Intro to Data Science - HW 5

Copyright Jeffrey Stanton, Jeffrey Saltz, and Jasmina Tacheva

```
# Enter your name here: Benjamin Tisinger
```

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

```
# 1. I did this homework by myself, with help from the book and the professor.
```

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 (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 (https://intro-datascience.s3.us-east-2.amazonaws.com/TB_data_dictionary_2021-02-06.csv).

```r
library(dbplyr)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::ident()  masks dbplyr::ident()
## ✕ dplyr::lag()    masks stats::lag()
## ✕ dplyr::sql()    masks dbplyr::sql()
```

```r
tb <- read.csv('https://intro-datascience.s3.us-east-2.amazonaws.com/who.csv')
head(tb,15)
```

```
##    iso2 year new_sp new_sp_m04 new_sp_m514 new_sp_m014 new_sp_m1524
## 1    AD 1989     NA         NA          NA          NA           NA
## 2    AD 1990     NA         NA          NA          NA           NA
## 3    AD 1991     NA         NA          NA          NA           NA
## 4    AD 1992     NA         NA          NA          NA           NA
## 5    AD 1993     15         NA          NA          NA           NA
## 6    AD 1994     24         NA          NA          NA           NA
## 7    AD 1996      8         NA          NA           0            0
## 8    AD 1997     17         NA          NA           0            0
## 9    AD 1998      1         NA          NA           0            0
## 10   AD 1999      4         NA          NA           0            0
## 11   AD 2000      1         NA          NA           0            0
## 12   AD 2001      3         NA          NA           0           NA
## 13   AD 2002      2         NA          NA           0            0
## 14   AD 2003      7         NA          NA           0            0
## 15   AD 2004      3         NA          NA           0            0
##    new_sp_m2534 new_sp_m3544 new_sp_m4554 new_sp_m5564 new_sp_m65 new_sp_mu
## 1            NA           NA           NA           NA         NA        NA
## 2            NA           NA           NA           NA         NA        NA
## 3            NA           NA           NA           NA         NA        NA
## 4            NA           NA           NA           NA         NA        NA
## 5            NA           NA           NA           NA         NA        NA
## 6            NA           NA           NA           NA         NA        NA
## 7             0            4            1            0          0        NA
## 8             1            2            2            1          6        NA
## 9             0            1            0            0          0        NA
## 10            0            1            1            0          0        NA
## 11            1            0            0            0          0        NA
## 12           NA            2            1           NA         NA        NA
## 13            0            1            0            0          0        NA
## 14            0            1            2            0          0        NA
## 15            0            1            1            0          0        NA
##    new_sp_f04 new_sp_f514 new_sp_f014 new_sp_f1524 new_sp_f2534 new_sp_f3544
## 1          NA          NA          NA           NA           NA           NA
## 2          NA          NA          NA           NA           NA           NA
## 3          NA          NA          NA           NA           NA           NA
## 4          NA          NA          NA           NA           NA           NA
## 5          NA          NA          NA           NA           NA           NA
## 6          NA          NA          NA           NA           NA           NA
## 7          NA          NA           0            1            1            0
## 8          NA          NA           0            1            2            3
## 9          NA          NA          NA           NA           NA           NA
## 10         NA          NA           0            0            0            1
## 11         NA          NA          NA           NA           NA           NA
## 12         NA          NA          NA           NA           NA           NA
## 13         NA          NA           0            1            0            0
## 14         NA          NA           0            1            1            1
## 15         NA          NA           0            0            1            0
##    new_sp_f4554 new_sp_f5564 new_sp_f65 new_sp_fu
## 1            NA           NA         NA        NA
## 2            NA           NA         NA        NA
## 3            NA           NA         NA        NA
```

```
## 4          NA          NA        NA        NA
## 5          NA          NA        NA        NA
## 6          NA          NA        NA        NA
## 7           0           1         0        NA
## 8           0           0         1        NA
## 9          NA          NA        NA        NA
## 10          0           0         0        NA
## 11         NA          NA        NA        NA
## 12         NA          NA        NA        NA
## 13          0           0         0        NA
## 14          0           0         0        NA
## 15          0           0         0        NA
```

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

```
min(tb$year)
```

```
## [1] 1980
```

```
max(tb$year)
```

```
## [1] 2008
```

```
#The measurements are taken at a frequency of every year starting at 1980 until 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()*

```
tb <- tb[!is.na(tb$iso2),]
head(tb,5)
```

```
##   iso2 year new_sp new_sp_m04 new_sp_m514 new_sp_m014 new_sp_m1524 new_sp_m2534
## 1   AD 1989     NA         NA          NA          NA           NA           NA
## 2   AD 1990     NA         NA          NA          NA           NA           NA
## 3   AD 1991     NA         NA          NA          NA           NA           NA
## 4   AD 1992     NA         NA          NA          NA           NA           NA
## 5   AD 1993     15         NA          NA          NA           NA           NA
##   new_sp_m3544 new_sp_m4554 new_sp_m5564 new_sp_m65 new_sp_mu new_sp_f04
## 1           NA           NA           NA         NA        NA         NA
## 2           NA           NA           NA         NA        NA         NA
## 3           NA           NA           NA         NA        NA         NA
## 4           NA           NA           NA         NA        NA         NA
## 5           NA           NA           NA         NA        NA         NA
##   new_sp_f514 new_sp_f014 new_sp_f1524 new_sp_f2534 new_sp_f3544 new_sp_f4554
## 1          NA          NA           NA           NA           NA           NA
## 2          NA          NA           NA           NA           NA           NA
## 3          NA          NA           NA           NA           NA           NA
## 4          NA          NA           NA           NA           NA           NA
## 5          NA          NA           NA           NA           NA           NA
##   new_sp_f5564 new_sp_f65 new_sp_fu
## 1           NA         NA        NA
## 2           NA         NA        NA
## 3           NA         NA        NA
## 4           NA         NA        NA
## 5           NA         NA        NA
```

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 <- subset(tb, tb$iso2 =='CA')
head(tbCan,5)
```

```
##     iso2 year new_sp new_sp_m04 new_sp_m514 new_sp_m014 new_sp_m1524
## 872   CA 1980    951         NA          NA          12           54
## 873   CA 1981    803         NA          NA           8           49
## 874   CA 1982    812         NA          NA           6           52
## 875   CA 1983    771         NA          NA           9           47
## 876   CA 1984    811         NA          NA           3           44
##     new_sp_m2534 new_sp_m3544 new_sp_m4554 new_sp_m5564 new_sp_m65 new_sp_mu
## 872           75           83          100          108        186        NA
## 873           61           64           87          103        141        NA
## 874           66           69           90           91        150        NA
## 875           63           62           90           92        123        NA
## 876           75           58           68           83        169        NA
##     new_sp_f04 new_sp_f514 new_sp_f014 new_sp_f1524 new_sp_f2534 new_sp_f3544
## 872         NA          NA          18           62           51           34
## 873         NA          NA           6           46           57           26
## 874         NA          NA           7           51           57           30
## 875         NA          NA          11           50           50           29
## 876         NA          NA           9           51           59           28
##     new_sp_f4554 new_sp_f5564 new_sp_f65 new_sp_fu
## 872           31           33        104        NA
## 873           28           35         92        NA
## 874           25           38         80        NA
## 875           24           35         86        NA
## 876           28           36        100        NA
```

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 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.

```
#This expressions is finding all the missing data located in tbCan and is filling it with NA so
 the database looks and behaves cleaner
```

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?

```
youngdudes <- tbCan$new_sp_m014[is.na(tbCan$new_sp_m014)]
youngfemales <- tbCan$new_sp_f014[is.na(tbCan$new_sp_f014)]

olddudes <- tbCan$new_sp_m65[is.na(tbCan$new_sp_m65)]
oldfemales <- tbCan$new_sp_f65[is.na(tbCan$new_sp_f65)]

head(youngdudes,5)
```

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

```
head(youngfemales,5)
```

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

```
head(olddudes,5)
```

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

```
head(oldfemales,5)
```

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

```
#An output of integer(0) simply means there is No NA/Missing data in these sets
```

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')
```

```
## Warning: package 'imputeTS' was built under R version 4.2.2
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

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

tbCan$new_sp_m014 <- na_interpolation(tbCan$new_sp_m014)
tbCan$new_sp_f014 <- na_interpolation(tbCan$new_sp_f014)

tbCan$new_sp_m65  <- na_interpolation(tbCan$new_sp_m65)
tbCan$new_sp_f65  <- na_interpolation(tbCan$new_sp_f65)
```

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

```
youngdudes <- tbCan$new_sp_m014[is.na(tbCan$new_sp_m014)]
youngfemales <- tbCan$new_sp_f014[is.na(tbCan$new_sp_f014)]

olddudes <- tbCan$new_sp_m65[is.na(tbCan$new_sp_m65)]
oldfemales <- tbCan$new_sp_f65[is.na(tbCan$new_sp_f65)]

head(youngdudes,5)
```

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

```
head(youngfemales,5)
```

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

```
head(olddudes,5)
```

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

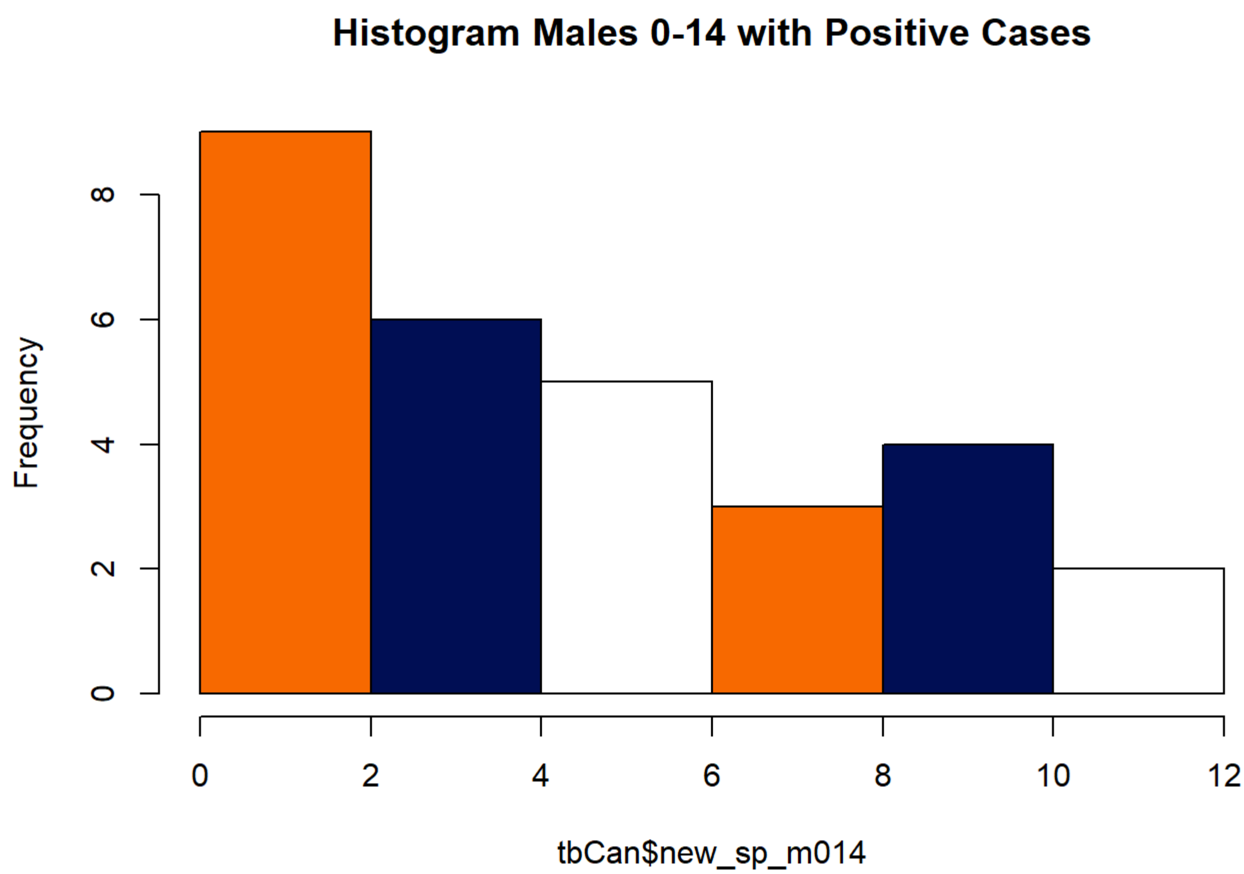
```
head(oldfemales,5)
```

