

Paper 364-2013

Creating a Useful Operational Report

Andy Hummel, Delta Air Lines, Atlanta, GA

Robert Goldman, Delta Air Lines, Atlanta, GA

ABSTRACT

Metrics and reports are highly valued by operational decision makers in order to make informed and data driven conclusions. However, there is a balance between presenting useful organized knowledge and displaying page upon page of raw useless data. SAS® has the power to produce a wide range of sophisticated graphs that are meaningful. The challenge is to produce a graph that quickly and accurately measures and displays the real world operation while allowing decision makers to make operationally beneficial determinations. There are numerous SAS papers that give step-by-step instructions on how to build a graph; this is not such a paper. The goal of this paper is to show how we approached the challenge of building an operational report and the techniques used.

INTRODUCTION

Delta Air Lines is fortunate to have a large amount of warehoused operational data. In particular, we recently began receiving aircraft ground surface movement data which is used to track the movement of aircraft from the time an airplane pushes back from the gate to the time it becomes airborne and from touchdown to the destination gate. This ground movement time is referred to as “taxi time”. Taxi time is primarily comprised to two ground movement locations: taxiways and ramp areas.

Airports can sometimes resemble rush hour traffic in New York City where there are too many aircraft and too little real estate; this capacity-demand imbalance causes excessive taxi time which results in unpredictability and congestion on the airport surface leading to customers missing connection flights, and excess fuel burn and CO2 emissions. Ultimately, excess taxi time costs an airline millions of dollars annually in operating expenses and reduces the asset utilization rates of its aircraft. Our assignment was to create a report that measures taxi time and this paper lays out the lessons learned in attempting to produce an operational taxi time report.

IT LOOKS PRETTY, BUT WHAT DOES IT TELL ME?

SAS has the ability to produce a wide range of charts and plots. Delta Air Lines use SAS Enterprise Guide and the point and click feature allows the user to take a data set and quickly graph the data in numerous ways. The results can be colorful and visually appealing but not always easily understandable. As a general rule, if the SAS user producing the graph cannot understand the message, the end recipient of the report will have no idea what the graph is trying to convey.

The below graph is a stacked pie chart and attempts to show aircraft movement time at for the six ramps at the Hartsfield-Jackson Atlanta International Airport. While visually impressive the message of the graph is not easily understood.

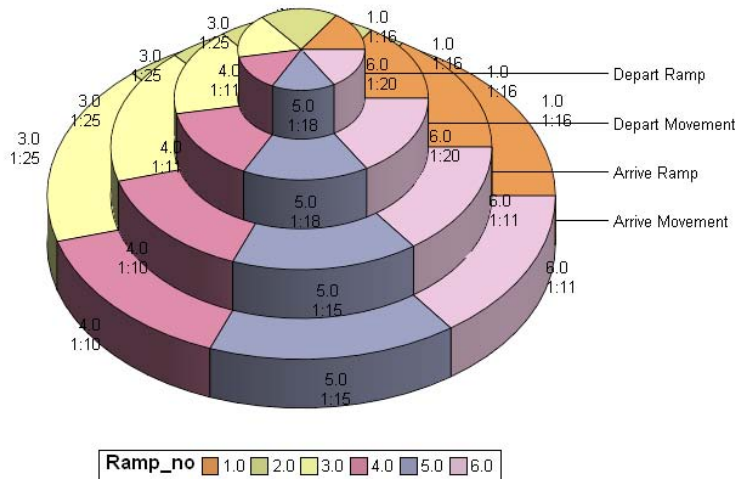


Figure 1. A colorful but confusing stacked pie chart.

GETTING STARTED

Our first step to building a useful operational report was to graph the ramp time for departing aircraft¹. The bar chart below shows the departing ramp time for Ramp 1 in Atlanta by hour of the day. A quick look at the chart shows that not every hour of the day is displayed. In order for the chart to be useful additional detail is needed.

