量化金融分析师（AQF®）全国统一考试

模拟题

适用场次：2023 年 3 月

使用本模拟题，您应该遵守：

1. 本模拟题仅提供给参加 2023 年 3 月份 AQF 全国统一考试的考生，考生仅可以出于准备个人考试的目的查阅和打印本模拟题；
2. 严禁出于任何目的的复制、网络发布和传播、抄袭本模考题内容，如有违反，可能导致违纪或违法行为；

量化金融标准委员会

Standard Committee of Quantitative Finance

量化金融分析师（AQF®）全国统一考试模拟题

说明：本场考试中的代码都应采用 Python 3.X 版本作答。

解答题（每题 20 分，本部分共 100 分）

1．某研究员正在构建一个配对交易策略，他选取了两种期货指数作为配对交易的对象。为了验证此两种期货具有一定相关性，他获取此两种期货合约指数 2021 年的价格数据作为训练集，分别存储于 csv 文件 training1 和 training2 中，部分数据如下：

trade_date	close
2021/12/31	4947.40
2021/12/30	4939.80
2021/12/29	4910.00
2021/12/28	4978.40

trade_date	close
2021/12/31	7377.80
2021/12/30	7336.20
2021/12/29	7288.40
2021/12/28	7309.20

其中，trade_date 为交易日日期，close 为期货指数当日收盘价。同时，他获取了 2022 年作为回测区间，将 2022 年的期货指数价格数据存储于 csv 文件 test 中，部分数据如下：

trade_date	close_1	close_2
2022/1/4	4922.20	7365.60
2022/1/5	4879.00	7232.60
2022/1/6	4820.60	7227.20
2022/1/7	4836.00	7198.00

1.1．为了检验这两个期货指数是否适合进行配对交易，该研究员准备使用训练集数据进行相关性检查，请你使用 training1 和 training2 中的数据，绘制两个期货指数的价格走势图，并计算秩相关系数。（8 分）

参考答案：

```
# 读取数据并进行拼接（4 分）
training1 = pd.read_csv('training1.csv', index_col=0,
parse_dates=True).sort_index()
```

```
training2 = pd.read_csv('training2.csv', index_col=0,
parse_dates=True).sort_index()training = training1.join(training2,
how='inner', lsuffix='_1', rsuffix='_2')
```

或 `training = pd.merge(training1, training2, left_index=True, right_index=True, suffixes=('_1', '_2'))`

```
# 绘制价格走势图（2 分）
training.plot(figsize= (8, 6))
# 计算相关系数（2 分）
training.corr(method='spearman')
```

1.2. 在检验了两个期货指数的相关性之后，他认为这两种期货指数适合进行配对交易，并准备使用 2021 年价格数据进行配对交易策略回测。该研究员认为，当两个期货指数价差偏离均值 2 倍标准差时存在错误定价，并进行开仓操作，回归正常价格区间时进行平仓。请你使用题目所提供的数据，计算交易信号。（9 分）

参考答案：

```
# 计算训练集均值和标准差（2 分）
training['spread'] = training['close_1'] - training['close_2']
mu = training['spread'].mean()
sigma = training['spread'].std()
# 读取测试集数据
test = pd.read_csv('test.csv', index_col=0, parse_dates=True)
# 计算测试集价差上下轨（4 分）
test['spread'] = test['close_1'] - test['close_2']
test['up'] = mu + 2 * sigma
test['down'] = mu - 2 * sigma
# 计算交易信号（3 分）
test['position_1'] = np.where(test['spread'] > test['up'], -1, 0)
test['position_1'] = np.where(test['spread'] < test['down'], 1,
```

```
test['position_1'])
test['position_2'] = -test['position_1']
```

1.3. 请问构建配对交易策略的前提条件是什么？答出任意两点即可。（3 分）

参考答案：

“卖空机制”

作为市场中性策略（Market Neutral Strategy）的一种，没有卖空机制是不可能执行配对交易的。虽然中国 A 股市场已经有了融资融券机制，但标的证券的种类太少，没有转融通机制导致能借到证券的数量也不多，所以诸多限制使得配对交易并没有大量兴起。一些券商研究部的金融工程组撰写的报告大多是从理论上进行分析。相信这一现象会随着市场交易机制的完善而改变。

“构建配对组”

广义的配对交易范畴很大，只要有两个风险收益特征非常相似的证券都可以配对。比如，两份交割日不同的沪深 300 股指期货合约可以配对，四川长虹和它的权证可以配对，招商银行的 A 股和 H 股可以配对，同一行业内基本面相似的公司（比如建行和工行）可以配对。在我们构建配对组时，我们既可以按照基本面来分析，比如寻找同一行业内主营业务同质化较高的公司。我们也可以完全按照统计原理进行筛选，寻找相关系数较高的股票。

“价差（Spread）收敛”

根据对历史数据的统计，两只股票的价差是长期稳定的，但稳定性会因投资者“追涨杀跌”的不理性交易而打破，导致价差扩大。这时我们就同时建立多空头寸，卖空相对高估的股票，买入相对低估的股票，等待价差收敛。价差收敛后，我们同时将多空头寸平仓，就可以获得收益。“价差收敛”是配对交易能否获利的关键假设，这种收敛的假设是根据过去配对股票价差的标准差来判断的，是数量分析的结果。如果某只公司的基本面发生了变化，价差就可能会进一步扩大，此时只能清仓止损了。举一个极端的例子就是在 2008 年下半年将高盛和雷曼兄弟的股票配对。

2. 在多因子策略中，对于有效因子的挖掘与测试是模型构建的重要前提。除了使用分层检验之外，回归法和因子 IC 值也是衡量因子对收益的预测能力的方法之一。李明，某量化研究员，将股票池中的股票在 2022 年上半年的月度因子值数据和月度收益率数据，分

别存储于 csv 文件 factor.csv 和 returns.csv 中。其中，factor.csv 包含股票代码（code）、交易日日期（trade_date）、净资产收益率（roe）数据，部分数据如下：

code	trade_date	roe
300120.SZ	20220104	2.6246
603595.SH	20220104	6.2389
300218.SZ	20220104	7.0669
300338.SZ	20220104	10.8223
002222.SZ	20220104	5.1984

returns.csv 中包含股票代码（code）、交易日日期（trade_date）、月度收益率（pct_chg）数据，部分数据如下：

code	trade_date	pct_chg
000001.SZ	20220123	-0.0553
000002.SZ	20220123	-0.0994
000004.SZ	20220123	0.0102
000005.SZ	20220123	-0.0388
000006.SZ	20220123	-0.0522

2.1. 李明想要使用题目中提供的数据，筛选出 2022 年 4 月因子值数据和收益率数据，使用最小二乘法回归分析，判断在 95%显著性水平下，该因子是否显著。另外，在数据处理的过程中，李明发现净资产收益率数据单位为%，即数据放大了 100 倍，将其调整为收益率同量纲对数据分析效果可能更好。请按照如上要求处理数据，并完成上述分析（10 分）

参考答案：

```
# 获取因子值数据并做基础数据处理（4 分）
factor = pd.read_csv('factor.csv')
factor['trade_date'] = pd.to_datetime(factor['trade_date'],
format='%Y%m%d')
factor.set_index('trade_date', inplace=True)
factor['roe'] = factor['roe'] / 100
# 获取收益率数据并做基础数据处理（1 分）
```

```
returns = pd.read_csv('returns.csv')
returns['trade_date'] = pd.to_datetime(returns['trade_date'],
format='%Y%m%d')
returns.set_index('trade_date', inplace=True)
# 回归分析（5 分）
data = pd.merge(factor.loc['2022-04'], returns.loc['2022-04'])
import statsmodels.api as sm
x = data['roe']
x = sm.add_constant(x)
y = data['pct_chg']
est = sm.OLS(y, x).fit()
est.summary()
```

2.2 请你根据题目提供的数据，计算 2022 年该因子显著的比例，以及当因子显著时，因子对收益预测的方向（1 表示因子值越大收益越高，-1 表示因子值越大收益越低），效果如下（部分数据）：（10 分）

	significant	direction
2022-04	True	1.0
2022-05	False	-1.0
2022-06	True	1.0

参考答案：

```
result = pd.DataFrame(columns=['significant', 'direction'])
for mo in range(1, 7):
    mostr = f'2022-{mo:02}'
    df = pd.merge(factor.loc[mostr], returns.loc[mostr])
    x = df['roe']
    x = sm.add_constant(x)
    y = df['pct_chg']
    est = sm.OLS(y,x).fit()
```

```
    significant = (est.pvalues['roe'] < 0.05)
    direction = np.sign(est.params['roe'])
    result.loc[mostr] = significant, direction
```

3．李明，AQF，某量化研究员，正在构建一个 CTA 趋势交易策略。他选取了某期货指数作为研究对象该期货指数 2021 年的收盘价的部分数据，储存在 future.csv 中。

trade_date	close
2021/12/31	2807.50
2021/12/30	2818.00
2021/12/29	2810.50
2021/12/24	2837.00
2021/12/23	2846.00
2021/12/22	2824.50

其中 trade_date 为交易日期，close 为期货当日收盘价。

3.1．读入数据，以 DataFrame 格式储存在变量 future 中，将索引改成日期的升序。然后计算出 20 日均线，储存在 future 的“MA20”列；计算出 60 日均线，储存在 future 的“MA60”列。（5 分）

参考答案：

```
import pandas as pd
import numpy as np
# 读入数据（2 分）
future = pd.read_csv('future.csv', index_col=0, parse_dates=True)
future.sort_index(inplace=True)
#计算均线（2 分）
future['MA20'] = future['close'].rolling(20).mean()
future['MA60'] = future['close'].rolling(60).mean()
```

3.2. 画出只包含期货指数 10 日均线及 60 日均线的折线图。（4 分）

参考答案：

```
import matplotlib.pyplot as plt
future[['MA20', 'MA60']].plot()
plt.show()
```

3.3. 基于各列的数据（收盘价、20 日均线、60 日均线），构建一个经典的趋势交易策略，简述策略的交易逻辑（4 分）

参考答案：

双均线交易策略：当 20 日均线上穿 60 日均线时，看多；

当 20 日均线下穿 60 日均线时，看空；

3.4. 根据前面的交易逻辑，在现有数据基础上计算策略的仓位趋势（0 表示未持仓，1 为持有多头，-1 为持有空头），保存在 future 的’positions’，并绘制全时段的仓位趋势图。（7 分）

参考答案：

```
# 计算仓位趋势 (4 分)
future['position'] = np.where(future['MA20'] > future['MA60'], 1, -1)
future.loc[future['MA20'].isna() | future['MA60'].isna(), 'position'] =
0
# 画出仓位趋势图(3 分)
future['position'].plot(ylim=[-1.1, 1.1],
                        title='Market Positioning',
                        figsize=(16, 8))
plt.show()
```

4. 在投资中，风险管理对于稳定投资收益具有非常重要的作用。在险价值 VaR 是风险管

理领域的一个重要工具。它是指在假定的市场条件下，在一定的时间段内，预期会在一定百分比的时间内以货币单位或占投资组合价值的百分比的最大损失。某私募公司的量化研究员想要使用 VaR 来度量某投资标的风险，他将该标的近半年的某股票日交易数据储存在名为 stock_data 的 csv 文件中，部分数据如下表：

date	open	high	low	returns
2021/7/1	7.11	7.29	7.11	NaN
2021/7/2	7.14	7.31	7.08	-0.0288
2021/7/5	7.37	7.51	7.25	0.024
2021/7/6	7.37	7.44	7.24	0.0207
2021/7/7	7.36	7.4	7.18	-0.0216
2021/7/8	7.33	7.63	7.31	0.0166
2021/7/9	7.86	7.86	7.25	-0.0095
2021/7/12	7.74	8.17	7.7	0.1043
2021/7/13	7.77	7.85	7.65	-0.0323
2021/7/14	7.49	7.78	7.45	-0.0064

请完成以下分析步骤。（已按惯例导入相关依赖库：`import pandas as pd; import numpy as np; import matplotlib.pyplot as plt`）

4.1. 已知 2021/7/1 的收盘价为 7.25，请根据日收益率序列计算该股票收盘价，并将其保存在新增加的一列变量 close 中，此处日收益率为算数收益率；（5 分）

参考答案：

```
# 读取数据（2 分）
df = pd.read_csv('stock_data.csv', index_col=0, parse_dates=True)
# 计算收盘价，并储存到新增一列（3 分）
df['close'] = 7.25 * (1 + df['returns']).cumprod()
df.iloc[0, 4] = 7.25
```

或 `df['close'] = 7.25 * (1 + df['returns'].fillna(0)).cumprod()`

4.2. 在险价值 VaR 的估计有多种方法，接下来使用蒙特卡罗模拟法来计算 VaR。我们假设股票价格服从几何布朗运动，股价变化的离散表达式为$S_{t+\Delta t} = S_t\left(1 + \mu\Delta t + \sigma\varepsilon\sqrt{\Delta t}\right)$，其中 $S_{t+\Delta t}$表示经过时间间隔Δt之后股票的价格，S_t表示t时刻股票的价格，μ表示股票对数收

益率的均值，σ表示股票收益率的标准差，ε表示正态分布随机数。对股票价格路径设置10000次的蒙特卡洛模拟，单次模拟步数为100次，模拟1天后股票可能的价格，请写出相关代码，将模拟生成的10000个第二天股票价格记录在simulated_price中。（11分）

参考答案：

```
# 设置蒙特卡罗模拟方法的参数（4 分）
M = 10000  # 蒙特卡罗模拟的次数
N = 100  # 设置单次蒙特卡罗模拟的步数
T = 1  # 计算一天之后的股票可能的价格
dt = T/N  # 单次蒙特卡罗模拟的步长
# 数据准备
simulated_price = np.zeros((M, 1))
close_ = df['close'].values
returns = np.log(close_[1:] / close_[:-1])

# 计算收益率均值和标准差（2 分）
mean_ = np.mean(returns)
std_ = np.std(returns)

# 蒙特卡罗模拟（5 分）
for i in range(M):
    S = close_ [-1]
    for j in range(N):
        S = S * (1 + mean_ * dt + std_ * dt ** 0.5 * np.random.normal())
        simulated_price[i - 1] = S
```

4.3. 根据模拟获得的收益率，计算95%和99%的置信水平下VaR对应的收益率并打印，保留4位小数，用百分比表示（4分）

参考答案：

??(simulated_price-close_)/close_

```
simulated_return = np.log(simulated_price / close_[-1])
VaR_95 = np.percentile(simulated_return, 5)
VaR_99 = np.percentile(simulated_return, 1)
print('95%VaR: {:.4%}'.format(VaR_95))
print('99%VaR: {:.4%}'.format(VaR_99))
```

5. 李明，某量化基金经理，正在测试逻辑回归模型的预测能力。他认为，根据个股的市盈率和市值大小可以判断出个股是否存在投资机会。他将个股市盈率和对数市值作为逻辑回归模型的特征数据储存在文件 data_x.csv 中，将个股当年的涨幅数据储存在文件 data_y.csv 中。

data_x 部分数据如下：

code	pe	log_mv
600000.SH	4.6834	17.1949
600004.SH	-77.0214	14.7886
600006.SH	16.9126	14.1939
600007.SH	21.0572	14.4083
600008.SH	13.382	14.6083

data_y 部分数据如下：

code	change
600000.SH	0.0331
600004.SH	-0.2088
600006.SH	-0.2267
600007.SH	0.4726
600008.SH	0.0888

现需要你根据以下步骤描述，编写相应代码。

5.1. 定义函数 standardize 用于数据标准化。假设函数参数为 data_x 中的一列特征数据 x，函数返回值为标准化之后的特征数据，标准化公式为：

$$x^* = \frac{x - \bar{x}}{\sigma}$$

使用上述函数对特征数据进行标准化，并将标准化后的数据保存在变量 data_2 中。（4 分）

参考答案：

```
# 定义函数（2 分）
def standardize(x):
    return (x - x.mean())/x.std()
# 读取数据
data_x = pd.read_csv('data_x.csv', index_col=0)
# 标准化（2 分）
data_2 = data_x.apply(standardize)
```

5.2. 将个股当年涨幅数据 data_y 转换成是否上涨的标签数据并保存在变量 data_3 中。（2 分）

参考答案：

```
# 读取数据
data_y = pd.read_csv('data_y.csv', index_col=0)
# 处理标签
data_3 = data_y.apply(np.sign)
```

5.3. 将特征数据和标签数据进行分组，其中 80%作为训练集，20%为测试集，分别将训练集和测试集的特征数据和标签数据存储于变量 train_x, train_y, test_x, test_y 中。（4 分）

参考答案：

```
from sklearn.model_selection import train_test_split
train_x, test_x, train_y, test_y = train_test_split(data_2,
data_3.values.flatten(), test_size=0.2)
```

5.4. 使用决策树模型进行模型训练。（4 分）

参考答案：

```
from sklearn.tree import DecisionTreeClassifier
```

```
clf_DT = DecisionTreeClassifier()
clf_DT.fit(train_x, train_y)
```

5.5 使用测试集数据对模型进行测试，根据测试集特征预测标签，并将预测的结果保存在变量 predict_y 中。（2 分）

参考答案：

```
predict_y = clf_DT.predict(test_x)
```

5.6. 根据以下公式计算模型准确率：

模型准确率=测试集中预测正确的次数/总的预测次数

（4 分）

参考答案：

```
accuracy = (test_y == predict_y).sum()/len(test_y)
```