2021-04-05-summarize-logit-transformed-frequencies-for-501Y.V1

April 5, 2021

1 Summary of logit transformed frequencies for 501Y.V1 (B.1.1.7)

[1]:
```
from augur.frequency_estimators import logit_transform
import matplotlib.pyplot as plt
import numpy as np
import pprint
from scipy.stats import linregress
from scipy.special import logit

%matplotlib inline
```

[2]:
```
dpi = 130
```

1.1 Define pivots and frequencies for 501Y.V1

[3]:
```
pivots = np.array([2020.011 , 2020.0301, 2020.0493, 2020.0684, 2020.0861, 2020.
↪1052,
       2020.1244, 2020.1436, 2020.1667, 2020.1858, 2020.205 , 2020.2242,
       2020.2433, 2020.261 , 2020.2801, 2020.2993, 2020.3184, 2020.3388,
       2020.358 , 2020.3771, 2020.3963, 2020.4155, 2020.4331, 2020.4523,
       2020.4714, 2020.4906, 2020.511 , 2020.5301, 2020.5493, 2020.5684,
       2020.5861, 2020.6052, 2020.6244, 2020.6436, 2020.6627, 2020.6804,
       2020.6995, 2020.7187, 2020.7379, 2020.7582, 2020.7774, 2020.7965,
       2020.8157, 2020.8333, 2020.8525, 2020.8717, 2020.8908, 2020.91  ,
       2020.9304, 2020.9495, 2020.9687, 2020.9879, 2021.0055, 2021.0246,
       2021.0438, 2021.063 , 2021.0821, 2021.0998, 2021.1189, 2021.1381,
       2021.1573, 2021.1831, 2021.2023, 2021.2214])
```

[4]:
```
node_frequencies = np.array([0.00000e+00, 0.00000e+00, 0.00000e+00, 0.
↪00000e+00, 0.00000e+00,
       0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00,
       0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00,
       0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00,
       0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00,
       0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00,
       0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00,
       0.00000e+00, 1.00000e-06, 3.00000e-06, 1.30000e-05, 6.80000e-05,
```

```
       2.38000e-04, 6.60000e-04, 1.54300e-03, 3.05500e-03, 5.88100e-03,
       1.06380e-02, 1.82880e-02, 3.00560e-02, 4.76210e-02, 6.81920e-02,
       9.09050e-02, 1.13454e-01, 1.32703e-01, 1.51767e-01, 1.69429e-01,
       1.86222e-01, 2.02518e-01, 2.17461e-01, 2.33414e-01, 2.49272e-01,
       2.65028e-01, 2.85990e-01, 3.01494e-01, 3.16859e-01])
```

[5]:
```
logit_node_frequencies = logit_transform(node_frequencies, pc=1e-3)
```

1.2 Inspect units of pivots

Pivots are floating point representations of years where one day equals approximately 1/365 (based on TreeTime's numeric date function).

[6]:
```
pivots[-1] - pivots[-2]
```

[6]:
```
0.019099999999980355
```

[7]:
```
one_day = 1 / 365
```

[8]:
```
one_day
```

[8]:
```
0.0027397260273972603
```

[9]:
```
days_between_pivots = (pivots[-1] - pivots[-2]) / one_day
```

[10]:
```
days_between_pivots
```

[10]:
```
6.9714999999928295
```

Calculate the delta time between pivots from the GitHub discussion. The actual pivots used in the delta frequency script are the last pivot and 5 pivots back (4 back from the last pivot = 4 weeks).

[11]:
```
delta_time = pivots[-1] - pivots[-4]
```

[12]:
```
delta_time
```

[12]:
```
0.06409999999982574
```

Inspect frequencies at pivots.

[13]:
```
node_frequencies[-1]
```

[13]:
```
0.316859
```

[14]:
```
node_frequencies[-4]
```

[14]:
```
0.265028
```

Inspect logit frequencies at pivots.

[15]:
```
logit_node_frequencies[-1]
```

[15]:
```
-0.7682444004574676
```

[16]:
```
logit_node_frequencies[-4]
```

[16]:
```
-1.0199969224671832
```

Calculate the slope between logit frequencies at the last timepoint and four timepoints back.

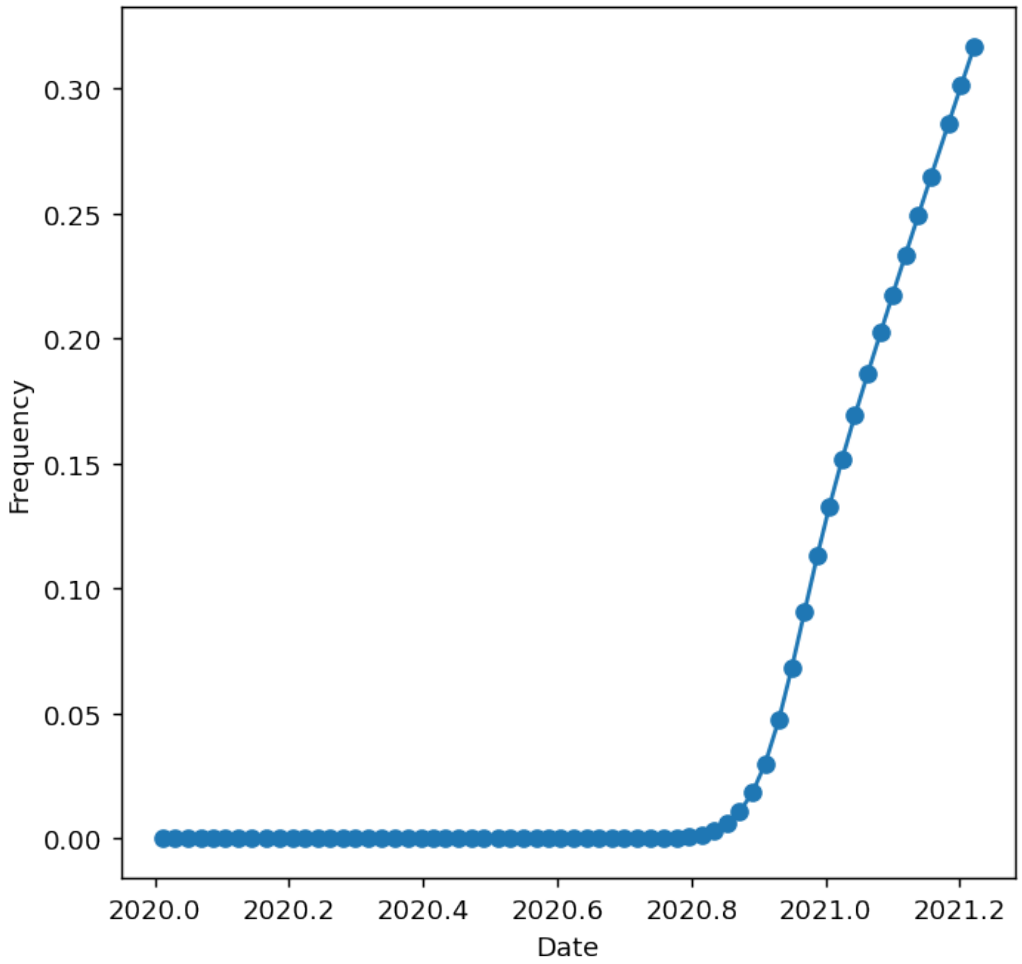
[17]:
```
(logit_node_frequencies[-1] - logit_node_frequencies[-4]) / delta_time
```

[17]:
```
3.9274964432199697
```

1.3 Plot frequencies

[18]:
```
fig, ax = plt.subplots(1, 1, figsize=(6, 6), dpi=dpi)
ax.plot(
    pivots,
    node_frequencies,
    "o-"
)
ax.set_xlabel("Date")
ax.set_ylabel("Frequency")
```

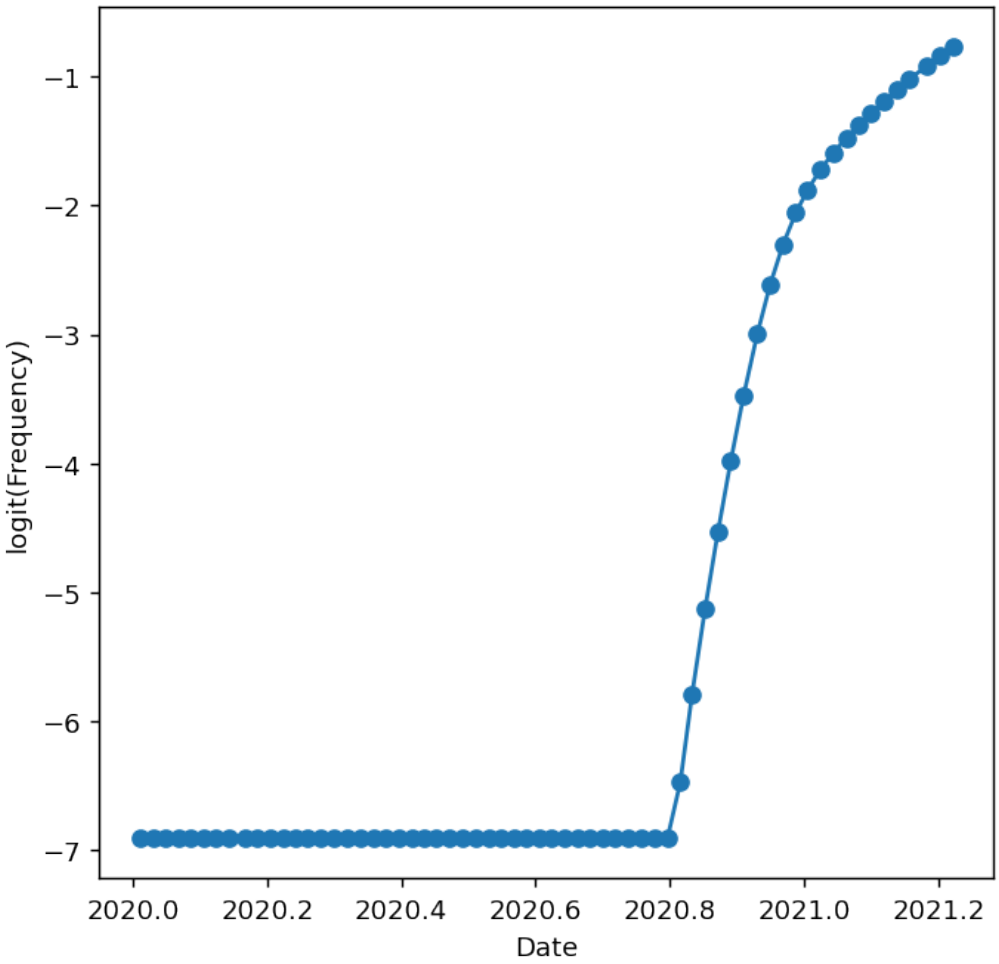
[18]:
```
Text(0, 0.5, 'Frequency')
```



1.4 Plot logit frequencies

[19]:
```
fig, ax = plt.subplots(1, 1, figsize=(6, 6), dpi=dpi)
ax.plot(
    pivots,
    logit_node_frequencies,
    "o-"
)
ax.set_xlabel("Date")
ax.set_ylabel("logit(Frequency)")
```

[19]:
```
Text(0, 0.5, 'logit(Frequency)')
```



1.5 Plot logit frequencies, highlighting pivots used in model fitting

[20]:
```
# 5 pivots back is 4 weeks prior to the last pivot.
first_pivot_index = -5
```

[21]:
```
pivots[first_pivot_index:]
```

[21]:
```
array([2021.1381, 2021.1573, 2021.1831, 2021.2023, 2021.2214])
```

[22]:
```
node_frequencies[first_pivot_index:]
```

[22]:
```
array([0.249272, 0.265028, 0.28599 , 0.301494, 0.316859])
```

[23]:
```
logit_node_frequencies[first_pivot_index:]
```

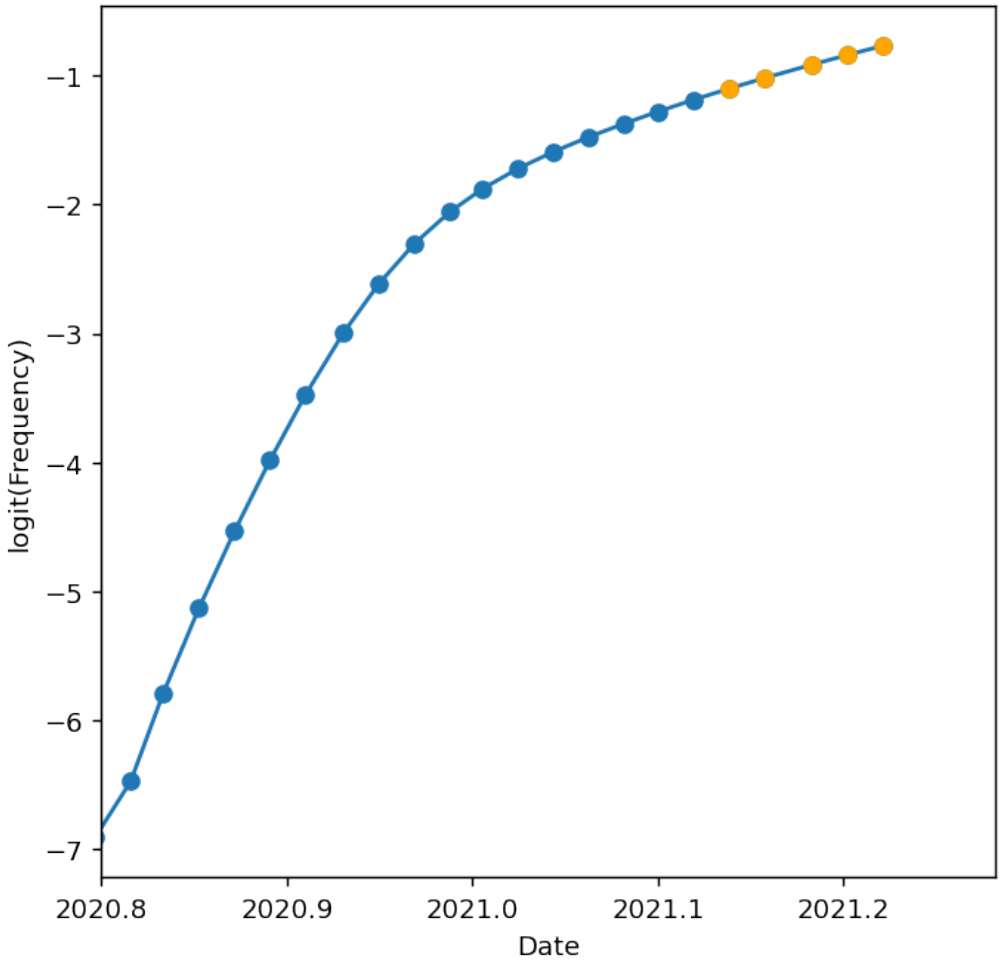
[23]:
```
array([-1.10249873, -1.01999692, -0.91494012, -0.84019365, -0.7682444 ])
```

[24]:
```
fig, ax = plt.subplots(1, 1, figsize=(6, 6), dpi=dpi)
ax.plot(
    pivots,
    logit_node_frequencies,
    "o-"
)

ax.plot(
    pivots[first_pivot_index:],
    logit_node_frequencies[first_pivot_index:],
    "o",
    color="orange"
)

ax.set_xlim(left=2020.8)
ax.set_xlabel("Date")
ax.set_ylabel("logit(Frequency)")
```

[24]:
```
Text(0, 0.5, 'logit(Frequency)')
```



1.6 Fit linear model to logit frequencies

[25]:
```
model = linregress(
    pivots[first_pivot_index:],
    logit_node_frequencies[first_pivot_index:]
)
```

[26]:
```
model
```

[26]:
```
LinregressResult(slope=4.011510776437146, intercept=-8108.916290950229,
rvalue=0.9997102240746145, pvalue=5.921182288488513e-06,
stderr=0.055768361693472164, intercept_stderr=112.7179218380822)
```

[27]:
```
x = np.linspace(pivots[first_pivot_index], pivots[-1], 20)
```

[28]:
```
y = model.slope * x + model.intercept
```

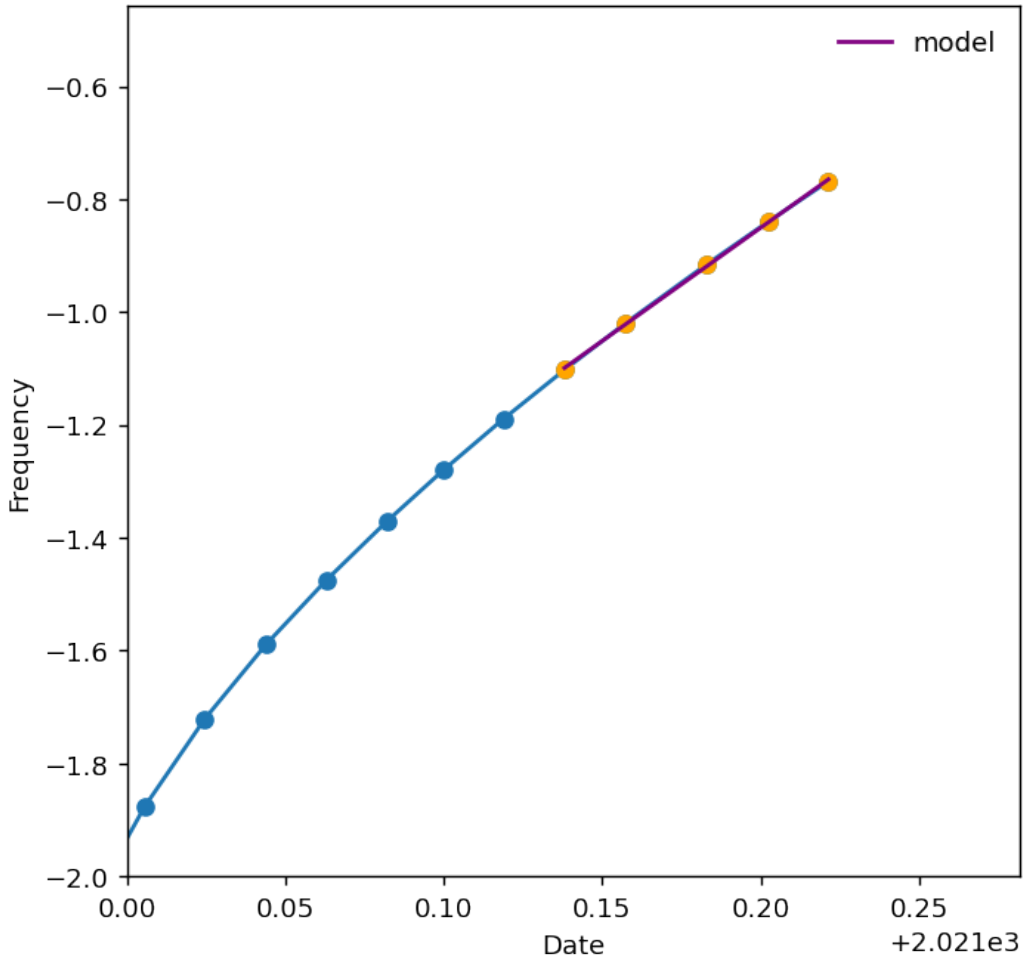
[29]:
```
fig, ax = plt.subplots(1, 1, figsize=(6, 6), dpi=dpi)
ax.plot(
    pivots,
    logit_node_frequencies,
    "o-"
)

ax.plot(
    pivots[first_pivot_index:],
    logit_node_frequencies[first_pivot_index:],
    "o",
    color="orange"
)

ax.plot(
    x,
    y,
    color="purple",
    label="model"
)

ax.legend(frameon=False)
ax.set_xlim(left=2021.0)
ax.set_ylim(bottom=-2)
ax.set_xlabel("Date")
ax.set_ylabel("Frequency")
```

[29]: Text(0, 0.5, 'Frequency')



1.7 Compare model based on log10 transformed frequencies to log transformed

[30]:
```
log10_logit_node_frequencies = np.log10(node_frequencies / (1 -␣
↪node_frequencies))
```

```
<ipython-input-30-e1f05b5457b4>:1: RuntimeWarning: divide by zero encountered in
log10
  log10_logit_node_frequencies = np.log10(node_frequencies / (1 -
node_frequencies))
```

[31]:
```
log10_logit_node_frequencies
```

[31]:
```
array([      -inf,       -inf,       -inf,       -inf,       -inf,
             -inf,       -inf,       -inf,       -inf,       -inf,
```

```
       -inf,        -inf,        -inf,        -inf,        -inf,
       -inf,        -inf,        -inf,        -inf,        -inf,
       -inf,        -inf,        -inf,        -inf,        -inf,
       -inf,        -inf,        -inf,        -inf,        -inf,
       -inf,        -inf,        -inf,        -inf,        -inf,
       -inf, -5.99999957, -5.52287744, -4.886051  , -4.16746155,
 -3.62331967, -3.18016934, -2.81096344, -2.51365999, -2.22798719,
 -1.96849524, -1.72981789, -1.50881548, -1.3010113 , -1.13559301,
 -1.0000215 , -0.89288146, -0.8152871 , -0.74733781, -0.69038902,
 -0.64047496, -0.59525726, -0.55612461, -0.51643398, -0.47880912,
 -0.44297903, -0.39735345, -0.36489147, -0.3336443 ])
```

[32]:
```
log10_model = linregress(
    pivots[first_pivot_index:],
    log10_logit_node_frequencies[first_pivot_index:],
)
```

[33]:
```
log10_model
```

[33]:
```
LinregressResult(slope=1.7421769943020815, intercept=-3521.6575993750675,
rvalue=0.9997102240746146, pvalue=5.9211822884851045e-06,
stderr=0.024219891748255007, intercept_stderr=48.95277146587176)
```

[34]:
```
model
```

[34]:
```
LinregressResult(slope=4.011510776437146, intercept=-8108.916290950229,
rvalue=0.9997102240746145, pvalue=5.921182288488513e-06,
stderr=0.055768361693472164, intercept_stderr=112.7179218380822)
```

[ ]:
```

```