

```
In [1]: import pandas as pd
```

```
In [2]: import matplotlib.pyplot as plt
```

```
In [3]: import seaborn as sns
```

```
In [4]: #ES_INDEX_DATE="2018-01-28"  
ES_INDEX_DATE="2018-02-02"  
DATA_FILE="../../rbcddata/%s/spv" % ES_INDEX_DATE
```

```
In [5]: #%matplotlib inline  
#import mpld3  
#mpld3.enable_notebook()
```

```
In [6]: %pylab inline  
pylab.rcParams['figure.figsize'] = (16, 7)  
  
Populating the interactive namespace from numpy and matplotlib
```

```
In [7]: data=pd.read_csv(DATA_FILE)
```

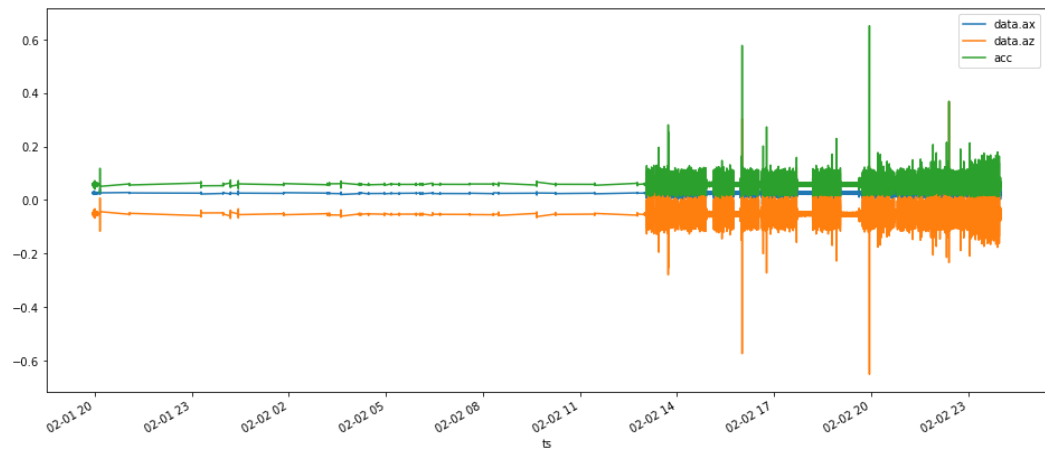
```
In [8]: section=75*60*60*48
```

```
In [9]: temp=data.head(section)  
#temp.head()  
temp["ts"]=pd.to_datetime(temp["timestamp"])  
temp=temp.set_index("ts")  
temp=temp.tz_localize('UTC').tz_convert('Asia/Kolkata')  
  
/home/sampad/Desktop/RBCCPS/lib/python2.7/site-packages/ipykernel_launcher  
r.py:3: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy  
This is separate from the ipykernel package so we can avoid doing imports until
```

```
In [10]: temp.loc[:, "acc"] = np.sqrt(temp["data.ax"]**2 + temp["data.az"]**2)
```

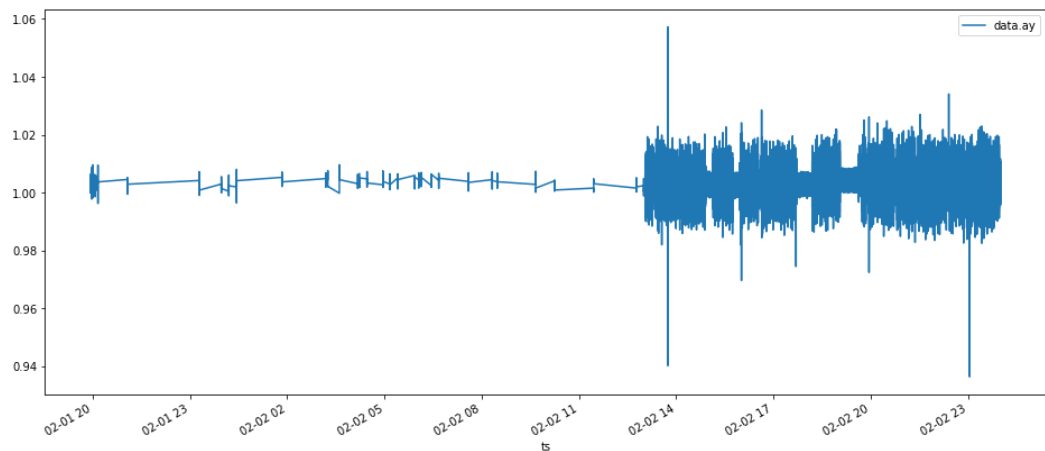
```
In [11]: temp[["data.ax","data.az","acc"]].plot()
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1d145a55d0>
```



```
In [12]: temp[["data.ay"]].plot()
```

