Use Case

Cooling Tower - Analysis of Temperature & Total Power Prediction

(Production across different area under Cooling Tower I Asset Tree)

Cooling Tower - Analysis of Temperature & Total Power Prediction

Pilot Guided Analytics

Tridiagonal® Solutions
compute | innovate | develop

Problem Objective

- For a cooling tower operations, where the rate of heat transfer between stream of water and air drive the overall efficiency and economy of the asset, it should be an interesting analysis to identify and establish a non-linear relationship between the temperature signals and the power for the equipment.
- Trying to optimize the compressor power could be a big challenge. Here, in this worksheet we try to figure out which asset's temperature signal could have more impact on the power.
- A predictive model could be built on top of this analysis.

What is required?

- Event Contextualization
- Feature Engineering
- Identify Synthetic Parameters
- Advanced Visualization
- Build Predictive Model
- Deploy

Cooling Tower - Analysis of Temperature & Total Power Prediction

Pilot Guided Analytics

Tridiagonal® Solutions
compute | innovate | develop

jupyter Python Notebook

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

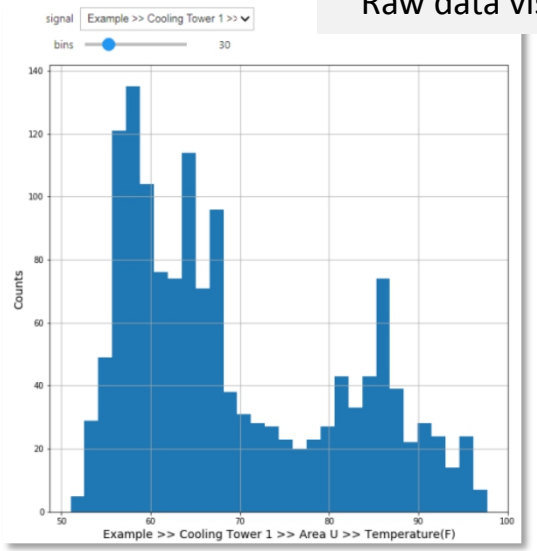
Run Code Voila

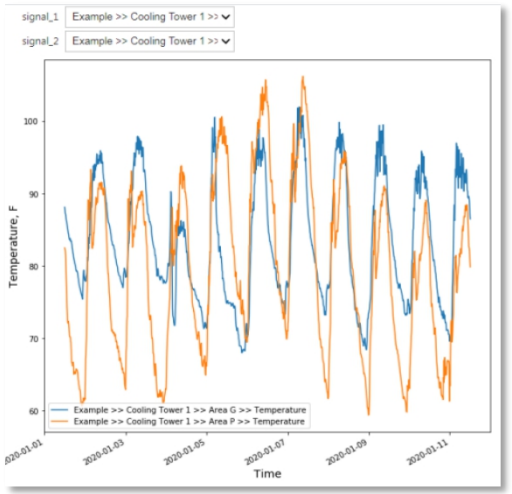
In [8]:
```
def apply_PCA(X_data):
    scaler=StandardScaler()
    X_train=scaler.fit_transform(X_data)
#     X_test=scaler.transform(X_test)

    apply_PCA.pca=PCA()
    apply_PCA.pca.fit(X_data)
    Xp_data=apply_PCA.pca.transform(X_data)

    components=apply_PCA.pca.components_
    explained_variance_ratio=apply_PCA.pca.explained_variance_ratio_
    explained_variance=apply_PCA.pca.explained_variance_[0:5]
    n_components=apply_PCA.pca.n_components_

    return Xp_data,components,explained_variance_ratio,explained_variance,n_components
```

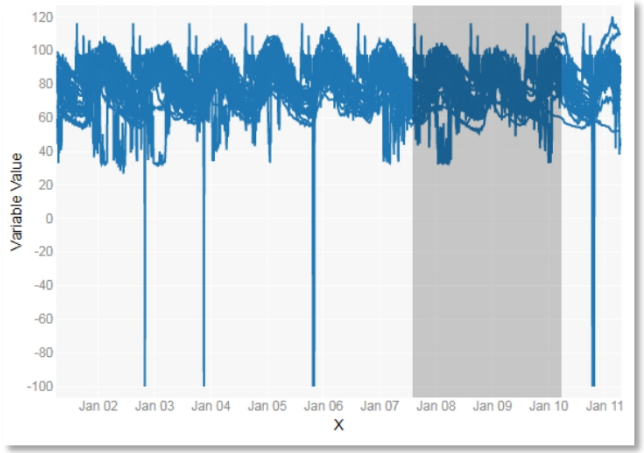
In [9]:
```
def plot_PC_variance(X_data,explained_variance_ratio):
    Np_comp=(1,len(X_data.columns),1)
    @interact(layout={'width':'10px'})
    def pca_comp_var(Np=Np_comp):
        fig,ax=plt.subplots(1,1,figsize=(10,10))
        plt.figure(4)
        ax.set_xlabel('Principal Components')
        ax.set_ylabel('Explained Variance')

        ax.plot(np.arange(Np),explained_variance_ratio[0:Np],label='variance explained by each component')
        ax.bar(np.arange(Np),explained_variance_ratio[0:Np],label='Bar plot for explained var for each comp')
        ax.plot(np.arange(Np),np.cumsum(explained_variance_ratio[0:Np]),label='Cumulative variance explained')
        ax.legend(loc='upper left')
```

Data Labs high level View

Raw data visualization







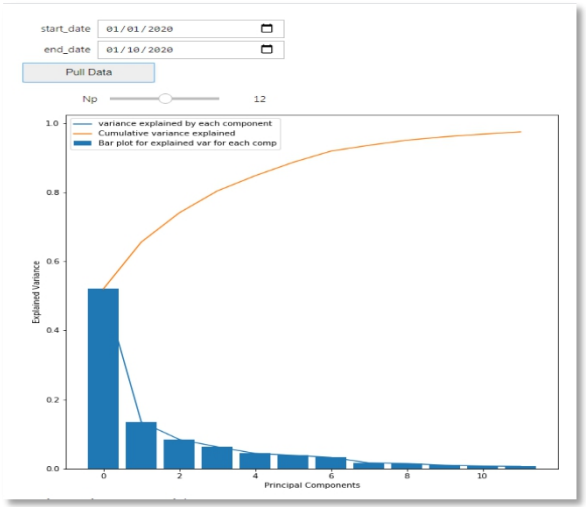
- Easy pull, push data from seeq workbench and create a first level of custom visualization in Seeq DataLabs.
- Apply PCA/PLS for identifying the critical input parameters and evaluate the state of the system, when deviation detected.

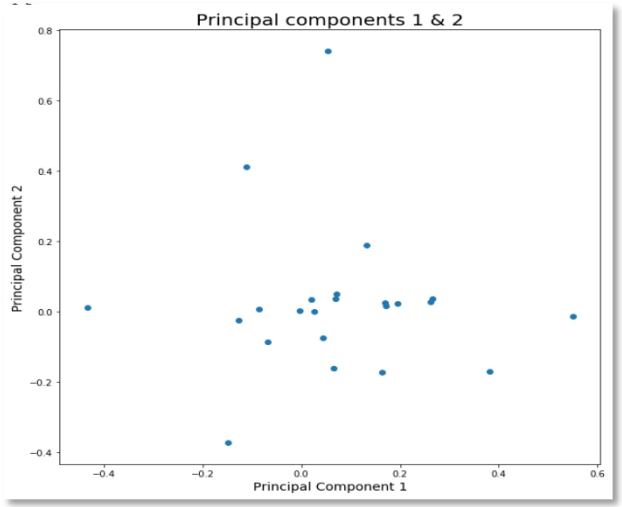
Cooling Tower - Analysis of Temperature & Total Power Prediction

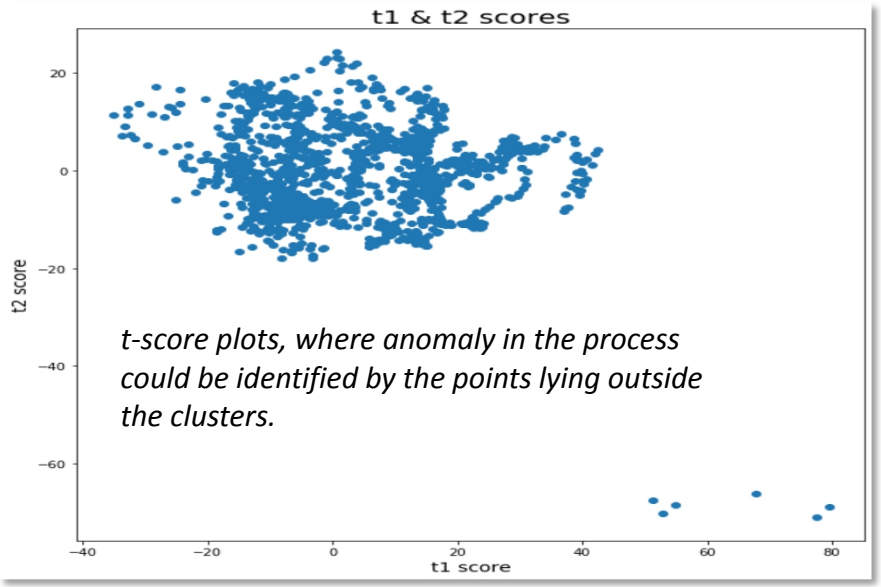
Pilot Guided Analytics

Tridiagonal® Solutions
compute | innovate | develop

Root Cause Analysis

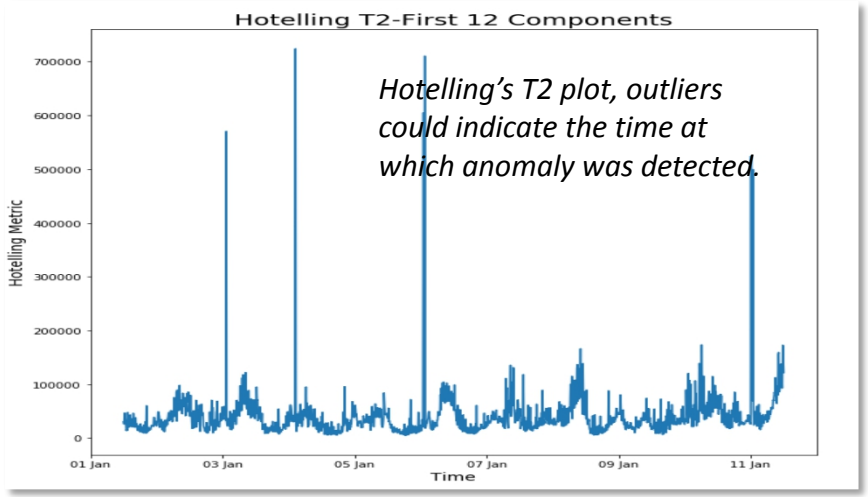






t-score plots, where anomaly in the process could be identified by the points lying outside the clusters.

- Python enables the user to apply advanced ML/AL modeling
- Principal component analysis to perform the root cause analysis, using the t-score plots and Hotelling's T2 plot
- These plots can be useful when performing fault detection based on historical data
- Deviation/Abnormal behavior detection



Hotelling's T2 plot, outliers could indicate the time at which anomaly was detected.

Cooling Tower - Analysis of Temperature & Total Power Prediction

Pilot Guided Analytics

Tridiagonal® Solutions
compute | innovate | develop

jupyter Python Notebook

Advanced AI / ML Model

- User can create a library of advanced Non-linear ML models for predictive analytics
- Compare the results from various model's results and accuracy and select the best performing model.

- Uses cases:
  - *Performance parameters*
  - *Early detection of an event*