# Class19

Job Rocha. PID: 59023124

## Pertussis and the CMI-PB project.

**Educational material for the CMI-PB project.**

## Background

Pertussis (more commonly known as whooping cough) is a highly contagious respiratory disease caused by the bacterium Bordetella pertussis (Figure 1). People of all ages can be infected leading to violent coughing fits followed by a characteristic high-pitched "whoop" like intake of breath. Children have the highest risk for severe complications and death. Recent estimates from the WHO indicate that ~16 million cases and 200,000 infant deaths are due to pertussis annually (Black et al. 2010).

## 1. Investigating pertussis cases by year.

The United States Centers for Disease Control and Prevention (CDC) has been compiling reported pertussis case numbers since 1922 in their National Notifiable Diseases Surveillance System (NNDSS). We can view this data on the CDC website here: https://www.cdc.gov/pertussis/surv-reporting/cases-by-year.html

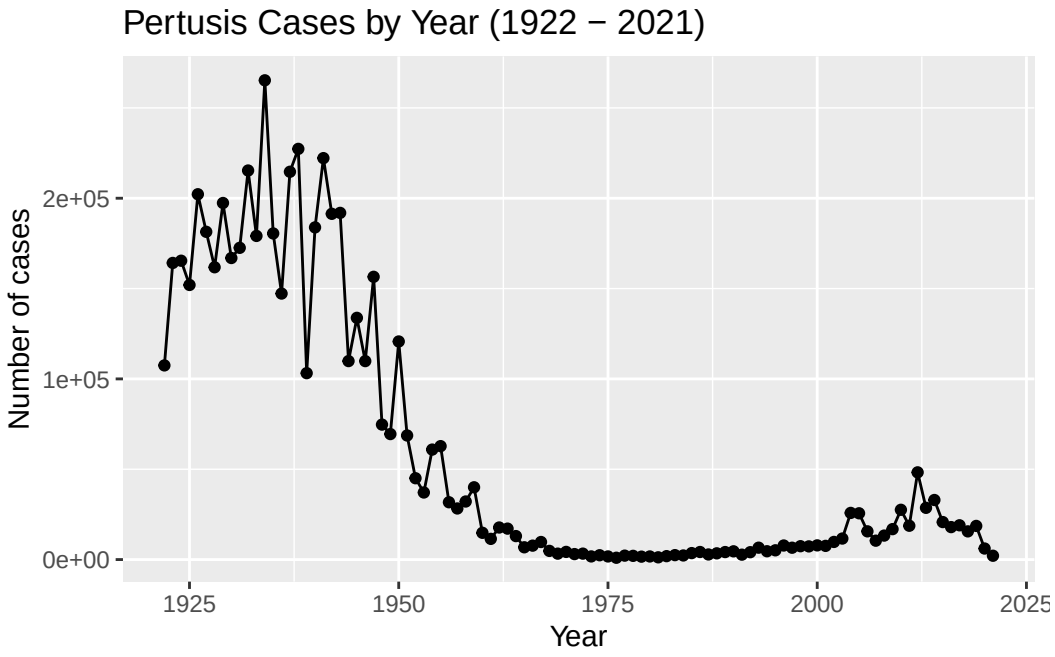
**Q1. With the help of the R "addin" package datapasta assign the CDC pertussis case number data to a data frame called cdc and use ggplot to make a plot of cases numbers over time.**

```
#install.packages("datapasta")
library(ggplot2)
library(datapasta)
```

```r
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,2020L,2021L),
  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,
                                     32971,20762,17972,18975,15609,18617,6124,
                                     2116)
)
```

```r
ggplot(cdc) +
  aes(Year, No..Reported.Pertussis.Cases) +
  geom_point() +
  geom_line() +
```

```
labs(y="Number of cases", x = "Year", title="Pertusis Cases by Year (1922 - 2021)")
```




Key point: Pertussis vaccination is, in general, highly effective at preventing the disease. In the pre-vaccine era (before 1946) pertussis was a much more common disease and a major cause of infant mortality 2. As we see clearly from analysis of the CDC tracking data above, introduction of the first pertussis vaccination in the United States in 1946 resulted in a dramatic reduction in the number of yearly cases from $>$ 200,000 in the 1940s to $<$ 2,000 in the 1970s.

## 2. A tale of two vaccines (wP & aP).

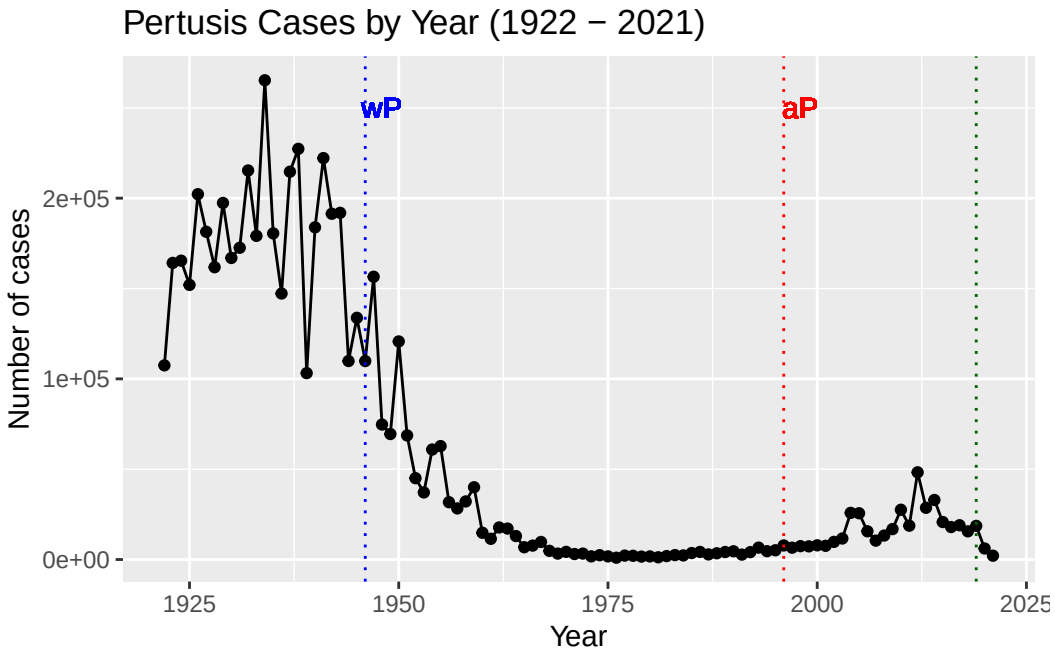
Two types of pertussis vaccines have been developed: whole-cell pertussis (wP) and acellular pertussis (aP). The first vaccines were composed of 'whole cell' (wP) inactivated bacteria. The latter aP vaccines use purified antigens of the bacteria (the most important pertussis components for our immune system, see Figure 2). These aP vaccines were developed to have less side effects than the older wP vaccines and are now the only form administered in the United States.

For motivated readers there is a nice historical account of the wP to aP vaccine switch in the US in Klein (2014) (Klein 2014)

Let's return to our CDC data plot and examine what happened after the switch to the acellular pertussis (aP) vaccination program.

**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?** ***Cases started to drop overtime after wP, but increased again after year 2000.***

```
ggplot(cdc) +
  aes(Year, No..Reported.Pertussis.Cases) +
  geom_point() +
  geom_line() +
  labs(y="Number of cases", x = "Year", title="Pertusis Cases by Year (1922 - 2021)") +
  geom_vline(xintercept = 1946, color="blue", linetype=3) +
  geom_vline(xintercept = 1996, color="red", linetype=3) +
  geom_vline(xintercept = 2019, color="darkgreen", linetype=3) +
  geom_text(aes(label="wP", x=1948, y = 250000), color="blue") + geom_text(aes(label="aP",
```



**Q3. Describe what happened after the introduction of the aP vaccine? Do you have a possible explanation for the observed trend?** ***After the introduction of aP cases started to slightly increase again. It could be either increase test sensitivity or reduction in vaccine rates.***

Additional points for discussion: How are vaccines currently approved?

1) Typically we first examine 'Correlates of protection' which are things that can be measured within weeks or months after vaccination, and which are thought to correlate with increased protection from disease. For the aP vaccine this was an induction of antibodies against pertussis toxin (PT) in infants at equivalent levels to those induced by the wP vaccine. The aP vaccines also had less side effects (reduction of sore arms, fever and pain).
2) Testing for protection induced by a new vaccine requires a lot of people exposed to the pathogen (like in a pandemic).
3) It is impossible to discover a effect 10 years post vaccination in the current trial system.
4) It is unclear what differentiates people that have been primed with aP vs. wP long term.
5) The CMI-PB project is an attempt to make data on this question open and examinable by all.

## 3. Exploring CMI-PB data.

Why is this vaccine-preventable disease on the upswing? To answer this question we need to investigate the mechanisms underlying waning protection against pertussis. This requires evaluation of pertussis-specific immune responses over time in wP and aP vaccinated individuals.

The new and ongoing CMI-PB project aims to provide the scientific community with this very information. In particular, CMI-PB tracks and makes freely available long-term humoral and cellular immune response data for a large number of individuals who received either DTwP or DTaP combination vaccines in infancy followed by Tdap booster vaccinations. This includes complete API access to longitudinal RNA-Seq, AB Titer, Olink, and live cell assay results directly from their website: https://www.cmi-pb.org/

### 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. An example of the JSON format for Ab titer data is shown below:

{ "specimen_id":1, "isotype":"IgG", "is_antigen_specific":true, "antigen":"PT", "ab_titer":68.5661390514946, "unit":"IU/ML", "lower_limit_of_detection":0.53 }

To read these types of files into R we will use the read_json() function from the jsonlite package. Note that if you want to do more advanced querys of APIs directly from R you will likely want to explore the more full featured rjson 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.

```
# 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. You can find out more about the content and format of this and other tables here: https://www.cmi-pb.org/blog/understand-data/

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

```
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      dataset
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 many aP and wP infancy vaccinated subjects are in the dataset?** ***60 aP and 58 wP.***

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

```
aP wP
60 58
```

**Q5. How many Male and Female subjects/patients are in the dataset?** ***79 Females and 39 Males.***

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

```
Female   Male
    79     39
```

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

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

```
                                        Female Male
  American Indian/Alaska Native              0    1
  Asian                                     21   11
  Black or African American                  2    0
  More Than One Race                         9    2
  Native Hawaiian or Other Pacific Islander  1    1
  Unknown or Not Reported                   11    4
  White                                     35   20
```

## 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:

```
#install.packages("lubridate")
library(lubridate)
```

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

```
The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

What is today's date (at the time I am writing this obviously)

```
today()
```

```
[1] "2023-12-07"
```

How many days have passed since new year 2000

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

```
Time difference of 8741 days
```

What is this in years?

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

```
[1] 23.93155
```

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?**

```
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
```

```
subject$age <- (today() - ymd(subject$year_of_birth))

ap <- subject %>% filter(infancy_vac == "aP")

print(paste("Mean age of aP: ", round( mean( time_length( ap$age, "years" ) ) ) ))
```

```
[1] "Mean age of aP:  26"
```

```
# wP
wp <- subject %>% filter(infancy_vac == "wP")
print(paste("Mean age of wP: ", round( mean( time_length( wp$age, "years" ) ) ) ))
```

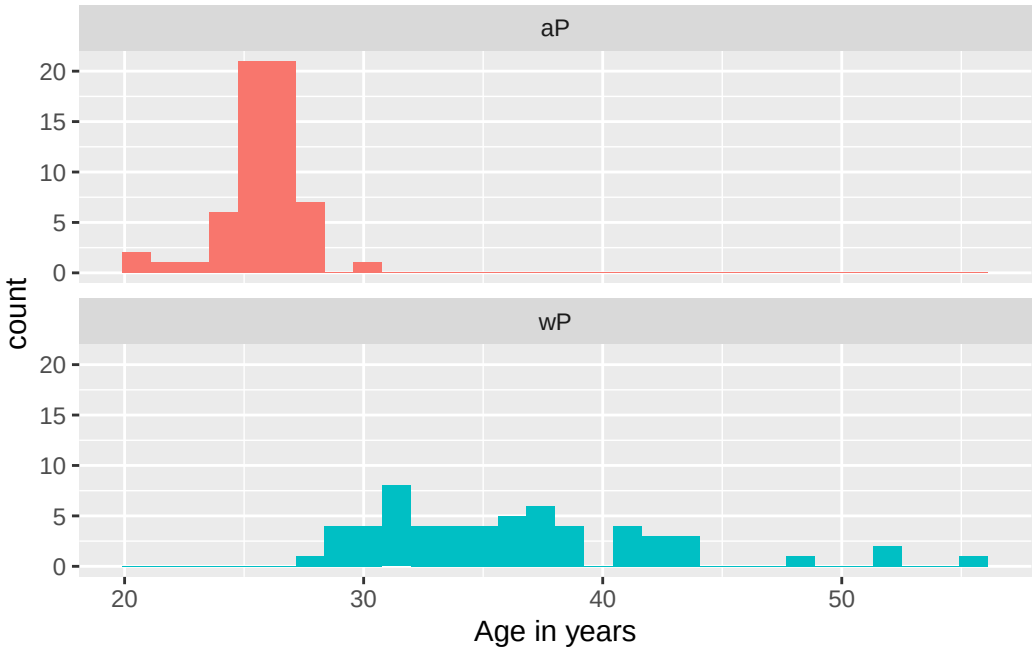
```
[1] "Mean age of wP:  36"
```

```
int <- ymd(subject$date_of_boost) - ymd(subject$year_of_birth)
age_at_boost <- time_length(int, "year")
head(age_at_boost)
```

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

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

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




### Joining multiple tables.

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

```
# 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/plasma_ab_titer", simplifyVector = TRUE)
```

To know whether a given specimen_id comes from an aP or wP individual we need to link (a.k.a. "join" or merge) our specimen and subject data frames. The excellent dplyr package (that we have used previously) 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 with `by = join_by(subject_id)`
```

```
dim(meta)
```

```
[1] 939  14
```

```
head(meta)
```

```
  specimen_id subject_id actual_day_relative_to_boost
1           1          1                           -3
2           2          1                            1
3           3          1                            3
4           4          1                            7
5           5          1                           11
6           6          1                           32
  planned_day_relative_to_boost specimen_type visit infancy_vac biological_sex
1                             0         Blood     1          wP         Female
2                             1         Blood     2          wP         Female
3                             3         Blood     3          wP         Female
4                             7         Blood     4          wP         Female
5                            14         Blood     5          wP         Female
6                            30         Blood     6          wP         Female
              ethnicity  race year_of_birth date_of_boost      dataset
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 13854 days
2 13854 days
3 13854 days
4 13854 days
5 13854 days
6 13854 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 with `by = join_by(specimen_id)`
```

```
dim(abdata)
```

```
[1] 41810    21
```

```
#abdata$antigen[abdata$antigen == "Fim2/3"] <- "FIM2/3"
```

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

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

```
 IgE  IgG IgG1 IgG2 IgG3 IgG4
6698 3240 7968 7968 7968 7968
```

Q12. What are the different $dataset values in abdata and what do you notice about the number of rows for the most "recent" dataset?

```
table(abdata$dataset)
```

```
2020_dataset 2021_dataset 2022_dataset
       31520         8085         2205
```

## 4. Examine IgG Ab titer levels.

Now using our joined/merged/linked abdata dataset filter() for IgG isotype.

```
igg <- abdata %>% filter(isotype == "IgG")
head(igg)
```

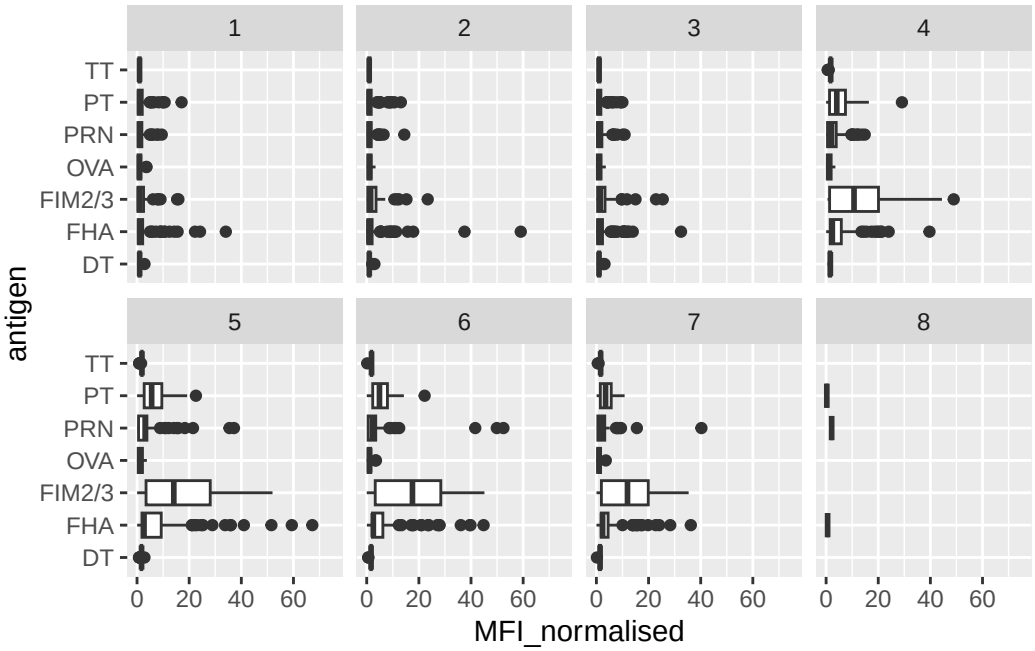
```
  specimen_id isotype is_antigen_specific antigen        MFI MFI_normalised
1           1     IgG                TRUE      PT   68.56614       3.736992
2           1     IgG                TRUE     PRN  332.12718       2.602350
3           1     IgG                TRUE     FHA 1887.12263      34.050956
4          19     IgG                TRUE      PT   20.11607       1.096366
```

```
5          19     IgG                TRUE     PRN  976.67419       7.652635
6          19     IgG                TRUE     FHA   60.76626       1.096457
   unit lower_limit_of_detection subject_id actual_day_relative_to_boost
1 IU/ML                 0.530000          1                           -3
2 IU/ML                 6.205949          1                           -3
3 IU/ML                 4.679535          1                           -3
4 IU/ML                 0.530000          3                           -3
5 IU/ML                 6.205949          3                           -3
6 IU/ML                 4.679535          3                           -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      dataset
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                Unknown White    1983-01-01    2016-10-10 2020_dataset
5                Unknown White    1983-01-01    2016-10-10 2020_dataset
6                Unknown White    1983-01-01    2016-10-10 2020_dataset
         age
1 13854 days
2 13854 days
3 13854 days
4 14950 days
5 14950 days
6 14950 days
```

Q13. Complete the following code to make a summary boxplot of Ab titer levels (MFI) for all antigens:

```
ggplot(igg) +
  aes(MFI_normalised, antigen) +
  geom_boxplot() +
    xlim(0,75) +
  facet_wrap(vars(visit), nrow=2)
```

```
Warning: Removed 5 rows containing non-finite values (`stat_boxplot()`).
```

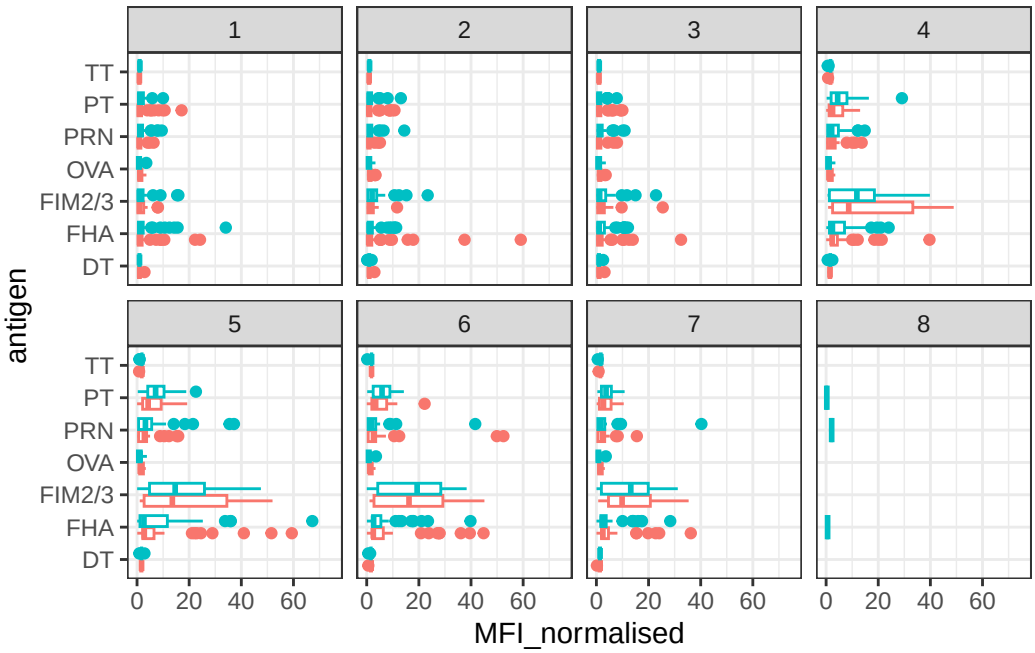



Q14. What antigens show differences in the level of IgG antibody titers recognizing them over time? Why these and not others? *PT, FIM2/3 and FHA*

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(igg) +
  aes(MFI_normalised, antigen, col=infancy_vac ) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(vars(visit), nrow=2) +
  xlim(0,75) +
  theme_bw()
```

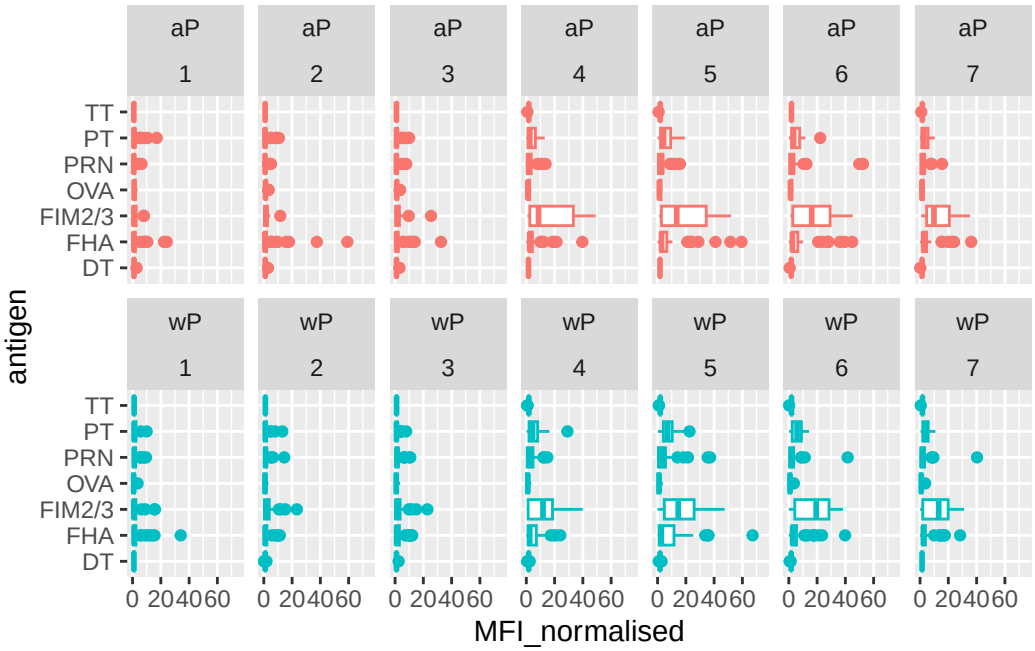
```
Warning: Removed 5 rows containing non-finite values (`stat_boxplot()`).
```




Another version of this plot adding infancy_vac to the faceting:

```
igg %>% filter(visit != 8) %>%
ggplot() +
  aes(MFI_normalised, antigen, col=infancy_vac ) +
  geom_boxplot(show.legend = FALSE) +
  xlim(0,75) +
  facet_wrap(vars(infancy_vac, visit), nrow=2)
```

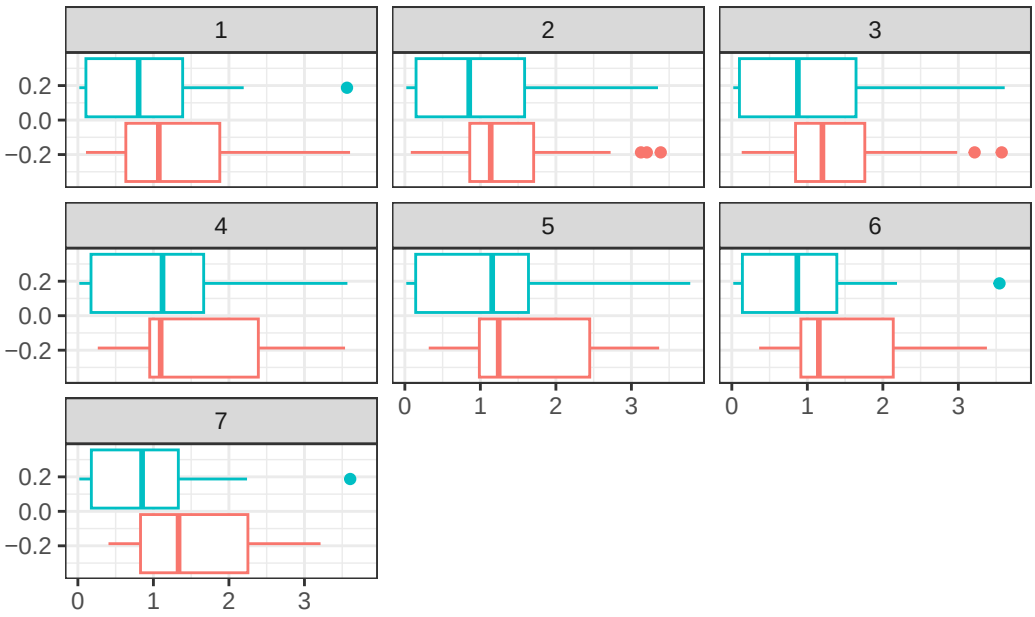
```
Warning: Removed 5 rows containing non-finite values (`stat_boxplot()`).
```




Side-note: If you don't like the overlapping x axis labels (and who would?) you can add a theme() layer where you set the text angle and horisontal adjustment relative to the axis. For example: theme(axis.text.x = element_text(angle = 45, hjust=1))

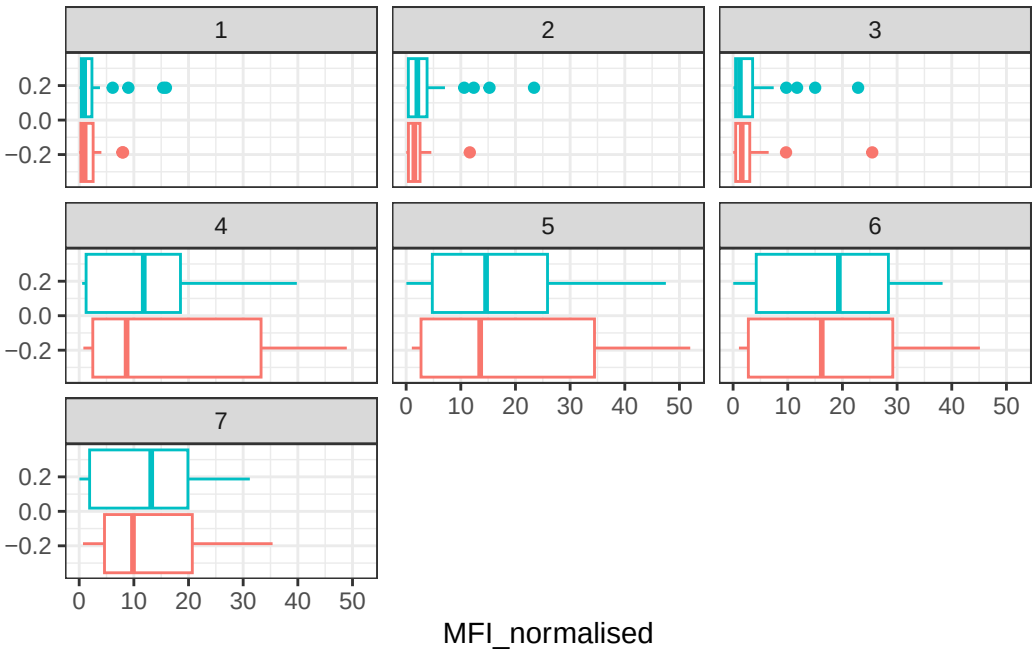
Q15. Filter to pull out only two specific antigens for analysis and create a boxplot for each. You can chose any you like. Below I picked a "control" antigen ("OVA", that is not in our vaccines) and a clear antigen of interest ("PT", Pertussis Toxin, one of the key virulence factors produced by the bacterium B. pertussis).

```
filter(igg, antigen=="OVA") %>%
  ggplot() +
  aes(MFI_normalised, col=infancy_vac) +
  geom_boxplot(show.legend = F) +
  facet_wrap(vars(visit)) +
  theme_bw()
```




and the same for antigen==“FIM2/3”

```
filter(igg, antigen=="FIM2/3") %>%
  ggplot() +
  aes(MFI_normalised, col=infancy_vac) +
  geom_boxplot(show.legend = F) +
  facet_wrap(vars(visit)) +
  theme_bw()
```



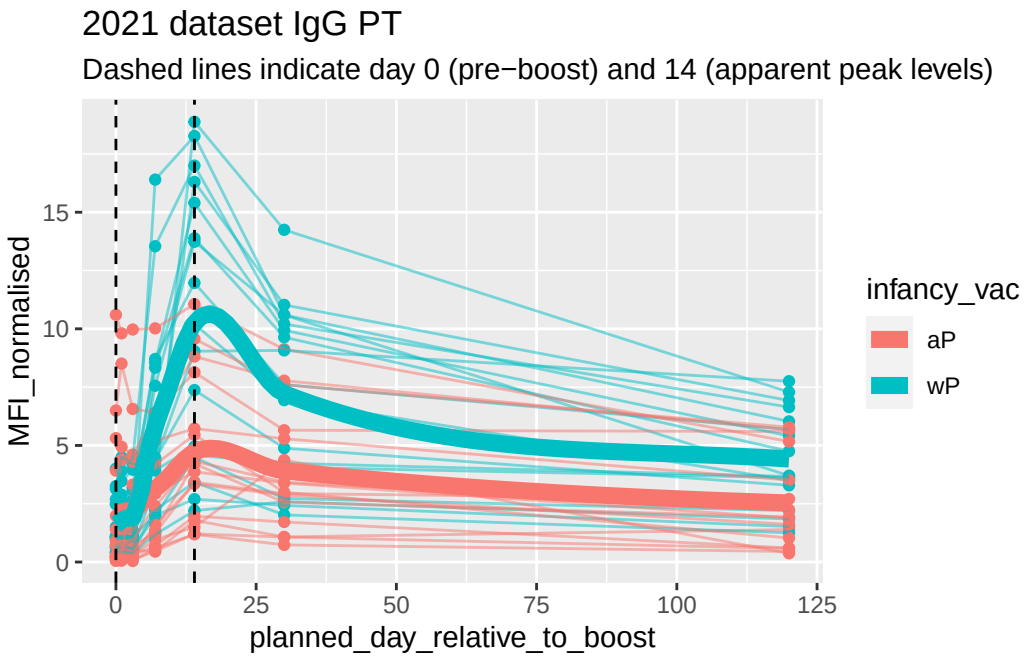

Q16. What do you notice about these two antigens time courses and the PT data in particular? *PT levels increase over time and they seem to decrease a little bit after visit number 5.*

Q17. Do you see any clear difference in aP vs. wP responses? *Not really, there are some small differences but hard to tell whether they are significant or not.*

Lets finish this section by looking at the 2021 dataset IgG PT antigen levels time-course:

```
abdata.21 <- abdata %>% filter(dataset == "2021_dataset")

abdata.21 %>%
  filter(isotype == "IgG",  antigen == "PT") %>%
  ggplot() +
    aes(x=planned_day_relative_to_boost,
        y=MFI_normalised,
        col=infancy_vac) +
    geom_point() +
    geom_line(aes(group=subject_id), alpha=0.5) +
    geom_smooth(se=F, span=0.4, linewidth=3) +
    geom_vline(xintercept=0, linetype="dashed") +
    geom_vline(xintercept=14, linetype="dashed") +
  labs(title="2021 dataset IgG PT",
       subtitle = "Dashed lines indicate day 0 (pre-boost) and 14 (apparent peak levels)")
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at -0.6

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 3.6

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 1.8382e-16

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 11364

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at -0.6

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 3.6

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 1.4316e-16

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 11364
```

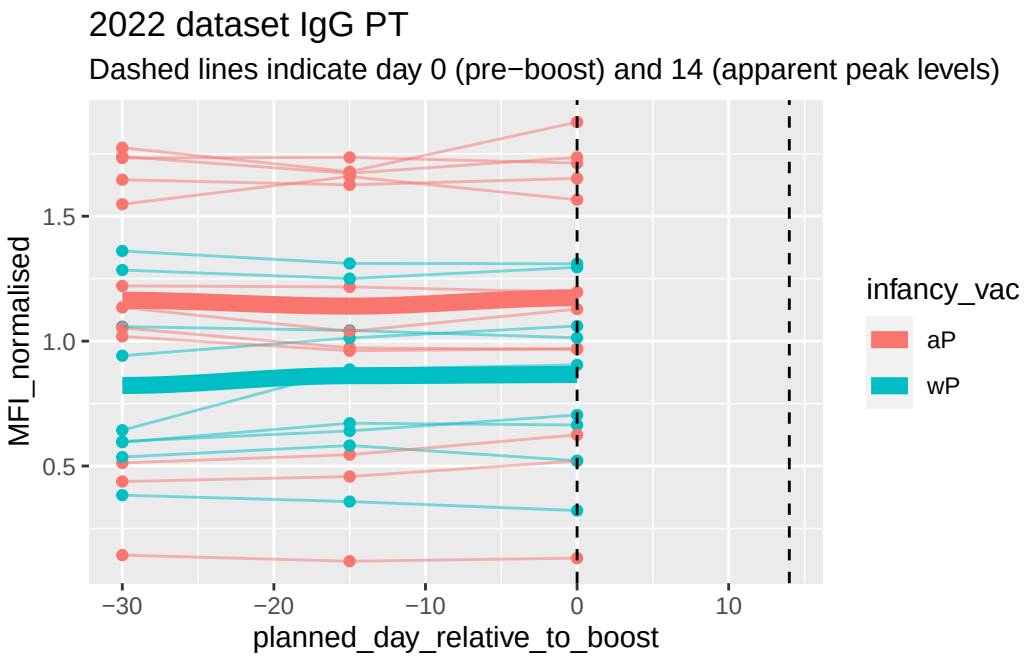



```
abdata.22 <- abdata %>% filter(dataset == "2022_dataset")

abdata.22 %>%
  filter(isotype == "IgG",  antigen == "PT") %>%
  ggplot() +
    aes(x=planned_day_relative_to_boost,
        y=MFI_normalised,
        col=infancy_vac) +
    geom_point() +
    geom_line(aes(group=subject_id), alpha=0.5) +
    geom_smooth(se=F, span=0.4, linewidth=3) +
    geom_vline(xintercept=0, linetype="dashed") +
    geom_vline(xintercept=14, linetype="dashed") +
  labs(title="2022 dataset IgG PT",
       subtitle = "Dashed lines indicate day 0 (pre-boost) and 14 (apparent peak levels)")
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at -30.15
```

```
Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 15.15

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 229.52

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: pseudoinverse used at -30.15

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: neighborhood radius 15.15

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: reciprocal condition number 0

Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,
: There are other near singularities as well. 229.52
```




Q18. Does this trend look similar for the 2020 dataset? *No, it doesn't.*

## 5. Obtaining CMI-PB RNASeq 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.

For example we can obtain RNA-Seq results for a specific ENSEMBLE gene identifier or multiple identifiers combined with the & character:

## For example use the following URL

https://www.cmi-pb.org/api/v2/rnaseq?versioned_ensembl_gene_id=eq.ENSG00000211896.7
The link above is for the key gene involved in expressing any IgG1 antibody, namely the IGHG1 gene. 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.

rna <- read_json(url, simplifyVector = TRUE)
```

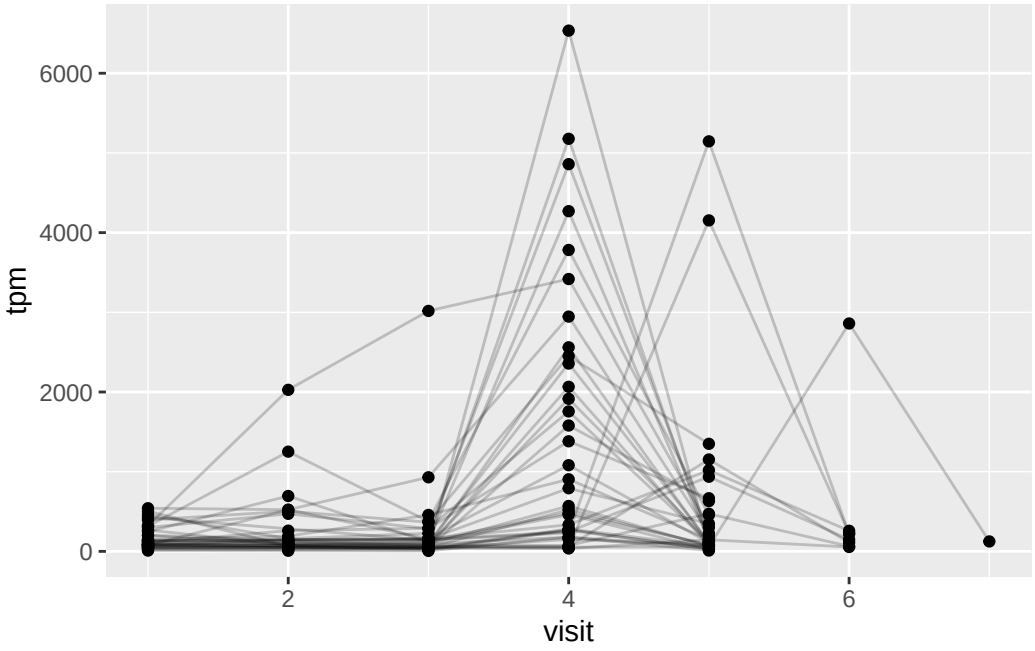
To facilitate further analysis we need to "join" the rna expression data with our metadata meta, which is itself a join of sample and specimen data. This will allow us to look at this genes TPM expression values over aP/wP status and at different visits (i.e. times):

```
#meta <- inner_join(specimen, subject)
ssrna <- inner_join(rna, meta)
```

```
Joining with `by = join_by(specimen_id)`
```

Q19. 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)
```

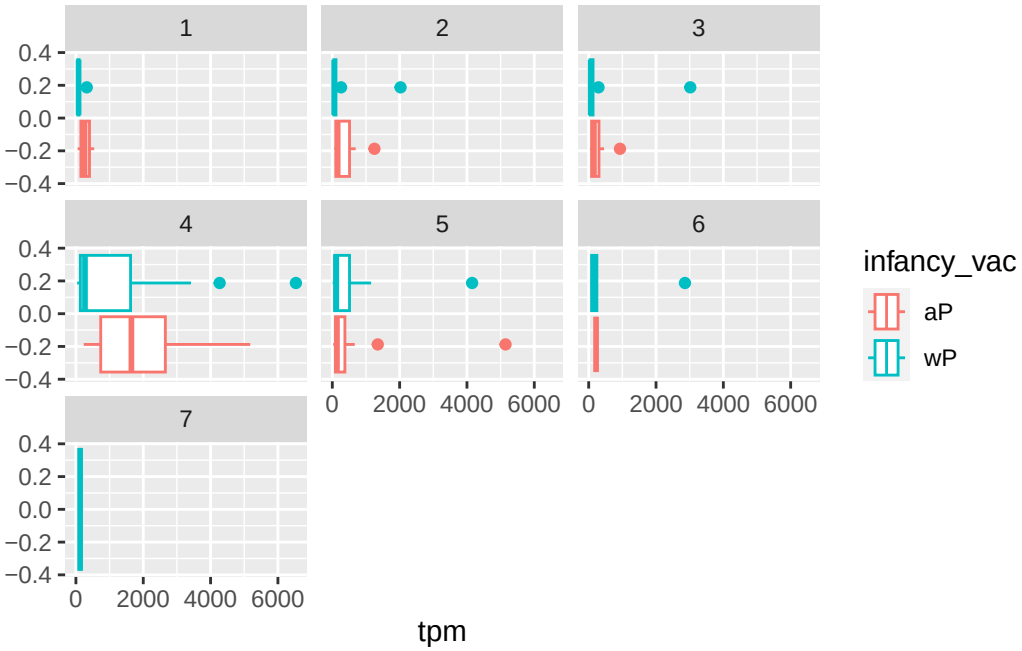



Q20.: What do you notice about the expression of this gene (i.e. when is it at it's maximum level)? *It's maximum value is at visit 4*

Q21. Does this pattern in time match the trend of antibody titer data? If not, why not? *Not really, expression turns on at visit 4 and then decreases but the proteins/ab remain in the blood.*

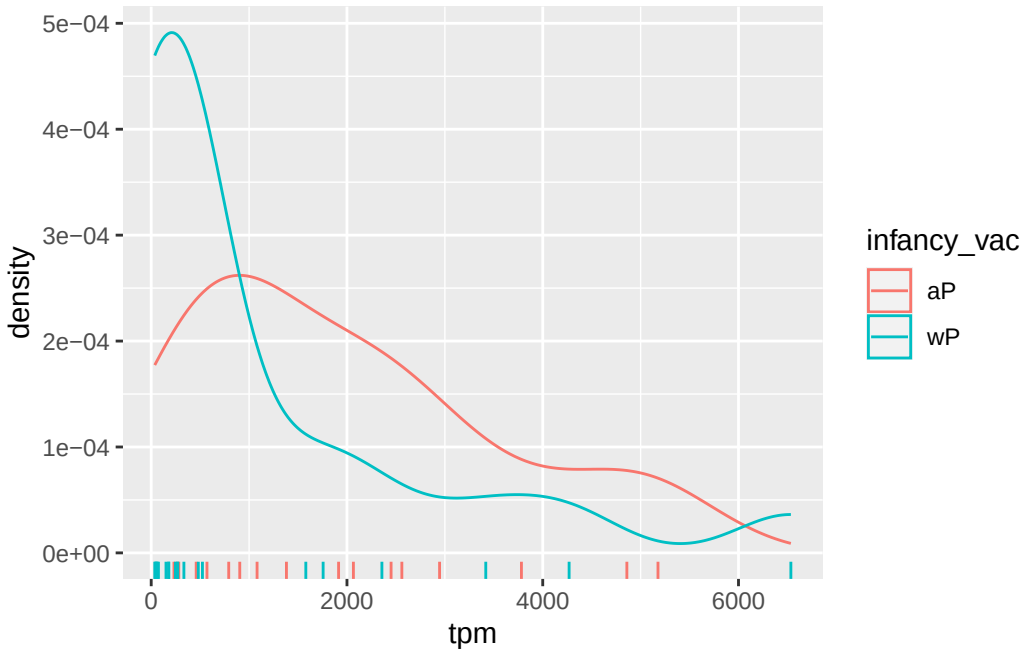
We can dig deeper and color and/or facet by infancy_vac status:

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




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

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


5e−04
4e−04
3e−04
2e−04
1e−04
0e+00
density
0
2000
4000
6000
tpm
infancy_vac
aP
wP