Intro to Data Science - HW 6

```
# Enter your name here: Chaithra Kopparam Cheluvaiah
```


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 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(tidyverse)  # loading the library required for importing csv data
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# parsing csv to data frame
tb <- read_csv("https://intro-datascience.s3.us-east-2.amazonaws.com/who.csv")
```

```
## Rows: 5769 Columns: 23
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): iso2
## dbl (22): year, new_sp, new_sp_m04, new_sp_m514, new_sp_m014, new_sp_m1524, ...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# examining the data
# head(tb)
# tail(tb)
```

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

```
# data is taken every year starting from 1980 till 2008 across various countries
```

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()*

```
summary(tb$iso2) # total count of countries is 5769
```

```
##    Length     Class      Mode
##      5769 character character
```

```
tb <- tb[!is.na(tb$iso2),] # removing NA

# However,
# NA seems to be country name Namibia. R is considering it as Not Available

# after removing NA
summary(tb$iso2) # total count of countries is 5746
```

```
##    Length     Class      Mode
##      5746 character character
```

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 <- tb[tb$iso2 == "CA",] # filter out TB records of Canada
# tbCan
```

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.

```
# is.na(tbCan$new_sp_m04)
# is.na() function returns true/false based on whether data is empty/NA

# tbCan$new_sp_m04[is.na(tbCan$new_sp_m04)]
# sub-setting new_sp_m04 vector based on results from is.na() function
```

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)]
```

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

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

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

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

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

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

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

```
# empty output implies individuals are tested positive for TB in all the years
# between 1980 - 2008 in Canada among children of age group 0 - 14 and elders
# of age over 65 years
```

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).

```
# install.packages("imputeTS")
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) #replacing missing values
# with mean of the neighboring points
```

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

```
# Part C
# testing for NA
tbCan$new_sp_m04[is.na(tbCan$new_sp_m04)]
```

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

```
# Part E
# testing for NA
tbCan$new_sp_f014[is.na(tbCan$new_sp_f014)]
```

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

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

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

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

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

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

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

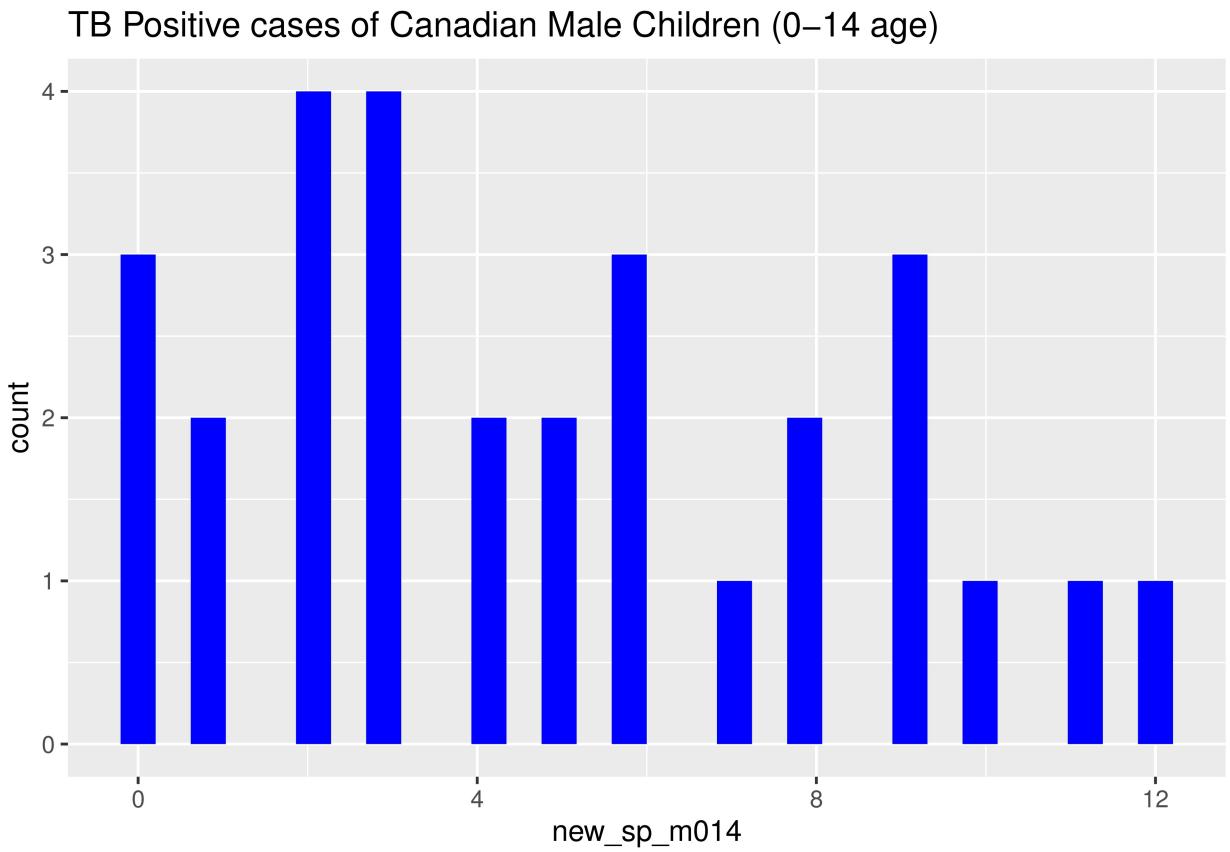
Step 3: Use ggplot to explore the distribution of each variable

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)
sp_m014_plot <- ggplot(tbCan, aes(x=new_sp_m014)) +
  ggtitle("TB Positive cases of Canadian Male Children (0-14 age) ") +
  geom_histogram(fill="blue")

sp_m014_plot
```

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



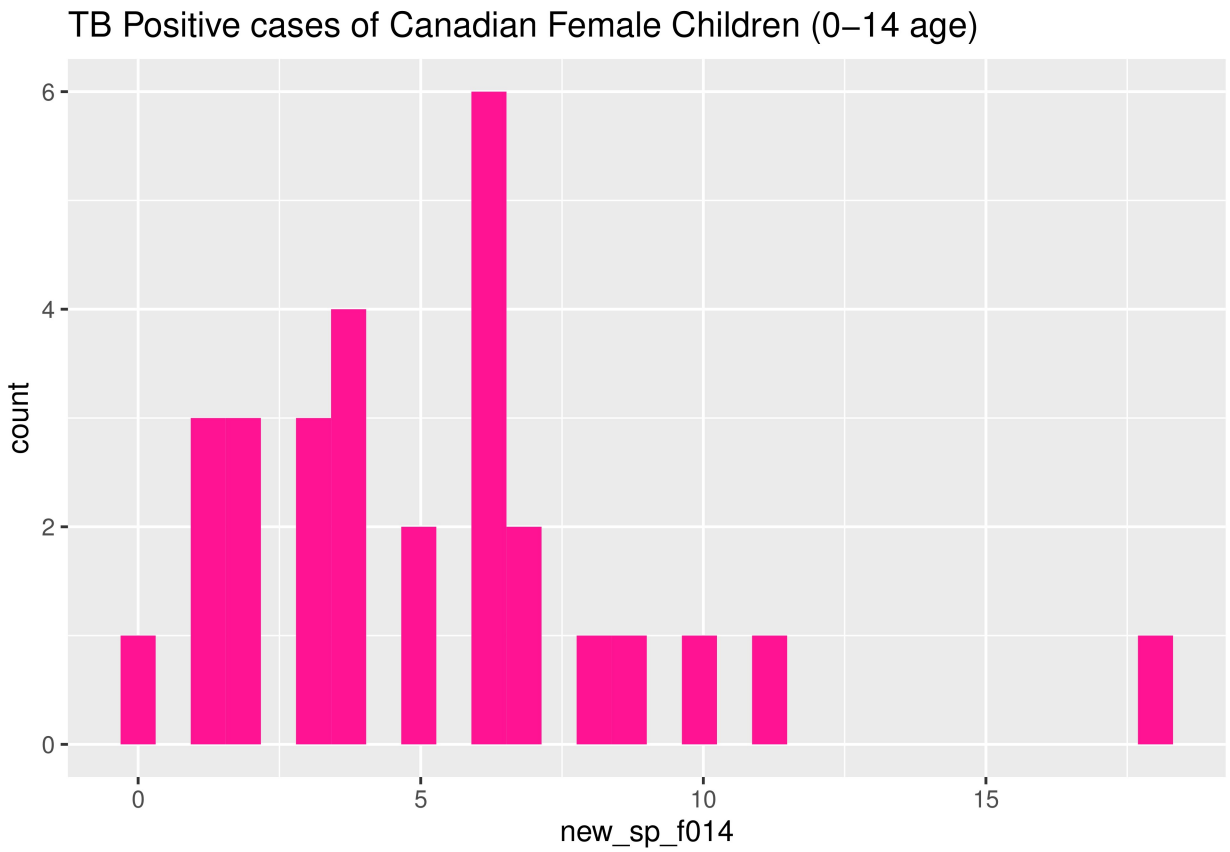

```
# histogram describes frequency distribution of number of positive TB cases
# among male children of 0-14 age group in Canada
```