```
## 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.

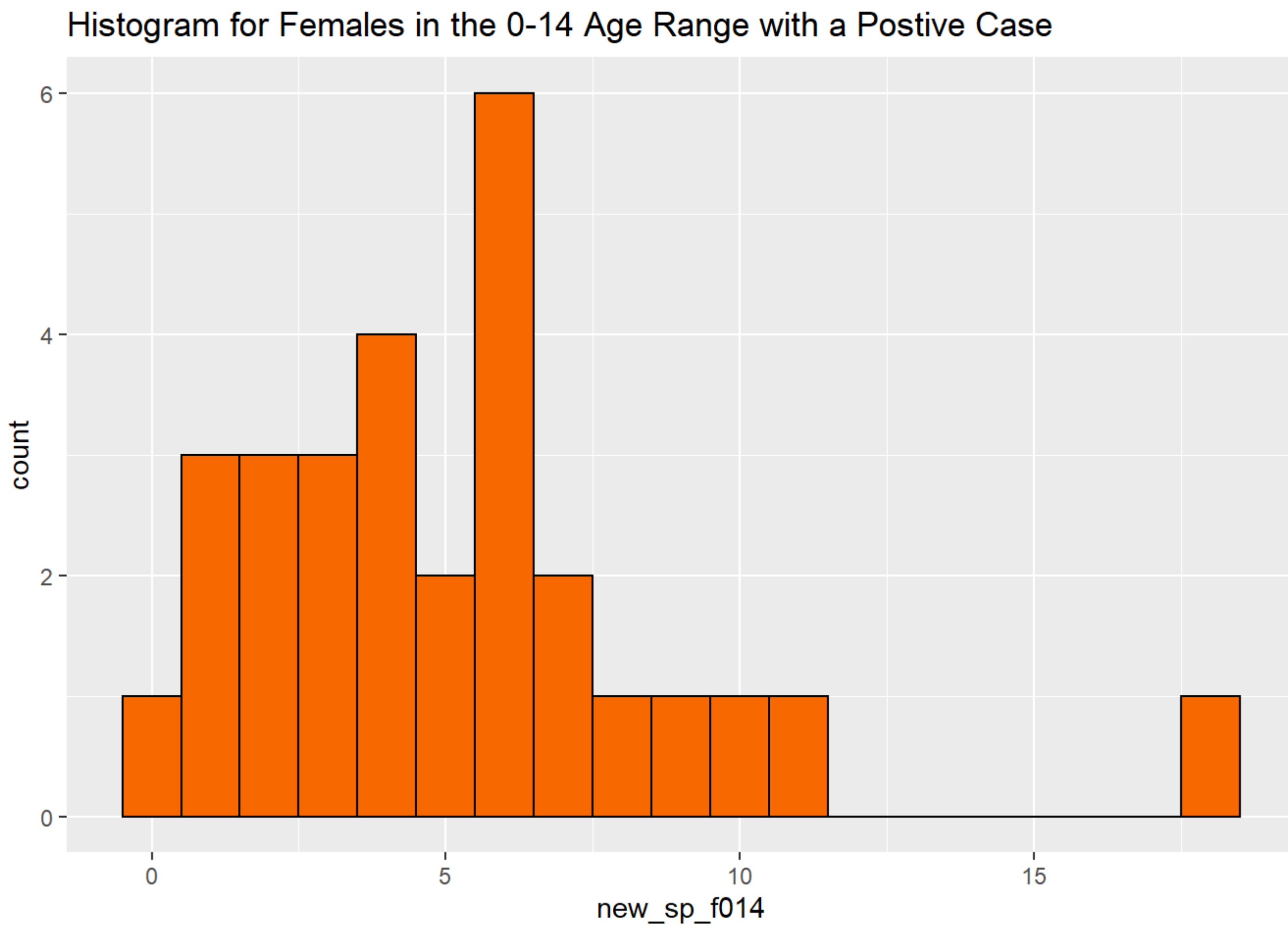
```
library(ggplot2)

hist(tbCan$new_sp_m014, main="Histogram Males 0-14 with Positive Cases",
     col = c("#F76900","#000E54","#FFFFFF"))
```



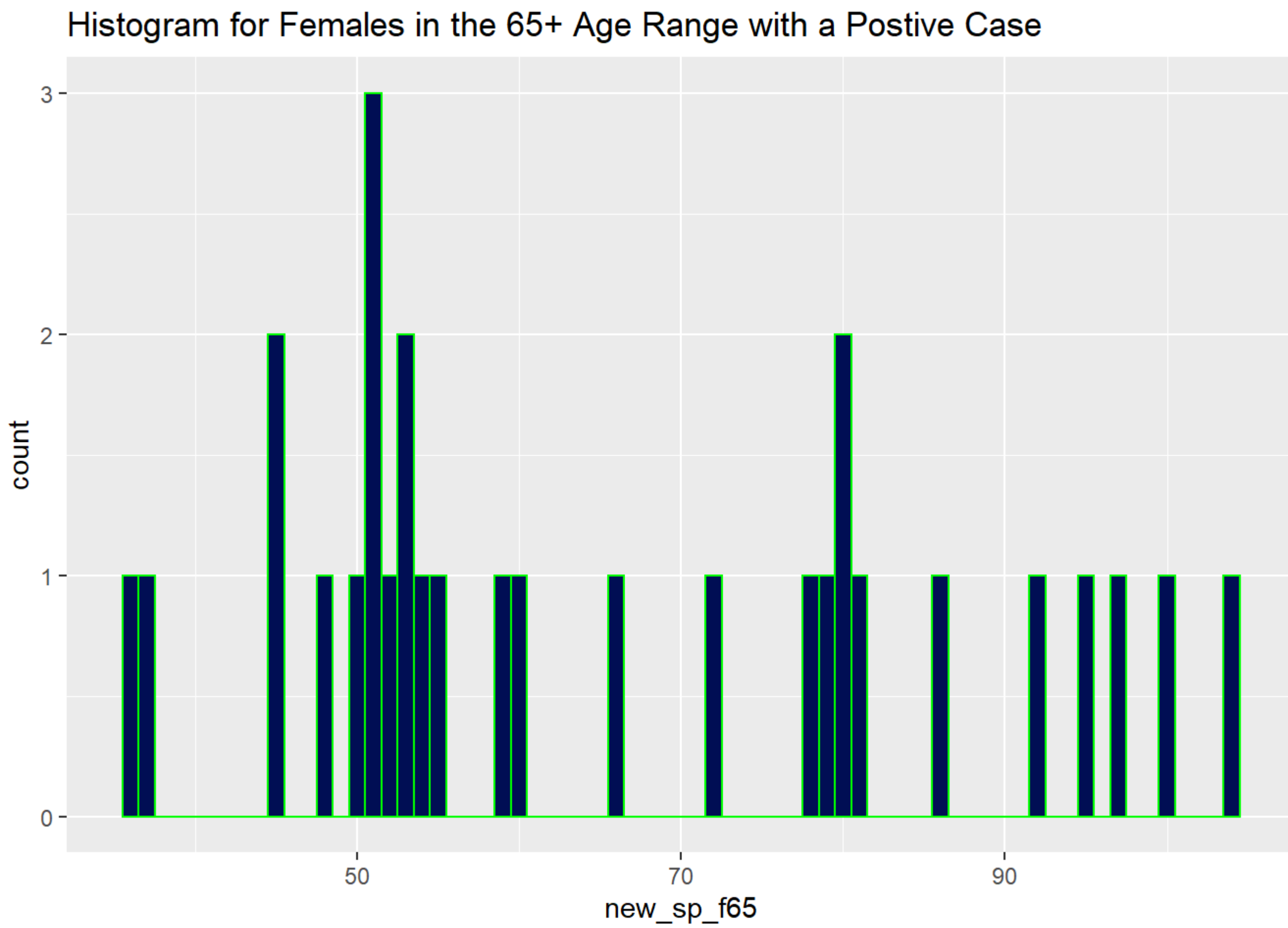

```
#histogram of Male positive cases. This set of data is not very good for measurement
```

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?

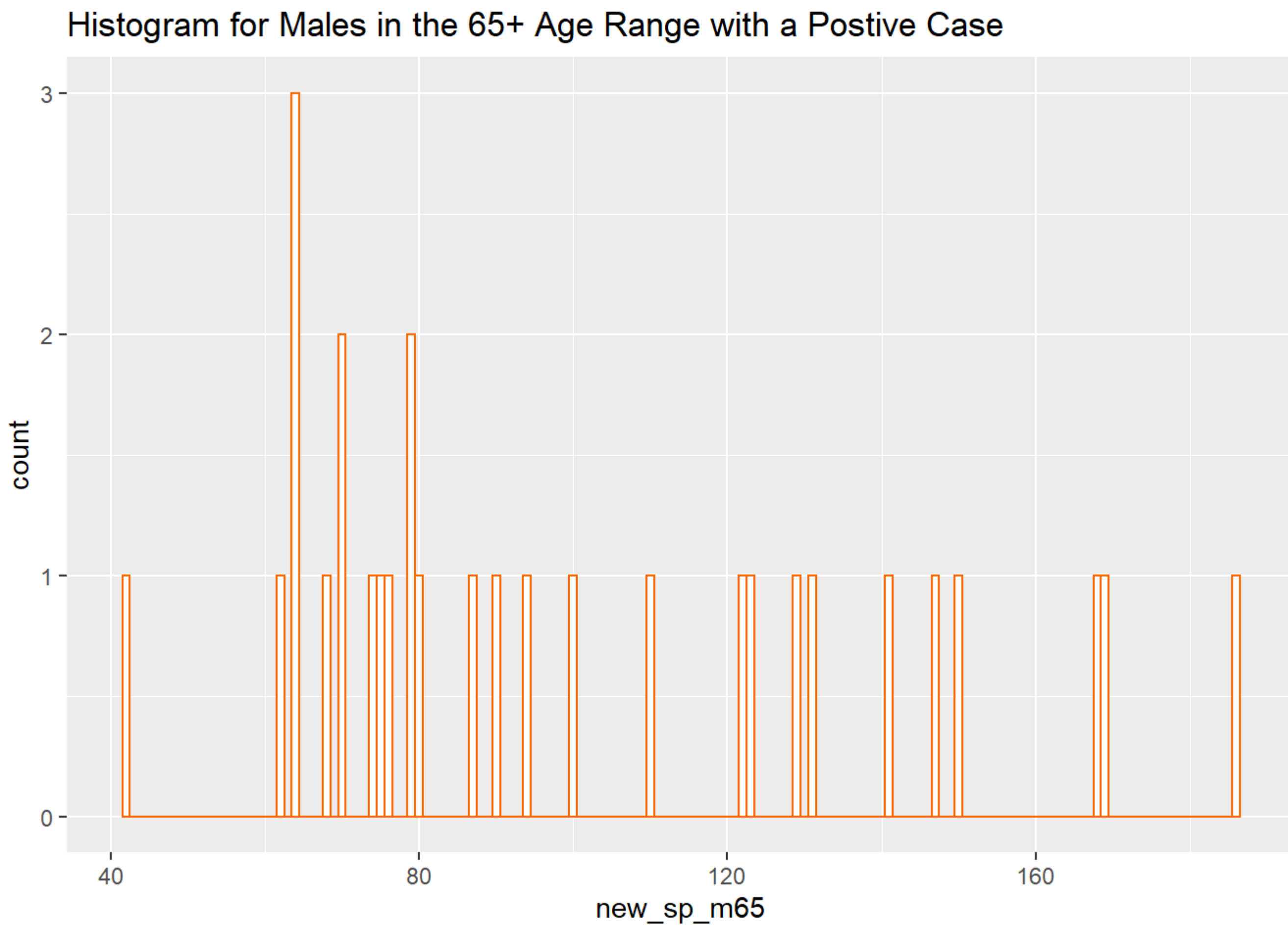
```
tbCan %>% ggplot() +
  geom_histogram(binwidth = 1,
                 fill="#F76900",
                 color="black",
                 aes(x=new_sp_f014)) +
                 ggtitle('Histogram for Females in the 0-14 Age Range with a Postive Case')
```




```
tbCan %>% ggplot() +
  geom_histogram(binwidth = 1,
                 fill="#000E54",
                 color="green",
                 aes(x=new_sp_f65)) +
                 ggtitle('Histogram for Females in the 65+ Age Range with a Postive Case')
```



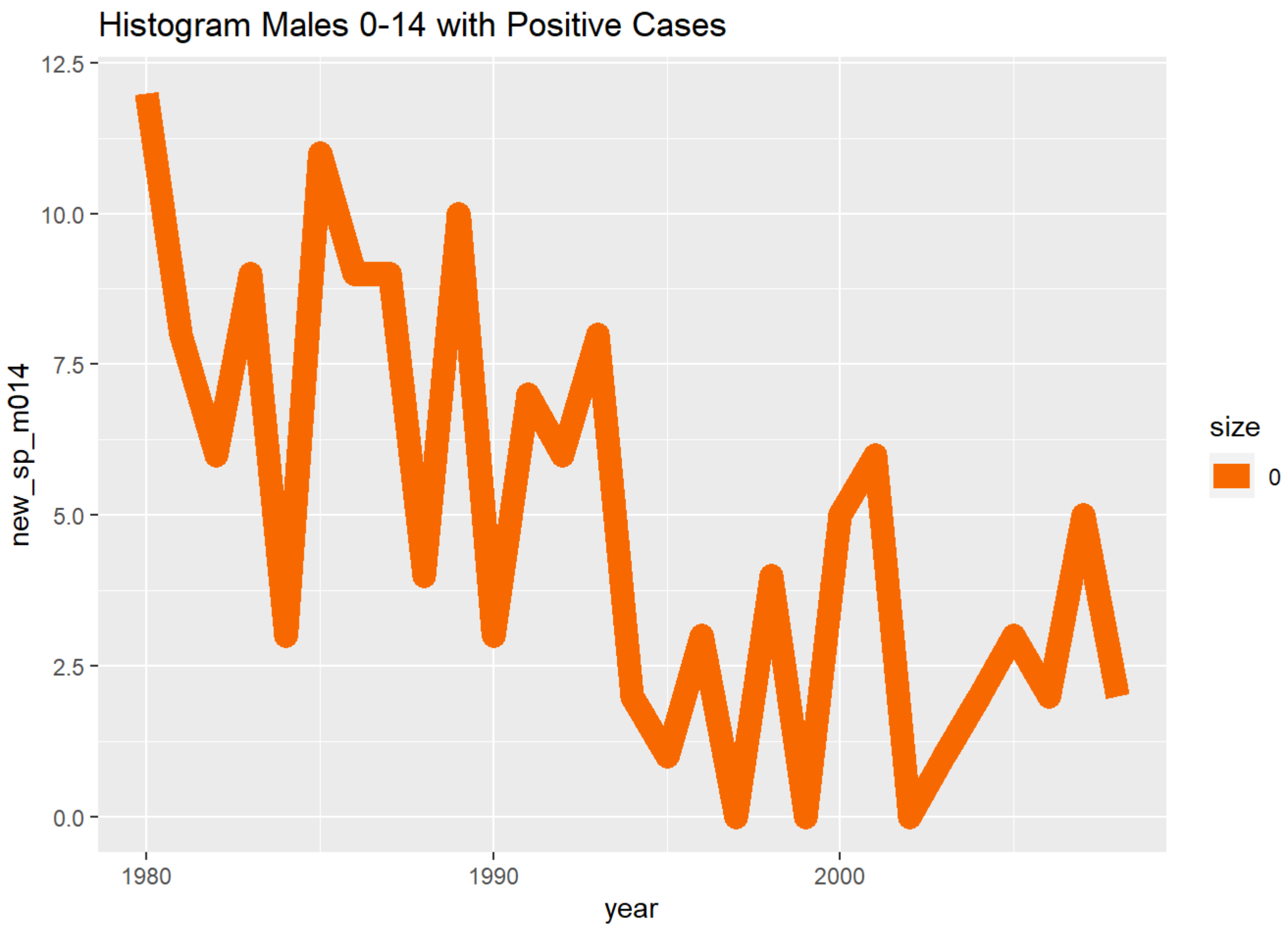

```
tbCan %>% ggplot() +
  geom_histogram(binwidth = 1,
                 fill="#FFFFFF",
                 color="#F76900",
                 aes(x=new_sp_m65)) +
                 ggtitle('Histogram for Males in the 65+ Age Range with a Postive Case')
```



