

AB Analysis - Python Package

User Guide



abanalysis==1.0.2

Steps:

1. To install the ab_analysis package follow the steps below –
2. Extract the .rar file and keep the folder in your local system
3. Go to cmd => type
pip install \\...\ab_analysis_version_1_0_2 mention the complete path of zip extract
Example,
pip install C:\Users\Abhishek.Kumar\Desktop\Python_Modules\ab_analysis\ab_analysis_version_1_0_2
4. Now you should have abanalysis module available for use.

#Below is the example of test use of the abanalysis module

```
import pandas as pd #import pandas as pd for reading your data into pandas DataFrame
import timeit      #this is required to measure the execution time
from abanalysis import ab_analysis as ab    #importing the abanalysis module as ab

s_time = timeit.default_timer()
Lift_Data = ab.ab_analysis(no_of_control = 1,
                           var2pickneighbors = ['Seasonality','Trend'],
                           with_replacement = 'T',
                           Lift_Threshold = 2,
                           Performance_Measure = 'Units',
                           Identifier = 'Store', Date_Col = 'Date', Measure = 'Sales',    Cluster = 'Cluster',
                           Changed = 'Changed', Test = 'Test',
                           trend_data = trend_ori_df, measure_data = sales_df, cluster_date = cluster_date)
e_time = timeit.default_timer()
print(f'Total Execution Time {e_time-s_time}')
```

Calling AB Module with data

```
Lift_Data = ab.ab_analysis(
    no_of_control = 10,
    var2pickneighbors = ['Seasonality','Trend'],
    with_replacement = 'T',
    Lift_Threshold = 2,
    Performance_Measure = 'Units',
    Identifier = 'Store', Date_Col = 'Date', Measure = 'Sales',
    Cluster = 'Cluster', Changed = 'Changed', Test = 'Test',
    trend_data = trend_ori_df, measure_data = sales_df, cluster_date = cluster_date)
```

User Provided Values / Data

- **no_of_control = #** (Provide the number of Controls) , # = Control per Treatment Store
- **with_replacement = T or F.**
- **Performance_Measure** = Units (Constant Values in the Output)
- **Lift_Threshold** = # (assign number, so while calculating. the lift at store level, it gives lift no. and threshold is manually assigned.)
- **Identifier** = **Store** (Map the Store Column from Trend Data table)
- **Date_Col** = **Date** (Map Date Column)
- **Measure** = **Sales** (Map Sales column from Sales data table)
- **Cluster** = Cluster (Column name should be same as the Cluster Data and Trend Data)
- **Lift_Data** (all the results are saved under this name which can be defined as per the user)

User input data: trend data / Sales data / Cluster data (Sample Data)

- Avoid using (_df) name like **trend_data** = trend_df or measure_data = sales_df , instead use **trend_data** = trend_ori_df
- Data structure of the data should be same. (Column names)
- Use data format as yyyy-mm-dd
- Cluster Column name of **Cluster Data** table should be same as Cluster Column of **Trend Data** table (user can assign other column names as well but both the table should have same column name)
 - Cluster Date
- Use the template for saving dates

Cluster	Test_StartDate	Test_EndDate	Prior_StartDate	Prior_EndDate	YoY_StartDate	YoY_EndDate
A	9/10/2017	10/7/2017	6/18/2017	9/9/2017	9/11/2016	10/8/2016
B	9/10/2017	10/7/2017	6/18/2017	9/9/2017	9/11/2016	10/8/2016
C	9/10/2017	10/7/2017	6/18/2017	9/9/2017	9/11/2016	10/8/2016
All	9/10/2017	10/7/2017	6/18/2017	9/9/2017	9/11/2016	10/8/2016

- Trend Data
- Avoid using Store values as Integer (1,2, etc) . Using store no. as (S001, S002. etc.) is recommended

Store	Trend	Seasonality	Test	Cluster	Changed
1	0.005192776	-0.002199118	FALSE	B	FALSE
2	0.066038106	-0.000428914	FALSE	B	TRUE

- Test / Changed = Boolean
- Trend / Seasonality = Integer
- Cluster = String
 - Sales Data

Store	Date	Sales
1	5/24/2015	74468.05
1	5/31/2015	87829.99
1	6/7/2015	88058.13