# Data analysis with python

Data has been imported into python in .xlsx format to preserve data types (»float«, »string«, »datetime«).

## Time format correction

Timestamp parameter was not structured uniformly in the raw data files collected from SQL database and needed to be pre-processed for each timeseries data file as shown below.

**import** **pandas** **as** **pd**

**import** **numpy** **as** **np**

**import** **os**

os.chdir(r'C:\path\Preprocessed')

rdf = pd.read\_excel (r'C:\path\Raw\1\_raw.xlsx')

rdf["timestamp"]=pd.to\_datetime(rdf["timestamp"])

rdf = rdf.set\_index('timestamp')

rdf.to\_excel('1\_preprocessed.xlsx')

## Visualisation

Time series visualisation was performed for each batch included in dataset. Visualisation is key for estalishing initial process validation. Atypical trends were investigated in details and decided whether data quality is suitable for consideration. Each timeseries csv file combines all batches that belong to a specific material code. Visualisation is needed for a separate batch, therefore a so called sub-dataframes code needed to be included in visualisation code. This enabled all batches to be plotted on separate graphs which was neccesary for visualisation purposes.

**import** **pandas** **as** **pd**

**import** **matplotlib.pyplot** **as** **plt**

**import** **csv**

**import** **numpy** **as** **np**

%**matplotlib** inline

rdf = pd.read\_excel (r'C:\path\Raw\1.xlsx')

rdf["timestamp"]=pd.to\_datetime(rdf["timestamp"])

rdf = rdf.set\_index('timestamp')

batch\_dfs = []

batches = list(rdf.batch.value\_counts().index)

batches.sort()

**for** num **in** batches:

name\_of\_df = 'rdf\_'+str(num)

name\_of\_df = rdf.loc[rdf['batch'] == num]

batch\_dfs.append(name\_of\_df)

**for** i **in** range(len(batch\_dfs)):

fig,ax = plt.subplots()

batch\_dfs[i]["tbl\_speed"].plot(linewidth=1)

plt.ylabel("Tablet press speed (tbl/h)")

ax=batch\_dfs[i]["waste"].plot(secondary\_y=**True**, style="r", linewidth=1)

plt.ylabel("Bad production (tablets)")

plt.title("Tablet press speed: " +str(batch\_dfs[i].batch.unique()))

plt.show()

## Feature extraction

Feature extraction from time series data was performed as explained in »Cholesterol-Lowering Drug Process and Quality Data«. An example code is available bellow for several features extracted.

As part of the feature extraction from time series data it was important to consider batch size which was included in normalization factor provided in the dataset. Normalization factor differs between material codes and needs to be applied correctly when needed.

### Tablet speed mean

Mean tablet speed without consideration of press speed at 0 (considered separately).

**import** **os**

**import** **pandas** **as** **pd**

rdf = pd.read\_excel (r'C:\path\1.xlsx')

rdf["timestamp"]=pd.to\_datetime(rdf["timestamp"])

rdf=rdf[rdf["tbl\_speed"] !=0]

batch\_dfs = []

batches = list(rdf.batch.value\_counts().index)

batches.sort()

**for** num **in** batches:

name\_of\_df = 'rdf\_'+str(num)

name\_of\_df = rdf.loc[rdf['batch'] == num]

batch\_dfs.append(name\_of\_df)

rows = []

**for** i **in** range(len(batch\_dfs)):

col\_a=batch\_dfs[i].batch.unique()

rows.append([",".join(col\_a), batch\_dfs[i]["tbl\_speed"].mean()])

new\_df = pd.DataFrame(rows, columns=["batch", "tbl\_speed\_mean"])

new\_df = new\_df.set\_index("batch")

os.chdir(r'C:\path\Extracted features')

writer = pd.ExcelWriter('1\_tbl\_speed.xlsx', engine='xlsxwriter')

new\_df.to\_excel(writer, 'Sheet1')

writer.save()

### Tablet speed change

Number of tablet press speed changes during the run of a batch. This needed to be normalized with batch size. Normalization factor is applied depending on product code used (see timeseries file for particular material code and normalization factor highlighted in yellow in code below).

**import** **os**

**import** **pandas** **as** **pd**

rdf = pd.read\_excel (r'C:\path\1.xlsx')

rdf["timestamp"]=pd.to\_datetime(rdf["timestamp"])

batch\_dfs = []

batches = list(rdf.batch.value\_counts().index)

batches.sort()

**for** num **in** batches:

name\_of\_df = 'rdf\_'+str(num)

name\_of\_df = rdf.loc[rdf['batch'] == num]

batch\_dfs.append(name\_of\_df)

rows = []

**for** i **in** range(len(batch\_dfs)):

col\_a=batch\_dfs[i].batch.unique()

col\_b=(batch\_dfs[i]["tbl\_speed"].diff()!=0).values.sum()

col\_b=col\_b/2.4

rows.append([",".join(col\_a), col\_b])

new\_df = pd.DataFrame(rows, columns=["batch", "tbl\_speed\_change"])

new\_df = new\_df.set\_index("batch")

os.chdir(r'C:\path\Extracted features')

writer = pd.ExcelWriter('1\_tbl\_speed\_change.xlsx', engine='xlsxwriter')

new\_df.to\_excel(writer, 'Sheet1')

writer.save()

### Tablet press speed at 0

Cumulative duration of tablet press speed at 0, normalized with batch size (material code and corresponding normalization factor is highlighted in yellow).

**import** **os**

**import** **pandas** **as** **pd**

rdf = pd.read\_excel (r'C:\path\1.xlsx')

rdf["timestamp"]=pd.to\_datetime(rdf["timestamp"])

rdf=rdf[rdf["tbl\_speed"] ==0]

batch\_dfs = []

batches = list(rdf.batch.value\_counts().index)

batches.sort()

**for** num **in** batches:

name\_of\_df = 'rdf\_'+str(num)

name\_of\_df = rdf.loc[rdf['batch'] == num]

batch\_dfs.append(name\_of\_df)

rows = []

**for** i **in** range(len(batch\_dfs)):

col\_a=batch\_dfs[i].batch.unique()

col\_b=batch\_dfs[i]["tbl\_speed"].count()

col\_b=col\_b/2.4

rows.append([",".join(col\_a), col\_b])

new\_df = pd.DataFrame(rows, columns=["batch", "tbl\_speed\_0\_duration"])

new\_df = new\_df.set\_index("batch")

os.chdir(r'C:\path\Extracted features')

writer = pd.ExcelWriter('1\_tbl\_speed\_0\_duration.xlsx', engine='xlsxwriter')

new\_df.to\_excel(writer, 'Sheet1')

writer.save()

### Total number of rejected tablets during startup

Number of rejected tablets before regular production starts (tablet press set up phase).

**import** **os**

**import** **pandas** **as** **pd**

rdf = pd.read\_excel (r'C:\path\1.xlsx')

rdf["timestamp"]=pd.to\_datetime(rdf["timestamp"])

rdf=rdf[rdf["produced"] ==0]

batch\_dfs = []

batches = list(rdf.batch.value\_counts().index)

batches.sort()

**for** num **in** batches:

name\_of\_df = 'rdf\_'+str(num)

name\_of\_df = rdf.loc[rdf['batch'] == num]

batch\_dfs.append(name\_of\_df)

rows = []

**for** i **in** range(len(batch\_dfs)):

col\_a=batch\_dfs[i].batch.unique()

col\_b=batch\_dfs[i]["waste"].max()

rows.append([",".join(col\_a), col\_b])

new\_df = pd.DataFrame(rows, columns=["batch", "startup\_waste"])

new\_df = new\_df.set\_index("batch")

os.chdir(r'C:\path\Extracted features')

writer = pd.ExcelWriter('1\_startup\_waste.xlsx', engine='xlsxwriter')

new\_df.to\_excel(writer, 'Sheet1')

writer.save()