Class 15: Investigating Pertussis Resurgence

Mirte Ciz Marieke Kuijpers

2022-03-11

Introduction

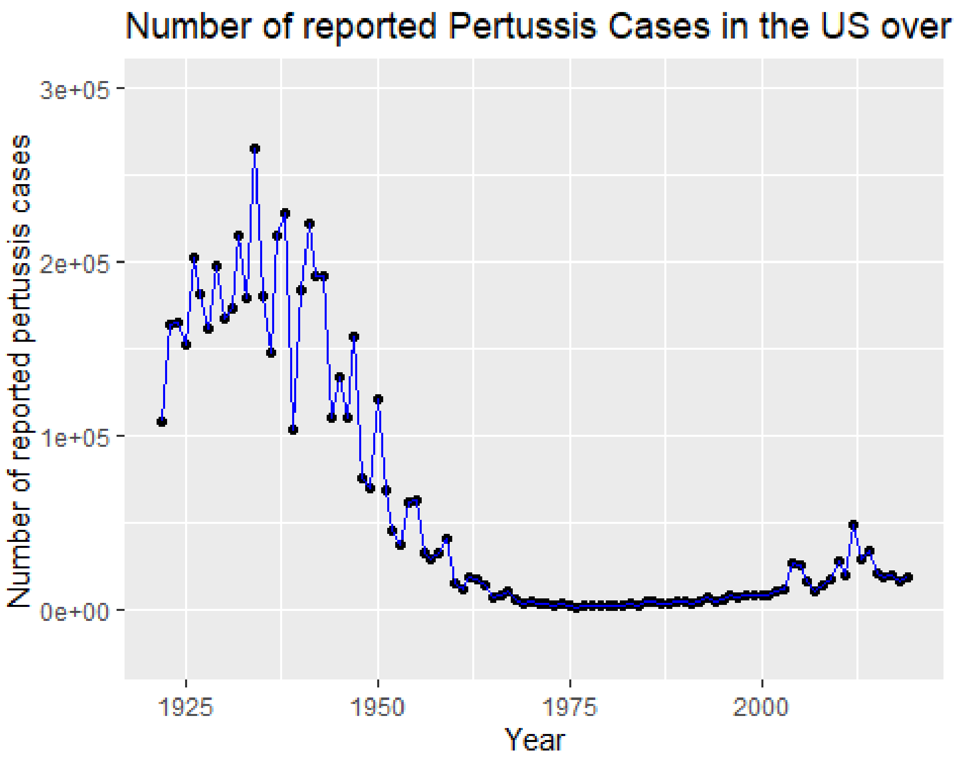
The United States Centers for Disease Control and Prevention (CDC) has been compiling reported pertussis case numbers since 1922 in their National Notifiable Diseases Surveillance System (NNDSS). This project will focus on analaysis of this data.

```
# Load packages
library("datapasta") # for easy import of copied data
library("ggplot2") # for plotting
library("jsonlite") # for reading, writing and processing JSON data
library("lubridate") # for dealing with dates
library("dplyr") # for manipulating tables
library("tidyr")
library("DESeq2") # for looking at gene expression

# Load data
dat <- read.table("pertussis.data.txt", header = TRUE)
```

Question 1

The first step that should be done is to plot the raw data to get a better idea of it.

```
# Convert reported pertussis cases to numeric
dat$No.Reported.Pertussis.Cases <- as.numeric(gsub(",", "",
dat$No.Reported.Pertussis.Cases))

# Plot raw data with ggplot2
ggplot(dat, aes(Year, No.Reported.Pertussis.Cases)) +
  geom_point() +
  labs(title = "Number of reported Pertussis Cases in the US over time", y =
"Number of reported pertussis cases") +
  geom_line(col = "blue") +
  ylim(c(-25000, 300000))
```

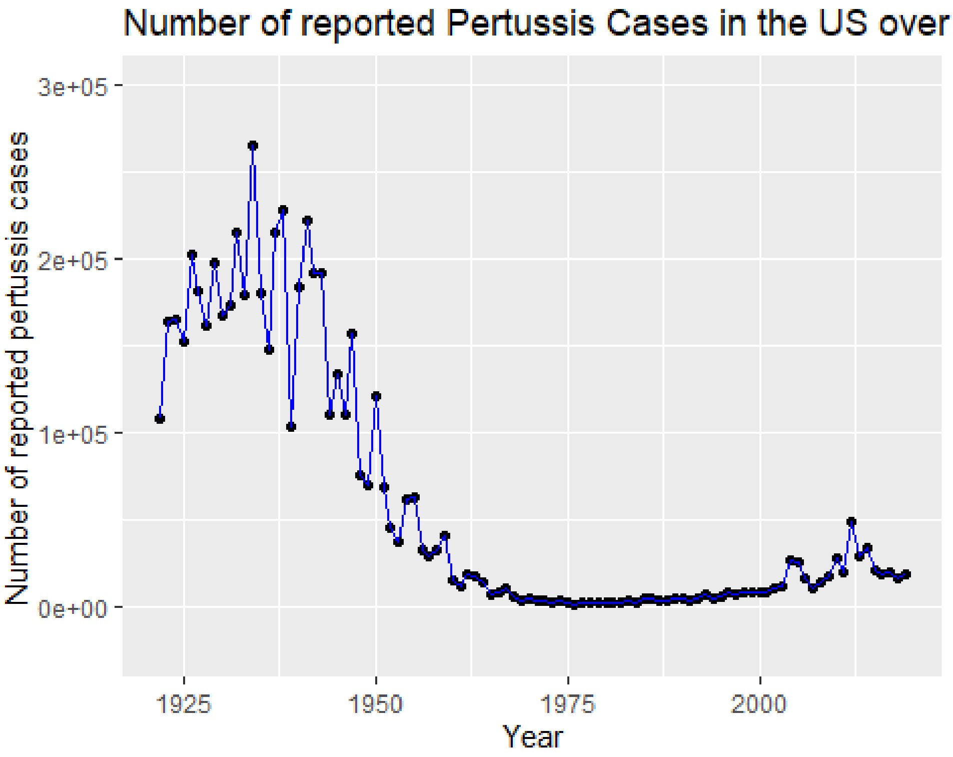


A simpler way to do this is to use the package `datapasta`. After installation and loading of `datapasta`; one can simply copy the data on the website and paste it into `R` using the addin drop-down menu to paste the data as a data.frame.

```
# Paste data in using datapasta
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)
    )
```

This provides a data.frame identical to that made by the `read.table()` function + the line of code required to change the second column to numeric. It is undeniably simpler and will proove useful. As an extra check, we can repeat the plotting for cdc.

```
# Plot raw data with ggplot2
p.dat <- ggplot(cdc, aes(Year, No..Reported.Pertussis.Cases)) +
  geom_point() +
  labs(title = "Number of reported Pertussis Cases in the US over time", y =
"Number of reported pertussis cases") +
  geom_line(col = "blue") +
  ylim(c(-25000, 300000))

p.dat
```

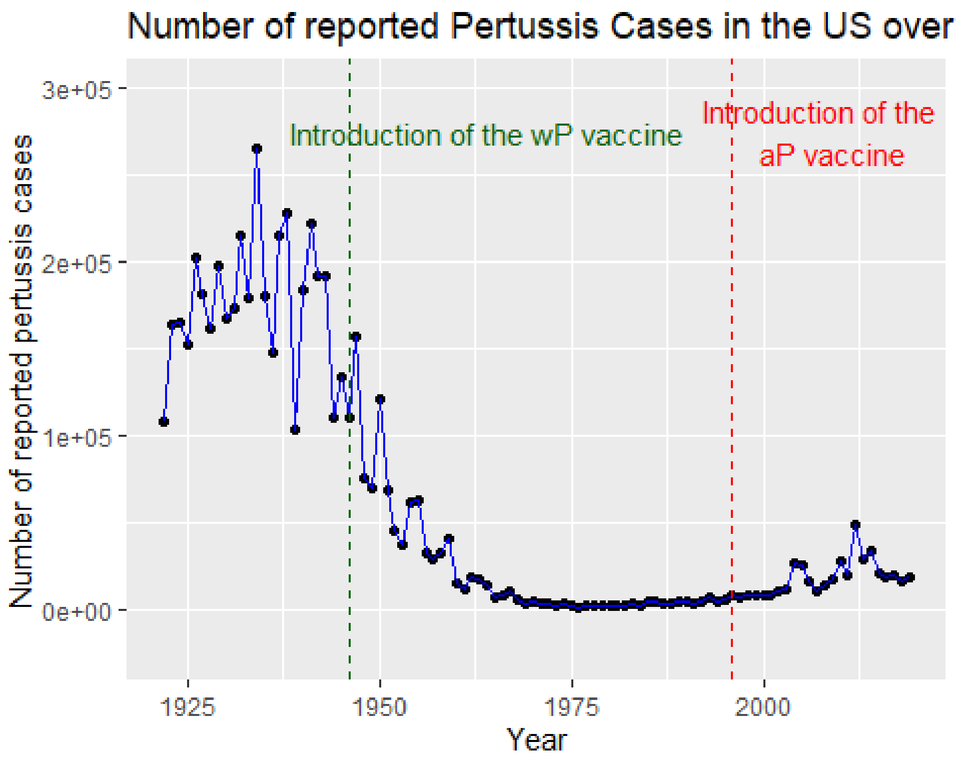


The plots appear identical as expected.

Question 2 and 3

We can also add information about important historical events, such as the advent of new vaccines.

```
# Plot add historical events to plot
p.dat.anno <- p.dat +
    geom_vline(xintercept = 1946, col = "darkgreen", lty = 2) +
    geom_vline(xintercept = 1996, col = "red", lty = 2) +
    annotate(geom = "text", x = 1964, y= 275000 , label="Introduction of the
wP vaccine", color="darkgreen") +
    annotate(geom = "text", x = 2008, y= 275000 , label="Introduction of the
  aP vaccine", color="red")

p.dat.anno
```



Introduction of the wP vaccine lead to a reduced case load. This took a while as a certain portion of the population needs to be vaccinated before the population rather than just the vaccinated individuals become protected. However, with a little time, it is clear that vaccination with wP, overtime, lead to practically 0 cases.

Unfortunately, after introduction of the aP vaccine, there seems to be a slight increase in cases. However, it is not clear whether this is a correlation or a causation. It might be possible that, vaccination rates have gone down, independent of the aP vaccine or wP vaccine being offered. So one possibility for this change is vaccine hesitancy. Another possibility is mutations of Pertussis, or it could be that the aP vaccine is less effective. Furthermore, another possibility is that increased travel has lead to an influx of unvaccinated populations, or unvaccinated individuals from the US becoming infected while traveling.

It seems likely, however, that the aP vaccine doesn't work as well, because it is mainly young adults, 10 year-olds etc, who caused the spike in infections and they were the first to recieve the aP vaccine. It thus seems likely the aP vaccine gives waning immunity, with immunity disappearing about 10+ years after vaccination.

Exploring CMI-PB data

The CMI-PB project aims to provide the scientific community with information on why the vaccine-preventable disease of Pertussis are seeing an increase in cases. Investigating this requires an understanding of the mechanisms of waning immunity to Pertussis, which is one of the goals of the project.

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

# Check this has read in correctly
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
```

Question 4

```
# Find number of people vaccinated with each vaccine
aP <- length(subject$infancy_vac == "aP")
wP <- length(subject$infancy_vac == "wP")

# Answers
aP

## [1] 96

wP

## [1] 96
```

There are 96 people who were vaccinated with the aP vaccination in infancy and 96 people who were vaccinated with the wP vaccination in infacny.

Question 5

```
# Find number of males and females in the dataset
male <- length(subject$biological_sex == "Male")
female <- length(subject$biological_sex == "Female")

# Answers
male

## [1] 96

female

## [1] 96
```

There are 96 males and 96 females in the dataset. Note, it would have been possible to find the number of one sex by subtracting the number of the other sex from the total, but this method is more robust, because if there was any missing data or unknowns they would not effect the method used, but would effect the method suggested.

Question 6

```
# Make a sex, race and ethnicity data.frame
#ber <- subject[, c("biological_sex", "ethnicity", "race")]
#bre <- subject[, c("biological_sex", "race", "ethnicity")]
reb <- subject[, c("race", "ethnicity", "biological_sex")]

# Get a table
#table(ber)
#table(bre)
table(reb)
```

```
## , , biological_sex = Female
## 
##                                             ethnicity
## race                                         Hispanic or Latino
##   American Indian/Alaska Native                              0
##   Asian                                                      0
##   Black or African American                                  0
##   More Than One Race                                         3
##   Native Hawaiian or Other Pacific Islander                  0
##   Unknown or Not Reported                                    8
##   White                                                      7
##                                             ethnicity
## race                                         Not Hispanic or Latino Unknown
##   American Indian/Alaska Native                                  0       0
##   Asian                                                         18       0
##   Black or African American                                      2       0
##   More Than One Race                                             5       0
##   Native Hawaiian or Other Pacific Islander                      1       0
##   Unknown or Not Reported                                        2       0
##   White                                                         19       1
## 
## , , biological_sex = Male
## 
##                                             ethnicity
## race                                         Hispanic or Latino
##   American Indian/Alaska Native                              0
##   Asian                                                      0
##   Black or African American                                  0
##   More Than One Race                                         1
##   Native Hawaiian or Other Pacific Islander                  0
##   Unknown or Not Reported                                    1
##   White                                                      3
##                                             ethnicity
## race                                         Not Hispanic or Latino Unknown
##   American Indian/Alaska Native                                  1       0
##   Asian                                                          9       0
##   Black or African American                                      0       0
##   More Than One Race                                             1       0
##   Native Hawaiian or Other Pacific Islander                      1       0
##   Unknown or Not Reported                                        1       2
##   White                                                          9       1
```

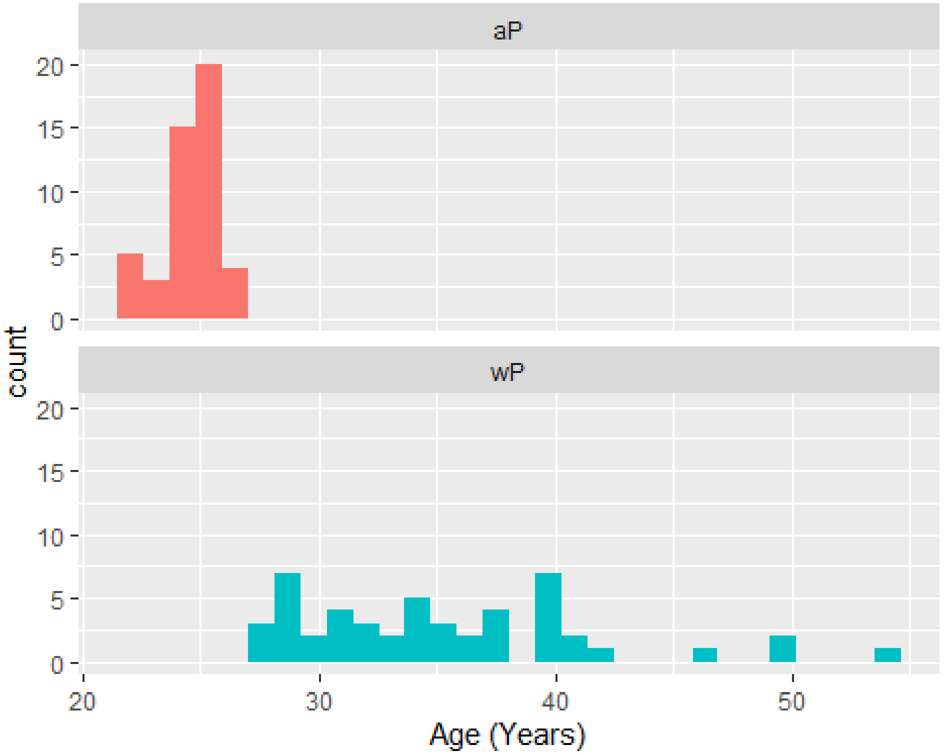
While this is not perfect, it a reasonable way to tabulate the three factors against each other. Note, that, dependent on the order of the three factors, the tables will be split differently. Placing biological sex last makes sense, because it means we only get two tables (if ethnicity was last, there would be a table for each ethnicity, containing biological sex against race, which is more difficult to interpret due to more tables to compare).

Question 7

```
# Find the age of individuals
subject$age <- today() - ymd(subject$year_of_birth)

# Find average age of aP individuals
ap.age <- subject[subject$infancy_vac == "aP", "age"]
time_length(mean(ap.age), "year")

## [1] 24.50808

# Find average age of wP individuals
wp.age <- subject[subject$infancy_vac == "wP", "age"]
time_length(mean(wp.age), "year")

## [1] 35.35253
```

The format of the `year_of_birth` column is year-month-date, so the `ymd()` function was used. To see if the two groups differ in age significantly we can probably use a student's t-test. However, as this requires parametric data, it would be wise to quickly plot the data to check whether it looks relatively normal.