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?

```
sp_f014_plot <- ggplot(tbCan, aes(x=new_sp_f014)) +
  ggtitle("TB Positive cases of Canadian Female Children (0-14 age) ") +
  geom_histogram(fill="deeppink")

sp_f014_plot
```

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

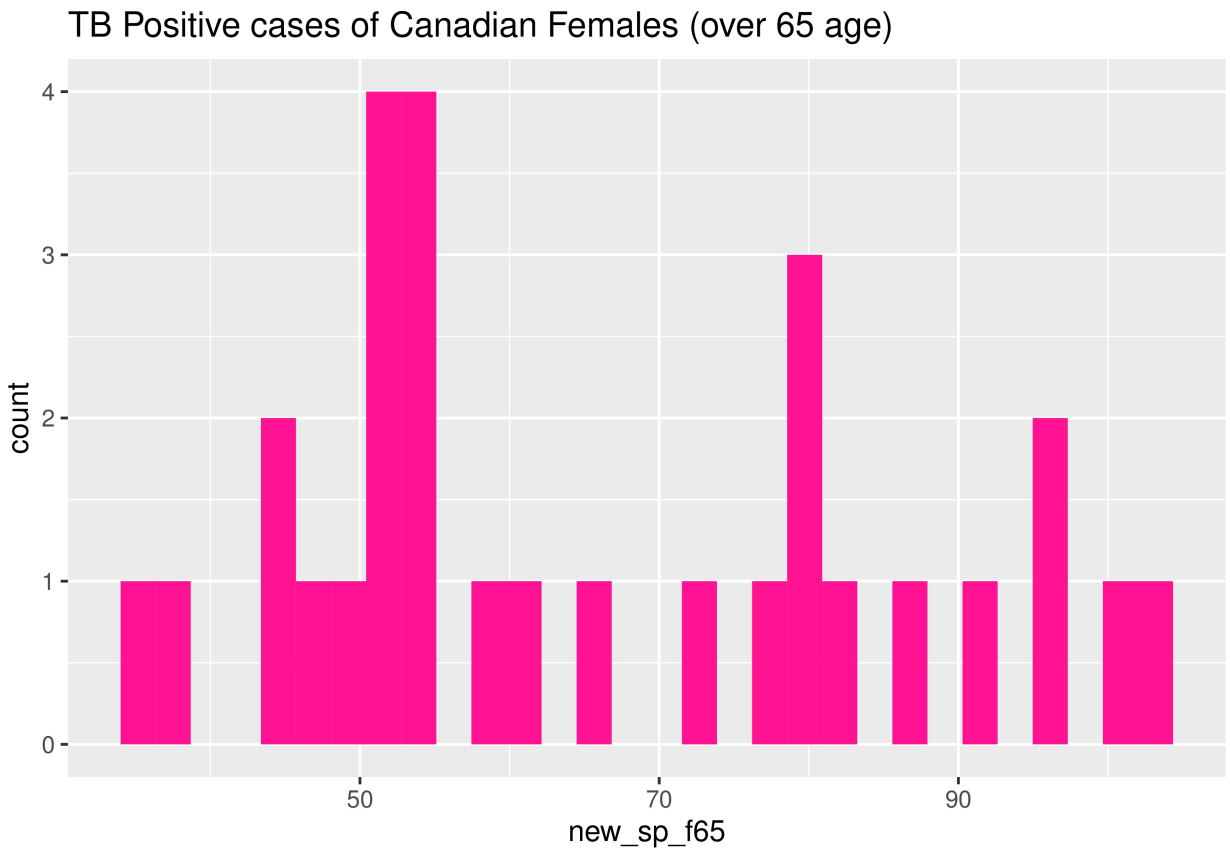



```
# histogram describes frequency distribution of number of positive TB cases
# among female children of 0-14 age group in Canada
```

```
sp_f65_plot <- ggplot(tbCan, aes(x=new_sp_f65)) +
  ggtitle("TB Positive cases of Canadian Females (over 65 age) ") +
  geom_histogram(fill="deeppink")

sp_f65_plot
```

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

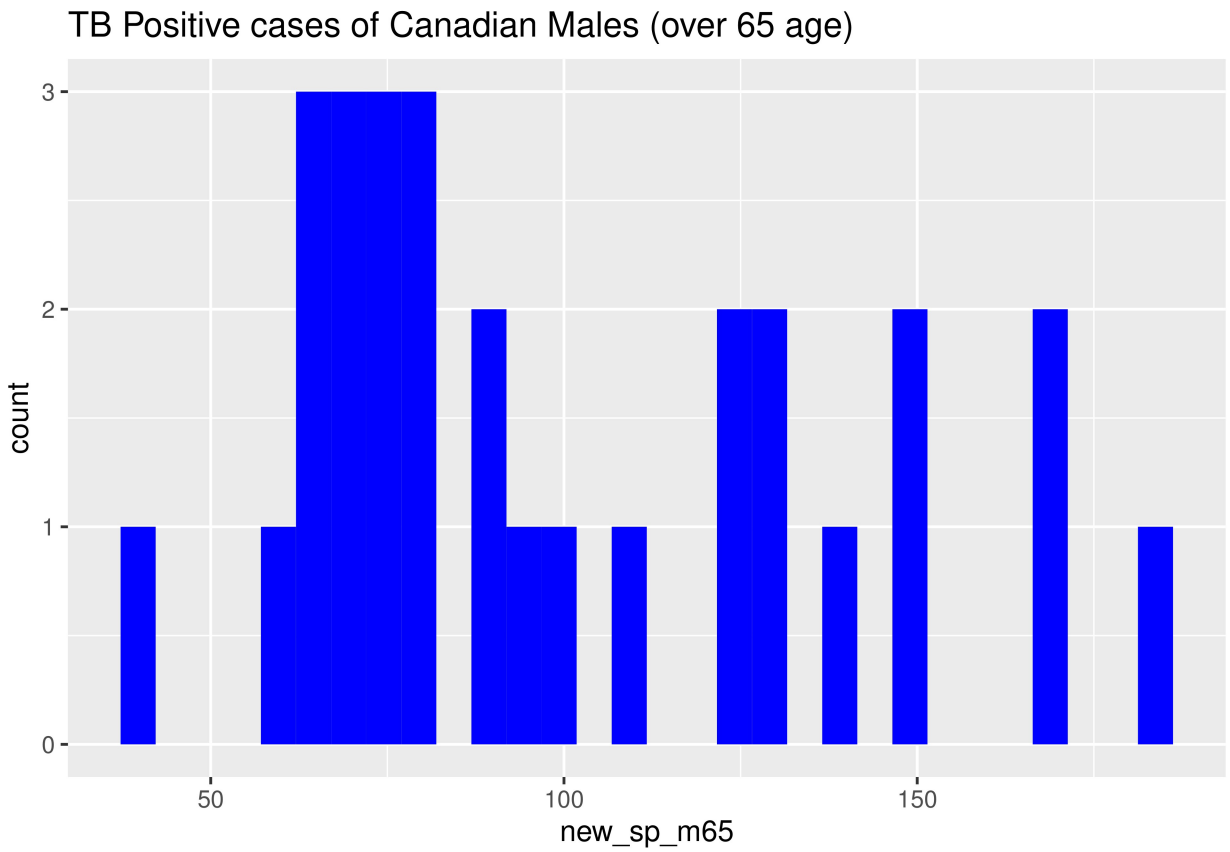



```
# histogram describes frequency distribution of number of positive TB cases
# among females of age over 65 years in Canada
```

```
sp_m65_plot <- ggplot(tbCan, aes(x=new_sp_m65)) +
  ggtitle("TB Positive cases of Canadian Males (over 65 age) ") +
  geom_histogram(fill="blue")

sp_m65_plot
```

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

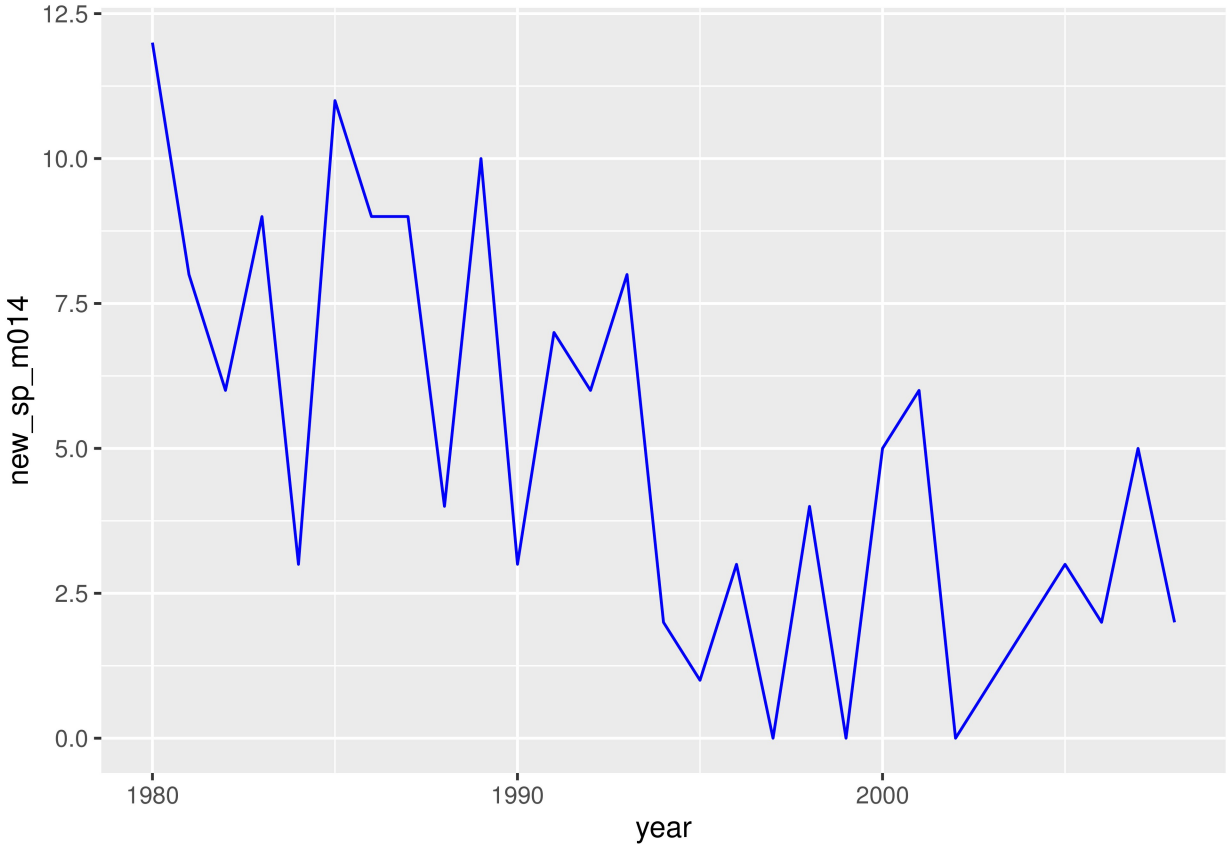



```
# histogram describes frequency distribution of number of positive TB cases
# among males of age over 65 years in Canada
```

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.

