

STSCI 5010 Homework 4

Pete Rigas (pbr43 cornell.edu)

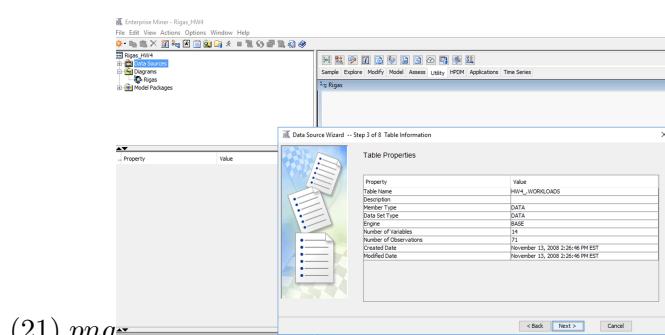
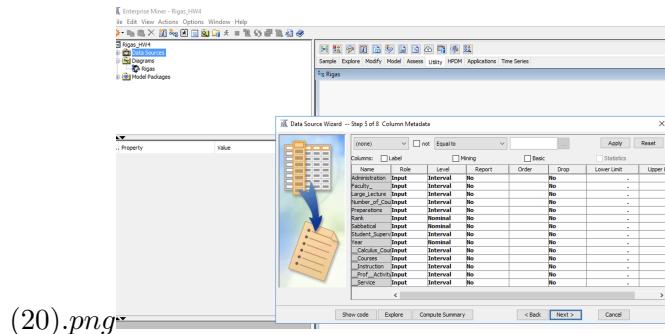
November 27, 2020

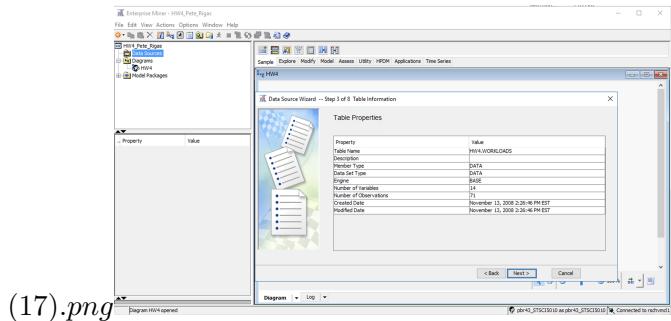
Note: Before taking these screenshots, I forgot that we had to name the diagram by last name. I have corrected this in my final code that I ran and the proc KDE procedures that are shown below, and to also change the name of the project with my last name by creating a new one. My mistake, sorry. Also, I just took screenshots and put them into Latex, which I hope is fine because the output is a PDF.

1 Setup

1.1 Creating the SAS Enterprise Miner Project, loading in the data from Canvas as a data source

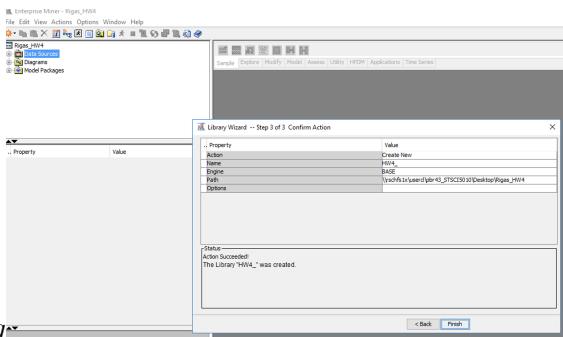
From the given data set workloads, we can assign it a data source as I have done so below, to the data sources folder in my HW 4 library. We do not find it necessary to edit any of the values from the dataset once we have imported it to the datasources section of the HW 4 library.





(17).png

After correctly defining the HW 4 library, as shown below, we can then proceed to creating the diagrams and the node in each diagram that we will use to output each of the plots with different BWM values.

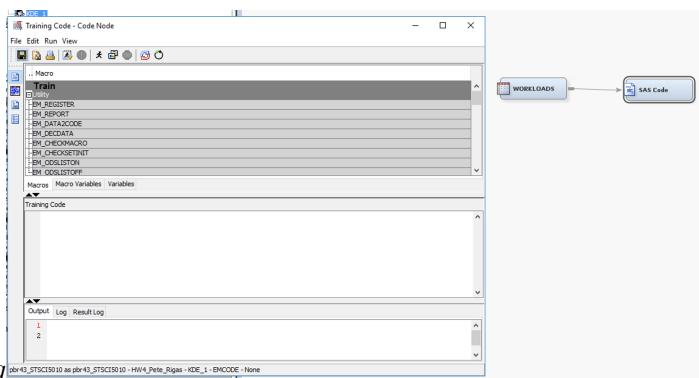


(22).png

1.2 Creating Diagrams for each of the proc KDE calls

We will create appropriate diagrams under the diagram "section" of the HW 4 library. As mentioned in the next section, once we have the white space in which the code is ready to compile, we will add a single node for each of the diagrams, for specified BWM values, that we will display. Again observe that we must create a node for each of the BWM values that we are using a proc KDE procedure for, in addition to the separate chunks of SAS code that are run for each of the diagrams for different BWM values.

After establishing the correct directionality between the workloads and SAS code from which we will run the proc KDE procedures, each of which are shown in the screenshots below, we input our SAS code for each of the proc KDE calls, each with their respective out statement.



(19).png

1.3 Format of code for the proc KDE diagrams (Part C Answer)

For each of the given diagrams and the specified bwm value, we can make use of the PROC kde code that is given in *Slide 67* of the first lecture for Module 2, in which we must specify the lower and upper values of the grid, in addition to the bwm value that is given to use for each of the plots that we must compile.

```

proc kde data = Rigas.workloads;
univar __service/gridl=0 gridu=100 bwm=2 out = Rigas.kde_2;
run;

```

- Observe that we output the name of the diagram with the specified BWM value, into the diagram that we have named. In my case, it is Rigas, that is correctly output into the diagram for each BWM value.
- service has 2 underscores in each proc KDE call
- the code above satisfies **C** because we can just change the bwm value to fit all other plots that we display in the next section
- setting the lowe and upper grid limits in the proc KDE call to 0 and 100, respectively, are appropriate axes for displaying all relevant information

1.4 Putting each of the proc KDE calls into the training code window

From previous steps, we make sure that each of the appropriate proc KDE calls, with the specified BWM values that we display from the plots in the **next** section, are appropriately shown.

```

proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=.5 out = SASUSER.kde_H;
run;

proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=1 out = SASUSER.kde_L;
run;

proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=2 out = SASUSER.kde_Z;
run;

proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=4 out = SASUSER.kde_4;
run;

proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=8 out = SASUSER.kde_8;
run;

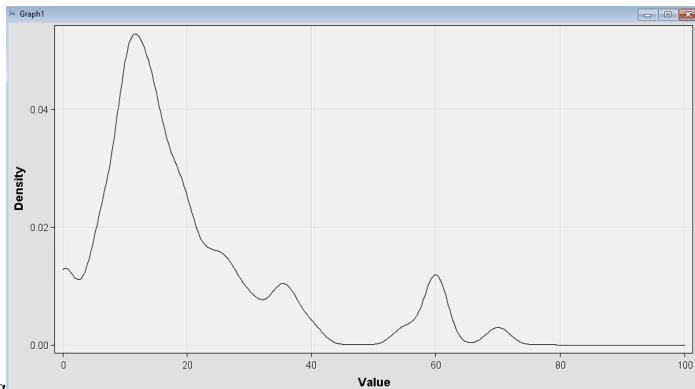
proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=10 out = SASUSER.kde_overlay;
by year;
run;

```

(31).png

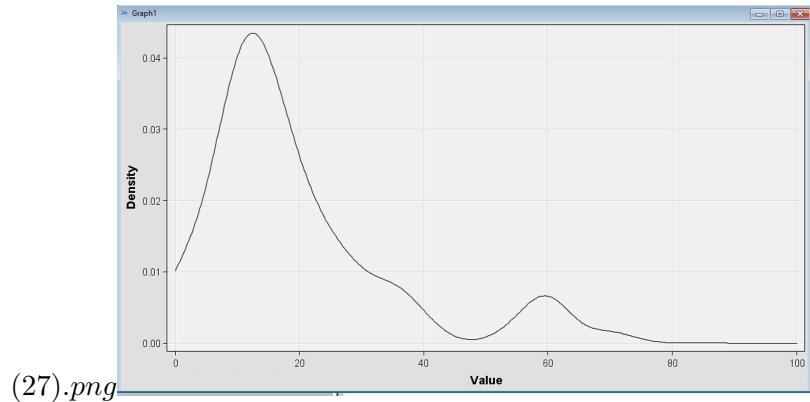
2 Diagram Output (Part A Answer)

2.1 bwm = 0.5

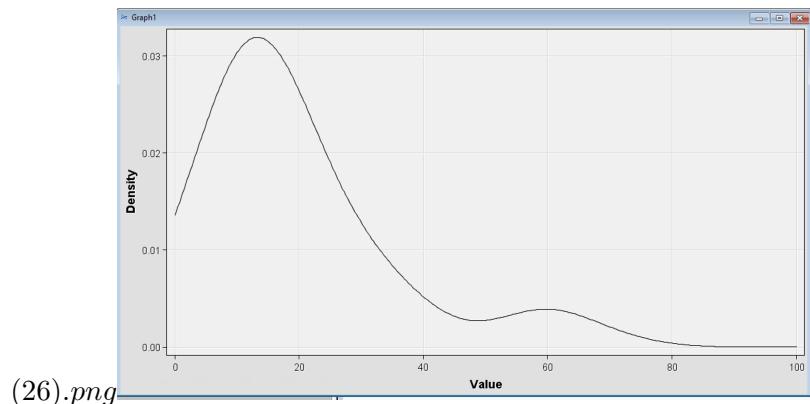


(28).png

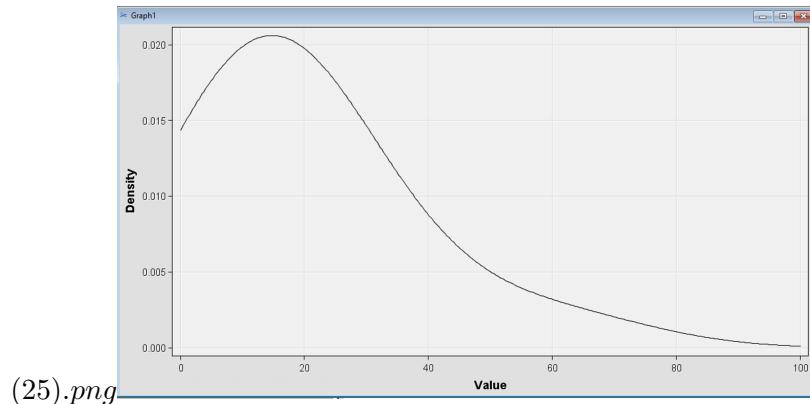
2.2 $bwm = 1$



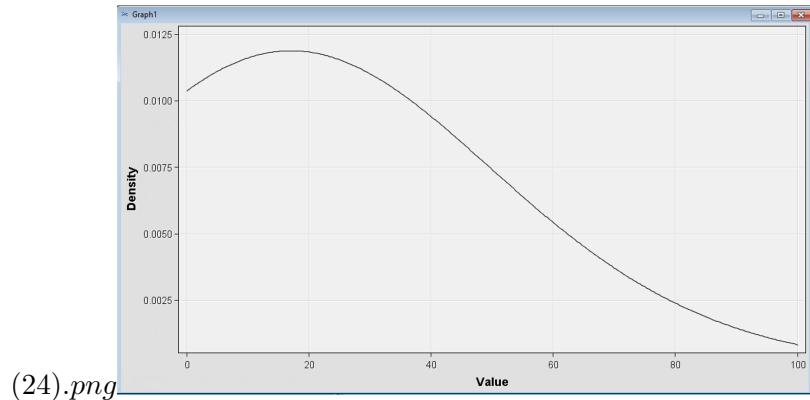
2.3 $bwm = 2$



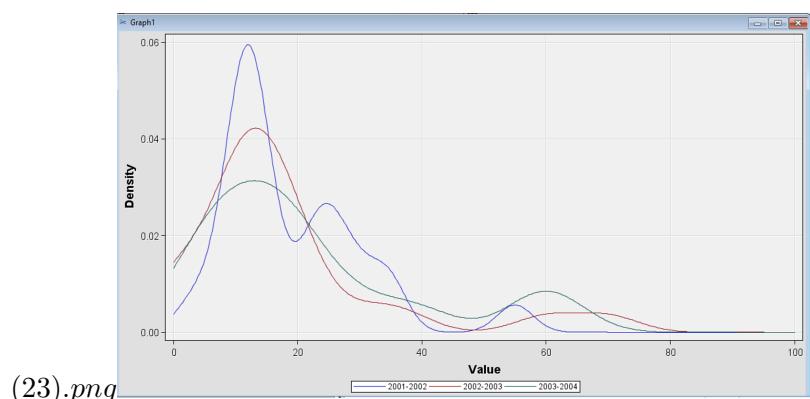
2.4 $bwm = 4$



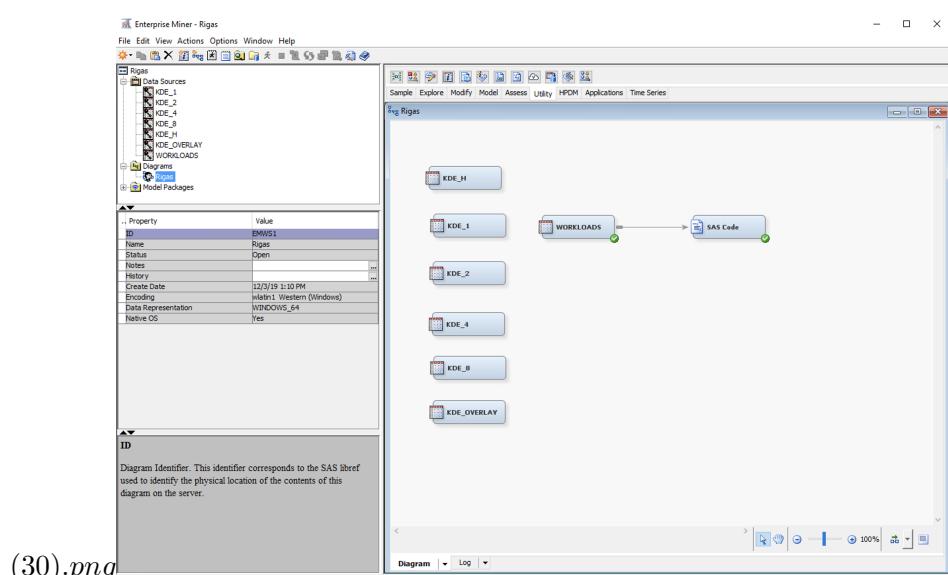
2.5 $bwm = 8$



3 Overlay of all the KDE graphs for all BWM values (Part B Answer)



4 Screenshot of all the proc KDE, Workloads, and SAS Code diagrams (Part D Answer)



5 Enterprise Miner Code

```
proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=0.5 out = SASUSER.kde_H;
  run;
proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=1 out = SASUSER.kde_1;
  run;
proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=2 out = SASUSER.kde_2;
  run;
proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=4 out = SASUSER.kde_4;
  run;
proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=8 out = SASUSER.kde_8;
  run;
proc kde data = RIGAS.WORKLOADS;
  univar __service/gridl=0 gridu=100 bwm=1 out = SASUSER.kde_overlay;
by year;
run;
```