IT137IU: Data Analysis
Lab#7/Assignment#7: Data Visualization (continue)

Introduction

2 Days On Zillow
HOUSE FOR SALE
$75,000 3 bds · 2 ba · 1,730 sqft
720 Saint Lewis Rd, Macclesfield, NC
CENTURY 21 The Combs Company
408 Days On Zillow
HOUSE FOR SALE
$290,000 3 bds · 4 ba · 4,304 sqft
48 Chipper Ct, Pinetops, NC
ALDRIDGE & SOUTHERLAND
Sponsored
Visit website
Trinity Creek: The Crest
2500-3800 sq.ft, 3-4 Bed with basement on wooded homesites: mid-$400s.

In this lab you will acquire the skills needed to process and present data in R. The objectives of the guide are as follows:

1. Learn basic data wrangling operations.
2. Learn how to use various ggplot functions to visualize data.

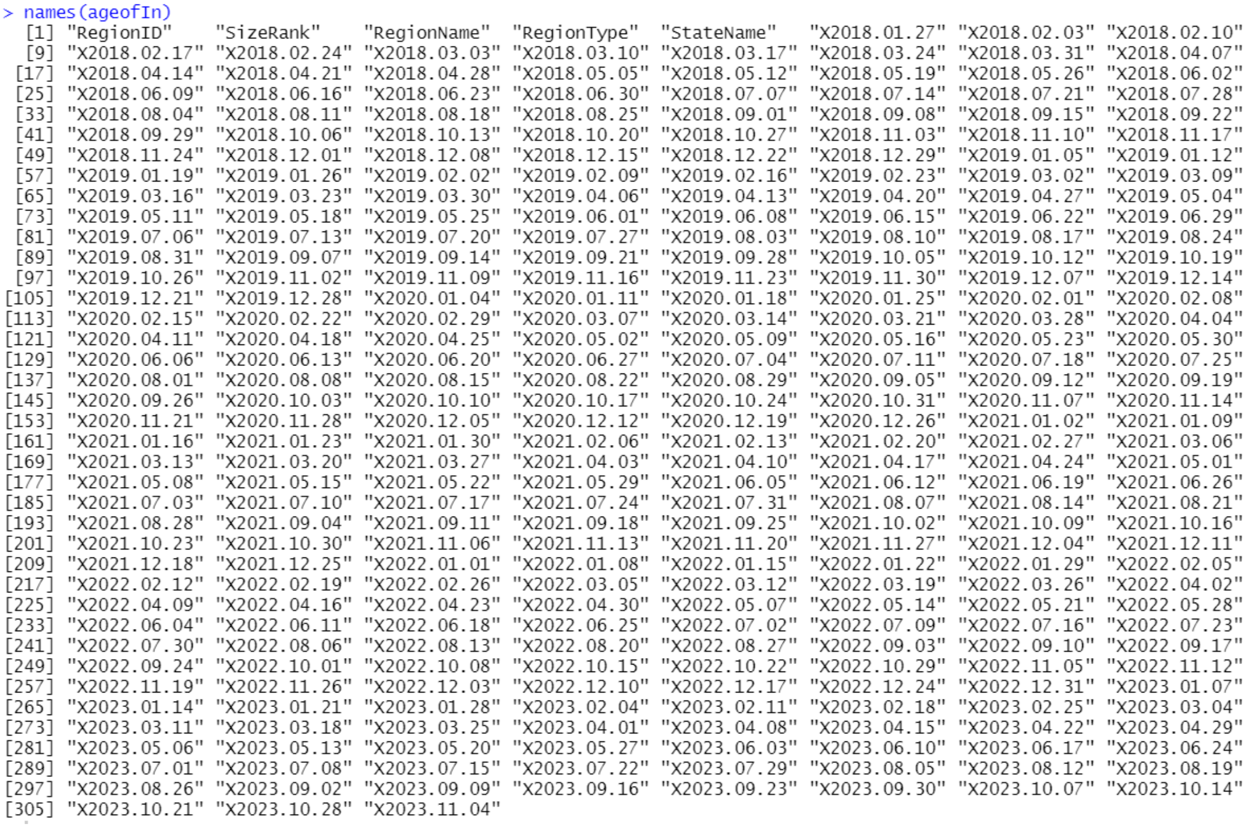
I uploaded a file on Blackboard containing housing price data at city levels taken from Zillow Group. Download the file ***AgeofInventory.csv*** and save it into the same folder where your Lab 7 file resides. Their list of data definitions at the bottom of the page includes the following entry:

Age of Inventory: Each Wednesday, age of inventory is calculated as the median number of days all active listings as of that Wednesday have been current. These medians are then aggregated into the number reported by taking the median across weekly values.

1. Data Wrangling

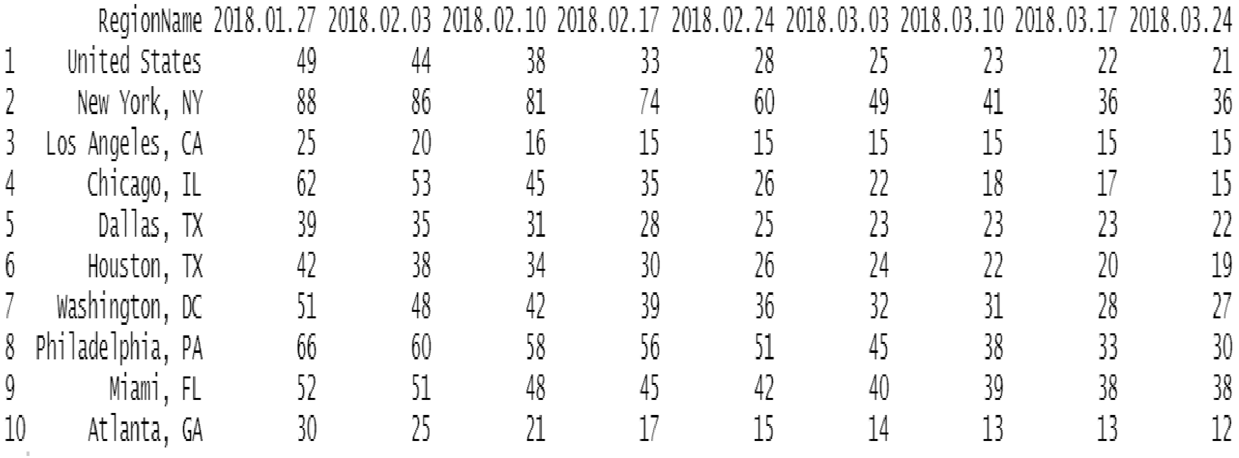
To see the names of variables in the dataset `ageofIn`, use the `names()` command.

Since this data set uses a separate column for each time period, the data is not yet tidy. Let's fix that.



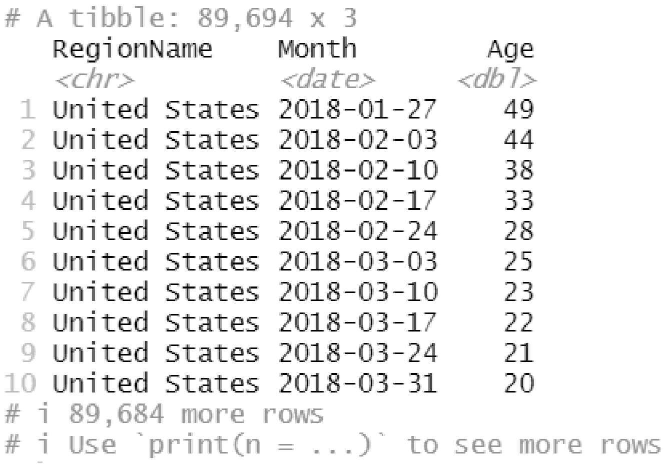
```
> names(ageofIn)
  [1] "RegionID"    "SizeRank"    "RegionName"  "RegionType"  "StateName"   "X2018.01.27" "X2018.02.03" "X2018.02.10"
  [9] "X2018.02.17" "X2018.02.24" "X2018.03.03" "X2018.03.10" "X2018.03.17" "X2018.03.24" "X2018.03.31" "X2018.04.07"
 [17] "X2018.04.14" "X2018.04.21" "X2018.04.28" "X2018.05.05" "X2018.05.12" "X2018.05.19" "X2018.05.26" "X2018.06.02"
 [25] "X2018.06.09" "X2018.06.16" "X2018.06.23" "X2018.06.30" "X2018.07.07" "X2018.07.14" "X2018.07.21" "X2018.07.28"
 [33] "X2018.08.04" "X2018.08.11" "X2018.08.18" "X2018.08.25" "X2018.09.01" "X2018.09.08" "X2018.09.15" "X2018.09.22"
 [41] "X2018.09.29" "X2018.10.06" "X2018.10.13" "X2018.10.20" "X2018.10.27" "X2018.11.03" "X2018.11.10" "X2018.11.17"
 [49] "X2018.11.24" "X2018.12.01" "X2018.12.08" "X2018.12.15" "X2018.12.22" "X2018.12.29" "X2019.01.05" "X2019.01.12"
 [57] "X2019.01.19" "X2019.01.26" "X2019.02.02" "X2019.02.09" "X2019.02.16" "X2019.02.23" "X2019.03.02" "X2019.03.09"
 [65] "X2019.03.16" "X2019.03.23" "X2019.03.30" "X2019.04.06" "X2019.04.13" "X2019.04.20" "X2019.04.27" "X2019.05.04"
 [73] "X2019.05.11" "X2019.05.18" "X2019.05.25" "X2019.06.01" "X2019.06.08" "X2019.06.15" "X2019.06.22" "X2019.06.29"
 [81] "X2019.07.06" "X2019.07.13" "X2019.07.20" "X2019.07.27" "X2019.08.03" "X2019.08.10" "X2019.08.17" "X2019.08.24"
 [89] "X2019.08.31" "X2019.09.07" "X2019.09.14" "X2019.09.21" "X2019.09.28" "X2019.10.05" "X2019.10.12" "X2019.10.19"
 [97] "X2019.10.26" "X2019.11.02" "X2019.11.09" "X2019.11.16" "X2019.11.23" "X2019.11.30" "X2019.12.07" "X2019.12.14"
[105] "X2019.12.21" "X2019.12.28" "X2020.01.04" "X2020.01.11" "X2020.01.18" "X2020.01.25" "X2020.02.01" "X2020.02.08"
[113] "X2020.02.15" "X2020.02.22" "X2020.02.29" "X2020.03.07" "X2020.03.14" "X2020.03.21" "X2020.03.28" "X2020.04.04"
[121] "X2020.04.11" "X2020.04.18" "X2020.04.25" "X2020.05.02" "X2020.05.09" "X2020.05.16" "X2020.05.23" "X2020.05.30"
[129] "X2020.06.06" "X2020.06.13" "X2020.06.20" "X2020.06.27" "X2020.07.04" "X2020.07.11" "X2020.07.18" "X2020.07.25"
[137] "X2020.08.01" "X2020.08.08" "X2020.08.15" "X2020.08.22" "X2020.08.29" "X2020.09.05" "X2020.09.12" "X2020.09.19"
[145] "X2020.09.26" "X2020.10.03" "X2020.10.10" "X2020.10.17" "X2020.10.24" "X2020.10.31" "X2020.11.07" "X2020.11.14"
[153] "X2020.11.21" "X2020.11.28" "X2020.12.05" "X2020.12.12" "X2020.12.19" "X2020.12.26" "X2021.01.02" "X2021.01.09"
[161] "X2021.01.16" "X2021.01.23" "X2021.01.30" "X2021.02.06" "X2021.02.13" "X2021.02.20" "X2021.02.27" "X2021.03.06"
[169] "X2021.03.13" "X2021.03.20" "X2021.03.27" "X2021.04.03" "X2021.04.10" "X2021.04.17" "X2021.04.24" "X2021.05.01"
[177] "X2021.05.08" "X2021.05.15" "X2021.05.22" "X2021.05.29" "X2021.06.05" "X2021.06.12" "X2021.06.19" "X2021.06.26"
[185] "X2021.07.03" "X2021.07.10" "X2021.07.17" "X2021.07.24" "X2021.07.31" "X2021.08.07" "X2021.08.14" "X2021.08.21"
[193] "X2021.08.28" "X2021.09.04" "X2021.09.11" "X2021.09.18" "X2021.09.25" "X2021.10.02" "X2021.10.09" "X2021.10.16"
[201] "X2021.10.23" "X2021.10.30" "X2021.11.06" "X2021.11.13" "X2021.11.20" "X2021.11.27" "X2021.12.04" "X2021.12.11"
[209] "X2021.12.18" "X2021.12.25" "X2022.01.01" "X2022.01.08" "X2022.01.15" "X2022.01.22" "X2022.01.29" "X2022.02.05"
[217] "X2022.02.12" "X2022.02.19" "X2022.02.26" "X2022.03.05" "X2022.03.12" "X2022.03.19" "X2022.03.26" "X2022.04.02"
[225] "X2022.04.09" "X2022.04.16" "X2022.04.23" "X2022.04.30" "X2022.05.07" "X2022.05.14" "X2022.05.21" "X2022.05.28"
[233] "X2022.06.04" "X2022.06.11" "X2022.06.18" "X2022.06.25" "X2022.07.02" "X2022.07.09" "X2022.07.16" "X2022.07.23"
[241] "X2022.07.30" "X2022.08.06" "X2022.08.13" "X2022.08.20" "X2022.08.27" "X2022.09.03" "X2022.09.10" "X2022.09.17"
[249] "X2022.09.24" "X2022.10.01" "X2022.10.08" "X2022.10.15" "X2022.10.22" "X2022.10.29" "X2022.11.05" "X2022.11.12"
[257] "X2022.11.19" "X2022.11.26" "X2022.12.03" "X2022.12.10" "X2022.12.17" "X2022.12.24" "X2022.12.31" "X2023.01.07"
[265] "X2023.01.14" "X2023.01.21" "X2023.01.28" "X2023.02.04" "X2023.02.11" "X2023.02.18" "X2023.02.25" "X2023.03.04"
[273] "X2023.03.11" "X2023.03.18" "X2023.03.25" "X2023.04.01" "X2023.04.08" "X2023.04.15" "X2023.04.22" "X2023.04.29"
[281] "X2023.05.06" "X2023.05.13" "X2023.05.20" "X2023.05.27" "X2023.06.03" "X2023.06.10" "X2023.06.17" "X2023.06.24"
[289] "X2023.07.01" "X2023.07.08" "X2023.07.15" "X2023.07.22" "X2023.07.29" "X2023.08.05" "X2023.08.12" "X2023.08.19"
[297] "X2023.08.26" "X2023.09.02" "X2023.09.09" "X2023.09.16" "X2023.09.23" "X2023.09.30" "X2023.10.07" "X2023.10.14"
[305] "X2023.10.21" "X2023.10.28" "X2023.11.04"
```

