Investigating Pertussis Resurgence Mini-Project

Joshua Cheung

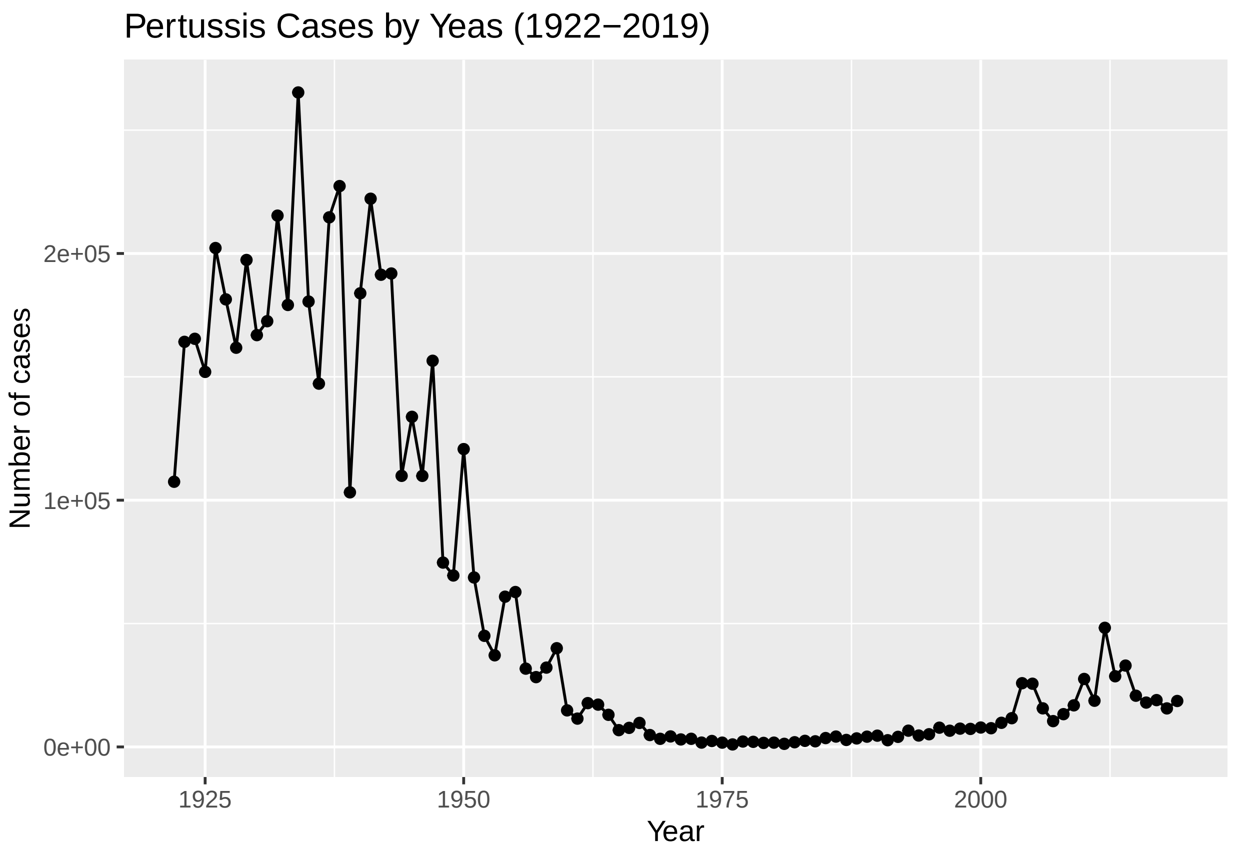
03/07/2022

1. Investigating pertussis cases by year

Q1. With the help of the R "addin" package datapasta assign the CDC pertussis case number data to a data frame called cdc and use ggplot to make a plot of cases numbers over time.

```
cdc <- data.frame(
                                 Year = c(1922L,1923L,1924L,1925L,
                                          1926L,1927L,1928L,1929L,1930L,1931L,
                                          1932L,1933L,1934L,1935L,1936L,
                                          1937L,1938L,1939L,1940L,1941L,1942L,
                                          1943L,1944L,1945L,1946L,1947L,
                                          1948L,1949L,1950L,1951L,1952L,
                                          1953L,1954L,1955L,1956L,1957L,1958L,
                                          1959L,1960L,1961L,1962L,1963L,
                                          1964L,1965L,1966L,1967L,1968L,1969L,
                                          1970L,1971L,1972L,1973L,1974L,
                                          1975L,1976L,1977L,1978L,1979L,1980L,
                                          1981L,1982L,1983L,1984L,1985L,
                                          1986L,1987L,1988L,1989L,1990L,
                                          1991L,1992L,1993L,1994L,1995L,1996L,
                                          1997L,1998L,1999L,2000L,2001L,
                                          2002L,2003L,2004L,2005L,2006L,2007L,
                                          2008L,2009L,2010L,2011L,2012L,
                                          2013L,2014L,2015L,2016L,2017L,2018L,
                                          2019L),
         No..Reported.Pertussis.Cases = c(107473,164191,165418,152003,
                                          202210,181411,161799,197371,
                                          166914,172559,215343,179135,265269,
                                          180518,147237,214652,227319,103188,
                                          183866,222202,191383,191890,109873,
                                          133792,109860,156517,74715,69479,
                                          120718,68687,45030,37129,60886,
                                          62786,31732,28295,32148,40005,
                                          14809,11468,17749,17135,13005,6799,
                                          7717,9718,4810,3285,4249,3036,
                                          3287,1759,2402,1738,1010,2177,2063,
                                          1623,1730,1248,1895,2463,2276,
                                          3589,4195,2823,3450,4157,4570,
                                          2719,4083,6586,4617,5137,7796,6564,
                                          7405,7298,7867,7580,9771,11647,
                                          25827,25616,15632,10454,13278,
                                          16858,27550,18719,48277,28639,32971,
                                          20762,17972,18975,15609,18617)
       )
```

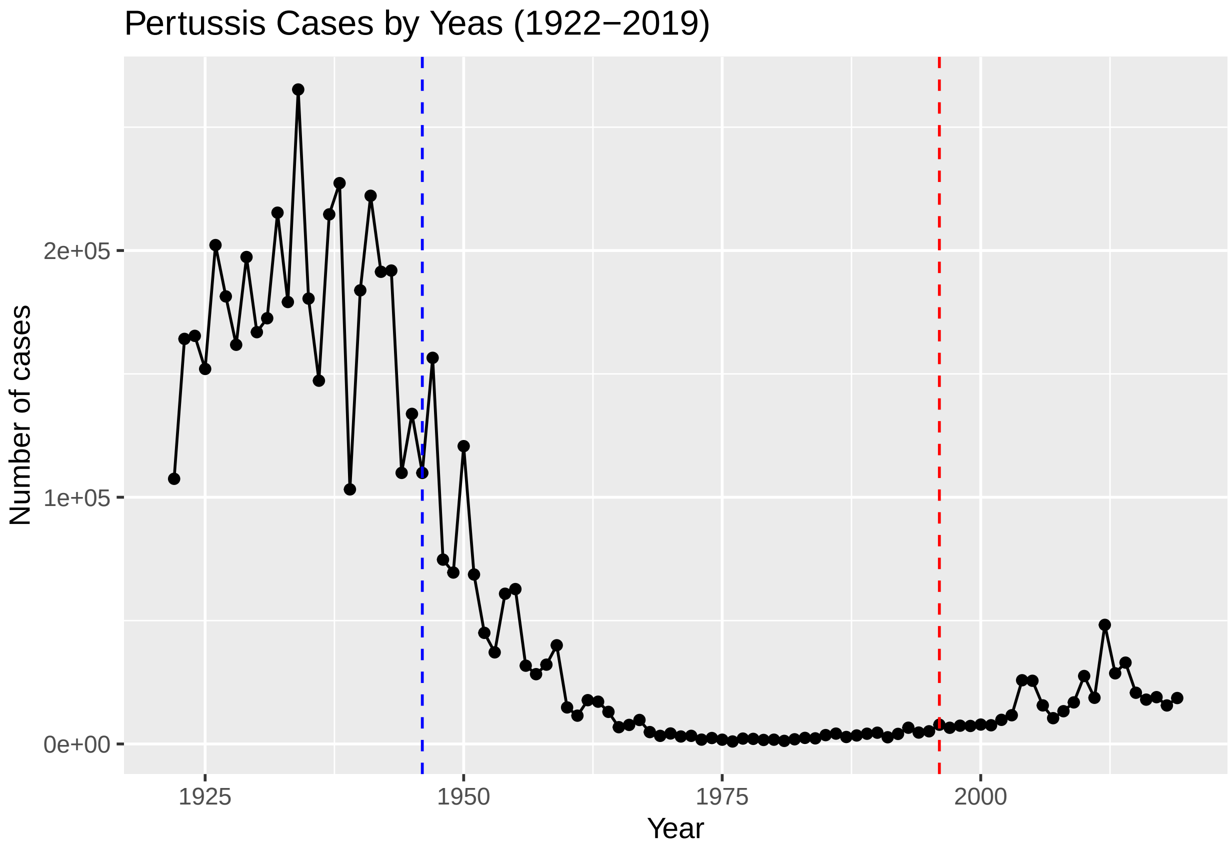
```
library(ggplot2)
ggplot(cdc) +
  aes(x=Year, y=No..Reported.Pertussis.Cases) +
  geom_point() +
  geom_line() +
  labs(title="Pertussis Cases by Yeas (1922-2019)",
       x="Year",
       y="Number of cases")
```



