

Intro to Data Science - HW 5

```
# Enter your name here: Victoria Haley
```

Copyright Jeffrey Stanton, Jeffrey Saltz, and Jasmina Tacheva

Attribution statement: (choose only one and delete the rest)

```
# 2. I did this homework with help from the book and the professor and these Internet sources: stackove
```

This module: Data visualization is important because many people can make sense of data more easily when it is presented in graphic form. As a data scientist, you will have to present complex data to decision makers in a form that makes the data interpretable for them. From your experience with Excel and other tools, you know that there are a variety of **common data visualizations** (e.g., pie charts). How many of them can you name?

The most powerful tool for data visualization in R is called **ggplot**. Written by computer/data scientist **Hadley Wickham**, this “**graphics grammar**” tool builds visualizations in layers. This method provides immense flexibility, but takes a bit of practice to master.

Step 1: Make a copy of the data

- A. Read the **who** dataset from this URL: <https://intro-datascience.s3.us-east-2.amazonaws.com/who.csv> into a new dataframe called **tb**.

Your new dataframe, **tb**, contains a so-called **multivariate time series**: a sequence of measurements on 23 Tuberculosis-related (TB) variables captured repeatedly over time (1980-2013). Familiarize yourself with the nature of the 23 variables by consulting the dataset’s codebook which can be found here: https://intro-datascience.s3.us-east-2.amazonaws.com/TB_data_dictionary_2021-02-06.csv.

```
library(readr)
tb <- data.frame(read.csv("https://intro-datascience.s3.us-east-2.amazonaws.com/who.csv"))
tbCodeBook <- data.frame(read.csv("https://intro-datascience.s3.us-east-2.amazonaws.com/TB_data_diction
```

- B. How often were these measurements taken (in other words, at what frequency were the variables measured)? Put your answer in a comment.

```
mean(rowSums(na.omit(tb[,2:23])))
```

```
## [1] 9208.751
```

```
#About 9,209 measurements were taken every year from 1980 - 2008
```

Step 2: Clean-up the NAs and create a subset

- A. Let’s clean up the **iso2** attribute in **tb**

Hint: use `is.na()` – well use `! is.na()`

```
! is.na(tb$iso2)
```

```
##      [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##      [13] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##      [25] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##      [37] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##      [49] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##      [61] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##      [73] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```
## [5269] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5281] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5293] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5305] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5317] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5329] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5341] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5353] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5365] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5377] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5389] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5401] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5413] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5425] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5437] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5449] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5461] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5473] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5485] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5497] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5509] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5521] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5533] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5545] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5557] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5569] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5581] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5593] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5605] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5617] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5629] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5641] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5653] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5665] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5677] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5689] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5701] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5713] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5725] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5737] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5749] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [5761] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

B. Create a subset of `tb` containing **only the records for Canada** (“CA” in the `iso2` variable). Save it in a new dataframe called `tbCan`. Make sure this new df has **29 observations and 23 variables**.

```
tbCan <- data.frame(subset(tb, tb$iso2 == "CA"))
```

C. A simple method for dealing with small amounts of **missing data** in a numeric variable is to **substitute the mean of the variable in place of each missing datum**. This expression locates (and reports to the console) all the missing data elements in the variable measuring the **number of positive pulmonary smear tests for male children 0-4 years old** (there are 26 data points missing)

```
tbCan$new_sp_m04[is.na(tbCan$new_sp_m04)]
```

```
## [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
## [26] NA
```

```
Error in eval(expr, envir, enclos): object 'tbCan' not found
Traceback:
```

D. Write a comment describing how that statement works.

```
#The statement checks for any NAs in the new pulmonary smear positive cases of males aged 0-4 in Canada
```

E. Write 4 more statements to check if there is missing data for the number of positive pulmonary smear tests for: **male and female** children 0-14 years old (**new_sp_m014** and **new_sp_f014**), and **male and female citizens 65 years of age and older**, respectively. What does empty output suggest about the number of missing observations?

```
tbCan$new_sp_f014[is.na(tbCan$new_sp_f014)]
```

```
## integer(0)
```

```
tbCan$new_sp_m014[is.na(tbCan$new_sp_m014)]
```

```
## integer(0)
```

```
tbCan$new_sp_f65[is.na(tbCan$new_sp_f65)]
```

```
## integer(0)
```

```
tbCan$new_sp_m65[is.na(tbCan$new_sp_m65)]
```

```
## integer(0)
```

```
#The empty output suggests that there is no missing data (NAs) for positive pulmonary smear tests for m
```

There is an R package called **imputeTS** specifically designed to repair missing values in time series data. We will use this instead of mean substitution. The **na_interpolation()** function in this package takes advantage of a unique characteristic of time series data: **neighboring points in time can be used to “guess” about a missing value in between.**

F. Install the **imputeTS** package (if needed) and use **na_interpolation()** on the variable from part C. Don't forget that you need to save the results back to the **tbCan** dataframe. Also update any attribute discussed in part E (if needed).

```
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
```

```
##   method      from
```

```
## as.zoo.data.frame zoo
```

```
tbCan$new_sp_m04 <- na_interpolation(tbCan$new_sp_m04)
```

G. Rerun the code from C and E above to check that all missing data have been fixed.

```
tbCan$new_sp_m04[is.na(tbCan$new_sp_m04)]
```

```
## numeric(0)
```

```
tbCan$new_sp_f014[is.na(tbCan$new_sp_f014)]
```

```
## integer(0)
```

```
tbCan$new_sp_m014[is.na(tbCan$new_sp_m014)]
```

```
## integer(0)
```

```
tbCan$new_sp_f65[is.na(tbCan$new_sp_f65)]
```

```
## integer(0)
```

```
tbCan$new_sp_m65[is.na(tbCan$new_sp_m65)]
```

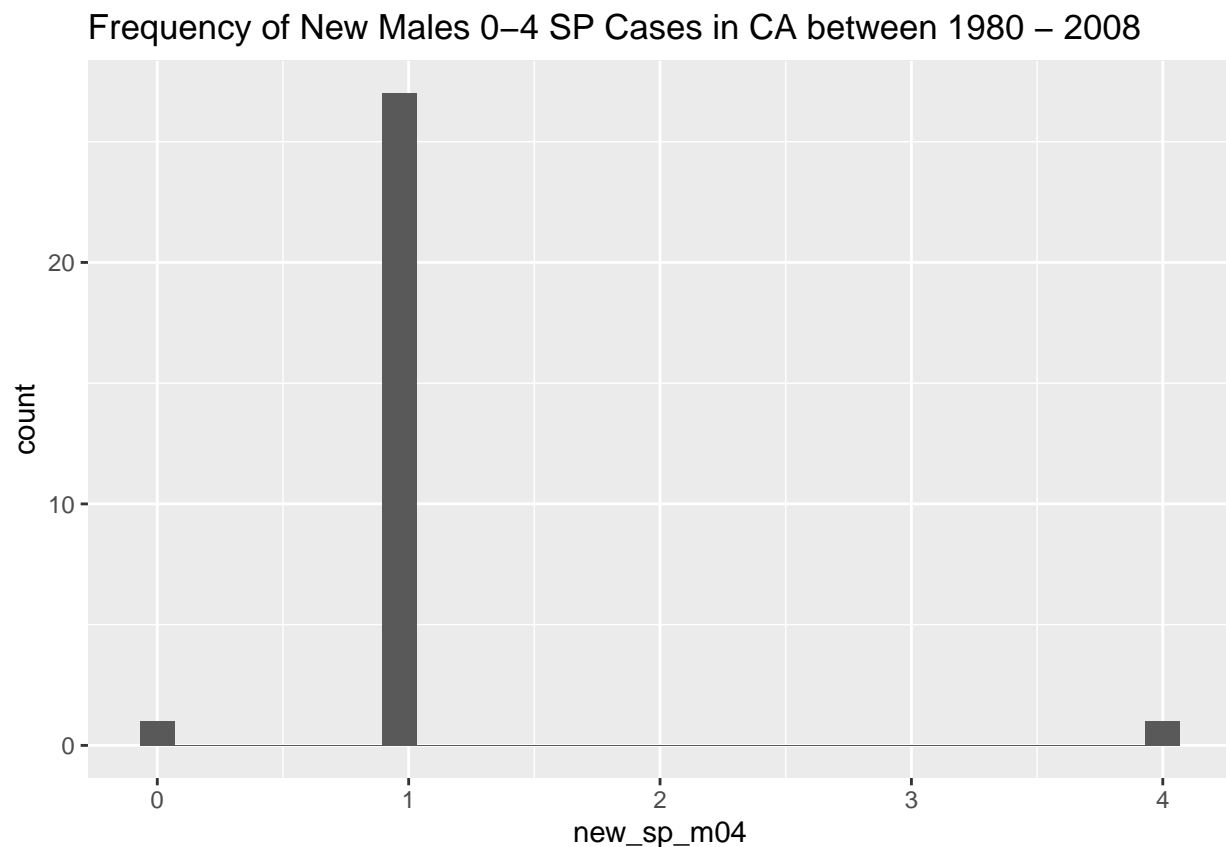
```
## integer(0)
```

Step 3: Use ggplot to explore the distribution of each variable

Don't forget to install and library the **ggplot2** package. Then: H. Create a histogram for **new_sp_m014**. Be sure to add a title and briefly describe what the histogram means in a comment.

