projectname

Joanna Doh

Table of contents

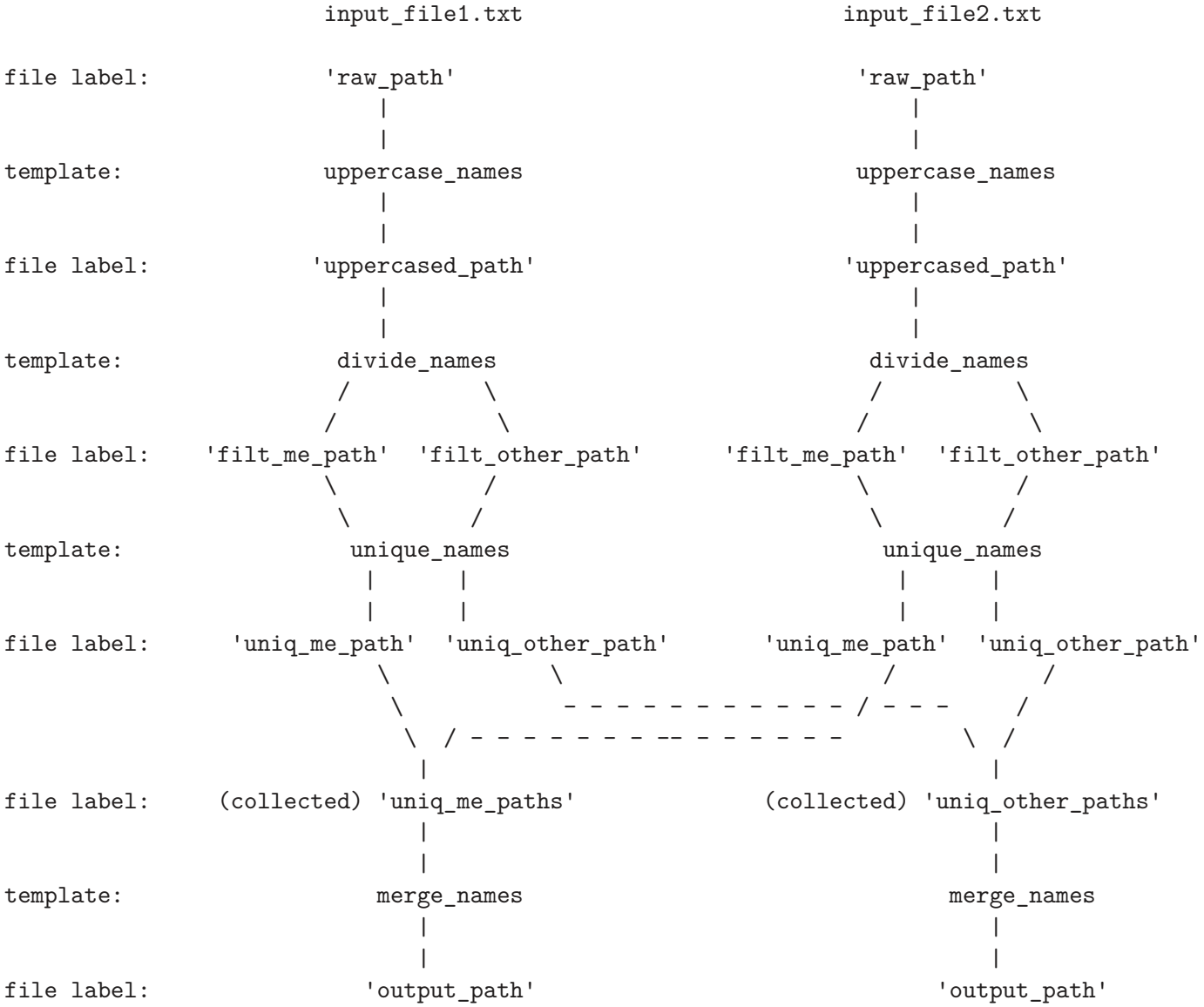
Title

These pages are generated from a Git repository…

References

1. GWF workflow

Example workflow using mapping between intput and output of each target. It is made to show all the ways information may be passed through an workflow.



1.1. Imports and utility functions

```
from pathlib import Path
from gwf import Workflow, AnonymousTarget
from gwf.workflow import collect
```

Instantiate the workflow with the name of the project folder:

```
# instantiate the workflow
gwf = Workflow(defaults={'account': 'your-project-folder-name'}
```

Utility functions:

```
# utility function
def modify_path(path, **kwargs):
    """
    Utility function for modifying file paths substituting
    the directory (dir), base name (base), or file suffix (suff
    """
    for key in ['dir', 'base', 'suffix']:
        kwargs.setdefault(key, None)
    assert len(kwargs) == 3

    par, name = os.path.split(path)
    name_no_suffix, suf = os.path.splitext(name)
    if type(kwargs['suffix']) is str:
        suf = kwargs['suffix']
    if kwargs['dir'] is not None:
        par = kwargs['dir']
    if kwargs['base'] is not None:
        name_no_suffix = kwargs['base']

    new_path = os.path.join(par, name_no_suffix + suf)
    if type(kwargs['suffix']) is tuple:
        assert len(kwargs['suffix']) == 2
        new_path, nsubs = re.subn(r'{}$'.format(kwargs['suffix'
        assert nsubs == 1, nsubs
    return new_path
```

1.2. Template functions:

```
# task template function
def uppercase_names(raw_path):
    """
    Formats names to uppercase.
    """
    # dir for files produces by task
    output_dir = 'steps/upper_cased'
    # path of output file
    uppercased_path = modify_path(raw_path, dir=output_dir, suffix='_uppercased.txt')

    # input specification
    inputs = [raw_path]
    # output specification mapping a label to each file
    outputs = {'uppercased_path': uppercased_path}
    # resource specification
    options = {'memory': '8g', 'walltime': '00:10:00'}

    # tmporary output file path
    tmp_uppercased_path = modify_path(raw_path, dir='/tmp')

    # commands to run in task (bash script)
    # we write to a tmp file and move that to the output directory
    # only if the command succeds (the && takes care of that)
    spec = f"""
    mkdir -p {output_dir}
    cat {raw_path} | tr [:lower:] [:upper:] > {tmp_uppercased_path} &&
        mv {tmp_uppercased_path} {uppercased_path}
    """
    # return target
    return AnonymousTarget(inputs=inputs, outputs=outputs, options=options, spec=spec)


# task template function
def divide_names(uppercased_path, me=None):
    """
    Splits names into two files. One with my name and one with other names.
    """
    # uppercased version of the me argument
    uppercased_me = me.upper()
```

```
    # dir for files produces by task
    output_dir = 'steps/filtered_names'
    # path of output file with names matching me
    filt_me_path = modify_path(uppercased_path, dir=output_dir,
    # path of output file with other names
    filt_other_path = modify_path(uppercased_path, dir=output_d

    # input specification
    inputs = [uppercased_path]
    # output specification mapping a label to each file
    outputs = {'filt_me_path': filt_me_path, 'filt_other_path':
    # resource specification
    options = {'memory': '8g', 'walltime': '00:10:00'}

    # tmporary output file paths
    tmp_filt_me_path = modify_path(filt_me_path, dir='/tmp')
    tmp_filt_other_path = modify_path(filt_other_path, dir='/tm

    # commands to run in task (bash script)
    # we write to tmp files and move them to the output directo
    # only if the command succeds (the && takes care of that)
    spec = f"""
    mkdir -p {output_dir}
    grep {uppercased_me} {uppercased_path} > {tmp_filt_me_path}
        grep -v {uppercased_me} {uppercased_path} > {tmp_filt_o
        mv {tmp_filt_me_path} {filt_me_path} &&
        mv {tmp_filt_other_path} {filt_other_path}
    """
    # return target
    return AnonymousTarget(inputs=inputs, outputs=outputs, opti


# task template function
def unique_names(filt_me_path, filt_other_path):
    """
    Extracts unique names from a file.
    """
    # dir for files produces by task
    output_dir = 'steps/unique_names'
    # path of output file with unique names matching me
    uniq_me_path = modify_path(filt_me_path, dir=output_dir, su
    # path of output file with unique other names
    uniq_other_path = modify_path(filt_other_path, dir=output_d
```

```
    # input specification
    inputs = [filt_me_path, filt_other_path]
    # output specification mapping a label to each file
    outputs = {'unique_me_path': uniq_me_path, 'unique_other_path': uniq_other_path}
    # resource specification
    options = {'memory': '8g', 'walltime': '00:10:00'}

    # tmporary output file paths
    tmp_uniq_me_path = modify_path(uniq_me_path, dir='/tmp')
    tmp_uniq_other_path = modify_path(uniq_other_path, dir='/tmp')

    # commands to run in task (bash script)
    # we write to tmp files and move them to the output directory
    # only if the command succeds (the && takes care of that)
    spec = f"""
    mkdir -p {output_dir}
    sort {filt_me_path} | uniq > {tmp_uniq_me_path} &&
        sort {filt_other_path} | uniq > {tmp_uniq_other_path} &&
        mv {tmp_uniq_me_path} {uniq_me_path} &&
        mv {tmp_uniq_other_path} {uniq_other_path}
    """
    # return target
    return AnonymousTarget(inputs=inputs, outputs=outputs, options=options, spec=spec)


# task template function
def merge_names(paths, output_path):
    """
    Merges names from many files.
    """
    # dir for files produces by task
    output_dir = modify_path(output_path, base='', suffix='')

    # input specification
    inputs = [paths]
    # output specification mapping a label to the file
    outputs = {'path': output_path}

    # tmporary output file path
    tmp_output_path =  modify_path(output_path, dir='/tmp')

    # resource specification
```

```
    options = {'memory': '8g', 'walltime': '00:10:00'}

    # commands to run in task (bash script)
    # we write to tmp files and move them to the output directo
    # only if the command succeds (the && takes care of that)
    spec = f"""
    mkdir -p {output_dir}
    cat {' '.join(paths)} > {tmp_output_path} &&
        mv {tmp_output_path} {output_path}
    """
    # return target
    return AnonymousTarget(inputs=inputs, outputs=outputs, opti
```

1.3. Workflow:

```
# instantiate the workflow
gwf = Workflow(defaults={'account': 'your-project-folder-name'}

# input files for workflow
input_file_names = ['data/input_file1.txt', 'data/input_file2.t

# workflow parameter
myname = 'Kasper'

# run an uppercase_names task for each input file
uppercase_names_targets = gwf.map(uppercase_names, input_file_n

# run an divide_names task for each output file from uppercase_
filter_names_targets = gwf.map(divide_names, uppercase_names_ta

# run an unique_names task for each output file from divide_nam
unique_names_targets = gwf.map(unique_names, filter_names_targe

# collect the outputs labelled 'unique_me_path' from all the ou
collected_outputs = collect(unique_names_targets.outputs, ['uni

# create a single task to merge all those files into one
merge_me_target = gwf.target_from_template(
    'merge_not_me_name_files',
    merge_names(collected_outputs['unique_me_paths'], "results/
```

```
    )

# collect the outputs labelled 'unique_other_path' from all the outputs of unique_names
collected_outputs = collect(unique_names_targets.outputs, ['unique_other_path'])

# create a single task to merge all those files into one
merge_other_target = gwf.target_from_template(
    'merge_me_name_files',
    merge_names(collected_outputs['unique_other_paths'], "results/merged_not_me_names.txt")
    )
```

2. GWF workflow for notebooks

```
from pathlib import Path
from gwf import Workflow, AnonymousTarget
from gwf.workflow import collect
```

Instantiate the workflow with the name of the project folder.

```
# instantiate the workflow
gwf = Workflow(defaults={'account': 'your-project-folder-name'})
```

2.1. Template functions:

```
# task template function
def run_notebook(path, memory='8g', walltime='00:10:00', cores=1):
    """
    Executes a notebook inplace and saves the output.
    """
    # path of output sentinel file
    sentinel = path.parent / '.' + path.name

    # input specification
    inputs = [path]
    # output specification mapping a label to each file
    outputs = {'sentinel': sentinel}
    # resource specification
    options = {'memory': memory, 'walltime': walltime, 'cores': cores}

    # commands to run in task (bash script)
    spec = f"""
```

```
    jupyter nbconvert --to notebook --execute --inplace path
    """
    # return target
    return AnonymousTarget(inputs=inputs, outputs=outputs, opti
```

2.2. Workflow:

Executes all notebooks in the `notebooks` directory in sorted order.

```
dependencies = []
# run notebooks in sorted order nb01_, nb02_, ...
for notebook in Path('notebooks/**.ipynb'):
    # run a notebook
    target = run_notebook(notebook, dependencies)
    # make each notebook dependent on all previous
    dependencies.append(target.outputs['sentinel'])
    # add target to workflow
    gwf.target(target)
```

Part I.

Notebooks

1. Workplace interaction

Import some plotting libraries and set some defaults:

```
import sys
import numpy as np
import pandas as pd
from IPython.display import display, Markdown
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
sns.set_style("whitegrid")

import random
random_seed = 5

sys.path.append('..')
from global_params import load_params
```

> **Tip:**
> Producing figures in svg format (scalable vector graphics) makes for sharp plots on webpages. However, if you make plots with thousands of observations you should set this to `'png'` instead:

```
%config InlineBackend.figure_formats = ['svg']
```

> **Tip:**
> Some values apply globally to your analysis. E.g., sample sizes, cutoffs, names, rates, etc. Keeping those in a a yml file like `interaction_params.yml` and loading them in each notebook avoids the risk of manually adding/updating them in each notebook where they are used. You can use the `load_globals` function imported above from `global_params.py` to produce an object holding all the global values:

```
params = load_params('../global_params.yml')
params
```

```
{'questions': ['Blah blah blah',
               'Blah blah blah',
               'Blah blah blah',
               'Blah blah blah'],
 'sample_size': 24}
```

```
params.sample_size
```

```
24
```

1.1. Sampling

```
subjects = pd.read_csv('../data/data_table.csv')
assert subjects.index.size == params.sample_size
```

> **Tip:**
> By adding a label and caption to a cell displaying a table, you can refer to that table elsewhere and insert it in a manuscript.

```
subjects
```

Table 1.1.: People included in the analysis.

	name	age	sex	position	nationality
0	Julie	27	F	PhDstudent	DK
1	Thomas	33	M	Postdoc	GB
2	Emilie	23	F	PhDstudent	CH
3	Sofie	31	F	Postdoc	DK
4	Sara	29	F	Postdoc	US
5	Cecilie	34	F	Postdoc	DK
6	Anders	32	M	PhDstudent	UK
7	Emma	42	F	Professor	DK
8	Caroline	31	F	PhDstudent	DK
9	Laura	30	F	Postdoc	DK

Table 1.1.: People included in the analysis.

	name	age	sex	position	nationality
10	Mikkel	33	M	Postdoc	NL
11	Jens	27	M	PhDstudent	DK
12	Andreas	29	M	PhDstudent	DK
13	Jakob	28	M	PhDstudent	DK
14	Mathilde	61	F	Professor	DK
15	Katrine	35	F	Postdoc	DK
16	Poul	30	M	Postdoc	DK
17	Anna	26	F	PhDstudent	DK
18	Peter	42	M	Professor	GB
19	Ida	53	F	Postdoc	DK
20	Freja	30	F	Postdoc	DK
21	Maria	39	F	Professor	UK
22	Amalie	29	F	PhDstudent	DK
23	Camilla	35	F	Postdoc	DK

Tip:
By generaing markdown for descriptions that will eventually end up in the manuscript, you can imbed python values. It also ensures that the manuscript exactly reflects the notebook.

The 24 subjects from workplaces in Denmark were interviewed .... blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah

1.2. Interviews

The 24 subjects were asked to score the follow statements:

1. Blah blah blah
2. Blah blah blah
3. Blah blah blah
4. Blah blah blah

In interviewed `{python} params.sample_size` workplace individuals were interviewed by .... blah, blah, blah, blah, blah, blah, blah, blah, blah, blah, blah, blah, blah, blah, blah, blah,

blah, blah, blah, blah, blah, blah, blah, blah, blah, blah, blah, blah,

```
df = pd.DataFrame({'name': subjects.name,
                   'seniority': np.random.randint(0, 5, params.
                   'age': subjects.age})
df['informality'] = np.random.normal(loc=10, scale=1, size=para
df
```

	name	seniority	age	informality
0	Julie	2	27	10.061504
1	Thomas	2	33	9.795845
2	Emilie	0	23	10.704674
3	Sofie	4	31	9.995386
4	Sara	1	29	9.384324
5	Cecilie	4	34	9.617037
6	Anders	3	32	8.822115
7	Emma	3	42	8.654268
8	Caroline	0	31	10.571754
9	Laura	3	30	12.278083
10	Mikkel	2	33	10.181373
11	Jens	4	27	11.043315
12	Andreas	3	29	10.194166
13	Jakob	2	28	9.740300
14	Mathilde	2	61	8.671047
15	Katrine	1	35	10.177327
16	Poul	4	30	9.894090
17	Anna	1	26	7.958770
18	Peter	2	42	11.287880
19	Ida	0	53	10.254190
20	Freja	0	30	11.093569
21	Maria	0	39	10.223915
22	Amalie	1	29	9.613552
23	Camilla	0	35	9.975990

```
sns.scatterplot(x='age', y='informality', data=df, hue='seniori
plt.ylabel('How informal you can be')
plt.xlabel('Age')
plt.legend(title='Seniority', loc='lower right', labels=['Under
plt.ylim(bottom=0) ;
```

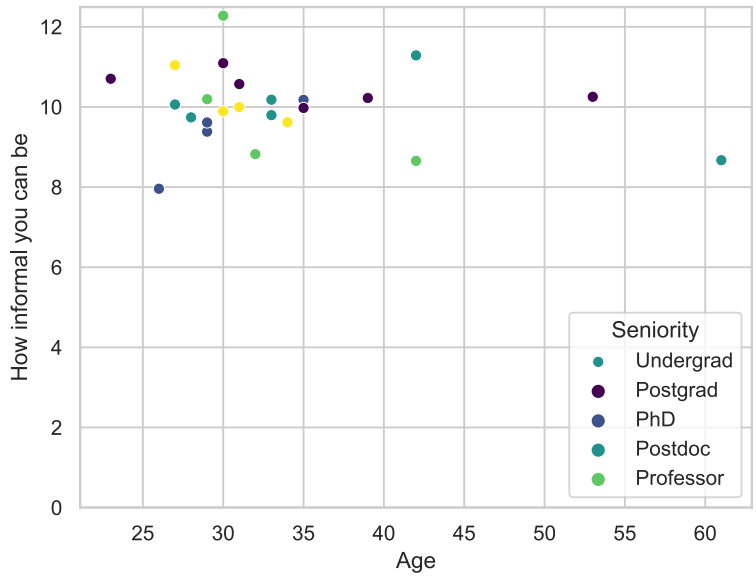


Figure 1.1.: Interaction among Danes: How Danes interact is has very little to do with age and seniority, compared to most other contries.

Seems Danish people act very informally unaffected by age and seniority.

```
informality_age_cor = df.informality.corr(df.age)
informality_age_cor
```

```
-0.1949220780248677
```

```
informality_seniority_cor = df.informality.corr(df.seniority)
informality_seniority_cor
```

```
-0.05515869516915789
```

The correlation between informality and age was -0.195 and the correlation between informality and seniority was -0.055.

```
sns.lmplot(x='age', y='informality', data=df, hue='seniority', palette='viridis')
plt.ylabel('How informal you can be')
plt.xlabel('Age') ;
```