A tale of two vaccines (wP & aP)

> **Q2. Using the ggplot geom_vline() function add lines to your previous plot for the 1946 introduction of the wP vaccine and the 1996 switch to aP vaccine (see example in the hint below). What do you notice?**

```
ggplot(cdc) +
  aes(x=Year, y=No..Reported.Pertussis.Cases) +
  geom_point() +
  geom_line() +
  labs(title="Pertussis Cases by Yeas (1922-2019)",
       x="Year",
       y="Number of cases") +
  geom_vline(xintercept=1946, col="blue",linetype="dashed")+
  geom_vline(xintercept=1996, col="red",linetype="dashed")
```



To answer question 2, we see a clear decline in the number of pertussis cases after the introduction of the wP vaccine. The level of cases remained very low for many years. However, after the introduction of the aP vaccine we notice that there it a slight upward trend in the number of pertussis cases.

> **Q3. Describe what happened after the introduction of the aP vaccine? Do you have a possible explanation for the observed trend?**

We can see from the data that we collected from the CDC website that pertussis cases have been increasing again in recent years. More specifically, in 2012, the CDC reported 48,277 cases of pertussis in the U.S., which was the largest number of cases reported since 1955, where 62,786 cases were reported. There are several plausible theories for this observed trend. One possibility is that improvements in biological testing such as more accurate, effective and sensitive PCR testing, have allowed us to detect more pertussis cases. Additionally, there may be a social implication due to vaccine hesitance from the general population in recent years. Another theory is that, adolescents who were given the newer aP vaccine in infancy may experience a decrease in their immunity to pertussis relative to individuals who received the older wP vaccine. Finally, it is also entirely possible that the bacteria responsible for pertussis pathogenesis has evolved to avoid immune responses generated by the vaccines due to significant selective pressure to do so.

3. Exploring CMI-PB data

The CMI-PB API returns JSON data

The CMI-PB API (like most APIs) sends responses in JSON format. Briefly, JSON data is formatted as a series of key-value pairs, where a particular word ("key") is associated with a particular value.

To read these types of files into R we will use the read_json() function from the jsonlite package. The big advantage of using jsonlite for our current purposes is that it can simplify JSON key-value pair arrays into R data frames without much additional effort on our part.

```
# We call the jsonlite package
# This allows us to read, write and process JSON data
library(jsonlite)
```

Let's now read the main subject database table from the CMI-PB API.

```
subject <- read_json("https://www.cmi-pb.org/api/subject", simplifyVector = TRUE)
```

```
# We now examine the first 3 rows
head(subject, 3)
```

```
##   subject_id infancy_vac biological_sex              ethnicity  race
## 1          1          wP         Female Not Hispanic or Latino White
## 2          2          wP         Female Not Hispanic or Latino White
## 3          3          wP         Female                Unknown White
##   year_of_birth date_of_boost   study_name
## 1    1986-01-01    2016-09-12 2020_dataset
## 2    1968-01-01    2019-01-28 2020_dataset
## 3    1983-01-01    2016-10-10 2020_dataset
```

Q4. How may aP and wP infancy vaccinated subjects are in the dataset?

```
table(subject$infancy_vac)
```

```
## 
## aP wP 
## 47 49 
```

There are 47 aP infancy vaccinated subjects in the dataset and 49 wP infancy vaccinated subjects in the dataset.

Q5. How many Male and Female subjects/patients are in the dataset?

```
table(subject$biological_sex)
```

```
## 
## Female   Male 
##     66     30 
```

There are 66 females and 30 males subjects/patients in the dataset.

> **Q6. What is the breakdown of race and biological sex (e.g. number of Asian females, White males etc…)?**

The breakdown of race and biological sex can be seen in the following table:

```
table(subject$biological_sex, subject$race)
```

```
##
##          American Indian/Alaska Native Asian Black or African American
##   Female                             0    18                         2
##   Male                               1     9                         0
##
##          More Than One Race Native Hawaiian or Other Pacific Islander
##   Female                  8                                         1
##   Male                    2                                         1
##
##          Unknown or Not Reported White
##   Female                      10    27
##   Male                         4    13
```

Side-Note: Working with dates

Two of the columns of subject contain dates in the Year-Month-Day format. Recall from our last mini-project that dates and times can be annoying to work with at the best of times. However, in R we have the excellent lubridate package, which can make life allot easier. Here is a quick example to get you started:

```
# Call the lubridate package
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

We observe what today's date is (at the time of writing this of course) with the following:

```
today()
```

```
## [1] "2022-03-08"
```

We observe how many days have passed since new year 2000 with the following:

```
today() - ymd("2000-01-01")
```

```
## Time difference of 8102 days
```

We convert this time difference in years as follows:

```
time_length( today() - ymd("2000-01-01"),  "years")
```

```
## [1] 22.18207
```

we note that here we are using the ymd() function to tell lubridate the format of our particular date and then the time_length() function to convert days to years.

> **Q7. Using this approach determine (i) the average age of wP individuals, (ii) the average age of aP individuals; and (iii) are they significantly different?**

```
# We use todays date to calculate age in days
subject$age <- today() - ymd(subject$year_of_birth)
```

```
# Call the dplyr package
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Analysis for aP
ap <- subject %>% filter(infancy_vac == "aP")
round( summary( time_length( ap$age, "years" ) ) )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      22      24      25      24      25      26
```

