SAS University Edition Challenge

Day 3 of 5 – Wednesday, 17th July 2018

Challenge Overview

You and your friend (let's call him) Kenny, have decided to start an online website that aggregates movie and TV show data, streamed by Netflix.

Kenny finished up with the 2017 releases data and sent it over to you today. You need to import this file to run a quick analysis on the data just like the other data. You notice that the delimiter in this file is different to the 2016 data. Kenny asks if you can merge the data sets together so the data can be uploaded to the website in one file.

Kenny also wants you to analyse the user rating scores of movies over the years to see which year has produced the best movies.

Let's get on it!

Summary of Skills Demonstrated

- Importing a delimited file
- Joining data sets
- Filtering data sets
- Understanding variable types
- Graphing with bar chart

Submission Details

Submissions close **Monday, 23rd July 2018, 12:00pm**.
Submit your answers here.

Be sure to use your SAS account when you submit your solution. If you don't use your SAS account, we can't put your entry in the draw!

If you haven't registered yet, it's not too late. Click here to register now.

Also, make sure to save your answers and tasks/code somewhere safe as a backup.

SAS University Edition

You'll need to download and install SAS University Edition to complete this challenge. Use the links below to install and set up SAS University Edition.

SAS University Edition download link:
https://www.sas.com/en_au/software/university-edition/download-software.html
How do I create a shared folder in VirtualBox?
https://support.sas.com/software/products/university-edition/faq/shared_folder_virtualbox.htm

Guided Exercises

Before we start

Make sure you've downloaded the resources for today's challenge from here, placed the data in your shared folder repository and launched SAS University Edition.

For this guided exercise, you will need to write some SAS code. Don't worry though, we'll guide you through it. It's really straight-forward.

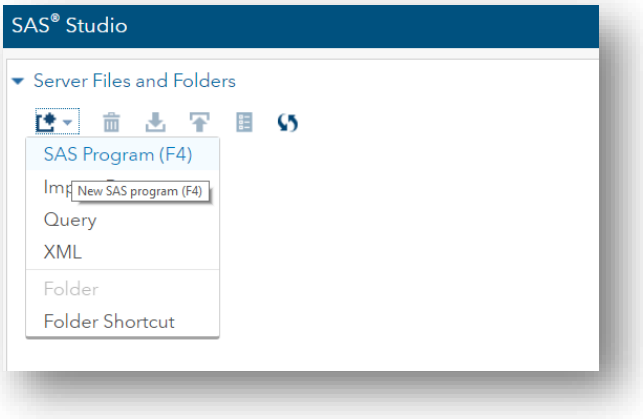
Joining Data Sets

To simulate how to join data sets, we are going to duplicate a data set and append the duplicates into one data set.

1. Expand **Libraries** in the left pane.
2. Expand the **SASHELP** library. This is an in-built SAS library that comes with each version of SAS University Edition.
3. Double-click the **CARS** data set to open it. You will notice that there are 428 rows or records.

In SAS, rows or records are referred to as **observations** and columns are referred to as **variables**.

4. Expand **Server Files and Folders** in the left pane and create a new program in **My Folders**.

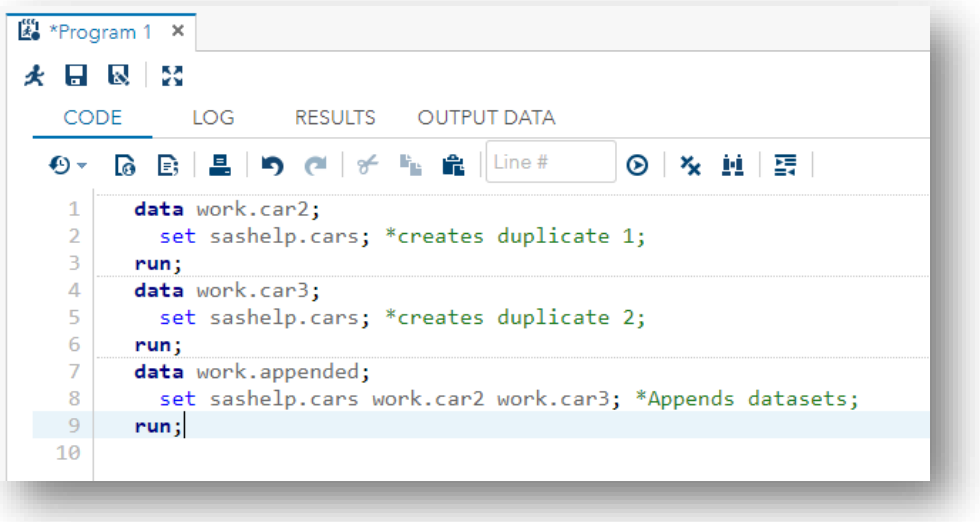


5. In the **SAS program editor**, copy and paste the following statements:

```
data work.cars2;
      set sashelp.cars; *creates duplicate data set 1;
run;

data work.cars3;
      set sashelp.cars; *creates duplicate data set 2;
run;

data work.appended;
      set sashelp.cars work.cars2 work.cars3; *Appends data
      sets;
run;
```

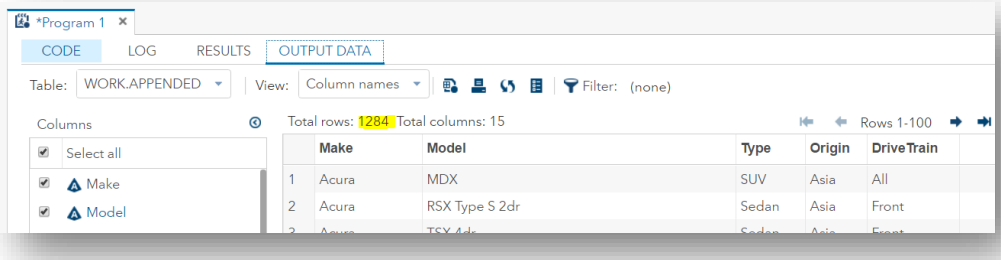


A SAS program is simply a sequence of steps. There are two kinds of steps in SAS; **DATA** steps and **PROC** (procedure) steps. DATA steps are used to build data sets. The code above is creating a new data set from an existing data set.

Click here to learn more about DATA step processing and/or check out the free SAS Programming 1 course.

6. Click **Run**
7. Navigate to **Libraries** in the left pane and expand the **WORK** library. A data set named **APPENDED** should appear.

8. Double-click on the **APPENDED** data set to open it. Remember that our **CARS** data set originally had 428 observations. Since we have joined 3 copies of the data set together, the number of rows will triple. The **Total rows** in the **APPENDED** data set is 428 × 3 = 1284. This means that the data sets have successfully been appended.

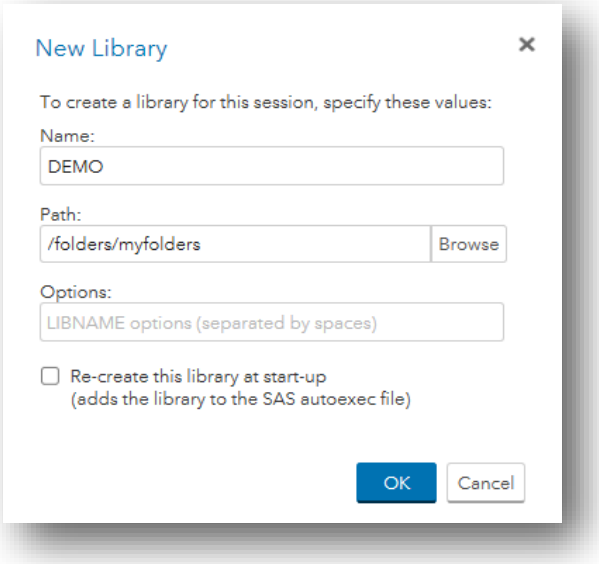


Understanding variable types

When dealing with data, it is important to understand the what is inside the data you are working with.

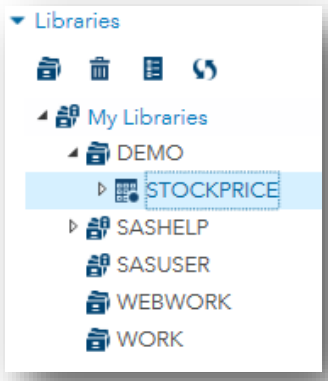
In this section we will work with a data set that contains a stock's price over a period of time. Let's run some tasks to see what variable types the data contains and if they are suitable for use. If not, we can write code to change the variable type!

1. Expand **Libraries** in the left pane.
2. Right click on **My Libraries** and select **New Library**.
3. Name the library **DEMO**.



4. Click **OK**.

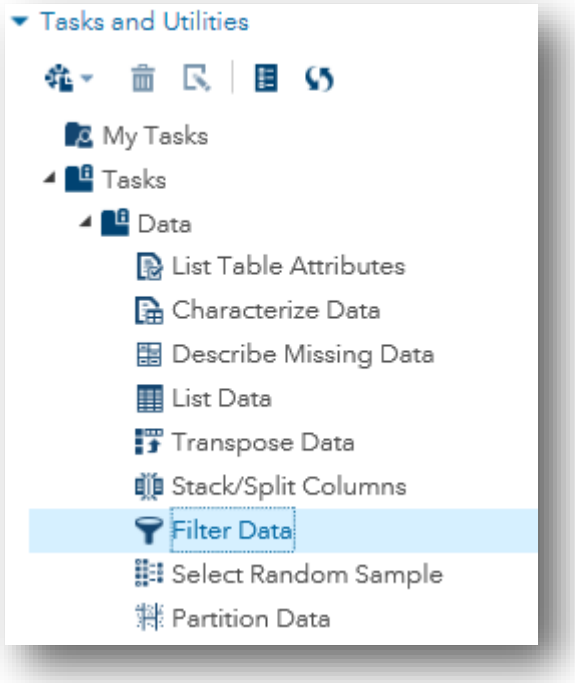
5. The **DEMO** library should automatically contain the **STOCKPRICE** data set you copied into your **myfolders** folder. If not, you can store the **STOCKPRICE** data set into the **DEMO** library by dragging and dropping.



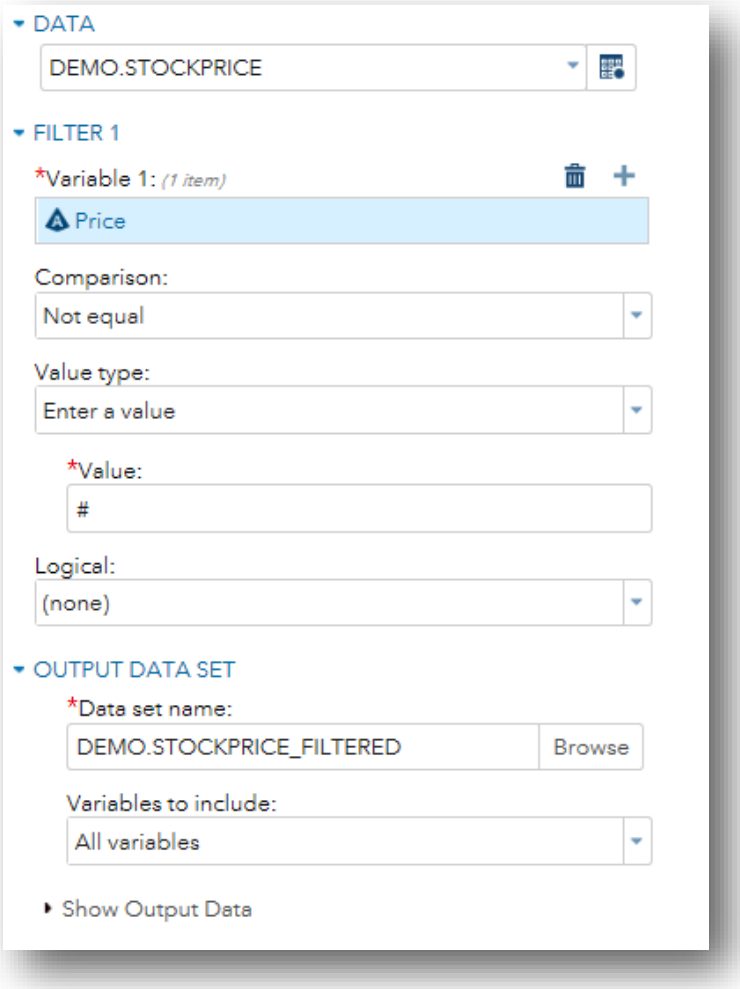
6. Double click on **STOCKPRICE** to open the data set.
7. Notice in the Price column, there are some missing prices (# instead of a number). There should be a total of 250 observations.

   Let's clean the data by removing the missing prices.

8. On the left pane, navigate to: **Tasks and Utilities** > **Tasks** > **Data** > **Filter Data**.



9. Double click on **Filter Data** to open the new filter task.
10. Select STOCKPRICE in the DEMO library as the data source.
11. Click the '**+**' for Variable 1 and select **Price**.
12. In **Comparison** select **Not equal**.
13. In Value type, select **Enter a value**.
14. In Value, type in **#**.
15. Output this filtered data set to the **DEMO** library and name it **STOCKPRICE_FILTERED**.

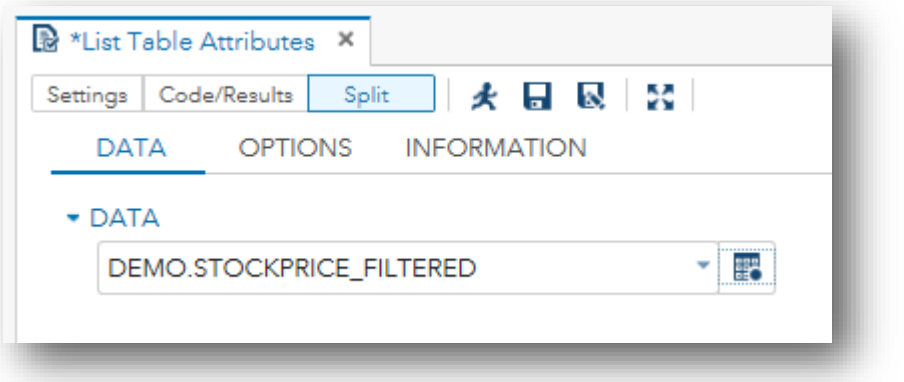


16. Click **Run**
17. Look at the output data set to confirm the '**#**' have been removed.
    There should now be a total of 233 observations.

Now that our data set is clean, let's have a look at the data types our data set contains to see if they are suitable for analysis later on.

18. In the left pane navigate to: **Tasks and Utilities** > **Tasks** > **Data** > **List Table Attributes**.
19. Double click on **List Table Attributes** to open the new task.

20. Choose the **STOCKPRICE_FILTERED** data set from **DEMO** library as the data source.



21. Click **Run**
22. Examine the output.

Data Set Name	DEMO.STOCKPRICE_FILTERED	Observations	233
Member Type	DATA	Variables	3
Engine	V9	Indexes	0
Created	06/27/2018 11:11:32	Observation Length	24
Last Modified	06/27/2018 11:11:32	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat
1	Date	Num	8	DATE.	DATE.
2	Price	Char	5		
3	Year	Num	8		

The **List Table Attributes** task generates summary information about the contents of a dataset. Referencing the above output:

- Top table – contains information such as the number of observations and variables in the data set and when the dataset was created.
- Bottom table – contains variable information such as the variables' names, types, and length.

In SAS, variable types are either **Numeric** or **Character**. Numeric values represent numbers. Character values can contain letters, numbers, and special characters.

Click here to learn more variables in SAS and/or check out the free SAS Programming 1 course.

23. Examine the list of variables and attributes.

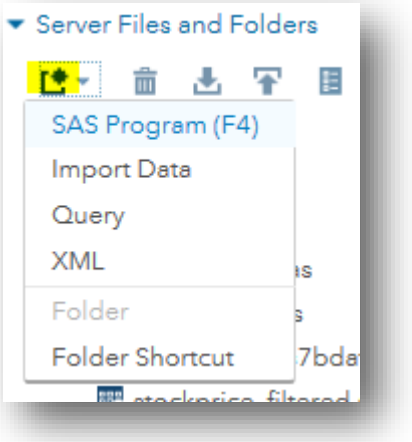
Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
1	Date	Num	8	DATE.	DATE.
2	Price	Char	5		
3	Year	Num	8		

Note that **Price** is a **character** type variable although it contains values that are numbers. This is because in the original data set, the **Price** variable contained '**NA**' values which are characters.

We want to analyse the price over time so we must change **Price** to a **numeric** type variable.

Let's do this manually with code.

24. Create a new SAS Program. Expand the **Server Files and Folders** section in the left pane, click on the icon shown below and choose **SAS Program**. Another way is to press **F4**.



25. Copy and paste the following code into the SAS program:

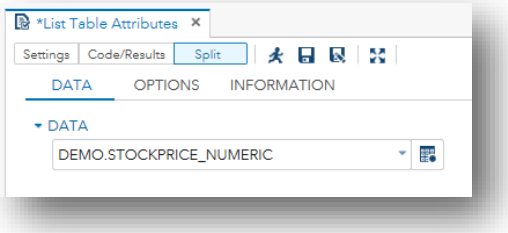
```
data demo.stockprice_numeric;
      set demo.stockprice_filtered;
      Price_Numeric=input(Price,8.);
      drop Price;
      rename Price_Numeric=Price;
run;
```

What is the code doing? Here is a brief explanation on how it works:

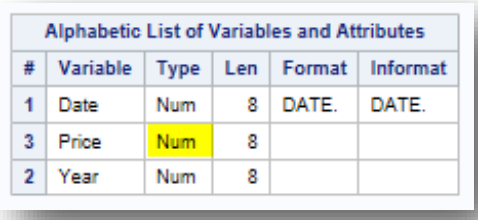
1. Line 1: The **DATA** step creates a new data set called **STOCKPRICE_NUMERIC** in the **DEMO** library.
2. Line 2: The **SET** statement sets the data source as **STOCKPRICE_FILTERED** from the **DEMO** library.
3. Line 3: To change a variable type from character to numeric, the INPUT function is used. The code **input(Price,8.)** reads in the **Price** variable and converts the value to a numeric value, giving it a length of 8. This value is then assigned a new variable called **Price_Numeric**.
4. Line 4: The DROP statement removes the old character variable **Price** from the output data set.
5. Line 5: The RENAME statement changes the name of the **Price_Numeric** variable to **Price**.

To learn more about SAS programming, check out the free SAS Programming 1 course.

26. Click **Run**
27. Verify that **STOCKPRICE_NUMERIC** data set has been created in the **DEMO** library by expanding the **Libraries** section in the left pane.
28. Create and run another **List Table Attributes** task for the new data set **STOCKPRICE_NUMERIC**.



29. Examine the output and note that the type for the **Price** variable is now **numeric**.



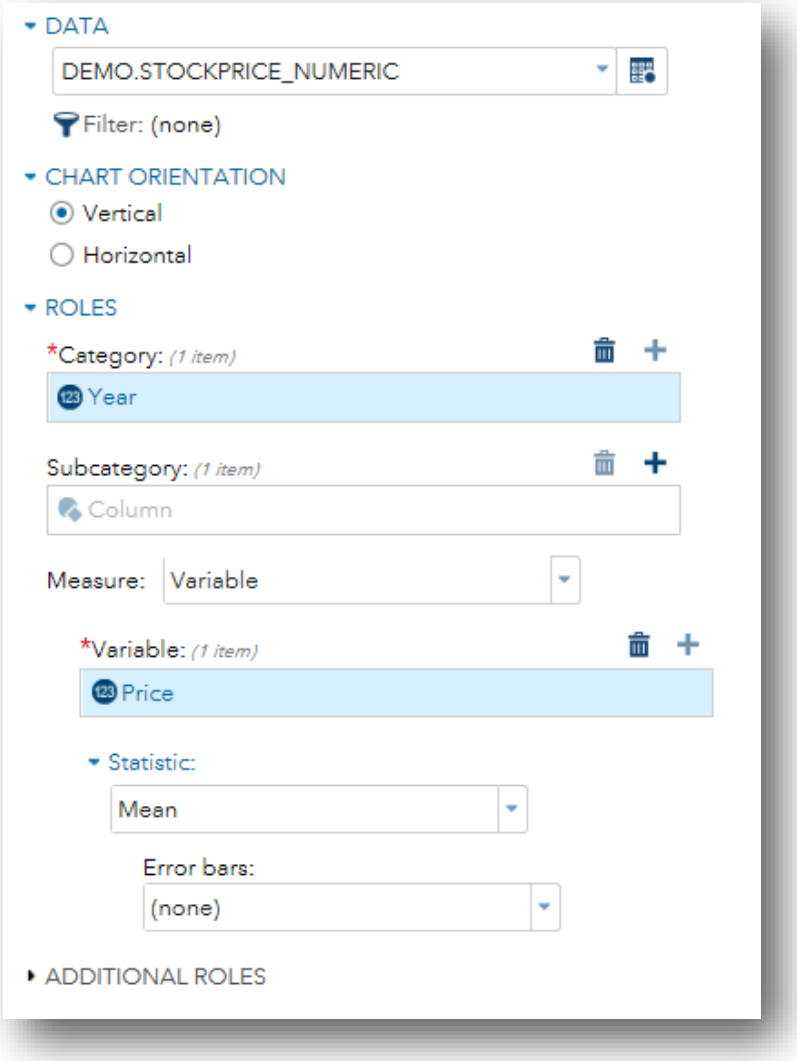
Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat
1	Date	Num	8	DATE.	DATE.
3	Price	Num	8		
2	Year	Num	8		

Graphing with Bar Chart

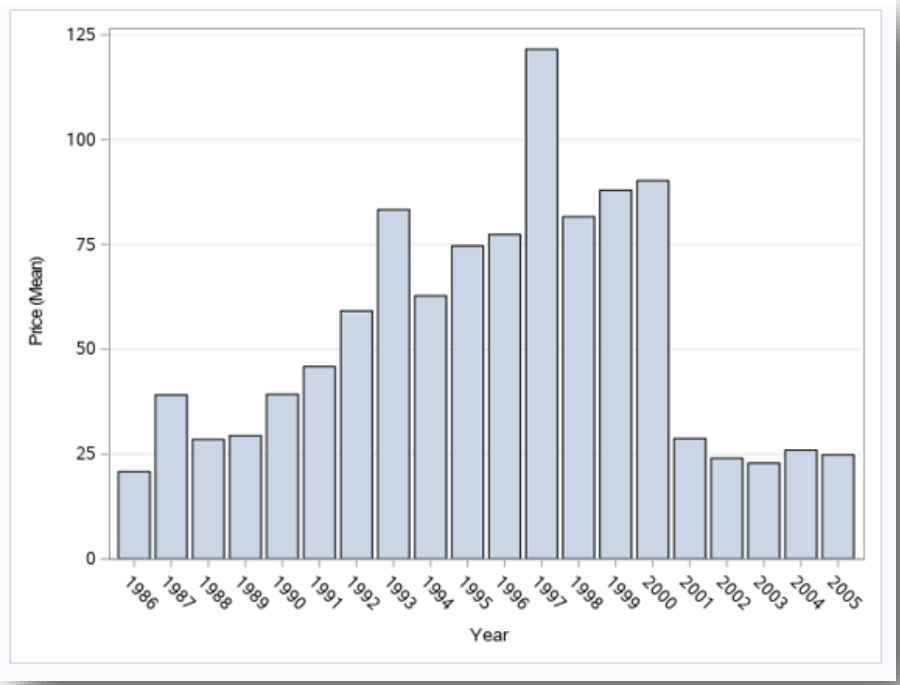
Now that we have cleaned the stocks data set, we can graph the data and analyse how the stock price has trended over the years.

1. Navigate in the left pane to: **Tasks and Utilities** > **Tasks** > **Graph** > **Bar Chart**.
2. Double click on **Bar Chart** to create a new bar chart task.
3. Select STOCKPRICE_NUMERIC from the DEMO library as the data source.
4. Click the '**+**' for Category and select **Year**.
5. In Measure, select **Variable** from the drop-down bar.
6. Click the '**+**' for Variable and select **Price**.
7. In Statistic, select **Mean** from the drop-down bar.



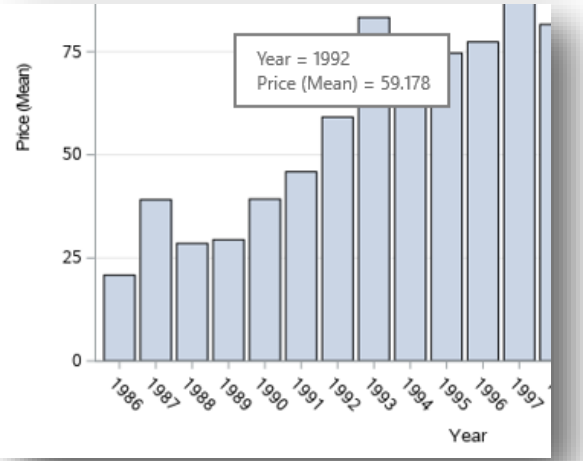
8. Click **Run**

9. Examine the output bar chart.



This bar chart shows the average price of the stock per year over the years 1986-2005.

10. Hover over the bars to see the values.



End of Guided Exercises

Challenge Exercises

Submit your answers here. Answer all 4 questions correctly to gain an entry into the prize draw.

This challenge requires information from the previous challenges. Be sure to complete the other challenges before completing this one.

Make sure you've downloaded the resources for today's challenge from here and placed it in **C:/SASUniversityEdition/myfolders**, the folder you created earlier.

It's a good idea to save your tasks in an accessible location once complete. If anything happens to your data, you can always rerun the tasks.

Import the **Netflix_2017.txt** file. Save the data set as **NETFLIX_2017** in your **NETFLIX** library.

Import the **Netflix_2016.csv** file. Save the data set **NETFLIX_2016** in your **NETFLIX** library.

Append the following data sets into one output data set: **NETFLIX_1940_2015**, **NETFLIX_2016** and **NETFLIX_2017**. Save the output data set as **NETFLIX_APPENDED** in the **NETFLIX** library.

Question 1: *How many observations are in the NETFLIX_APPENDED data set?*

Filter the **NETFLIX_APPENDED** data set to show the titles that **do not** have "**NA**" as a **user_rating_score**. Output the resulting data set as **NETFLIX_APPENDED_FILTERED** and store it in the **NETFLIX** library.

Question 2: *How many titles in NETFLIX_APPENDED do not have a user_rating_score of "NA"?*

In the **NETFLIX_APPENDED_FILTERED** data set, the user_rating_score is a character variable as it previously included the value "**NA**". It remains a character variable even after we remove "**NA**". Write code in a SAS Program to create a new data set **NETFLIX_APPENDED_NUMERIC** in the **NETFLIX** library that changes the user_rating_score to a numeric variable so that it can be analysed.

Question 3: *How many numeric variables are in the NETFLIX_APPENDED_NUMERIC data set?*

Using a bar chart with the **NETFLIX_APPENDED_NUMERIC** data set as the data source, analyse how the average user_rating_score for movies fluctuates over the years.

Question 4: *What is the mean of the year with the highest user_rating_score mean?*

Don't forget to visit our Facebook page tomorrow for the next challenge!

https://www.facebook.com/SASAustNZ