```
# Plot average ages
ggplot(subject, aes(time_length(age, "year"), fill=as.factor(infancy_vac))) +
  geom_histogram(show.legend = FALSE) +
  facet_wrap(vars(infancy_vac), nrow=2) +
  labs(x = "Age (Years)")

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



Neither of these look particularly parametric, so a t-test is not appropriate. By eye they do look significantly different. Instead of the parametric t-test we can instead use a non-parametric test such as a Wilcoxin test.

```
# wilcox test
wilcox.test(time_length(ap.age, "year"), time_length(wp.age, "year"),
alternative = "two.sided")

## Warning in wilcox.test.default(time_length(ap.age, "year"),
## time_length(wp.age, : cannot compute exact p-value with ties

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  time_length(ap.age, "year") and time_length(wp.age, "year")
## W = 0, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

This gives a p-value < 2.2e-16, and thus the two vaccine groups do have a significantly different age spread.

Question 8

The age at receiving a booster vaccination can be calculated in a similar way.

```
# Find the age of individuals
subject$age_at_boost <- time_length(ymd(subject$date_of_boost) -
ymd(subject$year_of_birth), "year")
head(subject$age_at_boost)

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

Joining multiple data tables

Question 9 and 10

We can now fetch the speciman and titer data as well, these include values for scientific experiments, while subject was mainly metadata on the subjects who gave samples for these experiments. To check what to join by we can use `col_names()`. If the columns to join by have the same data, but different column names, then `by.x` and `by.y` can be used instead of by.

```
# load data for specimens and ab_titer
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)

# check dimensions and colnames of each
dim(subject)

## [1] 96 10

colnames(subject)

##  [1] "subject_id"     "infancy_vac"    "biological_sex" "ethnicity"
##  [5] "race"           "year_of_birth"  "date_of_boost"  "study_name"
##  [9] "age"            "age_at_boost"

dim(specimen)

## [1] 729   6

colnames(specimen)

## [1] "specimen_id"                   "subject_id"
## [3] "actual_day_relative_to_boost"  "planned_day_relative_to_boost"
## [5] "specimen_type"                 "visit"

dim(titer)

## [1] 32675     7

colnames(titer)

## [1] "specimen_id"              "isotype"
## [3] "is_antigen_specific"      "antigen"
## [5] "ab_titer"                 "unit"
## [7] "lower_limit_of_detection"
```

```
# first join specimen and subject, as both have subject_id
meta <- inner_join(specimen, subject, by = "subject_id")

dim(meta)
```

```
## [1] 729  15
```

```
colnames(meta)
```

```
##  [1] "specimen_id"                   "subject_id"
##  [3] "actual_day_relative_to_boost"  "planned_day_relative_to_boost"
##  [5] "specimen_type"                 "visit"
##  [7] "infancy_vac"                   "biological_sex"
##  [9] "ethnicity"                     "race"
## [11] "year_of_birth"                 "date_of_boost"
## [13] "study_name"                    "age"
## [15] "age_at_boost"
```

```
# then join meta and titer as both have specimen_id
abdata <- inner_join(titer, meta, by = "specimen_id")

dim(abdata)
```

```
## [1] 32675    21
```

```
colnames(abdata)
```

```
##  [1] "specimen_id"                   "isotype"
##  [3] "is_antigen_specific"           "antigen"
##  [5] "ab_titer"                      "unit"
##  [7] "lower_limit_of_detection"      "subject_id"
##  [9] "actual_day_relative_to_boost"  "planned_day_relative_to_boost"
## [11] "specimen_type"                 "visit"
## [13] "infancy_vac"                   "biological_sex"
## [15] "ethnicity"                     "race"
## [17] "year_of_birth"                 "date_of_boost"
## [19] "study_name"                    "age"
## [21] "age_at_boost"
```

```
head(abdata)
```

```
##   specimen_id isotype is_antigen_specific antigen   ab_titer  unit
## 1           1     IgE               FALSE   Total 1110.21154 UG/ML
## 2           1     IgE               FALSE   Total 2708.91616 IU/ML
## 3           1     IgG                TRUE      PT   68.56614 IU/ML
## 4           1     IgG                TRUE     PRN  332.12718 IU/ML
## 5           1     IgG                TRUE     FHA 1887.12263 IU/ML
## 6           1     IgE                TRUE     ACT    0.10000 IU/ML
##   lower_limit_of_detection subject_id actual_day_relative_to_boost
## 1                      NaN          1                           -3
## 2                29.170000          1                           -3
## 3                 0.530000          1                           -3
## 4                 1.070000         1                           -3
## 5                 0.064000         1                           -3
## 6                 2.816431         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 age_at_boost
## 1 13218 days     30.69678
## 2 13218 days     30.69678
## 3 13218 days     30.69678
## 4 13218 days     30.69678
## 5 13218 days     30.69678
## 6 13218 days     30.69678
```

We now have a single table with the titer data related to specimen and subject data. This can now be used for analysis.

Question 11

To see how many specimens there are for each isotype we can use `table()`.

```
# How many of each isotype
table(abdata$isotype)

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

Question 12

```
# inspect visit 8 specimens
table(abdata$visit)
```

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

Visit 8 specimens are far fewer in number (likely there was a drop in subjects who made it to this late visit). It would thus be best to exclude this data poor visit from our analysis.

Examining IgG1 Ab titer levels

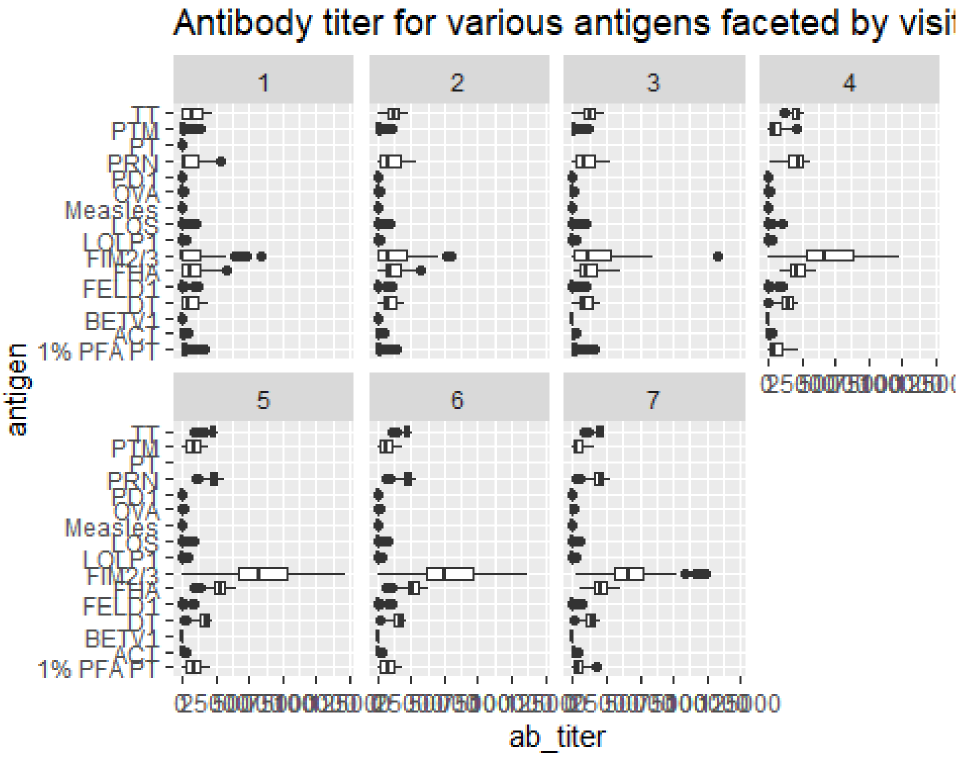
As previously mentioned, we should exclude visit 8 from our analysis. We are also going to focus on IgG1.

```
#filter data
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 age_at_boost
## 1 13218 days     30.69678
## 2 13218 days     30.69678
## 3 13218 days     30.69678
## 4 13218 days     30.69678
## 5 13218 days     30.69678
## 6 13218 days     30.69678
```

Question 13

As before, we should start by plotting our raw data.

```
# plot boxchart
ggplot(ig1, aes(ab_titer, antigen)) +
  geom_boxplot() +
  facet_wrap(vars(visit), nrow=2) +
  labs(title = "Antibody titer for various antigens faceted by visit")
```

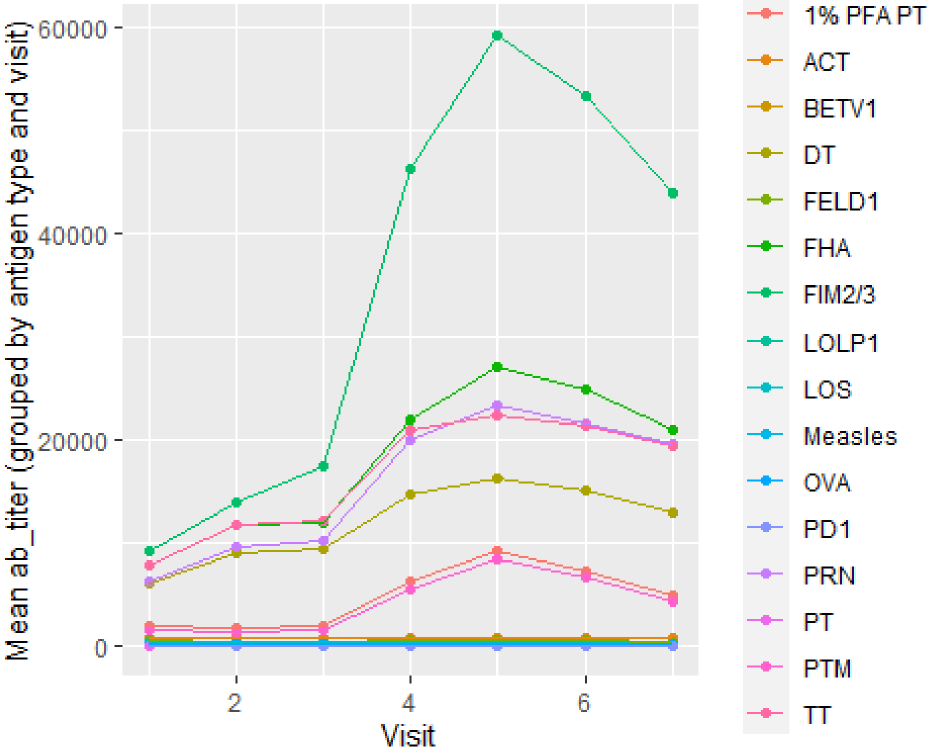


It might be more intuitive to group by antigen and make a time course.

```
# Create average data for this plot
ig1.avgs <- ig1 %>%
  group_by(antigen, visit) %>%
  summarize(mean = mean(ab_titer), n = n())
```