Code to produce the bar chart.

```
PROC GCHART DATA=bar_and_line_2;
  VBAR Hour / SUMVAR=avg_mins CLIPREF FRAME TYPE=MEAN
  COUTLINE=BLACK RAXIS=AXIS1 MAXIS=AXIS2 GAXIS=AXIS3 GAXIS=AXIS3;
  BY Ramp_no;
RUN; QUIT;
```

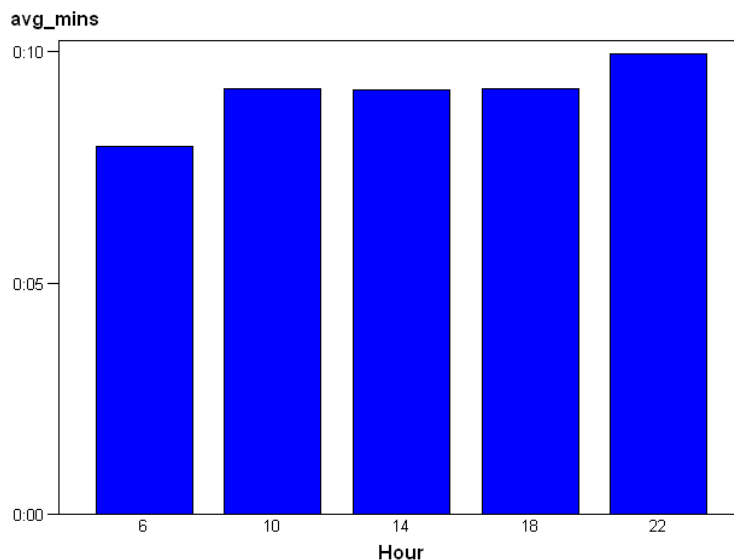


Figure 2. A basic bar chart.

By default SAS selects the number of bars to display. However, SAS also allows the user three options regarding the number of bars to use. The options are: 1) one bar for each unique data value, 2) a defined number of bars entered by the user and 3) specific bar values.

For our report we wanted to see one bar for every hour (even if the hour value was zero). By adding `DISCRETE` to the `PROC GCHART` code, each hour was displayed and the chart now showed the additional information we needed.

```
PROC GCHART DATA=bar_and_line_2;
  VBAR Hour / SUMVAR=avg_mins CLIPREF FRAME DISCRETE TYPE=MEAN
  COUTLINE=BLACK RAXIS=AXIS1 MAXIS=AXIS2 GAXIS=AXIS3 GAXIS=AXIS3;
  BY Ramp_no;
RUN; QUIT;
```

The graph on the following page displays departure ramp time for every hour of the day.

¹ Departure taxi time is the combination of “departure ramp time” and “departure movement time”. “Departure ramp time” is defined as the time the aircraft spends between pushing back from the departure gate to time it reaches the active movement area. “Departure movement time” is the time the aircraft spends from reaching the active taxi and runway areas to when the wheels leave the ground. In the case of Atlanta, each ramp is managed by a Delta Air Lines or Airport Authority “ramp controller.” The “movement area” is, as it is at all US airports, controlled by FAA ATC Ground Controllers.

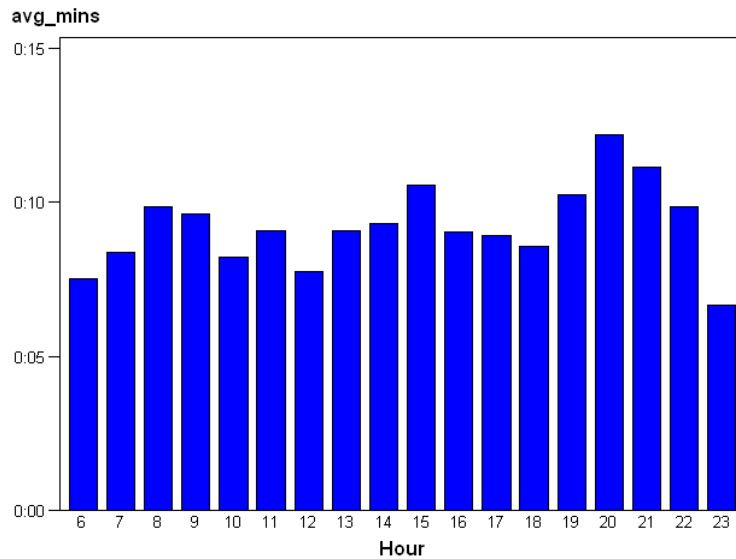


Figure 3. A bar chart using with every hour displayed.

However, we also have the option to specify only specific values. This would be useful if we wanted to see only the morning rush of 8:00 to 10:00 and afternoon rush of 16:00 to 18:00 on the same chart. To specify specific values we add `MIDPOINTS` to the `PROC GCHART` statement. The bars will be displayed in the order they appear in the `MIDPOINTS` statement. Additionally, the same bar can be displayed more than once by typing the value repeated times. For the bar chart below we defined the hours of 8:00, 9:00, 10:00, 16:00, 17:00 and 18:00.

```
PROC GCHART DATA=bar_and_line_2;
  VBAR Hour / SUMVAR=avg_mins CLIPREF FRAME MIDPOINTS=8 9 10 16 17 18 TYPE=MEAN
  COUTLINE=BLACK RAXIS=AXIS1 MAXIS=AXIS2 GAXIS=AXIS3 GAXIS=AXIS3;
  BY Ramp_no;
RUN; QUIT;
```

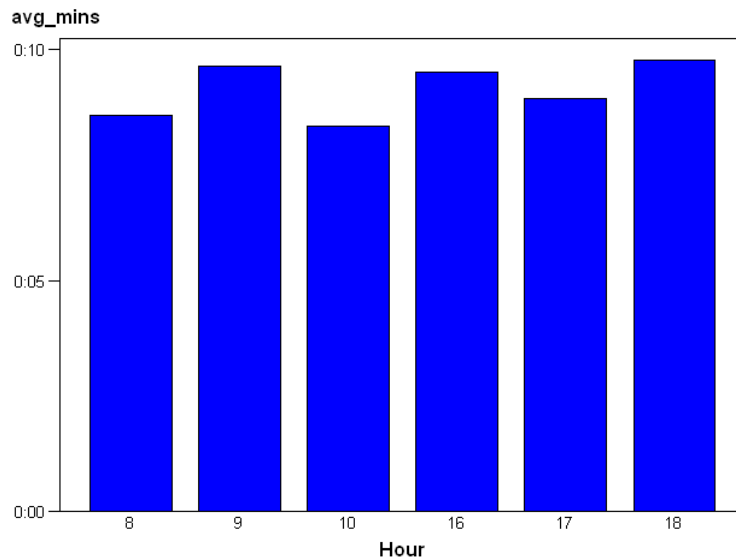


Figure 4. A bar chart with only specific hours displayed.

CONDENSING THE DATA

We now have a chart showing the departing ramp time for each of the six ramps in Atlanta by hour of the day. The next task was to condense the data. We have two taxi time values for each departing flight (outbound ramp time and outbound movement time) and two taxi time values for each inbound flight (inbound ramp time and inbound movement time). The Atlanta airport has six ramps so an Atlanta taxi time operational report would require a total of 24 charts. Such a large number of graphs would require a lot of time to fully view and would make comparisons between hours and ramps difficult. Furthermore, as aforementioned, ramp and movement time are both components of an overall taxi time. The data needs to be assimilated to be useful.

A stacked vertical bar chart will allow us to display the average outbound ramp and outbound movement time for each hour of the day. When the data is stacked together we can also view the cumulative total departure taxi time.

By adding `SUBGROUP` to the `PROC GCHART` statement we can create a stacked bar chart. Our data set included a variable called "Area". This variable contained our two components of departure taxi time, "Depart Ramp" and "Depart Movement". By adding `SUBGROUP=Area` to the `PROC GCHART` we created a stacked bar chart with departure ramp and departure movement time by hour.

The below code and output displays a stacked bar chart for departure taxi time that is comprised of "depart ramp" and "depart movement".

```
PROC GCHART DATA=bar_and_line_2;
  VBAR Hour / SUMVAR=avg_mins SUBGROUP=Area CLIPREF FRAME DISCRETE TYPE=SUM
  LEGEND=LEGEND1 COUTLINE=BLACK RAXIS=AXIS1 MAXIS=AXIS2 GAXIS=AXIS3 GAXIS=AXIS3;
  BY Ramp_no;
RUN; QUIT;
```