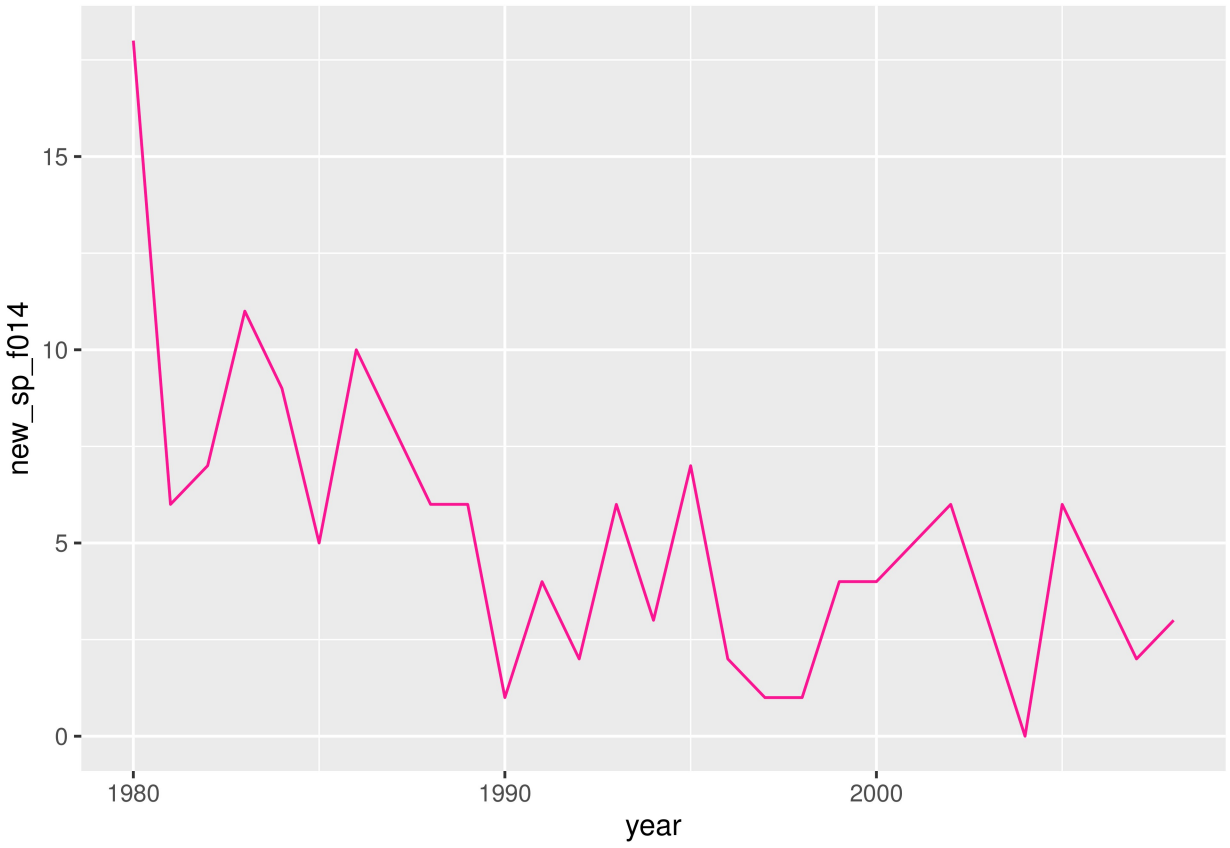
```
trend_sp_m014 <- ggplot(tbCan, aes(x=year, y=new_sp_m014)) +
  geom_line(col="blue")
trend_sp_m014
```




```
# number of positive test cases among male children of 0-14 age group has
# has decreased over time in Canada
```

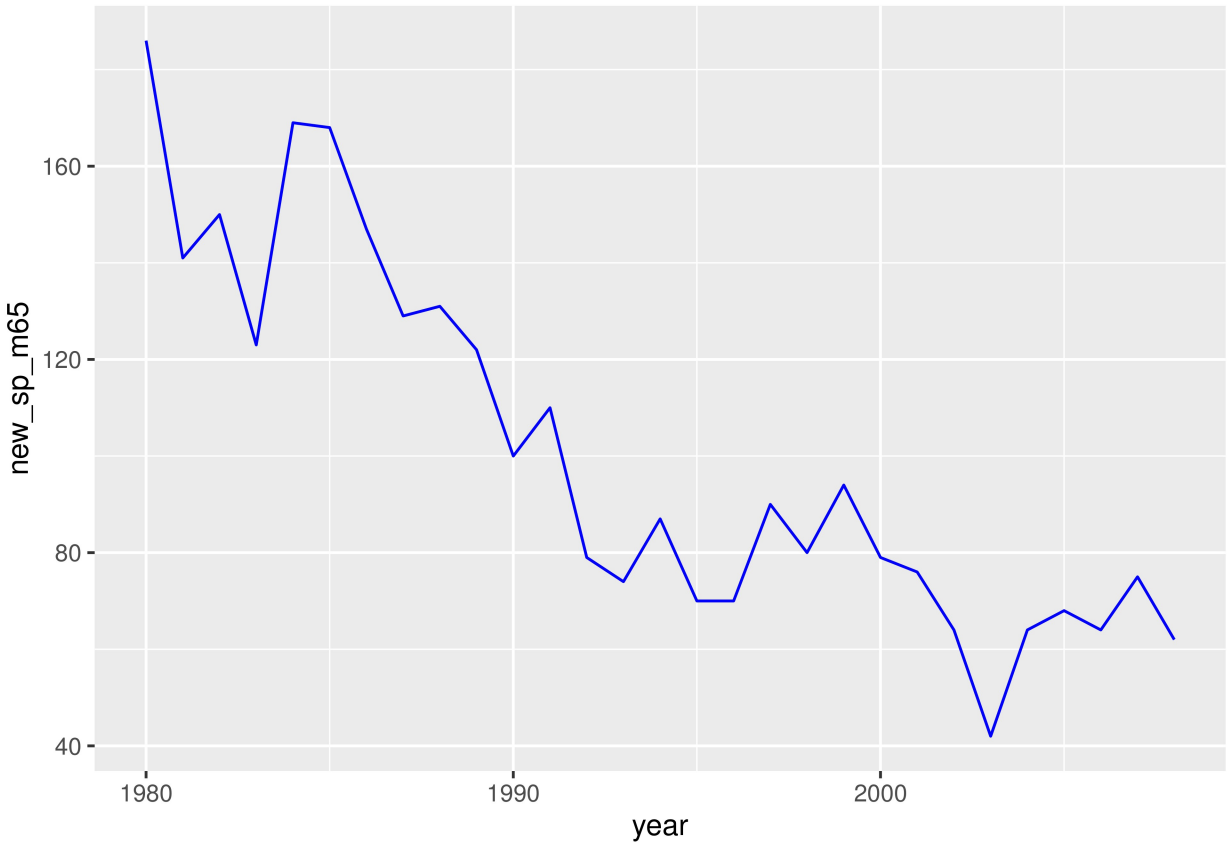
K. Next, create similar graphs for each of the other three variables. Change the **color** of the line plots (any color you want).

```
trend_sp_f014 <- ggplot(tbCan, aes(x=year, y=new_sp_f014)) +
  geom_line(col="deeppink")
trend_sp_f014
```



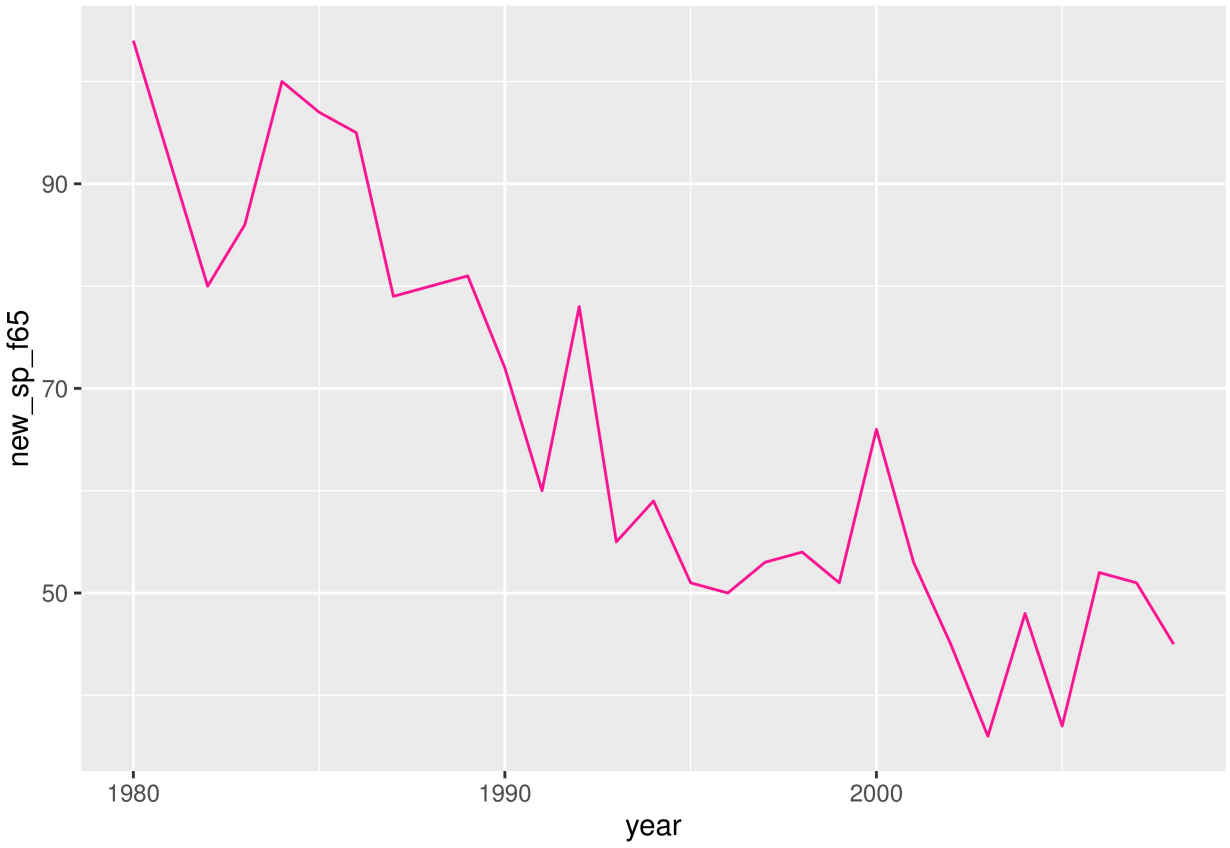

```
# number of positive test cases among female children of 0-14 age group has
# has decreased over time in Canada
```

```
trend_sp_m65 <- ggplot(tbCan, aes(x=year, y=new_sp_m65)) +
  geom_line(col="blue")
trend_sp_m65
```




```
# number of positive test cases among males of age over 65 years has
# has decreased over time in Canada
```

```
trend_sp_f65 <- ggplot(tbCan, aes(x=year, y=new_sp_f65)) +
  geom_line(col="deeppink")
trend_sp_f65
```



