ggplot2 Lab Part I

2025-09-11

Intro

Our goals for this lab is to

1. Make data visualizations with R
2. Describe said visualizations
3. Get the idea of how ggplot2 builds graphs one step at a time

The next ggplot2 lab will dive into the different types of graphs (eg boxplots) and give you a better variety of graphics. This one is suppose to be the minimally painful introduction.

Rebuilding the Nile River Graph

Preliminaries and Getting the Data

First we need to read in the data from my website and make sure it worked as expected.

```
#read the csv and call the data set my_nile
my_nile <- read.csv('https://vinnys-classes.github.io/data/Nile.csv')


#look at the first six rows using head() and the last six using tail() to make sure it worked like we e
head(my_nile)
```

```
##   flow year changepoint
## 1 1120 1871           0
## 2 1160 1872           0
## 3  963 1873           0
## 4 1210 1874           0
## 5 1160 1875           0
## 6 1160 1876           0
```

```
tail(my_nile)
```

```
##     flow year changepoint
## 95   912 1965           1
## 96   746 1966           1
## 97   919 1967           1
## 98   718 1968           1
## 99   714 1969           1
## 100  740 1970           1
```

```
#finally, we need to make sue the package is loaded. Note there isn't any quotations when we reference
library(ggplot2)
#HINT: if you get an error that read there is no package called 'ggplot2' then please run, in your cons


