Fall 2019 ISYE 6420
Final Project:
Tempur-Pedic® Effect on Sleep

Travis Jefferies

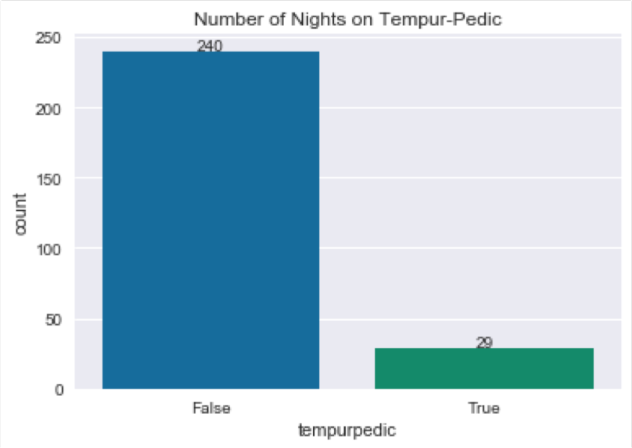
November 28, 2019

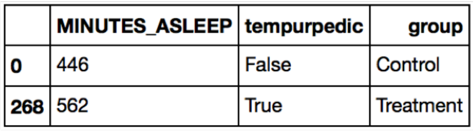
1 Problem Description

The problem I am interested in applying a Bayesian methodology to relates to sleep and restfulness. I recently invested in a Tempur-Pedic® mattress and want to test the hypothesis that I am consistently sleeping more minutes each night. The mattress was an expensive purchase and I'd like to validate that I am 'getting my money's worth'. From an outcome perspective, I am interested in both the difference in expected number of minutes asleep and the difference in standard deviations. The difference in standard deviations is a measure of consistency with regards to how well I am sleeping on the new mattress.

2 Data Collection

I used a Fitbit® Flex 2 to record how many minutes I sleep each night. Fitbit® Flex 2 is a fitness wristband with a removable tracker that tracks activity, exercise, and sleep measures such as number of steps, calories burned, and minutes asleep. Flex 2 records the data in .json format available for export from the Fitbit® website. Data capture and ETL from .json format to the comma separated DATA.csv file is outside the scope of this project. I have sleep data on record for 240 nights prior to the purchase of my Tempur-Pedic® mattress. It should be noted that I used the same traditional spring mattress for a majority of the 240 nights, minus a handful of nights where I was on vacation, in a hotel, etc. To the best of my knowledge, I never slept on a Tempur-Pedic® during this time. I recorded about a months worth (29 nights) of sleep data post Tempur-Pedic® purchase:





	MINUTES_ASLEEP	tempurpedic	group
0	446	False	Control
268	562	True	Treatment

Figure 1: Number of Samples Countplot and Data Preview

I'll denote tempurpedic = False as the 'control' group and tempurpedic = True as the 'treatment' group. These two sample groups will be used as the basis of comparison to measure the 'effect' of the Tempur-Pedic® mattress on how many minutes I sleep each night:

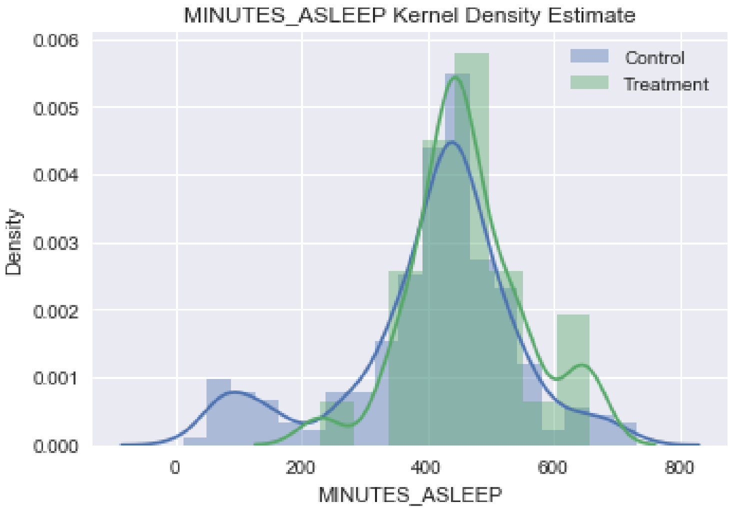


Figure 2: Treatment and Control Group Kernel Density Estimate (KDE) Plots

While not obviously visibly separable, there does appear to be a bit of a difference with regards to means and standard deviations between the treatment and control groups after one month of treatment exposure. There is some evidence of bimodality present in the data set - the smaller peak in the control group probably corresponds to 'naps'. For this project, I am interested not only in the difference of means, but the difference in standard deviations, so I decided to leave these data points in the analysis.

3 Methodology

I first performed a literature search related to Bayesian methodologies for comparing two samples of data drawn from the same population. The classical way to test for differences between two samples is the t test. Kruschke[1] lists the following limitations (among others) of the classical t test:

1. Only handles difference of means, not difference of standard deviations
2. Limited to rejecting null value and can't accept the null value, even when certainty in the estimate is high
3. Normality assumption of two samples is not very robust to outliers

Kruschke goes on to propose a Bayesian alternative to the t test. I decided to use the Bayesian methodology proposed by Kruschke to measure the difference of means and the difference in standard deviations. From the KDE in Figure 2, we can clearly see that there are some outliers present in our data set, something the classical t test is not capable of handling. Kruschke proposes using the heavy-tailed t distribution instead of the normal distribution to more robustly model data gathered under 'real-world' conditions. See page below for the entire formulation of Kruschke's method applied to my sleep data:

$$
\begin{aligned}
\mathsf{MINUTES\_ASLEEP}_{\text{group}} &\sim t\left(\mu_{\text{prior}}, \sigma_{\text{prior}}, \nu_{\text{prior}}\right)\\
\mu_{\text{prior}} &\sim \mathcal{N}\left(\mu_{\text{hyperprior}}, \sigma_{\text{hyperprior}}\right)\\
\sigma_{\text{prior}} &\sim (1000 \cdot \sigma_{\text{pooled samples}}) \cdot \mathcal{U} - \max((1/1000 \cdot \sigma_{\text{pooled samples}}) \cdot \mathcal{U}, 1)\\
\nu_{\text{prior}} &\sim \text{Exp}(\lambda = 1/29) + 1\\
\mu_{\text{hyperprior}} &= \mu_{\text{pooled samples}}\\
\sigma_{\text{hyperprior}} &= 1000 \cdot \sigma_{\text{pooled samples}}
\end{aligned}
$$

where

$$
\begin{aligned}
\text{group} &\in \{\text{control, treatment}\}\\
\mu_{\text{pooled samples}} &= \frac{\sum_{i=1}^{n} x_i}{n}\\
\sigma_{\text{pooled samples}} &= \sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \mu_{\text{pooled samples}}\right)}{n-1}}
\end{aligned}
$$

where n is all 269 observations present in the two pooled samples. Notice how our μ_{prior} and σ_{prior} are diffuse and non-informative, insuring that the prior has very little effect on the posterior. ν_{prior} was chosen based on Kruschke's recommendation of a shifted exponential distribution that balances nearly normal distributions ($\nu < 30$) with heavy tailed distributions ($\nu > 30$) [1]. $\mu_{\text{hyperprior}}$ is chosen to be $\mu_{\text{pooled samples}}$ so the output posteriors are on the same scale as our input data. Choosing $\sigma_{\text{hyperprior}}$ to be $1000 \cdot \sigma_{\text{pooled samples}}$ insures that our sampling algorithm explores a very large search space. A similar design is incorporated in the bounded uniform σ_{prior}. I bounded the σ_{prior} term to have a floor of one minute (having variation in sleeping patterns measured at the < 60 seconds level isn't realistic). Once we generate the $\mathsf{MINUTES\_ASLEEP}_{\text{group}}$ student t posteriors (denoted by a *), we can perform the hypothesis tests of interests:

$$
\begin{aligned}
\mu^*_{\text{treatment}} - \mu^*_{\text{control}} &> 0\\
\sigma^*_{\text{treatment}} - \sigma^*_{\text{control}} &< 0\\
\frac{(\mu^*_{\text{treatment}} - \mu^*_{\text{control}})}{(\sqrt{(\sigma^{*2}_{\text{treatment}} - \sigma^{*2}_{\text{control}})/2})} &> 0
\end{aligned}
$$

In English, these hypothesis tests correspond to a higher expected number of minutes asleep with less variability in the number of minutes of sleep I experience on the Tempur-Pedic® mattress vs a traditional spring mattress. In addition, I would also like to test for the presence of a positive 'effect size' as formulated by Kruschke.

4 Results and Discussion

I used python and the pandas, numpy, seaborn, matplotlib.pyplot, and the pymc3 libraries to implement the formulation above and generate the results below. I used the Hamiltonian No-U-Turn Sampler (NUTS) with 500 burn in samples and 12500 samples using four chains. Each of the five model parameters and three derived parameters yielded Gelman-Rubin potential scale reduction factors less than 0.0001 away from one, indicating MCMC convergence[2]. See Appendix A for traceplots and plots on individual group statistics and Appendix B for all source code.

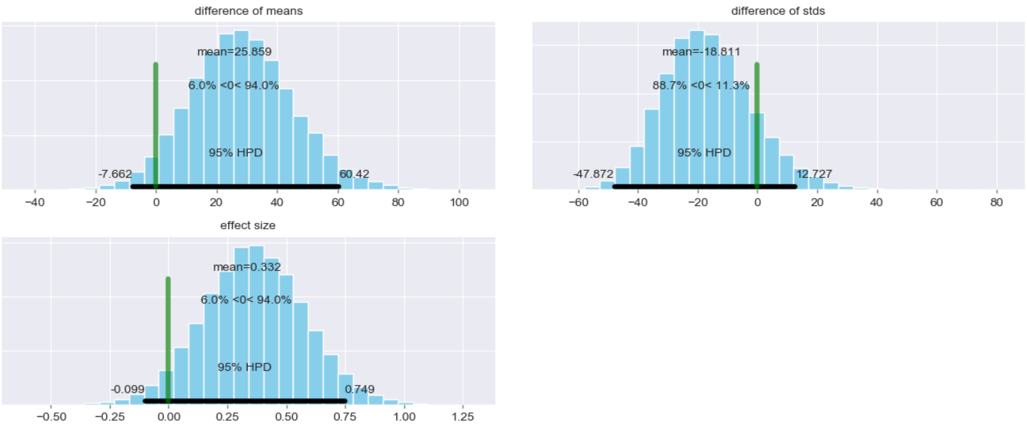


Figure 3: Treatment and Control Group Effect Size: Difference in Means and Standard Deviations

Hypothesis	μ	95% Credible Set
$\mu^*_{\text{treatment}} - \mu^*_{\text{control}} > 0$	25.859	$[-7.662, 60.42]$
$\sigma^*_{\text{treatment}} - \sigma^*_{\text{control}} < 0$	-18.511	$[-47.872, 12.727]$

On average, I have been sleeping ∼26 minutes more per night since I began sleeping on the Tempur-Pedic® mattress. In addition, there has been a reduction in the variance of my overall sleep activity to the tune of ∼19 minutes per night. Even though zero is contained in both of our hypothesis test 95% Credible Sets, we still have very strong evidence that the Tempur-Pedic® mattress improves the overall number of minutes I sleep each night (94% of $\mu^*_{\text{treatment}} - \mu^*_{\text{control}}$ values > 0) and decreases the overall variability in my sleep schedule (∼89% of $\sigma^*_{\text{treatment}} - \sigma^*_{\text{control}}$ values < 0). In addition to confirmation of the original hypothesis, we see that there is very strong evidence for an overall positive treatment effect (94% of draws have effect size > 0).

5 Conclusions and Future Work

One month into owning a Tempur-Pedic® mattress, I have experienced ≈ 26 minute average increase in minutes asleep each night and ≈ 25% reduction in the variance of my overall sleep activity. There is a 94% chance of a positive treatment effect where tempurpedic = True is defined to be the 'treatment' group. From a qualitative perspective, I have noticed a big difference in the quality of my sleep - I wake up feeling rested and refreshed each morning. A potential follow on to this analysis would be to control for 'naps' and only consider sleep corresponding to the maximum sleep start time for any given day or subset the data for MINUTES_ASLEEP > 210 minutes (three and a half hours). From a pure statistics perspective, this is bound to have an effect on the difference of standard deviations and potentially the difference of means. However there is validity to the current approach because from an outcome perspective, I should not need to nap as often if I am getting restful sleep at night (this perspective is subjective to each individual's sleep needs). I plan to keep using the Fitbit® Flex 2 to record my sleep data and will continue to use this methodology to measure the effects of my new Tempur-Pedic® mattress. Hopefully these initial results carry forward!

A Plots

Plot below is the traceplot of the 4 chains of 13,000 samples (12,500 samples and 500 Burn-In) drawn using Hamiltonian No-U-Turn Sampler (NUTS). We see good convergence characteristics amongst all four chains and full exploration of the specified parameter spaces.

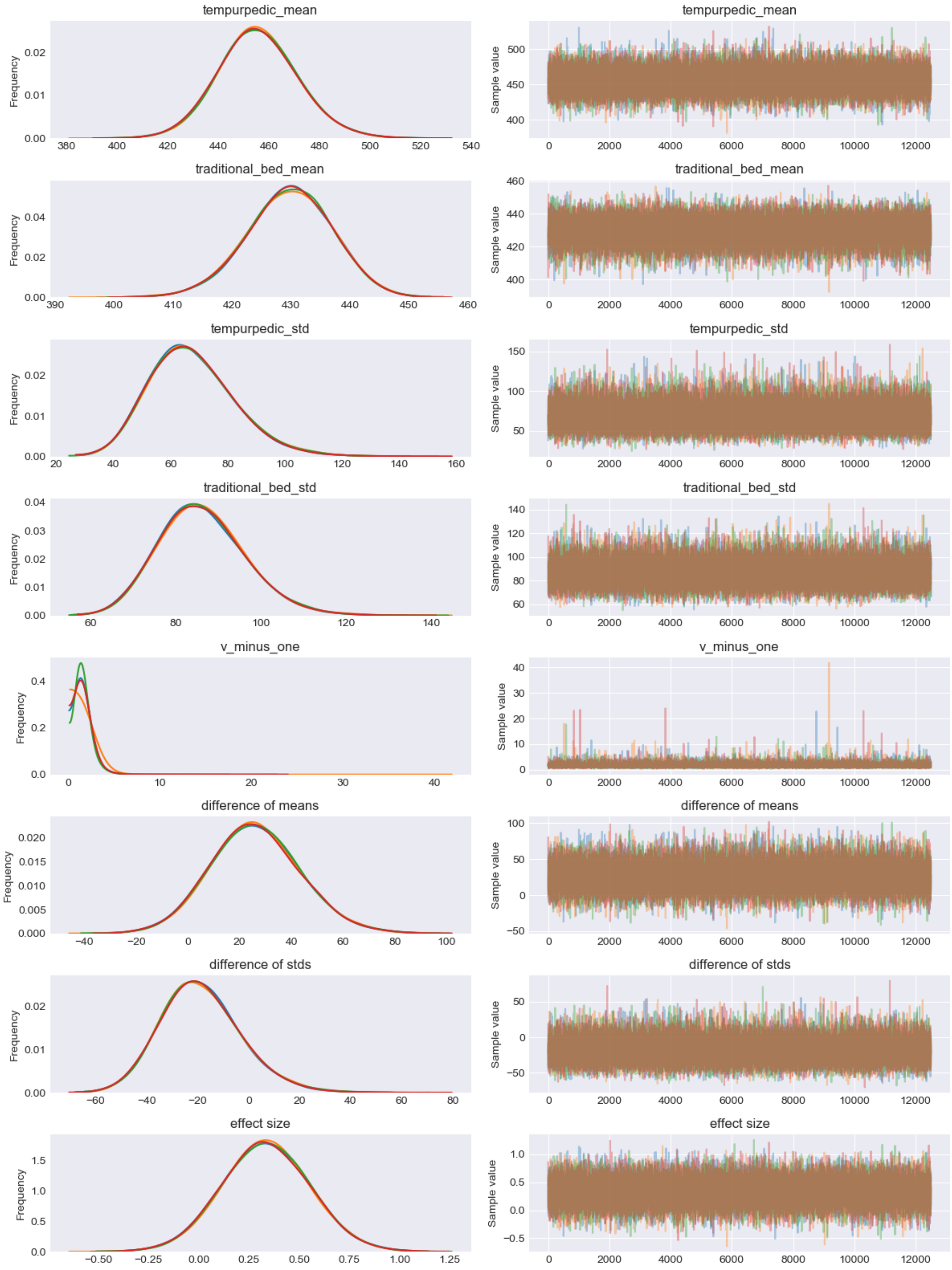


Figure 4: NUTS Traceplot

Plot below is of the individual group means, standard deviations, and the population ‘normality’ parameter ν:

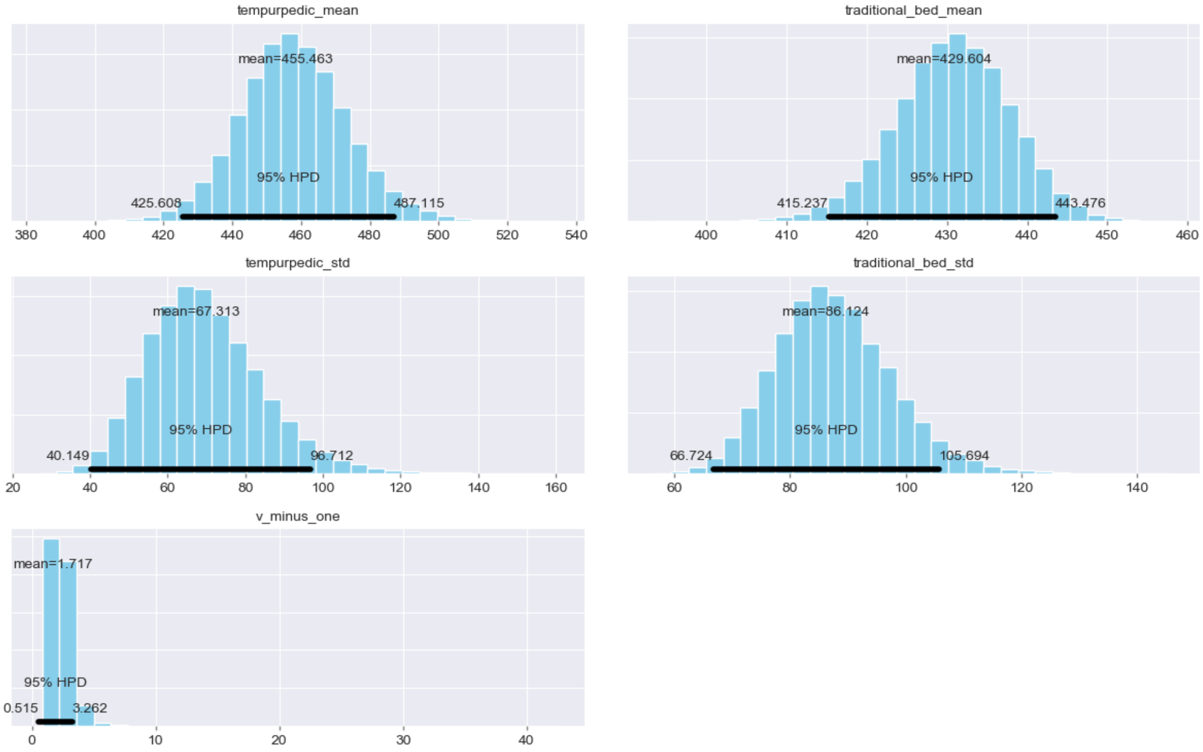


Figure 5: Treatment and Control Group Means and Standard Deviations

B Source Code

Python code below used to produce results. Built/tested using

python version: 3.5.5
pandas version: 0.23.4
numpy version: 1.15.2
seaborn version: 0.7.1
pymc3 version: 3.5

```
# coding: utf-8
# reference: https://docs.pymc.io/notebooks/BEST.html

# Load libraries

import pandas as pd
from glob import glob
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use(’seaborn-darkgrid’)
import numpy as np
np.random.seed(37)
import pymc3 as pm
from datetime import datetime
start = datetime.now()
print(’start␣time:␣{}’.format(start.strftime(’%Y-%m-%d␣%H:%M’)))
print(’Running␣on␣PyMC3␣v{}’.format(pm.__version__))

# Global Parameters

NSAMPLES = 12500
NCHAINS = 4
NCORES = 1
NTUNE = 500

g1 = ’tempurpedic’
g2 = ’traditional_bed’
measure = ’Sleep␣Efficiency’

dict_user_group_labels = {’group1’: g1,
                          ’group2’: g2,
                          ’measure’: measure}


# Load data

df = pd.read_csv(’DATA.csv’)

# Preprocessing

df[’value’] = df[’MINUTES_ASLEEP’].copy()
group1 = df[df[’group’] == ’Treatment’]
group2 = df[df[’group’] == ’Control’]
print(’\n\ngroup1:\n\n’)
print(group1.info())
print(’\n\ngroup2:\n\n’)
print(group2.info())

# Data exploration

graph = sns.categorical.countplot(df[’tempurpedic’], palette=’colorblind’)
```

```
# reference: https://stackoverflow.com/a/55105132
for p in graph.patches:
    height = p.get_height()
    graph.text(p.get_x()+p.get_width()/2., height + 0.2,height ,ha="center")

plt.title('Number␣of␣Nights␣on␣Tempur-Pedic')
plt.savefig('explore1.png', bbox_inches='tight')
plt.show()

sns.distplot(group1['value'],label='Treatment')
sns.distplot(group2['value'],label='Control')
plt.ylabel('Density')
plt.title('MINUTES_ASLEEP␣Kernel␣Density␣Estimate')
plt.legend()
plt.savefig('explore2.png', bbox_inches='tight')
plt.show()

group2_steps_list = group2.value.tolist()
group1_steps_list = group1.value.tolist()

y1 = np.array(group1_steps_list)
y2 = np.array(group2_steps_list)
y = pd.DataFrame(dict(value=np.r_[y1, y2],
        group=np.r_[['{}_steps_list'\
        .format(dict_user_group_labels['group2'])]*len(group2_steps_list),
        ['{}_steps_list'\
        .format(dict_user_group_labels['group1'])]*len(group1_steps_list)]))
y.hist('value', by='group')
print('Data:␣\n{}'.format(pd.concat([df.head(1), df.tail(1)])\
  .reset_index().drop(['index', 'level_0'],axis=1).to_string()))

# Hyperpriors from pooled data for mu prior by group

mu_m = y.value.mean()
mu_s = y.value.std() * 1000 # 1000 sigma spread on prior mean standard deviation

# Mu prior by group

with pm.Model() as model:
    group1_mean = pm.Normal('{}_mean'\
        .format(dict_user_group_labels['group1']),
        mu_m, sd=mu_s)
    group2_mean = pm.Normal('{}_mean'\
        .format(dict_user_group_labels['group2']),
        mu_m, sd=mu_s)

# Bounded sigma prior by group

sigma_low = max(y.value.std() * 1/1000, 1)
sigma_high = y.value.std() * 1000

with model:
    group1_std = pm.Uniform('{}_std'\
        .format(dict_user_group_labels['group1']),
        lower=sigma_low, upper=sigma_high)
    group2_std = pm.Uniform('{}_std'\
        .format(dict_user_group_labels['group2']),
        lower=sigma_low, upper=sigma_high)

# Shifted exponential "normality" parameter

with model:
    nu = pm.Exponential('nu_minus_one', 1/29.) + 1
```

```
# t distribution likelihood by group
## pymc3 StudentT distribution is parameterized by lambda which is precision

with model:
    lambda1 = group1_std**-2
    lambda2 = group2_std**-2

    group1 = pm.StudentT('{}'\
        .format(dict_user_group_labels['group2']),
        nu=nu, mu=group1_mean, lam=lambda1, observed=y1)
    group2 = pm.StudentT('{}'\
        .format(dict_user_group_labels['group1']),
        nu=nu, mu=group2_mean, lam=lambda2, observed=y2)

# Three hypothesis of interest

with model:
    diff_of_means = pm.Deterministic('difference␣of␣means',
        group1_mean - group2_mean)
    diff_of_stds = pm.Deterministic('difference␣of␣stds',
        group1_std - group2_std)
    effect_size = pm.Deterministic('effect␣size',
        diff_of_means / np.sqrt((group1_std**2 + group2_std**2) / 2))


# Sample draws using NUTS

with model:
    trace = pm.sample(NSAMPLES, chains=NCHAINS, cores=NCORES, tune=NTUNE)


# Gelman-Rubin statistics

gr = pm.diagnostics.gelman_rubin(trace)

print('Gelman-Rubin:␣\n{}'.format(gr))

# NUTS Traceplot

pm.traceplot(trace)
plt.savefig('trace.png')
plt.show()

# Model Runtime

plot_time = datetime.now()
model_hours, remainder = divmod((plot_time - start).total_seconds(), 3600)
model_minutes, model_seconds = divmod(remainder, 60)

print('model␣took␣{}␣hours,\
␣␣␣␣␣␣␣␣{}␣minutes,␣and␣{}␣seconds␣to␣run'\
        .format(round(model_hours),round(model_minutes),round(model_seconds)))


# Plot Figure 5

pm.plot_posterior(trace,
        varnames=['{}_mean'.format(dict_user_group_labels['group1']),
        '{}_mean'.format(dict_user_group_labels['group2']),
        '{}_std'.format(dict_user_group_labels['group1']),
        '{}_std'.format(dict_user_group_labels['group2']),
        'nu_minus_one'],
        color='#87ceeb')

plt.savefig('results1.png')
```

```
# Plot Figure 3

pm.plot_posterior(trace,
        varnames=[’difference␣of␣means’,’difference␣of␣stds’, ’effect␣size’],
        ref_val=0,
        color=’#87ceeb’)

plt.savefig(’results2.png’)
plt.show()
```

References

[1] John K. Kruschke. Bayesian estimation supersedes the t test. *Journal of Experimental Psychology: General*, 142(2):573603, 2013.

[2] Stephen P Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *J. Comput. Graph. Stat.*, 7(4):434–455, 1998.