# for a given day, get the popularity of a drive model: # cat data_Q3_2018.zip_folder/2018-07-27.csv | sed '1d' | cut -d',' -f3 | sort | uniq -c | sort -g -k1,1 # for every CSV file, get the date # find . | grep csv | while read fullpath; do echo

fullpath | sed 's/\//_/g' | sed 's/\.csv//g' | sed 's/zip_folder/_/g' | sed 's/data_//g'; done

# create a file per day containing the popularity of each model
https://stackoverflow.com/questions/17017732/changing-delimiter-of-the-uniq-command
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_fwf.html

date; find . | grep csv | while read fullpath; do
filename=`echo

fullpath | sed 's/zip_folder/_/g' | sed 's/data_//g' | sed 's/\.csv//g' | sed 's/\//_/g' | sed 's/\.//g' | sed 's/^_//g'`; cat

fullpath | sed '1d' | cut -d',' -f3 | sort | uniq -c | sort -g -k1,1 | sed 's/^ *//;s/ /,/' > count_of_models_on_

{filename}.dat; done; date

```
In [1]: import pandas
        print('pandas',pandas.__version__)
        import glob
        import pickle
        import numpy
        import seaborn
        import time
        import datetime
        import matplotlib.pyplot as plt

        pandas 0.23.4
```

```
In [2]: list_of_dat = glob.glob('data_synthesized_from_csvs/count_of_models_per_
        day/count_of_models_on_*.dat')
        print(len(list_of_dat))

        2088
```

```
In [3]: list_of_df=[]
        for path_to_dat in list_of_dat:
            date_str = path_to_dat[:-len('.dat')].split('_')[-1]
            date_as_dt = datetime.datetime.strptime(date_str, '%Y-%m-%d')
        #    print(path_to_dat)
            df = pandas.read_csv(path_to_dat,header=None)
            df.columns=[date_as_dt,'model']
            df=df.set_index('model')
            list_of_df.append(df)
```

list_of_models=[] for df in list_of_df: for model_name in df.index: list_of_models.append(model_name) list_of_models = list(set(list_of_models)) print(len(list_of_models))

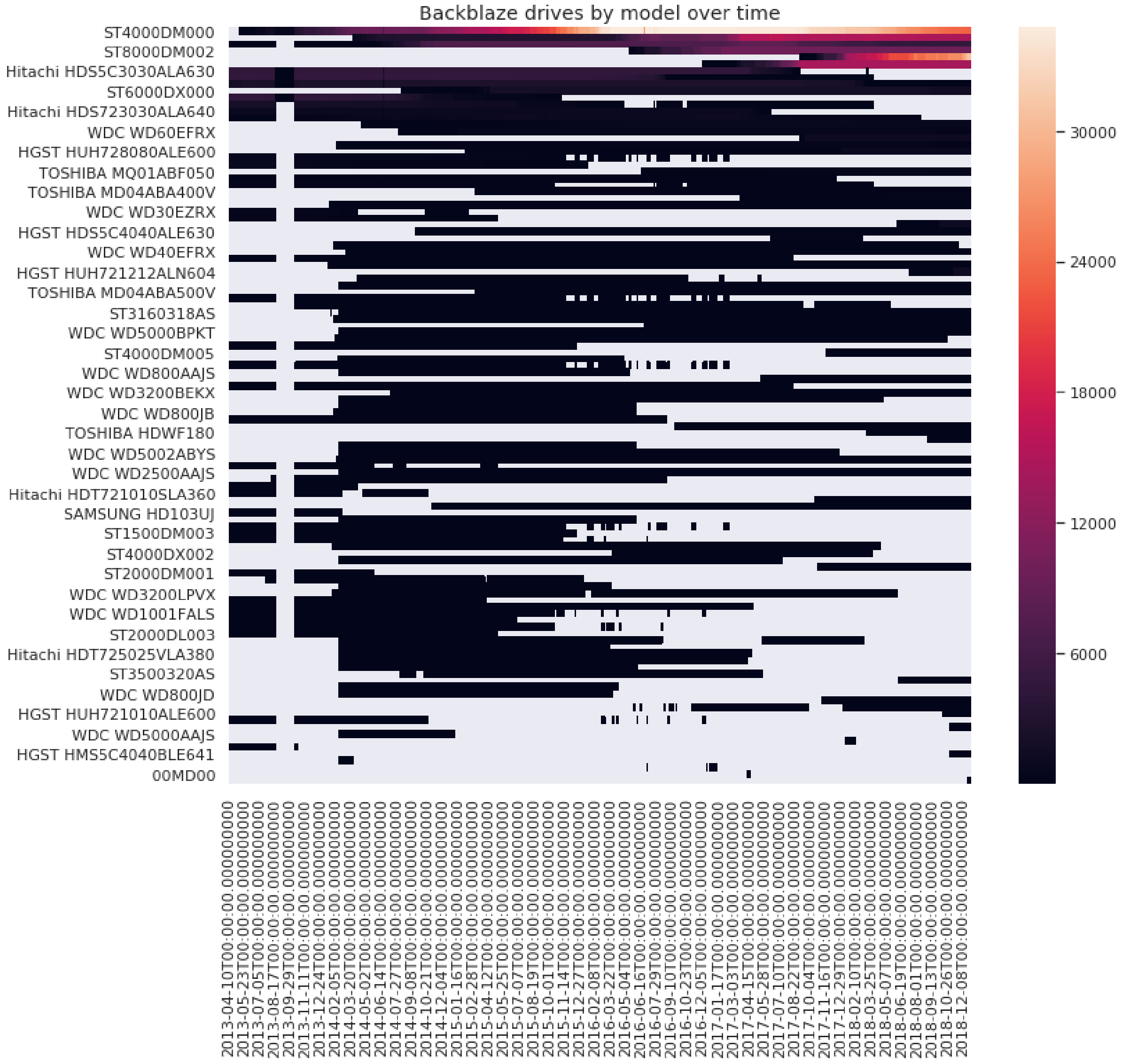
```
In [4]: df = pandas.concat(list_of_df,sort=False,axis=1) # join all the datafram
        es into a single df
        df = df.reindex(sorted(df.columns), axis=1) # order columns by calendar
         date
```

In [5]:
```
df.shape
```

Out[5]:
```
(113, 2088)
```

In [21]:
```
sorted_df = df.loc[df.sum(axis=1).sort_values(ascending=False).index]
```

In [27]:
```
seaborn.set(rc={'figure.figsize':(12,10)})
seaborn.heatmap(sorted_df);
plt.title('Backblaze drives by model over time',fontsize=14);
```


Backblaze drives by model over time
ST4000DM000
ST8000DM002
Hitachi HDS5C3030ALA630
ST6000DX000
Hitachi HDS723030ALA640
WDC WD60EFRX
HGST HUH728080ALE600
TOSHIBA MQ01ABF050
TOSHIBA MD04ABA400V
WDC WD30EZRX
HGST HDS5C4040ALE630
WDC WD40EFRX
HGST HUH721212ALN604
TOSHIBA MD04ABA500V
ST3160318AS
WDC WD5000BPKT
ST4000DM005
WDC WD800AAJS
WDC WD3200BEKX
WDC WD800JB
TOSHIBA HDWF180
WDC WD5002ABYS
WDC WD2500AAJS
Hitachi HDT721010SLA360
SAMSUNG HD103UJ
ST1500DM003
ST4000DX002
ST2000DM001
WDC WD3200LPVX
WDC WD1001FALS
ST2000DL003
Hitachi HDT725025VLA380
ST3500320AS
WDC WD800JD
HGST HUH721010ALE600
WDC WD5000AAJS
HGST HMS5C4040BLE641
00MD00
2013-04-10T00:00:00.000000000
2013-05-23T00:00:00.000000000
2013-07-05T00:00:00.000000000
2013-08-17T00:00:00.000000000
2013-09-29T00:00:00.000000000
2013-11-11T00:00:00.000000000
2013-12-24T00:00:00.000000000
2014-02-05T00:00:00.000000000
2014-03-20T00:00:00.000000000
2014-05-02T00:00:00.000000000
2014-06-14T00:00:00.000000000
2014-07-27T00:00:00.000000000
2014-09-08T00:00:00.000000000
2014-10-21T00:00:00.000000000
2014-12-04T00:00:00.000000000
2015-01-16T00:00:00.000000000
2015-02-28T00:00:00.000000000
2015-04-12T00:00:00.000000000
2015-05-25T00:00:00.000000000
2015-07-07T00:00:00.000000000
2015-08-19T00:00:00.000000000
2015-10-01T00:00:00.000000000
2015-11-14T00:00:00.000000000
2015-12-27T00:00:00.000000000
2016-02-08T00:00:00.000000000
2016-03-22T00:00:00.000000000
2016-05-04T00:00:00.000000000
2016-06-16T00:00:00.000000000
2016-07-29T00:00:00.000000000
2016-09-10T00:00:00.000000000
2016-10-23T00:00:00.000000000
2016-12-05T00:00:00.000000000
2017-01-17T00:00:00.000000000
2017-03-03T00:00:00.000000000
2017-04-15T00:00:00.000000000
2017-05-28T00:00:00.000000000
2017-07-10T00:00:00.000000000
2017-08-22T00:00:00.000000000
2017-10-04T00:00:00.000000000
2017-11-16T00:00:00.000000000
2017-12-29T00:00:00.000000000
2018-02-10T00:00:00.000000000
2018-03-25T00:00:00.000000000
2018-05-07T00:00:00.000000000
2018-06-19T00:00:00.000000000
2018-08-01T00:00:00.000000000
2018-09-13T00:00:00.000000000
2018-10-26T00:00:00.000000000
2018-12-08T00:00:00.000000000