoplot(pred.ch)+
  ggtitle("China")+
```

```

labs(x = "Year", y = "CO2 Emission")+
theme(axis.title.x = element_text(vjust = 0),
      axis.title.y = element_text(vjust = 2))+
ylim(c(2.75e+10, 6.5e+11))

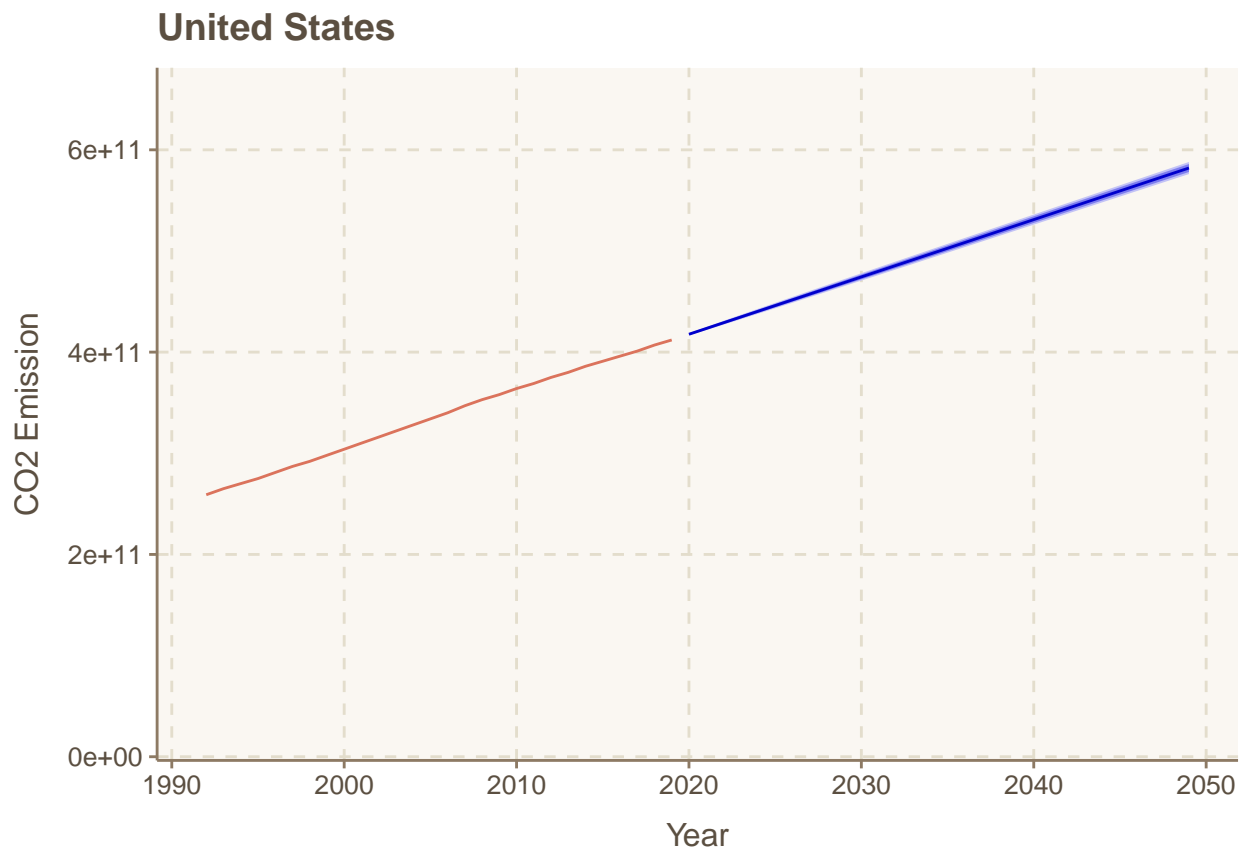
RU.scaled.plot<-autoplot(pred.ru)+
ggtitle("Russia")+
labs(x = "Year", y = "CO2 Emission")+
theme(axis.title.x = element_text(vjust = 0),
      axis.title.y = element_text(vjust = 2))+
ylim(c(2.75e+10, 6.5e+11))

FR.scaled.plot<-autoplot(pred.fr)+
ggtitle("France")+
labs(x = "Year", y = "CO2 Emission")+
theme(axis.title.x = element_text(vjust = 0),
      axis.title.y = element_text(vjust = 2))+
ylim(c(2.75e+10, 6.5e+11))

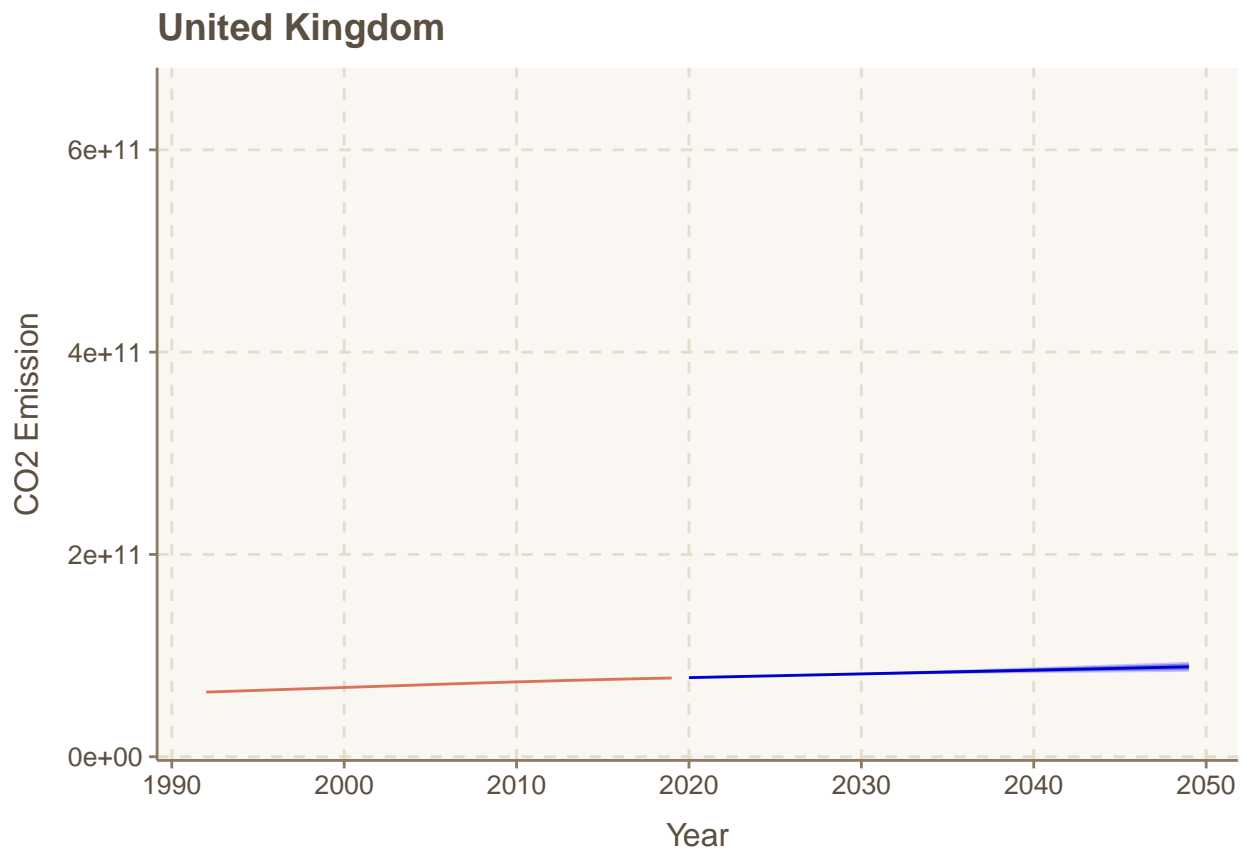
# Save the plots
ggsave("US.scaled.png",US.scaled.plot,width = 10,height = 7.5)
ggsave("UK.scaled.png",UK.scaled.plot,width = 10,height = 7.5)
ggsave("GE.scaled.png",GE.scaled.plot,width = 10,height = 7.5)
ggsave("CH.scaled.png",CH.scaled.plot,width = 10,height = 7.5)
ggsave("RU.scaled.png",RU.scaled.plot,width = 10,height = 7.5)
ggsave("FR.scaled.png",FR.scaled.plot,width = 10,height = 7.5)

US.scaled.plot

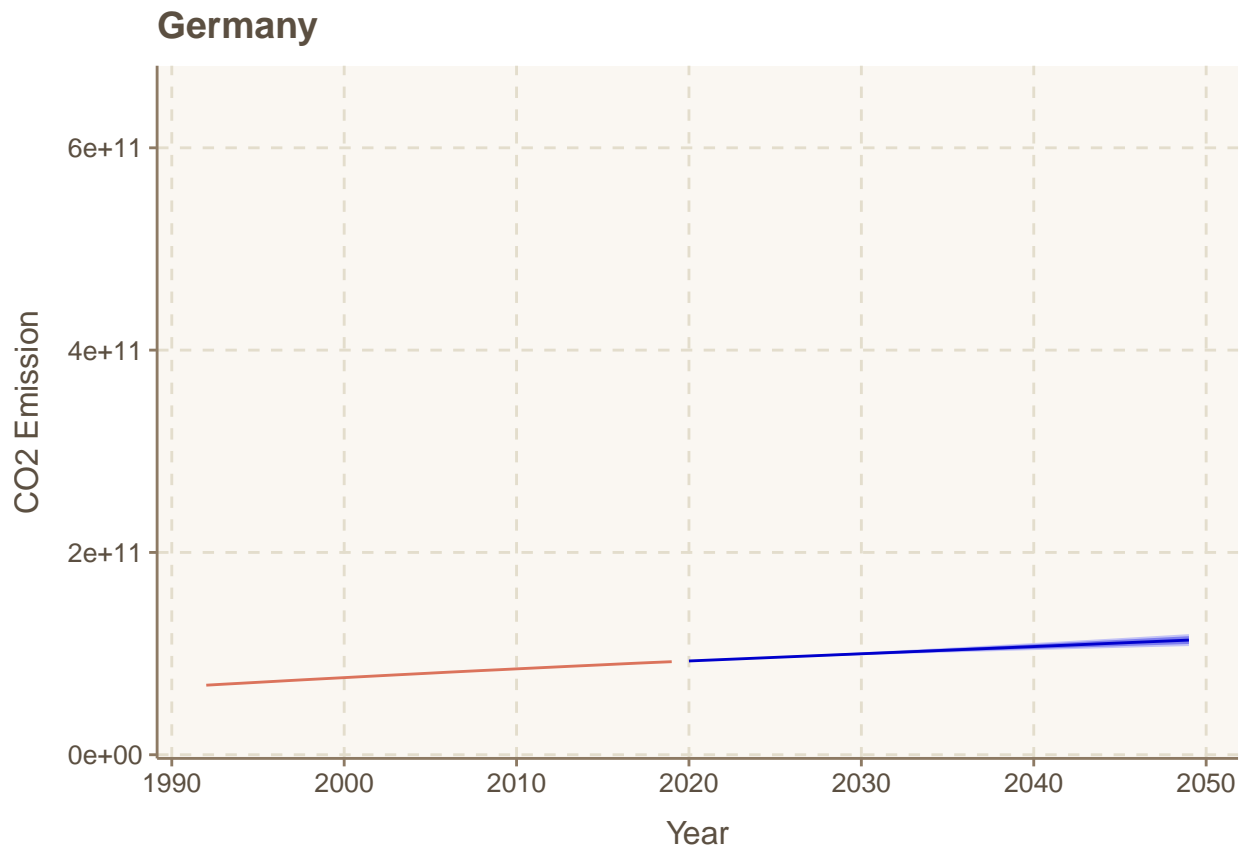
```

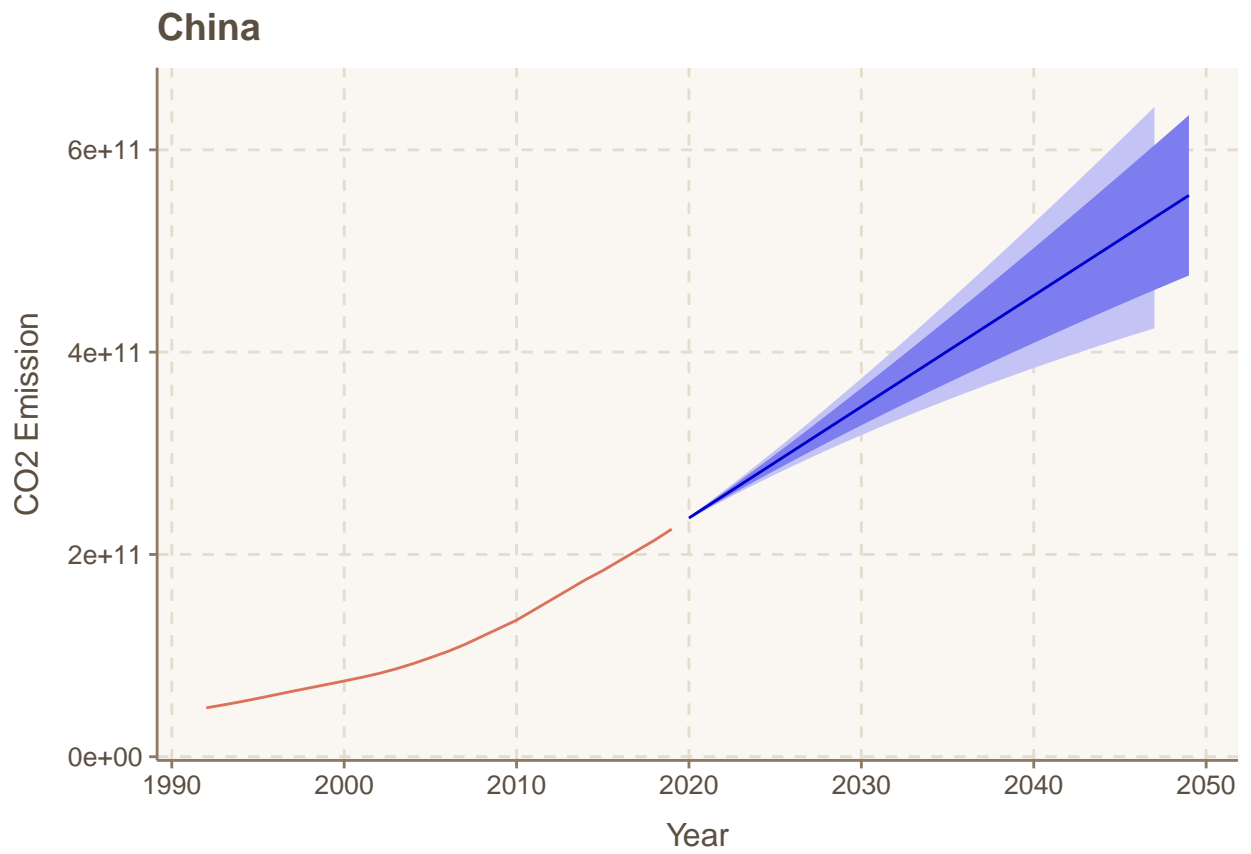
UK.scaled.plot



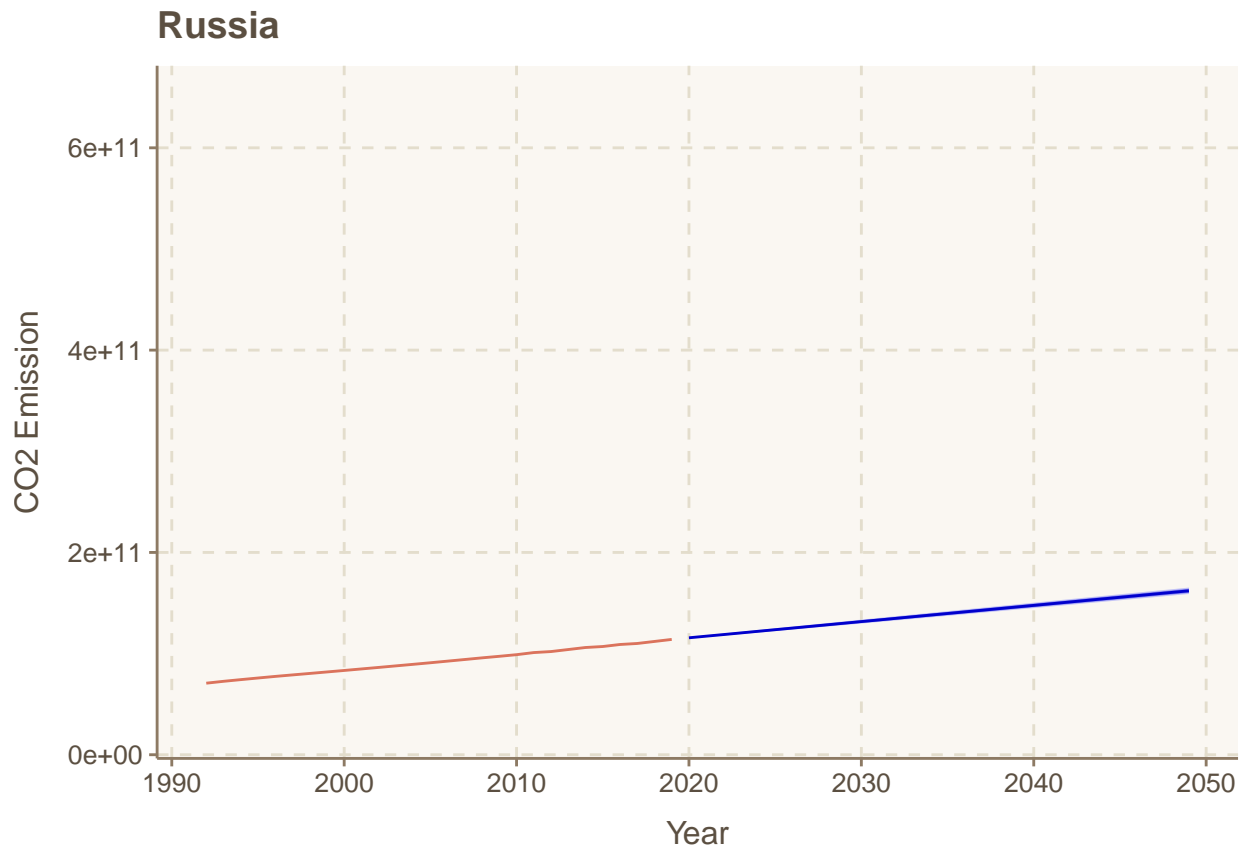
GE.scaled.plot



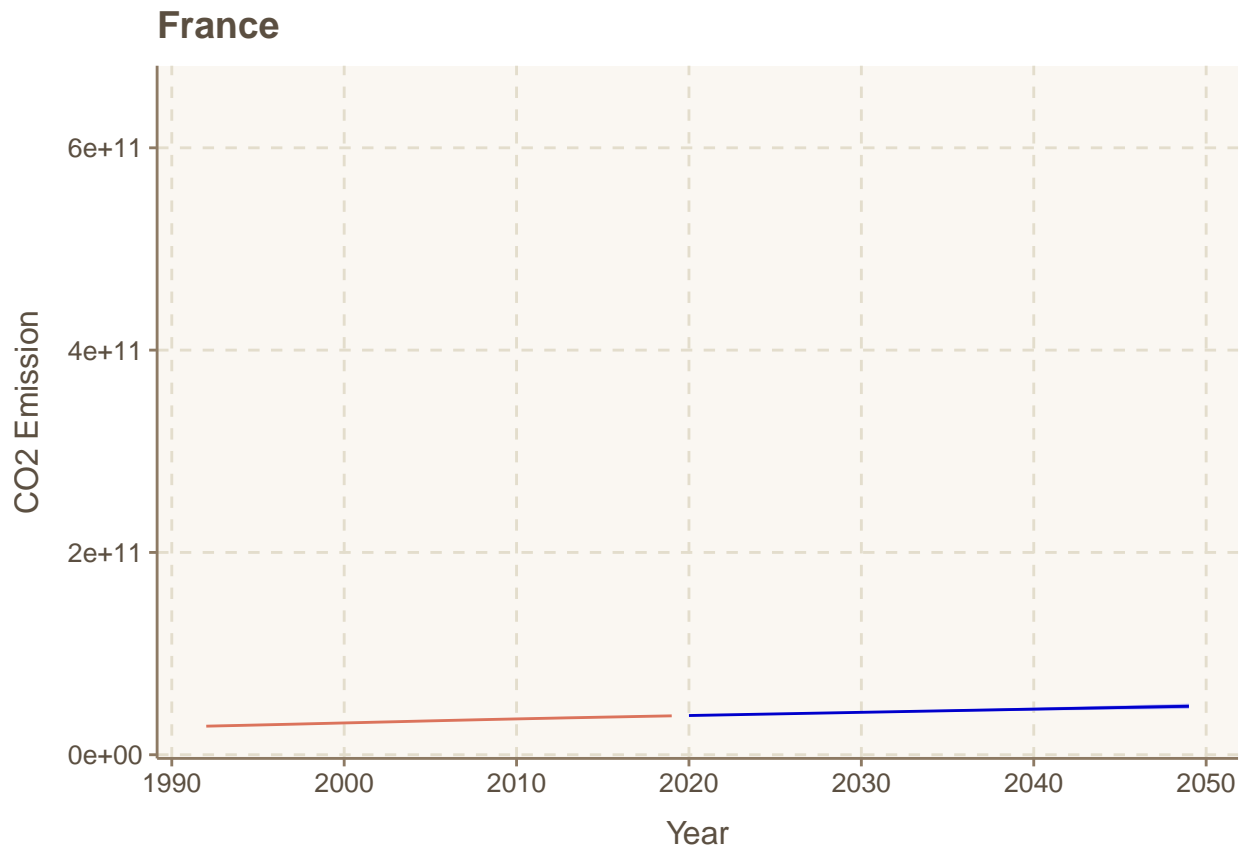
CH.scaled.plot



RU.scaled.plot



FR.scaled.plot



Predictions on continents

```
CO2$Continent <- countrycode(sourcevar = CO2[, "Country"],
                             origin = "country.name",
                             destination = "continent")
```

Time series analysis on five continents

```
## Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, : Some va
```

```
CO2_forecasting<-CO2 %>%
  dplyr::select(Country, Year, CO2.emission..Tons.,Continent)
# make sure there is no NA
dim(CO2_forecasting)
```

```
## [1] 55284      4
```

```
dim(na.omit(CO2_forecasting))
```

```
## [1] 54742      4
```

```
# Use 1992-2020 data to forecast
CO2_no_year_min[which(CO2_no_year_min$Year==max(CO2_no_year_min[]$Year)),]
```

```
##      Country Year
## 1:   Kosovo 2008
```

```
CO2_forecasting<-CO2_forecasting %>%
  filter(Year>1991) %>%
  group_by(Country)
# select continent
CO2_forecasting_continent<-CO2_forecasting %>%
  dplyr::group_by(Year, Continent) %>%
  dplyr::summarise(Sum=sum(CO2.emission..Tons.))
```

```
## 'summarise()' has grouped output by 'Year'. You can override using the
## '.groups' argument.
```

```
CO2_forecasting_continent<-na.omit(CO2_forecasting_continent)
# View(CO2_forecasting_continent)
```

```
Asia<-CO2_forecasting_continent %>%
  filter(Continent=="Asia")
Europe<-CO2_forecasting_continent %>%
  filter(Continent=="Europe")
Africa<-CO2_forecasting_continent %>%
  filter(Continent=="Africa")
Americas<-CO2_forecasting_continent %>%
  filter(Continent=="Americas")
Oceania<-CO2_forecasting_continent %>%
  filter(Continent=="Oceania")

# forecasting
# Asia
ts_data_as = ts(Asia$Sum,start=1992,frequency = 1)
arma.ts.as = auto.arima(ts_data_as)
pred.as = forecast(arma.ts.as,h=30)
# EU
ts_data_eu = ts(Europe$Sum,start=1992,frequency = 1)
arma.ts.eu = auto.arima(ts_data_eu)
pred.eu = forecast(arma.ts.eu,h=30)
# AF
ts_data_af = ts(Africa$Sum,start=1992,frequency = 1)
arma.ts.af = auto.arima(ts_data_af)
pred.af = forecast(arma.ts.af,h=30)
# AM
ts_data_am = ts(Americas$Sum,start=1992,frequency = 1)
arma.ts.am = auto.arima(ts_data_am)
pred.am = forecast(arma.ts.am,h=30)
# OC
ts_data_oc = ts(Oceania$Sum,start=1992,frequency = 1)
arma.ts.oc = auto.arima(ts_data_oc)
pred.oc = forecast(arma.ts.oc,h=30)
```

Plot the predicted CO2 emissions for the five continents We expect to see what's the increasing or decreasing trend of the CO2 emissions in the next 30 years.

```
Asia.plot<-autoplot(pred.as)+
  ggtitle("Asia")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

Europe.plot<-autoplot(pred.eu)+
  ggtitle("Europe")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

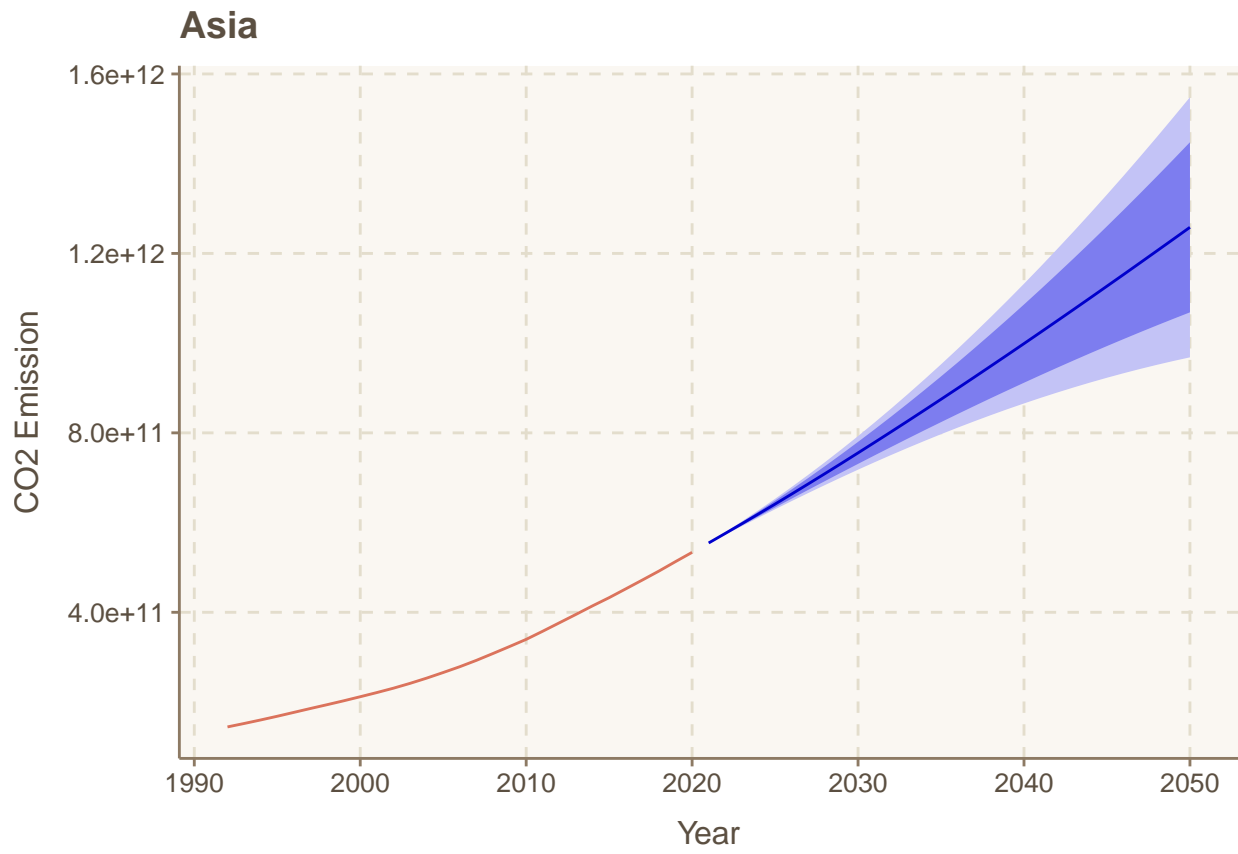
Africa.plot<-autoplot(pred.af)+
  ggtitle("Africa")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

Americas.plot<-autoplot(pred.am)+
  ggtitle("Americas")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

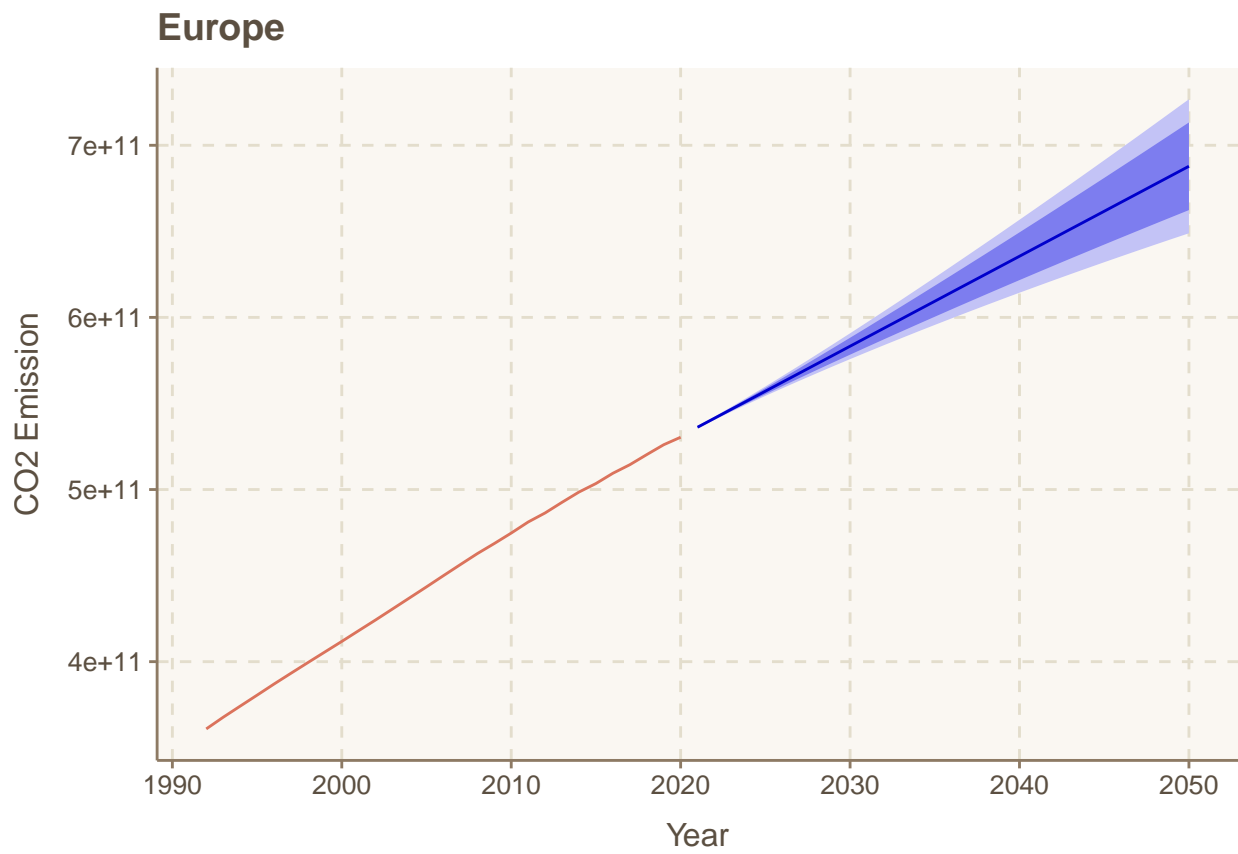
Oceania.plot<-autoplot(pred.oc)+
  ggtitle("Oceania")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))

# save the plots
ggsave("Oceania.png",Oceania.plot,width = 10,height = 7.5)
ggsave("Asia.png",Asia.plot,width = 10,height = 7.5)
ggsave("Europe.png",Europe.plot,width = 10,height = 7.5)
ggsave("Africa.png",Africa.plot,width = 10,height = 7.5)
ggsave("Americas.png",Americas.plot,width = 10,height = 7.5)

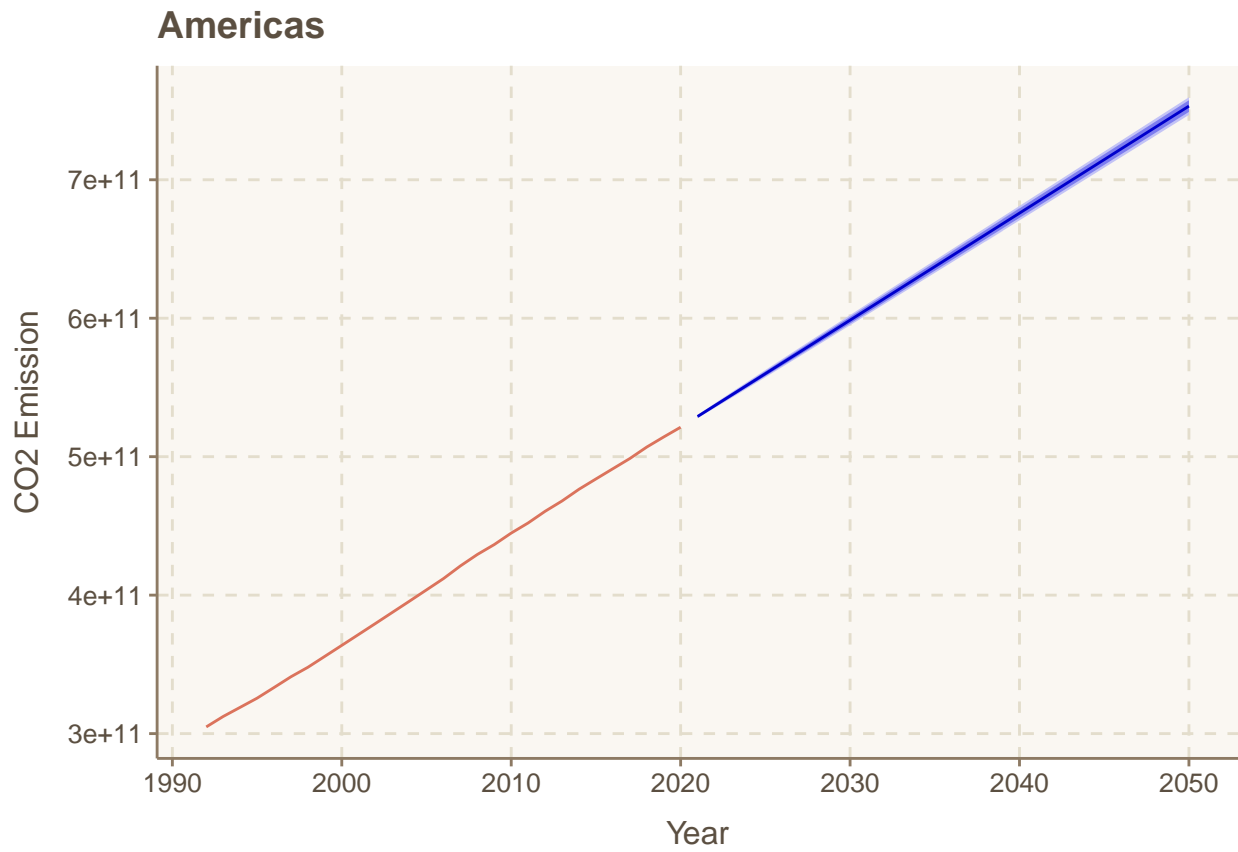
Asia.plot
```

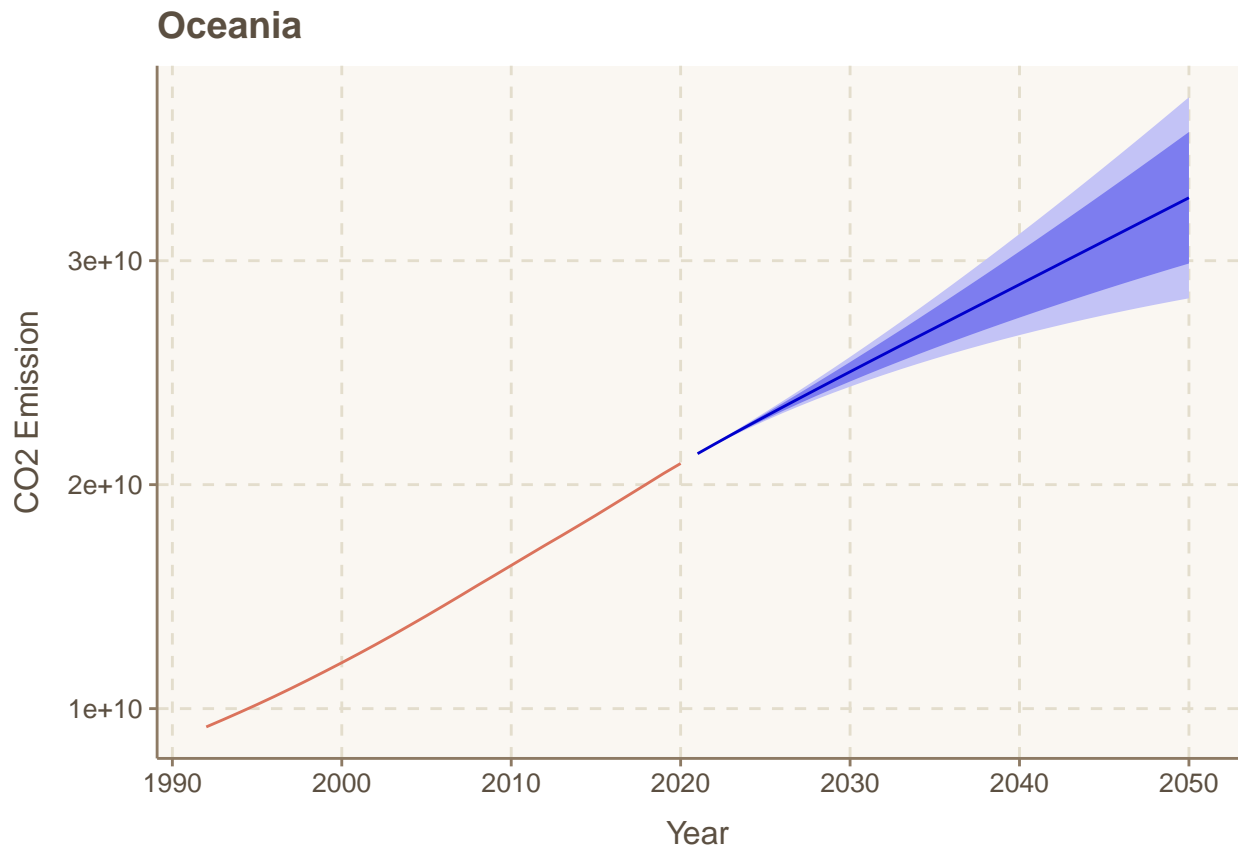
Europe.plot



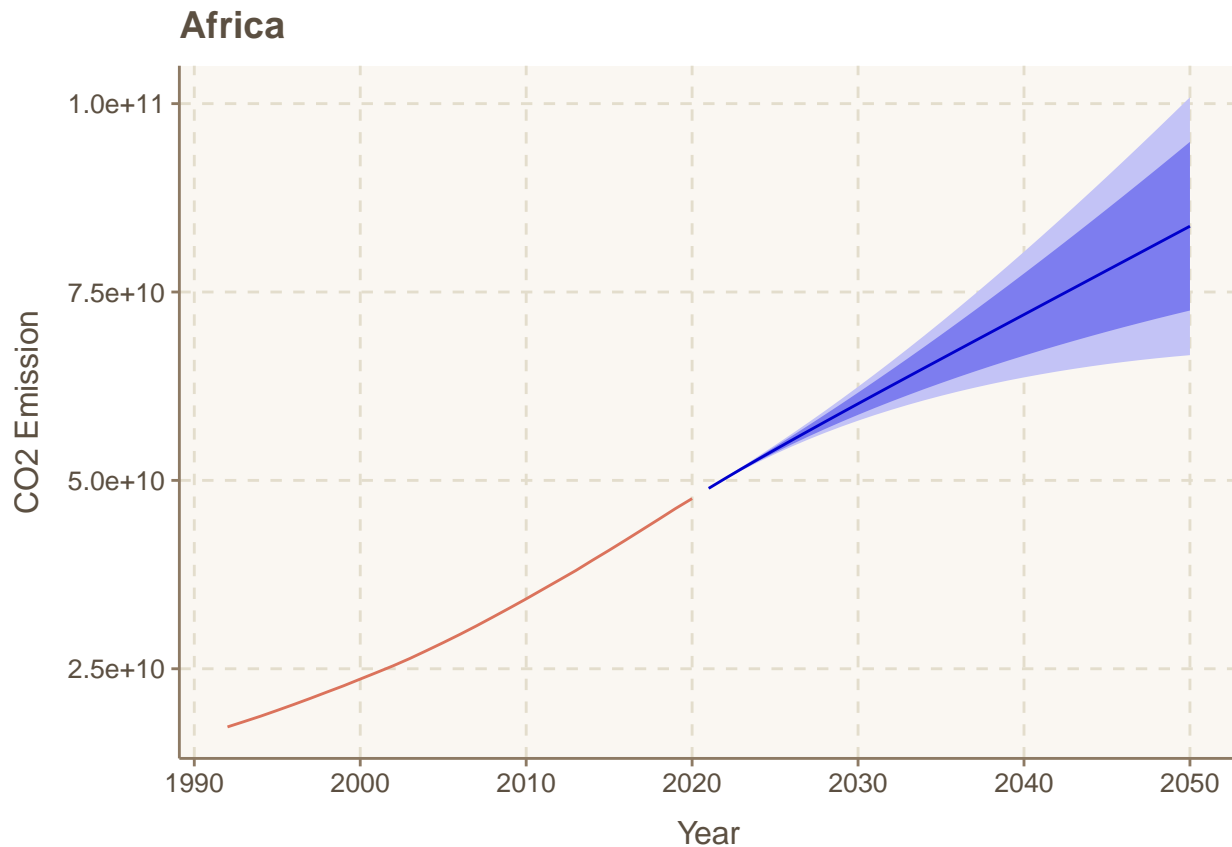
Americas.plot



Oceania.plot



Africa.plot



Next, plot the predictions on the same scales, from 1e+10 to 2e+12, to compare the predicted CO2 emissions among the five continents.

```
Asia.scaled.plot<-autoplot(pred.as)+
  ggtitle("Asia")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))+
  ylim(c(1e+10,2e+12))

Europe.scaled.plot<-autoplot(pred.eu)+
  ggtitle("Europe")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))+
  ylim(c(1e+10,2e+12))

Africa.scaled.plot<-autoplot(pred.af)+
  ggtitle("Africa")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))+
  ylim(c(1e+10,2e+12))

Americas.scaled.plot<-autoplot(pred.am)+
  ggtitle("Americas")+
  labs(x = "Year", y = "CO2 Emission")+
  theme(axis.title.x = element_text(vjust = 0),
        axis.title.y = element_text(vjust = 2))+
  ylim(c(1e+10,2e+12))
```

```

theme(axis.title.x = element_text(vjust = 0),
      axis.title.y = element_text(vjust = 2))+
ylim(c(1e+10,2e+12))

Oceania.scaled.plot<-autoplot(pred.oc)+
ggtitle("Oceania")+
labs(x = "Year", y = "CO2 Emission")+
theme(axis.title.x = element_text(vjust = 0),
      axis.title.y = element_text(vjust = 2))+
ylim(c(1e+10,2e+12))

# save the plots
ggsave("Oceania.scaled.png",Oceania.scaled.plot,width = 10,height = 7.5)

```

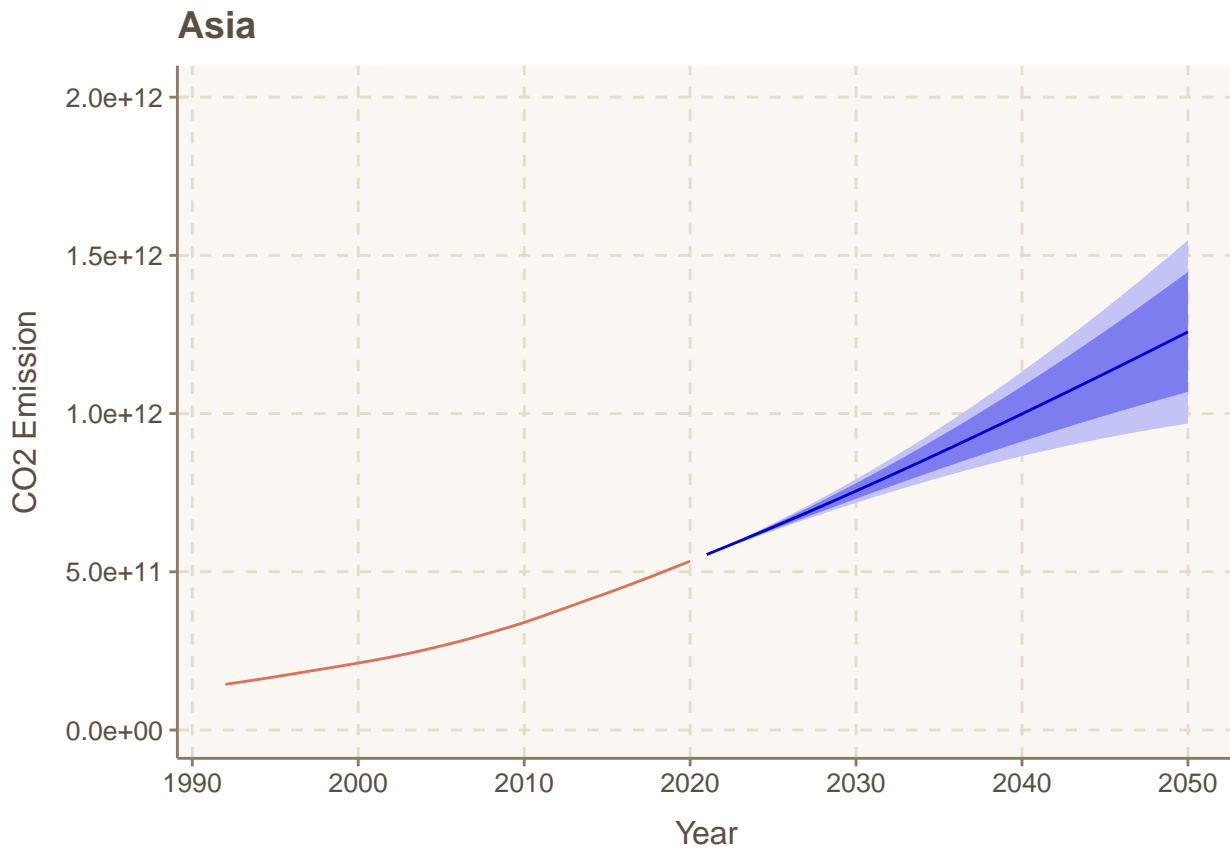
Warning: Removed 3 rows containing missing values (‘geom_line()’).

```

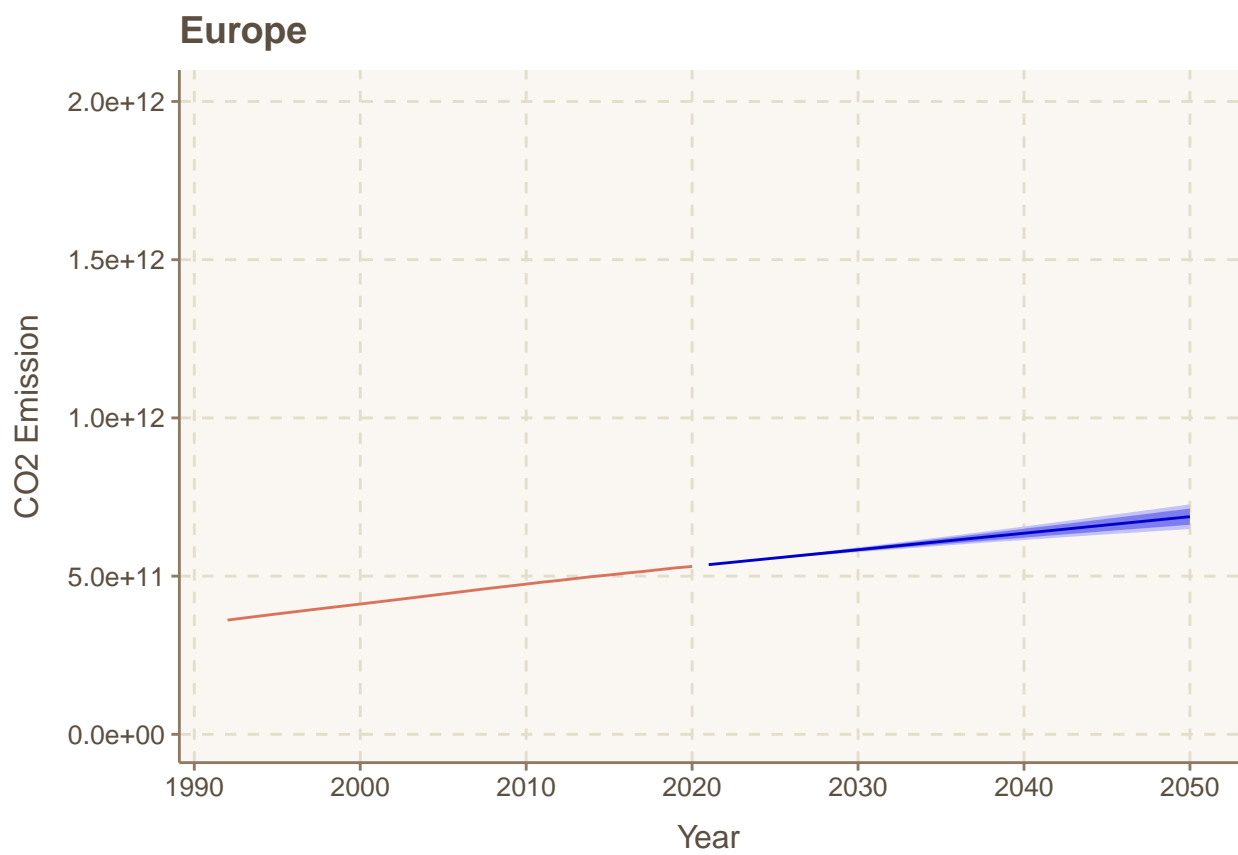
ggsave("Asia.scaled.png",Asia.scaled.plot,width = 10,height = 7.5)
ggsave("Europe.scaled.png",Europe.scaled.plot,width = 10,height = 7.5)
ggsave("Africa.scaled.png",Africa.scaled.plot,width = 10,height = 7.5)
ggsave("Americas.scaled.png",Americas.scaled.plot,width = 10,height = 7.5)

```

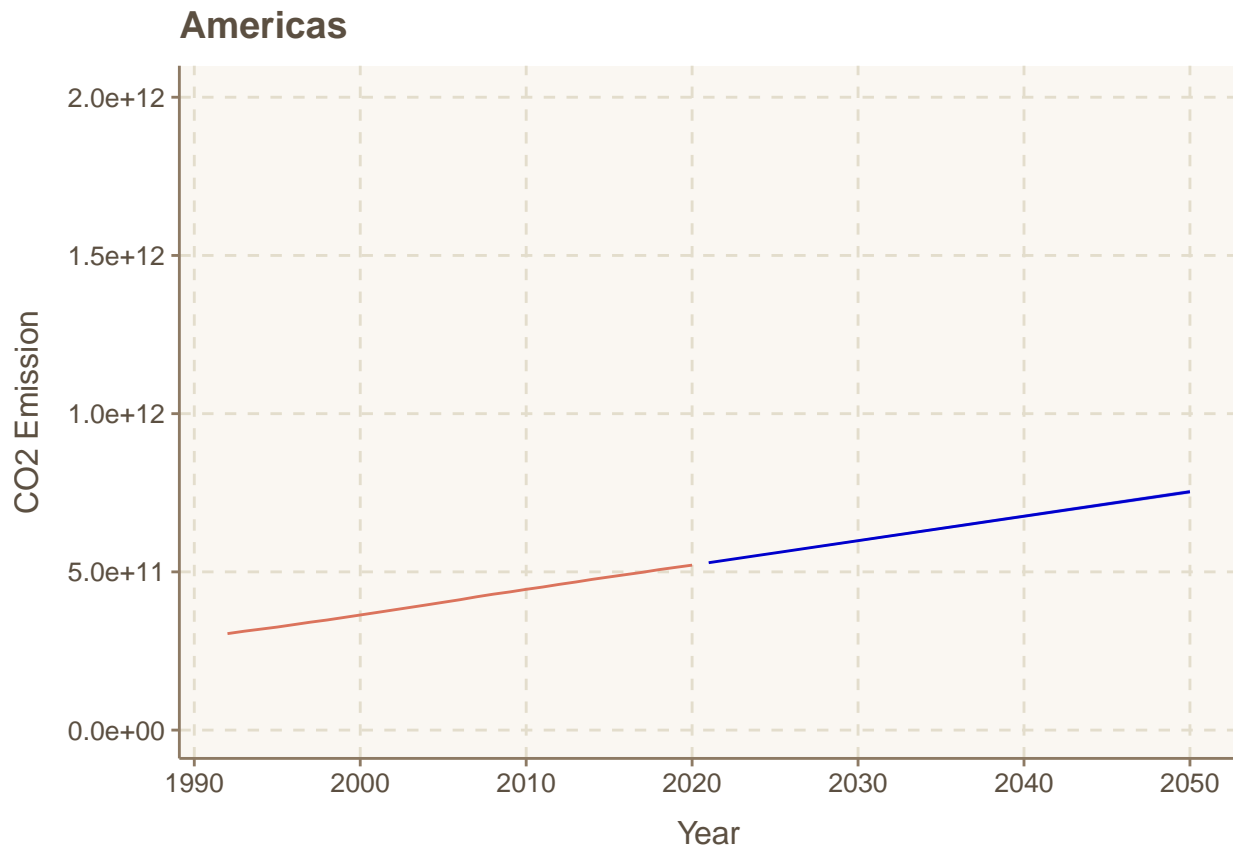
Asia.scaled.plot



Europe.scaled.plot

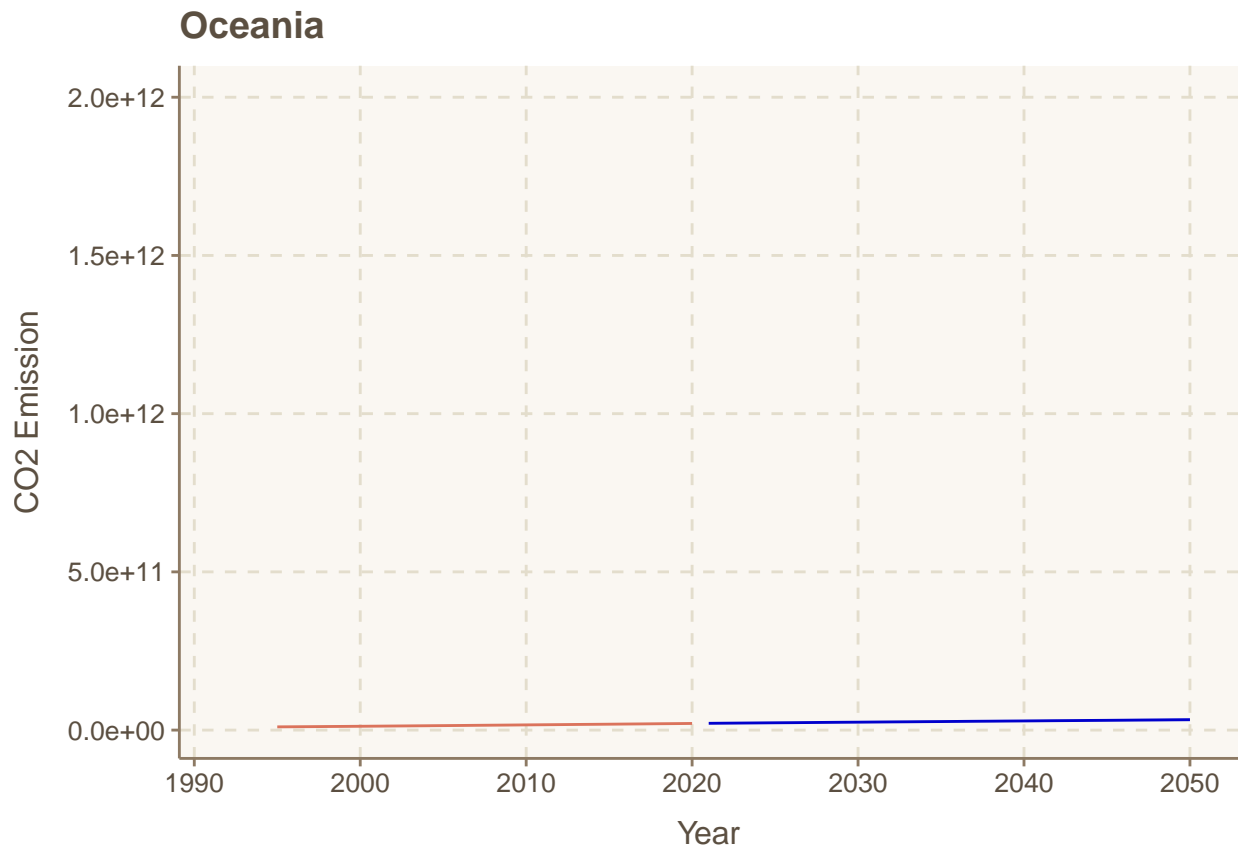


Americas.scaled.plot



```
Oceania.scaled.plot
```

```
## Warning: Removed 3 rows containing missing values ('geom_line()').
```

Africa.scaled.plot