<u>Exercise 1:</u> [10pts] Let's keep RegionName, and all the dates remove the prefix X to bring the dates back to their standard format and display the first ten rows and columns of the dataset as shown in below Figure.



```
      RegionName 2018.01.27 2018.02.03 2018.02.10 2018.02.17 2018.02.24 2018.03.03 2018.03.10 2018.03.17 2018.03.24
1  United States         49         44         38         33         28         25         23         22         21
2   New York, NY         88         86         81         74         60         49         41         36         36
3 Los Angeles, CA        25         20         16         15         15         15         15         15         15
4    Chicago, IL         62         53         45         35         26         22         18         17         15
5     Dallas, TX         39         35         31         28         25         23         23         23         22
6    Houston, TX         42         38         34         30         26         24         22         20         19
7 Washington, DC         51         48         42         39         36         32         31         28         27
8 Philadelphia, PA       66         60         58         56         51         45         38         33         30
9      Miami, FL         52         51         48         45         42         40         39         38         38
10   Atlanta, GA         30         25         21         17         15         14         13         13         12
```

<u>Exercise 2</u>: [10pts] Let's construct two new variables in the dataset `ageofIn:` namely month and age where month represents the dates and age is the values associated with the dates as shown in following Figure.

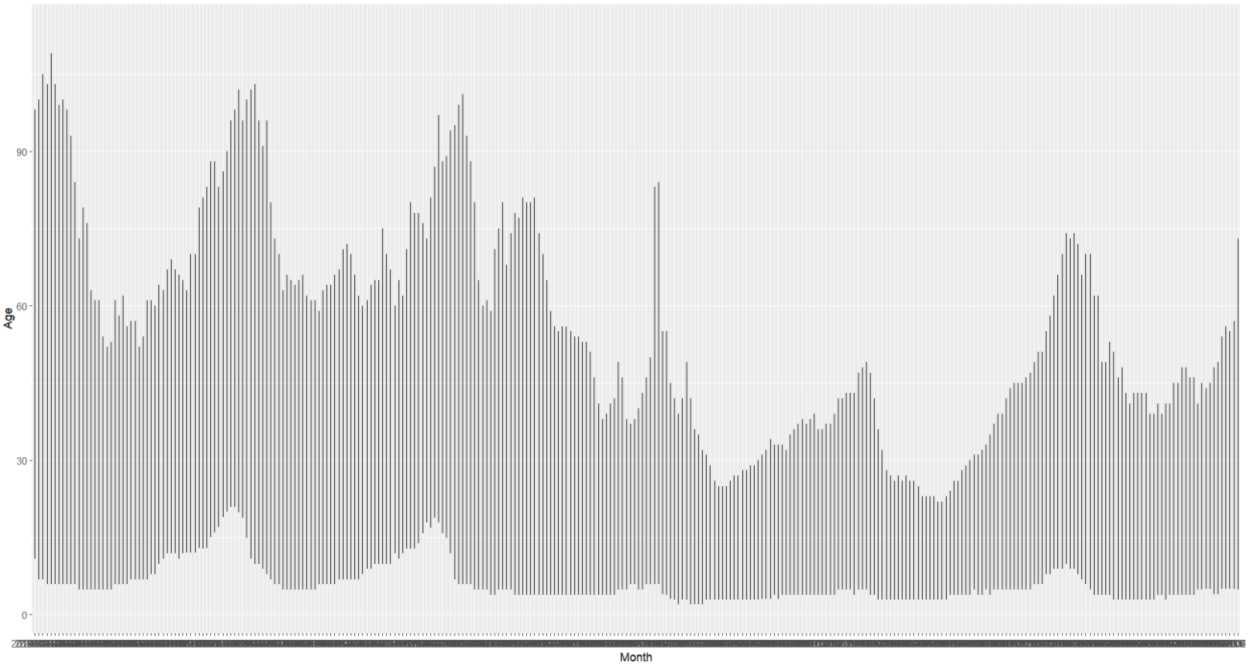


```
# A tibble: 89,694 x 3
   RegionName    Month        Age
   <chr>         <date>     <dbl>
 1 United States 2018-01-27    49
 2 United States 2018-02-03    44
 3 United States 2018-02-10    38
 4 United States 2018-02-17    33
 5 United States 2018-02-24    28
 6 United States 2018-03-03    25
 7 United States 2018-03-10    23
 8 United States 2018-03-17    22
 9 United States 2018-03-24    21
10 United States 2018-03-31    20
# i 89,684 more rows
# i Use `print(n = ...)` to see more rows
```

2. Starting Plot

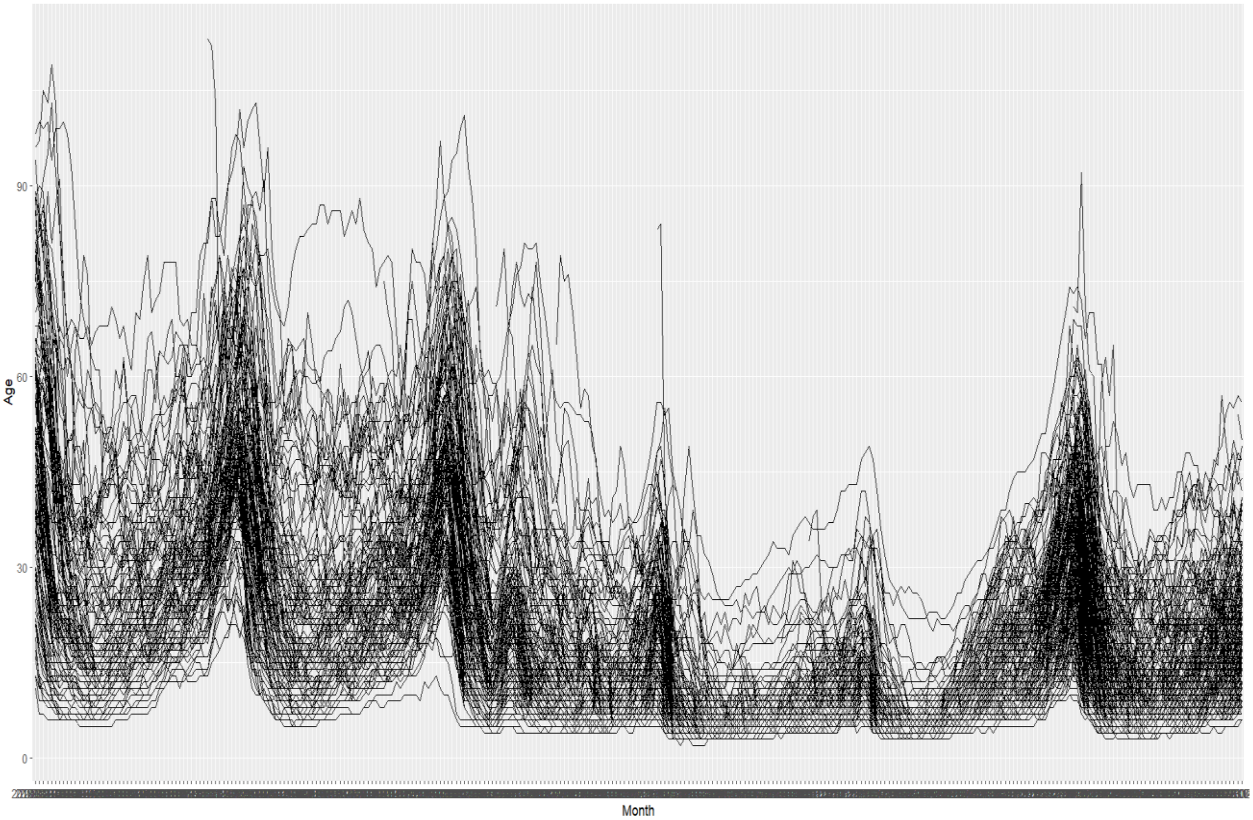
Let's start with a simple line plot of all these series. Let's show what happens if we leave out the group and color aesthetic.

```
ageofIn_long %>% ggplot(aes(Month, Age)) + geom_line()
```

