MATP-4400 COVID-19 Final Notebook

Thomas Hopkins

May 2020

Contents

SUBMISSION CHECKLIST: Things to check before you submit (DELETE)

- Have you given your file(s) a clear, sensible name? `my_notebook.Rmd` or `joey.Rmd` are **not* acceptable!
- Is every figure clearly labeled and titled?
- Does every figure serve a purpose?
  - What question does it answer?
  - You can put extra (non-essential) figues in the Appendix.
- Are your tables clearly labelled and legible?
  - We recommend using `kable()` (built into `knitr`)
  - `xtable()` might also work for you (not as easy as `kable()`)
- **CRITICAL:** Have you given enough information for someone to reproduce, understand and extend your results?
  - Where can they *find* the data that you used?
  - Have you *described* the data that used?
  - Have you *documented* your code?
  - Are your figures *clearly labeled*?
  - Did you *discuss your findings*?
  - Did you use good grammar and *proofread* your results?
  - Finally, have you *committed* your work to github and made a *pull request*?

github Submission Instructions (DELETE!)

Fill out the bullets at the beginning of this notebook...

- Provide your github ID.
- Indicate the issues associated with this work on github.
- Commit the Rmd and html versions of this notebook to the `MATP-4400-FINAL` directory on github and make a pull request.

- Provide the link to your github submission on LMS.
  - **NOTE:** You can post your link any time after it exists

  - We will grade the latest version available on Github starting at 3:00 on Thursday, 7 May.

- **NOTE:** RStudio Server will be unavailable due to maintenance for up to a week beginning the weekend of 9 May, so there can be no extensions or late submissions.

github help (DELETE!)

- DO THIS FIRST:
  - `git clone https://github.com/TheRensselaerIDEA/COVID-Notebooks` (if you haven't already)
    - NOTE: We may have added to the master since your last `git pull` so make sure to do a `git pull origin master`
  - `git checkout -b <your-new-branch>` (more instructions immediately below)
  - In R, create a new R notebook (ie a `.Rmd` file) in the subdirectory `MATP-4400-FINAL`
  - `git add` it to the repo (make sure you're in the correct directory when you do this!)
  - `git commit -a -m 'Created my notebook!'` as soon as you do this!
  - `git push origin <your-new-branch>` so we see you're working on something
  - Post a link in LMS; it will remain valid as you update (and check in) your work to github
  - for details, see instructions in next few bullets
  - Contact John Erickson erickj4@rpi.edu for further github questions
- See: **HOW TO USE THIS NOTEBOOK AND REPOSITORY** https://bit.ly/2wYQGXP
- See also: **git Cheat Sheet** https://bit.ly/2yCBSi8

DELETE THE SECTIONS ABOVE!

Final Project: Submission Links

This should be the first section of your final project notebook. Fill out the following according to how you submitted your notebook!

- github repository: https://github.com/TheRensselaerIDEA/COVID-Notebooks
- My github ID: *thomashopkins32*
- github issues addressed by this work: `#18` and `#24` (example)
- Github branch name of my submitted notebook: *feature-24* (example)
- link to merged notebook (post these to LMS!:
  - https://github.com/TheRensselaerIDEA/COVID-Notebooks/blob/master/MATP-4400-FINAL/ThomasHopkins_FINAL_2020.Rmd
  - https://github.com/TheRensselaerIDEA/COVID-Notebooks/blob/master/MATP-4400-FINAL/ThomasHopkins_FINAL_2020.html

Overview & Problems Tackled

Two key issues in understanding the overall impact of an epidemic are determining how rapidly the disease is spreading and predicting future outcomes of the disease based on similar experiences. I first decided to estimate the effective reproductive number R_e for New York State over time. This gives a baseline for the expected number of individuals to contract the disease if one individual were to have it in the population. Second, I decided to look at properties of diseases that would help in predicting when a second wave of the virus may occur. This is done under the assumption that a second wave will definitely occur (something that should be analyzed separately). The results from the first issue show that the R_e is decreasing over time while the results of the second issue suggest that a second wave will occur sometime in the Fall of this year.

Data Description

Include data sources/locations, versions/dates, etc. The first issue uses time series data from the data/csv/time-series directory on the GitHub Repository. It includes the totals of confirmed cases for each state in the US over time. The date range is from 1/22/20 to 5/5/20.

The second issue uses data from many sources that I compiled into a suitable format for the task. The sources I used for the numbers found in the data file can be found in the References section below. Here is a view of the data I compiled:

```
diseases <- read.csv('../data/csv/similar_diseases.csv')
rownames(diseases) <- diseases[,'NAME']
diseases$NAME <- NULL
diseases
```

```
##                   R0 mortality_rate month_first_peak seasonal airborne
## SARS           2.750          0.065         February    False     True
## MERS           4.275          0.344            April    False     True
## H1N1           1.480          0.264             July     True     True
## Measles       12.000          0.150            April     True     True
## Cholera        2.000          0.050          October     True    False
## Yellow fever   6.150          0.075          January     True     True
## Ebola         18.000          0.500        September    False    False
## Dengue fever  27.200          0.010          January    False    False
## COVID-19       2.400          0.070         February    False     True
##               month_second_peak
## SARS                        May
## MERS                       June
## H1N1                    October
## Measles                   April
## Cholera                   March
## Yellow fever            January
## Ebola                  November
## Dengue fever            October
## COVID-19                   <NA>
```

I took features of many similar (in terms of impact on society) diseases. + `R0` denotes the average basic reproduction number + `mortality_rate` denotes the rate at which individuals who are infected die from disease + `month_first_peak` denotes the first month in which the disease became an epidemic + `seasonal` denotes whether or not the disease has a tendency to fluctuate with the seasons in terms of the number of infected individuals + `airborne` denotes whether or not the disease can be spread through the air + `policy_action` denotes whether significant policy action has taken place (social distancing, travel bans, etc.) + `month_season_peak` denotes the season in which a reemergence of the disease appeared after the initial outbreak

The diseases I included in this analysis include SARS, MERS, H1N1, Measles, Cholera, Yellow fever, Ebola, and Dengue fever.

Results

Break out your results by each problem you attacked

Problem 1

Describe the problem you are examining. If there is background that is necessary for this problem, then put it here. Include any references.

The first problem tackled was to look into the effective reproductive number (R_0) for New York State. Since R_0 is difficult to calculate with our current data, I chose to utilize the EpiEstim package to estimate R_0 called R_e. This used the incidence of COVID-19 in New York State as well as a serial interval distribution with a mean of 3.96 and a standard deviation of 4.75 (see 1 (ADD LINK TO REFERENCE)).

Methods

How did you address the problem? What data did you use exactly? What methods did you use?

Read in the time-series data for New York State. We need to calculate the incidence of COVID-19 and differentiate between imported and local cases. We assume the entries with less than 70 cases are considered imported. By imported we mean that the infected individual came from outside the region. Local cases are the entries with more than 70 cases.

```
# read in the time-series US data
covid_TS_states <- read.csv('../data/csv/time_series/time_series_covid19_confirmed_US.csv')
# filter data by New York and sum up the total number of cases
covid_TS_NY <- covid_TS_states[covid_TS_states$Province_State=='New York',] %>%
  group_by(Province_State) %>%
  summarize_if(is.numeric, sum, na.rm=TRUE)
# change the date columns into actual date objects
col_names_dates <- str_replace_all(str_remove(colnames(covid_TS_NY)[7:ncol(covid_TS_NY)], 'X'), coll('.
dates <- as.Date(col_names_dates, format = "%m/%d/%y")
# extract the time-series for cases
covid_cases_NY <- as.numeric(covid_TS_NY[,7:ncol(covid_TS_NY)])
# take cases larger than 70 total infected as local infections
local_cases_NY <- covid_cases_NY[covid_cases_NY > 70]
# take cases less than 70 as imported (from outside NYS) infections
imported_cases_NY <- covid_cases_NY[covid_cases_NY <= 70]
# pad both arrays with 0s to make them the same length
local_cases_NY <- as.numeric(padarray(local_cases_NY, padsize=c(0,length(dates)-length(local_cases_NY))
imported_cases_NY <- as.numeric(padarray(imported_cases_NY, padsize=c(0,length(dates)-length(imported_ca
# create a dataframe with dates, local, and imported columns
covid_cases_NY.df <- data.frame(dates=dates, local=local_cases_NY, imported=imported_cases_NY)
# take only cases after the first 41 rows since there were 0 infected at that time
covid_incidence_NY.df <- covid_cases_NY.df[41:nrow(covid_cases_NY.df),]
# compute the incidence (number of new cases) for both imported and local cases
covid_incidence_NY.df$local <- ave(covid_incidence_NY.df$local, FUN=function(x) c(0, diff(x)))
covid_incidence_NY.df$imported <- ave(covid_incidence_NY.df$imported, FUN=function(x) c(0, diff(x)))
# quick fix for this specific value since it is the cutoff point between local and imported
covid_incidence_NY.df$local[6] <- covid_incidence_NY.df$local[6] - covid_cases_NY.df$imported[45]
covid_incidence_NY.df[covid_incidence_NY.df < 0] <- 0
head(covid_incidence_NY.df)
```

```
##         dates local imported
## 41 2020-03-02     0        0
## 42 2020-03-03     0        1
## 43 2020-03-04     0        9
## 44 2020-03-05     0       12
## 45 2020-03-06     0        8
## 46 2020-03-07    45        0
```

Results

What were the results on this problem?

Now we calculate the estimated effective reproductive number (R_e) for COVID-19 in New York State. This is done by using the EpiEstim `estimate_R()` function. We then plot the time-series for incidence, the explored serial intervals, and R_e. We use a parametric serial interval curve with a mean of 3.96 and a standard deviation of 4.75.

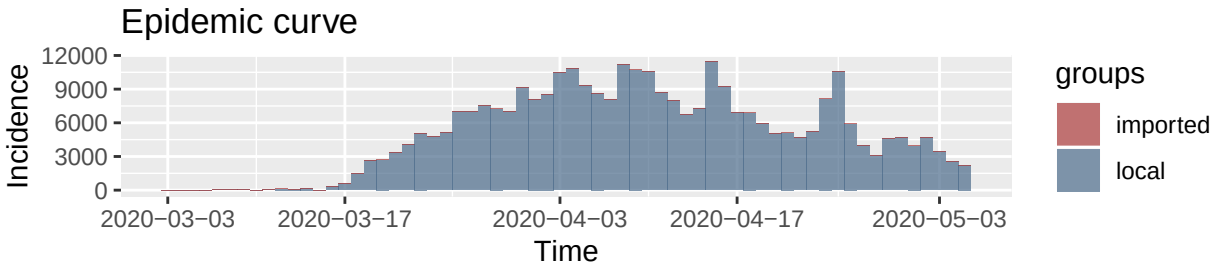
Source: https://www.medrxiv.org/content/10.1101/2020.02.19.20025452v4

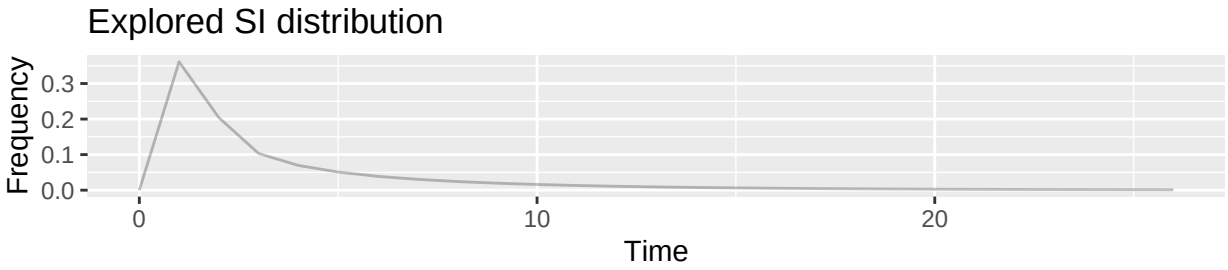
```
# set up plots so they can be viewed all at once
plot_Ri <- function(estimate_R_obj) {
    p_I <- plot(estimate_R_obj, "incid", add_imported_cases = TRUE)  # plots the incidence
    p_SI <- plot(estimate_R_obj, "SI")  # plots the serial interval distribution
    p_Ri <- plot(estimate_R_obj, "R")
    return(gridExtra::grid.arrange(p_I, p_SI, p_Ri, ncol = 1))
}
# calculate the curves using estimate_R function from EpiEstim package
NY_res_parametric_si <- estimate_R(covid_incidence_NY.df,
    method = "parametric_si", config = make_config(list(mean_si = 3.96, std_si = 4.75)))
```

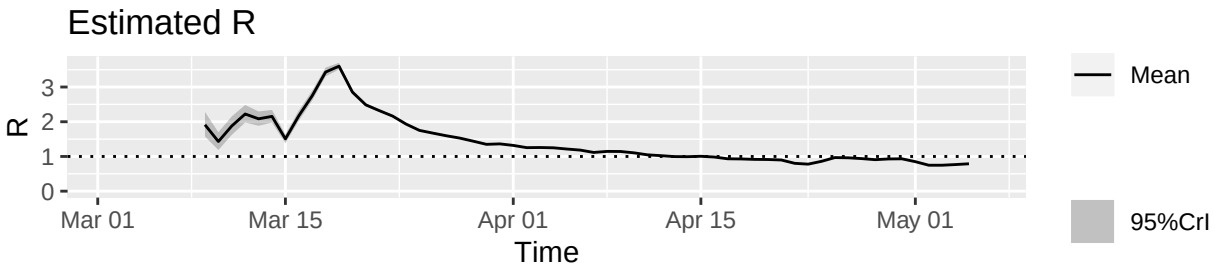
```
## Default config will estimate R on weekly sliding windows.
##     To change this change the t_start and t_end arguments.
```

```
# plot results
plot_Ri(NY_res_parametric_si)
```

```
## The number of colors (8) did not match the number of groups (2).
## Using `col_pal` instead.
```







Discussion

Interpret results. What were your findings? What do they say about COVID-19? What are the strengths and limitations of these results? Is there support for your findings from other sources? Include references as appropriate.

From the plots, it seems that the R_e for New York State currently hovers around the 1.0 line. This means that on average, one individual will transmit COVID-19 to one other individual. The goal for New York should be to get this number as close to zero as possible, although any R_e less than 1.0 is favorable. These plots also highlight what the R_e was earlier in the timeline and show how social distancing and other epidemic policies put in place have reduced R_e over time.

Problem 2

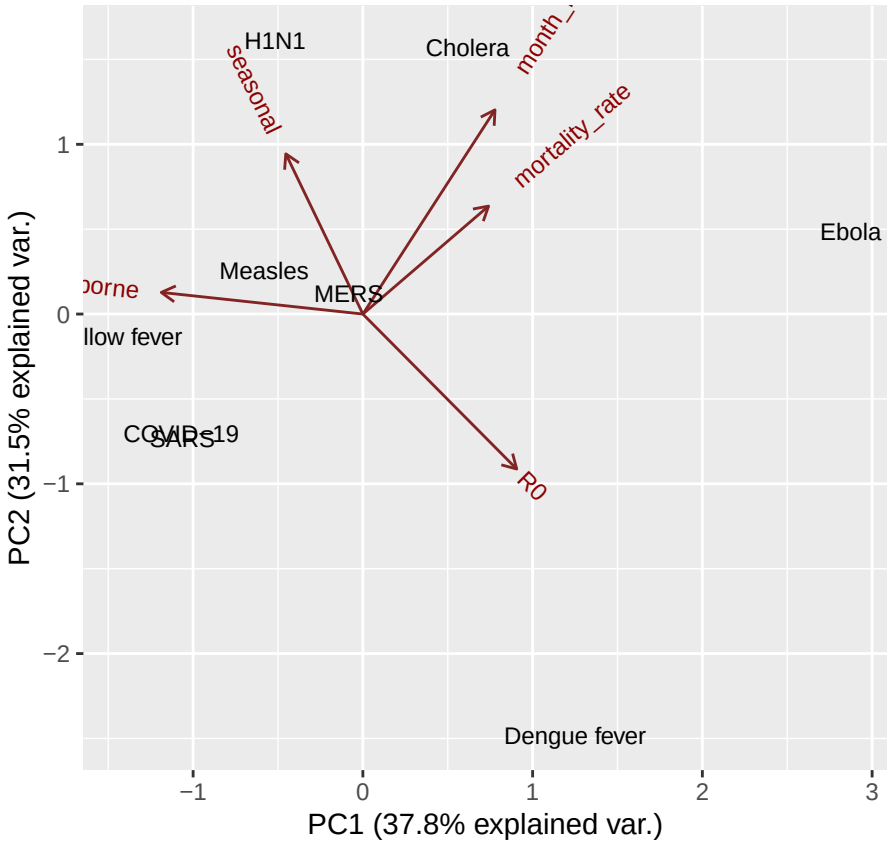
Methods

```
# read in the csv file containing the data
diseases <- read.csv('../data/csv/similar_diseases.csv')
# change rownames to match names of the disease
rownames(diseases) <- diseases[,'NAME']
# get rid of NAME column
diseases$NAME <- NULL
# change categorical data types to factors
months <- c('January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'Octo
diseases$month_first_peak <- as.numeric(factor(diseases$month_first_peak, levels=months))
diseases$seasonal <- as.numeric(as.factor(diseases$seasonal))
diseases$airborne <- as.numeric(as.factor(diseases$airborne))
diseases$month_second_peak <- factor(diseases$month_second_peak, levels=months)
```

```
# take out COVID-19 feature set for prediction later
covid_features <- diseases['COVID-19',]
covid_features$month_second_peak <- NULL
# create training data and training labels
train_data <- diseases[-c(nrow(diseases)),names(diseases) != 'month_second_peak']
train_labels <- diseases[-c(nrow(diseases)),'month_second_peak']
str(diseases)
```

```
## 'data.frame':    9 obs. of  6 variables:
##  $ R0               : num  2.75 4.28 1.48 12 2 ...
##  $ mortality_rate   : num  0.065 0.344 0.264 0.15 0.05 0.075 0.5 0.01 0.07
##  $ month_first_peak : num  2 4 7 4 10 1 9 1 2
##  $ seasonal         : num  1 1 2 2 2 2 1 1 1
##  $ airborne         : num  2 2 2 2 1 2 1 1 2
##  $ month_second_peak: Factor w/ 12 levels "January","February",..: 5 6 10 4 3 1 11 10 NA
```

```
# recreate dataframe with COVID-19 features
train_data_pca <- rbind(train_data, covid_features)
# scale numeric data
scaled_train_data <- scale(train_data_pca)
# perform PCA on scaled training data
my.pca <- prcomp(scaled_train_data, retx=TRUE)
# create biplot
ggbiplot(my.pca, choices=1:2, labels=rownames(train_data_pca), scale=0)
```




```
# must convert factors to numeric for predict function to scale correctly
train_data$month_first_peak <- as.numeric(train_data$month_first_peak)
train_data$seasonal <- as.numeric(train_data$seasonal)
```

```r
train_data$airborne <- as.numeric(train_data$seasonal)
# perform LDA on data
lda.fit <- lda(train_data, grouping=train_labels)
```

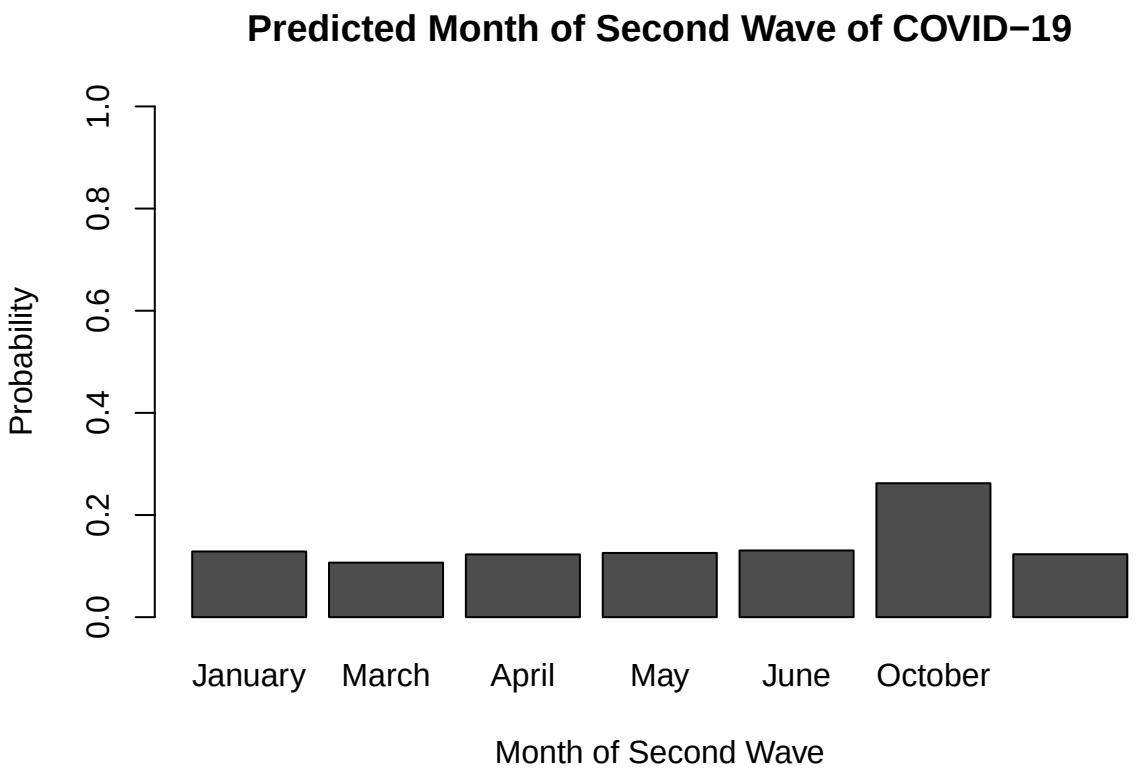
```
## Warning in lda.default(x, grouping, ...): groups February July August September
## December are empty
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```r
# get accuracy on training data
train_preds <- predict(lda.fit, train_data)$class
confusion.matrix <- table(train_labels, train_preds)
kable(confusion.matrix, type="html",digits = 2)
```