```
library(ggplot2)
ggplot(tbCan, aes(x=new_sp_m04)) + geom_histogram() +
  labs(title = "Frequency of New Males 0-4 SP Cases in CA between 1980 - 2008")
```

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



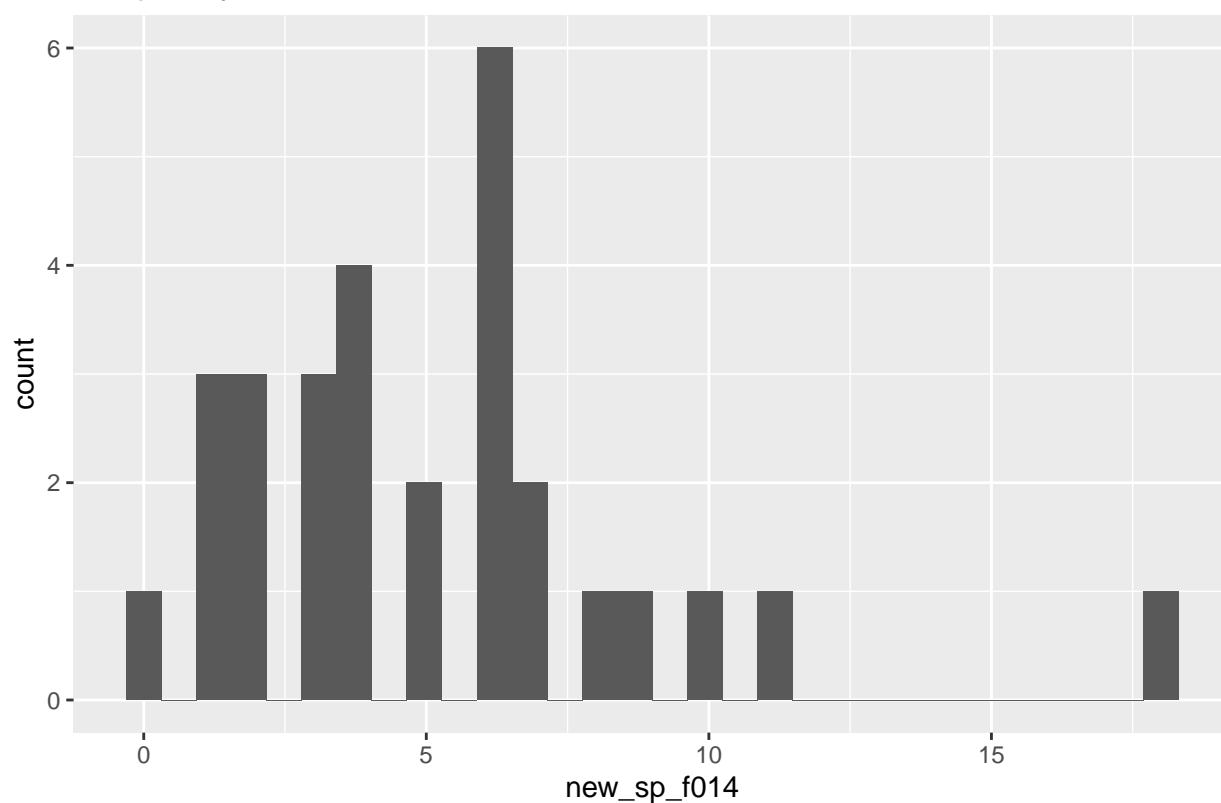
#This histogram shows that there was mostly 1 new pulmonary smear positive case in males aged 0-4 in Ca

- I. Create histograms (using ggplot) of each of the other three variables from E with ggplot(). Which parameter do you need to adjust to make the other histograms look right?

```
ggplot(tbCan, aes(x=new_sp_f014)) + geom_histogram() +
  labs(title = "Frequency of New Females 0-14 SP Cases in CA between 1980 - 2008")
```

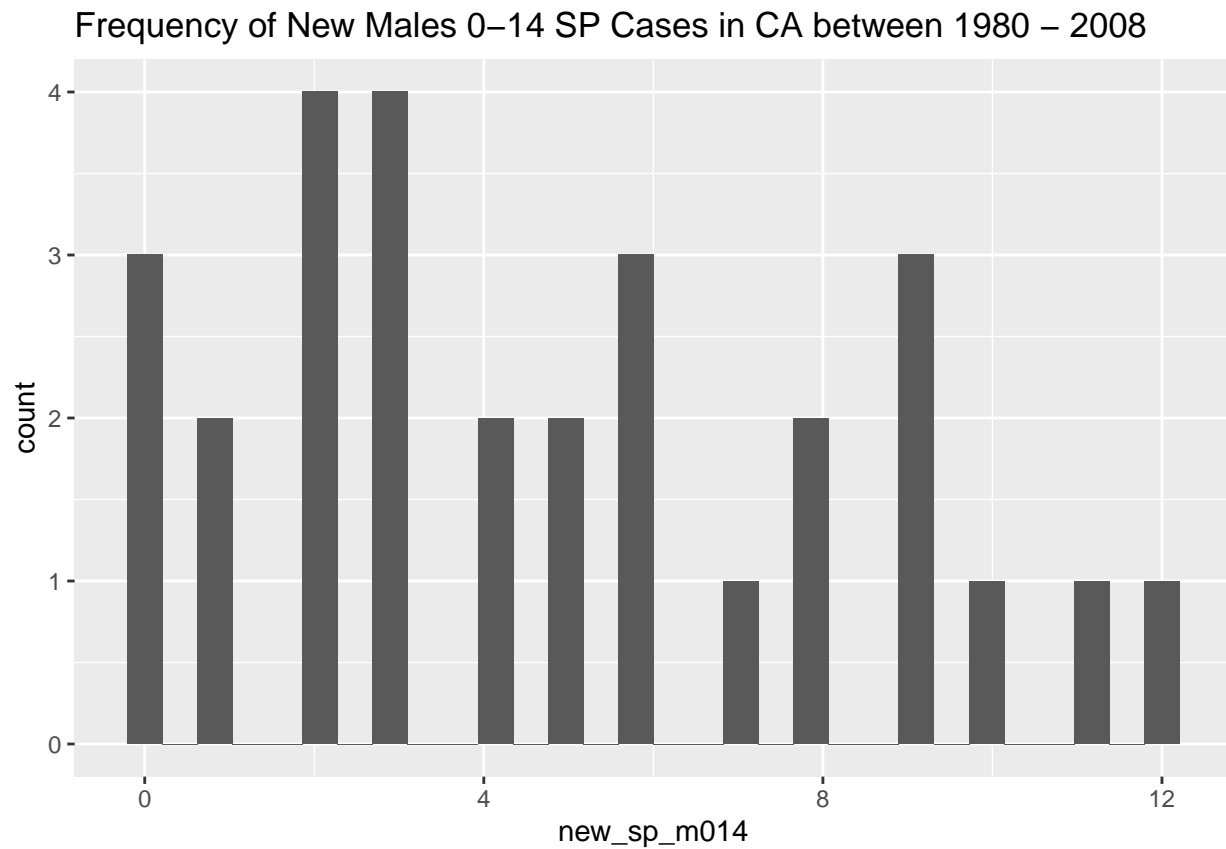
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Frequency of New Females 0–14 SP Cases in CA between 1980 – 2008



```
ggplot(tbCan, aes(x=new_sp_m014)) + geom_histogram() +  
  labs(title = "Frequency of New Males 0-14 SP Cases in CA between 1980 - 2008")
```

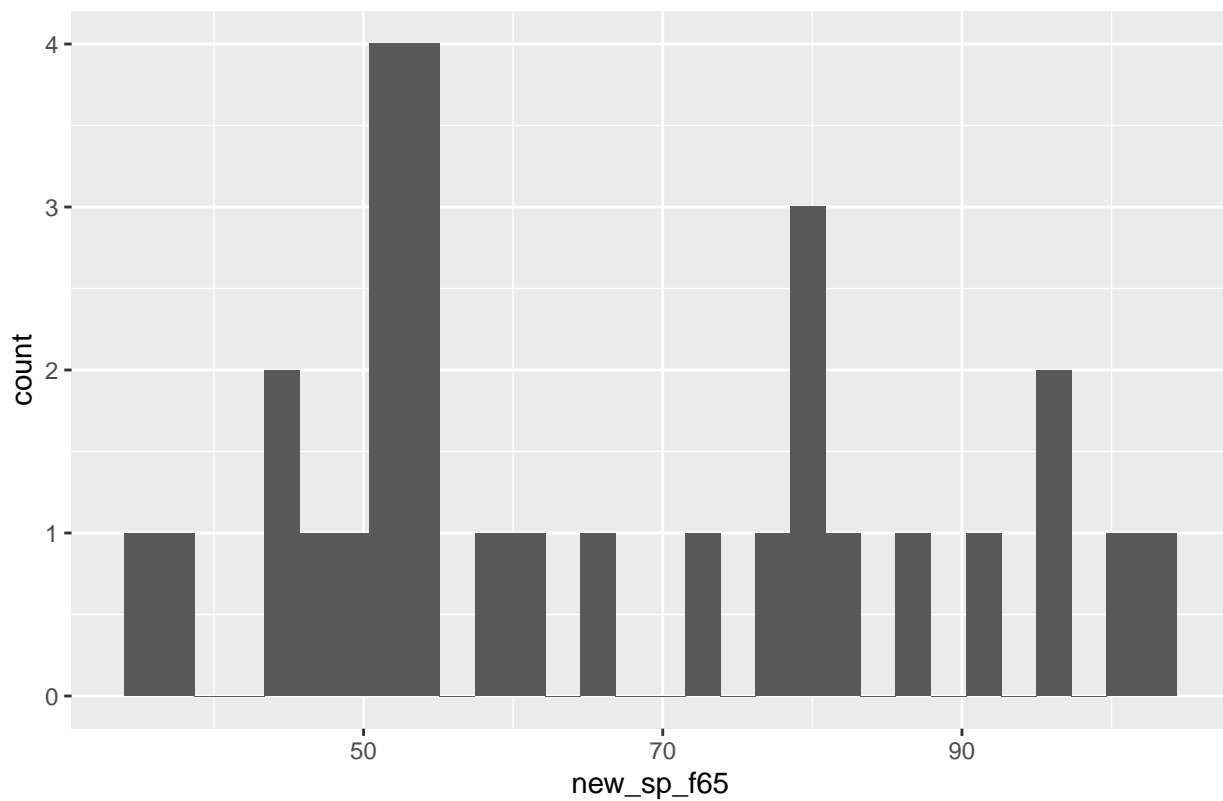
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(tbCan, aes(x=new_sp_f65)) + geom_histogram() +  
  labs(title = "Frequency of New Females 65+ SP Cases in CA between 1980 - 2008")
```

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

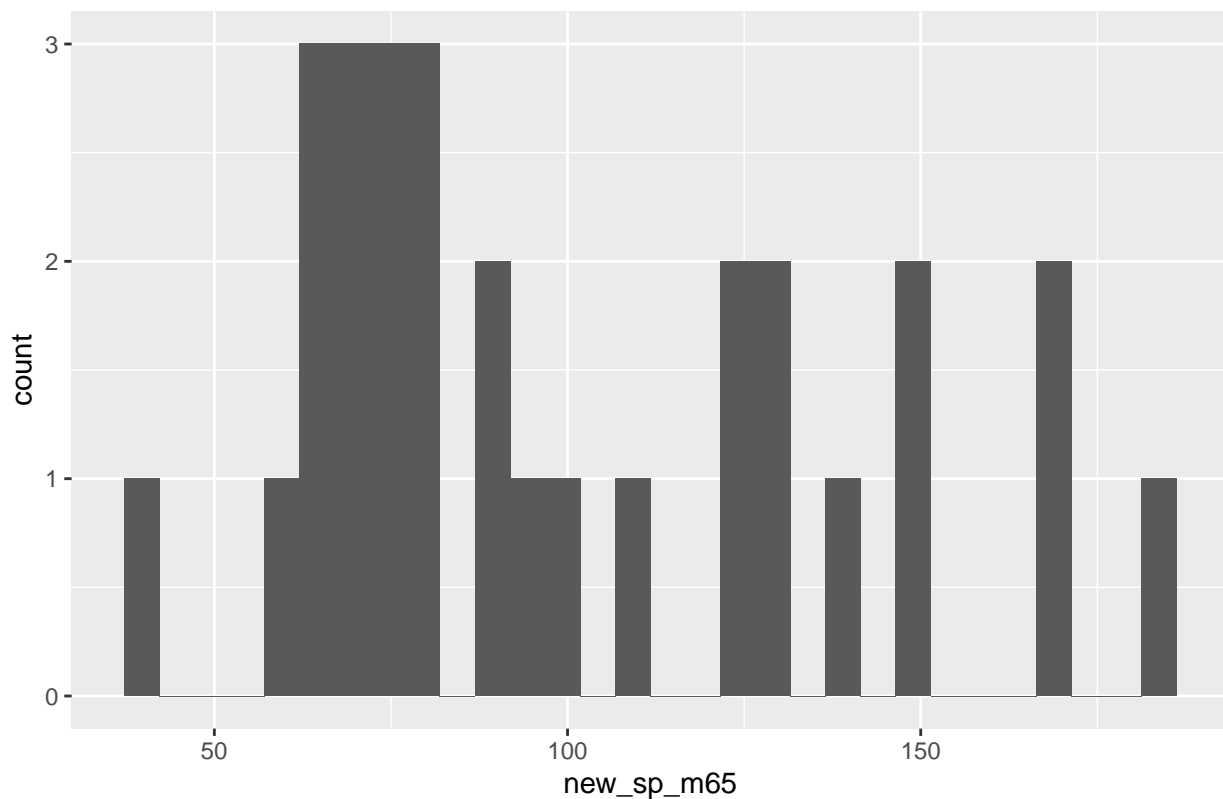
Frequency of New Females 65+ SP Cases in CA between 1980 – 2008



```
ggplot(tbCan, aes(x=new_sp_m65)) + geom_histogram() +  
  labs(title = "Frequency of New Males 65+ SP Cases in CA between 1980 - 2008")
```