```
## `summarise()` has grouped output by 'antigen'. You can override using the
## `.groups` argument.
```

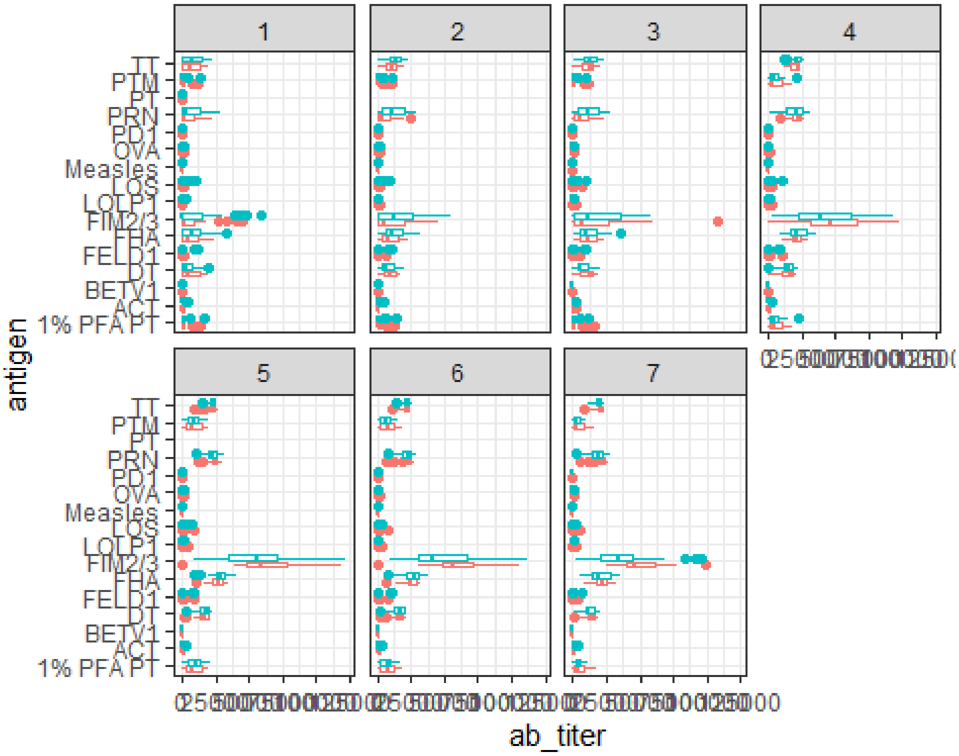
```
# Plot these averages
ggplot(ig1.avgs, aes(visit, mean, group = antigen, col = antigen)) +
  geom_point() +
  geom_line() +
  labs(x = "Visit", y = "Mean ab_titer (grouped by antigen type and visit)",
col = "Antigen")
```



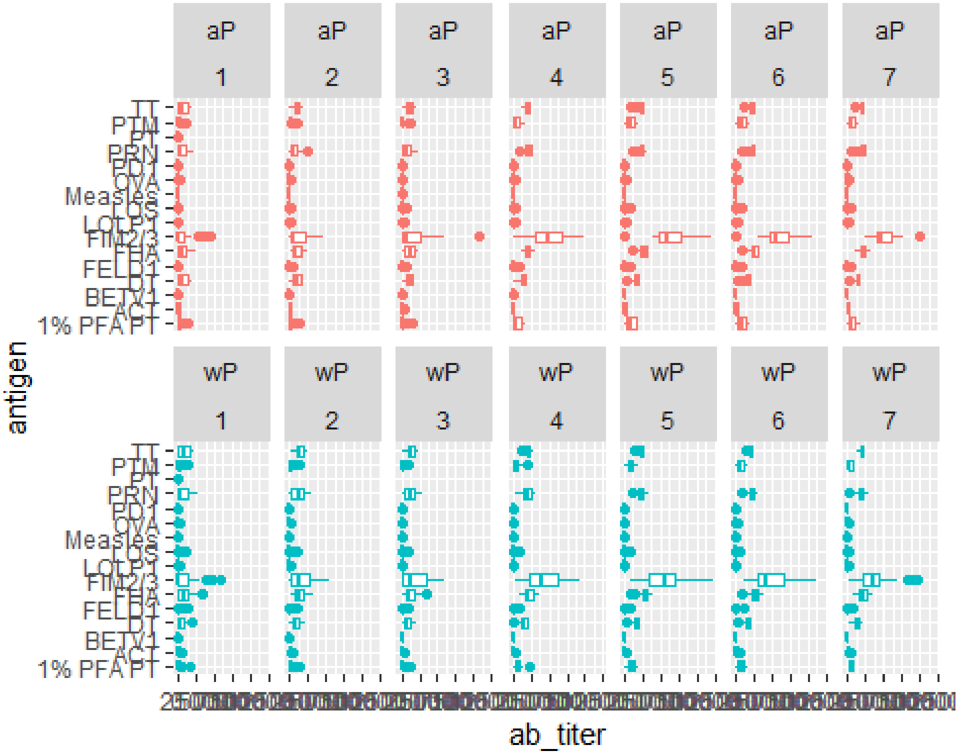
From this graph it seems that DT, FHA, FRIM2/3, PRN, 1% PFA PT, PTM and TT (an antigen for one of the other infectious agents that the tdap vaccine protects against) all have some change in titer while LOS, LOLP1, Measles, OVA, PD1, PT and BETV1 have no or minimal change in antibody titer. On the website we can look up what these antigens are. For example, PRN is pertactin autotransporter, a protein, a link to uniprot is provided, and there we can see it is likely virulence related, and so provided in the vaccine. This makes sense, antibodies against antigens for other infectious diseases should not go up, while components of the vaccine should see an increase in the antibodies targeting them.

We can also look at the differences between aP and wP vaccinated individuals.

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



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

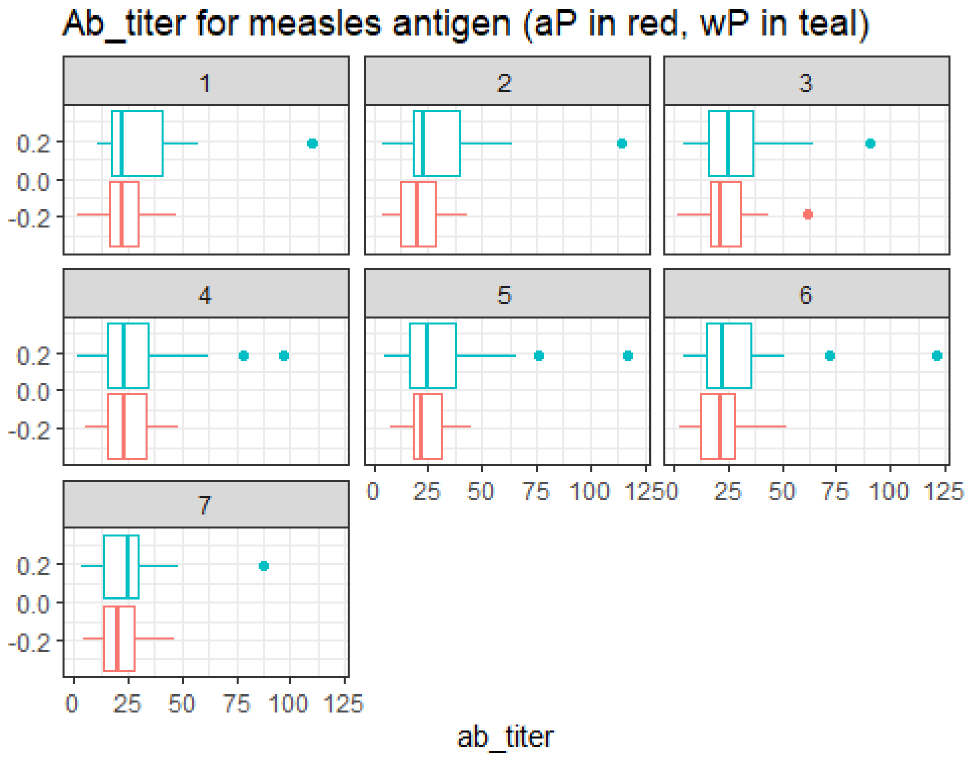


The second plot is better, as the first shows too much information to easily process. Although, it is useful for a quick and dirty comparison of the two.

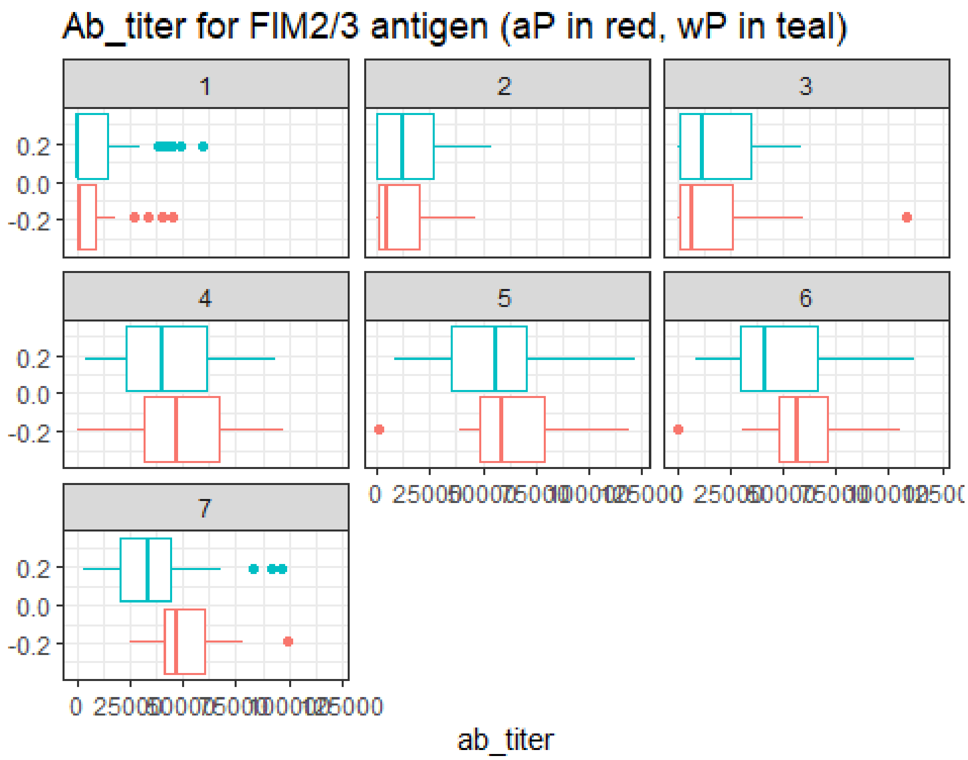
Question 15

We can now focus in on particular antigens, making them easier to look at.

```
# plot measles ab_titer
filter(ig1, antigen=="Measles") %>%
  ggplot() +
  aes(ab_titer, col=infancy_vac) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(vars(visit)) +
  theme_bw() +
  labs(title = "Ab_titer for measles antigen (aP in red, wP in teal)")
```



