Section Assignment 3

Liam Mueller

1/13/2022

Please Read!

This week, your assignment is to generate a high quality figure from the heart surgery data we worked with last week. We will do this one step at a time generating 7 figures along the way that build on each other, one line of code at a time. As this assignment builds, I want to be sure that you will catch any error you might make in an early step, so it wont tank your entire assignment in the later, more complex code. To accomplish this, I have given you the grading rubric for this assignment. In the "Week3_SA_Grading_Rubric.pdf" you will see exactly what each of the 7 figures you will be graded on need to look like. At the end of each step, make sure the figure you create looks exactly like mine before moving on!

One of the traits of a great programmer is their ability to solve a problem they haven't seen before. One of the best ways to solve a problem you have not seen before is to see if anyone else has. Great programmers are experts of web searching. I know you are not an expert programmer yet, so this week I have included links to webpages that I used the first time I had to teach myself how to use ggplot specifically related to the question you are being asked. Copying from the internet is one of the foundations of learning how to program, but it only works as learning if you reflect on why the code you used works. For that reason, this week and in future weeks, you will need to annotate your code (Use the lines starting with a "#Note:" to explain in your own words what your code is doing. Remember the hash tag # tells R not to read a line of text and it is the way programmers leave notes for each other) You will be graded on not just your figures, but your explanation. Remember, in this class, it is okay to copy code, but you still need to demonstrate independent thought.

First, be sure the author line above (line 3) contains your full name

(for example, my line 3 reads: 'author: "Liam Mueller" ')

Second, change the date in line 4 to today's date

(for example, my line 4 reads: 'date: "01/13/2022" ')

Second, be sure not to change anything other than your name in the author line (line 3) and the date in the date line (line 4) in lines 1-12 as it will change how the file knits and is saved.

Answer the following question by typing the appropriate code in the code box (denoted by "' signs).

Questions:

1. (0.5 point): The goal for this week is to generate high quality figures from the hospital data that you worked with last week. We will use the ggplot2 package to accomplish this. Install the ggplot2 package and load it into your function library.

```
install.packages("ggplot2")
```

```
## Installing package into 'C:/Users/liamm/Documents/R/win-library/4.1'
## (as 'lib' is unspecified)
```

```
## Error in contrib.url(repos, "source"): trying to use CRAN without setting a mirror
```

```
library(ggplot2)
```

2.(0.5 point): You will need to read in the data table that you created in last week's section activity. Assign that .csv file to an object on the line below:

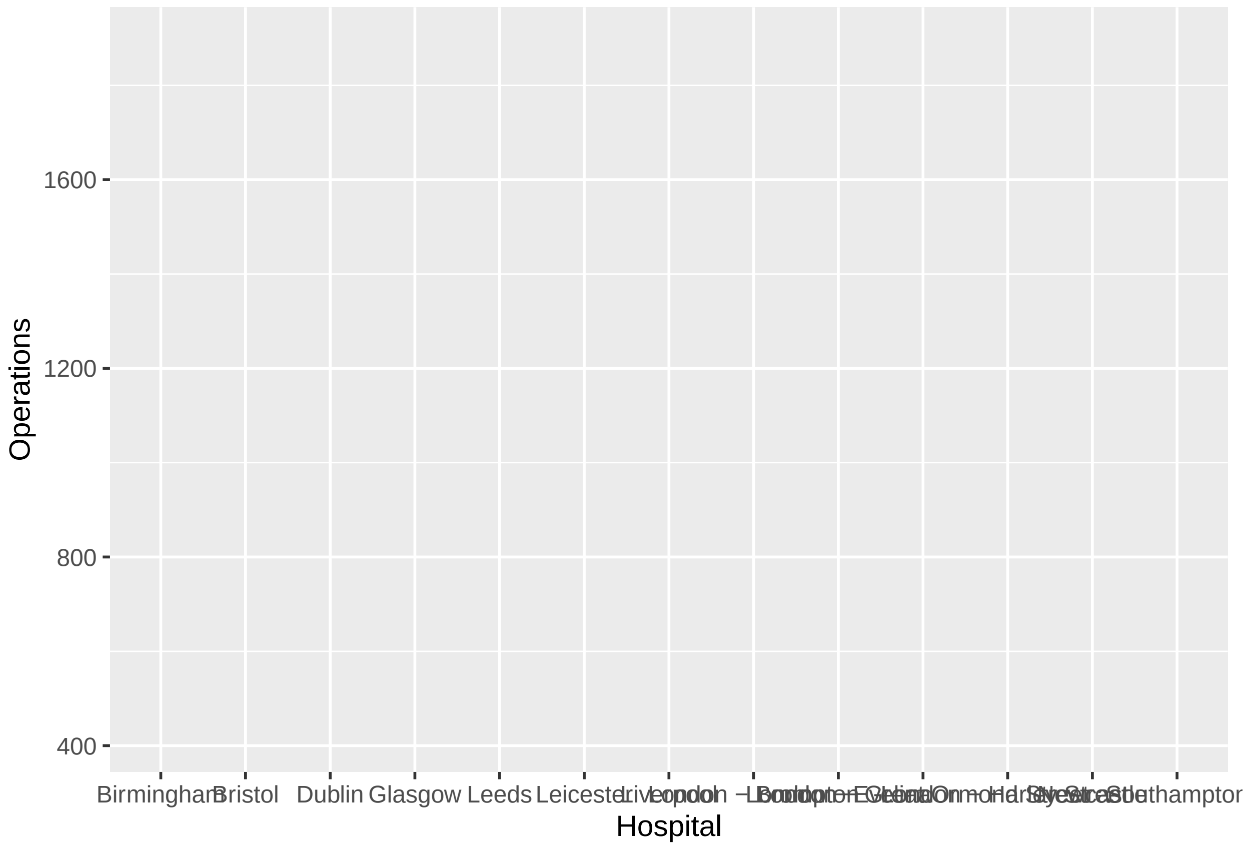
```
HospitalData<-read.csv(file = "week2SA.csv",header = TRUE,row.names = 1)
head(HospitalData)
```

```
##                  Hospital Operations Survivors Deaths SurvivalRate
## 1 London - Harley Street        418       413      5    0.9880383
## 2               Leicester        607       593     14    0.9769357
## 3               Newcastle        668       653     15    0.9775449
## 4                 Glasgow        760       733     27    0.9644737
## 5             Southampton        829       815     14    0.9831122
## 6                 Bristol        835       821     14    0.9832335
```

3. (9 points over 8 steps): Generate a bar chart representing the number of child heart operations each hospital performed during the study period where the color of the bar represents the survival rate in that hospital. We will do this in a number of steps:

Step 1 (1 point). Generate the empty space that we can fill with data. In this step we will define for ggplot the data set and the x and y axes. It will look super messy, and that is okay, in each step it will look more and more professional. Before moving on to step 2, check that your figure 1 looks identical to the one in the associated answer key.