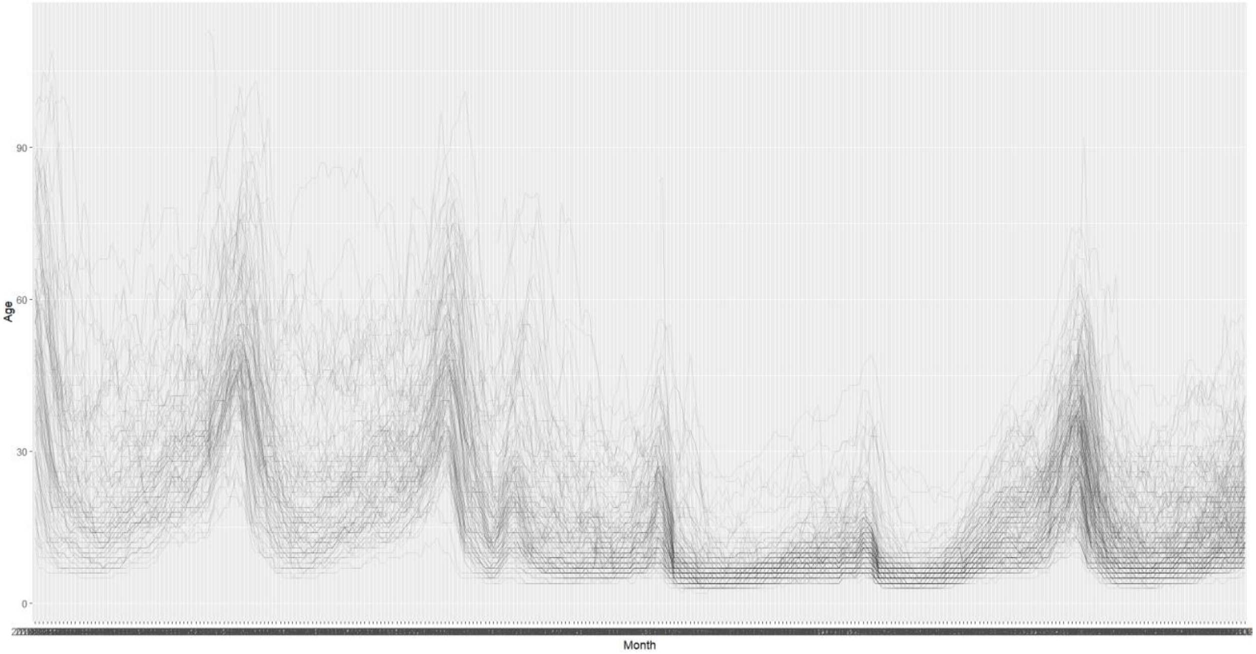


Now, let's add in the group aes

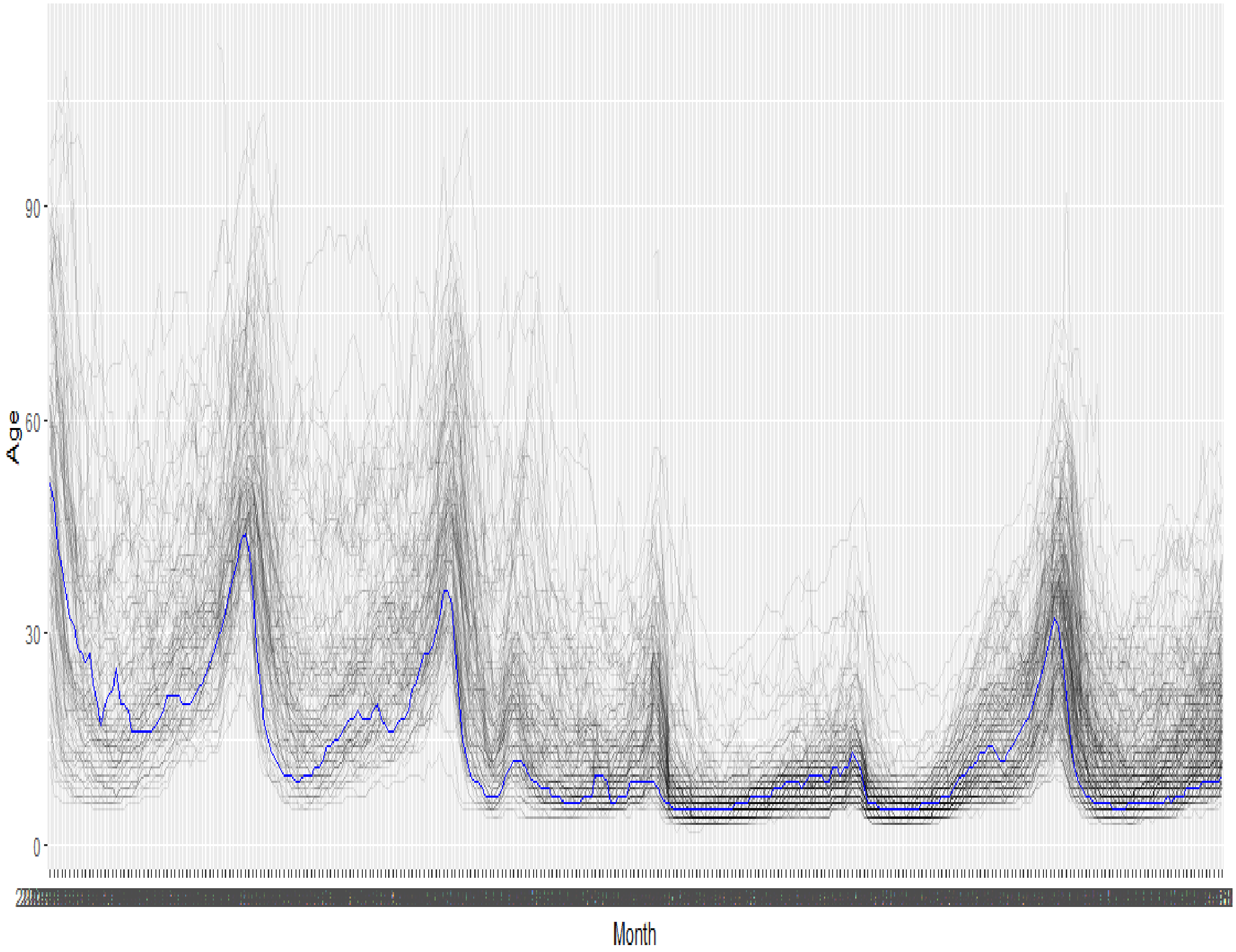
```
basic_plot <- ageofIn_long %>% ggplot(aes(Month, Age, group= RegionName)) +
geom_line()
```



Exercise 3: [**20pts**] It's hard to see what is happening in this tangled mess. Setting alpha to 0.1 will make this easier to untangle as shown in the Figure below.



