Business Forcasting

09/20/2023

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

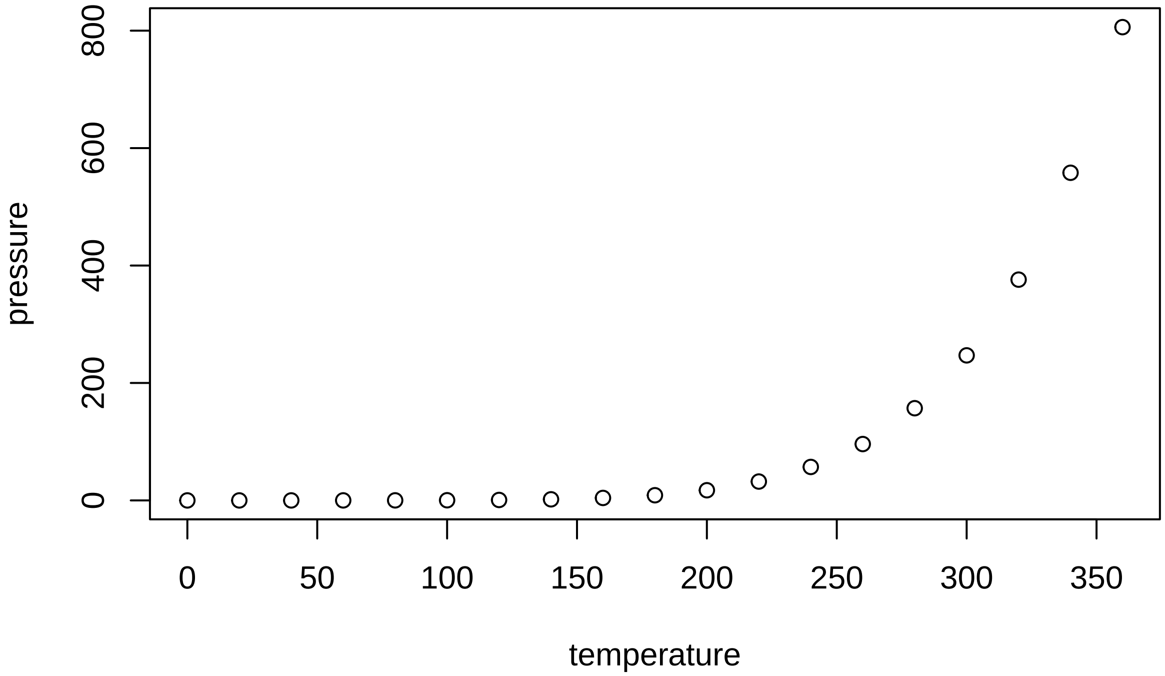
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:



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

We will be using gapminder data to look at info about GDP, life expectancy, and other parameters.

```
remove(list=ls())
library(tidyverse) ; library(gapminder)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
table(gapminder$year)
```

```
##
## 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
##  142  142  142  142  142  142  142  142  142  142  142  142
```

```
table(gapminder$country)
```

```
##
##              Afghanistan                  Albania                  Algeria
##                       12                       12                       12
##                   Angola                Argentina                Australia
##                       12                       12                       12
##                  Austria                  Bahrain               Bangladesh
##                       12                       12                       12
##                  Belgium                    Benin                  Bolivia
##                       12                       12                       12
##   Bosnia and Herzegovina                 Botswana                   Brazil
##                       12                       12                       12
##                 Bulgaria             Burkina Faso                  Burundi
##                       12                       12                       12
##                 Cambodia                 Cameroon                   Canada
##                       12                       12                       12
## Central African Republic                     Chad                    Chile
##                       12                       12                       12
##                    China                 Colombia                  Comoros
##                       12                       12                       12
##         Congo, Dem. Rep.              Congo, Rep.               Costa Rica
##                       12                       12                       12
##            Cote d'Ivoire                  Croatia                     Cuba
##                       12                       12                       12
##           Czech Republic                  Denmark                 Djibouti
##                       12                       12                       12
##       Dominican Republic                  Ecuador                    Egypt
##                       12                       12                       12
##              El Salvador        Equatorial Guinea                  Eritrea
##                       12                       12                       12
##                 Ethiopia                  Finland                   France
##                       12                       12                       12
##                    Gabon                   Gambia                  Germany
##                       12                       12                       12
##                    Ghana                   Greece                Guatemala 
##                       12                       12                       12 
##                   Guinea            Guinea-Bissau                    Haiti 
##                       12                       12                       12 
##                 Honduras         Hong Kong, China                  Hungary 
##                       12                       12                       12 
##                  Iceland                    India                Indonesia 
##                       12                       12                       12 
##                     Iran                     Iraq                  Ireland 
##                       12                       12                       12 
##                   Israel                    Italy                  Jamaica 
##                       12                       12                       12 
##                    Japan                   Jordan                    Kenya 
##                       12                       12                       12 
##         Korea, Dem. Rep.              Korea, Rep.                   Kuwait 
##                       12                       12                       12 
##                  Lebanon                  Lesotho                  Liberia 
##                       12                       12                       12 
##                    Libya               Madagascar                   Malawi 
##                       12                       12                       12 
##                 Malaysia                     Mali               Mauritania 
##                       12                       12                       12 
##                Mauritius                   Mexico                 Mongolia 
##                       12                       12                       12 
##               Montenegro                  Morocco               Mozambique 
##                       12                       12                       12 
##                  Myanmar                  Namibia                    Nepal 
##                       12                       12                       12 
##              Netherlands              New Zealand                Nicaragua 
##                       12                       12                       12 
##                    Niger                  Nigeria                   Norway 
##                       12                       12                       12 
##                     Oman                 Pakistan                   Panama 
##                       12                       12                       12 
##                 Paraguay                     Peru              Philippines 
##                       12                       12                       12 
##                   Poland                 Portugal              Puerto Rico 
##                       12                       12                       12 
##                  Reunion                  Romania                   Rwanda 
##                       12                       12                       12 
##    Sao Tome and Principe             Saudi Arabia                  Senegal 
##                       12                       12                       12 
##                   Serbia             Sierra Leone                Singapore 
##                       12                       12                       12 
##          Slovak Republic                 Slovenia                  Somalia 
##                       12                       12                       12 
##             South Africa                    Spain                Sri Lanka 
##                       12                       12                       12 
##                    Sudan                Swaziland                   Sweden 
##                       12                       12                       12 
##              Switzerland                    Syria                   Taiwan 
##                       12                       12                       12 
##                 Tanzania                 Thailand                     Togo 
##                       12                       12                       12 
##      Trinidad and Tobago                  Tunisia                   Turkey
##                       12                       12                       12
##                   Uganda           United Kingdom            United States
##                       12                       12                       12
##                  Uruguay                Venezuela                  Vietnam
##                       12                       12                       12
##       West Bank and Gaza              Yemen, Rep.                   Zambia
##                       12                       12                       12
##                 Zimbabwe
##                       12
```

```
table(gapminder$continent)
```

```
##
##   Africa Americas     Asia   Europe  Oceania
##      624      300      396      360       24
```

Find the data for 2002

```
data_2002 <- gapminder %>%
  filter(year == 2002)
data_2002
```

```
## # A tibble: 142 x 6
##    country     continent  year lifeExp       pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>     <int>     <dbl>
##  1 Afghanistan Asia       2002    42.1  25268405      727.
##  2 Albania     Europe     2002    75.7   3508512     4604.
##  3 Algeria     Africa     2002    71.0  31287142     5288.
##  4 Angola      Africa     2002    41.0  10866106     2773.
##  5 Argentina   Americas   2002    74.3  38331121     8798.
##  6 Australia   Oceania    2002    80.4  19546792    30688.
##  7 Austria     Europe     2002    79.0   8148312    32418.
##  8 Bahrain     Asia       2002    74.8    656397    23404.
##  9 Bangladesh  Asia       2002    62.0 135656790     1136.
## 10 Belgium     Europe     2002    78.3  10311970    30486.
## # i 132 more rows
```

Find the data for Brazil in 2002. What is the life expectancy?

Life Expectancy is 71.006

```
brazil_2002b <- gapminder %>%
  filter(year == "2002",country == "Brazil")
brazil_2002b
```

```
## # A tibble: 1 x 6
##   country continent  year lifeExp       pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Brazil  Americas   2002    71.0 179914212     8131.
```

Find which country has the lowest lifeExp.

Country has the lowest lifeExp is Rwanda

```
low_lifeExp <- gapminder %>%
  arrange(lifeExp)
low_lifeExp
```

```
## # A tibble: 1,704 x 6
##    country      continent  year lifeExp     pop gdpPercap
##    <fct>        <fct>     <int>   <dbl>   <int>     <dbl>
##  1 Rwanda       Africa     1992    23.6 7290203      737.
##  2 Afghanistan  Asia       1952    28.8 8425333      779.
##  3 Gambia       Africa     1952    30    284320      485.
##  4 Angola       Africa     1952    30.0 4232095     3521.
##  5 Sierra Leone Africa     1952    30.3 2143249      880.
##  6 Afghanistan  Asia       1957    30.3 9240934      821.
##  7 Cambodia     Asia       1977    31.2 6978607      525.
##  8 Mozambique   Africa     1952    31.3 6446316      469.
##  9 Sierra Leone Africa     1957    31.6 2295678     1004.
## 10 Burkina Faso Africa     1952    32.0 4469979      543.
## # i 1,694 more rows
```

Find which country has the lowest lifeExp in 2002.

Country has the lowest lifeExp in 2002 is Zambia

```
low_LifeExp_2002 <- gapminder %>%
  filter(year == "2002") %>%
  arrange(lifeExp)
low_LifeExp_2002
```

```
## # A tibble: 142 x 6
##    country                  continent  year lifeExp      pop gdpPercap
##    <fct>                    <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Zambia                   Africa     2002    39.2 10595811     1072.
##  2 Zimbabwe                 Africa     2002    40.0 11926563      672.
##  3 Angola                   Africa     2002    41.0 10866106     2773.
##  4 Sierra Leone             Africa     2002    41.0  5359092      699.
##  5 Afghanistan              Asia       2002    42.1 25268405      727.
##  6 Central African Republic Africa     2002    43.3  4048013      739.
##  7 Rwanda                   Africa     2002    43.4  7852401      786.
##  8 Liberia                  Africa     2002    43.8  2814651      531.
##  9 Swaziland                Africa     2002    43.9  1130269     4128.
## 10 Mozambique               Africa     2002    44.0 18473780      634.
## # i 132 more rows
```

Find the lifeExp in Japan in 2002.

The lifeExp in Japan in 2002 is 82

```
japan_2002 <- gapminder %>%
  filter(year == "2002", country == "Japan")
japan_2002
```

```
## # A tibble: 1 x 6
##   country continent  year lifeExp       pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Japan   Asia       2002      82 127065841    28605.
```

Find the countries whose lifeExp is higher than 80 in 2002. How many are there?

Seven

```
higher_lifeExp_2002 <- gapminder %>%
  filter(year == "2002",lifeExp > 80 ) %>%
arrange(lifeExp)
higher_lifeExp_2002
```

```
## # A tibble: 7 x 6
##   country          continent  year lifeExp       pop gdpPercap
##   <fct>            <fct>     <int>   <dbl>     <int>     <dbl>
## 1 Sweden           Europe     2002    80.0   8954175    29342.
## 2 Italy            Europe     2002    80.2  57926999    27968.
## 3 Australia        Oceania    2002    80.4  19546792    30688.
## 4 Iceland          Europe     2002    80.5    288030    31163.
## 5 Switzerland      Europe     2002    80.6   7361757    34481.
## 6 Hong Kong, China Asia       2002    81.5   6762476    30209.
## 7 Japan            Asia       2002    82   127065841    28605.
```

Find the lifeExp in Europe across the years. Which year is the highest lifeExp in Europe?

the highest lifeExp in Europe is 2007

```
europe_lifeExp_high = gapminder %>%
  filter(continent == "Europe") %>%
arrange(desc(lifeExp))
europe_lifeExp_high
```

```
## # A tibble: 360 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Iceland     Europe     2007    81.8   301931    36181.
##  2 Switzerland Europe     2007    81.7  7554661    37506.
##  3 Spain       Europe     2007    80.9 40448191    28821.
##  4 Sweden      Europe     2007    80.9  9031088    33860.
##  5 France      Europe     2007    80.7 61083916    30470.
##  6 Switzerland Europe     2002    80.6  7361757    34481.
##  7 Italy       Europe     2007    80.5 58147733    28570.
##  8 Iceland     Europe     2002    80.5   288030    31163.
##  9 Italy       Europe     2002    80.2 57926999    27968.
## 10 Norway      Europe     2007    80.2  4627926    49357.
## # i 350 more rows
```

8 Define gdp as it is equal to to gdpPercap * pop/10000 . Find the gdp of Europe in 2002.

```
gdp_europe <- gapminder %>%
  filter(continent == 'Europe', year == '2002') %>%
  mutate(gdp = gdpPercap * pop / 10000) %>%
  arrange(desc(gdp))
gdp_europe
```

```
## # A tibble: 30 x 7
##    country        continent  year lifeExp      pop gdpPercap        gdp
##    <fct>          <fct>     <int>   <dbl>    <int>     <dbl>      <dbl>
##  1 Germany        Europe     2002    78.7 82350671    30036. 247346845.
##  2 United Kingdom Europe     2002    78.5 59912431    29479. 176615850.
##  3 France         Europe     2002    79.6 59925035    28926. 173339350.
##  4 Italy          Europe     2002    80.2 57926999    27968. 162010799.
##  5 Spain          Europe     2002    79.8 40152517    24835.  99720670.
##  6 Netherlands    Europe     2002    78.5 16122830    33725.  54373854.
##  7 Poland         Europe     2002    74.7 38625976    12002.  46359820.
##  8 Turkey         Europe     2002    70.8 67308928     6508.  43805227.
##  9 Belgium        Europe     2002    78.3 10311970    30486.  31436952.
## 10 Austria        Europe     2002    79.0  8148312    32418.  26414878.
## # i 20 more rows
```

9 Which country has the highest gdp in Europe in 2002?

The highest gdp in Europe in 2002 is Italy

```
high_gdp_europe <- gapminder %>% filter(year==2002, continent=="europe") %>% mutate(gdp=gdpPercap*pop/1
high_gdp_europe
```

```
## # A tibble: 0 x 7
## # i 7 variables: country <fct>, continent <fct>, year <int>, lifeExp <dbl>,
## #   pop <int>, gdpPercap <dbl>, gdp <dbl>
```

```
gapminder %>% filter(year==2002, continent=="Europe") %>% mutate(gdp=gdpPercap*pop/10000) %>% arrange(d
```

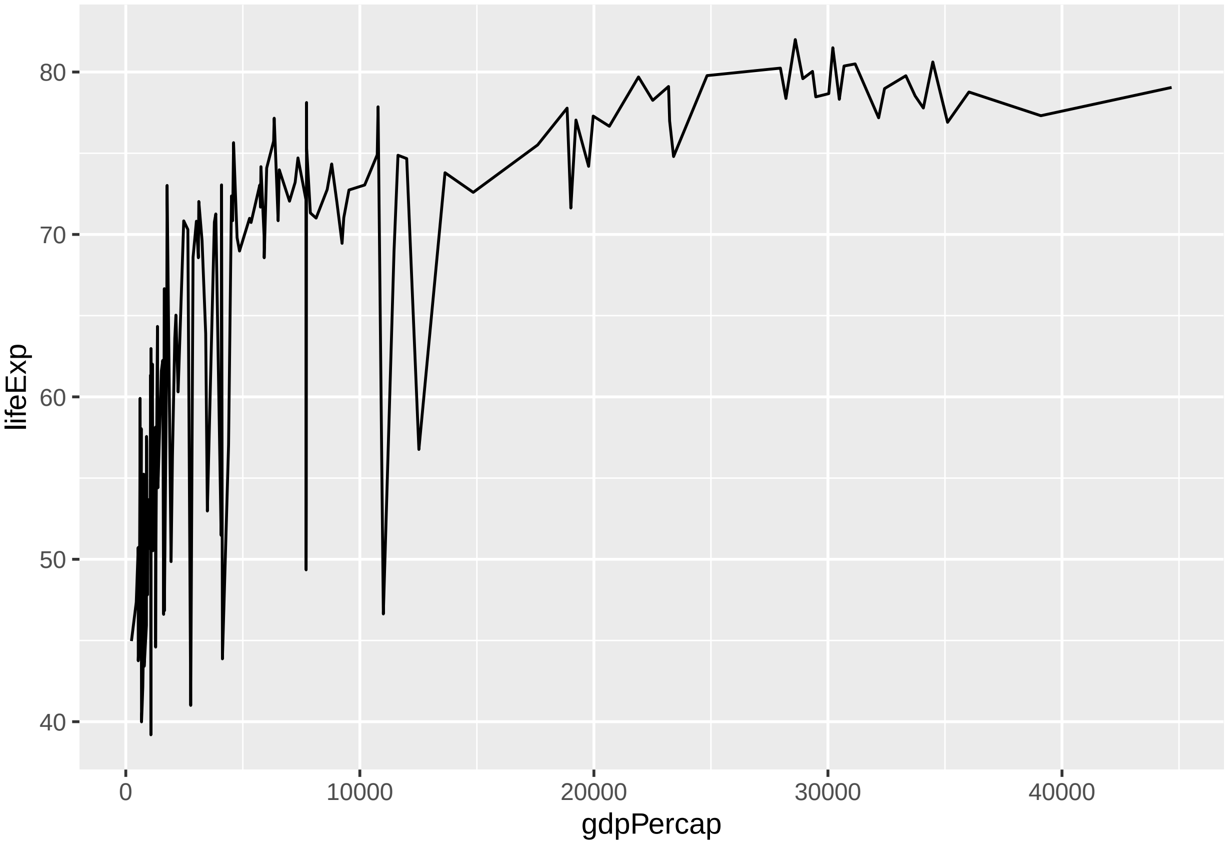
```
## # A tibble: 30 x 7
##    country        continent  year lifeExp      pop gdpPercap        gdp
##    <fct>          <fct>     <int>   <dbl>    <int>     <dbl>      <dbl>
##  1 Germany        Europe     2002    78.7 82350671    30036. 247346845.
##  2 United Kingdom Europe     2002    78.5 59912431    29479. 176615850.
##  3 France         Europe     2002    79.6 59925035    28926. 173339350.
##  4 Italy          Europe     2002    80.2 57926999    27968. 162010799.
##  5 Spain          Europe     2002    79.8 40152517    24835.  99720670.
##  6 Netherlands    Europe     2002    78.5 16122830    33725.  54373854.
##  7 Poland         Europe     2002    74.7 38625976    12002.  46359820.
##  8 Turkey         Europe     2002    70.8 67308928     6508.  43805227.
##  9 Belgium        Europe     2002    78.3 10311970    30486.  31436952.
## 10 Austria        Europe     2002    79.0  8148312    32418.  26414878.
## # i 20 more rows
```

Save the data in 2002. Call it data_2002.

```
save(data_2002,file ="data_2002.RData")
```

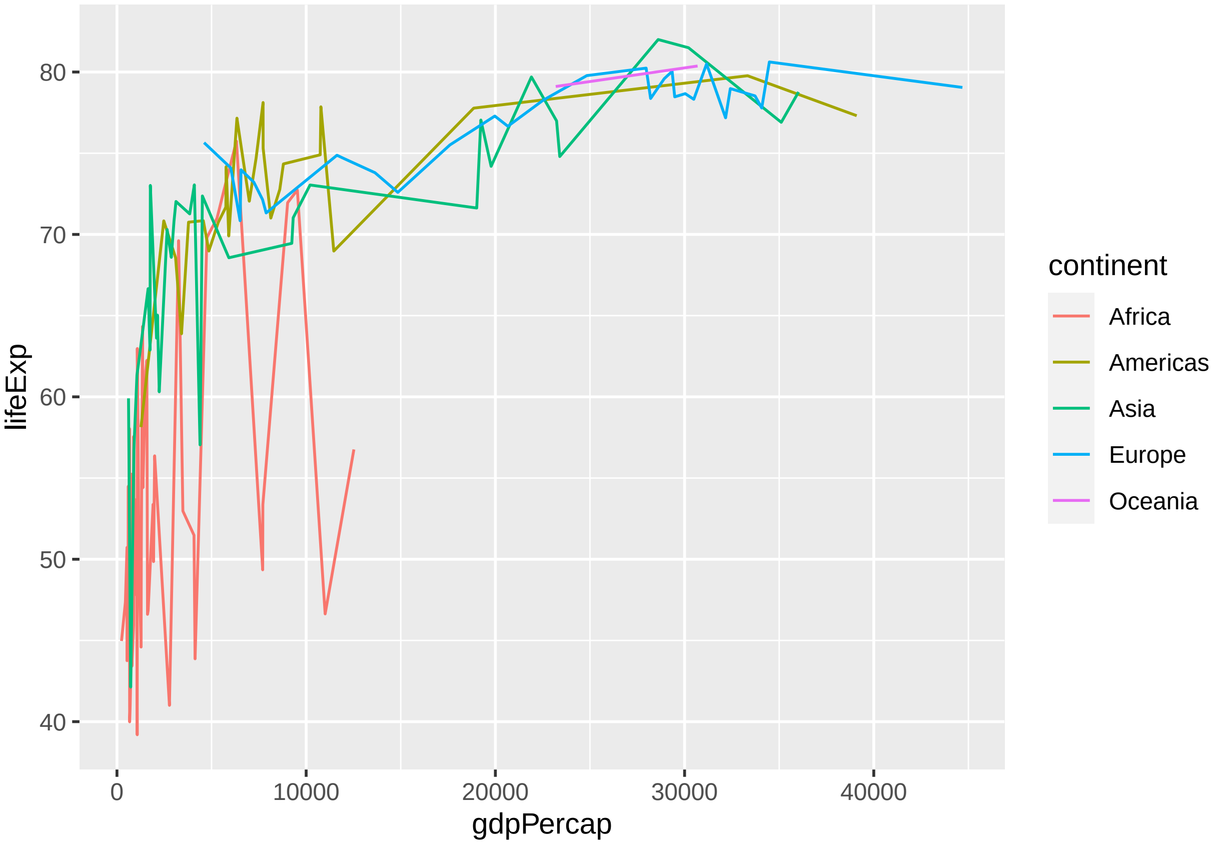
Use data_2002. Use ggplot. Plot gdpPercap vs lifeExp.

```
ggplot(data_2002,aes(x= gdpPercap, y=lifeExp )) +
  geom_line()
```



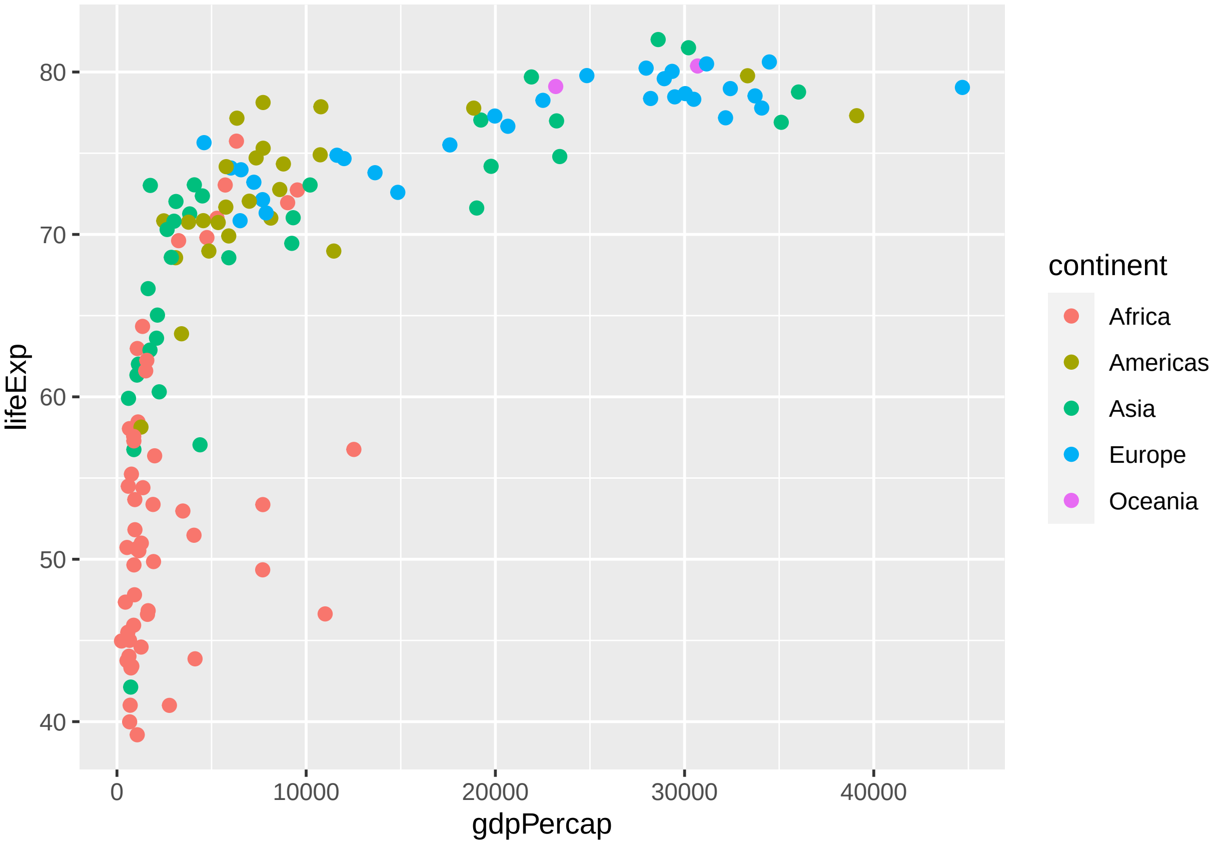
Use data_2002. Use ggplot. Plot gdpPercap vs lifeExp by continent (color)

```
ggplot(data_2002, aes(x= gdpPercap, y=lifeExp, color = continent )) +
 geom_line()
```



Use data_2002. Use ggplot. Plot gdpPercap vs lifeExp by continent and pop (color and size)

```
ggplot(data_2002, aes(x= gdpPercap, y=lifeExp, color = continent, size = pop)) +
 geom_point(size = 2)
```