#You don't need to know the following comment/understand the code but I wanted to give you a more full
#Finally we need to change the class of a variable. Currently R has changepoint as an integer and we ne
class(my_nile$changepoint)
```

```
## [1] "integer"
```

```
my_nile$changepoint <- as.factor(my_nile$changepoint)
class(my_nile$changepoint)
```

```
## [1] "factor"
```

Question 1

Please make a new Rmd file and indicate the names and types of variables we have in our data.

There is a second thing that is optional that I prefer. Next to the knit button is a cog wheel (options). "Chunk Output Inline" will make a plot/output in general in Rmd files appear immediately below the code. I prefer "Chunk Output in Console" where the plot will show up on the right hand side of your screen. My preferred option treats the code like you are running it in the console and avoids making me scroll between the code and the graph. Food for thought.

Making a (blank) Plot

The following code tells the function ggplot() that the data set is my_nile (what I named the data set, yours can be different). It's blank because we haven't given any mappings.

```
ggplot(data = my_nile)
```

Coordinates

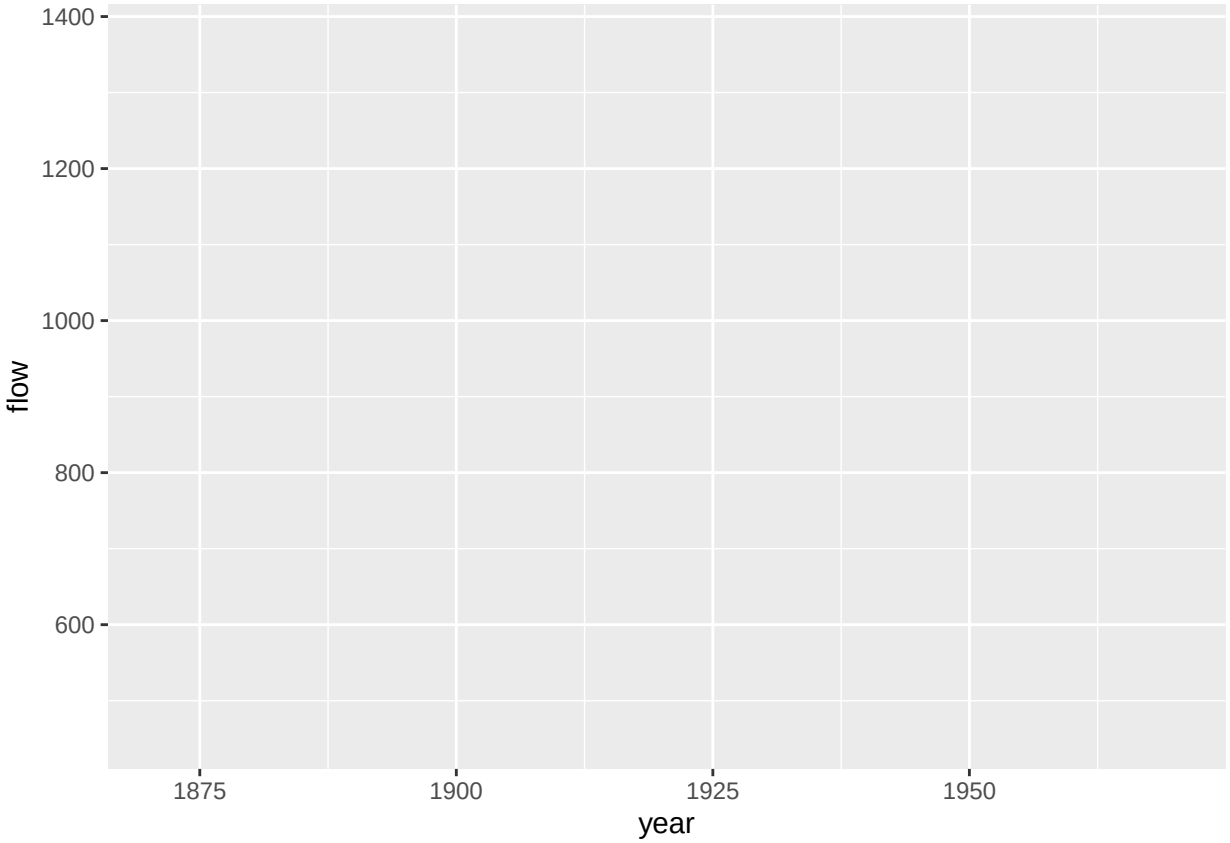
Generally ggplot is intelligent enough that we ourselves don't have to play with the coordinate system. Instead, we can just supply the variables of interest to the mapping parameter of ggplot(). In particular, we will use the aes(), short of aesthetics, to indicate the x and y variables.