```
# Anaysis for wP
wp <- subject %>% filter(infancy_vac == "wP")
round( summary( time_length( wp$age, "years" ) ) )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      27      31      34      35      39      54
```

Thus to answer the question 7, the average age of the wP individuals are 35 years. The average age of the aP individuals are 25 years. The aP and wP groups are indeed significantly different as their 5 number summaries do not align over the same spread.

> **Q8. Determine the age of all individuals at time of boost?**

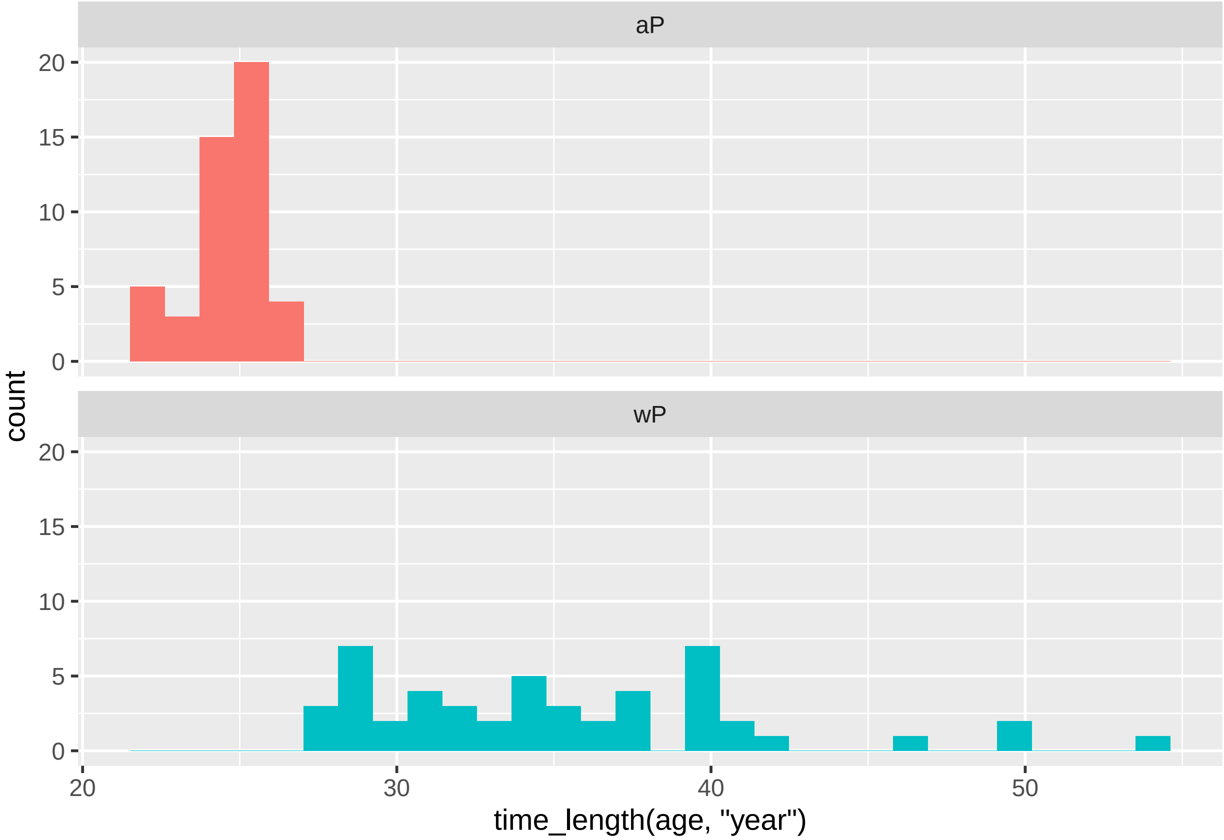
```
# The age of all individual at time of the boost is stored in R as follows:
int <- ymd(subject$date_of_boost) - ymd(subject$year_of_birth)
age_at_boost <- time_length(int, "year")
# We preview the first 6 ages as follows
head(age_at_boost)
```

```
## [1] 30.69678 51.07461 33.77413 28.65982 25.65914 28.77481
```

Q9. With the help of a faceted boxplot (see below), do you think these two groups are significantly different?

```
ggplot(subject) +
  aes(time_length(age, "year"),
      fill=as.factor(infancy_vac)) +
  geom_histogram(show.legend=FALSE) +
  facet_wrap(vars(infancy_vac), nrow=2)
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



The faceted bowplot reveals that the two groups are indeed signficantly different as their respective spread of data do not appear to overlap closely at all. We could even calculate a p-value as follows:

```
# Calculare p-value as follows:
x <- t.test(time_length( wp$age, "years" ),
       time_length( ap$age, "years" ))
x$p.value
```

```
## [1] 1.316045e-16
```

Joining multiple tables

Read the specimen and ab_titer tables into R and store the data as specimen and titer named data frames.

```
# We complete the API URLs
specimen <- read_json("https://www.cmi-pb.org/api/specimen",
                      simplifyVector = TRUE)
titer <- read_json("https://www.cmi-pb.org/api/ab_titer", simplifyVector = TRUE)
```

To know whether a given specimen_id comes from an aP or wP individual we need to link our specimen and subject data frames. The dplyr package has a family of join() functions that can help us with this common task:

> **Q9. Complete the code to join specimen and subject tables to make a new merged data frame containing all specimen records along with their associated subject details:**

```
meta <- inner_join(specimen, subject)
```

```
## Joining, by = "subject_id"
```

```
dim(meta)
```

```
## [1] 729  14
```

```
head(meta)
```

```
##   specimen_id subject_id actual_day_relative_to_boost
## 1           1          1                           -3
## 2           2          1                          736
## 3           3          1                            1
## 4           4          1                            3
## 5           5          1                            7
## 6           6          1                           11
##   planned_day_relative_to_boost specimen_type visit infancy_vac biological_sex
## 1                             0         Blood     1          wP         Female
## 2                           736         Blood    10          wP         Female
## 3                             1         Blood     2          wP         Female
## 4                             3         Blood     3          wP         Female
## 5                             7         Blood     4          wP         Female
## 6                            14         Blood     5          wP         Female
##                ethnicity  race year_of_birth date_of_boost   study_name
## 1 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 2 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 3 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 4 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 5 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 6 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
##          age
## 1 13215 days
## 2 13215 days
## 3 13215 days
## 4 13215 days
## 5 13215 days
## 6 13215 days
```

> **Q10. Now using the same procedure join meta with titer data so we can further analyze this data in terms of time of visit aP/wP, male/female etc.**

```
abdata <- inner_join(titer, meta)
```

```
## Joining, by = "specimen_id"
```

```
dim(abdata)
```

```
## [1] 32675    20
```

> **Q11. How many specimens (i.e. entries in abdata) do we have for each isotype?**

We use the table function on the isotype column to see how many specimens we have for each isotype as follows:

```
table(abdata$isotype)
```

```
##
##  IgE  IgG IgG1 IgG2 IgG3 IgG4
## 6698 1413 6141 6141 6141 6141
```

So to answer question 11, we have 6698, 1413, 6141, 6141, 6141, specimens for the IgE, IgG, IgG1, IgG2, IgG3, and IgG4 isotypes respectively.

> **Q12. What do you notice about the number of visit 8 specimens compared to other visits?**

We again use the table this time on the visit column to answer this question as follows:

```
table(abdata$visit)
```

```
##
##    1    2    3    4    5    6    7    8
## 5795 4640 4640 4640 4640 4320 3920   80
```

We notice that the number of visit 8 specimens is significantly lower relative to all the other visits.

4. Examine IgG1 Ab titer levels

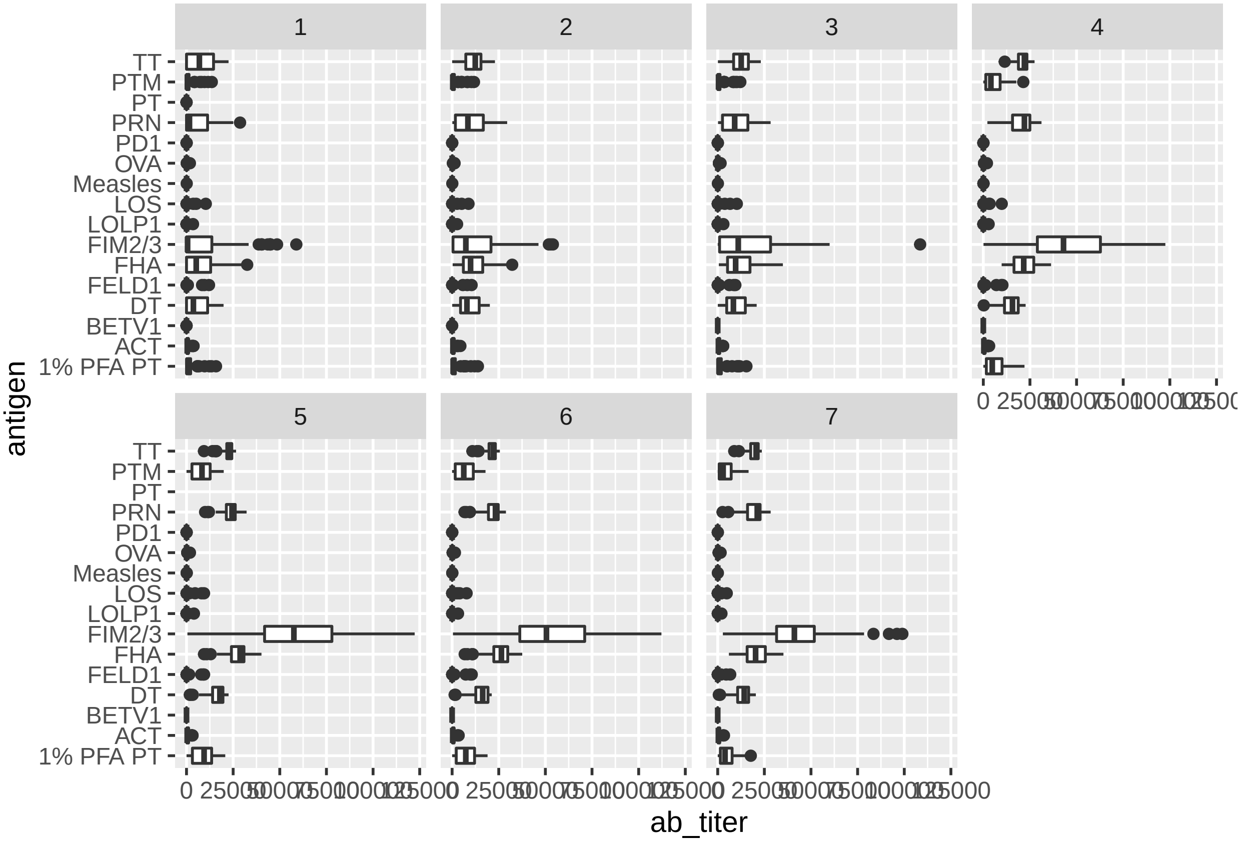
Now using our joined/merged/linked abdata dataset filter() for IgG1 isotype and exclude the small number of visit 8 entries.

```
ig1 <- abdata %>% filter(isotype == "IgG1", visit!=8)
head(ig1)
```

```
##   specimen_id isotype is_antigen_specific antigen   ab_titer  unit
## 1           1    IgG1                TRUE     ACT 274.355068 IU/ML
## 2           1    IgG1                TRUE     LOS  10.974026 IU/ML
## 3           1    IgG1                TRUE   FELD1   1.448796 IU/ML
## 4           1    IgG1                TRUE   BETV1   0.100000 IU/ML
## 5           1    IgG1                TRUE   LOLP1   0.100000 IU/ML
## 6           1    IgG1                TRUE Measles  36.277417 IU/ML
##   lower_limit_of_detection subject_id actual_day_relative_to_boost
## 1                 3.848750          1                           -3
## 2                 4.357917          1                           -3
## 3                 2.699944          1                           -3
## 4                 1.734784          1                           -3
## 5                 2.550606          1                           -3
## 6                 4.438966          1                           -3
##   planned_day_relative_to_boost specimen_type visit infancy_vac biological_sex
## 1                             0         Blood     1          wP         Female
## 2                             0         Blood     1          wP         Female
## 3                             0         Blood     1          wP         Female
## 4                             0         Blood     1          wP         Female
## 5                             0         Blood     1          wP         Female
## 6                             0         Blood     1          wP         Female
##                ethnicity  race year_of_birth date_of_boost   study_name
## 1 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 2 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 3 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 4 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 5 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
## 6 Not Hispanic or Latino White    1986-01-01    2016-09-12 2020_dataset
##          age
## 1 13215 days
## 2 13215 days
## 3 13215 days
## 4 13215 days
## 5 13215 days
## 6 13215 days
```

> **Q13. Complete the following code to make a summary boxplot of Ab titer levels for all antigens:**

```
ggplot(ig1) +
  aes(ab_titer, antigen) +
  geom_boxplot() +
  facet_wrap(vars(visit), nrow=2)
```

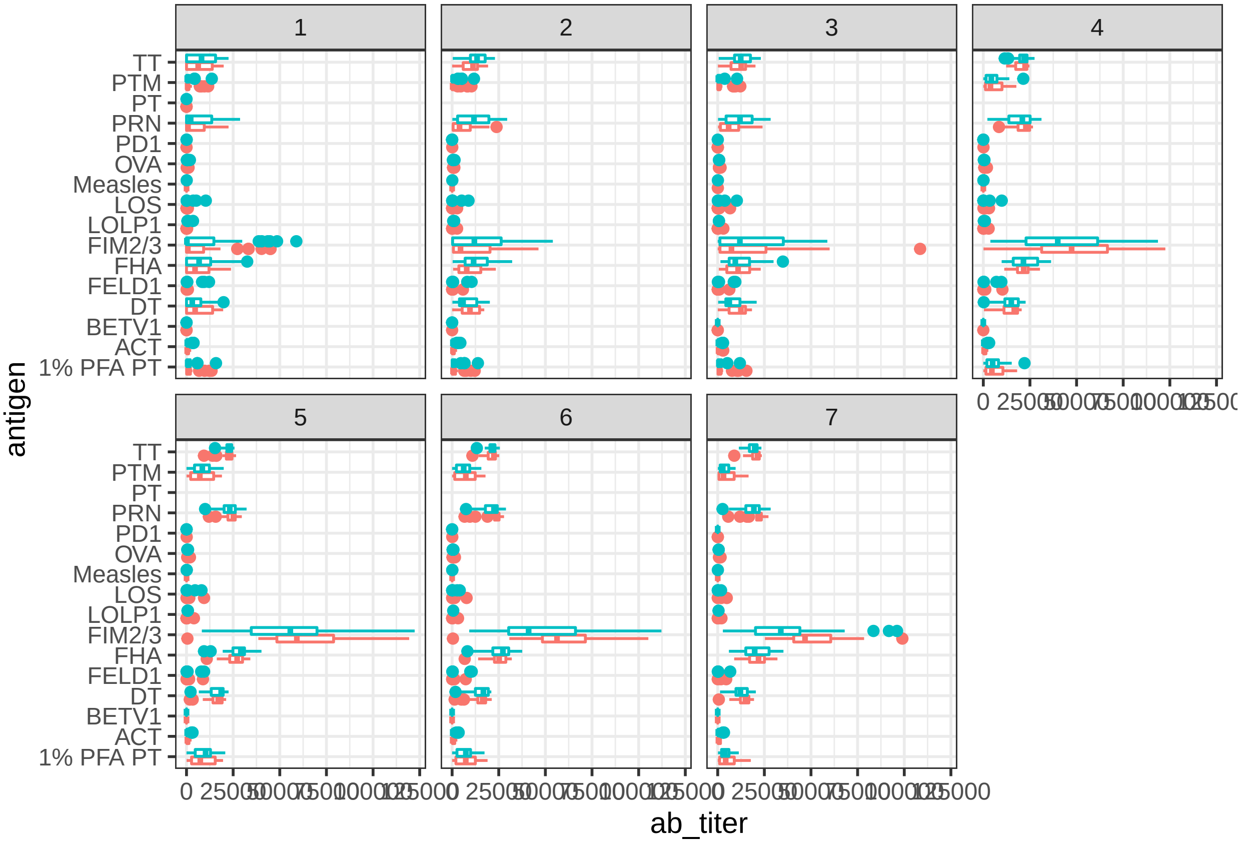


Q14. What antigens show differences in the level of IgG1 antibody titers recognizing them over time? Why these and not others?

We see that the FIM2/3 antigens show clear differences in the level if IgG1 antibody titers recognizing them over time. FIM2/3 are the extracellular fimbriae proteins from *B. pertussis* that participate in substrate attachment. Since we have already seen evidence of pertussis cases rising in recent years it would not be unreasonable to see these fimbriae proteins from *B. pertussis* exhibiting differences in the ability of the immues sytem to recognize them in the form of variations of the IgG1 antibodies titers recognition over time The other antigens do not exhibit this level of difference as they are likely not associated with being unique to *B. pertussis*

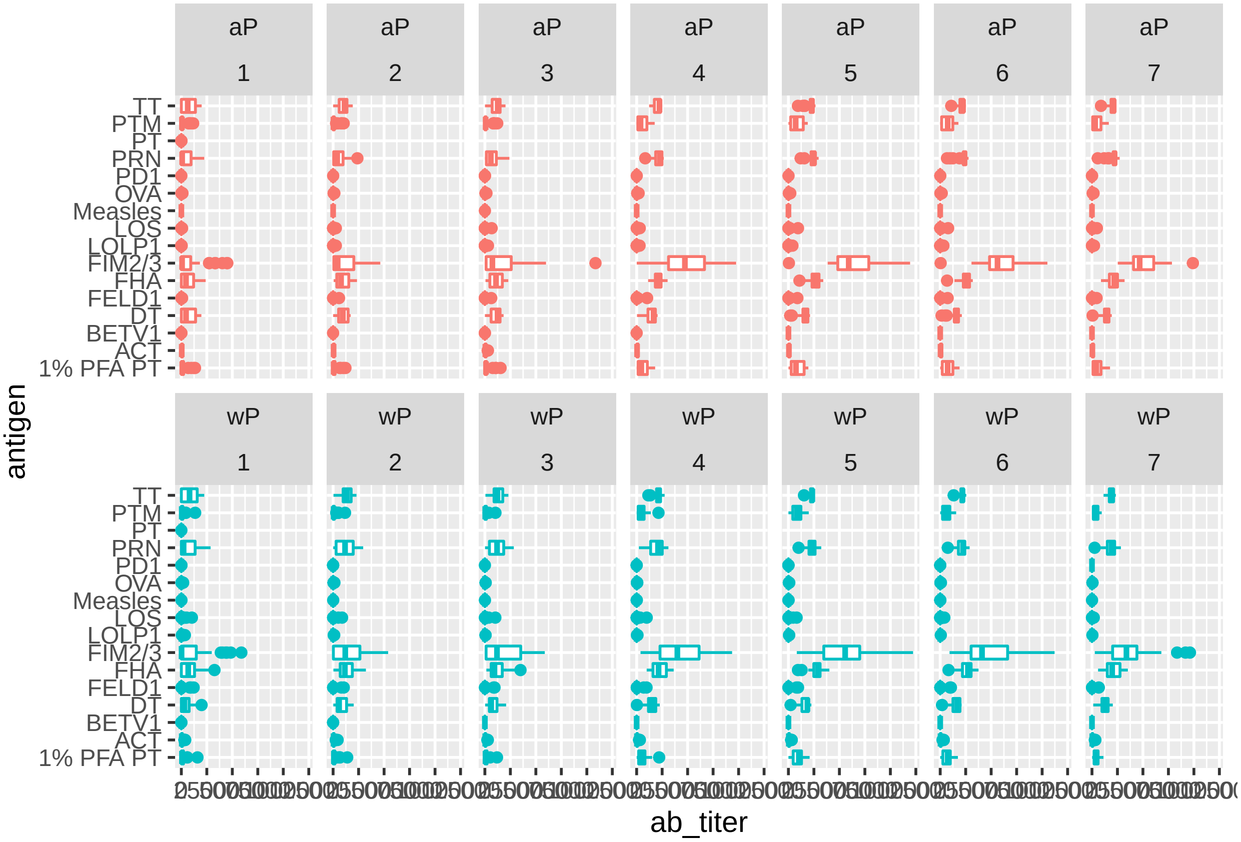
We can attempt to examine differences between wP and aP here by setting color and/or facet values of the plot to include infancy_vac status (see below). However these plots tend to be rather busy and thus hard to interpret easily.

```
ggplot(ig1) +
  aes(ab_titer, antigen, col=infancy_vac ) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(vars(visit), nrow=2) +
  theme_bw()
```



Another version of this plot adding infancy_vac to the faceting:

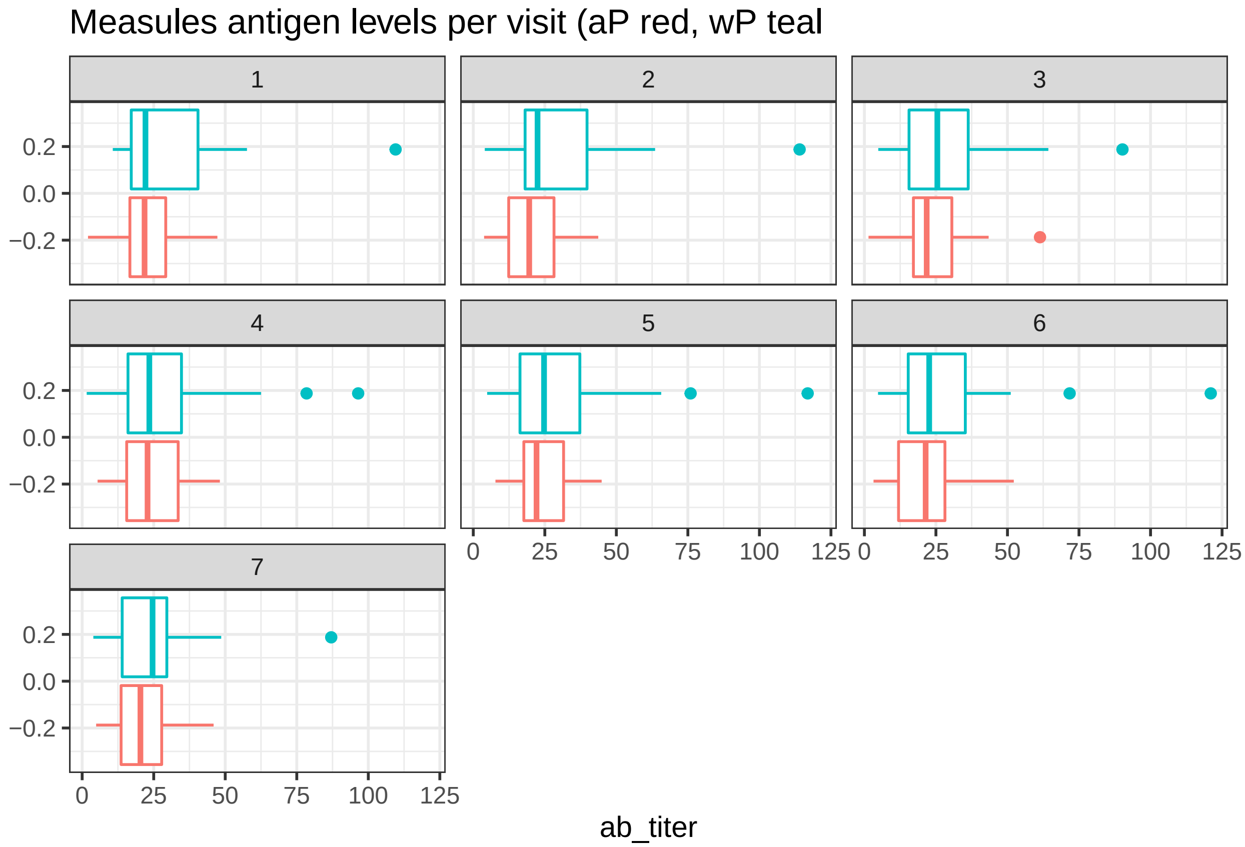
```
ggplot(ig1) +
  aes(ab_titer, antigen, col=infancy_vac ) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(vars(infancy_vac, visit), nrow=2)
```



Q15. Filter to pull out only two specific antigens for analysis and create a boxplot for each. You can chose any you like. Below I picked a "control" antigen ("Measles", that is not in our vaccines) and a clear antigen of interest ("FIM2/3", extra-cellular fimbriae proteins from B. pertussis that participate in substrate attachment).

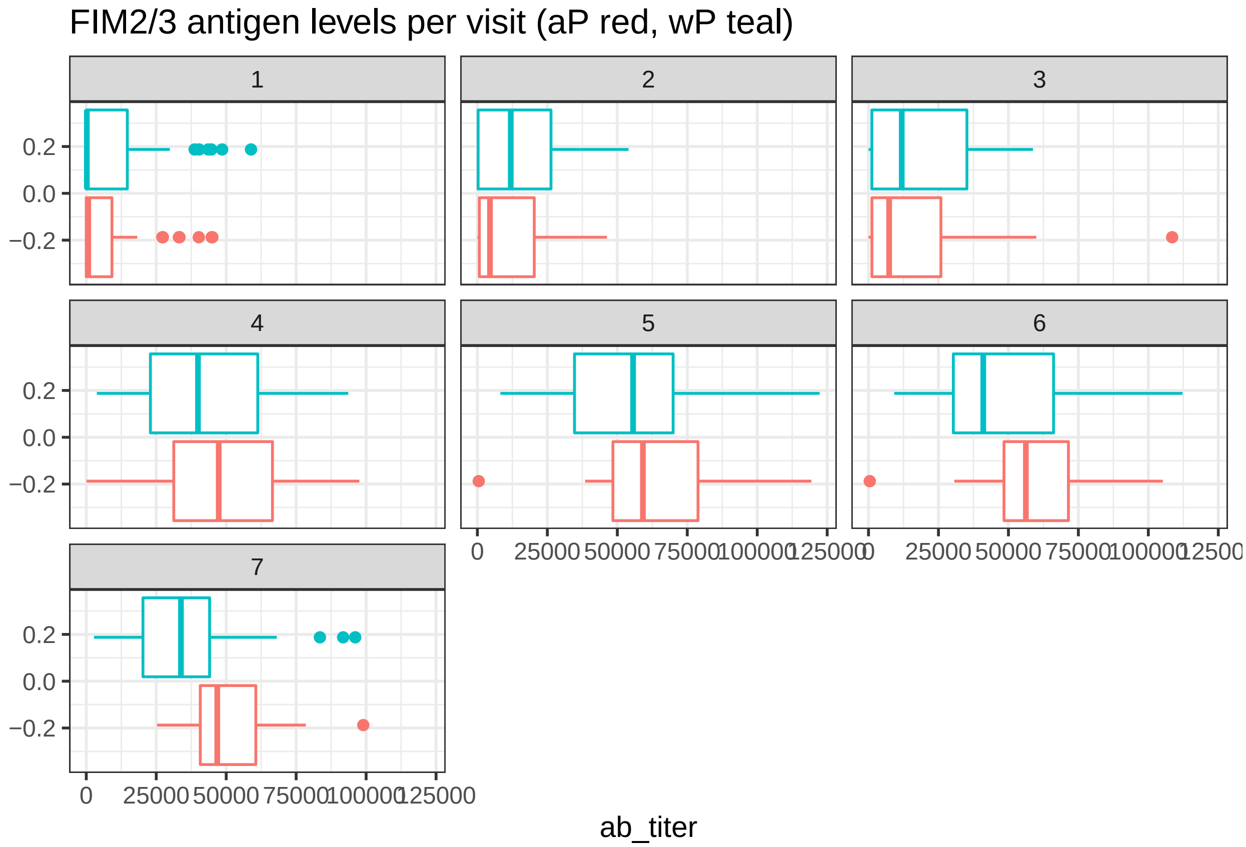
```r
# For the measles control antigens
filter(ig1, antigen=="Measles") %>%
  ggplot() +
  aes(ab_titer, col=infancy_vac) +
  geom_boxplot(show.legend = "none") +
  facet_wrap(vars(visit)) +
  theme_bw() +
  labs(title="Measules antigen levels per visit (aP red, wP teal")
```

```
## Warning: 'show.legend' must be a logical vector.
```



```
# For the FIM2/3 antigen
filter(ig1, antigen=="FIM2/3") %>%
  ggplot() +
  aes(ab_titer, col=infancy_vac) +
  geom_boxplot(show.legend = "none") +
  facet_wrap(vars(visit)) +
  theme_bw() +
  labs(title="FIM2/3 antigen levels per visit (aP red, wP teal)")
```

```
## Warning: 'show.legend' must be a logical vector.
```



Q16. What do you notice about these two antigens time course and the FIM2/3 data in particular?

We see that FIM2/3 levels are clearly rise over time. In fact the rising FIM2/3 levels far exceed those found in Measles. The FIM2/3 levels also appear to peak at visit 5 and then decline afterward. We note that This trend appears similar for both wP and aP subjects.

Q17. Do you see any clear difference in aP vs. wP responses?

in this case, we do not see clear differences in the aP and wP responses, for most of the visits the, spread of the data between the both aP and wP generally fall between the same range. However on the 7th visit, the aP response appear to have higher titer than thatt of the wP responses, as a majority of the spread of the aP response data on the 7th visit falls above the majority of the spread of wP response data for the 7th visit.

5. Obtaining CMI-PB RNA-Seq data

For RNA-Seq data the API query mechanism quickly hits the web browser interface limit for file size. We will present alternative download mechanisms for larger CMI-PB datasets in the next section. However, we can still do "targeted" RNA-Seq querys via the web accessible API.

Let's read available RNA-Seq data for this gene into R and investigate the time course of it's gene expression values.

```
url <- "https://www.cmi-pb.org/api/v2/rnaseq?versioned_ensembl_gene_id=eq.ENSG00000211896.7"
rna <- read_json(url, simplifyVector = TRUE)
```

To facilitate further analysis we need to "join" the rna expression data with our metadata meta, which is itself a join of sample and specimen data. This will allow us to look at this genes TPM expression values over aP/wP status and at different visits (i.e. times):

```
meta <- inner_join(specimen, subject)
```

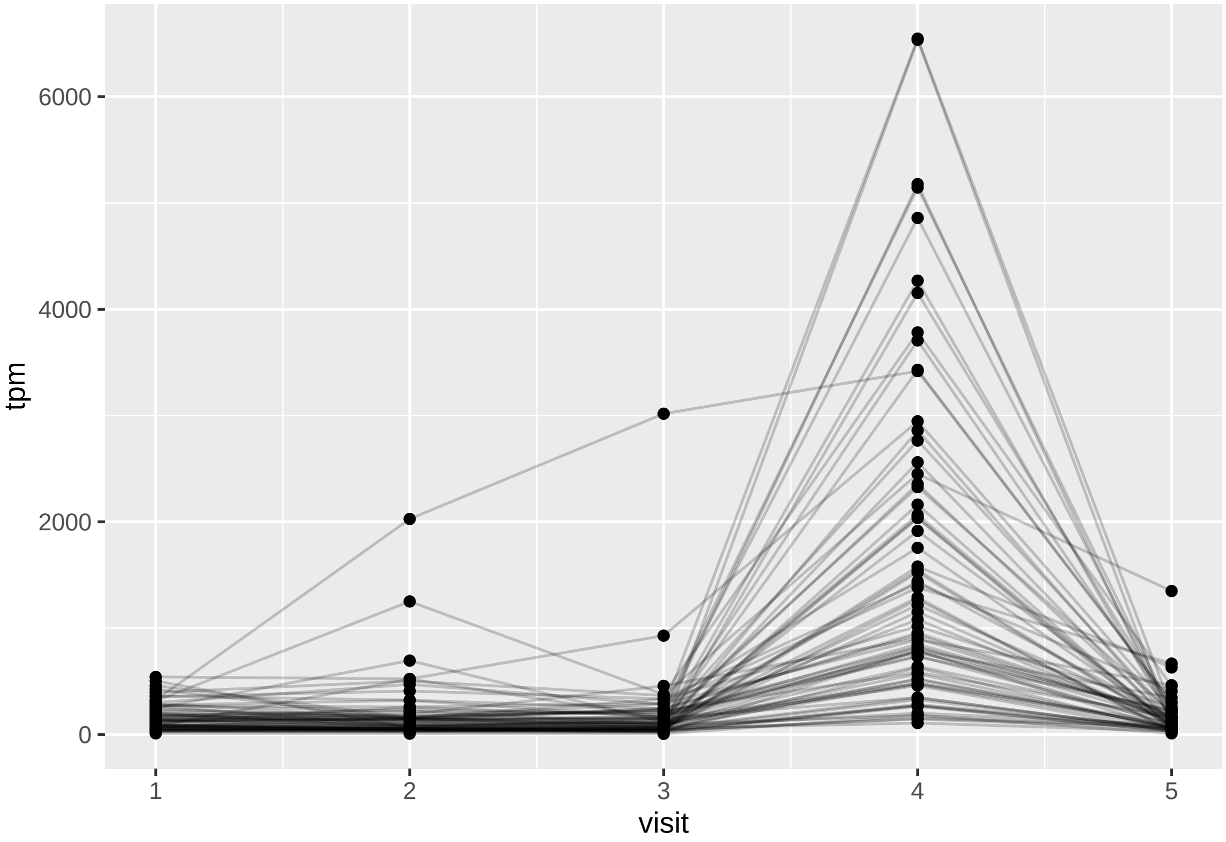
```
## Joining, by = "subject_id"
```

```
ssrna <- inner_join(rna, meta)
```

```
## Joining, by = "specimen_id"
```

> **Q18. Make a plot of the time course of gene expression for IGHG1 gene (i.e. a plot of visit vs. tpm).**

```
ggplot(ssrna) +
  aes(visit, tpm, group=subject_id) +
  geom_point() +
  geom_line(alpha=0.2)
```



Q19. What do you notice about the expression of this gene (i.e. when is it at it's maximum level)?

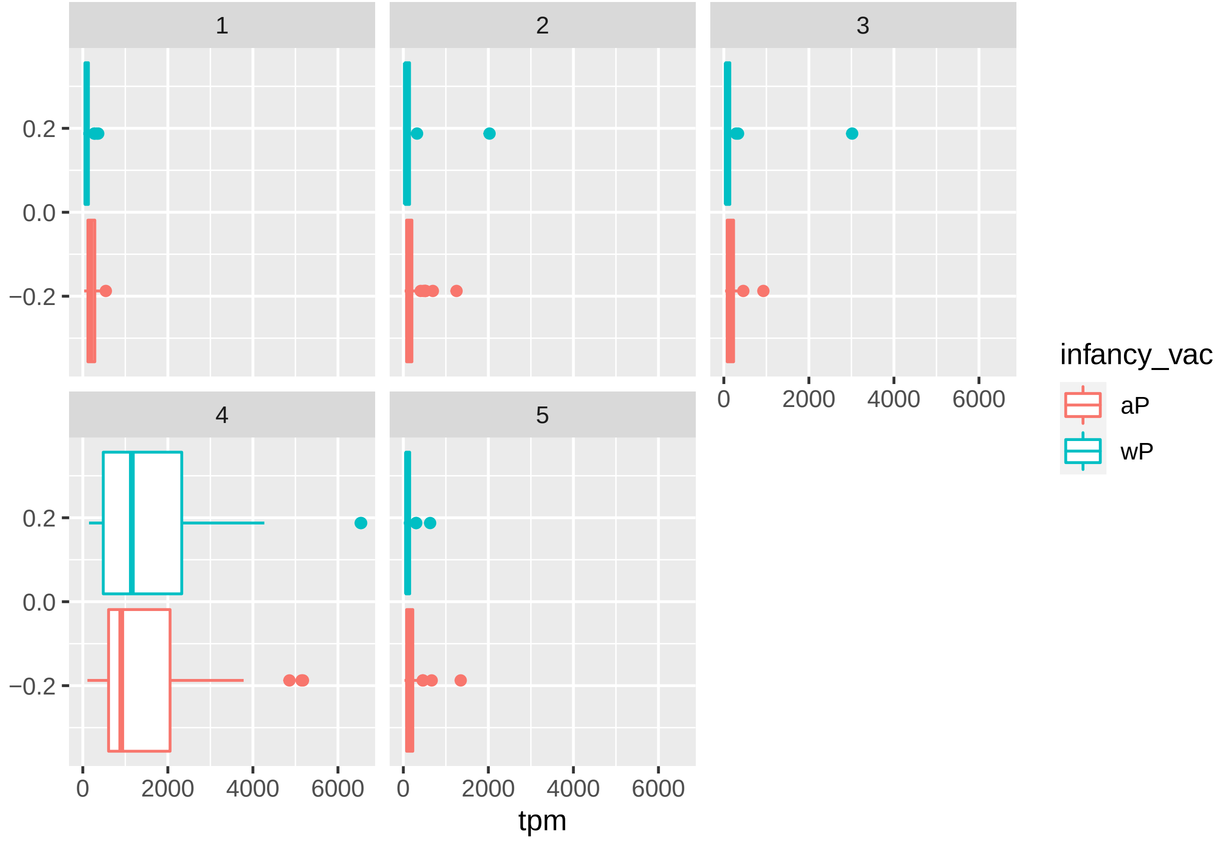
We see that the maximum expression of the IGHG1 gene occurs around the 4th visit.

Q20. Does this pattern in time match the trend of antibody titer data? If not, why not?

The time when maximum expression of the IGHG1 gene occurs is approximately the same time as when the FIM2/3 antigen levels begin to rise significantly as seen in the previous box plot. This makes sense as an increase inf the FIM2/3 antigen levels will require the immune system to generate express more IGHG1 to antagonize the antigen. Additionally, we see that the expression of the IGHG1 gene appear to decrease sharply after reaching a maximum. This is also in line with what is known about the immune system. Immune cells make long lived antibodies, so after generating them in high titers to antagonize the rising levels of the FIM2/3 antigen, there is no need to continue to maximize gene expression of the IGHG1 gene, as the antibodies generated are long lived and will continue to antagonize the FIM2/3 anigens.

