Digi Data Visualization: Lecture 3

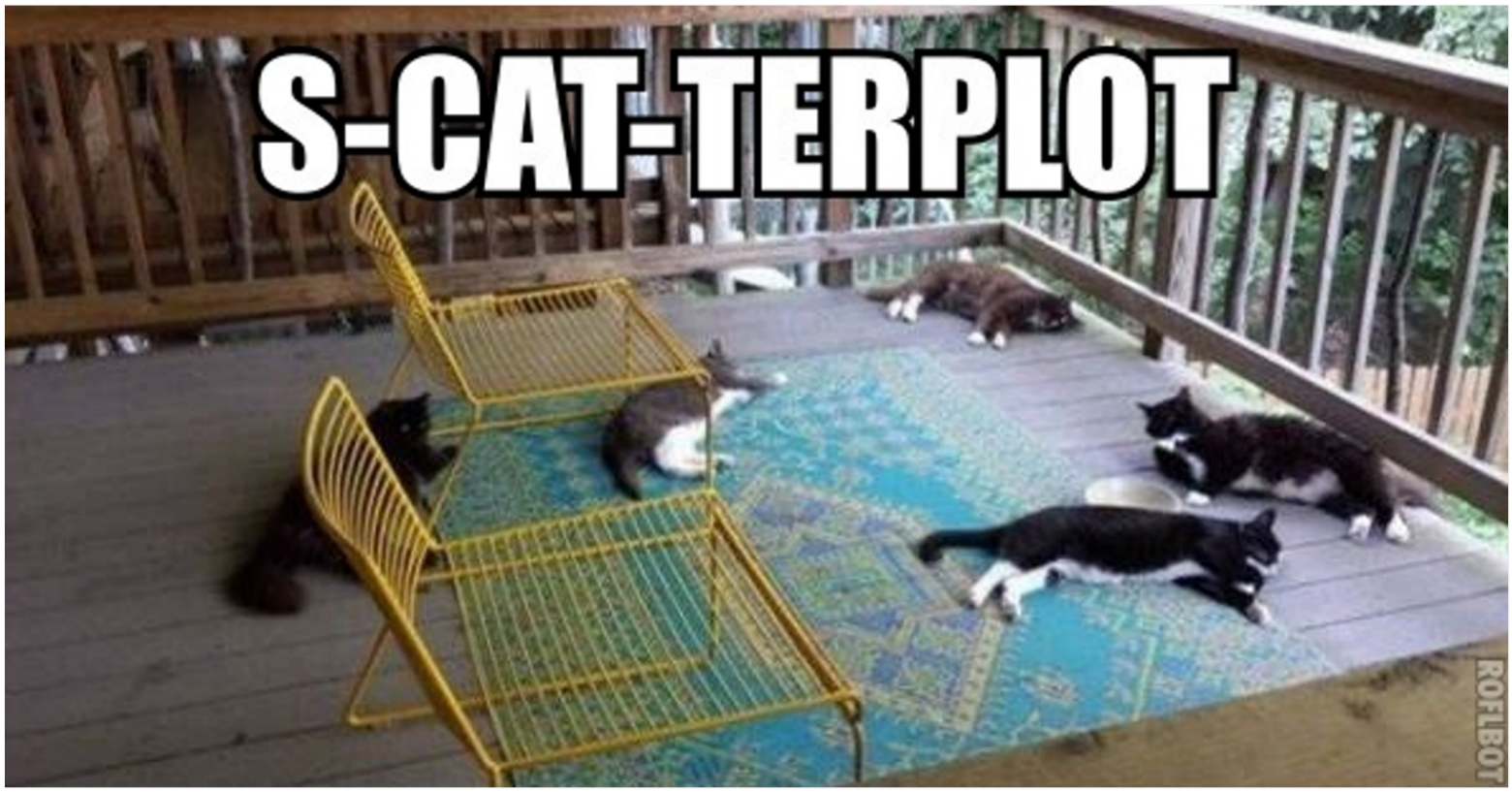
Katie Ireland Kuiper

11/11/2020

Scatterplots in Excel, Tableau, and R

Before we get started, be sure that you have Excel, Tableau public, R, and R Studio downloaded and installed.

Scatterplots



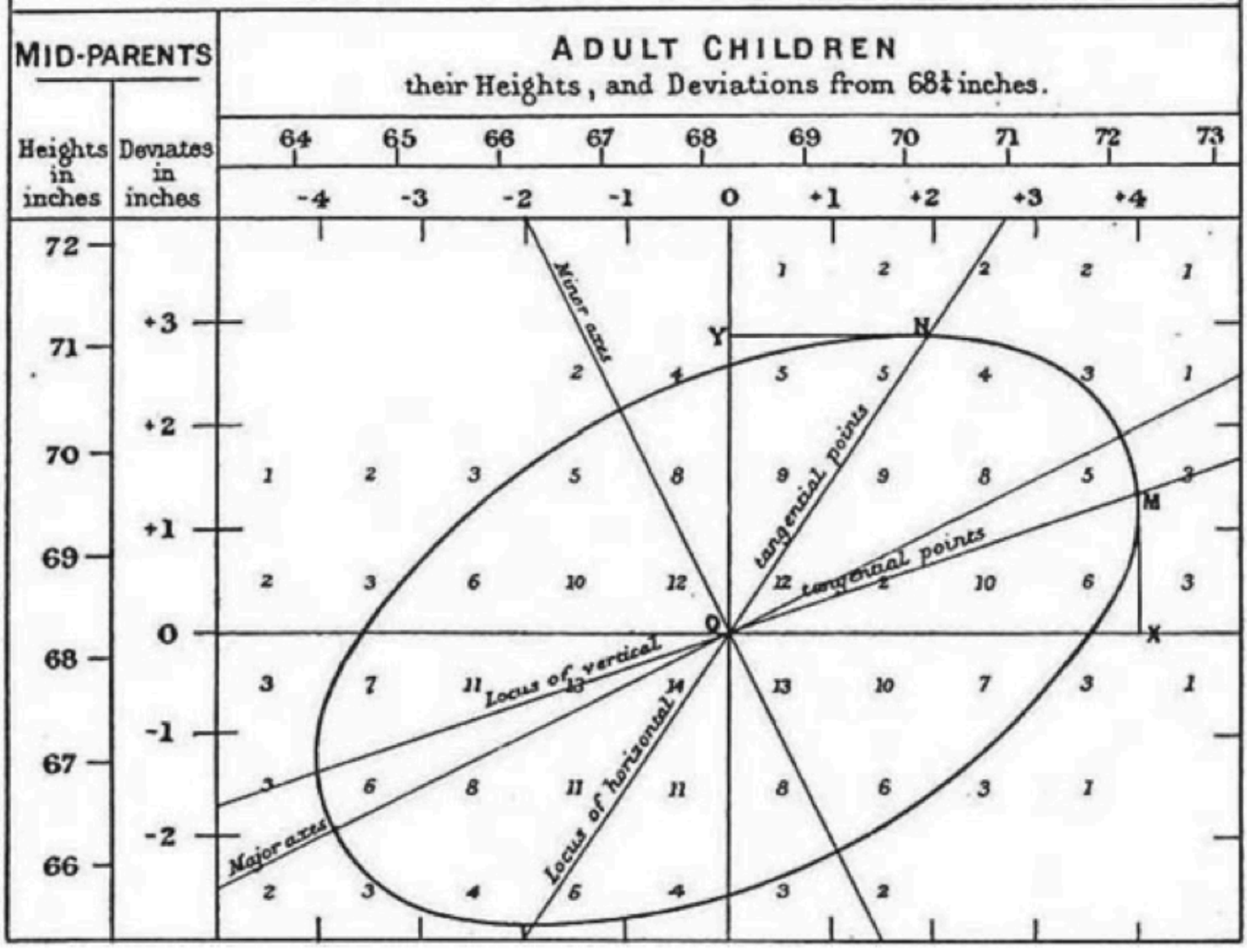
Today we will learn how to make scatterplots!

Who invented scatterplots?

The origin of the scatterplot is not entirely clear (Kopf 2018). They were historically called scatter diagrams and were often used to plot points on the cartesian coordinate system developed by Rene Descartes in the 17th century, with perpendicular lines visualizing the x and y axes (Kopf 2018). Although historically, they were most often used with geometry and algebra, today they are a widely popular visualization in a variety of disciplines and businesses and illustrate many interesting data relationships!

"Of all the graphic forms used today, the scatterplot is arguably the most versatile, polymorphic, and generally useful invention in the history of statistical graphics. Its use by Galton led to the discovery of correlation and regression, and ultimately to much of present multivariate statistics" (Friendly and Denis 2005).
Scatterplots dominate scientific journals; Tufte estimated that more than 70% of all charts in scientific publications are scatterplots. Friendly and Denis suggest that the original scatterplot was made in 1833 by English scientist John Frederick W. Herschel; he created scatterplots to display the parameters of the orbits of double stars (2005).

Francis Galton, one of the creators of the statistical concept of correlation, also played a role in recognizing the scatterplot in the late 1800s. He used correlation analysis to understand the relationship between the height of parents and their children.



Here is one of the visualizations he created: a smoothed correlation diagram of the heights of parents and children.

Other names for Scatterplots

Scatterplots have also been called scatter plots, scattergrams, scatter diagrams, scatter graphs, and scatter charts (Friendly & Dennis 2005; Yi 2019). In the early 1900s, the term scatter diagram was utilized to describe this particular kind of data visualization (Friendly & Denis 2005).

Common Uses for Scatterplots

These graphs are primarily used for observing data and displaying relationships between at least two (generally numeric) variables. The points or dots in the plot show the values of individual data points and patterns overall in the dataset when taken as an entire unit. For each datapoint, the value of its first variable is displayed on the X axis, and the second is represented on the Y axis. Scatterplots usually display values for two variables, but if the points are also matched to a color, shape, or size, an additional variable can be represented in the visualization.

Scatterplots are often associated with statistical measurements including correlation. Correlation is a specific type of data relationship, with different options. An example of correlation could be the relationship between square feet of a building and the use of electricity.

In terms of visualizing correlation, there are only three options: positive, negative, or no correlation. If there is a positive correlation, then as one variable increases, the other does as well. If we take our example of the square footage of a building and the use of electricity, as one increases so does the other. A negative correlation involves one variable decreasing and the other increasing; an example of this could be ice cream sales and declining temperatures.

Keep in mind…

It is also vitally important to remember, if you are interested in correlation in your data, correlation does not equal causation!! It is quite likely that there are other variables in the data that are responsible for the relationship, or possible that the pattern is coincidence. If your dataset has a great many number of points to plot, this can potentially create the issue of overplotting. Overplotting happens when data points overlap so much that it is hard to tell what the relationship is between the points and the variables. If the plot is extremely dense, then this might be occurring (Yi 2019). There are different options for overcoming this issue, including carefully subsetting the data, utilizing transparency for the points (which we will learn how to do in our R scatterplots today), or changing the size of the points.

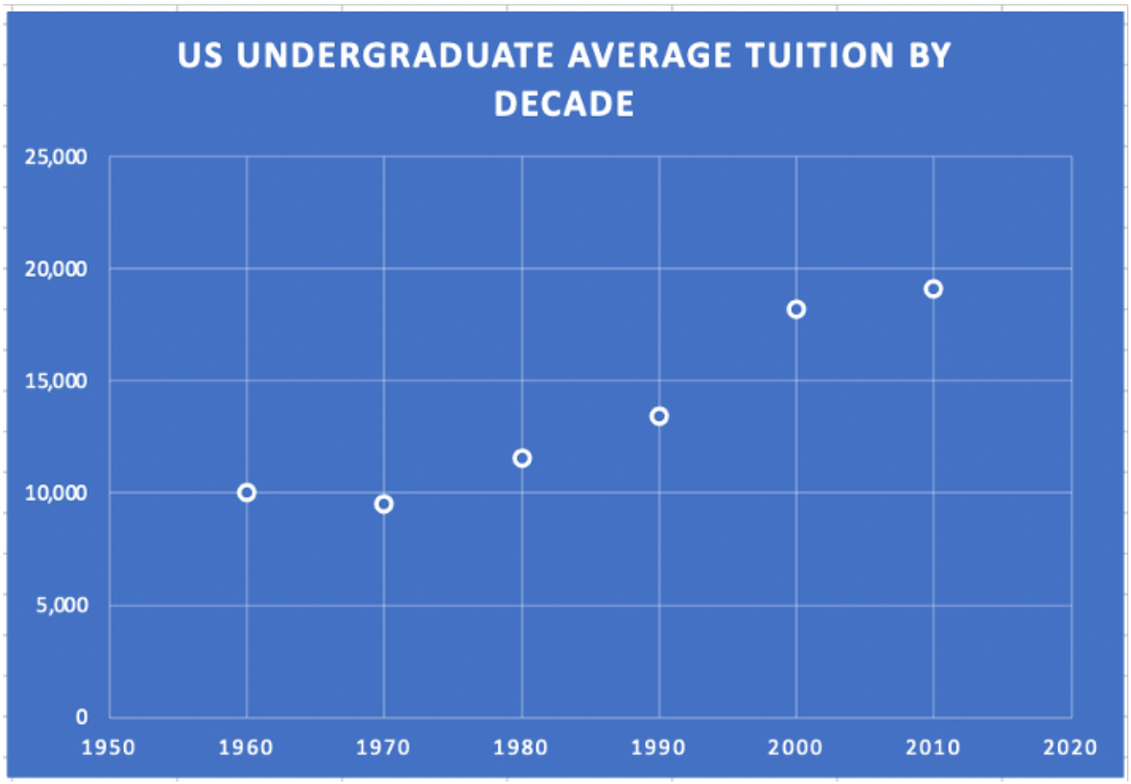
Excel

Excel is great for making quick and easy scatterplots! Let's make one in excel now!

Go ahead and open your dataset in excel. We are using this time series dataset of US undergraduate average tuition by decade, a subset of data courtesy of the National Center for Education Statistics. If you would like their full dataset, check it out or use the one shared in our zoom course.

Manually:

1. Highlight the two data columns: Decade and Average Tuition
2. Navigate to the main menu go to Insert -> Chart -> X Y (Scatter)



Here is an example image of the output in excel!

Hint: To view other types of charts with this dataset, click on Home (far left corner), then Ideas (far right corner). This will pull up other charts on the right side of your excel page and a pivot table for you to explore!

To adjust colors, axis labels, etc:

Go to Menu -> Chart Design and find Adjust Colors! If you would like different colors by the different points in time, highlight the data points in the chart. Navigate to Format Data Series (this should open a new menu on the right). In the menu, select Fill and then click Vary Colors by Point.

You can click into the chart names and axis names to change or delete them!

Note: I would recommend saving this under a different name from the original dataset, just for the sake of the other programs we will be using to make scatterplots during this tutorial!

Tableau Public

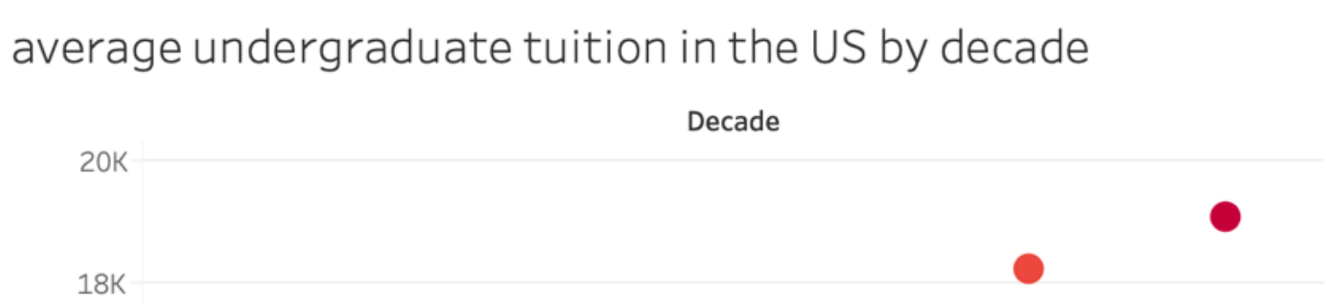
Open your original csv file.

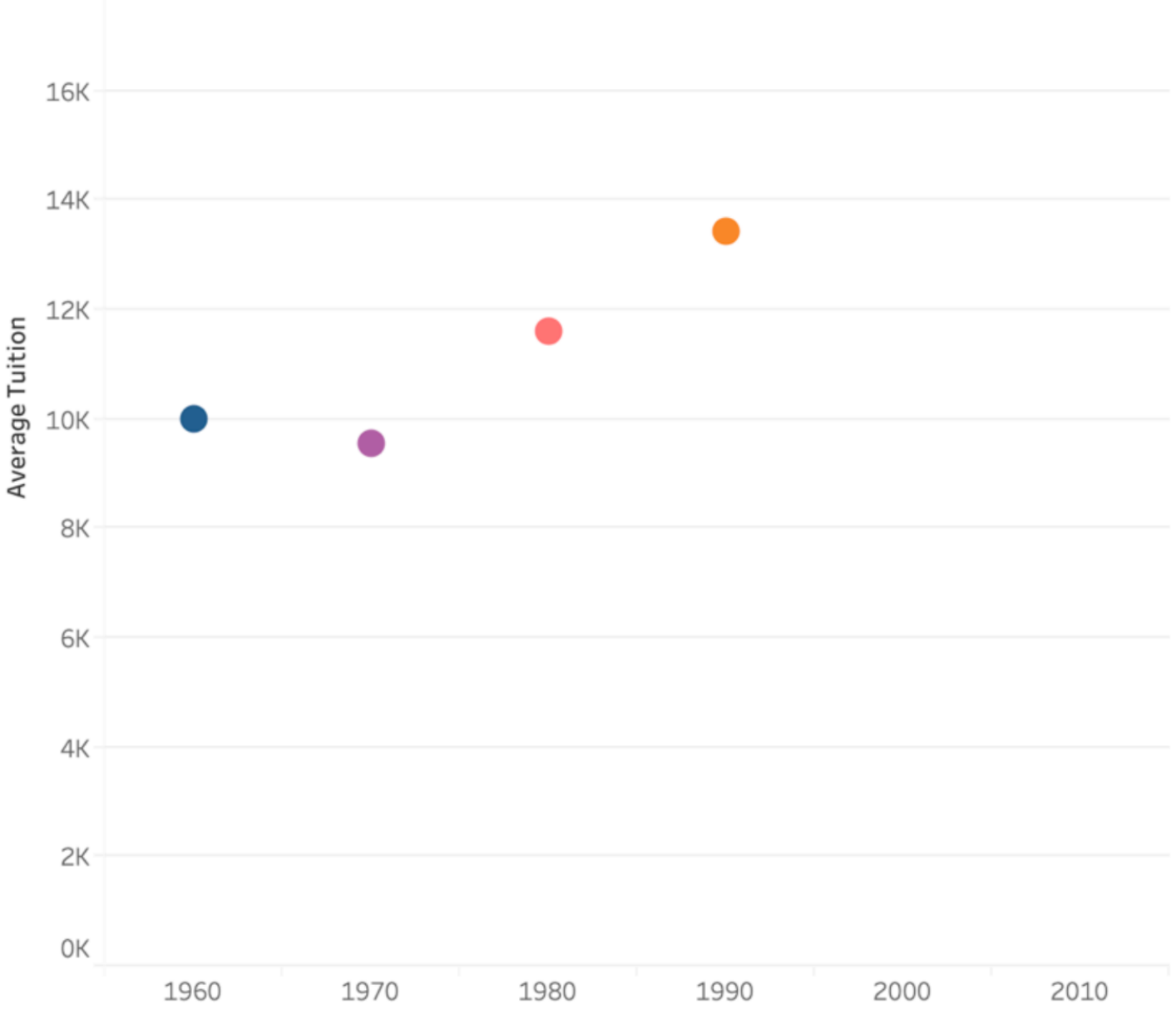
Navigate to Sheet 1 (bottom left corner).

Note: remember you have to convert the shape with the time series dataset bc tableau wants there to be a measure for it to be a 'scatterplot'

1. Click on Decade under the Data menu (left side) and duplicate it. Use the duplicated decade (called Decade copy) and select Change Data Type -> Date. Then move it to Columns.
2. Move Average Tuition to Rows.
3. Under Marks, change the Automatic Label field to Shape.
4. Select Lines continuous under the Show Me tab, then go to Marks. Change the top option from Automatic to Circle, and feel free to change the shape in the Marks menu.
5. You will then have a Scatterplot in Tableau. To adjust the length of the axes, move your cursor to the edge until it becomes a double-sided arrow. Then you can drag the arrow to adjust.
6. To change the names of the axes labels or the main plot name, click into it and adjust. Other settings can be changed under the Marks menu to the left of the chart.

Note: This is not labeled as a traditional Scatterplot in Tableau even though it is one, most likely because the dataset is a Time Series set. We will make a traditional Scatterplot in Tableau with our second dataset today.





Here is an example image of the output in tableau public!

7. You can also change the color palette by clicking into color. (Note: here I also dragged Decade onto Color in Marks so that each point in time would have a different color.) The color palette I used is called Sunrise-Sunset Diverging, but feel free to use others or create your own!

To change the colors, try dragging the data onto the colors under the Marks menu! Test it out, and see what happens. To adjust the color scheme, click into the colors in the Marks menu.

To adjust the length of any axis, move your cursor to the edge of the axis until it becomes an arrow, click and drag the cursor to change it.

To edit the names, click into the axis and type in your preferred title.

Additional chart settings can be further adjusted from the Marks Menu– Try selecting different marks and see what happens! Color, Size, Label, Detail, and Tooltip can all be adjusted there! If you want to publish your graph on tableau public you can save it there. If you'd rather just screen shot the image, then it won't be automatically saved to your tableau public account.

R

We are going to use car and ggplot2 to make our scatterplots in R. Open your R studio (integrated development environment) and run the scripts below.

```
avgtuition <- read.csv("/Users/kikuiper/Documents/data_dh/avgtuition.csv") #input your file path #here
head(avgtuition) #checks out the data #can fix in text edit (remove the quotes and commas)
```

	Decade <int>	Average.Tuition <int>
1	1960	10003
2	1970	9537
3	1980	11577
4	1990	13405
5	2000	18203
6	2010	19094

6 rows

```
#install.packages("tidyverse")
#install.packages("car")
library(car)
```

```
## Loading required package: carData
```

```
library(tidyverse) #make sure this package is installed using install.packages("tidyverse") before loading the
```

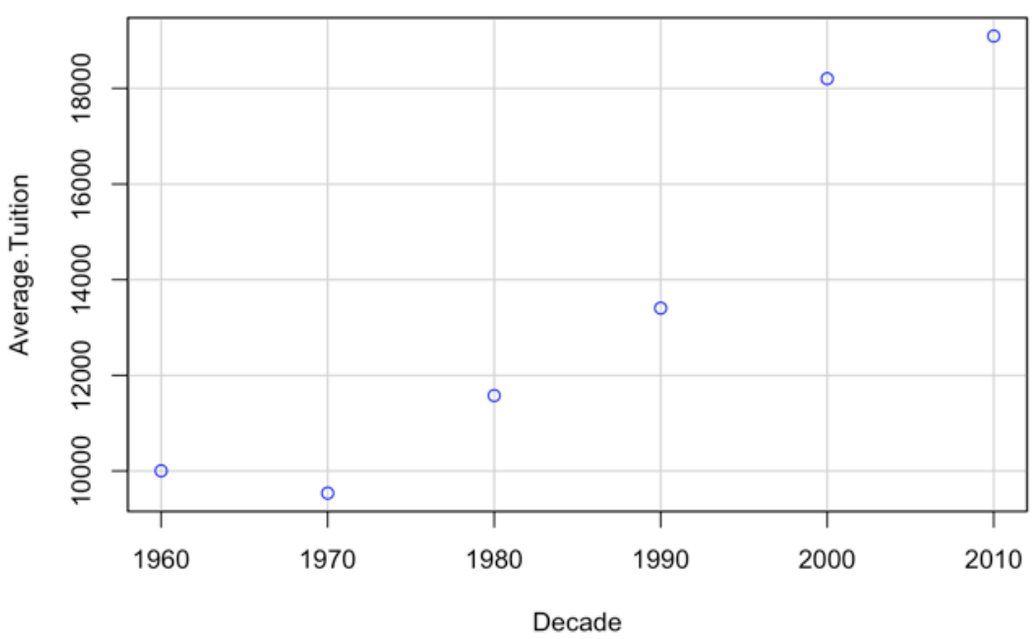
```
## ── Attaching packages ─────────────────────────────────────────────────────────────────────── tidyverse 1.3.0
──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.3     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
```

