

## 目录

1 引言 .....	2
1.1 研究背景: .....	2
1.2 研究方法: .....	2
2 数据获取与数据预处理 .....	3
2.1 数据集介绍: .....	3
2.2 数据预处理: .....	3
2.2.1 人口数据预处理: .....	3
2.2.2 老年服务数据预处理 .....	3
3 建模方法分析: .....	4
3.1 基本思想 .....	4
3.2 原理讲解 .....	4
3.3 建模假设: .....	5
4 数据建模: .....	5
4.1 数据处理与分析: 基于 Pandas 库的数据导入与处理 .....	6
4.2 人口统计与计算: 死亡人口和新出生人口的计算 .....	8
4.3 人口统计与计算: 新一阶段人口数量的计算 .....	9
4.4 人口数量转换和输出 .....	10
5 可视化相关数据计算 .....	13
5.1 老年服务的分区统计 .....	13
5.2 上海市数据统计与计算 .....	13
5.3 各区老年服务满足比计算 .....	14
5.4 老年服务设施数据归一化 .....	14
6 可视化 .....	15
6.1 预测未来年龄性别结构 .....	17
6.2 预测出的未来老年抚养比 .....	17
6.4 上海市老年人口分布现状 .....	15
6.5 当前各类设施满足比 .....	16
7 结论 .....	18
8 建议 .....	18

# 1 引言

## 1.1 研究背景：

作为中国最具活力的经济中心之一，上海面临着人口结构巨大变化的挑战。截至 2022 年 12 月 31 日，全市 65 岁及以上老年人口 424.40 万人，占总人口的 28.2%，远超国际公认的深度老龄化社会 14% 的标准。同时从 2021 年末至 2022 年末 65 岁及以上人口增加了 22.03 万人，增长 5.5%；15-64 岁劳动年龄人口抚养 65 岁及以上人口的老年抚养系数增加 3.0 个百分点。这一趋势带来的城市老年人的生活照料和精神赡养问题日益突出。

为了应对老龄化的挑战，上海市政府已经采取了一系列政策措施来促进养老服务的发展。在 2021 年上海市人民政府办公厅发布的《关于推进本市“十四五”期间养老服务设施建设的实施意见》中，强调了“十四五”期间本市养老服务设施建设围绕“增量、增能、增效”，落实布局专项规划，进一步织密网点，均衡布局，推动机构养老服务更加专业优质、社区养老服务更加方便可及、家庭照料能力不断增强；实现基本养老服务“应保尽保”，设施保障水平不断提高，功能不断提升，多层次多样化养老服务更加充分发展。

然而，随着老年人口数量的不断增加和老龄化程度的不断加深，养老服务供需矛盾日益突出。因此，需要开展深入的研究和分析，以便更好地了解老年人口及养老服务的现状及未来发展趋势。这样可以帮助我们更好地处理养老服务供需矛盾、评估政策实施效果、为上海市养老服务行业的发展提供参考。

## 1.2 研究方法：

基于 Leslie 矩阵模型，本研究旨在预测 2025-2050 年上海各区域老年人口分段情况，并计算老年人口抚养比及老年服务设施满足比。通过运用先进的数学模型和数据科学方法，我们将分析上海老年人口增长趋势和服务需求变化，以提出合理的老年服务设施建设建议。本研究将通过 Python 实现数据的处理、模型的构建，旨在为上海市政府和相关部门制定老年服务政策提供科学依据和决策支持。

# 2 数据获取与数据预处理

## 2.1 数据集介绍：

本次分析的主要数据来源于 2020 年上海市人口普查年鉴以及上海市养老服务平台。其中 2020 年上海市人口普查年鉴我使用了这些数据集：各地区分年龄、性别的人口，各地区分年龄、性别的死亡人口，各地区分性别、孩次的出生人口，各地区育龄妇女年龄别生育率及总和生育率。上海市养老服务平台我使用了长者照护之家，社区老年人日间照护机构，老年助餐服务场所，社区综合为老服务中心，农村睦邻点的信息。