```
# plot fim ab_titer
filter(ig1, antigen=="FIM2/3") %>%
  ggplot() +
  aes(ab_titer, col=infancy_vac) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(vars(visit)) +
  theme_bw() +
  labs(title = "Ab_titer for FIM2/3 antigen (aP in red, wP in teal)")
```



Question 16

FIM2/3 is part of the pertussis fimbriae, which is on the cell-surface, and so is easily found by the immune system. Thus it is a good candidate for an antigen in the vaccine, and we see it has high ab_titers. Measles antigens are not in the vaccine and, unsurprisingly, given this, shows little change.

Question 17

No, unfortunately not.

Obtaining CMI-PB RNASeq data

For RNA-Seq data the API query mechanism quickly hits the web browser interface limit for file size. We can do a more targeted search to minimize the size of the data we have to use. Specifically, we will use the ensembl_gene_id = eq.ENSG00000211896.7, which is for key gene involved in expressing any IgG1 antibody, namely the IGHG1 gene.

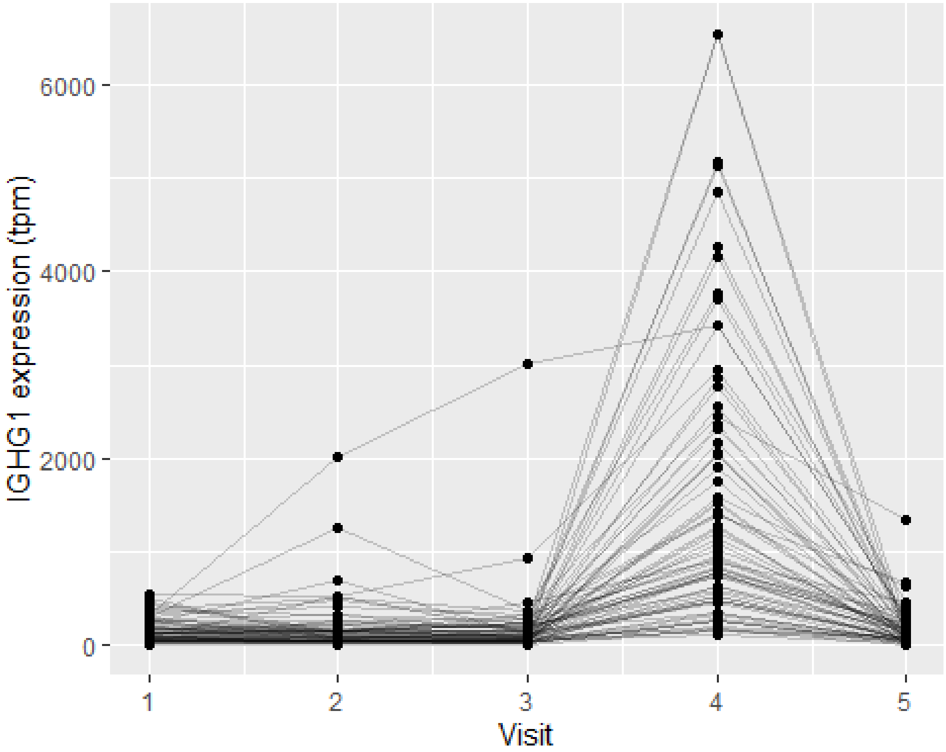
```
# url to use
url <- "https://www.cmi-
pb.org/api/v2/rnaseq?versioned_ensembl_gene_id=eq.ENSG00000211896.7"

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

```
# join this data to the meta data
ssrna <- inner_join(rna, meta)

## Joining, by = "specimen_id"

# check your work
head(ssrna)

##   versioned_ensembl_gene_id specimen_id raw_count      tpm subject_id
## 1         ENSG00000211896.7         344     18613  929.640         44
## 2         ENSG00000211896.7         243      2011  112.584         31
## 3         ENSG00000211896.7         261      2161  124.759         33
## 4         ENSG00000211896.7         282      2428  138.292         36
## 5         ENSG00000211896.7         345     51963 2946.136         44
## 6         ENSG00000211896.7         244     49652 2356.749         31
##   actual_day_relative_to_boost planned_day_relative_to_boost specimen_type
## 1                            3                             3         Blood
## 2                            3                             3         Blood
## 3                           15                            14         Blood
## 4                            1                             1         Blood
## 5                            7                             7         Blood
## 6                            7                             7         Blood
##   visit infancy_vac biological_sex              ethnicity
race
## 1     3          aP         Female     Hispanic or Latino More Than One
Race
## 2     3          wP         Female Not Hispanic or Latino
Asian
## 3     5          wP           Male     Hispanic or Latino More Than One
Race
## 4     2          aP         Female     Hispanic or Latino
White
## 5     4          aP         Female     Hispanic or Latino More Than One
Race
## 6     4          wP         Female Not Hispanic or Latino
Asian
##   year_of_birth date_of_boost   study_name        age age_at_boost
## 1    1998-01-01    2016-11-07 2020_dataset  8835 days     18.85010
## 2    1989-01-01    2016-09-26 2020_dataset 12122 days     27.73443
## 3    1990-01-01    2016-10-10 2020_dataset 11757 days     26.77344
## 4    1997-01-01    2016-10-24 2020_dataset  9200 days     19.81109
## 5    1998-01-01    2016-11-07 2020_dataset  8835 days     18.85010
## 6    1989-01-01    2016-09-26 2020_dataset 12122 days     27.73443
```

With the data loaded, we can first plot it to get a visualization of the data we will be working with.

Question 18

```
# plot ssrna
ggplot(ssrna, aes(visit, tpm, group=subject_id)) +
  geom_point() +
  geom_line(alpha=0.2) +
  labs(y = "IGHG1 expression (tpm)", x = "Visit")
```



### Question 19

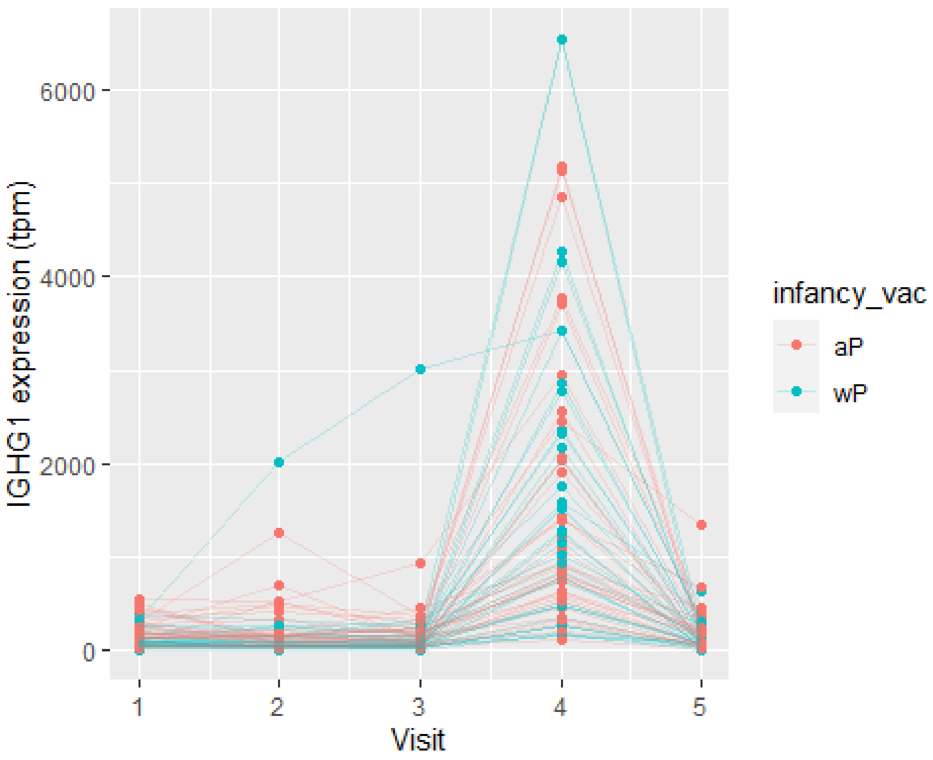
Interestingly, the major spike in expression, for most specimens, is around visit 4.

Question 20

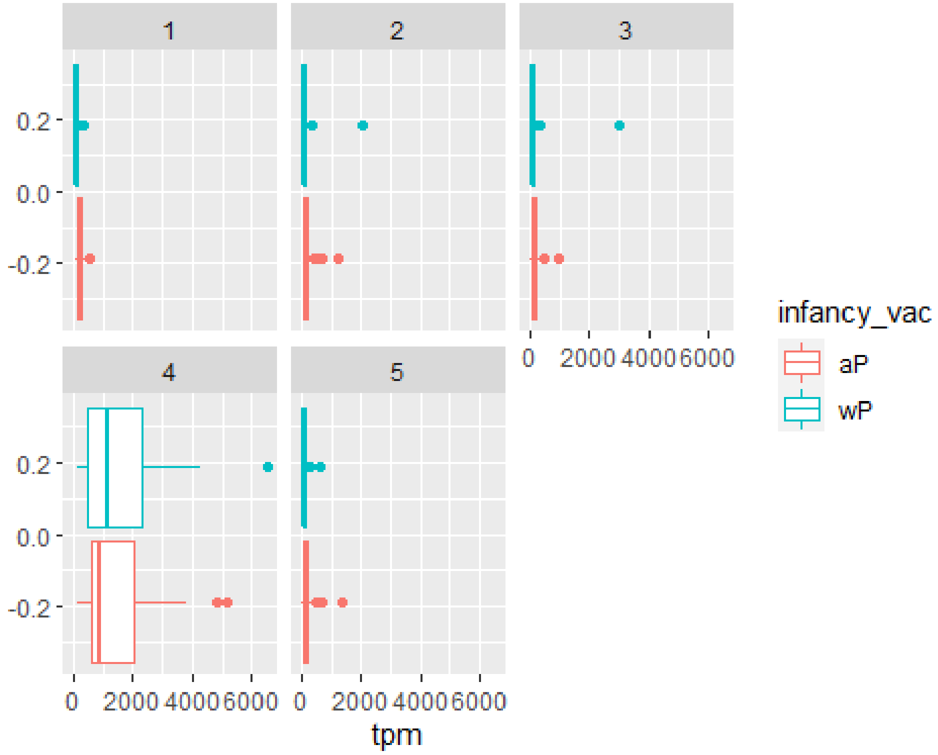
This trend does not match that in the antibody data perfectly, as the maximum for the antibody data is closer to 5. This, however, makes sense, we would expect antibodies to be long-lived, lasting much longer than gene expression. Furthermore, a small population of cells will continue to make the antibody (T-cells), even after the main immune response has ended.

We can again compare by vaccine. Colouring by vaccine in the previous plot is possible, but not particularly informative, as it is hard to interpret. Therefore using a boxplot is more informative in this case.

```
# plot as previously
ggplot(ssrna, aes(visit, tpm, group=subject_id, col = infancy_vac)) +
  geom_point() +
  geom_line(alpha=0.2) +
  labs(y = "IGHG1 expression (tpm)", x = "Visit")
```

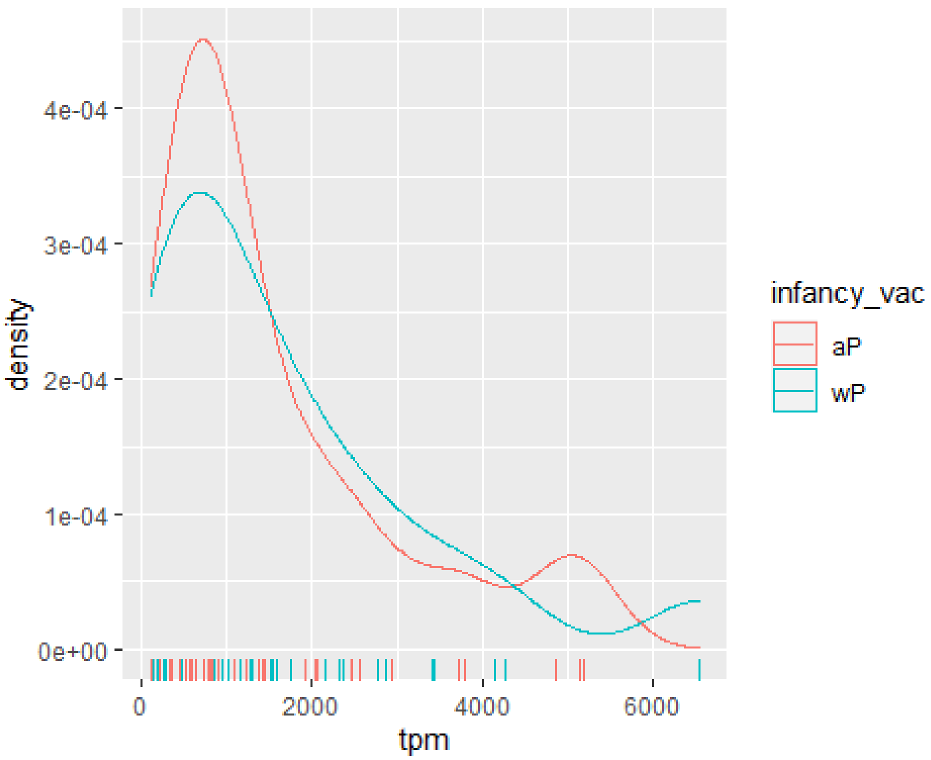


```r
# plot ssrn by vaccine
ggplot(ssrna, aes(tpm, col=infancy_vac)) +
  geom_boxplot() +
  facet_wrap(vars(visit))
```

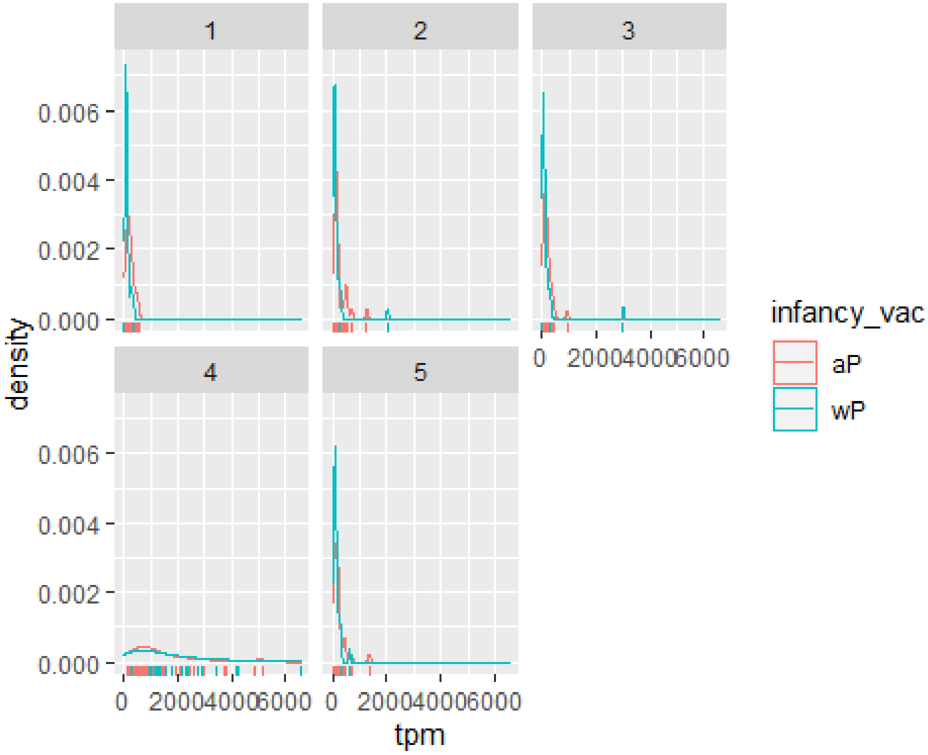


There are no obvious differences here. We can also look at a particular visit.

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



```
## ssrna per visit
ggplot(ssrna, aes(tpm, col=infancy_vac)) +
  geom_density() +
  geom_rug() +
  facet_wrap(~visit)
```



By visit there is some difference, but whether this is significant is unclear.

Working with larger datasets

```
# load data
rnaseq <- read.csv("2020LD_rnaseq.csv")

# check
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

dim(rnaseq)

## [1] 10502460        4
```

With the data loaded we can start exploring it.

```
# number of genes per specimen
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

# number of specimens
length(n_genes)

## [1] 180

# are there the same number of genes for all specimens
all(n_genes[1]==n_genes)

## [1] TRUE
```

Now we can convert to the wide format, which is easier to read, as it gives values for each gene in each location in a clear table

```
# convert to wide format
rna_wide <- rnaseq %>%
  select(versioned_ensembl_gene_id, specimen_id, tpm) %>%
  pivot_wider(names_from = specimen_id, values_from=tpm)

# get dimensions
dim(rna_wide)

## [1] 58347   181

# check results
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
```

The next step is to filter the data to remove any zero count genes, which are not required for further analysis.

```
# create a numbers only rna.wide
rna.wide <- as.data.frame(rna_wide[, -1])

# set first column of rna_wide as rownames for rna.wide
rownames(rna.wide) <- rna_wide$versioned_ensembl_gene_id

# check dimensions
dim(rna.wide)

## [1] 58347   180

dim(rna_wide)
```