<u>Exercise 4</u>: **[20pts]** Let's emphasize the line for Washington as demonstrated in Figure below.

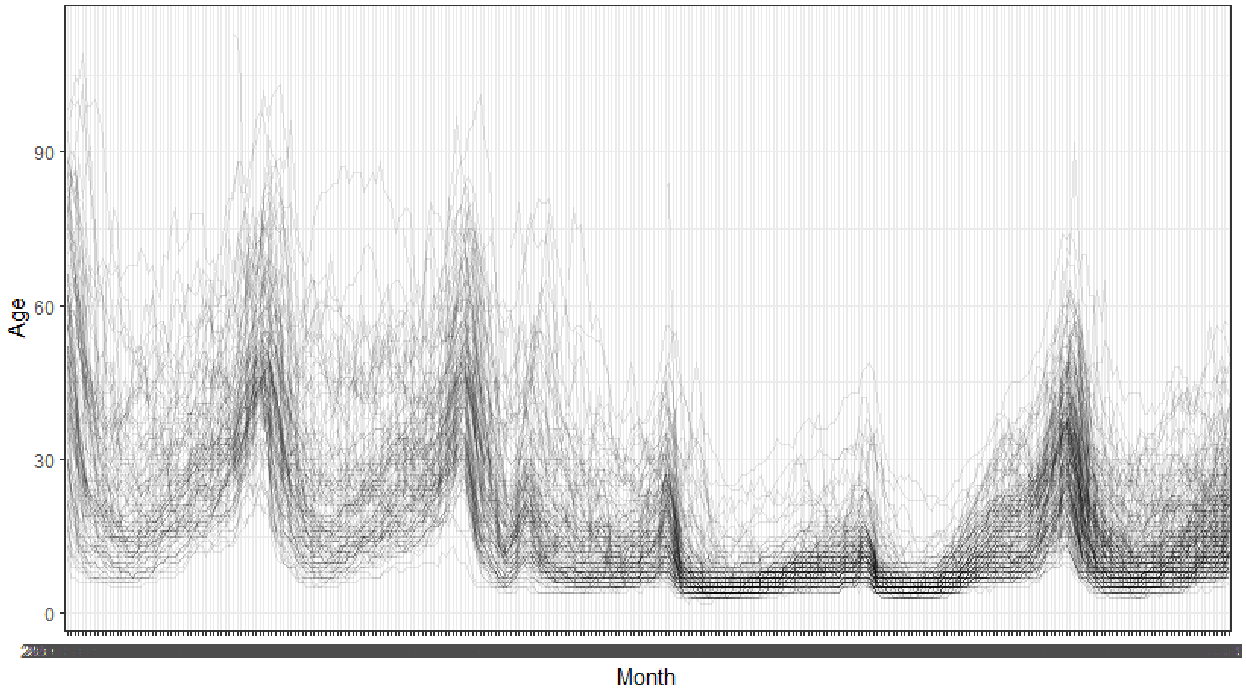


3. Themes

3.1. Complete Themes

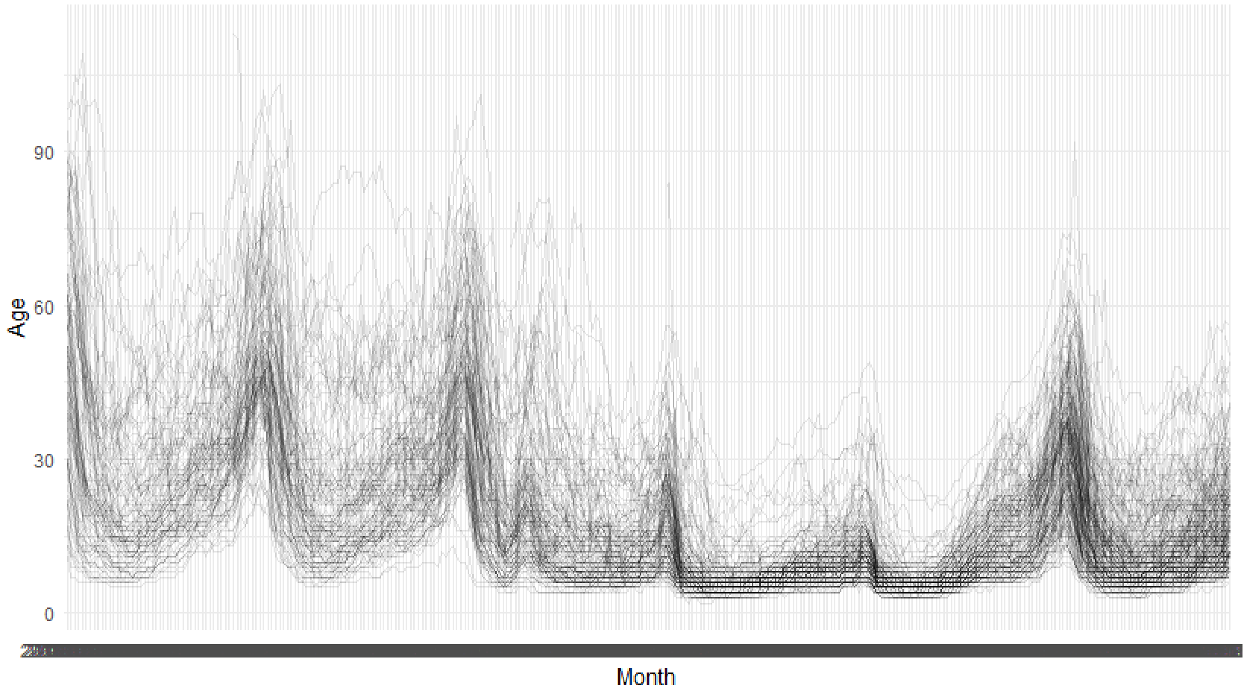
There are a variety of pre-made themes that can make our figures look cleaner (http://ggplot2.tidyverse.org/reference/ggtheme.html). theme_bw() is good if you don't want to print all the grey from the default, but you still want the same basic structure.

```
basic_plot + theme_bw()
```



theme_minimal() removes some of the visual clutter, removing the plot border and the axis ticks.

```
basic_plot + theme_minimal()
```



theme_void() goes the whole way and removes everything, but the data. It even removes the axis labels.

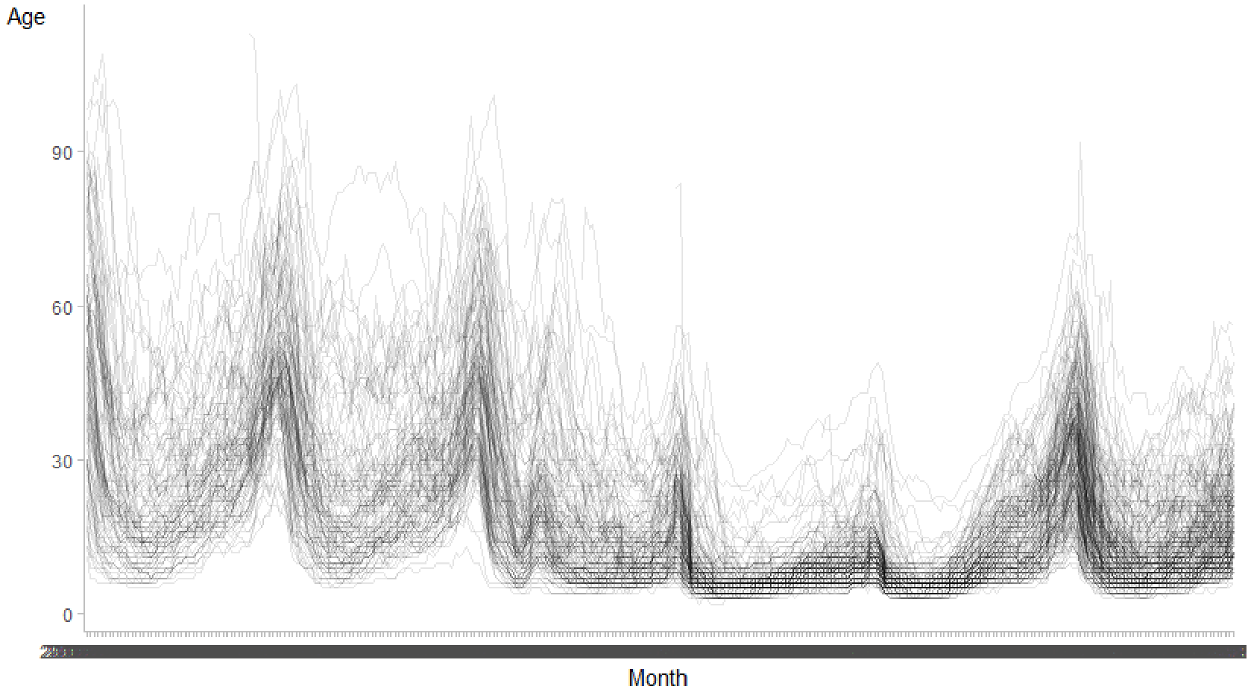
```
basic_plot + theme_void()
```

3.2. Modifying a theme

To modify a theme, we just add a call to the theme() function and assign new values to the parts of the plot we want to change (see the theme() reference for more examples: http://ggplot2.tidyverse.org/reference/theme.html).

Let’s start with the theme_bw() and make the chart more minimal.

```
basic_plot + theme_bw() + theme(
  panel.border = element_blank(),
  panel.grid = element_blank(),
  axis.line = element_line(color = "grey"),
  axis.ticks = element_line(color = "grey"),
  axis.title.y = element_text(angle = 0)
)
```



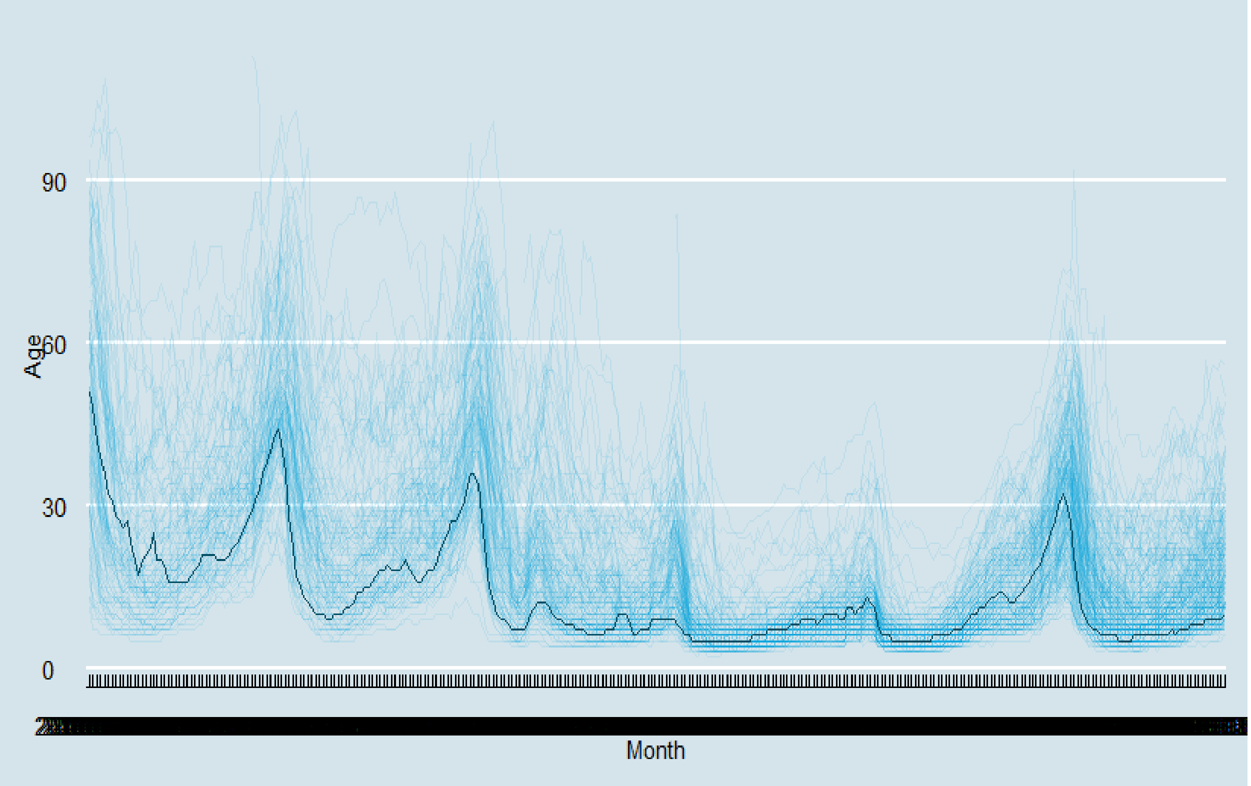
3.3. Modifying a theme

The ggthemes package adds a large set of fun themes. See the vignette at https://cran.r-project.org/web/packages/ggthemes/vignettes/ggthemes.html or enter the following command locally after installing the package

```
install.packages("ggthemes")
```

To make our chart look like it came out of the economist, let's use theme_economist(). To make the line colors work, we'll use the economist_pal() color pal.

Exercise 5: **[20pts]** Let's modify the theme using the theme_economist() to create the image as follows:

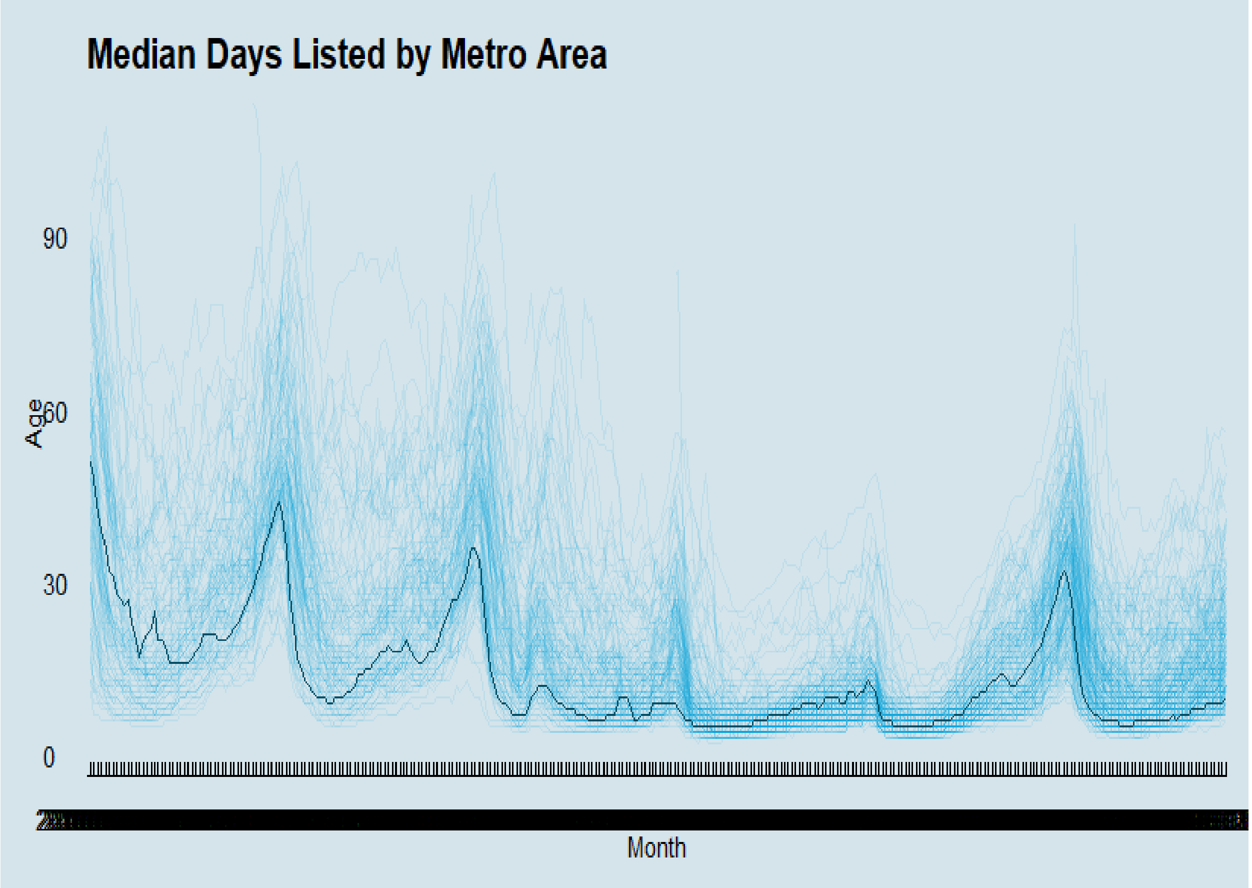


4. Labels

Now that we have a nice looking basic chart, we need to make sure our labels are in the right places and give enough information.

Let's start by adding a title. For a time series like this, using the name of the variable on the x-axis is a good start. We can also change the x-axis to break at each year, which makes the seasonality of this series even easier to pick out. With these added vertical lines, our chart will be more readable if we remove the horizontal gridlines (panel.grid.major.y).

<u>Exercise 6</u>: **[20pts]** Let's create a title for our image as illustrated below.



<u>What to submit</u>:

Your submission should include the following:

1. Lab report answers to all exercises above and source code.
2. Please create a folder called "yourname_studentID_lab7" that includes all the required files and generate a zip file called "yourname_studentID_lab7.zip".
3. Please submit your work (.zip) to Blackboard.