Lecture 1: Basic Prosody

1B: From Qualitative to Quantitative Analysis

Dafydd Gibbon
Bielefeld University, Germany
2022-04-25

II Brazilian Congress of Prosody
Minicourse 9, 25, 27, 39 April 2022
(09:00-11:30 Brazilian Standard Time)

Adding Quantitative to Qualitative Methods

Quantitative methods

- Qualitative methods in phonetics
  - epistemological basis of phonetics and linguistics
  - phonology
  - transcription and manual annotation

- Quantitative methods in phonetics
  - (semi)**automatic annotation with statistical training**
  - **analysis of annotations**
  - signal processing
  - (un)supervised machine learning

The deductive annotation-based approach

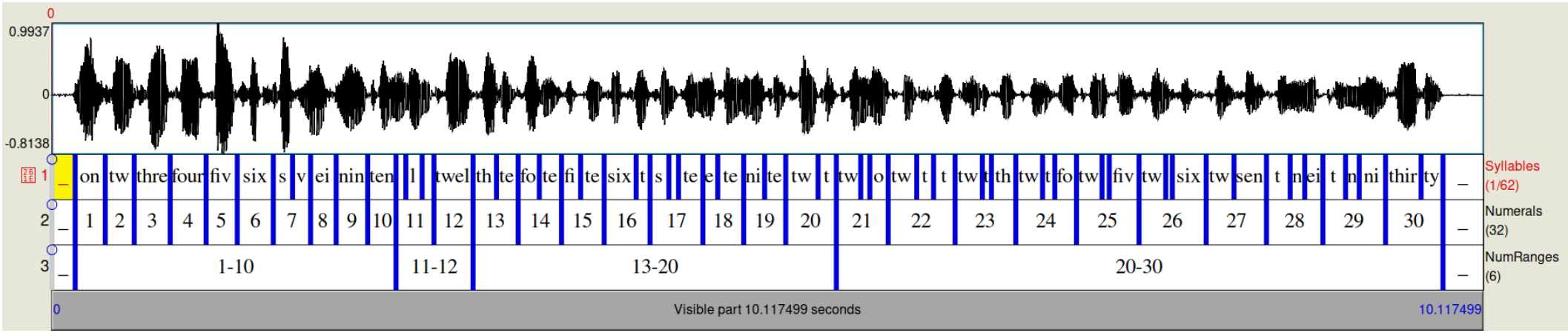
Decide on a set of linguistic categories and inventories

- phonetic
- phonological
- morphological
- grammatical (part of speech, PoS tags)
- semantic
- pragmatic
    - speech acts
    - turn-taking
    - discourse grammar

Analyse relevant speech data

- Search for and record data
- Listen, transcribe, annotate

Event annotation with ‘Praat’: intervals and labels



What goes in:

1. Recording of speech signal
2. Perception of boundaries
   1. visual, acoustic
   2. waveform, spectrum, F0
3. Perception of event intervals as speech sounds

A speech file (.WAV, etc.)

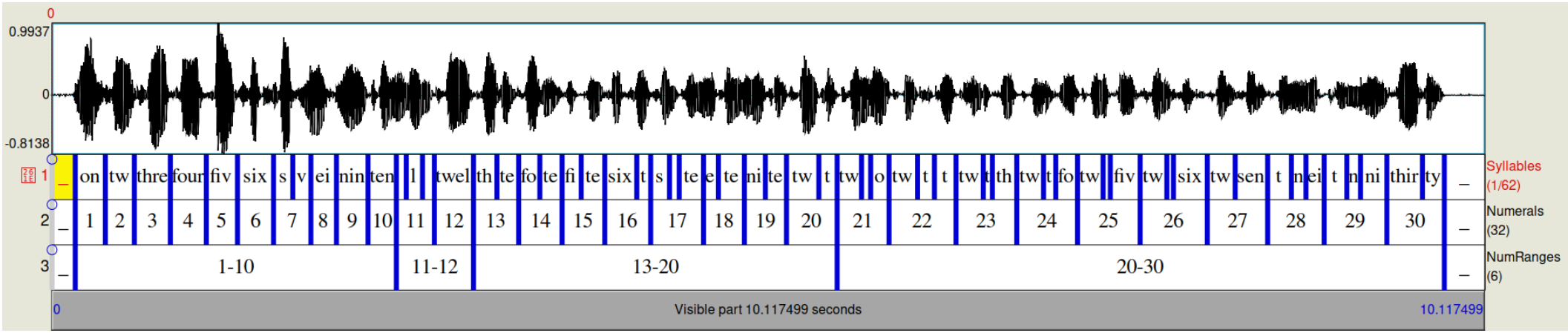
→

What comes out:

1. Linguistic (etc.) categories
2. Mapping of event intervals to categories
3. Durations of event intervals and a fortiori of categories

A text file (.TextGrid)

Event annotation with ‘Praat’: intervals and labels



Download Praat

https://www.fon.hum.uva.nl/praat/
https://www.praat.org

Data

Pre-recording

Design systematic filenames
Design data scenario
Prepare equipment and participants
You can record with Praat or Audacity

Recording

record with proper distance (1 span)
enough to drink

Post-recording

save with systematic filename
archive systematically

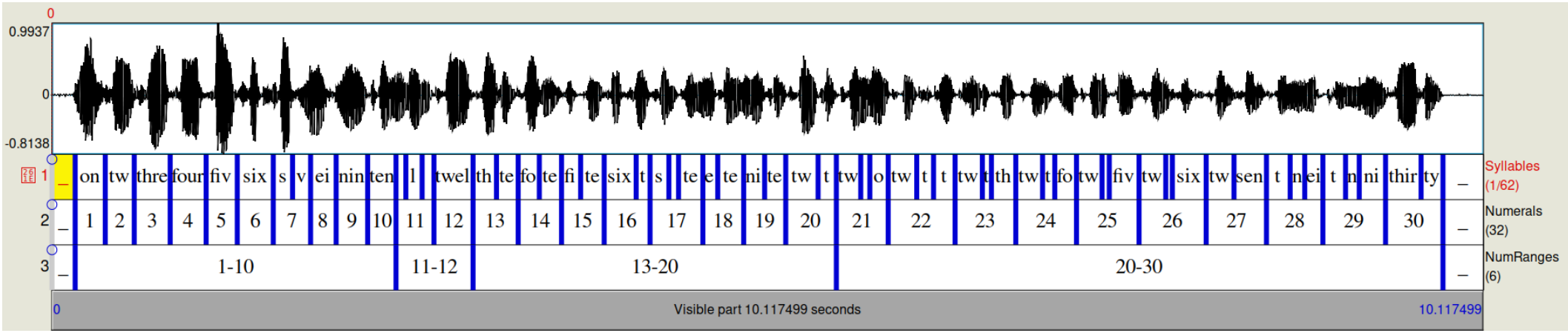
Annotate with Praat

Read into Praat
Select “Annotation”
Annotate.
Save with systematic filenameAnalysis
Convert the Praat format to CSV spreadsheet format
This can be done easily with a Python script. Contact me.

Analyse the spreadsheet file

With a spreadsheet.
With TGA (Time Group Analyser) online tool
http://wwwhomes.uni-bielefeld.de/gibbon/TGA/
speaker)

Event annotation with ‘Praat’: intervals and labels



What goes in:

1. Recording of speech signal
2. Perception of boundaries
   1. visual, acoustic
   2. waveform, spectrum, F0
3. Perception of event intervals as speech sounds

A speech file (.WAV, etc.)

→

What comes out:

1. Linguistic (etc.) categories
2. Mapping of event intervals to categories
3. Durations of event intervals and a fortiori of categories

A text file (.TextGrid)

Annotation Analysis

Event annotation with ‘Praat’: intervals and labels

What you get: TextGrid format:

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 10.117499318294643
tiers? <exists>
size = 3
item []:
    item [1]:
        class = "IntervalTier"
        name = "Syllables"
        xmin = 0
        xmax = 10.117499318294643
        intervals: size = 62
        intervals [1]:
            xmin = 0
            xmax = 0.1633600447898783
            text = "_"
        intervals [2]:
            xmin = 0.1633600447898783
            xmax = 0.37639112051691453
            text = "one"
        intervals [3]:
            xmin = 0.37639112051691453
            xmax = 0.5809009532148712
            text = "two"
        intervals [4]:
            xmin = 0.5809009532148712
            xmax = 0.8351180369157989
            text = "three"
        intervals [5]:
            xmin = 0.8351180369157989
            xmax = 1.0891320731869965
            text = "four"
        intervals [6]:
            xmin = 1.0891320731869965
            xmax = 1.3092641847716007
            text = "five"                    etc.
```

What can you do with the TextGrid format?

You can read it back into Praat.

But what about cross-platform use?

For example:

- Spreadsheet software:
  - LibreOffice Calc
  - Excel
- Statistical software:
  - R
  - MatLab
  - Stata
- General programming:
  - Python
- ...

Event annotation with ‘Praat’: intervals and labels

What you get: TextGrid format:

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 10.117499318294643
tiers? <exists>
size = 3
item []:
    item [1]:
        class = "IntervalTier"
        name = "Syllables"
        xmin = 0
        xmax = 10.117499318294643
        intervals: size = 62
        intervals [1]:
            xmin = 0
            xmax = 0.1633600447898783
            text = "_"
        intervals [2]:
            xmin = 0.1633600447898783
            xmax = 0.37639112051691453
            text = "one"
        intervals [3]:
            xmin = 0.37639112051691453
            xmax = 0.5809009532148712
            text = "two"
        intervals [4]:
            xmin = 0.5809009532148712
            xmax = 0.8351180369157989
            text = "three"
        intervals [5]:
            xmin = 0.8351180369157989
            xmax = 1.0891320731869965
            text = "four"
        intervals [6]:
            xmin = 1.0891320731869965
            xmax = 1.3092641847716007
            text = "five"                    etc.
```

What you need: CSV format:

Syllables	_	0.000	0.163	0.163
Syllables	one	0.163	0.376	0.213
Syllables	two	0.376	0.581	0.205
Syllables	three	0.581	0.835	0.254
Syllables	four	0.835	1.089	0.254
Syllables	five	1.089	1.309	0.220
Syllables	six	1.309	1.563	0.254
Syllables	se	1.563	1.700	0.137
Syllables	ven	1.700	1.828	0.128
Syllables	eight	1.828	2.008	0.180
Syllables	nine	2.008	2.231	0.223
Syllables	ten	2.231	2.431	0.200
Syllables	e	2.431	2.501	0.070
Syllables	le	2.501	2.616	0.115
Syllables	ven	2.616	2.699	0.083
Syllables	twelve	2.699	2.976	0.277
Syllables	thir	2.976	3.141	0.165
Syllables	teen	3.141	3.295	0.154
Syllables	four	3.295	3.444	0.149
Syllables	teen	3.444	3.601	0.157
Syllables	fif	3.601	3.736	0.135
Syllables	teen	3.736	3.902	0.166
Syllables	six	3.902	4.129	0.227
Syllables	teen	4.129	4.227	0.098
Syllables	se	4.227	4.360	0.133
Syllables	ven	4.360	4.428	0.068
Syllables	teen	4.428	4.600	0.172
Syllables	eigh	4.600	4.705	0.105
Syllables	teen	4.705	4.888	0.183
Syllables	nine	4.888	5.036	0.148
Syllables	teen	5.036	5.186	0.150
Syllables	twen	5.186	5.414	0.228
Syllables	ty	5.414	5.543	0.129
Syllables	twen	5.543	5.716	0.173
Syllables	ny	5.716	5.780	0.064
Syllables	one	5.780	5.910	0.130
Syllables	twen	5.910	6.112	0.202
Syllables	ty	6.112	6.230	0.118
Syllables	two	6.230	6.383	0.153
Syllables	twen	6.383	6.576	0.193
Syllables	ty	6.576	6.639	0.063
Syllables	three	6.639	6.812	0.173
Syllables	twen	6.812	7.001	0.189
Syllables	ty	7.001	7.087	0.086
Syllables	four	7.087	7.241	0.154
Syllables	twen	7.241	7.421	0.180
Syllables	ty	7.421	7.473	0.052
Syllables	five	7.473	7.683	0.210
Syllables	twen	7.683	7.872	0.189
Syllables	ty	7.872	7.919	0.047
Syllables	six	7.919	8.157	0.238
Syllables	twen	8.157	8.354	0.197
Syllables	sen	8.354	8.583	0.229
Syllables	twen	8.583	8.751	0.168
Syllables	ny	8.751	8.854	0.103
Syllables	eight	8.854	8.982	0.128
Syllables	twen	8.982	9.142	0.160
Syllables	ny	9.142	9.239	0.097
Syllables	nine	9.239	9.425	0.186
Syllables	thir	9.425	9.676	0.251
Syllables	ty	9.676	9.831	0.155
Syllables	_	9.831	10.117	0.286

Event annotation with ‘Praat’: intervals and labels

What you get: TextGrid format:

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 10.117499318294643
tiers? <exists>
size = 3
item []:
    item [1]:
        class = "IntervalTier"
        name = "Syllables"
        xmin = 0
        xmax = 10.117499318294643
        intervals: size = 62
        intervals [1]:
            xmin = 0
            xmax = 0.1633600447898783
            text = "_"
        intervals [2]:
            xmin = 0.1633600447898783
            xmax = 0.37639112051691453
            text = "one"
        intervals [3]:
            xmin = 0.37639112051691453
            xmax = 0.5809009532148712
            text = "two"
        intervals [4]:
            xmin = 0.5809009532148712
            xmax = 0.8351180369157989
            text = "three"
        intervals [5]:
            xmin = 0.8351180369157989
            xmax = 1.0891320731869965
            text = "four"
        intervals [6]:
            xmin = 1.0891320731869965
            xmax = 1.3092641847716007
            text = "five"                    etc.
```

What can you do with the TextGrid format?

You can read it back into Praat.

But what about cross-platform use?

For example:

- Spreadsheet software:
  - LibreOffice Calc
  - Excel
- Statistical software:
  - R
  - MatLab
  - Stata
- General programming:
  - Python
- ...

How can you get CSV format?

With a Python script...

Python script: **textgridtier2csv.py**

```
#!/usr/bin/python3

# textgrid2csv.py, D. Gibbon, 2015.02.12
# Convert Praat TextGrid to CSV

import os, re, sys
TAB = '\t'; NL = '\n'

# Check input from CLI
if len(sys.argv) < 3:

print('Usage:',sys.argv[0],'<filename.TextGrid>')
    exit()
textgridfile = sys.argv[1]
if not os.path.isfile(textgridfile):
    print('File', textgridfile, 'does not
exist.')
    exit()
csvfile = re.sub('.TextGrid','.csv',textgridfile)

# Preprocess textgrid format
textgrid = open(textgridfile,
'r' ).read().split(NL)
nugrid = []
for line in textgrid:
    line = re.sub(' *$','',line)    # Delete
final spaces
    line = re.sub('^ *','',line)    # Delete
initial spaces
    line = re.sub('\"','',line)     # Delete
quotes
    if line != '':
        nugrid += [line]
```

```
# Initialise CSV text string
csvstring = textgridfile + '\n'
csvstring += TAB.join(
    ['Tier', 'Label', 'xmin', 'xmax', 'xdiff'] ) +
NL

# Initialise row identifier flag
rowflag = False

# Loop through lines in preprocessed textgrid
for line in nugrid:
    if 'name = ' in line:                # Get tier
name
        name = line.split(" ")[-1]
    if 'intervals [' in line:              # Skip
the header
     rowflag = True
    if rowflag and 'xmin' in line:
        xmin = '%.3f'%float(line.split(' ')[-1])
    if rowflag and 'xmax' in line:
        xmax = '%.3f'%float(line.split(' ')[-1])
    if rowflag and "text" in line:
        text = line.split(" ")[-1]
        xdiff = '%.3f'%(float(xmax)-float(xmin))
        row = TAB.join([name, text, xmin, xmax,
xdiff])
        csvstring += row + NL
        rowflag = False

# Output CSV file
open(csvfile,'w').write(csvstring)
```

What can we do with a CSV annotation file?

Descriptive statistics:

- standard deviation
- normalised Pairwise Variability Index

Other spreadsheet calculations

Duration variability as a function, with visualisation

Event annotation with ‘Praat’: intervals and labels

What you get: TextGrid format:

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 10.117499318294643
tiers? <exists>
size = 3
item []:
    item [1]:
        class = "IntervalTier"
        name = "Syllables"
        xmin = 0
        xmax = 10.117499318294643
        intervals: size = 62
        intervals [1]:
            xmin = 0
            xmax = 0.1633600447898783
            text = "_"
        intervals [2]:
            xmin = 0.1633600447898783
            xmax = 0.37639112051691453
            text = "one"
        intervals [3]:
            xmin = 0.37639112051691453
            xmax = 0.5809009532148712
            text = "two"
        intervals [4]:
            xmin = 0.5809009532148712
            xmax = 0.8351180369157989
            text = "three"
        intervals [5]:
            xmin = 0.8351180369157989
            xmax = 1.0891320731869965
            text = "four"
        intervals [6]:
            xmin = 1.0891320731869965
            xmax = 1.3092641847716007
            text = "five"                    etc.
```

What you need: CSV format:

Syllables	_	0.000	0.163	0.163
Syllables	one	0.163	0.376	0.213
Syllables	two	0.376	0.581	0.205
Syllables	three	0.581	0.835	0.254
Syllables	four	0.835	1.089	0.254
Syllables	five	1.089	1.309	0.220
Syllables	six	1.309	1.563	0.254
Syllables	se	1.563	1.700	0.137
Syllables	ven	1.700	1.828	0.128
Syllables	eight	1.828	2.008	0.180
Syllables	nine	2.008	2.231	0.223
Syllables	ten	2.231	2.431	0.200
Syllables	e	2.431	2.501	0.070
Syllables	le	2.501	2.616	0.115
Syllables	ven	2.616	2.699	0.083
Syllables	twelve	2.699	2.976	0.277
Syllables	thir	2.976	3.141	0.165
Syllables	teen	3.141	3.295	0.154
Syllables	four	3.295	3.444	0.149
Syllables	teen	3.444	3.601	0.157
Syllables	fif	3.601	3.736	0.135
Syllables	teen	3.736	3.902	0.166
Syllables	six	3.902	4.129	0.227
Syllables	teen	4.129	4.227	0.098
Syllables	se	4.227	4.360	0.133
Syllables	ven	4.360	4.428	0.068
Syllables	teen	4.428	4.600	0.172
Syllables	eigh	4.600	4.705	0.105
Syllables	teen	4.705	4.888	0.183
Syllables	nine	4.888	5.036	0.148
Syllables	teen	5.036	5.186	0.150
Syllables	twen	5.186	5.414	0.228
Syllables	ty	5.414	5.543	0.129
Syllables	twen	5.543	5.716	0.173
Syllables	ny	5.716	5.780	0.064
Syllables	one	5.780	5.910	0.130
Syllables	twen	5.910	6.112	0.202
Syllables	ty	6.112	6.230	0.118
Syllables	two	6.230	6.383	0.153
Syllables	twen	6.383	6.576	0.193
Syllables	ty	6.576	6.639	0.063
Syllables	three	6.639	6.812	0.173
Syllables	twen	6.812	7.001	0.189
Syllables	ty	7.001	7.087	0.086
Syllables	four	7.087	7.241	0.154
Syllables	twen	7.241	7.421	0.180
Syllables	ty	7.421	7.473	0.052
Syllables	five	7.473	7.683	0.210
Syllables	twen	7.683	7.872	0.189
Syllables	ty	7.872	7.919	0.047
Syllables	six	7.919	8.157	0.238
Syllables	twen	8.157	8.354	0.197
Syllables	sen	8.354	8.583	0.229
Syllables	twen	8.583	8.751	0.168
Syllables	ny	8.751	8.854	0.103
Syllables	eight	8.854	8.982	0.128
Syllables	twen	8.982	9.142	0.160
Syllables	ny	9.142	9.239	0.097
Syllables	nine	9.239	9.425	0.186
Syllables	thir	9.425	9.676	0.251
Syllables	ty	9.676	9.831	0.155
Syllables	_	9.831	10.117	0.286

Descriptive statistics: static sets – *nPVI*

Annotation: segmentation and classification

(‘labelling’)

and the search for isochrony*

a hybrid deductive-inductive method

*__isochrony:__ equal timing, for example as an idealised phonetic interpretation of rhythm units like syllables or stress groups

Interval analysis and PVI – the search for isochrony

$$rPVI(D) = \sum |d_k - d_{k+1}| / (n-1)$$

raw Pairwise Variability Index

The measure defines an overall 'next-door neighbour distance'.

Interval analysis and PVI – the search for isochrony

$$rPVI(D) = \sum |d_k - d_{k+1}| / (n-1)$$

raw Pairwise Variability Index

$$nPVI(D) = 100 \times \sum \left| \frac{d_k - d_{k+1}}{(d_k + d_{k+1})/2} \right| / (n-1)$$

normalised Pairwise Variability Index

Interval analysis and PVI – the search for isochrony

$$rPVI(D) = \sum |d_k - d_{k+1}| / (n-1)$$

raw Pairwise Variability Index

$$nPVI(D) = 100 \times \sum \left| \frac{d_k - d_{k+1}}{(d_k + d_{k+1})/2} \right| / (n-1)$$

normalised Pairwise Variability Index

Similarity to Manhattan Distance

$$MD(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

Similarity to Canberra Distance (Normalised Manhattan Distance)

$$NormMD(x, y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

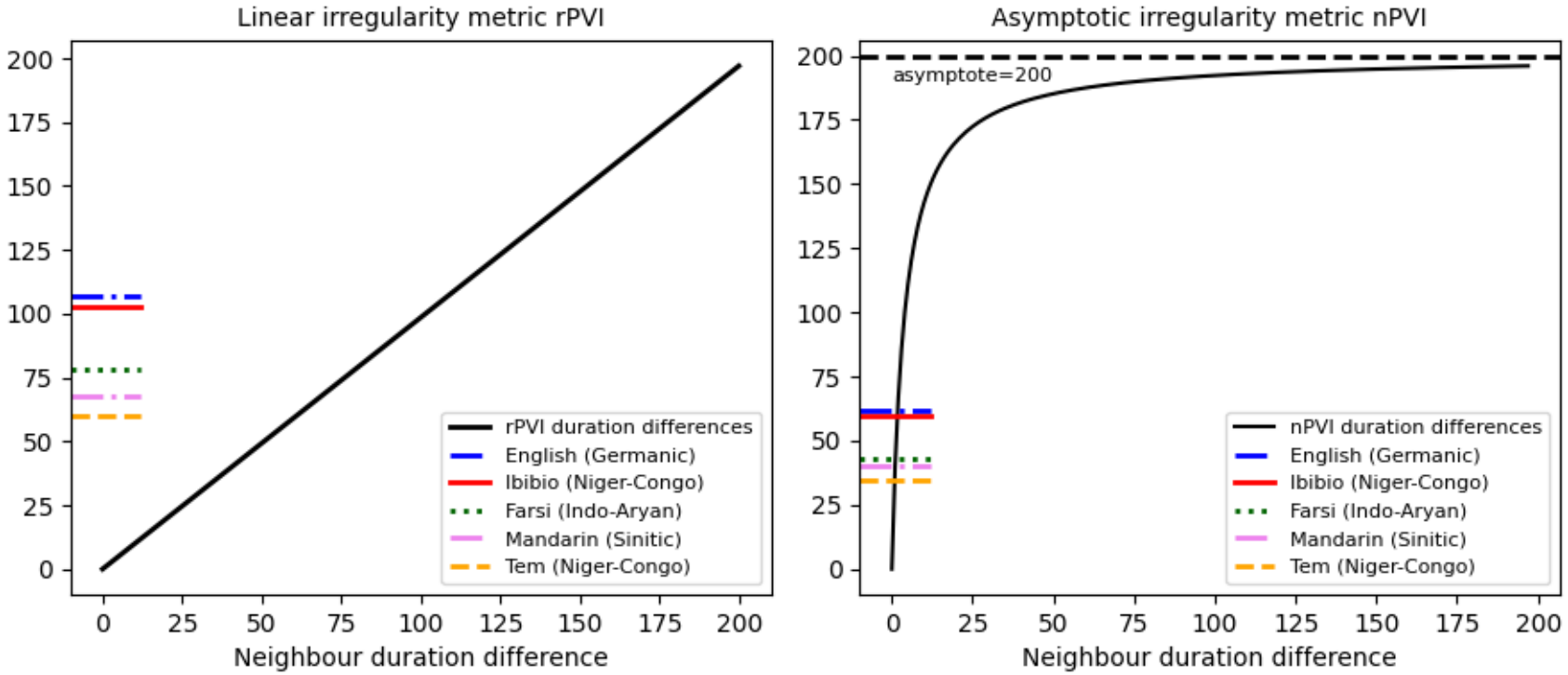
Interval analysis and PVI – the search for isochrony

$$rPVI(D) = \sum |d_k - d_{k+1}| / (n-1)$$

raw Pairwise Variability Index

$$nPVI(D) = 100 \times \sum \left| \frac{d_k - d_{k+1}}{(d_k + d_{k+1})/2} \right| / (n-1)$$

normalised Pairwise Variability Index



Interval analysis and PVI – the search for isochrony

$$rPVI(D) = \sum |d_k - d_{k+1}| / (n-1)$$

Similarity to Manhattan Distance

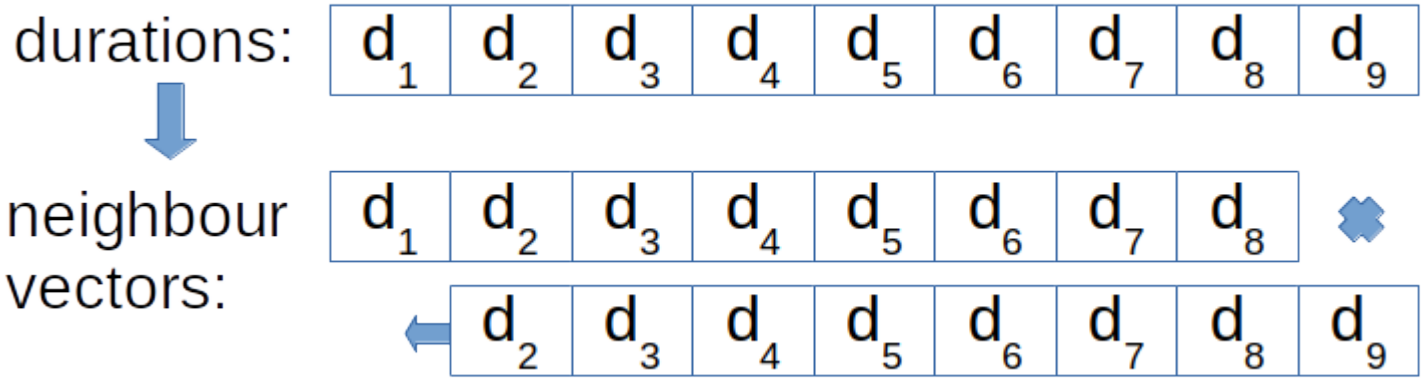
$$MD(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

$$nPVI(D) = 100 \times \sum \left| \frac{d_k - d_{k+1}}{(d_k + d_{k+1})/2} \right| / (n-1)$$

Similarity to Canberra Distance
(Normalised Manhattan Distance)

$$NormMD(x, y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

These measures show an overall 'next-door neighbour distance':



Assessment of interval duration measures

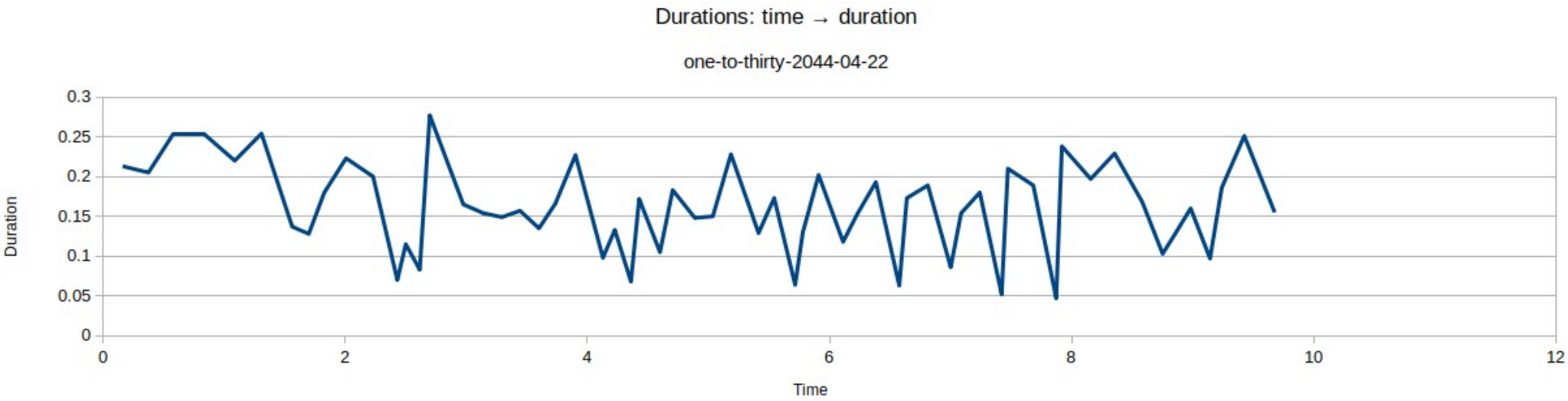
The interval duration measures can be useful heuristic measures.

They have the following properties:

- the interval duration measures ignore the alternation property of rhythm by using absolute values, and yielding the same values for positive and negative differences;
- they are often called ‘rhythm metrics’, but the interval duration measures calculate irregularity, not rhythmicality;
- the ‘irregularity measures’ do not model rhythm and do not constitute either a model or a theory of rhythm;
- through the annotation procedure the signal is filtered through the perceptual skills of an annotator and the signal is not analysed directly.

Duration variability as a function

Libre Office Calc (Excel, R, MatLab, Python, Stata...)

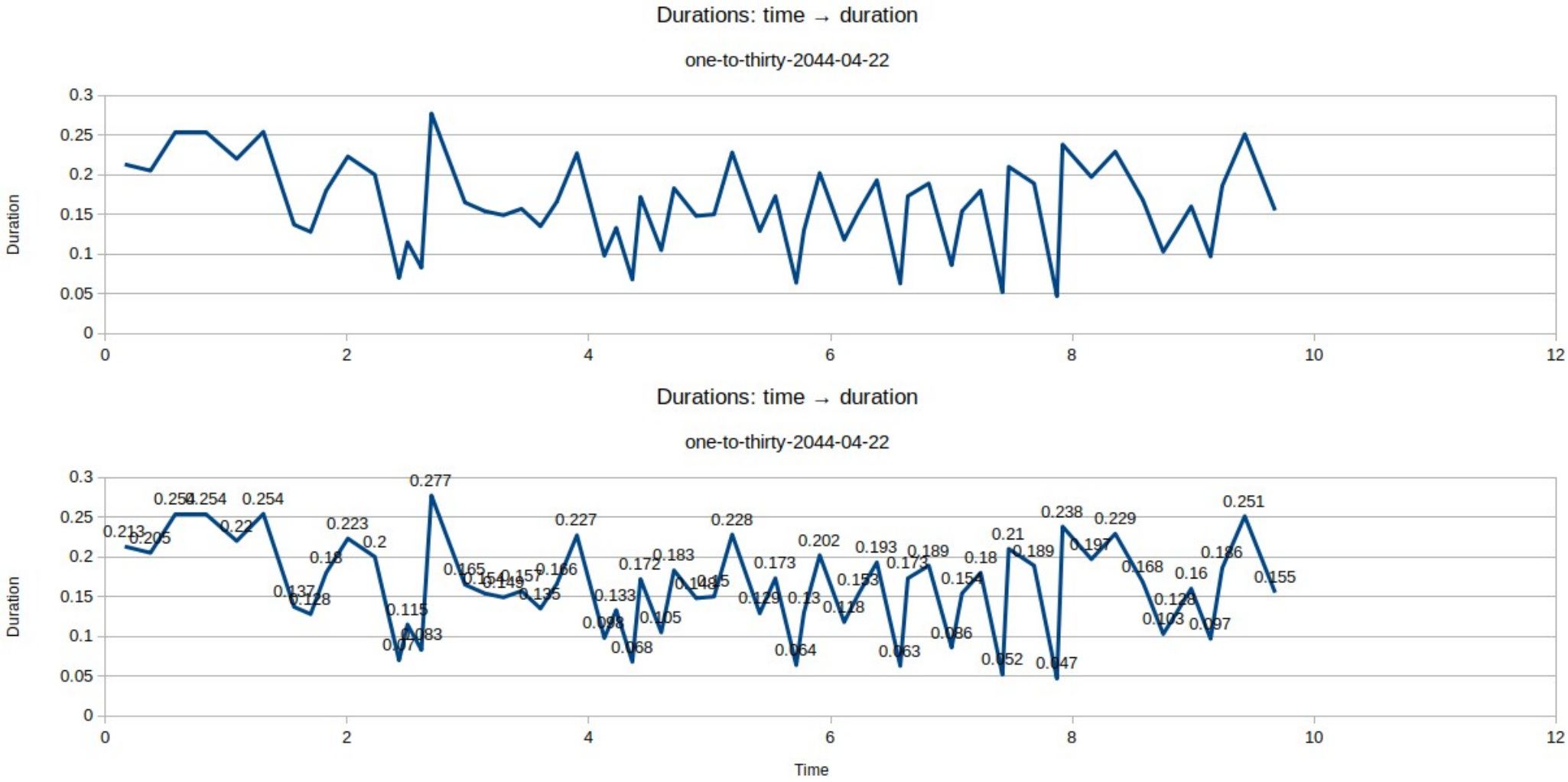


What does this tell you?
1. Durations are not constant
2. Duration differences are not constant
3. Durations vary through an utterance
4. Morphology is also relevant

So be careful with generalisations which give you a single index, like
1. Standard Deviation
2. nPVI

and ignore speech variability in languages

Libre Office Calc (Excel, R, MatLab, Python, Stata...)



What does this tell you?

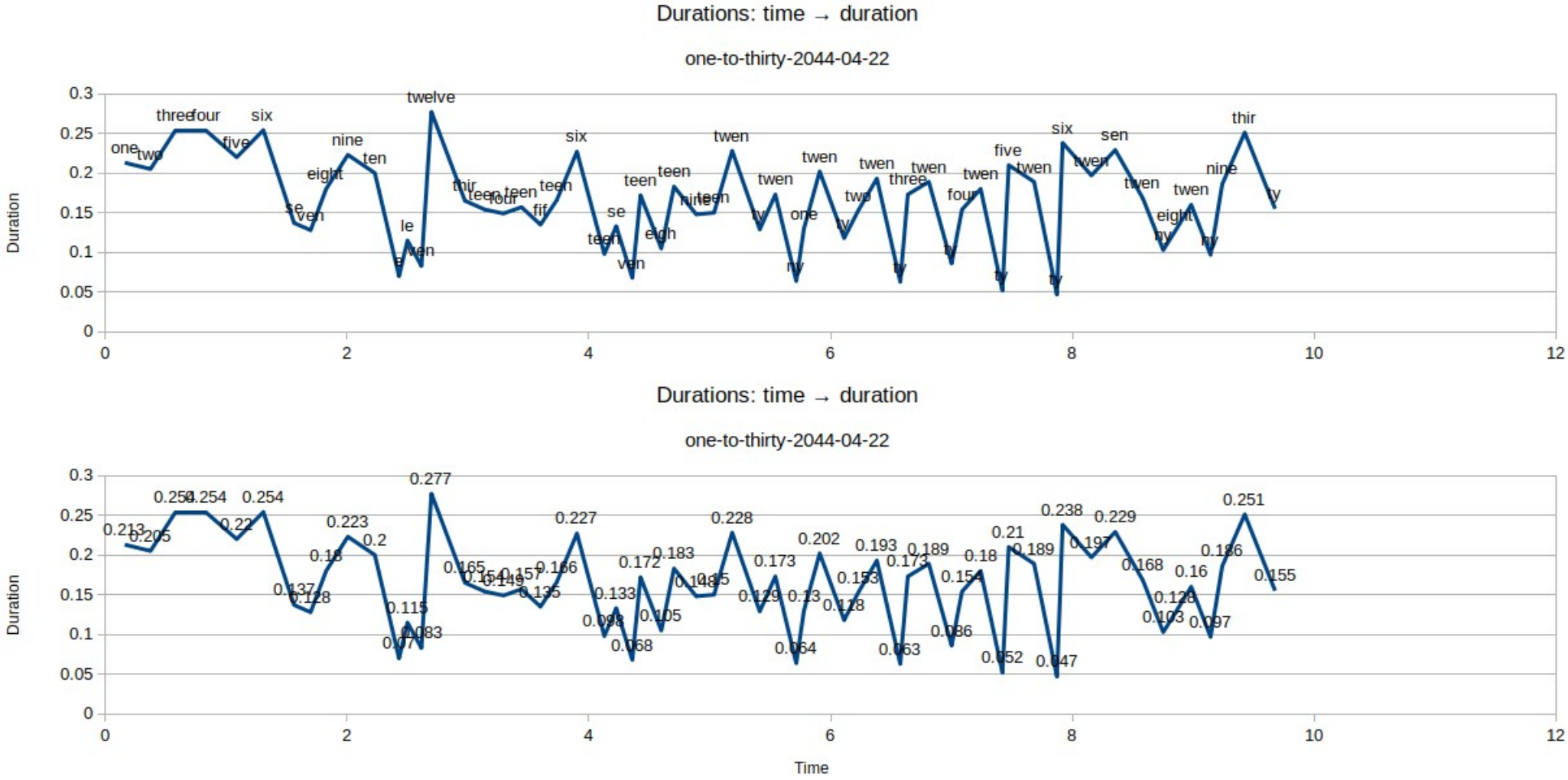
1. Durations are not constant
2. Duration differences are not constant
3. Durations vary through an utterance
4. Morphology is also relevant

So be careful with generalisations which give you a single index, like

1. Standard Deviation
2. nPVI

and ignore speech variability in languages

Libre Office Calc (Excel, R, MatLab, Python, Stata...)



What does this tell you?

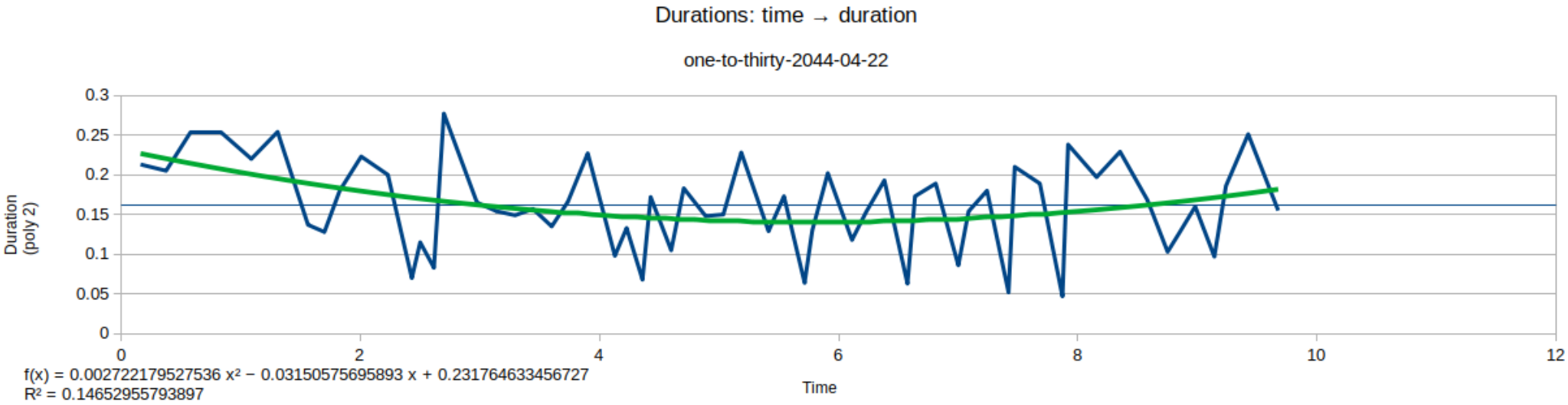
1. Durations are not constant
2. Duration differences are not constant
3. Durations vary through an utterance
4. Morphology is also relevant

So be careful with generalisations which give you a single index, like

1. Standard Deviation
2. nPVI

and ignore speech variability in languages

Libre Office Calc (Excel, R, MatLab, Python, Stata...)

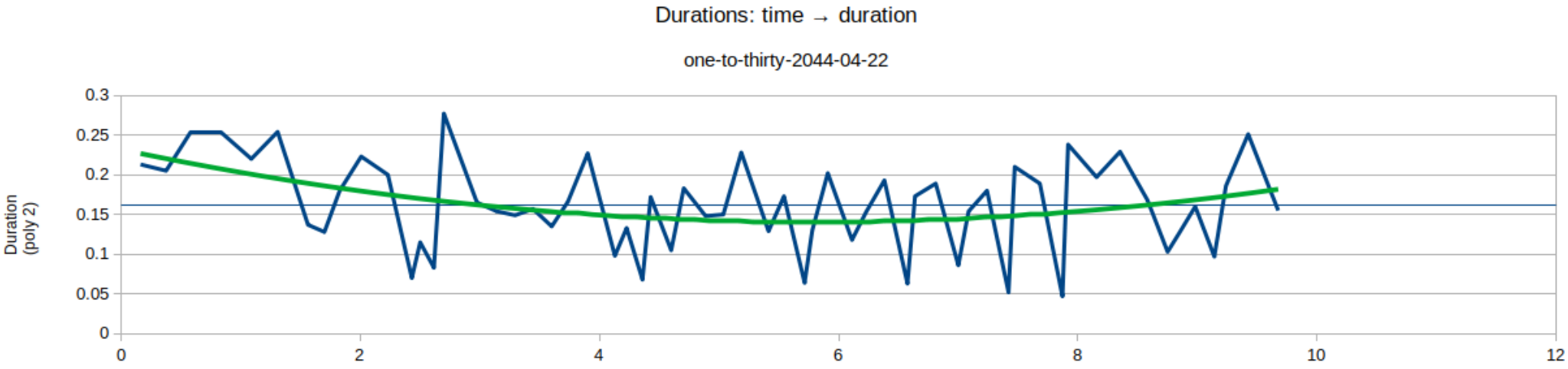


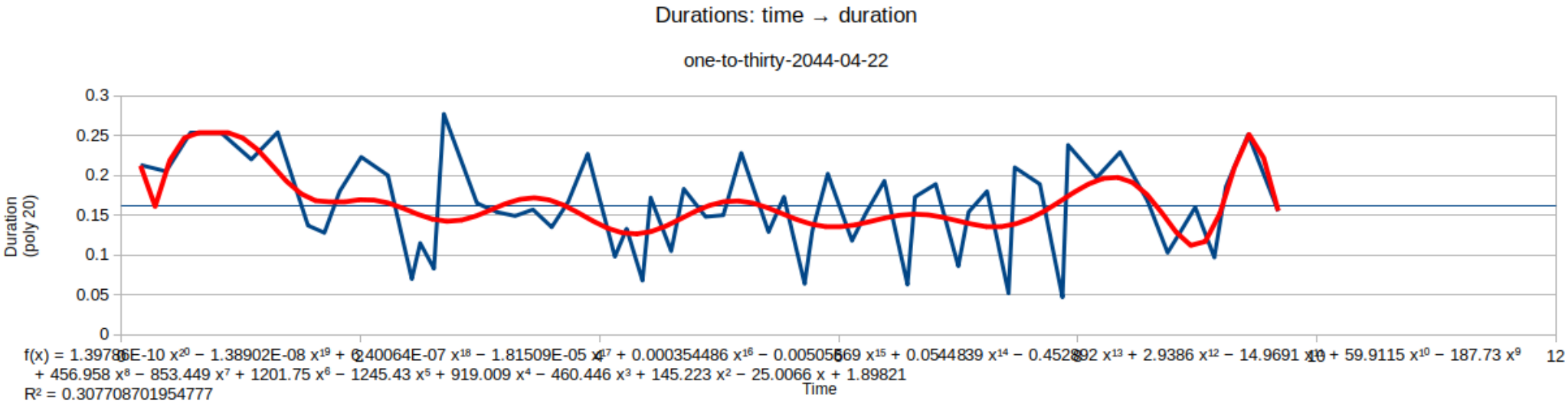
What does this tell you?
1. Durations are not constant
2. Duration differences are not constant
3. Durations vary through an utterance
4. Morphology is also relevant

So be careful with generalisations which give you a single index, like
1. Standard Deviation
2. nPVI

and ignore speech variability in languages

Libre Office Calc (Excel, R, MatLab, Python, Stata...)





What does this tell you?

1. Durations are not constant
2. Duration differences are not constant
3. Durations vary through an utterance
4. Morphology is also relevant

So be careful with generalisations which give you a single index, like

1. Standard Deviation
2. nPVI

and ignore speech variability in languages

For Wednesday and Friday

https://github.com/dafyddg/RFA/

For Pythonistas:

To get the code: git clone http://github.com/dafyddg/RFA/
To use the code:

```
pip install numpy
pip install matplotlib
pip install scipy
pip install graphviz
```

Also:

install GraphViz for your OS: https://graphviz.org/download/