data_exploration

January 26, 2017

0.1 loading data/libs

```
In [1]: import pandas as pd
        import calendar
        from bokeh.charts import output_notebook, Scatter, Bar, show, output_file, Line, BoxPlot
        from bokeh.plotting import figure
        from bokeh.io import hplot
        output_notebook()

In [2]: INPUT="data/device_failure.csv"
        dataset = pd.read_csv(INPUT,index_col=[0,1],parse_dates=[0])

        label_dset = dataset[["failure"]]
```

0.1.1 checking devices

```
In [3]: total_failures_per_device = label_dset.groupby(level=1).agg(sum)
        total_failures_per_device["failure"].value_counts()

Out[3]: 0    1062
        1     106
        Name: failure, dtype: int64
```

Each device fail at most once
~10% device failing
'only' 106 positive points

0.1.2 Checking Dates

```
In [4]: dates = label_dset.index.get_level_values(0)
        print "Range: from %s to %s" % (dates.min(), dates.max())


        total_failures_per_date = label_dset.groupby(level=0).agg(sum)
        print
        print " n failures per date"
        print str(total_failures_per_date["failure"].value_counts())
        print
```

```
        print "total: %i failures for %i days" % (total_failures_per_date["failure"].sum(),
                                                  total_failures_per_date[total_failures_per_dat
```

```
Range: from 2015-01-01 00:00:00 to 2015-11-02 00:00:00

 n failures per date
0    228
1     54
2     19
3      2
8      1
Name: failure, dtype: int64

total: 106 failures for 76 days
```

```
In [5]: from bokeh.plotting import figure
        data =total_failures_per_date.resample("M").sum()
        test = label_dset.reset_index("device").resample("M").agg(lambda d : d.nunique())
        data["n_devices"] = test["device"]
        data["failure_ratio_percent"] = data["failure"] / data["n_devices"] * 100
        data.index = (calendar.month_abbr[i] for i in data.index.month)
        l = Line(
            data["failure_ratio_percent"],
            title="failures per Month",
            ylabel="% failure",
            xlabel="month"
        )
        show(l)
```

```
In [6]: l = Line(
            data["n_devices"],
            title="n devices seen per Month",
            ylabel="n_devices",
            xlabel="month"
        )
        show(l)
```

```
In [7]: weekday_dset = total_failures_per_date.copy()
        weekday_dset.index = ["%i:%s" % (i,calendar.day_name[i]) for i in total_failures_per_dat

        per_day = weekday_dset.groupby(level=0).sum()

        print "failures per weekday"

        per_day.sort_index()
```

```
failures per weekday
```

```
Out[7]:              failure
        0:Monday          27
        1:Tuesday         18
        2:Wednesday       15
        3:Thursday        22
        4:Friday          12
        5:Saturday         8
        6:Sunday           4
```

- Long term trend with more failures in the past
- Less failures over the weekend
- The absence of weekend could be explained by maintenance hapening only during work-week (hence explaing more failures on monday

0.1.3 Per Device description

```
In [9]: import numpy as np
        dates = label_dset.swaplevel().reset_index("date")
        dd= dates["date"]
        devices = pd.DataFrame({"min_date":dd.groupby(level=0).min(),"failure":dates["failure"].
        devices["max_date"] =  dd.groupby(level=0).max()
        devices["n_lines"] = dd.groupby(level=0).count()
        devices["n_days"] = (devices["max_date"] - devices["min_date"] ) /np.timedelta64(1, 'D')
        devices["missing_values"] = devices["n_days"] - devices["n_lines"]

In [10]: devices["min_date"].value_counts()

Out[10]: 2015-01-01    1163
         2015-05-06       4
         2015-01-27       1
         Name: min_date, dtype: int64
```

0.1.4 Nb devices per month.

```
In [11]: #pd.DataFrame({"n_devices":devices["max_date"].dt.month.value_counts().sort_index()})
         montlhy_devices = pd.DataFrame({"n_devices":devices["max_date"].dt.month.value_counts()
         montlhy_devices.index = [calendar.month_abbr[i] for i in montlhy_devices.index]
         montlhy_devices

Out[11]:      n_devices
         Jan        399
         Feb         46
         Mar        184
         Apr        112
         May         72
         Jun          6
         Jul         15
         Aug        150
         Sep         38
```

```
Oct        115
Nov         31
```

0.1.5 Bucketing the n devices with missing day data

In [12]:
```
i = ( (devices["missing_values"] //20)*20).value_counts()
#i = ( (devices["missing_values"])).value_counts(bins=10)
i.index.name = "n missing days"
pd.DataFrame({"n devices":i.sort_index()})
```

Out[12]:

n missing days	n devices
-0.0	1077
20.0	26
40.0	21
60.0	8
80.0	3
100.0	28
120.0	4
140.0	1

In [13]:
```
i = devices["n_days"].value_counts(bins=10).sort_index()
i.index.name='n_days'
b = Bar(pd.DataFrame(
    {"n_devices":i}),
    xlabel="n days",
   title="devices distributed by ndays"
       )
show(b)
```

In [14]:
```
failing_devices = devices[devices["failure"]>0].index
failing_devices_t = pd.DataFrame({"failure":label_dset["failure"].unstack().filter(item
def max_date(date):
    return np.max(date)

def failing_date(date):
    data = withdate.ix[date.index]
    return data[data["failure"]>0]["date"][0]

withdate = failing_devices_t.reset_index(level=1)
max_vs_failingdates = withdate.groupby(level=0).agg( {"date": [ max_date, failing_date
max_vs_failingdates.columns = max_vs_failingdates.columns.droplevel()
max_vs_failingdates["td"] = (max_vs_failingdates["max_date"] - max_vs_failingdates["fai
print
print "dt in days between first failure and end of measurement :"
print max_vs_failingdates["td"].value_counts()
print
print "n failures"
print max_vs_failingdates["sum"].value_counts()
```

```
dt in days between first failure and end of measurement :
0.0     101
2.0       2
30.0      1
1.0       1
12.0      1
Name: td, dtype: int64

n failures
1.0    106
Name: sum, dtype: int64
```

```
In [15]: print "looking at weird failures"
         weird_devices = max_vs_failingdates[max_vs_failingdates["td"] > 0]
         weirdos = failing_devices_t.reset_index(level=1).ix[set(weird_devices.index)]
         print weirdos.set_index("date",append=True).unstack(level="device").to_string()
```

```
looking at weird failures
            failure
device     S1F0GPFZ S1F136J0 W1F0KCP2 W1F0M35B W1F11ZG9
date
2015-01-01      0.0      0.0      0.0      0.0      0.0
2015-01-02      0.0      0.0      0.0      0.0      0.0
2015-01-03      0.0      0.0      0.0      0.0      0.0
2015-01-04      0.0      0.0      0.0      0.0      0.0
2015-01-05      0.0      0.0      0.0      0.0      0.0
2015-01-06      0.0      0.0      0.0      0.0      0.0
2015-01-07      0.0      0.0      0.0      0.0      0.0
2015-01-08      0.0      0.0      0.0      0.0      0.0
2015-01-09      0.0      0.0      0.0      0.0      0.0
2015-01-10      0.0      0.0      0.0      0.0      0.0
2015-01-11      0.0      0.0      0.0      0.0      0.0
2015-01-12      0.0      0.0      0.0      0.0      0.0
2015-01-13      0.0      0.0      0.0      0.0      0.0
2015-01-14      0.0      0.0      0.0      0.0      0.0
2015-01-15      0.0      0.0      0.0      0.0      0.0
2015-01-16      0.0      0.0      0.0      0.0      0.0
2015-01-17      0.0      0.0      0.0      0.0      0.0
2015-01-18      0.0      0.0      0.0      0.0      0.0
2015-01-19      0.0      0.0      0.0      0.0      0.0
2015-01-20      0.0      0.0      0.0      0.0      0.0
2015-01-21      0.0      0.0      0.0      0.0      0.0
2015-01-22      0.0      0.0      0.0      0.0      0.0
2015-01-23      0.0      0.0      0.0      0.0      0.0
2015-01-24      0.0      0.0      0.0      0.0      0.0
2015-01-25      0.0      0.0      0.0      0.0      0.0
```

2015-01-26	0.0	0.0	0.0	0.0	0.0
2015-01-27	0.0	0.0	0.0	0.0	0.0
2015-01-28	0.0	0.0	0.0	0.0	0.0
2015-01-29	0.0	0.0	0.0	0.0	0.0
2015-01-30	0.0	0.0	0.0	0.0	0.0
2015-01-31	0.0	0.0	0.0	0.0	0.0
2015-02-01	0.0	0.0	0.0	0.0	0.0
2015-02-02	0.0	0.0	0.0	0.0	0.0
2015-02-03	0.0	0.0	0.0	0.0	0.0
2015-02-04	0.0	0.0	0.0	0.0	0.0
2015-02-05	0.0	0.0	0.0	0.0	0.0
2015-02-06	0.0	0.0	0.0	0.0	0.0
2015-02-07	0.0	0.0	0.0	0.0	0.0
2015-02-08	0.0	0.0	0.0	0.0	0.0
2015-02-09	0.0	0.0	0.0	0.0	0.0
2015-02-10	0.0	0.0	0.0	0.0	0.0
2015-02-11	0.0	0.0	0.0	0.0	0.0
2015-02-12	0.0	0.0	0.0	0.0	0.0
2015-02-13	0.0	0.0	0.0	0.0	0.0
2015-02-14	0.0	0.0	0.0	0.0	0.0
2015-02-15	0.0	0.0	0.0	0.0	0.0
2015-02-16	0.0	0.0	0.0	0.0	0.0
2015-02-17	0.0	0.0	0.0	0.0	0.0
2015-02-18	0.0	0.0	0.0	0.0	0.0
2015-02-19	0.0	0.0	0.0	0.0	0.0
2015-02-20	0.0	0.0	0.0	0.0	0.0
2015-02-21	0.0	0.0	0.0	0.0	0.0
2015-02-22	0.0	0.0	0.0	0.0	0.0
2015-02-23	0.0	0.0	0.0	0.0	0.0
2015-02-24	0.0	0.0	0.0	0.0	0.0
2015-02-25	0.0	0.0	0.0	0.0	0.0
2015-02-26	0.0	0.0	0.0	0.0	0.0
2015-02-27	0.0	0.0	0.0	0.0	0.0
2015-02-28	0.0	0.0	0.0	0.0	0.0
2015-03-01	0.0	0.0	0.0	0.0	0.0
2015-03-02	0.0	0.0	0.0	0.0	0.0
2015-03-03	0.0	0.0	0.0	0.0	0.0
2015-03-04	0.0	0.0	0.0	0.0	0.0
2015-03-05	0.0	0.0	0.0	0.0	0.0
2015-03-06	0.0	0.0	0.0	0.0	0.0
2015-03-07	0.0	0.0	0.0	0.0	0.0
2015-03-08	0.0	0.0	0.0	0.0	0.0
2015-03-09	0.0	0.0	0.0	0.0	0.0
2015-03-10	0.0	0.0	0.0	0.0	0.0
2015-03-11	0.0	0.0	0.0	0.0	0.0
2015-03-12	0.0	0.0	0.0	0.0	0.0
2015-03-13	0.0	0.0	0.0	0.0	0.0
2015-03-14	0.0	0.0	0.0	0.0	0.0

2015-03-15	0.0	0.0	0.0	0.0	0.0
2015-03-16	0.0	0.0	0.0	0.0	0.0
2015-03-17	0.0	0.0	0.0	0.0	0.0
2015-03-18	0.0	0.0	0.0	0.0	0.0
2015-03-19	0.0	0.0	0.0	0.0	0.0
2015-03-20	0.0	0.0	0.0	0.0	0.0
2015-03-21	0.0	0.0	0.0	0.0	0.0
2015-03-22	0.0	0.0	0.0	0.0	0.0
2015-03-23	0.0	0.0	0.0	0.0	0.0
2015-03-24	0.0	0.0	0.0	0.0	0.0
2015-03-25	0.0	0.0	0.0	0.0	0.0
2015-03-26	0.0	0.0	0.0	0.0	0.0
2015-03-27	0.0	0.0	0.0	0.0	0.0
2015-03-28	0.0	0.0	0.0	0.0	0.0
2015-03-29	0.0	0.0	0.0	0.0	0.0
2015-03-30	0.0	0.0	0.0	0.0	0.0
2015-03-31	0.0	0.0	0.0	0.0	0.0
2015-04-01	0.0	0.0	0.0	0.0	0.0
2015-04-02	0.0	0.0	0.0	0.0	0.0
2015-04-03	0.0	0.0	0.0	0.0	0.0
2015-04-04	0.0	0.0	0.0	0.0	0.0
2015-04-05	0.0	0.0	0.0	0.0	0.0
2015-04-06	0.0	0.0	0.0	0.0	0.0
2015-04-07	0.0	0.0	0.0	0.0	0.0
2015-04-08	0.0	0.0	0.0	0.0	0.0
2015-04-09	0.0	0.0	0.0	0.0	0.0
2015-04-10	0.0	0.0	0.0	0.0	0.0
2015-04-11	0.0	0.0	0.0	0.0	0.0
2015-04-12	0.0	0.0	0.0	0.0	0.0
2015-04-13	0.0	0.0	0.0	0.0	0.0
2015-04-14	0.0	0.0	0.0	0.0	0.0
2015-04-15	0.0	0.0	0.0	0.0	0.0
2015-04-16	0.0	0.0	0.0	0.0	0.0
2015-04-17	0.0	0.0	0.0	0.0	0.0
2015-04-18	0.0	0.0	0.0	0.0	0.0
2015-04-19	0.0	0.0	0.0	0.0	0.0
2015-04-20	0.0	0.0	0.0	0.0	0.0
2015-04-21	0.0	0.0	0.0	0.0	0.0
2015-04-22	0.0	0.0	0.0	0.0	0.0
2015-04-23	0.0	0.0	0.0	0.0	0.0
2015-04-24	0.0	0.0	0.0	0.0	0.0
2015-04-25	0.0	0.0	0.0	0.0	0.0
2015-04-26	0.0	0.0	0.0	0.0	0.0
2015-04-27	0.0	0.0	0.0	0.0	0.0
2015-04-28	0.0	0.0	0.0	0.0	0.0
2015-04-29	0.0	0.0	0.0	0.0	0.0
2015-04-30	0.0	0.0	0.0	0.0	0.0
2015-05-01	0.0	0.0	0.0	0.0	0.0

2015-05-02	0.0	0.0	0.0	0.0	0.0
2015-05-03	0.0	0.0	0.0	0.0	0.0
2015-05-04	0.0	0.0	0.0	0.0	0.0
2015-05-05	0.0	1.0	0.0	0.0	0.0
2015-05-06	0.0	0.0	0.0	0.0	0.0
2015-05-07	0.0	NaN	0.0	0.0	0.0
2015-05-08	0.0	NaN	0.0	0.0	0.0
2015-05-09	0.0	NaN	1.0	1.0	0.0
2015-05-10	0.0	NaN	0.0	0.0	0.0
2015-05-11	0.0	NaN	0.0	0.0	0.0
2015-05-12	0.0	NaN	NaN	NaN	0.0
2015-05-13	0.0	NaN	NaN	NaN	0.0
2015-05-14	0.0	NaN	NaN	NaN	0.0
2015-05-15	0.0	NaN	NaN	NaN	0.0
2015-05-16	0.0	NaN	NaN	NaN	0.0
2015-05-17	0.0	NaN	NaN	NaN	0.0
2015-05-18	0.0	NaN	NaN	NaN	0.0
2015-05-19	0.0	NaN	NaN	NaN	0.0
2015-05-20	0.0	NaN	NaN	NaN	0.0
2015-05-21	0.0	NaN	NaN	NaN	0.0
2015-05-22	0.0	NaN	NaN	NaN	0.0
2015-05-23	0.0	NaN	NaN	NaN	0.0
2015-05-24	0.0	NaN	NaN	NaN	0.0
2015-05-25	0.0	NaN	NaN	NaN	0.0
2015-05-26	0.0	NaN	NaN	NaN	0.0
2015-05-27	0.0	NaN	NaN	NaN	0.0
2015-05-28	0.0	NaN	NaN	NaN	0.0
2015-05-29	0.0	NaN	NaN	NaN	0.0
2015-05-30	0.0	NaN	NaN	NaN	0.0
2015-05-31	0.0	NaN	NaN	NaN	0.0
2015-06-01	0.0	NaN	NaN	NaN	0.0
2015-06-02	0.0	NaN	NaN	NaN	0.0
2015-06-03	0.0	NaN	NaN	NaN	0.0
2015-06-04	0.0	NaN	NaN	NaN	0.0
2015-06-05	0.0	NaN	NaN	NaN	0.0
2015-06-06	0.0	NaN	NaN	NaN	0.0
2015-06-07	0.0	NaN	NaN	NaN	0.0
2015-06-08	0.0	NaN	NaN	NaN	0.0
2015-06-09	0.0	NaN	NaN	NaN	0.0
2015-06-10	0.0	NaN	NaN	NaN	0.0
2015-06-11	0.0	NaN	NaN	NaN	0.0
2015-06-12	0.0	NaN	NaN	NaN	0.0
2015-06-13	0.0	NaN	NaN	NaN	0.0
2015-06-14	0.0	NaN	NaN	NaN	0.0
2015-06-15	0.0	NaN	NaN	NaN	0.0
2015-06-16	0.0	NaN	NaN	NaN	0.0
2015-06-17	0.0	NaN	NaN	NaN	0.0
2015-06-18	0.0	NaN	NaN	NaN	0.0

2015-06-19	0.0	NaN	NaN	NaN	0.0
2015-06-20	0.0	NaN	NaN	NaN	0.0
2015-06-21	0.0	NaN	NaN	NaN	0.0
2015-06-22	0.0	NaN	NaN	NaN	0.0
2015-06-23	0.0	NaN	NaN	NaN	0.0
2015-06-24	0.0	NaN	NaN	NaN	0.0
2015-06-25	0.0	NaN	NaN	NaN	0.0
2015-06-26	0.0	NaN	NaN	NaN	0.0
2015-06-27	0.0	NaN	NaN	NaN	0.0
2015-06-28	0.0	NaN	NaN	NaN	0.0
2015-06-29	0.0	NaN	NaN	NaN	0.0
2015-06-30	0.0	NaN	NaN	NaN	0.0
2015-07-01	0.0	NaN	NaN	NaN	0.0
2015-07-02	0.0	NaN	NaN	NaN	0.0
2015-07-03	0.0	NaN	NaN	NaN	0.0
2015-07-04	0.0	NaN	NaN	NaN	0.0
2015-07-05	0.0	NaN	NaN	NaN	0.0
2015-07-06	0.0	NaN	NaN	NaN	0.0
2015-07-07	0.0	NaN	NaN	NaN	0.0
2015-07-08	0.0	NaN	NaN	NaN	0.0
2015-07-09	0.0	NaN	NaN	NaN	0.0
2015-07-10	0.0	NaN	NaN	NaN	0.0
2015-07-11	0.0	NaN	NaN	NaN	0.0
2015-07-12	1.0	NaN	NaN	NaN	0.0
2015-07-13	0.0	NaN	NaN	NaN	0.0
2015-07-14	0.0	NaN	NaN	NaN	0.0
2015-07-15	0.0	NaN	NaN	NaN	0.0
2015-07-16	0.0	NaN	NaN	NaN	0.0
2015-07-17	0.0	NaN	NaN	NaN	0.0
2015-07-18	0.0	NaN	NaN	NaN	1.0
2015-07-19	0.0	NaN	NaN	NaN	0.0
2015-07-20	0.0	NaN	NaN	NaN	0.0
2015-07-21	0.0	NaN	NaN	NaN	0.0
2015-07-22	0.0	NaN	NaN	NaN	0.0
2015-07-23	0.0	NaN	NaN	NaN	0.0
2015-07-24	0.0	NaN	NaN	NaN	0.0
2015-07-25	NaN	NaN	NaN	NaN	0.0
2015-07-26	NaN	NaN	NaN	NaN	0.0
2015-07-27	NaN	NaN	NaN	NaN	0.0
2015-07-28	NaN	NaN	NaN	NaN	0.0
2015-07-29	NaN	NaN	NaN	NaN	0.0
2015-07-30	NaN	NaN	NaN	NaN	0.0
2015-07-31	NaN	NaN	NaN	NaN	0.0
2015-08-01	NaN	NaN	NaN	NaN	0.0
2015-08-02	NaN	NaN	NaN	NaN	0.0
2015-08-03	NaN	NaN	NaN	NaN	0.0
2015-08-04	NaN	NaN	NaN	NaN	0.0
2015-08-05	NaN	NaN	NaN	NaN	0.0

```
2015-08-06      NaN      NaN      NaN      NaN      0.0
2015-08-07      NaN      NaN      NaN      NaN      0.0
2015-08-08      NaN      NaN      NaN      NaN      0.0
2015-08-09      NaN      NaN      NaN      NaN      0.0
2015-08-10      NaN      NaN      NaN      NaN      0.0
2015-08-11      NaN      NaN      NaN      NaN      0.0
2015-08-12      NaN      NaN      NaN      NaN      0.0
2015-08-13      NaN      NaN      NaN      NaN      0.0
2015-08-14      NaN      NaN      NaN      NaN      0.0
2015-08-15      NaN      NaN      NaN      NaN      0.0
2015-08-16      NaN      NaN      NaN      NaN      0.0
2015-08-17      NaN      NaN      NaN      NaN      0.0
```

- Identified a list of devices, which are still measured after having failed.

Three hypothesis: - The device is still functionnal after maintenance - The failure was a fluke - The measurement thereafter are false

==> If we cannot distinguish between these hypothesis, need to remove these devices from the dataset

In [ ]: