DataNights

BIG DATANIGHTS - TIME SERIES

MAYA BERCOVITCH
Director of Data Science
Anaplan

LIRON MORGENSTERN
Data Scientist
Anaplan

HILA PAZ HERSZFANG
Data Science Team Leader
Zscaler

ALICE FRIDBERG
Data Science Team Leader
SKAI

NAOMI FRIDMAN
Deep Learning Consultant
Freelance

ZIV FREUND
Head of AI Research Group
Elbit Systems

RAZ TAMIR
Data Science Team Leader
Owlytics Healthcare

2.11- 7.12

Wednesdays
18:30 - 20:30

Amnon Wahle
Program Leader
Data Scientist
Wiliot

בראל

Datahack

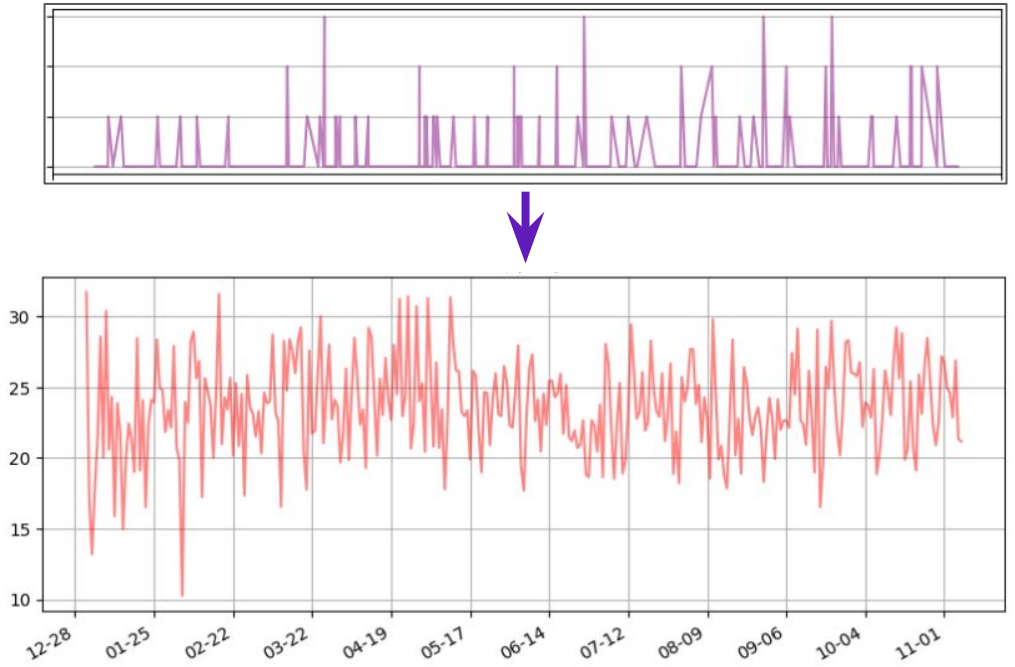
SUIT Method

SAMPLE

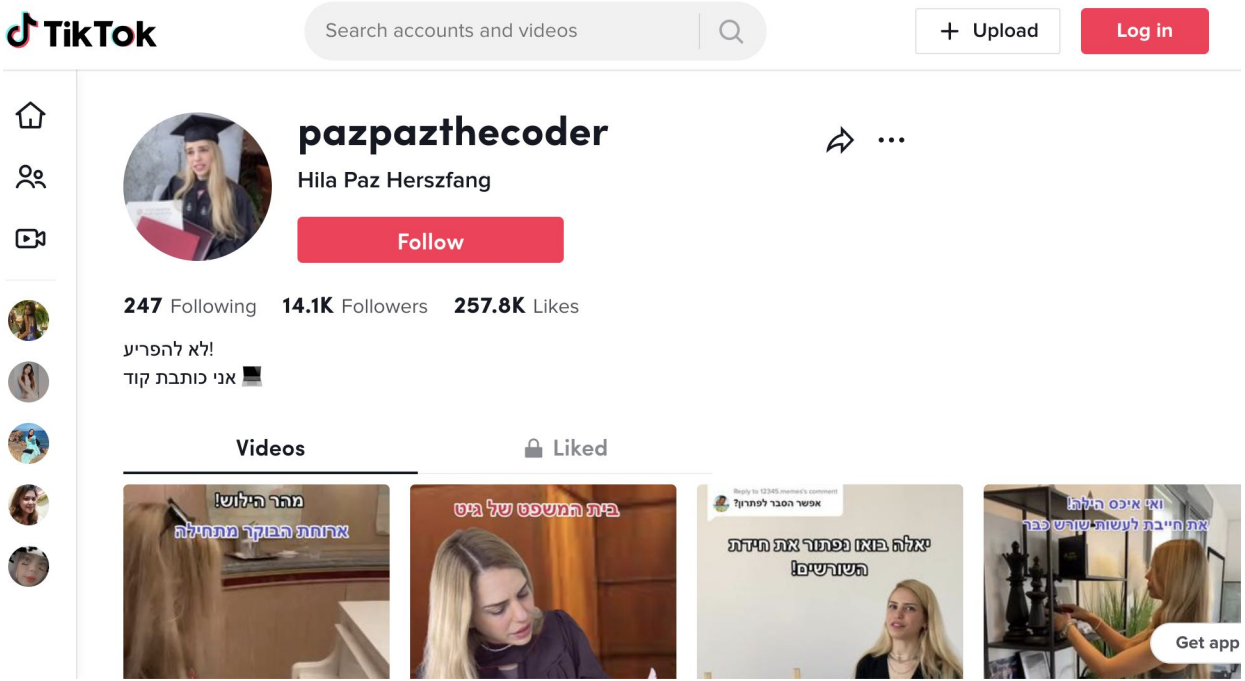
UNIVARIATE

ISOLATE

TRANSFORM

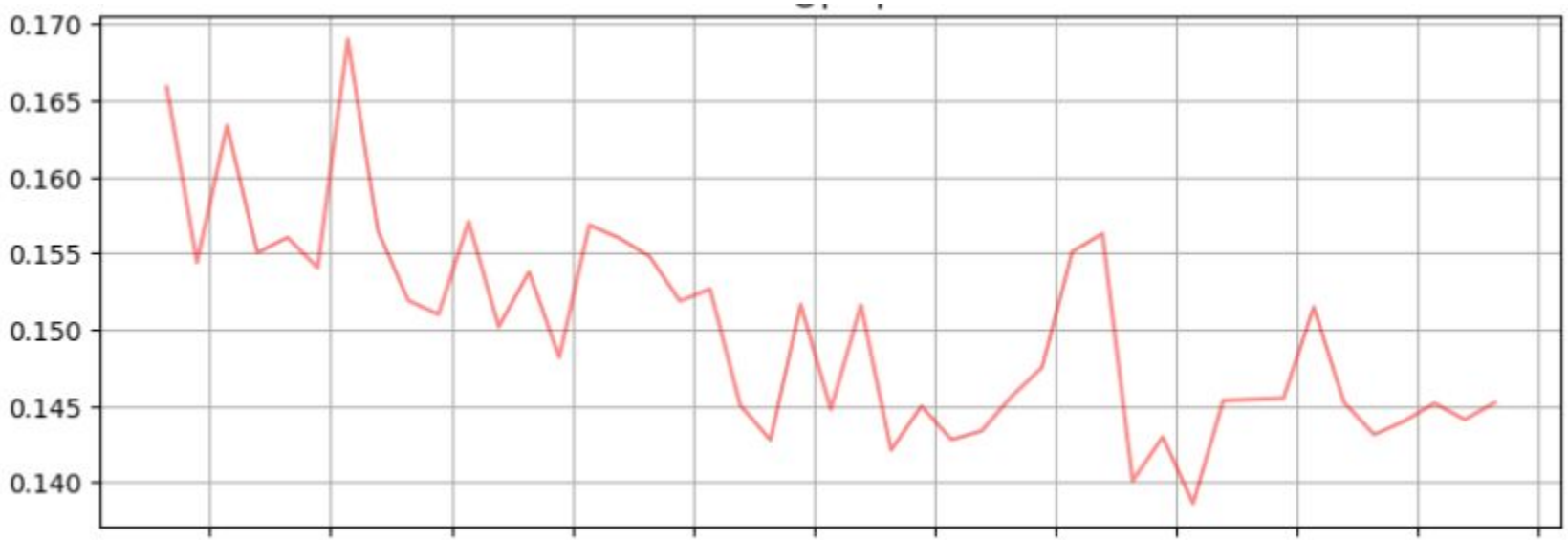


OUR DATA


TikTok
Search accounts and videos
Upload
Log in
pazpazthecoder
Hila Paz Herszfang
Follow
247 Following
14.1K Followers
257.8K Likes
!לא להפריע
אני כותבת קוד
Videos
Liked
מהר הילוש!
ארוחת הבוקר מתחילה
בית המשפט של גיט
אפשר הסבר לפתרון?
יאלה בואו נפתור את חידת השורשים!
Get app

OUR GOAL

Understand why the like ratio is decreasing



DATA TYPES

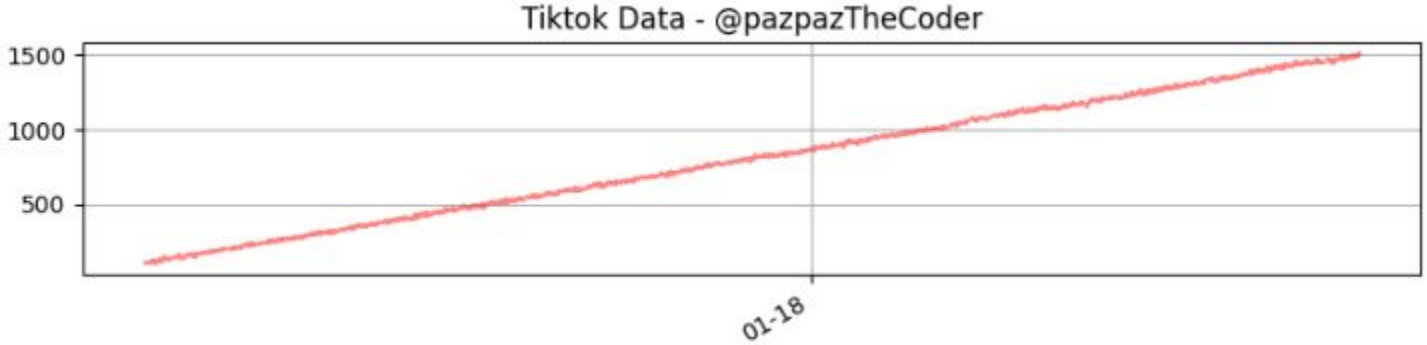
datetime	followers
2022-01-01 00:00:00	112.0
2022-01-01 00:30:00	NaN
2022-01-01 01:00:00	102.0
2022-01-01 01:30:00	106.0
2022-01-01 02:00:00	105.0
2022-01-01 02:30:00	105.0
2022-01-01 03:00:00	111.0
2022-01-01 03:30:00	108.0
2022-01-01 04:00:00	108.0
2022-01-01 04:30:00	101.0

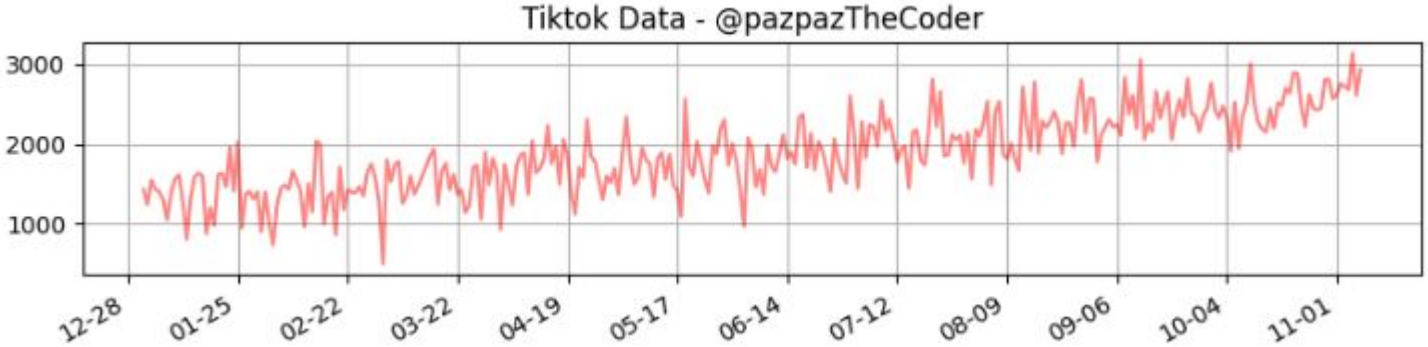
date	views	likes
2022-01-01	1431	157
2022-01-02	1241	124
2022-01-03	1547	185
2022-01-04	1435	200
2022-01-05	1407	196
2022-01-06	1304	104
2022-01-07	1052	115
2022-01-08	1382	193
2022-01-09	1559	265
2022-01-10	1615	209

DATA TYPES

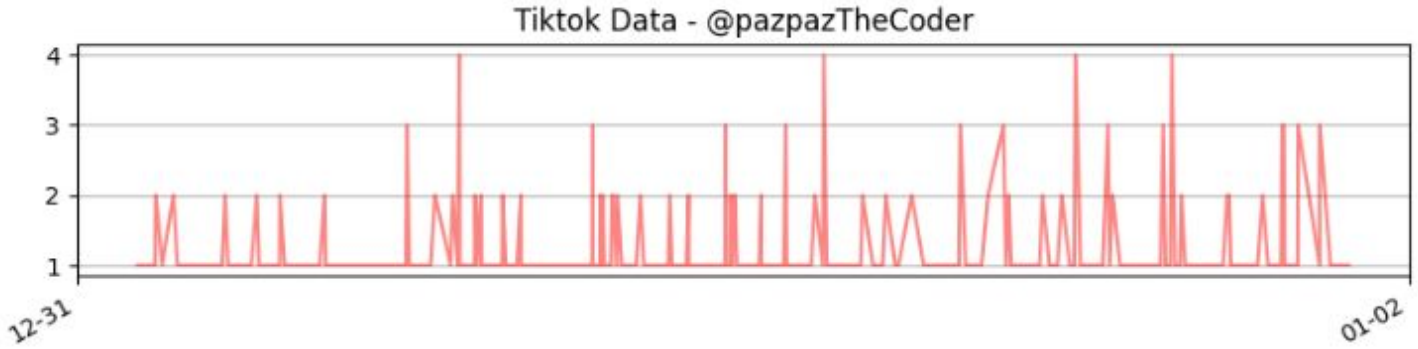
datetime	views	likes	city	age_group
2022-01-01 00:00:00	1	0	rishon	18-21
2022-01-01 00:00:00	1	0	kadima	13-15
2022-01-01 00:00:00	1	0	ramat-gan	18-21
2022-01-01 00:00:00	1	0	tel-aviv	Prefer not to Say
2022-01-01 00:00:27	1	0	tel-aviv	21-25
2022-01-01 00:08:19	1	0	tel-aviv	21-25
2022-01-01 00:10:12	1	0	tel-aviv	18-21
2022-01-01 00:10:22	1	0	tel-aviv	18-21
2022-01-01 00:12:20	1	0	tel-aviv	30+
2022-01-01 00:21:02	1	0	ramat-gan	21-25

DATA TYPES

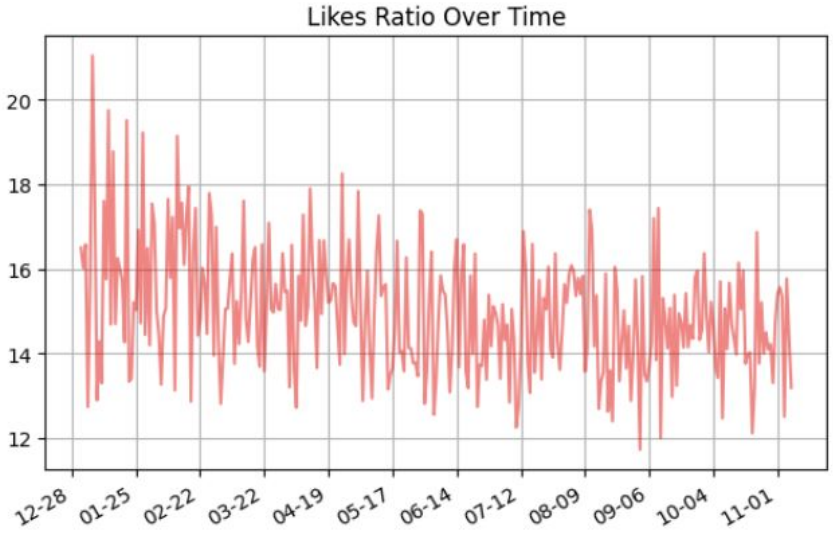

Tiktok Data - @pazpazTheCoder
1500
1000
500
01-18

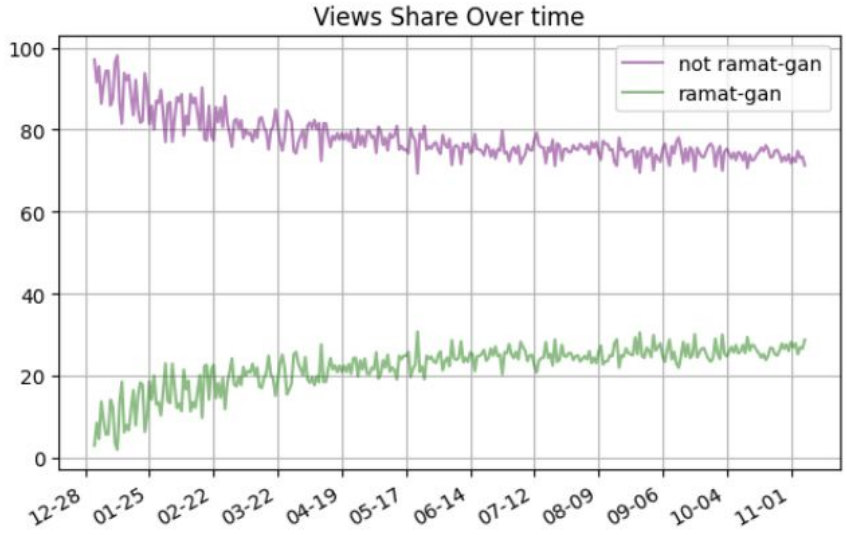

Tiktok Data - @pazpazTheCoder
3000
2000
1000
12-28
01-25
02-22
03-22
04-19
05-17
06-14
07-12
08-09
09-06
10-04
11-01

DATA TYPES



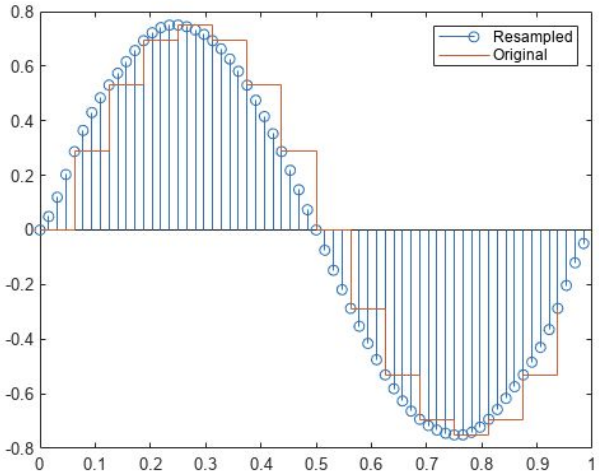
APPLY SUIT


Likes Ratio Over Time
20
18
16
14
12
12-28
01-25
02-22
03-22
04-19
05-17
06-14
07-12
08-09
09-06
10-04
11-01


Views Share Over time
not ramat-gan
ramat-gan
100
80
60
40
20
0
12-28
01-25
02-22
03-22
04-19
05-17
06-14
07-12
08-09
09-06
10-04
11-01

SAMPLING

2.1 DownSampling Cumulative Data



UNIVARIATE

```python
def get_tlv_likes(df: pd.DataFrame) -> int:
    tlv = df[df['city'] == 'tel-aviv']
    likes = sum(tlv['likes'])
    return likes

events.resample('1D').apply(get_tlv_likes)
```

```
datetime
2022-01-01     48
2022-01-02     55
2022-01-03     94
2022-01-04     44
2022-01-05     69
```

ISOLATION

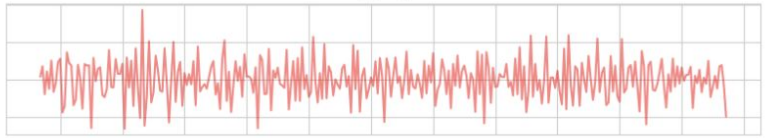
```
1  groupers = [pd.Grouper(freq='1D'),'city']
2
3  by_date_and_city = events.groupby(groupers).likes.sum()
4  by_date = events.groupby(pd.Grouper(freq='1D')).likes.sum()
5
6  (by_date_and_city / by_date).unstack()
```

city datetime	kadima	ramat-gan	rishon	tel-aviv
2022-01-01	0.061538	0.000000	0.200000	0.738462
2022-01-02	0.120482	0.012048	0.204819	0.662651
2022-01-03	0.025210	0.008403	0.176471	0.789916

311 rows × 4 columns Open in new tab

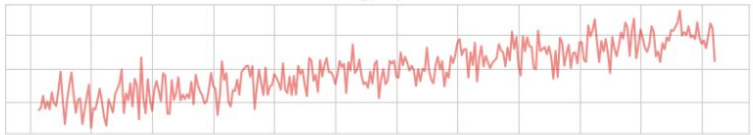
TRANSFORMATION

Stationary Data



The unconditional joint probability distribution does not change when shifted in time
E.g. mean and variance are constant

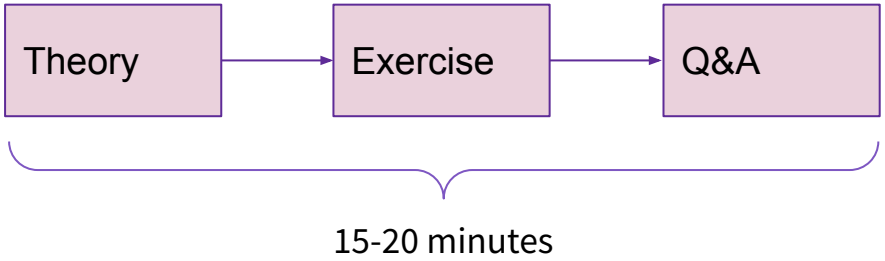
Non-Stationary Data



META STRUCTURE

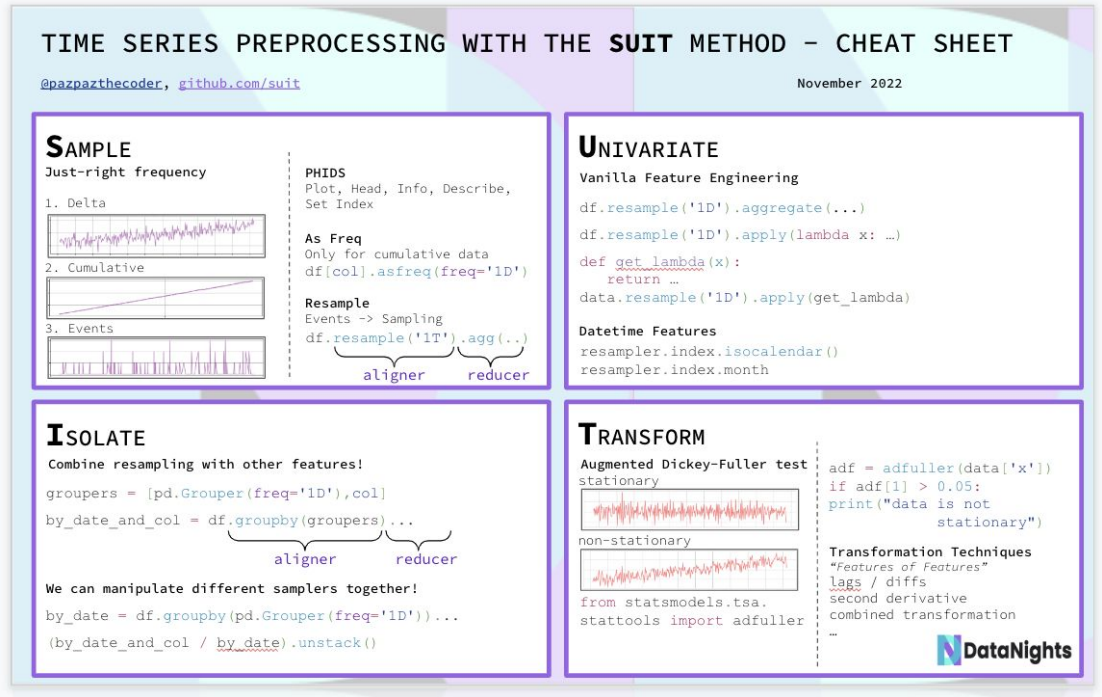
```
warmup() # We are here
```

```
for i in range[1,4]:
```




```
closure()
```

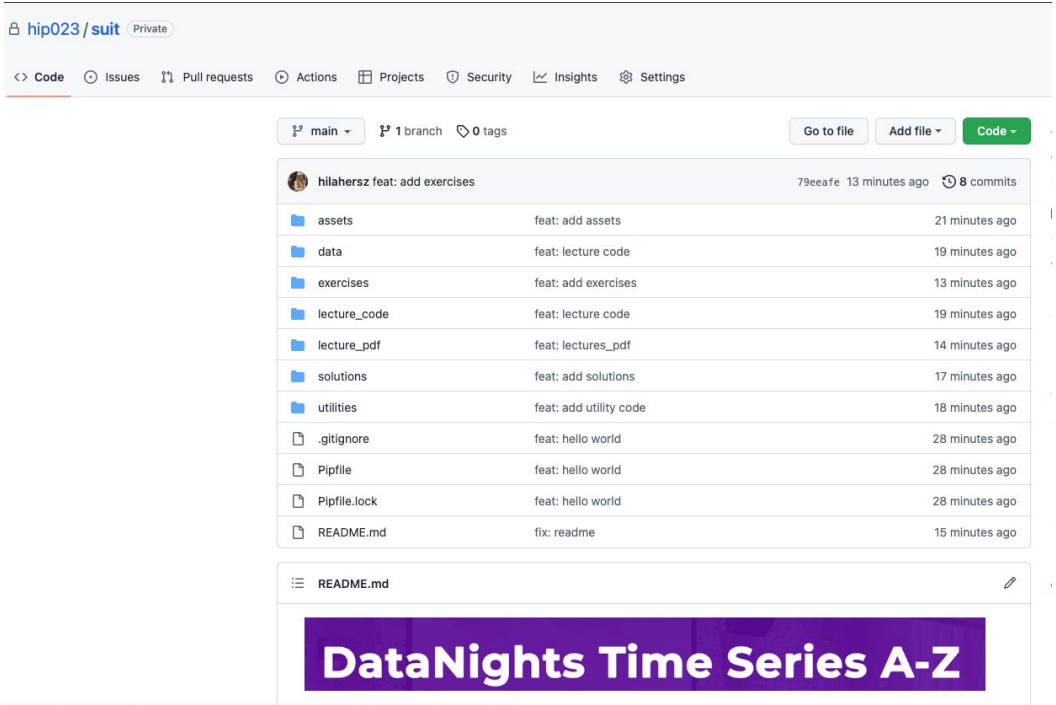
CHEAT SHEET


TIME SERIES PREPROCESSING WITH THE SUIT METHOD - CHEAT SHEET
@pazpazthecoder, github.com/suit
November 2022
SAMPLE
Just-right frequency
1. Delta
2. Cumulative
3. Events
PHIDS
Plot, Head, Info, Describe, Set Index
As Freq
Only for cumulative data
df[col].asfreq(freq='1D')
Resample
Events -> Sampling
df.resample('1T').agg(..)
aligner
reducer
UNIVARIATE
Vanilla Feature Engineering
df.resample('1D').aggregate(...)
df.resample('1D').apply(lambda x: …)
def get_lambda(x):
return …
data.resample('1D').apply(get_lambda)
Datetime Features
resampler.index.isocalendar()
resampler.index.month
ISOLATE
Combine resampling with other features!
groupers = [pd.Grouper(freq='1D'),col]
by_date_and_col = df.groupby(groupers)...
aligner
reducer
We can manipulate different samplers together!
by_date = df.groupby(pd.Grouper(freq='1D'))...
(by_date_and_col / by_date).unstack()
TRANSFORM
Augmented Dickey-Fuller test
stationary
non-stationary
from statsmodels.tsa.stattools import adfuller
adf = adfuller(data['x'])
if adf[1] > 0.05:
print("data is not stationary")
Transformation Techniques
"Features of Features"
lags / diffs
second derivative
combined transformation
…
DataNights

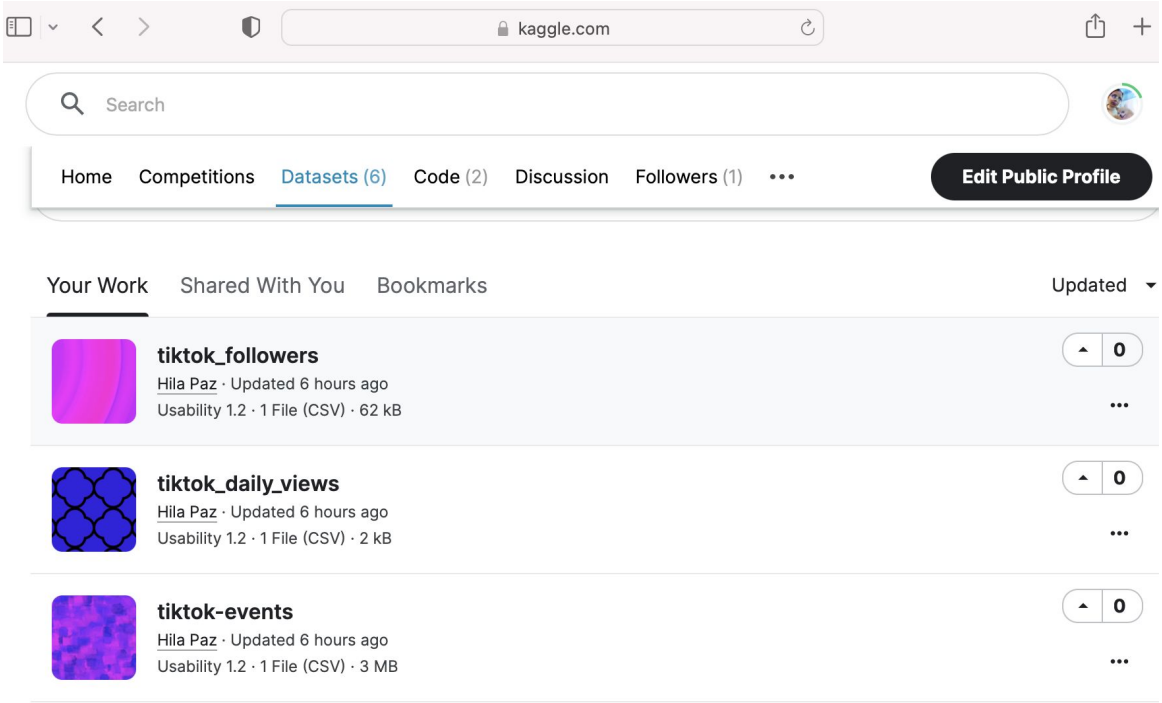
GIT REPO

https://github.com/hip023/suit

GIT REPO


hip023 / suit
Private
Code
Issues
Pull requests
Actions
Projects
Security
Insights
Settings
main
1 branch
0 tags
Go to file
Add file
Code
hilahersz feat: add exercises
79eeafe 13 minutes ago
8 commits
assets feat: add assets 21 minutes ago
data feat: lecture code 19 minutes ago
exercises feat: add exercises 13 minutes ago
lecture_code feat: lecture code 19 minutes ago
lecture_pdf feat: lectures_pdf 14 minutes ago
solutions feat: add solutions 17 minutes ago
utilities feat: add utility code 18 minutes ago
.gitignore feat: hello world 28 minutes ago
Pipfile feat: hello world 28 minutes ago
Pipfile.lock feat: hello world 28 minutes ago
README.md fix: readme 15 minutes ago
README.md
DataNights Time Series A-Z

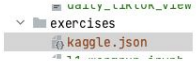
KAGGLE DATA


kaggle.com
Search
Home
Competitions
Datasets (6)
Code (2)
Discussion
Followers (1)
Edit Public Profile
Your Work
Shared With You
Bookmarks
Updated
tiktok_followers
Hila Paz · Updated 6 hours ago
Usability 1.2 · 1 File (CSV) · 62 kB
0
tiktok_daily_views
Hila Paz · Updated 6 hours ago
Usability 1.2 · 1 File (CSV) · 2 kB
0
tiktok-events
Hila Paz · Updated 6 hours ago
Usability 1.2 · 1 File (CSV) · 3 MB
0

KAGGLE DATA

Make sure that the name of the downloaded file is kaggle.json (if it's not rename it :))

5. Upload the kaggle.json file to the working directory of the exercise folder - and that's it! you're good to go.



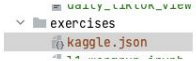

```
# Run the following cell to collect the first exercise data from kaggle
! mkdir ~/.kaggle
! mv kaggle.json ~/.kaggle/
! chmod 600 ~/.kaggle/kaggle.json
```

```
# Run the following commands to import the datasets from Kaggle
```

KAGGLE DATA

Make sure that the name of the downloaded file is kaggle.json (if it's n... rename it :))

5. Upload the kaggle.json file to the working directory of the exercise folder - and that's it! you're good to go.




```
# Run the following cell to collect the first exercise data from kaggle
! mkdir ~/.kaggle
! mv kaggle.json ~/.kaggle/
! chmod 600 ~/.kaggle/kaggle.json
```

```
# Run the following commands to import the datasets from Kaggle
```

YOUTUBE CODE

VIDEO ON

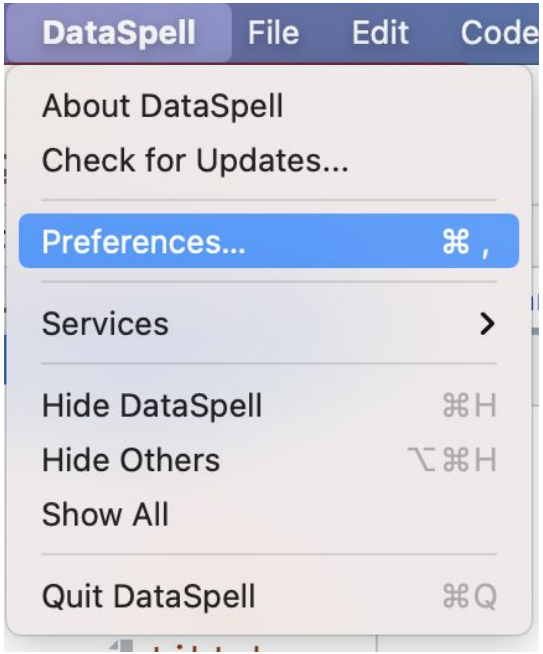
MIC Closed

WARM UP

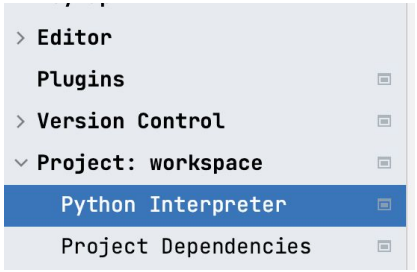

DataspellProjects — hila@Hila — -zsh — 64×16
..spellProjects
DataspellProjects git clone git@github.com:hip023/suit.git

WARM UP

1


DataSpell
File
Edit
Code
About DataSpell
Check for Updates...
Preferences... ⌘ ,
Services
Hide DataSpell ⌘H
Hide Others ⌥⌘H
Show All
Quit DataSpell ⌘Q

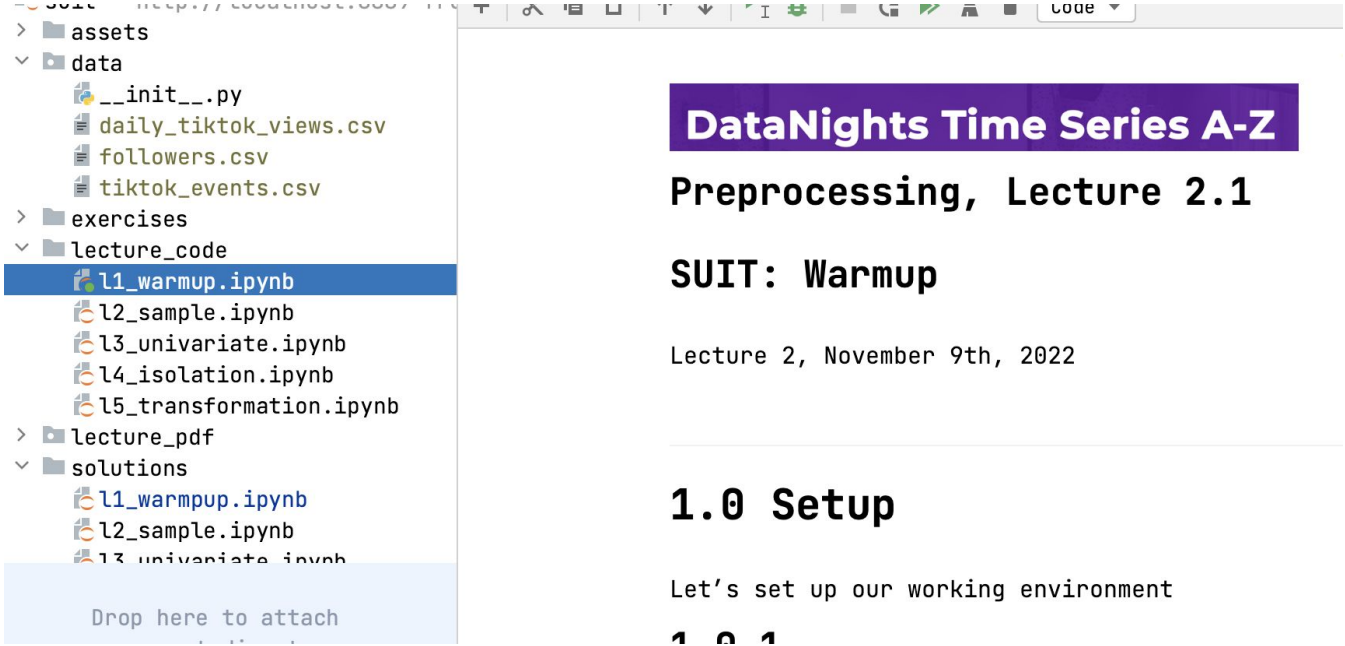
2


Editor
Plugins
Version Control
Project: workspace
Python Interpreter
Project Dependencies

3


Base interpreter:
/usr/local/bin/python3.9
Install packages from Pipfile
Pipenv executable:
Auto-detected: /usr/local/ana

WARM UP


assets
data
__init__.py
daily_tiktok_views.csv
followers.csv
tiktok_events.csv
exercises
lecture_code
l1_warmup.ipynb
l2_sample.ipynb
l3_univariate.ipynb
l4_isolation.ipynb
l5_transformation.ipynb
lecture_pdf
solutions
l1_warmpup.ipynb
l2_sample.ipynb
Drop here to attach
Code
DataNights Time Series A-Z
Preprocessing, Lecture 2.1
SUIT: Warmup
Lecture 2, November 9th, 2022
1.0 Setup
Let's set up our working environment

WARM UP

l1_warmup.ipynb

1. Move kaggle.json
2. Download 3 datasets from kaggle

EXERCISE

exercises/l1_warmup.ipynb

DataNights Time Series A-Z

Preprocessing, Exercise 1 (With Hila)

Warmpup

Lecture 2, November 9th, 2022

1.0 Setup

Make sure that you import datasets as explained in lecture_code/l1_warmup.

1.1 Let's import some packages

```
1 import pandas as pd
```

EXERCISE

exercises/l1_warmup.ipynb

Time for PHIDS!

1. plot

```
dataframe.column.plot()
```

2. head

```
dataframe.head()
```

3. info

```
dataframe.info()
```

4. describe

```
dataframe.describe()
```

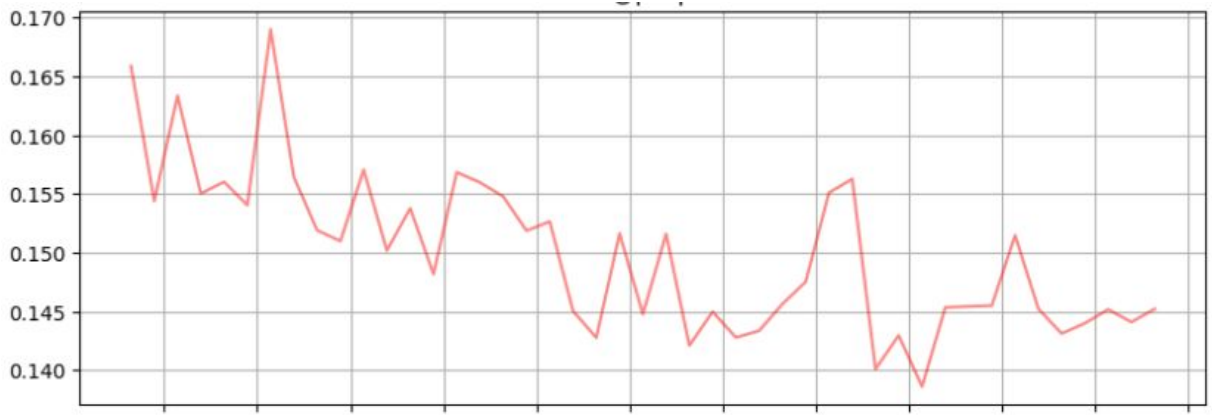
5. set index

```
dataframe.set_index("column")
```

L1 SUMMARY

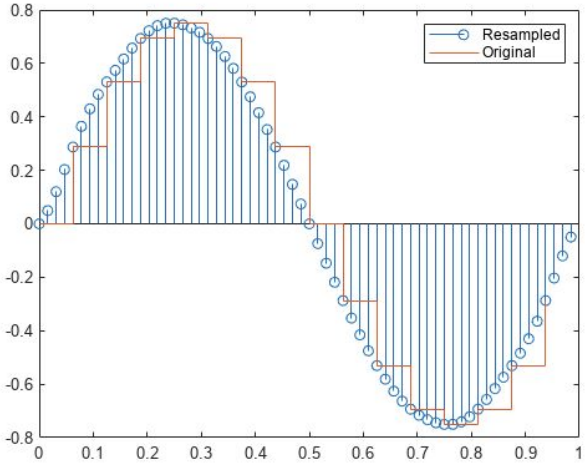
SUIT method for Time Series Preprocessing

All material is available at github.com/hip023/suit



SAMPLE

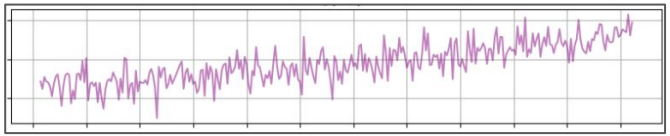
2.1 DownSampling Cumulative Data



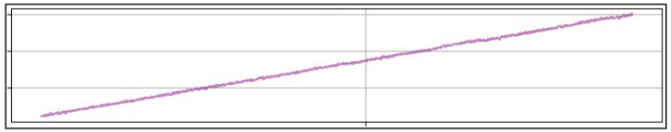
SAMPLE

Just-right frequency

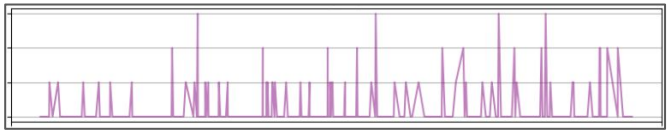
1. Delta



2. Cumulative



3. Events



PHIDS

Plot, Head, Info, Describe, Set Index

As Freq

Only for cumulative data

```
df[col].asfreq(freq='1D')
```

Resample

Events -> Sampling

```
df.resample('1T').agg(..)
```

aligner reducer

TIME SERIES PREPROCESSING WITH THE **SUIT** METHOD - CHEAT SHEET

@pazpazthecoder, github.com/hip023/suit

November 2022

DataNights

TIME SERIES PREPROCESSING WITH THE **SUIT** METHOD - CHEAT SHEET

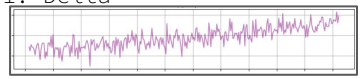
@pazpazthecoder, github.com/hip023/suit

November 2022

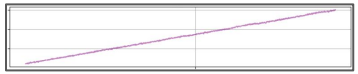
SAMPLE

Just-right frequency

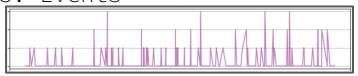
1. Delta



2. Cumulative



3. Events



PHIDS

Plot, Head, Info, Describe, Set Index

As Freq

Only for cumulative data

```
df[col].asfreq(freq='1D')
```

Resample

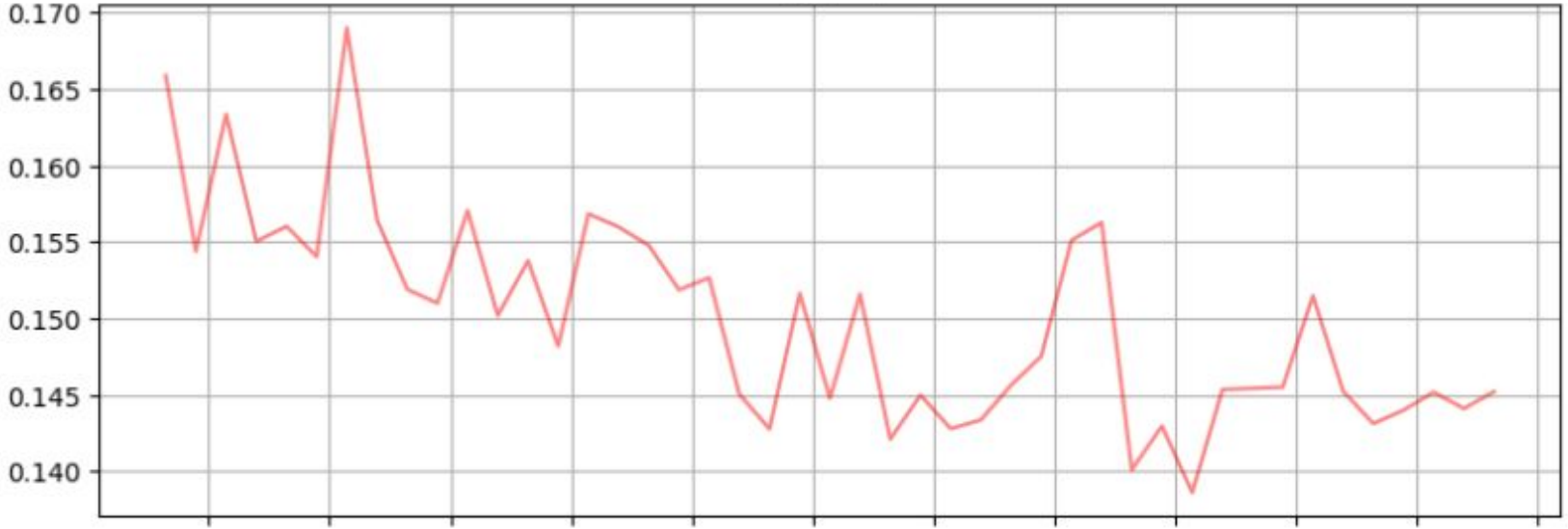
Events -> Sampling

```
df.resample('1T').agg(..)
```

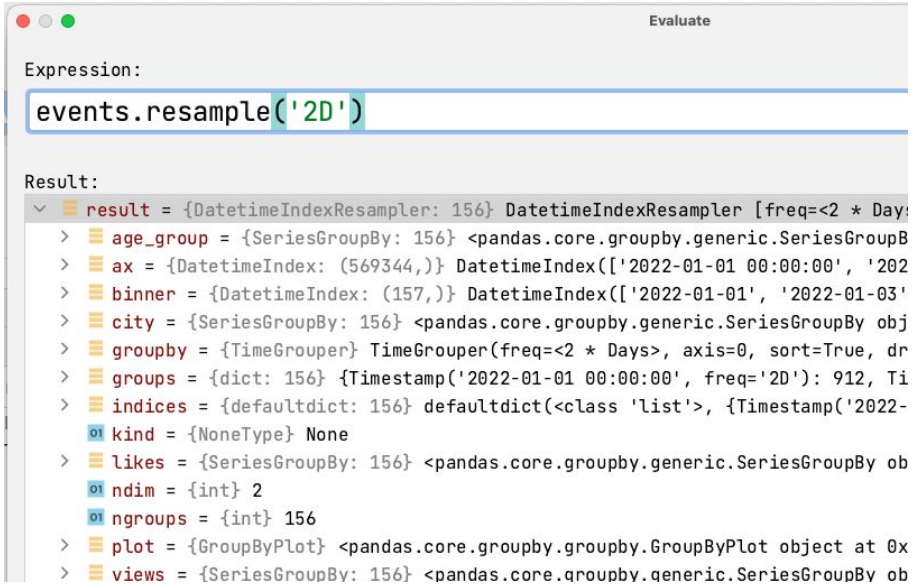
aligner reducer

DataNights

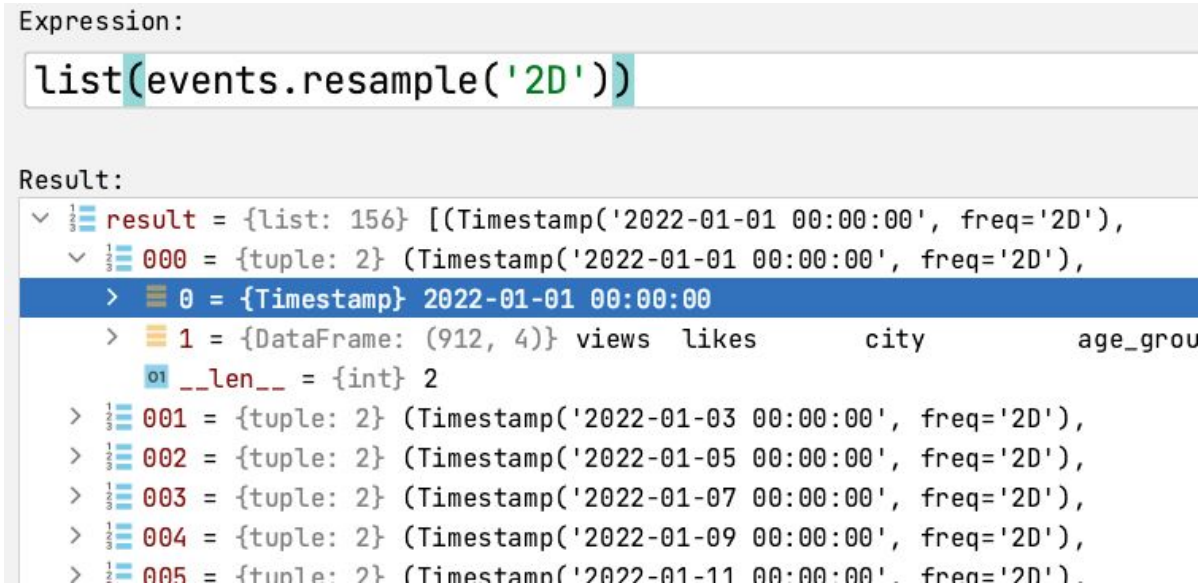
UNIVARIATE



RESAMPLER


Evaluate
Expression:
events.resample('2D')
Result:
result = {DatetimeIndexResampler: 156} DatetimeIndexResampler [freq=<2 * Day
age_group = {SeriesGroupBy: 156} <pandas.core.groupby.generic.SeriesGroupB
ax = {DatetimeIndex: (569344,)} DatetimeIndex(['2022-01-01 00:00:00', '202
binner = {DatetimeIndex: (157,)} DatetimeIndex(['2022-01-01', '2022-01-03'
city = {SeriesGroupBy: 156} <pandas.core.groupby.generic.SeriesGroupBy obj
groupby = {TimeGrouper} TimeGrouper(freq=<2 * Days>, axis=0, sort=True, dr
groups = {dict: 156} {Timestamp('2022-01-01 00:00:00', freq='2D'): 912, Ti
indices = {defaultdict: 156} defaultdict(<class 'list'>, {Timestamp('2022-
kind = {NoneType} None
likes = {SeriesGroupBy: 156} <pandas.core.groupby.generic.SeriesGroupBy ob
ndim = {int} 2
ngroups = {int} 156
plot = {GroupByPlot} <pandas.core.groupby.groupby.GroupByPlot object at 0x
views = {SeriesGroupBy: 156} <pandas.core.groupby.generic.SeriesGroupBy ob

RESAMPLER


Expression:
list(events.resample('2D'))
Result:
result = {list: 156} [(Timestamp('2022-01-01 00:00:00', freq='2D'),
000 = {tuple: 2} (Timestamp('2022-01-01 00:00:00', freq='2D'),
0 = {Timestamp} 2022-01-01 00:00:00
1 = {DataFrame: (912, 4)} views likes city age_grou
__len__ = {int} 2
001 = {tuple: 2} (Timestamp('2022-01-03 00:00:00', freq='2D'),
002 = {tuple: 2} (Timestamp('2022-01-05 00:00:00', freq='2D'),
003 = {tuple: 2} (Timestamp('2022-01-07 00:00:00', freq='2D'),
004 = {tuple: 2} (Timestamp('2022-01-09 00:00:00', freq='2D'),

RESAMPLING AGAIN

In 24
```python
def get_tlv_likes(df: pd.DataFrame) -> int:
    tel_aviv = df[df['city'] == 'tel-aviv']
    likes = tel_aviv['likes'].sum()
    return likes

events.resample('1D').apply(get_tlv_likes)
```

Out 24
```
datetime
2022-01-01    48
2022-01-02    55
2022-01-03    94
2022-01-04    44
2022-01-05    69
```

RESAMPLING AGAIN

In 24

```python
def get_tlv_likes(df: pd.DataFrame) -> int:
    tel_aviv = df[df['city'] == 'tel-aviv']
    likes = tel_aviv['likes'].sum()
    return likes

events.resample('1D').apply(get_tlv_likes)
```

Out 24

```
datetime
2022-01-01    48
2022-01-02    55
2022-01-03    94
2022-01-04    44
2022-01-05    69
```

UNIVARIATE

Vanilla Feature Engineering

```
df.resample('1D').aggregate(...)

df.resample('1D').apply(lambda x: …)

def get_lambda(x):
    return …
data.resample('1D').apply(get_lambda)
```

Datetime Features

```
resampler.index.isocalendar()
resampler.index.month
```

TIME SERIES PREPROCESSING WITH THE **SUIT** METHOD - CHEAT SHEET

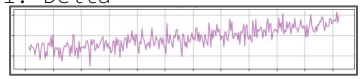
@pazpazthecoder, github.com/hip023/suit

November 2022

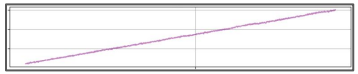
SAMPLE

Just-right frequency

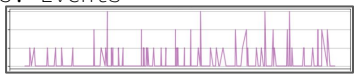
1. Delta



2. Cumulative



3. Events



PHIDS

Plot, Head, Info, Describe, Set Index

As Freq

Only for cumulative data

```
df[col].asfreq(freq='1D')
```

Resample

Events -> Sampling

```
df.resample('1T').agg(..)
```

aligner reducer

DataNights

TIME SERIES PREPROCESSING WITH THE **SUIT** METHOD - CHEAT SHEET

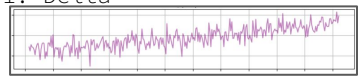
@pazpazthecoder, github.com/hip023/suit

November 2022

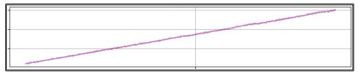
SAMPLE

Just-right frequency

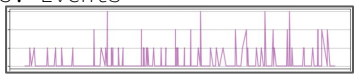
1. Delta



2. Cumulative



3. Events



PHIDS

Plot, Head, Info, Describe, Set Index

As Freq

Only for cumulative data

```
df[col].asfreq(freq='1D')
```

Resample

Events -> Sampling

```
df.resample('1T').agg(..)
```

aligner reducer

UNIVARIATE

Vanilla Feature Engineering

```
df.resample('1D').aggregate(...)
df.resample('1D').apply(lambda x: …)
def get_lambda(x):
    return …
data.resample('1D').apply(get_lambda)
```

Datetime Features

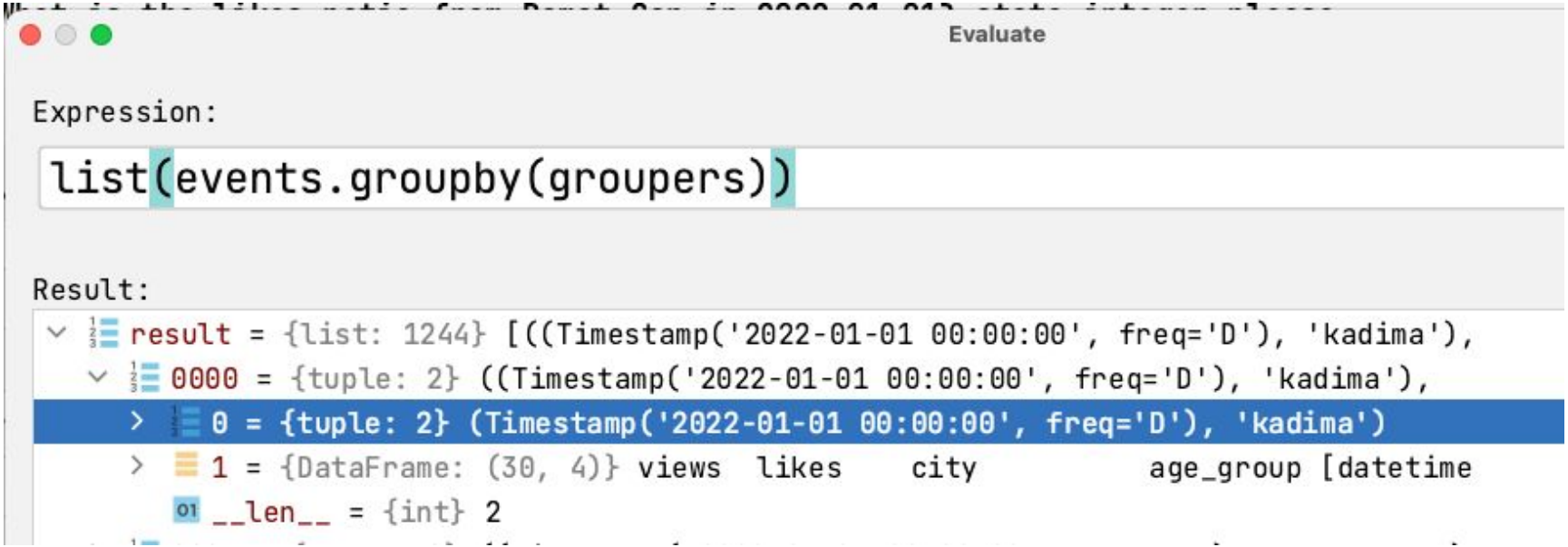
```
resampler.index.isocalendar()
resampler.index.month
```

DataNights

ISOLATION

city datetime	kadima	ramat-gan	rishon	tel-aviv
2022-01-01	0.061538	0.000000	0.200000	0.738462
2022-01-02	0.120482	0.012048	0.204819	0.662651
2022-01-03	0.025210	0.008403	0.176471	0.789916
2022-01-04	0.044118	0.014706	0.294118	0.647059
2022-01-05	0.031579	0.000000	0.242105	0.726316
2022-01-06	0.047619	0.009524	0.323810	0.619048
2022-01-07	0.061947	0.000000	0.061947	0.876106
2022-01-08	0.027027	0.027027	0.324324	0.621622

ISOLATION


Evaluate
Expression:
list(events.groupby(groupers))
Result:
result = {list: 1244} [((Timestamp('2022-01-01 00:00:00', freq='D'), 'kadima'),
0000 = {tuple: 2} ((Timestamp('2022-01-01 00:00:00', freq='D'), 'kadima'),
0 = {tuple: 2} (Timestamp('2022-01-01 00:00:00', freq='D'), 'kadima')
1 = {DataFrame: (30, 4)} views likes city age_group [datetime
__len__ = {int} 2

ISOLATE

Combine resampling with other features!

```
groupers = [pd.Grouper(freq='1D'),col]
by_date_and_col = df.groupby(groupers)...
```

We can manipulate different samplers together!

```
by_date = df.groupby(pd.Grouper(freq='1D'))...
(by_date_and_col / by_date).unstack()
```

TIME SERIES PREPROCESSING WITH THE **SUIT** METHOD - CHEAT SHEET

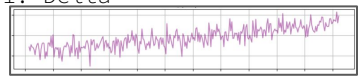
@pazpazthecoder, github.com/hip023/suit

November 2022

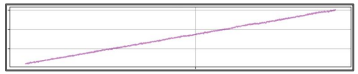
SAMPLE

Just-right frequency

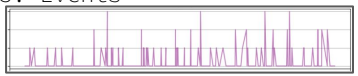
1. Delta



2. Cumulative



3. Events



PHIDS

Plot, Head, Info, Describe, Set Index

As Freq

Only for cumulative data

```
df[col].asfreq(freq='1D')
```

Resample

Events -> Sampling

```
df.resample('1T').agg(..)
```

aligner reducer

UNIVARIATE

Vanilla Feature Engineering

```
df.resample('1D').aggregate(...)
df.resample('1D').apply(lambda x: …)
def get_lambda(x):
    return …
data.resample('1D').apply(get_lambda)
```

Datetime Features

```
resampler.index.isocalendar()
resampler.index.month
```

DataNights

TIME SERIES PREPROCESSING WITH THE **SUIT** METHOD - CHEAT SHEET

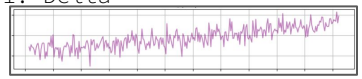
@pazpazthecoder, github.com/hip023/suit

November 2022

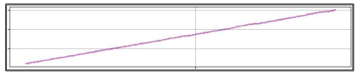
SAMPLE

Just-right frequency

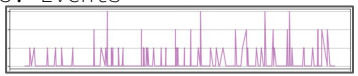
1. Delta



2. Cumulative



3. Events



PHIDS

Plot, Head, Info, Describe, Set Index

As Freq

Only for cumulative data

```
df[col].asfreq(freq='1D')
```

Resample

Events -> Sampling

```
df.resample('1T').agg(..)
```

aligner reducer

UNIVARIATE

Vanilla Feature Engineering

```
df.resample('1D').aggregate(...)
df.resample('1D').apply(lambda x: …)
def get_lambda(x):
    return …
data.resample('1D').apply(get_lambda)
```

Datetime Features

```
resampler.index.isocalendar()
resampler.index.month
```

ISOLATE

Combine resampling with other features!

```
groupers = [pd.Grouper(freq='1D'),col]
by_date_and_col = df.groupby(groupers)...
```

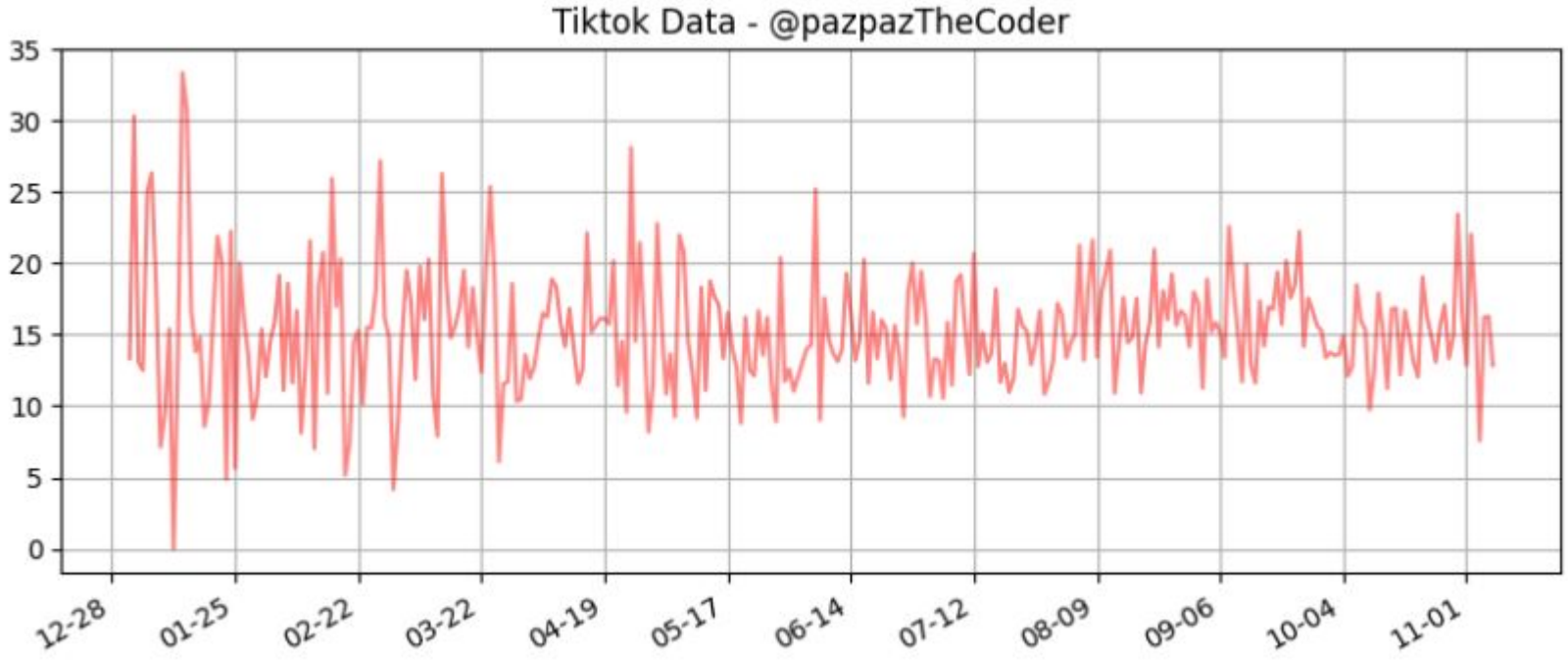
aligner reducer

We can manipulate different samplers together!

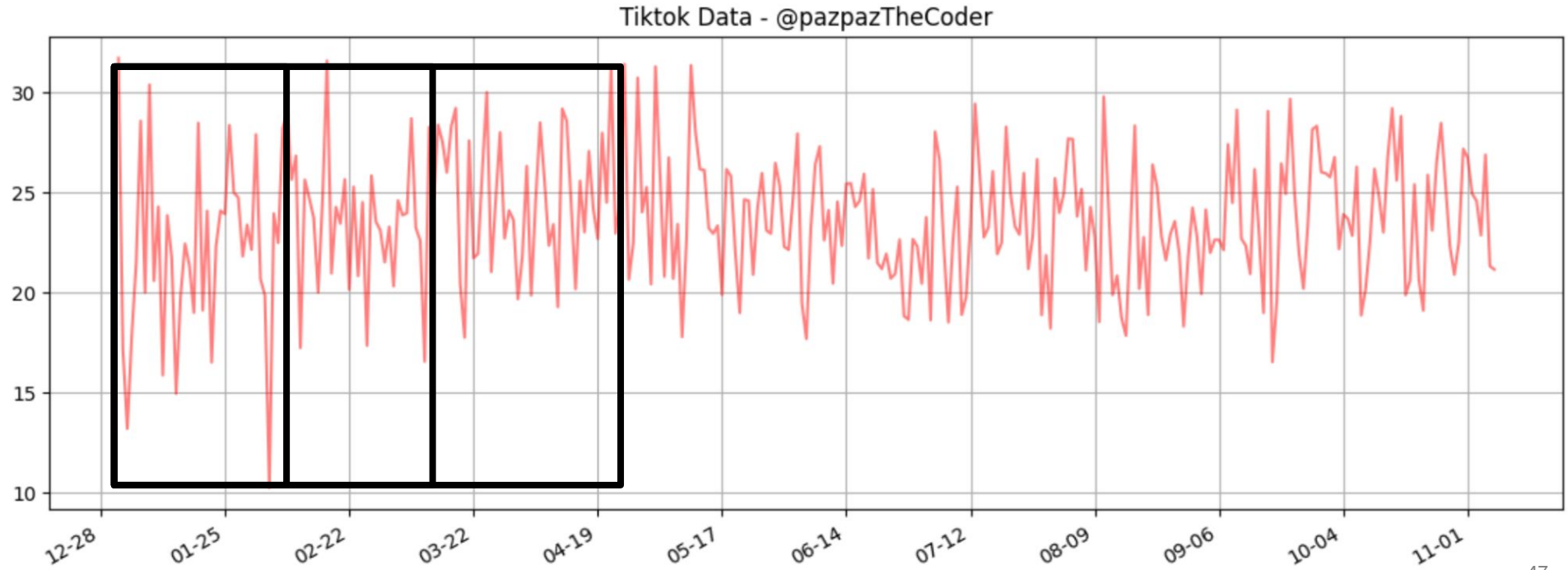
```
by_date = df.groupby(pd.Grouper(freq='1D'))...
(by_date_and_col / by_date).unstack()
```

DataNights

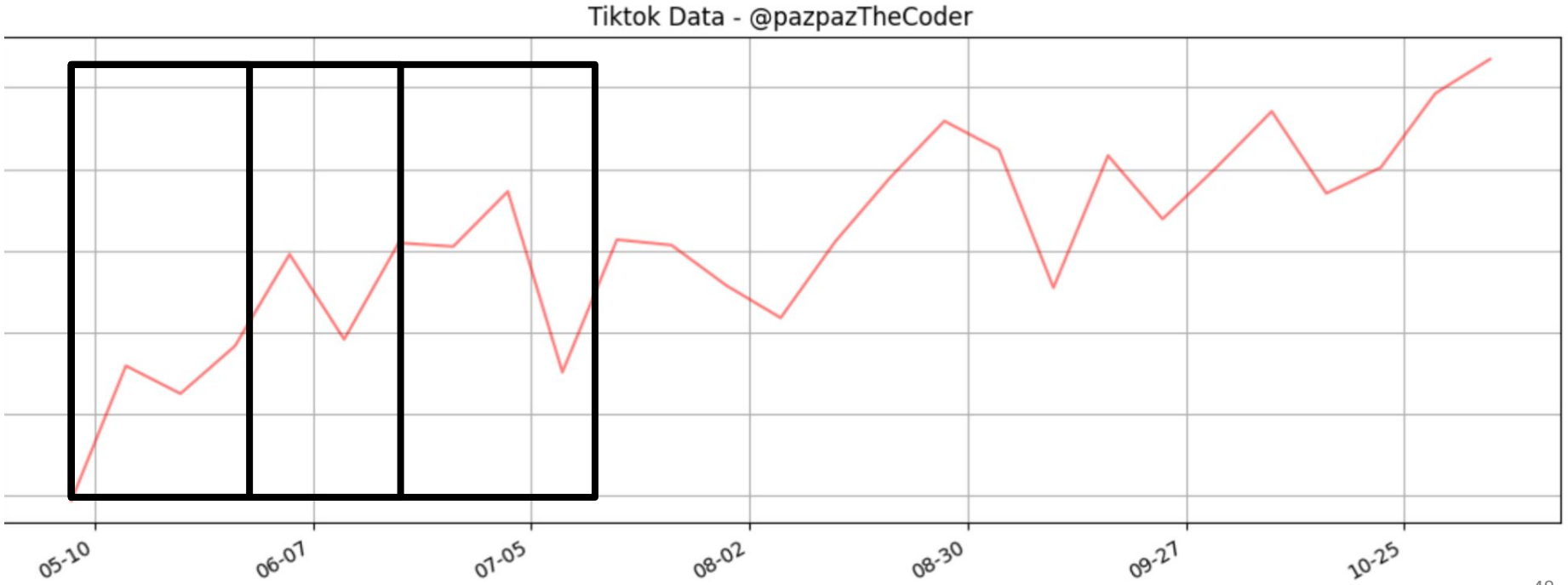
TRANSFORMATION



STATIONARY DATA



NON STATIONARY



STATIONARY TEST

The distribution of

$$(y_t, y_{t+1}, \cdots, y_{t+s})$$

For every s, is independent of t.

STATIONARY TEST

Mean is constant

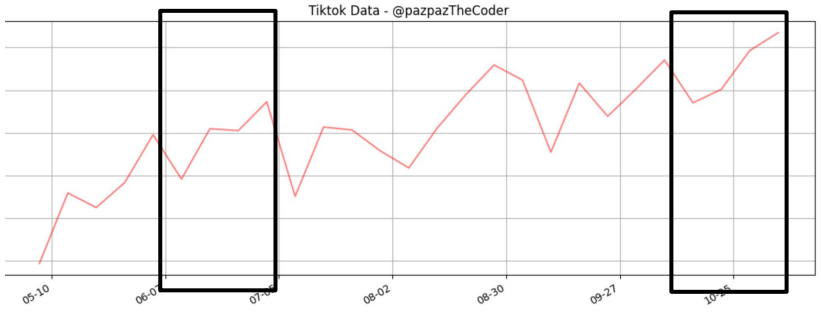
Std is constant

No seasonality

$$y_t = \mu + u_t$$

STATIONARY TEST

Global- local test



- $\mu \backslash \sigma$ change
- seasonality

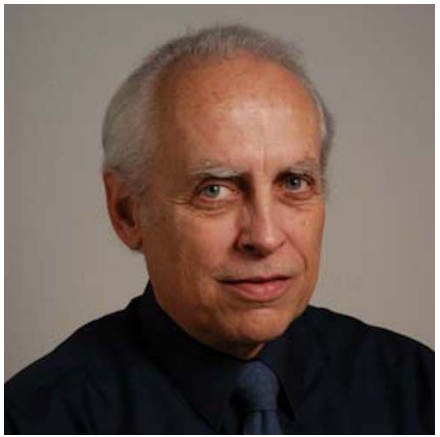
$$(y_{t_1}, \cdots, y_{t_1+s}) \quad (y_{t_2}, \cdots, y_{t_2+s})$$

STATIONARY TEST

Production ready?

ADF TEST

Augmented **D**ickey-**F**uller Test



DF TEST

Dickey-**F**uller Test

$$y_t = \mu + u_t$$

$$y_t = \mu + \rho y_{t-1} + u_t$$

Null Hypothesis
$\rho = 1$

Alternative Hypothesis
$\rho < 1$

ADF TEST

Augmented **D**ickey-**F**uller Test

$$y_t = \mu + u_t$$
$$t$$

$$y_t = \mu + \sum_{1} \rho_i \cdot y_{t-i} + u_t$$

Null Hypothesis

$\rho = 1$

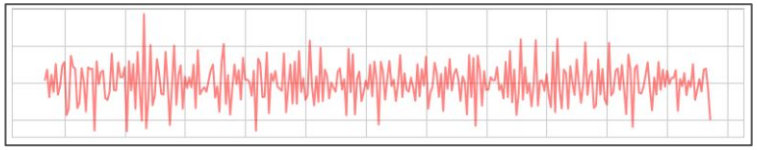
Alternative Hypothesis

$\rho < 1$

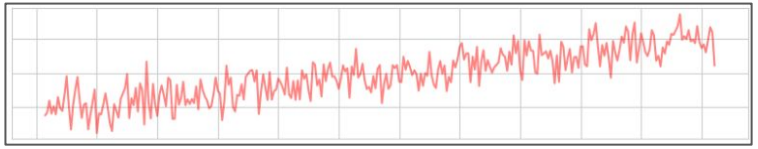
Transform

Augmented Dickey-Fuller test

stationary



non-stationary



```
from statsmodels.tsa.
stattools import adfuller
```

```
adf = adfuller(data['x'])
if adf[0] > 0.05:
print("data is not
                stationary")
```

Transformation Techniques

“Features of Features”

lags / diffs
second derivative
combined transformation

TIME SERIES PREPROCESSING WITH THE **SUIT** METHOD - CHEAT SHEET

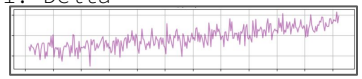
@pazpazthecoder, github.com/hip023/suit

November 2022

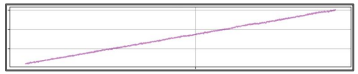
SAMPLE

Just-right frequency

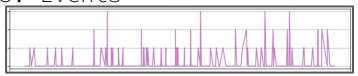
1. Delta



2. Cumulative



3. Events



PHIDS

Plot, Head, Info, Describe, Set Index

As Freq

Only for cumulative data

```
df[col].asfreq(freq='1D')
```

Resample

Events -> Sampling

```
df.resample('1T').agg(..)
```

aligner reducer

UNIVARIATE

Vanilla Feature Engineering

```
df.resample('1D').aggregate(...)
df.resample('1D').apply(lambda x: …)
def get_lambda(x):
    return …
data.resample('1D').apply(get_lambda)
```

Datetime Features

```
resampler.index.isocalendar()
resampler.index.month
```

ISOLATE

Combine resampling with other features!

```
groupers = [pd.Grouper(freq='1D'),col]
by_date_and_col = df.groupby(groupers)...
```

aligner reducer

We can manipulate different samplers together!

```
by_date = df.groupby(pd.Grouper(freq='1D'))...
(by_date_and_col / by_date).unstack()
```

DataNights

TIME SERIES PREPROCESSING WITH THE **SUIT** METHOD - CHEAT SHEET

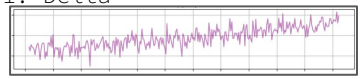
@pazpazthecoder, github.com/hip023/suit

November 2022

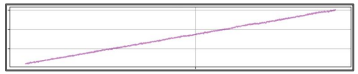
SAMPLE

Just-right frequency

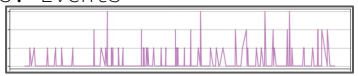
1. Delta



2. Cumulative



3. Events



PHIDS

Plot, Head, Info, Describe, Set Index

As Freq

Only for cumulative data

```
df[col].asfreq(freq='1D')
```

Resample

Events -> Sampling

```
df.resample('1T').agg(..)
```

aligner reducer

UNIVARIATE

Vanilla Feature Engineering

```
df.resample('1D').aggregate(...)
df.resample('1D').apply(lambda x: …)
def get_lambda(x):
    return …
data.resample('1D').apply(get_lambda)
```

Datetime Features

```
resampler.index.isocalendar()
resampler.index.month
```

ISOLATE

Combine resampling with other features!

```
groupers = [pd.Grouper(freq='1D'),col]
by_date_and_col = df.groupby(groupers)...
```

aligner reducer

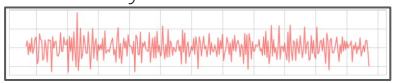
We can manipulate different samplers together!

```
by_date = df.groupby(pd.Grouper(freq='1D'))...
(by_date_and_col / by_date).unstack()
```

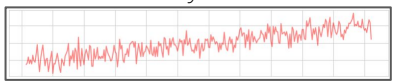
TRANSFORM

Augmented Dickey-Fuller test

stationary



non-stationary



```
from statsmodels.tsa.
stattools import adfuller
```

```
adf = adfuller(data['x'])
if adf[0] > 0.05:
print("data is not
                stationary")
```

Transformation Techniques

"Features of Features"

lags / diffs

second derivative

combined transformation

…

DataNights