Get data for Europe in 2002. Call it data_Europe

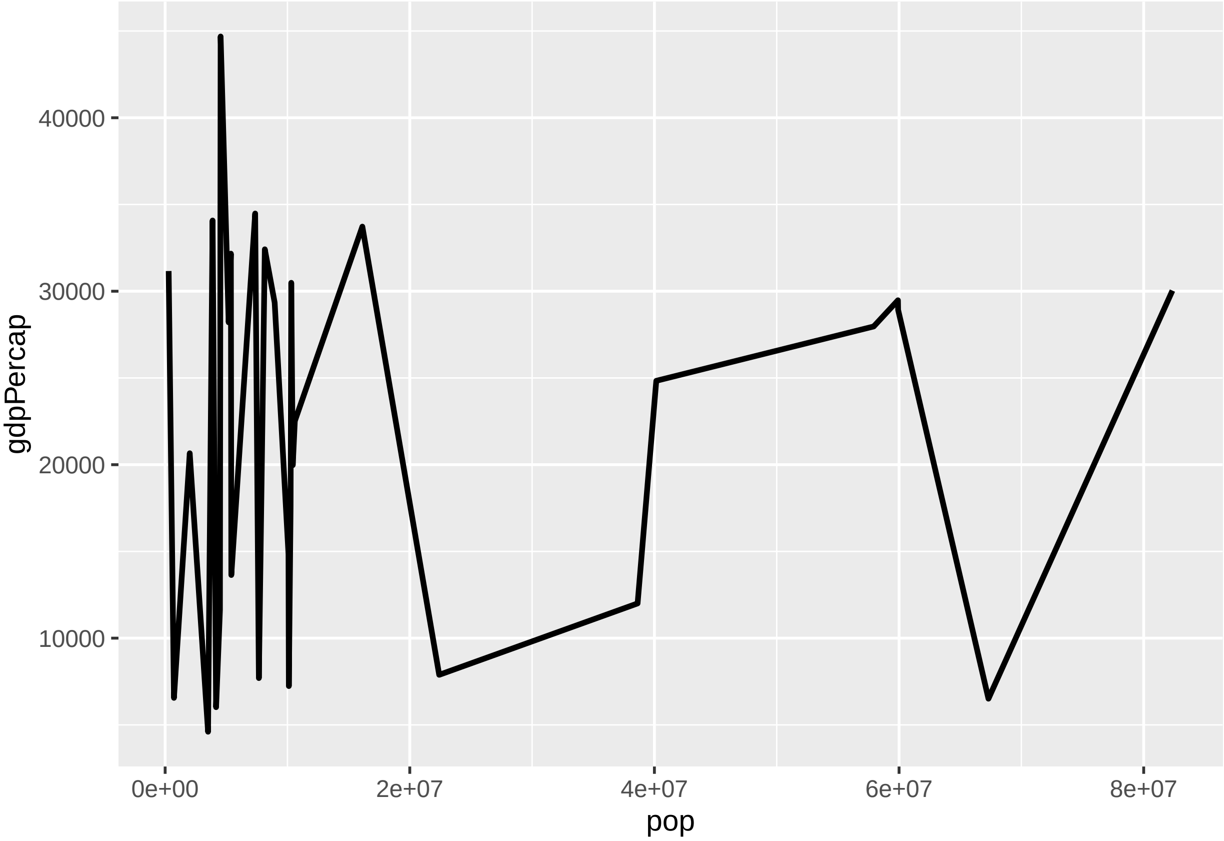
```
data_Europe <- gapminder %>%
  filter(year == "2002", continent == "Europe")
data_Europe
```

```
## # A tibble: 30 x 6
##    country                continent  year lifeExp      pop gdpPercap
##    <fct>                  <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Albania                Europe     2002    75.7  3508512     4604.
##  2 Austria                Europe     2002    79.0  8148312    32418.
##  3 Belgium                Europe     2002    78.3 10311970    30486.
##  4 Bosnia and Herzegovina Europe     2002    74.1  4165416     6019.
##  5 Bulgaria               Europe     2002    72.1  7661799     7697.
##  6 Croatia                Europe     2002    74.9  4481020    11628.
##  7 Czech Republic         Europe     2002    75.5 10256295    17596.
##  8 Denmark                Europe     2002    77.2  5374693    32167.
##  9 Finland                Europe     2002    78.4  5193039    28205.
## 10 France                 Europe     2002    79.6 59925035    28926.
## # i 20 more rows
```

Use data_Europe. Use ggplot. Plot pop vs gdpPercap.

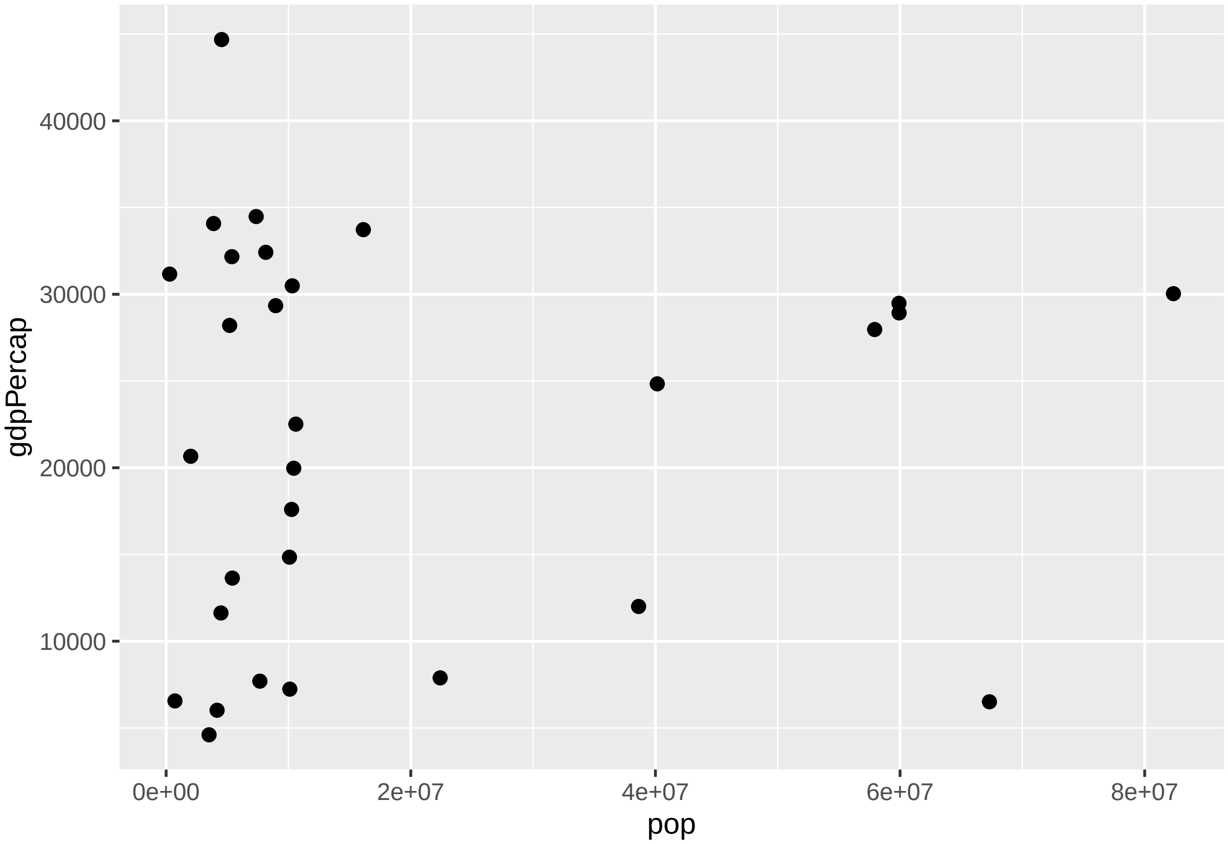
```
ggplot(data_Europe, aes(x= pop, y=gdpPercap)) +
 geom_line(size = 1)
```

```
## 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.
```



