注：作图所用的 Python 代码见文末附录

第 1 题：

（a）证明如下：

$$\because\ \bar{u} = F_X(X)$$

$$\therefore\ F_{\bar{u}}(u) = P(\bar{u} \le u) = P(F_X(X) \le u)$$

$$\because F_X(x)\text{严格递增}$$

$$\therefore\ P(F_X(X) \le u) = P\left[F_X^{-1}\left(F_X(X)\right) \le F_X^{-1}(u)\right] = P[X \le F_X^{-1}(u)] = F_X[F_X^{-1}(u)] = u$$

$$\therefore\ F_{\bar{u}}(u) = u\text{，这恰好是均匀分布的累积分布函数}$$

$$\because\ F_X(X)\text{的值域为}[0,1]$$

$$\therefore\ \bar{u}\text{ 服从}[0,1]\text{上的均匀分布}$$

$$\text{故有，}\bar{u} := F_X(X) \sim U[0,1]$$

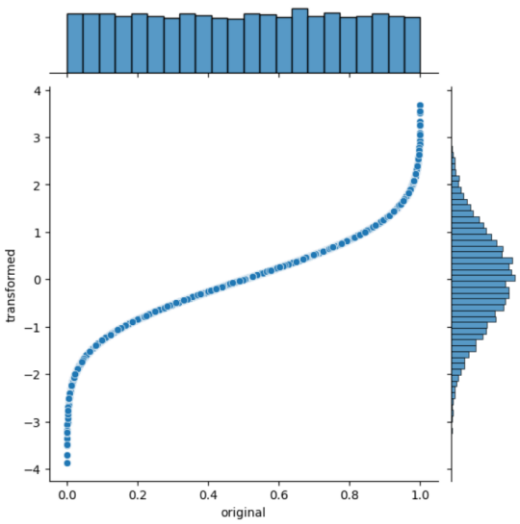
命题得证。

（b）证明如下：

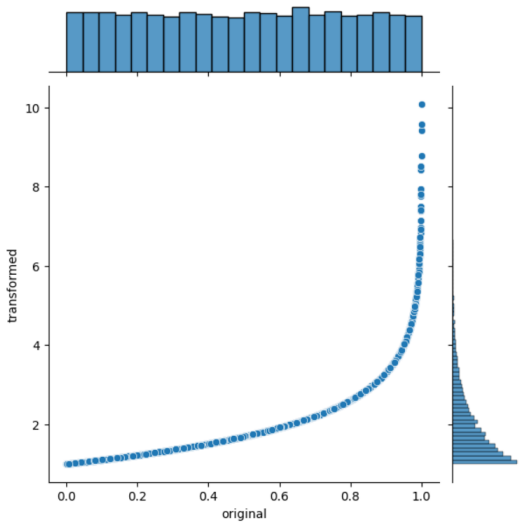
$$\because\ Q_X(\bar{u}) = F_X^{-1}(\bar{u}) = F_X^{-1}(F_X(X)) = X$$

$$\therefore X \sim Q_X(\bar{u})$$

（c）正态分布（均值为 0，方差为 1）的概率积分变换：



指数分布（$\lambda = 1$）的概率积分变换：



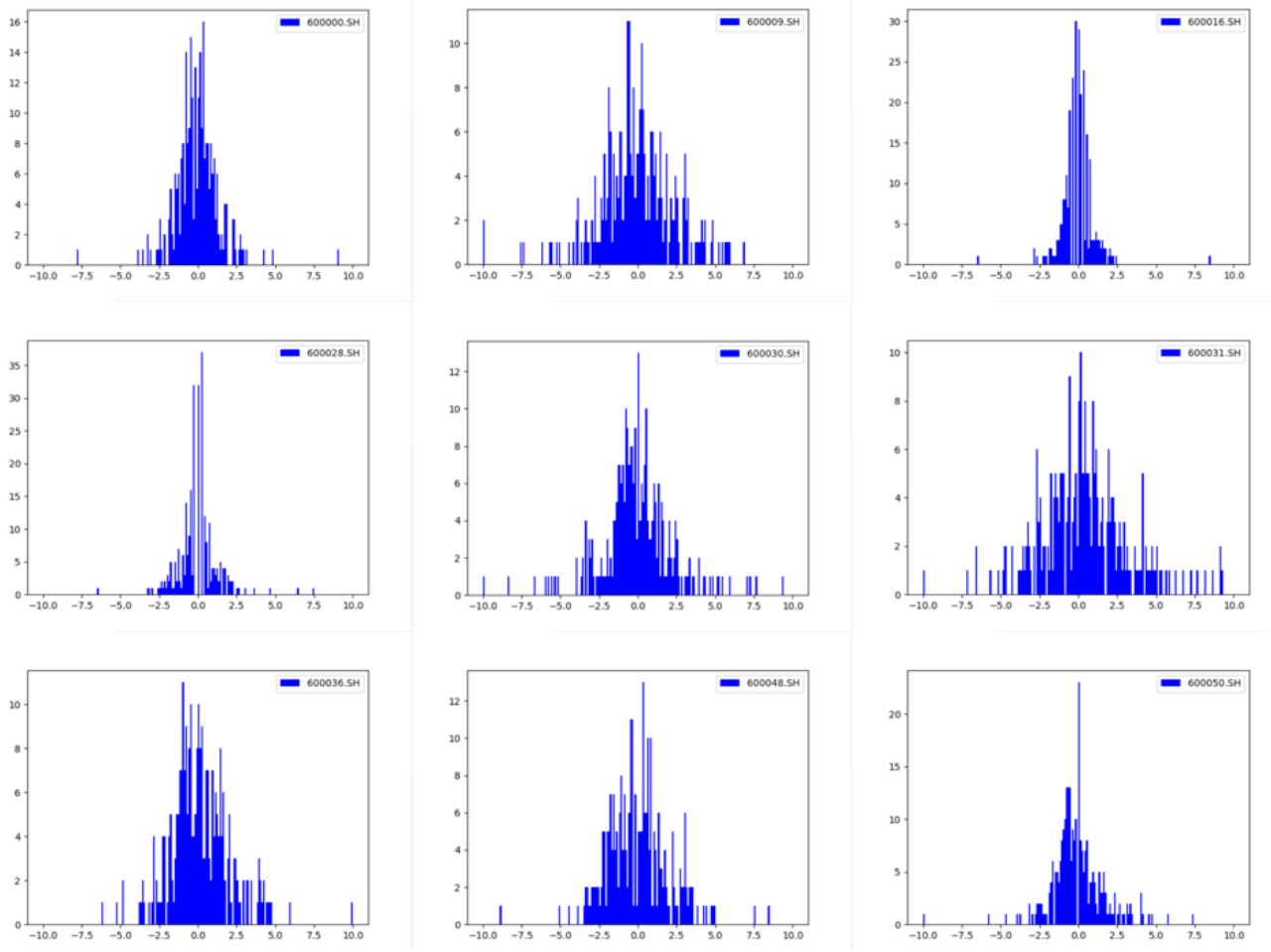
第 2 题：

（a）Location indices 包括均值、中位数和众数。考虑到股票的涨幅几乎不可能出现同样的值，因此直接计算其众数意义不大。我考虑将涨幅的取值范围$[-10\%, 10\%]$平均分为 20 个区间，也就是每 0.1%的涨幅为一个区间，然后统计各只股票的涨幅落在各个区间的频数，找出频数最大的区间作为众数的近似。各只股票涨幅的均值、中位数及众数的情况如下表所示：

名称	浦发银行	上海机场	民生银行	中国石化	中信证券	三一重工	招商银行	保利地产	中国联通	上汽集团
代码	600000.SH	600009.SH	600016.SH	600028.SH	600030.SH	600031.SH	600036.SH	600048.SH	600050.SH	600104.SH
均值	-0.0270	-0.0568	-0.0440	-0.0089	0.0396	0.3884	0.1348	0.0386	-0.0792	0.0118
中位数	-0.0958	-0.1193	0.0000	0.0000	-0.1371	0.2710	0.0000	-0.1229	-0.2058	-0.2419
众数	0.3~0.4	-0.6~-0.7	-0.1~-0.2	0.2~0.3	0~0.1	0.1~0.2	-0.9~-1	0.3~0.4	0~0.1	-0.3~0.4

若我是产品经理，我会选择招商银行（600036.SH）。我认为产品经理首先要避开经常出现较大跌幅的股票，保证投资组合稳健。因此，分布图像（见下图）在左侧出现厚尾的股票我不会选择，比如上海机场（600009.SH）、中信证券（600030.SH）和三一重工（600031.SH）。此外，产品经理要保证产品的收益率，因此我偏好平均收益率靠前的股票，招商银行的平均收益率排在第二位，表现较好（第一名为三一重工，由于左侧厚尾已被剔除）。招商银行的收益率中位数为 0，在十只股票中并列第 2（第一名同样为三一重工）。此外，从招商银行的收益率分布图来看，其呈现出右偏分布，也就是有可能出现较大涨幅，这是一个比较好的特征。

在向客户推荐招商银行这只股票时，我会向客户展示其收益率的均值，同时隐去众数的信息。首先，招商银行的收益率均值为正，比较有吸引力，应当展示；而收益率众数为负，为避免给客户留下不好印象，应当隐去。此外，对于投资者或是产品经理而言，重要的是一段时期内的回报率而非个别交易日的盈亏，因此收益率的均值本身就比众数更有说服力。



（b）如果我是客户，我会比较喜欢三一重工，因为它的平均日收益率最高。在选择股票时，我会比较看重收益率的均值，因为相比于中位数和众数，均值反映出的信息更充分。均值更能够体现一只股票在一段时期内的综合表现，与股票作为中长期投资方式的身份相吻合。我不太关注股价收益率的众数，原因如（a）题中的陈述类似。收益率作为连续变量，直接计算众数没有意义。如果按照（a）的思路，计算各收益率区间的出现次数，然后选择出现最多的区间作为众数的近似，也会带来问题。如果区间过宽，则最后的结果意义不大（比如区间[-10%，10%]必然包含所有的样本点）；如果区间过窄，那对数据量的要求会比较大，数据较少时，样本点会零散的出现在一些区间，使得各个区间的频数都很小且很接近。综上所述，众数通常不是一个好的选择。

第 3 题：

证明：

$$Z_X^2 = \frac{[X - Loc\{X\}]^2}{Dis\{X\}^2} = \frac{X^2 - 2 \cdot Loc\{X\} \cdot X + Loc\{X\}^2}{Dis\{X\}^2}$$

根据讲义中的式（1.22）和式（1.32）（Affine Equivalence），有：

$$Loc\{a + bX\} = a + bLoc\{X\}$$

$$Dis\{a + bX\} = |b| \cdot Dis\{X\}$$

故：

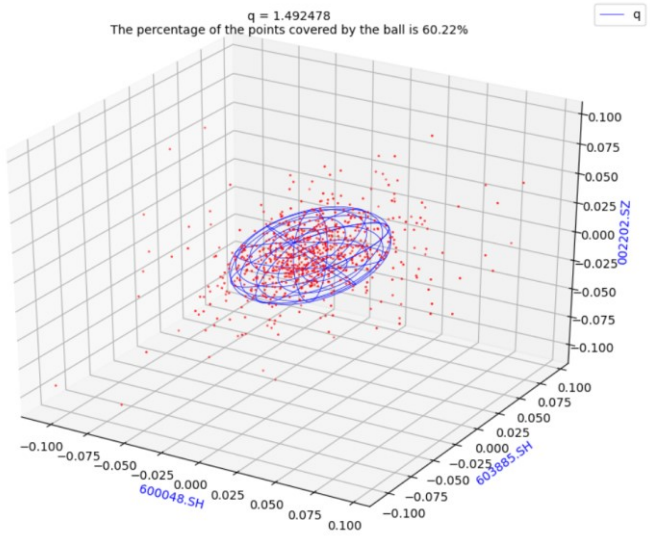
$$Z_{a+bX} = \frac{a + bX - Loc\{a + bX\}}{Dis\{a + bX\}} = \frac{a + bX - a - b \cdot Loc\{X\}}{|b| \cdot Dis\{X\}} = \frac{bX - b \cdot Loc\{X\}}{|b| \cdot Dis\{X\}}$$

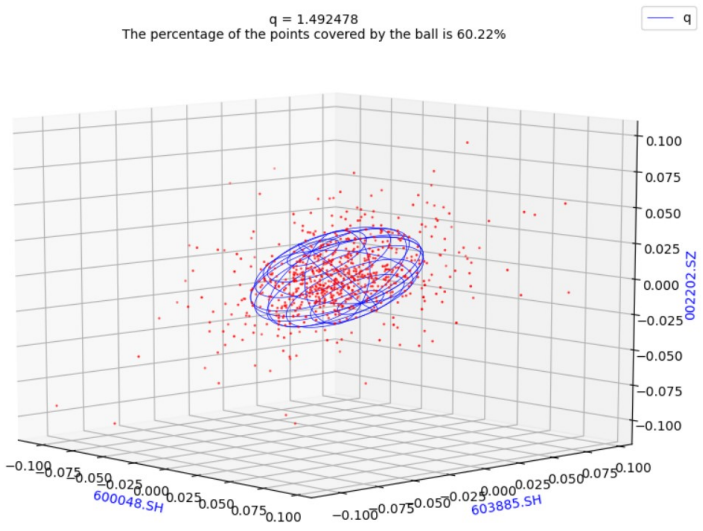
$$\therefore Z_{a+bX}^2 = \frac{b^2 \cdot [X - Loc\{X\}]^2}{b^2 \cdot Dis\{X\}^2} = Z_X^2$$

第 4 题：

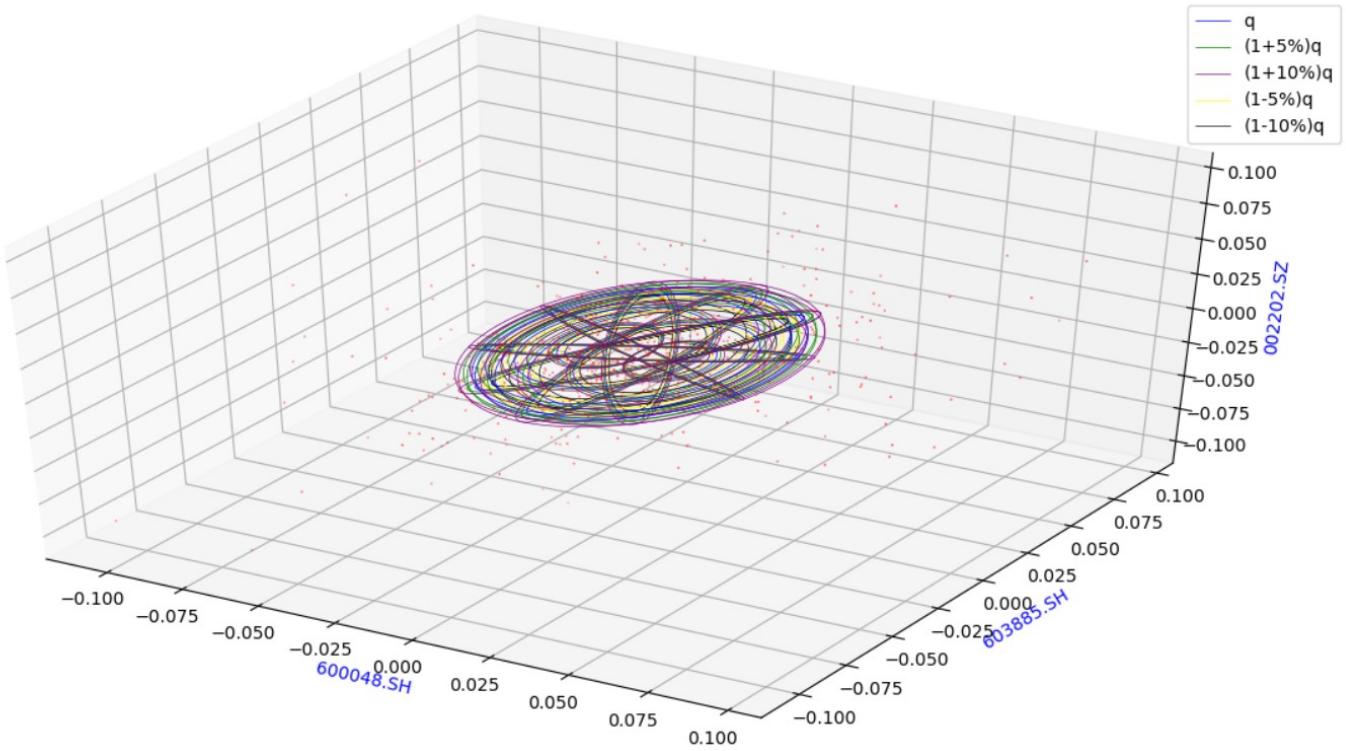
（a）我选择了吉祥航空（603885.SH）、金风科技（002202.SZ）和保利地产（600048.SH）三只股票。我选择对日收益率 $\ln\frac{P_t}{P_{t-1}}$ 而非股价本身进行分析。

（b）首先采用 20018 年 1 月 1 日-2020 年 12 月 31 日的数据进行分析。Location – Dispersion Ellipsoid 如下图所示。所有样本点距离均值点的马氏距离的均值为 $q = 1.492478$，有 60.22%的样本点被包含在 Ellipsoid 以内。





（c）q 取不同值时 Location – Dispersion Ellipsoid 如下图所示：（为突出 Ellipsoid，股价散点已做淡化处理）

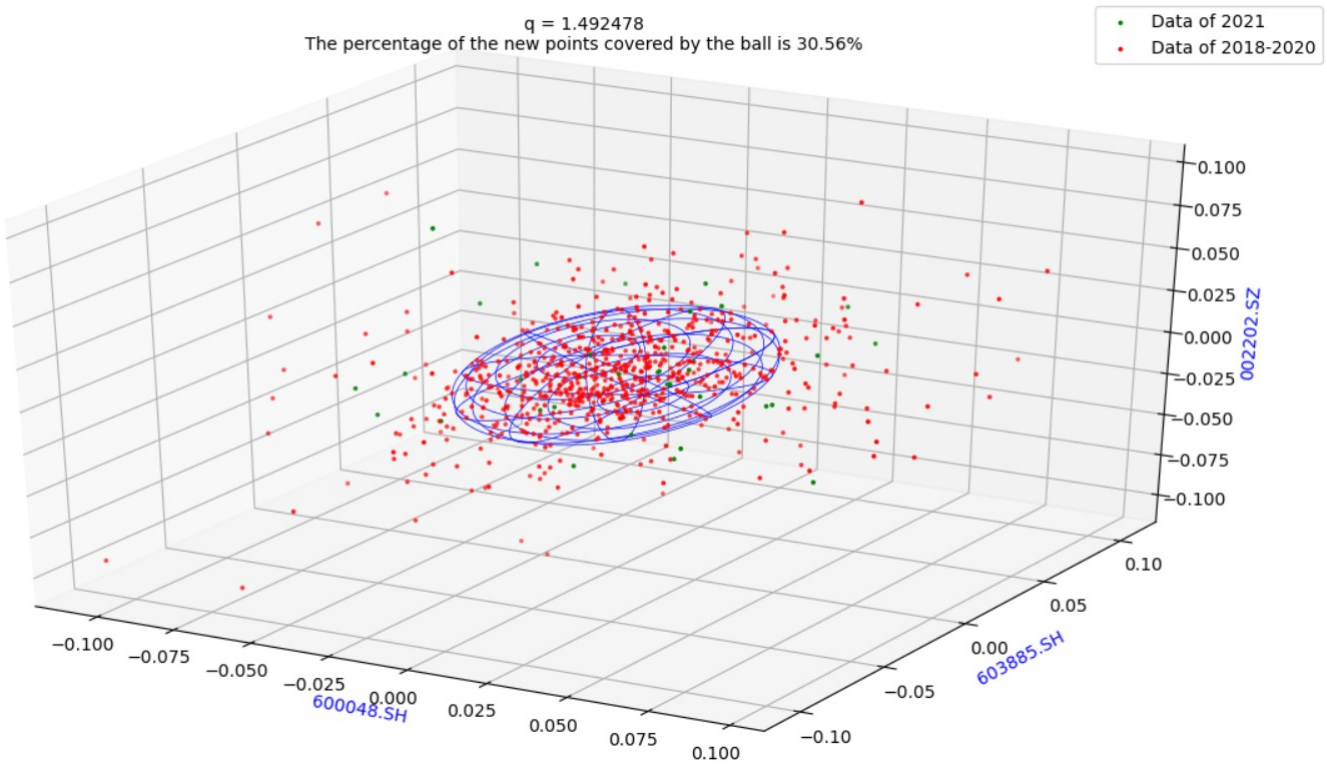


（d）不同的 Location – Dispersion Ellipsoid 包含的样本点的比例如下表：

The value of q	The percentage of the points covered by the ball
$(1-10\%)q$	53.09%
$(1-5\%)q$	55.97%
q	60.22%
$(1+5\%)q$	63.10%
$(1+10\%)q$	66.53%

由表可知，随着 q 的增大，被 Ellipsoid 包含的样本点的比重也在变大。

（e）下图中用红色的点标注了 2018-2020 的数据，用绿色的点标注了 2021 年的数据。计算之后发现，只有 30.56% 的新数据可以被依据历史数据刻画的 Location – Dispersion Ellipsoid 包含，说明我选择的三只股票在 2021 年的分布与在 2018-2020 年的分布差异较大，用历史数据去推测它们未来的表现也许不太合适。结合现实来看，在 2018-2020 和 2021 年两个阶段，中国的新能源政策、房地产调控政策发生了一些变化。“十四五”规划提出了雄心勃勃的碳排放削减计划，并要求稳定地价、房价和预期。因此，金风科技（新能源）与保利地产（房地产）的股价，必然会受到“十四五”规划出台的影响，使得它们在 2021 年的表现（分布）与 2018-2020 不大相同。航空业在 2020 遭受疫情的巨大冲击，在 2021 年缓慢恢复，因此吉祥航空在 2021 年表现出与 2018-2020 不同的股价分布也能够理解。



附录

第 1 题所用 Python 代码如下：

```
import numpy as np
import scipy.stats as st
import seaborn as sns

np.random.seed(0)

# 产生正态分布随机数
x1 = np.random.normal(0, 1, size=1000000)

# 产生指数分布随机数
x2 = np.random.exponential(1, size=1000000)

# 进行概率积分变换
u = np.random.uniform(0, 1, size=10000)

# 累积函数的逆
q1 = st.norm(0, 1).ppf(u)
q2 = st.expon(1).ppf(u)

# 画图
h1 = sns.jointplot(u, q1)
h1.set_axis_labels("original", "transformed", fontsize=10)
h1.savefig("正态分布概率积分变换.png")

h2 = sns.jointplot(u, q2)
```

```
h2.set_axis_labels("original", "transformed", fontsize=10)
h2.savefig("指数分布概率积分变换.png")
```

第 2 题所用 Python 代码如下：

```
import pandas as pd
import os
import numpy as np
import matplotlib.pyplot as plt

# 导入数据
os.chdir("C:\\Users\\liufengqi\\Desktop")
df = pd.read_table("第二题数据.txt", sep="\t", header=0, encoding="UTF-8", index_col=0)
data = np.asarray(df)

# 画直方图
plt.hist(data[:, 0], bins=200, color="blue", range=(-10, 10), label=df.columns[0])
plt.legend()
plt.show()
```

第 4 题所用 Python 代码如下：

```
import pandas as pd
import os
import numpy as np
import matplotlib.pyplot as plt

# 读取股价历史数据
os.chdir("C:\\Users\\liufengqi\\Desktop")
df = pd.read_table("股价历史数据.txt", sep="\t", header=0, encoding="UTF-8", index_col=0)
data = np.asarray(df)
stock1 = data[:, 0]
stock2 = data[:, 1]
stock3 = data[:, 2]

# 计算日收益率均值、协方差矩阵及逆矩阵
E = np.asarray(df.mean(axis=0))
cov = np.asarray(df.cov())
inv_cov = np.linalg.inv(cov)

# 求协方差矩阵的特征值和特征向量
eigvals, eigvecs = np.linalg.eig(cov)
# print('特征值数组:\n', eigvals)
# print('特征向量:\n', eigvecs)
ld = np.asarray([[eigvals[0] ** 0.5, 0, 0], [0, eigvals[1] ** 0.5, 0], [0, 0, eigvals[2] ** 0.5]])

# 求所有样本点距离均值点的马氏距离的均值 q
q = sum([np.sqrt((x - E).T @ inv_cov @ (x - E)) for x in data]) / df.shape[0]
# print(q)

# 求出有多少样本点会被包含在 Ellipsoid 之内
```

```
count = 0
for i in data:
    dis = np.sqrt((i - E).T @ inv_cov @ (i - E))
    if dis < q:
        count += 1

per = count / df.shape[0]
print("The percentage of the points covered by the ball is %.2f" % (per * 100) + "%")

#  下面开始画 Ellipsoid
def DrawEllipsoid(Distance, MeanValue):
    #  先画一个单位球
    # center and radius
    center = MeanValue
    radius = Distance

    # data
    u = np.linspace(0, 2 * np.pi, 100)
    v = np.linspace(0, np.pi, 100)
    x = radius * np.outer(np.cos(u), np.sin(v))
    y = radius * np.outer(np.sin(u), np.sin(v))
    z = radius * np.outer(np.ones(np.size(u)), np.cos(v))
    ball = np.asarray([x, y, z])

    #  通过矩阵乘法（线性变换）将单位球拉伸、旋转为 Ellipsoid
    ellpsoid = []
    for i in range(0, 100):
        ellpsoid.append(np.dot(np.dot(eigvecs, ld), ball[:, :, i]))

    ellpsoid = np.asarray(ellpsoid)
    x = ellpsoid[:, 0, :] + center[0]
    y = ellpsoid[:, 1, :] + center[1]
    z = ellpsoid[:, 2, :] + center[2]
    return x, y, z

#  开始绘图
# plot
fig = plt.figure(figsize=(100, 100))

# wire frame
x1, y1, z1 = DrawEllipsoid(q, E.tolist())
x2, y2, z2 = DrawEllipsoid(1.05 * q, E.tolist())
x3, y3, z3 = DrawEllipsoid(1.1 * q, E.tolist())
x4, y4, z4 = DrawEllipsoid(0.95 * q, E.tolist())
x5, y5, z5 = DrawEllipsoid(0.9 * q, E.tolist())

ax = fig.add_subplot(projection='3d')
ax.plot_wireframe(x1, y1, z1, rstride=10, cstride=10, color="blue", linewidth=0.5, label = "q")
# ax.plot_wireframe(x2, y2, z2, rstride=10, cstride=10, color="green", linewidth=0.5, label="(1+5%)q")
```

```
# ax.plot_wireframe(x3, y3, z3, rstride=10, cstride=10, color="purple", linewidth=0.5, label="(1+10%)q")
# ax.plot_wireframe(x4, y4, z4, rstride=10, cstride=10, color="yellow", linewidth=0.5, label="(1-5%)q")
# ax.plot_wireframe(x5, y5, z5, rstride=10, cstride=10, color="black", linewidth=0.5, label="(1-10%)q")

#  把样本点也画出来
ax.scatter(stock1, stock2, stock3, s=0.1, c="red", zorder=1)

ax.set_zlabel('002202.SZ', fontdict={'size': 10, 'color': 'blue'})
ax.set_ylabel('603885.SH', fontdict={'size': 10, 'color': 'blue'})
ax.set_xlabel('600048.SH', fontdict={'size': 10, 'color': 'blue'})
# plt.title("q = %f" % q + "\n" + "The percentage of the points covered by the ball is %.2f" % (per * 100) + "%",
              # fontsize=10)
plt.legend(loc=1, ncol=1)

#  以下为最后一小题的代码，请勿与前面的代码同时运行

#  读取 2021 年股价数据
df1 = pd.read_table("2021 股价数据.txt", sep="\t", header=0, encoding="UTF-8", index_col=0)
data1 = np.asarray(df1)
newstock1 = data1[:, 0]
newstock2 = data1[:, 1]
newstock3 = data1[:, 2]

#  绘制 2021 股价散点图

ax.scatter(newstock1, newstock2, newstock3, s=3, c="green", zorder=1, label = "Data of 2021")

#  历史数据的散点也绘制出来，以供对比
ax.scatter(stock1, stock2, stock3, s=3, c="red", zorder=1, label = "Data of 2018-2020")

ax.set_zlabel('002202.SZ', fontdict={'size': 10, 'color': 'blue'})
ax.set_ylabel('603885.SH', fontdict={'size': 10, 'color': 'blue'})
ax.set_xlabel('600048.SH', fontdict={'size': 10, 'color': 'blue'})

#  计算新数据有多少被包含在 Ellipsoid 之内
count = 0
for i in data1:
    dis = np.sqrt((i - E).T @ inv_cov @ (i - E))
    if dis < q:
        count += 1
print(count)
per = count / df1.shape[0]
print(per)
print("The percentage of the points covered by the ball is %.2f" % (per * 100) + "%")
plt.title("q = %f" % q + "\n" + "The percentage of the new points covered by the ball is %.2f" % (per * 100) + "%",
          fontsize=10)
plt.legend(loc=1, ncol=1)

plt.show()
```