SESUG Paper 238-2019

The Gender Gap in the Education System

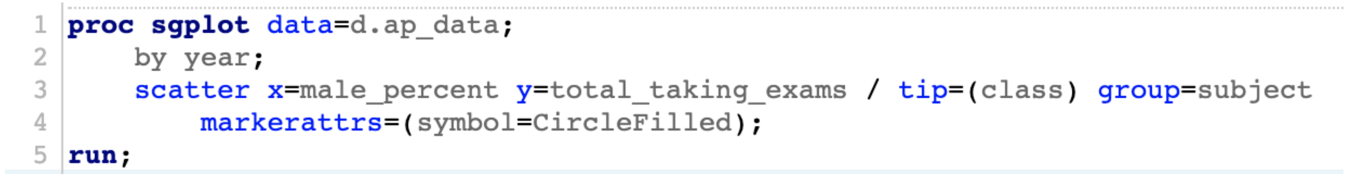
Geoffrey Dean, Green Hope High School

INTRODUCTION

This paper looks into how to create several compelling graphics using SAS. It utilizes SAS to produce several different types of charts, each with a unique purpose in presenting data. The data in these charts visualizes the gender gap in the education system, primarily in science, technology, engineering, and math (STEM) courses through advanced placement (AP) exams. The AP program is an academic option run by The College Board, allowing high school students to learn college-level material, and in many cases earn college credit by demonstrating mastery. The AP Exams are nationally standardized tests administered by The College Board, testing students on the material learned in the respective courses.

COMPARING VARIABLES USING A SCATTERPLOT

The first plot you will see is the scatterplot, which is a clear and efficient method of representing data. It is used when comparing two numeric variables. In this case, looking for patterns in the percentage of students taking each AP Exam that are male, to the total number of students who took the exam that year. The code below uses the SGPLOT procedure to create the graphs shown in Figure 1 and Figure 2.



```
proc sgplot data=d.ap_data;
    by year;
    scatter x=male_percent y=total_taking_exams / tip=(class) group=subject
        markerattrs=(symbol=CircleFilled);
run;
```

There are several elements in the code above that enhance the graphs. The first key element is the BY statement in the second line. This statement creates scatterplots for each year in the data, which will help visualize the data in the context of each year. To conserve space, only figures 1 and 2 are shown below.