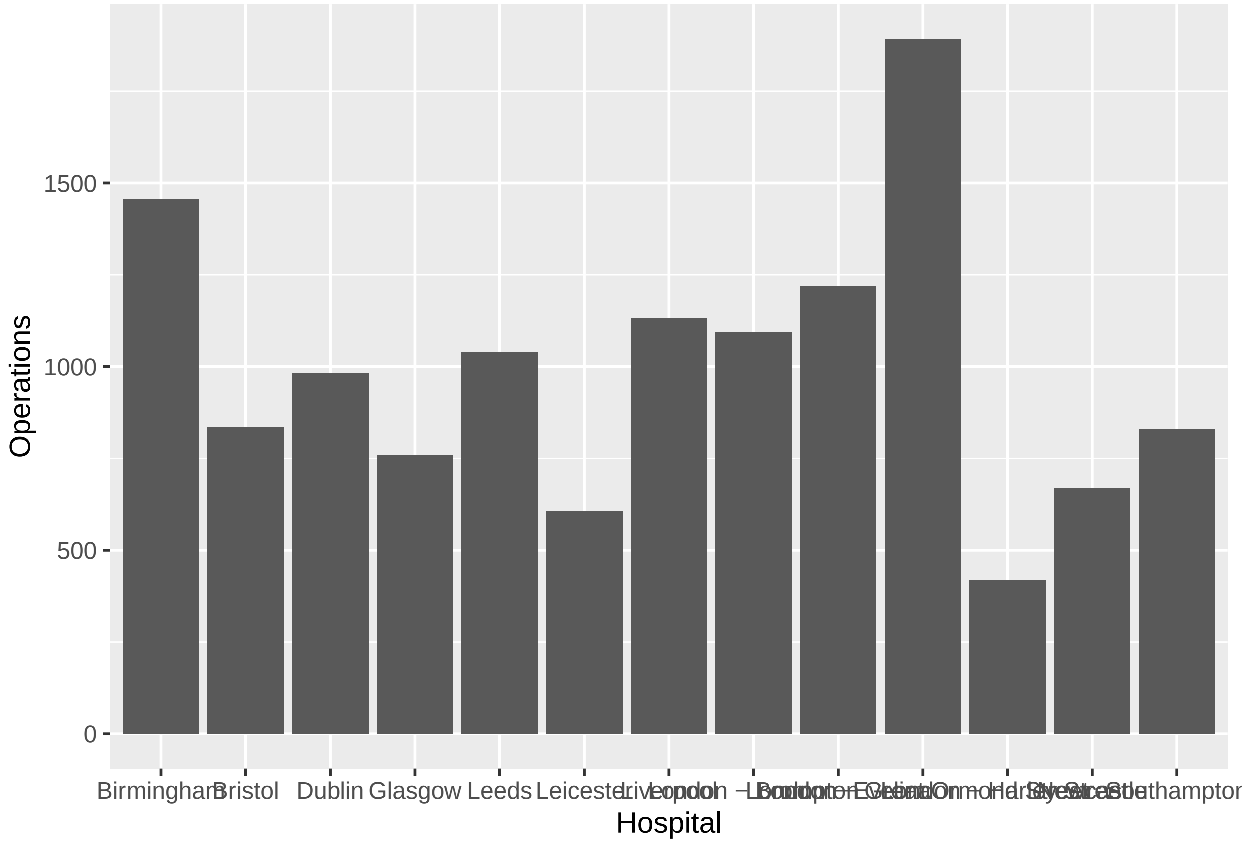
```r
fig1<- ggplot(data = HospitalData , aes(x= Hospital,y =Operations))
fig1
```



```r
#Note:
```

Step 2 (1 point): Now it is time to set up the type of plot we want to visualize, in this case a bar plot. Copy your code from step 1 into the box below, then add the appropriate code on a new line. Again, check your results with the associated pdf and move on to step 3 if your graph is identical to mine.

```
fig2<- ggplot(data = HospitalData , aes(x= Hospital,y =Operations))+
  geom_bar(stat="identity")
fig2
```

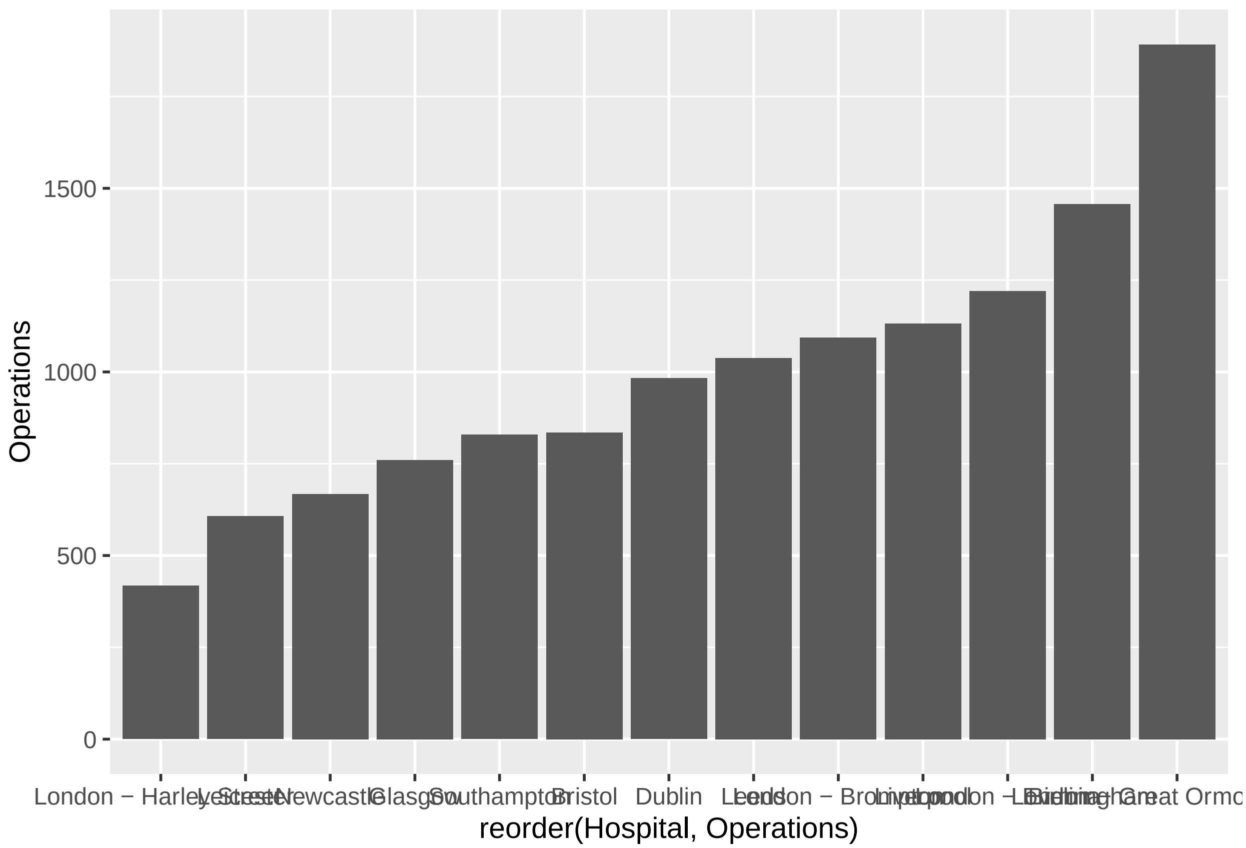


```
#Note:
```

step 3 (1 point): In this step we are going to reorder the bars in acceding order by the number of operations. Copy the code from step 2 into the box below and modify the code that reorders the x axis. Hint: Instead of adding a new line, this time we need to change an argument in how we assigned the X axis in the aes() argument that we first put in step 1. Check out the “sorting bars by some numeric variable” section of this link: https://sebastiansauer.github.io/ordering-bars/

Check your fig 3 with the answer key and if they look identical, move on to step 4.

```r
fig3<- ggplot(data = HospitalData , aes(x= reorder(Hospital,Operations),y =Operations))+
  geom_bar(stat="identity")
fig3
```

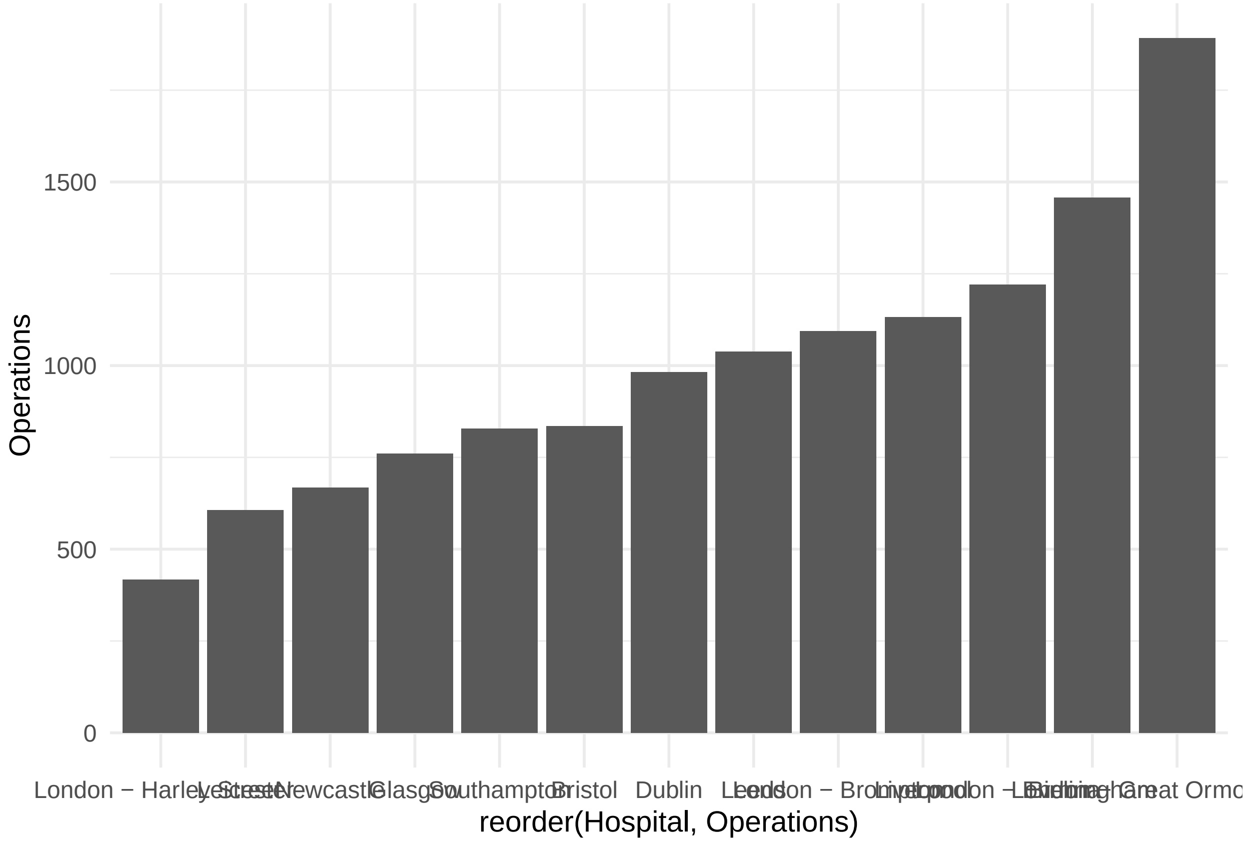


```r
#Note:
```

step 4 (1 point): Now that we have an organized plot, it is time to start working on legibility. First, we will start making changes that make it look more professional and less like an EXCEL output. The main thing making it look like an excel plot is that grey background that journals hate (to be fair, it just wastes ink). And don't worry, we will fix that messy x axis soon, we just need to have this line happen first. The easiest way to get rid of the grey background is by changing the plot theme. Check out how to create the "theme_minimal" theme here: https://r-charts.com/ggplot2/themes/ Copy your code from step 3 into the box below and add a line that modifies the theme to minimal.

Check your fig 4 with the answer key and if they look identical, move on to step 5.

```r
fig4<- ggplot(data = HospitalData , aes(x= reorder(Hospital,Operations),y =Operations))+
  geom_bar(stat="identity") +
  theme_minimal()
fig4
```

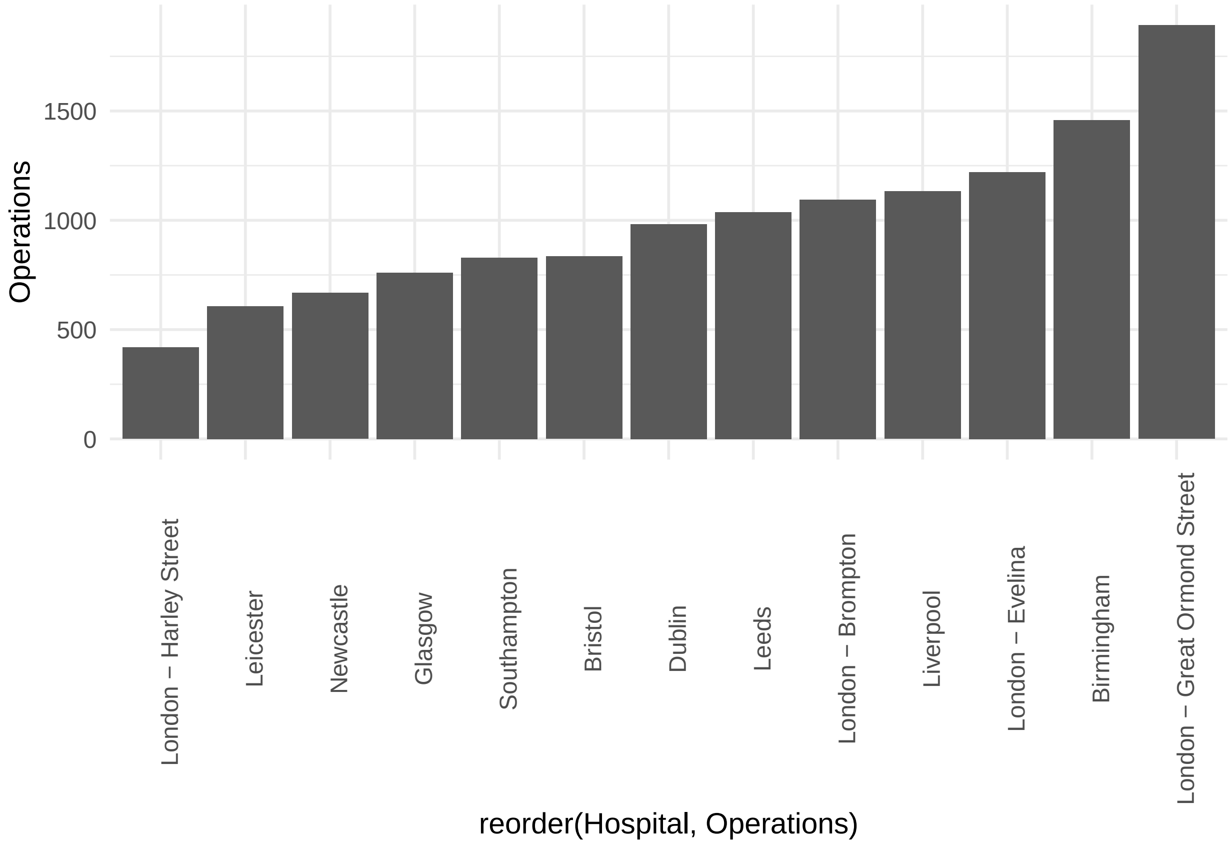


```r
#Note:
```

step 5 (1 point): In this step we are going to rotate the axis text on the x axis so we can see the names of each hospital more easily. Copy your code from step 4 into the box below and add a new line that changes the axis text of the x axis. Start here: https://datavizpyr.com/rotate-x-axis-text-labels-in-ggplot2/

Check your fig 5 with the answer key and if they look identical, move on to step 6.

```
fig5<- ggplot(data = HospitalData , aes(x= reorder(Hospital,Operations),y =Operations))+
  geom_bar(stat="identity") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle=90))
fig5
```

