Lab 02: Around the World in 80 Slays, SOLUTIONS

PSTAT 100: Spring 2024 (Instructor: Ethan P. Marzban)

MEMBER 1 (NetID 1) MEMBER 2 (NetID 2)
MEMBER 3 (NetID 3)

April 14, 2024

Required Packages

```
library(ottr)        # for checking test cases (i.e. autograding)
library(pander)      # for nicer-looking formatting of dataframe outputs
library(reshape2)    # for 'melting' data frames
library(tidyverse)   # for graphs, data wrangling, etc.
```

Logistical Details

Logistical Details

- This lab is due by **11:59pm on Wednesday, April 17, 2024**.
- Collaboration is allowed, and encouraged!
  - If you work in groups, list ALL of your group members' names and NetIDs (not Perm Numbers) in the appropriate spaces in the YAML header above.
  - Please delete any "MEMBER X" lines in the YAML header that are not needed.
  - No more than 3 people in a group, please.
- Ensure your Lab properly renders to a `.pdf`; non-`.pdf` submissions will not be graded and will receive a score of 0.
- Ensure all test cases pass (test cases that have passed will display a message stating `"All tests passed!"`)

Lab Overview and Objectives

In this lab, we will discuss:

- Plotting in `R` using the `ggplot2` package
- Applying transformations, and the underlying theory on how this ties back to the grammar of graphics
- Formatting plot and code outputs of `.qmd` files.

The Data at a Glance

This week, we will be exploring an amalgamation of a few datasets provided by the World Bank's Data Site. For those unaware, the World Bank Group is a collection of five organizations, with staff members from over 170 countries and offices in over 130 locations, aiming to study the effects of poverty globally and, ideally, eliminate it (you can read more about the World Bank's mission here).

I've taken the liberty of merging (and partially cleaning) several datasets to create one dataset, contained in a file called `country_info.csv`, which contains the following column headers:

- `Country Name`: the (verbose) name of the country
- `Country Code`: an abbreviation for the country (each country has a unique `Country Code`)
- `Indicator Name`: the name of the indicator being measured. Names are one of the following 8 values:
  - `GDP (current US$)`
  - `GNI, Atlas method (current US$)`
  - `Life expectancy at birth, female (years)`
  - `Life expectancy at birth, male (years)`
  - `Literacy rate, adult female (% of females ages 15 and above)`
  - `Literacy rate, adult male (% of males ages 15 and above)`
  - `Literacy rate, adult total (% of people ages 15 and above),`
  - `Population, total`
- Column headers `1960` through `2023`, indicating the year of observation

Also included in the `data/` subfolder is a file called `continents.csv`, which contains a list of countries of the world and the continent in which they lie.

! Question

Import the `country_info.csv` dataframe, and assign it to a variable called `country_info`. Display the first 5 column names, and check that they are: `"Country Name"`, `"Country Code"`, `"Indicator Name"`, `"1960"`, and `"1961"`, respectively. **Pay close attention** - if you have `X`'s in your column names, try playing around with the `check.names` argument of the `read.csv()` function.
Solution:

```
## replace this line with your code

country_info <- read.csv("data/country_info.csv",
                         check.names = F)

(country_info %>% names())[1:5]
```

```
[1] "Country Name"   "Country Code"   "Indicator Name" "1960"
[5] "1961"
```

Answer Check:

```
# DO NOT EDIT THIS LINE
invisible({check("tests/q1.R")})
```

```
All tests passed!
```

You may note that, as previously mentioned, the `country_info` dataframe does not include any information about continents. Let's fix that!

! Question

Read in the `continents.csv` file, and left-join its content onto the `country_info` variable you created in the previous question. Assign this merged dataframe to a variable called `country_info_merged`; ensure that the column encoding continent information has the title `Continent` (pay attention to case - this will be important for the autograder).

Solution:

```
## replace this line with your code

continents <- read.csv("data/continents.csv",
                         check.names = F)

country_info_merged <- country_info %>%
  left_join(
    continents,
    by = join_by(`Country Name` == `Country`)
  )
```

Answer Check:

```
# DO NOT EDIT THIS LINE
invisible({check("tests/q2.R")})
```

```
All tests passed!
```

Focusing on 2020

To start off, we'll focus exclusively on data collected in the year 2020.

! Question 3

From your `country_info_merged` dataframe, select only the `"Country Name"`, `"Country Code"`, `"Indicator Name"`, and `"2020"` column, and assign the subsetted dafatframe to a variable called `data_2020`.

Solution:

```
## replace this line with your code

data_2020 <- country_info_merged %>%
  select(
    "Country Name",
    "Country Code",
    "Indicator Name",
    "2020",
    "Continent"
  )
```

Answer Check:

```
# DO NOT EDIT THIS LINE
invisible({check("tests/q3.R")})
```

```
Test q3 failed:

ncol(data_2020) == 4 is not TRUE

  `actual`:   FALSE
  `expected`: TRUE
```

GDP vs. GNI

Now, it is often stated that GNI (Gross National Income) and GDP (Gross Domestic Product) can be used interchangeably as a metric of how "wealthy" a country is. We won't tackle the question of whether or not they are actually accurate measures of a countries "wealth", but we will investigate the claim that the two variables contain roughly the same information.

Ultimately, we'd like to make a scatterplot of `GDP` (on the y-axis) vs `GNI` (on the x-axis). But, as it stands, the format of our `data_2020` variable isn't really conducive to that. Specifically, the `GNI` and `GDP` values are all mixed up in the dataframe. There are a couple of ways we can overcome this; here's the general procedure we're going to use:

1) Filter out only the GDP and GNI values (using the `Indicator Name` column)

2) Melt our dataframe, to resolve the issue of having the "year" as a column name,

3) **Pivot** the molten data frame to extract separate columns for GDP and GNI.

We'll also drop the `Continent` column, for now. At the end of this succession of operations, we should have a dataframe whose first few rows look like:

Country Name	Country Code	Year	GDP (current US$)	GNI, Atlas method (current US$)
Aruba	ABW	2020	2.558906e+09	2.487356e+09
Afghanistan	AFG	2020	1.995593e+10	1.923686e+10
Angola	AGO	2020	4.850156e+10	5.633140e+10
Albania	ALB	2020	1.516273e+10	1.494621e+10

! Question 4

Copy and paste the following template code into the answer portion below, and fill in the blanks (indicated with ellipses, …) to perform the melting and pivoting outlined above. Assign the reformatted reformatted dataframe to a variable called `data_2020_plotting`, and display the first 4 rows - visually check that these 4 rows match the table above.

```
data_2020 %>%
  select(
    ...         # select everything except the Continent column
  ) %>%
  filter(
    ...         # isolate only GDP and GNI values
  ) %>%
  melt(
    ...         # resolve the issue of 'year' value in col. name
  ) %>%
  pivot_wider(
    names_from = ...,  # where should the new column names come from?
    values_from = ...  # where should the new values come from?
  )
```

Solution:

```r
## copy-paste the above skeleton code, and fill in the blanks.

data_2020_plotting <- data_2020 %>%
  select(!Continent) %>%
  filter((`Indicator Name` == "GDP (current US$)") |
           (`Indicator Name` == "GNI, Atlas method (current US$)")) %>%
  melt(
    id.vars = c("Country Name", "Country Code", "Indicator Name"),
    variable.name = "Year",
    value.name = "Value"
  ) %>%
  pivot_wider(
    names_from = `Indicator Name`,
    values_from = `Value`
  )

data_2020_plotting %>% head(4) %>% pander()
```

Table 2: Table continues below

Country Name	Country Code	Year	GDP (current US$)
Aruba	ABW	2020	2.559e+09
Afghanistan	AFG	2020	1.996e+10
Angola	AGO	2020	4.85e+10
Albania	ALB	2020	1.516e+10

GNI, Atlas method (current US$)
2.487e+09
1.924e+10
5.633e+10
1.495e+10

Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Alright, now we can generate the plot we set out to create!

! Question 5

Use your `data_2020_plotting` variable from the previous quetion to generate a scatterplot of `GDP` (on the y-axis) vs `GNI` (on the x-axis) (again, unless otherwise specified, consider only data collected from 2020). Make sure your plots have appropriately labeled axes, and include a title. An easy way to include a title using `ggplot` is to add a call to `ggtitle()`: e.g.

```
ggplot(...) +
  ... +
  ggtitle("Title of Plot")
```

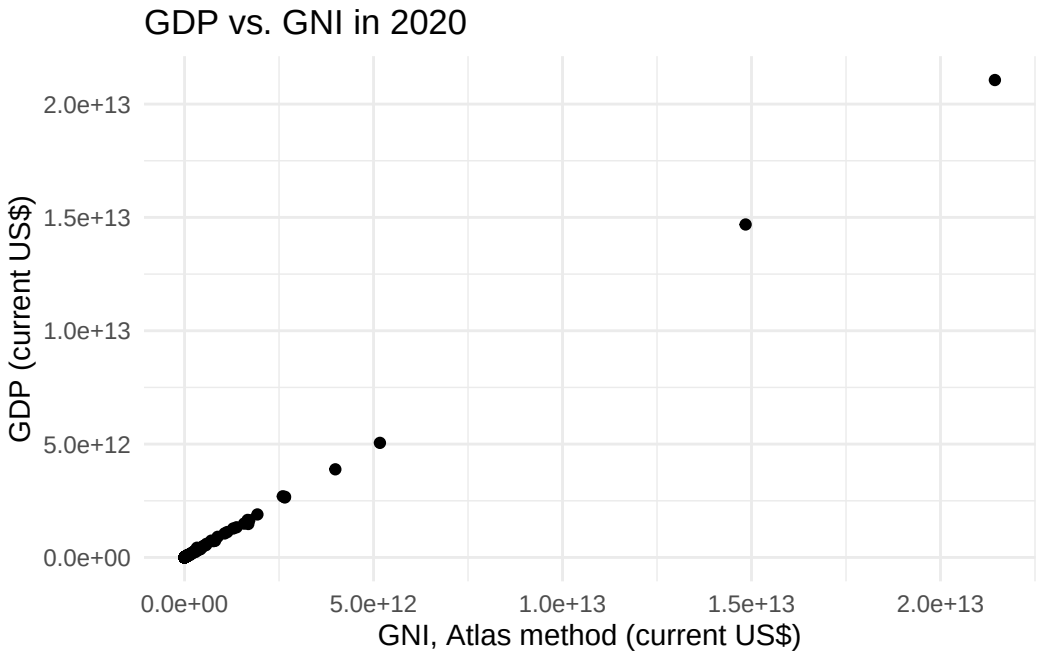
Interpret your plot - specifically, do you think GNI and GDP can be used interchangeably? What specifically about the plot makes you believe that?

