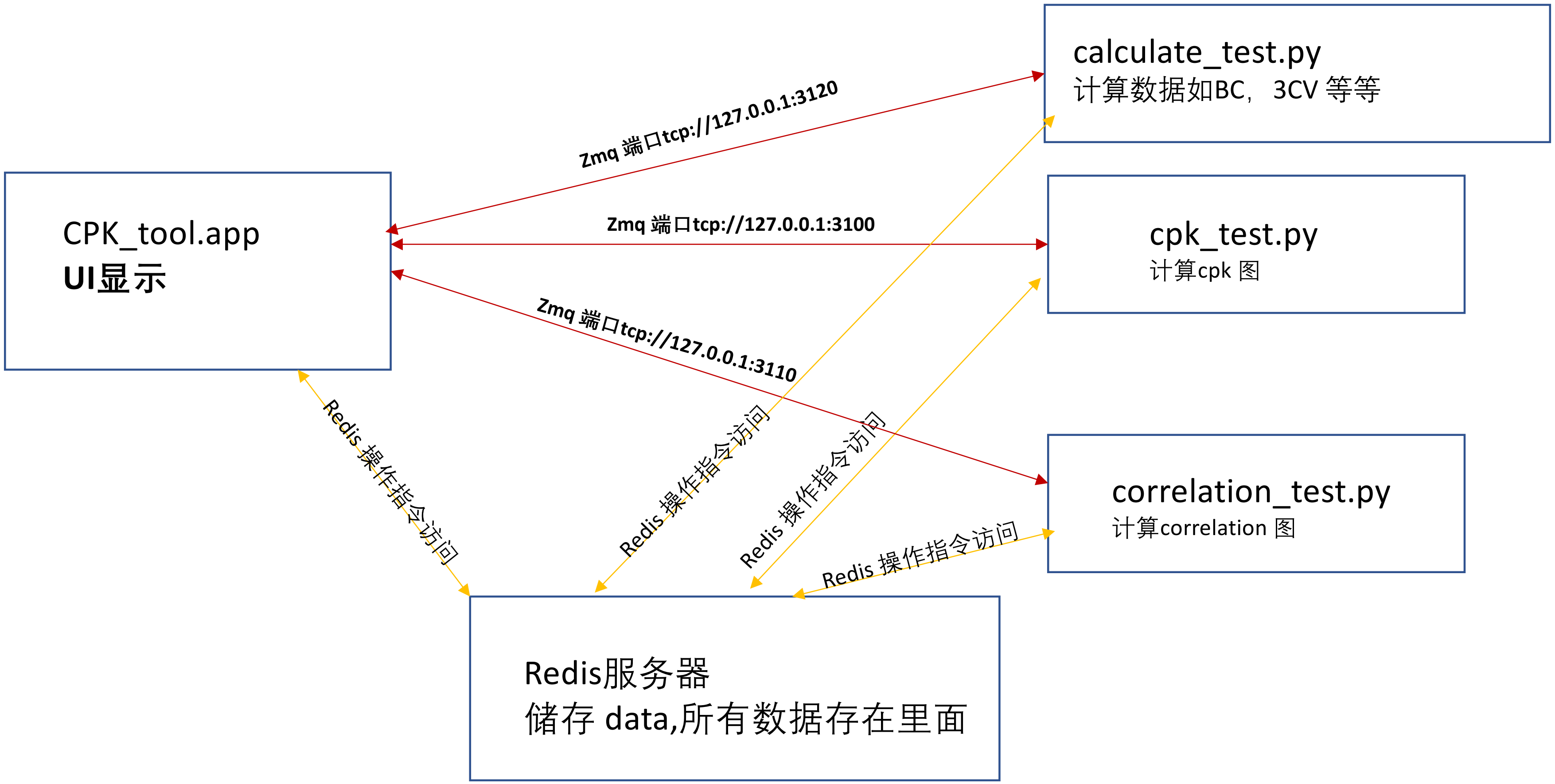


CPK_tool.app
UI显示

calculate_test.py
计算数据如BC，3CV 等等

cpk_test.py
计算cpk 图

correlation_test.py
计算correlation 图

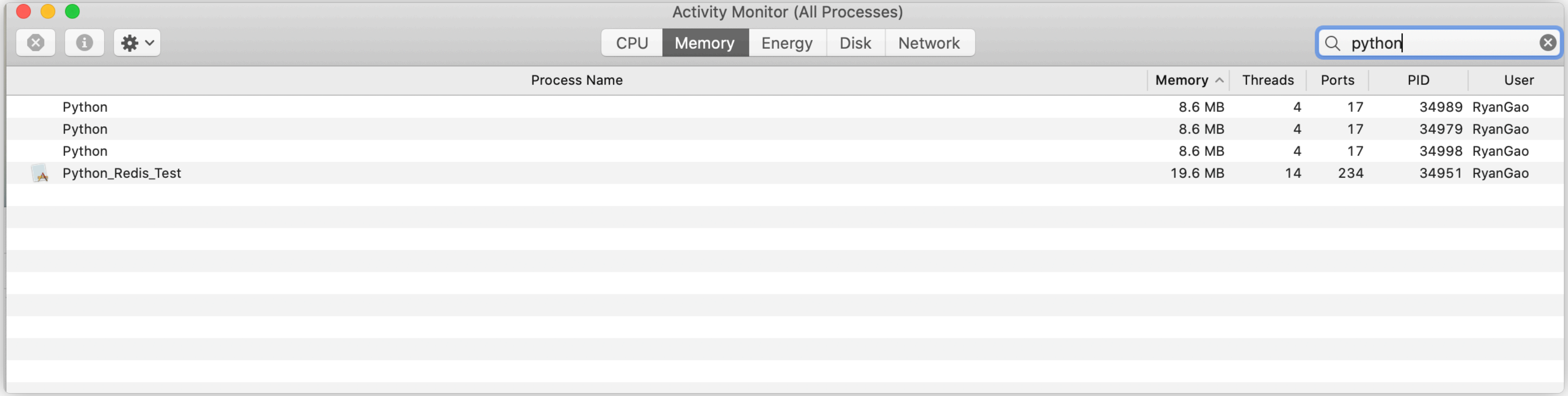
Redis服务器
储存 data,所有数据存在里面

Zmq 端口tcp://127.0.0.1:3120
Zmq 端口tcp://127.0.0.1:3100
Zmq 端口tcp://127.0.0.1:3110
Redis 操作指令访问
Redis 操作指令访问
Redis 操作指令访问
Redis 操作指令访问

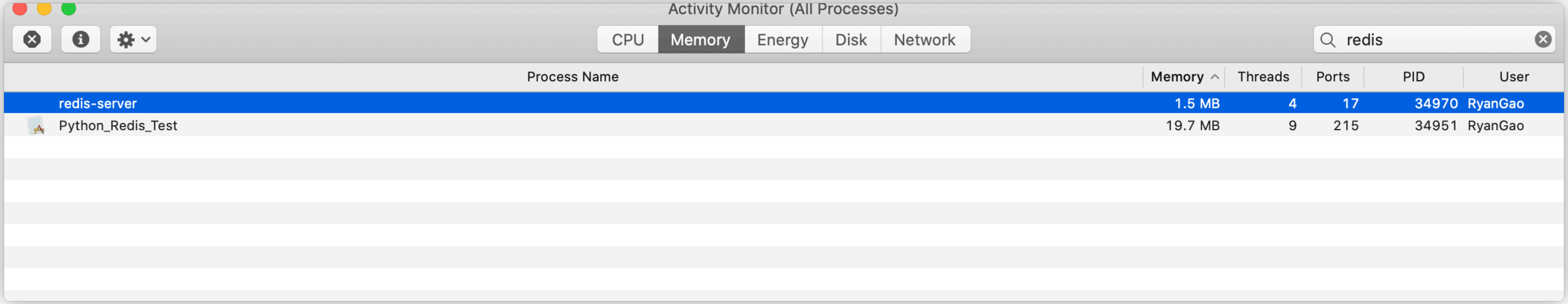
首先安装 pip3 install zmq

启动app，可以看到拉起3个python 程序和一个redis-server.

calculate_test.py 和cpk_test.py 和correlation_test.py



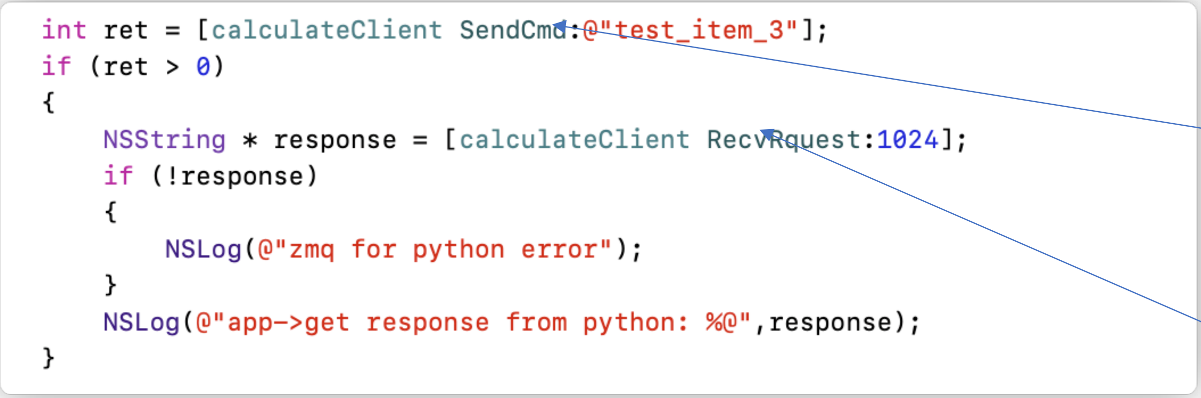
Process Name	Memory	Threads	Ports	PID	User
Python	8.6 MB	4	17	34989	RyanGao
Python	8.6 MB	4	17	34979	RyanGao
Python	8.6 MB	4	17	34998	RyanGao
Python_Redis_Test	19.6 MB	14	234	34951	RyanGao



Process Name	Memory	Threads	Ports	PID	User
redis-server	1.5 MB	4	17	34970	RyanGao
Python_Redis_Test	19.7 MB	9	215	34951	RyanGao

注意事项：

是建立的REP-REQ 模式，也就是说只有app 往python 发数据，再接受python 返回数据。

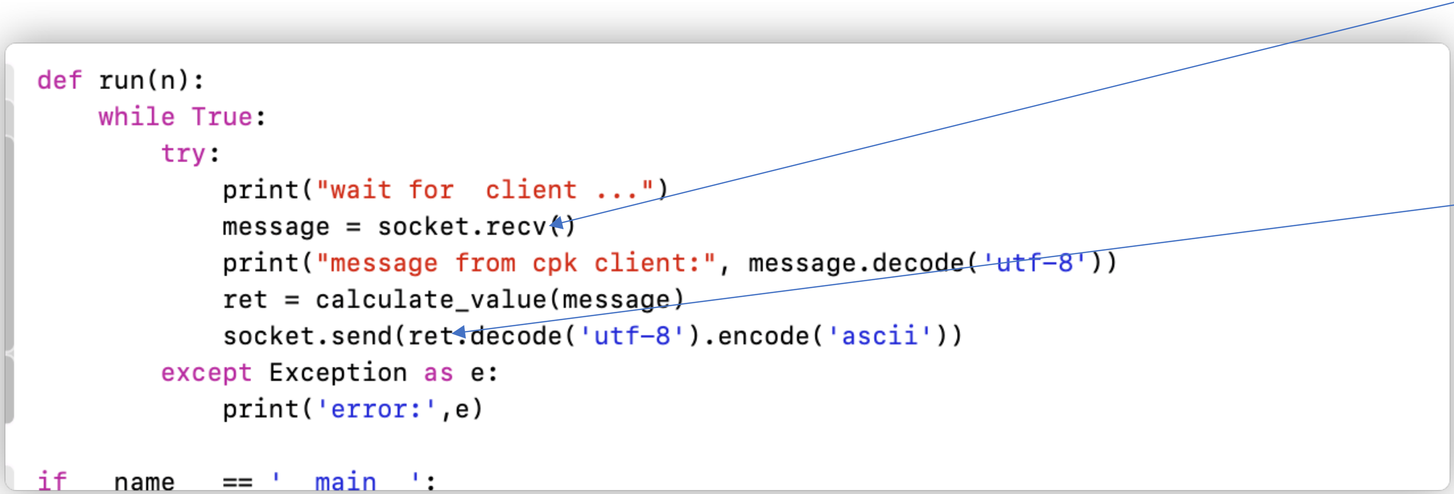


```
int ret = [calculateClient SendCmd:@"test_item_3"];
if (ret > 0)
{
    NSString * response = [calculateClient RecvRquest:1024];
    if (!response)
    {
        NSLog(@"zmq for python error");
    }
    NSLog(@"app->get response from python: %@",response);
}
```

App那边主动发送

等待python 处理完后的数据回复

等待app那边发送的数据

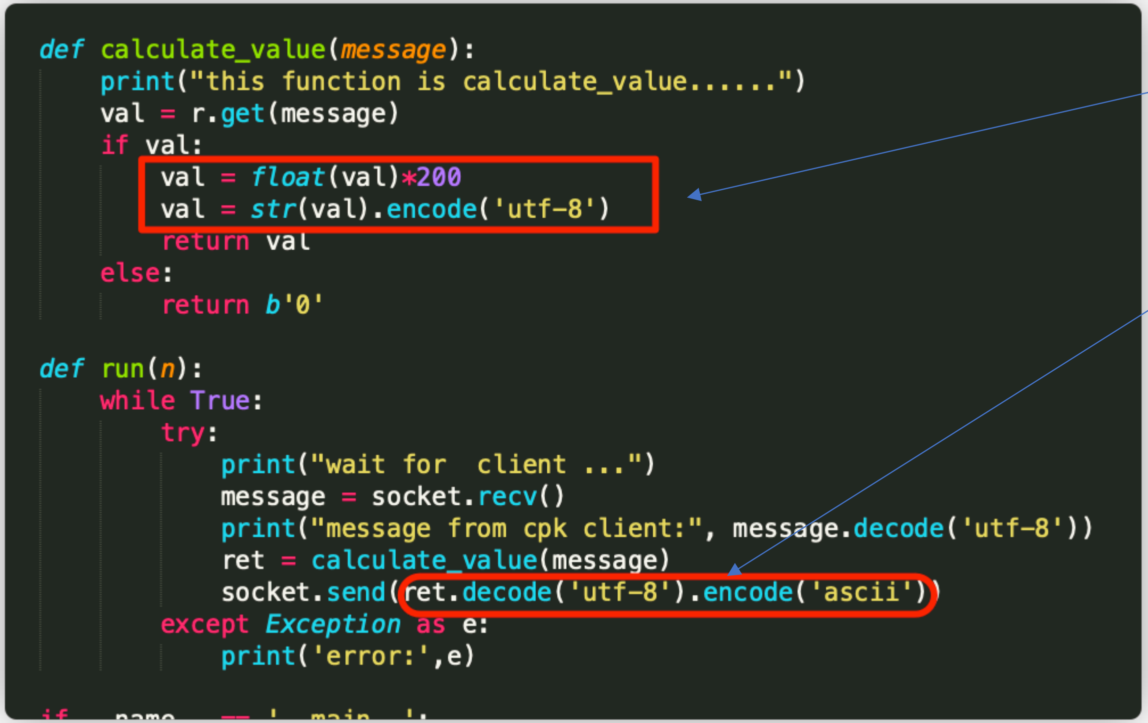


```
def run(n):
    while True:
        try:
            print("wait for  client ...")
            message = socket.recv()
            print("message from cpk client:", message.decode('utf-8'))
            ret = calculate_value(message)
            socket.send(ret.decode('utf-8').encode('ascii'))
        except Exception as e:
            print('error:',e)

if __name__ == '__main__':
```

Python 处理完，发送给app 处理结果

注意事项，由于现在采用的是python3.8模式
所以对python发送，send 的时候要用**bytes类型字符串，这点是和python2.7的区别**



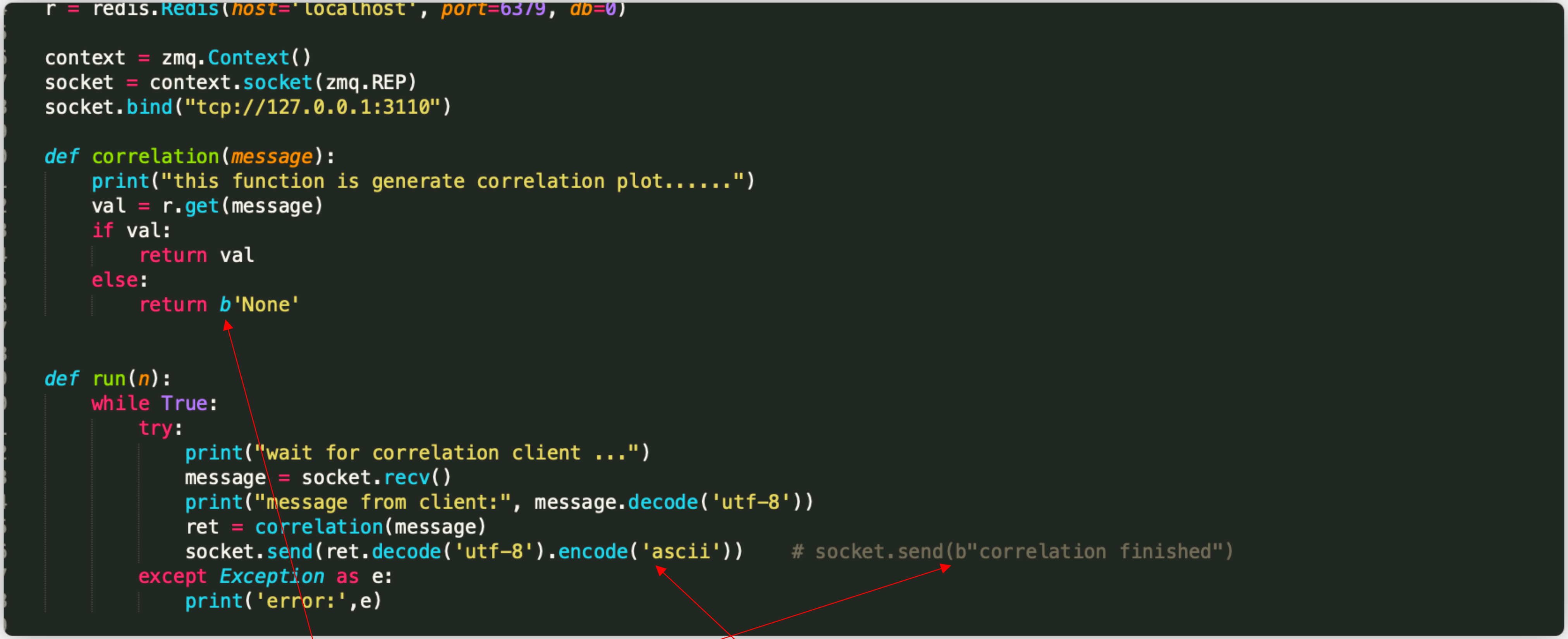
```
def calculate_value(message):
    print("this function is calculate_value......")
    val = r.get(message)
    if val:
        val = float(val)*200
        val = str(val).encode('utf-8')
        return val
    else:
        return b'0'

def run(n):
    while True:
        try:
            print("wait for  client ...")
            message = socket.recv()
            print("message from cpk client:", message.decode('utf-8'))
            ret = calculate_value(message)
            socket.send(ret.decode('utf-8').encode('ascii'))
        except Exception as e:
            print('error:',e)

if __name__ == '__main__':
```

如果是字符串，转换数字计算，再转换成字符

发送的时候要assii 码，这是由于python3.8关系
如果是python2.7 就不需要



```
r = redis.Redis(host='localhost', port=6379, db=0)

context = zmq.Context()
socket = context.socket(zmq.REP)
socket.bind("tcp://127.0.0.1:3110")

def correlation(message):
    print("this function is generate correlation plot......")
    val = r.get(message)
    if val:
        return val
    else:
        return b'None'


def run(n):
    while True:
        try:
            print("wait for correlation client ...")
            message = socket.recv()
            print("message from client:", message.decode('utf-8'))
            ret = correlation(message)
            socket.send(ret.decode('utf-8').encode('ascii'))    # socket.send(b"correlation finished")
        except Exception as e:
            print('error:',e)
```

注意发送的时候，前面的b

注意ascii，这是由于python3.8 才这样，
python2.7就不用这么麻烦