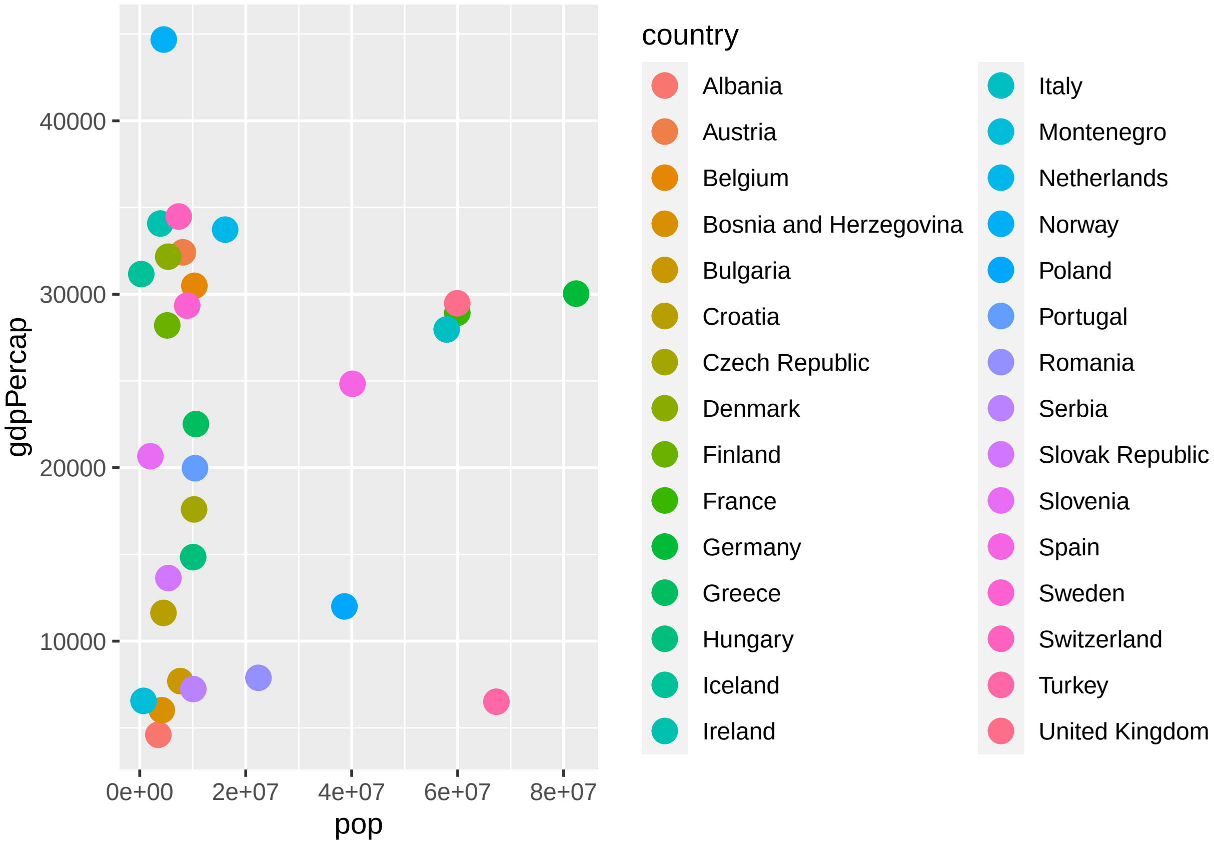
Use data_Europe. Use ggplot. Plot pop vs gdpPercap. Scale population by log10

```
ggplot(data_Europe, aes(x= pop, y=gdpPercap )) +
 geom_point(size=2)
```



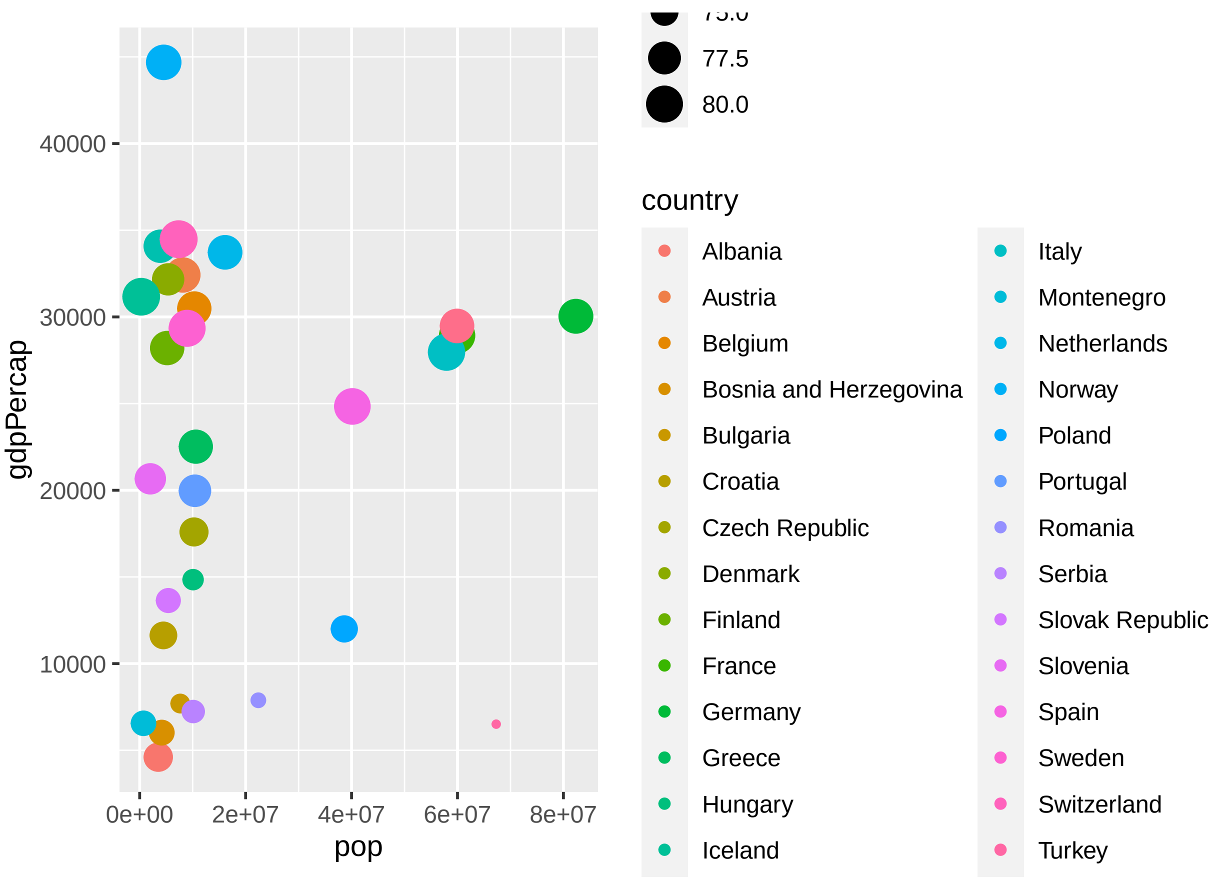
Use data_Europe. Use ggplot. Plot pop vs gdpPercap. Scale population by log10. Color the data by country.

```
ggplot(data_Europe, aes(x= pop, y=gdpPercap, color = country )) +
 geom_point(size = 4)
```



Use data_Europe. Use ggplot. Plot pop vs gdpPercap. Scale population by log10. Color the data by country and size it by lifeExp.

```
ggplot(data_Europe, aes(x= pop, y=gdpPercap, color = country, size = lifeExp)) +
 geom_point()
```



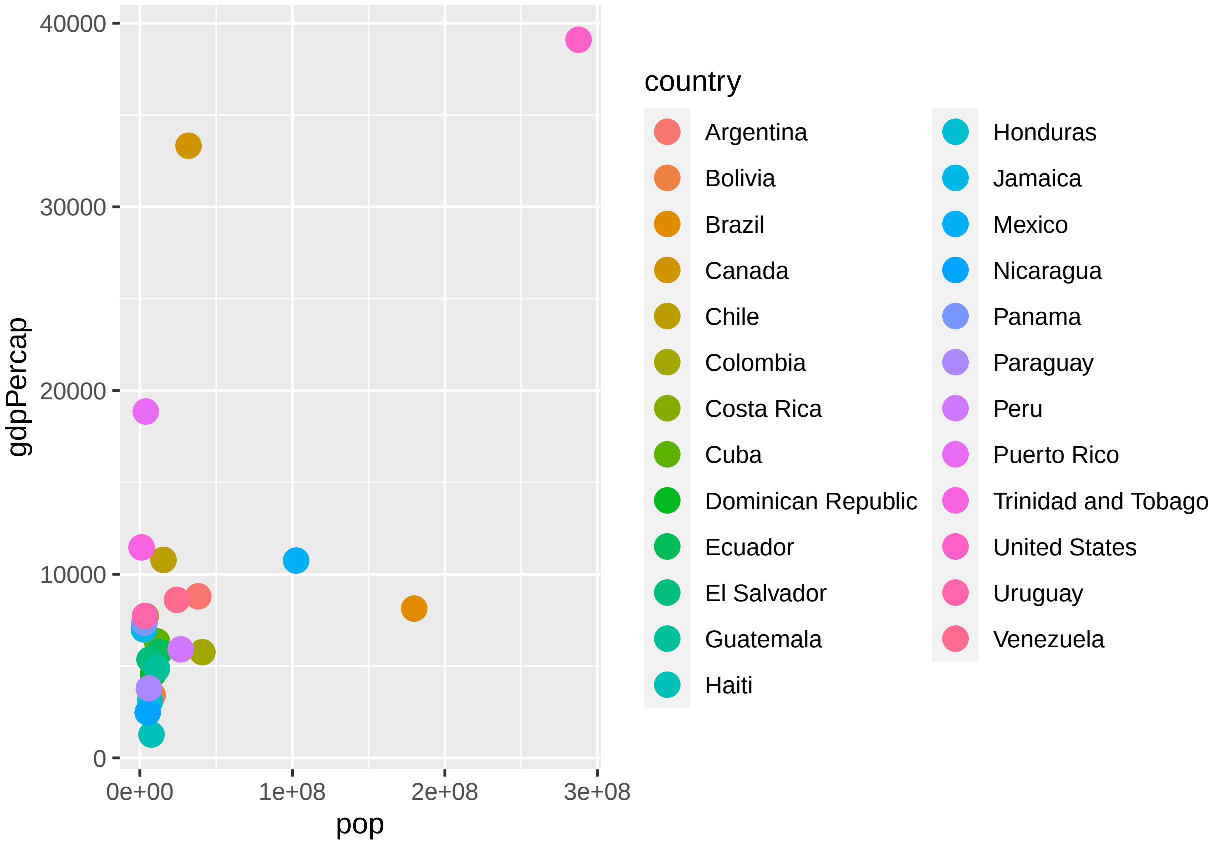
Save the data for Americas. Call it data_Americas.

```
data_Americas <- gapminder %>%
  filter(year == "2002", continent == "Americas")
data_Americas
```

```
## # A tibble: 25 x 6
##    country            continent  year lifeExp       pop gdpPercap
##    <fct>              <fct>     <int>   <dbl>     <int>     <dbl>
##  1 Argentina          Americas   2002    74.3  38331121     8798.
##  2 Bolivia            Americas   2002    63.9   8445134     3413.
##  3 Brazil             Americas   2002    71.0 179914212     8131.
##  4 Canada             Americas   2002    79.8  31902268    33329.
##  5 Chile              Americas   2002    77.9  15497046    10779.
##  6 Colombia           Americas   2002    71.7  41008227     5755.
##  7 Costa Rica         Americas   2002    78.1   3834934     7723.
##  8 Cuba               Americas   2002    77.2  11226999     6341.
##  9 Dominican Republic Americas   2002    70.8   8650322     4564.
## 10 Ecuador            Americas   2002    74.2  12921234     5773.
## # i 15 more rows
```