Solution:

```
## replace this line with your code

data_2020_plotting %>%
  ggplot(aes(x = `GNI, Atlas method (current US$)`,
             y = `GDP (current US$)`)) +
  geom_point() +
  theme_minimal() +
  ggtitle("GDP vs. GNI in 2020")
```

```
Warning: Removed 18 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Answer Check:

There is no autograder for this question; your TA will manually check that your answers are correct.

Since the plot is pretty much a straight line, we can see that GDP and GNI contain roughly the same information, and hence can be used more or less interchangeably.

The large cluster of points near the origin makes things a little difficult to see. We suspect, therefore, that viewing our GDP and GNI on a **log-log** scale might be useful - in other words, we would like to apply a logarithm transformation to both the x- and y-axes.

! Question 6

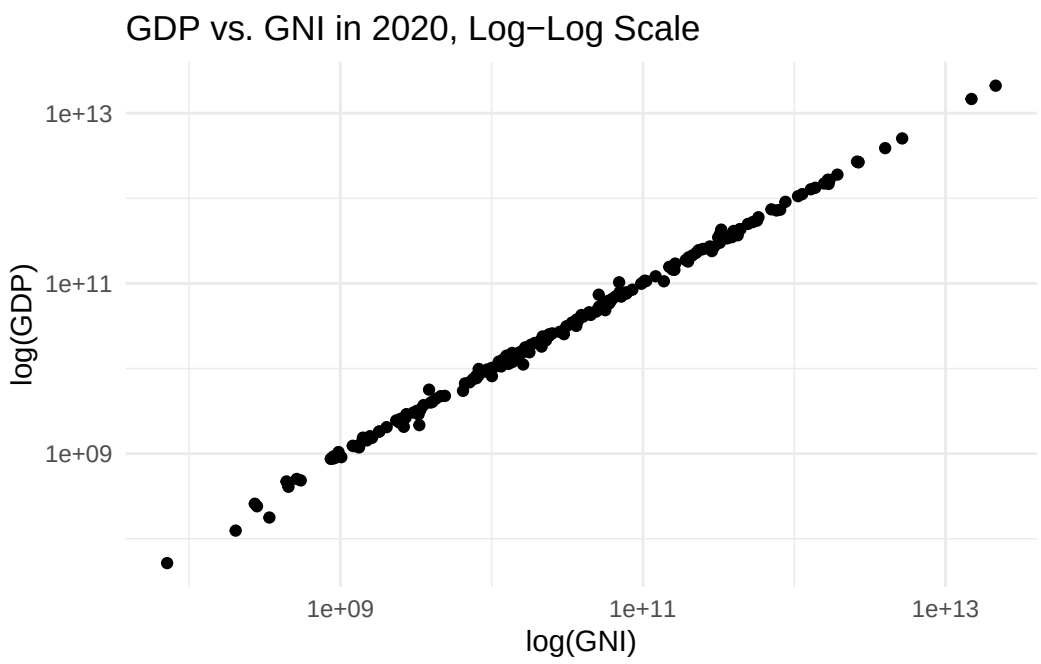
Replicate your plot from the previous question, and now use the `scale_x_log10()` and `scale_y_log10()` functions to log-transform both the x- and y-axes. **Update your axis labels and plot title appropriately!** Do your conclusions about whether or not GDP and GNI can be used interchangeably change from the previous question?

Solution:

```
## replace this line with your code

data_2020_plotting %>%
  ggplot(aes(x = `GNI, Atlas method (current US$)`,
             y = `GDP (current US$)`)) +
  geom_point() +
  theme_minimal() +
  ggtitle("GDP vs. GNI in 2020, Log-Log Scale") +
  xlab("log(GNI)") +
  ylab("log(GDP)") +
  scale_x_log10() +
  scale_y_log10()
```

```
Warning: Removed 18 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Conclusions do not change between the previous question and this one - this is to be expected, since the logarithm is a monotonically increasing transformation.

Adult Literacy Rates

Let's take a look at how adult literacy rates are related to GDP.

! Question 7

Return to the `data_2020` variable.

1) Extract out only GDP and Adult Literacy Rate values
2) Melt, using "Country Name", "Country Code", "Indicator Name", and "Continent" as colvars
3) Pivot to create separate columns for GDP and Adult Literacy Rates
4) Generate a scatterplot of Adult Literacy Rates (on the y-axis) vs GDP (on the x-axis).

