

# Intro to Data Science - Lab 6

Copyright 2022, Jeffrey Stanton and Jeffrey Saltz Please do not post online.

## Week 6 - Using ggplot to Build Complex Data Displays

```
# Enter your name here: Hendi Kushta
```

Please include nice comments.

### Instructions:

Run the necessary code on your own instance of R-Studio.

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

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

Creating graphical displays of data is an essential skill for all data scientists because so much of what data scientists do involves communicating with other people about data. The **ggplot2** package developed by data scientist Hadley Wickham ([https://en.wikipedia.org/wiki/Hadley\\_Wickham](https://en.wikipedia.org/wiki/Hadley_Wickham)) provides excellent power and flexibility for graphically displaying data. Whole books have been written about **ggplot2** (e.g.: <https://www.springer.com/gp/book/9780387981413>), so we will only be able to scratch the surface, but we will master the basic grammar that you need in order to use this package. For this lab we will also use a dataset that comes delivered in R, called **\*\* economics \*\***. It contains 574 snapshots of U.S. economic conditions between 1967 and 2014.

Here are two lines of starter code:

```
MyPlot <- ggplot(economics, aes(x=date))
myPlot <- myPlot + geom_line(aes(y=psavert))
```

1. Run these two lines of code below. What happens? How do you actually invoke the plot (i.e., how do you get it to draw in the plot window)?

```
# install.packages('tidyverse')
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

MyPlot <- ggplot(economics, aes(x=date))
MyPlot <- MyPlot + geom_line(aes(y=psavert))
# we plot by just writing MyPlot
MyPlot
```



2. Run `help("economics")` to find out the meaning of the `psavert` variable.

```
help('economics')
# psavert -> personal savings rate
```

3. Examine the plot to estimate when the personal savings rate reached its maximum value. Also examine the plot to estimate when the personal savings rate reached its minimum value.

```
# From the graph we can see that the maximum personal savings rate was reached
# in 1975 and the minimum in 2005
```

4. Use `which.max( )` and `which.min( )` to verify your guesses from problem 3.

```
economics[which.max(economics$psavert),]

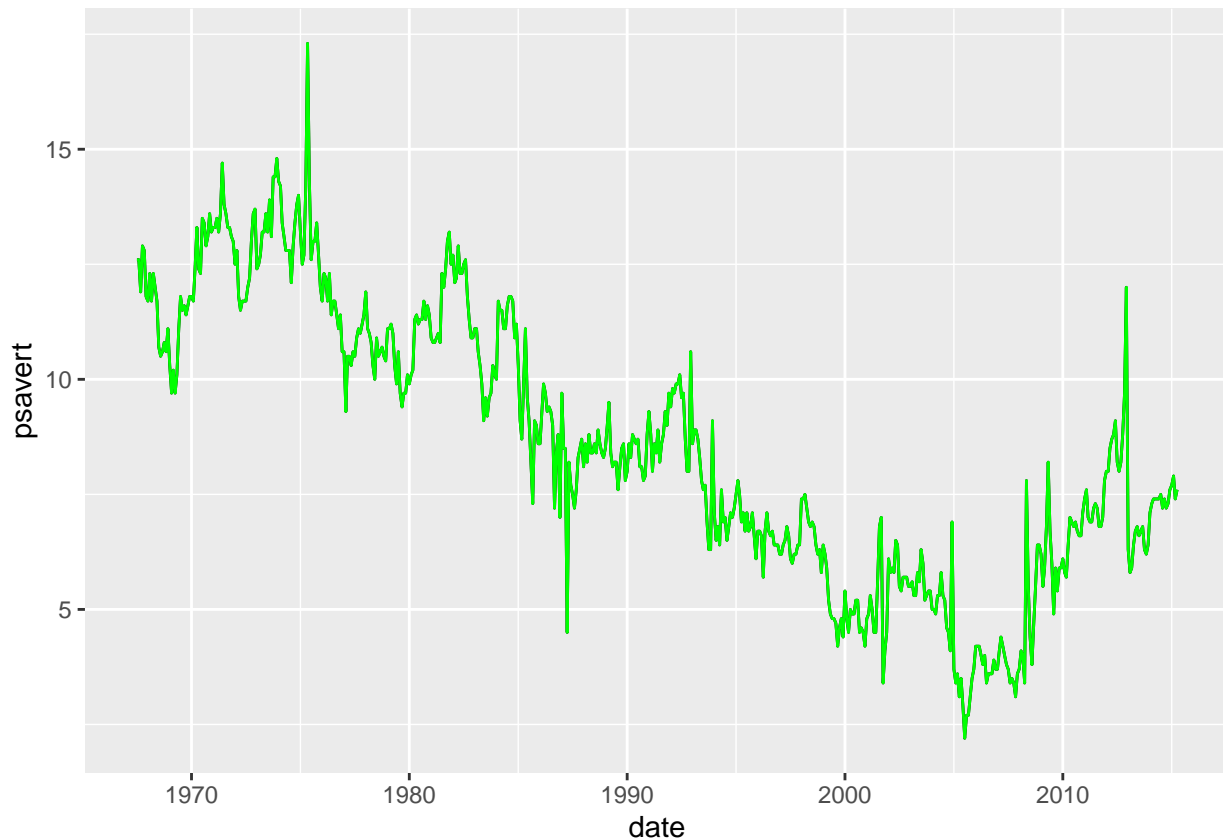
## # A tibble: 1 x 6
##   date      pce    pop psavert uempmed unemploy
##   <date>    <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 1975-05-01 1019. 215523   17.3     9.4    8433

economics[which.min(economics$psavert),]

## # A tibble: 1 x 6
##   date      pce    pop psavert uempmed unemploy
##   <date>    <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 2005-07-01 8830. 296186    2.2     8.8    7406
```

5. Change the color of the plot line to green. **Hint:** Changing a line to a constant color happens in the specification of the `geometry`.

```
MyPlot <- MyPlot + geom_line(aes(y=psavert), color="green")
MyPlot
```



6. Add a title to the plot with the `ggtitle("Put title here")` sub-command. The title **"Personal Savings Rate: 1967-2014"** would be a good choice.

```
MyPlot + ggtitle("Personal Savings Rate: 1967-2014")
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

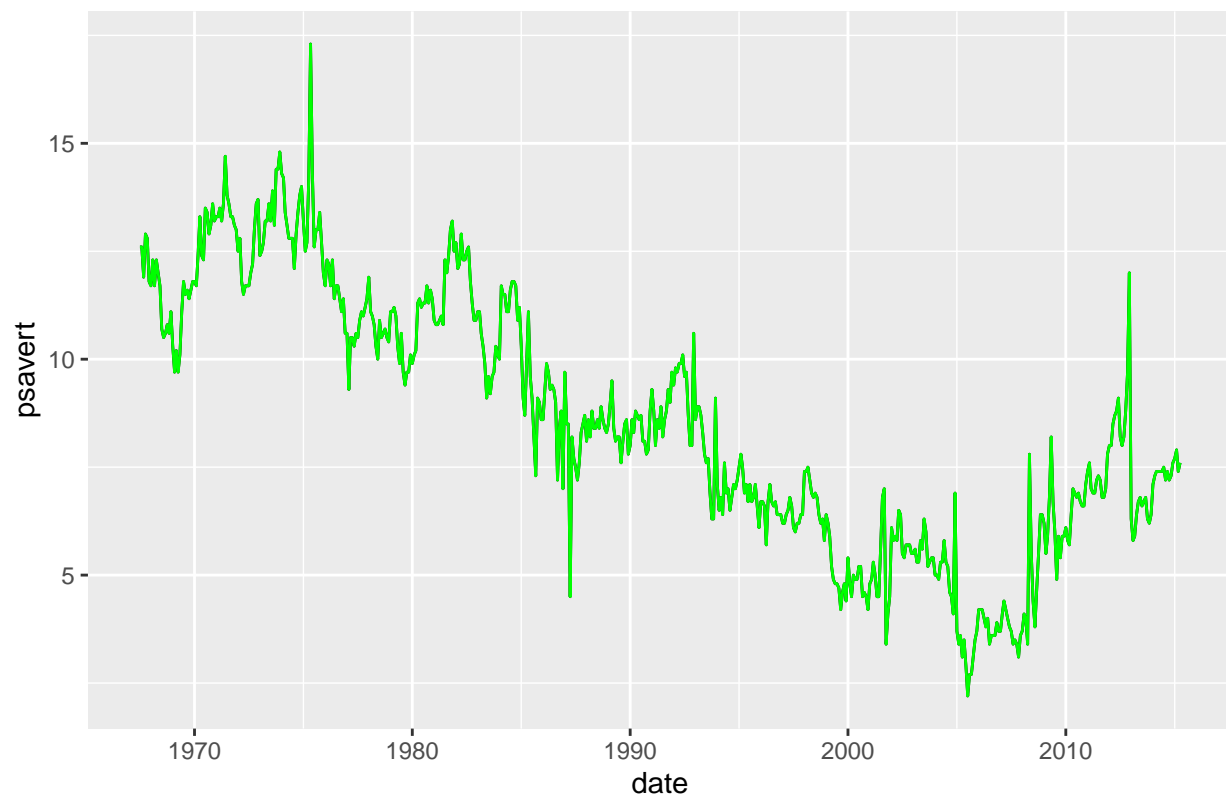
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

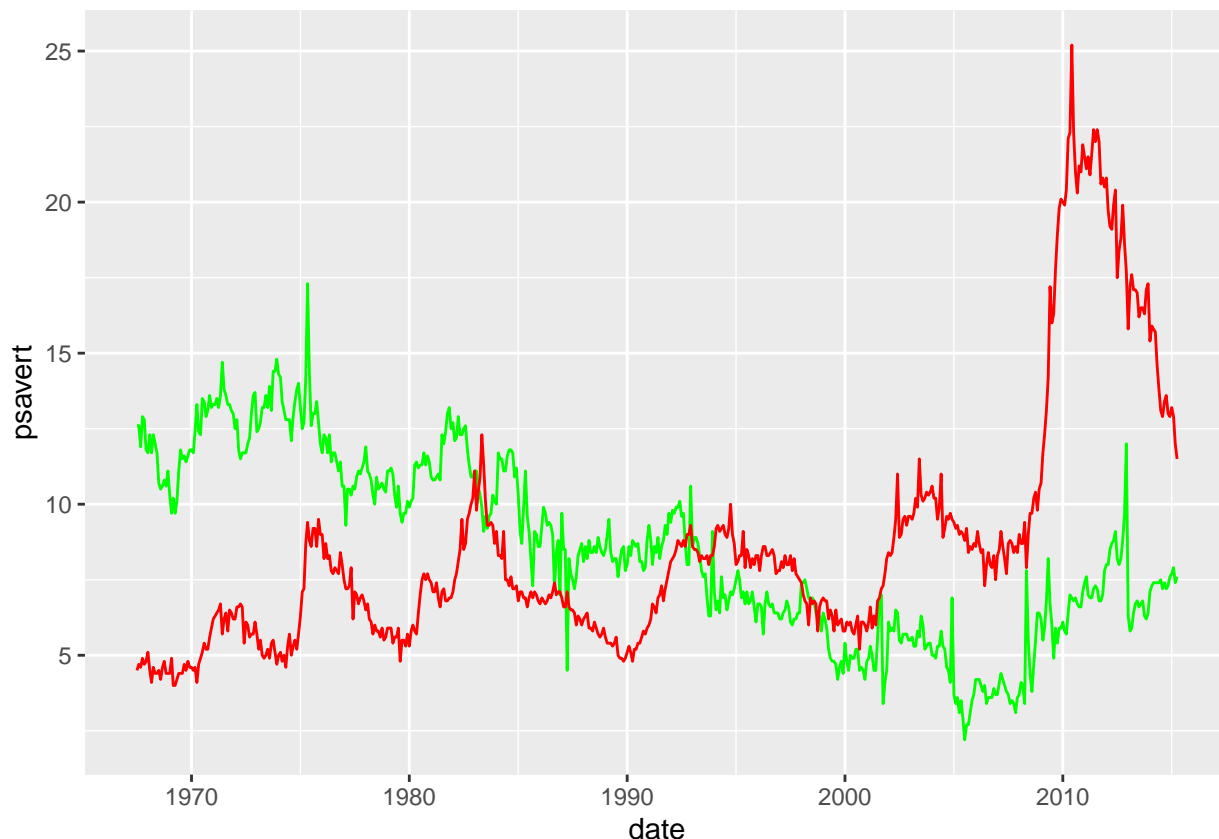


## Personal Savings Rate: 1967–2014



7. Add another data series to your plot to show the variable **uempmed** as a red line.

