Initial FrED Statistical Exploration 2

Exploration of basic statistics on initial FrED data.

Uses data - "Run_Comparison_2_A_Open_Loop.csv", "Run_Comparison_2_B_Open_Loop.csv"

J. Cuiffi - Penn State New Kensington

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
# import data - no path - data should be in same folder as notebook
dataA = pd.read_csv('Run_Comparison_2_A_Open_Loop.csv')
dataB = pd.read_csv('Run_Comparison_2_B_Closed_Loop.csv')
dataB
```

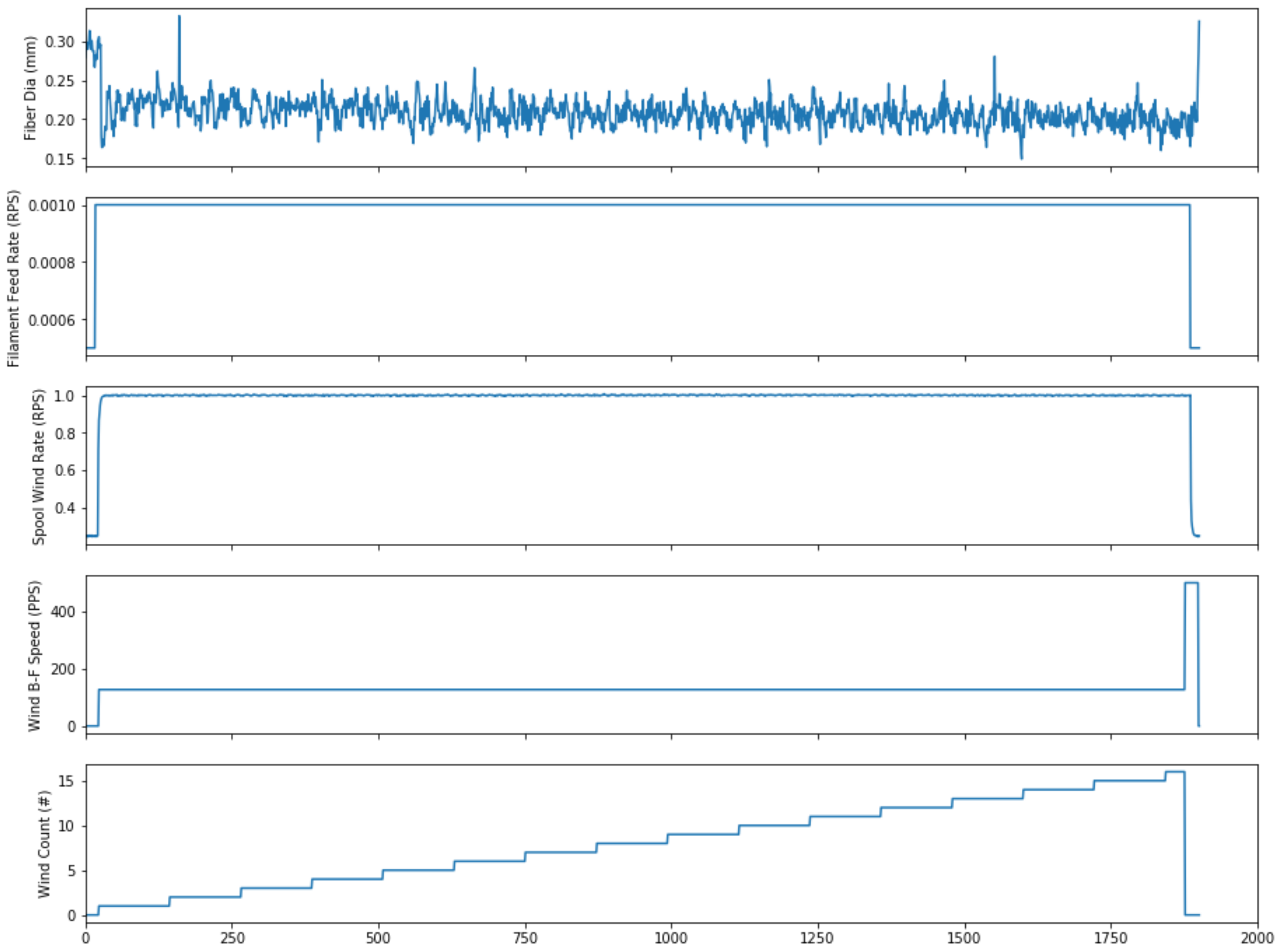
Out[2]:

	Run Time (sec)	Filament Feed Rate Set (RPS)	Spool Wind Rate Set (RPS)	Wind B-F Speed (PPS)	Filament Diameter Set (mm)	Spool Wind Rate Actual (RPS)	Wind Count (#)	Filament Diameter Actual (mm)
0	0.000000	0.0005	0.25	0	0.0	0.247	0	0.306
1	0.980893	0.0005	0.25	0	0.0	0.252	0	0.290
2	1.981293	0.0005	0.25	0	0.0	0.248	0	0.297
3	2.979438	0.0005	0.25	0	0.0	0.248	0	0.308
4	3.980020	0.0005	0.25	0	0.0	0.248	0	0.293
...	...	...	...	...	...	...	...	...
1985	1985.019386	0.0005	0.25	0	0.0	0.263	0	0.282
1986	1986.024497	0.0005	0.25	0	0.0	0.254	0	0.282
1987	1987.019836	0.0005	0.25	0	0.0	0.253	0	0.283
1988	1988.024751	0.0005	0.25	0	0.0	0.255	0	0.319
1989	1989.018169	0.0005	0.25	0	0.0	0.253	0	0.300

1990 rows × 8 columns

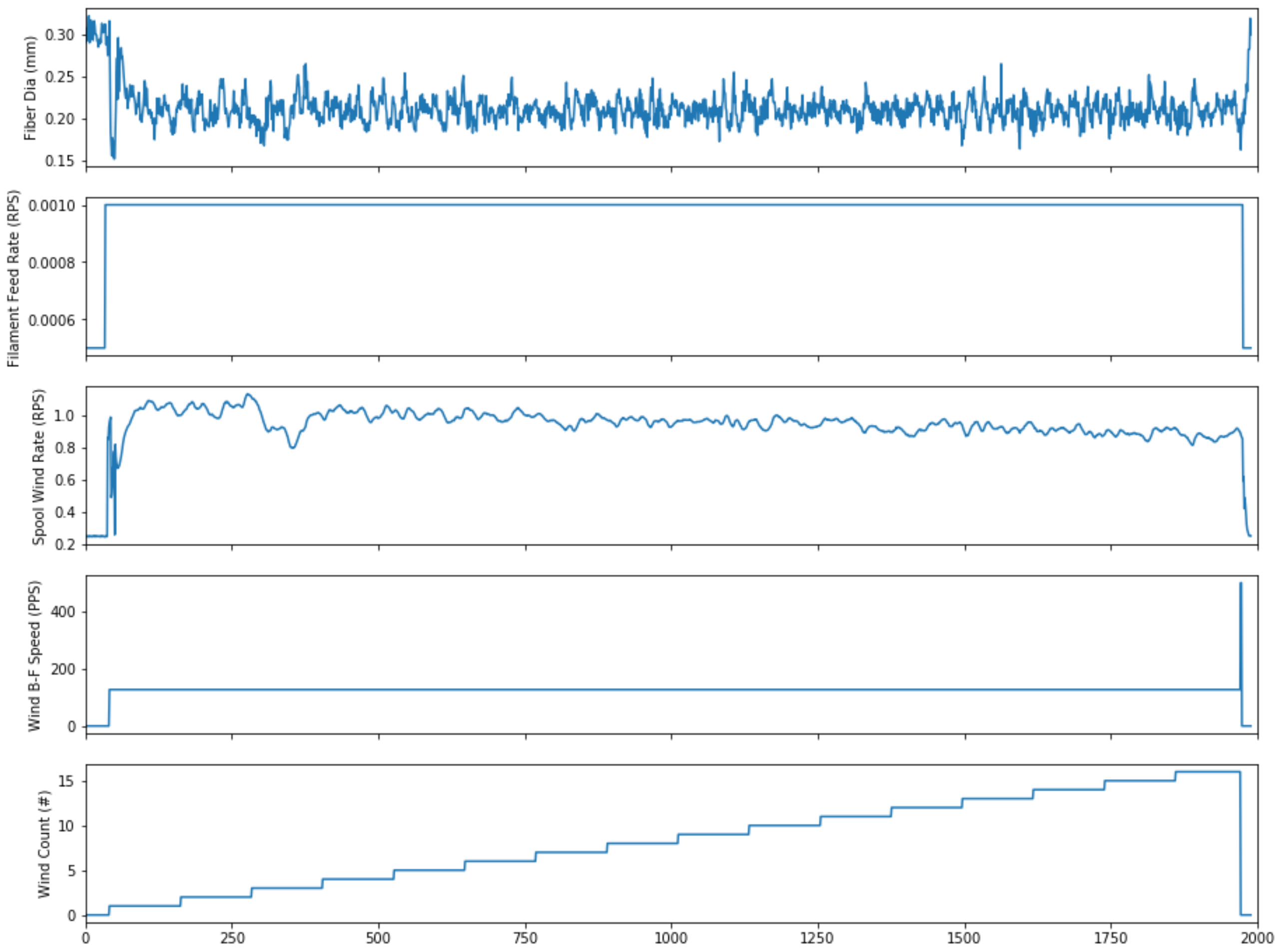
```
In [13]: # plot Run A
         fig, axs = plt.subplots(5, 1, sharex=True)
         fig.set_size_inches(15,12)
         axs[0].set_xlim(0,2000)
         axs[0].plot(dataA['Run Time (sec)'],dataA['Filament Diameter Actual (mm)'])
         axs[0].set_ylabel('Fiber Dia (mm)')
         axs[1].plot(dataA['Run Time (sec)'],dataA['Filament Feed Rate Set (RPS)'])
         axs[1].set_ylabel('Filament Feed Rate (RPS)')
         axs[2].plot(dataA['Run Time (sec)'],dataA['Spool Wind Rate Actual (RPS)'])
         axs[2].set_ylabel('Spool Wind Rate (RPS)')
         axs[3].plot(dataA['Run Time (sec)'],dataA['Wind B-F Speed (PPS)'])
         axs[3].set_ylabel('Wind B-F Speed (PPS)')
         axs[4].plot(dataA['Run Time (sec)'],dataA['Wind Count (#)'])
         axs[4].set_ylabel('Wind Count (#)')
         print('Run A Plot Summary')
```

```
Run A Plot Summary
```



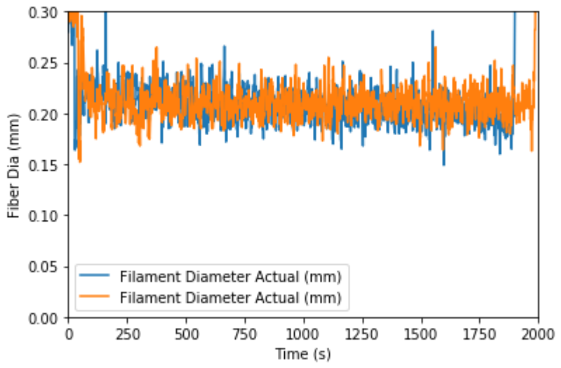
```
In [21]: # plot Run B
         fig, axs = plt.subplots(5, 1, sharex=True)
         fig.set_size_inches(15,12)
         axs[0].set_xlim(0,2000)
         axs[0].plot(dataB['Run Time (sec)'],dataB['Filament Diameter Actual (mm)'])
         axs[0].set_ylabel('Fiber Dia (mm)')
         axs[1].plot(dataB['Run Time (sec)'],dataB['Filament Feed Rate Set (RPS)'])
         axs[1].set_ylabel('Filament Feed Rate (RPS)')
         axs[2].plot(dataB['Run Time (sec)'],dataB['Spool Wind Rate Actual (RPS)'])
         axs[2].set_ylabel('Spool Wind Rate (RPS)')
         axs[3].plot(dataB['Run Time (sec)'],dataB['Wind B-F Speed (PPS)'])
         axs[3].set_ylabel('Wind B-F Speed (PPS)')
         axs[4].plot(dataB['Run Time (sec)'],dataB['Wind Count (#)'])
         axs[4].set_ylabel('Wind Count (#)')
         print('Run B Plot Summary')
```

```
Run B Plot Summary
```



```
In [29]: # Run Comaprison
         plt.plot(dataA['Run Time (sec)'],dataA['Filament Diameter Actual (mm)'])
         plt.plot(dataB['Run Time (sec)'],dataB['Filament Diameter Actual (mm)'])
         plt.ylim(0,.3)
         plt.xlim(0,2000)
         plt.legend()
         plt.ylabel('Fiber Dia (mm)')
         plt.xlabel('Time (s)')
         print('Run A - Open Loop versus Run B - Closed Loop')
```

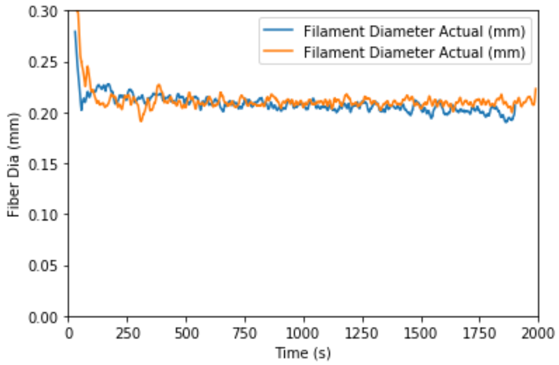
```
Run A - Open Loop versus Run B - Closed Loop
```



In [32]:

```
# Run Comaprison - 30 sec rolling ave
plt.plot(dataA['Run Time (sec)'],dataA['Filament Diameter Actual (mm)'].rolling(window=30).mean())
plt.plot(dataB['Run Time (sec)'],dataB['Filament Diameter Actual (mm)'].rolling(window=30).mean())
plt.ylim(0,.3)
plt.xlim(0,2000)
plt.legend()
plt.ylabel('Fiber Dia (mm)')
plt.xlabel('Time (s)')
print('Run A - Open Loop versus Run B - Closed Loop')
```

```
Run A - Open Loop versus Run B - Closed Loop
```



In [42]:

```
# clip data to include only filament production data
# use data between 200-1800s only
dataA_clip = dataA[(dataA['Run Time (sec)'] >= 200) & (dataA['Run Time (sec)'] <= 1800)]
dataB_clip = dataB[(dataB['Run Time (sec)'] >= 200) & (dataB['Run Time (sec)'] <= 1800)]
dataA_clip
```

Out[42]:

	Run Time (sec)	Filament Feed Rate Set (RPS)	Spool Wind Rate Set (RPS)	Wind B-F Speed (PPS)	Filament Diameter Set (mm)	Spool Wind Rate Actual (RPS)	Wind Count (#)	Filament Diameter Actual (mm)
200	200.034864	0.001	1.0	127	0	1.002	2	0.220
201	201.047124	0.001	1.0	127	0	1.002	2	0.219
202	202.003518	0.001	1.0	127	0	1.000	2	0.222
203	203.001034	0.001	1.0	127	0	1.003	2	0.229
204	203.999888	0.001	1.0	127	0	1.000	2	0.221
...	...	...	...	...	...	...	...	...
1795	1795.032491	0.001	1.0	127	0	1.001	15	0.220
1796	1796.031234	0.001	1.0	127	0	1.002	15	0.247
1797	1797.031941	0.001	1.0	127	0	0.998	15	0.220
1798	1798.030225	0.001	1.0	127	0	1.002	15	0.213
1799	1799.034034	0.001	1.0	127	0	1.000	15	0.204

1600 rows × 8 columns

In [45]:

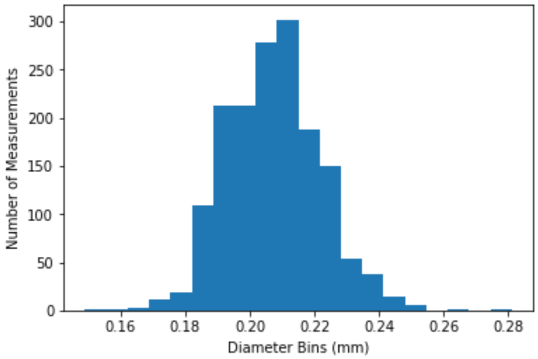
```
# basic statistics
print('Run A Model Target Diameter:',6.73*np.sqrt(.001/1.0))
print('Run A - Open Loop Mean:', dataA_clip['Filament Diameter Actual (mm)'].mean())
print('Run A Std Dev:', dataA_clip['Filament Diameter Actual (mm)'].std())
print('Run B Target Diamter: .210')
print('Run B Closed Loop Mean:', dataB_clip['Filament Diameter Actual (mm)'].mean())
print('Run B Std Dev:', dataB_clip['Filament Diameter Actual (mm)'].std())
```

```
Run A Model Target Diameter: 0.21282128652933194
Run A - Open Loop Mean: 0.2073737500000006
Run A Std Dev: 0.014609456135252106
Run B Target Diamter: .210
Run B Closed Loop Mean: 0.2096762500000002
Run B Std Dev: 0.013733009799798131
```

In [48]:

```
# histogram - A
n, bins, patches = plt.hist(dataA_clip['Filament Diameter Actual (mm)'],20)
plt.xlabel('Diameter Bins (mm)')
plt.ylabel('Number of Measurements')
print('Run A Histogram')
```

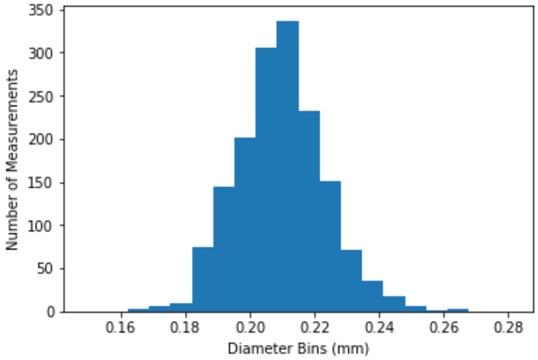
```
Run A Histogram
```



In [49]:

```
# histogram - B
plt.hist(dataB_clip['Filament Diameter Actual (mm)'],bins)
plt.xlabel('Diameter Bins (mm)')
plt.ylabel('Number of Measurements')
print('Run B Histogram')
```

```
Run B Histogram
```



In [ ]: