|  |
| --- |
| 附录一：问题一的求解代码（python3.8） |
| import numpy as np  import pandas as pd  import math  #p就是上文提到的β  n=11000000  i0=1  p=0  k=[1,1,2,5,5,5,8,8,12,14,16,25,29,37,40,45,47,49,59,68,78,90,102]  for i in range(2,len(k)):  cnt=(n\*i0/k[i]-i0)/(n-i0)  p+=(-math.log(cnt))/(i+1)  p/=len(k)-2  print(str(p))  #p=0.2534751872377965 |
| 附录二：国内β在前期的拟合曲线（python3.8） |
| import numpy as np  import pandas as pd  from matplotlib import pyplot as plt  import math  n=11000000  i0=1  p=0  k=[1,1,2,5,5,5,8,8,12,14,16,25,29,37,40,45,47,49,59,68,78,90,102]  k2=list()  for i in range(3,len(k)):  cnt=(n\*i0/k[i]-i0)/(n-i0)  k2.append((-math.log(cnt))/(i+1))  x = np.arange(1, len(k2)+1,1)  y = np.array(k2)  z1 = np.polyfit(x, y,5)#用5次多项式拟合  p1 = np.poly1d(z1)  print(p1) #在屏幕上打印拟合多项式  yvals=p1(x)  plot1=plt.plot(x, y, '\*')  plot2=plt.plot(x, yvals, 'r')  plt.show()    5 4 3 2  #-1.983e-06 x + 0.0001218 x - 0.002787 x + 0.02899 x - 0.1378 x + 0.5078 |
| 附录3：武汉封城后μ+η的拟合曲线（python3.8） |
| # coding: utf-8  import requests  import json  import numpy as np  import pandas as pd  from matplotlib import pyplot as plt  from datetime import datetime  s=11000000  china = pd.read\_csv('wuhan\_30.csv', encoding='GBK')  data=china.iloc[::].values  k=list()  for i in range(30):  k.append((data[i][4]\*2-data[i][5])/data[i][1])  print(k[i])  x = np.arange(1, len(k)+1, 1)  y = np.array(k)  z1 = np.polyfit(x, y,2)#用2次多项式拟合  p1 = np.poly1d(z1)  print(p1) #在屏幕上打印拟合多项式  yvals=p1(x)  plot1=plt.plot(x, y, '\*')  plot2=plt.plot(x, yvals, 'r')  plt.legend(loc=4)  plt.show()    # 2  #-6.84e-06 x + 0.0008303 x + 0.003564 |
| 附录4： logic函数推导出不采取措施的数据（python3.8） |
| import math  for i in range(1,101):  k=math.e\*\*(-i\*0.254596492)  k=k\*(11000000-1)+1  k=11000000/k  print(str(k)) |
| 附录5： 校验国内前期β的准确性（python3.8） |
| import math  ave=0.2534751872377965  sum=0  n=11000000  i0=1  p=0  k2=[1,1,2,5,5,5,8,8,12,14,16,25,29,37,40,45,47,49,59,68,78,90,102]  k=list()  for i in range(3,len(k2)):  cnt=(n\*i0/k2[i]-i0)/(n-i0)  k.append((-math.log(cnt))/(i+1))  for x in range(1,21):  p=-1.983e-06\*x\*\*5 + 0.0001218\*x\*\*4 - 0.002787\*x\*\*3 + 0.02899\*x\*\*2 - 0.1378\*x + 0.5078  sum=sum+(p-k[x-1])\*\*2  sum2=0  for x in range(1,21):  p=-1.983e-06\*x\*\*5 + 0.0001218\*x\*\*4 - 0.002787\*x\*\*3 + 0.02899\*x\*\*2 - 0.1378\*x + 0.5078  sum2+=(p-0.254596492)\*\*2  sum3=1-(sum/sum2)  print(str(sum3))  #sum3=0.9605702849424397 |
| 附录6： 武汉封城后λ的拟合（python3.8） |
| # coding: utf-8  import numpy as np  import pandas as pd  from matplotlib import pyplot as plt  from datetime import datetime  s=11000000  china = pd.read\_csv('wuhan\_30.csv', encoding='GBK')  data=china.iloc[::].values  k=list()  for i in range(30):  k.append(data[i][4]/((s-data[i][1])\*data[i][1]))  print(k[i])  '''  for i in range(30):  k.append(data[i][1]/((s-data[i][2])\*data[i][2]))  print(k[i])  '''  x = np.arange(1, len(k)+1, 1)  y = np.array(k)  z1 = np.polyfit(x, y,5)#用5次多项式拟合  p1 = np.poly1d(z1)  print(p1) #在屏幕上打印拟合多项式  yvals=p1(x)  plot1=plt.plot(x, y, '\*')  plot2=plt.plot(x, yvals, 'r')  plt.legend(loc=4)  plt.show()  # 5 4 3 2  #1.387e-15 x - 1.928e-13 x + 9.214e-12 x - 1.569e-10 x - 2.564e-10 x + 2.653e-08 |
| 附录7： 迭代出正常情况下感染人数增长情况（python3.8） |
| # coding: utf-8  import requests  import json  import math  import numpy as np  import pandas as pd  from matplotlib import pyplot as plt  from datetime import datetime  e = pd.read\_csv('wuhan\_30.csv', encoding='GBK')  s=11000000  #it=46063  '''  for i in range(1,31):  x=i  la=1.387e-15\*i\*\*5 - 1.928e-13\*x\*\*4 + 9.214e-12\*x\*\*3 - 1.569e-10\*x\*\*2 - 2.564e-10\*x + 2.653e-08  u=-6.782e-06\*x\*\*2 + 0.0008275\*x + 0.003428  st=(la\*it+1+u+la\*(s-it)-math.sqrt(((la\*it+1+u+la\*(s-it))\*\*2-4\*la\*(1+u))))/(2\*la)  it=it/(1+u+-la\*st)  print(str(it))  '''  for i in range(31,61):  st=0.9593030450458995  it=45256.602628212655  x=i  la=1.387e-15\*i\*\*5 - 1.928e-13\*x\*\*4 + 9.214e-12\*x\*\*3 - 1.569e-10\*x\*\*2 - 2.564e-10\*x + 2.653e-08  u=-6.782e-06\*x\*\*2 + 0.0008275\*x + 0.003428  st=(la\*it+1+u+la\*(s-it)-math.sqrt(((la\*it+1+u+la\*(s-it))\*\*2-4\*la\*(1+u))))/(2\*la)  it=it/(1+u+-la\*st)  print(str(int(it))) |