Technical Documentation

Dictionary

Attribute	Details	Example
date	Release Date weekly on Mondays	2025-01-20
ticker	Ticker	HUBS
BBticker	Bloomberg Ticker	HUBS
source	For Example GitHub, Freelancer etc. or Aggregate	Freelancer
RawCount	Raw Count of Observations	24
ActChng	Change in Activity, standardized (normal distributed)	-0.03
TrendRank	Trend of Activity, ranked against universe (0-100%)	84%

Example:

```
date       ticker BBticker    source      RawCount ActChng TrendRank
<date>     <chr>  <chr>       <chr>          <dbl>   <dbl>     <dbl>
2025-01-06 SNOW   SNOW US     Freelancer         2  0.0675    0.568
2025-01-06 SNOW   SNOW US     GH_forks          69 -0.294     0.0123
2025-01-06 SNOW   SNOW US     GH_pushes       1635 -0.312     0.851
2025-01-06 SNOW   SNOW US     GH_stars         590  0.119     0.0795
2025-01-06 SNOW   SNOW US     StackO            20  0.559     0.955
2025-01-06 WIX    WIX US      Aggregate         NA -2.09      0.211
2025-01-06 WIX    WIX US      Apps          154603 -0.602     0.144
2025-01-06 WIX    WIX US      Freelancer        54  0.0254    0.608
2025-01-06 WIX    WIX US      GH_pushes          0 -0.352     0.161
2025-01-06 WIX    WIX US      Google_GB         41 -1.01      0.153
```

Explanation of Data Fields

Date:

The Data is a weekly timeseries, defined and released on Mondays (e.g. 20.01.2025).

The Counts are point-in-time data with the exception of Google Trends.

In the case of Google Trends and Apps, the observations are already 1 week old as scraping takes several days.

Ticker: Ticker Symbol

Source:

The Web Platform on which Activity was measured. The occurrence of keywords / brands on those platforms is measured each week.

Freelancer	Project Postings on Freelancer.com
StackO	Discussions on StackOverflow
GH_forks, GH_pushes, GH_stars	GitHub Repositories can be forked, pushed and starred
Google_glob, Google_GB	Google Search Counts as provided by Google Trends. Regions Global and UK are available. The Search Counts are expressed as a value between 0-100. And the whole history changes each week (!).
Apps	App Downloads on Google Play and Apple Store

Aggregate	An aggregation of all other sources. It calculates the sum of Sources, normalized by Standard Deviation. Only a Value for Activity Changes, not RawCount is provided.
	
	`merge( TRND_adj_glob / sd(tail(TRND_adj_glob,60)), TRND_adj_stars / sd(tail(TRND_adj_stars,60)) … ,join="left")`
	`TRND_adj_comb <- colCombine(TRND_adj_comb, fun="sum")`

Links to Sources:

https://www.freelancer.com/jobs/2/?keyword=twilio&status=all

https://stackoverflow.com/search?tab=newest&q=twilio

https://github.com/topics/twilio

https://trends.google.com/trends/explore?date=today%205-y&q=twilio

RawCount:

The Number of occurrences for a ticker found on a platform each week. Each ticker/company is measured via its software brands which form the keywords in a query on the platform.

Weekly Change	`TRND - stats::lag(TRND,1)`
Standardization of each Ticker	`TRND / matrix(rep( sapply(TRND , sd)+2 , dim(TRND)[1]), ncol=dim(TRND)[2], byrow=T)`
4 week change	`TRND[,] <- mav(TRND,4)`

ActivityChange:

Just a function of RawCount to make tickers comparable with each other: The rolling 4-week change in RawCount, normalized by its standard deviation.

Calculating 1-year linear trend	`SIG <- sig_trend(TRND, 52 ,cusum=T)`
Ranking Trends over all Tickers	`SIG[,] <- rowRanks(SIG, ties.method="average")/ncol(SIG)`

TrendRank:

Just a function of RawCount to identify longer term trends and rank them within tickers. 1. calculating the **1-year linear trend**, 2. Rank Significance of Trend within all tickers.

Delivery of Data

A Data Excerpt is openly accessible from GoogleCloud:

https://storage.googleapis.com/antedata_open/AllDat_excerpt_time.csv

https://storage.googleapis.com/antedata_open/AllDat_excerpt_ticker.csv

To test our full data sample, please sign a standard data test license with us.

Integration of Data – Creating Timeseries

Example R Code - Create a TimeSeries of Counts on Freelancer.com (Tickers in Columns):

1. Load CSV

```
df <- read.csv(
"https://storage.googleapis.com/antedata_open/AllDat_excrpt.csv")
```

2. Filter and Spread Data to Matrix (Date vertical, Tickers horizontal)

```
df <- spread(
  df[ which(df$source == "Freelancer"),c("date","ticker","RawCount")
], key="ticker", value="RawCount")
```

3. Create Timeseries Object xts

```
df <- as.xts( df[,2:ncol(df)] , order.by=as.Date(df[,1]) )
```

Example Python Code:

```
df = pd.read_csv(

"https://storage.googleapis.com/antedata_open/AllDat_excrpt.csv")

df = df.loc[df['source'] == 'Freelancer', ['date', 'ticker',
'RawCount']]

df = df.pivot(index='date', columns='ticker', values='RawCount')
```

Example Result TimeSeries

```
           CRM DBX HUBS NVDA OKTA SNOW WIX
2024-11-18  50   5   17    7    1    2  63
2024-11-25  55   5   20    3    2    4  66
2024-12-02  60  10   18   12    3    2  49
2024-12-09  58   9   25    4    5    1  52
2024-12-16  53   7   15    7    4    2  39
2024-12-23  41   4   24    4    8    3  34
2024-12-30  22   7   17    3    1    2  30
2025-01-06  44   7   24   12    3    2  54
2025-01-13  44   4   24    6    0    3  54
```

Contact

If you have technical questions, please don't hesitate to contact:

Florian.Boehringer@antedata.ch

+41 78 69 111 67