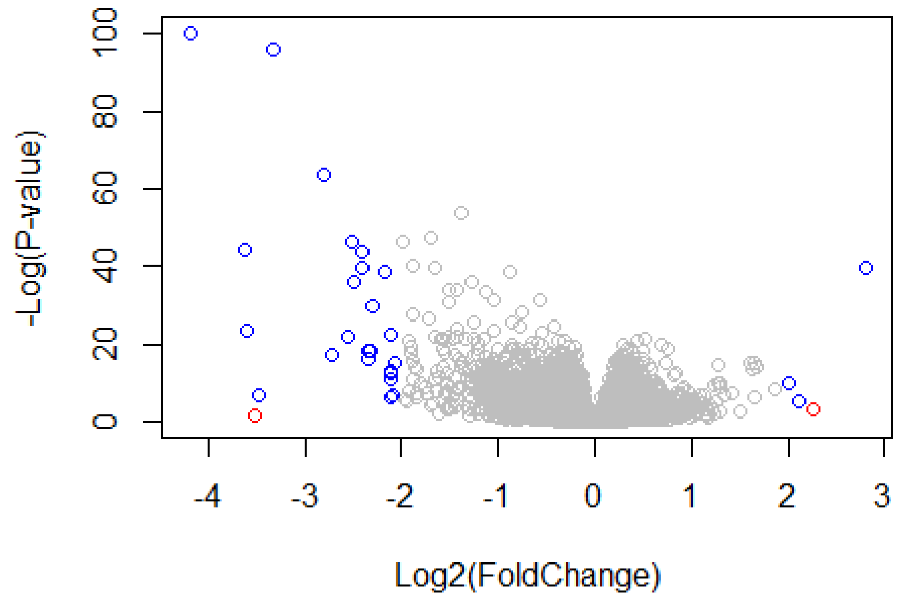
```
## [1] 58347   181

# find rows with a total of zero
ind <- rowSums(rna.wide) != 0

# use the indices to remove zero count genes
rna.wide <- rna.wide[ ind , ]

# check
sum(ind)

## [1] 45219

dim(rna.wide)

## [1] 45219   180
```

All zero count genes have now been removed from the object and analysis can begin. The next step might be to use DESeq2.

```
# order rna.wide and specimen
rna.wide <- rna.wide[order(colnames(rna.wide))]
meta <- meta[order(meta$specimen_id),]

# remove specimen's not in the data
met <- meta[c(colnames(rna.wide)),]

# remove NAs
rna.wide[is.na(rna.wide)] = 0

# DESeq
dds = DESeqDataSetFromMatrix(countData = round(as.matrix(rna.wide)),
                             colData = met,
                             design = ~ as.factor(infancy_vac))

## converting counts to integer mode

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables
in
## design formula are characters, converting to factors

dds = DESeq(dds)

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates
## fitting model and testing

## -- replacing outliers and refitting for 114 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

Now this can be used to look at results and make a volcano plot.

```
# look at results
res <- results(dds)
res

## log2 fold change (MLE): as.factor.infancy vac.wP
## Wald test p-value: as.factor.infancy vac.wP
## DataFrame with 45219 rows and 6 columns
##                       baseMean log2FoldChange     lfcSE       stat
pvalue
##                      <numeric>      <numeric> <numeric>  <numeric>
<numeric>
## ENSG00000229704.1  0.00000000             NA        NA         NA
NA
## ENSG00000229711.1  0.00000000             NA        NA         NA
NA
## ENSG00000229715.4  1.26754140     -0.0570342  0.195581 -0.2916148
0.770581
## ENSG00000229716.2  0.00528201     -0.0417721  2.919420 -0.0143083
0.988584
## ENSG00000229717.2  0.00000000             NA        NA         NA
NA
## ...                       ...            ...       ...        ...
...
## ENSG00000170439.6    0.178432     -0.2653791 0.7319677  -0.362556
0.7169367
## ENSG00000170442.11   2.228797     -0.4517177 0.2524462  -1.789362
0.0735565
## ENSG00000170445.12  24.397618     -0.0150468 0.0444863  -0.338235
0.7351858
## ENSG00000170448.11  11.923953      0.1607262 0.0854792   1.880296
0.0600678
## ENSG00000170456.15   4.293316     -0.1393194 0.1069225  -1.302994
0.1925767
##                          padj
##                     <numeric>
## ENSG00000229704.1         NA
## ENSG00000229711.1         NA
## ENSG00000229715.4   0.883721
## ENSG00000229716.2         NA
## ENSG00000229717.2         NA
## ...                      ...
## ENSG00000170439.6         NA
## ENSG00000170442.11  0.204272
## ENSG00000170445.12  0.862762
## ENSG00000170448.11  0.177175
## ENSG00000170456.15  0.394493

summary(res)

##
## out of 34788 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)       : 2440, 7%
## LFC < 0 (down)     : 3599, 10%
## outliers [1]       : 0, 0%
## low counts [2]     : 12613, 36%
## (mean count < 0)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

#
res05 <- results(dds, alpha=0.05)
summary(res05)

##
## out of 34788 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 1981, 5.7%
## LFC < 0 (down)     : 2869, 8.2%
## outliers [1]       : 0, 0%
## low counts [2]     : 15268, 44%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

##### Plotting
# Setup our custom point color vector
mycols <- rep("gray", nrow(res))
mycols[ abs(res$log2FoldChange) > 2 ]  <- "red"

inds <- (res$padj < 0.01) & (abs(res$log2FoldChange) > 2 )
mycols[ inds ] <- "blue"

# Volcano plot with custom colors
plot( res$log2FoldChange,  -log(res$padj), col=mycols, ylab="-Log(P-value)",
xlab="Log2(FoldChange)" )
```



While this leaves out many important steps, such as testing other key variables such as sex and age (the differences here could be due to age rather than vaccine given that vaccine groups vary significantly in age), and adding gene names to the results. This code provides a start and previous labs could be used to further flesh it out.

Session Information

```
sessionInfo()

## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] DESeq2_1.34.0               SummarizedExperiment_1.24.0
##  [3] Biobase_2.54.0              MatrixGenerics_1.6.0
##  [5] matrixStats_0.61.0          GenomicRanges_1.46.1
##  [7] GenomeInfoDb_1.30.1         IRanges_2.28.0
##  [9] S4Vectors_0.32.3            BiocGenerics_0.40.0
## [11] tidyr_1.2.0                 dplyr_1.0.8
## [13] lubridate_1.8.0             jsonlite_1.8.0
## [15] ggplot2_3.3.5               datapasta_3.1.0
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2             bit64_4.0.5            splines_4.1.2
##  [4] highr_0.9              blob_1.2.2             GenomeInfoDbData_1.2.7
##  [7] yaml_2.2.2             pillar_1.7.0           RSQLite_2.2.10
## [10] lattice_0.20-45        glue_1.6.1             digest_0.6.29
## [13] RColorBrewer_1.1-2     XVector_0.34.0         colorspace_2.0-2
## [16] htmltools_0.5.2        Matrix_1.3-4           XML_3.99-0.8
## [19] pkgconfig_2.0.3        genefilter_1.76.0      zlibbioc_1.40.0
## [22] purrr_0.3.4            xtable_1.8-4           scales_1.1.1
## [25] BiocParallel_1.28.3    tibble_3.1.6           annotate_1.72.0
## [28] KEGGREST_1.34.0        farver_2.1.0           generics_0.1.2
## [31] ellipsis_0.3.2         cachem_1.0.6           withr_2.5.0
## [34] cli_3.2.0              survival_3.2-13        magrittr_2.0.2
## [37] crayon_1.5.0           memoise_2.0.1          evaluate_0.15
## [40] fansi_1.0.2            tools_4.1.2            lifecycle_1.0.1
## [43] stringr_1.4.0          locfit_1.5-9.4         munsell_0.5.0
## [46] DelayedArray_0.20.0    AnnotationDbi_1.56.2   Biostrings_2.62.0
## [49] compiler_4.1.2         rlang_1.0.1            grid_4.1.2
## [52] RCurl_1.98-1.6         rstudioapi_0.13        labeling_0.4.2
## [55] bitops_1.0-7           rmarkdown_2.11         gtable_0.3.0
## [58] DBI_1.1.2              R6_2.5.1               knitr_1.37
## [61] fastmap_1.1.0          bit_4.0.4              utf8_1.2.2
## [64] stringi_1.7.6          parallel_4.1.2         Rcpp_1.0.8
## [67] vctrs_0.3.8            geneplotter_1.72.0     png_0.1-7
## [70] tidyselect_1.1.2       xfun_0.29
```