Step 4: Explore how the data changes over time

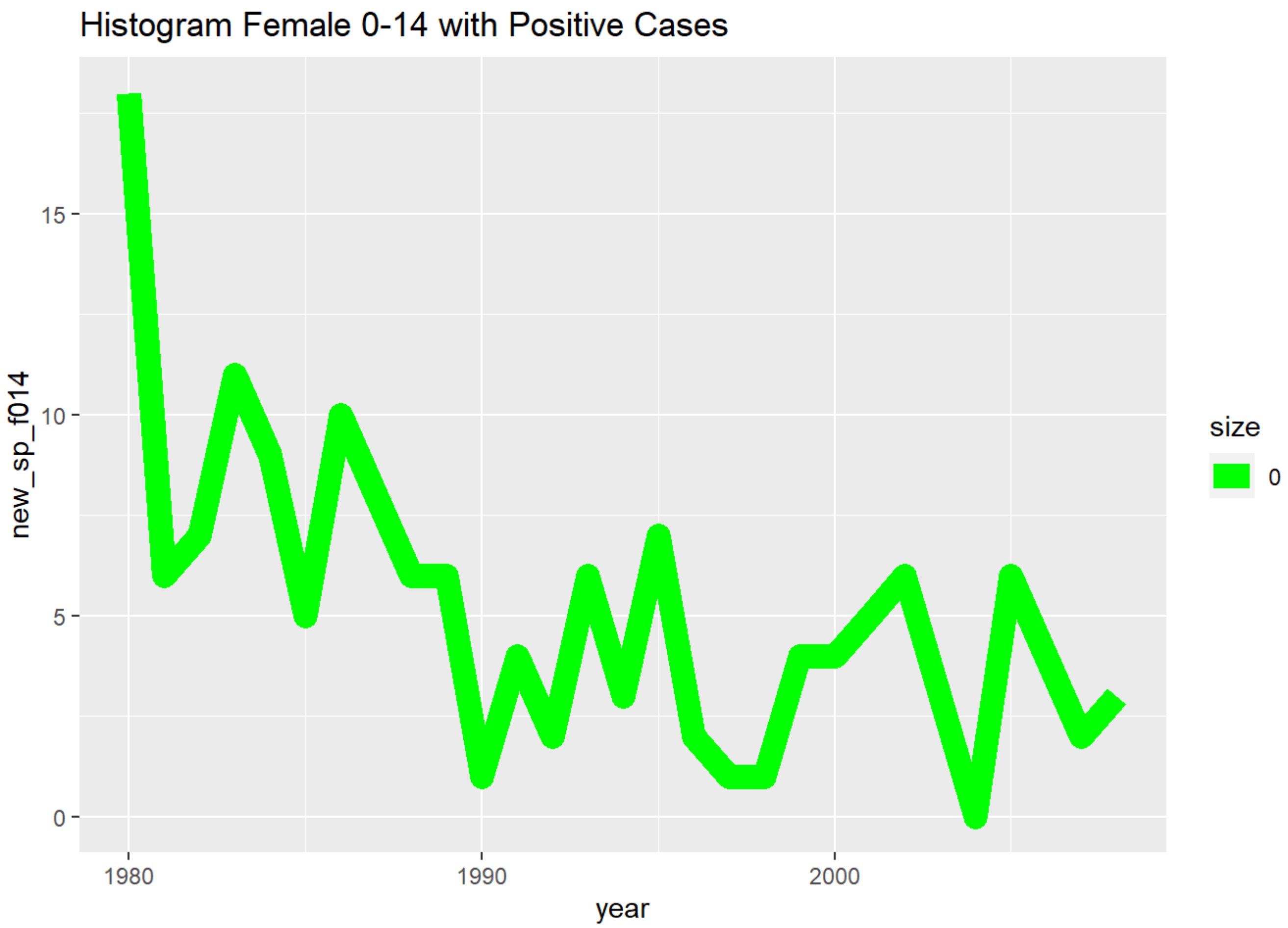
J. These data were collected in a period of several decades (1980-2013). You can thus observe changes over time with the help of a line chart. Create a **line chart**, with **year** on the X-axis and **new_sp_m014** on the Y-axis.

```
tbCan %>% ggplot() +
geom_line(color = '#F76900',
aes(x=year, y=new_sp_m014, size=0 )) +
ggtitle('Histogram Males 0-14 with Positive Cases')
```

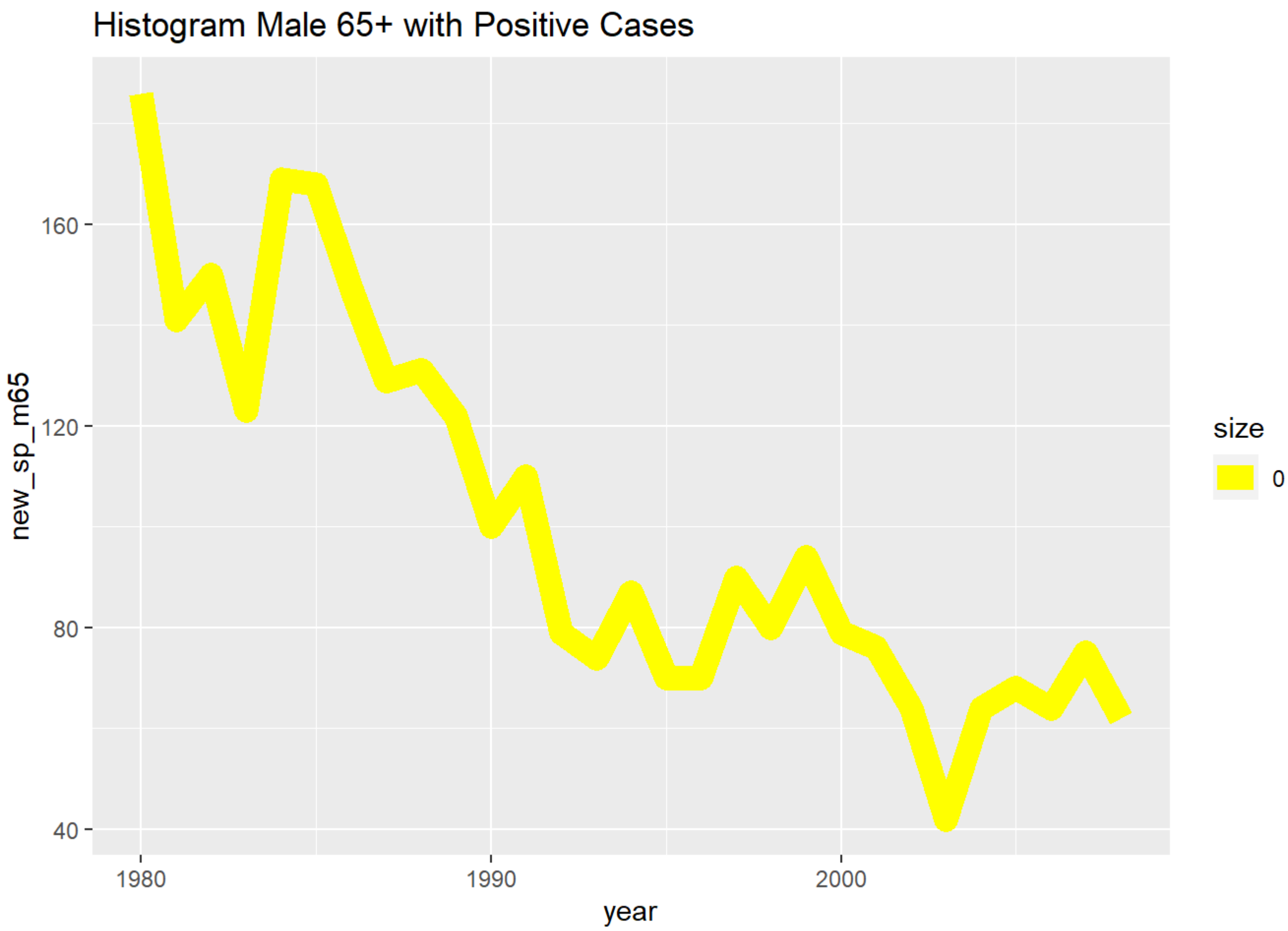


K. Next, create similar graphs for each of the other three variables. Change the **color** of the line plots (any color you want).

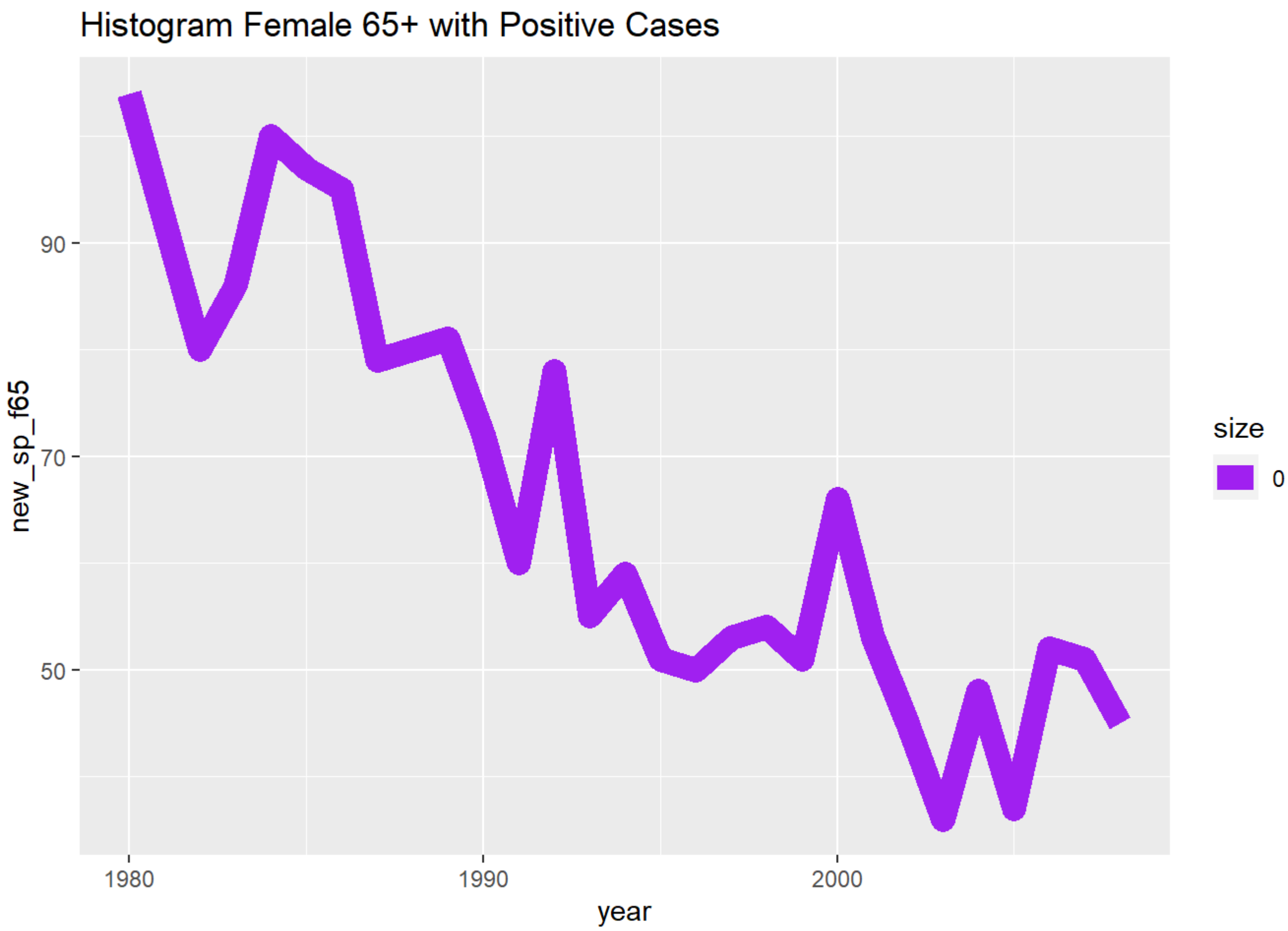
```
tbCan %>% ggplot() +
geom_line(color = 'green',
aes(x=year, y=new_sp_f014, size=0 )) +
ggtitle('Histogram Female 0-14 with Positive Cases')
```




```
tbCan %>% ggplot() +
geom_line(color = 'yellow',
aes(x=year, y=new_sp_m65, size=0 )) +
ggtitle('Histogram Male 65+ with Positive Cases')
```




```
tbCan %>% ggplot() +
geom_line(color = 'purple',
aes(x=year, y=new_sp_f65, size=0 )) +
ggtitle('Histogram Female 65+ with Positive Cases')
```



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$new_sp_combined014 <- (tbCan$new_sp_m014 + tbCan$new_sp_f014)
tbCan$new_sp_combined65 <- (tbCan$new_sp_m65 + tbCan$new_sp_f65)

show(tbCan$new_sp_combined014)
```

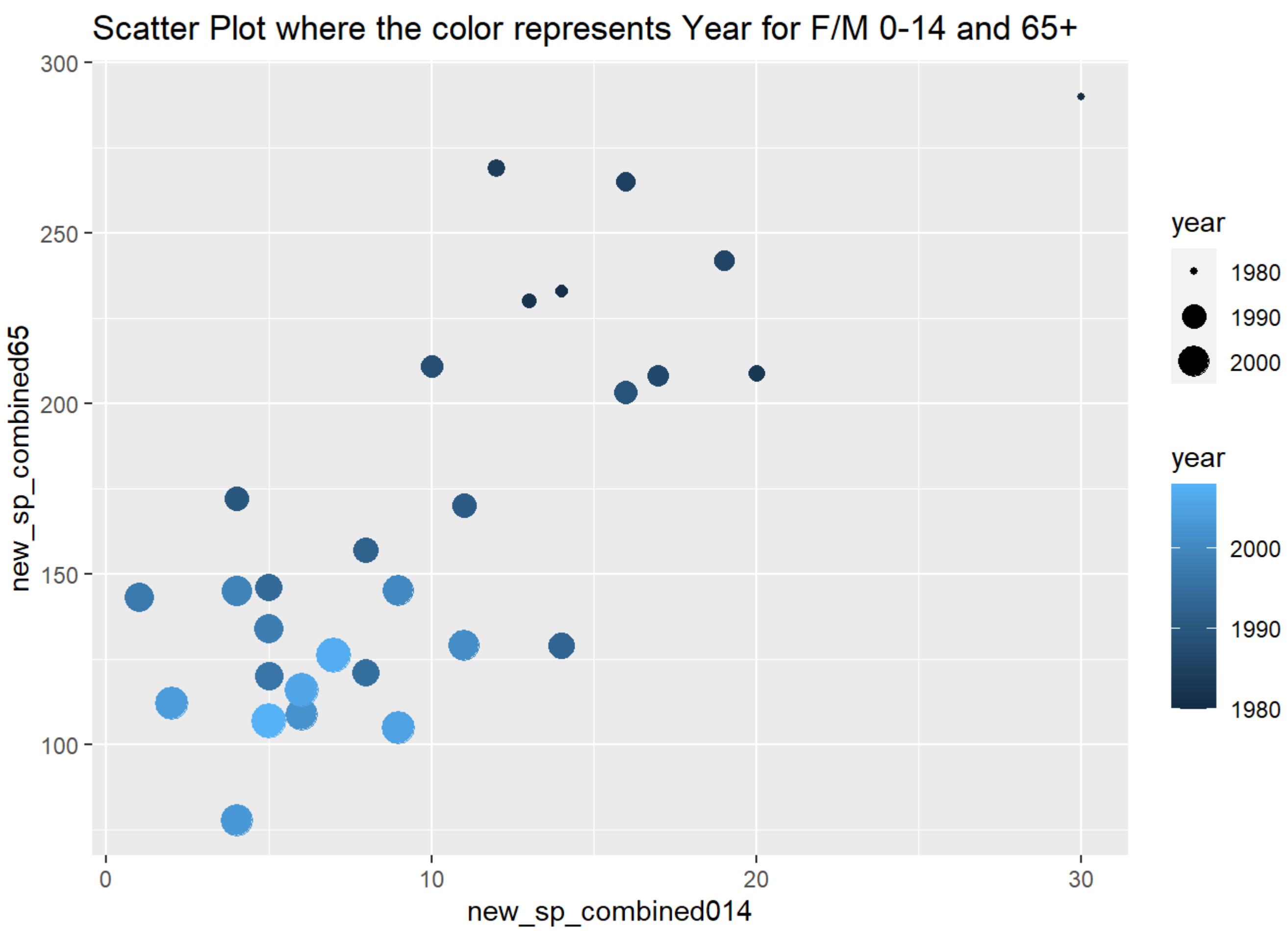
```
##  [1] 30 14 13 20 12 16 19 17 10 16  4 11  8 14  5  8  5  1  5  4  9 11  6  4  2
## [26]  9  6  7  5
```

```
show(tbCan$new_sp_combined65)
```

```
##  [1] 290 233 230 209 269 265 242 208 211 203 172 170 157 129 146 121 120 143 134
## [20] 145 145 129 109  78 112 105 116 126 107
```

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**.

```
tbCan %>% ggplot() +
geom_point() +
aes(x=new_sp_combined014, y=new_sp_combined65, size=year, color=year ) +
ggtitle('Scatter Plot where the color represents Year for F/M 0-14 and 65+')
```



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

```
# The data shows that the Older People had more tests done, but the test count droppped as time
 increased.
```