Class 18: Pertussis mini project

AUTHOR
Nundini Varshney (PID: A16867985)

First we will examine and explore the case numbers as tracked by the CDC: https://www.cdc.gov/pertussis/surv-reporting/cases-by-year.html

We can use the datapasta package to scrape this data from the website into R:

> 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,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)
       )
View(cdc)
```

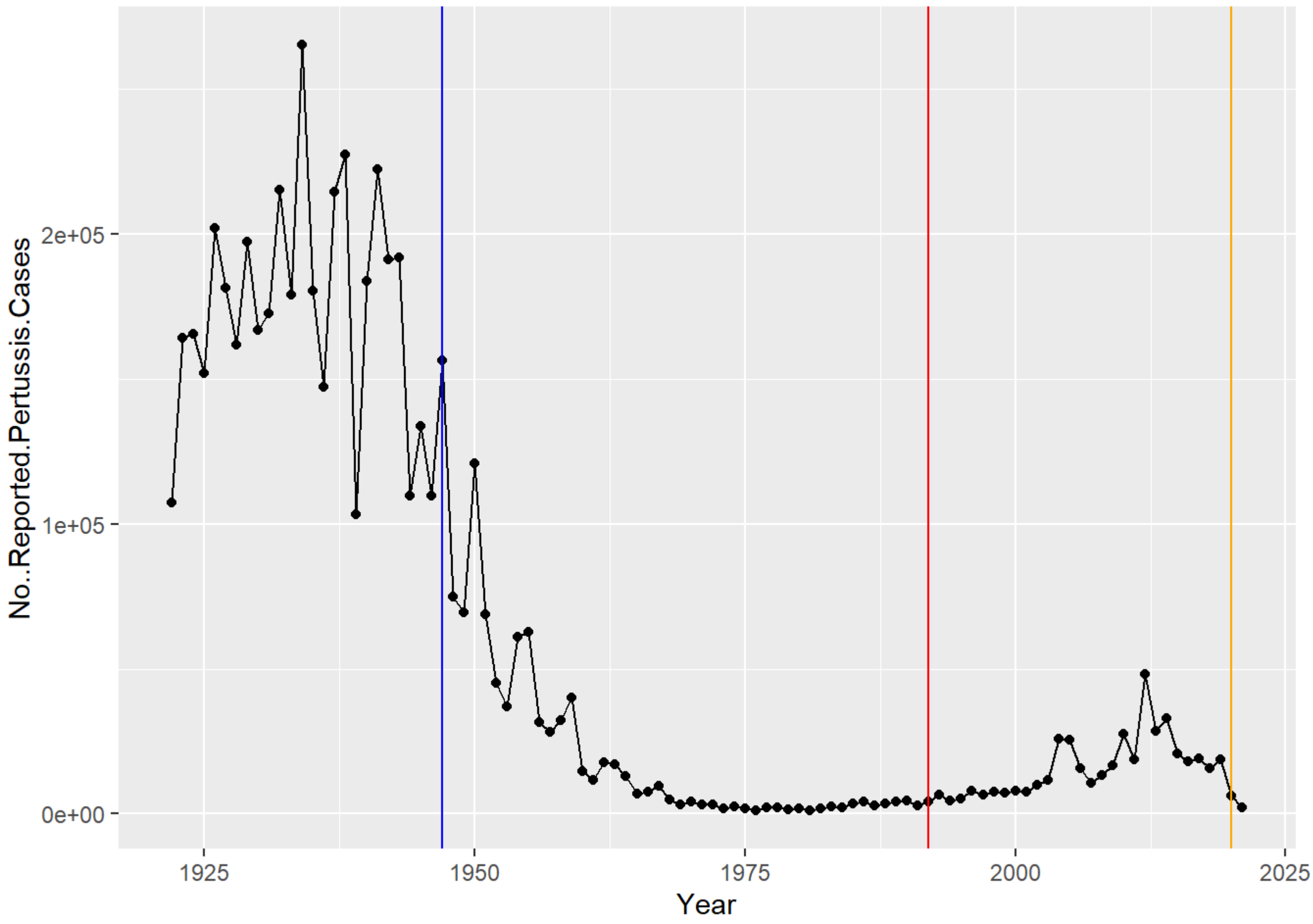
```
head(cdc)
```

```
  Year No..Reported.Pertussis.Cases
1 1922                       107473
2 1923                       164191
3 1924                       165418
4 1925                       152003
5 1926                       202210
6 1927                       181411
```

I want a plot of cases per year

```
library(ggplot2)

ggplot(cdc) +
  aes(Year, No..Reported.Pertussis.Cases) +
  geom_point() +
  geom_line() +
  geom_vline(xintercept = 1947, col = "blue") +
  geom_vline(xintercept = 1992, col = "red") +
  geom_vline(xintercept = 2020, col = "orange")
```



Access data from CMI-PB project

This database uses an API to return JSON format data

We will use the R package `jsonlite`.

```
library(jsonlite)
```

```
Warning: package 'jsonlite' was built under R version 4.3.3
```

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


head(subject)
```

```
  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
4          4          wP           Male Not Hispanic or Latino Asian
5          5          wP           Male Not Hispanic or Latino Asian
6          6          wP         Female Not Hispanic or Latino 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
4    1988-01-01    2016-08-29 2020_dataset
5    1991-01-01    2016-08-29 2020_dataset
6    1988-01-01    2016-10-10 2020_dataset
```

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

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

```

aP wP 
60 58 
```

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

```
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

We can use the lubridate package to ease the pain of doing math with dates

```
library(lubridate)
```

```
Warning: package 'lubridate' was built under R version 4.3.3
```

```
Attaching package: 'lubridate'

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

    date, intersect, setdiff, union
```

```r
time_length( today() - mdy("5-15-2002"), "years")
```

```
[1] 21.81246
```

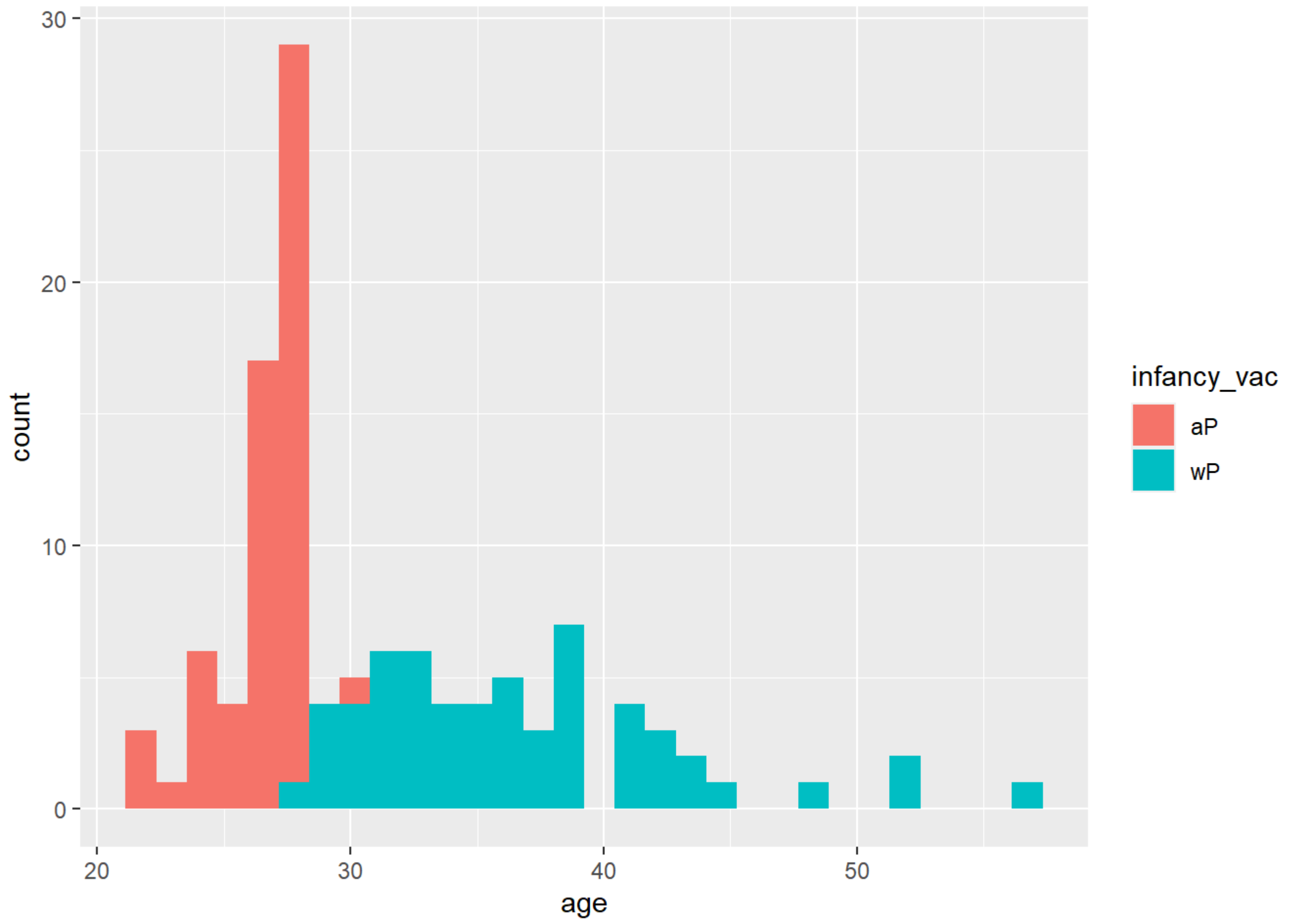
```r
today() - ymd(subject$year_of_birth)
```

```
Time differences in days
  [1] 13945 20520 15041 13215 12119 13215 15771 14310 10293 15406 13945 15406
 [13]  9927 11388 12849 13580 16137  9927 11023 15771 15041 14310 12119 11754
 [25] 13215 15041  9927 15406  9927 13215 12849  9927 12484 15041 12119  9927
 [37]  9562  9927 14310 11023 14310  9927  9562  9562  9927  9562 10293  9562
 [49]  9927  9927  9927  9562  9562  9927  9927  9927 10293  9927  9927  9927
 [61] 13580 11388 10658 11388 12484 17598 19059 19059 12484  9562  9562 12119
 [73] 10658 10658  9562  9562 13215 11388 13580 11754 11388  9562  9197  9927
 [85]  8832  9562  8832  8832  9927  9197  9562  8832 10293  9197  9562  8832
 [97] 13945 11388  9197  8466  7736  7736 11023 12849 11023 10293  9562 10658
[109] 12849  9927 10293 10293 10293 12484  8101  8832 11023  9562
```

```r
subject$age <- time_length( today() - ymd(subject$year_of_birth), "years")
```

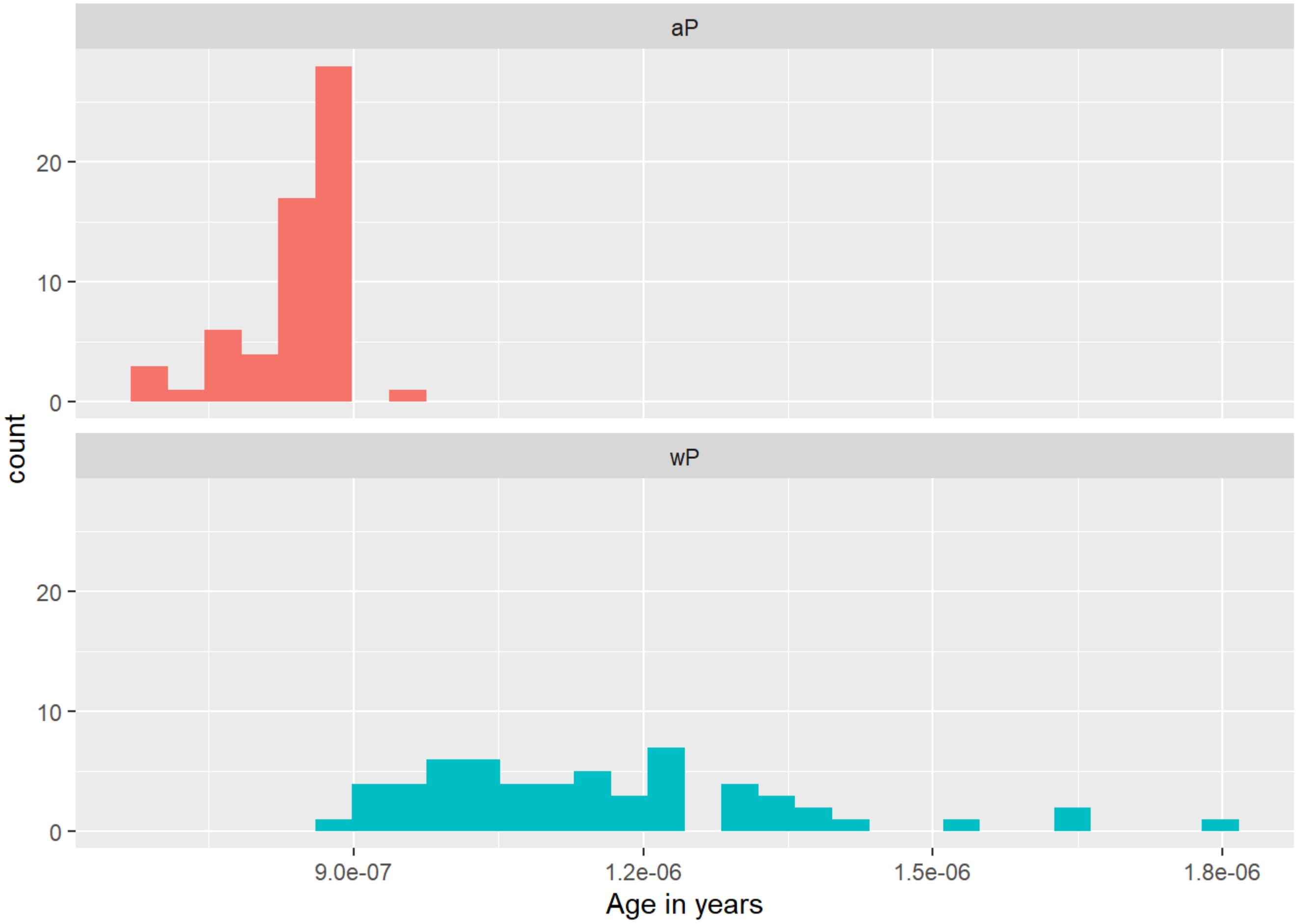
```r
ggplot(subject) +
  aes(age, fill=infancy_vac) +
  geom_histogram()
```

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



```r
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`.
```



Get more data

```r
specimen <- read_json("http://cmi-pb.org/api/specimen", simplifyVector = TRUE)
head(specimen)
```

```
  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
1                             0         Blood     1
2                             1         Blood     2
3                             3         Blood     3
4                             7         Blood     4
5                            14         Blood     5
6                            30         Blood     6
```

```r
head(subject)
```

```
  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
4          4          wP           Male Not Hispanic or Latino Asian
5          5          wP           Male Not Hispanic or Latino Asian
6          6          wP         Female Not Hispanic or Latino White
  year_of_birth date_of_boost      dataset      age
1    1986-01-01    2016-09-12 2020_dataset 38.17933
2    1968-01-01    2019-01-28 2020_dataset 56.18070
3    1983-01-01    2016-10-10 2020_dataset 41.18001
4    1988-01-01    2016-08-29 2020_dataset 36.18070
5    1991-01-01    2016-08-29 2020_dataset 33.18001
6    1988-01-01    2016-10-10 2020_dataset 36.18070
```

We need to **join** these two tables (subject and specimen) to make a single new “meta” table with all our metadata. We will use the `dplyr` function to do this.

```
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
```

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

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

```
head(meta)
```

```
  subject_id infancy_vac biological_sex              ethnicity  race
1          1          wP         Female Not Hispanic or Latino White
2          1          wP         Female Not Hispanic or Latino White
3          1          wP         Female Not Hispanic or Latino White
4          1          wP         Female Not Hispanic or Latino White
5          1          wP         Female Not Hispanic or Latino White
6          1          wP         Female Not Hispanic or Latino White
  year_of_birth date_of_boost      dataset      age specimen_id
1    1986-01-01    2016-09-12 2020_dataset 38.17933           1
2    1986-01-01    2016-09-12 2020_dataset 38.17933           2
3    1986-01-01    2016-09-12 2020_dataset 38.17933           3
4    1986-01-01    2016-09-12 2020_dataset 38.17933           4
5    1986-01-01    2016-09-12 2020_dataset 38.17933           5
```

```
6    1986-01-01    2016-09-12 2020_dataset 38.17933           6
  actual_day_relative_to_boost planned_day_relative_to_boost specimen_type
1                           -3                             0         Blood
2                            1                             1         Blood
3                            3                             3         Blood
4                            7                             7         Blood
5                           11                            14         Blood
6                           32                            30         Blood
  visit
1     1
2     2
3     3
4     4
5     5
6     6
```

Now we can read some of the other data from CMI-PB

```r
ab_titer <- read_json("http://cmi-pb.org/api/v4/plasma_ab_titer", simplifyVector = TRUE)
head(ab_titer)
```

```
  specimen_id isotype is_antigen_specific antigen        MFI MFI_normalised
1           1     IgE               FALSE   Total 1110.21154       2.493425
2           1     IgE               FALSE   Total 2708.91616       2.493425
3           1     IgG                TRUE      PT   68.56614       3.736992
4           1     IgG                TRUE     PRN  332.12718       2.602350
5           1     IgG                TRUE     FHA 1887.12263      34.050956
6           1     IgE                TRUE     ACT    0.10000       1.000000
   unit lower_limit_of_detection
1 UG/ML                 2.096133
2 IU/ML                29.170000
3 IU/ML                 0.530000
4 IU/ML                 6.205949
5 IU/ML                 4.679535
6 IU/ML                 2.816431
```

One more `inner_join()` to add all our metadata in `meta` on to our `ab_data` table:

```r
abdata <- inner_join(ab_titer, meta)
```

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

```r
head(abdata)
```

```
  specimen_id isotype is_antigen_specific antigen        MFI MFI_normalised
1           1     IgE               FALSE   Total 1110.21154       2.493425
2           1     IgE               FALSE   Total 2708.91616       2.493425
3           1     IgG                TRUE      PT   68.56614       3.736992
4           1     IgG                TRUE     PRN  332.12718       2.602350
5           1     IgG                TRUE     FHA 1887.12263      34.050956
```

```
6           1     IgE                TRUE     ACT    0.10000       1.000000
   unit lower_limit_of_detection subject_id infancy_vac biological_sex
1 UG/ML                 2.096133          1          wP         Female
2 IU/ML                29.170000          1          wP         Female
3 IU/ML                 0.530000          1          wP         Female
4 IU/ML                 6.205949          1          wP         Female
5 IU/ML                 4.679535          1          wP         Female
6 IU/ML                 2.816431          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 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 actual_day_relative_to_boost planned_day_relative_to_boost
1 38.17933                           -3                             0
2 38.17933                           -3                             0
3 38.17933                           -3                             0
4 38.17933                           -3                             0
5 38.17933                           -3                             0
6 38.17933                           -3                             0
  specimen_type visit
1         Blood     1
2         Blood     1
3         Blood     1
4         Blood     1
5         Blood     1
6         Blood     1
```

Our first exploratory plot:

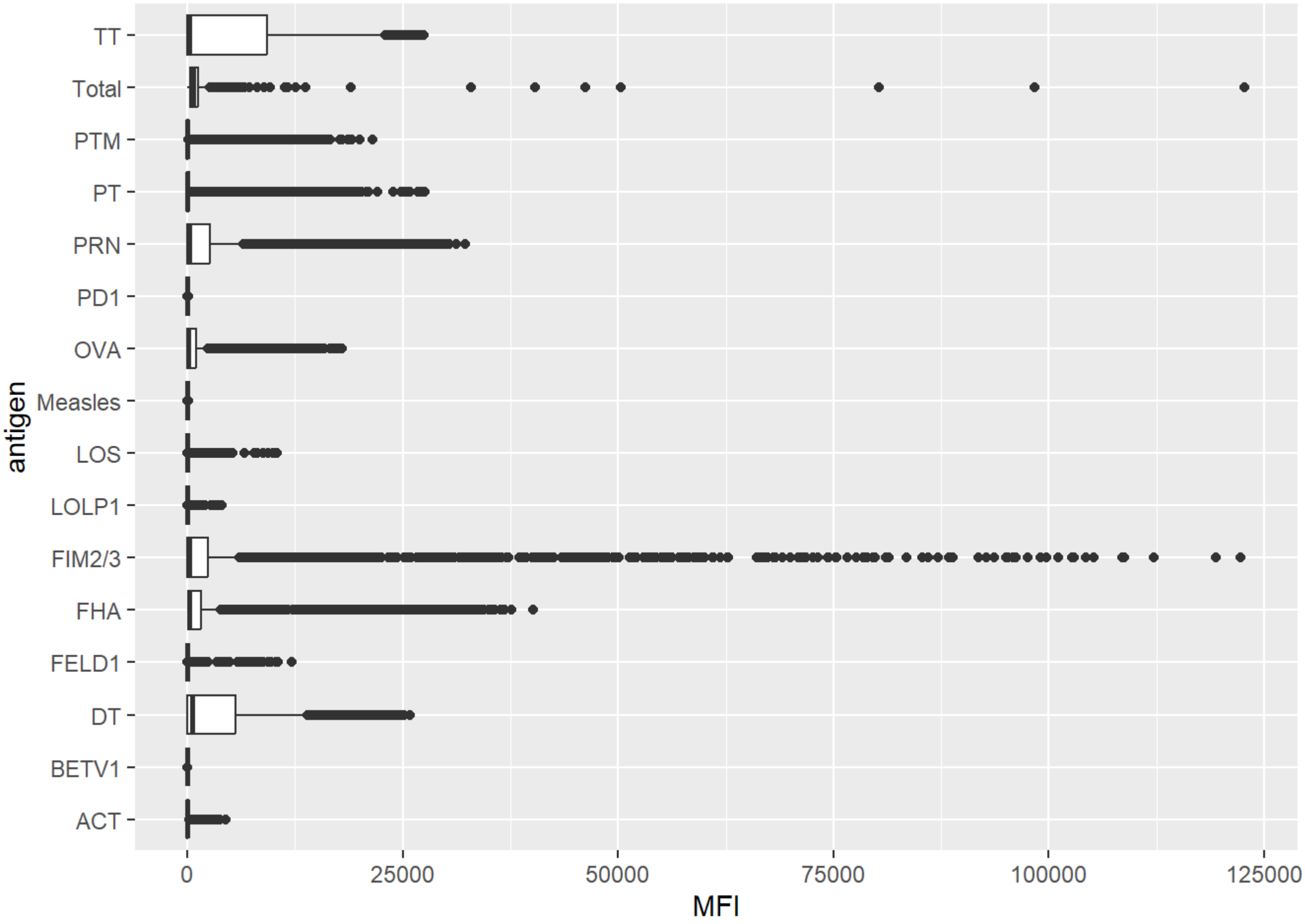
```
table(abdata$antigen)
```

```

    ACT   BETV1      DT   FELD1     FHA  FIM2/3   LOLP1     LOS Measles     OVA 
   1970    1970    3435    1970    3829    3435    1970    1970    1970    3435 
    PD1     PRN      PT     PTM   Total      TT 
   1970    3829    3829    1970     788    3435 
```

```
ggplot(abdata) +
  aes(MFI, antigen) +
  geom_boxplot()
```

```
Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```

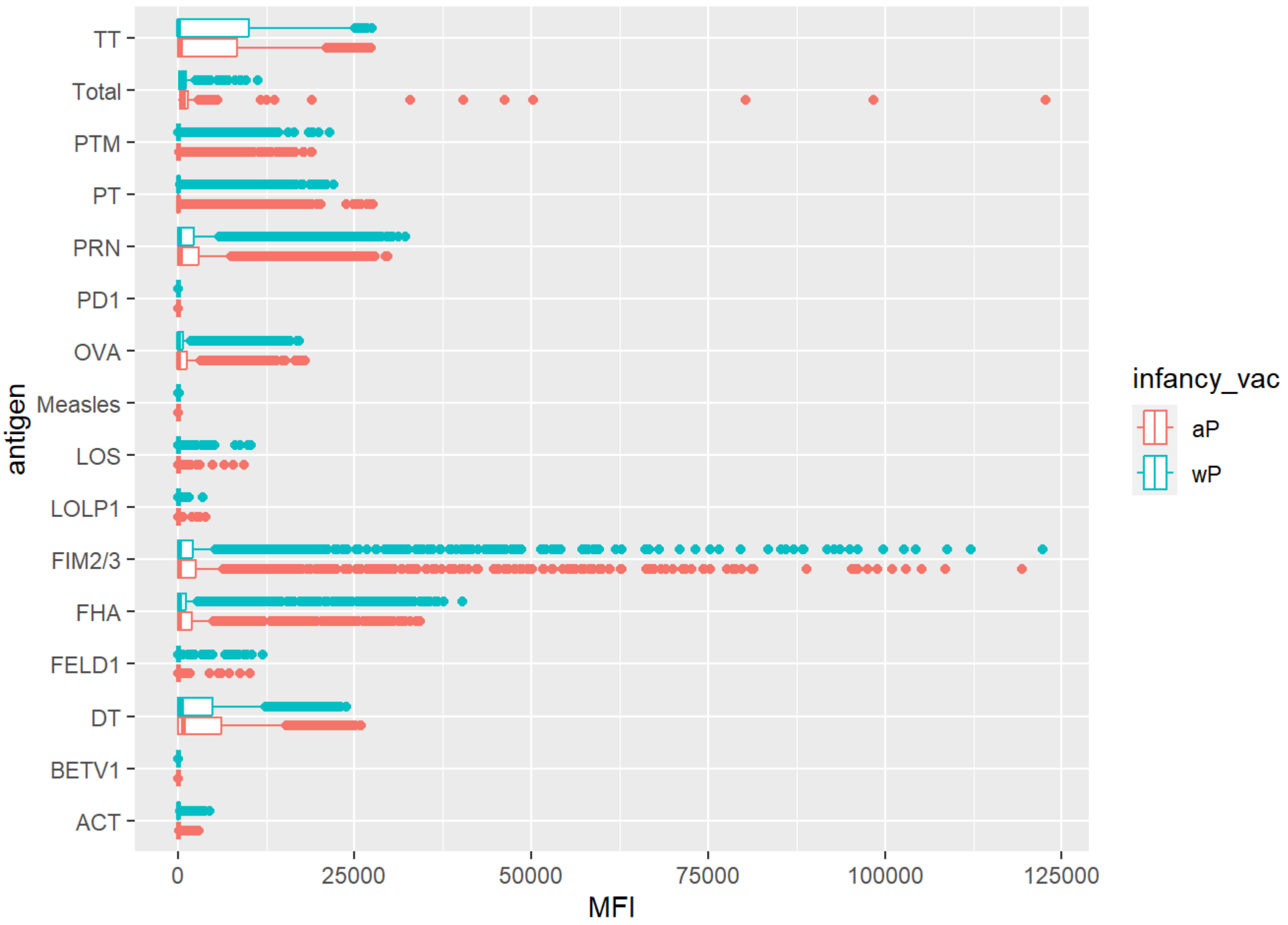


Why are certain antigens and not others very variable in their detected levels here?

Can you facet or even just color by infancy_vac? Is there some difference?

```
ggplot(abdata) +
  aes(MFI, antigen, col=infancy_vac) +
  geom_boxplot()
```

```
Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```



There are potentially some differences here but in general it is hard to tell with this whole dataset overview…

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

```
2020_dataset 2021_dataset 2022_dataset 
       31520         8085         2170 
```

Let’s focus in on just the 2021 dataset.

```
abdata.21 <- filter(abdata, dataset == "2021_dataset")
table(abdata.21$dataset)
```

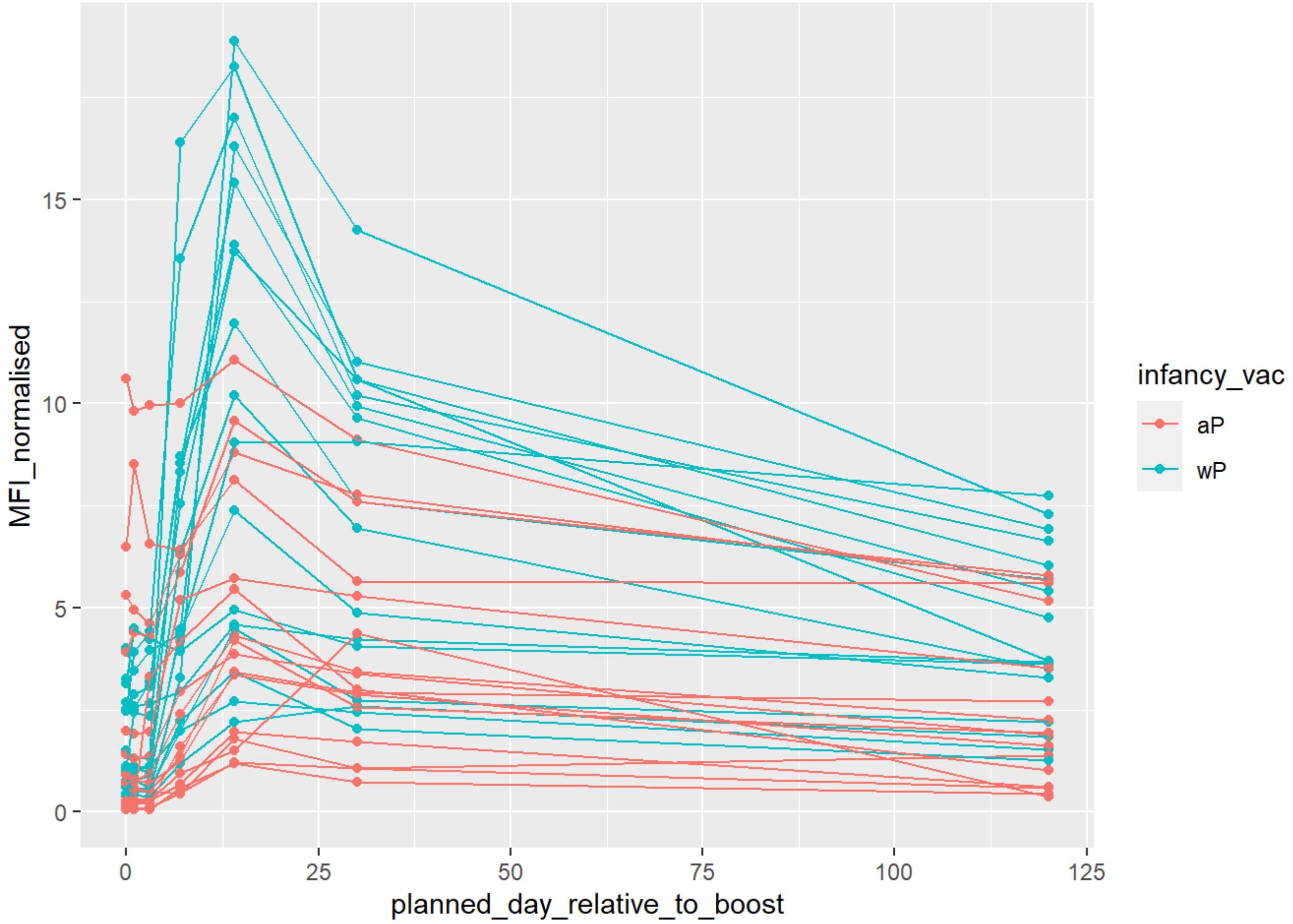
```
2021_dataset 
        8085 
```

Focus on PT antigen IgG levels

```
pt.21 <- filter(abdata.21, isotype == "IgG", antigen == "PT")
```

plot of days (time) relative to boost vs MFI levels

```r
ggplot(pt.21) +
    aes(x=planned_day_relative_to_boost,
        y=MFI_normalised,
        col=infancy_vac,
        group=subject_id) +
    geom_point() +
    geom_line()
```



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

The wP data is higher overall and reaches a taller peak than the aP responses, around day 12, but towards the end, past day 100, there is not a clear difference