Here is some skeleton code you can copy-paste and modify, which will accomplish the above four steps in a single command (or, rather, series of commands chained together into one using several pipe operations):

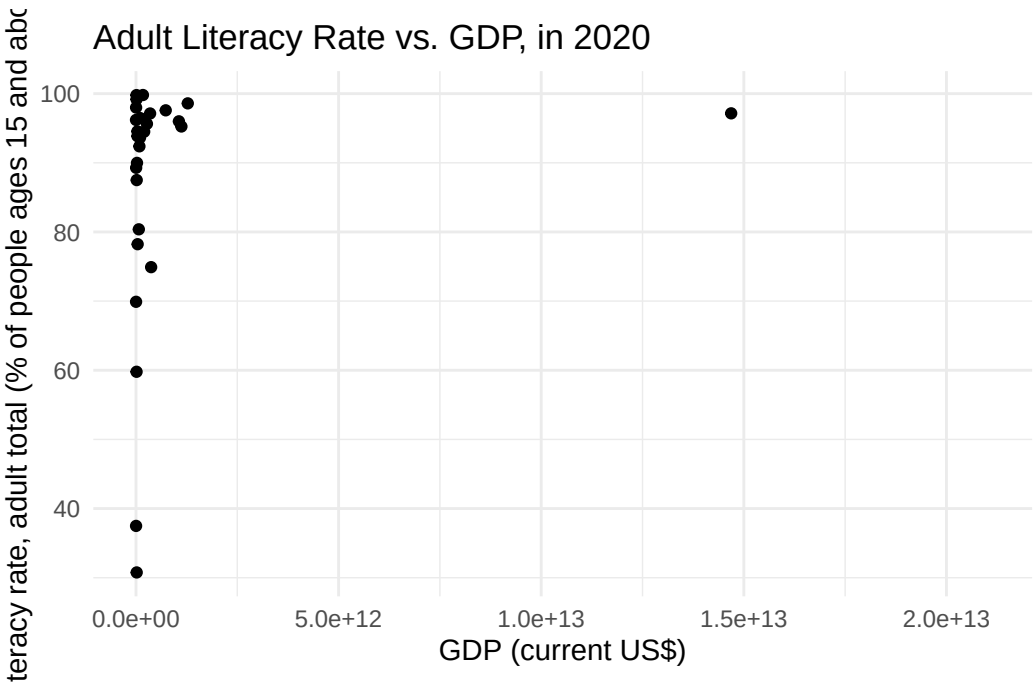
```
data_2020 %>%
  filter(
    ...              # step 1
  ) %>%
  melt(
    ...              # step 2
  ) %>%
  pivot_wider(
    ...              # step 3
  )  %>%
  ggplot(aes(
    ...              # step 4
  )) +
  geom_point() +
  theme_minimal() +
  ...                # additional plotting layers, as needed
```

Solution:

```
## replace this line with your code

data_2020 %>%
  filter((`Indicator Name` == "GDP (current US$)") |
           (`Indicator Name` == "Literacy rate, adult total (% of people ages 15 and above)
  melt(
    id.vars = c("Country Name", "Country Code", "Indicator Name", "Continent"),
    variable.name = "Year",
    value.name = "Value"
  ) %>%
  pivot_wider(
    names_from = `Indicator Name`,
    values_from = `Value`
  )  %>%
  ggplot(aes(
    x = `GDP (current US$)`,
    y = `Literacy rate, adult total (% of people ages 15 and above)`
  )) +
  geom_point() +
  theme_minimal() +
  ggtitle("Adult Literacy Rate vs. GDP, in 2020")
```

```
Warning: Removed 189 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Not bad - we can, however, improve things a bit.

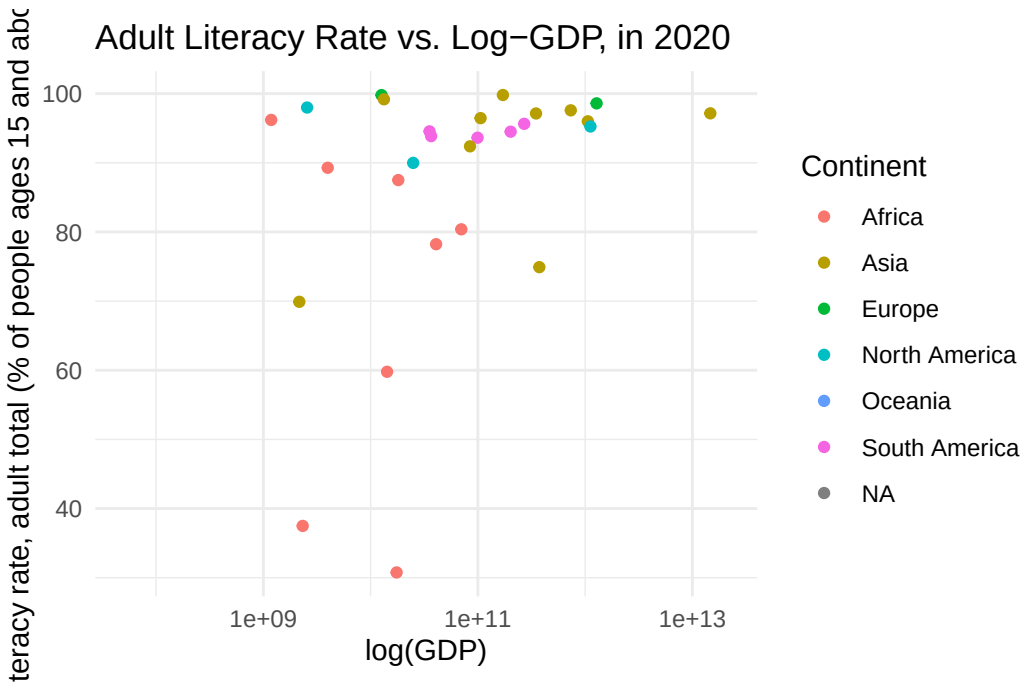
! Question 8

Replicate your plot from the previous question. Now, scale the x-axis using a logarithm transformation (**again, be sure to modify axis labels and plot titles as necessary!**); also color points by continent.
Solution:

```r
## replace this line with your code

data_2020 %>%
  filter((`Indicator Name` == "GDP (current US$)") |
           (`Indicator Name` == "Literacy rate, adult total (% of people ages 15 and above)
  melt(
    id.vars = c("Country Name", "Country Code", "Indicator Name", "Continent"),
    variable.name = "Year",
    value.name = "Value"
  ) %>%
  pivot_wider(
    names_from = `Indicator Name`,
    values_from = `Value`
  )  %>%
  ggplot(aes(
    x = `GDP (current US$)`,
    y = `Literacy rate, adult total (% of people ages 15 and above)`,
    colour = Continent
  )) +
  geom_point() +
  theme_minimal() +
  scale_x_log10() +
  ggtitle("Adult Literacy Rate vs. Log-GDP, in 2020") +
  xlab("log(GDP)")
```

```
Warning: Removed 189 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Though the color is adding *some* information, it's a little hard to read. This is where the notion of **facetting** becomes very useful. Recall, from lecture, that is is typically not a good idea to use color to encode a categorical variable with more than around 5 categories. An alternative to modifying aesthetics to encode additional information is *facetting*, where we effectively "unlayer" our plot. For example, facetting our plot from the previous question would distenangle the continents from one another and reveal a panel of 7 (8, if we include the one country with `NA` continent - don't worry too much about what's going on with that) graphs, one for each continent.

! Question 9

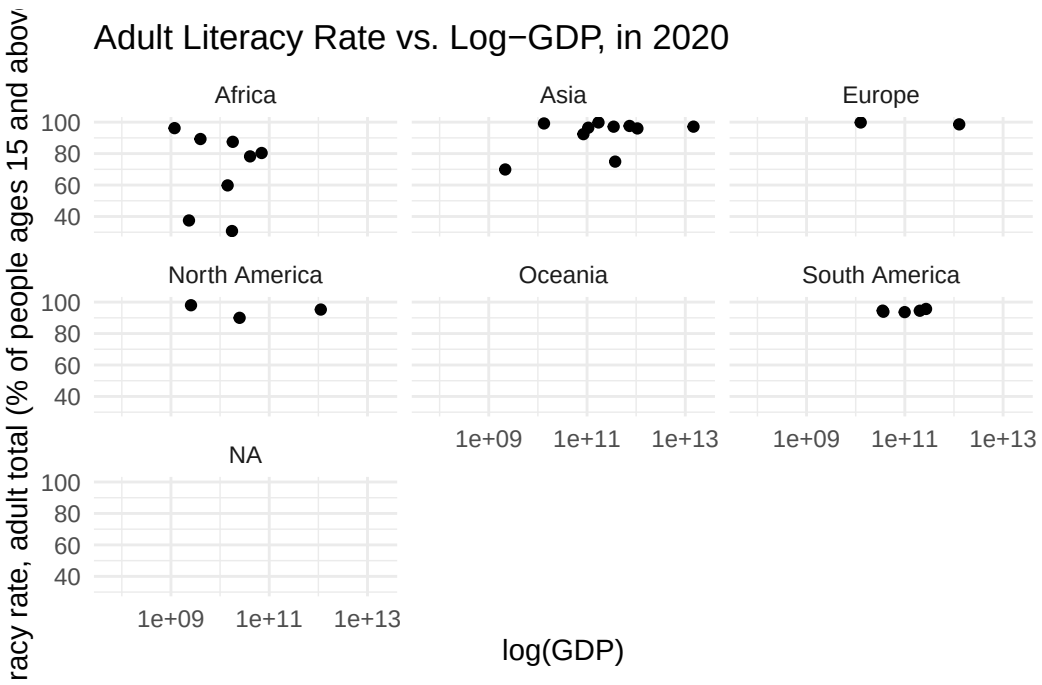
Replicate your plot from the previous question, but instead of coloring by continent use `facet_wrap()` to facet your plot based on continent.

Solution:

```r
## replace this line with your code

data_2020 %>%
  filter((`Indicator Name` == "GDP (current US$)") |
           (`Indicator Name` == "Literacy rate, adult total (% of people ages 15 and above)
  melt(
    id.vars = c("Country Name", "Country Code", "Indicator Name", "Continent"),
    variable.name = "Year",
    value.name = "Value"
  ) %>%
  pivot_wider(
    names_from = `Indicator Name`,
    values_from = `Value`
  )  %>%
  ggplot(aes(
    x = `GDP (current US$)`,
    y = `Literacy rate, adult total (% of people ages 15 and above)`
  )) +
  geom_point() +
  facet_wrap(~Continent) +
  theme_minimal() +
  scale_x_log10() +
  ggtitle("Adult Literacy Rate vs. Log-GDP, in 2020") +
  xlab("log(GDP)")
```

```
Warning: Removed 189 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

A Note

You might notice that the graphs within each panel (of the facetted graph) seem a little "sparse". This is due to the relatively high volume of missing literacy rates included in the dataset, which is likely due to the COVID-19 pandemic.

Comparisons Across Time

Everything we did above was restricted to the year 2020. It might, however, be interesting to consider how (if at all) various metrics have varied over time.

GDP

Question 10

Display a line graph of GDP (on the y-axis) vs. Year (on the x-axis), including only years from 2000 onwards. Color by continent. Keep in mind that you may have to, again, do some filtering, melting, and piping first!

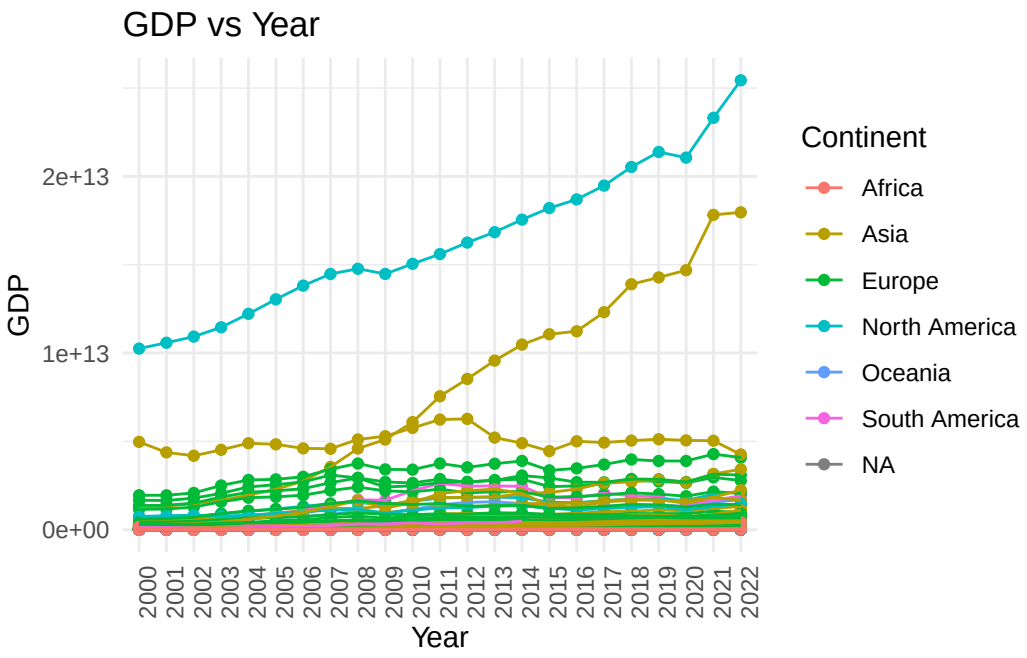
Solution:

```r
## replace this line with your code

country_info_merged %>%
  filter(
    `Indicator Name` == "GDP (current US$)"
  ) %>%
  select(
    "Country Name",
    "Country Code",
    "Indicator Name",
    "2000":"2022",
    "Continent"
  ) %>%
  select(
    !`Indicator Name`
  ) %>%
  melt(
    id.vars = c("Country Name", "Country Code", "Continent"),
    variable.name = "Year",
    value.name = "GDP"
  ) %>%
  ggplot(aes(x = Year,
             y = GDP)) +
  geom_point(aes(colour = Continent)) +
  geom_line(aes(group = `Country Name`,
                colour = Continent)) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 90)
  )  +
  ggtitle("GDP vs Year")
```

```
Warning: Removed 223 rows containing missing values or values outside the scale range
(`geom_point()`).
```

