bayes_models_diagnostics

March 21, 2023

1 Overview

This is a python notebook to present all the model diagnostics of bayseian models used in Exploring Biopsychosocial Profiles in Individuals with Anorexia Nervosa and Healthy Controls paper.

For more information on how the data was pre-processed and how the models where built please see https://github.com/WMDA/BB_data/

1.1 Content

Section 1: Overview of notebook

Section 2: Libraries used in this note book and reading in the data/models .

Section 3: A Heatmap Plot

Section 4: KDE and trace plots for the mixed effects models

Section 5: Diagnostic Summaries of the mixed effects models

Section 6: Autocorrelation plots

2 Libraries used in this note book and reading in the data/models

Skip this section if just interested in model diagnostics

[1]:
```
from functions.data_functions import save_pickle, load_pickle, load_data
import seaborn as sns
sns.set_style('darkgrid')
import matplotlib.pyplot as plt
import numpy as np
import arviz as az
```

[2]:
```
fitted_models = load_pickle('fitted_models')
null_models = load_pickle('fitted_models_additional_null_models')
```

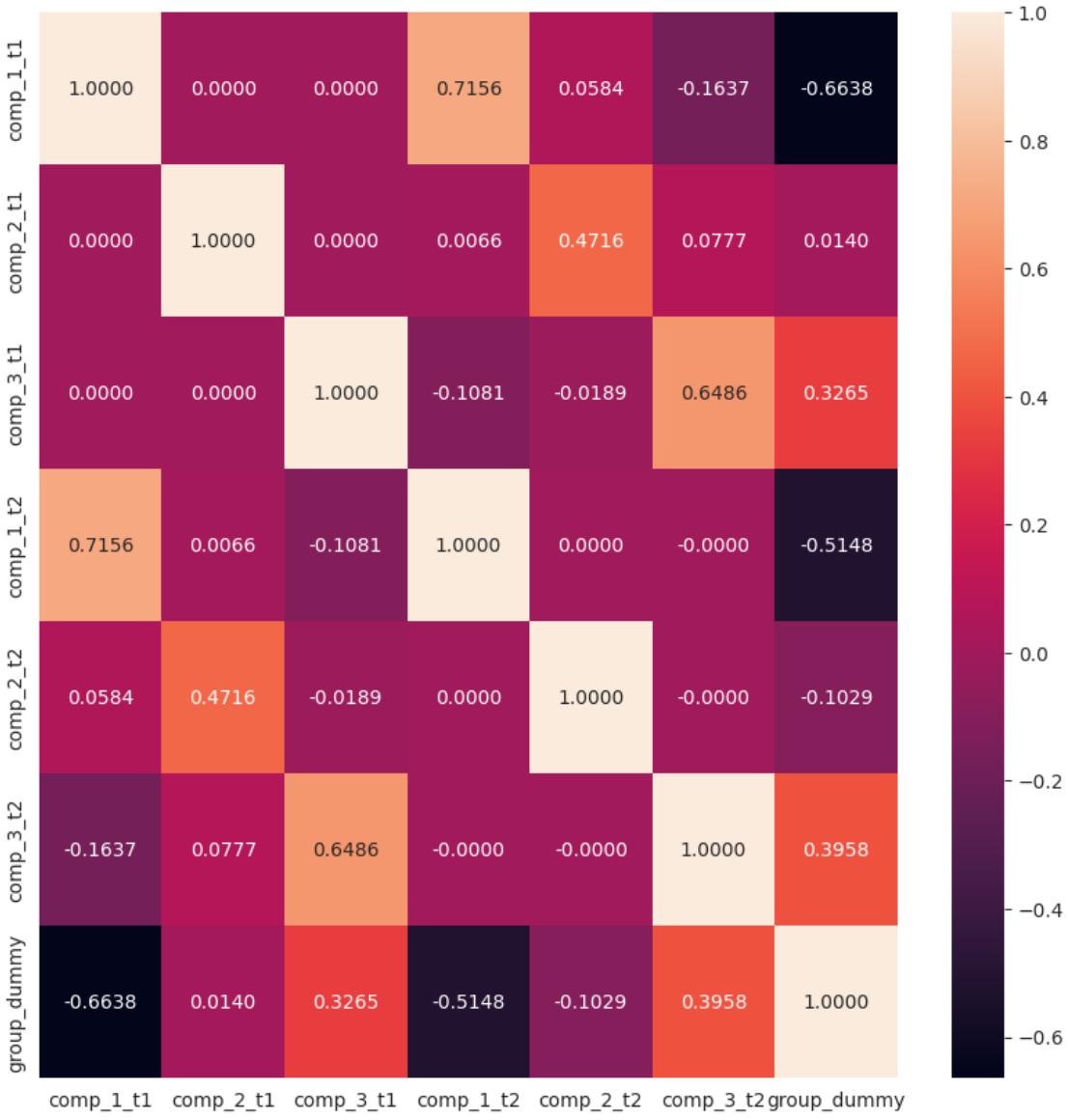
[3]:
```
pca_df = load_data('BEACON', 'pca_df')
#Define number of components to loop over
comp = ['comp_1', 'comp_2','comp_3']
```

3 Heatmap

This is a heatmap of all the variables in the used in the models to check for multicollinearity. Plotted is each of components at time points one and two, along with the R2.

[4]:
```
pca_df['group_dummy'] = pca_df['group'].apply(lambda group: 0 if group =='AN'␣
 ↪else 1)
plt.figure(figsize=(10, 10))
sns.heatmap(pca_df[['comp_1_t1','comp_2_t1', 'comp_3_t1', 'comp_1_t2',␣
 ↪'comp_2_t2', 'comp_3_t2','group_dummy']].corr(),  annot=True,  fmt=".4f")
```

[4]: <AxesSubplot: >



4 KDE and trace plots for the mixed effects models

4.1 Alternative hypothesis models

Below are the kernel denisty plots for the alternative hypothesis model (that there will be an effect of group at time points) as well as trace plots for each parameter in the model.

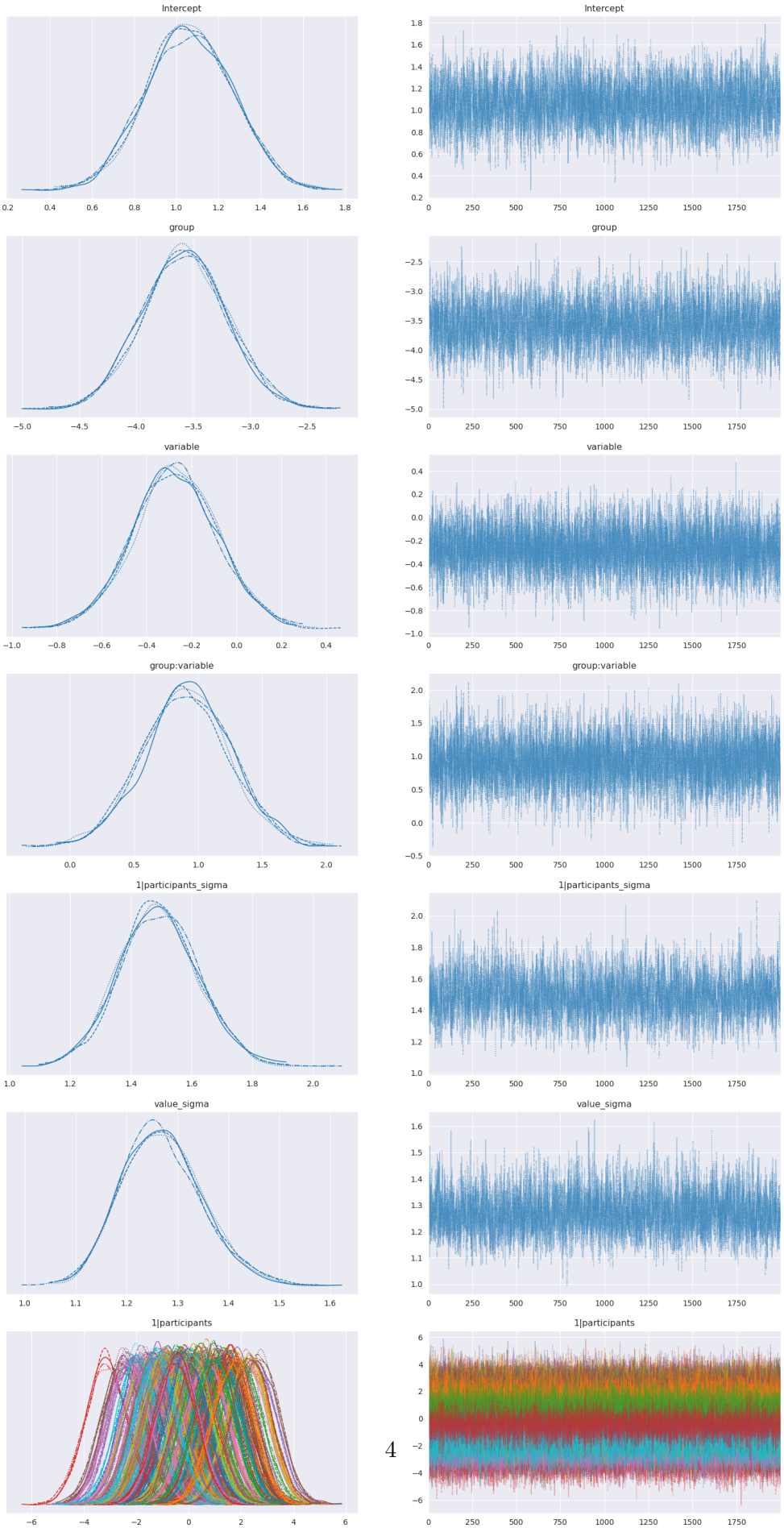
Note value is component scores at both time point and variable is time

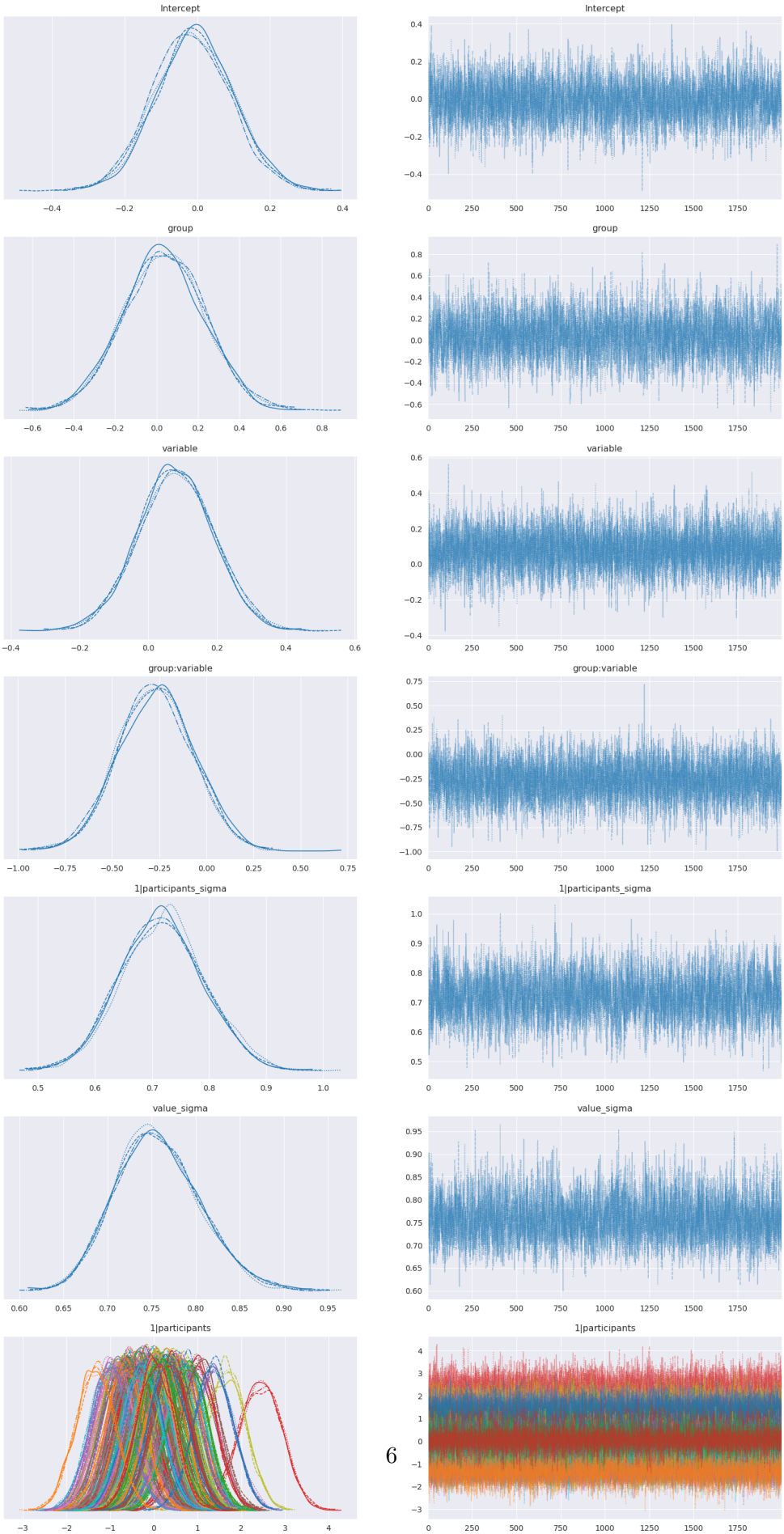
First plot: Component One

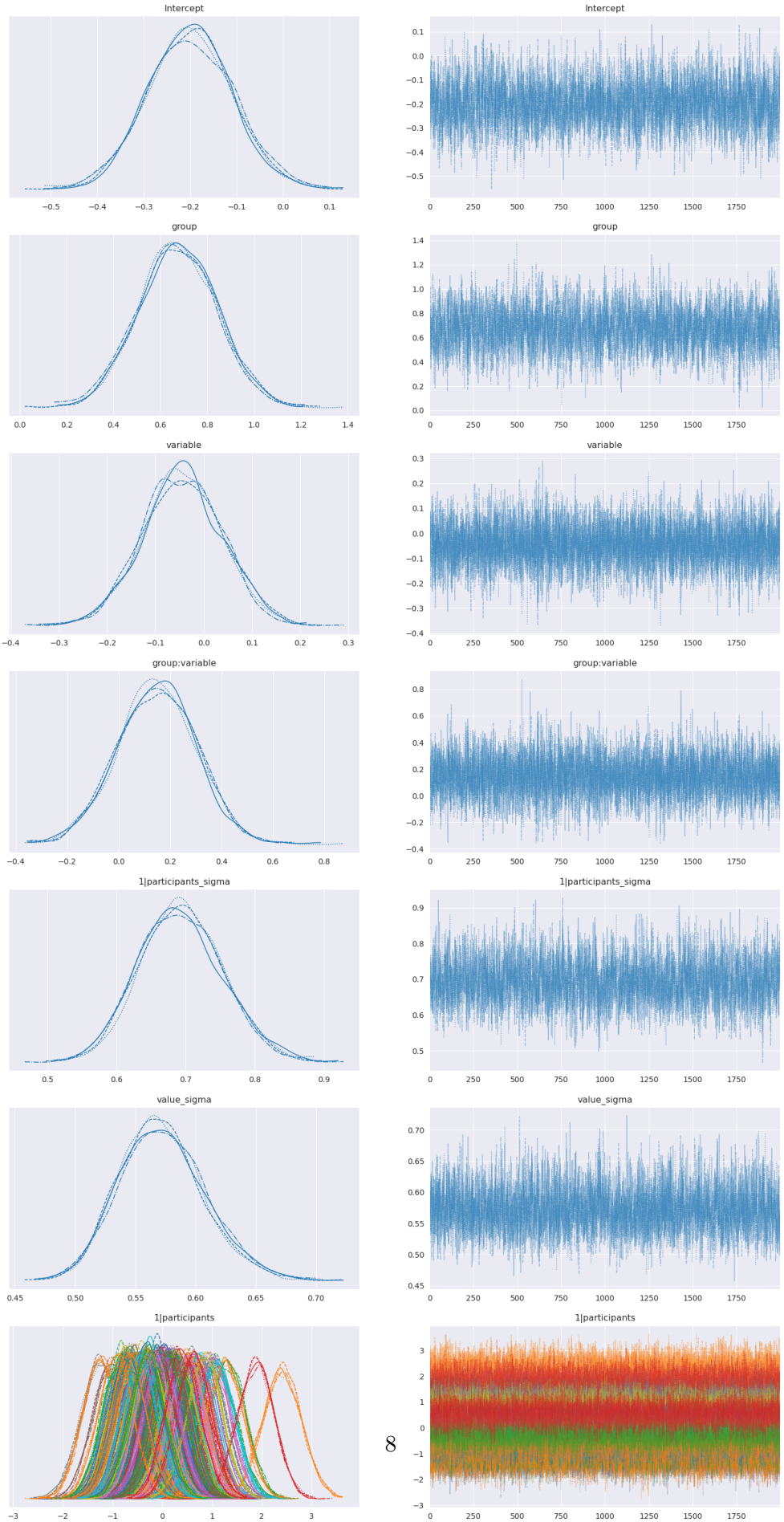
Second plot: Component Two

Third plot: Component Three

[5]:
```
for component in comp:
    az.plot_trace(fitted_models['alt'][component], figsize=(18,35))
```


Intercept
Intercept
group
group
variable
variable
group:variable
group:variable
1|participants_sigma
1|participants_sigma
value_sigma
value_sigma
1|participants
1|participants


Intercept
Intercept
group
group
variable
variable
group:variable
group:variable
1|participants_sigma
1|participants_sigma
value_sigma
value_sigma
1|participants
1|participants


Intercept
Intercept
group
group
variable
variable
group:variable
group:variable
1|participants_sigma
1|participants_sigma
value_sigma
value_sigma
1|participants
1|participants

4.2 For the null hypothesis models

Below are the kernel denisty plots for the null hypothesis model (that there will be no group difference in component scores) as well as trace plots for each parameter in the model.

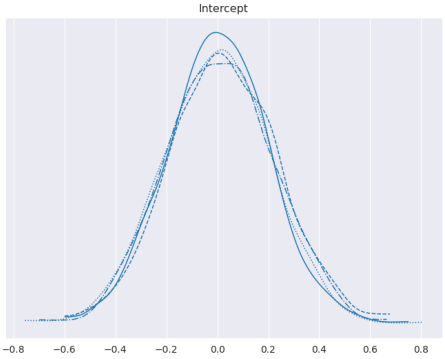
Note value is component scores at both time point and variable is time

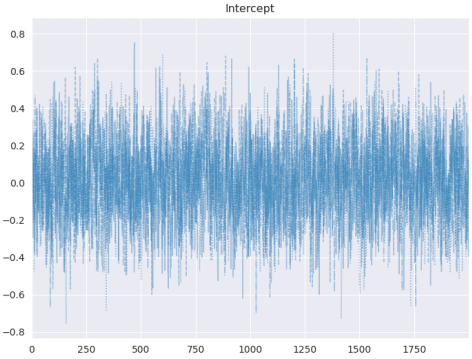
First plot: Component One

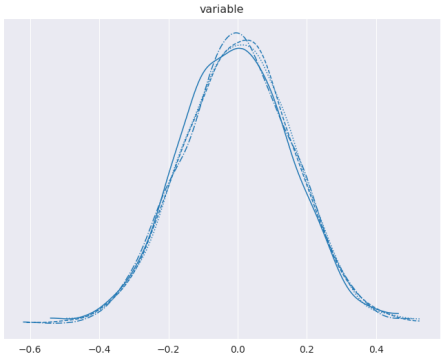
Second plot: Component Two

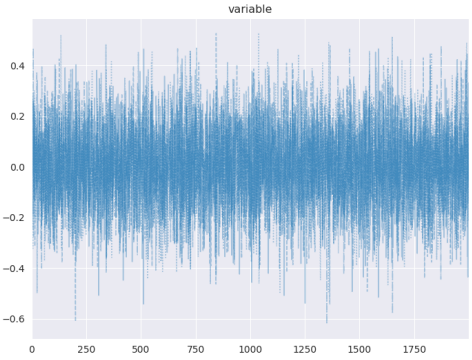
Third plot: Component Three

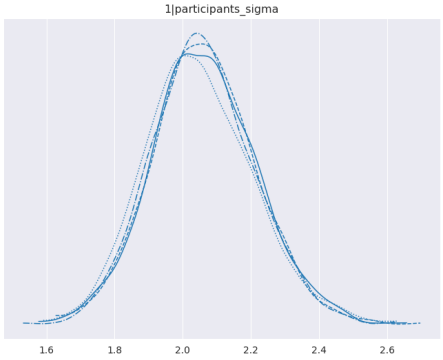
[6]:
```
for component in comp:
    az.plot_trace(null_models[component], figsize=(18,35))
```

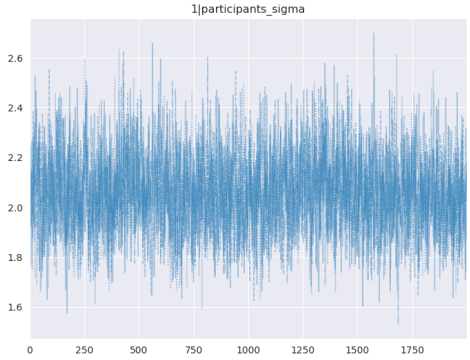

Intercept

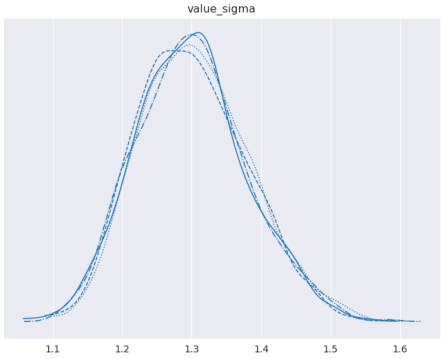

Intercept

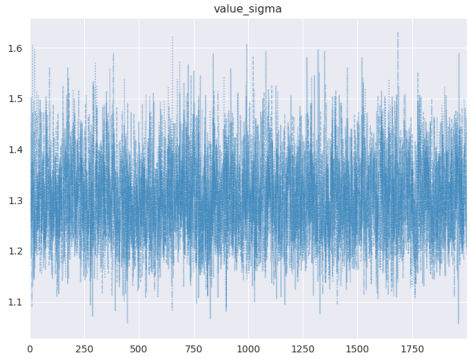

variable

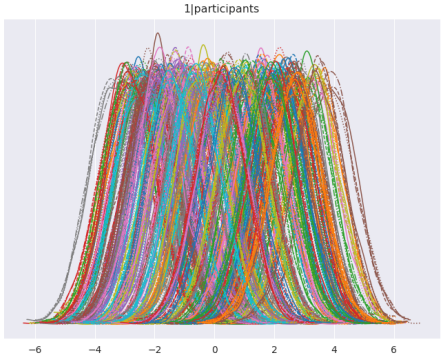

variable

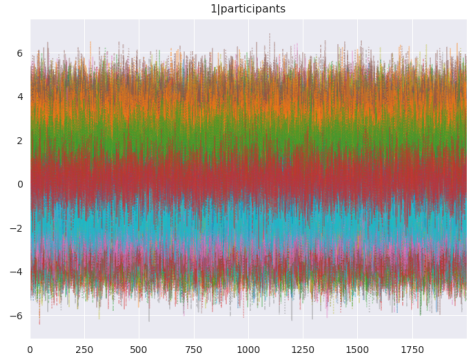

1|participants_sigma

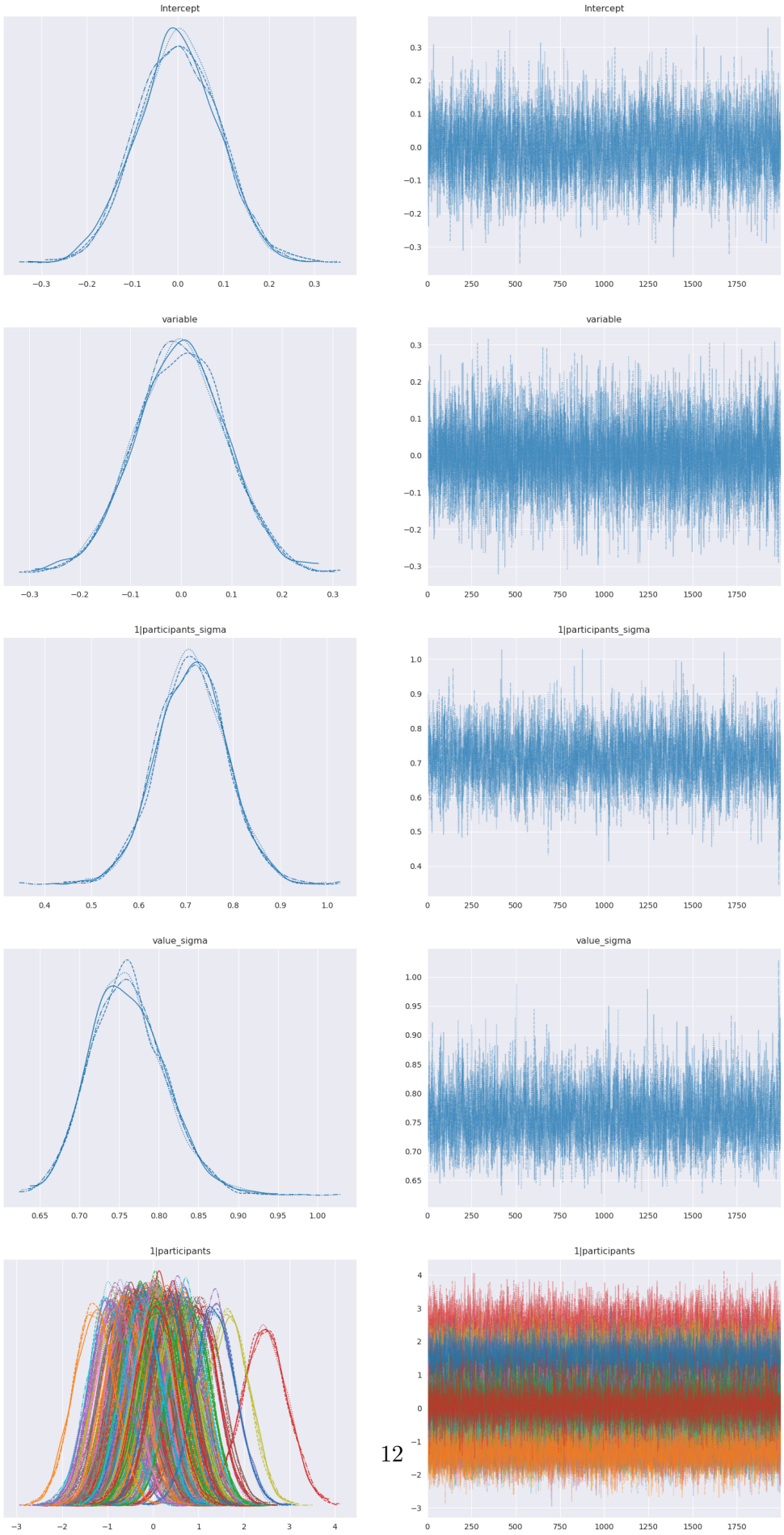

1|participants_sigma

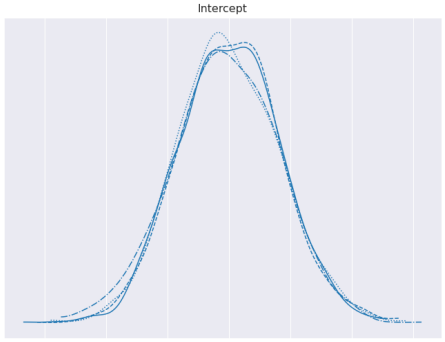

value_sigma

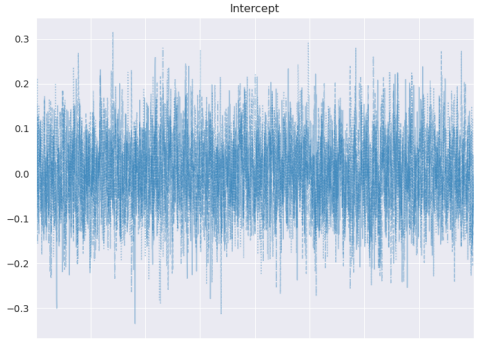

value_sigma

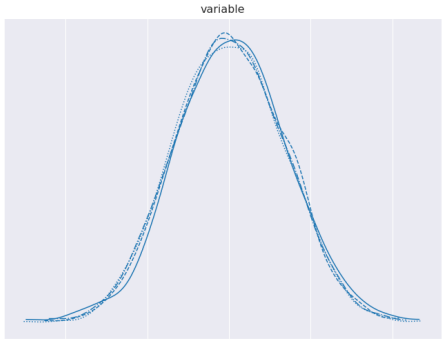

1|participants

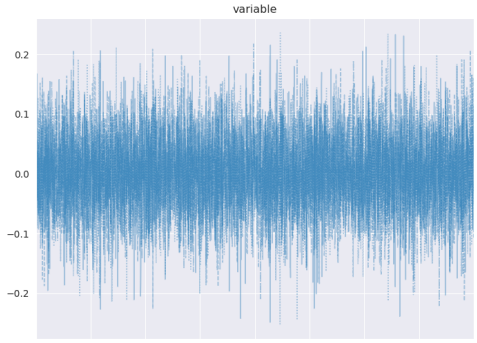

1|participants

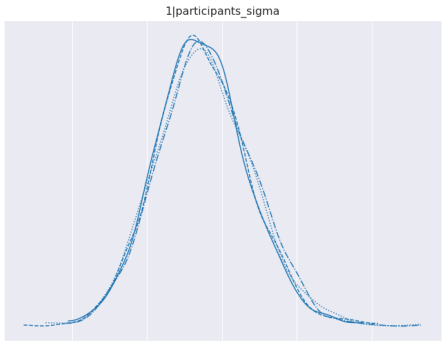

Intercept
Intercept
variable
variable
1|participants_sigma
1|participants_sigma
value_sigma
value_sigma
1|participants
1|participants

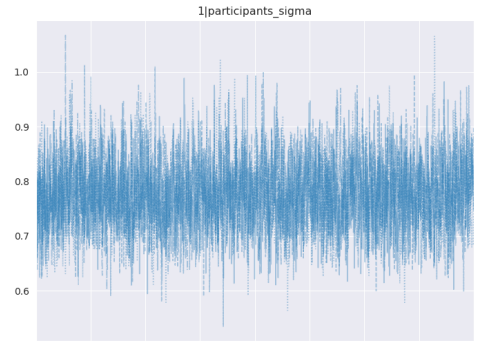

Intercept
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3

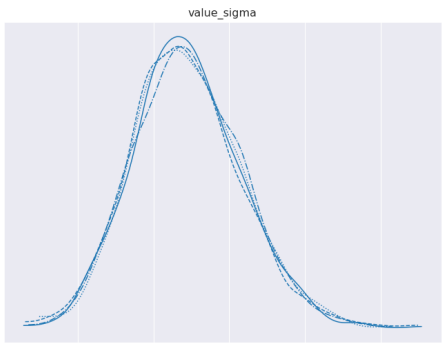

Intercept
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
250
500
750
1000
1250
1500
1750

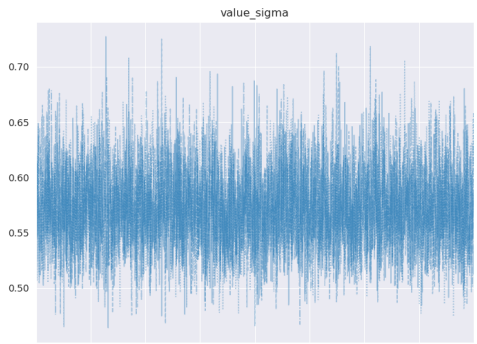

variable
−0.2
−0.1
0.0
0.1
0.2

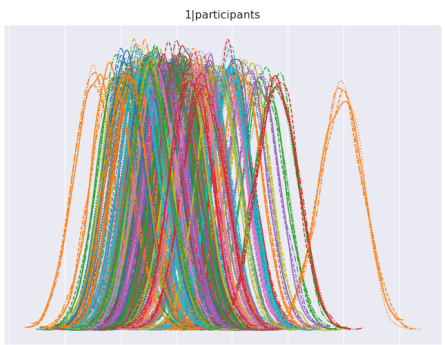

variable
0.2
0.1
0.0
−0.1
−0.2
0
250
500
750
1000
1250
1500
1750

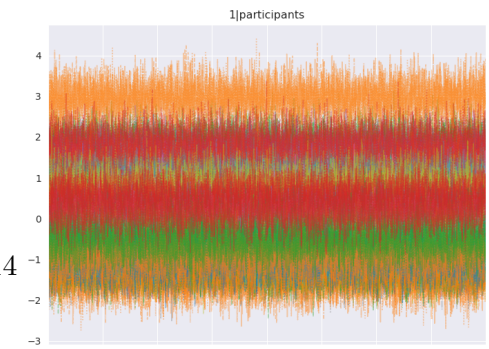

1|participants_sigma
0.6
0.7
0.8
0.9
1.0


1|participants_sigma
1.0
0.9
0.8
0.7
0.6
0
250
500
750
1000
1250
1500
1750


value_sigma
0.50
0.55
0.60
0.65
0.70


value_sigma
0.70
0.65
0.60
0.55
0.50
0
250
500
750
1000
1250
1500
1750


1|participants
−3
−2
−1
0
1
2
3
4


1|participants
4
3
2
1
0
−1
−2
−3
0
250
500
750
1000
1250
1500
1750

5 Diagnostic Summaries of the mixed effects models

5.1 Alternative Hypothesis

Model diagnostics for the MCMC chains for the alternative hypothesis (that there will be a group effect at time points).

Abbreviations

mcse_mean: Monte Carlo standard error for the mean

mcse_sd: Monte Carlo standard error for the standard deviations

ess_bulk: Bulk effective sample size estimate

ess_tail: Tail effective sample size estimate

[7]:
```
alt_variables = ['Intercept', 'group', 'group:variable', 'value_sigma',␣
↪'1|participants_sigma', 'variable']
null_variables = ['value_sigma', '1|participants_sigma', 'variable']
```

[8]:
```
az.summary(fitted_models['alt']['comp_1'], kind='diagnostics',␣
↪var_names=alt_variables )
```

[8]:

	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Intercept	0.004	0.003	3194.0	4645.0	1.0
group[HC]	0.006	0.005	3311.0	5059.0	1.0
group:variable[HC, t2]	0.004	0.003	6509.0	5701.0	1.0
value_sigma	0.002	0.001	2697.0	4278.0	1.0
1|participants_sigma	0.003	0.002	2101.0	3993.0	1.0
variable[t2]	0.002	0.002	7103.0	5956.0	1.0

[9]:
```
az.summary(fitted_models['alt']['comp_2'], kind='diagnostics',␣
↪var_names=alt_variables)
```

[9]:

	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Intercept	0.002	0.001	2852.0	4230.0	1.0
group[HC]	0.004	0.003	2847.0	4499.0	1.0
group:variable[HC, t2]	0.003	0.002	5177.0	6231.0	1.0
value_sigma	0.001	0.001	2708.0	4788.0	1.0
1|participants_sigma	0.002	0.001	2341.0	4528.0	1.0
variable[t2]	0.001	0.001	5868.0	5791.0	1.0

[10]:
```
az.summary(fitted_models['alt']['comp_3'], kind='diagnostics',␣
↪var_names=alt_variables)
```

[10]:

	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Intercept	0.002	0.001	2229.0	3852.0	1.0
group[HC]	0.003	0.002	2557.0	3974.0	1.0
group:variable[HC, t2]	0.002	0.002	5459.0	5436.0	1.0
value_sigma	0.001	0.000	2866.0	5044.0	1.0
1|participants_sigma	0.001	0.001	2333.0	4097.0	1.0
variable[t2]	0.001	0.001	6040.0	5976.0	1.0

Extracting the number of chain divergences, tree depth, energy steps and acceptance rate.

[11]:
```
for component in comp:
    print(f'Number of divergences for {component} alt model:', int(np.
↪sum(fitted_models['alt'][component].sample_stats.diverging)))
    print(f'Number of times tree depth hit 10 for {component} alt model:',␣
↪int(np.sum(fitted_models['alt'][component].sample_stats.tree_depth == 10)))
    print(f'Number of energy steps > 0.3 for {component} alt model:', int(np.
↪sum(fitted_models['alt'][component].sample_stats.energy < 0.3)))
    print(f'Number of steps where acceptance rate < 0.95 for {component} alt␣
↪model:', int(np.sum(fitted_models['alt'][component].sample_stats.
↪acceptance_rate < 0.95)))
    print('\n\n')
```

```
Number of divergences for comp_1 alt model: 0
Number of times tree depth hit 10 for comp_1 alt model: 0
Number of energy steps > 0.3 for comp_1 alt model: 0
Number of steps where acceptance rate < 0.95 for comp_1 alt model: 3074



Number of divergences for comp_2 alt model: 0
Number of times tree depth hit 10 for comp_2 alt model: 0
Number of energy steps > 0.3 for comp_2 alt model: 0
Number of steps where acceptance rate < 0.95 for comp_2 alt model: 3213



Number of divergences for comp_3 alt model: 0
Number of times tree depth hit 10 for comp_3 alt model: 0
Number of energy steps > 0.3 for comp_3 alt model: 0
Number of steps where acceptance rate < 0.95 for comp_3 alt model: 3186
```

5.2 Null Hypothesis

Model diagnostics for the MCMC chains for the Null hypothesis (that there will be a group difference in component scores).

Abbreviations

mcse_mean: Monte Carlo standard error for the mean

mcse_sd: Monte Carlo standard error for the standard deviations

ess_bulk: Bulk effective sample size estimate

ess_tail: Tail effective sample size estimate

[12]:
```
az.summary(null_models['comp_1'], kind='diagnostics', var_names=null_variables)
```

[12]:

	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
value_sigma	0.002	0.001	2818.0	4951.0	1.0
1|participants_sigma	0.004	0.003	1646.0	3571.0	1.0
variable[t2]	0.002	0.002	7698.0	5886.0	1.0

[13]:
```
az.summary(null_models['comp_2'], kind='diagnostics', var_names=null_variables)
```

[13]:

	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
value_sigma	0.001	0.001	2977.0	4179.0	1.0
1|participants_sigma	0.002	0.001	2307.0	3788.0	1.0
variable[t2]	0.001	0.001	11087.0	6391.0	1.0

[14]:
```
az.summary(null_models['comp_3'], kind='diagnostics', var_names=null_variables)
```

[14]:

	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
value_sigma	0.001	0.000	3271.0	4974.0	1.0
1|participants_sigma	0.001	0.001	2269.0	4210.0	1.0
variable[t2]	0.001	0.001	10590.0	5855.0	1.0

Extracting the number of divergences, tree depth, energy steps and acceptance rate.

[15]:
```
for component in comp:
    print(f'Number of divergences for {component} alt model:', int(np.
↪sum(null_models[component].sample_stats.diverging)))
    print(f'Number of times tree depth hit 10 for {component} alt model:',␣
↪int(np.sum(null_models[component].sample_stats.tree_depth == 10)))
    print(f'Number of energy steps > 0.3 for {component} alt model:', int(np.
↪sum(null_models[component].sample_stats.energy < 0.3)))
    print(f'Number of steps where acceptance rate < 0.95 for {component} alt␣
↪model:', int(np.sum(null_models[component].sample_stats.acceptance_rate < 0.
↪95)))
    print('\n\n')
```

```
Number of divergences for comp_1 alt model: 0
Number of times tree depth hit 10 for comp_1 alt model: 0
Number of energy steps > 0.3 for comp_1 alt model: 0
Number of steps where acceptance rate < 0.95 for comp_1 alt model: 3270
```

```
Number of divergences for comp_2 alt model: 0
Number of times tree depth hit 10 for comp_2 alt model: 0
Number of energy steps > 0.3 for comp_2 alt model: 0
Number of steps where acceptance rate < 0.95 for comp_2 alt model: 3082
```

```
Number of divergences for comp_3 alt model: 0
Number of times tree depth hit 10 for comp_3 alt model: 0
Number of energy steps > 0.3 for comp_3 alt model: 0
Number of steps where acceptance rate < 0.95 for comp_3 alt model: 3298
```

6 Autocorrelation plots

6.1 Alternative Hypothesis models

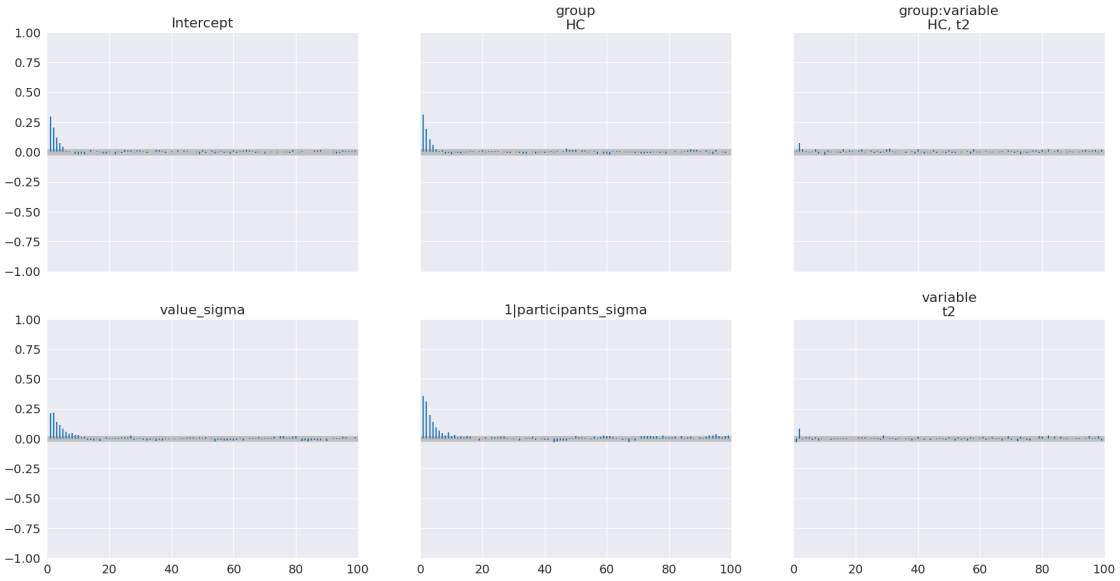
Autocorrelation plots for the alterantive hypothesis models

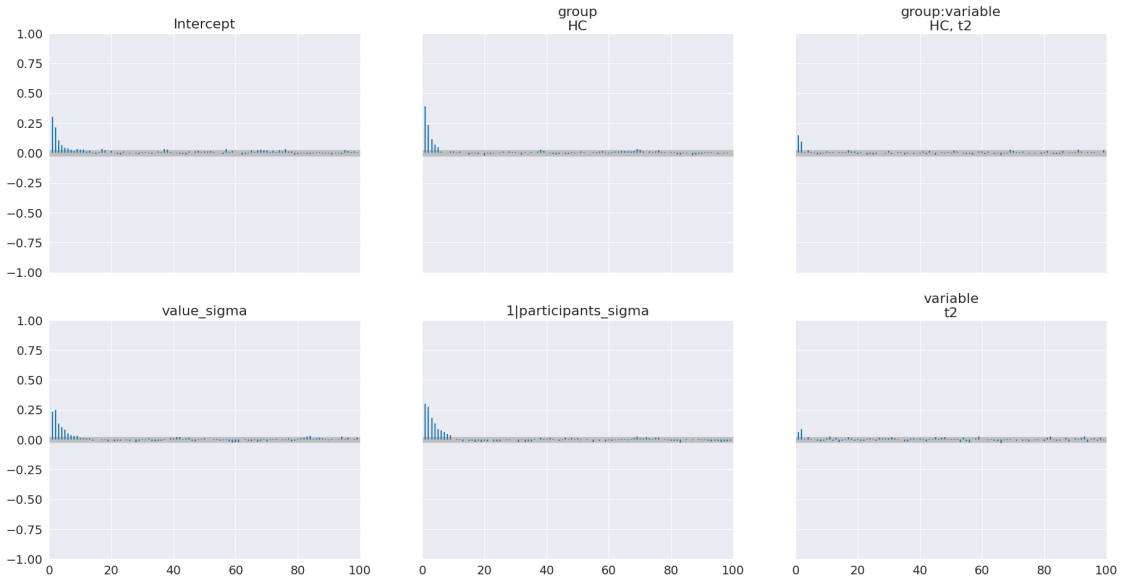
Plot one: Component one

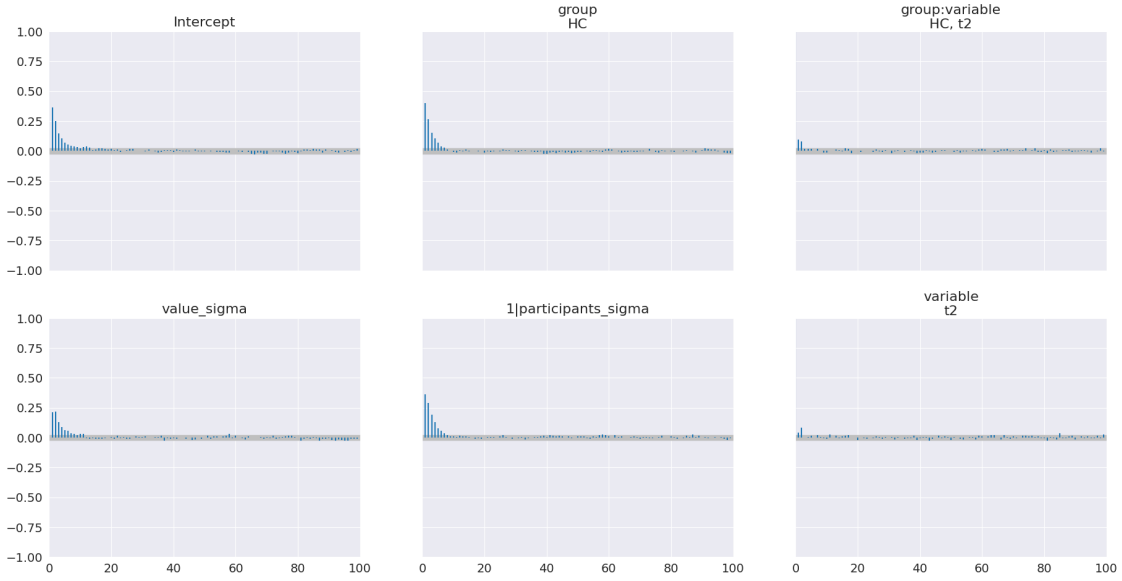
Plot two: Component two

Plot three: Component three

[16]:
```
for component in comp:
    az.plot_autocorr(fitted_models['alt'][component] ,combined=True,␣
 ↪var_names=alt_variables)
```







6.2 Null Hypothesis models

Autocorrelation plots for the null hypothesis models

Plot one: Component one

Plot two: Component two

Plot three: Component three

[17]:
```
for component in comp:
    az.plot_autocorr(null_models[component] ,combined=True,␣
 ↪var_names=null_variables)
```