analysis

January 9, 2021

[37]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import kurtosis
%matplotlib inline
```

[38]:
```
dividend_df = pd.read_csv('logs/dividend_log.txt', header=None)
dividend_df = pd.DataFrame(dividend_df[0].str.split().tolist()).astype(int)
dividend_df.columns = ['return_value', 'dividend_size', 'divisor_size',␣
 ↪'runtime']
dividend_df
```

[38]:

	return_value	dividend_size	divisor_size	runtime
0	1	10	10	0
1	1	12	10	0
2	1	12	10	0
3	1	13	10	0
4	1	15	10	0
…	…	…	…	…
9985	1	9995	10	18
9986	1	9997	10	18
9987	1	9998	10	18
9988	1	9999	10	19
9989	1	10000	10	18

[9990 rows x 4 columns]

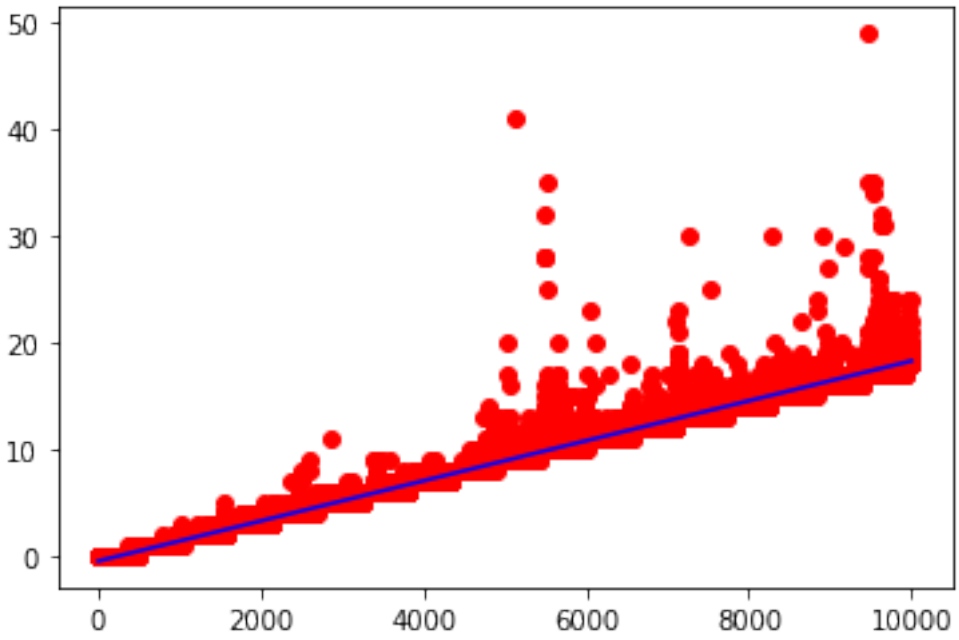
[39]:
```
def predict(poly, data, order=2):
    return sum([poly[i]*(data**(2 - i)) for i in range(len(poly) - 1)]) +␣
 ↪poly[-1]
```

[40]:
```
fit = np.polyfit(dividend_df['dividend_size'], dividend_df['runtime'], 2)
order_map = dict(zip(fit, reversed(list(range(len(fit))))))
order_map = {round(term, 5): order for term, order in order_map.items()}
order_map = {k: v for k, v in order_map.items() if abs(k) > 0}
print(' + '.join(f'{term}{f"x^{order}" if order != 0 else ""}' for term, order␣
 ↪in order_map.items()))
```

```
0.00189x^1 + -0.38735
```

[41]:
```
dividend_df['regression'] = dividend_df['dividend_size'].map(lambda x:␣
 ↪predict(fit, x)).tolist()
plt.scatter(dividend_df['dividend_size'], dividend_df['runtime'], color='red')
plt.plot(dividend_df['dividend_size'], dividend_df['regression'], color='blue')
```

[41]:
```
[<matplotlib.lines.Line2D at 0x12622da90>]
```



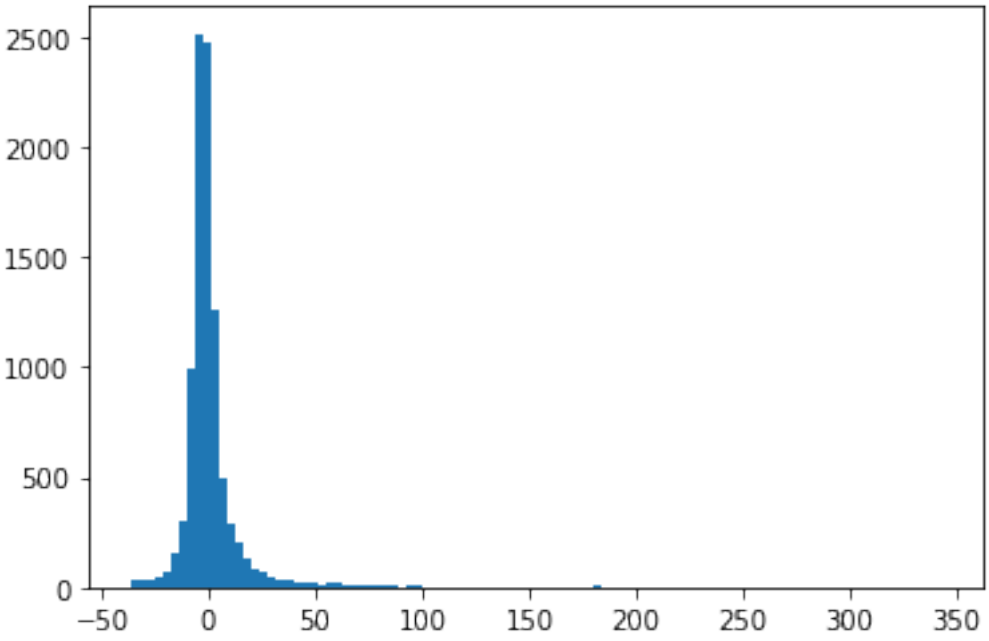
[42]:
```
dividend_dist = dividend_df.loc[dividend_df['runtime'] != 0]
dividend_dist['pct_err'] = 100*(dividend_df['runtime'] -␣
 ↪dividend_df['regression'])/dividend_df['regression']
plt.hist(dividend_dist['pct_err'], bins=100)
```

```
<ipython-input-42-69cd2f3aa961>:2: 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: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  dividend_dist['pct_err'] = 100*(dividend_df['runtime'] -
dividend_df['regression'])/dividend_df['regression']
```

[42]:
```
(array([2.900e+01, 3.100e+01, 3.600e+01, 5.200e+01, 7.100e+01, 1.550e+02,
        3.040e+02, 9.910e+02, 2.512e+03, 2.476e+03, 1.262e+03, 4.940e+02,
        2.930e+02, 2.050e+02, 1.300e+02, 8.600e+01, 6.700e+01, 5.100e+01,
```

```
       3.000e+01, 3.900e+01, 2.500e+01, 2.500e+01, 1.700e+01, 1.100e+01,
       1.600e+01, 1.900e+01, 1.000e+01, 8.000e+00, 1.000e+01, 1.300e+01,
       1.000e+01, 9.000e+00, 4.000e+00, 0.000e+00, 5.000e+00, 6.000e+00,
       2.000e+00, 3.000e+00, 1.000e+00, 1.000e+00, 0.000e+00, 2.000e+00,
       1.000e+00, 1.000e+00, 0.000e+00, 0.000e+00, 1.000e+00, 0.000e+00,
       0.000e+00, 2.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 1.000e+00,
       0.000e+00, 0.000e+00, 0.000e+00, 4.000e+00, 0.000e+00, 0.000e+00,
       0.000e+00, 1.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
       0.000e+00, 0.000e+00, 1.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
       0.000e+00, 0.000e+00, 0.000e+00, 1.000e+00, 0.000e+00, 0.000e+00,
       0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
       0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
       0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
       0.000e+00, 0.000e+00, 0.000e+00, 1.000e+00]),
array([-36.61805298, -32.81246539, -29.00687779, -25.20129019,
       -21.3957026 , -17.590115  , -13.7845274 ,  -9.97893981,
        -6.17335221,  -2.36776461,   1.43782298,   5.24341058,
         9.04899818,  12.85458577,  16.66017337,  20.46576097,
        24.27134856,  28.07693616,  31.88252376,  35.68811135,
        39.49369895,  43.29928655,  47.10487414,  50.91046174,
        54.71604934,  58.52163693,  62.32722453,  66.13281213,
        69.93839972,  73.74398732,  77.54957492,  81.35516251,
        85.16075011,  88.96633771,  92.7719253 ,  96.5775129 ,
       100.3831005 , 104.18868809, 107.99427569, 111.79986329,
       115.60545088, 119.41103848, 123.21662607, 127.02221367,
       130.82780127, 134.63338886, 138.43897646, 142.24456406,
       146.05015165, 149.85573925, 153.66132685, 157.46691444,
       161.27250204, 165.07808964, 168.88367723, 172.68926483,
       176.49485243, 180.30044002, 184.10602762, 187.91161522,
       191.71720281, 195.52279041, 199.32837801, 203.1339656 ,
       206.9395532 , 210.7451408 , 214.55072839, 218.35631599,
       222.16190359, 225.96749118, 229.77307878, 233.57866638,
       237.38425397, 241.18984157, 244.99542917, 248.80101676,
       252.60660436, 256.41219196, 260.21777955, 264.02336715,
       267.82895475, 271.63454234, 275.44012994, 279.24571753,
       283.05130513, 286.85689273, 290.66248032, 294.46806792,
       298.27365552, 302.07924311, 305.88483071, 309.69041831,
       313.4960059 , 317.3015935 , 321.1071811 , 324.91276869,
       328.71835629, 332.52394389, 336.32953148, 340.13511908,
       343.94070668]),
<a list of 100 Patch objects>)
```



[43]:
```
kurtosis(dividend_dist['pct_err'])
```

[43]:
```
70.66361836124753
```

[44]:
```
divisor_df = pd.read_csv('logs/divisor_log.txt', header=None)
divisor_df = pd.DataFrame(divisor_df[0].str.split().tolist()).astype(int)
divisor_df.columns = ['return_value', 'dividend_size', 'divisor_size',␣
 ↪'runtime']
divisor_df
```

[44]:

	return_value	dividend_size	divisor_size	runtime
0	1	10000	2	6
1	1	10000	3	7
2	1	10000	4	9
3	1	10000	5	10
4	1	10000	6	12
…	…	…	…	…
9994	1	10000	9996	92
9995	1	10000	9997	92
9996	1	10000	9998	89
9997	1	10000	9999	89
9998	0	10000	10000	0

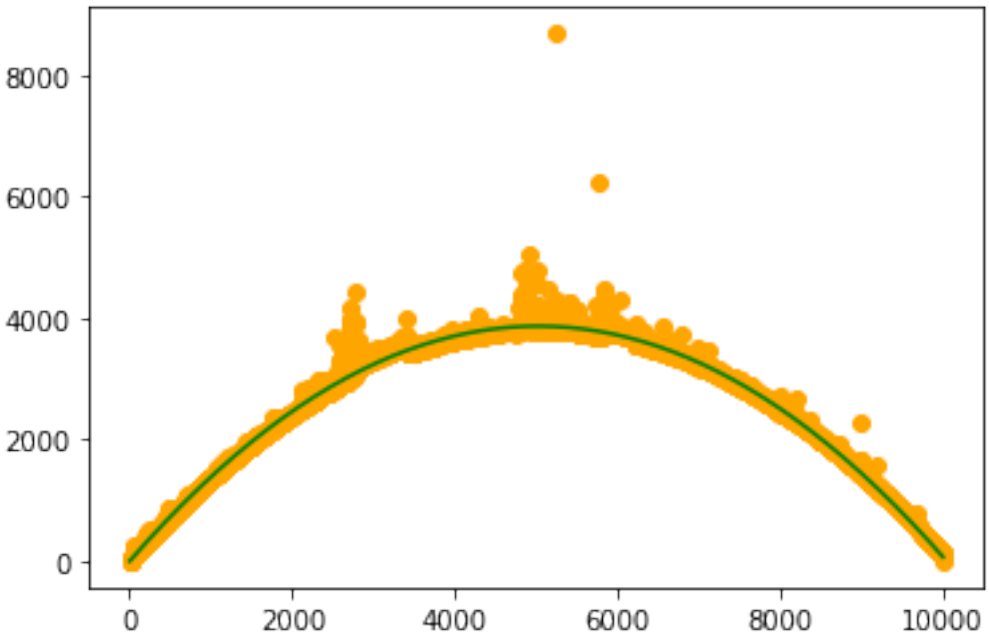
[9999 rows x 4 columns]

[45]:
```
fit = np.polyfit(divisor_df['divisor_size'], divisor_df['runtime'], 2)
order_map = dict(zip(fit, reversed(list(range(len(fit))))))
order_map = {round(term, 5): order for term, order in order_map.items()}
order_map = {k: v for k, v in order_map.items() if abs(k) > 0}
print(' + '.join(f'{term}{f"x^{order}" if order != 0 else ""}' for term, order␣
 ↪in order_map.items()))
```

```
-0.00015x^2 + 1.54136x^1 + -11.27207
```

[46]:
```
divisor_df['regression'] = divisor_df['divisor_size'].map(lambda x:␣
 ↪predict(fit, x)).tolist()
plt.scatter(divisor_df['divisor_size'], divisor_df['runtime'], color='orange')
plt.plot(divisor_df['divisor_size'], divisor_df['regression'], color='green')
```

[46]: [<matplotlib.lines.Line2D at 0x126a35b50>]



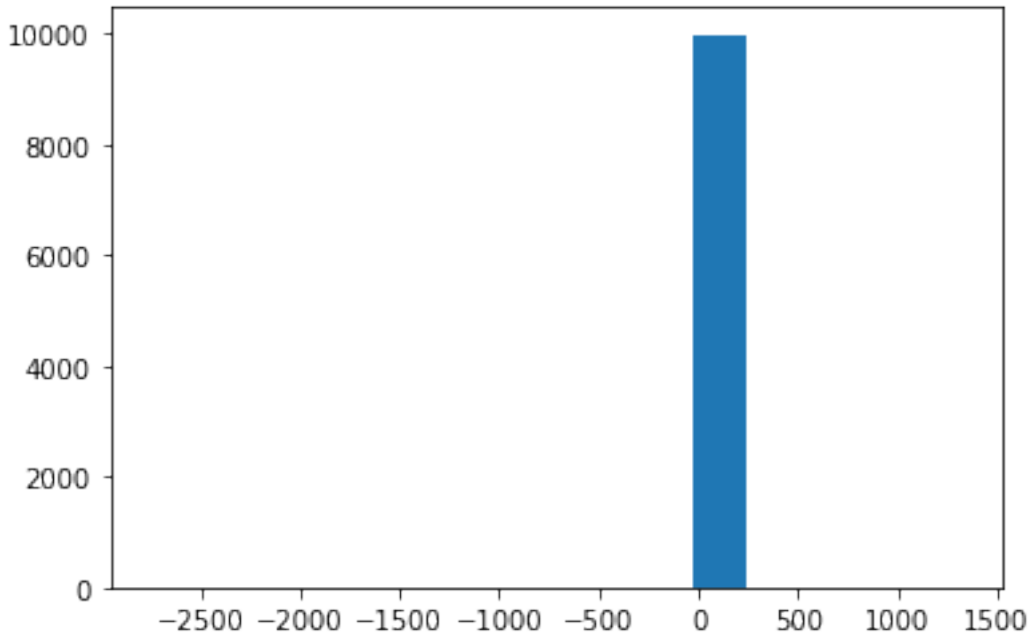
[47]:
```
divisor_dist = divisor_df.loc[divisor_df['runtime'] != 0]
divisor_dist['pct_err'] = 100*(divisor_dist['runtime'] -␣
 ↪divisor_dist['regression'])/divisor_dist['regression']
plt.hist(divisor_dist['pct_err'], bins=15)
```

```
<ipython-input-47-c27847415f69>:2: 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: https://pandas.pydata.org/pandas-
```

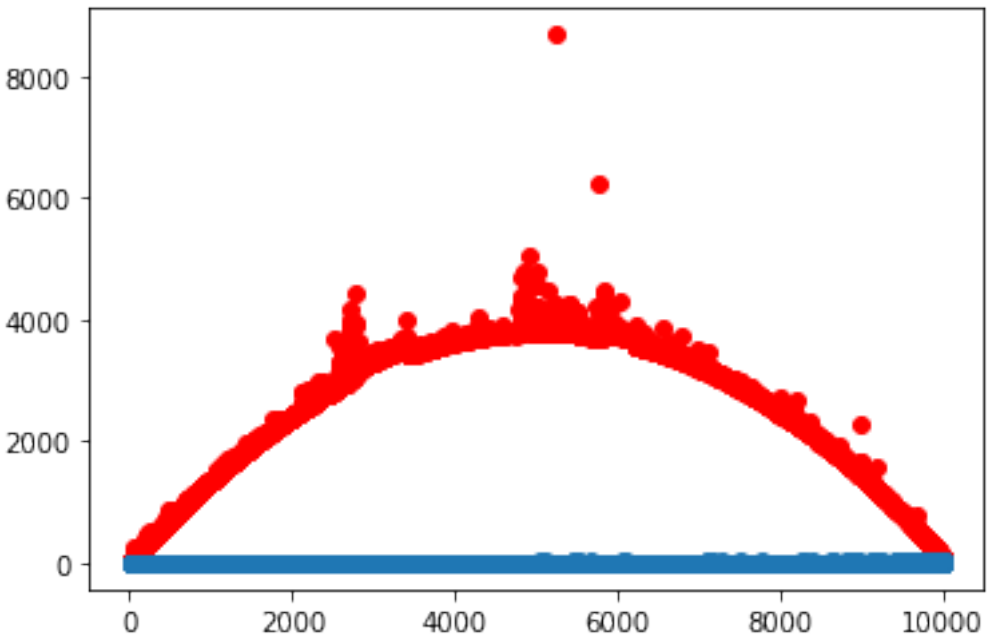
```
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  divisor_dist['pct_err'] = 100*(divisor_dist['runtime'] -
divisor_dist['regression'])/divisor_dist['regression']
```

[47]:
```
(array([1.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
        0.000e+00, 1.000e+00, 1.000e+00, 3.000e+00, 9.988e+03, 2.000e+00,
        1.000e+00, 0.000e+00, 1.000e+00]),
 array([-2752.78850115, -2480.60798028, -2208.4274594 , -1936.24693852,
        -1664.06641764, -1391.88589677, -1119.70537589,  -847.52485501,
         -575.34433414,  -303.16381326,   -30.98329238,   241.1972285 ,
          513.37774937,   785.55827025,  1057.73879113,  1329.919312  ]),
 <a list of 15 Patch objects>)
```



[48]:
```
plt.scatter(divisor_df['divisor_size'], divisor_df['runtime'], color='red')
plt.scatter(dividend_df['dividend_size'], dividend_df['runtime'])
# plt.plot(dividend_df['dividend_size'], /divisor_df['regression'],␣
 ↪color='green')
```

[48]:
```
<matplotlib.collections.PathCollection at 0x1252c27f0>
```


8000
6000
4000
2000
0
0
2000
4000
6000
8000
10000

[ ]: