Problem Chosen	2020 MCM/ICM Summary Sheet	Team Control Number
A		2010067

Pay, Profit--- How to Survival Under the Change

Herring and mackerel are one of the important marine fish resources in the North Pacific. Their rich nutritional value and large quantity attract the attention of hundreds of fishing companies in coastal countries. They are pelagic fish that feed mainly on plankton. In the trend of ocean temperature warming, the area suitable for their survival is moving northward year by year. Therefore, we speculate that the survival area of fish stocks will play a significant role in the development of Fisheries in coastal countries and even in the world.

First of all, we established the logistic growth model of fish. We compared the physiological characteristics of Atlantic herring and mackerel, and found that there was a great similarity between the two kinds of fish. Therefore, we did not distinguish the two kinds of fish and analyzed them with herring as an example. After obtaining the data of quantity change in the past 16 years, we made curve fitting and got the growth model of fish. Then we predict the population of herring in the Atlantic Ocean, and we conclude that the population of herring will be stable at about 200 thousandths of metric tons in the future without great changes in the natural environment

Then, we analyzed the data of Sea Surface Temperatures(SST) change in the past 33 years and predicted three possible temperature changes in the future. From the feature that herring and mackerel are influenced by the sea temperature, we assume that the key point of solving this problem convert from hammering out a time when small fishing company not making harvest, to foresee the turning point of the temperature in North Atlantic Ocean when it is too hot to breed this two species of fish. Therefore we get Ocean temperature data from year 1985 to 2018 from National Oceanic and Atmospheric Administration (NOAA). Then, we use Basemap and Matplotlib to draw the thermodynamic diagram and SST average change line diagram. The thermodynamic diagrams and line chart show that the temperature fluctuate irregularly, but have an upward trend. To predict the unsure temperature change, we use linear regression to fit the data that have already knew. And base on assumptions, we create three kinds of function to predict future ocean temperature. Additionally, to make our prediction more close the reality, each assumption we add a small random number in each year’s temperature. When we have such data by presumption, we draw the future ocean temperature graph and find that whatever type of assumption we choose, the shoals will definitely move to the north, just a matter of time and distant. And when the year come to the temperature around Scotland overpass the limit of fish survival range(15℃), there will be no fish in that area anymore.

Next, according to the current price level and the development of fishing industry, we simulate the cost and benefit of small fishing companies. And in view of the most ideal conditions, this paper puts forward suggestions for improving the company's business strategy. We conclude that without changing the business strategy, the profit will decrease by 22%, while after the change, the profit will increase by 40%.

Finally, we analyze our business strategy in accordance with International Law and the Common Fisheries Policy of the UK and the Faroe Islands. And we contribute to the magazine based on our entire analysis and unique personal experience. Last but not least, we evaluate the established models.

Content

1 Introduction

1.1 Background

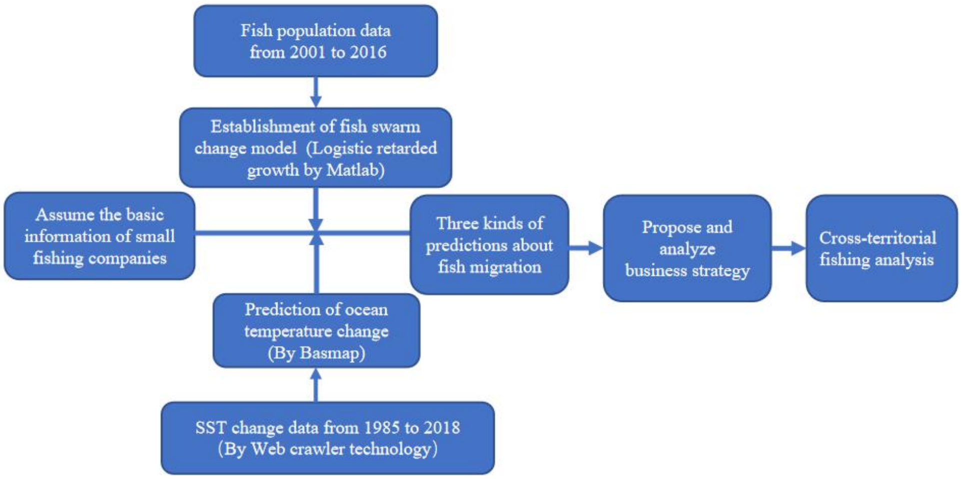
Climate change forces animals to find new environment to meet the needs of survival through the impact on biological habitat. Take herring, which live widely in the North Atlantic, as an example. In 2001-2015, the herring in the North Sea fishery experienced three stages of change: Firstly, the stable period is 2001-2003, during which the quantity and abundance of Atlantic herring is very high. Then, it began to decline in 2004. In 2012, things changed and herring began to recover until 2015[1].

Since the 20th century, the global temperature has begun to rise gradually. According to a meteorological statement issued by WMO(World Meteorological Organization), the change in the northeast of the Atlantic Ocean is particularly obvious. The long-term impact of climate change in the North Atlantic has led to the rise of the sea surface temperature at an average annual rate of 0.6℃[2]. Moreover, studies have shown that marine fish have the most direct response to the change of sea water temperature, so the temperature has an vital impact on the early life cycle of herring, and the water temperature is positively correlated with the larval abundance[1]. Therefore, in the global warming trend, herring is bound to migrate.

As a consultant to the North Atlantic Fisheries Management Association of Scotland, we are obliged to make predictions on fish migration over the next 50 years and provide timely advice to fishing companies, especially small fishing companies with poor risk resistance.

1.2 Analysis and Approach Overview

The Sea Surface Temperature(SST) has a great influence on the quantity and location of fish school. Under the trend of global warming and rising sea temperature, the quantity and location of fish school will change. First of all, according to the Logistic retarded growth principle, we use MATLAB as a tool to establish the growth model of fish. Then we draw the thermodynamical diagram of temperature change in the sea area near Scotland and predict the future SST changes, using Basemap as a tool. After that, the operating conditions of small fishing companies are assumed and analyzed. Then we get three different changes of fish swarm in the optimal, the most ideal and the most extreme conditions. Finally, according to the relevant knowledge of economics, we put forward suggestions about the strategy of small fishing companies.


Fish population data from 2001 to 2016
Establishment of fish swarm change model (Logistic retarded growth by Matlab)
Assume the basic information of small fishing companies
Three kinds of predictions about fish migration
Propose and analyze business strategy
Cross-territorial fishing analysis
Prediction of ocean temperature change (By Basmap)
SST change data from 1985 to 2018 (By Web crawler technology)

2 Assumptions and Justification

· Suppose Atlantic mackerel and herring belong to the same species, according to their similar physiological characteristics.

· Suppose that all fishing boats used by small fishing companies can be bought on the market, and the relevant prices are in line with the current price level.

· Suppose that in order to obtain the maximum fishing capacity, the fishing boat will use up the gasoline in the tank every time it goes out to sea.

· Suppose that the fish only changes with temperature, not with other factors such as ocean pH.

· Suppose people can catch a small amount of these fish at 13 ℃.

· Suppose the fish can't survive at 15 ℃.

· Suppose that the rise of sea surface temperature(SST) is linear from the known data.

3 Symbols and Notations

Symbol:	Description:
t	The time
y	The estimated herring stocks
r	The growth rate
N_0	The initial number of herring
K	The environmental capacity
N	The herring stocks
L	The Maximum range of small fishing boat
D	Full tank gas
d	Gas consumption rate
v	Speed of fishing boat
k	The slope of linear regression calculate from temperature of single point
rand[n-2019]	An array of random number getting from the average temperature
singleStd	The standard deviation calculate from each single point
n	The year
w	The fisherman wage
wholeStd	The standard deviation calculate from the whole North Atlantic.
Tn	The temperature of a single point at n year
$Temp_n$	The temperature of a single point at n year

4 The Model of shoal of fish Variation

4.1 The basic description of fish school characteristics

Atlantic herring (Clupea harengus Figure 4.1.1) is a kind of pelagic fish that feeds on plankton. It is widely distributed in the West and east of the Atlantic Ocean[3]. Atlantic herring can be as long as 17 inches (45.72 cm) and weigh up to 1.5 pounds (0.68 kg). Their body color is dark blue or green blue, and their two outer sides and abdomen are silver. The ventral and anal fins are translucent white[4]. Herring like low temperature, its incubation and development temperature range is 3.3-15 ℃, if the water temperature is higher than 66F, they will not survive. Their minimum survival temperature is 5 ~ 8.5 ℃[5].

Atlantic mackerel (Scomber scombrus Fig. 4.1.2) is a kind of pelagic fish, which lives in the large shoals of the upper ocean. Their sides are silver, purplish, and their bellies are white. Atlantic mackerel is a migratory species that has populations in the East and west of the North Atlantic[6]. Their body length is about 20-40 cm and their weight is 150-400 G. Their hatching is closely related to their growth and water temperature. Their lowest survival temperature is 5 ~ 9.5 ℃[5].

Based on the above description, we regard Atlantic mackerel and herring as the same kind of fish without distinction. Because the Atlantic herring and the Atlantic mackerel in shape, weight, temperature of existence there is a great cross. In addition, studies have shown that the effects of temperature on Atlantic herring and Atlantic mackerel are very similar[5].

Therefore, we choose Atlantic herring to establish the Growth-Time Model of herring.

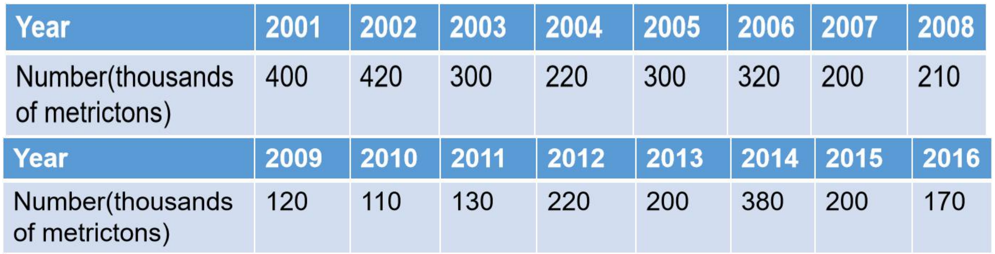
Figure 4.1.1[7]

Figure 4.1.2[8]

4.2 Growth model of fish school

First of all, in order to consider the physical characteristics of the fish population, we studied the relationship between the population number of Atlantic herring and time. Considering the above changes are continuous, we established a logistic growth differential equation.

From the website of the Atlantic States Marine Fisheries Commission[9], we obtained data on the changes in herring population in 2001-2016 (Figure 4.2.1). According to the principle of logistic block growth, we fit a curve of fish population growth(Fig.4.2.2).



Year	2001	2002	2003	2004	2005	2006	2007	2008
Number(thousands of metrictons)	400	420	300	220	300	320	200	210
Year	**2009**	**2010**	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**
Number(thousands of metrictons)	120	110	130	220	200	380	200	170

Figure 4.2.1

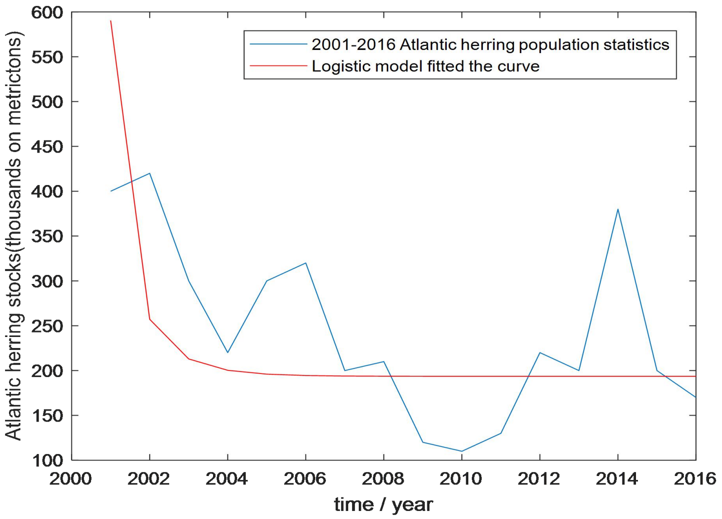


Figure 4.2.2

Logistic equation is： $\frac{dN}{dx} = rN\left(\frac{K-N}{K}\right)$

The initial condition is： $N|_{t=0}=N_0$

Using the method of separating variables, we can get:

t=$\int \frac{K}{rN(K-N)} dN = \int \frac{1}{r}\left(\frac{1}{N} - \frac{1}{K-N}\right) dN = \frac{1}{r}[\ln|N| - \ln|K-N|] + c$

Combined with the initial conditions:

c= - $\frac{1}{r}\ln\frac{N_0}{K-N_0}$

Then：

t=$\frac{1}{r}\ln\frac{N}{K-N}\frac{K-N}{N_0}$

Arrange it, we get:

N=$\frac{K}{1+\frac{K-N_0}{N_0}e^{-rt}}$

After the above derivation, we get the general logistic fish prediction model. In order to facilitate the calculation, we further collate the above model to get the basic form of logistic model:

y=$\frac{1}{a+be^{-t}}$

Next, the logistic model is transformed into a linear model through changes. The process is as follows:

$$y' = \frac{1}{y}$$

$$x' = e^{-t}$$

$$y' = a + bx'$$

Through the calculation of MATLAB program, we get :

a=0.0052 b=-0.0094

That is:

$$y = \frac{1}{0.0052 - 0.0094e^{-t}}$$

5 Temperature analysis of the sea area near Scotland

5.1 Overview of the temperature changes in the North Atlantic from 1985-2015

Atlantic herring and mackerel are pelagic fishes, which are widely found in the Gulf of Maine, the Irish sea, the North Sea, the Norwegian Sea and other North Pacific waters. In the past, mackerel and herring resources in the North Atlantic region were jointly managed by the European Union, Norway, Faroe Islands and other parties according to the agreement. In recent years, due to global warming, many fishing grounds have begun to move north, such as a large number of fish entering the exclusive economic zone of Iceland and the Faroe Islands and its surrounding waters. And in the summer of 2012, mackerel swarmed into Greenland[10].

We use Python web crawler technology to obtain the SST data of the North Atlantic from the National Oceanic and Atmospheric Administration(NOAA) of the United States from 1985 to 2018 and draw the thermodynamical diagram of temperature change with Basemap. As shown in Figure 5.1.1-5.1.4, we analyze the SST changes in the past 33 years with an interval of 11 years. According to the four thermodynamic diagrams, the temperature in the North Pacific is on the rise. Between the equator and N60°, the yellow area obviously expands, the green area moves to higher latitude, and the blue area also tends to move to higher latitude.

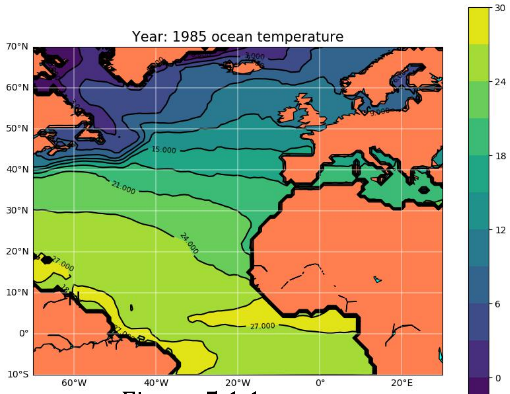


Figure 5.1.1

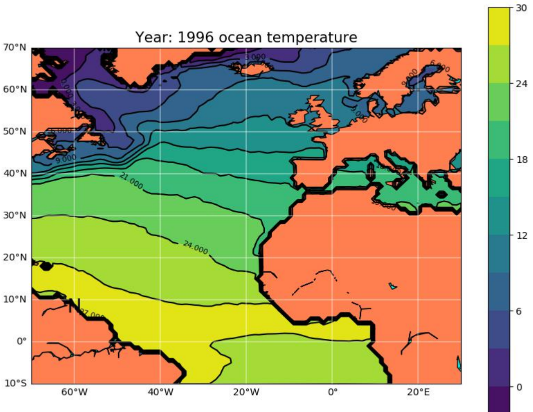


Figure 5.1.2

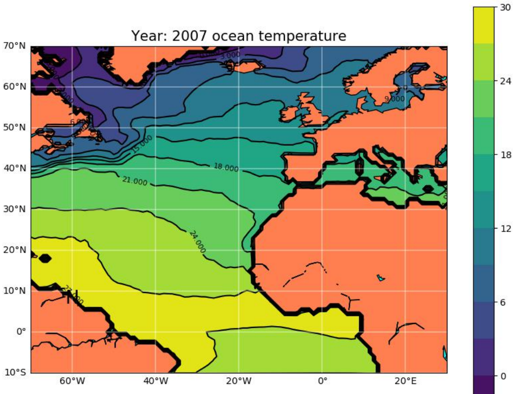


Figure 5.1.3

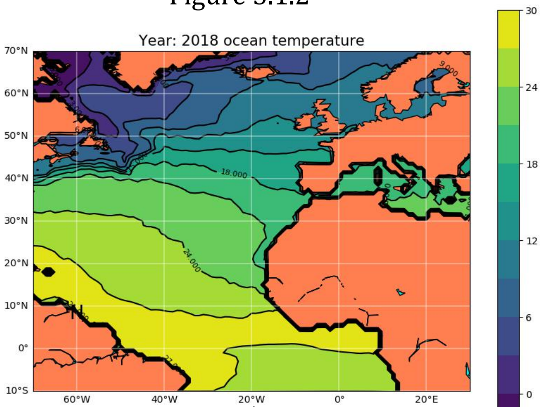


Figure 5.1.4

Then, we use Python as a tool to get the line graph of the mean temperature and time by using the SST average temperature of the North Atlantic. It can be seen from the Fig.5.1.5

that the North Atlantic Ocean temperature changes periodically and has an obvious upward trend.

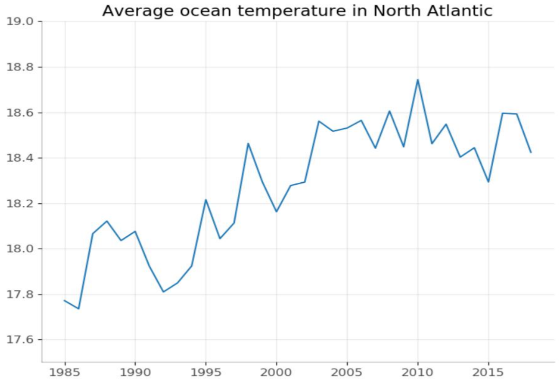


Figure 5.1.5

5.1.1 Temperature changes in the North Sea and Ireland

The North Sea in eastern Scotland and the Irish Sea in the southwest are areas where herring and mackerel are abundant. Therefore, the SST(sea surface temperature) changes in these two places affect the resources of herring and mackerel in Scotland. As shown in Fig.5.1.6, we select point 1(longitude354°latitude53°) and point 2(longitude0 latitude57°) of these two sea areas, and compare the sea water temperature changes in 1985 and 2018, as shown in Fig5.1.7, and then draw a line chart of temperature time changes, as shown in Fig.5.1.8.

The temperature range of herring and mackerel is 3-15 ℃, and we assume that their optimum temperature is 11 ℃. As can be seen from Fig.5.1.8, the temperature of the North Sea and the Irish sea has increased by about 1 ℃ in past 33 years. As can be seen from Fig. 5.1.7, the optimum living temperature zone of the fish group moves to the northwest. Therefore, we can infer that the fish swarm moves to the northwest because of the increase of ocean temperature.

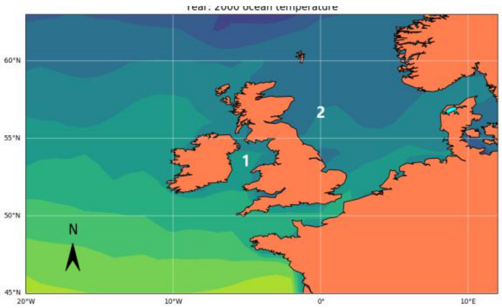


Figure 5.1.6

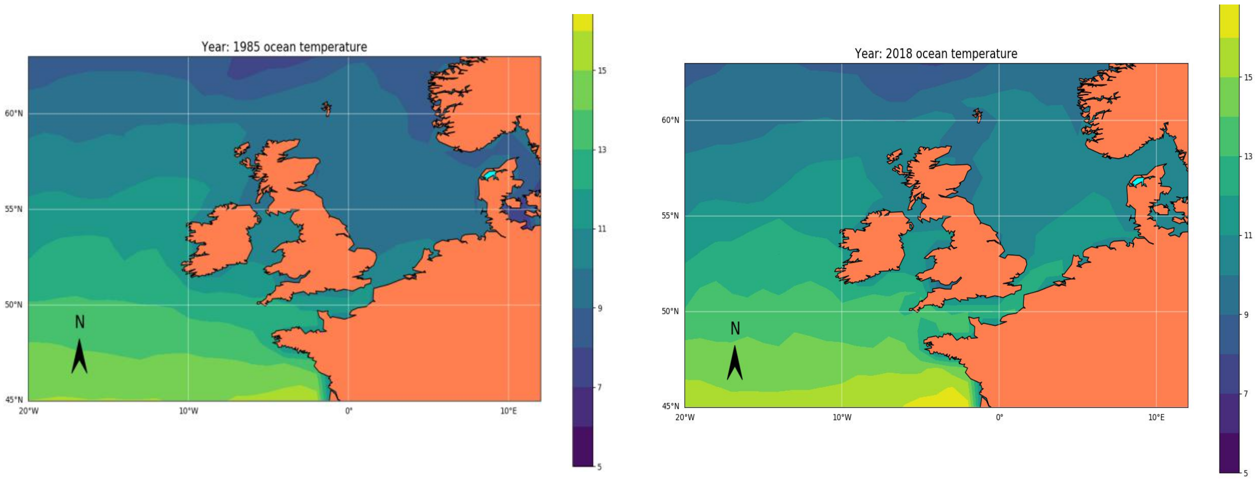


Figure 5.1.7

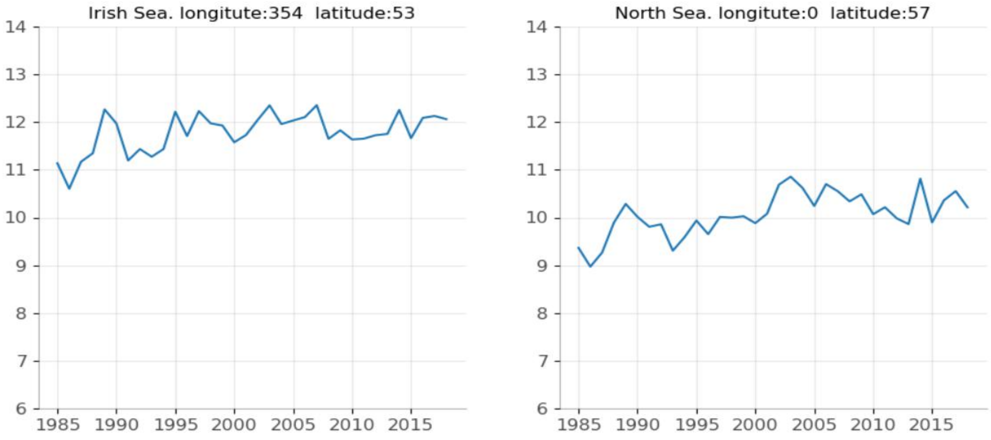


Figure 5.1.8

6 The operation of small fishing companies

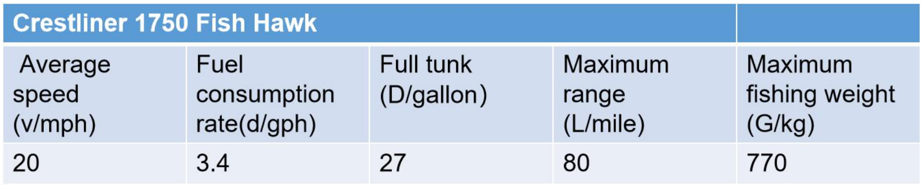
The common characteristics of small fishing companies are small scale, limited capital source and poor risk resistance. The reasonable assumption of company size provides a practical basis for changing the business operation in response to fish migration.

SM-FISHING is a small fishing company based in Peterhead, Scotland. Peterhead is the easternmost port in Scotland and one of the most important ports in the UK. The company has 15 employees, 5 Crestliner 1750 Fish Hawk, which are small fishing boats, boats parameters as shown in the Fig.6.1[11]. According to AAA gas prices website, the price of one gallon of gas is about \$3[12]. According to Alibaba's pricing for frozen Atlantic herring mackerel, we assume that the price of herring mackerel is \$1500 per ton[13]. Assuming that at least two people fishing at a time (n = 2), the company pays \$100 for one person to go out to sea. It is assumed that the fuel in the fuel tank will be exhausted each time, and the fishing yield is related to the ocean temperature and carrying capacity.

Maximum range of small fishing boat : L=D/d*v

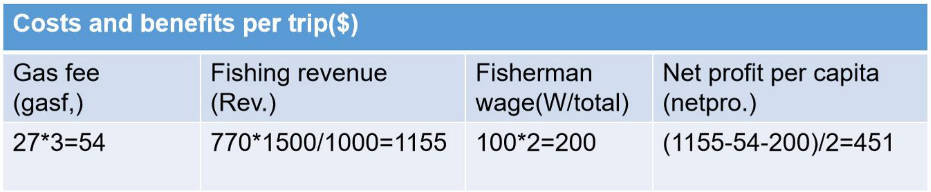
Per capita net profit per time of fishing : *netpro.*=(*Rev.*-*gasf.*-W)/N

According to the above information and formula, we get the information in the table below(Fig.6.2), and the per capita net profit of the sea going is 451 US dollars.



Crestliner 1750 Fish Hawk				
Average speed (v/mph)	Fuel consumption rate(d/gph)	Full tunk (D/gallon)	Maximum range (L/mile)	Maximum fishing weight (G/kg)
20	3.4	27	80	770

Figure 6.1



Costs and benefits per trip($)			
Gas fee (gasf,)	Fishing revenue (Rev.)	Fisherman wage(W/total)	Net profit per capita (netpro.)
27*3=54	770*1500/1000=1155	100*2=200	(1155-54-200)/2=451

Figure 6.2

7 Three predictions of fish school migration

According to the aforementioned of the shoals, which is herring and mackerel are probably influenced by the sea temperature, we assume that the key point of solving this problem convert from hammering out a time when small fishing company not making harvest, to foresee the turning point of the temperature in North Atlantic Ocean when it is too hot to breed this two species of fish.

Therefore, we need to draw the borderline of the boat **Crestliner 1750 Fish Hawk's** voyage and predict the temperature hereabout, and when the temperature go up and overpass the limit of herring and mackerel's survival range, which is about 15 centigrade, there will be no fish anymore. But to make harvest, 13 centigrade is agreeable, that small fish companies can make harvest.

From data, the small boat's maximum distance of voyage is 160 sea mile, considering the return ticket will halve the distance to 80 sea mile, which is nearly 1.27 degree of latitude, no more than 1.9 degree of longitude. we mark it as white circle in figure 7.1.

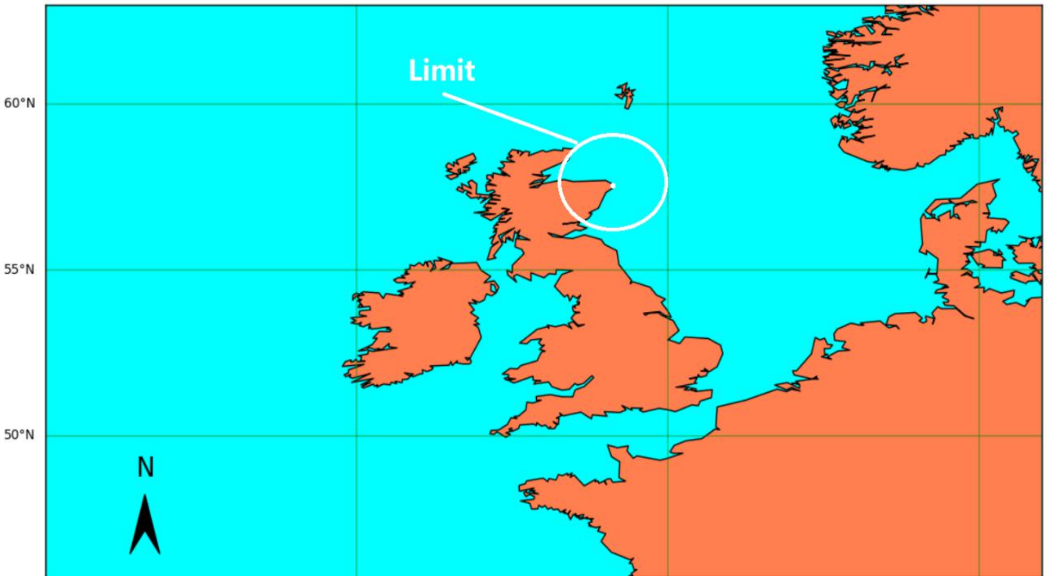
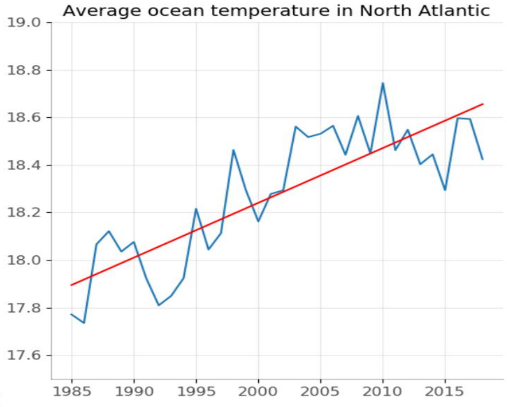


Figure 7.1

To make the unpredictable ocean changing into predictable, in the first step, we would like to show the picture following, this is the average temperature of North Atlantic, so that we presume the function of the temperature increasing is linear. Using linear regression analyze and draw the result in red line(Figure 7.2).



$$\gamma = 0.02304\alpha - 27.83940$$

Figure 7.2

Forecasting the future temperature tendency, or in another word regression, nothing is better than deep learning. However unlike the stock market, the temperature data is too few to do so. Therefore, function fitting is all we could use.

To simulate the randomly and periodically choppy wave of the line chart, firstly, we subtract the multiply of slope and years from temperature every year, so that make this line graph bobbing around beeline of x=0. Calculate the standard deviation and divide 1.5 , and the result is 0.10430.

$$\begin{cases} \gamma = \mathrm{T} - 0.02304 \times \overbrace{\Delta year}^{np.array} \\ std = \dfrac{numpy.std(\gamma)}{1.5} \end{cases}$$

Then, we create a normal distribution with the standard deviation above, adding the multiply of slope and years and drawing the ideal condition graph (Figure 7.3) without interfered by any external factor.

$$\begin{cases} rand = np.random.normal(loc = 0, scale = std, size = (50,)) \\ Future = 0.2304 \times year - 27.8394 + rand \end{cases}$$

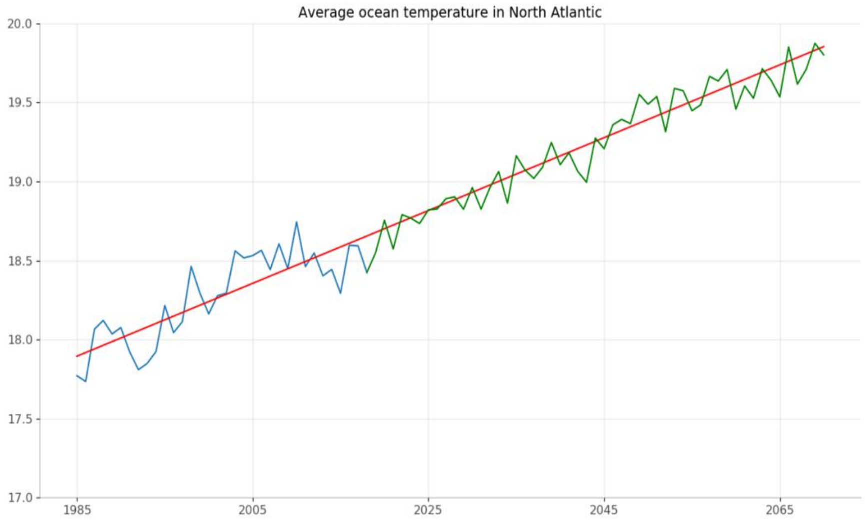


Figure 7.3

From the experience of predict the whole temperature, we trying to calculate each single point's changing and predict it separately. Later the consequence out of hand, turn out that the two point stand shoulder by shoulder become quite discrepancy. After explore our pseudo random algorithm we make it more closely to reality.

In consequence, we calculate each single point as before, extracting the slope of function fitting. What is different is that we use the slope as a parameter only, multiplying it and Delta year, which is the increment. Then, add another random quantity, which is the standard deviation getting from the whole temperature change. Because all of the point were added the same random quantity, the temperature will fluctuate wholly.

To make some improvement, we conclude from Figure 7.2 and Fig.5.18 about Irish Sea, that the change in a single point is more corrugate than the whole. So we calculate the standard deviation each point, dividing the whole standard deviation, computing the square root of the division result as a coefficient of the random number in order to make the corrugate more approximate to each point. What's more, slope of each point also need to make a limit between 0.8~1.2 times of the whole point.

$$\begin{cases} coef = \sqrt{whloleStd/singleStd} \\ T_n = rand[n-2019] \times coef + k \times (n-2018) + T_{2018} \end{cases} \quad ①$$

Note*:

wholeStd: the standard deviation calculate from the whole North Atlantic.
singleStd: the standard deviation calculate from each single point
rand[n-2019]: an array of random number getting from the average temperature
n: year
k: the slope of linear regression calculate from temperature of single point
T_{2018}: the temperature of a single point in 2018

7.1 The Ideal Condition of SST Variation

In the very ideal condition, temperature in North Sea will go as same as before. Using the same random arithmetic, we create the future water temperature guesstimate below. Visually from figure 7.4 we can find that around the year 2122, the temperature will raise and surpass red flag so that small fish company cannot harvest. And the thermodynamics figure7.5 show that the port are nearly besieged by deadly hot water.

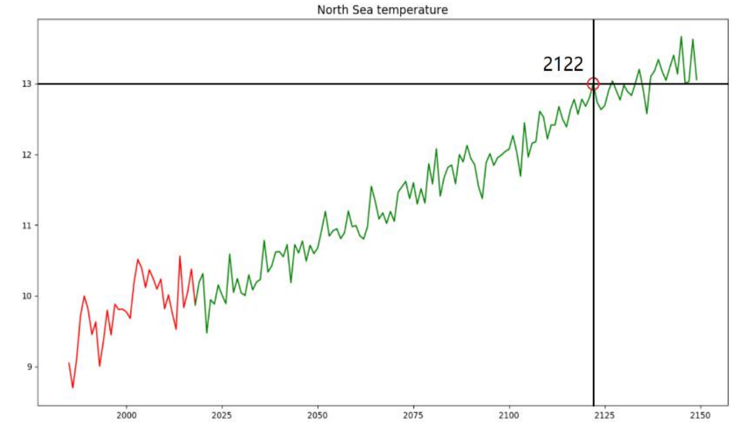


Figure 7.4

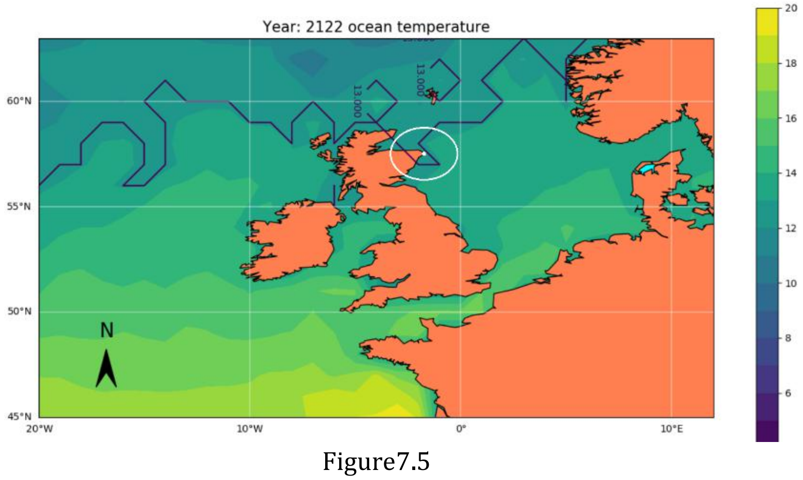


Figure7.5

7.2 Optimal Condition of SST Variation

According to the Paris Agreement signed in 2015, the temperature rise will and must be controlled in the limit of 2 degree in this century. To performance the last straw of rein up the galloping steed of temperature, at least, 55 party of the agreement must pare the carbon dioxide release below 45 percent[14]. And we assume that will be done successfully. We predict that air temperature is well proportion with ocean temperature, and the rate of increasing only relate to the carbon release.

$$\begin{cases} \Delta AirT = \alpha CarbonR \\ \Delta SeaT = \beta \Delta AirT \end{cases}$$

Therefore, if carbon release become 0.45 times as before, the rate of temperature increasing will be like also. And here we continually assume the rate of decrease carbon release will be linear, which means from year 2018(simplify the model) to 2100 the rate of increasing will be slow down. So formula ① will be improve as following:

$$\begin{cases} \lambda = 1 - \dfrac{1-0.45}{2100-2018} \times (n-2018) \\ Temp_n = k \times \lambda + T_{n-1} \end{cases}$$

After this recursive calculative formula, we also need to add a disturbance to make the result more randomly:

$$\begin{cases} coef = \sqrt{whloleStd/singleStd} \\ T_n = rand[n-2019] + Temp_n \end{cases}$$

The graph(Fig.7.7) is about the average ocean temperature modal in North Sea. From this graph we can conclude that the sea temperature will not raising up too quick to harvest the fish, at least before the year of 2150. It will be meaningless to predict thereafter. In that way, we believe in the optimal condition, small company can always catch enough fish.

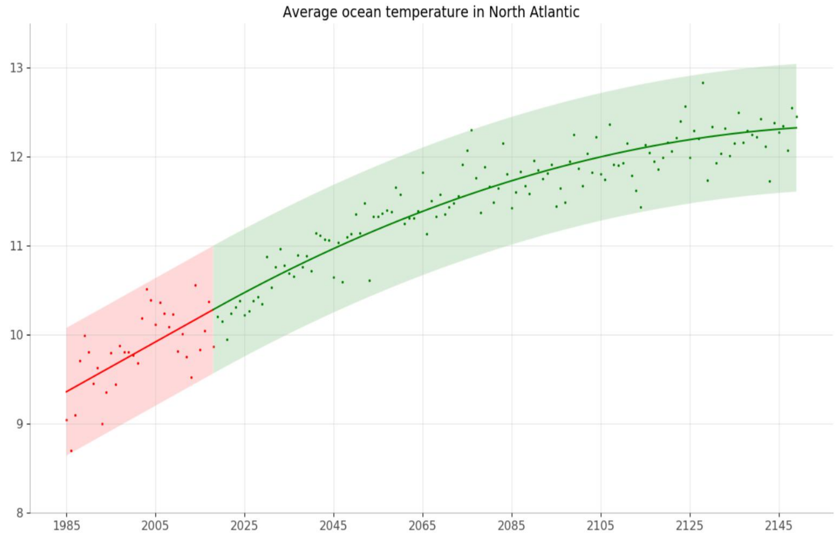


Figure 7.7

7.3 Extreme Condition of SST Variation

Base on the optimal condition we have discussed about, we take a step further. Because of the chain reaction of the global warming, countless species go extinct without our knowing. Even forest, cannot waiting to the year when it can adapt climate change and voraciously absorb the thick carbon dioxide penetrate throw the glade, withered group by group. The higher of the temperature, the less of the forest, meanwhile the higher rate of temperature increasing. So, we make some basic assumption to simplify our calculation.

First, is the same assumption we have made in the optimal condition. Additionally we assume that the Delta temperature of air equal to Delta temperature of sea.

Second, each degree of temperature increasing, will cause 15 percent of trees dead. But at present, all of the trees are still alive.

Third, trees can absorb carbon dioxide. But if some of the trees dead, the whole ability of neutralize carbon emission will pare away.

$$\begin{cases} \Delta SeaT = \Delta AirT \\ \Delta TreeAbsorb = 1 - 0.15 \times \Delta AirT \\ \Delta Relase = 1 - \dfrac{1-0.45}{2100-2018} \times (n - 2018) \\ Temp_n = k \times \lambda + T_{n-1} + \beta \times (1 - \Delta TreeAbsorb) \end{cases}$$

To continually simplify this model, we assume β in the formula is 0.04, and that can be prove reasonably later.

In this condition, even if we try painstakingly, we still fail to curb the tendency of the increasing temperature. As the forest break down, more carbon seal up in the wood will be rejoin with oxygen and become carbon dioxide to accelerate the road of undoing. Around the year of 2105, the temperature will surpass the limit and fisherman cannot harvest anymore

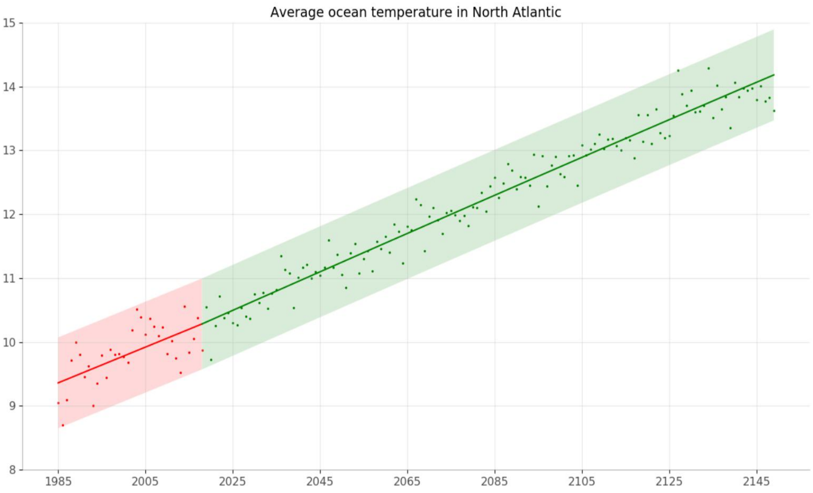


Figure 7.8

8 Analysis of future management strategy in three conditions

Based on the prediction of the sea surface temperature(SST) changes in the next 50 years, we believe that under the optimal and most extreme conditions, the company does not need to change its business strategy. Under the optimal conditions, the sea surface temperature of Scotland's appendages will not change significantly, and the suitable temperature zone of fish school will not move too much. Under the most extreme conditions, the suitable temperature zone for the survival of fish groups has a significant northward migration. However, the deviation is too large to be suitable for fishing. Therefore, the change of the company's business strategy in this state will not be considered.

Under the most ideal conditions, we provide suggestions for the improvement of the company's business strategy through assumptions.

The First Strategy:
the small companies do not change any business strategy, but still fish according to the original model. According to the marine thermodynamic map of 2070(Figure.), the temperature of the fishing area was 12 ℃. The optimum survival temperature of the fish group is 10 ℃. Under this 12℃, the maximum fishing capacity is reduced by 50% to 385 kg. The net profit per capita for each trip to the sea is $350.
The Second Strategy:
The small fishing company can sell five small fishing boats, Crestliner 1750 Fish Hawk, and buy one with refrigeration boat, Crestliner 1780 Fish Haskek as shown in the Figure 8.2, which is three times the size of the original fishing boat. According to the thermodynamic diagram of 2070(Fig.8.1), we assume that the company is at point P(2°W,57°N) and the new optimal fishing site is Q(7°W,62°N). The distance between PQ is about 416 miles from geographical knowledge and Pythagorean theorem. As a result of the increase of the maximum mileage and the maximum carrying capacity, the number of people going to sea each time is at least 6, and the salary per person is increased to $300. At this time, the cost profit of each sea trip is as shown in the Fig.8.3, and the net profit per capita is 630 US dollars.

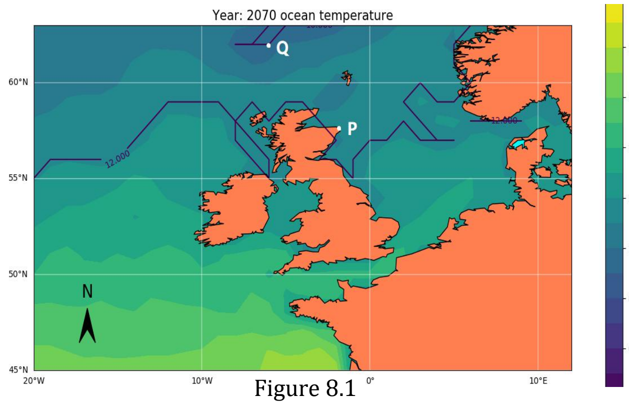
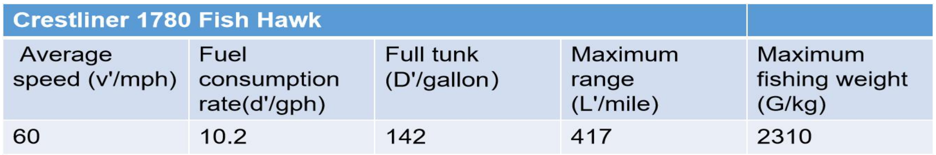
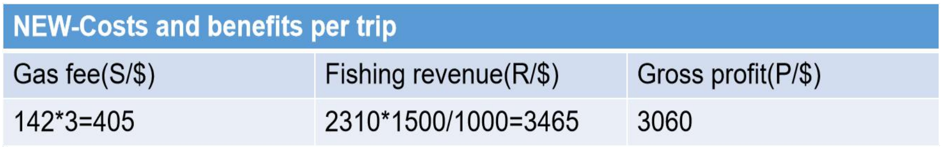


Figure 8.1



Crestliner 1780 Fish Hawk				
Average speed (v'/mph)	Fuel consumption rate(d'/gph)	Full tunk (D'/gallon)	Maximum range (L'/mile)	Maximum fishing weight (G/kg)
60	10.2	142	417	2310

Figure 8.2



NEW-Costs and benefits per trip		
Gas fee(S/$)	Fishing revenue(R/$)	Gross profit(P/$)
142*3=405	2310*1500/1000=3465	3060

Figure 8.3

All in all, for the normal development of the company, the per capita net profit of 2070 should be equal to or higher than the past. Without changing the business strategy, the per capita net profit will be reduced by 22%, while after buying new fishing boats, the per capita net profit will be increased by 40%. Therefore, under the best conditions, the company should buy new fishing boats to ensure the operation of the company.

9 Analysis of cross territorial fishing

According to content 8, the new catch point Q is located at 7 ° W, 62 ° N. According to the latitude and longitude of Google map, Q belongs to the territorial sea area of Faroe Islands(Figure 9.1).

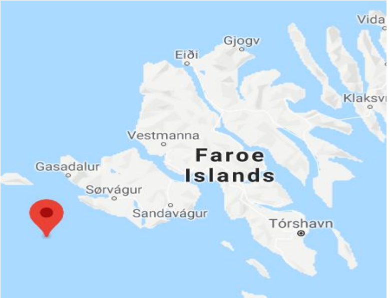


Figure 9.1

The Faroe Islands is a semi autonomous region, which belongs to The Danish. However, unlike Denmark, the Faroe Islands are not part of the European Union and all trade with EU countries is governed by specially negotiated treaties drawn up in consultation and co-operation with the Danish foreign ministry[15]. The fishing industry has always been an important source of economic resources for the Faroe Islands, accounting for about 20% of the total economy. The labor force related to fishing accounts for 15% of the total. Since the announcement of brexit, the Faroe Islands and Britain have reached a new fishing agreement. "The UK remains one of the most important markets for Faroese seafood exports. A new trade agreement with the UK to ensure future exports is therefore a priority of the Faroese government," it wrote in a statement [16].

However, the archipelago has rejected the Common Fisheries Policy, a set of rules that manage European fishing fleets, to freely decide its own quotas[16]. This means that the number of fish caught will be determined by the Faroe Islands alone. But we haven't found a policy on whether British fishermen can enter the Faroe Islands for commercial fishing. According to International Law, ships of other countries have the right of innocent passage when they enter the territorial waters of other countries. The right of innocent passage does not apply to submerged submarines or to aircraft, nor does it include a right to fish[17]. This shows that British ships cannot enter the Faroe Islands for commercial fishing, according to current International Law.

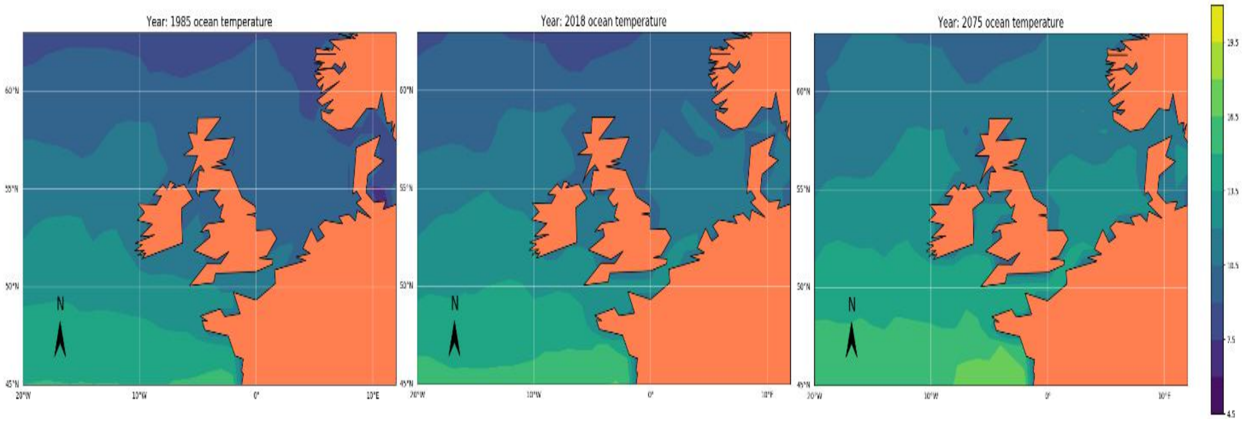
So, we don't think the suggestion that we've proposed for small fishing companies is practical when it comes to territorial issues.

10 An Article to Hook Line and Sinker Magazine

Beware the Temperature

When I was in the year of playing slingshot, nothing is more joyful than making a snowman, rambling around the lane covered with white, look up at the mountain capped by a snow quilt. But now, it is over. Not because I come to an age and lose my childlike mindset, but due to the global warming. A global change, or in another word upheaval, nowhere can wriggle out of it, not in my hometown Beijing, not in Scottish, even not in the South Pole. Catastrophe happened more and more frequent, the conflagration in Australia, the risen sea level and the levee in Maldives. Arid turn oasis to desert, creek to wadi. Yet deluge destroy the crop and the dam. This extreme weather I can list for you a whole night.

If you are an innocent fisherman, enjoying your lighthearted time rowing on the sea, even without knowing what happen out there, then you need to beware. You may inquest “how so? Global warming, nothing to do with my business“. Then you are wrong. Just look the picture below, yellow stand for hot. Ocean around your home are becoming hotter and hotter. And the most adaptive temperature for herring and mackerel is 10 centigrade, which means sooner or later you will not catch fish anymore if you wouldn’t do any preparation.


Year: 1985 ocean temperature
Year: 2018 ocean temperature
Year: 2075 ocean temperature
60°N
55°N
50°N
45°N
20°W
10°W
0°
10°E
N
19.5
16.5
13.5
10.5
7.5
4.5

Whether you would like to cultivate your progeny not to inherit your small boat and profession, or you would like to think the big picture and change your vessel to a bigger one in order to fit the change, you’d better make it fast, time wait for no man. But if you choose the later, in some point of view, we could help you.

Crestliner 1780 Fish Hawk will be an agreeable choice for you. The scale of this ship is three times than your small ship, and it can sail to further cooler area, which can allow you catching more profitable fish in the background of global temperature increasing, while at the same time other fisherman can only covet your exquisite cruise ship. According to our precisely calculate, it won’t take you too long to recovery you cost so don’t worry, and after that, ceaseless coin will swell up your little pocket and lead you to the fat city. I swear I never intent to make an advertisement of it, but just in passing, it is really awesome.

Picture from https://www.yachting-rental.com/activities/fishing

11 Strengths and Weaknesses

11.1 Strengths

In terms of fish growth models, Logistic growth model can be used to calculate the intra-species resource competition. Therefore it can be well used in the prediction of fish shoal. In this problem, we established the Logistic equation, which is $\frac{dN}{dx} = rN\left(\frac{K-N}{K}\right)$, with Initial conditions $N|_{t=0}$=N_0. Second, the model is simple. In the last stage of mathematical derivation, we got the simplify form of the Logistic Equation, which is y=$\frac{1}{a+be^{-t}}$. Besides the exponent, there is only one extra parameter and will be easy to program. Finally, the model can be extend to a wide range of applications, such as in population prediction.

For sea surface temperature prediction, we use thermal imaging to determine the location of future schools of fish. First of all, it can be considered more accurate. Because as mentioned above, shoals of fish are directly related to ocean temperature. Second, it's more intuitive. The thermodynamic diagram and the line chart make it easy to point out where is the proper area of fish, even without doing a lot of calculations. What's more, our data is download from National Oceanic and Atmospheric Administration. With proper extrapolate and calculate, the future we predict is more convincing than blind speculation.

As for the business strategy,we take the actual small fishing company as a reference, and make the company situation assumption in this paper. Then, we modify and simplify the boat model according to the existing fishing boats on the market. In this way, the rationality of the hypothesis and the accuracy of the result are improved to a certain extent.

11.2 Weaknesses

For the fish growth models,Firstly, we only considered the population of fish (N), the natural growth (r), the environmental capacity in the North Atlantic (k), to build the basic model. What we didn’t put into consideration, for an example, is the increasing levels of carbon dioxide which will cause the change of seawater acidity. In that way, an error between the predicted result and the reality is unavoidable. Finally, we searched the data from the network, but we cannot evaluate the reliability of it.

In terms of sea surface temperature prediction, we need to fit the data to predict future temperature from the history data. Due to the different assumptions, the future forecast may become entirely different. And it also affected by many external factors, for example politics and economics.

In terms of business strategy, we simplified the forms of companies and fishing boats to facilitate calculation and suggestions. However, in real life, there are many factors that affect the company's decision-making, such as policies, equipment maintenance costs and so on. Our strategy can only serve as a reference, because the actual situation is much more complex than the hypothetical situation

12 Reference

[1]S. Chen, X.Chen, “Study on the interannual change of fish community structure in the North Sea fishing ground of the Northeast Atlantic Ocean”, Journal of Oceanography (Chinese version)

[2]*WMO Statement On The Status Of The Global Climate In 2012*. 2012. Accessed 14 Feb 2020.

[3] "Atlantic Herring". *Cs.Mcgill.Ca*, 2020, https://www.cs.mcgill.ca/~rwest/wikispeedia/wpcd/wp/a/Atlantic_herring.htm.

[4] "Clupea Harengus (Melker)". *Animal Diversity Web*, 2020, https://animaldiversity.org/accounts/Clupea_harengus/#physical_description. Accessed 17 Feb 2020.

[5]Turner, Sara M. et al. "Using Habitat Association Models To Predict Alewife And Blueback Herring Marine Distributions And Overlap With Atlantic Herring And Atlantic Mackerel: Can Incidental Catches Be Reduced?". *ICES Journal Of Marine Science*, vol 73, no. 7, 2015, pp. 1912-1924. *Oxford University Press (OUP)*, doi:10.1093/icesjms/fsv166.

[6]"Atlantic Mackerel". *NOAA*, 2020, https://www.fisheries.noaa.gov/species/atlantic-mackerel.

[7]"About - MSC Fisheries". *Fisheries.Msc.Org*, 2020, https://fisheries.msc.org/en/fisheries/from-nord-north-sea-and-eastern-channel-pelagic-trawl-herring/about/.

[8]"Mackerel Piccata | Marine Stewardship Council". *Msc.Org*, 2020, https://www.msc.org/what-you-can-do/sustainable-fish-seafood-recipes/recipe/mackerel-piccata.

[9]"Species - Atlantic States Marine Fisheries Commission". *Asmfc.Org*, 2020, http://www.asmfc.org/species/atlantic-herring.

[10]The dispute on fishery resources caused by climate change——mackerel war intensified in Europe 李励年,林龙山,缪圣赐,LI Li-nian,LIN Long-shan,MIAO Sheng-ci- Modern Fisheries Information,vol(23). 2013

[11]"New Crestliner 1750 Fish Hawk | Aluminum Multi Species Fishing Boats". *Crestliner.Com*, 2020, https://www.crestliner.com/fish-hawk/1750-fish-hawk/#full-specs.

[12]"AAA Gas Prices". *AAA Gas Prices*, 2020, https://gasprices.aaa.com/. Accessed 17 Feb 2020.

[13]"Frozen Atlantic Herring, Frozen Atlantic Herring Suppliers And Manufacturers At Alibaba.Com". *Alibaba.Com*, 2020, https://www.alibaba.com/showroom/frozen-atlantic-herring.html. Accessed 18 Feb 2020.

[14] *Unfccc.Int*, 2020, https://unfccc.int/process-and-meetings/the-paris-agreement/what-is-the-paris-agreement.

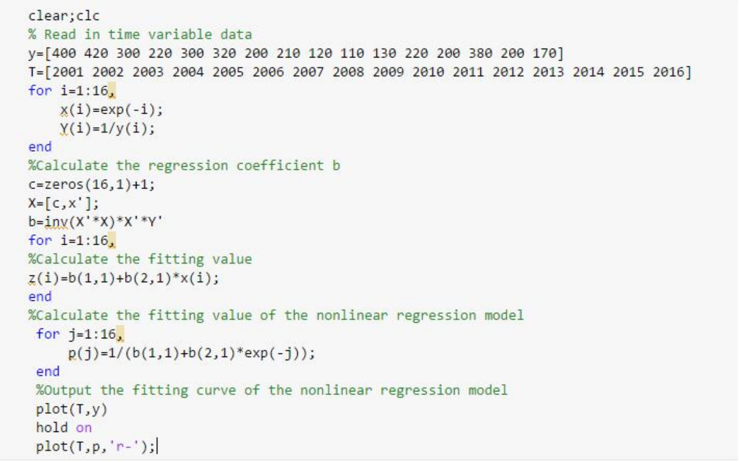
[15]"Government & Politics Of The Faroe Islands - Visit Faroe Islands". *Visit Faroe Islands*, 2020, https://visitfaroeislands.com/about/history-governance-economy/government-politics/. Accessed 17 Feb 2020.

[16]*Thelocal.Dk*, 2020, https://www.thelocal.dk/20170405/faroe-islands-eye-fishing-agreement-with-uk-after-brexit. Accessed 18 Feb 2020.

[17]"Territorial Waters | International Law". Encyclopedia Britannica, 2020, https://www.britannica.com/topic/territorial-waters.

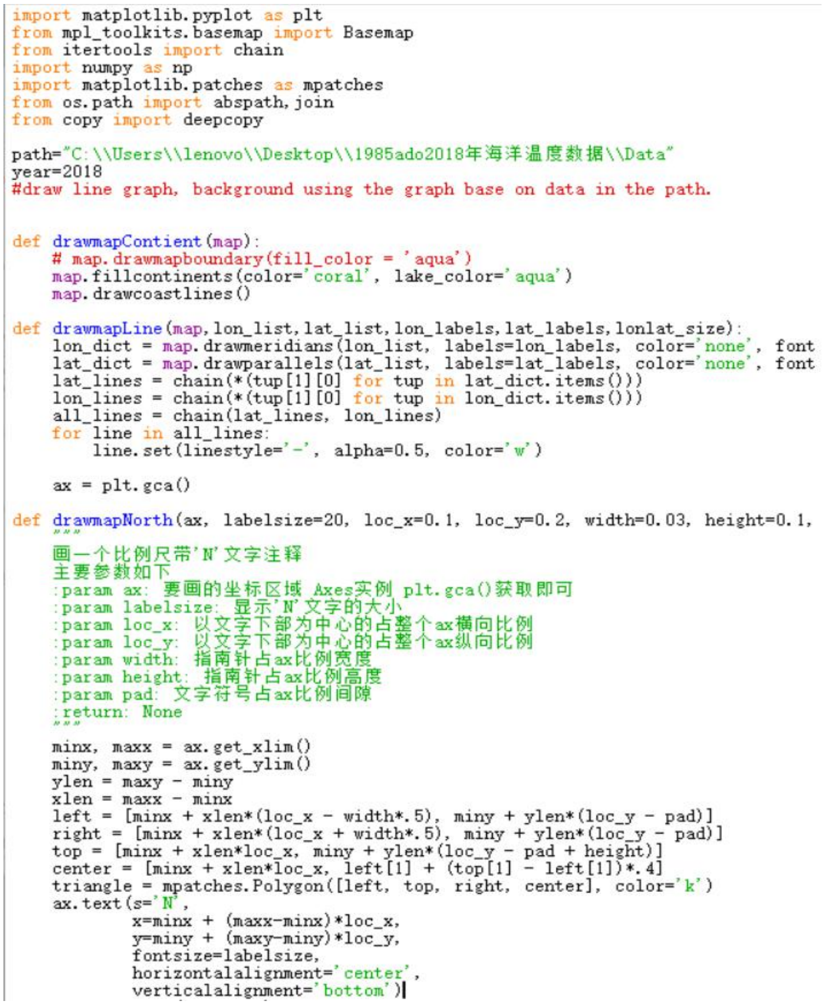
13 Appendix

Code1.Fish growth model.MATLAB

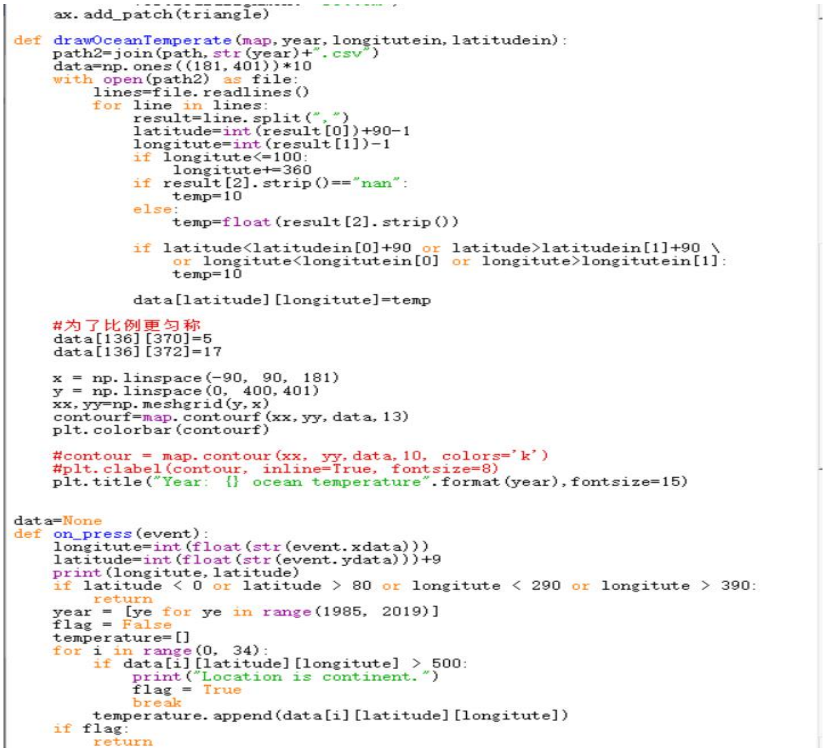


```
clear;clc
% Read in time variable data
y=[400 420 300 220 300 320 200 210 120 110 130 220 200 380 200 170]
T=[2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016]
for i=1:16,
    x(i)=exp(-i);
    Y(i)=1/y(i);
end
%Calculate the regression coefficient b
c=zeros(16,1)+1;
X=[c,x'];
b=inv(X'*X)*X'*Y'
for i=1:16,
%Calculate the fitting value
z(i)=b(1,1)+b(2,1)*x(i);
end
%Calculate the fitting value of the nonlinear regression model
 for j=1:16,
     p(j)=1/(b(1,1)+b(2,1)*exp(-j));
 end
 %Output the fitting curve of the nonlinear regression model
 plot(T,y)
 hold on
 plot(T,p,'r-');
```

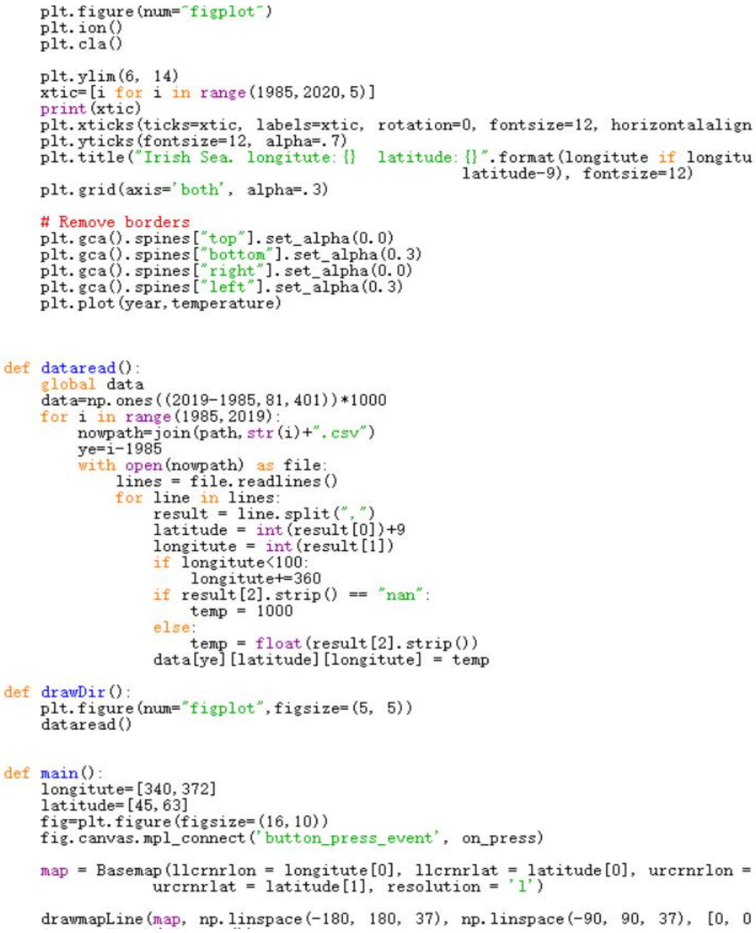
Code2. Click on the map to draw a line chart



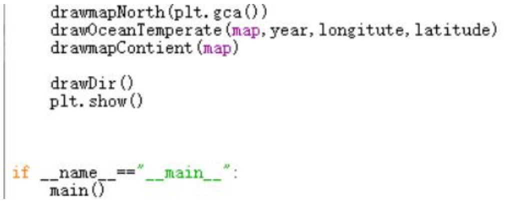
```
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from itertools import chain
import numpy as np
import matplotlib.patches as mpatches
from os.path import abspath, join
from copy import deepcopy

path="C:\\Users\\lenovo\\Desktop\\1985ado2018年海洋温度数据\\Data"
year=2018
#draw line graph, background using the graph base on data in the path.


def drawmapContient(map):
    # map.drawmapboundary(fill_color = 'aqua')
    map.fillcontinents(color='coral', lake_color='aqua')
    map.drawcoastlines()

def drawmapLine(map,lon_list,lat_list,lon_labels,lat_labels,lonlat_size):
    lon_dict = map.drawmeridians(lon_list, labels=lon_labels, color='none', font
    lat_dict = map.drawparallels(lat_list, labels=lat_labels, color='none', font
    lat_lines = chain(*(tup[1][0] for tup in lat_dict.items()))
    lon_lines = chain(*(tup[1][0] for tup in lon_dict.items()))
    all_lines = chain(lat_lines, lon_lines)
    for line in all_lines:
        line.set(linestyle='-', alpha=0.5, color='w')

    ax = plt.gca()

def drawmapNorth(ax, labelsize=20, loc_x=0.1, loc_y=0.2, width=0.03, height=0.1,
    """
    画一个比例尺带'N'文字注释
    主要参数如下
    :param ax: 要画的坐标区域 Axes实例 plt.gca()获取即可
    :param labelsize: 显示'N'文字的大小
    :param loc_x: 以文字下部为中心的占整个ax横向比例
    :param loc_y: 以文字下部为中心的占整个ax纵向比例
    :param width: 指南针占ax比例宽度
    :param height: 指南针占ax比例高度
    :param pad: 文字符号占ax比例间隙
    :return: None
    """
    minx, maxx = ax.get_xlim()
    miny, maxy = ax.get_ylim()
    ylen = maxy - miny
    xlen = maxx - minx
    left = [minx + xlen*(loc_x - width*.5), miny + ylen*(loc_y - pad)]
    right = [minx + xlen*(loc_x + width*.5), miny + ylen*(loc_y - pad)]
    top = [minx + xlen*loc_x, miny + ylen*(loc_y - pad + height)]
    center = [minx + xlen*loc_x, left[1] + (top[1] - left[1])*.4]
    triangle = mpatches.Polygon([left, top, right, center], color='k')
    ax.text(s='N',
            x=minx + (maxx-minx)*loc_x,
            y=miny + (maxy-miny)*loc_y,
            fontsize=labelsize,
            horizontalalignment='center',
            verticalalignment='bottom')
```



```
    ax.add_patch(triangle)

def drawOceanTemperate(map,year,longitutein,latitudein):
    path2=join(path,str(year)+".csv")
    data=np.ones((181,401))*10
    with open(path2) as file:
        lines=file.readlines()
        for line in lines:
            result=line.split(",")
            latitude=int(result[0])+90-1
            longitute=int(result[1])-1
            if longitute<=100:
                longitute+=360
            if result[2].strip()=="nan":
                temp=10
            else:
                temp=float(result[2].strip())

            if latitude<latitudein[0]+90 or latitude>latitudein[1]+90 \
                or longitute<longitutein[0] or longitute>longitutein[1]:
                temp=10

            data[latitude][longitute]=temp

    #为了比例更匀称
    data[136][370]=5
    data[136][372]=17

    x = np.linspace(-90, 90, 181)
    y = np.linspace(0, 400,401)
    xx,yy=np.meshgrid(y,x)
    contourf=map.contourf(xx,yy,data,13)
    plt.colorbar(contourf)

    #contour = map.contour(xx, yy,data,10, colors='k')
    #plt.clabel(contour, inline=True, fontsize=8)
    plt.title("Year: {} ocean temperature".format(year),fontsize=15)


data=None
def on_press(event):
    longitute=int(float(str(event.xdata)))
    latitude=int(float(str(event.ydata)))+9
    print(longitute,latitude)
    if latitude < 0 or latitude > 80 or longitute < 290 or longitute > 390:
        return
    year = [ye for ye in range(1985, 2019)]
    flag = False
    temperature=[]
    for i in range(0, 34):
        if data[i][latitude][longitute] > 500:
            print("Location is continent.")
            flag = True
            break
        temperature.append(data[i][latitude][longitute])
    if flag:
        return
```

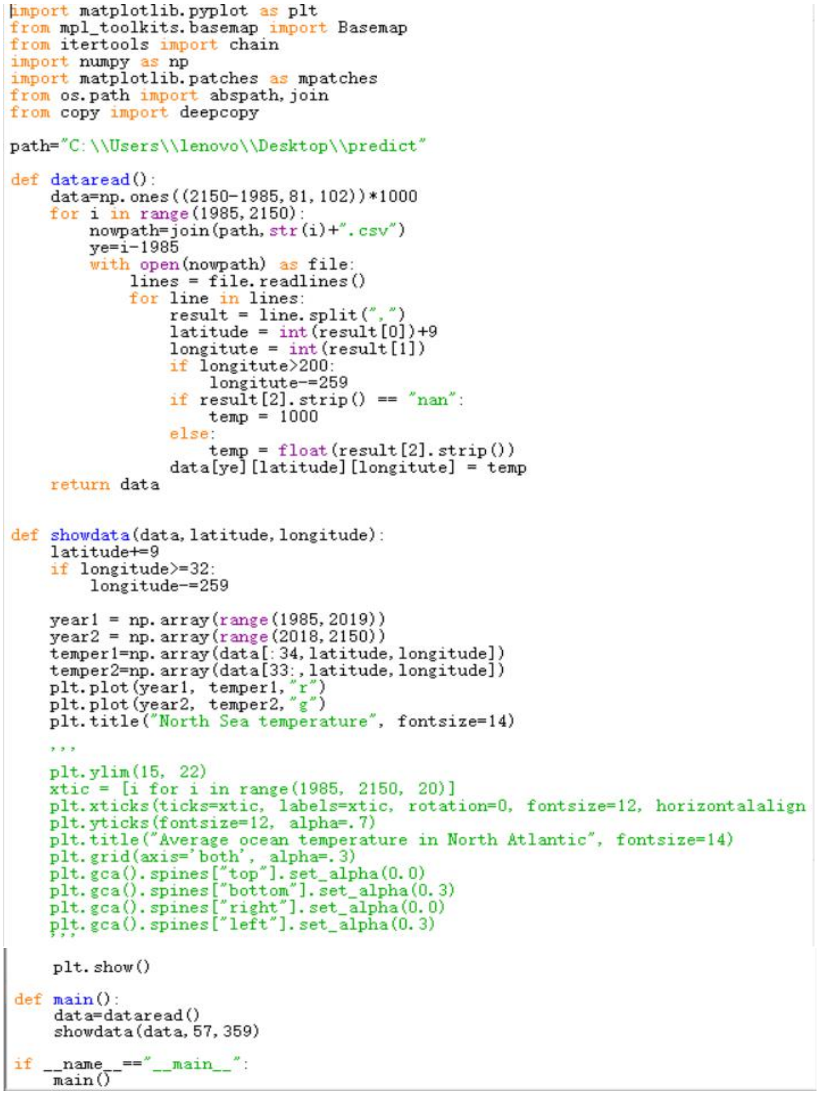


```
    plt.figure(num="figplot")
    plt.ion()
    plt.cla()

    plt.ylim(6, 14)
    xtic=[i for i in range(1985,2020,5)]
    print(xtic)
    plt.xticks(ticks=xtic, labels=xtic, rotation=0, fontsize=12, horizontalalign
    plt.yticks(fontsize=12, alpha=.7)
    plt.title("Irish Sea. longitute:{}  latitude:{}".format(longitute if longitu
                                latitude-9), fontsize=12)
    plt.grid(axis='both', alpha=.3)

    # Remove borders
    plt.gca().spines["top"].set_alpha(0.0)
    plt.gca().spines["bottom"].set_alpha(0.3)
    plt.gca().spines["right"].set_alpha(0.0)
    plt.gca().spines["left"].set_alpha(0.3)
    plt.plot(year,temperature)



def dataread():
    global data
    data=np.ones((2019-1985,81,401))*1000
    for i in range(1985,2019):
        nowpath=join(path,str(i)+".csv")
        ye=i-1985
        with open(nowpath) as file:
            lines = file.readlines()
            for line in lines:
                result = line.split(",")
                latitude = int(result[0])+9
                longitute = int(result[1])
                if longitute<100:
                    longitute+=360
                if result[2].strip() == "nan":
                    temp = 1000
                else:
                    temp = float(result[2].strip())
                data[ye][latitude][longitute] = temp

def drawDir():
    plt.figure(num="figplot",figsize=(5, 5))
    dataread()


def main():
    longitute=[340,372]
    latitude=[45,63]
    fig=plt.figure(figsize=(16,10))
    fig.canvas.mpl_connect('button_press_event', on_press)

    map = Basemap(llcrnrlon = longitute[0], llcrnrlat = latitude[0], urcrnrlon =
                urcrnrlat = latitude[1], resolution = 'l')

    drawmapLine(map, np.linspace(-180, 180, 37), np.linspace(-90, 90, 37), [0, 0
```



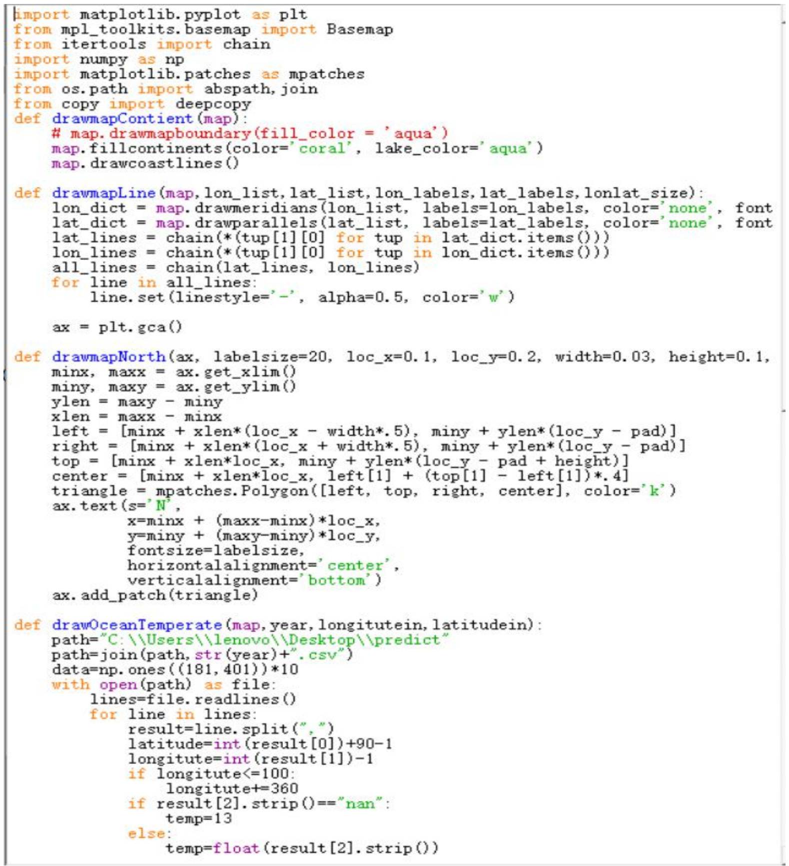
```
    drawmapNorth(plt.gca())
    drawOceanTemperate(map,year,longitute,latitude)
    drawmapContient(map)

    drawDir()
    plt.show()


if __name__=="__main__":
    main()
```

Code3. Draw known line graphs based on coordinates and years

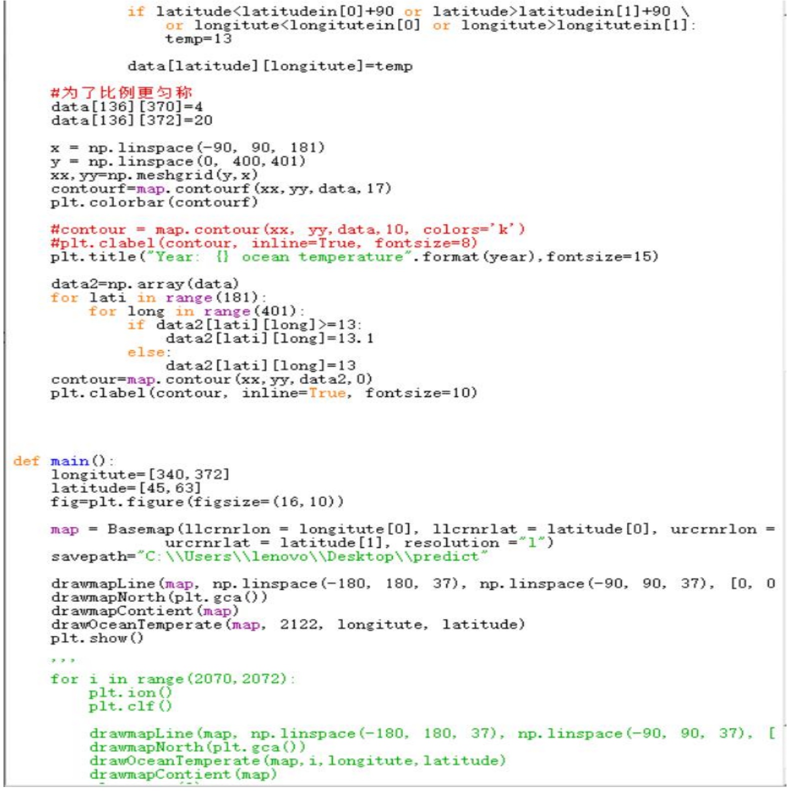


```
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from itertools import chain
import numpy as np
import matplotlib.patches as mpatches
from os.path import abspath,join
from copy import deepcopy

path="C:\\Users\\lenovo\\Desktop\\predict"

def dataread():
    data=np.ones((2150-1985,81,102))*1000
    for i in range(1985,2150):
        nowpath=join(path,str(i)+".csv")
        ye=i-1985
        with open(nowpath) as file:
            lines = file.readlines()
            for line in lines:
                result = line.split(",")
                latitude = int(result[0])+9
                longitute = int(result[1])
                if longitute>200:
                    longitute-=259
                if result[2].strip() == "nan":
                    temp = 1000
                else:
                    temp = float(result[2].strip())
                data[ye][latitude][longitute] = temp
    return data


def showdata(data,latitude,longitude):
    latitude+=9
    if longitude>=32:
        longitude-=259

    year1 = np.array(range(1985,2019))
    year2 = np.array(range(2018,2150))
    temper1=np.array(data[:34,latitude,longitude])
    temper2=np.array(data[33:,latitude,longitude])
    plt.plot(year1, temper1,"r")
    plt.plot(year2, temper2,"g")
    plt.title("North Sea temperature", fontsize=14)

    '''
    plt.ylim(15, 22)
    xtic = [i for i in range(1985, 2150, 20)]
    plt.xticks(ticks=xtic, labels=xtic, rotation=0, fontsize=12, horizontalalign
    plt.yticks(fontsize=12, alpha=.7)
    plt.title("Average ocean temperature in North Atlantic", fontsize=14)
    plt.grid(axis='both', alpha=.3)
    plt.gca().spines["top"].set_alpha(0.0)
    plt.gca().spines["bottom"].set_alpha(0.3)
    plt.gca().spines["right"].set_alpha(0.0)
    plt.gca().spines["left"].set_alpha(0.3)
    '''

    plt.show()

def main():
    data=dataread()
    showdata(data,57,359)

if __name__=="__main__":
    main()
```

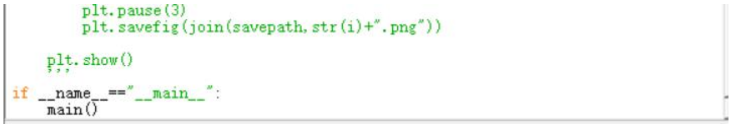
Code4. Thermodynamics figure



```
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
from itertools import chain
import numpy as np
import matplotlib.patches as mpatches
from os.path import abspath,join
from copy import deepcopy
def drawmapContient(map):
    # map.drawmapboundary(fill_color = 'aqua')
    map.fillcontinents(color='coral', lake_color='aqua')
    map.drawcoastlines()

def drawmapLine(map,lon_list,lat_list,lon_labels,lat_labels,lonlat_size):
    lon_dict = map.drawmeridians(lon_list, labels=lon_labels, color='none', font
    lat_dict = map.drawparallels(lat_list, labels=lat_labels, color='none', font
    lat_lines = chain(*(tup[1][0] for tup in lat_dict.items()))
    lon_lines = chain(*(tup[1][0] for tup in lon_dict.items()))
    all_lines = chain(lat_lines, lon_lines)
    for line in all_lines:
        line.set(linestyle='-', alpha=0.5, color='w')

    ax = plt.gca()

def drawmapNorth(ax, labelsize=20, loc_x=0.1, loc_y=0.2, width=0.03, height=0.1,
    minx, maxx = ax.get_xlim()
    miny, maxy = ax.get_ylim()
    ylen = maxy - miny
    xlen = maxx - minx
    left = [minx + xlen*(loc_x - width*.5), miny + ylen*(loc_y - pad)]
    right = [minx + xlen*(loc_x + width*.5), miny + ylen*(loc_y - pad)]
    top = [minx + xlen*loc_x, miny + ylen*(loc_y - pad + height)]
    center = [minx + xlen*loc_x, left[1] + (top[1] - left[1])*.4]
    triangle = mpatches.Polygon([left, top, right, center], color='k')
    ax.text(s='N',
            x=minx + (maxx-minx)*loc_x,
            y=miny + (maxy-miny)*loc_y,
            fontsize=labelsize,
            horizontalalignment='center',
            verticalalignment='bottom')
    ax.add_patch(triangle)

def drawOceanTemperate(map,year,longitutein,latitudein):
    path="C:\\Users\\lenovo\\Desktop\\predict"
    path=join(path,str(year)+".csv")
    data=np.ones((181,401))*10
    with open(path) as file:
        lines=file.readlines()
        for line in lines:
            result=line.split(",")
            latitude=int(result[0])+90-1
            longitute=int(result[1])-1
            if longitute<=100:
                longitute+=360
            if result[2].strip()=="nan":
                temp=13
            else:
                temp=float(result[2].strip())
```

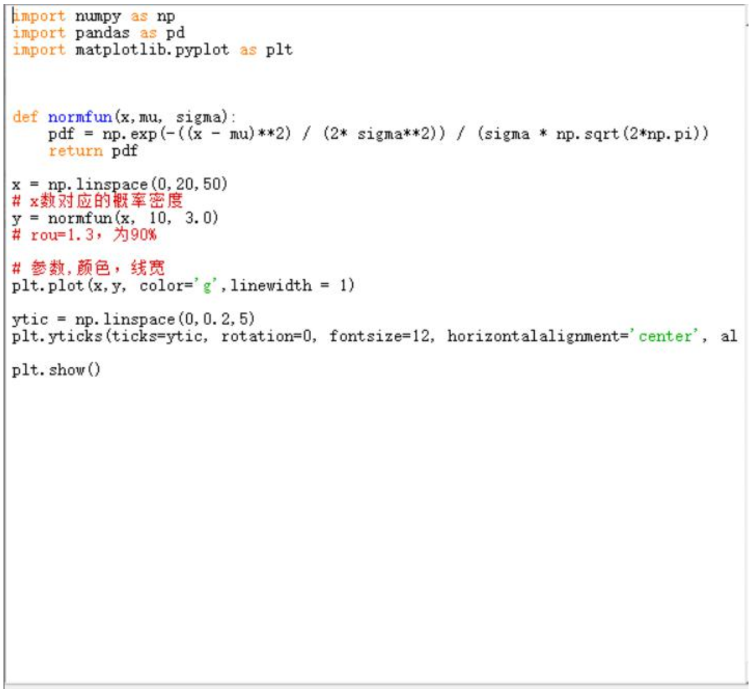


```
            if latitude<latitudein[0]+90 or latitude>latitudein[1]+90 \
                or longitute<longitutein[0] or longitute>longitutein[1]:
                temp=13

            data[latitude][longitute]=temp

    #为了比例更匀称
    data[136][370]=4
    data[136][372]=20

    x = np.linspace(-90, 90, 181)
    y = np.linspace(0, 400,401)
    xx,yy=np.meshgrid(y,x)
    contourf=map.contourf(xx,yy,data,17)
    plt.colorbar(contourf)

    #contour = map.contour(xx, yy,data,10, colors='k')
    #plt.clabel(contour, inline=True, fontsize=8)
    plt.title("Year: {} ocean temperature".format(year),fontsize=15)

    data2=np.array(data)
    for lati in range(181):
        for long in range(401):
            if data2[lati][long]>=13:
                data2[lati][long]=13.1
            else:
                data2[lati][long]=13
    contour=map.contour(xx,yy,data2,0)
    plt.clabel(contour, inline=True, fontsize=10)



def main():
    longitute=[340,372]
    latitude=[45,63]
    fig=plt.figure(figsize=(16,10))

    map = Basemap(llcrnrlon = longitute[0], llcrnrlat = latitude[0], urcrnrlon =
                  urcrnrlat = latitude[1], resolution ="l")
    savepath="C:\\Users\\lenovo\\Desktop\\predict"

    drawmapLine(map, np.linspace(-180, 180, 37), np.linspace(-90, 90, 37), [0, 0
    drawmapNorth(plt.gca())
    drawmapContient(map)
    drawOceanTemperate(map, 2122, longitute, latitude)
    plt.show()
    '''
    for i in range(2070,2072):
        plt.ion()
        plt.clf()

        drawmapLine(map, np.linspace(-180, 180, 37), np.linspace(-90, 90, 37), [
        drawmapNorth(plt.gca())
        drawOceanTemperate(map,i,longitute,latitude)
        drawmapContient(map)
```



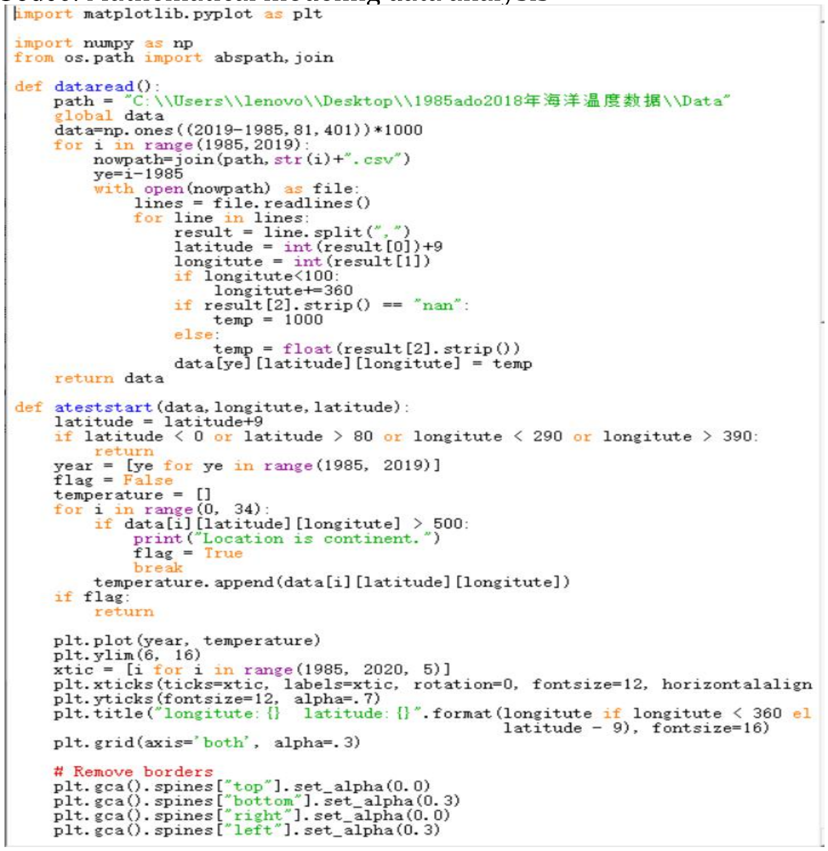
```
        plt.pause(3)
        plt.savefig(join(savepath,str(i)+".png"))

    plt.show()
    '''
if __name__=="__main__":
    main()
```

Code5. Normal distribution

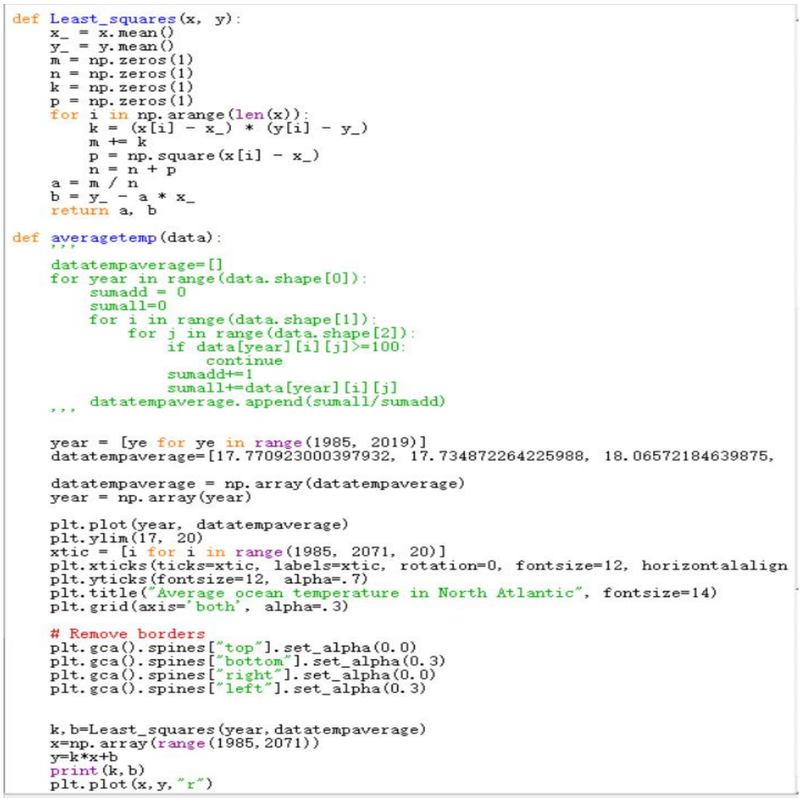


```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt



def normfun(x,mu, sigma):
    pdf = np.exp(-((x - mu)**2) / (2* sigma**2)) / (sigma * np.sqrt(2*np.pi))
    return pdf

x = np.linspace(0,20,50)
# x数对应的概率密度
y = normfun(x, 10, 3.0)
# rou=1.3，为90%

# 参数,颜色，线宽
plt.plot(x,y, color='g',linewidth = 1)

ytic = np.linspace(0,0.2,5)
plt.yticks(ticks=ytic, rotation=0, fontsize=12, horizontalalignment='center', al

plt.show()
```

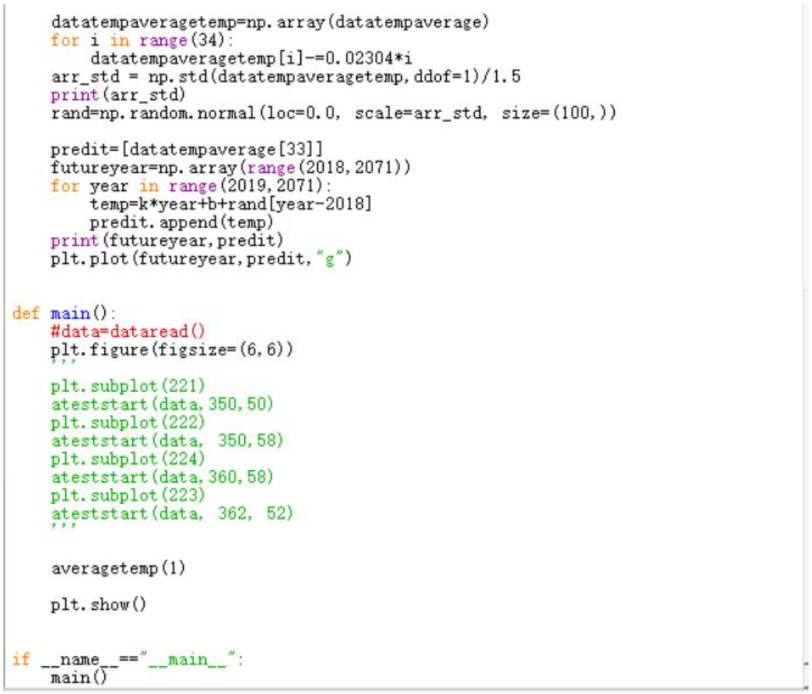
Code6. Mathematical modeling data analysis



```
import matplotlib.pyplot as plt

import numpy as np
from os.path import abspath,join

def dataread():
    path = "C:\\Users\\lenovo\\Desktop\\1985ado2018年海洋温度数据\\Data"
    global data
    data=np.ones((2019-1985,81,401))*1000
    for i in range(1985,2019):
        nowpath=join(path,str(i)+".csv")
        ye=i-1985
        with open(nowpath) as file:
            lines = file.readlines()
            for line in lines:
                result = line.split(",")
                latitude = int(result[0])+9
                longitute = int(result[1])
                if longitute<100:
                    longitute+=360
                if result[2].strip() == "nan":
                    temp = 1000
                else:
                    temp = float(result[2].strip())
                data[ye][latitude][longitute] = temp
    return data

def ateststart(data,longitute,latitude):
    latitude = latitude+9
    if latitude < 0 or latitude > 80 or longitute < 290 or longitute > 390:
        return
    year = [ye for ye in range(1985, 2019)]
    flag = False
    temperature = []
    for i in range(0, 34):
        if data[i][latitude][longitute] > 500:
            print("Location is continent.")
            flag = True
            break
        temperature.append(data[i][latitude][longitute])
    if flag:
        return

    plt.plot(year, temperature)
    plt.ylim(6, 16)
    xtic = [i for i in range(1985, 2020, 5)]
    plt.xticks(ticks=xtic, labels=xtic, rotation=0, fontsize=12, horizontalalign
    plt.yticks(fontsize=12, alpha=.7)
    plt.title("longitute:{}  latitude:{}".format(longitute if longitute < 360 el
                                               latitude - 9), fontsize=16)
    plt.grid(axis='both', alpha=.3)

    # Remove borders
    plt.gca().spines["top"].set_alpha(0.0)
    plt.gca().spines["bottom"].set_alpha(0.3)
    plt.gca().spines["right"].set_alpha(0.0)
    plt.gca().spines["left"].set_alpha(0.3)
```

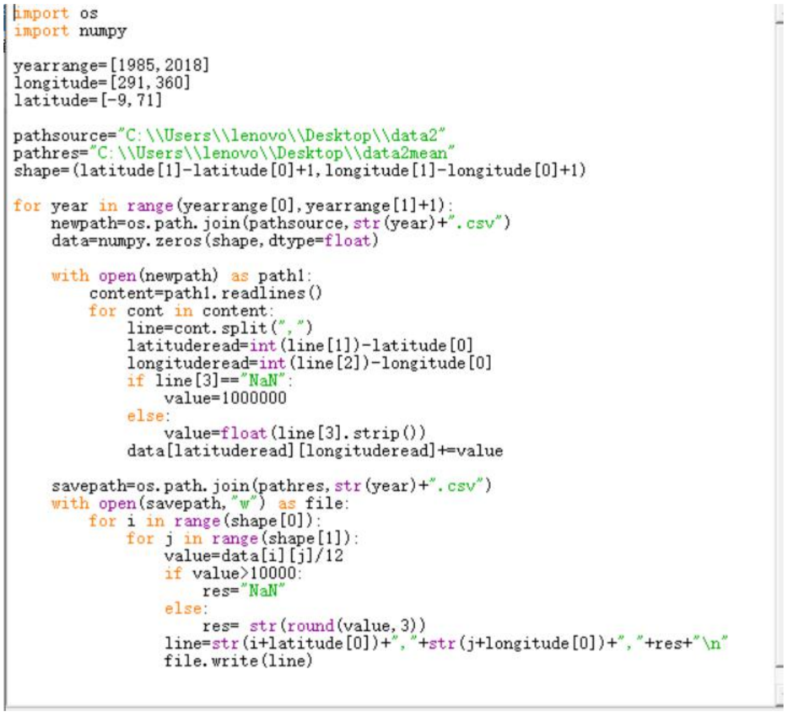


```
def Least_squares(x, y):
    x_ = x.mean()
    y_ = y.mean()
    m = np.zeros(1)
    n = np.zeros(1)
    k = np.zeros(1)
    p = np.zeros(1)
    for i in np.arange(len(x)):
        k = (x[i] - x_) * (y[i] - y_)
        m += k
        p = np.square(x[i] - x_)
        n = n + p
    a = m / n
    b = y_ - a * x_
    return a, b

def averagetemp(data):
    '''
    datatempaverage=[]
    for year in range(data.shape[0]):
        sumadd = 0
        sumall=0
        for i in range(data.shape[1]):
            for j in range(data.shape[2]):
                if data[year][i][j]>=100:
                    continue
                sumadd+=1
                sumall+=data[year][i][j]
    '''    datatempaverage.append(sumall/sumadd)

    year = [ye for ye in range(1985, 2019)]
    datatempaverage=[17.770923000397932, 17.734872264225988, 18.06572184639875,

    datatempaverage = np.array(datatempaverage)
    year = np.array(year)

    plt.plot(year, datatempaverage)
    plt.ylim(17, 20)
    xtic = [i for i in range(1985, 2071, 20)]
    plt.xticks(ticks=xtic, labels=xtic, rotation=0, fontsize=12, horizontalalign
    plt.yticks(fontsize=12, alpha=.7)
    plt.title("Average ocean temperature in North Atlantic", fontsize=14)
    plt.grid(axis='both', alpha=.3)

    # Remove borders
    plt.gca().spines["top"].set_alpha(0.0)
    plt.gca().spines["bottom"].set_alpha(0.3)
    plt.gca().spines["right"].set_alpha(0.0)
    plt.gca().spines["left"].set_alpha(0.3)


    k,b=Least_squares(year,datatempaverage)
    x=np.array(range(1985,2071))
    y=k*x+b
    print(k,b)
    plt.plot(x,y,"r")
```



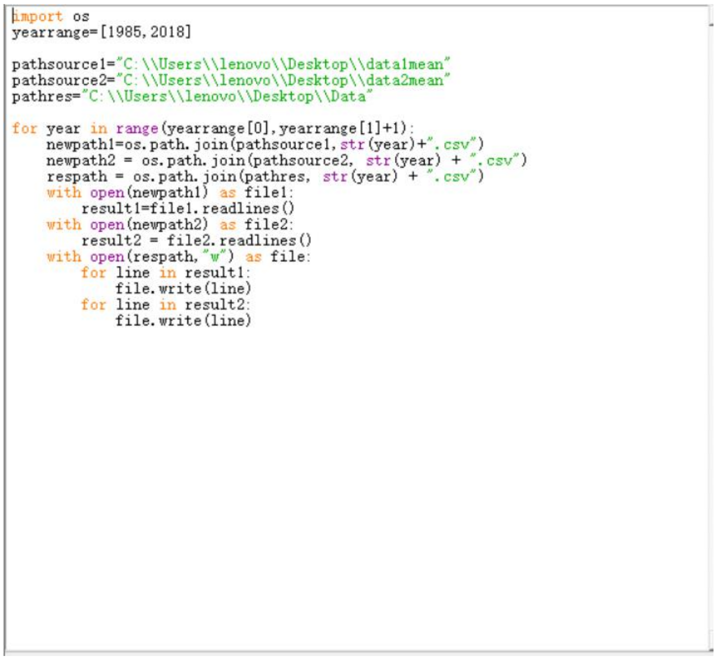
```
    datatempaveragetemp=np.array(datatempaverage)
    for i in range(34):
        datatempaveragetemp[i]-=0.02304*i
    arr_std = np.std(datatempaveragetemp,ddof=1)/1.5
    print(arr_std)
    rand=np.random.normal(loc=0.0, scale=arr_std, size=(100,))

    predit=[datatempaverage[33]]
    futureyear=np.array(range(2018,2071))
    for year in range(2019,2071):
        temp=k*year+b+rand[year-2018]
        predit.append(temp)
    print(futureyear,predit)
    plt.plot(futureyear,predit,"g")


def main():
    #data=dataread()
    plt.figure(figsize=(6,6))
    '''
    plt.subplot(221)
    ateststart(data,350,50)
    plt.subplot(222)
    ateststart(data, 350,58)
    plt.subplot(224)
    ateststart(data,360,58)
    plt.subplot(223)
    ateststart(data, 362, 52)
    '''

    averagetemp(1)

    plt.show()


if __name__=="__main__":
    main()
```

Code7. Mathematical modeling data merge



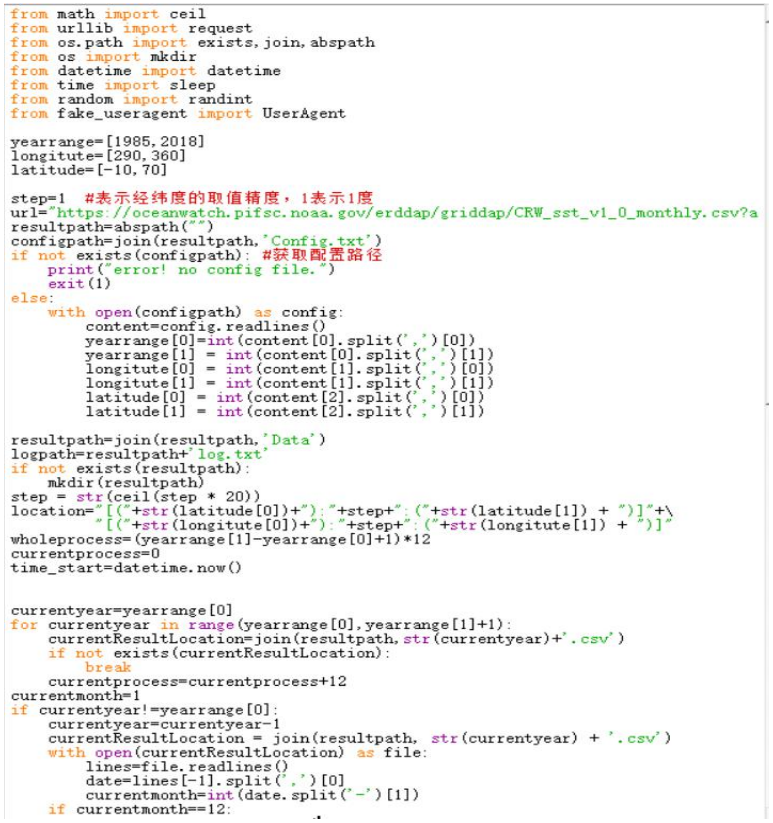
```
import os
import numpy

yearrange=[1985,2018]
longitude=[291,360]
latitude=[-9,71]

pathsource="C:\\Users\\lenovo\\Desktop\\data2"
pathres="C:\\Users\\lenovo\\Desktop\\data2mean"
shape=(latitude[1]-latitude[0]+1,longitude[1]-longitude[0]+1)

for year in range(yearrange[0],yearrange[1]+1):
    newpath=os.path.join(pathsource,str(year)+".csv")
    data=numpy.zeros(shape,dtype=float)

    with open(newpath) as path1:
        content=path1.readlines()
        for cont in content:
            line=cont.split(",")
            latituderead=int(line[1])-latitude[0]
            longituderead=int(line[2])-longitude[0]
            if line[3]=="NaN":
                value=1000000
            else:
                value=float(line[3].strip())
            data[latituderead][longituderead]+=value

    savepath=os.path.join(pathres,str(year)+".csv")
    with open(savepath,"w") as file:
        for i in range(shape[0]):
            for j in range(shape[1]):
                value=data[i][j]/12
                if value>10000:
                    res="NaN"
                else:
                    res= str(round(value,3))
                line=str(i+latitude[0])+","+str(j+longitude[0])+","+res+"\n"
                file.write(line)
```

Code8. Mathematical modeling data mean

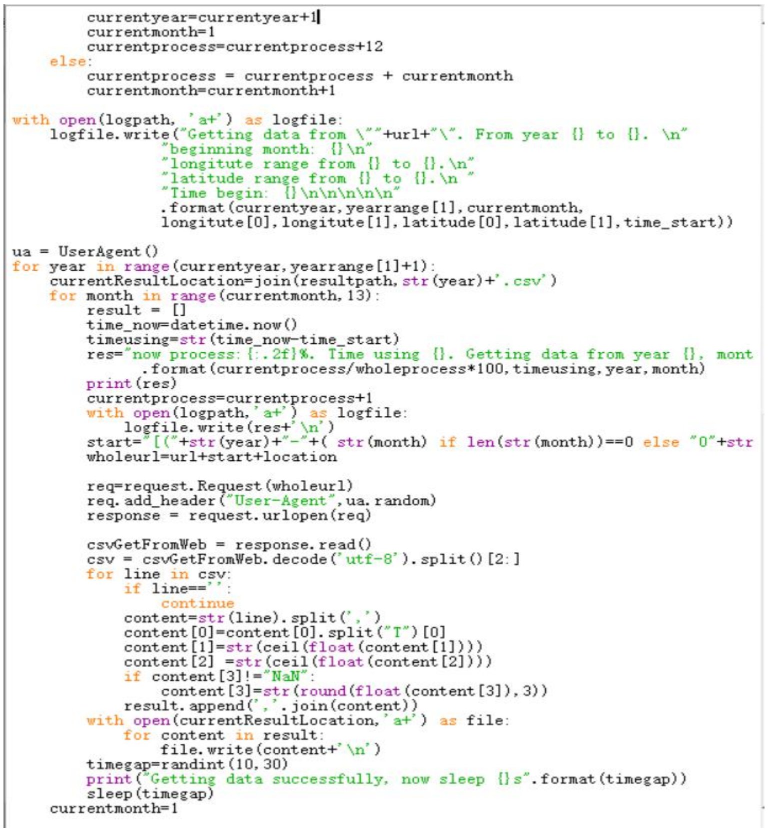


```
import os
yearrange=[1985,2018]

pathsource1="C:\\Users\\lenovo\\Desktop\\data1mean"
pathsource2="C:\\Users\\lenovo\\Desktop\\data2mean"
pathres="C:\\Users\\lenovo\\Desktop\\Data"

for year in range(yearrange[0],yearrange[1]+1):
    newpath1=os.path.join(pathsource1,str(year)+".csv")
    newpath2 = os.path.join(pathsource2, str(year) + ".csv")
    respath = os.path.join(pathres, str(year) + ".csv")
    with open(newpath1) as file1:
        result1=file1.readlines()
    with open(newpath2) as file2:
        result2 = file2.readlines()
    with open(respath,"w") as file:
        for line in result1:
            file.write(line)
        for line in result2:
            file.write(line)
```

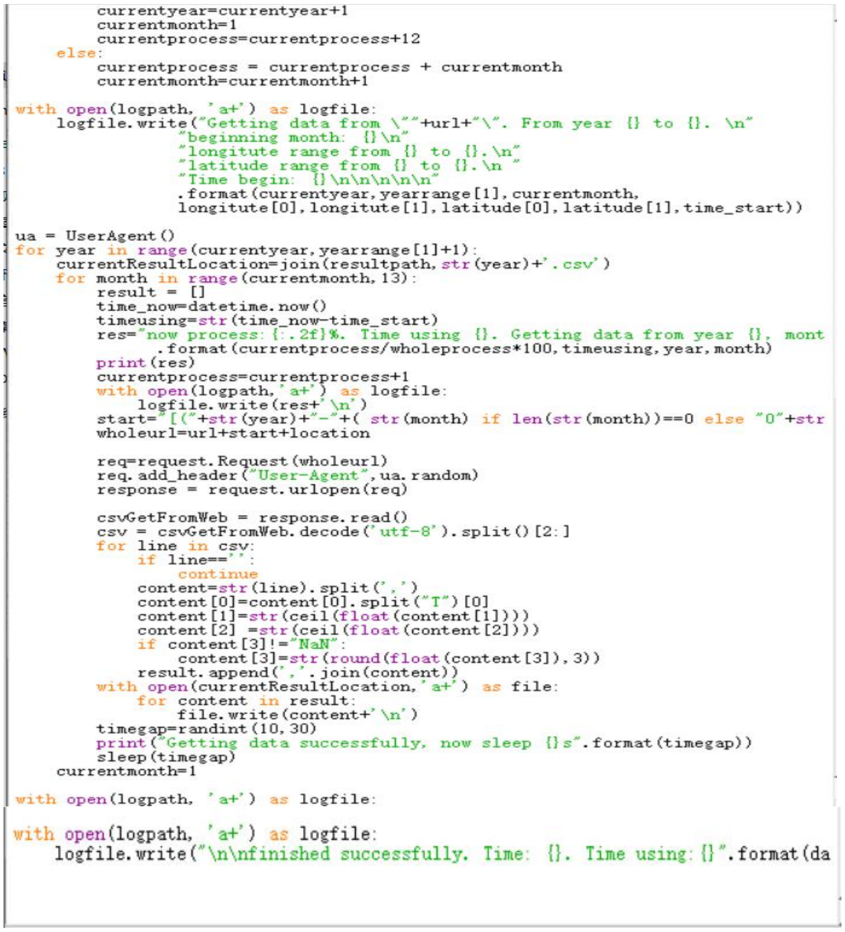
Code9. Mathematical modeling data acquisition



```
from math import ceil
from urllib import request
from os.path import exists,join,abspath
from os import mkdir
from datetime import datetime
from time import sleep
from random import randint
from fake_useragent import UserAgent

yearrange=[1985,2018]
longitute=[290,360]
latitude=[-10,70]

step=1  #表示经纬度的取值精度，1表示1度
url="https://oceanwatch.pifsc.noaa.gov/erddap/griddap/CRW_sst_v1_0_monthly.csv?a
resultpath=abspath("")
configpath=join(resultpath,'Config.txt')
if not exists(configpath): #获取配置路径
    print("error! no config file.")
    exit(1)
else:
    with open(configpath) as config:
        content=config.readlines()
        yearrange[0]=int(content[0].split(',')[0])
        yearrange[1] = int(content[0].split(',')[1])
        longitute[0] = int(content[1].split(',')[0])
        longitute[1] = int(content[1].split(',')[1])
        latitude[0] = int(content[2].split(',')[0])
        latitude[1] = int(content[2].split(',')[1])

resultpath=join(resultpath,'Data')
logpath=resultpath+'log.txt'
if not exists(resultpath):
    mkdir(resultpath)
step = str(ceil(step * 20))
location="[("+str(latitude[0])+"):"+step+":("+str(latitude[1]) + ")]"+\
         "[("+str(longitute[0])+"):"+step+":("+str(longitute[1]) + ")]"
wholeprocess=(yearrange[1]-yearrange[0]+1)*12
currentprocess=0
time_start=datetime.now()


currentyear=yearrange[0]
for currentyear in range(yearrange[0],yearrange[1]+1):
    currentResultLocation=join(resultpath,str(currentyear)+'.csv')
    if not exists(currentResultLocation):
        break
    currentprocess=currentprocess+12
currentmonth=1
if currentyear!=yearrange[0]:
    currentyear=currentyear-1
    currentResultLocation = join(resultpath, str(currentyear) + '.csv')
    with open(currentResultLocation) as file:
        lines=file.readlines()
        date=lines[-1].split(',')[0]
        currentmonth=int(date.split('-')[1])
    if currentmonth==12:
```

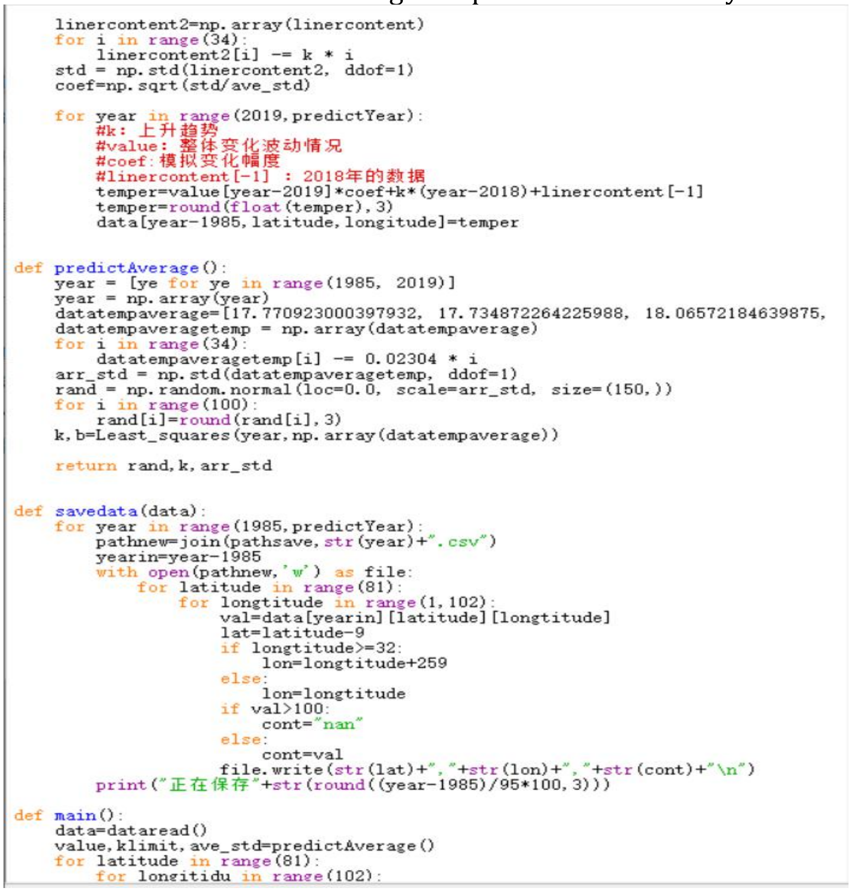


```
        currentyear=currentyear+1
        currentmonth=1
        currentprocess=currentprocess+12
    else:
        currentprocess = currentprocess + currentmonth
        currentmonth=currentmonth+1

with open(logpath, 'a+') as logfile:
    logfile.write("Getting data from \""+url+"\". From year {} to {}. \n"
                  "beginning month: {}\n"
                  "longitute range from {} to {}.\n"
                  "latitude range from {} to {}.\n "
                  "Time begin: {}\n\n\n\n\n"
                  .format(currentyear,yearrange[1],currentmonth,
                  longitute[0],longitute[1],latitude[0],latitude[1],time_start))

ua = UserAgent()
for year in range(currentyear,yearrange[1]+1):
    currentResultLocation=join(resultpath,str(year)+'.csv')
    for month in range(currentmonth,13):
        result = []
        time_now=datetime.now()
        timeusing=str(time_now-time_start)
        res="now process:{:.2f}%. Time using {}. Getting data from year {}, mont
              .format(currentprocess/wholeprocess*100,timeusing,year,month)
        print(res)
        currentprocess=currentprocess+1
        with open(logpath,'a+') as logfile:
            logfile.write(res+'\n')
        start="[("+str(year)+"-"+( str(month) if len(str(month))==0 else "0"+str
        wholeurl=url+start+location

        req=request.Request(wholeurl)
        req.add_header("User-Agent",ua.random)
        response = request.urlopen(req)

        csvGetFromWeb = response.read()
        csv = csvGetFromWeb.decode('utf-8').split()[2:]
        for line in csv:
            if line=='':
                continue
            content=str(line).split(',')
            content[0]=content[0].split("T")[0]
            content[1]=str(ceil(float(content[1])))
            content[2] =str(ceil(float(content[2])))
            if content[3]!="NaN":
                content[3]=str(round(float(content[3]),3))
            result.append(','.join(content))
        with open(currentResultLocation,'a+') as file:
            for content in result:
                file.write(content+'\n')
        timegap=randint(10,30)
        print("Getting data successfully, now sleep {}s".format(timegap))
        sleep(timegap)
    currentmonth=1
```

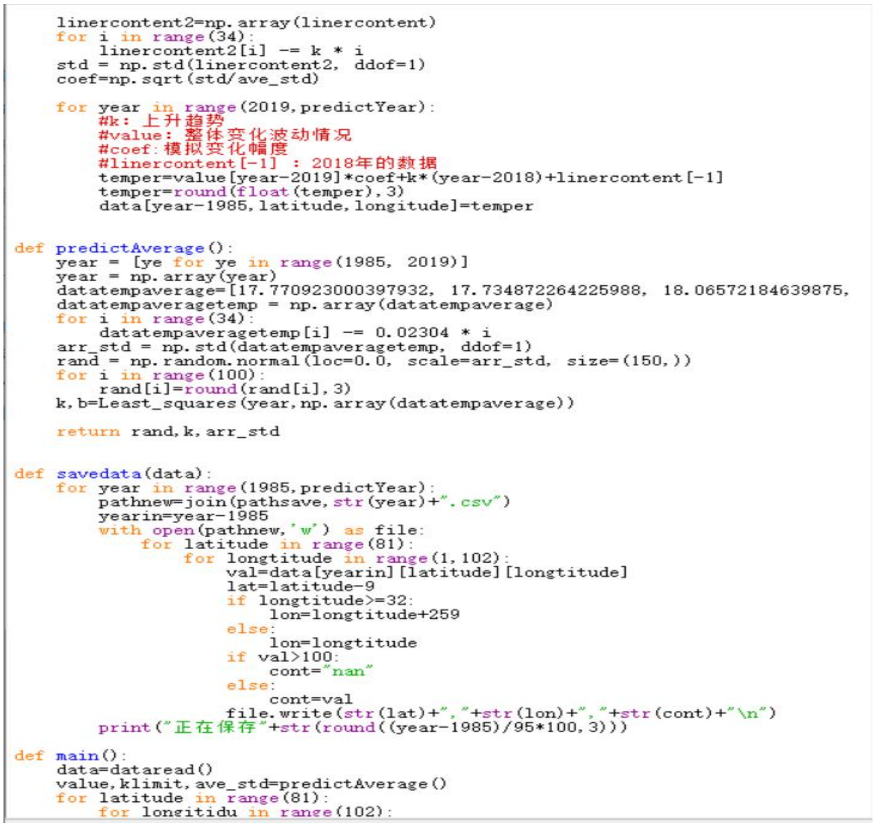


```
        currentyear=currentyear+1
        currentmonth=1
        currentprocess=currentprocess+12
    else:
        currentprocess = currentprocess + currentmonth
        currentmonth=currentmonth+1

with open(logpath, 'a+') as logfile:
    logfile.write("Getting data from \""+url+"\". From year {} to {}. \n"
                  "beginning month: {}\n"
                  "longitute range from {} to {}.\n"
                  "latitude range from {} to {}.\n "
                  "Time begin: {}\n\n\n\n\n"
                  .format(currentyear,yearrange[1],currentmonth,
                  longitute[0],longitute[1],latitude[0],latitude[1],time_start))

ua = UserAgent()
for year in range(currentyear,yearrange[1]+1):
    currentResultLocation=join(resultpath,str(year)+'.csv')
    for month in range(currentmonth,13):
        result = []
        time_now=datetime.now()
        timeusing=str(time_now-time_start)
        res="now process:{:.2f}%. Time using {}. Getting data from year {}, mont
              .format(currentprocess/wholeprocess*100,timeusing,year,month)
        print(res)
        currentprocess=currentprocess+1
        with open(logpath,'a+') as logfile:
            logfile.write(res+'\n')
        start="[("+str(year)+"-"+( str(month) if len(str(month))==0 else "0"+str
        wholeurl=url+start+location

        req=request.Request(wholeurl)
        req.add_header("User-Agent",ua.random)
        response = request.urlopen(req)

        csvGetFromWeb = response.read()
        csv = csvGetFromWeb.decode('utf-8').split()[2:]
        for line in csv:
            if line=='':
                continue
            content=str(line).split(',')
            content[0]=content[0].split("T")[0]
            content[1]=str(ceil(float(content[1])))
            content[2] =str(ceil(float(content[2])))
            if content[3]!="NaN":
                content[3]=str(round(float(content[3]),3))
            result.append(','.join(content))
        with open(currentResultLocation,'a+') as file:
            for content in result:
                file.write(content+'\n')
        timegap=randint(10,30)
        print("Getting data successfully, now sleep {}s".format(timegap))
        sleep(timegap)
    currentmonth=1

with open(logpath, 'a+') as logfile:

with open(logpath, 'a+') as logfile:
    logfile.write("\n\nfinished successfully. Time: {}. Time using:{}".format(da
```

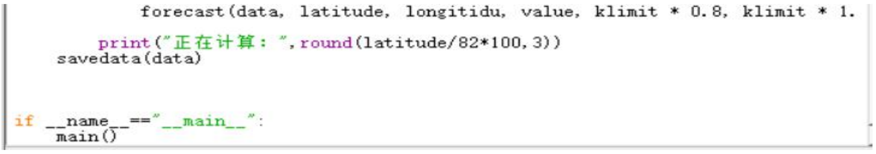
Code10. Mathematical modeling data prediction -- linearity



```
    linercontent2=np.array(linercontent)
    for i in range(34):
        linercontent2[i] -= k * i
    std = np.std(linercontent2, ddof=1)
    coef=np.sqrt(std/ave_std)

    for year in range(2019,predictYear):
        #k: 上升趋势
        #value: 整体变化波动情况
        #coef:模拟变化幅度
        #linercontent[-1] : 2018年的数据
        temper=value[year-2019]*coef+k*(year-2018)+linercontent[-1]
        temper=round(float(temper),3)
        data[year-1985,latitude,longitude]=temper


def predictAverage():
    year = [ye for ye in range(1985, 2019)]
    year = np.array(year)
    datatempaverage=[17.770923000397932, 17.734872264225988, 18.06572184639875,
    datatempaveragetemp = np.array(datatempaverage)
    for i in range(34):
        datatempaveragetemp[i] -= 0.02304 * i
    arr_std = np.std(datatempaveragetemp, ddof=1)
    rand = np.random.normal(loc=0.0, scale=arr_std, size=(150,))
    for i in range(100):
        rand[i]=round(rand[i],3)
    k,b=Least_squares(year,np.array(datatempaverage))

    return rand,k,arr_std


def savedata(data):
    for year in range(1985,predictYear):
        pathnew=join(pathsave,str(year)+".csv")
        yearin=year-1985
        with open(pathnew,'w') as file:
            for latitude in range(81):
                for longtitude in range(1,102):
                    val=data[yearin][latitude][longtitude]
                    lat=latitude-9
                    if longtitude>=32:
                        lon=longtitude+259
                    else:
                        lon=longtitude
                    if val>100:
                        cont="nan"
                    else:
                        cont=val
                    file.write(str(lat)+","+str(lon)+","+str(cont)+"\n")
        print("正在保存"+str(round((year-1985)/95*100,3)))

def main():
    data=dataread()
    value,klimit,ave_std=predictAverage()
    for latitude in range(81):
        for longitidu in range(102):
```



```
    linercontent2=np.array(linercontent)
    for i in range(34):
        linercontent2[i] -= k * i
    std = np.std(linercontent2, ddof=1)
    coef=np.sqrt(std/ave_std)

    for year in range(2019,predictYear):
        #k: 上升趋势
        #value: 整体变化波动情况
        #coef:模拟变化幅度
        #linercontent[-1] : 2018年的数据
        temper=value[year-2019]*coef+k*(year-2018)+linercontent[-1]
        temper=round(float(temper),3)
        data[year-1985,latitude,longitude]=temper


def predictAverage():
    year = [ye for ye in range(1985, 2019)]
    year = np.array(year)
    datatempaverage=[17.770923000397932, 17.734872264225988, 18.06572184639875,
    datatempaveragetemp = np.array(datatempaverage)
    for i in range(34):
        datatempaveragetemp[i] -= 0.02304 * i
    arr_std = np.std(datatempaveragetemp, ddof=1)
    rand = np.random.normal(loc=0.0, scale=arr_std, size=(150,))
    for i in range(100):
        rand[i]=round(rand[i],3)
    k,b=Least_squares(year,np.array(datatempaverage))

    return rand,k,arr_std


def savedata(data):
    for year in range(1985,predictYear):
        pathnew=join(pathsave,str(year)+".csv")
        yearin=year-1985
        with open(pathnew,'w') as file:
            for latitude in range(81):
                for longtitude in range(1,102):
                    val=data[yearin][latitude][longtitude]
                    lat=latitude-9
                    if longtitude>=32:
                        lon=longtitude+259
                    else:
                        lon=longtitude
                    if val>100:
                        cont="nan"
                    else:
                        cont=val
                    file.write(str(lat)+","+str(lon)+","+str(cont)+"\n")
        print("正在保存"+str(round((year-1985)/95*100,3)))

def main():
    data=dataread()
    value,klimit,ave_std=predictAverage()
    for latitude in range(81):
        for longitidu in range(102):
```



```
            forecast(data, latitude, longitidu, value, klimit * 0.8, klimit * 1.

        print("正在计算: ",round(latitude/82*100,3))
    savedata(data)


if __name__=="__main__":
    main()
```