# BSTA 477/677 – Winter 2021

Tutorial 1 - Jan 31st, 2021



Data used: Bike sharing  Washington D.C. dataset
Software used: SAS EG

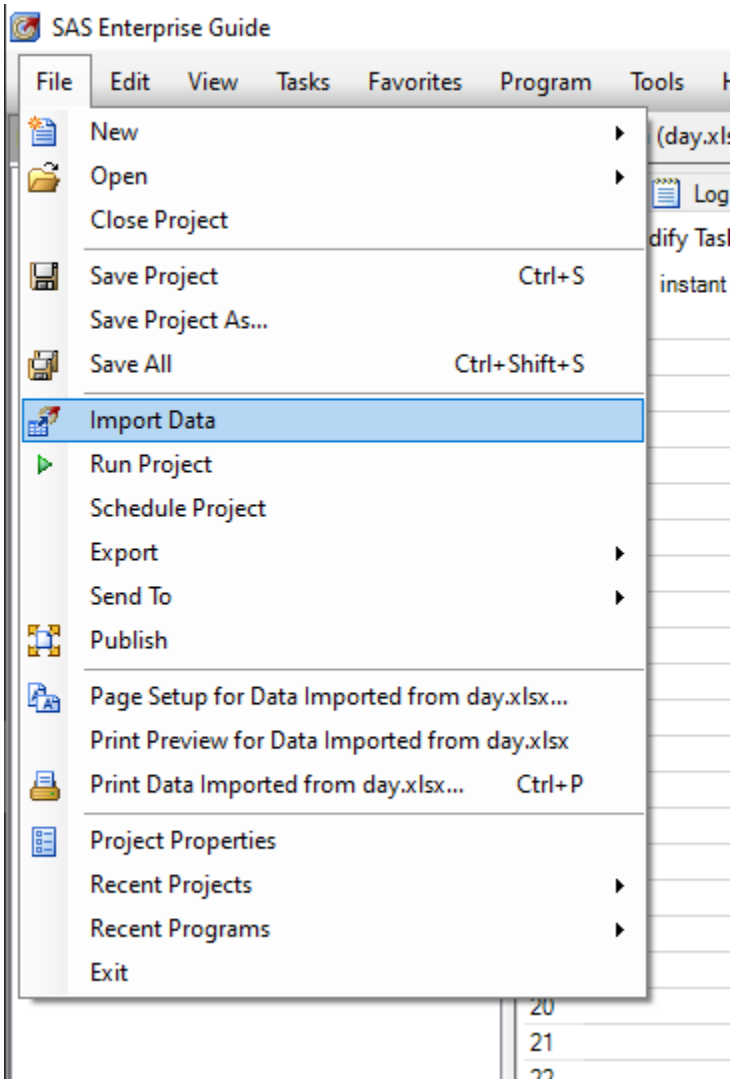
# SAS Enterprise Guide - Basic

- Process flow
- Task bar
- Library bar - Data storage
- Note/Warning/Error bar

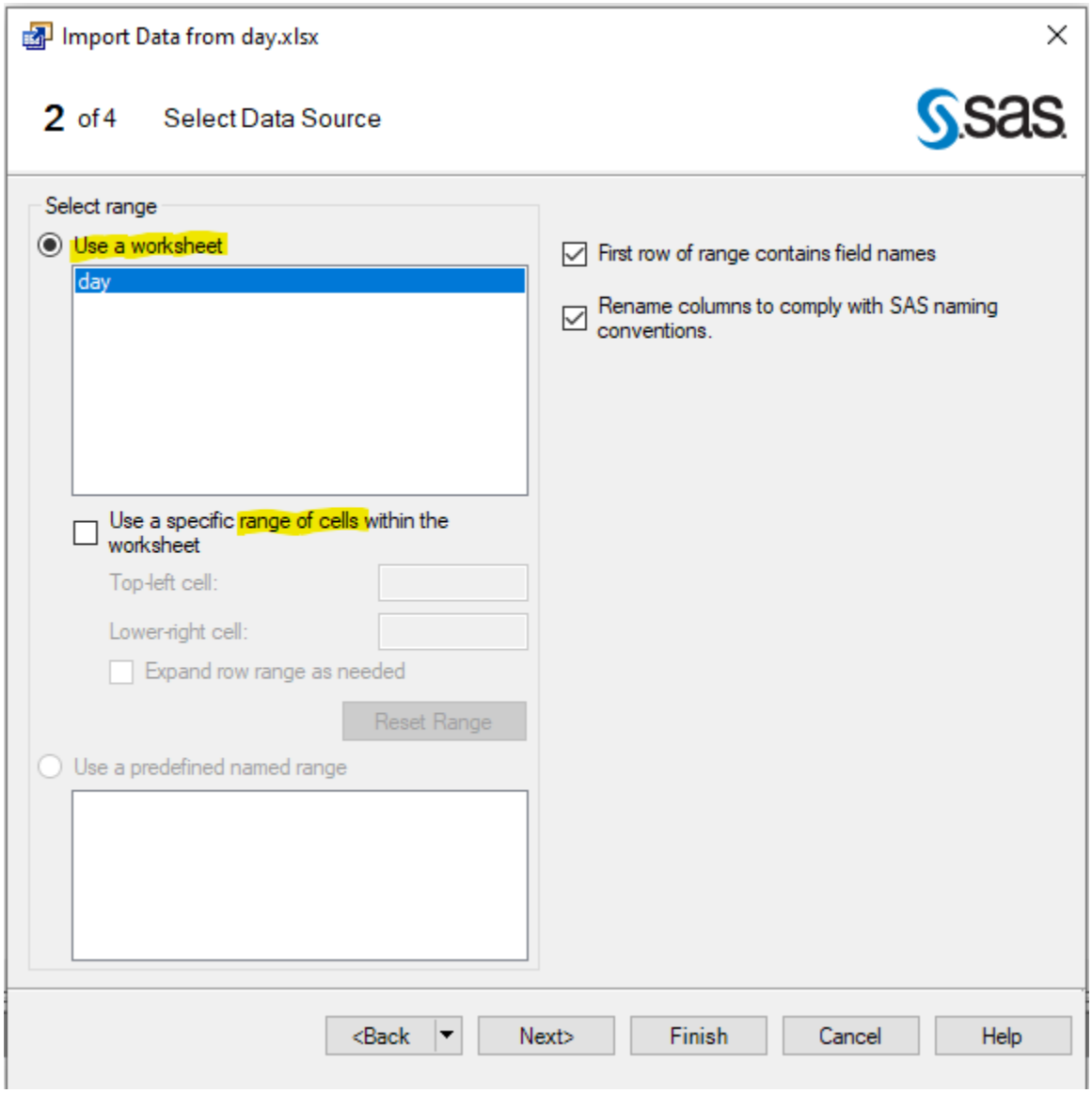
# Basic data manipulation

## Data import

File > Import Data > Open > (optional) Browse SAS output data set location > (optional) Adjust variable source format



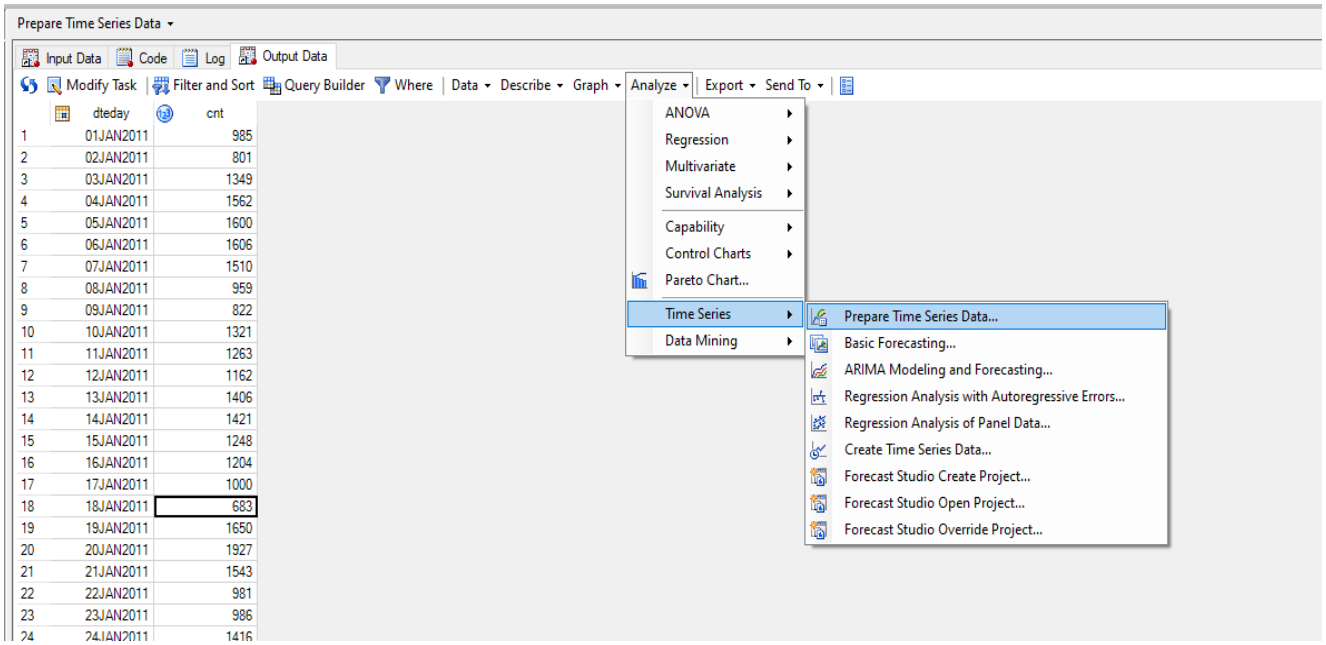
Note: For importing excel files, there are options to select the sheet to import and/or select a range of cells for the sheet.



# Data preparation

Simple aggregation of time series data:

Output data tab > Analyze > Time series > Prepare Time Series data > (optional) Adjust Existing Time ID variable interval > (optional) Interpolation method: Simple aggregation



# Data summary

- Code: Proc content

```
proc contents data=bsta477.bike_sharing_day_data;
run;
```

Information given:

*The CONTENTS Procedure*

| | | | |
|---|---|---|---|
| Data Set Name | BSTA477.BIKE_SHARING_DAY_DATA | Observations | 731 |
| Member Type | DATA | Variables | 16 |
| Engine | V9 | Indexes | 0 |
| Created | 01/29/2021 16:47:02 | Observation Length | 128 |
| Last Modified | 01/29/2021 16:47:02 | Deleted Observations | 0 |
| Protection | | Compressed | NO |
| Data Set Type | | Sorted | NO |
| Label | | | |
| Data Representation | WINDOWS_64 | | |
| Encoding | wlatin1 Western (Windows) | | |

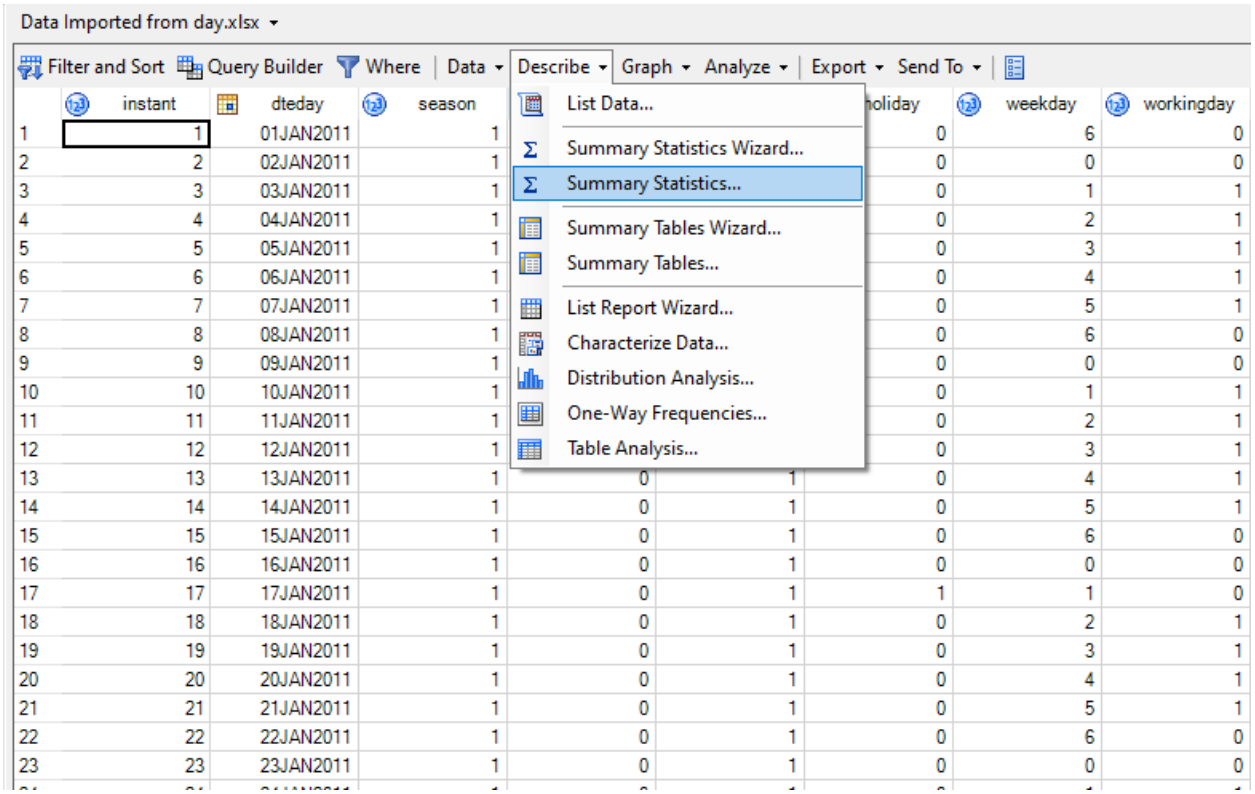
| Engine/Host Dependent Information | |
|---|---|
| Data Set Page Size | 65536 |
| Number of Data Set Pages | 2 |
| First Data Page | 1 |
| Max Obs per Page | 511 |
| Obs in First Data Page | 490 |
| Number of Data Set Repairs | 0 |
| ExtendObsCounter | YES |
| Filename | C:\Users\mathd\Documents\BSTA477\bike_sharing_day_data.sas7bdat |
| Release Created | 9.0401M4 |
| Host Created | X64_10PRO |
| Owner Name | MATHIEU-PC\mdugre |
| File Size | 192KB |
| File Size (bytes) | 196608 |

Alphabetic List of Variables and Attributes

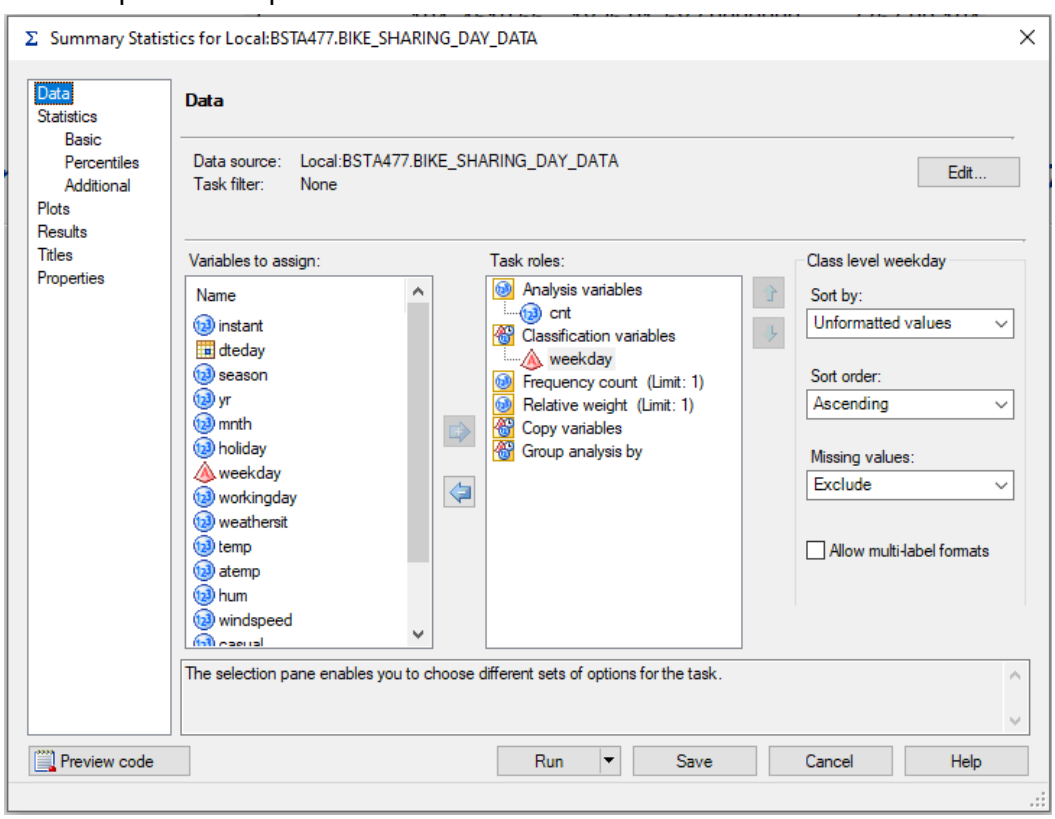
| # | Variable | Type | Len | Format | Informat |
|---|---|---|---|---|---|
| 11 | atemp | Num | 8 | BEST12. | BEST12. |
| 14 | casual | Num | 8 | BEST12. | BEST12. |
| 16 | cnt | Num | 8 | BEST12. | BEST12. |
| 2 | dteday | Num | 8 | DATE9. | DATE9. |
| 6 | holiday | Num | 8 | BEST12. | BEST12. |
| 12 | hum | Num | 8 | BEST12. | BEST12. |
| 1 | instant | Num | 8 | BEST12. | BEST12. |
| 5 | mnth | Num | 8 | BEST12. | BEST12. |
| 15 | registered | Num | 8 | BEST12. | BEST12. |
| 3 | season | Num | 8 | BEST12. | BEST12. |
| 10 | temp | Num | 8 | BEST12. | BEST12. |
| 9 | weathersit | Num | 8 | BEST12. | BEST12. |
| 7 | weekday | Num | 8 | BEST12. | BEST12. |
| 13 | windspeed | Num | 8 | BEST12. | BEST12. |
| 8 | workingday | Num | 8 | BEST12. | BEST12. |
| 4 | yr | Num | 8 | BEST12. | BEST12. |

Page Break

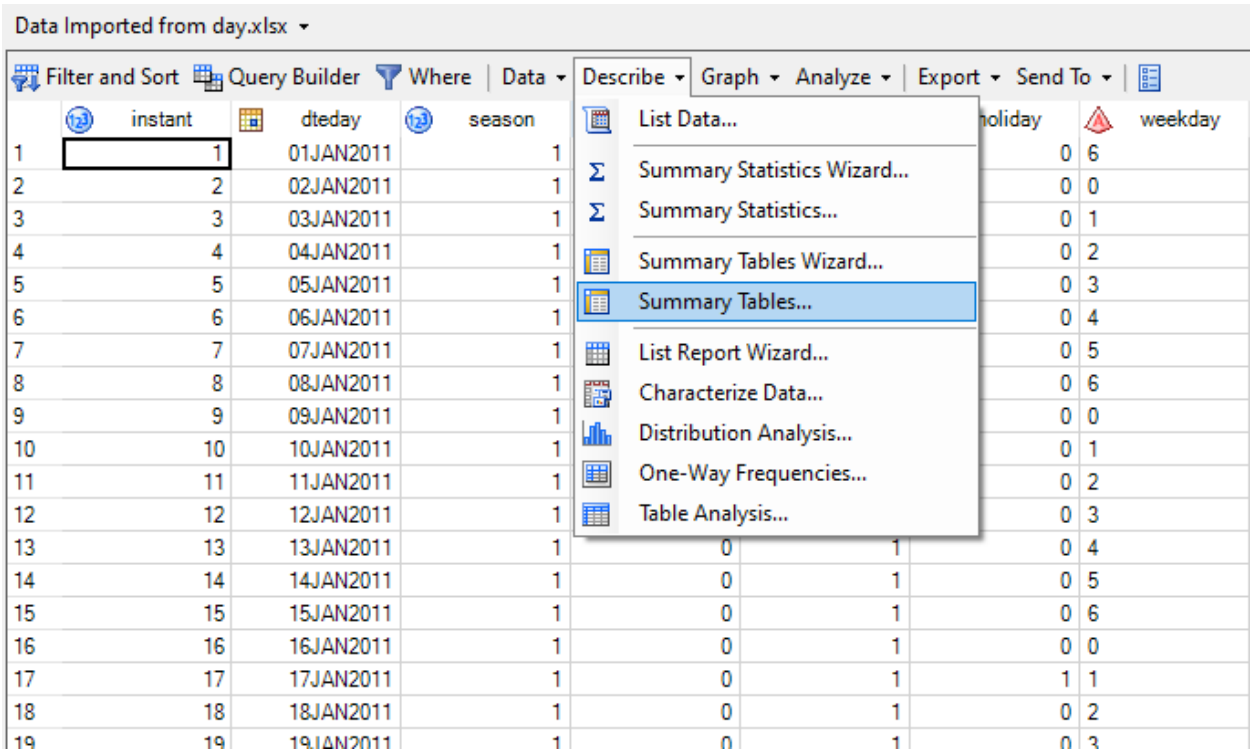
- Task:
  - Output data tab > Describe > Summary statistics (choose an analysis variable)




Options to explore variables:



- Output data tab > Describe > Summary tables

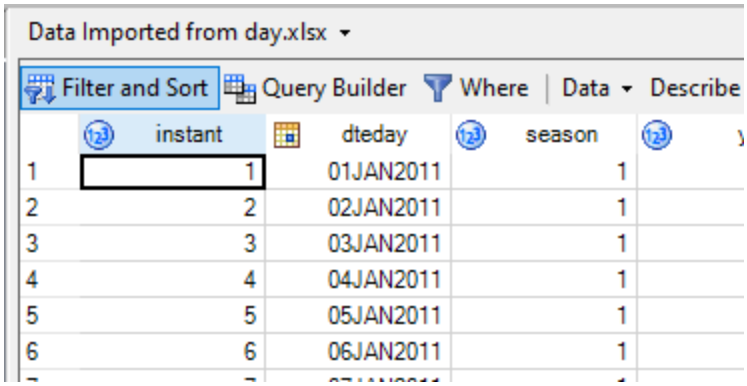


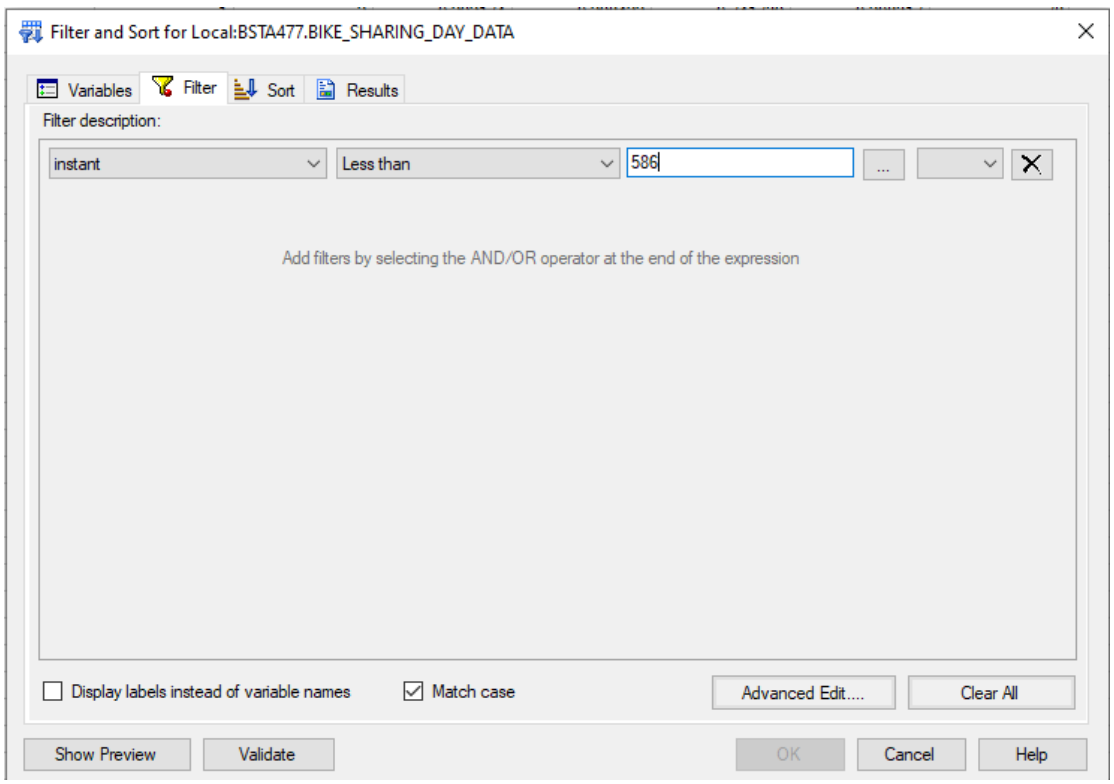

## Data partitioning

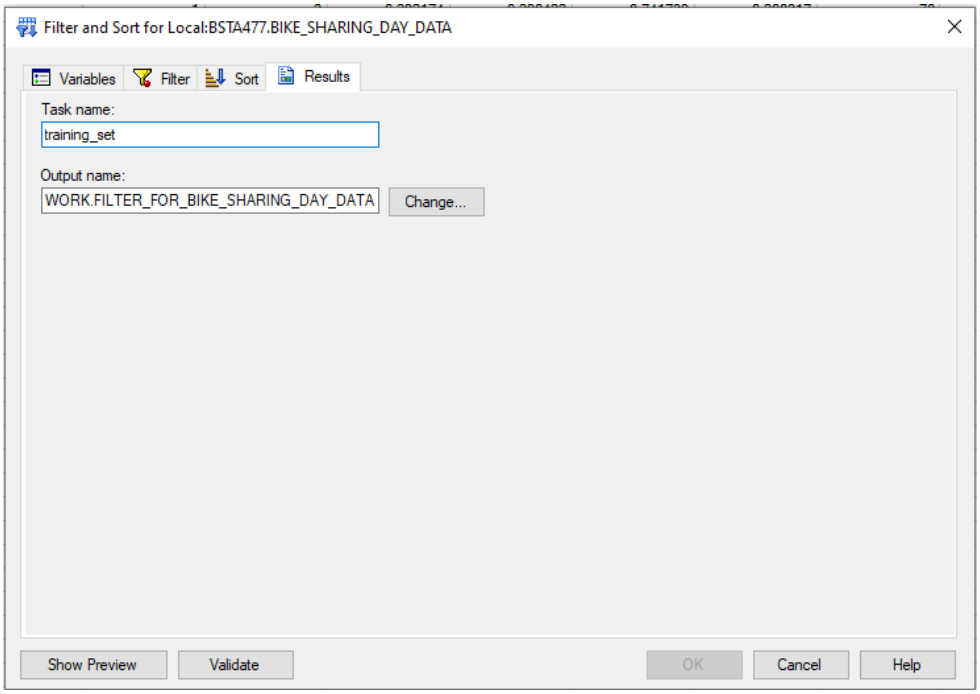
Goal of data partitioning is to partition the time series into a training set and a validation set.

**Example**: Data set with 731 observations. Goal: 80% observation in training set, 20% validation set. Because 80% of 731 is 585, time ID variable < 586 will be 80% of data set. Therefore, time ID variable > 585 will be 20% of the data set.

Training set:
Output data tab > Filter and Sort > Variable tab: Choose variables in the output set > Filter: Choose ID variable > Filter: Less than > Filter: … Choose value at 586 (because 80% of 731)


Data Imported from day.xlsx
Filter and Sort
Query Builder
Where
Data
Describe
instant
dteday
season
1 1 01JAN2011 1
2 2 02JAN2011 1
3 3 03JAN2011 1
4 4 04JAN2011 1
5 5 05JAN2011 1
6 6 06JAN2011 1


Filter and Sort for Local:BSTA477.BIKE_SHARING_DAY_DATA
Variables
Filter
Sort
Results
Filter description:
instant
Less than
586
...
Add filters by selecting the AND/OR operator at the end of the expression
Display labels instead of variable names
Match case
Advanced Edit....
Clear All
Show Preview
Validate
OK
Cancel
Help




Validation set:
Output data tab > Filter and Sort > Variable tab: Choose variables in the output set > Filter: Choose ID variable > Filter: More than > Filter: … Choose value at 585

## Data merge

Reference: [SAS Help Center MERGE statement]

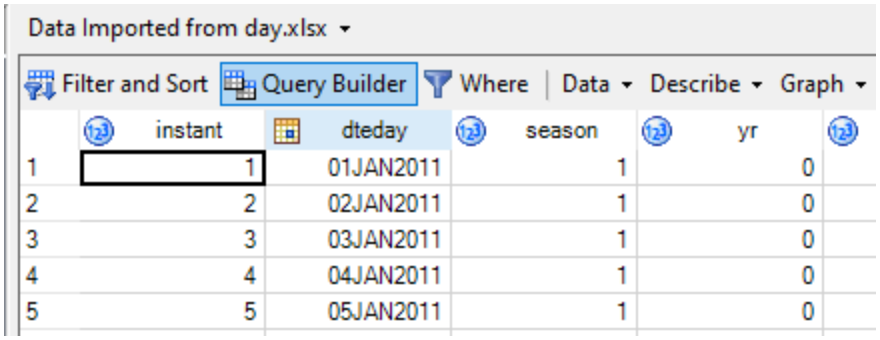
**Example**: Data_1 (var: ID, name, job), Data_2 (var: ID, name, age). Merge two data sets by column name column.

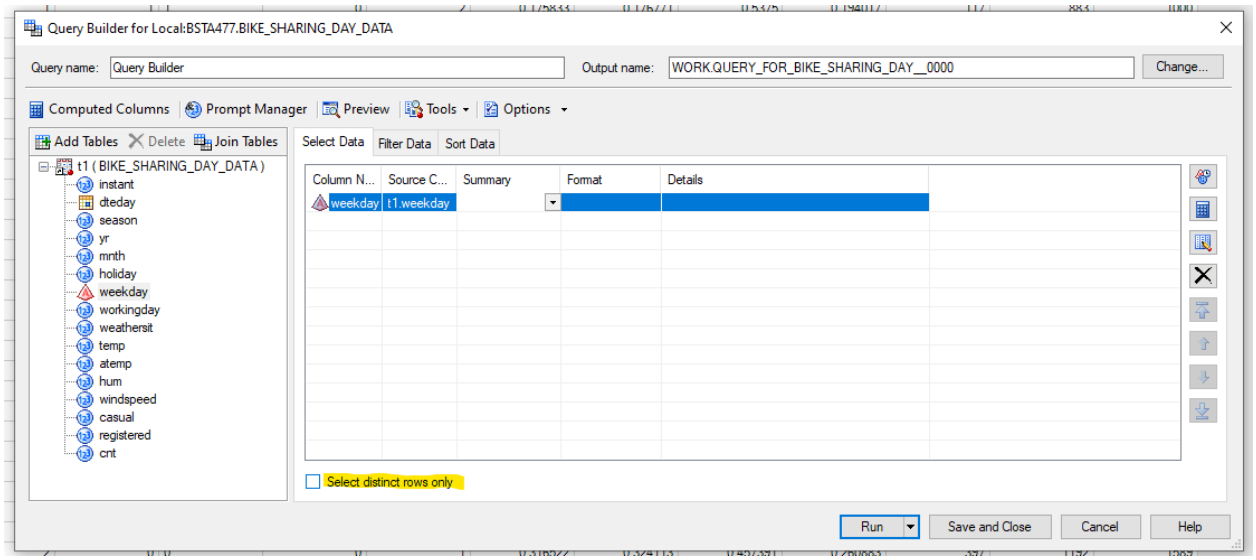
```
Data merged;
Merge data_1 data_2;
By name;
Run;
```

## Other cleaning techniques

- Remove duplicates:

Output data tab > Query Builder > Drag columns to be in output data > Select only distinct rows



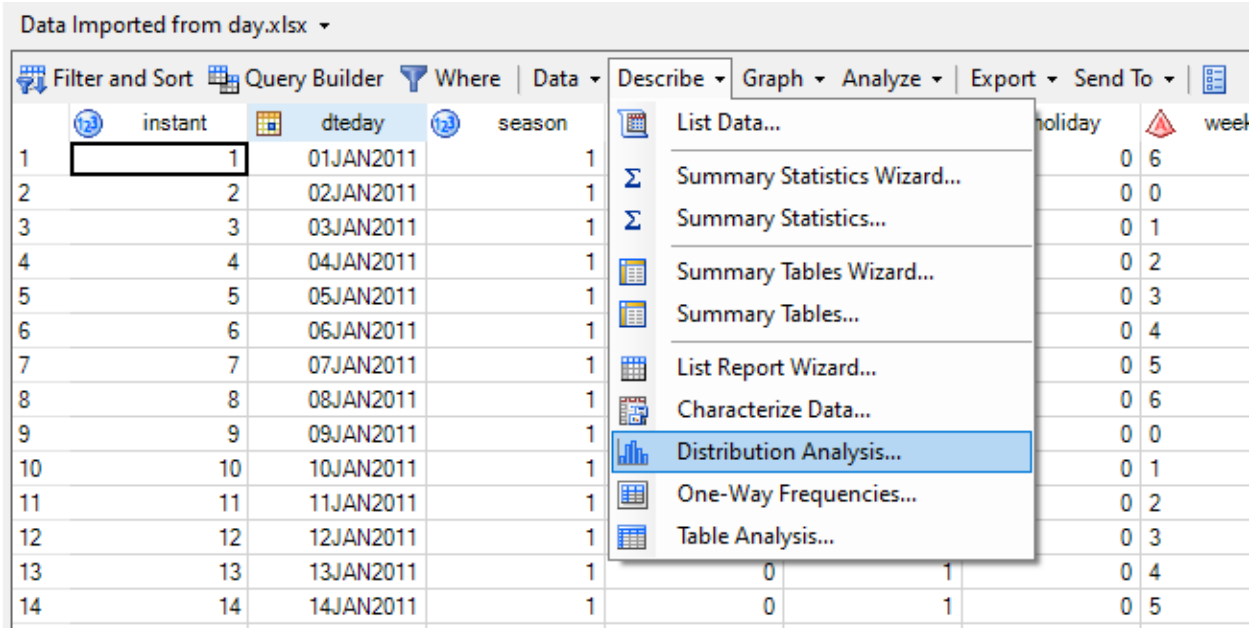

- Missing data: numerical and categorical: Proc Content

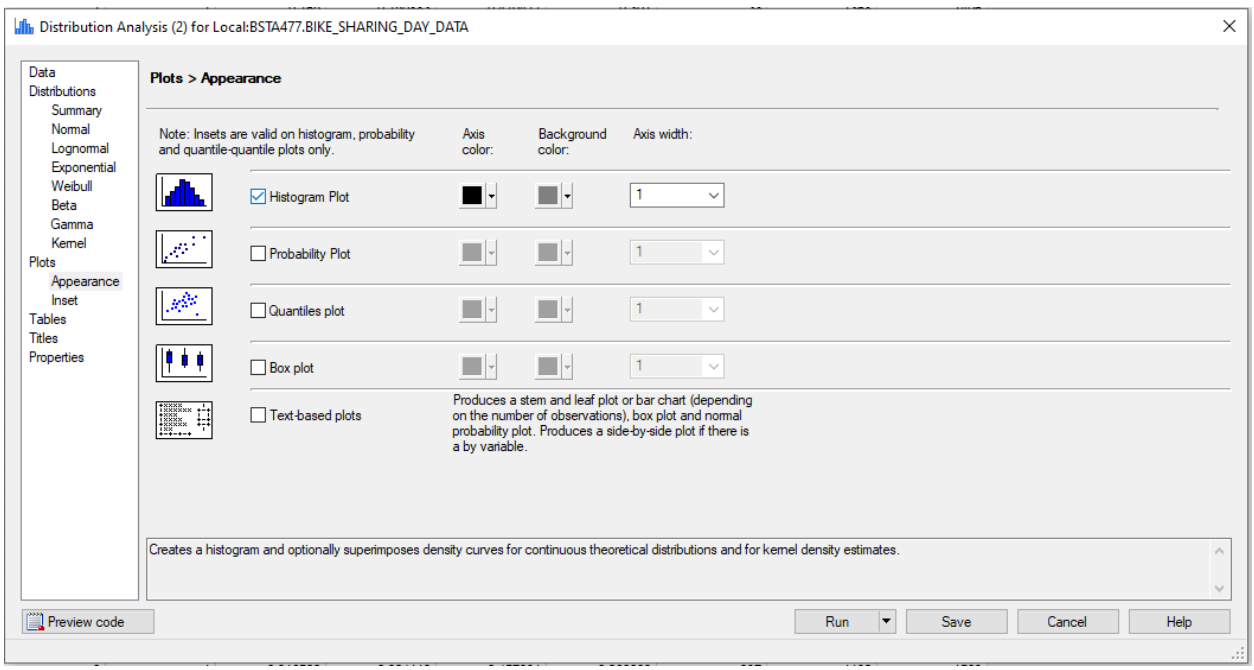
Reference: [Managing missing data using SAS Enterprise Guide]

# Time Series Graphics

## Histograms

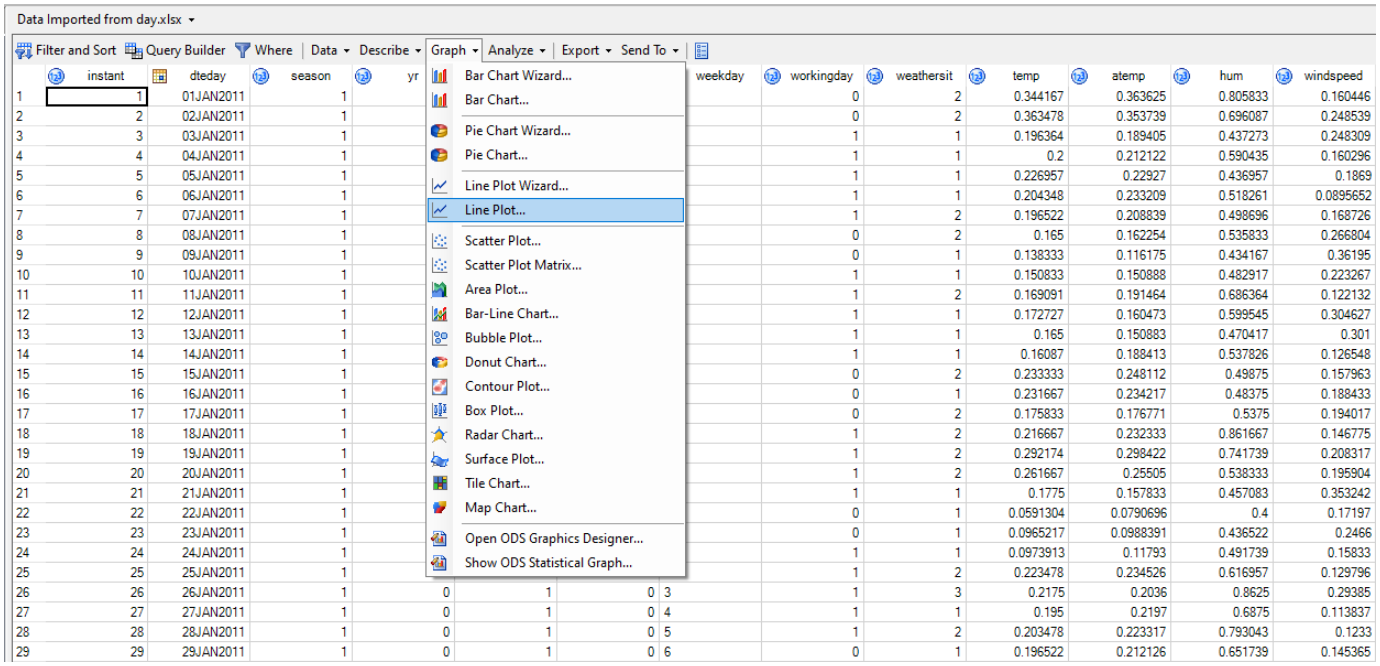
Output data tab > Describe > Distribution analysis > Select analysis variable > Distribution Summary: Normal selected > Plots Appearance: Histogram plot > Run



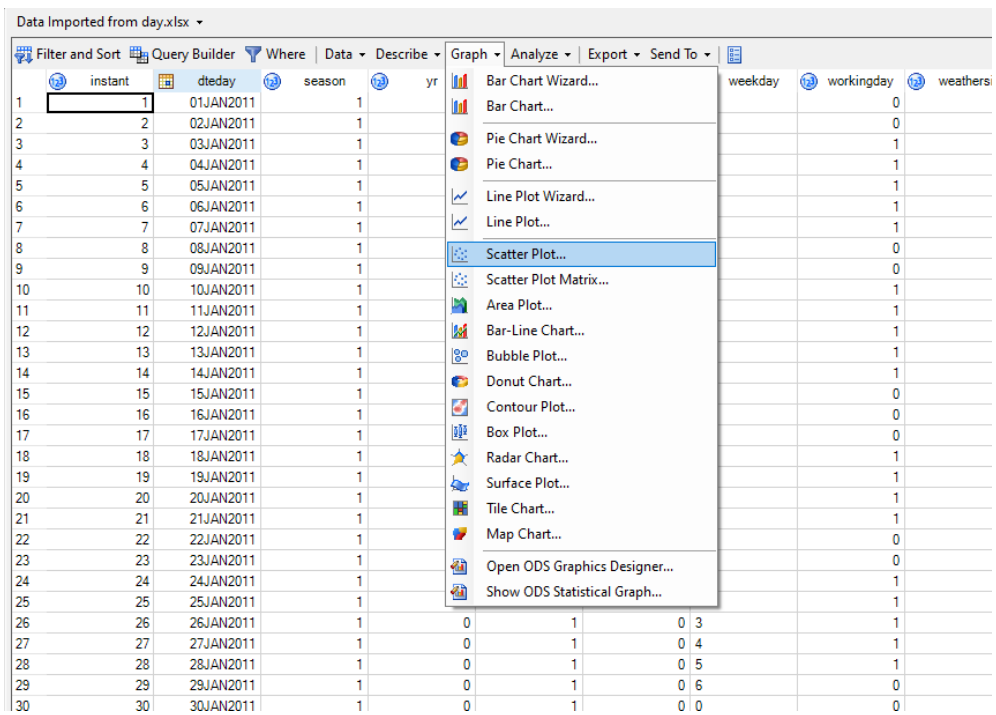

# Line plot

Output data tab > Graph > Line plot




# Scatter plot

Output data tab > Graph > Scatter plot




# ACF, PACF

Example code: Use this PROC TIMESERIES for general ACF, PACF plots

```
ods graphics on;
proc timeseries data=bsta477.bike_sharing_day_data plots=(acf pacf);
var cnt;
run;
```

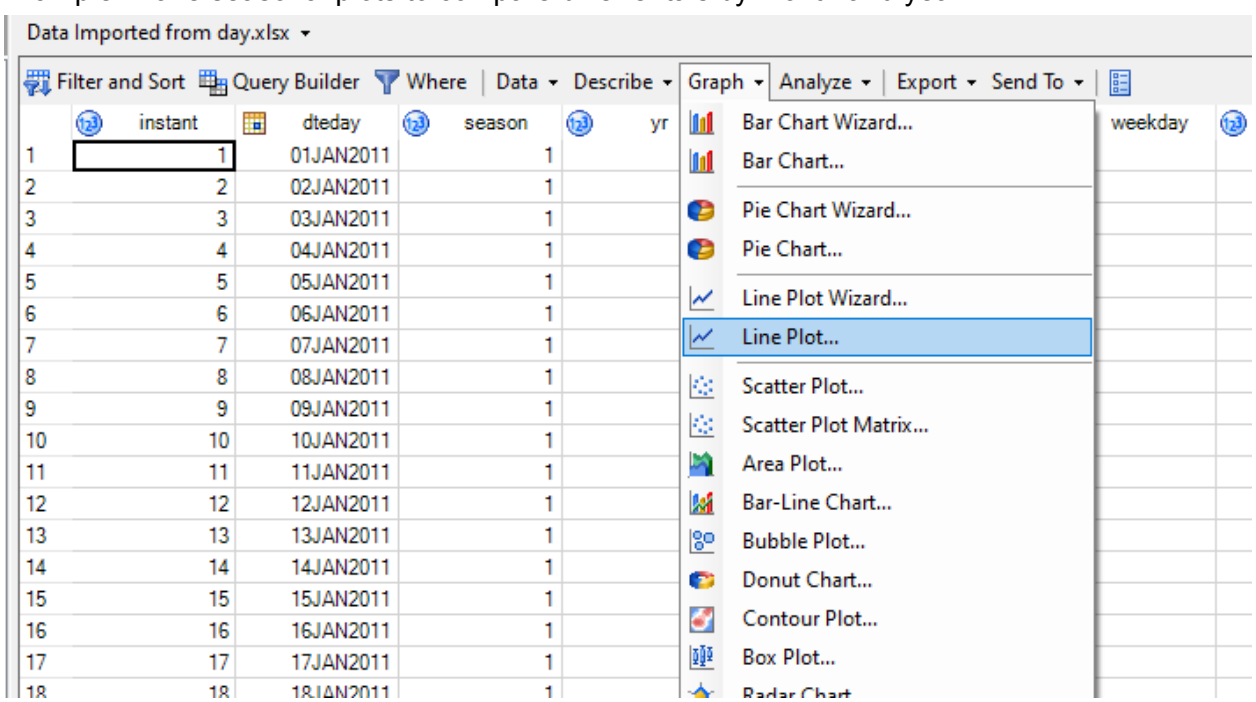
Example code: Use PROC ARIMA below for specific ACF, PACF lags required.

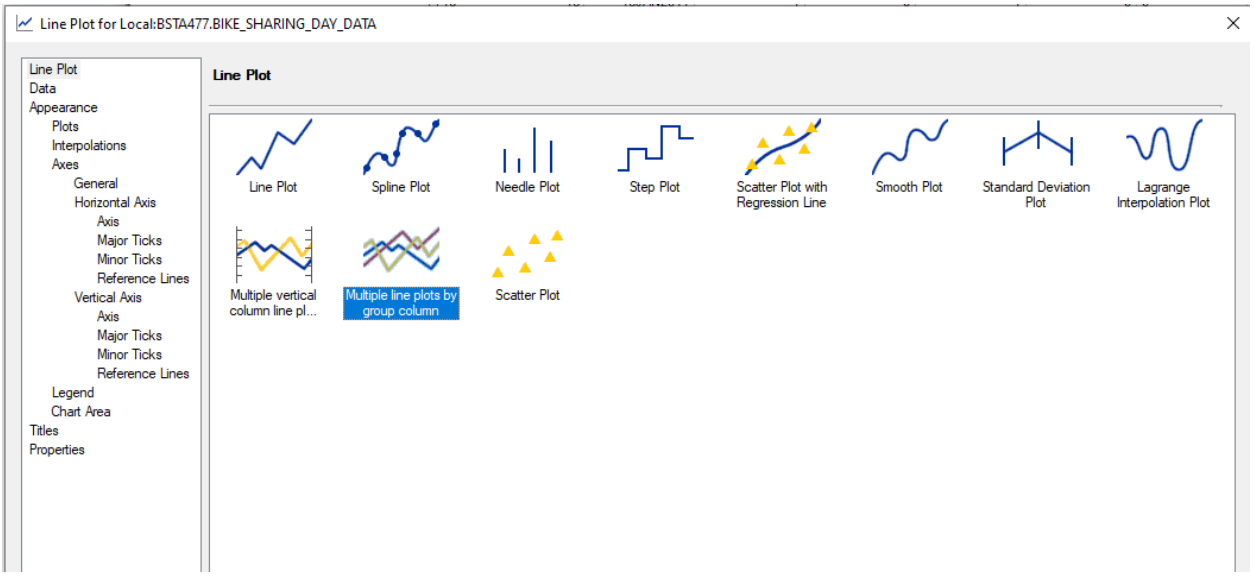
```
proc arima data=bsta477.bike_sharing_day_data;
identify var=cnt nlag=20;
run;
```

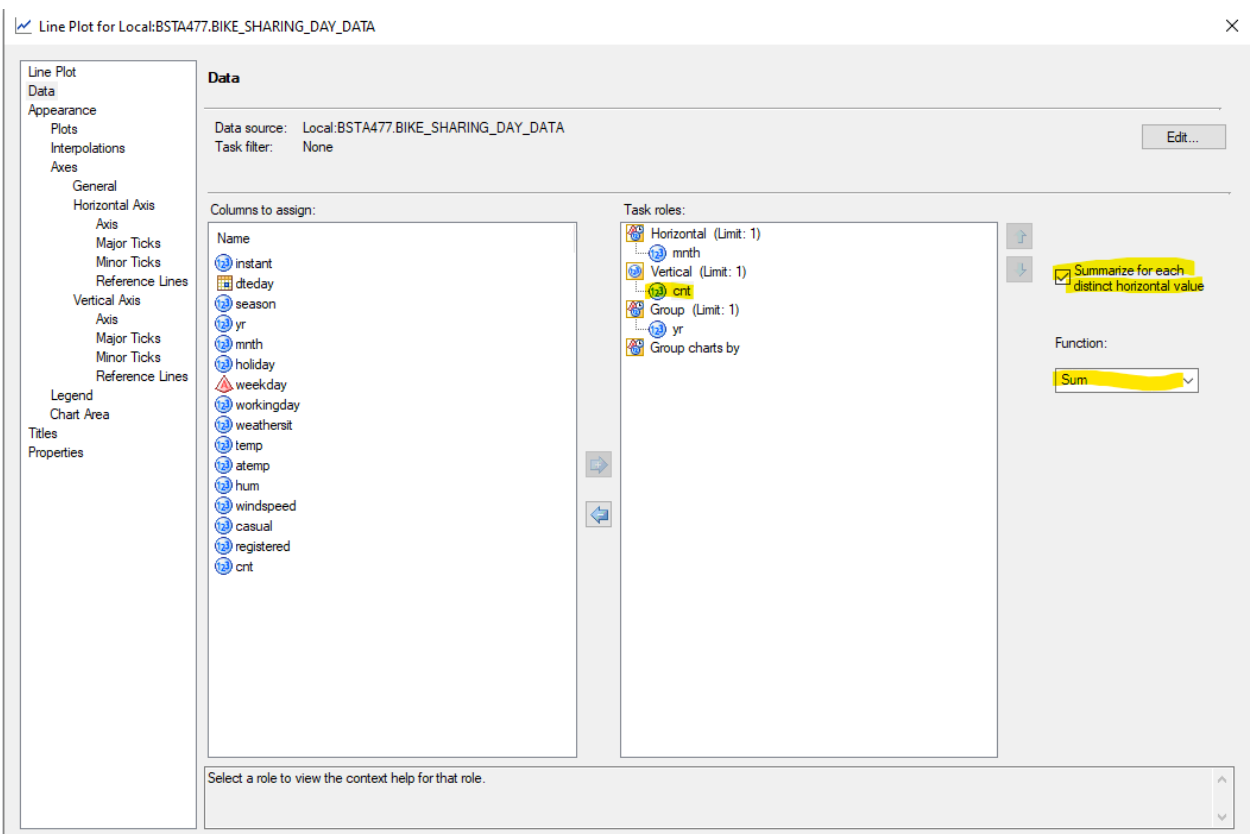
## Seasonal plot

Seasonal plots can be created using line plots with information about months and years.

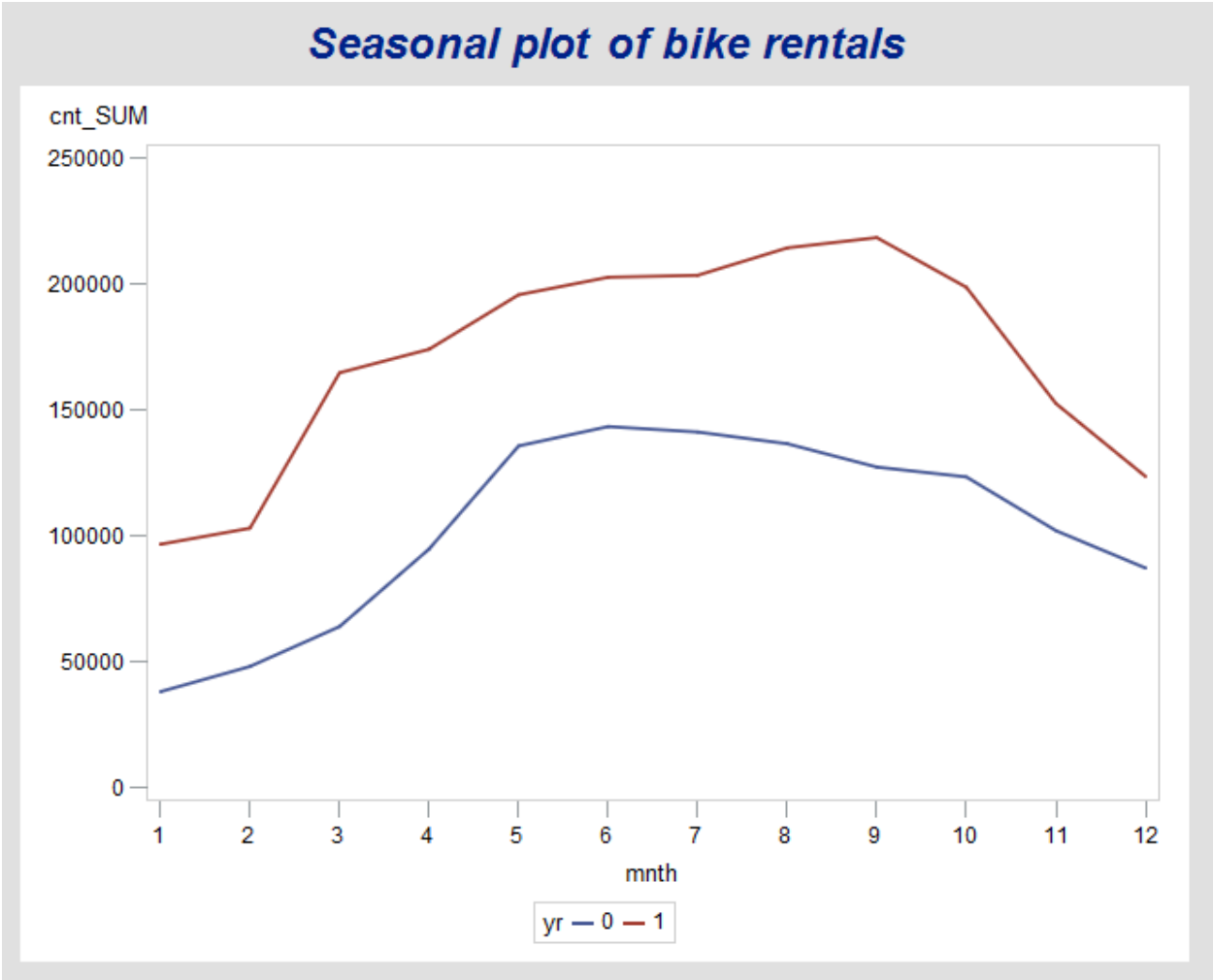
Example: Make seasonal plots to compare bike rentals by month and year.






Here, we put the bike rentals variable (cnt) in vertical axis, month variable (mnth) which is the common time indicator for each year in the horizontal axis, and year variable (yr) as a group value. **Remember to click on the numerical variable to indicate the summarize options.** (Clicked on cnt variable and chose the summarize option)




Year 0 here is 2011, year 1 is 2012.

# Time series decomposition

## Classical Decomposition

Reference:

- [How to visualize time series decomposition]
- [The TIMESERIES procedure]

### Automatic SAS EG inferred classical decomposition.

**Attention**:

- Interval variable is the seasonal length. Example: interval = qrt - quarter => m = 4

- The data either has to have the interval frequency or has use accumulate for the operation to work
- Interval options
    - Day => m = 7
    - Hour => m=24
    - Minute => m= 60
    - Second => m=60
    - Week => m=52
    - Month => m=12
    - Year => m=1
    - QRT => m=4

**<u>Code example</u>**: Decomposing time series with data accumulated in quarterly frequency with m-MA = 4.

```
proc timeseries data=bsta477.bike_sharing_day_data outdecomp=outdecomp plots=(series decomp sc tc cc residual);
   id dteday interval=qtr accumulate=total;
   var cnt;
   decomp orig tcs tcc sic tc sc cc ic / mode=add;
run;
```

## SAS Seasonality User manually defined m-MA

Used this option to define the seasonal period m. This option is also used to predict one step ahead using the **Moving average** method. For **Centered moving average method**, one has to calculate manually (in Excel)

**SEASONALITY=** *number*

specifies the length of the seasonal cycle. For example, SEASONALITY=3 means that every group of three time periods forms a seasonal cycle. By default, the length of the seasonal cycle is one (no seasonality) or the length implied by the INTERVAL= option specified in the ID statement. For example, INTERVAL=MONTH implies that the length of the seasonal cycle is 12.

**<u>Notes</u>**:

- Do not use Seasonality syntax with Interval syntax above.
- Seasonality syntax does not require the data to have the same frequency (opposed to Interval syntax).

**<u>Code example</u>**: Use moving average operation to predict one step ahead with MA(4), simply enter seasonality = 4.

```
proc timeseries data=bsta477.bike_sharing_day_data outdecomp=outdecomp_odd plots=(series decomp sc tc cc residual) seasonality=4;
   var cnt;
   decomp orig tcs tcc sic tc sc cc ic / mode=add;
run;
```

=> The prediction one head ahead is in the Trend Cycle component of the result output. The residuals are the Irregular component of the result output.

## X11 Decomposition

Reference: [X11 Decomposition SAS Documentation]

## X13 Decomposition

Reference: [X13 Decomposition SAS Documentation]

# Box-cox transformation

Please refer to your textbook for the theory on Box-cox transformation.
To use box-cox transformation, you have to test with all lambda possible to find the best one that creates the best normal distribution curve for the chosen variable.

For lambda selection, SAS automated the process of Box cox transformation. You need: a chosen variable, and a column of zeros.

Ex: Apply box cox transformation to transform bike rentals variable to have normal distribution.

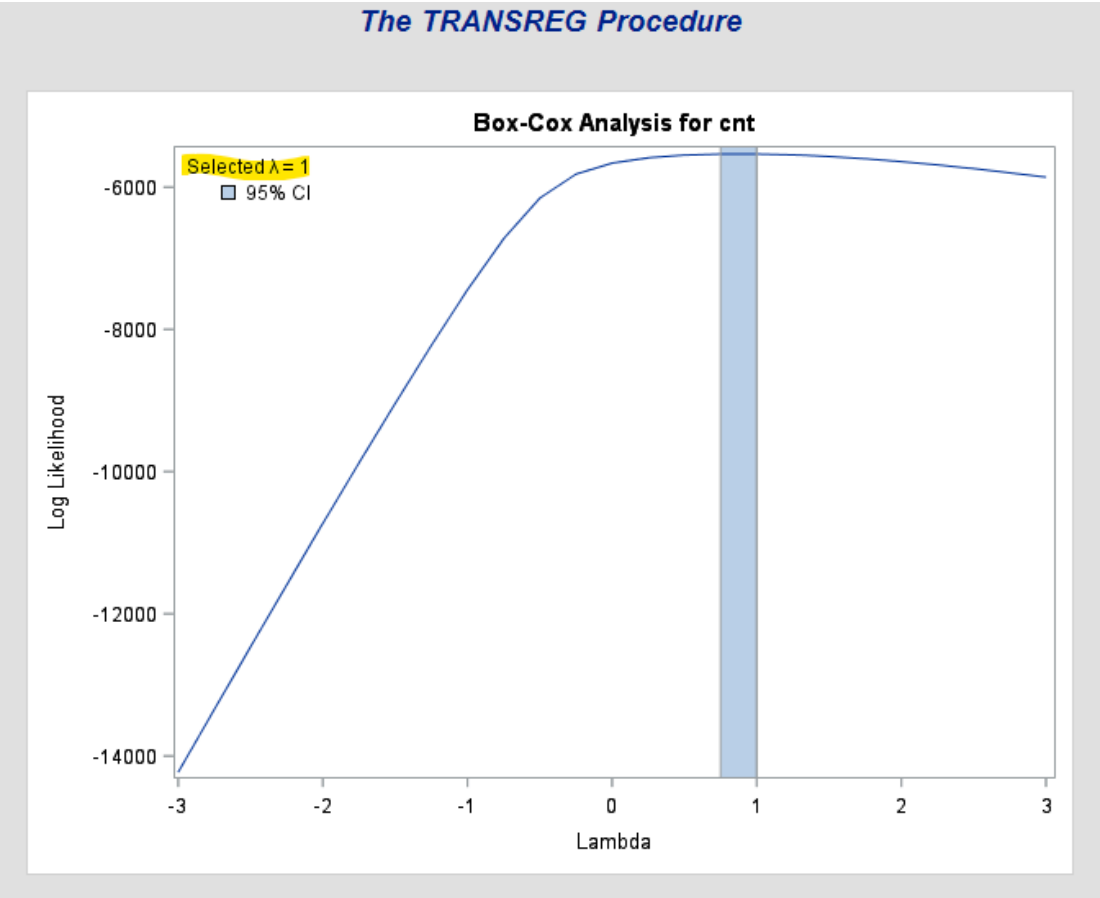
1. Add a zero column

```
data bsta477.bike_sharing_data_day;
 set bsta477.bike_sharing_data_day;
 zero = 0;
 run;
```

2. Use PROC TRANSREG

```
proc transreg data=bsta477.bike_sharing_data_day maxiter=0 nozeroconstant;
    model BoxCox(cnt) = identity(zero);
    output;
 run;
```

3. Check for optimal lambda




# SAS Resources

## SAS Student hub

- Free SAS e-Learning for Academics for Students: https://www.sas.com/en_ca/learn/academic-programs/resources/free-sas-e-learning.html#for-students

- SAS Enterprise Guide:
  - Learning: https://support.sas.com/en/software/enterprise-guide-support.html#tutorials
  - Documentation: https://documentation.sas.com/?activeCdc=egdoccdc&cdcId=egcdc&cdcVersion=8.3&docsetId=egug&docsetTarget=titlepage.htm&locale=en