```
MyPlot <- ggplot(economics, aes(x=date))  
MyPlot <- MyPlot + geom_line(aes(y=psavert), colour="green") +  
               geom_line(aes(y=uempmed), colour="red")  
MyPlot
```



8. Change the title of the plot to mention both variables.

```
MyPlot + ggtitle("Personal Savings Rate And Unemployment: 1967-2014")
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

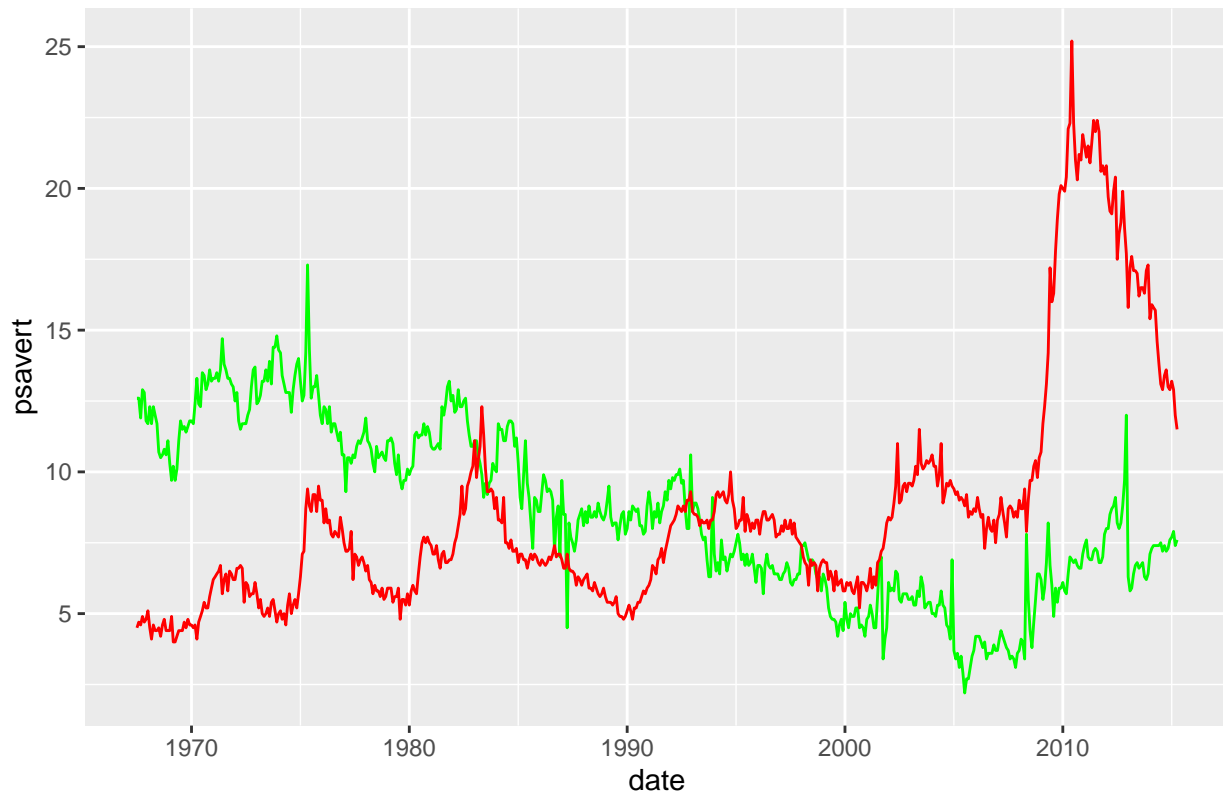
```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9
```

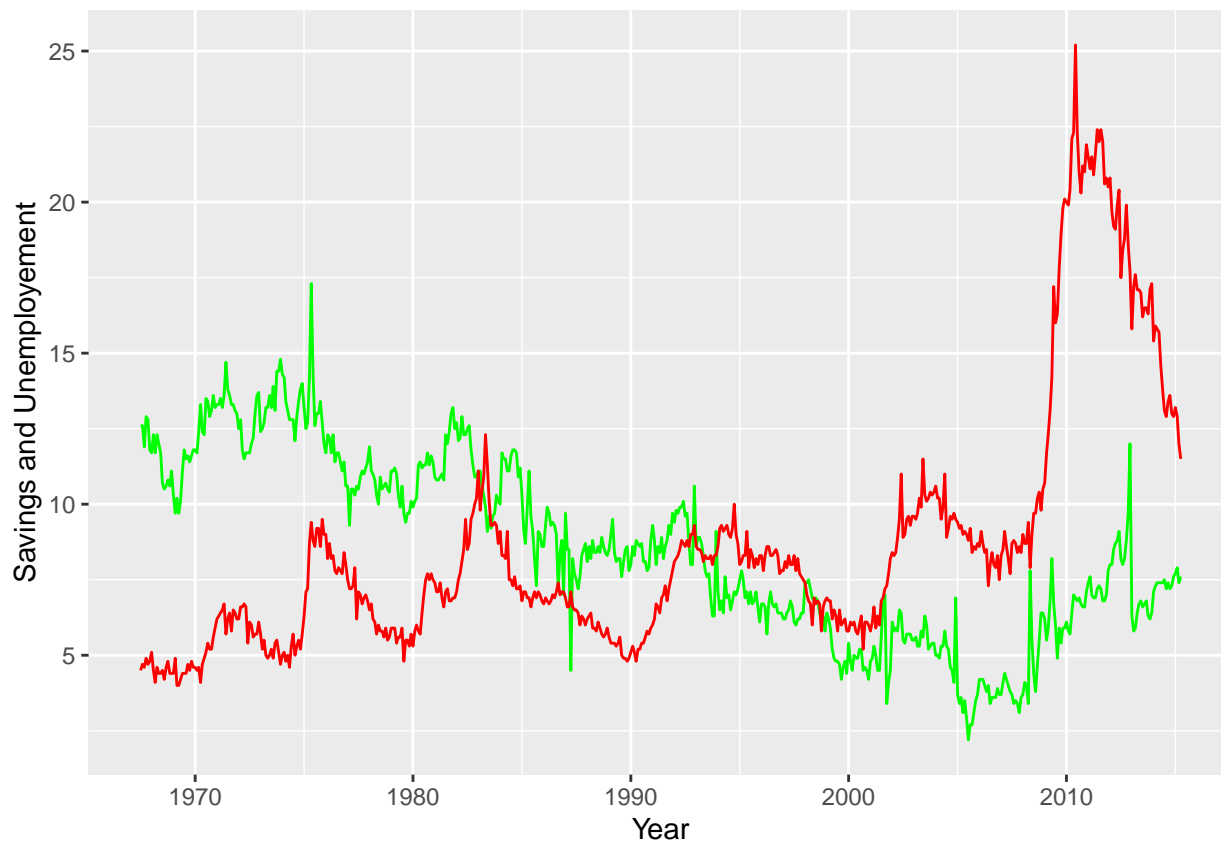
```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x9
```

## Personal Savings Rate And Unemployment: 1967–2014



9. You can modify the axis labels in a ggplot with `ylab( )` and `xlab( )` subcommands. Change the axis labeling as needed to account for plotting both `psavert` and `uempmed` in the same window.