Use data_Americas. Plot year vs gdpPercap. Scale gdpPercap by log10. Color the data by country.

```
ggplot(data_Americas, aes(x= pop, y=gdpPercap, color = country )) +
 geom_point(size = 4)
```



```
library(fpp3)
```

```
## -- Attaching packages --------------------------------------------- fpp3 0.5 --

## v tsibble     1.1.3     v fable       0.3.3
## v tsibbledata 0.4.1     v fabletools  0.3.3
## v feasts      0.3.1

## -- Conflicts ------------------------------------------------- fpp3_conflicts --
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x dplyr::lag()         masks stats::lag()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
```

```
library(tidyverse)
```

#Here is a tsibble which is identical to the tourism tsibble #from the tsibble package.

```
library(readxl)
my_tourism <- tourism
#View(my_tourism) ## To download as pdf, I have hide this command
head(my_tourism)
```

```
## # A tsibble: 6 x 5 [1Q]
## # Key:       Region, State, Purpose [1]
##   Quarter Region   State           Purpose  Trips
##     <qtr> <chr>    <chr>           <chr>    <dbl>
## 1 1998 Q1 Adelaide South Australia Business  135.
## 2 1998 Q2 Adelaide South Australia Business  110.
## 3 1998 Q3 Adelaide South Australia Business  166.
## 4 1998 Q4 Adelaide South Australia Business  127.
## 5 1999 Q1 Adelaide South Australia Business  137.
## 6 1999 Q2 Adelaide South Australia Business  200.
```

```r
library(readxl)
library(tsibble)

my_tourism <- tourism %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(
    index = Quarter,
    key = c(Region, State, Purpose)
  )

my_tourism
```

```
## # A tsibble: 24,320 x 5 [1Q]
## # Key:       Region, State, Purpose [304]
##    Quarter Region   State           Purpose  Trips
##      <qtr> <chr>    <chr>           <chr>    <dbl>
##  1 1998 Q1 Adelaide South Australia Business  135.
##  2 1998 Q2 Adelaide South Australia Business  110.
##  3 1998 Q3 Adelaide South Australia Business  166.
##  4 1998 Q4 Adelaide South Australia Business  127.
##  5 1999 Q1 Adelaide South Australia Business  137.
##  6 1999 Q2 Adelaide South Australia Business  200.
##  7 1999 Q3 Adelaide South Australia Business  169.
##  8 1999 Q4 Adelaide South Australia Business  134.
##  9 2000 Q1 Adelaide South Australia Business  154.
## 10 2000 Q2 Adelaide South Australia Business  169.
## # i 24,310 more rows
```

> Find what combination of Region and Purpose had the maximum number of overnight trips on average.

*Region: Sydney, Purpose: Visiting, Mean:747.27 **

```r
my_tourism %>%
  as_tibble() %>%
  group_by(Region, Purpose) %>%
  summarise(avg_trips = mean(Trips)) %>%
  ungroup() %>%
  filter(avg_trips == max(avg_trips))
```

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

```
## # A tibble: 1 x 3
##   Region Purpose  avg_trips
##   <chr>  <chr>        <dbl>
## 1 Sydney Visiting      747.
```

> Create a new tsibble which combines the Purposes and State,and just has total trips by State.

```r
state_tourism <- my_tourism %>%
  group_by(State) %>%
  summarise(Trips = sum(Trips)) %>%
  ungroup()

state_tourism
```

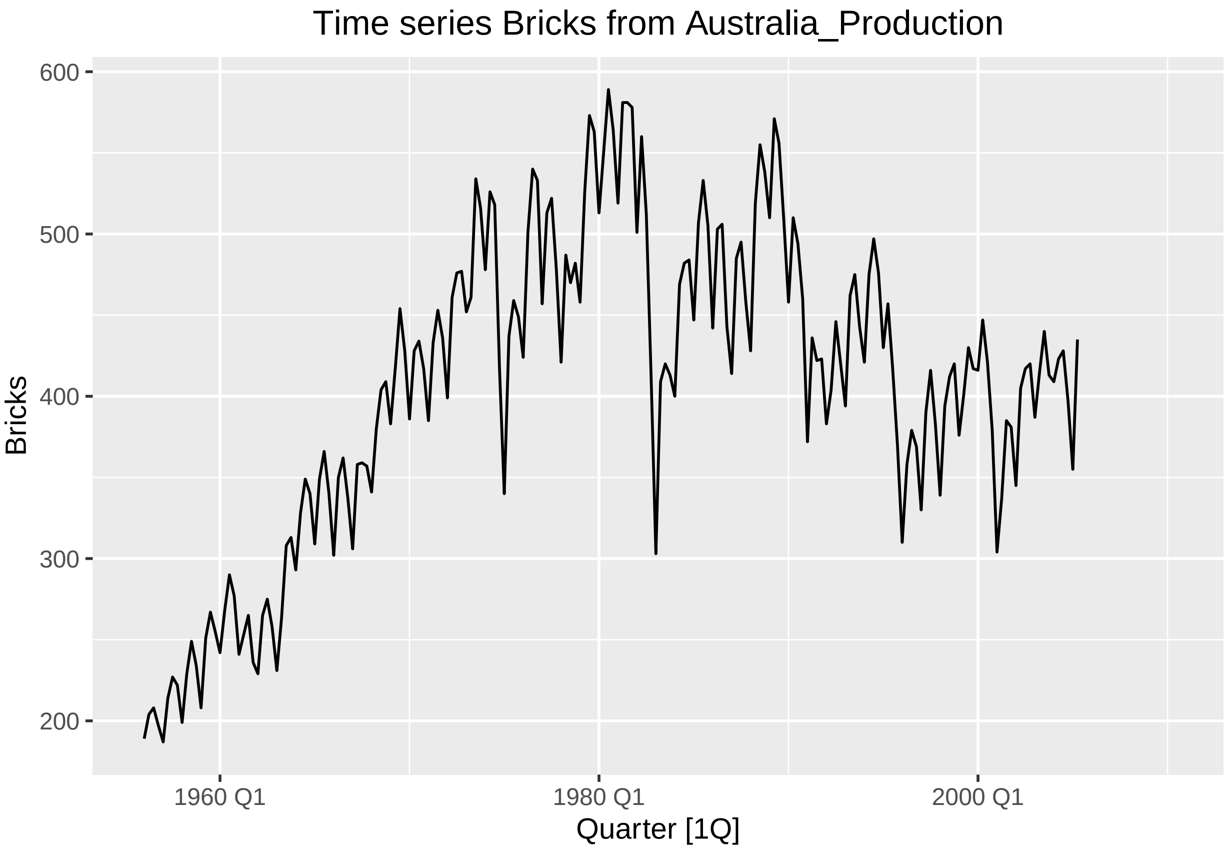
```
## # A tsibble: 640 x 3 [1Q]
## # Key:       State [8]
##    State Quarter Trips
##    <chr>   <qtr> <dbl>
##  1 ACT   1998 Q1  551.
##  2 ACT   1998 Q2  416.
##  3 ACT   1998 Q3  436.
##  4 ACT   1998 Q4  450.
##  5 ACT   1999 Q1  379.
##  6 ACT   1999 Q2  558.
##  7 ACT   1999 Q3  449.
##  8 ACT   1999 Q4  595.
##  9 ACT   2000 Q1  600.
## 10 ACT   2000 Q2  557.
## # i 630 more rows
```

Plotting timeseries

Create plot for the timeseries of Bricks from the aus_production dataset

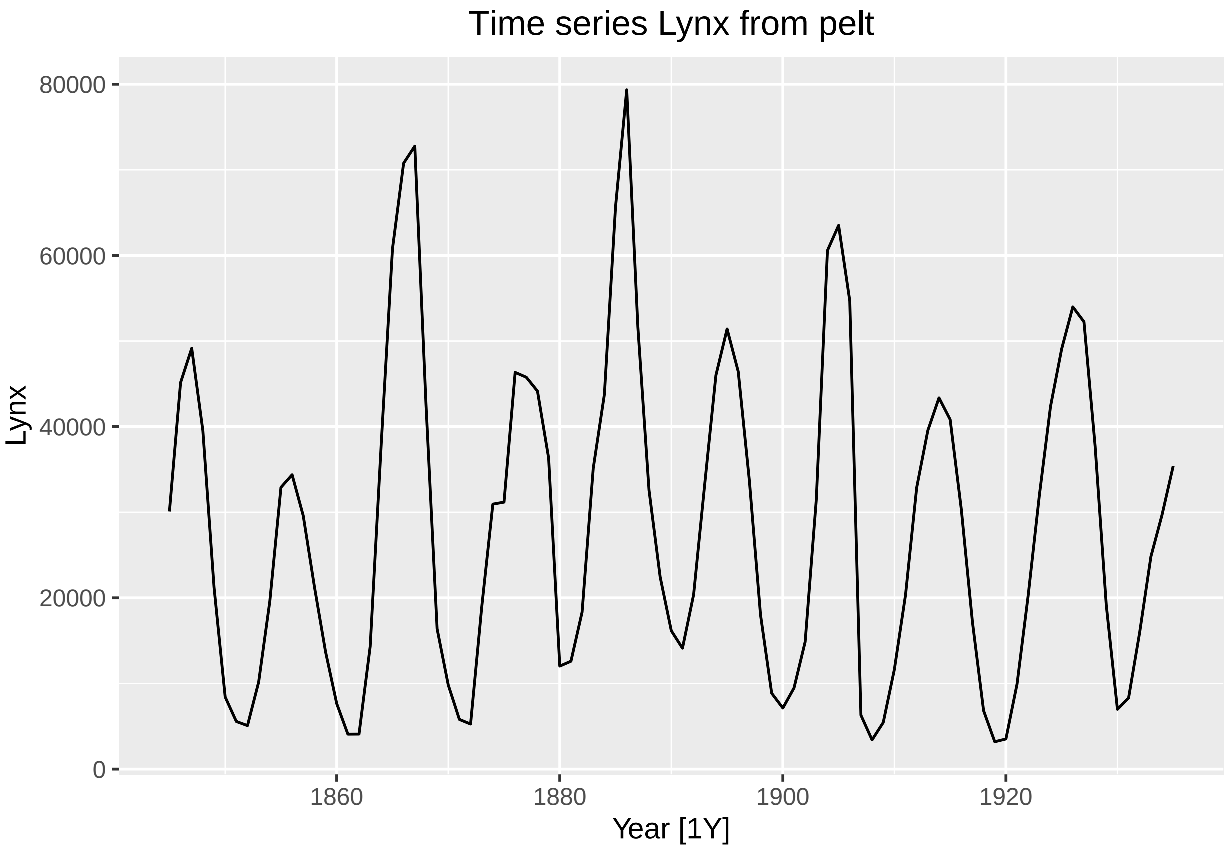
```r
library(ggplot2)

autoplot(aus_production, Bricks) +
  ggtitle("Time series Bricks from Australia_Production") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
## Warning: Removed 20 rows containing missing values (`geom_line()`).
```



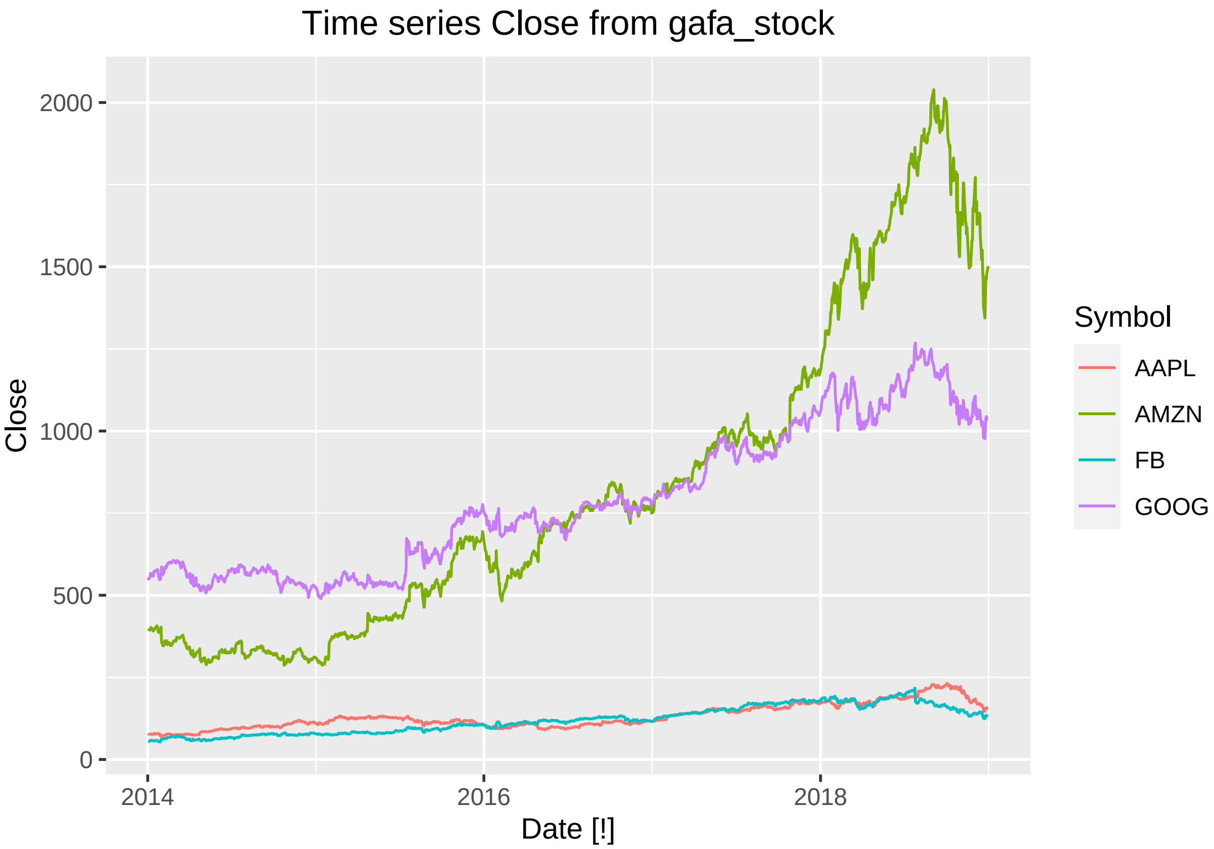
Create plot for the timeseries of Lynx from the pelt dataset

```
library(ggplot2)

autoplot(pelt, Lynx) +
  ggtitle("Time series Lynx from pelt") +
  theme(plot.title = element_text(hjust = 0.5))
```



Create a plot of the timeseries of Close from gafa_stock dataset

```
library(ggplot2)

autoplot(gafa_stock, Close) +
  ggtitle("Time series Close from gafa_stock") +
  theme(plot.title = element_text(hjust = 0.5))
```



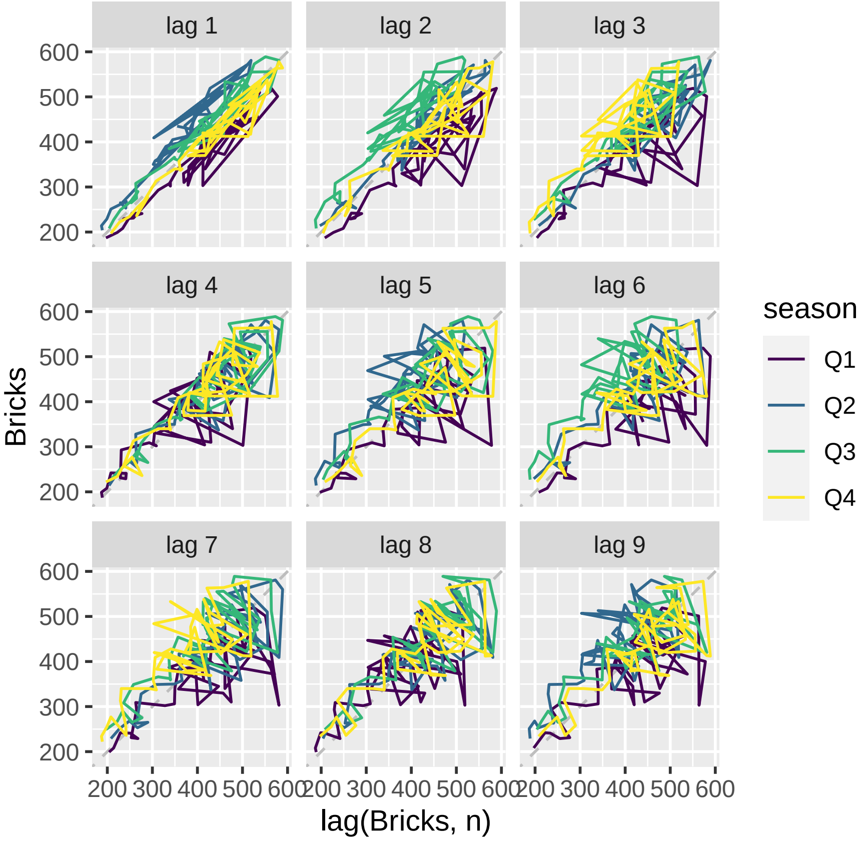
ACF AND GG

> We have introduced the following functions:gg_lag, ACF. Use these functions to explore the following time series: Bricks from aus_production dataset. Where does the seasonality show up as peaks? What is the trend?

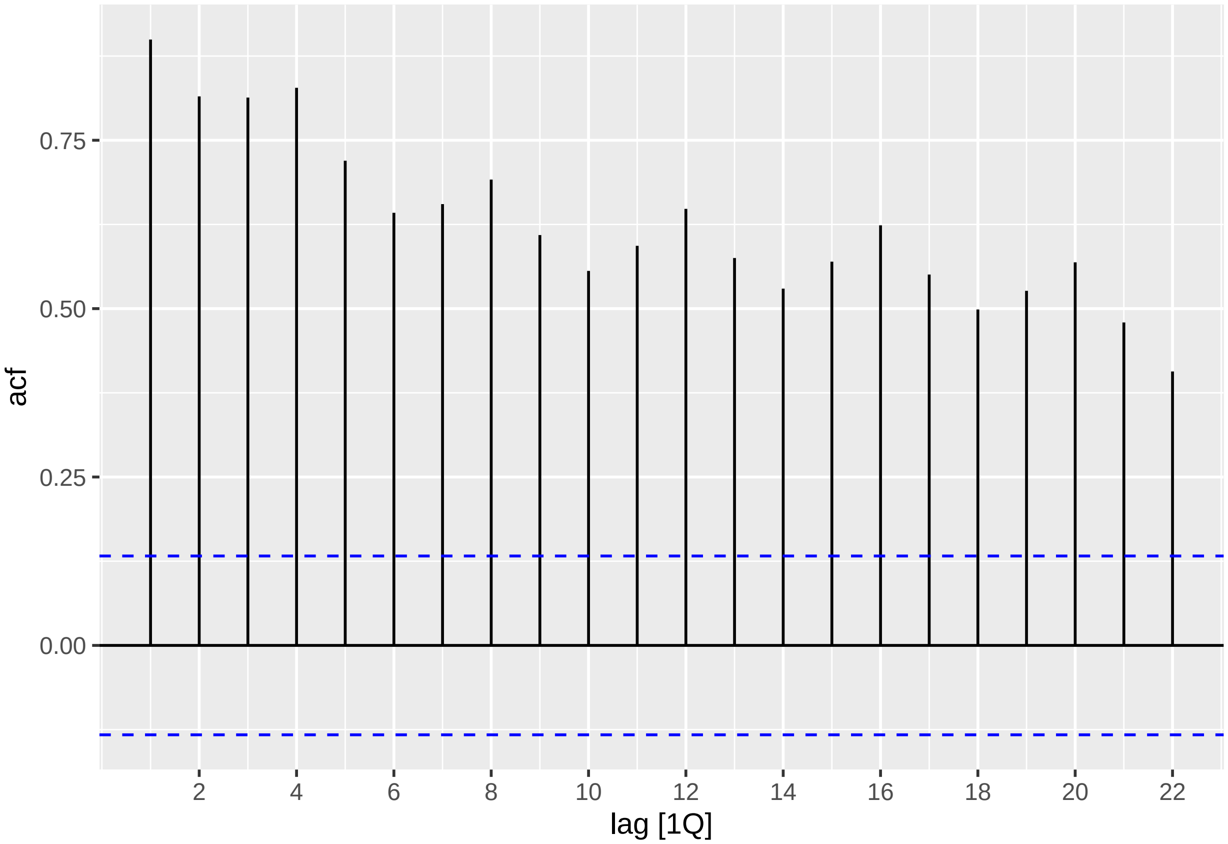
The presence of seasonality is evident at lags 1, 4, 8, 12, 16, and 20, where a positive upward trend is observed.

```
aus_production %>%
  gg_lag(Bricks)
```

```
## Warning: Removed 20 rows containing missing values (gg_lag).
```



```
aus_production %>%
  ACF(Bricks) %>%
  autoplot()
```



You can compute the daily changes in the Google stock price in 2018 using the code below:

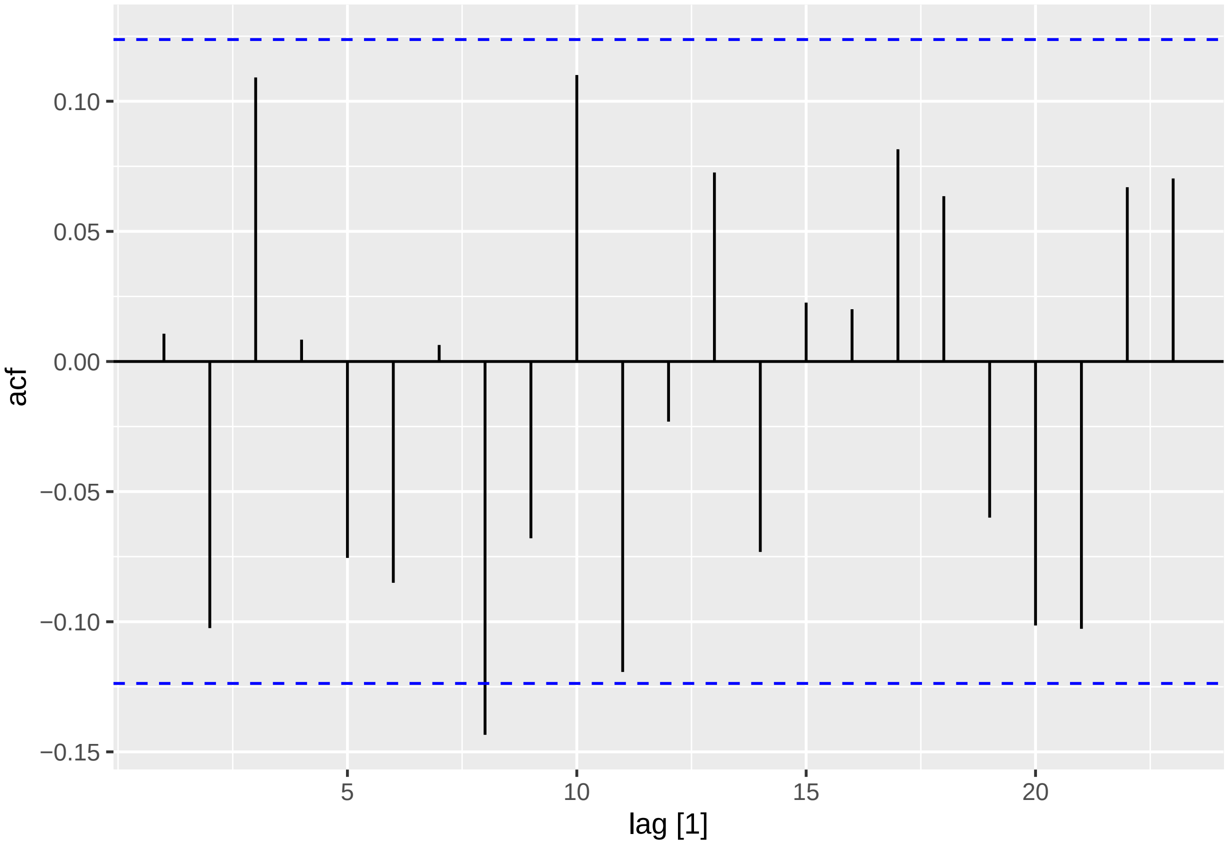
```
dgoog <- gafa_stock %>%
  filter(Symbol == "GOOG", year(Date) >= 2018) %>%
  mutate(trading_day = row_number()) %>%
  update_tsibble(index = trading_day, regular = TRUE) %>%
  mutate(diff = difference(Close))
```

The tsibble needed re-indexing as trading happens irregularly. The new index is based only on trading days.

Use autoplot() and ACF(). Do you see white noise?

From the above lib used, I have notice some white noise in the graph

```
dgoog %>%
  mutate(diff = difference(Close)) %>%
  ACF(diff) %>%
  autoplot()
```


0.10
0.05
0.00
−0.05
−0.10
−0.15
acf
5
10
15
20
lag [1]