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

Frequency of New Males 65+ SP Cases in CA between 1980 – 2008

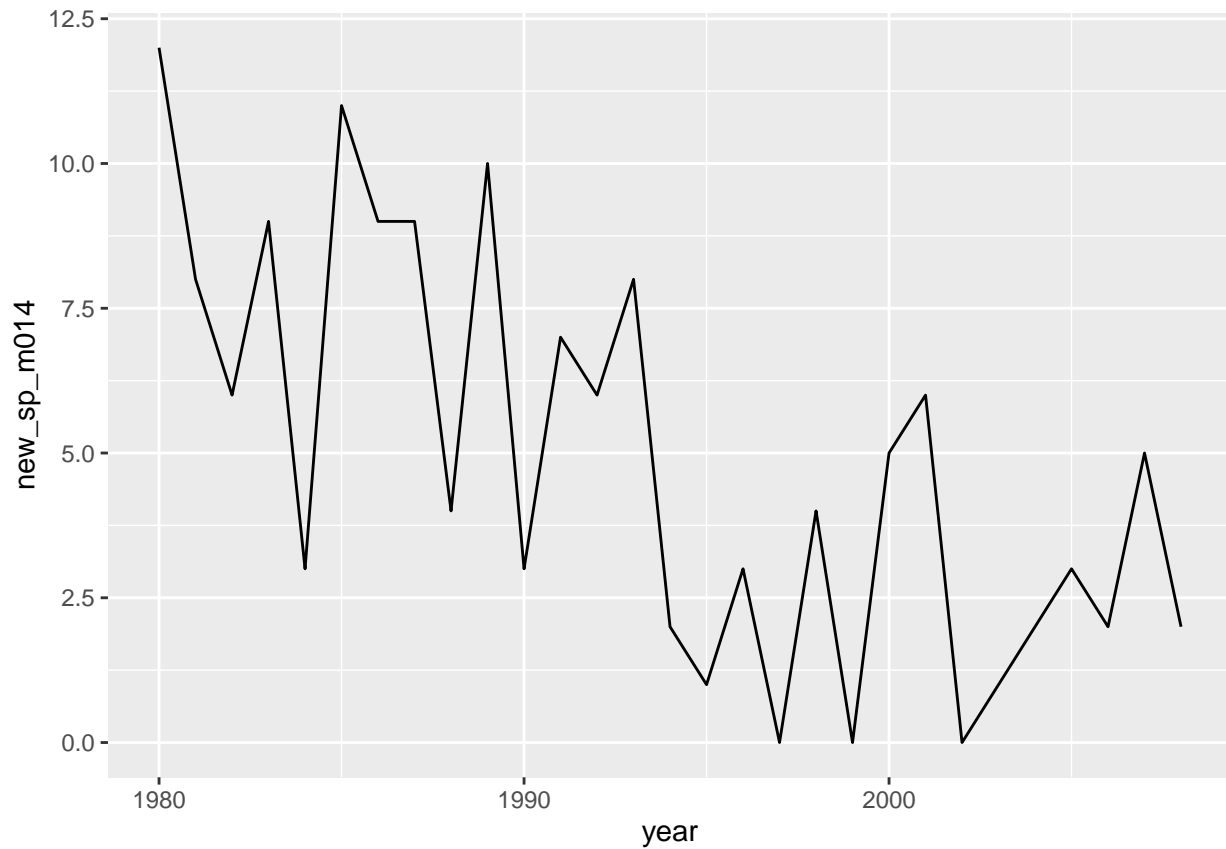


#The binwidth should be adjusted so that the histograms look right

Step 4: Explore how the data changes over time

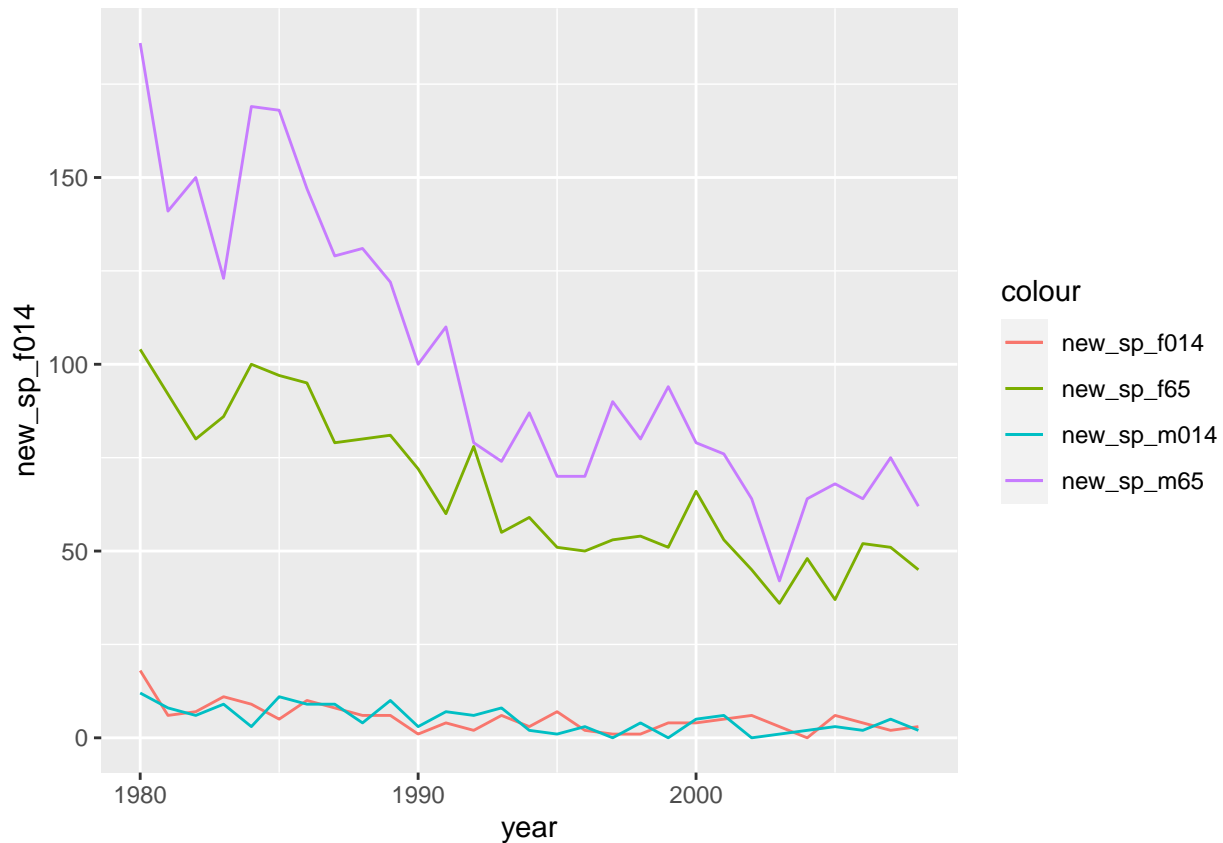
J. These data were collected in a period of several decades (1980-2013). You can thus observe changes over time with the help of a line chart. Create a **line chart**, with **year** on the X-axis and **new_sp_m014** on the Y-axis.

```
ggplot(tbCan, aes(x=year, y=new_sp_m014)) + geom_line()
```

K. Next, create similar graphs for each of the other three variables. Change the **color** of the line plots (any color you want).

```
ggplot(tbCan, aes(x=year)) +  
  geom_line(aes(y=new_sp_f014, color="new_sp_f014")) +  
  geom_line(aes(y=new_sp_m014, color="new_sp_m014")) +  
  geom_line(aes(y=new_sp_f65, color="new_sp_f65")) +  
  geom_line(aes(y=new_sp_m65, color="new_sp_m65"))
```

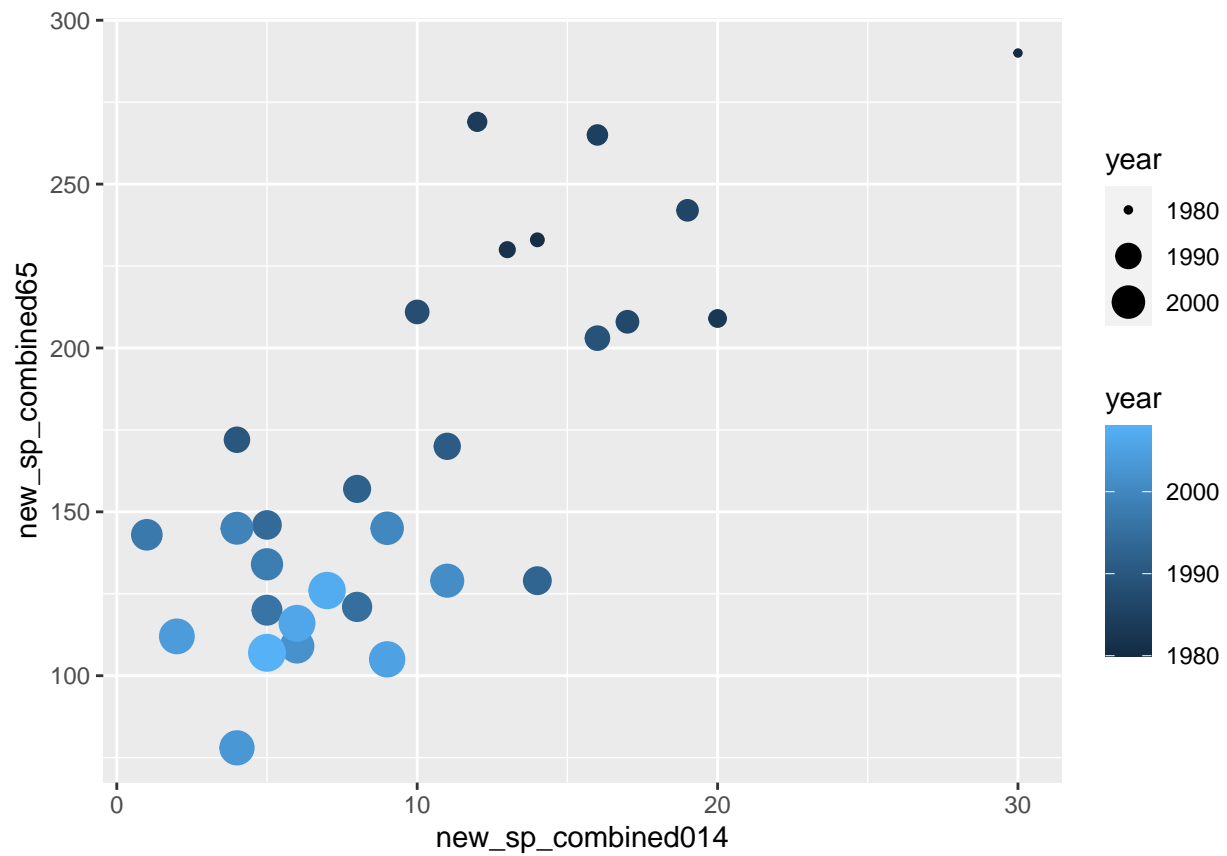


L. Using vector math, create a new variable by combining the numbers from **new_sp_m014** and **new_sp_f014**. Save the resulting vector as a new variable in the **tbCan** df called **new_sp_combined014**. This new variable represents the number of positive pulmonary smear tests for male AND female children between the ages of 0 and 14 years of age. Do the same for SP tests **among citizens 65 years of age and older** and save the resulting vector in the **tbCan** variable called **new_sp_combined65**.

```
tbCan$new_sp_combined014 <- rowSums(cbind(tbCan$new_sp_f014, tbCan$new_sp_m014))
tbCan$new_sp_combined65 <- rowSums(cbind(tbCan$new_sp_f65, tbCan$new_sp_m65))
```

M. Finally, create a **scatter plot**, showing **new_sp_combined014** on the x axis, **new_sp_combined65** on the y axis, and having the **color and size** of the point represent **year**.

```
ggplot(tbCan, aes(x=new_sp_combined014, y=new_sp_combined65)) + geom_point(aes(size=year, color=year))
```



N. Interpret this visualization – what insight does it provide?

#This scatterplot shows that the number of new positive smear cases between male and females in both 0-