	January	February	March	April	May	June	July	August	September	October	November	De
January	0	0	0	0	0	0	0	0	0	1	0	
February	0	0	0	0	0	0	0	0	0	0	0	
March	0	0	0	0	0	0	0	0	0	1	0	
April	0	0	0	0	0	0	0	0	0	1	0	
May	0	0	0	0	0	0	0	0	0	1	0	
June	0	0	0	0	0	0	0	0	0	1	0	
July	0	0	0	0	0	0	0	0	0	0	0	
August	0	0	0	0	0	0	0	0	0	0	0	
September	0	0	0	0	0	0	0	0	0	0	0	
October	0	0	0	0	0	0	0	0	0	2	0	
November	0	0	0	0	0	0	0	0	0	1	0	
December	0	0	0	0	0	0	0	0	0	0	0	

```r
# predict outcome for COVID-19
covid_features$month_first_peak <- as.numeric(covid_features$month_first_peak)
covid_features$seasonal <- as.numeric(covid_features$seasonal)
covid_features$airborne <- as.numeric(covid_features$airborne)
covid_pred <- predict(lda.fit, covid_features)
barplot(covid_pred$posterior,
        ylab="Probability",
        xlab="Month of Second Wave",
        ylim=c(0, 1.0),
        main="Predicted Month of Second Wave of COVID-19")
```



Results

Discussion

Do at least 2 problems. Add more as necessary

Summary and COVIDMINDER Recommendations

- Overall, what insights did you find about the COVID-19 epidemic in your analysis?
- What recommendations do you have for COVIDMINDER for Data utilization, Analytics, Visualizations, User interface design, etc.
  - Would you recommend that your analysis be included in COVIDMINDER? Why or Why not?
  - If not, is there additional work that might improve the results? Note that it is perfectly okay for you to recommend to not to include your analysis. Research doesn't always work out. As a team we need to try a lot of different ideas in hopes of finding a few good ones to pursue. *

Think of yourself as a consultant reporting back on a particular aspect of the analysis and application design!

References

Include any references to literature in support of your work You might also wish to include references to unusual R packages essential to your work 1. https://www.medrxiv.org/content/10.1101/2020.02.19.20025452v4 2. EpiEstim CRAN package* https://www.cdc.gov/about/history/sars/timeline.htm https://www.hindawi.com/journals/av/2011/734690/ https://wwwnc.cdc.gov/eid/article/26/2/19-0697_article https://www.ncbi.nlm.nih.gov/pubmed/31813836 https://ajph.aphapublications.org/doi/pdf/10.2105/AJPH.2013.301704r https://www.cdc.gov/h1n1flu/surveillanceqa.htm https://www.ncbi.nlm.nih.gov/pubmed/28757186 https://journals.plos.org/plosntds/article/file?rev=2&id=10.1371/journal.pntd.0006158&type=printable https://www.ncbi.

nlm.nih.gov/pubmed/27846442 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3381442/ https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4397933/

Appendix

Include here whatever you think is relevant to support the main content of your notebook. For example, you may have only include example figures above in your main text but include additional ones here