IMPORTANT: It can be tempting to write the whole code on a single line but I recommend against that. Because functions can have many parameters (especially ggplot2) it is good practice to break them up onto multiple lines. R knows to ignore a new line between parameters of a function or (). The rational is that it makes your code

1. Easier to read
2. Easier to find things
3. Easier to comment/annotate

I will also note that generally R is smart enough that it makes indents of the right size so parameters to the same function are aligned. For example below "data" and "mapping" are both parameters for the ggplot() function so they are aligned. Similarly "x" and "y" are parameters for the aes() function and they too are aligned.

```
ggplot(data = my_nile,
       mapping = aes(x = year,
                     y = flow))
```

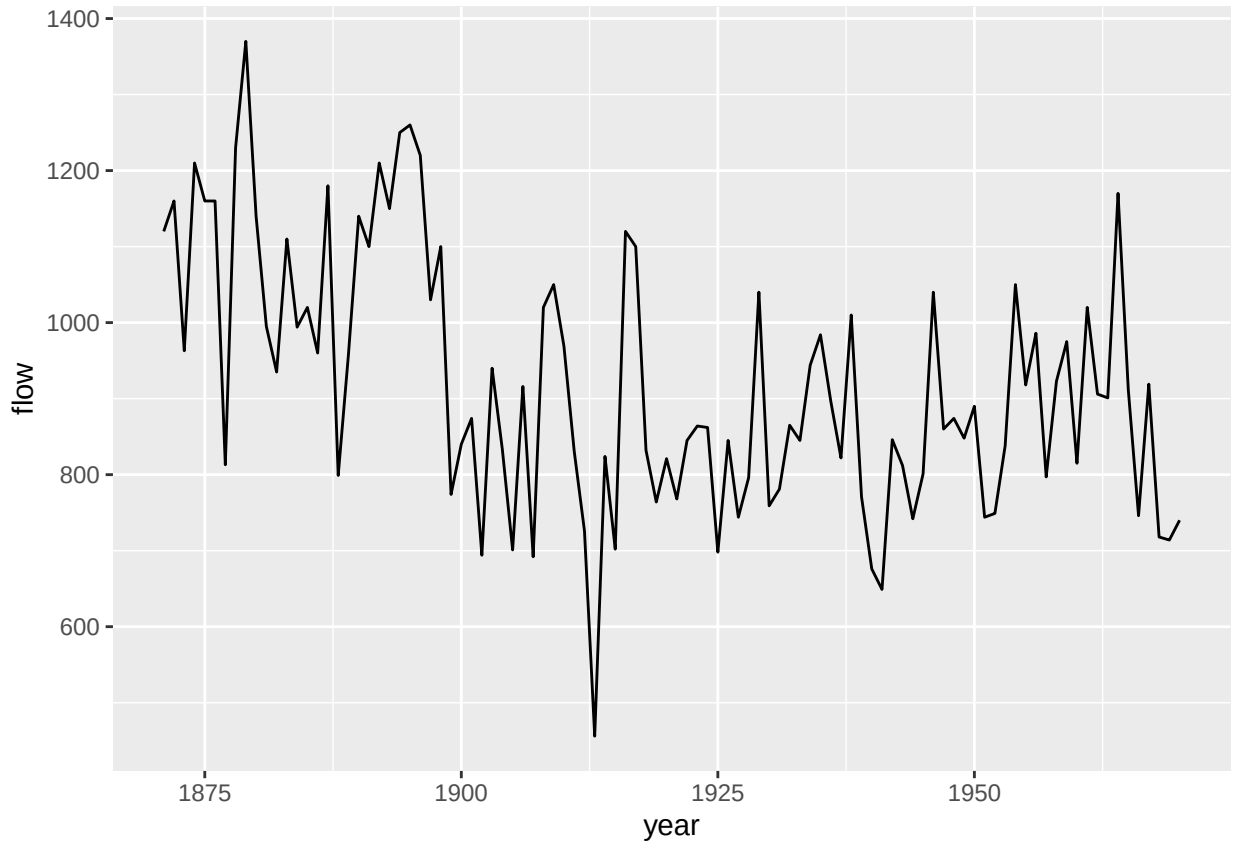


Alright, we now have a coordiante system on our graph. Our graph is still blank though but we are getting there. The next thing we need to add is going to be our first geometry

Geometry

The relevant geometry we are after is geom_line(). As previously stated, a lot of geometries you want will start with geom_. Not all but the majority. Please note that we use an "+" to add a new layer/geometry to our current graph

```
ggplot(data = my_nile,
       mapping = aes(x = year,
                     y = flow)) +
  geom_line()
```




```
#But we don't have to use geom_line(). Lets try another one like geom_point() which is used for scatter
```

Question 2

By editing the above code, please create a scatterplot that has points plotted instead of connected lines. Note that we do not (right now) want points *and* lines. This question just wants a scatterplot.

Question 3

Please comment on which of the two graphs you prefer and why. Work under the assumption we are trying to communicate to our audience how the Nile's flow has changed over time.

Two other popular geom_'s for our type of data are

1. geom_smooth() where R makes something akin to a curvy "free-hand' ' best fit line through the data. There is math involved but that is the idea (the math deals with fitting a polynomial to the data in a small region of the graph)
2. geom_jitter() which is very similar to geom_point() except the data points are *jittered*. That is, the points have a small amount of "white noise" added to them. An example of why this can be useful is in a later lab.

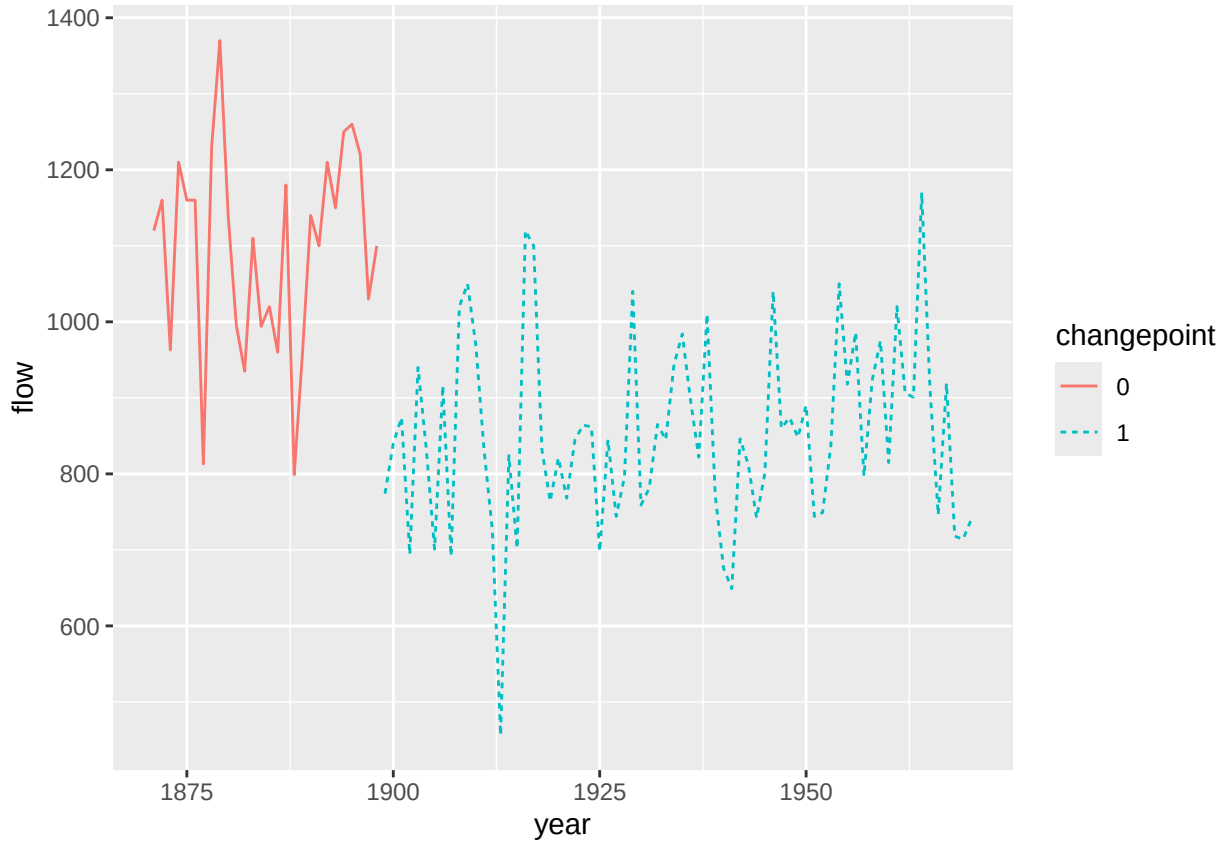
If you'd like you may try the above two and see what happens on the plot we have now (OPTIONAL)

Scales

We still have a variable (changepoint) that we have not used yet. It is believed that the flow rate of the River Nile changed in the late 1800's and changepoint has that information encoded. Let's add this variable to our graph as it could be useful to see the changes over time.

To do this we will edit geom_line(). It too has an aes() function that we can pass aesthetics to. We will add two scales at once. One scale will be color and the other will be the type of line.

```
ggplot(data = my_nile,
       mapping = aes(x = year,
                     y = flow)) +

  geom_line(mapping = aes(color = changepoint,
                          linetype = changepoint))
```



Question 4

Please discuss the difficulty needed to add line type in addition to color in order to make the graph color blind friendly.

Question 5

Let's go back to the scatterplot idea from Quesiton 2. Please remake that graph (feel free to copy and paste) but this time add **just** color to the graph.

Question 6

Because it's points and not lines the parameter linetype won't be useful. Instead, the parameter we'd like to set in it's place is called "shape". Please do this now.

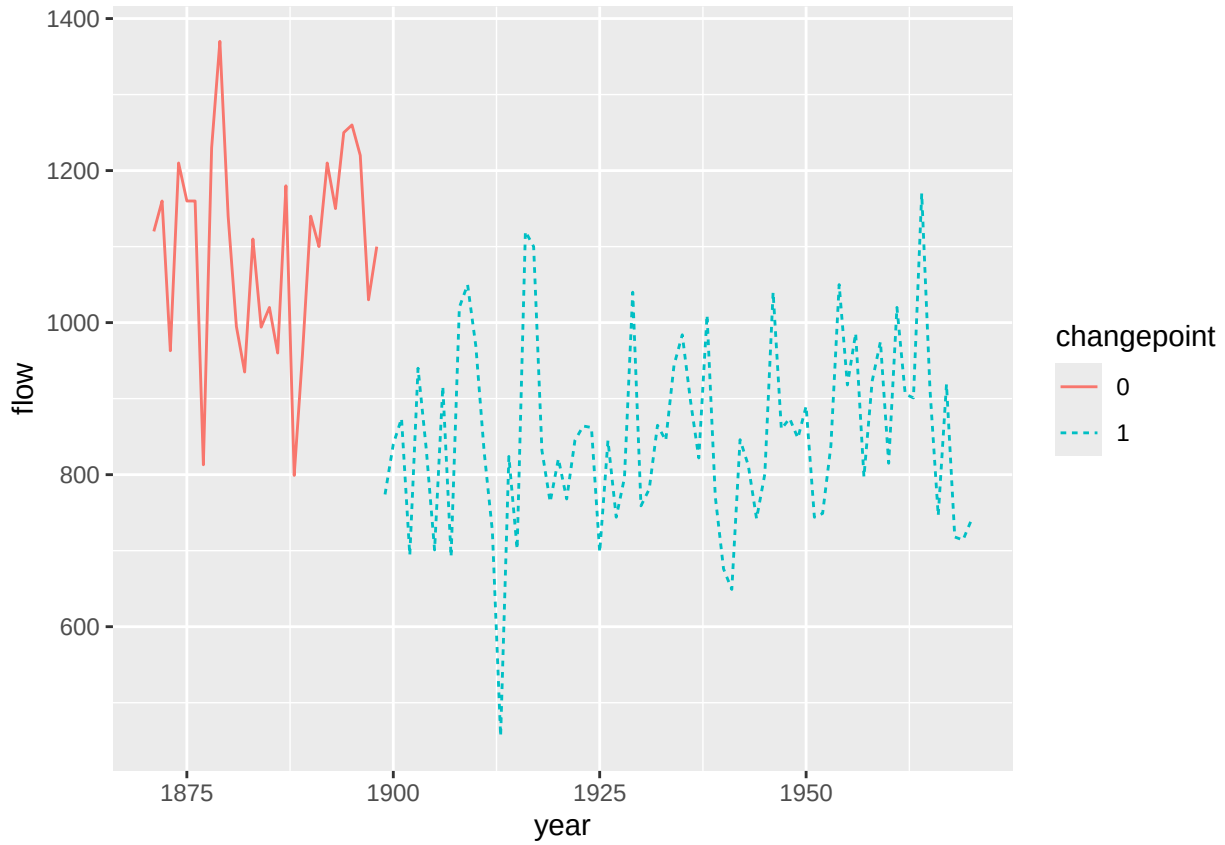
Question 7

Using any of the graphs made so far (with strong preferrence to the scatterplot), please describe the relationship between years and flow rate. You do not need to discuss the change point; I'm wanting you to practice describing a scatterplot. Mention the form, direction, strength, and outliers please.

Theme

We will do two different things with theme. First, which is less important of the two, is to change the background and general feeling of the graph. To do this go to the help page for theme_grey. First, please note how terrible Wickham's help pages are. It is one of the worst drawbacks of his work imo.

```
ggplot(data = my_nile,
       mapping = aes(x = year,
                     y = flow)) +

  geom_line(mapping = aes(color = changepoint,
                          linetype = changepoint)) +

  theme_grey()
```



Question 8

Please change the theme in the above code to another theme that you like more. Also comment on if anything look *good*. The second part won't be graded but I want you to refine your tastes/views in graphs and appreciating what isn't nice is a thing. HINT: We only need to worry about the bottom line of the code.

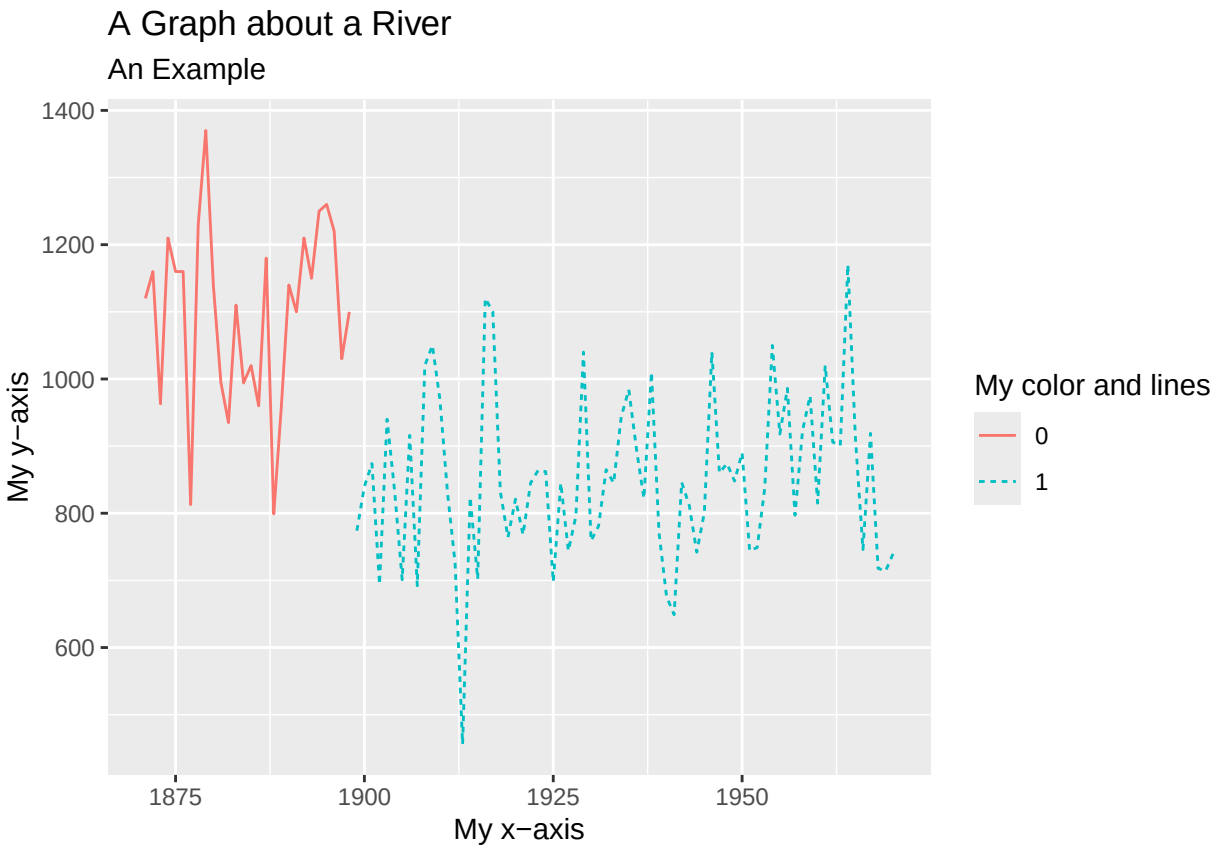
Our next goal is more important and helps make the graph accessible. We need to increase the font size. Currently I consider the default text of ggplot's to be too small (and grey for that matter....). Copy and paste the code you made for question 7 below.

Question 9

Again, on the help page for theme_grey there seems to be four parameters that are set. Read the description of them under the Arguments section and then set the parameter associated with font size to 15. HINT: whatever theme you choose, the parameter to set the font size will be the same

Officially I don't know if label is considered a thing under theme or scale or coordinates (I've never read Leland's book) but for us I don't think it matters. Let's work on adding labels to our graph. The relevant functions we want are labs() and xlab() and ylab() which we will add onto our current running chunks of code.

```
ggplot(data = my_nile,
       mapping = aes(x = year,
                     y = flow)) +

  geom_line(mapping = aes(color = changepoint,
                          linetype = changepoint)) +

  theme_grey() +

  xlab("My x-axis") +
  ylab("My y-axis") +

  labs(title = 'A Graph about a River', #Main title
       subtitle = 'An Example', #Subtitle
       color = 'My color and lines',
       linetype = 'My color and lines') #color and linetype; both of these need the same name otherwise
```



Question 10

Edit the above code so that... 1. The x- and y- axis have informative names

2. Make the title informative
3. List you and your lab mates in the subtitle (like you are authors)
4. Make the legend's text less wordy. HINT: Both color and linetype need to have the same name otherwise ggplot will make two seperate legends which is annoying and limits the point of **redundent coding**

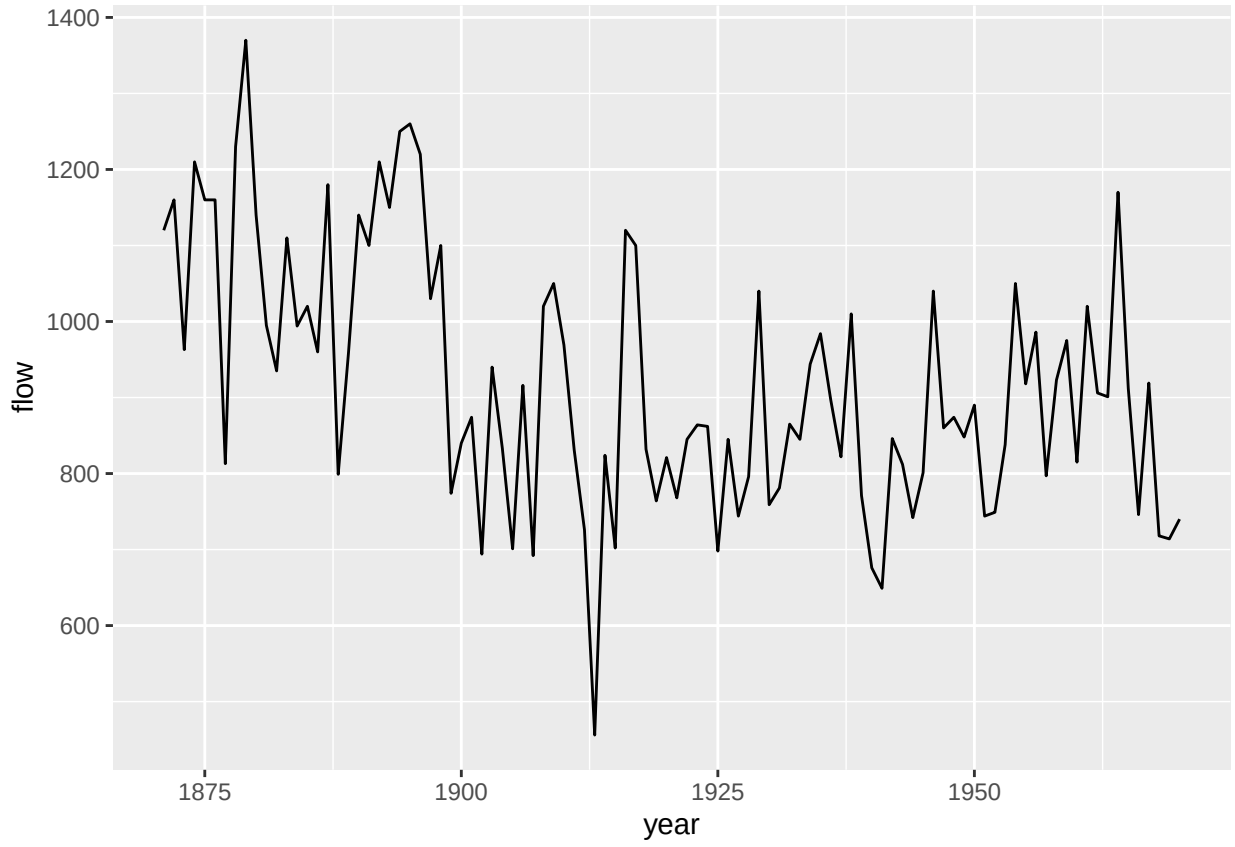
At this point you should have a graph that looks very similar to the ones in the notes with the addition of your names and different labels & theme.

Question 11

Let's go further for accessibility and create alt text. Look at the help page for labs() and read about the parameter called "alt". Follow the link in the description of the parameter to see an example of it used (bottom of the resulting page). Please add alt text to your graph. To do this, inside labs() you'll make a new line and set alt = "YOUR DESCRIPTION". Be sure to list the major takeaways of this graph that you'd like someone with poor vision to understand. Eg

NOTE: It's hard to see if it has worked but 1) there is generally a warning or error if it went wrong and 2) we can actually save our ggplot like we would a variable using the <- notation. Afterwards we can run get_alt_text() to see if it's there. For example....

```
my_example <- ggplot(data = my_nile,
                     mapping = aes(x = year,
                                   y = flow)) +
              geom_line() +
              labs(alt = 'This is example graph without colors')

my_example
```




```
get_alt_text(my_example)
```

```
## [1] "This is example graph without colors"
```

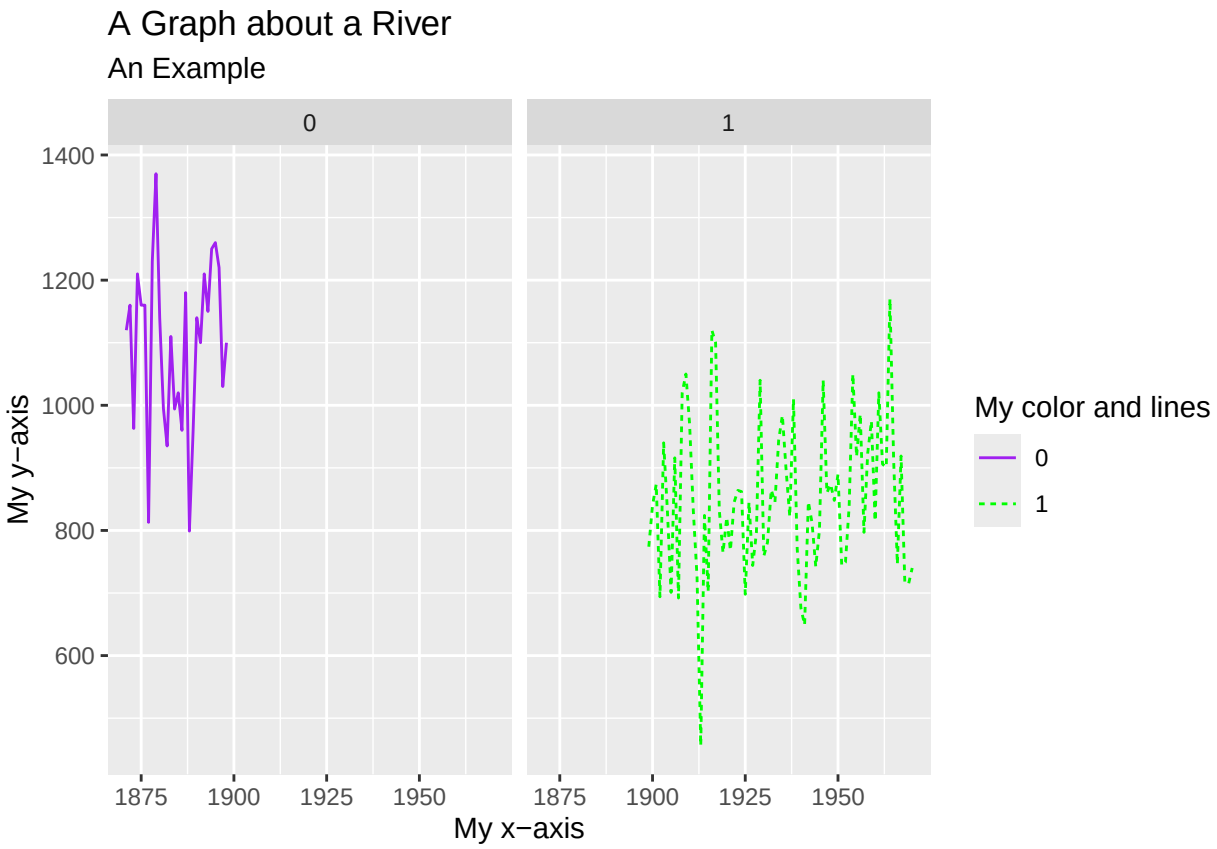
Facets

This isn't really a graph that we would facet but I want to give you the experience and round out this example. There are other things I'm not yet showing you. These will be dealt with later.

Facets allow us to subset the data on different (categorical) variables to see how each subpopulation of data behaves. Again, this example isn't a great one but it's worth seeing. We will use changepoint as our facet variable

```
ggplot(data = my_nile,
       mapping = aes(x = year,
                     y = flow)) +
```

```
geom_line(mapping = aes(color = changepoint,
                        linetype = changepoint)) +

theme_grey() +

xlab("My x-axis") +
ylab("My y-axis") +

labs(title = 'A Graph about a River',
     subtitle = 'An Example',
     color = 'My color and lines',
     linetype = 'My color and lines') +

facet_grid(. ~ changepoint)  +

scale_color_manual(values = c('purple', 'green'))
```



Question 12

Please explain why there is "blank" space to the right of the purple solid line on the left graph and empty space to the left of the green dashed line.

Question 13

Let's get rid of that blank space and instead make the axis "free" so they can have the most natural axis for the data in each pane. To do this, go to the help page for facet_grid() and look at the parameter called "scales". Choose the option that allows *only* the x-axis to be "free".

Question 14

Copy the graph made above in question 13 and change the side the tilda (the squiggly line) is on from the left side of the "changepoint" variable to the right side and then put a dot (eg changepoint ~ .) and then rerun the graph and comment on what changed.

Question 15

Above the code that changes the color of the graph to purple and green is the scale_color_manual(). Feel free to change the colors to whatever you prefer. You can use colors() to see all your color choices.