```
Warning: Removed 208 rows containing missing values or values outside the scale range
(`geom_line()`).
```



Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

It's a little hard to see variations across continents. Let's restrict our focus to just North America, for the time being.

! Question 11

Re-do your plot from the previous question, but now only display countries in North America. Color by country (yes, this is a bad idea since there are more than 5 countries in North America! But, for practice purposes, we'll use color anyway.) Apply a log-transform to the y-axis, and adjust labels and titles as necessary.

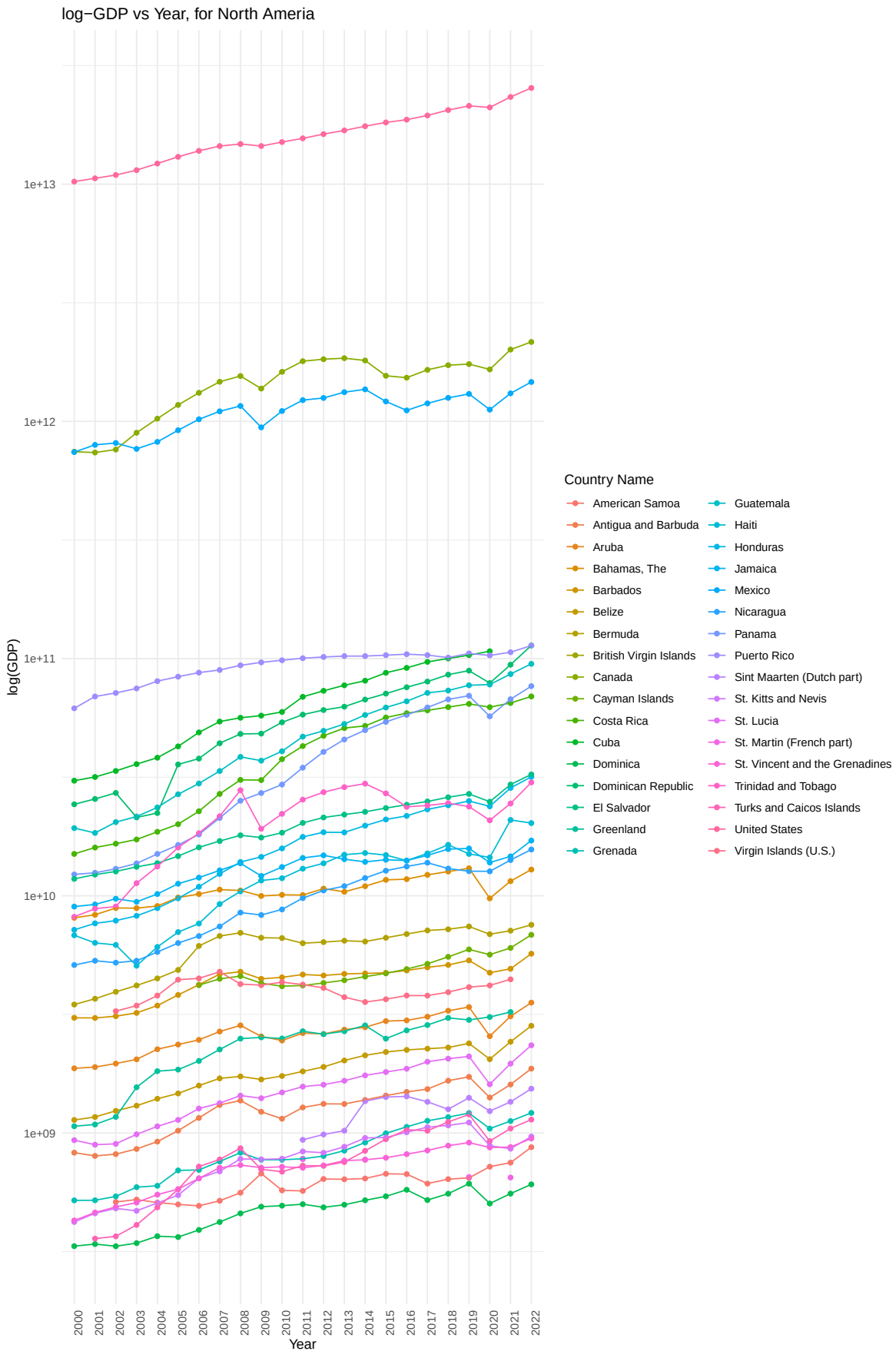
Solution:

```r
## replace this line with your code

country_info_merged %>%
  filter(
    `Indicator Name` == "GDP (current US$)",
    Continent == "North America"
  ) %>%
  select(
    "Country Name",
    "Country Code",
    "Indicator Name",
    "2000":"2022",
    "Continent"
  ) %>%
  select(
    !`Indicator Name`
  ) %>%
  melt(
    id.vars = c("Country Name", "Country Code", "Continent"),
    variable.name = "Year",
    value.name = "GDP"
  ) %>%
  ggplot(aes(x = Year,
             y = GDP)) +
  geom_point(aes(colour = `Country Name`)) +
  geom_line(aes(group = `Country Name`,
                colour = `Country Name`)) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 90)
  ) +
  scale_y_log10() +
  ylab("log(GDP)") +
  ggtitle("log-GDP vs Year, for North Ameria")
```

```
Warning: Removed 68 rows containing missing values or values outside the scale range
(`geom_point()`).
```