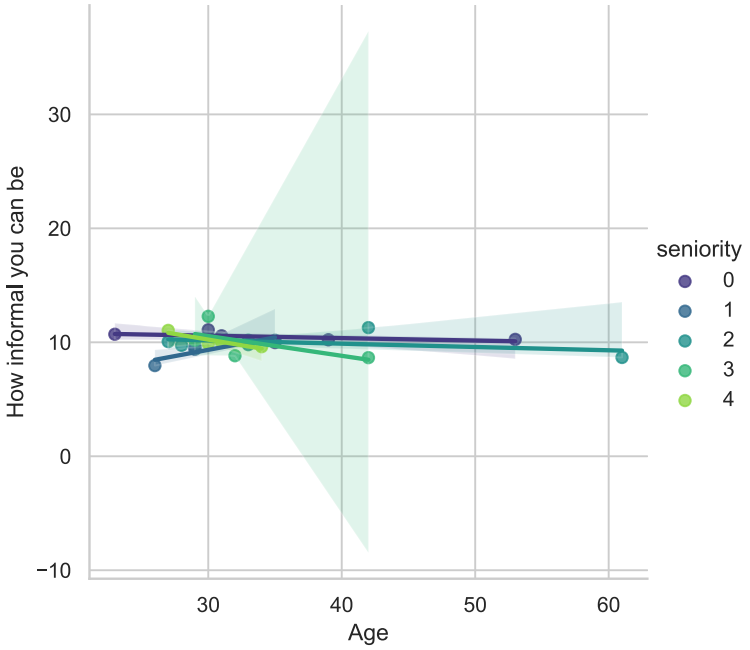


Figure 1.2.: Interaction among Danes: Regressions of informality against age for five levels of seniority.

Part II.

Reports

1. My manuscript

1.1. Abstract

Denmmark … Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sagittis posuere ligula sit amet lacinia. Duis dignissim pellentesque magna, rhoncus congue sapien finibus mollis. Ut eu sem laoreet, vehicula ipsum in, convallis erat. Vestibulum magna sem, blandit pulvinar augue sit amet, auctor malesuada sapien. Nullam faucibus leo eget eros hendrerit, non laoreet ipsum lacinia. Curabitur cursus diam elit, non tempus ante volutpat a. Quisque hendrerit blandit purus non fringilla. Integer sit amet elit viverra ante dapibus semper. Vestibulum viverra rutrum enim, at luctus enim posuere eu. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

1.2. Introduction

Denmark is …. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis sagittis posuere ligula sit amet lacinia. Duis dignissim pellentesque magna, rhoncus congue sapien finibus mollis. Ut eu sem laoreet, vehicula ipsum in, convallis erat. Vestibulum magna sem, blandit pulvinar augue sit amet, auctor malesuada sapien. Nullam faucibus leo eget eros hendrerit, non laoreet ipsum lacinia. Curabitur cursus diam elit, non tempus ante volutpat a. Quisque hendrerit blandit purus non fringilla. Integer sit amet elit viverra ante dapibus semper. Vestibulum viverra rutrum enim, at luctus enim posuere eu. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nunc ac dignissim magna. Vestibulum vitae egestas elit. Proin feugiat leo quis ante condimentum, eu ornare mauris feugiat. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris cursus laoreet

ex, dignissim bibendum est posuere iaculis. Suspendisse et maximus elit. In fringilla gravida ornare. Aenean id lectus pulvinar, sagittis felis nec, rutrum risus. Nam vel neque eu arcu blandit fringilla et in quam. Aliquam luctus est sit amet vestibulum eleifend. Phasellus elementum sagittis molestie. Proin tempor lorem arcu, at condimentum purus volutpat eu. Fusce et pellentesque ligula. Pellentesque id tellus at erat luctus fringilla. Suspendisse potenti. Etiam maximus accumsan gravida. Maecenas at nunc dignissim, euismod enim ac, bibendum ipsum. Maecenas vehicula velit in nisl aliquet ultricies. Nam eget massa interdum, maximus arcu vel, pretium erat. Maecenas sit amet tempor purus, vitae aliquet nunc. Vivamus cursus urna velit, eleifend dictum magna laoreet ut. Duis eu erat mollis, blandit magna id, tincidunt ipsum. Integer massa nibh, commodo eu ex vel, venenatis efficitur ligula. Integer convallis lacus elit, maximus eleifend lacus ornare ac. Vestibulum scelerisque viverra urna id lacinia. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Aenean eget enim at diam bibendum tincidunt eu non purus. Nullam id magna ultrices, sodales metus viverra, tempus turpis.

1.3. Results

1.3.1. Social norms

In Denmark, the workplace interaction is very informal and largely unaffected by seniority and age.

The 24 subjects from workplaces in Denmark were interviewed .... blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah

I found that neither academic seniority or age of workplace individuals much affected how informal our interaction was (see Figure 1.2).

The correlation between informality and age was -0.195 and the correlation between informality and seniority was -0.055.

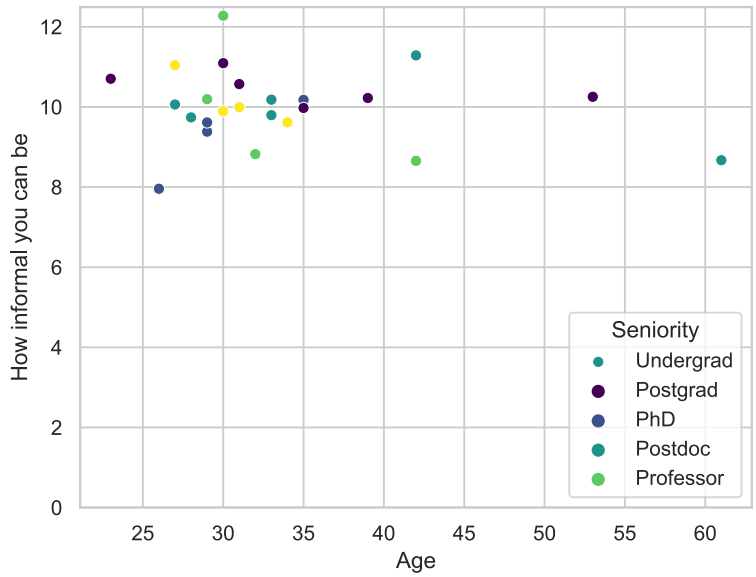


Figure 1.1.: Interaction among Danes: How Danes interact is has very little to do with age and seniority, compared to most other contries.

Duis ornare ex ac iaculis pretium. Maecenas sagittis odio id erat pharetra, sit amet consectetur quam sollicitudin. Vivamus pharetra quam purus, nec sagittis risus pretium at. Nullam feugiat, turpis ac accumsan interdum, sem tellus blandit neque, id vulputate diam quam semper nisl. Donec sit amet enim at neque porttitor aliquet. Phasellus facilisis nulla eget placerat eleifend. Vestibulum non egestas eros, eget lobortis ipsum. Nulla rutrum massa eget enim aliquam, id porttitor erat luctus. Nunc sagittis quis eros eu sagittis. Pellentesque dictum, erat at pellentesque sollicitudin, justo augue pulvinar metus, quis rutrum est mi nec felis. Vestibulum efficitur mi lorem, at elementum purus tincidunt a. Aliquam finibus enim magna, vitae pellentesque erat faucibus at. Nulla mauris tellus, imperdiet id lobortis et, dignissim condimentum ipsum. Morbi nulla orci, varius at aliquet sed, facilisis id tortor. Donec ut urna nisi.

1.4. Discussion

This this investigation of 24 Danes…,

Vestibulum ultrices, tortor at mattis porta, odio nisi rutrum nulla, sit amet tincidunt eros quam facilisis tellus. Fusce eleifend lectus in elementum lacinia. Nam auctor nunc in massa ullamcorper, sit amet auctor ante accumsan. Nam ut varius metus. Curabitur eget tristique leo. Cras finibus euismod erat eget elementum. Integer vel placerat ex. Ut id eros quis lectus lacinia venenatis hendrerit vel ante.

1.5. Methods

1.5.1. Interaction analysis

Duis urna urna, pellentesque eu urna ut, malesuada bibendum dolor. Suspendisse potenti. Vivamus ornare, arcu quis molestie ultrices, magna est accumsan augue, auctor vulputate erat quam quis neque. Nullam scelerisque odio vel ultricies facilisis. Ut porta arcu non magna sagittis lacinia. Cras ornare vulputate lectus a tristique. Pellentesque ac arcu congue, rhoncus mi id, dignissim ligula.

Table 1.1 lists the samples included in the analysis.

Table 1.1.: People included in the analysis.

	name	age	sex	position	nationality
0	Julie	27	F	PhDstudent	DK
1	Thomas	33	M	Postdoc	GB
2	Emilie	23	F	PhDstudent	CH
3	Sofie	31	F	Postdoc	DK
4	Sara	29	F	Postdoc	US
5	Cecilie	34	F	Postdoc	DK
6	Anders	32	M	PhDstudent	UK
7	Emma	42	F	Professor	DK
8	Caroline	31	F	PhDstudent	DK
9	Laura	30	F	Postdoc	DK
10	Mikkel	33	M	Postdoc	NL
11	Jens	27	M	PhDstudent	DK
12	Andreas	29	M	PhDstudent	DK
13	Jakob	28	M	PhDstudent	DK
14	Mathilde	61	F	Professor	DK
15	Katrine	35	F	Postdoc	DK
16	Poul	30	M	Postdoc	DK

Table 1.1.: People included in the analysis.

	name	age	sex	position	nationality
17	Anna	26	F	PhDstudent	DK
18	Peter	42	M	Professor	GB
19	Ida	53	F	Postdoc	DK
20	Freja	30	F	Postdoc	DK
21	Maria	39	F	Professor	UK
22	Amalie	29	F	PhDstudent	DK
23	Camilla	35	F	Postdoc	DK

Duis ornare ex ac iaculis pretium. Maecenas sagittis odio id erat pharetra, sit amet consectetur quam sollicitudin. Vivamus pharetra quam purus, nec sagittis risus pretium at. Nullam feugiat, turpis ac accumsan interdum, sem tellus blandit neque, id vulputate diam quam semper nisl. Donec sit amet enim at neque porttitor aliquet. Phasellus facilisis nulla eget placerat eleifend. Vestibulum non egestas eros, eget lobortis ipsum. Nulla rutrum massa eget enim aliquam, id porttitor erat luctus. Nunc sagittis quis eros eu sagittis. Pellentesque dictum, erat at pellentesque sollicitudin, justo augue pulvinar metus, quis rutrum est mi nec felis. Vestibulum efficitur mi lorem, at elementum purus tincidunt a. Aliquam finibus enim magna, vitae pellentesque erat faucibus at. Nulla mauris tellus, imperdiet id lobortis et, dignissim condimentum ipsum. Morbi nulla orci, varius at aliquet sed, facilisis id tortor. Donec ut urna nisi.

The 24 subjects were asked to score the follow statements:

1. Blah blah blah
2. Blah blah blah
3. Blah blah blah
4. Blah blah blah

1.6. References

Part III.

Tables

1. Result tables

This is a table of results from a csv file:

Table 1.1.: This could be a table listing results of an analysis.

	pos	when_DAF_is_half	when_mutation_has_freq2	population
0	10122953	-1.390610	-6.29509	CDX
1	11859476	-2.693320	-6.64483	CDX
2	11864438	-2.693320	-6.64483	CDX
3	32635171	-2.224110	-6.58804	CDX
4	105249963	-1.406140	-6.11750	CDX
5	3712725	-1.086690	-6.50623	CHB
6	3713920	-2.062530	-9.86443	CHB
7	3717514	-2.635040	-6.90527	CHB
8	3720564	-2.399250	-6.07486	CHB
9	3720591	-2.399250	-6.07486	CHB
10	3721203	-2.399250	-6.07486	CHB
11	3721452	-2.399250	-6.07486	CHB
12	32812742	-3.157760	-6.19174	CHB
13	47349640	-1.544110	-6.06731	CHB
14	47352820	-2.056060	-8.20242	CHB
15	151669901	-1.644880	-8.53576	CHB
16	3706646	-2.436120	-6.34196	CHS
17	10214581	-2.539500	-9.20117	CHS
18	122431333	-0.937105	-6.22082	CHS
19	141669308	-1.019170	-6.12830	CHS
20	49962327	-1.766100	-6.61933	JPT
21	142111497	-1.409290	-6.26781	JPT
22	20348765	-2.042520	-6.05583	KHV
23	20349644	-1.773100	-8.27088	KHV
24	69126919	-1.338150	-6.76224	KHV
25	151669901	-2.155580	-7.15834	KHV
26	153982797	-1.289470	-6.11440	KHV

Show more tables (these are the same again):

```
import pandas as pd
pd.read_csv('../results/result_table.csv')
```

	pos	when_DAF_is_half	when_mutation_has_freq2	popu
0	10122953	-1.390610	-6.29509	CDX
1	11859476	-2.693320	-6.64483	CDX
2	11864438	-2.693320	-6.64483	CDX
3	32635171	-2.224110	-6.58804	CDX
4	105249963	-1.406140	-6.11750	CDX
5	3712725	-1.086690	-6.50623	CHB
6	3713920	-2.062530	-9.86443	CHB
7	3717514	-2.635040	-6.90527	CHB
8	3720564	-2.399250	-6.07486	CHB
9	3720591	-2.399250	-6.07486	CHB
10	3721203	-2.399250	-6.07486	CHB
11	3721452	-2.399250	-6.07486	CHB
12	32812742	-3.157760	-6.19174	CHB
13	47349640	-1.544110	-6.06731	CHB
14	47352820	-2.056060	-8.20242	CHB
15	151669901	-1.644880	-8.53576	CHB
16	3706646	-2.436120	-6.34196	CHS
17	10214581	-2.539500	-9.20117	CHS
18	122431333	-0.937105	-6.22082	CHS
19	141669308	-1.019170	-6.12830	CHS
20	49962327	-1.766100	-6.61933	JPT
21	142111497	-1.409290	-6.26781	JPT
22	20348765	-2.042520	-6.05583	KHV
23	20349644	-1.773100	-8.27088	KHV
24	69126919	-1.338150	-6.76224	KHV
25	151669901	-2.155580	-7.15834	KHV
26	153982797	-1.289470	-6.11440	KHV

```
import pandas as pd
pd.read_csv('../results/result_table.csv')
```

	pos	when_DAF_is_half	when_mutation_has_freq2	popu
0	10122953	-1.390610	-6.29509	CDX
1	11859476	-2.693320	-6.64483	CDX
2	11864438	-2.693320	-6.64483	CDX
3	32635171	-2.224110	-6.58804	CDX

	pos	when_DAF_is_half	when_mutation_has_freq2	population
4	105249963	-1.406140	-6.11750	CDX
5	3712725	-1.086690	-6.50623	CHB
6	3713920	-2.062530	-9.86443	CHB
7	3717514	-2.635040	-6.90527	CHB
8	3720564	-2.399250	-6.07486	CHB
9	3720591	-2.399250	-6.07486	CHB
10	3721203	-2.399250	-6.07486	CHB
11	3721452	-2.399250	-6.07486	CHB
12	32812742	-3.157760	-6.19174	CHB
13	47349640	-1.544110	-6.06731	CHB
14	47352820	-2.056060	-8.20242	CHB
15	151669901	-1.644880	-8.53576	CHB
16	3706646	-2.436120	-6.34196	CHS
17	10214581	-2.539500	-9.20117	CHS
18	122431333	-0.937105	-6.22082	CHS
19	141669308	-1.019170	-6.12830	CHS
20	49962327	-1.766100	-6.61933	JPT
21	142111497	-1.409290	-6.26781	JPT
22	20348765	-2.042520	-6.05583	KHV
23	20349644	-1.773100	-8.27088	KHV
24	69126919	-1.338150	-6.76224	KHV
25	151669901	-2.155580	-7.15834	KHV
26	153982797	-1.289470	-6.11440	KHV

Part IV.

Slides

1. Main talk

1.1. Admixture displacement in each geographical region

1.1.1. This is a subtitle

Here we have some text that may run over several lines of the slide frame, depending on how long it is.

- first item
  - A sub item

Next, we'll brief review some theme-specific components.

- Note that *all* of the standard Reveal.js features can be used with this theme, even if we don't highlight them here.

1.2. Additional theme classes

1.2.1. Some extra things you can do with the clean theme

Special classes for emphasis

- `.alert` class for default emphasis, e.g. important note.
- `.fg` class for custom colour, e.g. important note.
- `.bg` class for custom background, e.g. important note.

Cross-references

- `.button` class provides a Beamer-like button, e.g. Summary

1.3. Social norms

1.3.1. Sampling

We used a sample size of 24.

In Denmark, the workplace interaction is very informal and largely unaffected by seniority and age.

The 24 subjects from workplaces in Denmark were interviewed .... blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah blah

1.4. Social norms

1.4.1. Neither academic seniority or age affected interaction

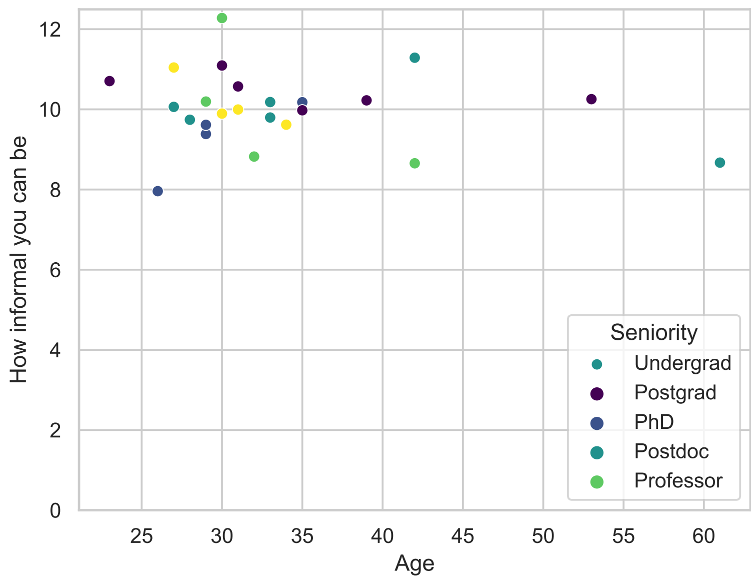


Figure 1.1.: Interaction among Danes: How Danes interact is has very little to do with age and seniority, compared to most other contries.

The correlation between informality and age was -0.195 and the correlation between informality and seniority was -0.055.

1.5. Slide title

- Eat spaghetti
- Drink wine

1.6. Slide title

Left column

- One
- Two
- Three

1.7. Admixture displacement in each geographical region

The correlation between informality and age was -0.195 and the correlation between informality and seniority was -0.055.

1.8. Slide Title

Slide content

Schumer et al. (2018)

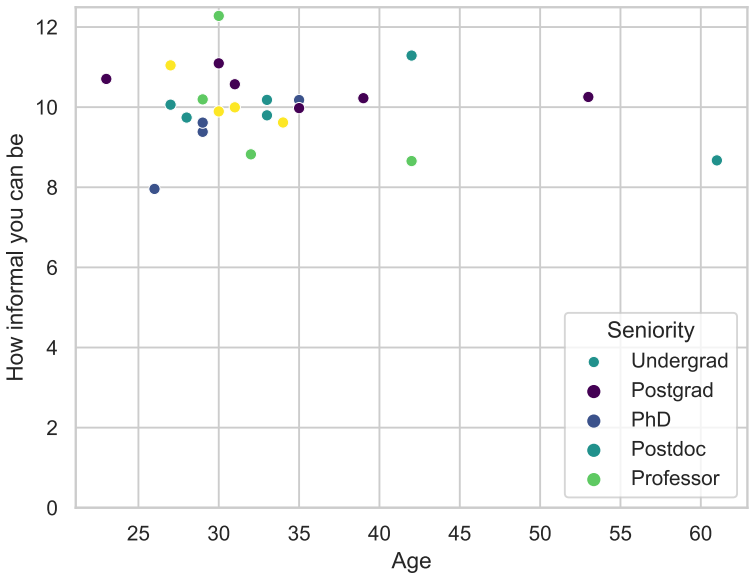


Figure 1.2.: Interaction among Danes: How Danes interact is has very little to do with age and seniority, compared to most other contries.