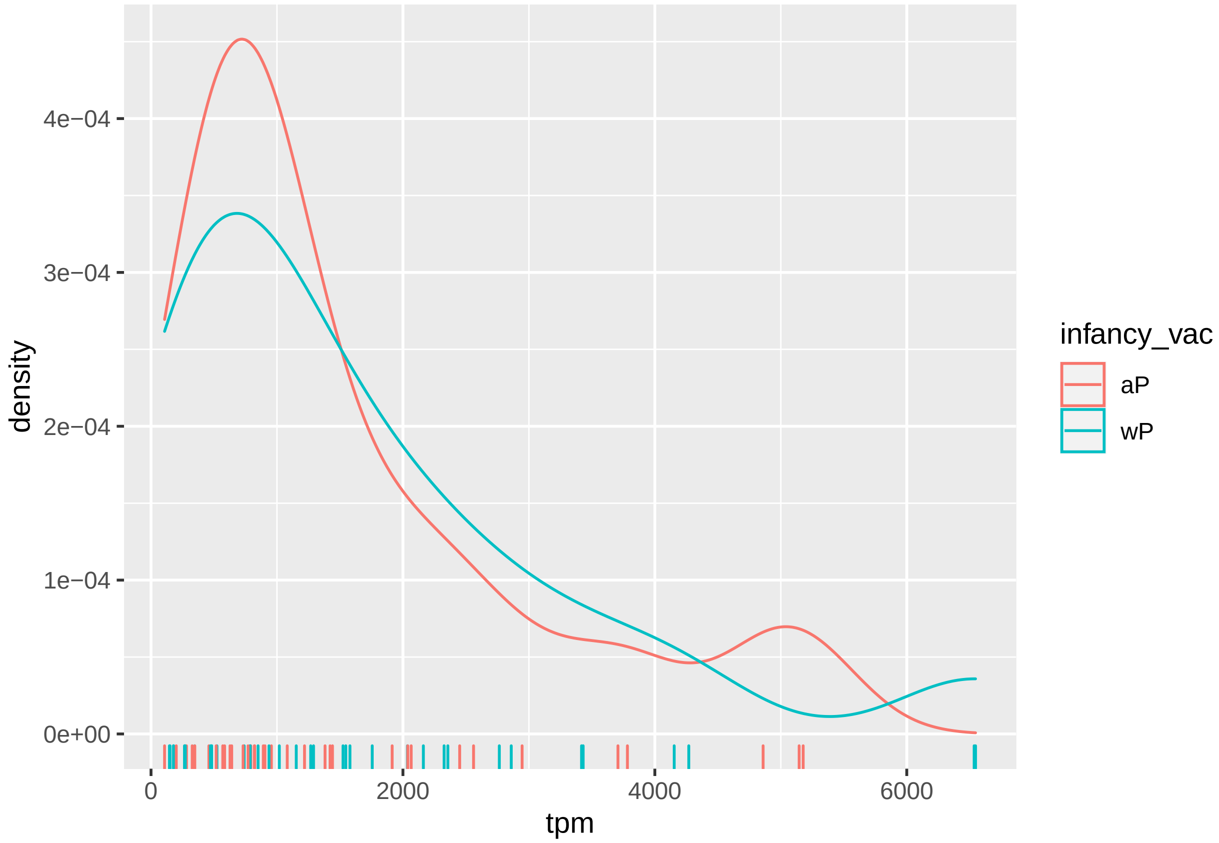
We can dig deeper and color and/or facet by infancy_vac status:

```
ggplot(ssrna) +
  aes(tpm, col=infancy_vac) +
  geom_boxplot() +
  facet_wrap(vars(visit))
```



There is no obvious wP vs. aP differences here even if we focus in on a particular visit:

```r
ssrna %>%
  filter(visit==4) %>%
  ggplot() +
    aes(tpm, col=infancy_vac) + geom_density() +
    geom_rug()
```



6. Working with larger datasets [OPTIONAL]

As API based RNA-Seq querys can quickly become large requests, CMI-PB makes CSV files available for download. We will take their “2020 longitudinal RNA-Seq data” file (named 2020LD_rnaseq.csv).

```r
# Change for your downloaded file path
setwd("c:/users/caleb/downloads")
rnaseq <- read.csv("2020LD_rnaseq.csv")
head(rnaseq,3)
```

```
##   versioned_ensembl_gene_id specimen_id raw_count tpm
## 1        ENSG00000229704.1         209         0   0
## 2        ENSG00000229707.1         209         0   0
## 3        ENSG00000229708.1         209         0   0
```

This is rather large, look at the number of rows we have here:

```
dim(rnaseq)
```

```
## [1] 10502460        4
```

Working with long format data

Note that our rnadata is in so-called “long” format rather than the more conventional “wide” format tables of expression data that we have worked with in the past (where rows represent genes and different columns represent counts in different experiments with a column per experiment). Here we have genes in the rows but have we counts for all experiments collated together in one column. Along with this we have our now familiar specimen_id column to tell us which experiment the values come from.

Lets have a look at how many genes we have reported for each specimen_id with the table() function.

```
n_genes <- table(rnaseq$specimen_id)
head( n_genes , 10)
```

```
## 
##     1     3     4     5     6    19    20    21    22    23 
## 58347 58347 58347 58347 58347 58347 58347 58347 58347 58347
```

We now seek to answer many specimens there are.

```
length(n_genes)
```

```
## [1] 180
```

We now check if there are the same number of genes for each specimen.

```
all(n_genes[1]==n_genes)
```

```
## [1] TRUE
```

Convert to “wide format”

All looks good at this stage so we convert to wider format with the pivot_wider() function from the tidyr package:

```
# Call the tidyr package
library(tidyr)
# Convert to wider format
rna_wide <- rnaseq %>%
  select(versioned_ensembl_gene_id, specimen_id, tpm) %>%
  pivot_wider(names_from = specimen_id, values_from=tpm)
dim(rna_wide)
```

```
## [1] 58347   181
```

```
head(rna_wide[,1:7], 3)
```

```
## # A tibble: 3 x 7
##   versioned_ensembl_gene_id ‘209‘  ‘74‘ ‘160‘  ‘81‘ ‘102‘ ‘163‘
##   <chr>                     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ENSG00000229704.1             0     0     0     0     0     0
## 2 ENSG00000229707.1             0     0     0     0     0     0
## 3 ENSG00000229708.1             0     0     0     0     0     0
```

Filtering

As usual we have lots of zero count genes that can be removed (i.e. filtered out) before further analysis. Generally we will want to remove genes which are expressed at low levels or show only small changes in expression.

Key remaining questions

Once you have got this far we can begin to investigate the relationship between mRNA levels on different days (e.g. visit 1 vs. visit 4 etc.) and start to address the next set of pressing questions:

1. Is RNA-Seq expression levels predictive of Ab titers?
2. What differentiates aP vs. wP primed individuals?
3. What about decades after their first immunization? Do you know?

These are all ares of active research. What is clear is that there are immune responses not captured in antibody titers. We are trying to capture the ‘systems level’ response to antigen encounters (here: vaccination). Something must be different in aP vs. wP primed individuals. That is why CMI-PB make all their data available for the wider community to explore and contribute new analysis methods.

About this document

Here we use the sessionInfo() function to report on our R systems setup at the time of document execution.

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
```

```
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] tidyr_1.2.0     dplyr_1.0.8     lubridate_1.8.0 jsonlite_1.8.0
## [5] ggplot2_3.3.5
##
## loaded via a namespace (and not attached):
##  [1] highr_0.9        pillar_1.7.0     compiler_4.1.2   tools_4.1.2
##  [5] digest_0.6.29    evaluate_0.14    lifecycle_1.0.1  tibble_3.1.6
##  [9] gtable_0.3.0     pkgconfig_2.0.3  rlang_1.0.1      cli_3.1.1
## [13] DBI_1.1.2        rstudioapi_0.13  yaml_2.2.2       xfun_0.29
## [17] fastmap_1.1.0    withr_2.4.3      stringr_1.4.0    knitr_1.37
## [21] generics_0.1.2   vctrs_0.3.8      grid_4.1.2       tidyselect_1.1.1
## [25] glue_1.6.1       R6_2.5.1         fansi_1.0.2      rmarkdown_2.11
## [29] farver_2.1.0     purrr_0.3.4      magrittr_2.0.2   scales_1.1.1
## [33] ellipsis_0.3.2   htmltools_0.5.2  assertthat_0.2.1 colorspace_2.0-2
## [37] labeling_0.4.2   utf8_1.2.2       stringi_1.7.6    munsell_0.5.0
## [41] crayon_1.5.0
```