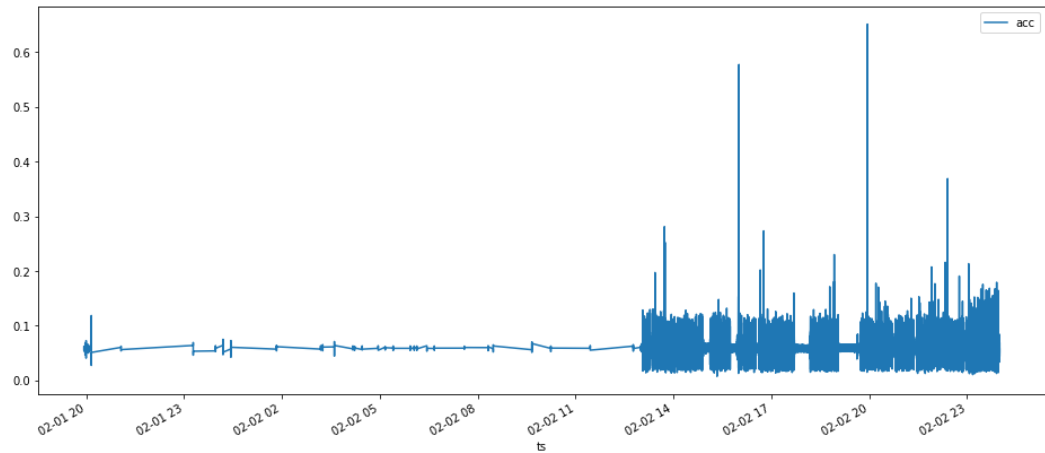
```
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1d21a1a9d0>
```



```
In [13]: temp=temp.drop(columns=["timestamp","data.gx","data.gy","data.gz","data.
ax","data.ay","data.az"])
```

```
In [14]: #temp=temp.head(75*60*60)
temp.plot()
```

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1d119fb350>



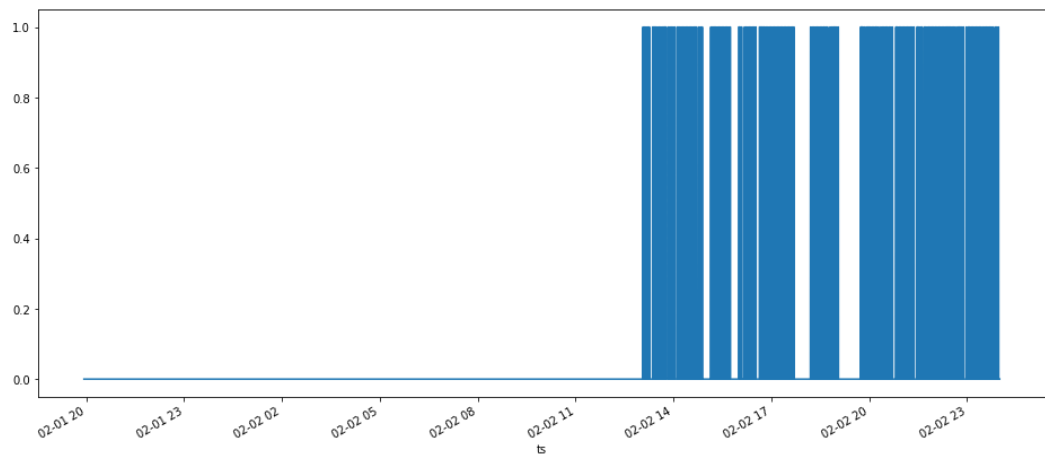
```
In [84]: x=temp.rolling(75*30).std()
```

```
In [ ]: x.plot()
```

```
In [96]: x['acc_clipped'] = np.where(x['acc']>=0.003, 1,0)
```

```
In [89]: x["acc_clipped"].plot()
```

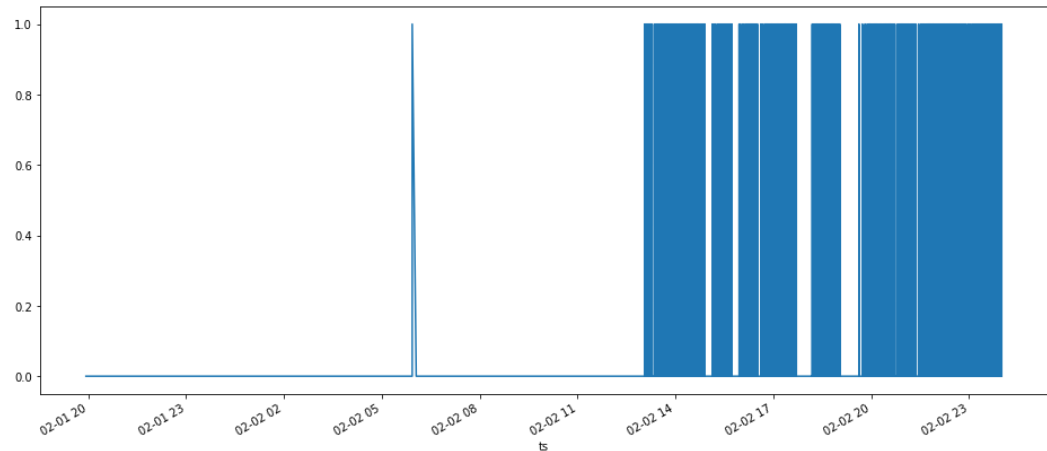
Out[89]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1c50aa8590>



```
In [97]: x['change'] = x['acc_clipped'] - x['acc_clipped'].shift(1)
x['change'] = np.where(x['change']>0, 1,0)
```

```
In [98]: x["change"].plot()
```

```
Out[98]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1ae9c42810>
```

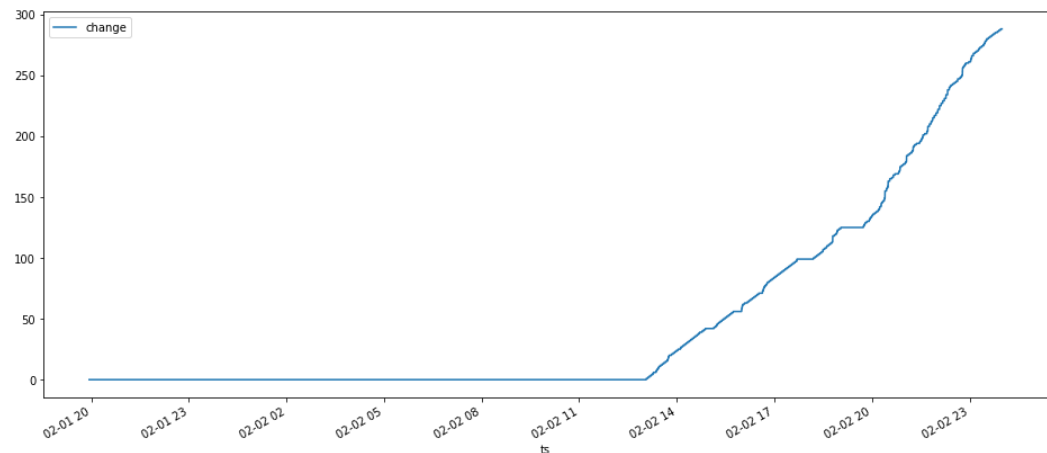


```
In [99]: y=x[["change"]]
```

```
In [100]: z=y.cumsum()
```

```
In [71]: z.plot()
```

```
Out[71]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1b5de9f850>
```



```
In [101]: y.change.sum()
```

```
Out[101]: 424
```

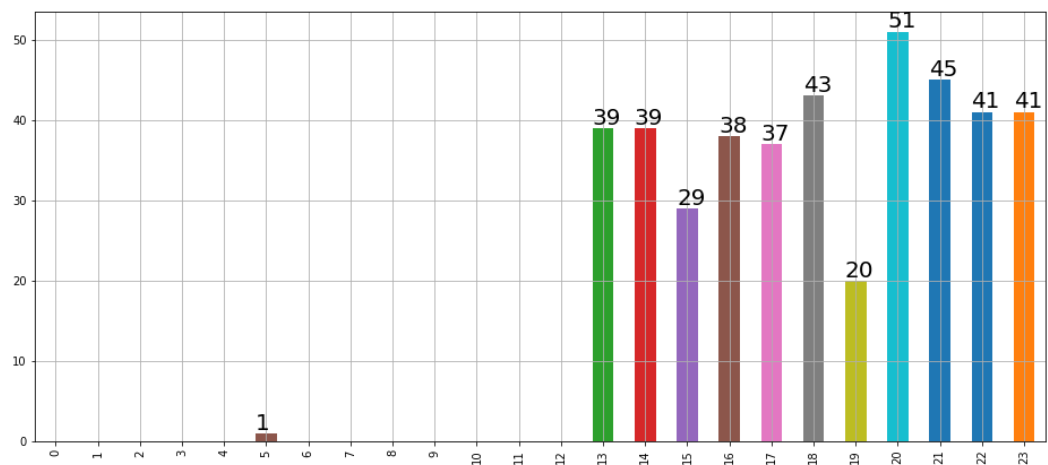
```
In [102]: dformat= ES_INDEX_DATE + " %s:%s"
npcb=[]
for hr in range(0,24):
    start=dformat%(hr,"00:00")
    end=dformat%(hr,"59:59")
    zz=y.loc[start:end]
    ww=zz.change.sum()
    #print ww
    npcb.append(ww)
    print [hr,ww],

print "\n"
print "TOTAL--->",sum(npcb)
disection=y.loc[dformat("00","00:00"):dformat("23","59:59")]
print disection.change.sum()

[0, 0] [1, 0] [2, 0] [3, 0] [4, 0] [5, 1] [6, 0] [7, 0] [8, 0] [9, 0] [10, 0] [11, 0] [12, 0] [13, 39] [14, 39] [15, 29] [16, 38] [17, 37] [18, 43] [19, 20] [20, 51] [21, 45] [22, 41] [23, 41]

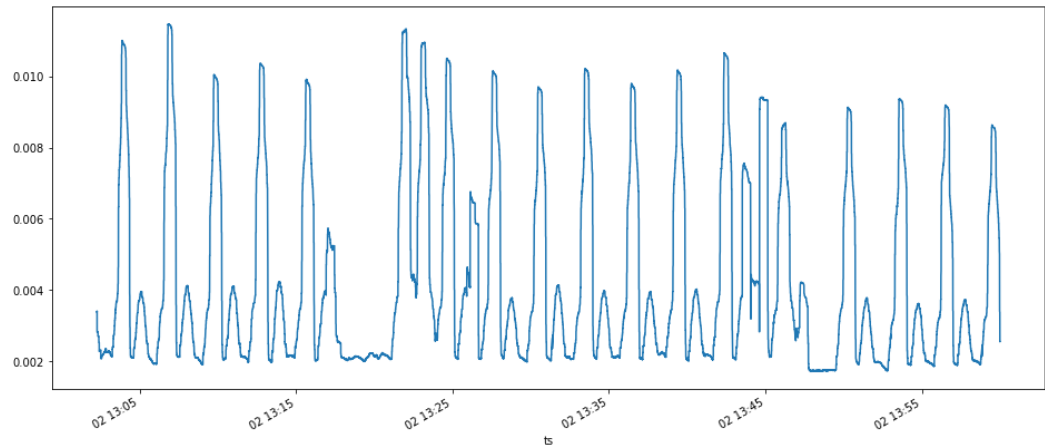
TOTAL---> 424
424
```

```
In [103]: ax=pd.Series(npcb).plot.bar(grid=True)
for i in ax.patches:
    # get_x pulls left or right; get_height pushes up or down
    if i.get_height():
        ax.text(i.get_x(), i.get_height()+0.5, i.get_height() ,fontsize=20,color='black')
```



```
In [87]: _hour=13  
x.loc[dformat%(_hour,"00:00"):dformat%(_hour,"59:59")].acc.plot()
```

```
Out[87]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1c50b04310>
```



```
In [104]: x.acc_clipped.loc[dformat%(_hour,"00:00"):dformat%(_hour,"59:59")].plot()  
( )
```

```
Out[104]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1ad96595d0>
```