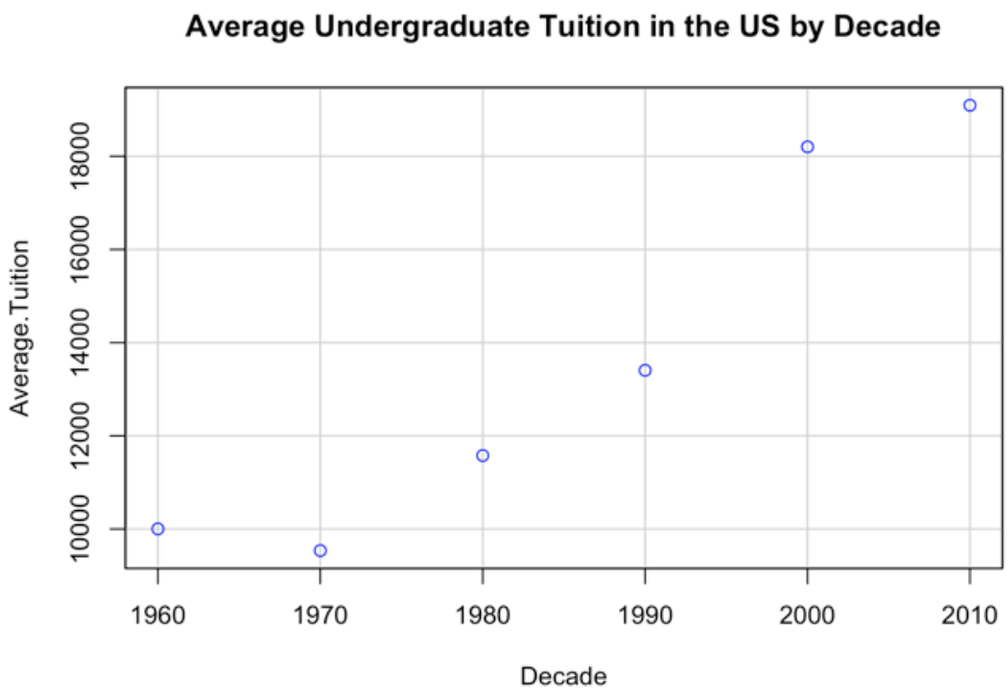
```
## ── Conflicts ────────────────────────────────────────────────────────────────────────── tidyverse_conflicts()
──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::recode() masks car::recode()
## x purrr::some()   masks car::some()
```

Now that the data is uploaded, run these commands to prepare it for making the charts.

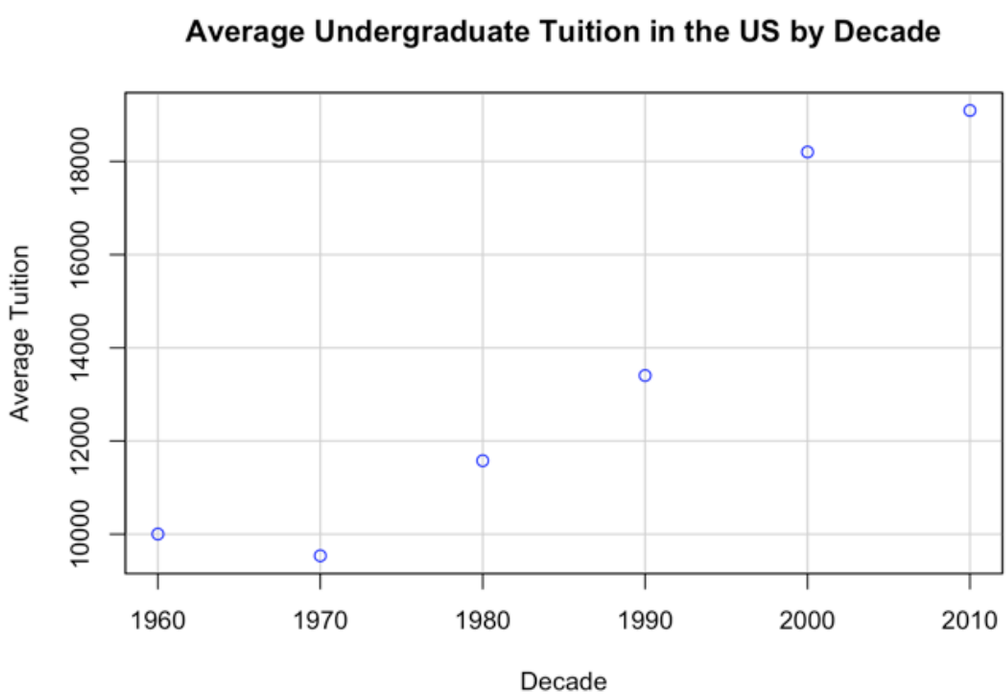
```
car::scatterplot(Average.Tuition ~ Decade, data=avgtuition,
   smooth = F, regLine= F, boxplot = F)
```



```
#Note: car has a lot of built in features that were not necessary for this plot. They are all turned off using th
e F command above. Try using T to turn them back on and see what happens!


#add in the main title
car::scatterplot(Average.Tuition ~ Decade, data=avgtuition,
   smooth = F, regLine= F, boxplot = F, main = "Average Undergraduate Tuition in the US by Decade")
```

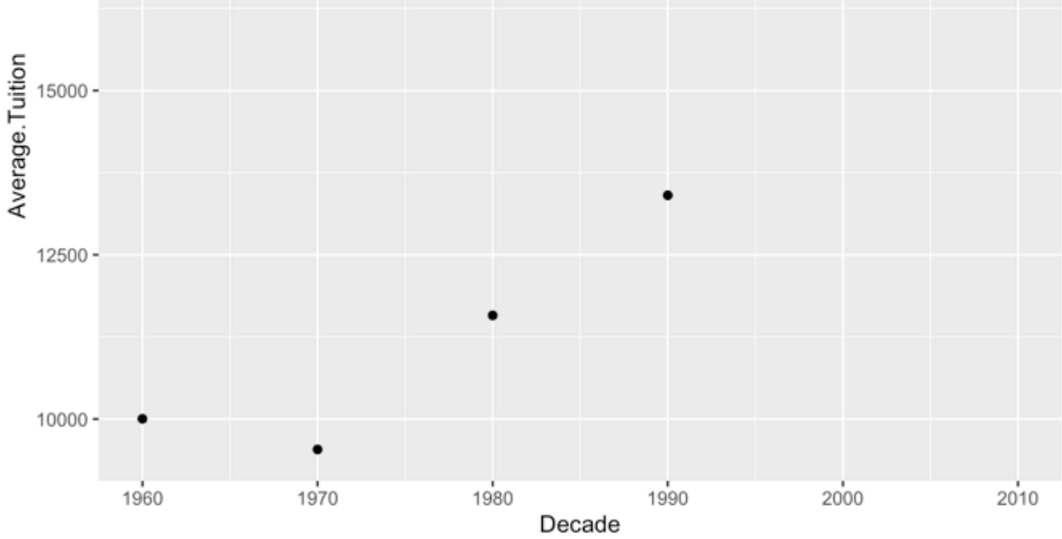



```
#add in x axis and y axis labels
car::scatterplot(Average.Tuition ~ Decade, data=avgtuition,
   smooth = F, regLine= F, boxplot = F, main = "Average Undergraduate Tuition in the US by Decade", xlab = "Decad
e", ylab = "Average Tuition")
```

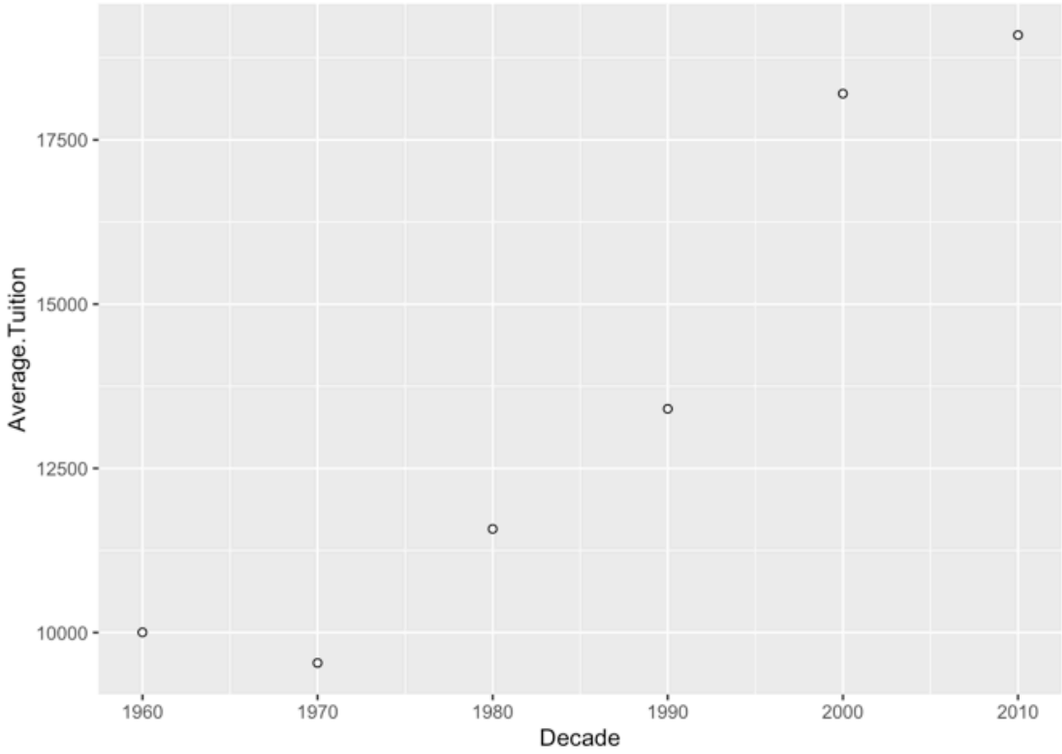


Let's plot it with ggplot2.

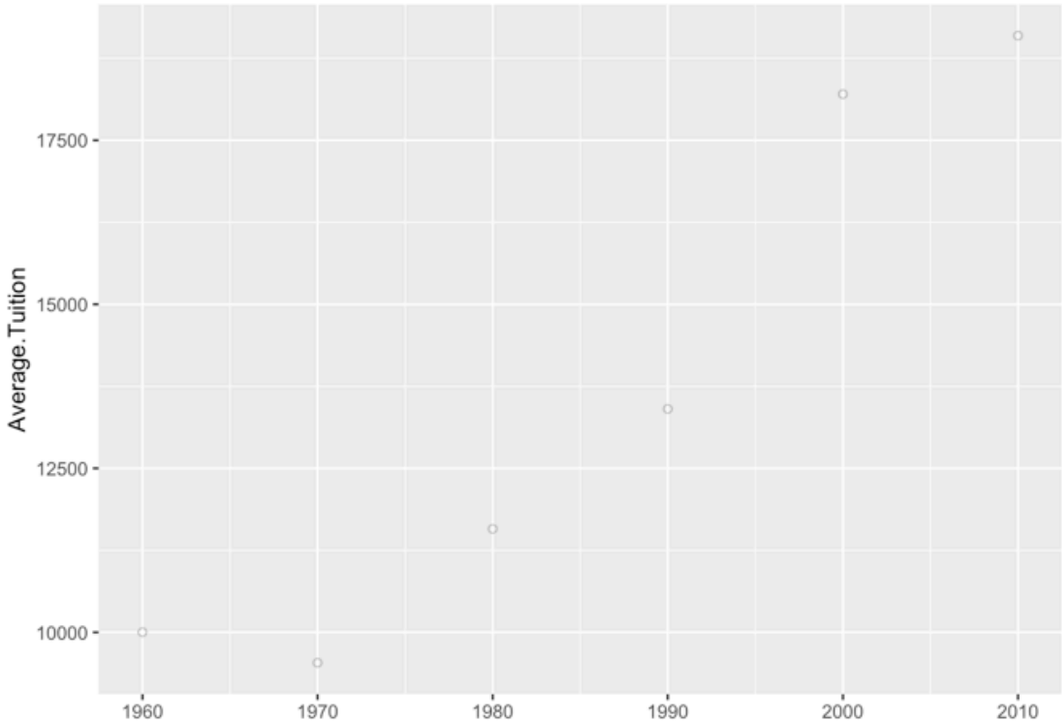
```
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point()
```

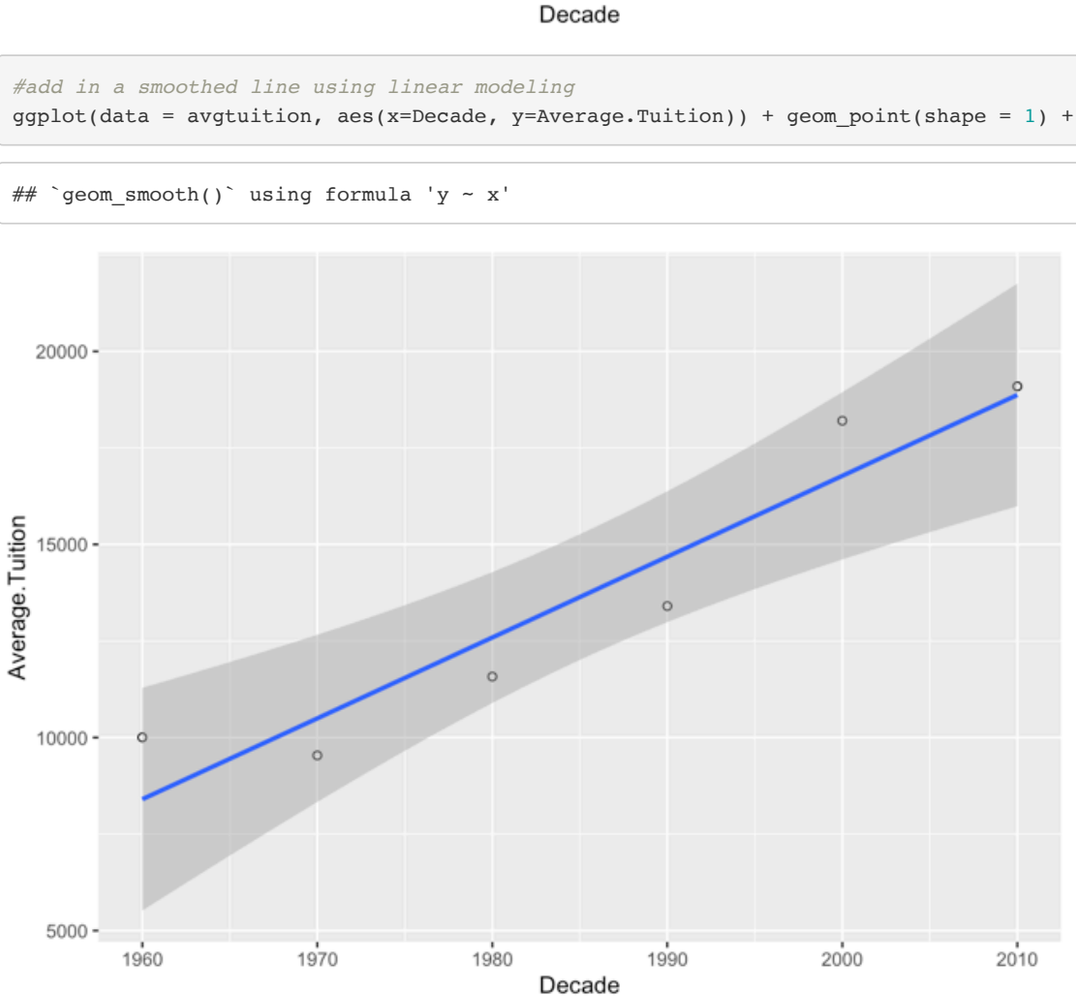



```
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(shape = 1)
```




```
#adjust opacity of the points
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(shape = 1, alpha=1/4)
```



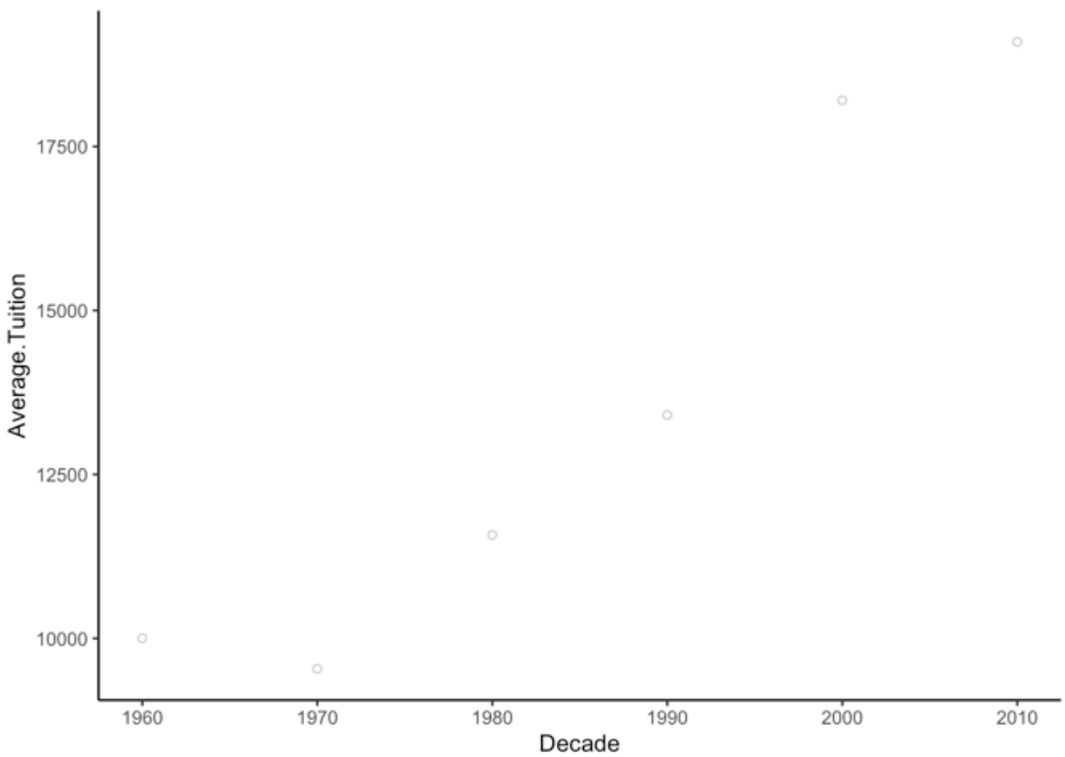



```
#add in a smoothed line using linear modeling
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(shape = 1) +  geom_smooth(method=lm)
```

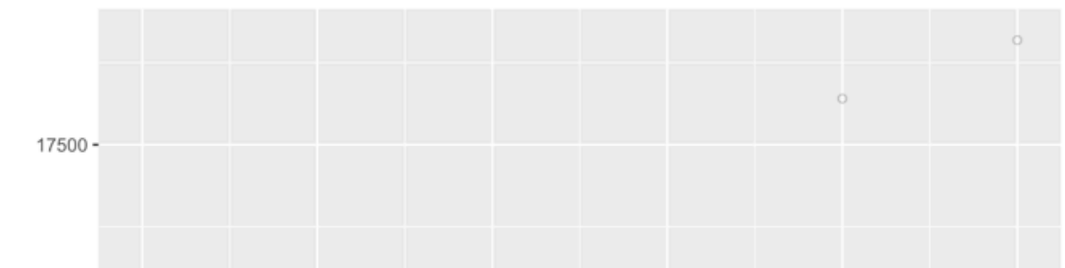
```
## `geom_smooth()` using formula 'y ~ x'
```

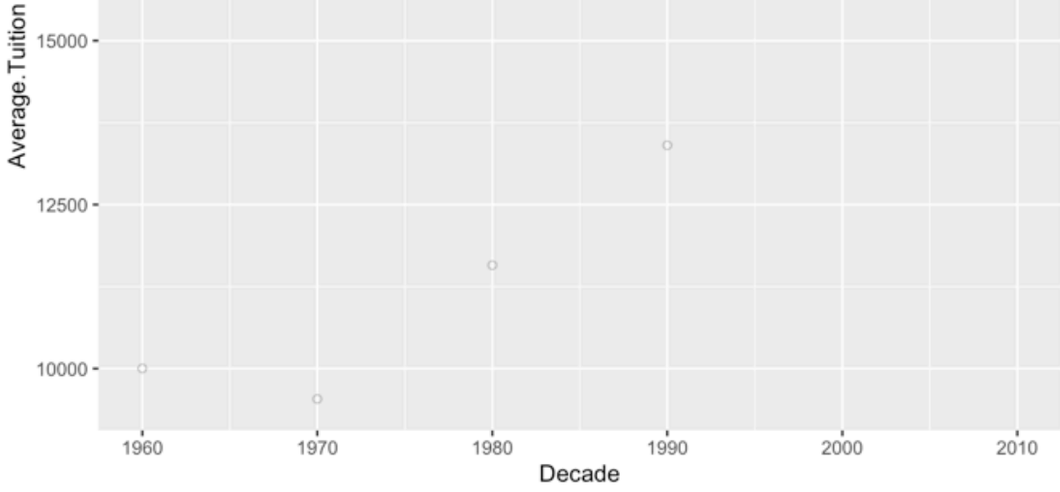


```
#add in a theme
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(shape = 1, alpha=1/4) + theme_classic()
```

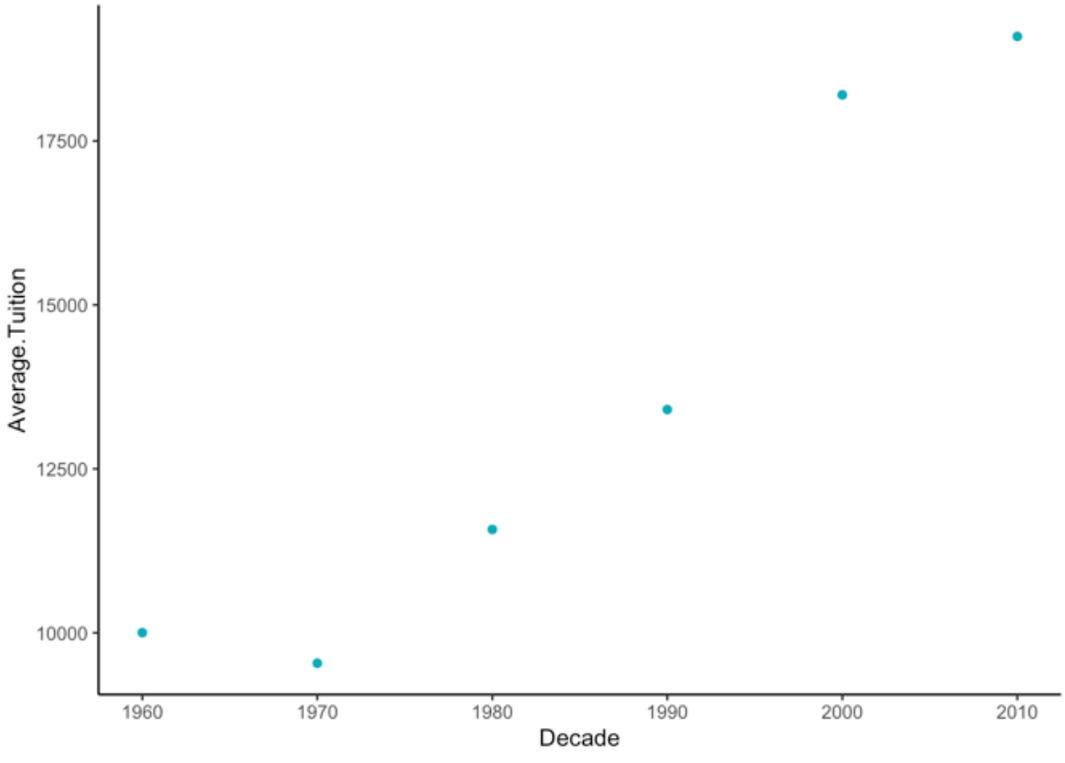



```
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(shape = 1, alpha=1/4) + theme_gray()
```

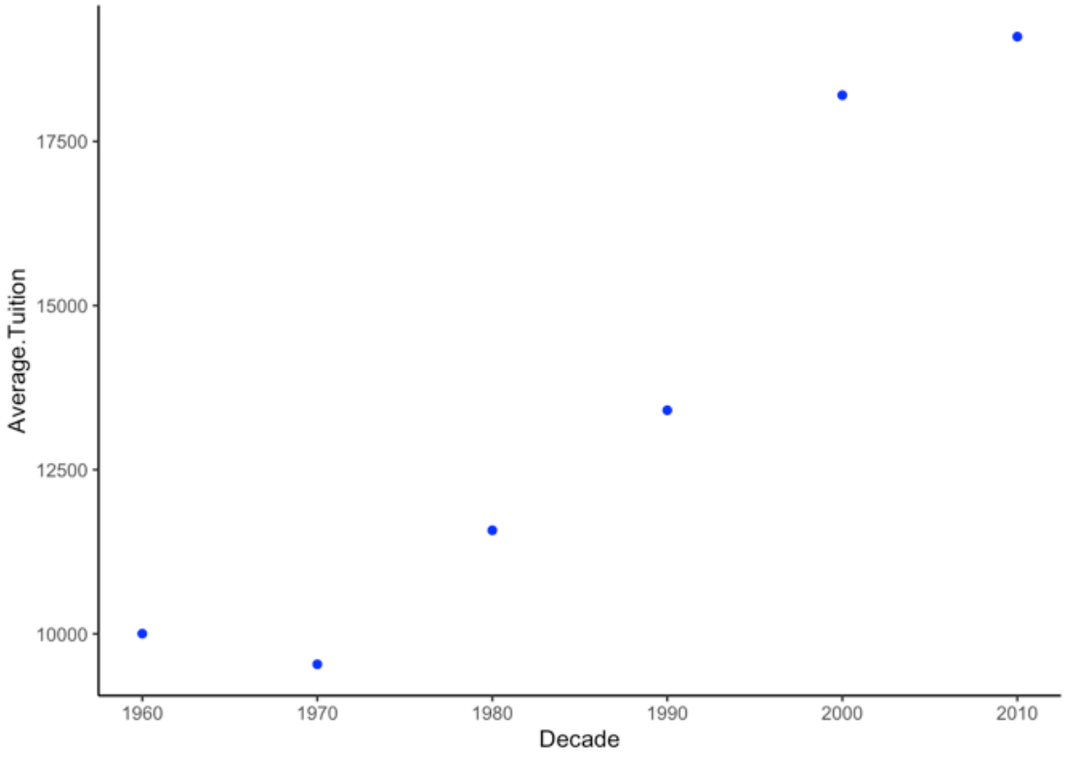




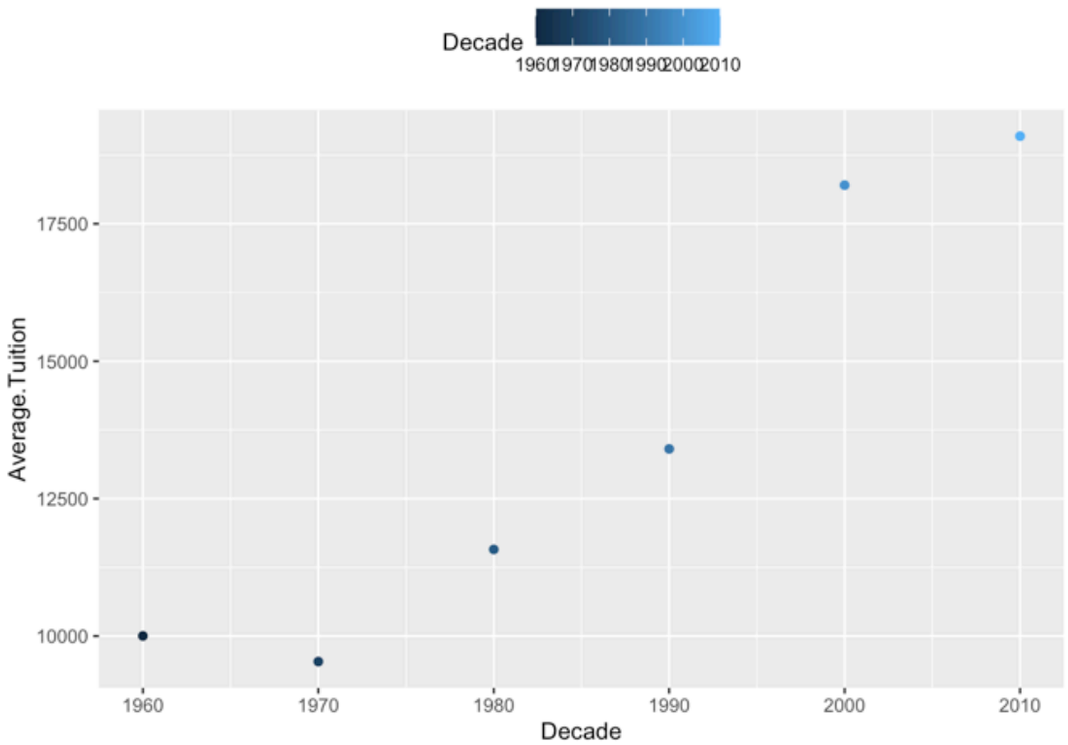

```
#add in single color for the points, can be done using a color name or Hex colors
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(color = "#00AFBB") + theme_classic()
```



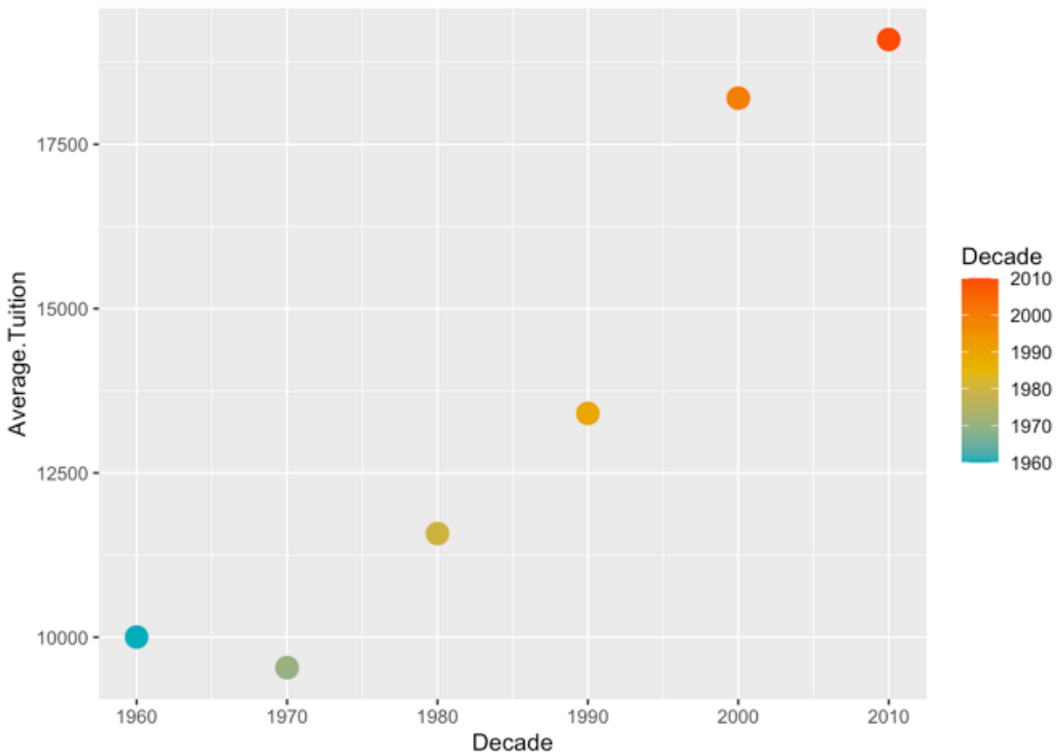

```
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(color = "blue") + theme_classic()
```



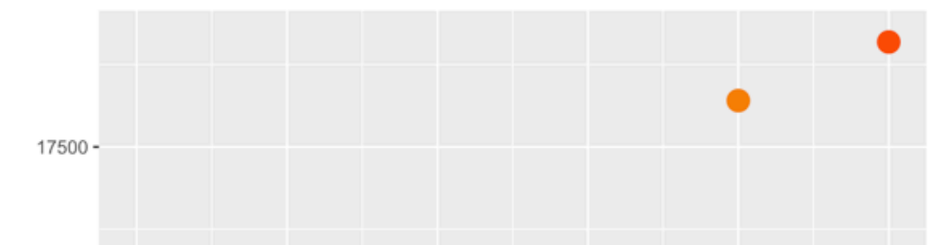
```
#this can also be applied to the different variable values
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(aes(color = Decade))+ theme(legend.posit
ion = "top")
```

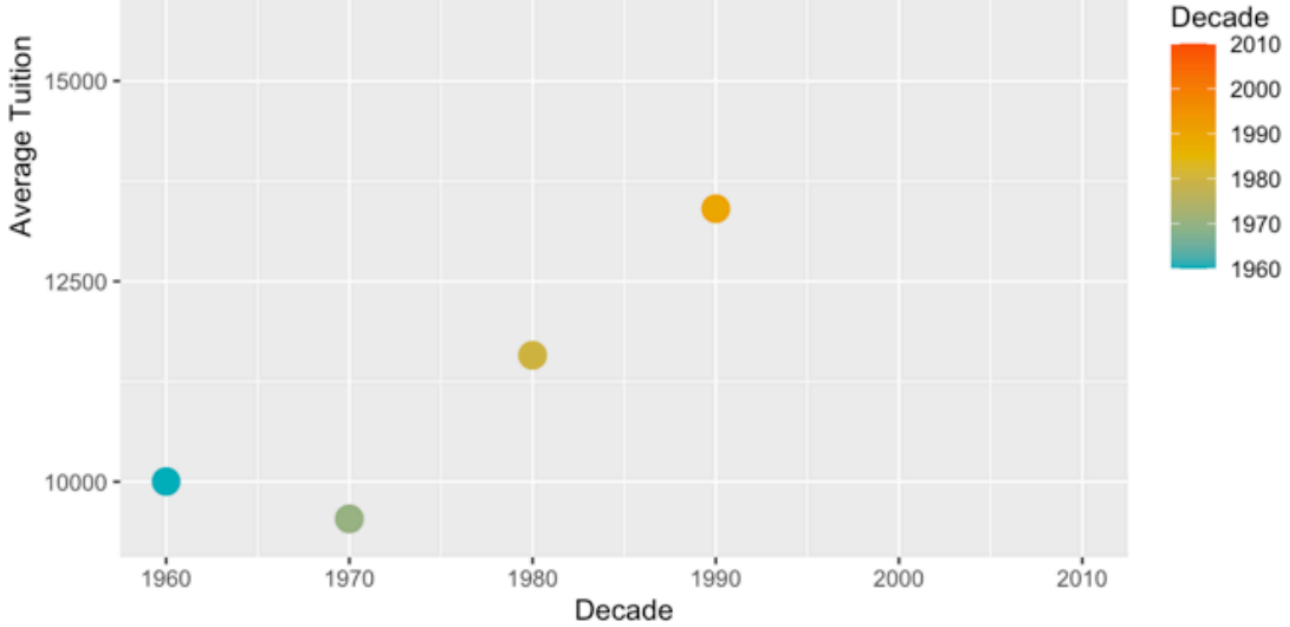



```
#adjust the colors!
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(aes(size = 5, color = Decade)) + scale_c
olor_gradient2(midpoint = 1985, low ="#00AFBB", mid="#E7B800", high="#FC4E07") + scale_size(guide="none")
```

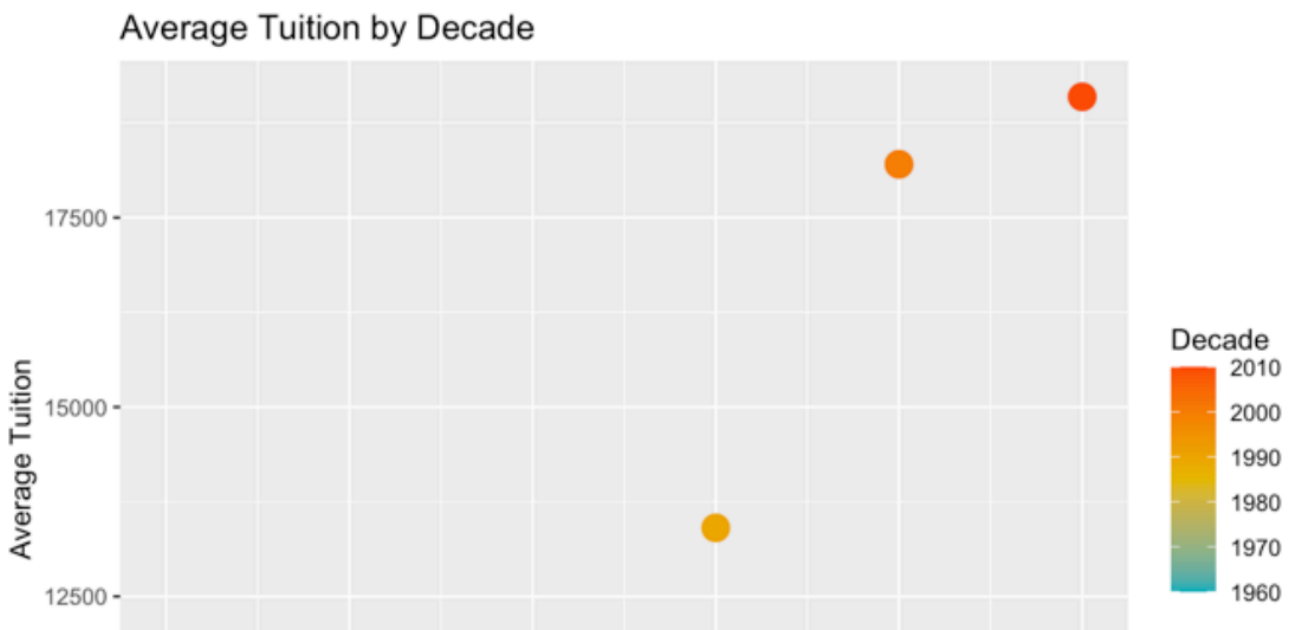


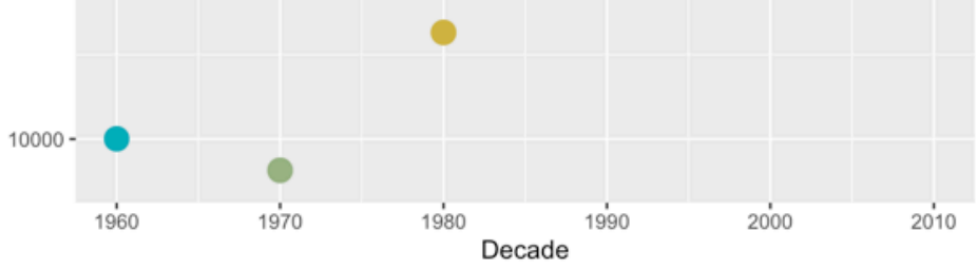

```
#add in x, y axis labels
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(aes(size = 5, color = Decade)) + scale_c
olor_gradient2(midpoint = 1985, low ="#00AFBB", mid="#E7B800", high="#FC4E07") + scale_size(guide="none")  + labs
(x= "Decade", y = "Average Tuition")
```






```
#add in main title
ggplot(data = avgtuition, aes(x=Decade, y=Average.Tuition)) + geom_point(aes(size = 5, color = Decade)) + scale_c
olor_gradient2(midpoint = 1985, low ="#00AFBB", mid="#E7B800", high="#FC4E07") + scale_size(guide="none")  + labs
(x= "Decade", y = "Average Tuition", title = "Average Tuition by Decade")
```





Now, we will use a dataset with numerical variables only, rather than time-series data. This is an abbreviated dataset from kaggle.com:Apartment sales by size of the living area.

Excel

1. Open your csv file: aptsales.csv
2. Highlight the columns GrLivArea and SalePrice.
3. Go to Insert -> Chart -> X Y (Scatter)

Here is an example image of the output in excel!

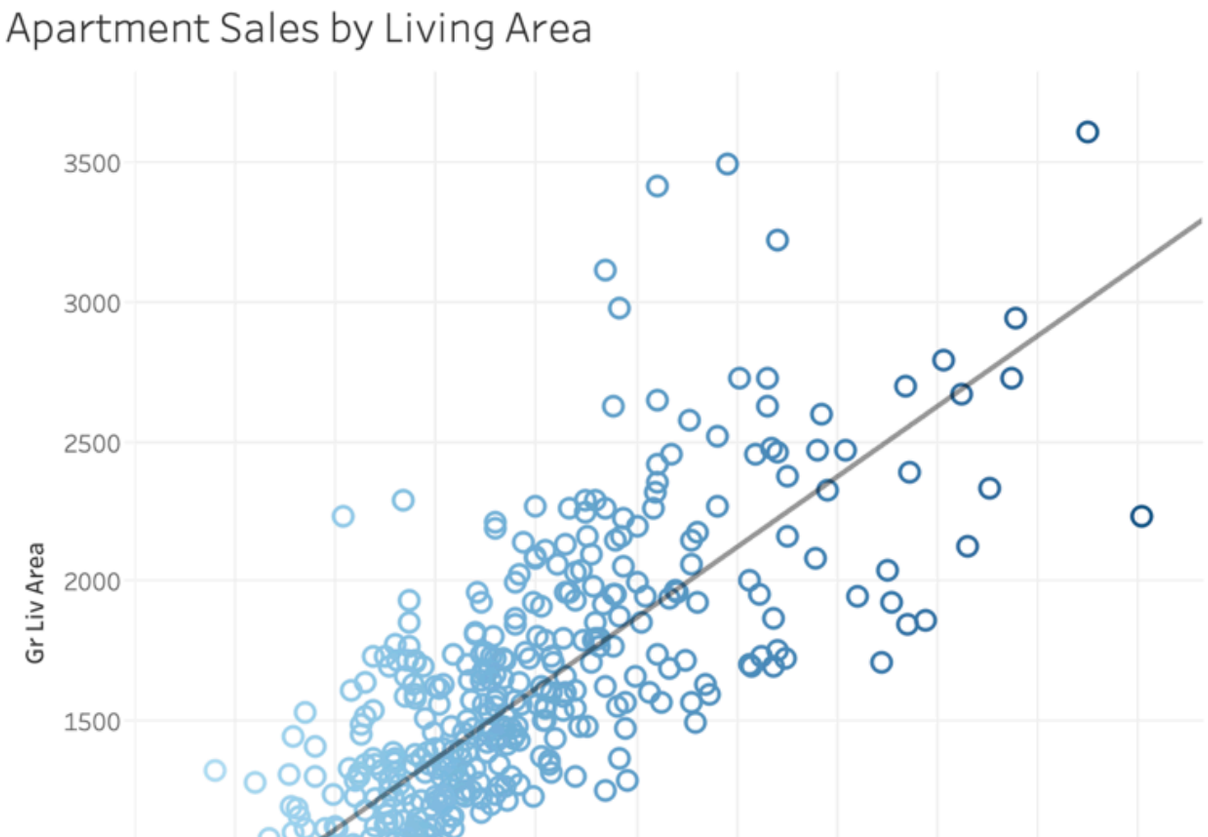
4. Alternatively, click on the Home menu button (far left), then navigate to the Ideas symbol on the far right.
5. Click on ideas, and a pop-up menu on the right will be displayed beside the spreadsheet. This will include a list of possible visualizations with this dataset, including a pivot table cluster chart. This is still a scatterplot.
6. If you select it, the chart will be displayed on your spreadsheet and you will be able to edit the title, axes, and colors from the other menu options!
7. Note: If you wish to add a trend line to this chart, navigate to the menu Chart Design.
8. Select Add Chart Element on the far left. Now choose Trendline -> Linear (or Exponential). There are different types of trendlines in excel based on different models. Select the best one for your data or play with them now!

Here is an example image of the output in Excel, with the exponential trendline!

Tableau

1. Open the dataset and navigate to sheet 1.
2. Duplicate the variables Gr Living Area and Sale Price.
3. Convert each duplicated variable to a dimension by clicking into the variable and selecting Convert to Dimension.
4. Move measure variables: SUM(Sale Price) to Columns and SUM(Gr Liv Area) to Rows.
5. Move dimension variables to Marks.
6. Double click Scatterplot under Show Me Menu on the left.
7. To change the colors and make them vary, move SUM(Sale Price) from the Tables menu on the left onto Color in the Marks menu.

Note: If you wish to add a “trend line”, go to the top menu Analysis. Then select Trend Lines -> Show Trend Lines. If you click into this, you can also find out details of the trend line output in a pop-up window. Here is the beginning of the results: Trend Lines Model A linear trend model is computed for sum of Gr Liv Area given sum of Sale Price. The model may be significant at p <= 0.05.



Here is an example image of the output in Tableau, with the trend line!

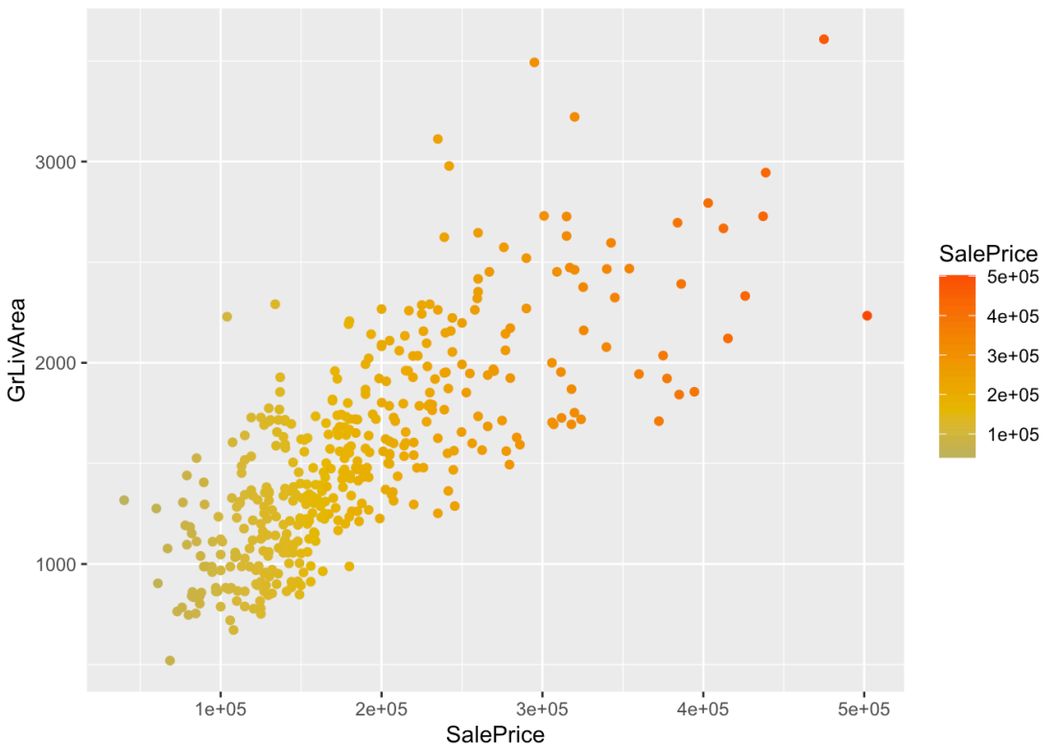
R

```
#upload and create the data object in R
aptsales <- read.csv("/Users/kikuiper/Documents/data_dh/aptsales.csv") #input your file path #here
head(aptsales)
```

	GrLivArea <int>	SalePrice <int>
1	1710	208500
2	1262	181500
3	1786	223500
4	1717	140000
5	2198	250000
6	1362	143000

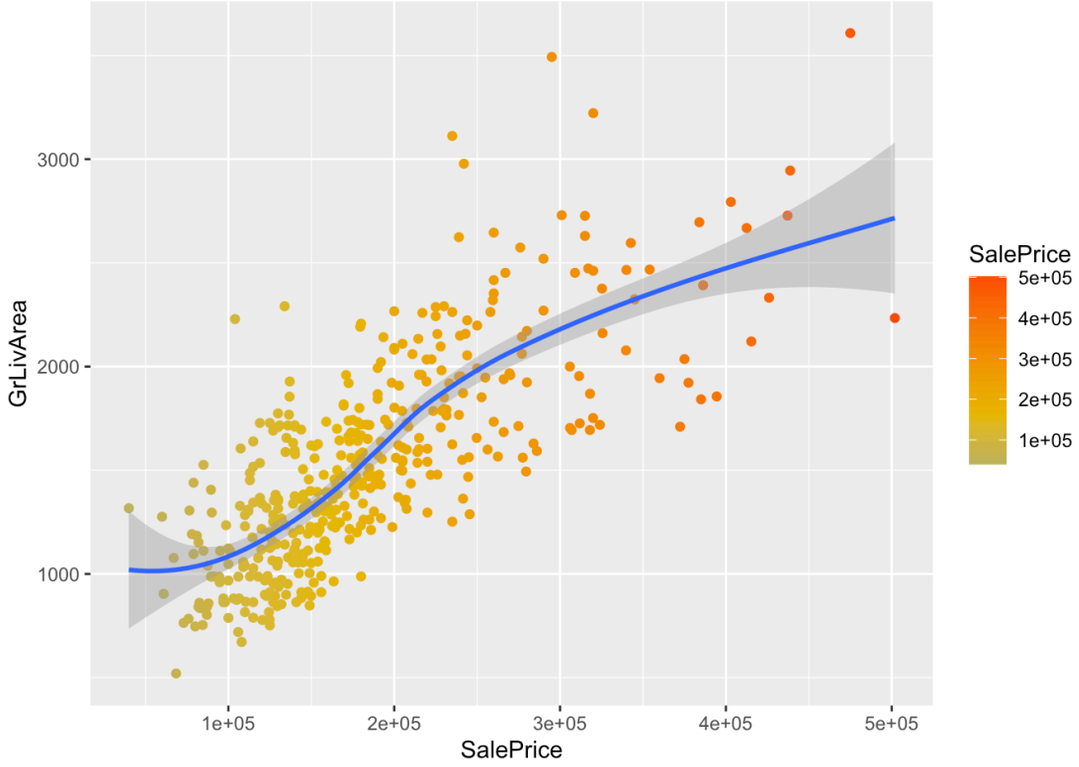
6 rows

```
ggplot(data = aptsales, aes(x=SalePrice, y=GrLivArea)) + geom_point(aes(color = SalePrice)) + scale_color_gradien
t2(midpoint = 160000, low ="#00AFBB", mid="#E7B800", high="#FC4E07")
```



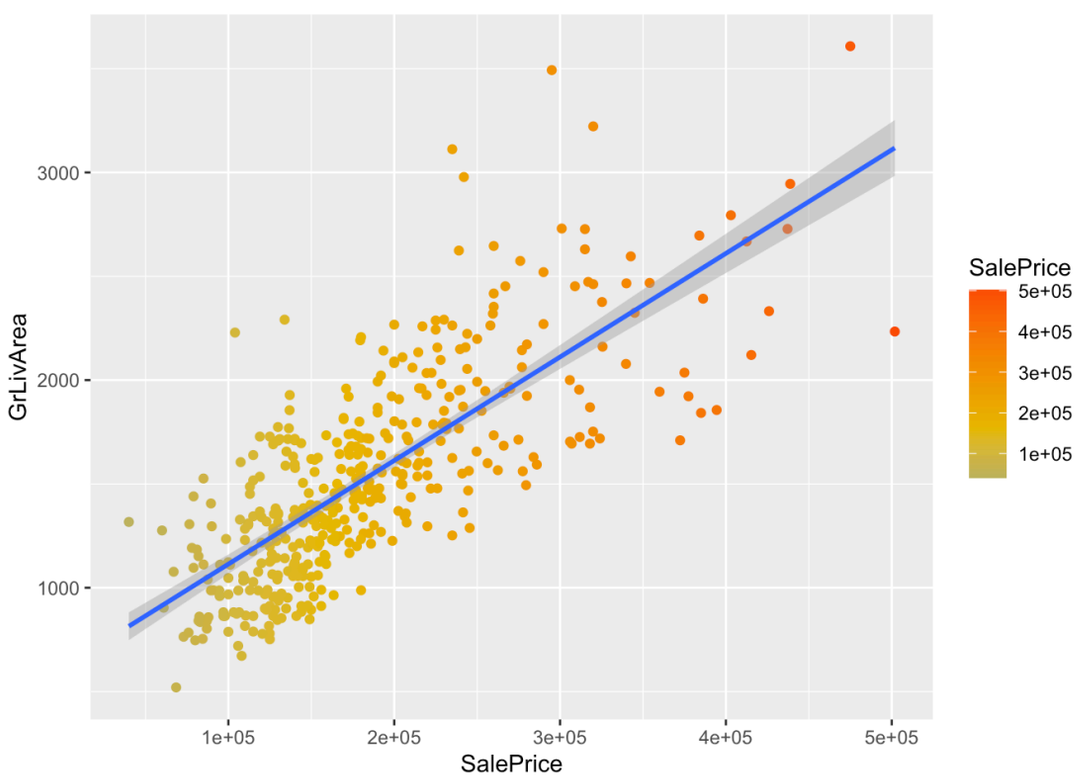

```
#add a line

ggplot(data = aptsales, aes(x=SalePrice, y=GrLivArea)) + geom_point(aes(color = SalePrice)) + scale_color_gradien
t2(midpoint = 160000, low ="#00AFBB", mid="#E7B800", high="#FC4E07") +geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



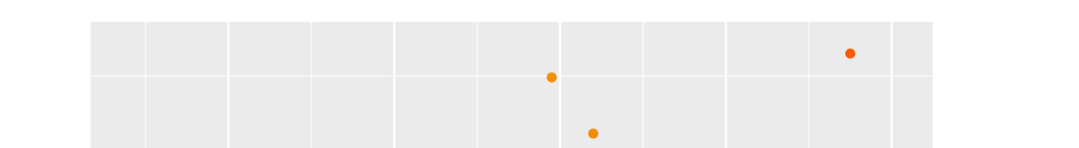

```
#can adjust the settings so that the line is straight by changing the method that geom_smooth is using; there are
additional options for the geom_smooth method.
ggplot(data = aptsales, aes(x=SalePrice, y=GrLivArea)) + geom_point(aes(color = SalePrice)) + scale_color_gradien
t2(midpoint = 160000, low ="#00AFBB", mid="#E7B800", high="#FC4E07") +geom_smooth(method = "lm")
```

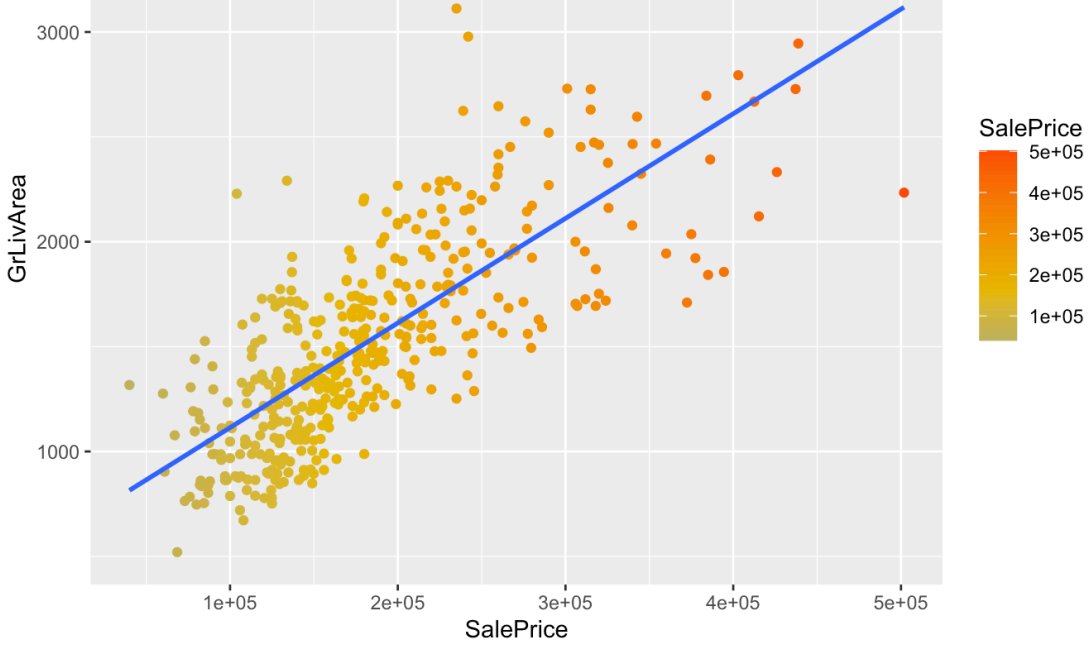
```
## `geom_smooth()` using formula 'y ~ x'
```




```
# turn off the confidence interval with se = F
ggplot(data = aptsales, aes(x=SalePrice, y=GrLivArea)) + geom_point(aes(color = SalePrice)) + scale_color_gradien
t2(midpoint = 160000, low ="#00AFBB", mid="#E7B800", high="#FC4E07") +geom_smooth(method = "lm", se= F)
```

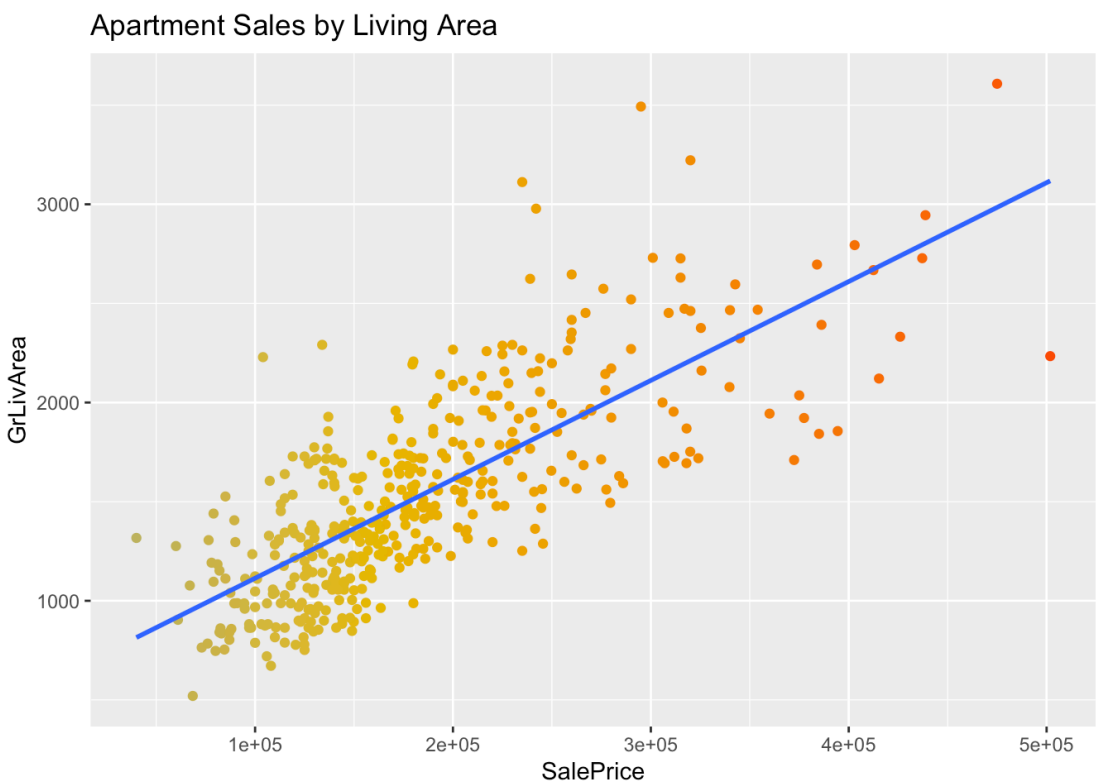
```
## `geom_smooth()` using formula 'y ~ x'
```





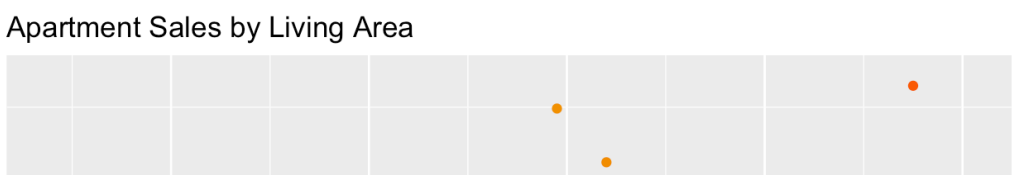

```
#add in the main title
ggplot(data = aptsales, aes(x=SalePrice, y=GrLivArea)) + geom_point(aes(color = SalePrice)) + labs( title = "Apar
tment Sales by Living Area")+ scale_color_gradient2(midpoint = 160000, low ="#00AFBB", mid="#E7B800", high="#FC4E
07", guide = "none") +geom_smooth(method = "lm", se= F)    #can turn off the confidence interval this way
```

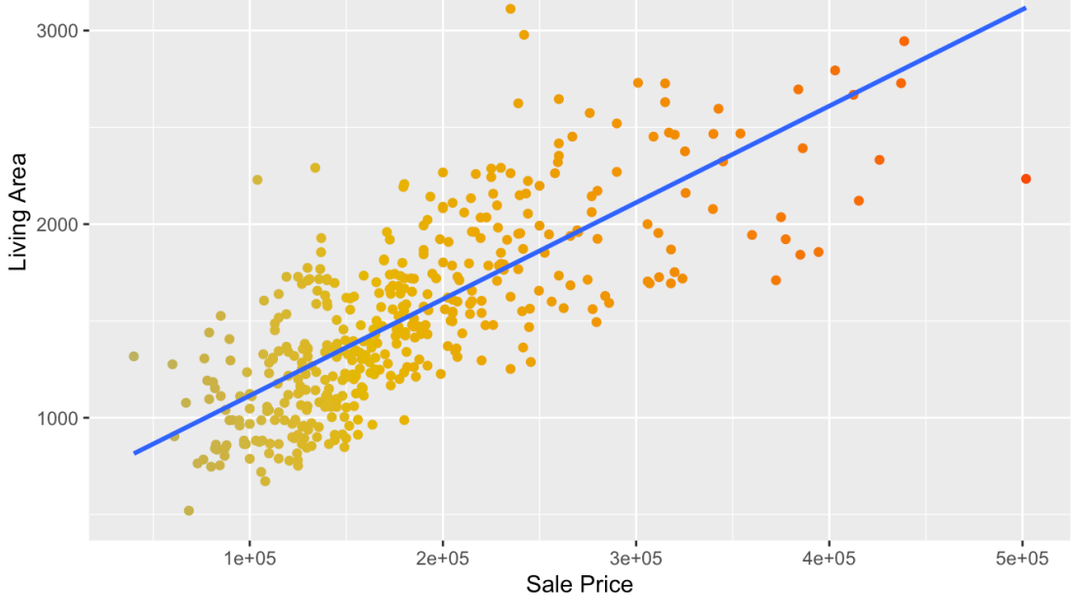
```
## `geom_smooth()` using formula 'y ~ x'
```




```
#add in x and y axis labels
ggplot(data = aptsales, aes(x=SalePrice, y=GrLivArea)) + geom_point(aes(color = SalePrice)) + labs(x= "Sale Price
", y = "Living Area", title = "Apartment Sales by Living Area")+ scale_color_gradient2(midpoint = 160000, low ="#
00AFBB", mid="#E7B800", high="#FC4E07", guide = "none") +geom_smooth(method = "lm", se= F)
```

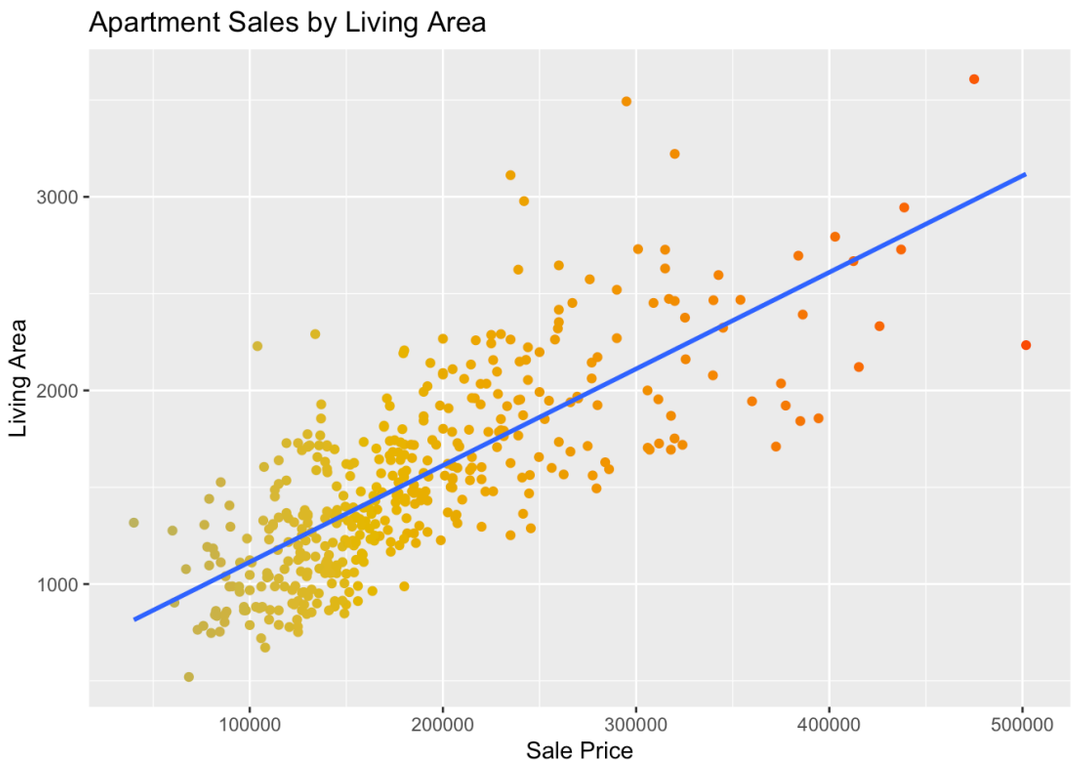
```
## `geom_smooth()` using formula 'y ~ x'
```





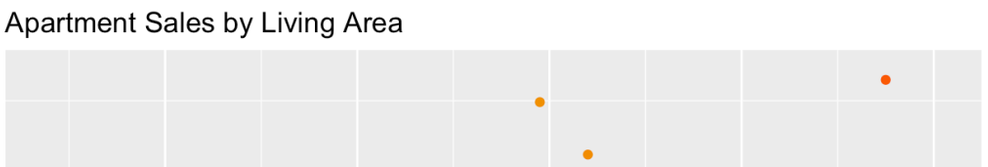

```
#turn off scientific notation for the x axis
options(scipen = 999)

#plot it again!
ggplot(data = aptsales, aes(x=SalePrice, y=GrLivArea)) + geom_point(aes(color = SalePrice)) + labs(x= "Sale Price
", y = "Living Area", title = "Apartment Sales by Living Area")+ scale_color_gradient2(midpoint = 160000, low ="#
00AFBB", mid="#E7B800", high="#FC4E07", guide = "none") +geom_smooth(method = "lm", se= F)    #can turn off the c
onfidence interval this way
```

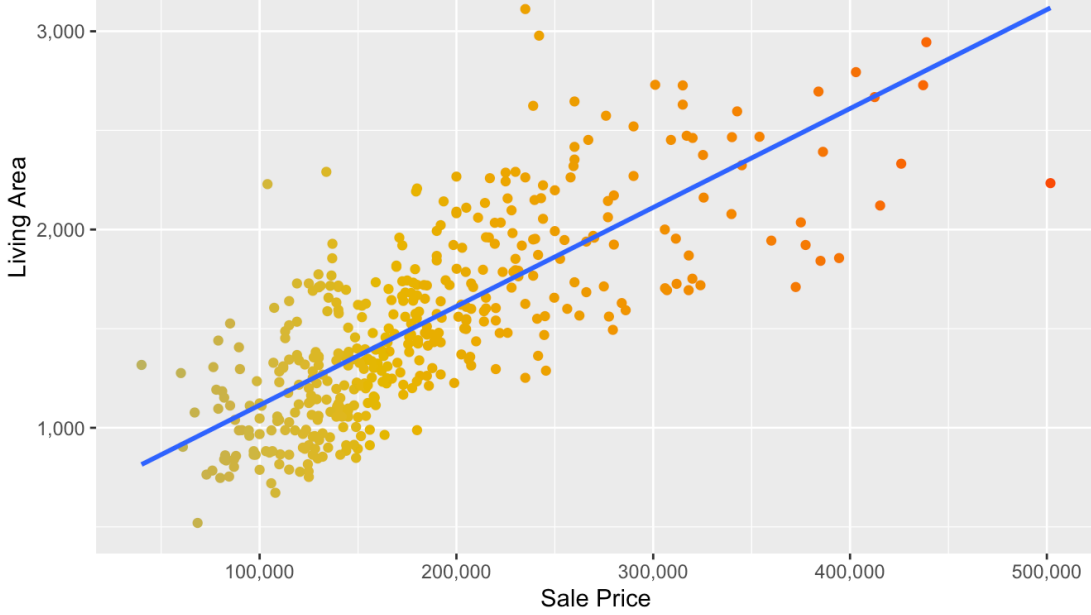
```
## `geom_smooth()` using formula 'y ~ x'
```




```
#install.packages("scales")
#library(scales)
ggplot(data = aptsales, aes(x=SalePrice, y=GrLivArea)) + geom_point(aes(color = SalePrice)) + labs(x= "Sale Price
", y = "Living Area", title = "Apartment Sales by Living Area")+ scale_color_gradient2(midpoint = 160000, low ="#
00AFBB", mid="#E7B800", high="#FC4E07", guide = "none") +geom_smooth(method = "lm", se= F)+scale_x_continuous(lab
els = scales::comma)+scale_y_continuous(labels = scales::comma)
```

```
## `geom_smooth()` using formula 'y ~ x'
```





Read more about the options for the geom_smooth method here. The ggplot2 cheatsheet is also very helpful for exploring the different endless options for your ggplots!

Looking ahead!

We will learn how to make area graphs in Excel, Tableau, and R, also via zoom 4pm est on Nov. 18th to conclude our lecture series this semester on data visualization.

Please reach out with any questions, comments, or feedback at katherine.kuiper25@uga.edu.

Works Cited

Cash, Ivan. https://i.imgur.com/DNxvj.png.
Chang, Agnes. 2020. Introduction to Data Visualization: Course Syllabus. Columbiaviz.githib.io
Cieslak, Jack. 13 Powerful ways to visualize your data. Sisense.com
Durcevel, Sandra. Designing Charts and Graphs: How to Choose the Right Data visualization Types. May 2nd, 2019. datapine.
https://www.smithsonianmag.com/history/surprising-history-infographic-180959563/
Durcevel, Sandra. Designing Charts and Graphs: How to Choose the Right Data visualization Types. May 2nd, 2019. datapine.
ETA+. 2020. Unsplash.
Friendly, Michael. Historical Milestones. http://www.datavis.ca/
Friendly, Michael. 2008, “The Golden Age of Statistical Graphics,”Statistical Science 8(4), 502—535.
https://ggplot2.tidyverse.org
Friendly, Michael and Daniel Denis. 2005. The Early Origins and Development of the Scatterplot. Journal of the History of the Behavioral Sciences. Wiley Publishing.http://datavis.ca/papers/friendly-scat.pdf
Hill, Asha. 2018. 9 Useful R Data Visualization Packages for Any Discipline. Mode. https://mode.com/blog/r-data-visualization-packages/
Hirschman, Dan. 2014. Scatterplot: the unruly darings of public sociology.https://scatter.wordpress.com/category/fun/
Holtz, Yan. 2018. 2_TwoNum. data_to_viz Example_dataset. https://github.com/holtzy/data_to_viz/blob/master/Example_dataset/2_TwoNum.csv
Kiss, Zoltan. 2018. Data and Data Analysis in Linguistics. http://seas3.elte.hu/lingtheo/notes/gkiss-data-in-lcs.html.
Kopf, Dan. 2018. A brief history of the scatterplot-data visualization’s greatest invention. https://qz.com/1235712/the-origins-of-the-scatter-plot-data-visualizations-greatest-invention/
Open Learn. 2020. Session 3: Discrete and Continuous Data. Openlearn Edu: Uk.
https://www.open.edu/openlearn/ocw/mod/oucontent/view.php?id=85587&section=1
https://vocal.media/futurism/most-powerful-star-wars-jedi
Simmons, Ryan. 2016. ENC:2135 Research, Genre & Context. https://dfb1585.wixsite.com/enc2135/single-post/2016/1/10/Blog-Prompt-1
Spiske, Markus. 2020. Unsplash
Stephanie. 2020. Statistics How To: Statistics for the rest of us!. Types of Variable.
Yi, Mike. 2019. A Complete Guide to Scatter Plots. Chartio: Data Tutorials. https://chartio.com/learn/charts/what-is-a-scatter-plot/
Tableau. 2020. Data visualization beginner’s guide: a definition, examples, and learning resources. https://www.tableau.com/learn/articles/data-visualization
Tufte, Edward. 1983,The Visual Display of Quantitative Information.
Tufte, Edward. 1990. Envisioning Information.
Tufte, Edward. 2006. Beautiful Evidence.
UC Denver Libraries. Research Guides: auraria Library. https://guides.auraria.edu/c.php?g=528873&p=3964258. Wickam, Hadley. 2010. A layered grammar of graphics. Journal of Computational and Graphical Statistics.
Wickam, Hadley. 2016. ggplot2: Elegant Graphics for Data Analysis(Use R!). 2nd Edition.
Wiediger, Christian. 2020. Unsplash.