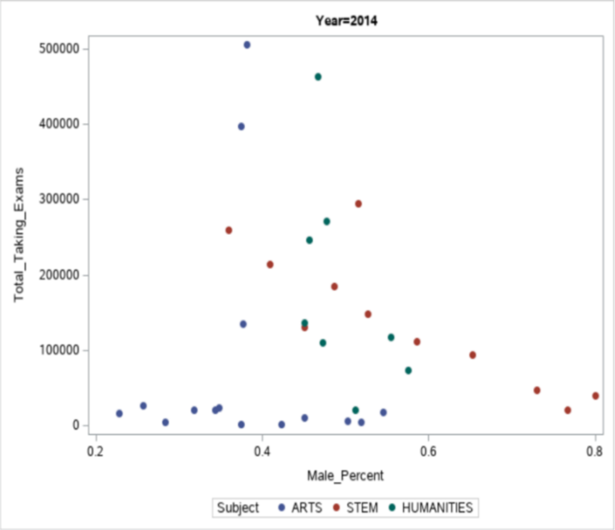


Figure 1 Percentage of students taking AP exams by subject in 2014 that are male

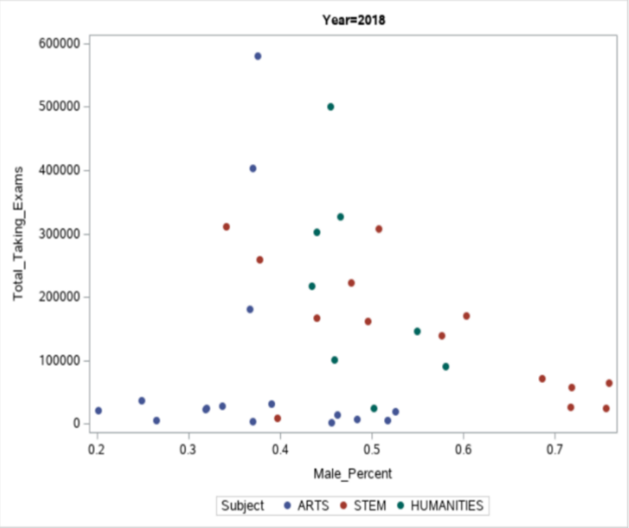


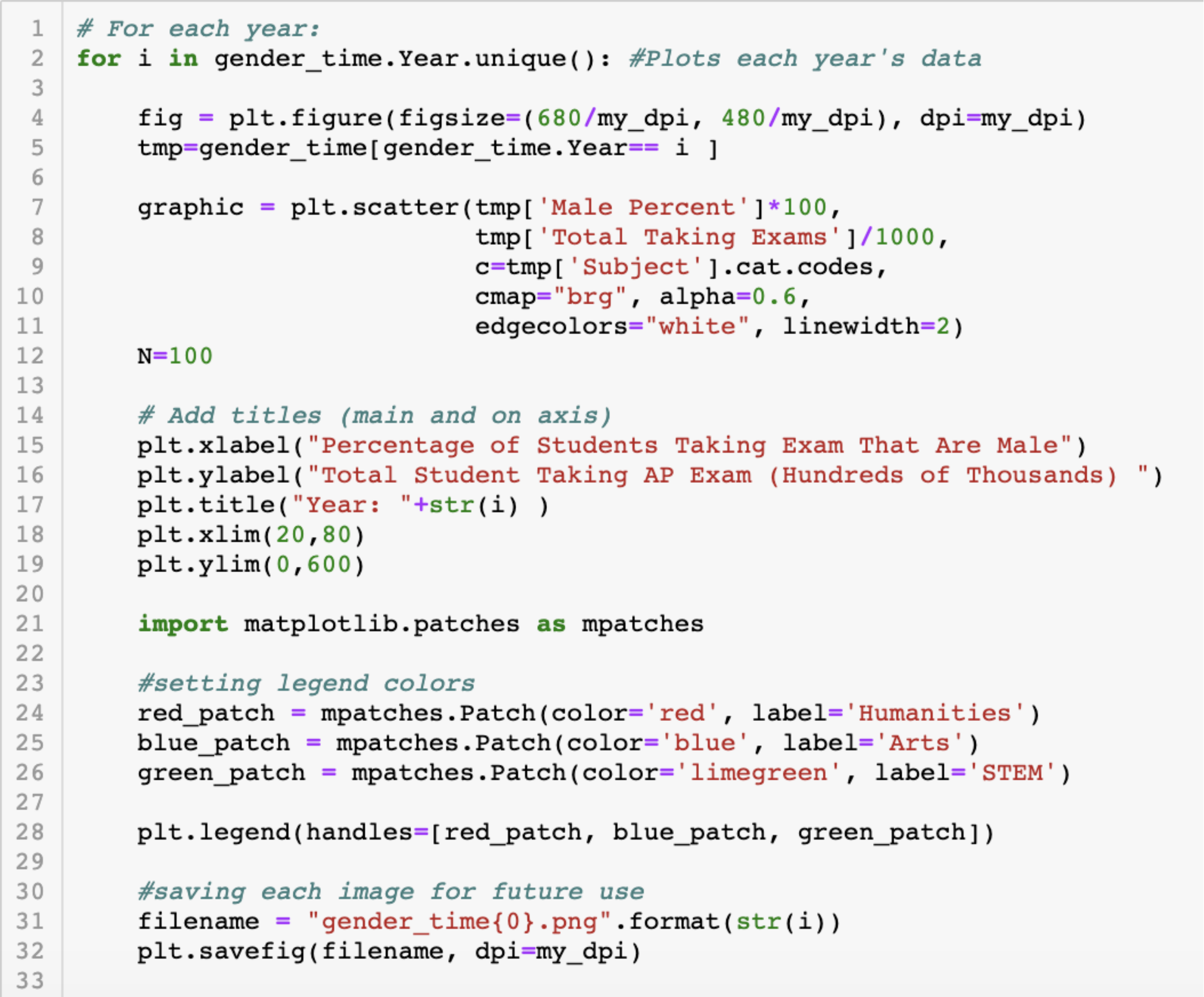
Figure 2 Percentage of students taking AP exams by subject in 2018 that are male

The next element, the TIP = (CLASS) portion of the SCATTER statement, produces tooltips so that readers in an online setting can get additional information on individual points. In doing so, outliers can be identified on both axes, the outliers on the Y-axis are extreme amounts of students, while outliers on the X-axis reveal substantial gaps in gender representation.

The final part of this SAS code block is the GROUP option, which in this case is set to GROUP=SUBJECT. This option makes the subjects, and their corresponding data points, distinct colors. This is helpful because you can see that there are far more male students taking AP exams in STEM classes, but there is also a dearth of males represented in fine arts classes such as visual art or French.

ADVANTAGES OF SAS OVER OPEN SOURCE

From my experience, SAS has a significant advantage over other programming languages in its efficiency in creating these plots. I used the matplotlib package in Python to create a virtually identical plot. However, it took considerably more code than was required in the above SAS example. Below is the commented code I used to create scatterplots for each year's data showing the same gender breakdown.



```python
# For each year:
for i in gender_time.Year.unique(): #Plots each year's data

    fig = plt.figure(figsize=(680/my_dpi, 480/my_dpi), dpi=my_dpi)
    tmp=gender_time[gender_time.Year== i ]

    graphic = plt.scatter(tmp['Male Percent']*100,
                          tmp['Total Taking Exams']/1000,
                          c=tmp['Subject'].cat.codes,
                          cmap="brg", alpha=0.6,
                          edgecolors="white", linewidth=2)
    N=100

    # Add titles (main and on axis)
    plt.xlabel("Percentage of Students Taking Exam That Are Male")
    plt.ylabel("Total Student Taking AP Exam (Hundreds of Thousands) ")
    plt.title("Year: "+str(i) )
    plt.xlim(20,80)
    plt.ylim(0,600)

    import matplotlib.patches as mpatches

    #setting legend colors
    red_patch = mpatches.Patch(color='red', label='Humanities')
    blue_patch = mpatches.Patch(color='blue', label='Arts')
    green_patch = mpatches.Patch(color='limegreen', label='STEM')

    plt.legend(handles=[red_patch, blue_patch, green_patch])

    #saving each image for future use
    filename = "gender_time{0}.png".format(str(i))
    plt.savefig(filename, dpi=my_dpi)
```

The above code produced Figure 3, compared with the previously discussed Figure 2 (reprinted here for ease of comparison):

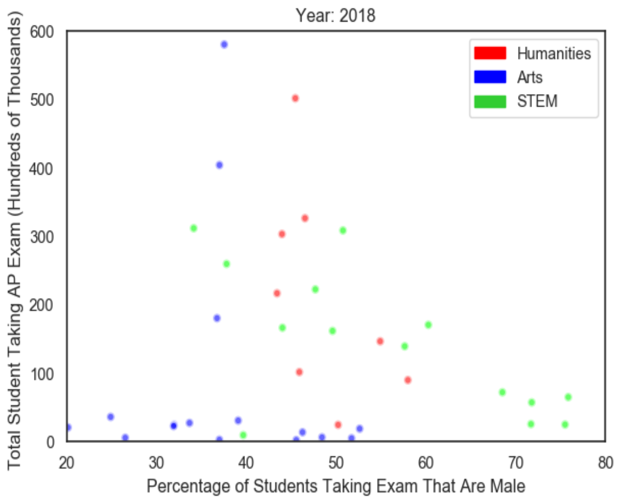


Figure 3 Percentage of male students taking AP exams by subject in 2018 using Python Matplotlib

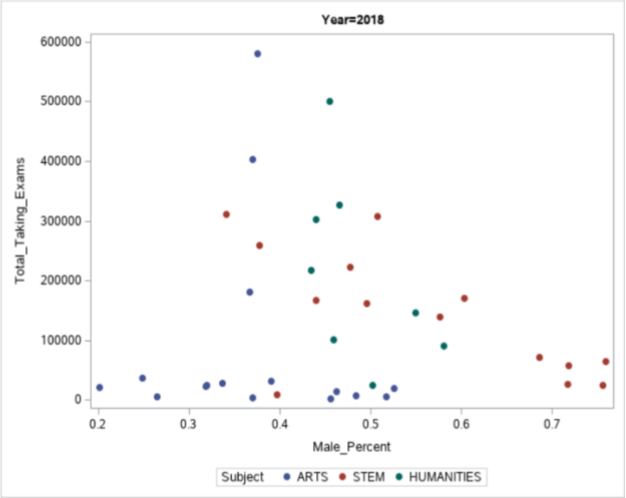
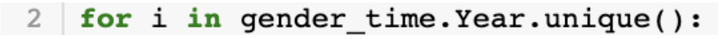


Figure 4 Reprint of Figure 2 for easy comparison

These two charts are nearly indistinguishable from one another from a functionality standpoint, however creating Figure 2 required a fraction of the code. One specific area in which SAS was far more efficient was in creating separate charts for each year. In Python, you loop through each year by using the leading:



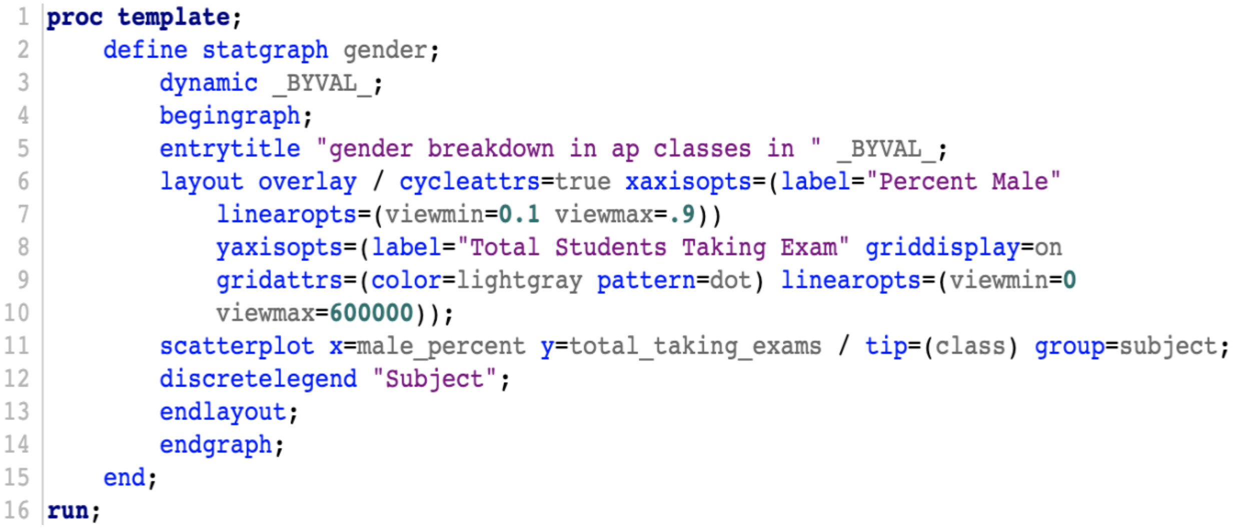
```
for i in gender_time.Year.unique():
```

statement. This line of code required the program to apply the indented processes for each unique year in the data set, 2014-2018, something that SAS accomplishes with the BY statement.

ANIMATING GRAPHICS TO SHOW CHANGES OVER TIME

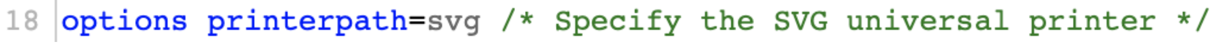
A more sophisticated way to represent this shift is to create a time-lapse animation of the chart for each year. This is an attractive option if the display media is online. The initial step in creating a time-lapse animation is through the TEMPLATE procedure. Creating a template for graphs, enables SAS to use the same format to create different graphs for each individual year, streamlining the animation process. Using PROC TEMPLATE, you define STATGRAPH as GENDER, a variable referenced later.

There are several parts to the template statement, the first being the dynamic variable defined in line 3. This sets YEAR as the variable that will change through the different charts, and as you can see in line 5, the dynamic variable helps create the title. Within the DEFINE statement, you see several settings that determine the appearance of the chart, all of which can be changed according to your preference. Note at the bottom of this block you add a SCATTERPLOT statement that is virtually identical to the code used to define Figure 1 and 2. This indicates that the data is to be represented using a scatter plot, but it can be adjusted to a different chart type if desired.



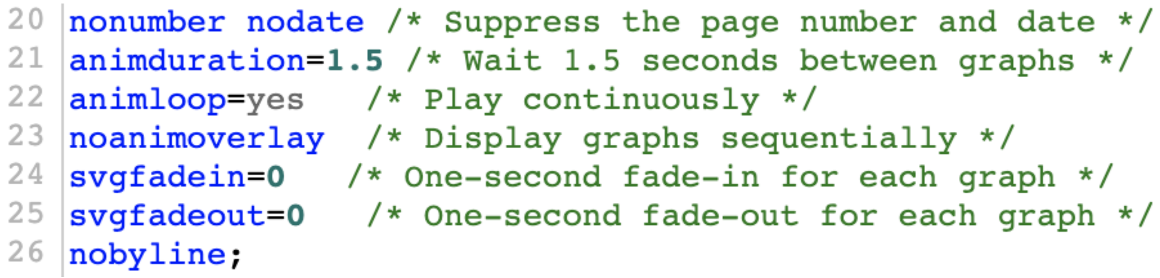
```
 1 proc template;
 2     define statgraph gender;
 3         dynamic _BYVAL_;
 4         begingraph;
 5         entrytitle "gender breakdown in ap classes in " _BYVAL_;
 6         layout overlay / cycleattrs=true xaxisopts=(label="Percent Male"
 7             linearopts=(viewmin=0.1 viewmax=.9))
 8             yaxisopts=(label="Total Students Taking Exam" griddisplay=on
 9             gridattrs=(color=lightgray pattern=dot) linearopts=(viewmin=0
10             viewmax=600000));
11         scatterplot x=male_percent y=total_taking_exams / tip=(class) group=subject;
12         discretelegend "Subject";
13         endlayout;
14         endgraph;
15     end;
16 run;
```

This OPTIONS statement below specifies which filetype to create in rendering the chart; in this case, it is a Scalable Vector Graphic (SVG). SVG is an image format that is particularly useful for charts and visualizations, because of its ability to support interactivity.



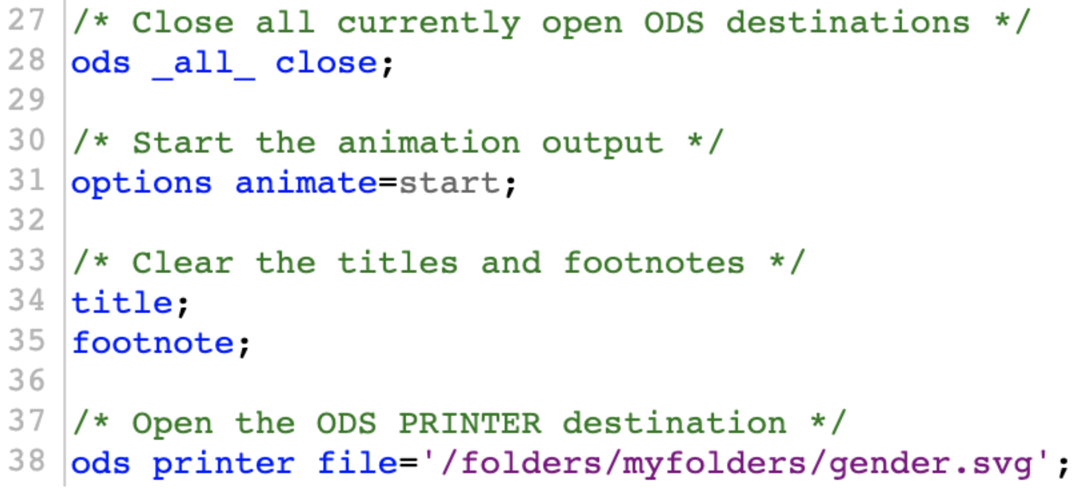
```
18 options printerpath=svg /* Specify the SVG universal printer */
```

Below are settings to control the animation display. For example, the ANIMDURATION option indicates how long the chart will be displayed. Here it is set one and a half seconds, obviously a personal preference.



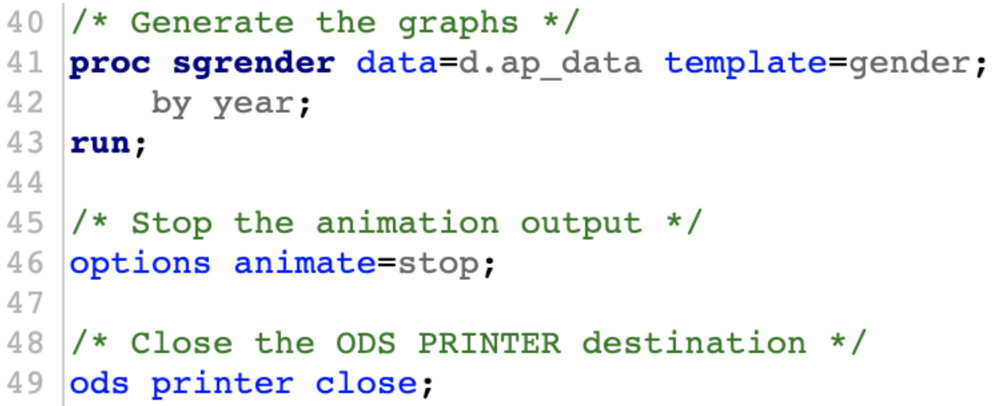
```
nonumber nodate /* Suppress the page number and date */
animduration=1.5 /* Wait 1.5 seconds between graphs */
animloop=yes   /* Play continuously */
noanimoverlay  /* Display graphs sequentially */
svgfadein=0   /* One-second fade-in for each graph */
svgfadeout=0   /* One-second fade-out for each graph */
nobyline;
```

These next few steps reset options that may affect the animation creation process, such as closing the ODS destinations and setting the titles and footnotes to blanks. Additionally, you initialize the animation using the OPTIONS ANIMATE = START. You also need to specify the location to save the file. Since the PRINTERPATH was previously defined as SVG, then the file should have an '.svg' extension. If a specific file format is not specified in the ODS PRINTER FILE= statement, the file will automatically be saved as a '.txt' file. A .txt extension will prevent the animation from displaying properly in your web browser.



```
/* Close all currently open ODS destinations */
ods _all_ close;

/* Start the animation output */
options animate=start;

/* Clear the titles and footnotes */
title;
footnote;

/* Open the ODS PRINTER destination */
ods printer file='/folders/myfolders/gender.svg';
```

Below you create the graphs once again, using the SGRENDER procedure and setting the DATA=d.ap_data. This is where you reference the template created earlier, named GENDER. The template is referenced using TEMPLATE=GENDER, which generates the graphs for each year using the BY statement. Then, the animation is stopped, the ODS printer is closed, and the creation of the animation ends.



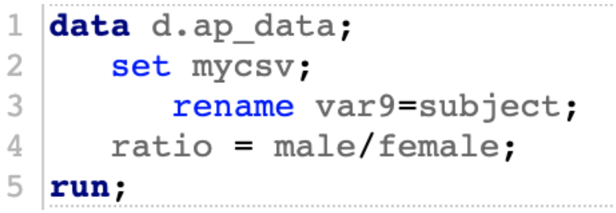
```
/* Generate the graphs */
proc sgrender data=d.ap_data template=gender;
    by year;
run;

/* Stop the animation output */
options animate=stop;

/* Close the ODS PRINTER destination */
ods printer close;
```

After this code has been run, you will have an '.svg' file that can be opened in a web browser. This can be viewed locally or embedded into a web page. No access to SAS is needed for the animation to run in a web browser.

SHOWING CHANGES OVER TIME USING A LINE PLOT

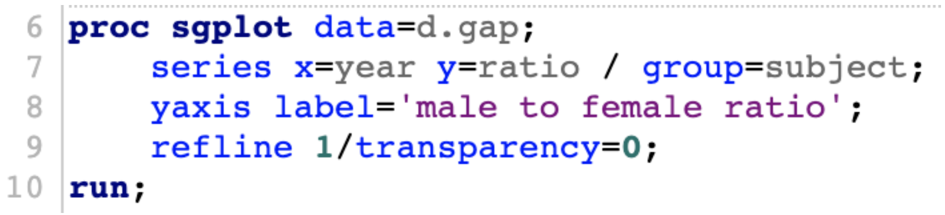
Another way to represent changes over time is the line plot, referred to in SAS as a SERIES plot. This style of graphic helps to show changes, or in this case the lack of change, that occur over time. In this case, the change of interest is a shift in the ratio of males to females who take certain AP classes, from 2014 to 2018.

To create this line plot, we will need to define a ratio variable, shown below in the DATA step code:



```
data d.ap_data;
   set mycsv;
      rename var9=subject;
   ratio = male/female;
run;
```

Ratio values close to zero indicate the AP exam for that particular subject is taken almost exclusively by females. A ratio of 1 is perfectly balanced between female and male students, and a large ratio indicates the AP exam is taken predominately by males. After creating this variable, the next step is creating the line plot using the following code:



```
proc sgplot data=d.gap;
    series x=year y=ratio / group=subject;
    yaxis label='male to female ratio';
    refline 1/transparency=0;
run;
```

Note: While this variable is being used to reveal a gender gap, it is not using a gap in the literal sense, as no subtraction operation is taking place.

Once again, the data is grouped by subject using a GROUP=SUBJECT option in the SERIES statement. This draws attention to the differences in the ratio across the various subject areas.

To ease reading, you should add a reference line at y=1 using the REFLINE statement. This reference line represents a perfectly balanced class. The transparency option ranges from zero to one, zero (fully visible), to one (fully transparent). This helps to visualize the slight ratio change toward more females in the humanities, as shown in Figure 5.

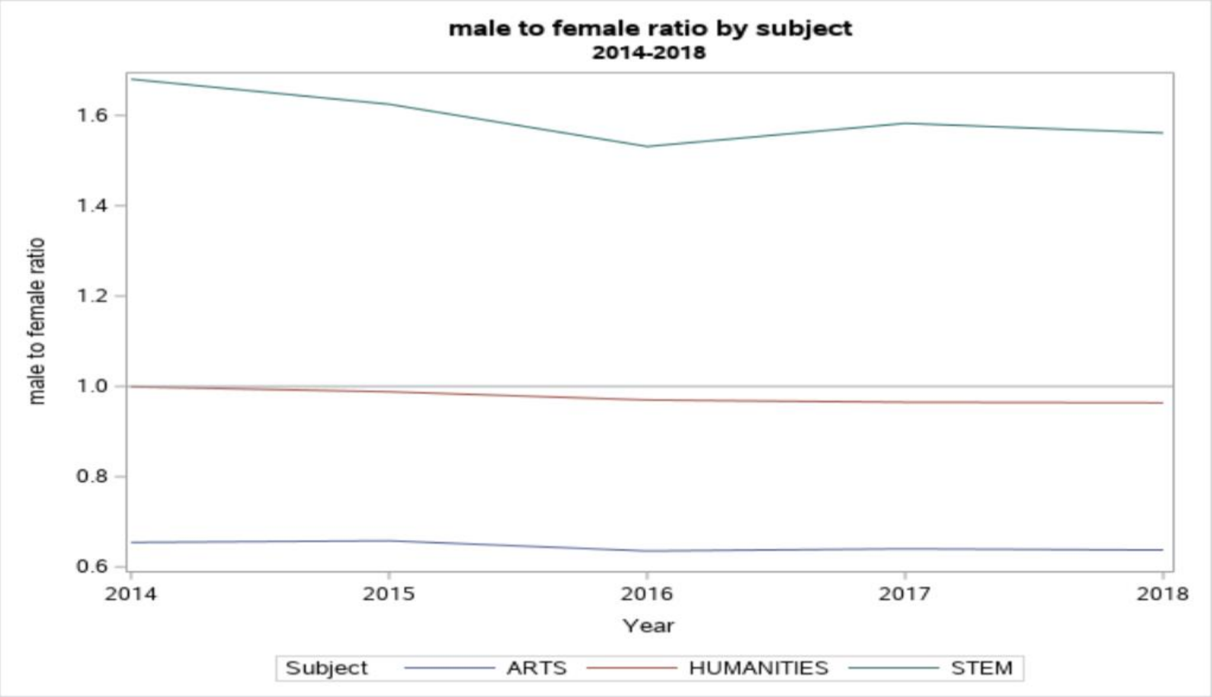
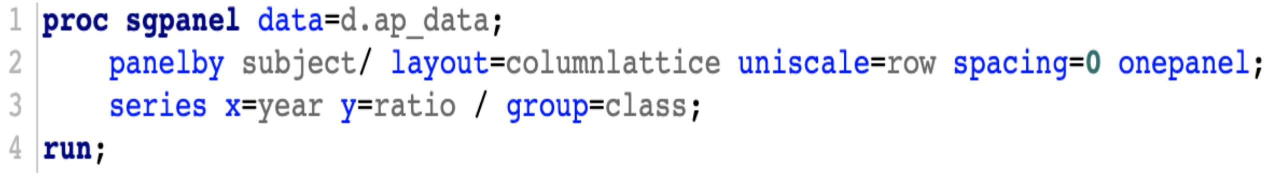


Figure 5 The ratio of males to females of AP Exam test-takers in the three subject groups from 2014-2018

From Figure 5, it is apparent that despite efforts to improve gender equity in education, little change has occurred between 2014 and 2018. There is still a relatively equal distribution of males and females in humanities classes, there are still far more females in art classes, and the severe underrepresentation of females in STEM classes has not improved. To investigate further, you can break down each line representing the subjects and expand them to show the individual classes within subjects.

USING MULTIPLE LINE PLOTS FOR FURTHER ANALYSIS

We can make multiple plots and place them horizontally to compare the gender ratio of all classes across the three subject groups. This will help you compare the different classes within the subjects across the five years. This can be done with the SGPANEL procedure, and by using the PANELBY SUBJECT statement to create the three subject panels. Next, you set the layout option to COLUMNLATTICE to align the panels in columns. Additionally, the GROUP=CLASS in the SERIES statement allows us to distinguish between the various classes within the subject panels.



```
proc sgpanel data=d.ap_data;
    panelby subject/ layout=columnlattice uniscale=row spacing=0 onepanel;
    series x=year y=ratio / group=class;
run;
```

This creates three separate panels, one with each subject inside of it, and places them in three distinct columns. You can also use the ROWLATTICE option to flip the orientation. In this case, I think COLUMNLATTICE generates a better visual. The COLUMNLATTICE chart is shown in Figure 6.

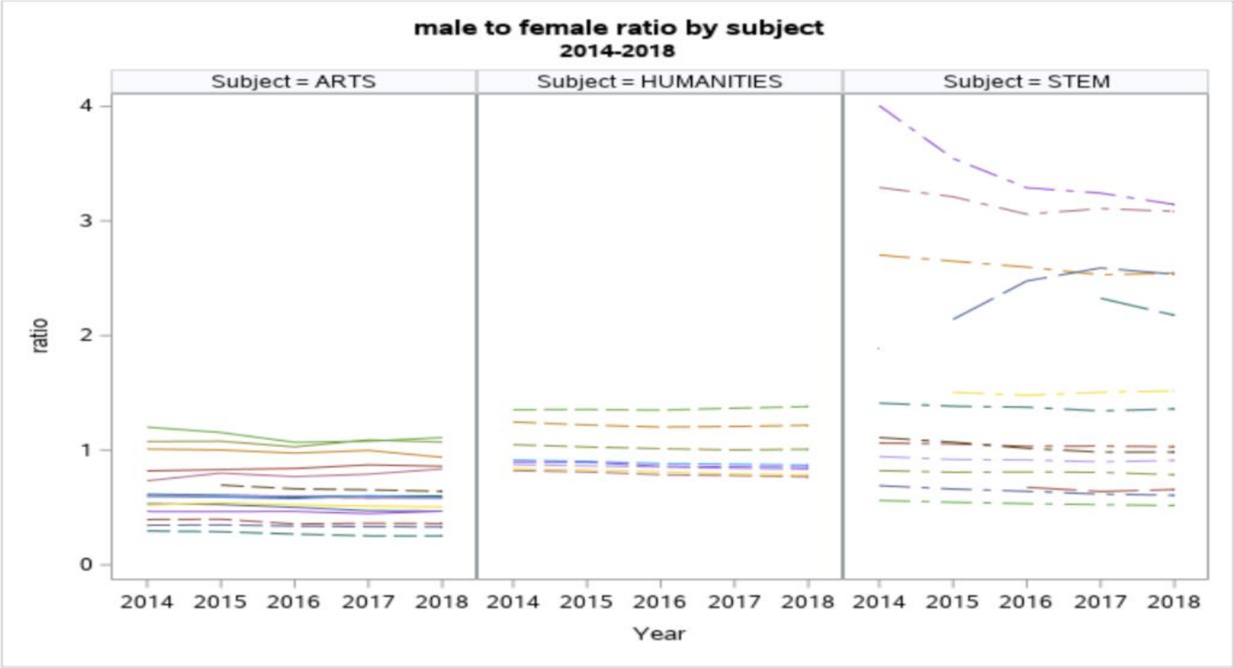


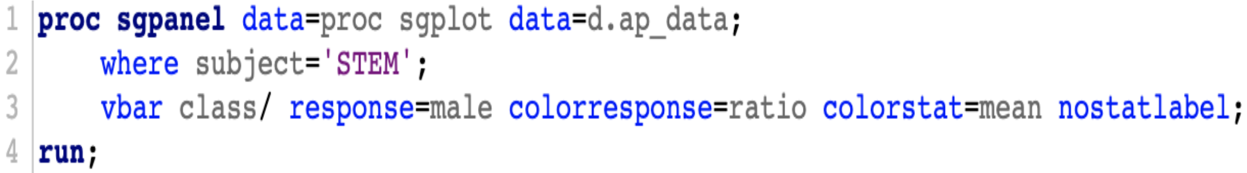
Figure 6 The ratio of males to females who took the AP Exam in all subjects, categorized by subject, 2014-2018.

By plotting the classes in panels by subject, you can compare all the classes across the five years, and something interesting is revealed. This chart shows that the severe gender gap does not exist in your average STEM class, but instead, it is a few outlier STEM classes that increases the mean.

The next step is to identify those five outlier classes, which you can do with a bar chart.

USING A BAR CHART AND COLOR RAMP TO SHOW THE GENDER GAP

As shown in Figure 6, the gender gap is not ubiquitous across all STEM classes; in reality, it is only extreme in a few STEM classes. To compare the gender gap across STEM courses, you can use a bar chart. The bar chart is particularly useful in visualizing quantities and total amounts. Additionally, you can customize your bar chart using several options. You specify the orientation of the chart within PROC SGPLOT, by calling the VBAR statement. You can also use the HBAR statement to perform the horizontal analog. I chose VBAR because I think it is easier to visually compare the different values. The WHERE statement filters the data by subject, allowing you to look exclusively at STEM class data.



```
proc sgpanel data=proc sgplot data=d.ap_data;
    where subject='STEM';
    vbar class/ response=male colorresponse=ratio colorstat=mean nostatlabel;
run;
```

Figure 7 shows the number of males taking each exam, indicated by the RESPONSE = MALE option. Additionally, by adding the COLORRESPONSE condition, you can visualize the gender gap for each STEM class, using a color ramp to denote the gender ratio. The metric used is specified by the COLORSTAT=MEAN condition, indicating the color scale used is based on the mean gender ratio from the data. The resulting chart is shown in Figure 7.

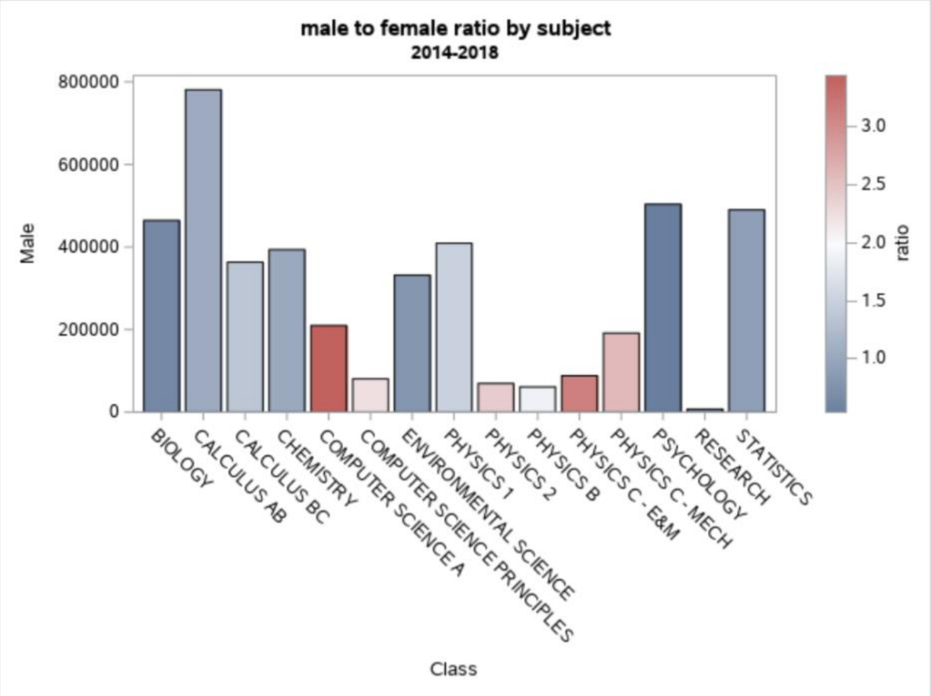


Figure 7 The cumulative number of students taking the AP Exam in STEM classes, set to a color scale of the average male to female ratio in those classes, 2014-2018.

From Figure 7, you can conclude that the classes in which there is the highest disparity between genders, have among the fewest students taking the AP exam. The three most disparate classes are Computer Science A, Physics C – Mechanical, and Physics C-Electric and Mechanical, some of the smallest STEM classes by student enrollment.

CONCLUSION

There are several ways to represent data using SAS, primarily through the SGPLOT procedure. Each type of representation brings a unique approach to the data, allowing one to investigate multiple facets of a data-based question. Additionally, you can use these charts to discern things that may not be apparent at first glance, as a deeper analysis of the gender gap in STEM classes revealed.

REFERENCES

“AP Archived Data – Research – College Board.” *Research*, College Board, 5 Mar. 2018, research.collegeboard.org/programs/ap/data/archived.

SAS. “Creating a Graph with Data Tips in an HTML Page.” *Documentation.sas.com*, SAS Institute, documentation.sas.com/?docsetId=grstatug&docsetTarget=n19378esl0uxoun1gsipgcqrjl2s.htm&docsetVersion=9.4&locale=en.

SAS. “Create an Animated Graph.” *Documentation.sas.com*, SAS Institute, documentation.sas.com/?docsetId=grstatug&docsetTarget=p1xw4cbegqxrshn184m0n13yjfky.htm&docsetVersion=9.4&locale=en.

Slaughter, Susan J, and Lora D Delwhiche. “SAS Global Forum 2010.” *Using PROC SGPLOT for Quick High-Quality Graphs*, p. 7.

ACKNOWLEDGMENTS

Firstly, I would like to thank my father, Jared Dean, for introducing me to SAS and patiently answering all of my questions during the learning process. Secondly, my mother, Katie Dean, who has given me boundless amounts of feedback, and has been extraordinarily helpful in the drafting process. Additionally, I would like to thank my high school teacher, Mr. Colin Richardson, for inspiring me to think more critically about the world and challenging me to question why things are the way they are.

CONTACT INFORMATION

Your comments and critiques are greatly valued. Feel free to reach me at:

Geoffrey Dean
Green Hope High School

Email: dean.geoffrey@gmail.com
Twitter: @geoffreycdean
Website: geoffreydean38.wordpress.com

APPENDIX A. CODE

```
FILENAME CSV "/folders/myfolders/ap_data_all.csv";

/** Import the CSV file.  **/
PROC IMPORT DATAFILE=CSV OUT=WORK.MYCSV DBMS=CSV REPLACE;
     guessingrows=200;
RUN;

/** Unassign the file reference.  **/
FILENAME CSV;
libname d '/folders/myfolders';

/* Sorting data by year, makes data easier to process */
proc sort data=d.ap_data;
     by year;
run;

/* Figure 1 and Figure 2 code */
proc sgplot data=d.ap_data;
     by year;
     scatter x=male_percent y=total_taking_exams / tip=(class)
group=subject
          markerattrs=(symbol=CircleFilled);
run;

/* SVG animation code */
proc template;
     define statgraph gender;
          dynamic _BYVAL_;
          begingraph;
          entrytitle "gender breakdown in ap classes in "
_BYVAL_;
          layout overlay / cycleattrs=true
xaxisopts=(label="Percent Male"
               linearopts=(viewmin=0.1 viewmax=.9))
               yaxisopts=(label="Total Students Taking Exam"
griddisplay=on
               gridattrs=(color=lightgray pattern=dot)
linearopts=(viewmin=0
               viewmax=600000));
          scatterplot x=male_percent y=total_taking_exams /
tip=(class) group=subject;
```

```
          discretelegend "Subject";
          endlayout;
          endgraph;
     end;
run;

options printerpath=svg /* Specify the SVG universal printer */
nonumber nodate /* Suppress the page number and date */
animduration=1.5 /* Wait 1.5 seconds between graphs */
animloop=yes   /* Play continuously */
noanimoverlay  /* Display graphs sequentially */
svgfadein=0   /* One-second fade-in for each graph */
svgfadeout=0   /* One-second fade-out for each graph */
nobyline;

/* Close all currently open ODS destinations */
ods _all_ close;

/* Start the animation output */
options animate=start;

/* Clear the titles and footnotes */
title;
footnote;

/* Open the ODS PRINTER destination */
ods printer file='/folders/myfolders/gender.svg';

/* Generate the graphs */
proc sgrender data=d.ap_data template=gender;
     by year;
run;

/* Stop the animation output */
options animate=stop;

/* Close the ODS PRINTER destination */
ods printer close;

/* Code used to create ratio variable for Figure 5 */
data d.ap_data;
     set mycsv;
     rename var9=subject;
     ratio=male/female;
run;

/* Figure 5 code */
```

```
proc sgplot data=d.gap;
     series x=year y=ratio / group=subject;
     yaxis label='male to female ratio';
     refline 1/transparency=0;
run;

/* Figure 6 code */
proc sgpanel data=d.ap_data;
     panelby subject/ layout=columnlattice uniscale=row
spacing=0 onepanel;
     series x=year y=ratio / group=class;
run;

/* Figure 7 code */
proc sgplot data=d.ap_data;
     where subject='STEM';
     vbar class/ response=male colorresponse=ratio
colorstat=mean nostatlabel;
run;
```