```
Warning: Removed 61 rows containing missing values or values outside the scale range
(`geom_line()`).
```


log−GDP vs Year, for North Ameria
log(GDP)
1e+13
1e+12
1e+11
1e+10
1e+09
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Year
Country Name
American Samoa
Antigua and Barbuda
Aruba
Bahamas, The
Barbados
Belize
Bermuda
British Virgin Islands
Canada
Cayman Islands
Costa Rica
Cuba
Dominica
Dominican Republic
El Salvador
Greenland
Grenada
Guatemala
Haiti
Honduras
Jamaica
Mexico
Nicaragua
Panama
Puerto Rico
Sint Maarten (Dutch part)
St. Kitts and Nevis
St. Lucia
St. Martin (French part)
St. Vincent and the Grenadines
Trinidad and Tobago
Turks and Caicos Islands
United States
Virgin Islands (U.S.)

Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Life Expectancy

Finally, let's see how female life expectancies have changed over time.

! Question 12

Plot female life expectancy vs time (again, only including years from 2000 onwards) for South American countries. Color by country, and comment on your plot. Which country (if any) has consistently had the lowest female life expectancies each year? Which (if any) has had the highest?

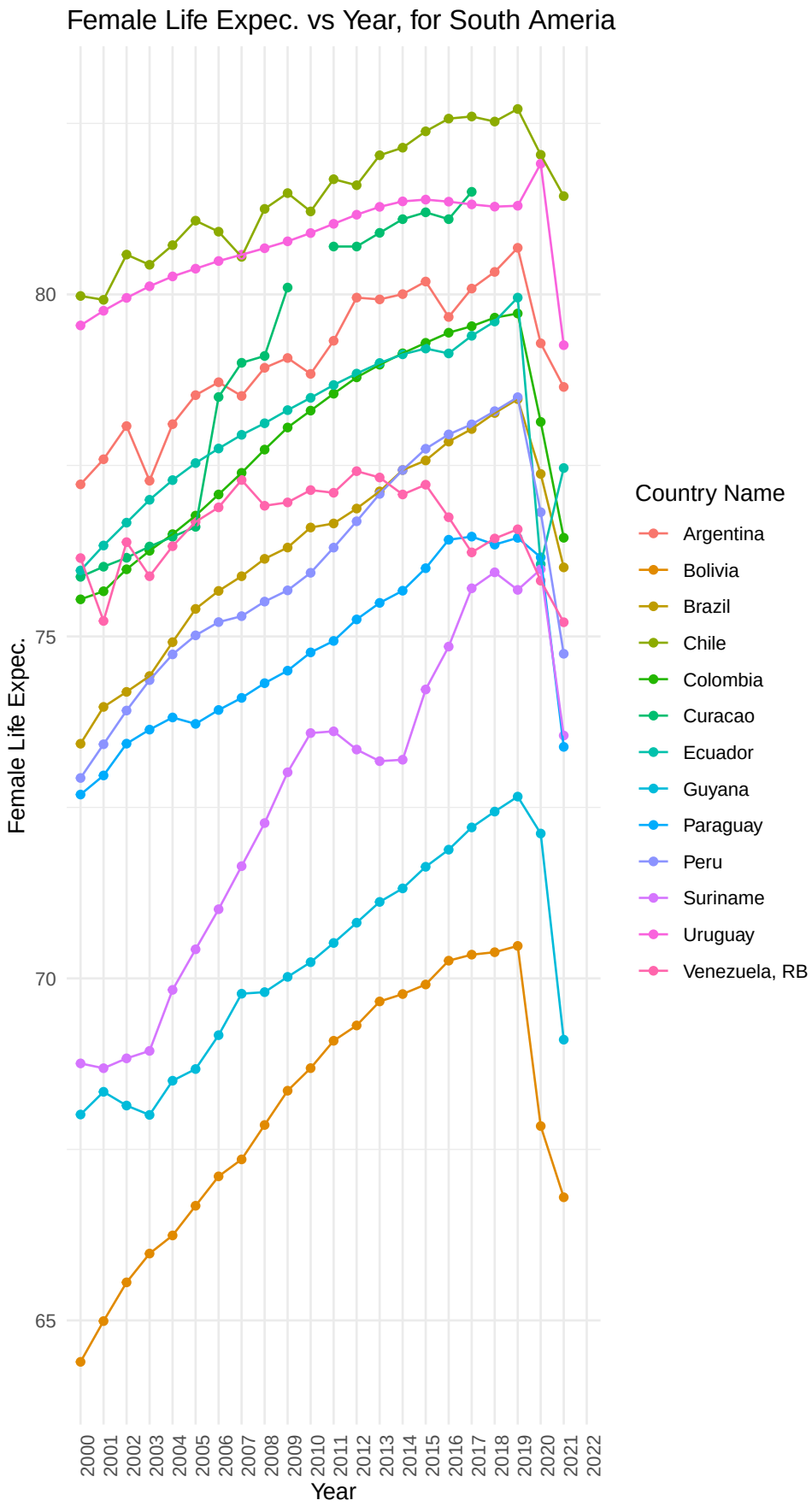
Solution:

```r
## replace this line with your code

country_info_merged %>%
  filter(
    `Indicator Name` == "Life expectancy at birth, female (years)",
    Continent == "South America"
  ) %>%
  select(
    "Country Name",
    "Country Code",
    "Indicator Name",
    "2000":"2022",
    "Continent"
  ) %>%
  select(
    !`Indicator Name`
  ) %>%
  melt(
    id.vars = c("Country Name", "Country Code", "Continent"),
    variable.name = "Year",
    value.name = "Female Life Expec."
  ) %>%
  ggplot(aes(x = Year,
             y = `Female Life Expec.`)) +
  geom_point(aes(colour = `Country Name`)) +
  geom_line(aes(group = `Country Name`,
                colour = `Country Name`)) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 90)
  ) +
  ggtitle("Female Life Expec. vs Year, for South Ameria")
```

```
Warning: Removed 18 rows containing missing values or values outside the scale range
(`geom_point()`).
```

```
Warning: Removed 17 rows containing missing values or values outside the scale range
(`geom_line()`).
```


Female Life Expec. vs Year, for South Ameria
Female Life Expec.
80
75
70
65
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
Year
Country Name
Argentina
Bolivia
Brazil
Chile
Colombia
Curacao
Ecuador
Guyana
Paraguay
Peru
Suriname
Uruguay
Venezuela, RB

Answer Check:
There is no autograder for this question; your TA will manually check that your answers are correct.

Submission Details

Congrats on finishing the lab! Please carry out the following steps:

Submission Details

1) Check that all of your tables, plots, and code outputs are rendering correctly in your final `.pdf`.
2) Check that you passed all of the test cases (on questions that have autograders). You'll know that you passed all tests for a particular problem when you get the message "All tests passed!".
3) Submit **ONLY** your `.pdf` to Gradescope. Make sure to match pages to your questions - we'll be lenient on the first few labs, but after a while failure to match pages will result in point penalties.