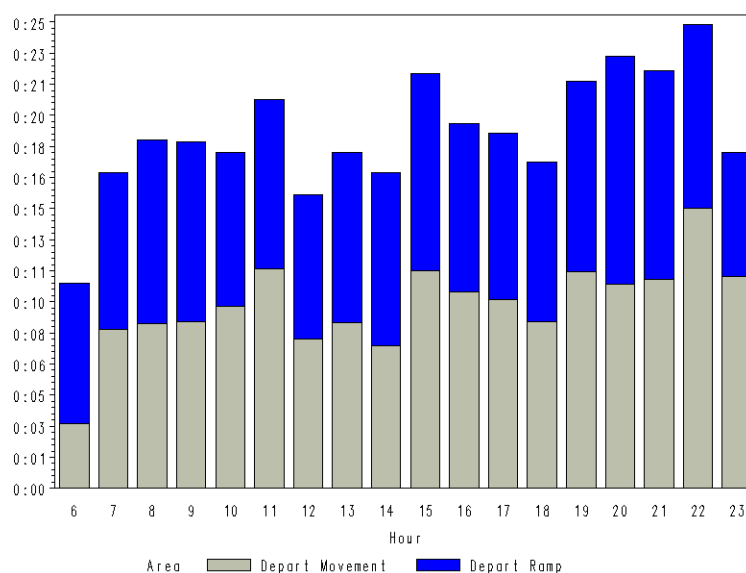


Figure 5. A stacked bar chart containing two stacks.

By combining the ramp and movement time into a stacked bar chart we have reduced our number of graphs from 24 down to 12. However, the report still contains a large number of charts and additional condensing of the data would be beneficial for quick analysis. Combining taxi-in and taxi-out data helps develop a better picture of total traffic saturation.

The next step was to use a stacked bar chart to display four sets of data in each stack. The stacked bar chart has two departure measurements (depart ramp, depart movement) and two arrival measurements (arrive ramp, arrive movement). The graph on the following page displays the results of our stacked bar chart with four taxi time components in each stack.

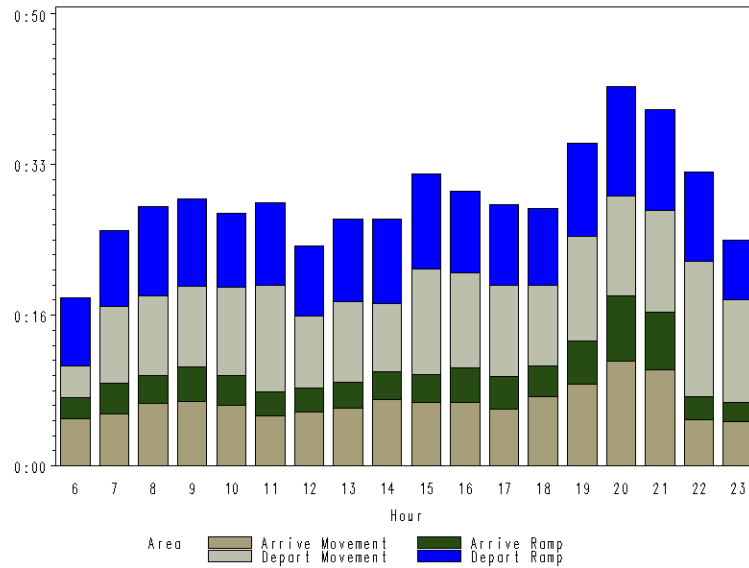


Figure 6. A stacked bar chart containing four stacks.

Our graphs now displayed departure and arrival taxi time information. The stacked bars allowed us to distinguish which hour had the longest total taxi time but it is difficult to separately analyze departure and arrival data since they are stacked on top of each other. It would be more beneficial if the inbound and outbound data were somehow separated, yet on the same graph. In order to create separation we decided to assign the departure taxi time measurements a positive value and the arrival metrics a negative value. This moved the x-axis that was previously at the bottom of the graph higher on the chart. This allowed us to easily compare the departure taxi time data to other departure data and arrival taxi time data to other arrival data. From the chart below, we now see that on the departure side the 22:00 hour had the longest taxi time and on the arrival side 20:00 had the longest taxi time.

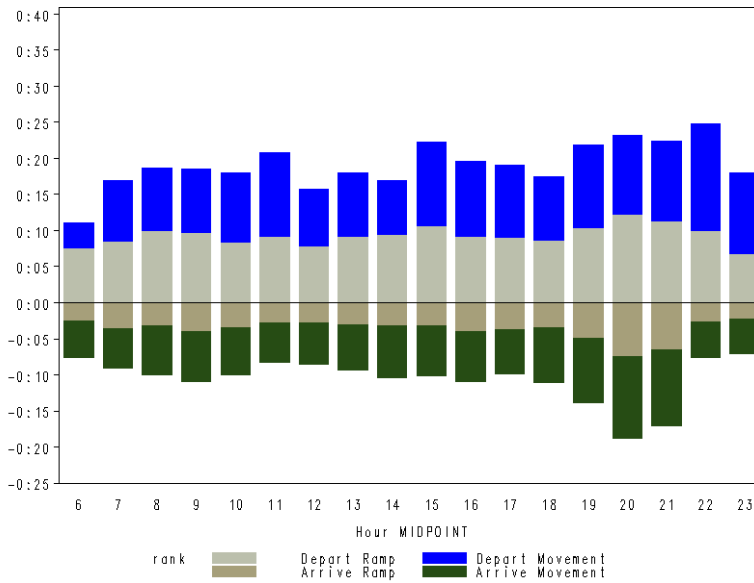


Figure 7. A stacked bar chart with an elevated x-axis.

ANNOTATE

We were happy with the progress we had made in producing our graphs. We had been able to condense the number of graphs down from 24 to 6 by using stacked bar charts. We showed the report to our Manager, and like all Managers she said “that is great, but can you add gate latency time?”² Obviously “no” was not the answer.

Gate latency time is a component of “depart ramp” time and we would like to see how gate latency varies as part of the departing ramp time so creating a fifth stack would not accurately display the data.

We decided that adding a marker inside the depart ramp stack would be the best approach to displaying gate latency. Using the point and click method in SAS Enterprise Guide an individual can easily create a bar-line chart which would allow us to overlay a marker on top of a bar chart. However, when using the point and click method the bar can have only one stack. Adding a marker to a stacked graph is a bit more involved and our solution was to annotate.

We turned to the Knowledge Base at Support.SAS.com and using “Sample 24865: Overlay a plot line on a vertical bar chart” we were able to create a maker showing gate latency inside of the depart ramp stack. Another valuable resource we found was David Mink and David J. Pasta’s SUGI 31 paper “Improving Your Graphics Using SAS/GRAPH® Annotate Facility”. The annotate facility requires a detailed explanation to fully understand and explaining annotation is not the goal of this paper. Our goal is to show how we used annotate to produce a marker inside a stacked bar chart. The results can be seen below.

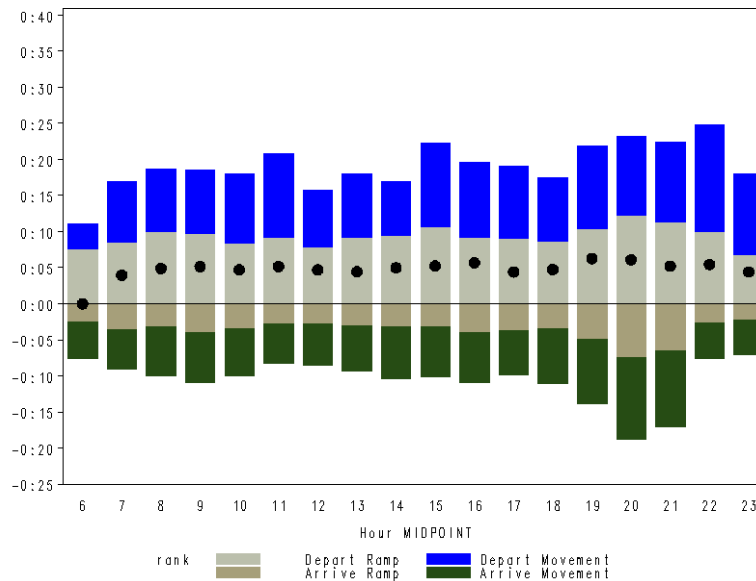


Figure 8. A stacked bar chart with annotations.

We now had a marker which displayed gate latency by hour of the day. The marker allowed us to understand how much of the depart ramp time is made up of gate latency. This information is very useful in determining which hours we need to focus our attention on. A high level of gate latency indicates an inefficient operation and wasted resources.

STANDARDIZING THE AXIS

By default SAS sets the minimum and maximum values for the y-axis. In our example the minimum value is -25 minutes and the maximum value is 40 minutes. For our chart we would like an equal range above and below the x-axis. Additionally, Ramp 1 may have a maximum value of 50 minutes while Ramp 2 may have a maximum value of 25 minutes. We believe that comparing max time across all six ramps is useful and to accurately accomplish the comparison the min and max values must be standardized for all six ramps.

² “Gate latency time” is time that an aircraft has the doors closed and is ready to push back from the gate but has not yet moved. This can be caused by traffic behind the aircraft or a ground crew that is not ready to push the aircraft. An increased gate latency time indicates an inefficient operation since the aircraft is ready to move but is sitting idle. Not only does latency have a direct operating cost, but it also has a negative effect on asset utilization.

The min and max values are defined in the `AXIS` statement. By using the `ORDER` option the programmer can define the min and max values as well as the frequency of the order display. The below code sets the y-axis min value to -30 minutes and the max value to 30 minutes and displays the minute values every 5 minutes.

```
AXIS2 ORDER=(' -00:30't to '00:30't by '00:05't);
```

Using a fixed value for the min and max values is acceptable if the data does not vary from day to day. However, while a typical day in Atlanta may have a max taxi time of 30 minutes, a high travel summer day with severe thunderstorms could push that value to over an hour. Allowing the min and max value to be dynamic would be much more useful. With a little programming, dynamic variables can be easily created. For our SAS program we used `PROC SQL` and the `MAX` function. The below code calculates the max time value for “avg_mins” for the entire data set using the `MAX` function. The new variable created is called “max_avg_mins”.

```
PROC SQL;
  CREATE TABLE dynamic_avg_mins_1 AS SELECT DISTINCT
    MAX(avg_mins) FORMAT TIME6.9 as max_avg_mins
  FROM complete_data
;QUIT;
```

Once the maximum average taxi time is found we added five minutes to create a buffer so that the maximum time was not stretched to the boarder of the graph. We then used `CALL SYMPUT` to create a macro variable named `max_mac` to pass to the `AXIS` statement. By calculating the maximum value and passing it to the `AXIS` statement we are able to make our report dynamic and reduce the manual effort needed to run the report. This is extremely useful if the report is ran daily or the user would like to automate the report without having to worry about axis ranges.

```
DATA dynamic_avg_mins_2;
  SET dynamic_avg_mins_1;

  max_avg_mins2=max_avg_mins1 + '00:05't;
  CALL SYMPUT('max_mac', (max_avg_mins2));
run;
```

The `AXIS` statement containing the dynamic value is below. The same value is used for above and below the x-axis. The negative sign in front of the macro value allows the macro variable to be negative.

```
AXIS2 ORDER=(-&max_mac. to &max_mac. by '00:05't);
```

CLEANING UP THE GRAPH

We now had a chart that contained our data in a stacked bar format with additional data displayed using a marker. We had also standardized the range of the y-axis and have made the maximum y-axis value dynamic. The one item that we still needed to clean up was the negative signs on the y-axis values below the x-axis. While we as the SAS user understood why the negative signs were present, the negative values would cause confusion for the end user. Remember, the negative values were used so that the arrival data could be displayed below the x-axis.

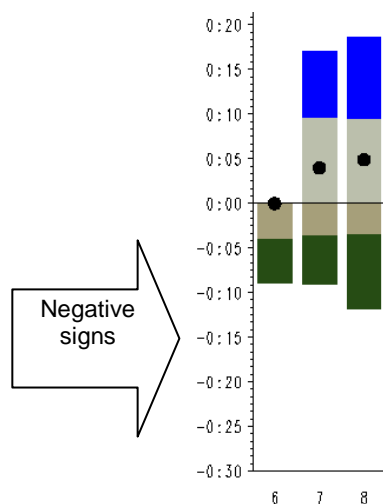


Figure 9. A bar char with the negative signs called out.

Our end report is outputted in a PDF format and we used `ODS LAYOUT` to position our graphs on the page. `ODS LAYOUT` is a very powerful function and there are several good reference papers regarding `ODS LAYOUT`.³

Our solution to remove the negative signs was to use `ODS LAYOUT` to produce a box overtop the negative signs in order to cover them. This approach worked very well, and after some trial and error of where to position the box and how large to make it, we were able to cover the negative signs. This allowed us to retain the negative values in the data set and display the values below the x-axis without causing confusion to the end user by having negative signs display on the charts.

THE COLOR WHEEL IS NOT YOUR FRIEND

If you are reading a color version of this paper will notice the four areas of the stacked bar chart are all a different colors. However, if you are reading a grayscale version of the paper the stacked areas will appear as different shades of gray. While the different shades of gray are distinguishable, matching the area of the stack to the legend can be difficult. Additionally, individuals who are colorblind will struggle with both color and gray scale versions of the paper.

Our solution to better distinguish the stacked areas was to use the `PATTERN` statement to add patterns to the areas of our stacked bar. Again we referenced the Knowledge Base at Support.SAS.com for information on this topic. By searching on “pattern statement” we were given a wealth of information. We decided to use patterns and the color “brownish gray” for our stacked bar areas. Additionally, Richard A. DeVenezia has put together a very good website which gives a list of SAS colors available to use. The address of the website is: <http://www.devenezia.com/docs/SAS/sas-colors.html>.

By experimenting with the different pattern settings we were able to easily distinguish each of the four stack areas. For the chart below, starting with the top stack and moving down, we used `PATTERN VALUE` of X1, R1, L5 and no `PATTERN VALUE` for the bottom stack. The code for setting the color to “brownish gray” and the pattern to an upper right diagonal line is:

```
PATTERN1 COLOR=BRGR VALUE=R1;
```

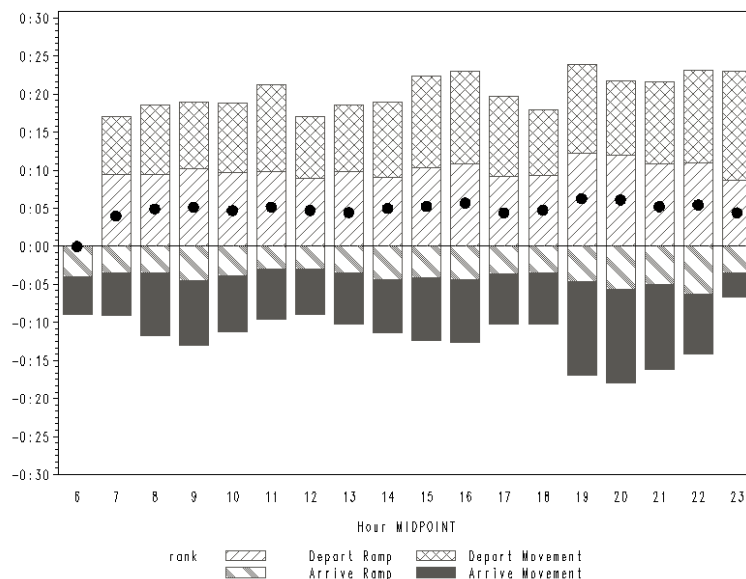


Figure 10. A stacked bar chart with patterns.

The stack areas can now be easily matched with the legend no matter if the chart is printed in color or gray scale. This allows the report to be more robust and we don't have to worry about the message of the report getting lost as the report is photocopied or passed from one end user to another.

³ We used Rich May's SUGI 31 paper "ODS LAYOUT is Like Onion" and Daniel O'Connor and Scott Huntley's SAS Global Forum 2010 paper "Breaking New Ground with SAS® 9.2 ODS Layout Enhancements".

ADDITIONAL TECHNICAL COMMENTS

During this project we found that graphs sometimes did not display as expected. We used SAS Enterprise Guide 4.1 and discovered that in order for the graphs to accurately displayed we had to adjust the "Graph Format" setting. Our default setting was "ActiveX" and this caused us frustration as the graphs did not always properly display. Once we changed the "Graph Format" to "GIF" or "JPEG" the graphs displayed as we anticipated. In our version of Enterprise Guide "Graph Format" can be found by selecting "Tools", "Options", "Results", and then "Graph".

CONCLUSION

This paper has attempted to show how we approached the process of creating an operational report. The report itself takes a new set of data elements and presents them in a unique and meaningful way. The development process involved a great deal of trial and error and there is always room for improvements. We believe that our final report incorporated a number of different SAS functions and is a good example of how SAS allows the programmer to create a highly customized and beneficial report.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

Andy Hummel
Delta Air Lines
General Manager – Pilot Crew Resources
Andrew.Hummel@Delta.com
404-715-1270

Robert Goldman
Delta Air Lines
Technical Pilot – Flight Operations
Robert.S.Goldman@Delta.com
404-368-8242

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.