DATA ANALYTICS REFERENCE DOCUMENT	
Document Title:	Python Data Analytics references
Document No.:	1541783112
Author(s):	Gerhard van der Linde
Contributor(s):	

REVISION HISTORY

Revision	Details of Modification(s)	Reason for modification	Date	By
0	Draft release	A Rererence summary to the libabries and commands	2018/11/09 17:05	Gerhard van der Linde

Python Reference

There's thousands of tutorial and references out there, so this is not another one, rather just a reference page to provide quick samples to analytics essentials and links to the detailed summaries instead.

Pandas

Query a dataframe for a string in a field

df_query.py

```
	df[df['Description'].str.contains('[a-z_,: ]6mm')]
```

The string in the sample is over complicated and includes a regulate expression, so can be a simple string, but has the ability to include regular expressions too.

```
#read a csv into a dataframe
csv=pd.read_csv('joblog.csv')
df=pd.read_csv("HF250-210C POR 020519.csv",skiprows=[1,2],encoding='ansi')
h5=pd.read_hdf('joblog_20180502.h5')
csv['Time Mark']=pd.to_datetime(csv['Time Mark'])
csv['RunTimeRecipe']=pd.to_timedelta(csv['RunTimeRecipe'])
csv['RunTimeRecipe']=csv['RunTimeRecipe']/pd.to_timedelta('1 hour')
h5.append(csv)
new=h5.append(csv)
new=new.reset_index(drop=True)
new.to_hdf('joblog_20180724.h5', key='index')

#merging dataframes on column
b2=pd.merge(bom,baan, on='PartNo', how='left')

#vlookup for dataframes
df5.loc[df5.Item.isin(dfu.ms),['WhereUsed']]=dfu[103:104]['usage'].item()
```

```
db=pd.read_hdf('stat/joblog_20181121.h5')
# db=pd.read_hdf('stat/joblog_20180502.h5')
#db=pd.read_csv('stat/joblog.csv')

#specify column names
db.columns=['UniqueJobIdentifier', 'JobName', 'TimeMark', 'UserName',\
'SampleTime', 'EntryAngle', 'ExitAngle', 'LoadedWafers', 'ToolID',\
'Recipename', 'RunTimeRecipe', 'VacuumMedian', 'VacuumElement',\
'TempMax', 'FieldMax', 'TempDeltaMax', 'LogFilename',\
'LogFilePath']

# Filer on tool ID
db=db[db['ToolID'].str.contains('ABCO-06')]
# Filter on Job ID - B4300 - A3225
db=db[db['JobName'].str.contains('B4300')]
# remove the runtime outliers
db=db[abs(db['RunTimeRecipe']-
db['RunTimeRecipe'].mean())<db['RunTimeRecipe'].std()*TimeOutLierFilter]
# remove the runtime outliers
db=db[abs(db['TempMax']-db['TempMax'].mean())<db['TempMax'].std()*OutLierFilter]
# remove the runtime outliers
db=db[abs(db['TempDeltaMax']-db['TempDeltaMax'].mean())<db['TempDeltaMax'].std()*OutLierFilter]
#remove times before jan2017
db=db[db['TimeMark']>pd.to_datetime('2016-01-01')]
#remove times greater than now
db=db[db['TimeMark']<pd.to_datetime(time.asctime())]
```