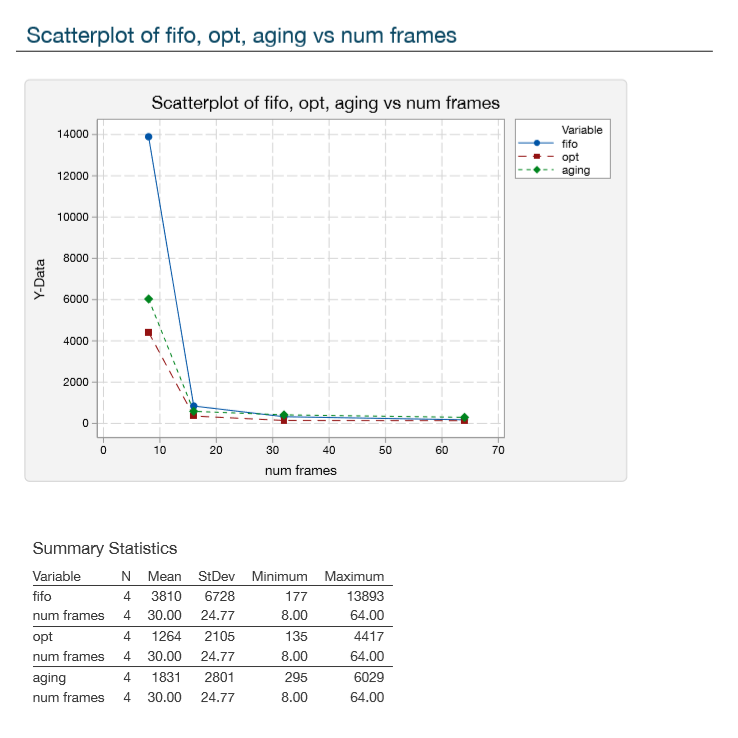
**1550 – Project 3 Write Up** Nicole Dash

1. In order to determine what refresh period works well I graphed the page faults vs refresh as shown above. I graphed both the results when I execute gcc.trace and swim.trace. As shown from the graph both drop off towards the beginning but level off at different times. This is one of the indicators that there is not a perfect refresh period as it may be better to have a lower number for gcc.trace but a higher number for swim.trace. The number that I chose as the ideal refresh rate is 88. I chose this number based on the page faults when I ran swim.trace. 88 refresh cycles give the lowest number of page faults and is also in the range where the variability in page faults decreases. I also ran the algorithm with 8-page frames. This number would change if I use a different number of page frames, thus why with aging there is no perfect refresh period.



1. For each of the three algorithm I compared the number of page faults with different page frames entered. I ran all three with the swim.trace file. The comparison is shown on the graph above. I compared the numbers at 8, 16, 32, and 64-page frames. I ran the aging algorithm with the refresh rate of 88 that I determined above. Opt as shown from the graph (and its name) is the most optimal algorithm and reduces the number of page faults at each number of frames. However optimal requires a pre-trace and is impossible to implement in an operating system because it requires future knowledge. The next best is the aging algorithm with a refresh of 88 ran with swim.trace you can see that the number of page faults is very similar to the number of page faults for opt and thus would be the most appropriate for an actual operating system.
2. Belady’s Anomaly is when increasing the number of page frames increases the number of page faults. This occurs in the FIFO page replacement algorithm. In this implementation this occurs when the number of page frames is increased from 91 to 92. I found this by running fifo on gcc.trace. In gcc.trace when I run the program with 91 frames the result is 457 page faults, however when fifo is run on gcc.trace with 92 and 93 frames the resulting page faults is 458. Although very small increase this is an increase in the number of page faults as the number of frames increases. At 94 frames it decreases again to 453-page faults. Thus, Belady’s Anomaly occurs at 92-page frames.