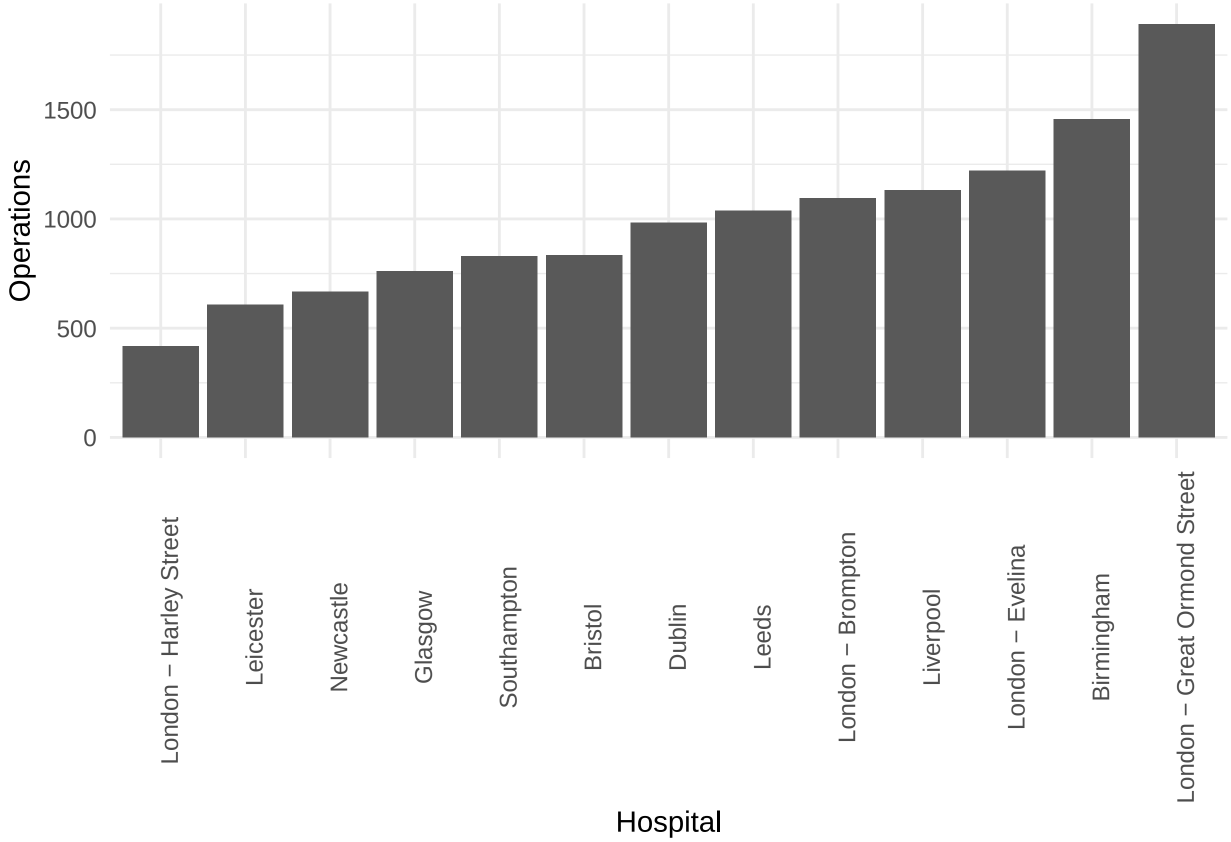


```
#Note:
```

step 6 (1 point): It is really coming together! We are in the home stretch now! Since we reordered the x axis, the x axis label no longer looks good. Lets rename the X axis back to simply: "Hospital". Copy the code you wrote in step 5 into the box below and add a new line that changes the x label. If lost, start here: http://www.sthda.com/english/wiki/ggplot2-title-main-axis-and-legend-titles

Check your fig 6 with the answer key and if they look identical, move on to step 7.

```
fig6<- ggplot(data = HospitalData , aes(x= reorder(Hospital,Operations),y =Operations))+
  geom_bar(stat="identity") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle=90))+
  xlab("Hospital")
fig6
```

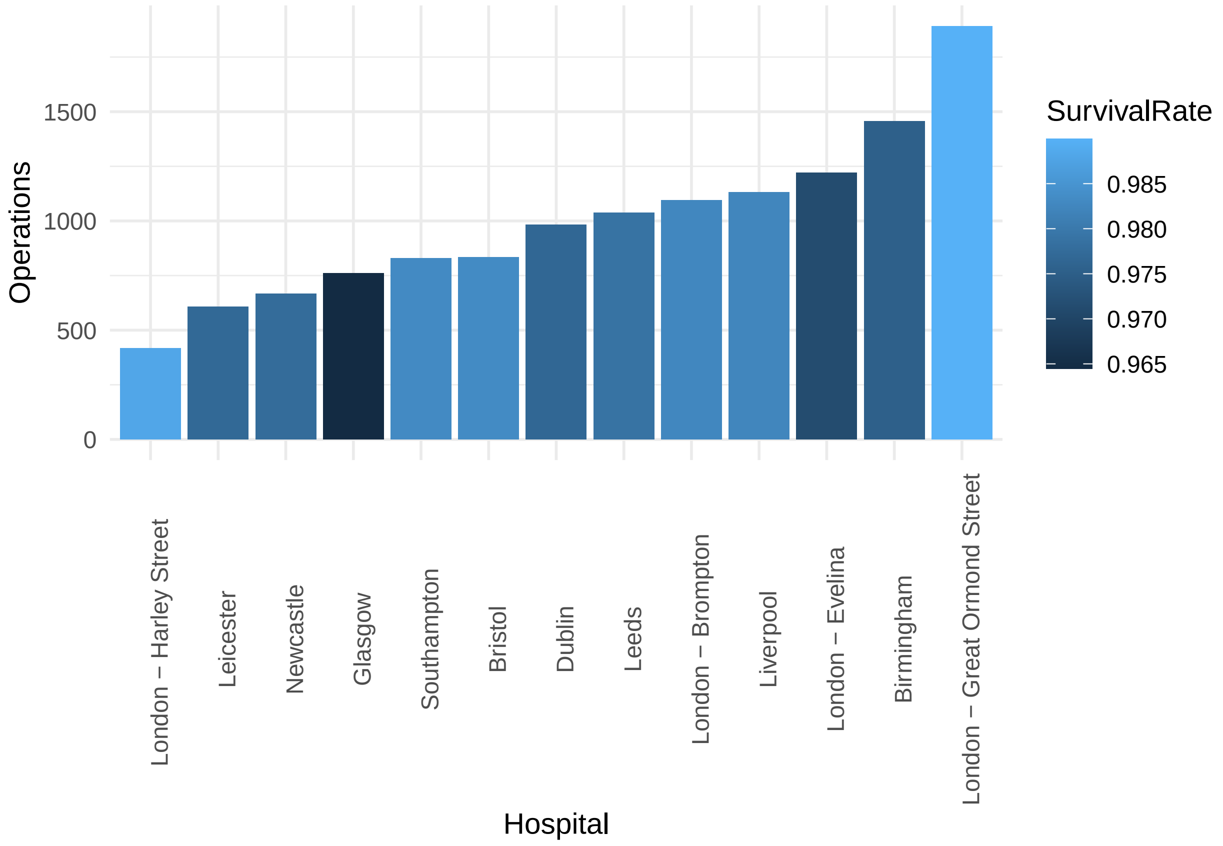


```
#Note:
```

Step 7 (2 points): Now the big step. Copy your code from step 6 into the box below and once again add to the aes() function to add a gradient fill color to the bars associated with the survival rate you calculated in last week's assignment. Check out how the "fill =" argument is used with a third data column here: https://r-graphics.org/recipe-bar-graph-colors

Check your fig 7 with the answer key and if they look identical, move on to step 8.

```
fig7<- ggplot(data = HospitalData , aes(x= reorder(Hospital,Operations),y =Operations,fill=SurvivalRate
  geom_bar(stat="identity") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle=90))+
  xlab("Hospital")
fig7
```



```
#Note:
```

step 8 (1 point): Now you have a pretty good looking plot! There is still plenty to learn and explore in ggplot but I think this is enough for one assignment! Now lets save this figure we made as an image file (a .tiff) using the ggsave function. Hint: the only three arguments in the ggsave function you need to assign today are the “filename =”, “plot =”, and “device =” arguments. You can leave the others as defaults.

```
?ggsave
```

```
## starting httpd help server ... done
```

```
ggsave(filename = 'Week3figure.tiff',plot = fig7, device = "tiff")
```

```
## Saving 6.5 x 4.5 in image
```

Once you are done, click the “Knit” button above(It looks like a blue ball of yarn). Save the file with your name and the week number in the file name:

(for example: “Liam_Mueller_Section_3”).

Then upload the pdf file to canvas/gradescope under the Week 1 Section Assignment before the deadline.

And that is it for this week!