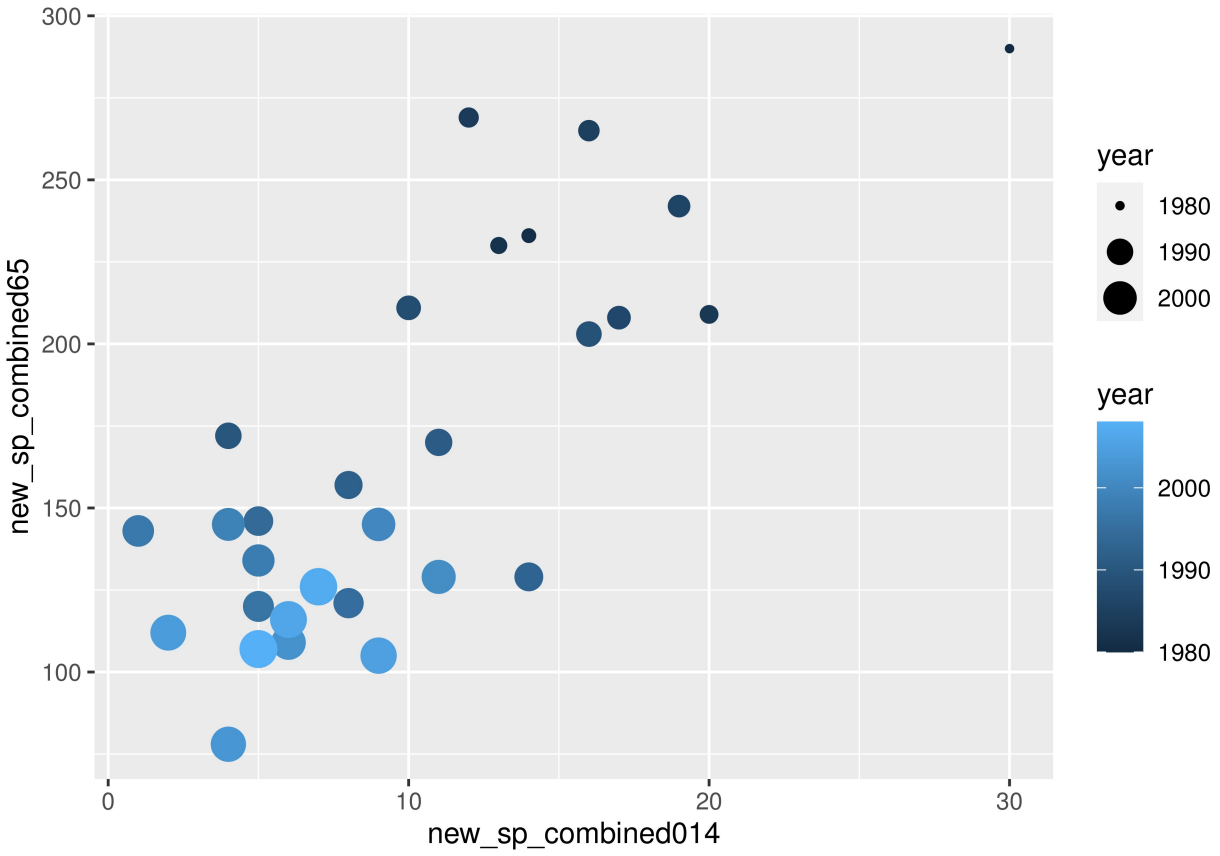

```
# number of positive test cases among females of age over 65 years has
# has decreased over time in Canada
```

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_f014 + tbCan$new_sp_m014
tbCan$new_sp_combined65 <- 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**.

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



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

```
# Inference
# 1) more number of individuals were tested positive for TB among both the age
# groups between 1980 - 1990
# 2) less number of individuals were tested positive for TB among both the age
# groups post 2000
# 3) with increase in TB cases among children, there is an increase in TB cases
# among elders as well and viz.,
```