Homework 2: Data Statistics

J.R. Powers-Luhn

I. Abstract

Prior to generating *post hoc* models from data it is necessary to examine the data to validate any assumptions made in collecting it. That examination should include explicitly stated assumptions as well as attempts to validate those assumptions prior to drawing conclusions about the data. Generating models from biased data can obviously result in biased models, but those biases may be more subtle in the model than in the data that produced it. In order to demonstrate the form that such an examination might take, statistical summary calculations were performed on data collected for a 1985 body weight study. It was determined that the sampled data did not conform to expected (Gaussian) distributions, and that at least one significant outlier was present in a parameter used to calculate lean body mass in the 1985 paper. An examination was made of the cross-correlations between the measured data and the body fat percentage predictions, revealing that abdominal circumference may be a good predictor of body fat percentage (correlation value of 0.81).

II. Introduction

Understanding the distribution of a signal with a random component is a necessary first step in analyzing or modeling that signal. In order to recover the true nature of a physical phenomena, we measure the value of the signal plus some random noise, $M = S + N$. Misunderstanding the noise results in misunderstanding the recovered signal, which may lead to incorrect conclusions about the data. If the noise is underestimated then this could lead to a true hypothesis being rejected. Similarly, mischaracterizing the distribution from which the noise is generated could result in a systematic bias in models.

A. Gaussian Distribution

A common physical model for a single sampled quantity is a mean value plus some variation drawn from a Gaussian distribution, as in equation 1.

$$G(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1}$$

This assumption, however, is not always valid. In order to test this, we can compare the properties of the measured distribution to the hypothetical distribution by measuring the mathematical properties called *moments*. The nth moment of a distribution is described by equation 2.

$$\mu_n = \int_{\infty}^{\infty}(x-c)^n f(x)dx \tag{2}$$

The moments of a set of samples (plus some other characteristics: the median, trimmed mean, and standard deviation) are characteristic of a distribution. In order to illustrate this, statistical properties were calculated for a real-life data set made of sampled data [1]. This data set consists of 252 samples each of fourteen quantities:

- Age (yr)
- Weight (lbs)
- Height (inches)
- Adiposity index $(\mathrm{kg/m^2})$
- Neck circumference (cm)
- Chest circumference (cm)
- Ab circumference (cm)
- Hip circumference (cm)
- Thigh circumference (cm)
- Knee circumference (cm)
- Ankle circumference (cm)
- Extended bicep circumference (cm)
- Forearm circumference (cm)
- Wrist circumference (cm)

143 of these measured values were used to derive the lean body weight equation 3, with abdominal (Ab) and wrist (Wr) circumferences measured in cm, weight (Wt) measured in kg, and height (Hgt) measured in m. This was then validated with 109 additional measurements. From this it was expected that lean body weight would be strongly positively correlated with weight and height, weakly positively correlated with wrist circumference, and somewhat negatively correlated with abdominal circumference. Because of the nonlinear correlation between age and lean body weight, it was expected that the correlation between these two values would be low. Additionally it was expected that percent body fat would inversely correlate with lean body weight.

$$\begin{aligned} LBW = 17.298 &+ 0.89946(Wt) \\ &- 0.2783(Age) + 0.002617(Age^2) \\ &+ 17.819(Hgt) - 0.6798(Ab - Wr) \end{aligned} \tag{3}$$

Prior to examination, it was predicted that age would appear to conform to a nearly uniform distribution (possibly with a tail on the high end of the distribution) while all other sampled variable would appear to be Gaussian.

III. Results

A. Summary Statistics

In order to determine the distribution of the sampled input variables, summary statistics were calculated for each variable. The results of these calculations are reported in table I.

These statistics revealed that the distribution of the input variables was neither uniform nor Gaussian. Only three measured variables (neck circumference, abdomen circumference, and thigh circumference) had kurtosis values within ±1 of 3.

	Parameter	Max	Min	Mean	Median	20% Trimmed Mean	Standard Deviation	Variance	Skewness	Kurtosis
0	Age	81.00	22.0	44.88	43.00	44.44	12.58	158.18	0.28	-0.43
1	Weight	363.15	118.5	178.92	176.50	176.55	29.33	860.30	1.20	5.14
2	Height	77.75	29.5	70.15	70.00	70.25	3.66	13.36	-5.35	58.34
3	Adiposity Index	48.90	18.1	25.44	25.05	25.056	3.64	13.26	1.55	6.56
4	Neck circumference	51.20	31.1	37.99	38.00	37.93	2.43	5.88	0.55	2.64
5	Chest circumference	136.20	79.3	100.82	99.65	100.13	8.41	70.79	0.68	0.94
6	Abdomen circumference	148.10	69.4	92.56	90.95	91.81	10.76	115.81	0.83	2.18
7	Hip circumference	147.70	85.0	99.90	99.30	99.33	7.15	51.12	1.49	7.30
8	Thigh circumference	87.30	47.2	59.40	59.00	59.13	5.24	27.45	0.82	2.59
9	Knee circumference	49.10	33.0	38.59	38.50	38.49	2.41	5.79	0.51	1.02
10	Ankle circumference	33.90	19.1	23.10	22.80	22.91	1.69	2.86	2.24	11.68
11	Extended bicep circumference	45.00	24.8	32.27	32.05	32.16	3.02	9.09	0.28	0.46
12	Forearm circumference	34.90	21.0	28.66	28.70	28.70	2.02	4.07	-0.22	0.82
13	Wrist circumference	21.40	15.8	18.23	18.30	18.22	0.93	0.87	0.28	0.36
14	% Bodyweight	45.10	0.0	18.94	19.00	18.86	7.74	59.84	0.14	-0.32

TABLE I
TABLE OF SUMMARY STATISTICS FOR DATA RECORDED IN [1]

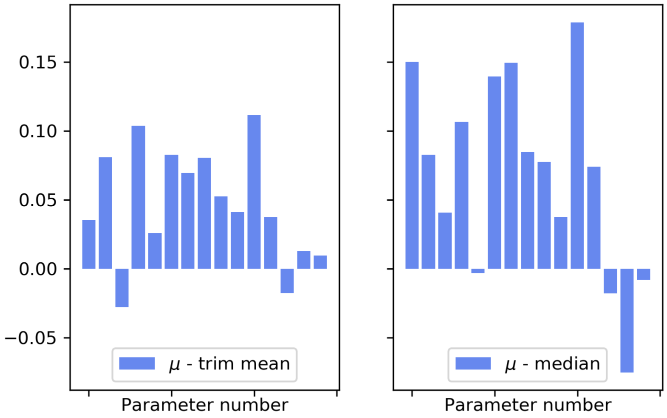


Fig. 1. Separation between the mean and trimmed mean (left) and median (right), in units of standard deviation from the sampled data set. Height, the parameter with the furthest outlier, is represented by the third bar from the left. In spite of having the furthest outlier, it is in the smaller half of differences between the mean and either the trimmed mean or median.

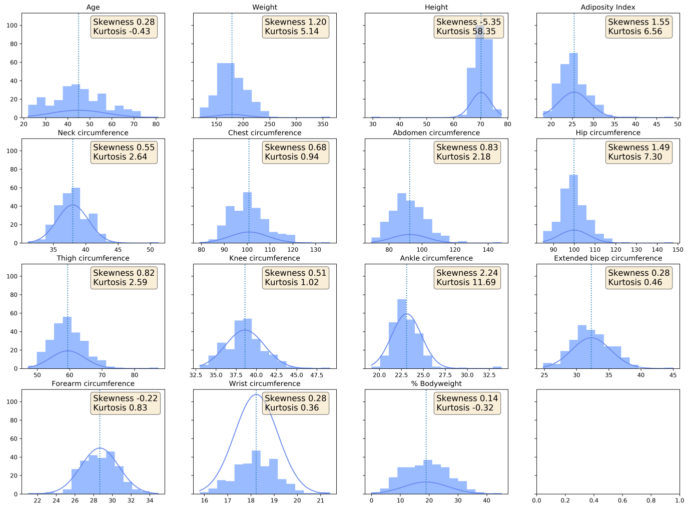


Fig. 2. Comparison of measured data to Gaussian distribution. The histogram for each measured parameter is overlayed with a Gaussian probability distribution function with the same mean and standard deviation. Higher-order moments (skewness and kurtosis) are listed in the insets.

Height had a kurtosis value of 58.35, likely due to the presence of a single outlier 11.1σ below the mean value of $70.15\,\mathrm{in}$. In spite of the median and 20% trimmed mean being "robust to outliers" [2], in general, the mean, median, and trimmed mean were close to each other (within 0.2 standard deviations of the sample), as shown in figure 1.

In order to quickly process these distribution relative to the assumed Gaussian, histograms of the each distribution were plotted on the same axes as a Gaussian probability distribution function scaled to the number of observations in the data set (figure 2). Vertical lines were drawn to identify the mean of each variable. This serves to illustrate the deviation of each sampled variable from a Gaussian distribution.

This does not necessarily indicate that the underlying population does not follow a Gaussian distribution. To illustrate this, figure 3 shows four normalized histograms drawn from the same population. The sample size for each histogram is shown above the plot.

B. Correlation

Adiposity index (equation 4, where h is hip circumference in cm and H is height in m) has been shown to correlate strongly with body fat percentage–but the correlation factors cited range from 0.5 to 0.6 [3]. This data set shows a stronger correlation, reaching 0.76.

$$BAI = \frac{h}{H^{3/2}} - 18 \tag{4}$$

Contrary to expectations, neither weight nor height were strongly correlated with body fat percentage in spite of the large coefficient in equation 3. This may be partially explained by the relationship between height and body adipose index in equation 4, which ascribes an inverse relationship to height. Since BAI is strongly correlated with body fat (correlation coefficient 0.73), the relationship between height and body fat would also be inverse. Since the correlation coefficient only captures linear relationships, this low value could obscure

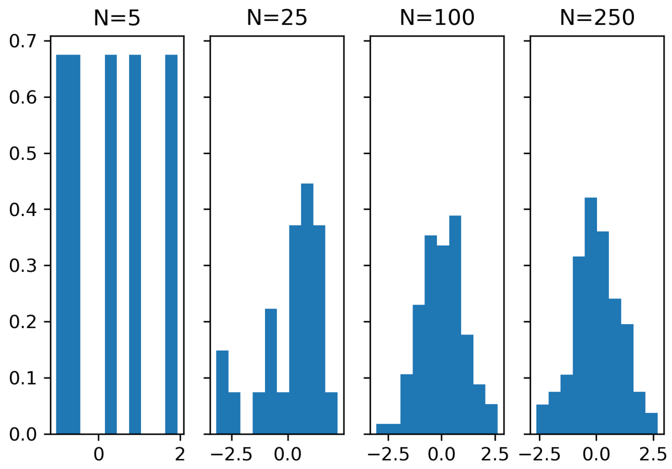
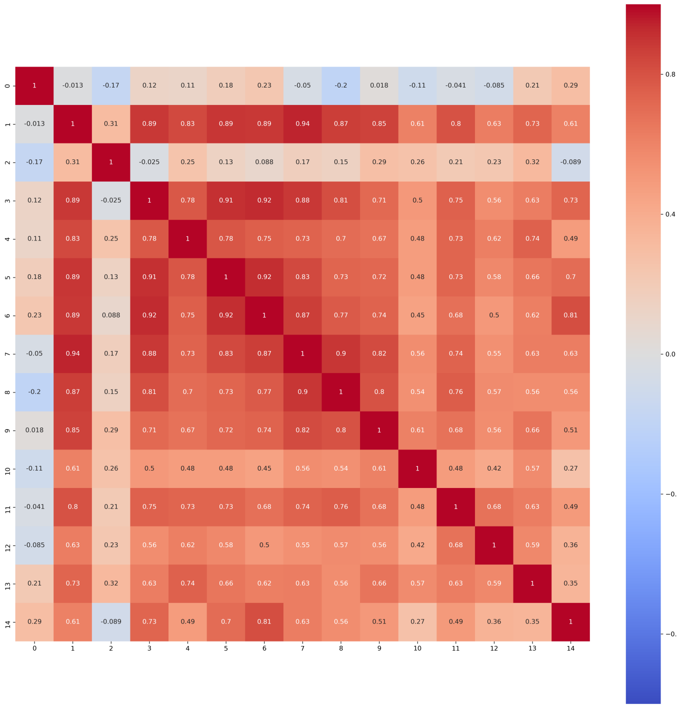


Fig. 3. Four normalized sample distributions drawn from the same population. Small sample sizes result in difficulty identifying the distribution of the underlying population, and even large sample sizes may still result in a sample with different moments than the population that produced them.

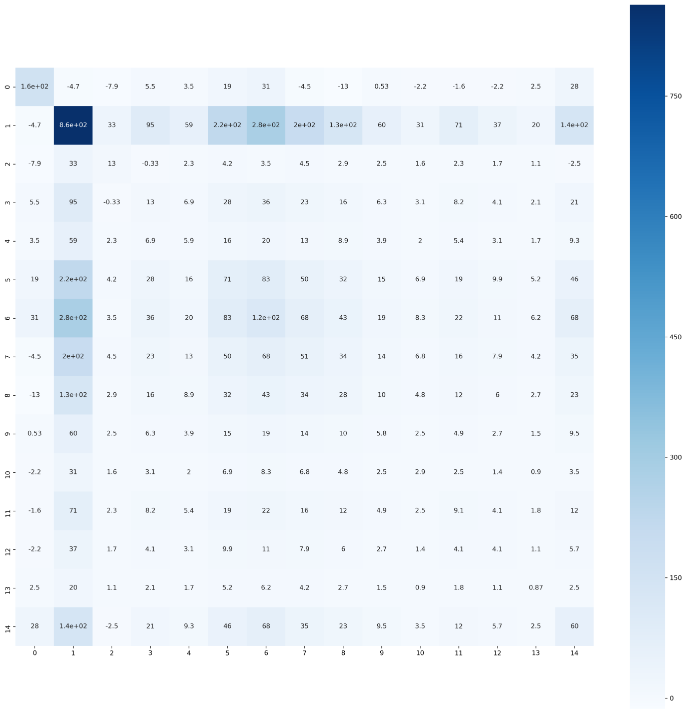


	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1	-0.013	-0.17	0.12	0.11	0.18	0.23	-0.05	-0.2	0.018	-0.11	-0.041	-0.085	0.21	0.29
1	-0.013	1	0.31	0.89	0.83	0.89	0.89	0.94	0.87	0.85	0.61	0.8	0.63	0.73	0.61
2	-0.17	0.31	1	-0.025	0.25	0.13	0.088	0.17	0.15	0.29	0.26	0.21	0.23	0.32	-0.089
3	0.12	0.89	-0.025	1	0.78	0.91	0.92	0.88	0.81	0.71	0.5	0.75	0.56	0.63	0.73
4	0.11	0.83	0.25	0.78	1	0.78	0.75	0.73	0.7	0.67	0.48	0.73	0.62	0.74	0.49
5	0.18	0.89	0.13	0.91	0.78	1	0.92	0.83	0.73	0.72	0.48	0.73	0.58	0.66	0.7
6	0.23	0.89	0.088	0.92	0.75	0.92	1	0.87	0.77	0.74	0.45	0.68	0.5	0.62	0.81
7	-0.05	0.94	0.17	0.88	0.73	0.83	0.87	1	0.9	0.82	0.56	0.74	0.55	0.63	0.63
8	-0.2	0.87	0.15	0.81	0.7	0.73	0.77	0.9	1	0.8	0.54	0.76	0.57	0.56	0.56
9	0.018	0.85	0.29	0.71	0.67	0.72	0.74	0.82	0.8	1	0.61	0.68	0.56	0.66	0.51
10	-0.11	0.61	0.26	0.5	0.48	0.48	0.45	0.56	0.54	0.61	1	0.48	0.42	0.57	0.27
11	-0.041	0.8	0.21	0.75	0.73	0.73	0.68	0.74	0.76	0.68	0.48	1	0.68	0.63	0.49
12	-0.085	0.63	0.23	0.56	0.62	0.58	0.5	0.55	0.57	0.56	0.42	0.68	1	0.59	0.36
13	0.21	0.73	0.32	0.63	0.74	0.66	0.62	0.63	0.56	0.66	0.57	0.63	0.59	1	0.35
14	0.29	0.61	-0.089	0.73	0.49	0.7	0.81	0.63	0.56	0.51	0.27	0.49	0.36	0.35	1

Fig. 4. Correlations between recorded parameters in the data

other functional relationships between the two parameters. Age was weakly correlated with body fat percentage (as well as all other parameters), which raises the question of how the fit parameters were obtained by Penrose, et al. Abdomen circumference was most strongly correlated with body fat percentage (0.81). Adiposity index also correlated strongly with central circumference measurements (closer to the abdomen/torso), with coefficients >0.7 for all measured circumferences except wrist, forearm, and ankle.

The covariance was also computed between each parameter in the data set. Because the covariance is not normalized, it is harder to use as a tool. This is demonstrated in figure 5, where most values fall into the 0 to 50 (as shown in figure 6). Covariance is not normalized, and the varying scale makes it difficult to use intuitively to identify relationships between data.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	1.6e+02	-4.7	-7.9	5.5	3.5	19	31	-4.5	-13	0.53	-2.2	-1.6	-2.2	2.5	28
1	-4.7	8.6e+02	33	95	59	2.2e+02	2.8e+02	2e+02	1.3e+02	60	31	71	37	20	1.4e+02
2	-7.9	33	13	-0.33	2.3	4.2	3.5	4.5	2.9	2.5	1.6	2.3	1.7	1.1	-2.5
3	5.5	95	-0.33	13	6.9	28	36	23	16	6.3	3.1	8.2	4.1	2.1	21
4	3.5	59	2.3	6.9	5.9	16	20	13	8.9	3.9	2	5.4	3.1	1.7	9.3
5	19	2.2e+02	4.2	28	16	71	83	50	32	15	6.9	19	9.9	5.2	46
6	31	2.8e+02	3.5	36	20	83	1.2e+02	68	43	19	8.3	22	11	6.2	68
7	-4.5	2e+02	4.5	23	13	50	68	51	34	14	6.8	16	7.9	4.2	35
8	-13	1.3e+02	2.9	16	8.9	32	43	34	28	10	4.8	12	6	2.7	23
9	0.53	60	2.5	6.3	3.9	15	19	14	10	5.8	2.5	4.9	2.7	1.5	9.5
10	-2.2	31	1.6	3.1	2	6.9	8.3	6.8	4.8	2.5	2.9	2.5	1.4	0.9	3.5
11	-1.6	71	2.3	8.2	5.4	19	22	16	12	4.9	2.5	9.1	4.1	1.8	12
12	-2.2	37	1.7	4.1	3.1	9.9	11	7.9	6	2.7	1.4	4.1	4.1	1.1	5.7
13	2.5	20	1.1	2.1	1.7	5.2	6.2	4.2	2.7	1.5	0.9	1.8	1.1	0.87	2.5
14	28	1.4e+02	-2.5	21	9.3	46	68	35	23	9.5	3.5	12	5.7	2.5	60

Fig. 5. Correlations between recorded parameters in the data

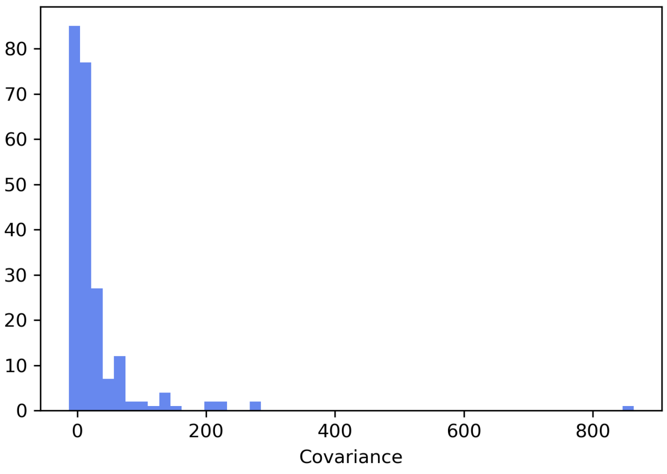


Fig. 6. The covariance values were strongly clustered around zero, with one value (auto-covariance of weight with itself) dominating by nearly three orders of magnitude.

IV. Conclusions

Summary statistics calculated for a real-world data set revealed that assumptions about sampled data (e.g. that it closely follows a Gaussian or uniform distribution) may not be valid–even with a significant sample size. Statistical moments as well as other properties of a distribution such as median and trimmed mean provide a straightforward tool to approximately determine whether or not a sample conforms to a standard distribution.

These analyses reveal that the data collected deviated from the normal distribution. Further, it showed the value of carful analysis of the parameters of the sampled data, including recognizing the thresholds required to confirm or reject distribution hypotheses and the sometimes subtle nature of outliers in the data.

Values closely correlated with body fat percentage (and therefore inversely correlated with lean body mass) were identified, and the correlation of those values with other measured quantities was calculated and discussed. The correlation between circumference measurements of the torso, neck, and thighs were found to be linked (correlation >0.7), though a causal relationship could not be inferred.

Now that the measured data are better understood, further work can be performed to see if the lean body mass formula described in equation 3 can be improved upon, possibly with a ridge regression or some other hypothesis class.

References

[1] K W Penrose, A G Nelson, and A G Fisher. “Generalized Body Composition Prediction Equation For Men Using Simple Measurement Techniques”. In: *Medicine & Science in Sports & Exercise* 17.2 (1985). ISSN: 0195-9131.

[2] J Coble. *Evaluating Data: Time Domain Statistics*. Aug. 2018.

[3] Richard N. Bergman, Darko Stefanovski, Thomas A. Buchanan, et al. “A Better Index of Body Adiposity”. In: *Obesity* 19.5 (May 2011), pp. 1083–1089. ISSN: 1930-7381. DOI: 10.1038/oby.2011.38. URL: http://doi.wiley.com/10.1038/oby.2011.38.

V. Appendix

Python code used to perform calculations and generate graphics.

```
#!/usr/bin/env python
# coding: utf 8

# # NE 579 Homework Number 2: Data Statistics

# ## Import required libraries

# In[1]:


get_ipython().run_line_magic('matplotlib', 'inline')
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import scipy.io as sio
from scipy.stats import trim_mean, skew, kurtosis, norm


# In[2]:


sns.color_palette('coolwarm')


# ## Load the data from the file

# In[3]:


fn = 'hwkdata.mat'
data_dict = sio.loadmat(fn)
print(data_dict.keys())


# In[4]:


data_dict['x'].shape


# In[5]:


data_dict['y'].shape


# In[6]:


data = np.append(data_dict['x'], data_dict['y'], 1)


# In[7]:
```

```
data.shape


# Now the 2d array ‘data‘ contains 252 samples, each with 14 input variables and 1 output var

# I will want to label the columns later...

# In[8]:


parameter_names = [
    'Age',
    'Weight',
    'Height',
    'Adiposity␣Index',
    'Neck␣circumference',
    'Chest␣circumference',
    'Abdomen␣circumference',
    'Hip␣circumference',
    'Thigh␣circumference',
    'Knee␣circumference',
    'Ankle␣circumference',
    'Extended␣bicep␣circumference',
    'Forearm␣circumference',
    'Wrist␣circumference',
    '%␣Bodyweight'
]


# ## Calculate statistical properties of the data:
#    Maximum
#    Minimum
#    Mean
#    Median
#    20% trimmed mean
#    Standard deviation
#    Variance
#    Skewness
#    Kurtosis

# ### Maximum

# In[9]:


np.max(data, axis=0)


# ### Minimum

# In[10]:


np.min(data, axis=0)


# ### Mean
# In[11]:


np.mean(data, axis=0)


# ### Median

# In[12]:


np.median(data, axis=0)


# ### 20% Trimmed Mean

# In[13]:


trim_mean(data, 0.2, axis=0)


# ### Standard Deviation

# In[14]:


np.std(data, axis=0)


# ### Variance

# In[15]:


np.var(data, axis=0)


# ### Skewness

# In[16]:


skew(data, axis=0)


# ### Kurtosis

# In[17]:


kurtosis(data, axis=0)


# #### Also calculate $Kurt   3$

# In[18]:
```

```
kurtosis(data, axis=0)  3 * np.ones_like(kurtosis(data, axis=0))


# #### Difference between mean and median

# In[70]:


with sns.color_palette('coolwarm'):
    f = plt.figure()
    s = np.divide(np.mean(data, axis=0)  np.median(data, axis=0), np.std(data, axis=0))
    ax = plt.subplot(121)
    plt.bar(np.arange(s.shape[0]), s, label=r'$\mu$␣ ␣median')
    #plt.bar(np.arange(s.shape[0]), np.mean(data, axis=0)  np.median(data, axis=0), label=r'$
    #plt.bar(np.arange(s.shape[0]), kurtosis(data, axis=0) / kurtosis(data, axis=0).max(axis=0
    #         alpha=0.5, label='Kurtosis')
    #plt.bar(np.arange(s.shape[0]), skew(data, axis=0),
    #         alpha=0.5, label='Skewness')
    plt.legend(loc="upper␣right")
    plt.savefig('images/mu_minus_median.png', dpi=300, bbox_inches='tight')
    plt.show()


# From this plot we see that the mean (sensitive to outliers) exceeds the median (insensitive

# In[69]:


with sns.color_palette('coolwarm'):
    plt.bar(np.arange(np.mean(data, axis=0).shape[0]),
        np.divide(np.mean(data, axis=0)  trim_mean(data, 0.2, axis=0), np.std(data, axis=0)),
    plt.legend(loc='upper␣right')
    #plt.bar(np.arange(trim_mean(data, 0.2, axis=0).shape[0]),
    #         trim_mean(data, 0.2, axis=0))
    plt.savefig('images/mu_minus_trimmed_mean.png', dpi=300, bbox_inches='tight')
    plt.show()


# In[85]:


with sns.color_palette('coolwarm'):
    f = plt.figure()
    ax1 = f.add_subplot(121)
    ax1.bar(np.arange(np.mean(data, axis=0).shape[0]),
            np.divide(np.mean(data, axis=0)  trim_mean(data, 0.2, axis=0),
                      np.std(data, axis=0)),
            label=r'$\mu$␣ ␣trim␣mean')
    ax1.legend(loc='lower␣center')

    ax2 = f.add_subplot(122, sharey=ax1)
    s = np.divide(np.mean(data, axis=0)  np.median(data, axis=0), np.std(data, axis=0))
    ax2.bar(np.arange(s.shape[0]), s, label=r'$\mu$␣ ␣median')
    plt.setp(ax2.get_yticklabels(), visible=False)
    plt.legend(loc="lower␣center")
    plt.setp(ax1.get_xticklabels(), visible=False)
    plt.setp(ax2.get_xticklabels(), visible=False)

    ax1.set_xlabel(’Parameter␣number’)
    ax2.set_xlabel(’Parameter␣number’)

    plt.savefig(’images/robust_statistics.png’, dpi=300, bbox_inches=’tight’)



#
# # Now look at covariance and correlation

# In[103]:


plt.figure(figsize=(20,20))
sns.heatmap(np.corrcoef(data.T), vmin= 1, vmax=1, center=0, cmap=’coolwarm’,
            square=True, annot=True)
plt.savefig(’images/correlation_heatmap.png’, dpi=300, bbox_inches=’tight’)
plt.show()


# ## ...and covariance

# In[108]:


plt.figure(figsize=(20,20))
sns.heatmap(np.cov(data.T), cmap=’Blues’,
            square=True, annot=True)
plt.savefig(’images/covariance_heatmap.png’, dpi=300, bbox_inches=’tight’)
plt.show()


# ## Distribution of covariance

# In[120]:


with sns.color_palette(’coolwarm’):
    plt.hist(np.cov(data.T).flatten(), bins=50)
    plt.xlabel(’Covariance’)
    plt.savefig(’images/covariance_dist.png’, dpi=300, bbox_inches=’tight’)
    plt.show()


# # Pandas

# In[23]:


stat_names = [
    ”Parameter”, ”Max”, ”Min”, ”Mean”, ”Median”, ”20%␣Trimmed␣Mean”,
    ”Standard␣Deviation”, ”Variance”, ”Skewness”, ”Kurtosis”
]
```

```
# In[24]:


summary_statistics = pd.DataFrame(columns=stat_names)
summary_statistics['Parameter'] = parameter_names
summary_statistics['Max'] = np.max(data, axis=0)
summary_statistics['Min'] = np.min(data, axis=0)
summary_statistics['Mean'] = np.mean(data, axis=0)
summary_statistics['Median'] = np.median(data, axis=0)
summary_statistics['20%␣Trimmed␣Mean'] = trim_mean(data, 0.2, axis=0)
summary_statistics['Standard␣Deviation'] = np.std(data, axis=0)
summary_statistics['Variance'] = np.var(data, axis=0)
summary_statistics['Skewness'] = skew(data, axis=0)
summary_statistics['Kurtosis'] = kurtosis(data, axis=0)


# In[25]:


summary_statistics


# In[58]:


with open('summary_statistics_table.tex', 'w') as f:
    f.write(summary_statistics.to_latex(index=True))


# In[26]:


(summary_statistics['Max']   summary_statistics['Min'] ) / summary_statistics['Standard␣Deviat


# In[27]:


summary_statistics['Mean']


# In[86]:


props = dict(boxstyle='round', facecolor='wheat', alpha=0.5)
with sns.color_palette('coolwarm'):
    f, ax = plt.subplots(4, 4, figsize=(20,15), sharey=True)
#    for i in range(len(parameter_names)):
    for i in range(4):
        for j in range(4):
            if (i*4+j) < len(parameter_names):
                x = np.linspace(np.min(data[:,i*4+j]), np.max(data[:,i*4+j]))
                y = data[:,i*4+j].shape[0]*norm.pdf(x, loc=np.mean(data[:,i*4+j]), scale=np.st
                ax[i,j].plot(x, y)
                ax[i,j].hist(data[:,i*4+j], bins=15)
                ax[i,j].axvline(x=np.mean(data[:,i*4+j]), linestyle='dotted')
                ax[i,j].set_title(f'{parameter_names[i*4+j]}')
                textstr = f'Skewness␣{skew(data[:,i*4+j]):0.2f}\nKurtosis␣{kurtosis(data[:,i*4
                ax[i,j].text(0.5, 0.95, textstr, transform=ax[i,j].transAxes,
                             fontsize=14, verticalalignment='top', bbox=props)
plt.savefig('images/histogram_array.png', dpi=300, bbox_inches='tight')
plt.show()


# In[101]:


N = [5, 25, 100, 250]
D = [norm.rvs(size=n) for n in N]

f, axs = plt.subplots(1, len(N), sharey='all')

for _ in range(len(N)):
    axs[_].hist(D[_], density=True)
    axs[_].set_title(f'N={N[_]}')

plt.savefig('sample_size.png', dpi=300, bbox_inches='tight')
plt.show()


# In[ ]:
```