

Section 11 Report

PS 270, Spring 2020

Michael Masterson

13 August, 2020

Be sure to install stargazer and download library.bib to your rmd folder before running this document. You will need to install a LaTeX distribution (see below) before you can knit the document

YAML

The part at the very beginning of the R Markdown document that gives the title, author, date, and output format of the document written in a way of organizing data called YAML. For our purposes, all you need to know is that you will want to follow the title field with the title of your paper in quotes, the author field with your name in quotes, the date field either with the date in quotes or the code I put in all of my documents, including this one, that automatically includes the current date, and the type of output, which for papers should be pdf_document. In documents where you have citations, you will also need to include the name of the .bib file after bibliography (more on this later). I have added linkcolor: blue, so that the links I include will be blue in the knitted document. The last lines make the document double spaced. If you want single spacing, you can delete the two lines starting

with header-includes.

Componets of a Research Paper

A research paper should have an introduction that sets up the topic and states what the main argument will be. At the end of the introduction, there should be a road map paragraph that describes what each section of the paper will contain.

Each main section should be set off with a heading using `#` at the beginning of a line followed by the heading and with a blank line before and after. You can make subsections using `##`. Generally you do not want to include code in the knitted document. For this reason, I have included `knitr::opts_chunk$set(echo = FALSE, warning = FALSE, message = FALSE)` in my setup chunk, so that by default, the code in the code chunks will not be printed to the knitted document. I will still need to set `include=FALSE` for each chunk where I do not want the results of the code to be printed to the document.

A research paper should generally have main sections that cover:

1. The literature review and theory
2. The method and data
3. The results
4. The conclusion

A sample paper on voter id and voter turnout might look something like this:

An introduction setting up the question, why it is interesting, laying out the main argument and ending in a road map. Then it might have the following sections and subsections:

Voter ID and Turnout

Method

The data

Modeling Strategy

Results

Discussion

Conclusion

Knitting to PDF

When you create a document that you want to send to others, it is important to knit to pdf format rather than other formats. This is because only pdf documents will look the same regardless of what computer they are opened on.

In order to knit to pdf, you need to have a Latex distribution installed on your computer. This will be required for the final paper, so start working on getting it set up now so you can get help if there are problems.

If you use a Windows computer, you should install MiKTeX. You can download it [here](#).

If you use a Mac you should install MACtex, which you can download [here](#).

If you use Linux, just follow the instructions [here](#) to install TexLive.

Once you have a LaTeX distribution installed on your computer, you should be able to change the part at the beginning of the document to: `output: pdf_document`

Then your document will be knitted to pdf.

Basic formatting

You can make spaces between paragraphs simply by making two returns in the document.

Like this.

You can make a footnote by using the `^` character followed by square brackets with the text of the footnote.¹

Putting text between a single asterisk will make it italic. **Putting text between two asterisks will make it bold**

Hitting F7 in Rstudio will start spell check.

You can find more information about formatting [here](#).

You can make equations by putting them between dollar signs like this $2 + 2 = 4$. There are lot of different mathematical symbols you can use. [Here](#) is a helpful list of some of them.

Citations

In R markdown, you can (and should) generate your citations automatically based on a .bib file. You will need a .bib file that contains your citations in the same folder as your R markdown file. I recommend using a citation manager like Zotero (which is free). These will allow you to export a .bib file based on the citations you have in your manager. You can also

¹This is the text of a footnote.

make the .bib file manually. A good way to do this is to search for the document you want to cite on Google scholar, click the double quote button to get the citation, then click that you want the citation in BibTeX format. After that, you can simply copy and paste it into the document. I have included a sample of what a .bib files looks like called library.bib along with this R markdown file.

Citing Something in a Document

Once you have your .bib file set up and have referenced it at the beginning of a document, citing it is easy! All you need to do for a parenthetical citation is use square brackets that contain @ and the name of the document in your .bib file.

For example, this sentence cites my adviser's book on audience costs and war in authoritarian regimes (Weeks 2014).

If you want to make an in text citation without the parentheses, simply remove the square brackets. For example, according to Keele, Stevenson, and Elwert (2020) you should not interpret the coefficients on control variables in a regression as telling you the casual effect of those variables.

You can also cite specific page numbers like this: The rational mechanisms of war are private information, commitment problems, and issue indivisibilities (Fearon 1995, 281–82).

Any document that you cite in your Rmdarkdown document this way will automatically be cited in full at the end of your document.

Tables

Until now we have just viewed our results using the tidy() function. This is fine when you are viewing your results on your own, but it is not good enough for including results in a paper.

Lets say I want to include a regression in my paper with gdpPerCap as the independent variable and lifeExp as the dependent variable. First, I need to create the model, but I will set include=FALSE so the ugly output will not print to my paper.

Now I need to turn this model into something I can include in my paper. To do this, I can use the stargazer package. I will set results=“asis”, which means as is, so that the table will show up correctly in the paper.

stargazer takes the name of your model and a type argument, which should be set to “latex” when you are knitting to pdf. You can add a title to your paper using the title argument. I recommend setting the header argument=FALSE, so that it does not print a line saying that this table was made with stargazer. The covariate.labels argument takes what you want to label your independent variables in order. The dep.var.labels argument takes what you want to label your dependent variable.

Table 1: Effect of GDP Per Capita	
	<i>Dependent variable:</i>
	Life Exectancy
GDP Per Capita	0.001*** (0.00003)
Constant	53.956*** (0.315)
Observations	1,704
R ²	0.341
Adjusted R ²	0.340
Residual Std. Error	10.491 (df = 1702)
F Statistic	879.577*** (df = 1; 1702)
<i>Note:</i>	*p<0.1; **p<0.05; ***p<0.01

For the paper, you also need to include a table of summary statistics. This can also be accomplished using stargazer. Just give stargazer the dataframe that you want summary statistics from instead of a model. Stargazer does not like tibbles, so make sure you turn

your data frame into a data frame before giving it to stargazer.

I have added the `omit.summary.stat` argument to leave out the quartiles, which are not usually included in summary statistic tables.

Table 2: Gapminder Summary Statistics

Statistic	N	Mean	St. Dev.	Min	Max
Year	1,704	1,979.500	17.265	1,952	2,007
Life Expectancy	1,704	59.474	12.917	23.599	82.603
Population	1,704	29,601,212.000	106,157,897.000	60,011	1,318,683,096
GDP Per Capita	1,704	7,215.327	9,857.455	241.166	113,523.100

Practice 1

1. Create a `log_gdp` variable in `gapminder` that is the log of `gdp`.
2. Make a model with life Expectancy as the dependent variable and `log_gdp` as the independent variable.
3. Make a stargazer table of this model that will print correctly to a document that is knitted to pdf.

Figures

You can decide where in your paper you want a figure to show up as well as what to caption it. First lets make the graph that we want to display as a figure and assign it to an R object in a code chunk where we have set `include=FALSE`. For this example, lets graph the predicted effect of `gdp` on life expectancy.

Now that we have the plot the way we want it to show up in the paper. Lets make a special code chunk just to include the plot. At the top of the code chunk, you can set the caption of

the figure using the `fig.cap` argument. You can set the height and the width using `fig.width` and `fig.height`. By default, these sizes are in inches.

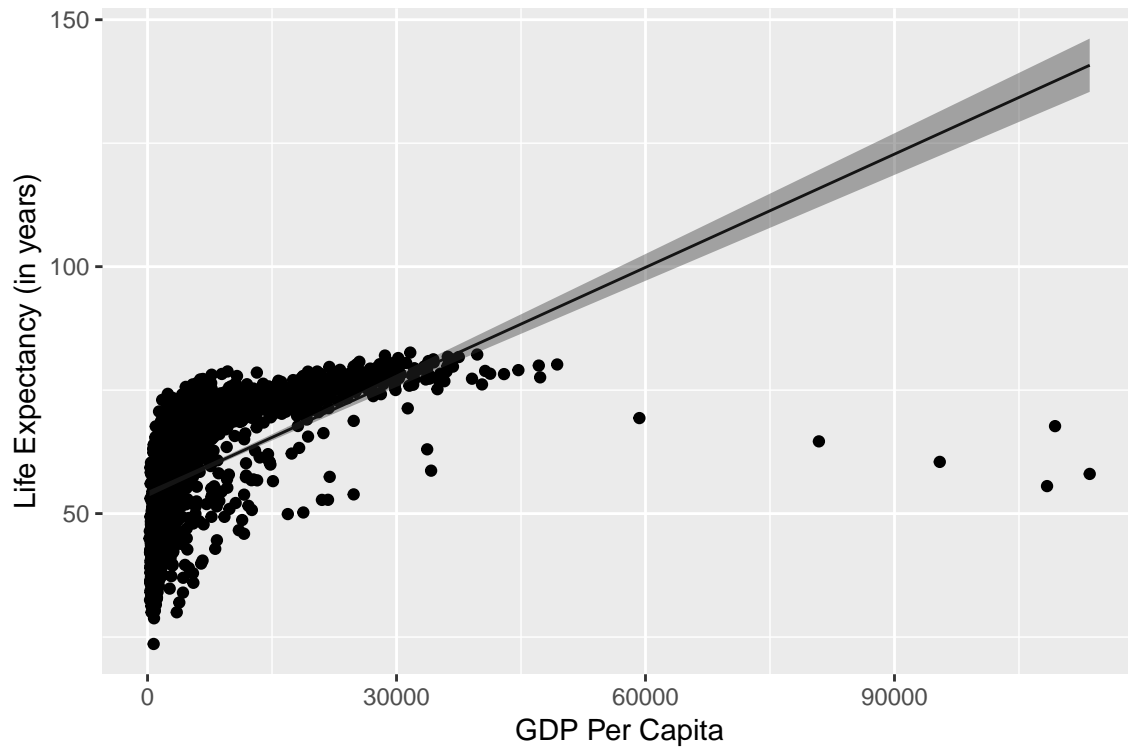


Figure 1: Predicted Value of Life Expectancy

Practice 2

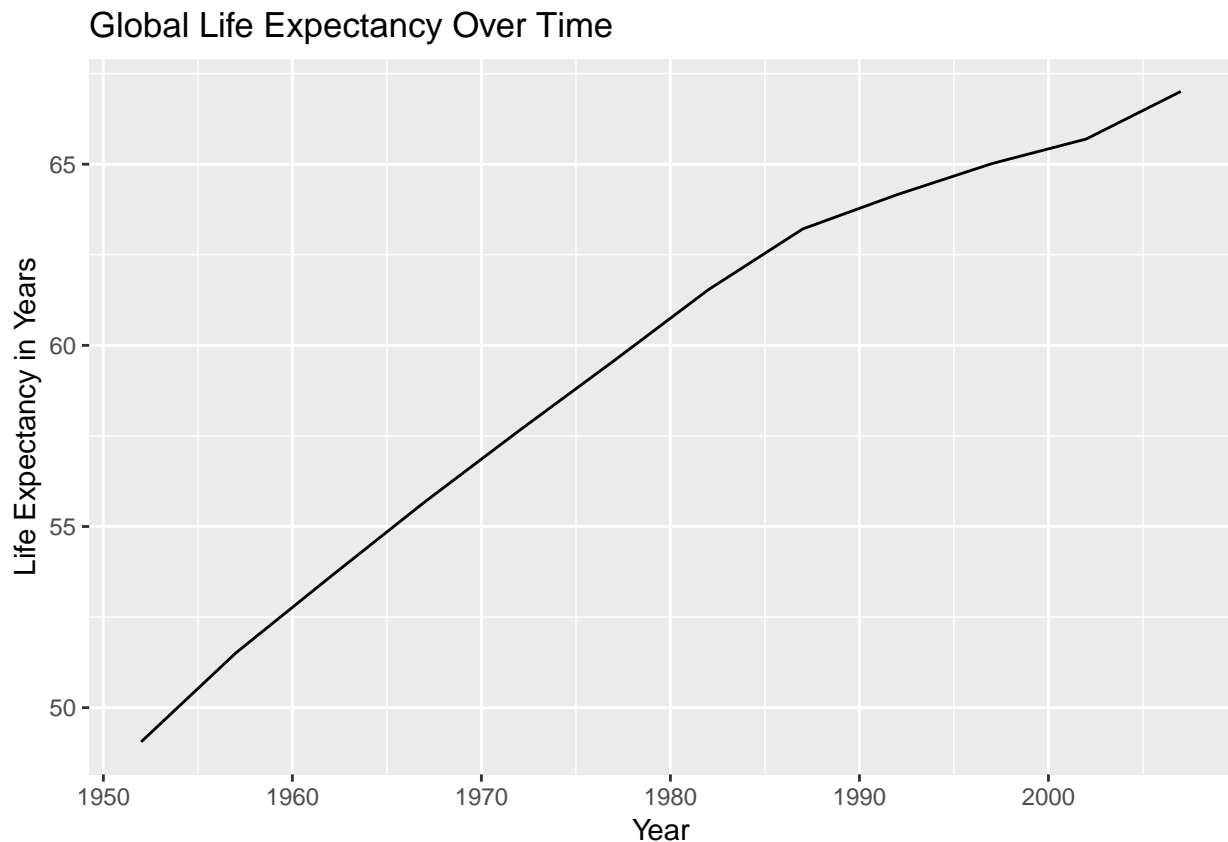
1. Use `augment` to get the predicted values of the dependent variable from your model in practice 1.
2. Calculate a confidence interval and graph the model.
3. Make the graph so that it will display correctly in the knitted pdf document. Give it a caption and set the height and width.

Plots Over Time

Gapminder is a panel data set because it has repeated observations of different countries over time. What if I want to plot the average value of life expectancy over time?

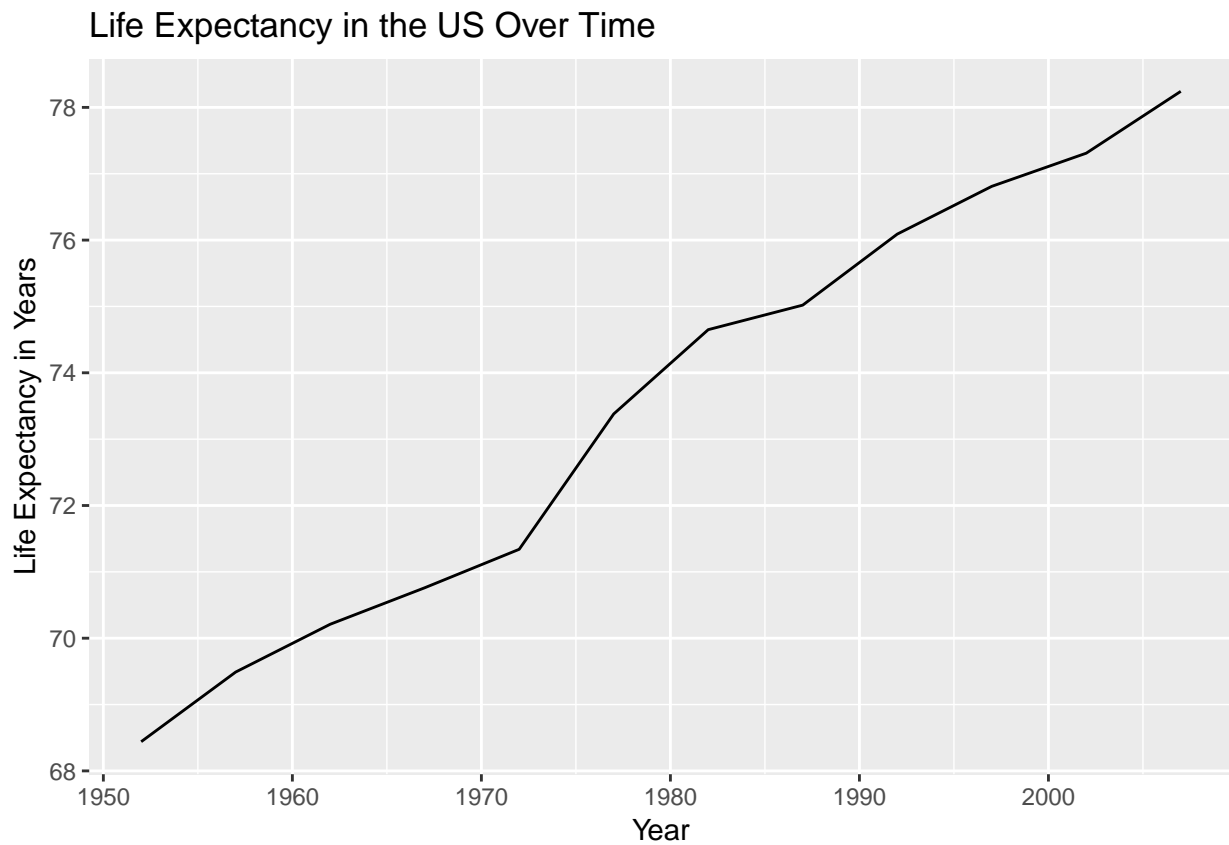
First I would create a data set containing the average life expectancy for each year by using `group_by` and `summarize`.

Then I could use this to make a plot over time with `ggplot`.

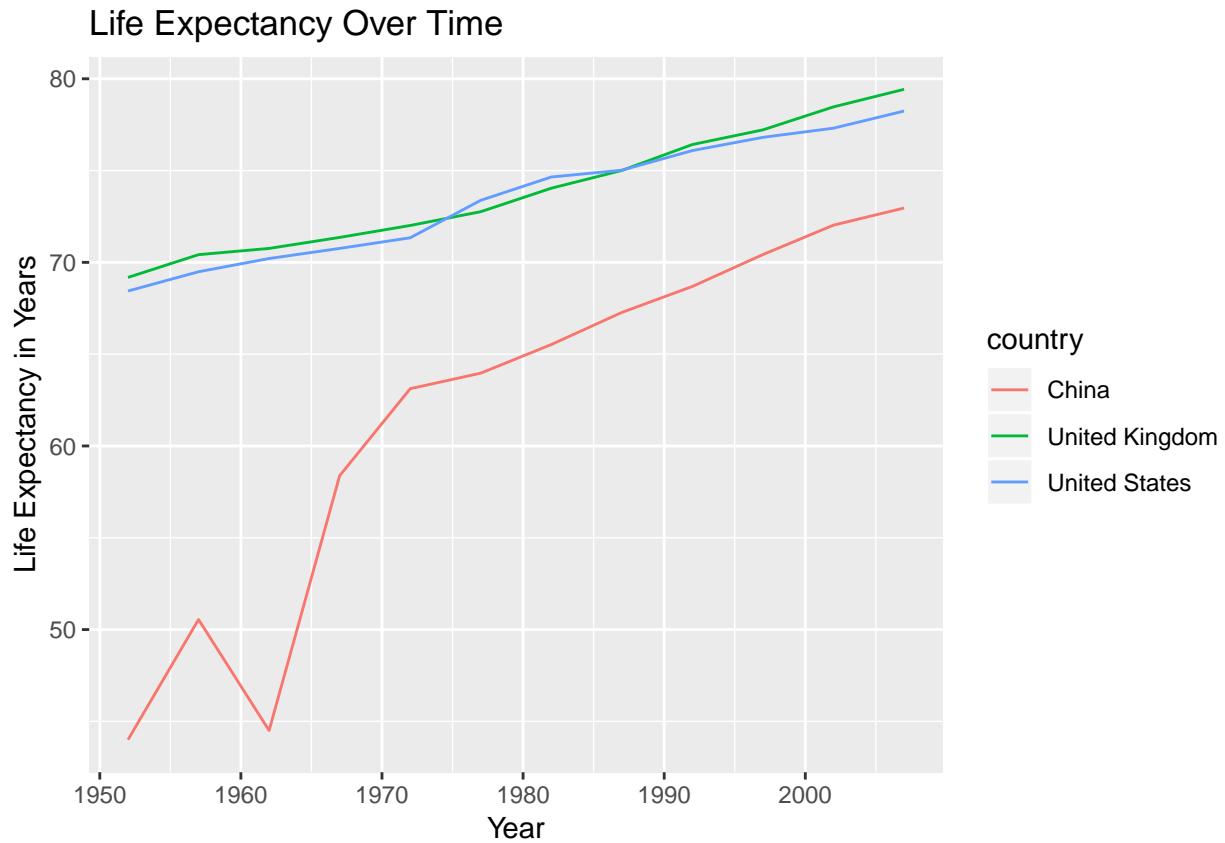


What if instead of the average global life expectancy, I just want to plot the life expectancy over time in a single country like the US?

I can use `filter` to select the country I want and then either assign that to a new data frame to use for the plot or just pipe it directly to the data frame argument of `ggplot`.



I should not plot every country because then there will be so many lines that it will be hard to see my plot, but what if I wanted to plot this for the US, the UK, and China? I can filter for those 3 countries and then use the group and color aesthetics to plot a different line in a different color for each.



Practice 3

1. Make a plot that shows the global average gdp per capita over time.
2. Make a plot that shows the gdp per capita over time for Mexico, the Netherlands, and Senegal.

References

Normally I could just end the document after `# References` and my references would print, but because we want to include a code Appendix after the references, I use the following line

to tell R to place my references here.

Fearon, James D. 1995. "Rationalist Explanations for War." *International Organization* 49 (3): 379–414.

Keele, Luke, Randolph T Stevenson, and Felix Elwert. 2020. "The Causal Interpretation of Estimated Associations in Regression Models." *Political Science Research and Methods* 8 (1): 1–13.

Weeks, Jessica LP. 2014. *Dictators at War and Peace*. Cornell University Press.

Code Appendix

```
my.model <- lm(lifeExp ~ gdpPercap, data=gapminder)
tidy(my.model)

stargazer(my.model, type="latex",
           title = "Effect of GDP Per Capita",
           header = FALSE,
           covariate.labels = c(
             "GDP Per Capita"
           ),
           dep.var.labels = "Life Exectancy"
)

gapminder <- data.frame(gapminder)

stargazer(gapminder, type="latex", header = FALSE,
           title="Gapminder Summary Statistics",
           covariate.labels = c("Year", "Life Expectancy"),
```

```

                                "Population", "GDP Per Capita"),
                                omit.summary.stat = c("p25", "p75"))

prediction <- augment(my.model)

prediction <- mutate(prediction,
  upperbound = .fitted + (.se.fit * 1.96),
  lowerbound = .fitted - (.se.fit * 1.96)
)

my.plot <- ggplot(prediction, aes(x=gdpPercap, y=lifeExp)) +
  geom_point() +
  geom_line(aes(y=.fitted)) +
  geom_ribbon(aes(ymin=lowerbound, ymax=upperbound), alpha = 0.4) +
  labs(x = "GDP Per Capita", y = "Life Expectancy (in years)")

my.plot

my.plot

summary.data <- gapminder %>%
  group_by(year) %>%
  summarise(avg_life_exp = mean(lifeExp, na.rm=TRUE)
  )

```

```

ggplot(summary.data, aes(x = year, y = avg_life_exp)) +
  geom_line() + # this line with make life expectancy
  labs(x = "Year", y = "Life Expectancy in Years", title = "Global Life Expectancy Over

filter(gapminder, country == "United States") %>%
  ggplot(aes(x = year, y = lifeExp)) +
  geom_line() +
  labs(x = "Year", y = "Life Expectancy in Years", title = "Life Expectancy in the US Over
filter(gapminder, country == "United States" | country == "United Kingdom" | country ==
  geom_line() + # this line with make life expectancy
  labs(x = "Year", y = "Life Expectancy in Years", title = "Life Expectancy Over Time")

```