Group 10 HW3 for Digital Economy & Decision Analytics

Shaoyuan Xia
Jiayi Cong
Danni Luo

Team NO.10

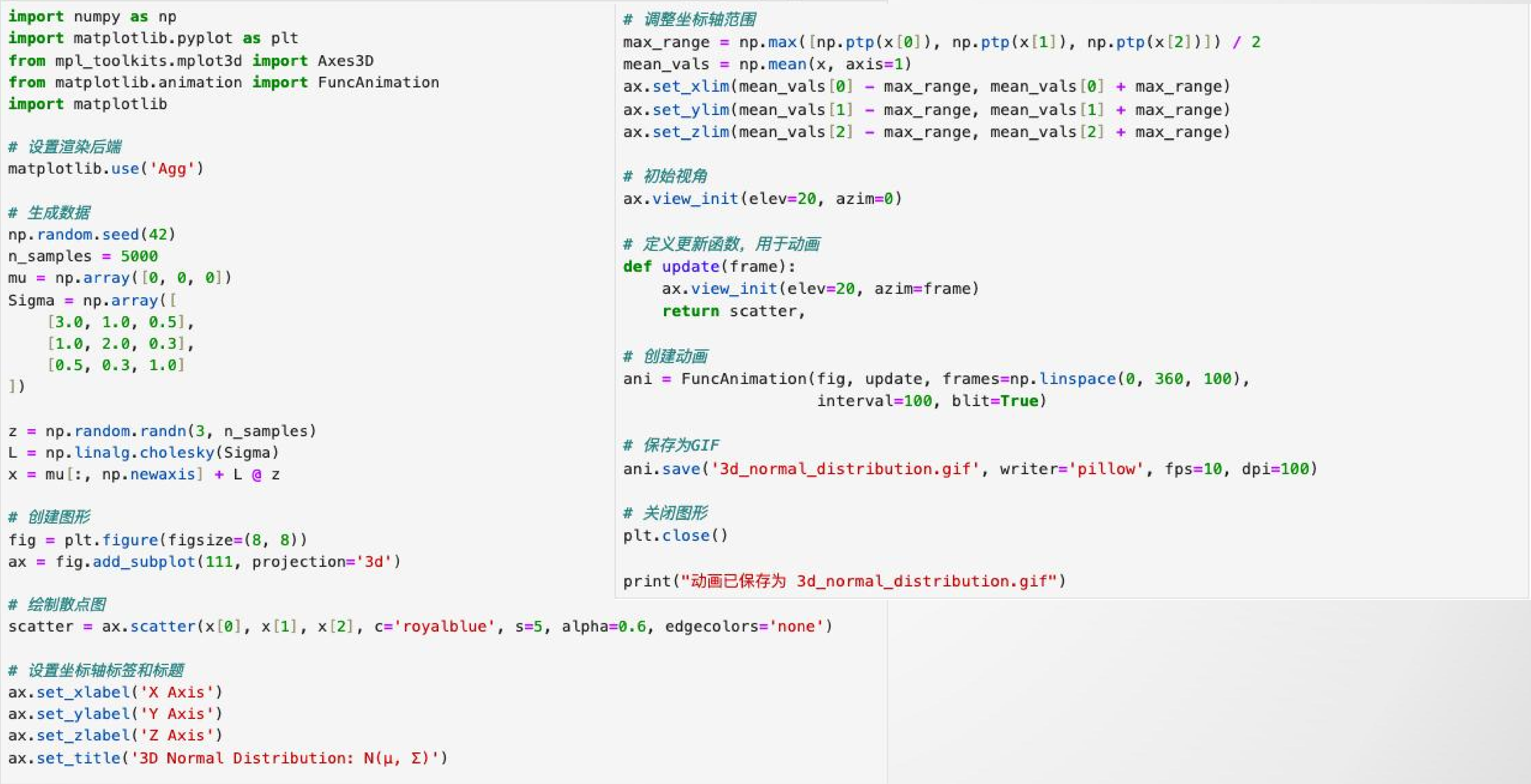
Capital University of Economics and Business, Beijing

Outline of July 10 HW

1. create a N (mu, SIGMA), try in a rotating plot in 3D
2. Redo the LDA example IRTG1792

1. Code for creating a N(μ, Σ) and its rotating plot

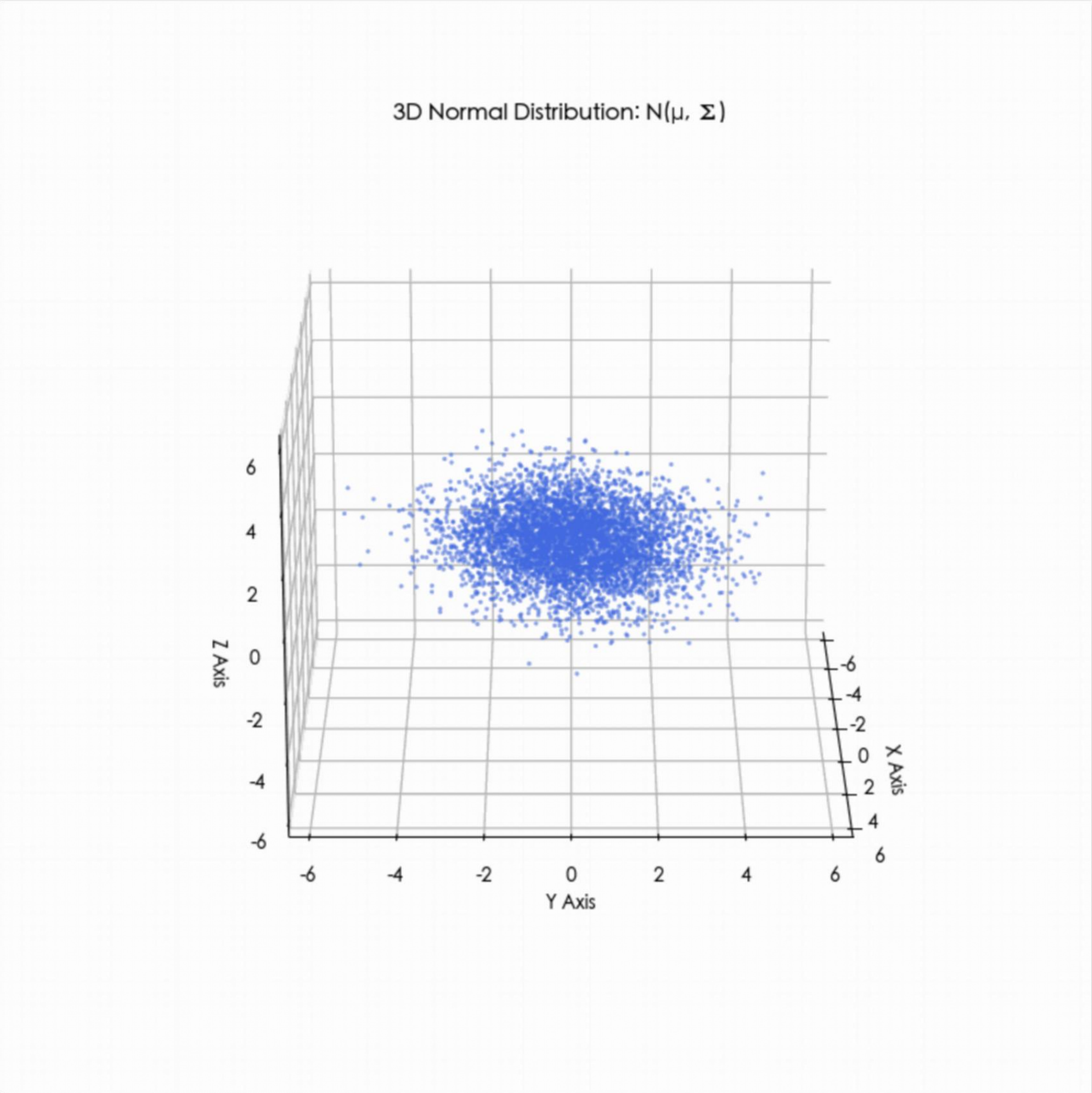
method 1: generate a rotating animation GIF



```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.animation import FuncAnimation
import matplotlib

# 设置渲染后端
matplotlib.use('Agg')

# 生成数据
np.random.seed(42)
n_samples = 5000
mu = np.array([0, 0, 0])
Sigma = np.array([
    [3.0, 1.0, 0.5],
    [1.0, 2.0, 0.3],
    [0.5, 0.3, 1.0]
])

z = np.random.randn(3, n_samples)
L = np.linalg.cholesky(Sigma)
x = mu[:, np.newaxis] + L @ z

# 创建图形
fig = plt.figure(figsize=(8, 8))
ax = fig.add_subplot(111, projection='3d')

# 绘制散点图
scatter = ax.scatter(x[0], x[1], x[2], c='royalblue', s=5, alpha=0.6, edgecolors='none')

# 设置坐标轴标签和标题
ax.set_xlabel('X Axis')
ax.set_ylabel('Y Axis')
ax.set_zlabel('Z Axis')
ax.set_title('3D Normal Distribution: N(μ, Σ)')
```

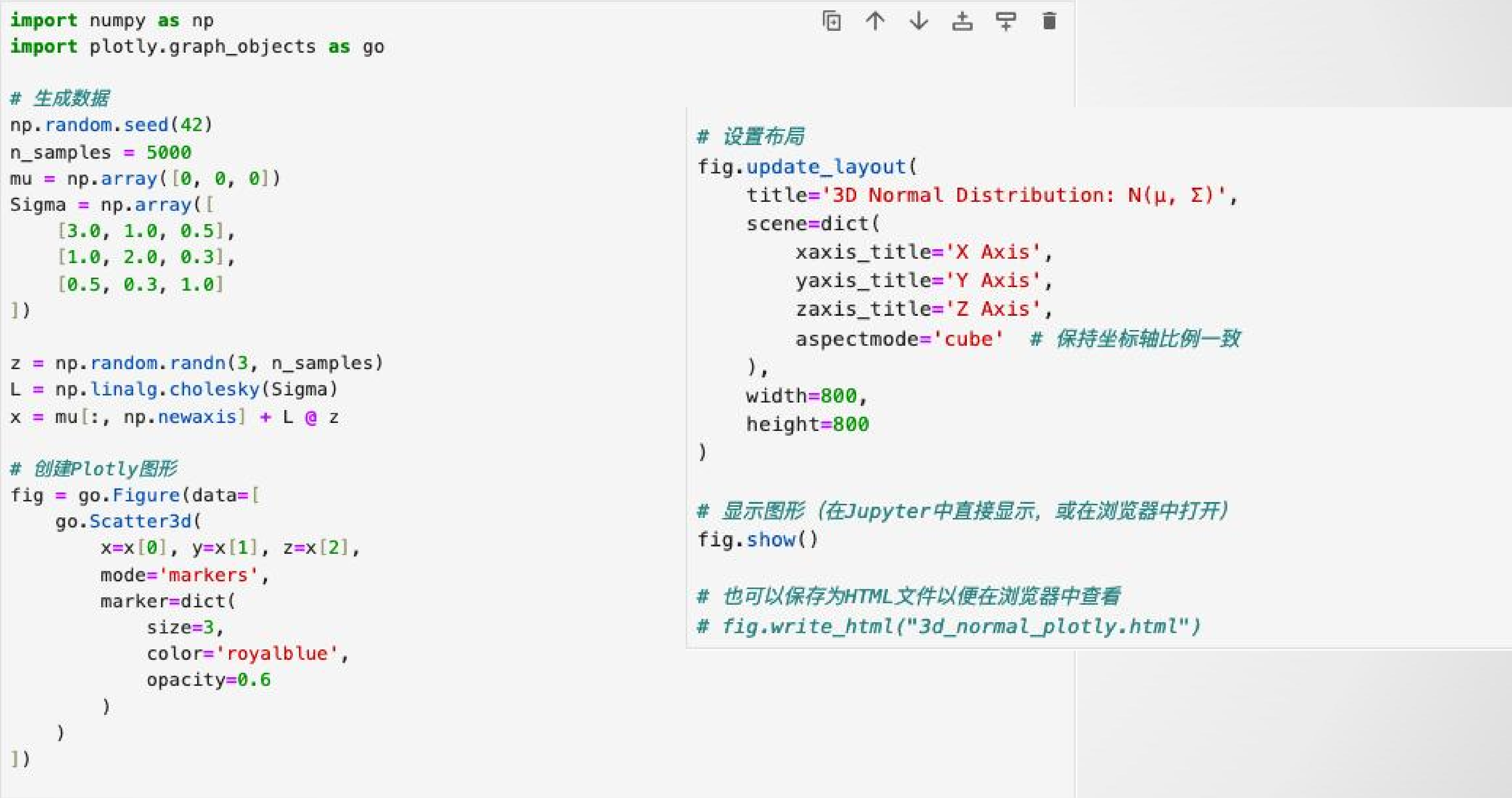
```python
# 调整坐标轴范围
max_range = np.max([np.ptp(x[0]), np.ptp(x[1]), np.ptp(x[2])]) / 2
mean_vals = np.mean(x, axis=1)
ax.set_xlim(mean_vals[0] - max_range, mean_vals[0] + max_range)
ax.set_ylim(mean_vals[1] - max_range, mean_vals[1] + max_range)
ax.set_zlim(mean_vals[2] - max_range, mean_vals[2] + max_range)

# 初始视角
ax.view_init(elev=20, azim=0)

# 定义更新函数，用于动画
def update(frame):
    ax.view_init(elev=20, azim=frame)
    return scatter,

# 创建动画
ani = FuncAnimation(fig, update, frames=np.linspace(0, 360, 100),
                    interval=100, blit=True)

# 保存为GIF
ani.save('3d_normal_distribution.gif', writer='pillow', fps=10, dpi=100)

# 关闭图形
plt.close()

print("动画已保存为 3d_normal_distribution.gif")
```

A rotating plot for N(μ, Σ)



1. Code for creating a N(μ, Σ) and its rotating plot

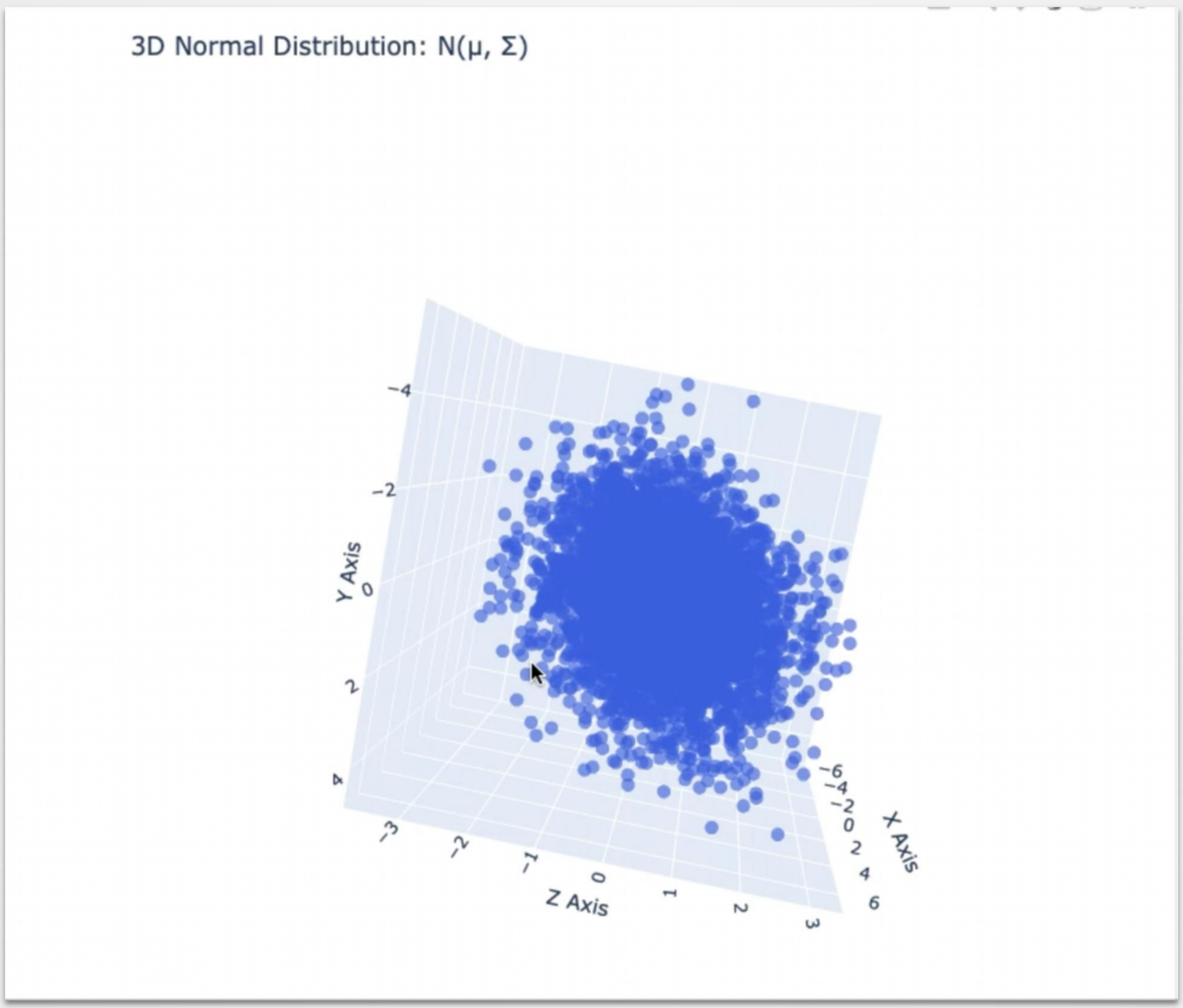
method 2: implementing interactive 3D graphs using Plotly



```python
import numpy as np
import plotly.graph_objects as go

# 生成数据
np.random.seed(42)
n_samples = 5000
mu = np.array([0, 0, 0])
Sigma = np.array([
    [3.0, 1.0, 0.5],
    [1.0, 2.0, 0.3],
    [0.5, 0.3, 1.0]
])

z = np.random.randn(3, n_samples)
L = np.linalg.cholesky(Sigma)
x = mu[:, np.newaxis] + L @ z

# 创建Plotly图形
fig = go.Figure(data=[
    go.Scatter3d(
        x=x[0], y=x[1], z=x[2],
        mode='markers',
        marker=dict(
            size=3,
            color='royalblue',
            opacity=0.6
        )
    )
])
```

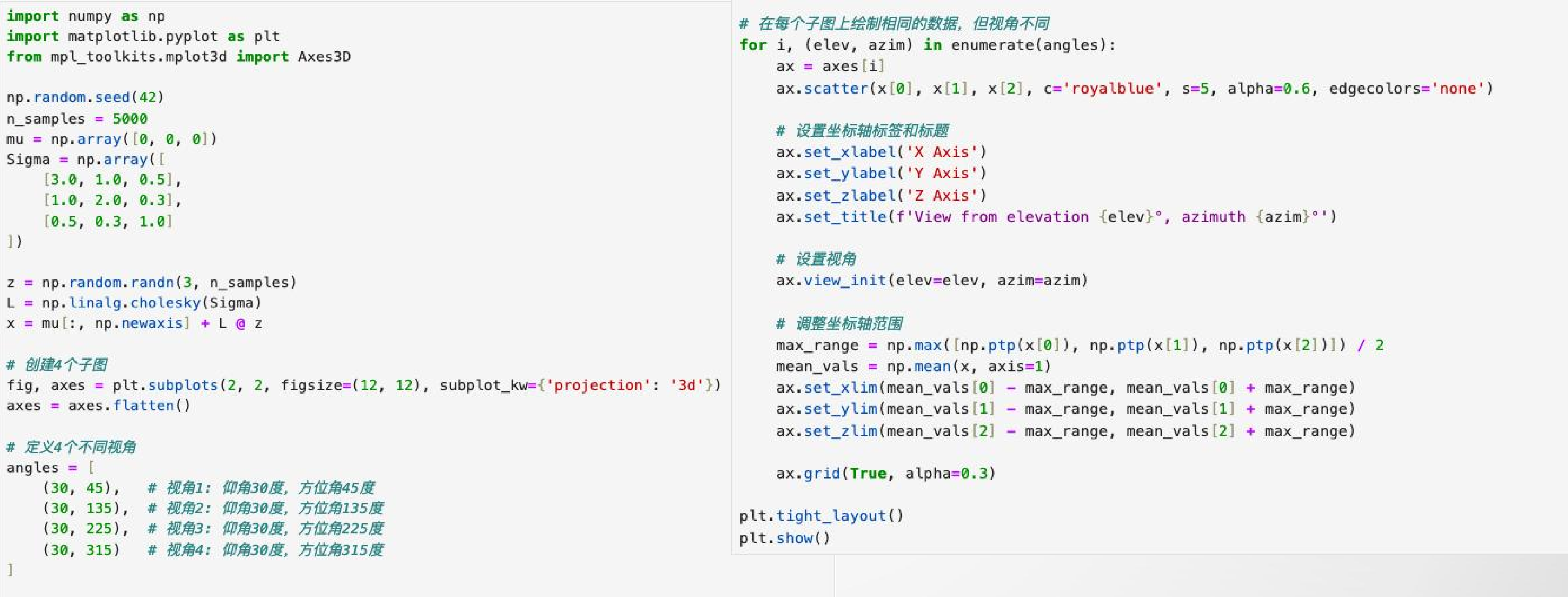
```python
# 设置布局
fig.update_layout(
    title='3D Normal Distribution: N(μ, Σ)',
    scene=dict(
        xaxis_title='X Axis',
        yaxis_title='Y Axis',
        zaxis_title='Z Axis',
        aspectmode='cube'  # 保持坐标轴比例一致
    ),
    width=800,
    height=800
)

# 显示图形（在Jupyter中直接显示，或在浏览器中打开）
fig.show()

# 也可以保存为HTML文件以便在浏览器中查看
# fig.write_html("3d_normal_plotly.html")
```

A rotating plot for N(μ, Σ)



1. Code for creating a N(μ, Σ) and its rotating plot

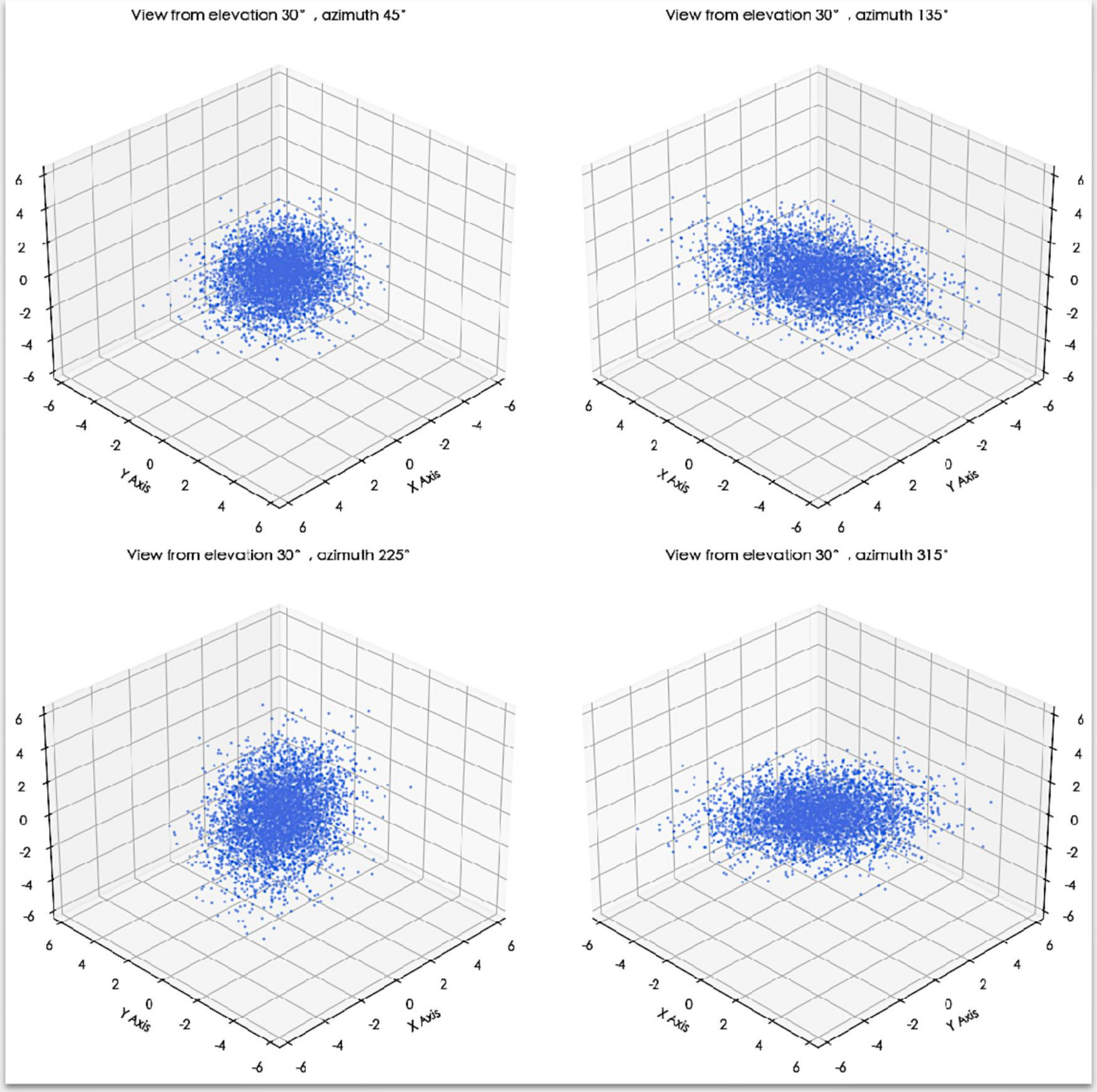
method 3: generate static multi-angle views



```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

np.random.seed(42)
n_samples = 5000
mu = np.array([0, 0, 0])
Sigma = np.array([
    [3.0, 1.0, 0.5],
    [1.0, 2.0, 0.3],
    [0.5, 0.3, 1.0]
])

z = np.random.randn(3, n_samples)
L = np.linalg.cholesky(Sigma)
x = mu[:, np.newaxis] + L @ z

# 创建4个子图
fig, axes = plt.subplots(2, 2, figsize=(12, 12), subplot_kw={'projection': '3d'})
axes = axes.flatten()

# 定义4个不同视角
angles = [
    (30, 45),   # 视角1: 仰角30度, 方位角45度
    (30, 135),  # 视角2: 仰角30度, 方位角135度
    (30, 225),  # 视角3: 仰角30度, 方位角225度
    (30, 315)   # 视角4: 仰角30度, 方位角315度
]
```

```python
# 在每个子图上绘制相同的数据, 但视角不同
for i, (elev, azim) in enumerate(angles):
    ax = axes[i]
    ax.scatter(x[0], x[1], x[2], c='royalblue', s=5, alpha=0.6, edgecolors='none')

    # 设置坐标轴标签和标题
    ax.set_xlabel('X Axis')
    ax.set_ylabel('Y Axis')
    ax.set_zlabel('Z Axis')
    ax.set_title(f'View from elevation {elev}°, azimuth {azim}°')

    # 设置视角
    ax.view_init(elev=elev, azim=azim)

    # 调整坐标轴范围
    max_range = np.max([np.ptp(x[0]), np.ptp(x[1]), np.ptp(x[2])]) / 2
    mean_vals = np.mean(x, axis=1)
    ax.set_xlim(mean_vals[0] - max_range, mean_vals[0] + max_range)
    ax.set_ylim(mean_vals[1] - max_range, mean_vals[1] + max_range)
    ax.set_zlim(mean_vals[2] - max_range, mean_vals[2] + max_range)

    ax.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

A rotating plot for N(μ, Σ)



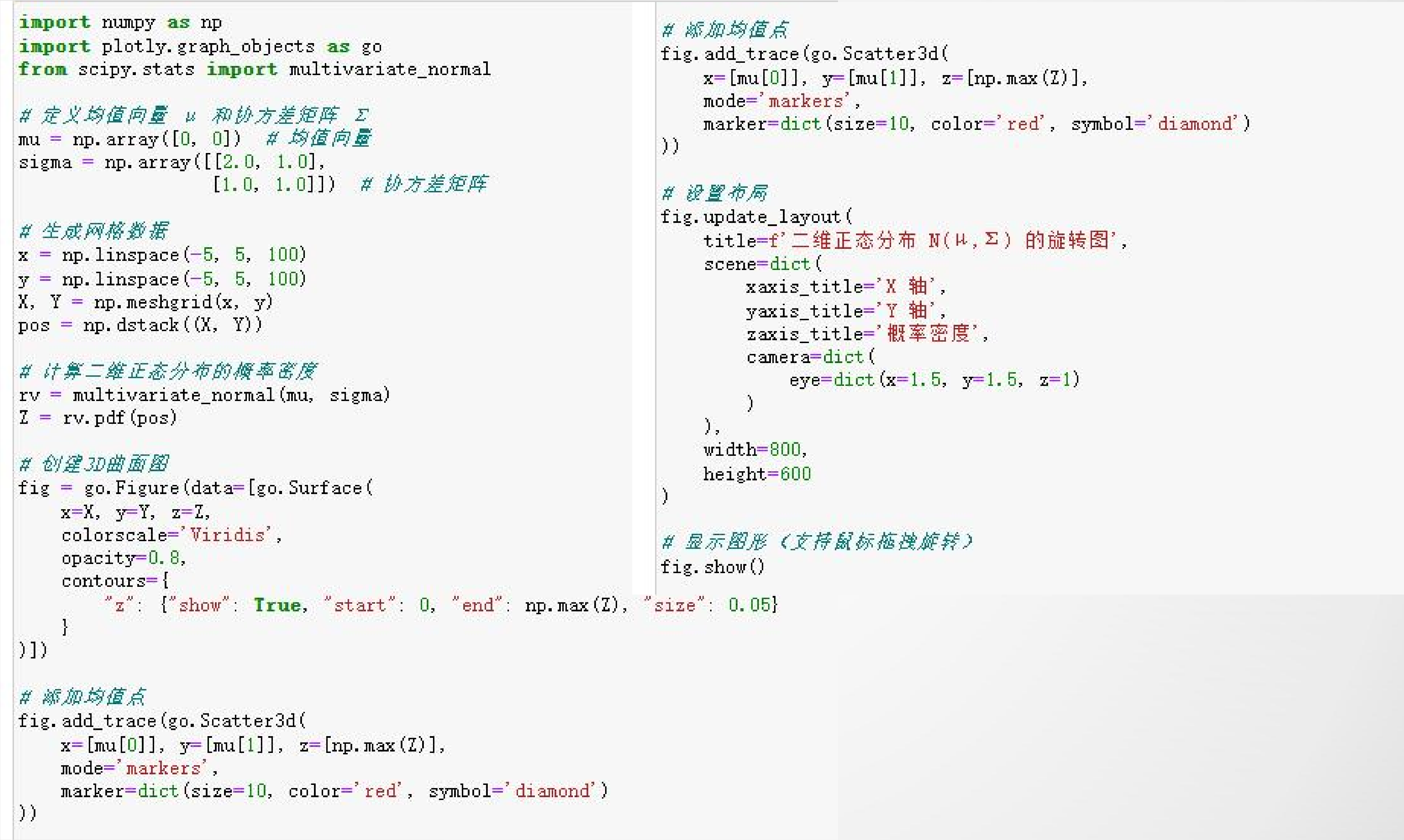
View from elevation 30° , azimuth 45°

View from elevation 30° , azimuth 135°

View from elevation 30° , azimuth 225°

View from elevation 30° , azimuth 315°

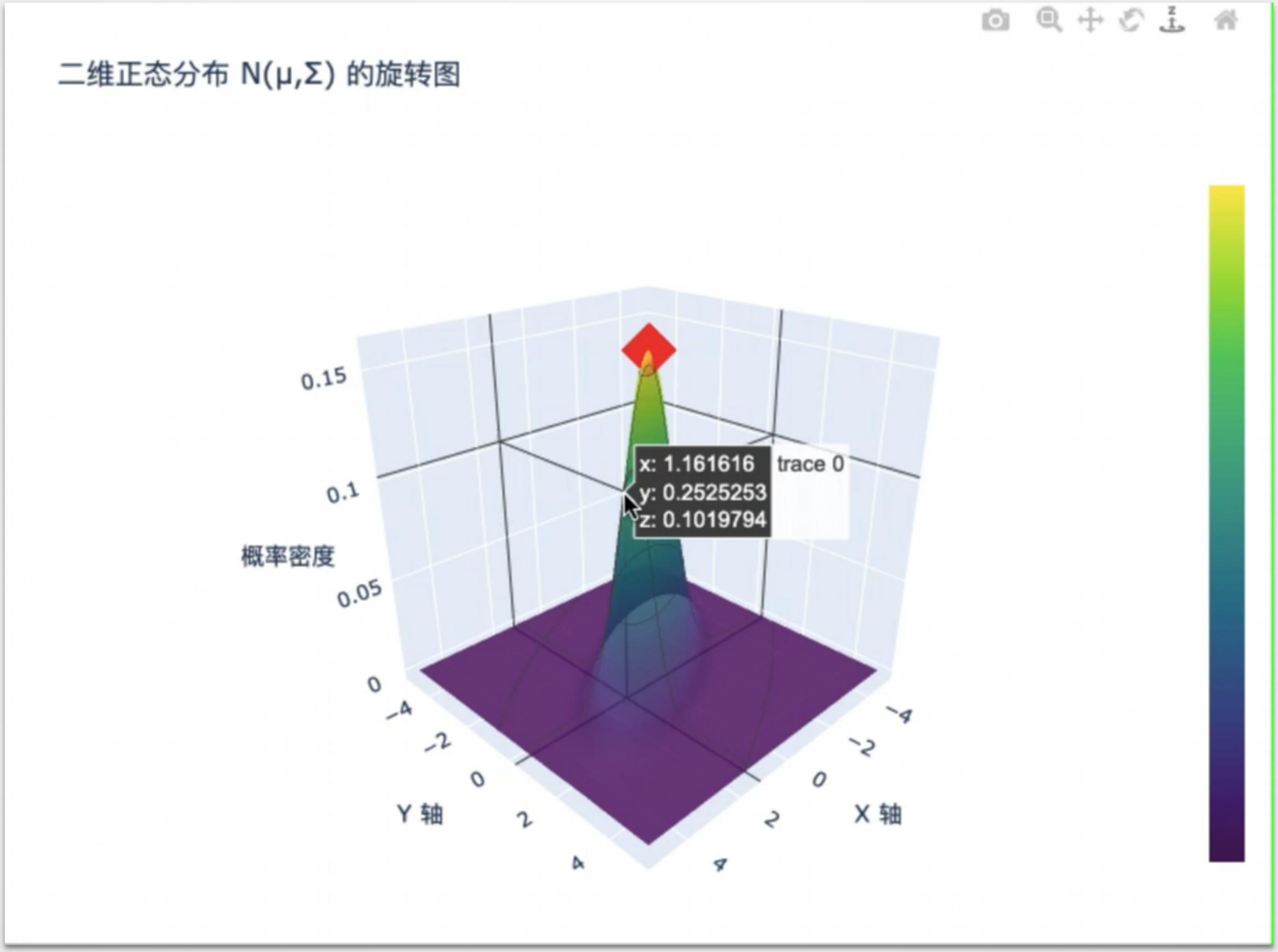
Code for creating a N(μ, Σ) and its PDF rotating plot



```python
import numpy as np
import plotly.graph_objects as go
from scipy.stats import multivariate_normal

# 定义均值向量 μ 和协方差矩阵 Σ
mu = np.array([0, 0])  # 均值向量
sigma = np.array([[2.0, 1.0],
                  [1.0, 1.0]])  # 协方差矩阵

# 生成网格数据
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(x, y)
pos = np.dstack((X, Y))

# 计算二维正态分布的概率密度
rv = multivariate_normal(mu, sigma)
Z = rv.pdf(pos)

# 创建3D曲面图
fig = go.Figure(data=[go.Surface(
    x=X, y=Y, z=Z,
    colorscale='Viridis',
    opacity=0.8,
    contours={
        "z": {"show": True, "start": 0, "end": np.max(Z), "size": 0.05}
    }
)])

# 添加均值点
fig.add_trace(go.Scatter3d(
    x=[mu[0]], y=[mu[1]], z=[np.max(Z)],
    mode='markers',
    marker=dict(size=10, color='red', symbol='diamond')
))
```

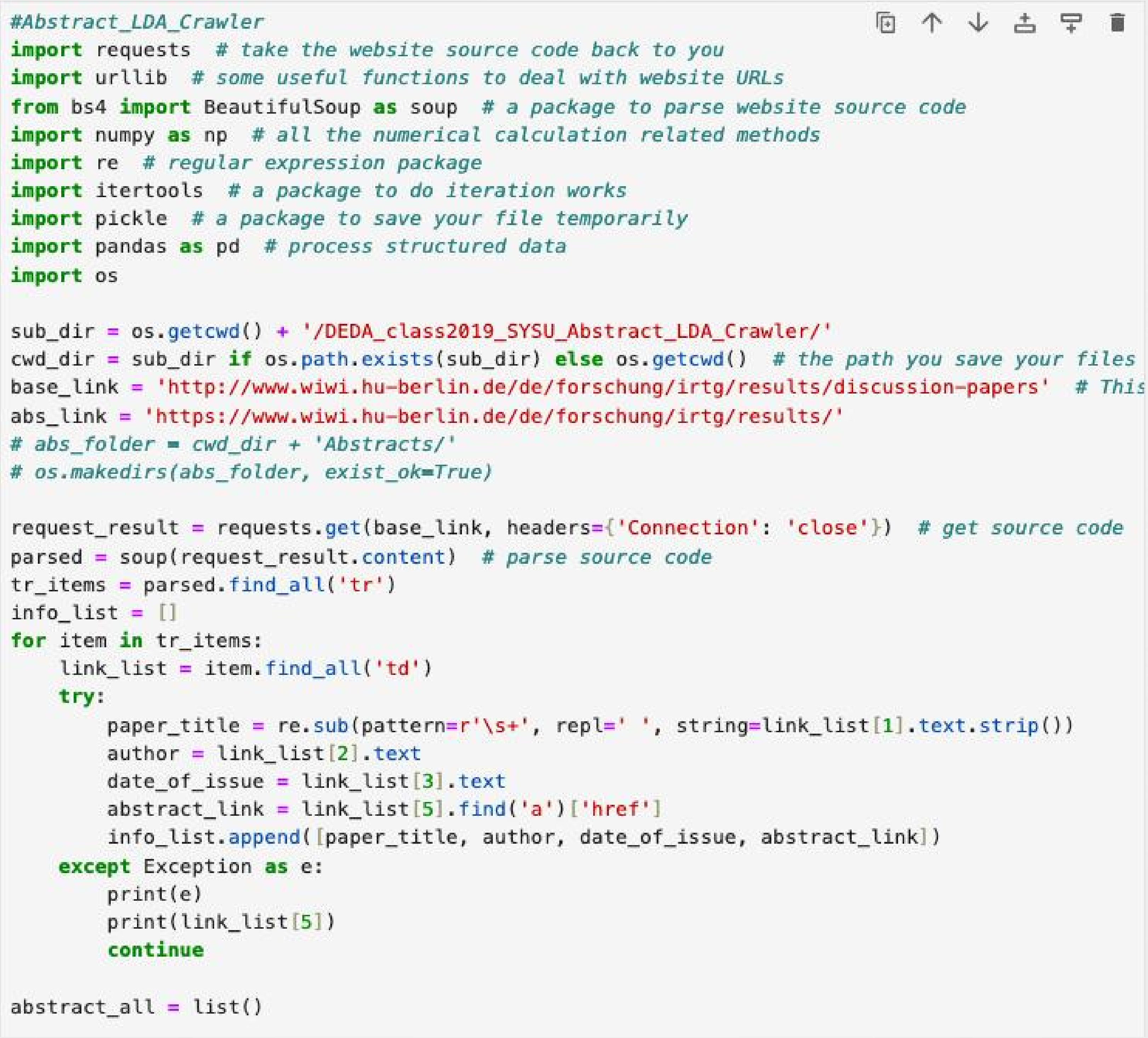
```python
# 添加均值点
fig.add_trace(go.Scatter3d(
    x=[mu[0]], y=[mu[1]], z=[np.max(Z)],
    mode='markers',
    marker=dict(size=10, color='red', symbol='diamond')
))

# 设置布局
fig.update_layout(
    title=f'二维正态分布 N(μ,Σ) 的旋转图',
    scene=dict(
        xaxis_title='X 轴',
        yaxis_title='Y 轴',
        zaxis_title='概率密度',
        camera=dict(
            eye=dict(x=1.5, y=1.5, z=1)
        )
    ),
    width=800,
    height=600
)

# 显示图形（支持鼠标拖拽旋转）
fig.show()
```

A Possibility Density Function plot for N(μ, Σ)



2. the LDA analysis example IRTG1792

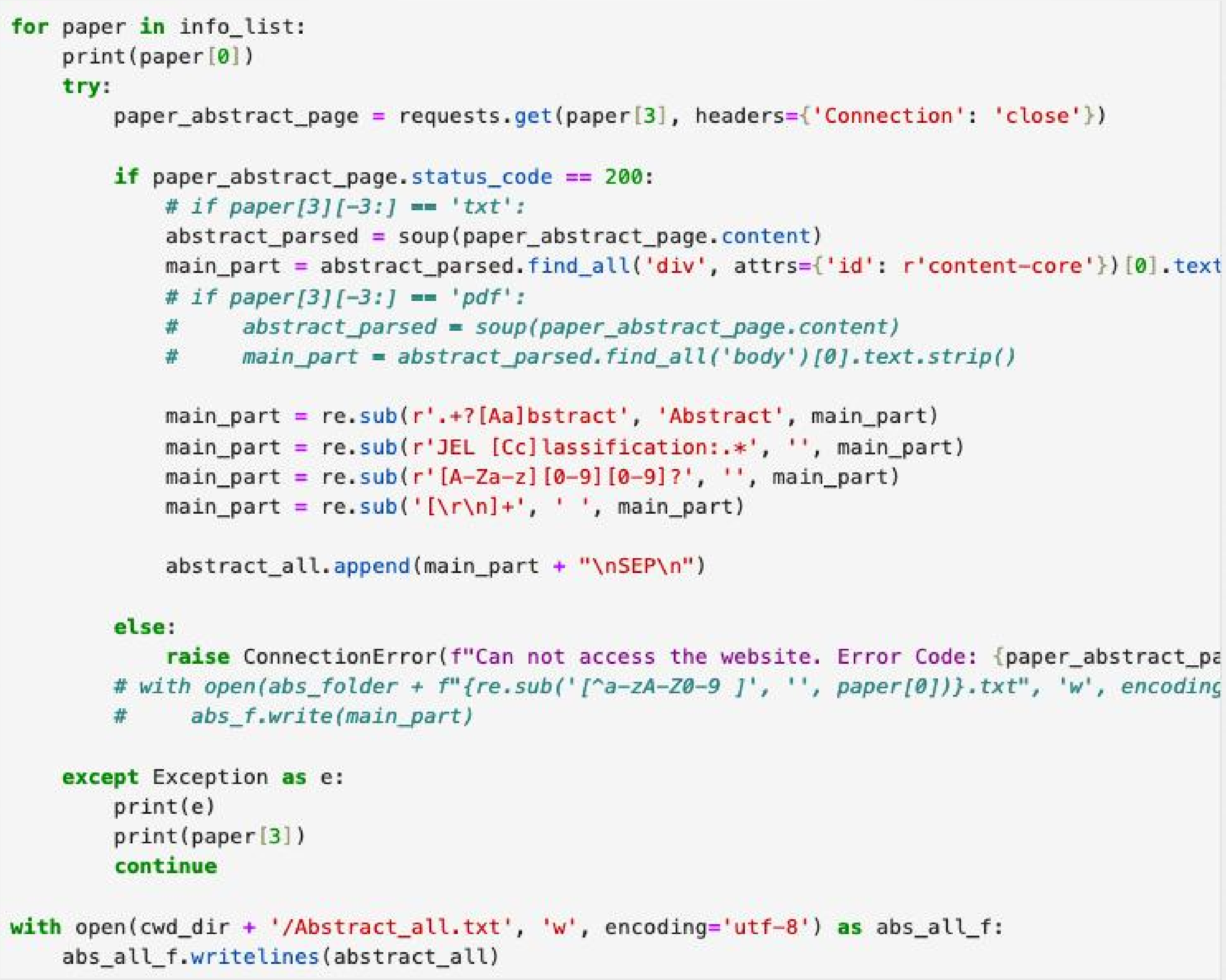
Step1: web scraping



```python
#Abstract_LDA_Crawler
import requests  # take the website source code back to you
import urllib  # some useful functions to deal with website URLs
from bs4 import BeautifulSoup as soup  # a package to parse website source code
import numpy as np  # all the numerical calculation related methods
import re  # regular expression package
import itertools  # a package to do iteration works
import pickle  # a package to save your file temporarily
import pandas as pd  # process structured data
import os

sub_dir = os.getcwd() + '/DEDA_class2019_SYSU_Abstract_LDA_Crawler/'
cwd_dir = sub_dir if os.path.exists(sub_dir) else os.getcwd()  # the path you save your files
base_link = 'http://www.wiwi.hu-berlin.de/de/forschung/irtg/results/discussion-papers'  # This
abs_link = 'https://www.wiwi.hu-berlin.de/de/forschung/irtg/results/'
# abs_folder = cwd_dir + 'Abstracts/'
# os.makedirs(abs_folder, exist_ok=True)

request_result = requests.get(base_link, headers={'Connection': 'close'})  # get source code
parsed = soup(request_result.content)  # parse source code
tr_items = parsed.find_all('tr')
info_list = []
for item in tr_items:
    link_list = item.find_all('td')
    try:
        paper_title = re.sub(pattern=r'\s+', repl=' ', string=link_list[1].text.strip())
        author = link_list[2].text
        date_of_issue = link_list[3].text
        abstract_link = link_list[5].find('a')['href']
        info_list.append([paper_title, author, date_of_issue, abstract_link])
    except Exception as e:
        print(e)
        print(link_list[5])
        continue

abstract_all = list()
```

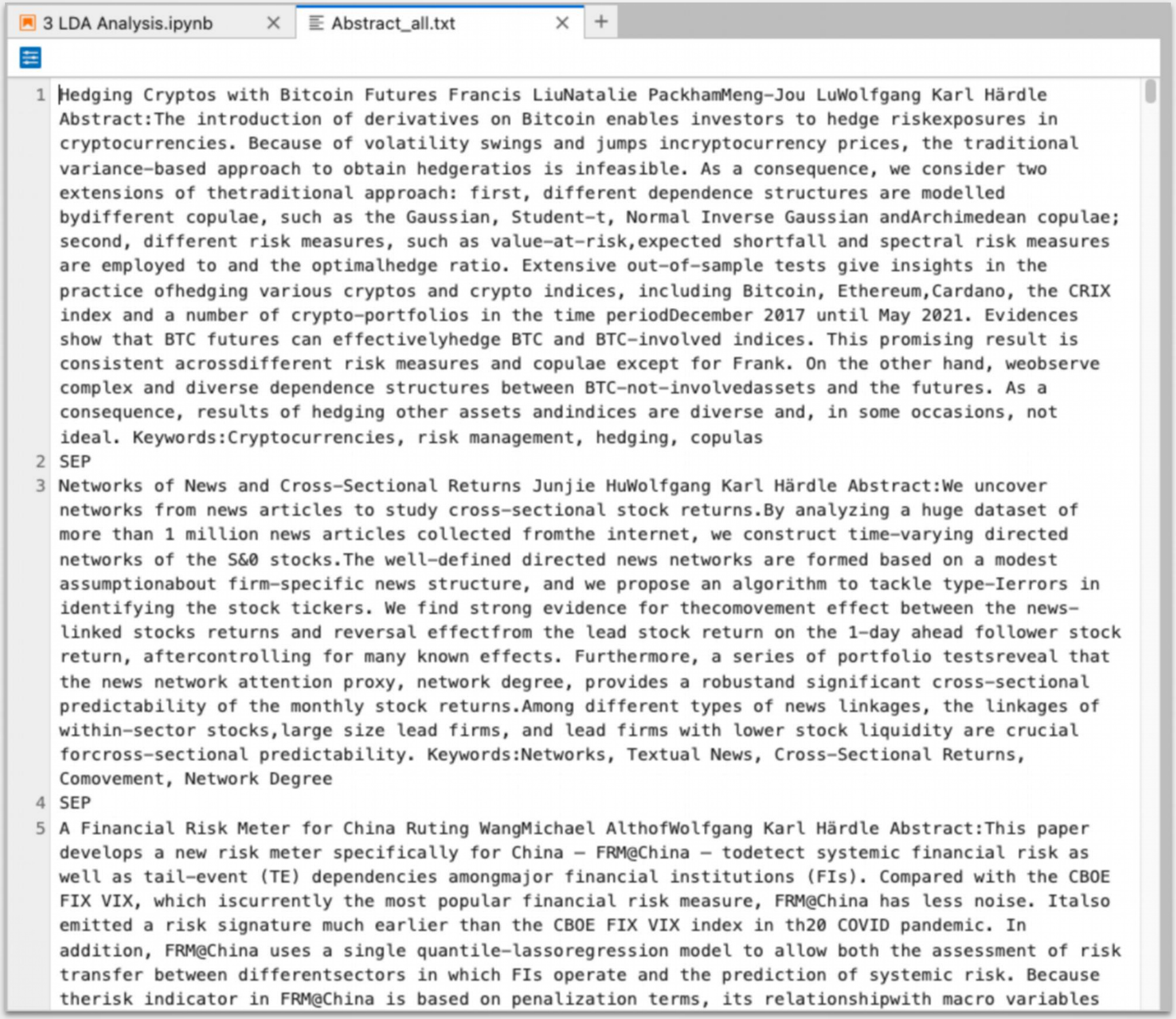
2. the LDA analysis example IRTG1792

Step1: web scraping



```
for paper in info_list:
    print(paper[0])
    try:
        paper_abstract_page = requests.get(paper[3], headers={'Connection': 'close'})

        if paper_abstract_page.status_code == 200:
            # if paper[3][-3:] == 'txt':
            abstract_parsed = soup(paper_abstract_page.content)
            main_part = abstract_parsed.find_all('div', attrs={'id': r'content-core'})[0].text
            # if paper[3][-3:] == 'pdf':
            #     abstract_parsed = soup(paper_abstract_page.content)
            #     main_part = abstract_parsed.find_all('body')[0].text.strip()

            main_part = re.sub(r'.+?[Aa]bstract', 'Abstract', main_part)
            main_part = re.sub(r'JEL [Cc]lassification:.*', '', main_part)
            main_part = re.sub(r'[A-Za-z][0-9][0-9]?', '', main_part)
            main_part = re.sub('[\r\n]+', ' ', main_part)

            abstract_all.append(main_part + "\nSEP\n")

        else:
            raise ConnectionError(f"Can not access the website. Error Code: {paper_abstract_pa
        # with open(abs_folder + f"{re.sub('[^a-zA-Z0-9 ]', '', paper[0])}.txt", 'w', encoding
        #     abs_f.write(main_part)

    except Exception as e:
        print(e)
        print(paper[3])
        continue

with open(cwd_dir + '/Abstract_all.txt', 'w', encoding='utf-8') as abs_all_f:
    abs_all_f.writelines(abstract_all)
```

2. the LDA analysis example IRTG1792



Hedging Cryptos with Bitcoin Futures Francis LiuNatalie PackhamMeng-Jou LuWolfgang Karl Härdle Abstract:The introduction of derivatives on Bitcoin enables investors to hedge riskexposures in cryptocurrencies. Because of volatility swings and jumps incryptocurrency prices, the traditional variance-based approach to obtain hedgeratios is infeasible. As a consequence, we consider two extensions of thetraditional approach: first, different dependence structures are modelled bydifferent copulae, such as the Gaussian, Student-t, Normal Inverse Gaussian andArchimedean copulae; second, different risk measures, such as value-at-risk,expected shortfall and spectral risk measures are employed to and the optimalhedge ratio. Extensive out-of-sample tests give insights in the practice ofhedging various cryptos and crypto indices, including Bitcoin, Ethereum,Cardano, the CRIX index and a number of crypto-portfolios in the time periodDecember 2017 until May 2021. Evidences show that BTC futures can effectivelyhedge BTC and BTC-involved indices. This promising result is consistent acrossdifferent risk measures and copulae except for Frank. On the other hand, weobserve complex and diverse dependence structures between BTC-not-involvedassets and the futures. As a consequence, results of hedging other assets andindices are diverse and, in some occasions, not ideal. Keywords:Cryptocurrencies, risk management, hedging, copulas

SEP

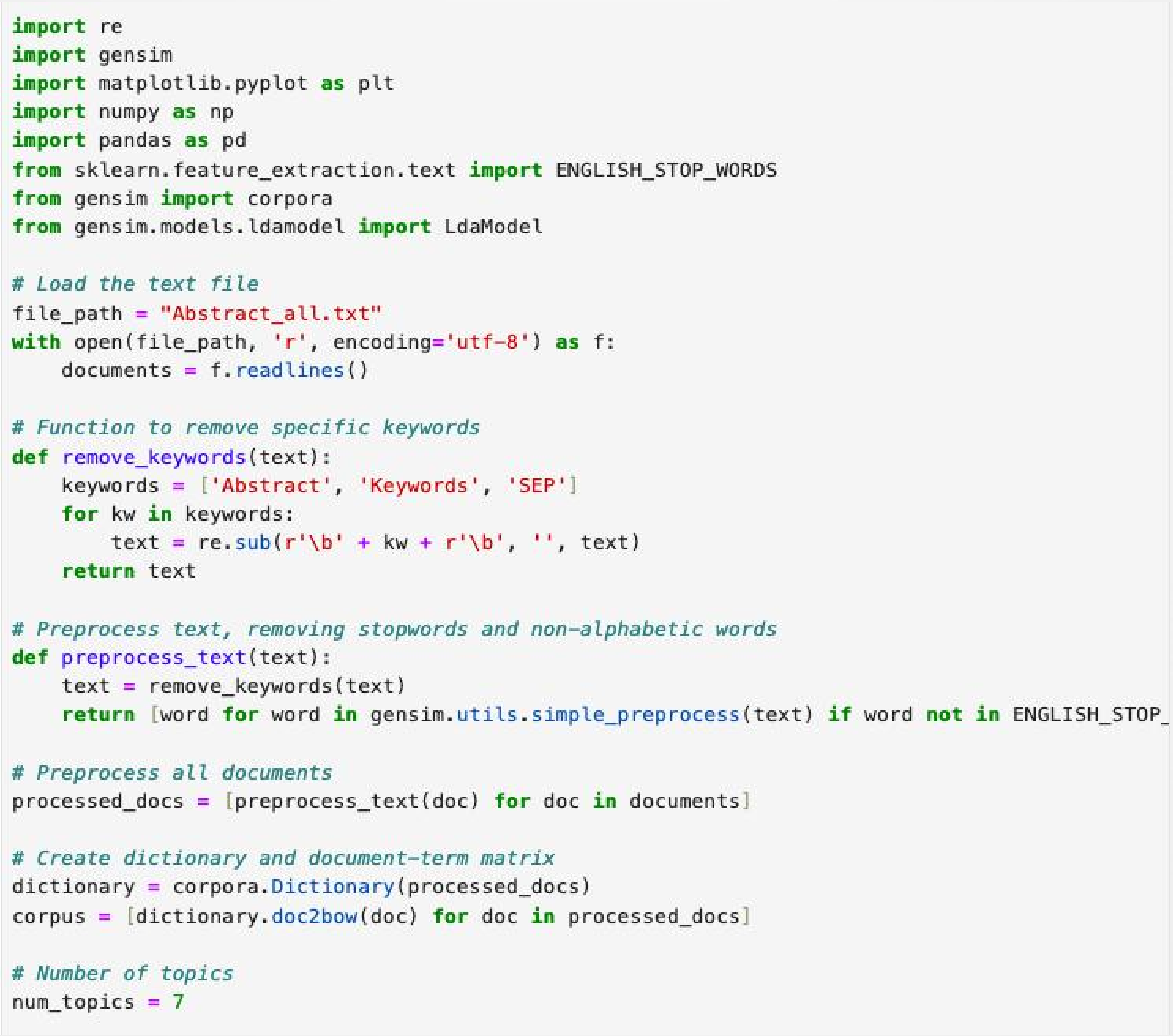
Networks of News and Cross-Sectional Returns Junjie HuWolfgang Karl Härdle Abstract:We uncover networks from news articles to study cross-sectional stock returns.By analyzing a huge dataset of more than 1 million news articles collected fromthe internet, we construct time-varying directed networks of the S&0 stocks.The well-defined directed news networks are formed based on a modest assumptionabout firm-specific news structure, and we propose an algorithm to tackle type-Ierrors in identifying the stock tickers. We find strong evidence for thecomovement effect between the news-linked stocks returns and reversal effectfrom the lead stock return on the 1-day ahead follower stock return, aftercontrolling for many known effects. Furthermore, a series of portfolio testsreveal that the news network attention proxy, network degree, provides a robustand significant cross-sectional predictability of the monthly stock returns.Among different types of news linkages, the linkages of within-sector stocks,large size lead firms, and lead firms with lower stock liquidity are crucial forcross-sectional predictability. Keywords:Networks, Textual News, Cross-Sectional Returns, Comovement, Network Degree

SEP

A Financial Risk Meter for China Ruting WangMichael AlthofWolfgang Karl Härdle Abstract:This paper develops a new risk meter specifically for China – FRM@China – todetect systemic financial risk as well as tail-event (TE) dependencies amongmajor financial institutions (FIs). Compared with the CBOE FIX VIX, which iscurrently the most popular financial risk measure, FRM@China has less noise. Italso emitted a risk signature much earlier than the CBOE FIX VIX index in th20 COVID pandemic. In addition, FRM@China uses a single quantile-lassoregression model to allow both the assessment of risk transfer between differentsectors in which FIs operate and the prediction of systemic risk. Because therisk indicator in FRM@China is based on penalization terms, its relationshipwith macro variables

2. the LDA analysis example IRTG1792

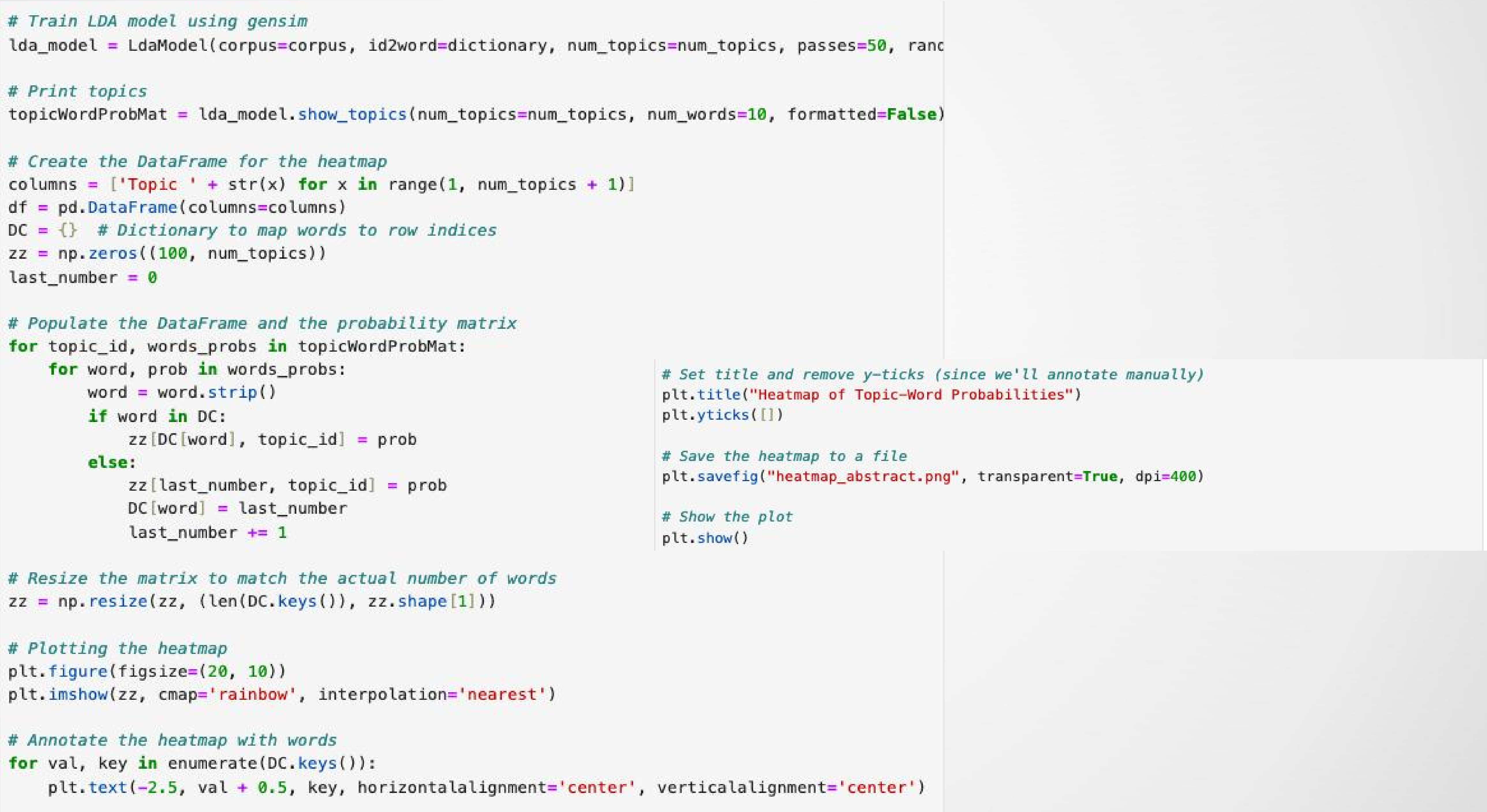
Step2: heat map to find correlations and topics



```
import re
import gensim
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import ENGLISH_STOP_WORDS
from gensim import corpora
from gensim.models.ldamodel import LdaModel

# Load the text file
file_path = "Abstract_all.txt"
with open(file_path, 'r', encoding='utf-8') as f:
    documents = f.readlines()

# Function to remove specific keywords
def remove_keywords(text):
    keywords = ['Abstract', 'Keywords', 'SEP']
    for kw in keywords:
        text = re.sub(r'\b' + kw + r'\b', '', text)
    return text

# Preprocess text, removing stopwords and non-alphabetic words
def preprocess_text(text):
    text = remove_keywords(text)
    return [word for word in gensim.utils.simple_preprocess(text) if word not in ENGLISH_STOP_

# Preprocess all documents
processed_docs = [preprocess_text(doc) for doc in documents]

# Create dictionary and document-term matrix
dictionary = corpora.Dictionary(processed_docs)
corpus = [dictionary.doc2bow(doc) for doc in processed_docs]

# Number of topics
num_topics = 7
```

2. the LDA analysis example IRTG1792

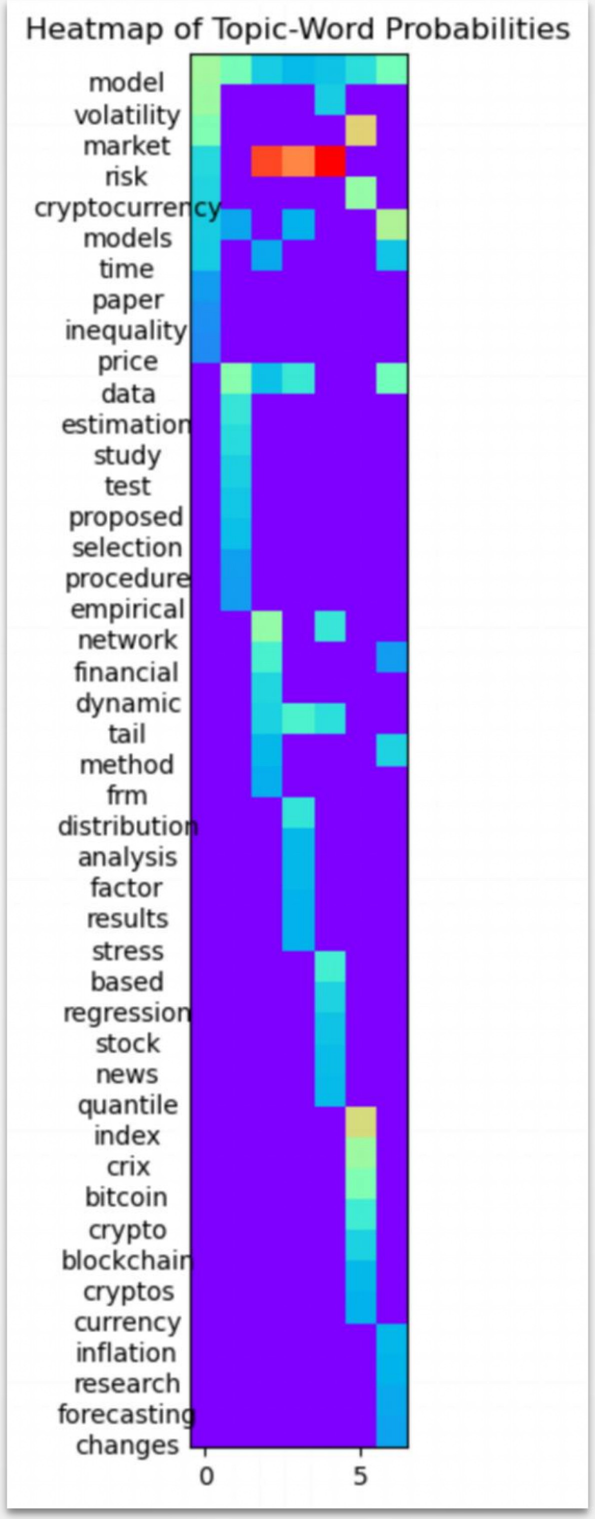
Step2: heat map to find correlations and topics



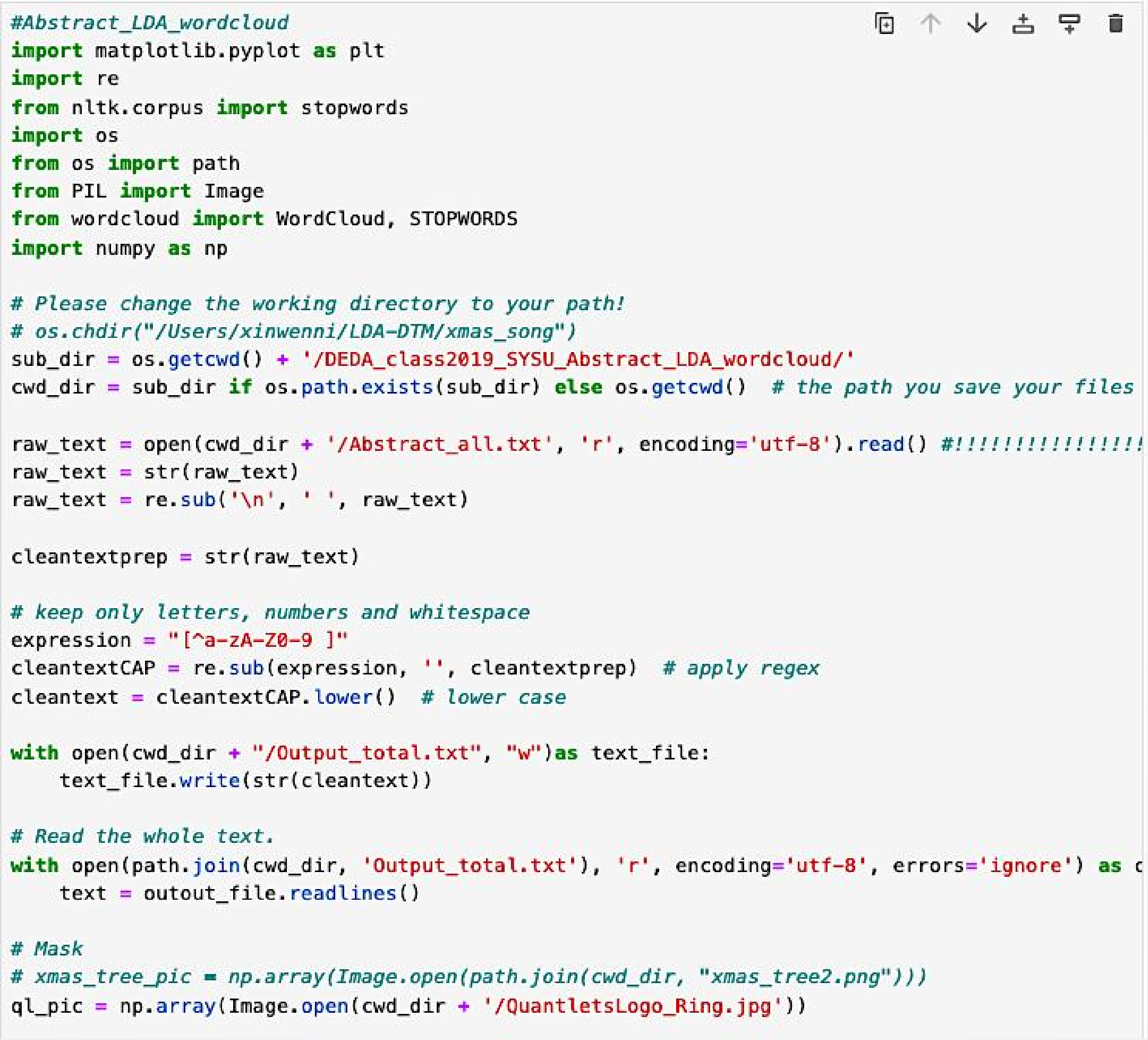
```python
# Train LDA model using gensim
lda_model = LdaModel(corpus=corpus, id2word=dictionary, num_topics=num_topics, passes=50, ranc

# Print topics
topicWordProbMat = lda_model.show_topics(num_topics=num_topics, num_words=10, formatted=False)

# Create the DataFrame for the heatmap
columns = ['Topic ' + str(x) for x in range(1, num_topics + 1)]
df = pd.DataFrame(columns=columns)
DC = {}  # Dictionary to map words to row indices
zz = np.zeros((100, num_topics))
last_number = 0

# Populate the DataFrame and the probability matrix
for topic_id, words_probs in topicWordProbMat:
    for word, prob in words_probs:
        word = word.strip()
        if word in DC:
            zz[DC[word], topic_id] = prob
        else:
            zz[last_number, topic_id] = prob
            DC[word] = last_number
            last_number += 1

# Resize the matrix to match the actual number of words
zz = np.resize(zz, (len(DC.keys()), zz.shape[1]))

# Plotting the heatmap
plt.figure(figsize=(20, 10))
plt.imshow(zz, cmap='rainbow', interpolation='nearest')

# Annotate the heatmap with words
for val, key in enumerate(DC.keys()):
    plt.text(-2.5, val + 0.5, key, horizontalalignment='center', verticalalignment='center')
```

```python
# Set title and remove y-ticks (since we'll annotate manually)
plt.title("Heatmap of Topic-Word Probabilities")
plt.yticks([])

# Save the heatmap to a file
plt.savefig("heatmap_abstract.png", transparent=True, dpi=400)

# Show the plot
plt.show()
```

2. the LDA analysis example IRTG1792

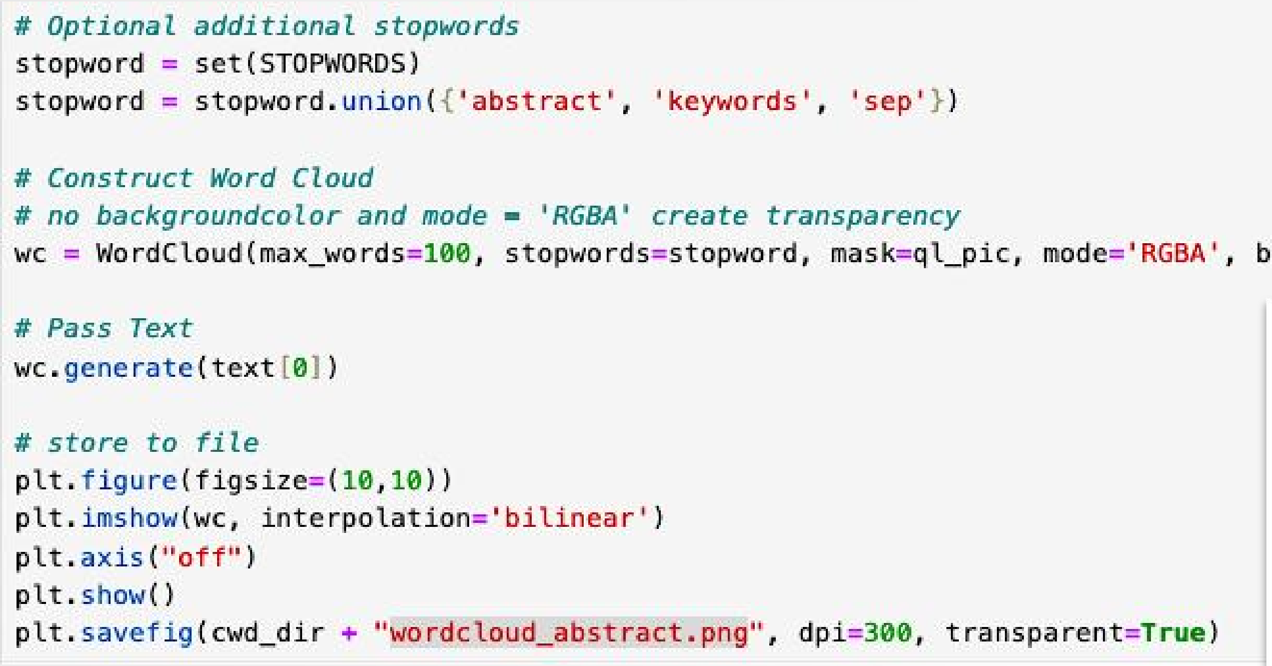


WordCloud of the LDA analysis example IRTG1792

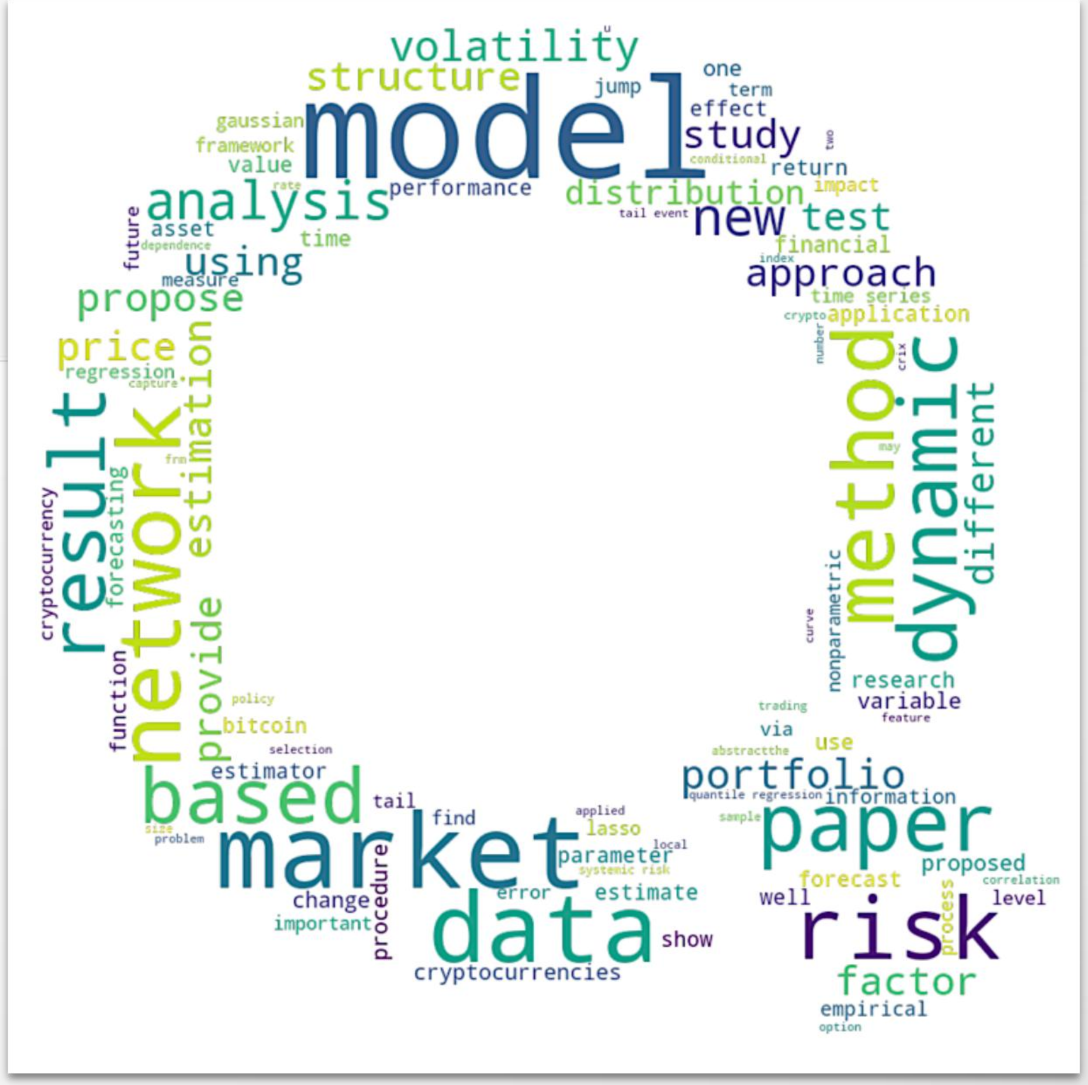


```
#Abstract_LDA_wordcloud
import matplotlib.pyplot as plt
import re
from nltk.corpus import stopwords
import os
from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS
import numpy as np

# Please change the working directory to your path!
# os.chdir("/Users/xinwenni/LDA-DTM/xmas_song")
sub_dir = os.getcwd() + '/DEDA_class2019_SYSU_Abstract_LDA_wordcloud/'
cwd_dir = sub_dir if os.path.exists(sub_dir) else os.getcwd()  # the path you save your files

raw_text = open(cwd_dir + '/Abstract_all.txt', 'r', encoding='utf-8').read() #!!!!!!!!!!!!!!!!
raw_text = str(raw_text)
raw_text = re.sub('\n', ' ', raw_text)

cleantextprep = str(raw_text)

# keep only letters, numbers and whitespace
expression = "[^a-zA-Z0-9 ]"
cleantextCAP = re.sub(expression, '', cleantextprep)  # apply regex
cleantext = cleantextCAP.lower()  # lower case

with open(cwd_dir + "/Output_total.txt", "w")as text_file:
    text_file.write(str(cleantext))

# Read the whole text.
with open(path.join(cwd_dir, 'Output_total.txt'), 'r', encoding='utf-8', errors='ignore') as c
    text = outout_file.readlines()

# Mask
# xmas_tree_pic = np.array(Image.open(path.join(cwd_dir, "xmas_tree2.png")))
ql_pic = np.array(Image.open(cwd_dir + '/QuantletsLogo_Ring.jpg'))
```

WordCloud of the LDA analysis example IRTG1792



```
# Optional additional stopwords
stopword = set(STOPWORDS)
stopword = stopword.union({'abstract', 'keywords', 'sep'})

# Construct Word Cloud
# no backgroundcolor and mode = 'RGBA' create transparency
wc = WordCloud(max_words=100, stopwords=stopword, mask=ql_pic, mode='RGBA', background_color=N

# Pass Text
wc.generate(text[0])

# store to file
plt.figure(figsize=(10,10))
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
plt.savefig(cwd_dir + "wordcloud_abstract.png", dpi=300, transparent=True)
```


volatility
structure
model
jump
one
term
effect
study
two
gaussian
framework
value
conditional
return
analysis
performance
distribution
impact
rate
tail event
new
test
future
asset
dependence
time
financial
using
index
measure
approach
propose
time series
crypto
application
price
regression
capture
estimation
result
network
forecasting
frm
cryptocurrency
number
crix
method
dynamic
different
may
curve
nonparametric
research
variable
feature
trading
provide
policy
function
bitcoin
selection
via
use
abstractthe
estimator
portfolio
quantile regression
information
based
tail
find
applied
lasso
sample
paper
size
problem
market
local
parameter
systemic risk
proposed
correlation
procedure
error
estimate
forecast
change
well
level
important
process
data
risk
show
cryptocurrencies
factor
empirical
option