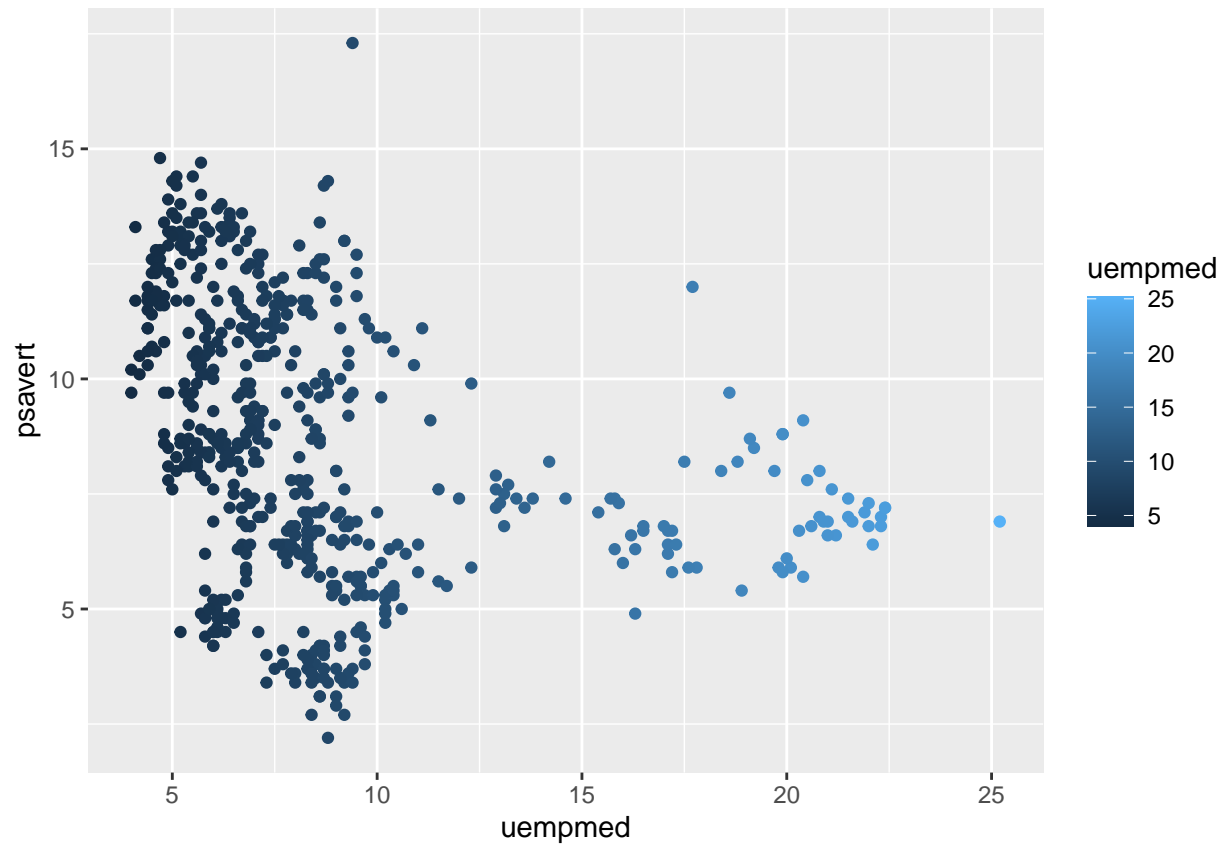
```
MyPlot + xlab("Year") + ylab("Savings and Unemployment")
```



10. Create one last plot, creating a scatter plot, having the **unemploy** on the x-axis, **psavert** on the y-axis. Color each point based on the **uempmed**.

```
ggplot(economics) + aes(x=uempmed, y=psavert, color=uempmed) + geom_point()
```





11. Interpret (using comments in R) what you see in this last graph.

```
# As a median, people who have been unemployed between 5-10 weeks have personal  
# saving rates between 0-15, while people who have been unemployed for 15-20 weeks  
# have constant savings of 5-10
```