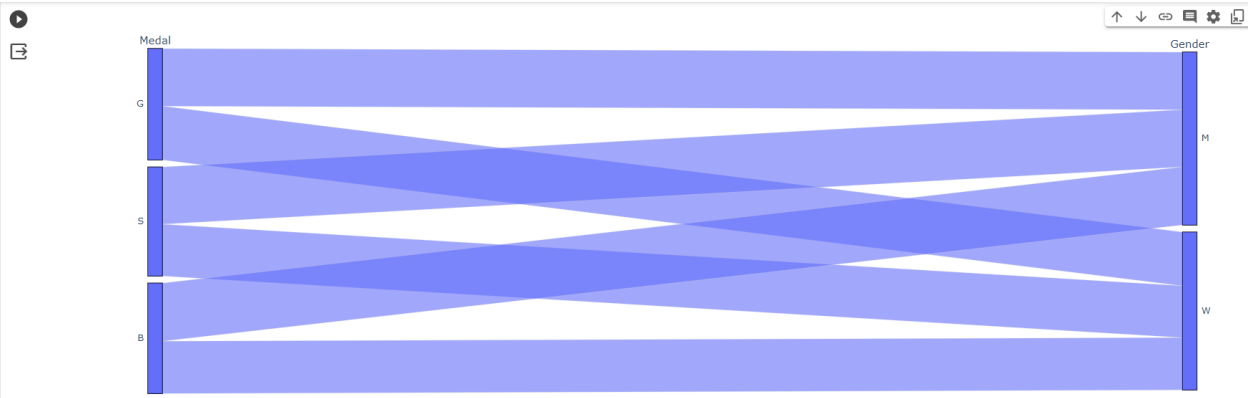
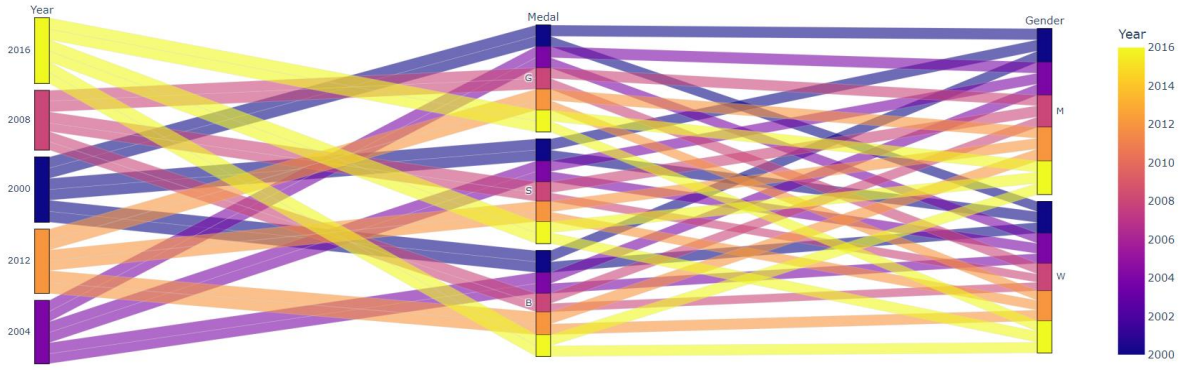
Homework Assignment #4

Q1 Please use Pandas to read olympic_medals.csv and use parallel_categories function from plotly.express to visualize proportions of medal type for each gender from since year 2000. Please see the example in the Python notebook we walked through in the class.

```
[53] #please use this cell to read and select your data
     df=pd.read_csv('/content/drive/MyDrive/DATA/olympic_medals.csv')
     df.head()
     df=df[(df['Year']>=2000)]
     plt.style.use('ggplot')
     px.parallel_categories(df[['Medal', 'Gender']])
```

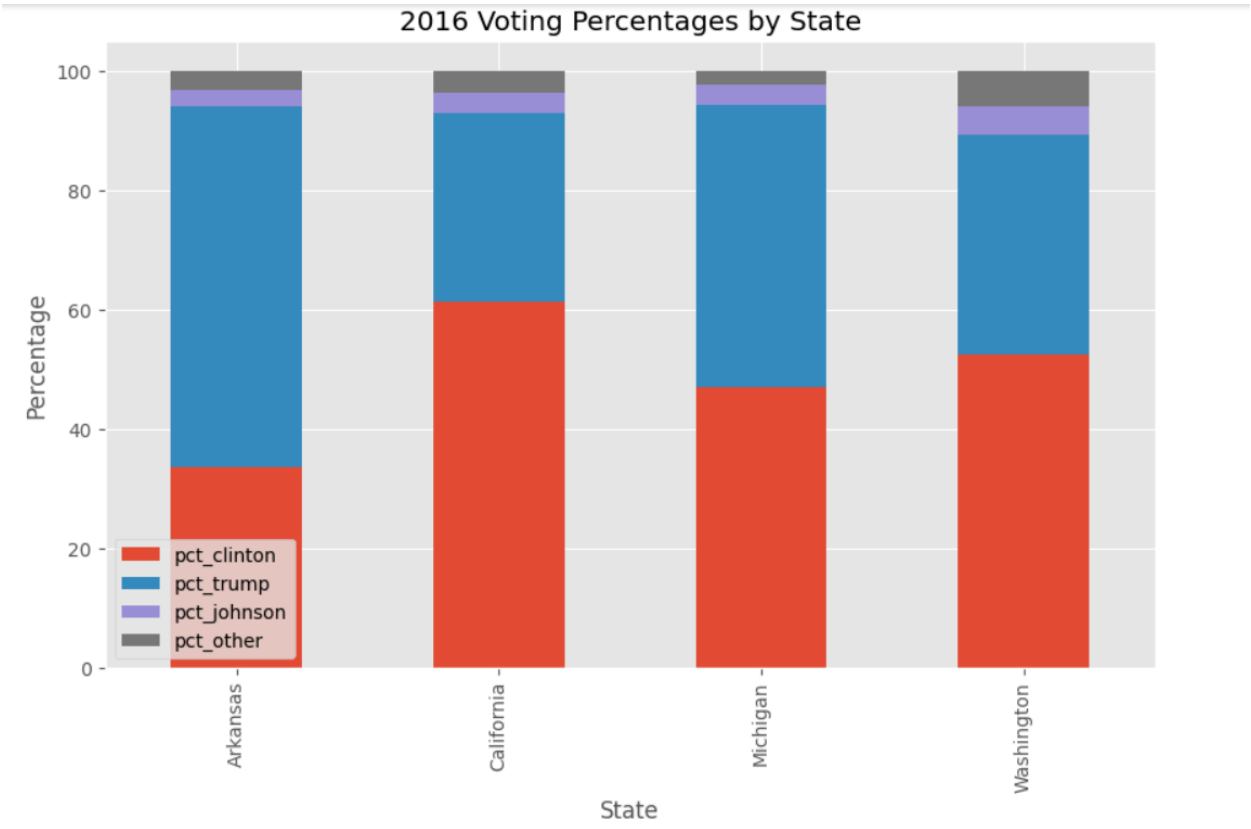



```
#Please use this cell to create your your figure. Please use Year column to color your graph.
px.parallel_categories(df, dimensions=['Year', 'Medal', 'Gender'], color="Year")
```



Q2 Please inspect the code below and observe how values are plotted by running it. Then, read the 2016elections.csv from the DATA folder and select rows for AR, MI, CA, and WI. Then, utilize stacked bar plot, to stack vote percentages for Trump, Clinton, Johnson, and Others. Please see 'pct_clinton', 'pct_trump', 'pct_johnson', 'pct_other' columns. Make sure that your x tick labels are those four states above.

```
[56] #You can use this cell to write your code. It is doable at most 4 lines of code.
     allowedStates=['CA','AR','MI','WA']
     df=pd.read_csv('/content/drive/MyDrive/DATA/2016elections.csv')[pd.read_csv('/content/drive/MyDrive/DATA/2016elections.csv')['st'].isin(allowedStates)]
     (df.set_index('state')[['pct_clinton','pct_trump', 'pct_johnson', 'pct_other']]
        .plot(kind='bar', stacked=True, figsize=(10, 6))
        .set(xlabel='State', ylabel='Percentage', title='2016 Voting Percentages by State'))
```



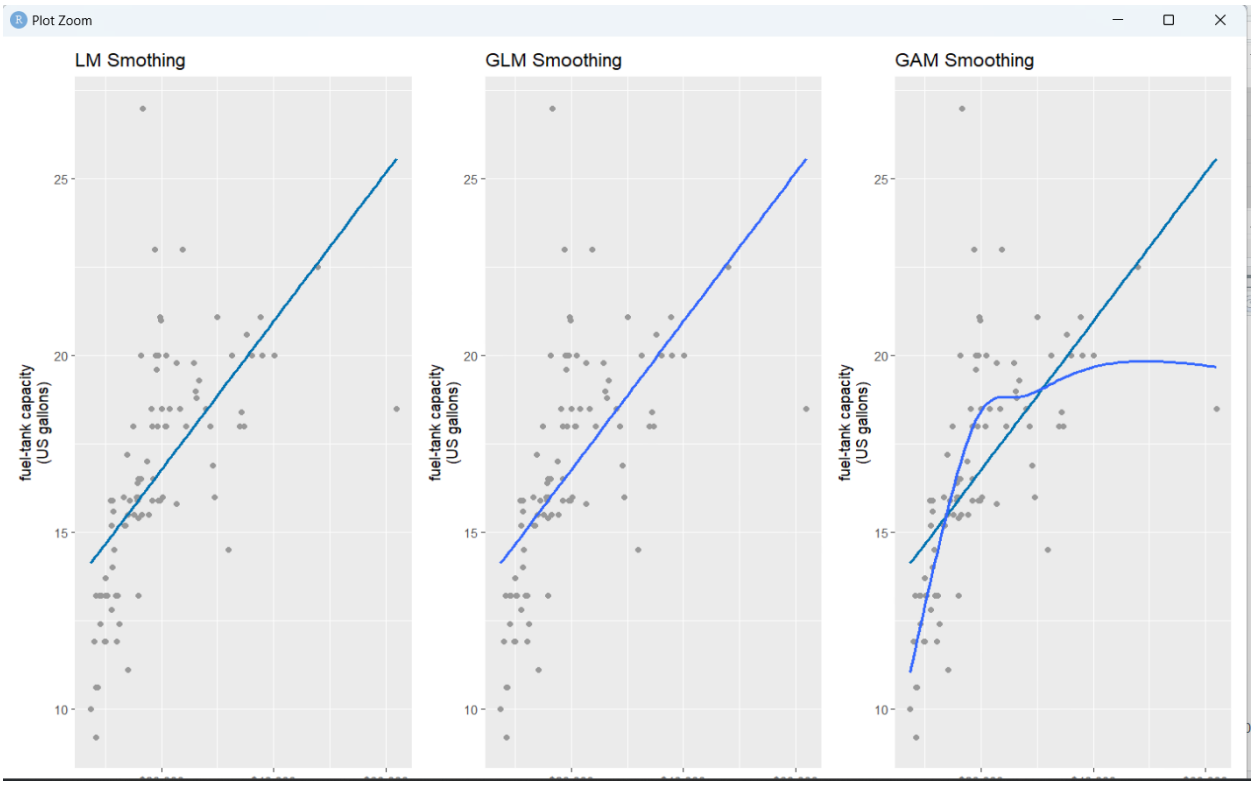
3.A. (a) Use ”lm”, ”glm”, ”gam” methods in the geom_smooth() function to create three figures.

kframbro_Homework4_partII.R* | Untitled1*

Source on Save | Run | Source

```
1
2  library(tidyr)
3  library(ggplot2)
4  library(dplyr)
5  library(gridExtra)
6  library(patchwork)
7  cars93 <- MASS::Cars93
8
9  #Question #3A (plot variables = Question#+Plot#)
10
11 q3a1<-ggplot(cars93, aes(x = Price, y = Fuel.tank.capacity)) +
12   geom_point(color = "grey60") +
13   geom_smooth(se = FALSE, method = "lm",formula = y ~ x, color = "#0072B2") +
14   scale_x_continuous(
15     name = "price (USD)",
16     breaks = c(20, 40, 60),
17     labels = c("$20,000", "$40,000", "$60,000")
18   ) +
19   scale_y_continuous(name = "fuel-tank capacity\n(US gallons)")+
20   ggtitle('LM Smothing')
21 q3a2<-q3a1 +geom_smooth(se=FALSE,method="glm")+
22   ggtitle('GLM Smoothing')
23 q3a3<-q3a1 +geom_smooth(se=FALSE,method="gam")+
24   ggtitle('GAM Smoothing')
25 q3a1+q3a2+q3a3
```

25:1 (Top Level) R Script

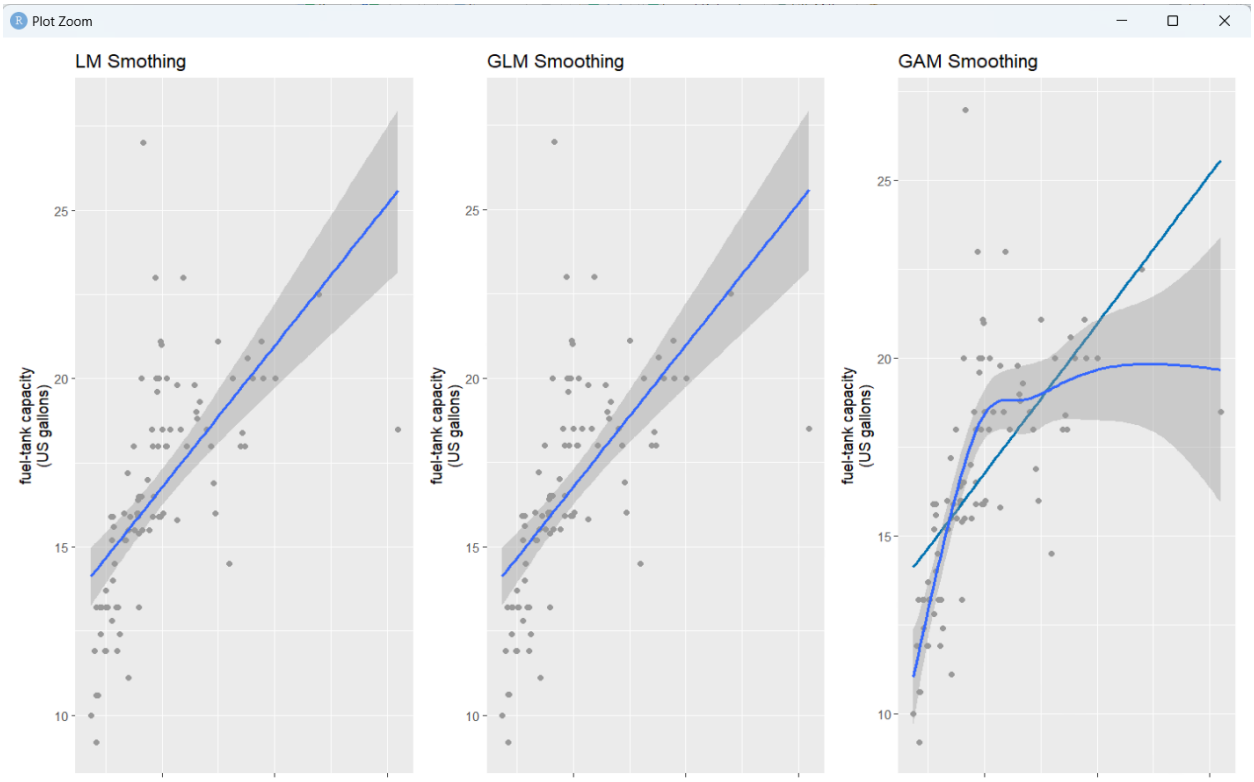


Question 3B: (b) Set the se parameter to TRUE to show the standard error (shaded area around the fitted line)

kframbro_Homework4_partII.R* | Untitled1*

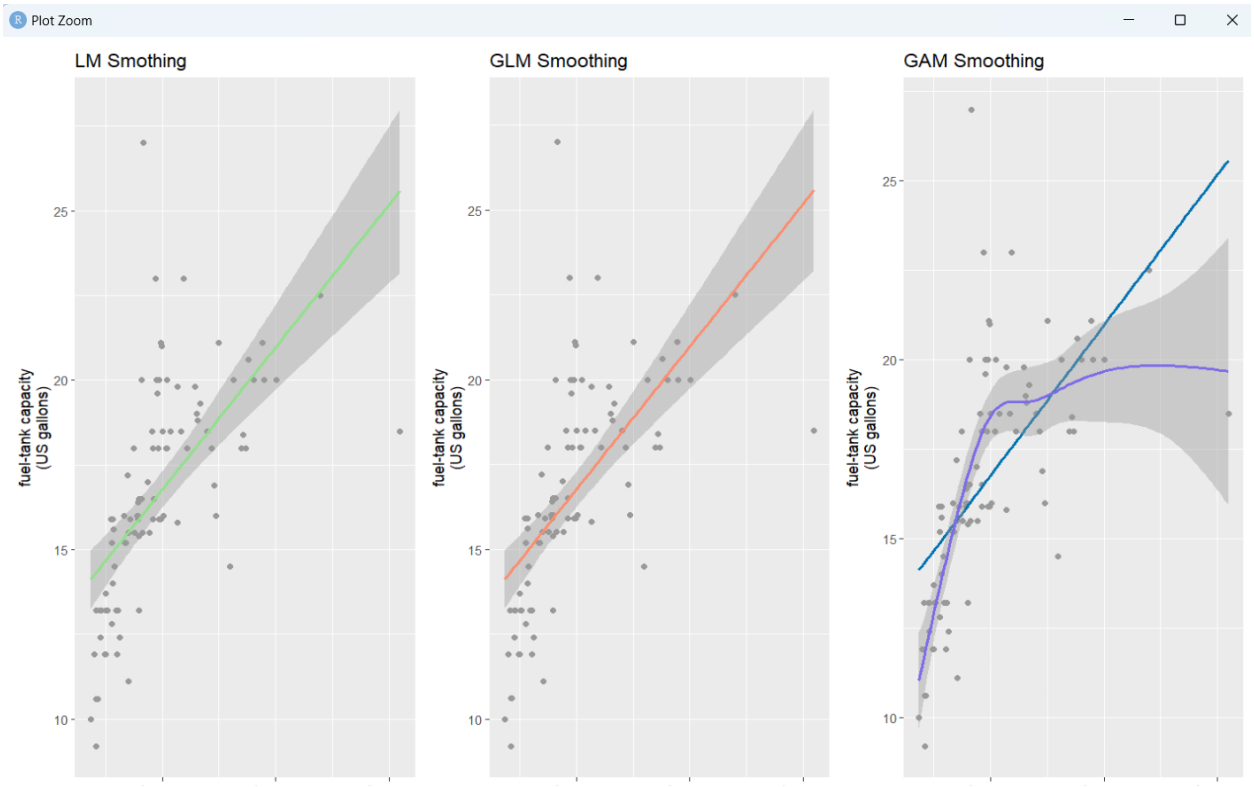
Source on Save | Run | Source

```
#Question #3A (plot variables = Question#+Plot#)

q3a1<-ggplot(cars93, aes(x = Price, y = Fuel.tank.capacity)) +
  geom_point(color = "grey60") +
  geom_smooth(se = FALSE, method = "lm",formula = y ~ x, color = "#0072B2") +
  scale_x_continuous(
    name = "price (USD)",
    breaks = c(20, 40, 60),
    labels = c("$20,000", "$40,000", "$60,000")
  ) +
  scale_y_continuous(name = "fuel-tank capacity\n(US gallons)")+
  ggtitle('LM Smothing')
q3a2<-q3a1 +geom_smooth(se=FALSE,method="glm")+
  ggtitle('GLM Smoothing')
q3a3<-q3a1 +geom_smooth(se=FALSE,method="gam")+
  ggtitle('GAM Smoothing')
q3a1+q3a2+q3a3

#Question3B Using the variables from the previous question
q3b1<-q3a1+geom_smooth(se=TRUE,method="lm")
q3b2<-q3a2 +geom_smooth(se=TRUE,method="glm")
q3b3<-q3a3 +geom_smooth(se=TRUE,method="gam")

q3b1+q3b2+q3b3
```

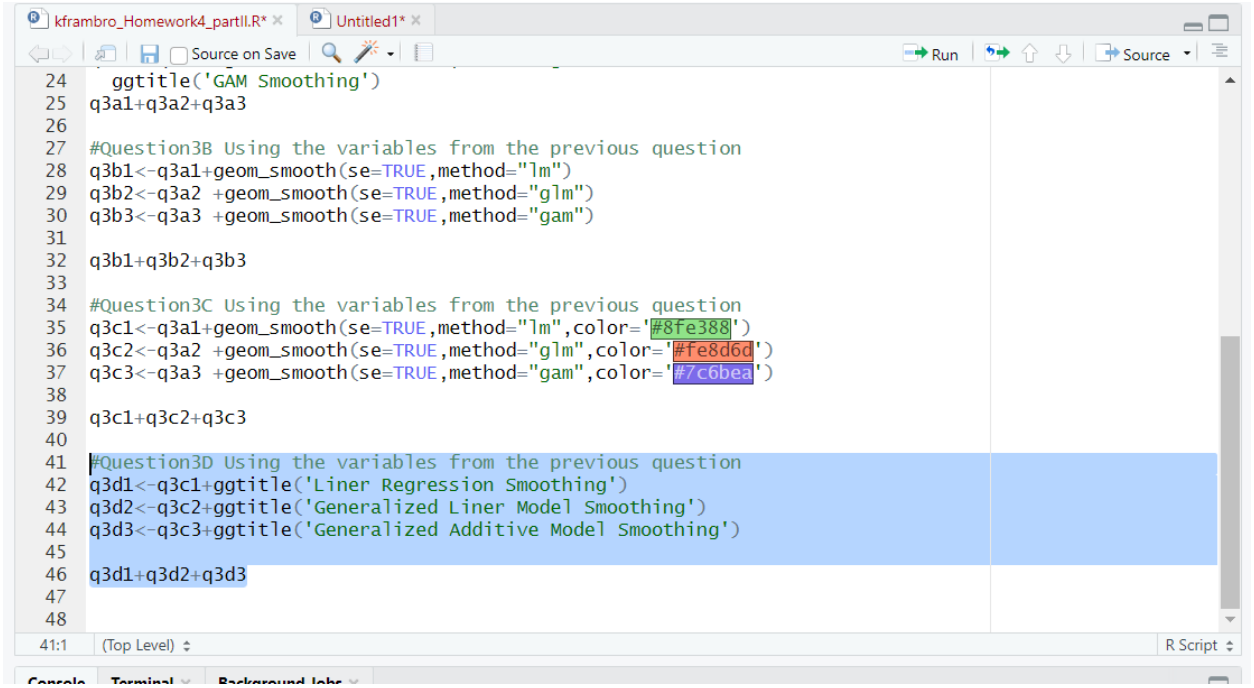


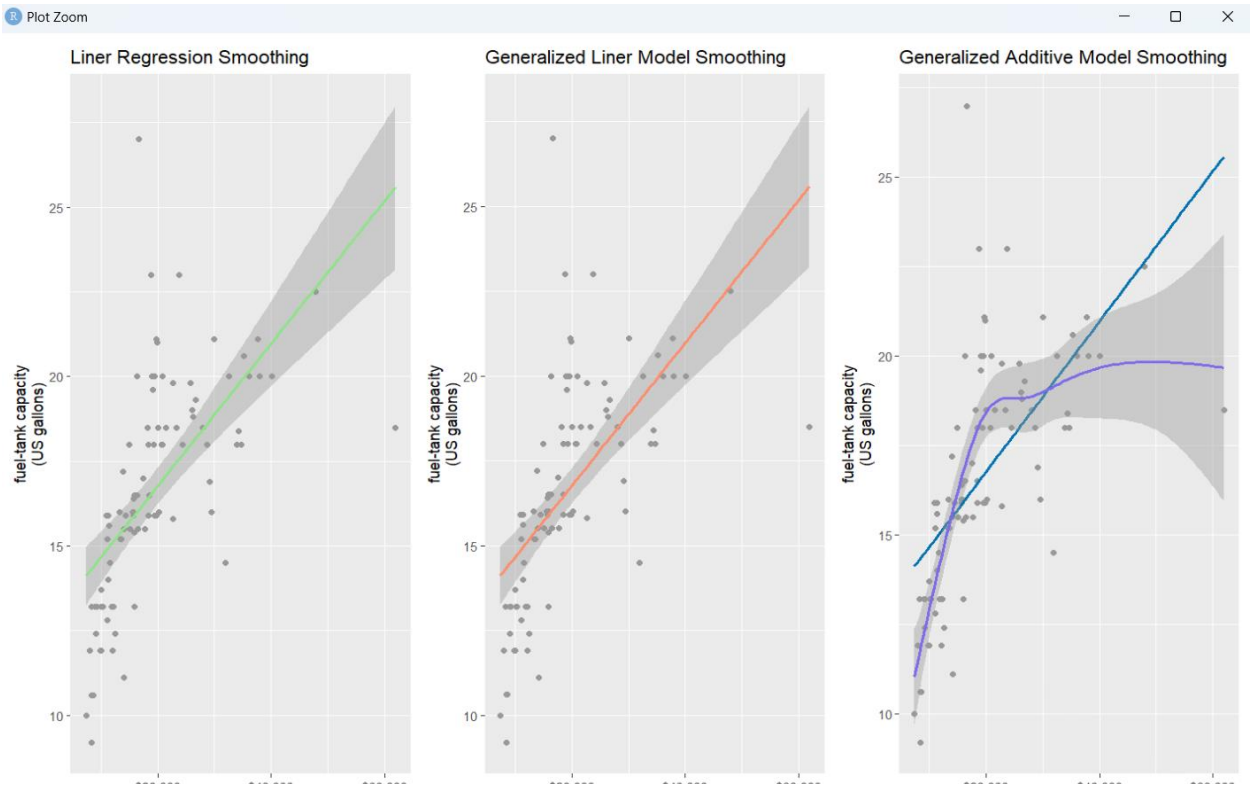
Question 3C: For every method above change the color of the line with the following color codes: #8fe388, #fe8d6d, #7c6bea

```
    breaks = c(20, 40, 60),
    labels = c("$20,000", "$40,000", "$60,000")
  ) +
  scale_y_continuous(name = "fuel-tank capacity\n(US gallons)")+
  ggtitle('LM Smothing')
q3a2<-q3a1 +geom_smooth(se=FALSE,method="glm")+
  ggtitle('GLM Smoothing')
q3a3<-q3a1 +geom_smooth(se=FALSE,method="gam")+
  ggtitle('GAM Smoothing')
q3a1+q3a2+q3a3

#Question3B Using the variables from the previous question
q3b1<-q3a1+geom_smooth(se=TRUE,method="lm")
q3b2<-q3a2 +geom_smooth(se=TRUE,method="glm")
q3b3<-q3a3 +geom_smooth(se=TRUE,method="gam")

q3b1+q3b2+q3b3

#Question3C Using the variables from the previous question
q3c1<-q3a1+geom_smooth(se=TRUE,method="lm",color='#8fe388')
q3c2<-q3a2 +geom_smooth(se=TRUE,method="glm",color='#fe8d6d')
q3c3<-q3a3 +geom_smooth(se=TRUE,method="gam",color='#7c6bea')

q3c1+q3c2+q3c3
```



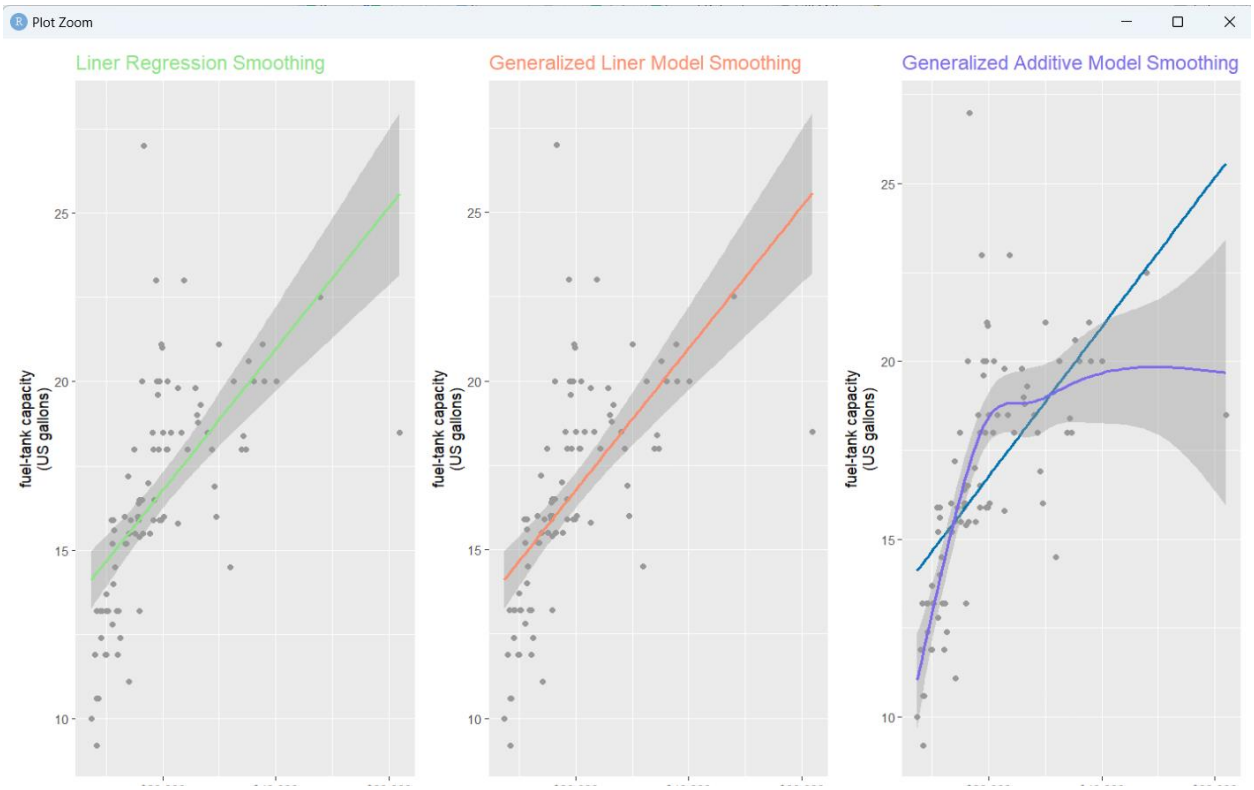
Question 3d: Please search for the method to add a title to your ggplot figure and add titles for each figure to indicate the method that you used for smoothing.





Question 3E: Please search for the theme() function for ggplot and change the font size of the titles to 14 and match their colors with the line colors you used above.

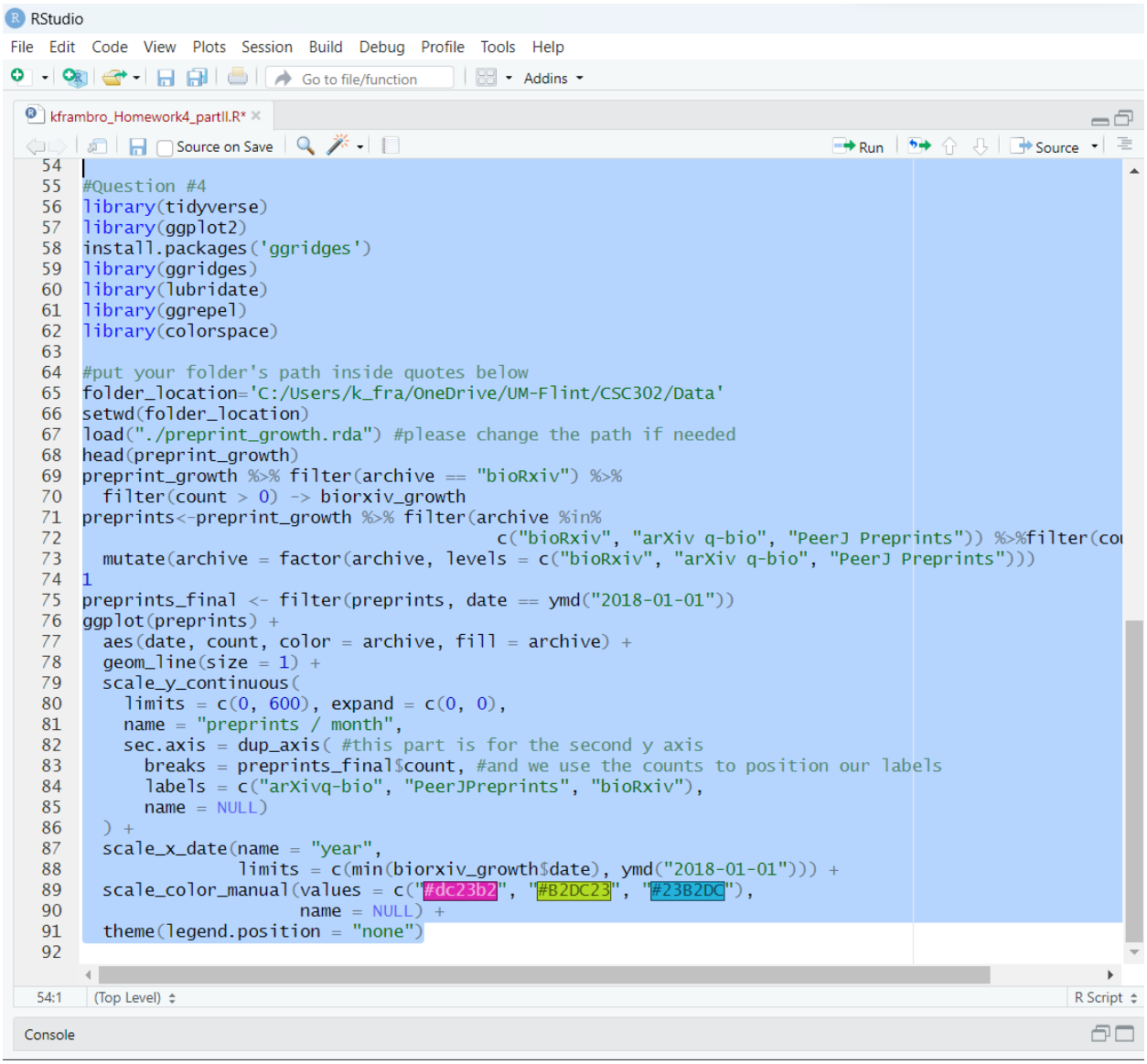
```
kframbro_Homework4_partII.R*   Untitled1*
Source on Save   Run   Source
29 q3b2<-q3a2 +geom_smooth(se=TRUE,method="glm")
30 q3b3<-q3a3 +geom_smooth(se=TRUE,method="gam")
31
32 q3b1+q3b2+q3b3
33
34 #Question3C Using the variables from the previous question
35 q3c1<-q3a1+geom_smooth(se=TRUE,method="lm",color='#8fe388')
36 q3c2<-q3a2 +geom_smooth(se=TRUE,method="glm",color='#fe8d6d')
37 q3c3<-q3a3 +geom_smooth(se=TRUE,method="gam",color='#7c6bea')
38
39 q3c1+q3c2+q3c3
40
41 #Question3D Using the variables from the previous question
42 q3d1<-q3c1+ggtitle('Liner Regression Smoothing')
43 q3d2<-q3c2+ggtitle('Generalized Liner Model Smoothing')
44 q3d3<-q3c3+ggtitle('Generalized Additive Model Smoothing')
45
46 q3d1+q3d2+q3d3
47
48 #Question3e Using the variables from the previous question
49 q3e1<-q3d1+theme(plot.title=element_text(size=14,color='#8fe388'))
50 q3e2<-q3d2+theme(plot.title=element_text(size=14,color='#fe8d6d'))
51 q3e3<-q3d3+theme(plot.title=element_text(size=14,color='#7c6bea'))
52
53 q3e1+q3e2+q3e3
48:1   (Top Level)   R Script
```

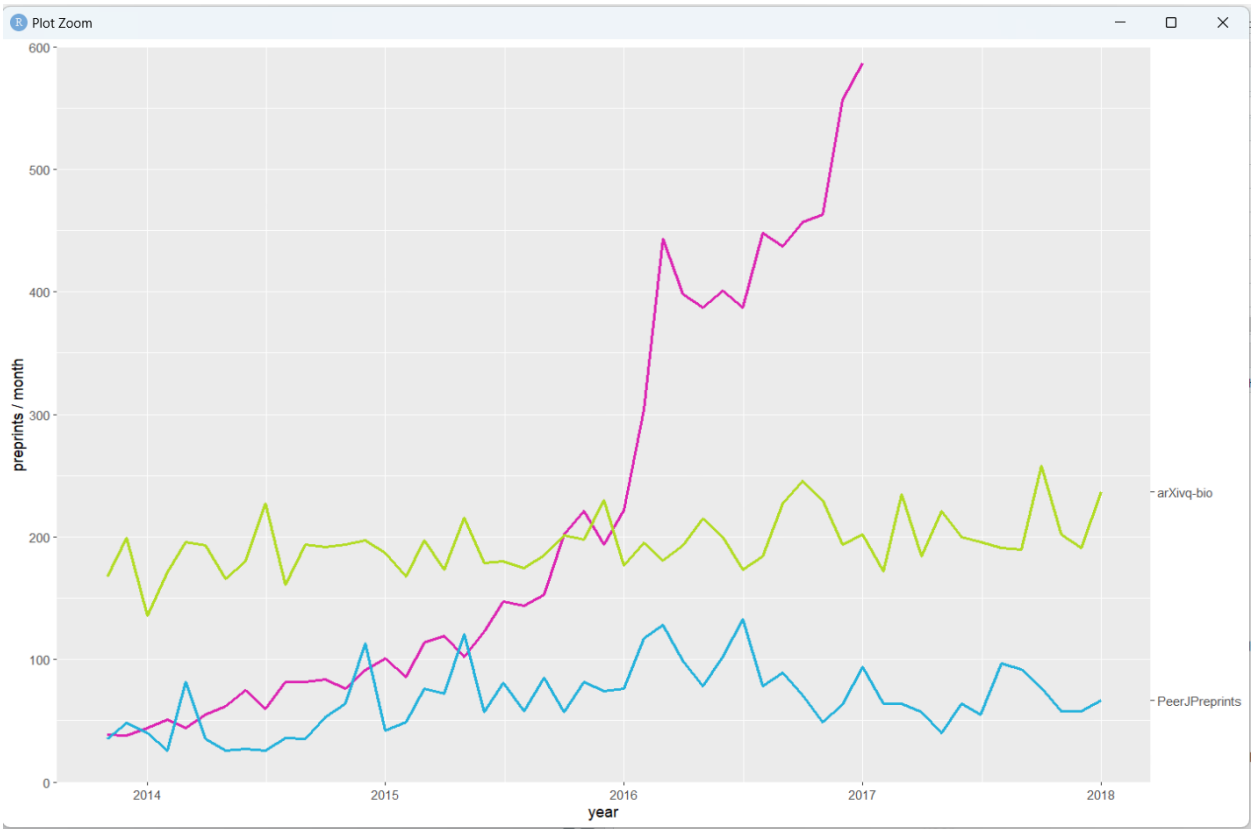


\

Question 4: Please inspect the following code which can be also found in TimeSeries_Trends.R and try to run how it generates three time series in a single plot. Then, modify the start date and the manual coloring as you want to get a different version of the chart. Please indicate what you changed and submit the figure you created as a response to this question.

Changes made, date changed to 2018-01-01, colors changed to "#dc23b2", "#B2DC23", "#23B2DC"


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
kframbro_Homework4_partII.R*
Source on Save
Run
Source
#Question #4
library(tidyverse)
library(ggplot2)
install.packages('ggridges')
library(ggridges)
library(lubridate)
library(ggrepel)
library(colorspace)
#put your folder's path inside quotes below
folder_location='C:/Users/k_fra/OneDrive/UM-Flint/CSC302/Data'
setwd(folder_location)
load("./preprint_growth.rda") #please change the path if needed
head(preprint_growth)
preprint_growth %>% filter(archive == "bioRxiv") %>%
filter(count > 0) -> biorxiv_growth
preprints<-preprint_growth %>% filter(archive %in%
c("bioRxiv", "arXiv q-bio", "PeerJ Preprints")) %>%filter(cou
mutate(archive = factor(archive, levels = c("bioRxiv", "arXiv q-bio", "PeerJ Preprints")))
1
preprints_final <- filter(preprints, date == ymd("2018-01-01"))
ggplot(preprints) +
aes(date, count, color = archive, fill = archive) +
geom_line(size = 1) +
scale_y_continuous(
limits = c(0, 600), expand = c(0, 0),
name = "preprints / month",
sec.axis = dup_axis(#this part is for the second y axis
breaks = preprints_final$count, #and we use the counts to position our labels
labels = c("arXivq-bio", "PeerJPreprints", "bioRxiv"),
name = NULL)
) +
scale_x_date(name = "year",
limits = c(min(biorxiv_growth$date), ymd("2018-01-01"))) +
scale_color_manual(values = c("#dc23b2", "#B2DC23", "#23B2DC"),
name = NULL) +
theme(legend.position = "none")
54:1
(Top Level)
R Script
Console


Plot Zoom
600
500
400
300
200
100
0
preprints / month
2014
2015
2016
2017
2018
year
arXivq-bio
PeerJPreprints