Weight Project

April 13, 2020

In [2]:
```
#Start by cleaning up the hackers diet excel sheet (removing unhelpful rows, etc.)
import pandas as pd
df = pd.read_excel("hackdiet_db.sv.xlsx")

df_clean_columns = df.rename(columns={"2020-04-12T18:35:41Z": "weight", "Epoch": "weigh
df_clean_columns = df_clean_columns.dropna(subset=['weight'])
df_clean_columns_rows = df_clean_columns[df_clean_columns['weigh_in_date'] != 'Date']
df_clean_columns_rows_2 = df_clean_columns_rows[df_clean_columns_rows['weigh_in_date']
df_clean_columns_rows_3 = df_clean_columns_rows_2[df_clean_columns_rows_2['weigh_in_da
df_clean_columns_rows_4 = df_clean_columns_rows_3[df_clean_columns_rows_3['weigh_in_da
df_clean_columns_rows_5 = df_clean_columns_rows_4[df_clean_columns_rows_4['weigh_in_da
df_clean_columns_rows_6 = df_clean_columns_rows_5.drop(columns=['C', 'D', 'E', 'F','G'
df_clean_columns_rows_6["weight"] = pd.to_numeric(df_clean_columns_rows_6["weight"], d
# we need to convert kg to lbs; if <90. then weight *= 2.20462
df_clean_columns_rows_6.loc[df_clean_columns_rows_6['weight'] <= 90., 'weight'] *= 2.2
# print to Excel
df_clean_columns_rows_6.to_excel("HackerDietClean.xlsx", engine='xlsxwriter', index=Fa
print("done without errors")
df_clean_columns_rows_6
```

```
done without errors
```

Out[2]:

	weigh_in_date	weight
12	2008-02-08	175.000000
14	2008-02-10	175.199997
15	2008-02-11	172.800003
16	2008-02-12	171.600006
18	2008-02-14	169.000000
...	...	...
1714	2020-03-26	185.500000
1718	2020-03-30	185.500000
1719	2020-03-31	184.600006
1722	2020-04-01	183.800003
1726	2020-04-05	187.100006

[739 rows x 2 columns]

In [5]:
```
#Do MFP
import pandas as pd_mfp
df_mfp = pd_mfp.read_excel("mfp.xlsx")
df_clean_columns_mfp = df_mfp.rename(columns={"Date": "weigh_in_date", "Body Fat %": "
df_clean_columns_rows_mfp_2 = df_clean_columns_mfp.drop(columns=['B', 'C', 'D'], axis=
df_clean_columns_rows_mfp_2 = df_clean_columns_rows_mfp_2.dropna(subset=['weight'])
df_clean_columns_rows_mfp_2["weight"] = pd_mfp.to_numeric(df_clean_columns_rows_mfp_2[
df_clean_columns_rows_mfp_2.to_excel("MFPClean.xlsx", engine='xlsxwriter', index=False
print("done without errors")
df_clean_columns_rows_mfp_2
```

```
done without errors
```

Out[5]:

	weigh_in_date	weight
0	2019-01-02	184.800003
1	2019-01-03	184.399994
2	2019-01-04	184.100006
3	2019-01-05	183.300003
4	2019-01-06	183.899994
..	...	...
270	2020-04-06	188.399994
271	2020-04-07	185.000000
272	2020-04-08	184.899994
273	2020-04-10	188.399994
274	2020-04-11	185.199997

[156 rows x 2 columns]

In [4]:
```
#Do TrendWeight
import pandas as pd_tw
df_tw = pd_tw.read_excel("TrendWeight.xlsx")
df_clean_columns_tw = df_tw.rename(columns={"Date": "weigh_in_date", "WeightActual": "
df_clean_columns_rows_tw_2 = df_clean_columns_tw.drop(columns=['C', 'D', 'E', 'F', 'G'
df_clean_columns_rows_tw_2 = df_clean_columns_rows_tw_2.dropna(subset=['weight'])
df_clean_columns_rows_tw_2["weight"] = pd_tw.to_numeric(df_clean_columns_rows_tw_2["we
df_clean_columns_rows_tw_2.to_excel("TrendWeightClean.xlsx", engine='xlsxwriter', inde
print("done without errors")
df_clean_columns_rows_tw_2
```

```
done without errors
```

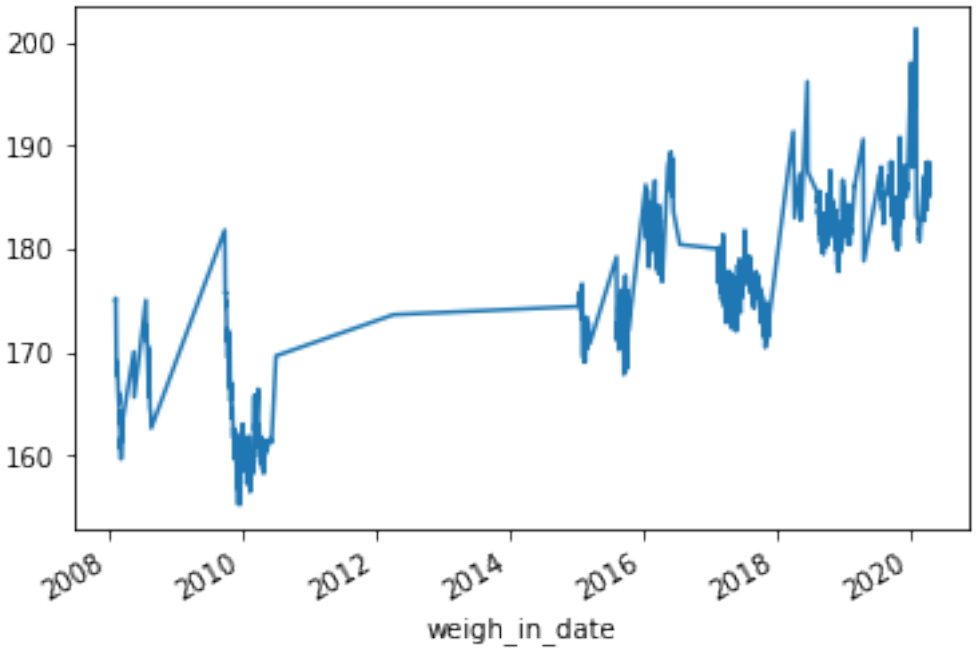
Out[4]:

	weigh_in_date	weight
0	2018-08-10	185.440002
1	2018-08-11	185.429993
2	2018-08-12	184.539993
3	2018-08-14	185.160004
4	2018-08-15	184.100006

```
..            ...         ...
242    2020-04-06  188.380005
243    2020-04-07  184.990005
244    2020-04-08  184.940002
245    2020-04-10  188.389999
246    2020-04-11  185.210007

[247 rows x 2 columns]
```

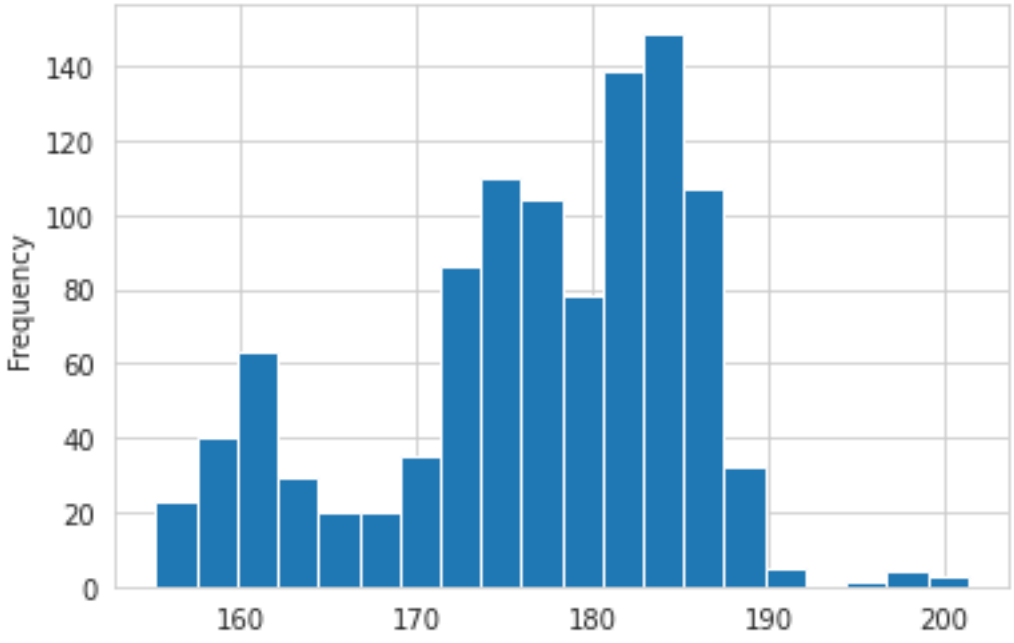
In [6]:
```
#concatinate all MFP and Hackers Diet sheets so there are no duplicates
import pandas as pd_clean_join
mfp_clean_join = pd_clean_join.read_excel("MFPClean.xlsx")
hd_clean_join = pd_clean_join.read_excel("HackerDietClean.xlsx")
tw_clean_join = pd_clean_join.read_excel("TrendWeightClean.xlsx")

outer_join_first = mfp_clean_join.merge(hd_clean_join, how="outer")
outer_join_second = outer_join_first.merge(tw_clean_join, how="outer")
#Extra effort to make sure the outer join worked appropriately
outer_join_second.drop_duplicates()
#Write the concatenated file to disk
outer_join_second.to_excel("ThreeCleanSheets.xlsx",index=False)
```

In [5]:
```
#Now let's analyze ourself!
import pandas as pd_a
from fbprophet import Prophet
import numpy as np

data = pd_a.read_excel("ThreeCleanSheets.xlsx")
data["weigh_in_date"] = pd_a.to_datetime(data["weigh_in_date"])
data = data.sort_values(by="weigh_in_date")
data.set_index('weigh_in_date')['weight'].plot();

#data.describe()
```

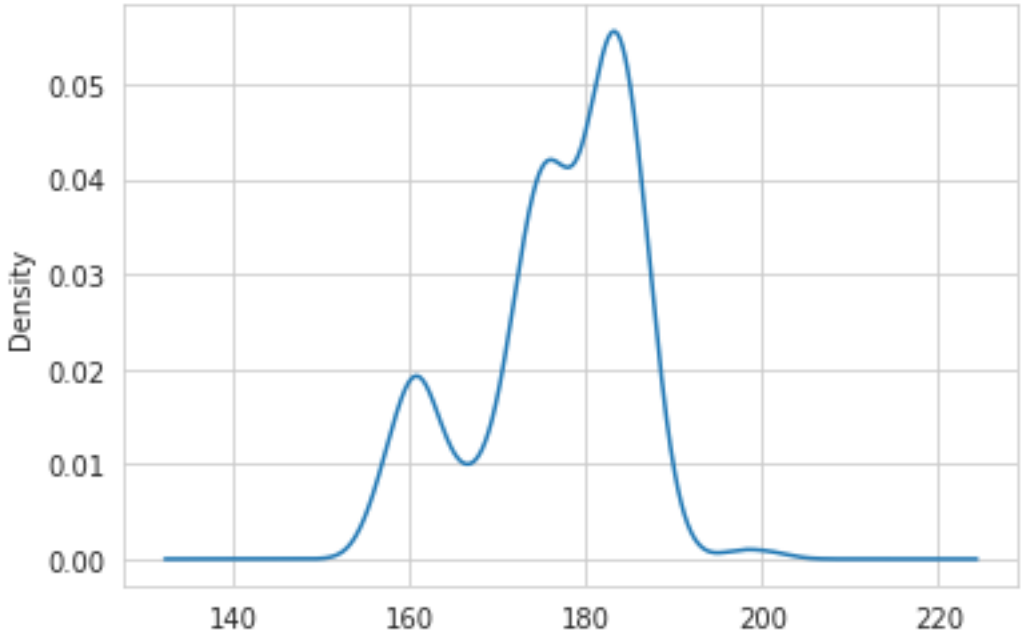



```
In [61]: data['weight'].plot(kind='hist', bins=20)
```

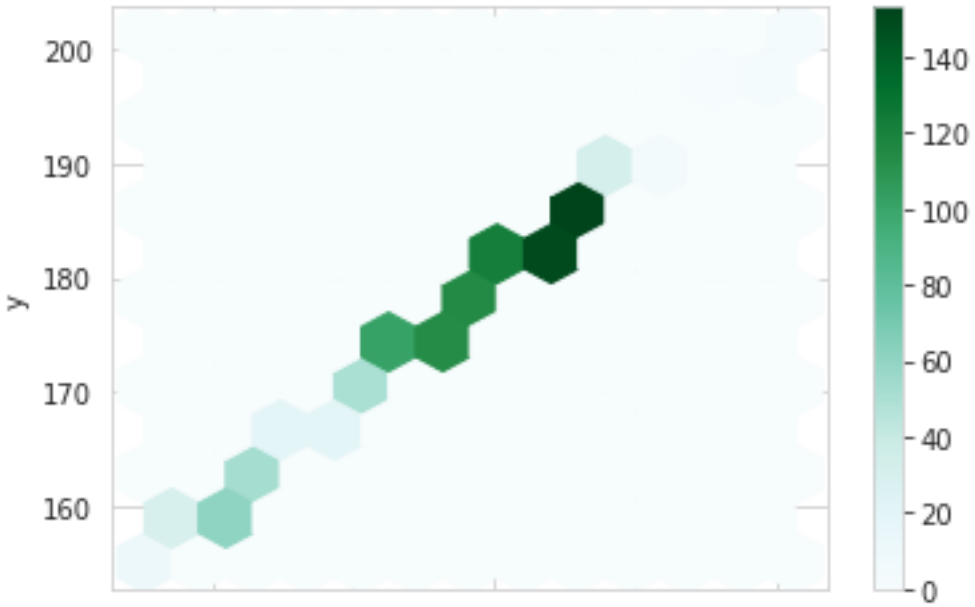
```
Out[61]: <matplotlib.axes._subplots.AxesSubplot at 0x7f5226277390>
```



In [62]:
```
data.set_index('weigh_in_date')['weight'].plot(kind='kde');
```



In [94]:
```
df = pd_a.DataFrame({'x': data["weight"], 'y': data["weight"]})
ax = df.plot.hexbin(x='x', y='y', gridsize=12)
```



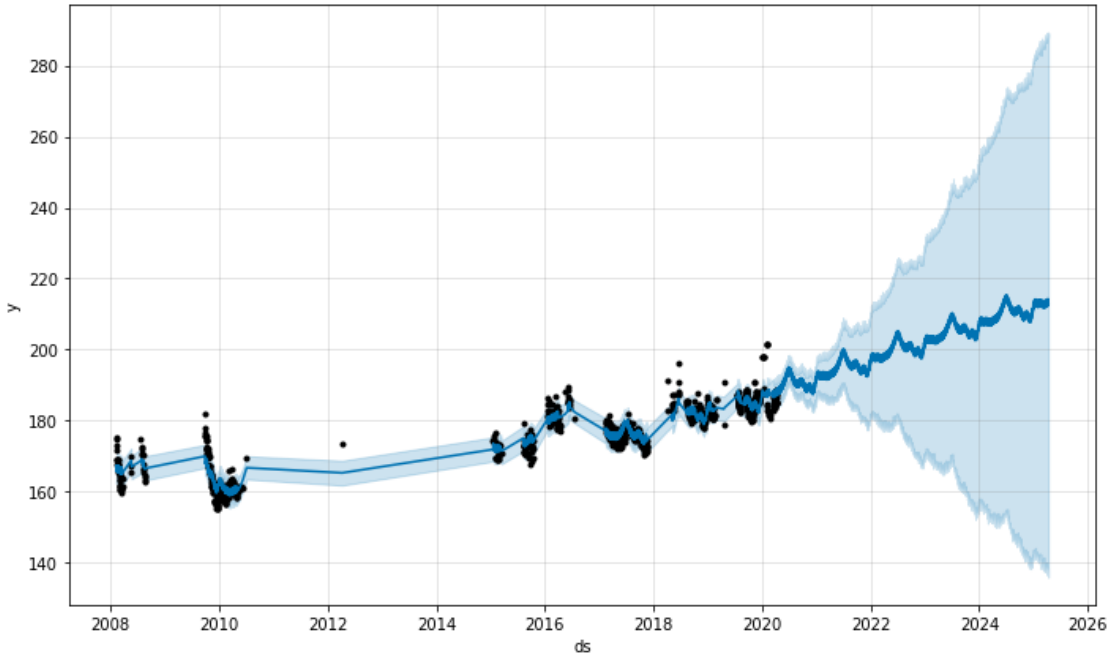
In [105]:
```
data.describe(percentiles=[0,1/10, 2/10, 3/10,4/10,5/10,6/10,7/10,8/10,9/10])
```

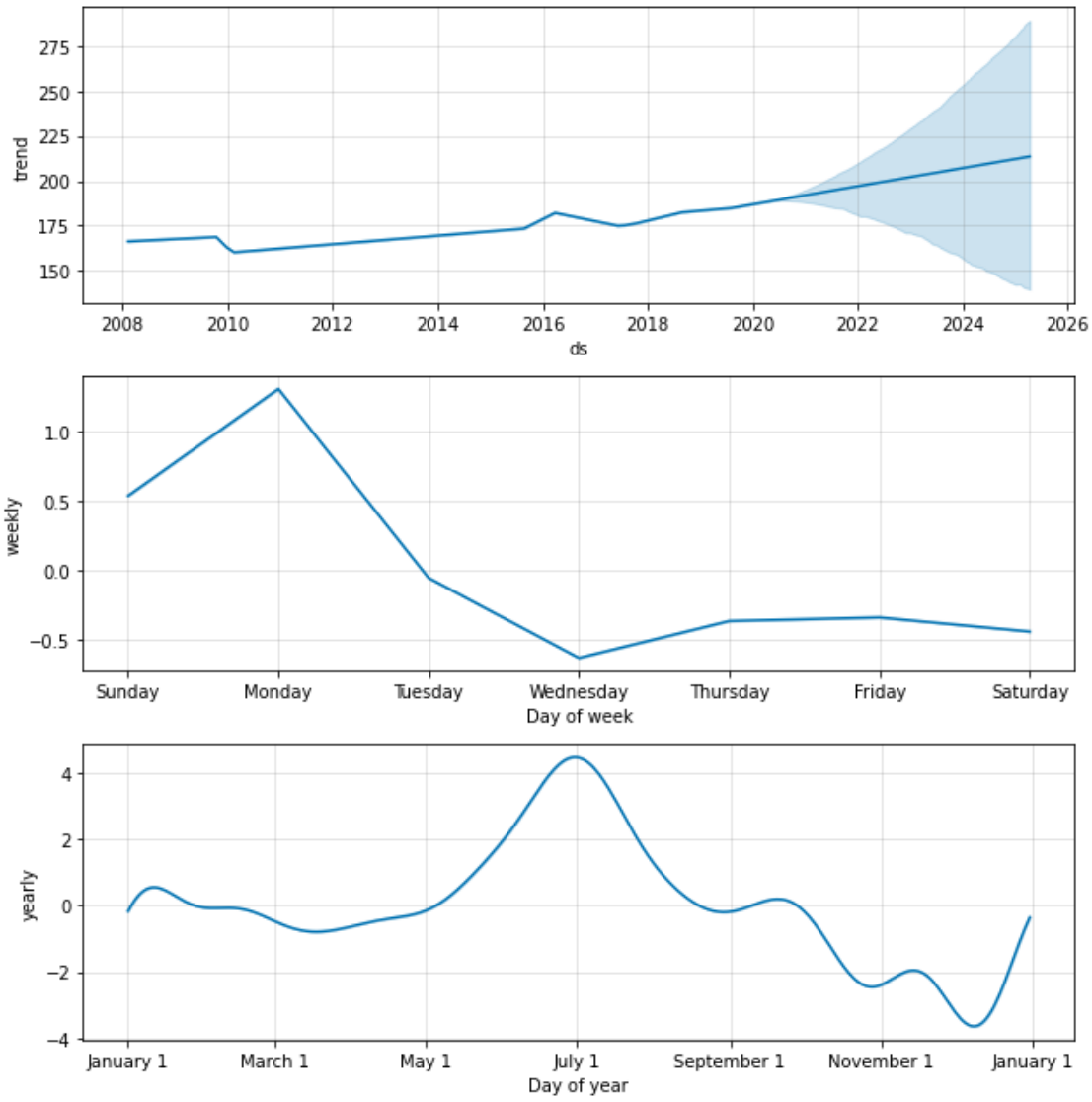
Out[105]:

	weight
count	1048.000000
mean	176.656119
std	8.785274
min	155.199997
0%	155.199997
10%	161.399994
20%	170.199997
30%	173.600006
40%	175.800003
50%	178.199997
60%	181.007999
70%	182.687003
80%	184.199997
90%	185.800003
max	201.330002

In [16]:
```
#predict the dad bod of 200 lbs.

        # Prophet requires columns ds (Date) and y (value)
        m = data.rename(columns={'weigh_in_date': 'ds', 'weight': 'y'})
        n = Prophet()
        n.fit(m)
        future = n.make_future_dataframe(periods=1825)
        future.tail()
        forecast = n.predict(future)
        forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
        fig1 = n.plot(forecast)
        fig2 = n.plot_components(forecast)

        # Make the prophet model and fit on the data
        #gm_prophet = fbprophet.Prophet(changepoint_prior_scale=0.15, daily_seasonality=False
        #gm_prophet.fit(gm)
```

```
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to overrid
```


280
260
240
220
200
180
160
140
y
2008
2010
2012
2014
2016
2018
2020
2022
2024
2026
ds



In [17]:
```
from fbprophet.plot import plot_plotly
import plotly.offline as py
py.init_notebook_mode()

fig = plot_plotly(n, forecast)  # This returns a plotly Figure
py.iplot(fig)
```

In [ ]: