# Canada_GDP

## Including Plots

You can also embed plots, for example:

```
# Load required packages
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ forcats   1.0.0     ✔ readr     2.1.4
## ✔ ggplot2   3.4.3     ✔ stringr   1.5.0
## ✔ lubridate 1.9.2     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# Load the data (replace 'your_data.csv' with your actual data file name)
data <- read.csv("C:/Users/Admin/Desktop/R DA Projects-portfolio/CAN-GDP/CAN-GDP.csv")
head(data)
```

```
##   REF_DATE                       GEO       DGUID                 Prices
## 1     2010 Newfoundland and Labrador 2016A000210 Chained (2012) dollars
## 2     2011 Newfoundland and Labrador 2016A000210 Chained (2012) dollars
## 3     2012 Newfoundland and Labrador 2016A000210 Chained (2012) dollars
## 4     2013 Newfoundland and Labrador 2016A000210 Chained (2012) dollars
## 5     2014 Newfoundland and Labrador 2016A000210 Chained (2012) dollars
## 6     2015 Newfoundland and Labrador 2016A000210 Chained (2012) dollars
##                       Estimates     UOM UOM_ID SCALAR_FACTOR SCALAR_ID
## 1 Final consumption expenditure Dollars     81      millions         6
## 2 Final consumption expenditure Dollars     81      millions         6
## 3 Final consumption expenditure Dollars     81      millions         6
## 4 Final consumption expenditure Dollars     81      millions         6
## 5 Final consumption expenditure Dollars     81      millions         6
## 6 Final consumption expenditure Dollars     81      millions         6
##      VECTOR COORDINATE VALUE STATUS SYMBOL TERMINATED DECIMALS
## 1 v62787341      2.1.1 21779            NA         NA        0
## 2 v62787341      2.1.1 22169            NA         NA        0
## 3 v62787341      2.1.1 22448            NA         NA        0
## 4 v62787341      2.1.1 22730            NA         NA        0
## 5 v62787341      2.1.1 22869            NA         NA        0
## 6 v62787341      2.1.1 23200            NA         NA        0
```

```r
# Select and rename columns
selected_data <- data %>%
  select(REF_DATE, GEO, Estimates, VALUE) %>%
  rename(Year = REF_DATE,
         Province = GEO,
         Measure = Estimates,
         Millions = VALUE)

# Show the first few rows of the selected and renamed data
head(selected_data)
```

```
##   Year                  Province                       Measure Millions
## 1 2010 Newfoundland and Labrador Final consumption expenditure    21779
## 2 2011 Newfoundland and Labrador Final consumption expenditure    22169
## 3 2012 Newfoundland and Labrador Final consumption expenditure    22448
## 4 2013 Newfoundland and Labrador Final consumption expenditure    22730
## 5 2014 Newfoundland and Labrador Final consumption expenditure    22869
## 6 2015 Newfoundland and Labrador Final consumption expenditure    23200
```

```r
unique(data$GEO)
```

```
##  [1] "Newfoundland and Labrador"
##  [2] "Prince Edward Island"
##  [3] "Nova Scotia"
##  [4] "New Brunswick"
##  [5] "Quebec"
##  [6] "Ontario"
##  [7] "Manitoba"
##  [8] "Saskatchewan"
##  [9] "Alberta"
## [10] "British Columbia"
## [11] "Yukon"
## [12] "Northwest Territories including Nunavut"
## [13] "Northwest Territories"
## [14] "Nunavut"
## [15] "Outside Canada"
```

```
################################################################################
```

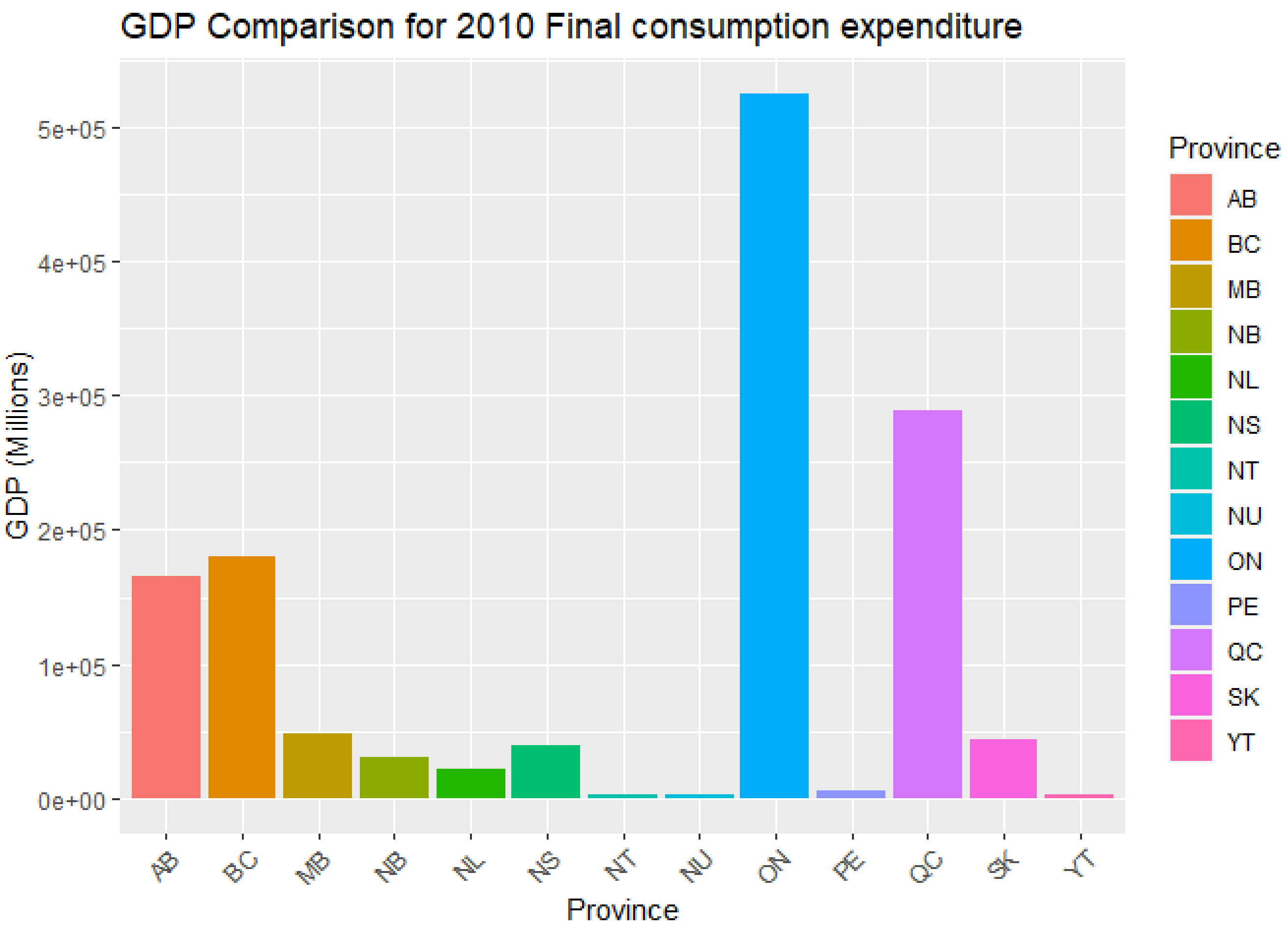
```
# Define a mapping of province names to their abbreviations
province_mapping <- c("Alberta" = "AB",
                      "British Columbia" = "BC",
                      "Manitoba" = "MB",
                      "New Brunswick" = "NB",
                      "Newfoundland and Labrador" = "NL",
                      "Northwest Territories" = "NT",
                      "Nova Scotia" = "NS",
                      "Nunavut" = "NU",
                      "Ontario" = "ON",
                      "Prince Edward Island" = "PE",
                      "Quebec" = "QC",
                      "Saskatchewan" = "SK",
                      "Yukon" = "YT")

# Replace province names with abbreviations and filter out "Outside Canada"
final_data <- selected_data %>%
  mutate(Province = ifelse(Province == "Outside Canada", NA, Province)) %>%
  drop_na() %>%
  mutate(Province = province_mapping[Province])


# Show the first few rows of the final data
head(final_data)
```

```
##   Year Province                       Measure Millions
## 1 2010       NL Final consumption expenditure    21779
## 2 2011       NL Final consumption expenditure    22169
## 3 2012       NL Final consumption expenditure    22448
## 4 2013       NL Final consumption expenditure    22730
## 5 2014       NL Final consumption expenditure    22869
## 6 2015       NL Final consumption expenditure    23200
```

```
###############################################################################
#Calculate Total GDP for Each Year
total_gdp_by_year <- final_data %>%
  group_by(Year) %>%
  summarize(Total_GDP = sum(Millions))

print(total_gdp_by_year)
```

```
## # A tibble: 11 × 2
##     Year Total_GDP
##    <int>     <int>
##  1  2010   2643330
##  2  2011   2722814
##  3  2012   2766273
##  4  2013   2838688
##  5  2014   2920336
##  6  2015   2959855
##  7  2016   3004614
##  8  2017   3117072
##  9  2018   3208741
## 10  2019   3258106
## 11  2020   3074167
```

```
#Compare GDP between Provinces:
library(ggplot2)

year_to_compare <- 2010
measure_to_compare <- "Final consumption expenditure"

gdp_comparison_plot <- final_data %>%
  filter(Year == year_to_compare, Measure == measure_to_compare) %>%
  ggplot(aes(x = Province, y = Millions, fill = Province)) +
  geom_bar(stat = "identity") +
  labs(title = paste("GDP Comparison for", year_to_compare, measure_to_compare),
       x = "Province",
       y = "GDP (Millions)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(gdp_comparison_plot)
```



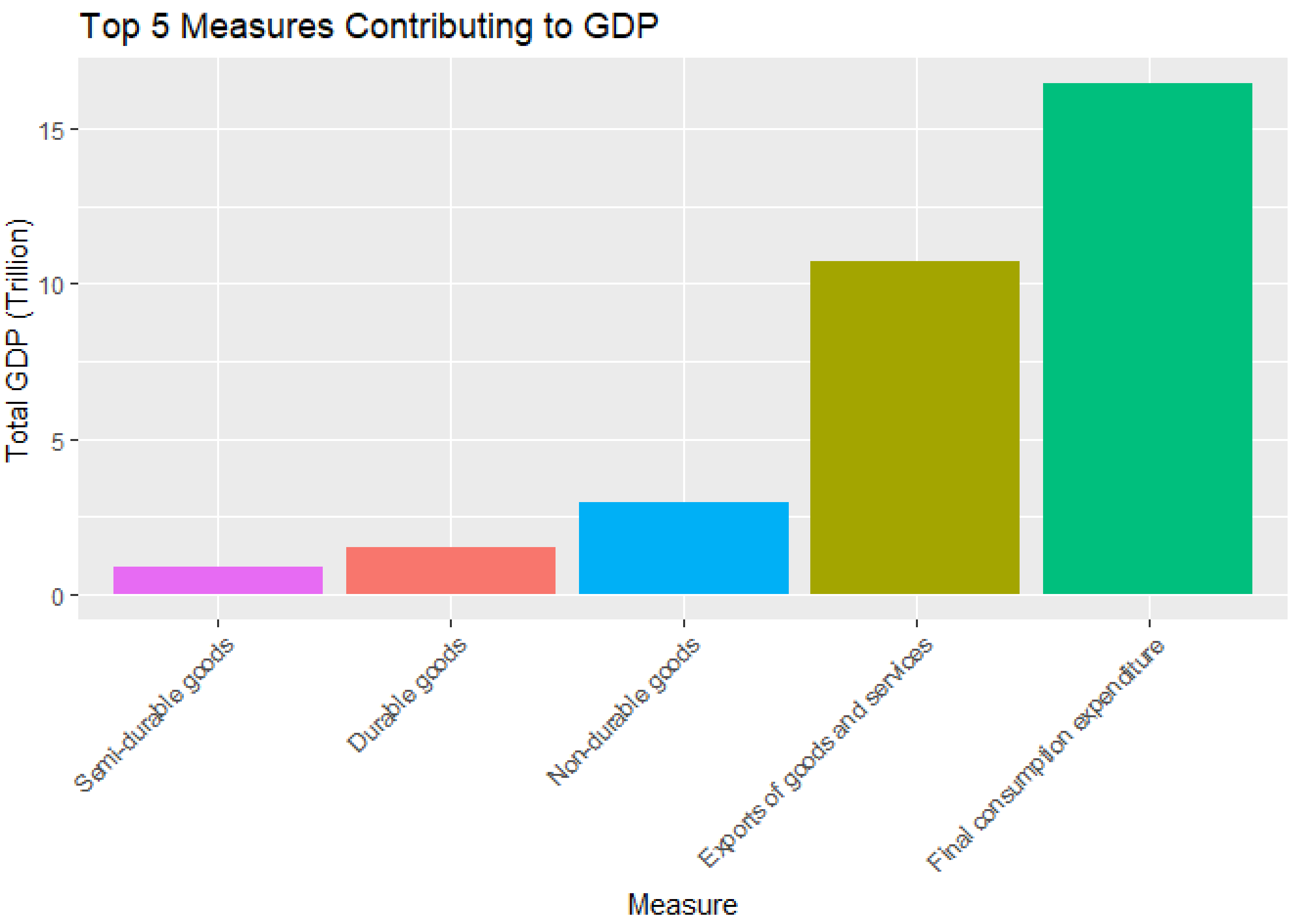
```
###############################################################################
#Identify Top Measures for NB
province_to_analyze <- "NB"

top_measures_for_province <- final_data %>%
  filter(Province == province_to_analyze) %>%
  group_by(Measure) %>%
  summarize(Total_GDP = sum(Millions)) %>%
  arrange(desc(Total_GDP))
```

```r
print(top_measures_for_province)
```

```
## # A tibble: 6 × 2
##   Measure                        Total_GDP
##   <chr>                              <int>
## 1 Final consumption expenditure     349749
## 2 Exports of goods and services     278945
## 3 Non-durable goods                  72304
## 4 Durable goods                      32334
## 5 Semi-durable goods                 17112
## 6 Investment in inventories           1333
```

```r
###############################################################################
library(ggplot2)
library(dplyr)

# Calculate total GDP for each measure across all provinces
top_measures_all_provinces <- final_data %>%
  group_by(Measure) %>%
  summarize(Total_GDP = sum(Millions)) %>%
  arrange(desc(Total_GDP))

# Select the top N measures (you can adjust N as needed)
top_n_measures <- 5

top_measures_for_plot <- top_measures_all_provinces %>%
  top_n(top_n_measures, Total_GDP) %>%
  arrange(Total_GDP)

# Create a bar plot to visualize top measures
ggplot(top_measures_for_plot, aes(x = reorder(Measure, Total_GDP), y = Total_GDP, fill = Measure)) +
  geom_bar(stat = "identity") +
  labs(title = paste("Top", top_n_measures, "Measures Contributing to GDP"),
       x = "Measure",
       y = "Total GDP (Trillion)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  guides(fill = FALSE) +  # Remove legend for fill
  scale_y_continuous(labels = scales::comma_format(scale = 1e-6))  # Format y-axis labels in millions
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
###############################################################################
library(ggplot2)
library(dplyr)

# Filter data for the "Exports of goods and services" measure
exports_data <- final_data %>%
  filter(Measure == "Exports of goods and services")

# Calculate total exports for each province
exports_by_province <- exports_data %>%
  group_by(Province, Year) %>%
  summarize(Total_Exports = sum(Millions))
```

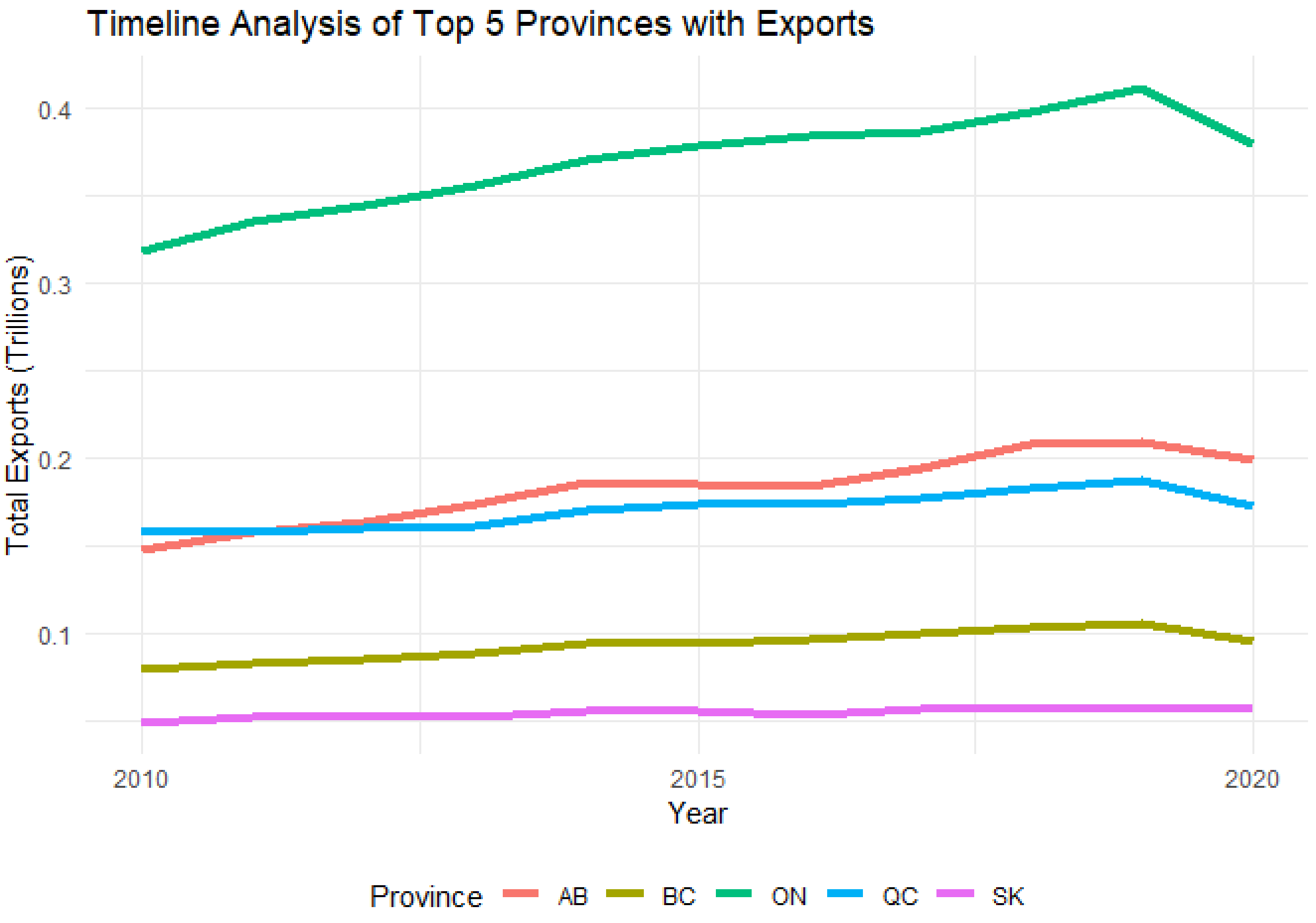
```
## `summarise()` has grouped output by 'Province'. You can override using the
## `.groups` argument.
```

```
# Select the top 5 provinces by total exports
top_n_provinces <- 5
top_exports_provinces <- exports_by_province %>%
  group_by(Province) %>%
  summarize(Total_Exports = sum(Total_Exports)) %>%
  arrange(desc(Total_Exports)) %>%
  top_n(top_n_provinces)
```

```
## Selecting by Total_Exports
```

```
# Create a timeline plot for top 5 provinces
ggplot(exports_by_province %>%
         filter(Province %in% top_exports_provinces$Province),
       aes(x = as.Date(paste0(Year, "-01-01")), y = Total_Exports, color = Province)) +
  geom_line(size = 1.5) +
  labs(title = paste("Timeline Analysis of Top", top_n_provinces, "Provinces with Exports"),
       x = "Year",
       y = "Total Exports (Trillions)") +
  scale_y_continuous(labels = scales::comma_format(scale = 1e-6)) +
  theme_minimal() +
  theme(legend.position = "bottom")  # Adjust legend position
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



##############################################################

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.