## 2.2 数据预处理：

### 2.2.1 人口数据预处理：

为了进行合理预测，我们使用 2020 年上海市人口普查年鉴，得到了上海各地区分性别、孩次的出生人口，各地区分年龄、性别的死亡人口，各地区分年龄、性别的人口，各地区育龄妇女年龄别生育率及总和生育率，在后续过程中，我们需要得到不同年龄阶段的存活率，新生儿男女数量。

$$\text{分年龄死亡率} = \frac{\text{该年龄段死亡人数}}{\text{该年龄段总人数}}$$

为了计算不同年龄阶段的死亡率，我们将对数据进行处理，使用 excel 进行数据预处理，因为数值过小，为了方便展示，将数值放大一千倍，使用千分比进行展示，并且保留两位小数。在接下来的预测中，不同年龄的死亡率将和此表保持一致（此表见附录）。为了得到每年的新生儿数量，我将用各地区育龄妇女年龄别生育率\*分年龄育龄妇女人数去计算新生儿数量，再根据性别比计算男婴，女婴数量。观察统计表格，发现存在数据缺失，但是经过验算，发现数值应为 0（单位%）。在接下来的预测中，不同年龄的生育率将和此表保持一致（此表见附录）。

### 2.2.2 老年服务数据预处理

将数据在 Excel 中运用字段提取函数，在 D 列中提取出相应服务所处的行政区。

D2    :    ✕    ✓    fx    =MID(C2,FIND(":",C2)+1,FIND("区",C2)+1-FIND(":",C2)-1)			
A	B	C	D
1 字段1	字段2	字段3	
2 上海崇明三星镇至行长者照护之家	地址：崇明区三星镇宏海公路4318	地址：崇明区三星镇宏海公路4318号	崇明区
3 上海崇明堡镇全程玖玖长者照护之家	电话：021-59690298	地址：崇明区堡镇通富路37号	崇明区

	A	B	C	D
1	字段1	字段2	字段3	
2	上海崇明三星镇至行长者照护之家	地址：崇明区三星镇宏海公路4318号	崇明区三星镇宏海公路4318号	崇明区
3	上海崇明堡镇全程玖玖长者照护之家	电话：021-59690298	地址：崇明区堡镇通富路37号	崇明区
4	上海崇明三星镇至远长者照护之家	电话：59601261	地址：崇明区三星镇协进村497号	崇明区
5	上海崇明区城桥鳌山村乐祥长者照护之家	电话：59620317	地址：崇明区城桥镇崇明区城桥镇鳌山村888号	崇明区
6	上海崇明新村乡新洲长者照护之家	电话：18918801520	地址：崇明区新村乡新洲村	崇明区
7	上海青浦区夏阳社区旭昇长者照护之家	电话：13701738221	地址：青浦区夏阳街道沪青平公路5251号	青浦区
8	上海市华新社区万德福长者照护之家	电话：69792689	地址：青浦区华新镇青浦区华新镇纪鹤公路2828号	青浦区
9	上海奉贤区浦秀青春里长者照护之家	电话：021-57401013, 1891826677	地址：奉贤区庄行镇浦秀村耀东1328号	奉贤区
10	上海奉贤区高桥长者照护之家	电话：37580501	地址：奉贤区奉城镇奉旺路659号	奉贤区
11	上海松江广富林街道三新园区长者照护之家	地址：松江区方松街道松江区广富林街道三新北路1800弄	松江区方松街道松江区广富林街道三新北路1800弄	松江区
12	上海市松江区岳阳街道景德路社区晓驻长者照护之家	电话：021-37790020	地址：松江区岳阳街道松江区中山中路36号	松江区
13	上海市松江区泖港镇黄桥长者照护之家	电话：57866030	地址：松江区泖港镇黄桥中心公路1083号	松江区
14	上海松江区佘山镇长者照护之家	电话：57655006	地址：松江区佘山镇千新公路1200号	松江区
15	上海松江慧享福长者照护之家	电话：021-67676726	地址：松江区方松街道松江区方松街道江学路381弄50-56	松江区
16	上海松江慧享福长者照护之家	电话：021-67676726	地址：松江区方松街道松江区方松街道江学路381弄50-56	松江区
17	上海永丰社区长者照护之家	电话：021-51219837	地址：松江区永丰街道富强路1621-1637	松江区
18	上海泖港社区怡鑫长者照护之家	电话：18917169735	地址：松江区泖港镇五库大街58号	松江区
19	上海岳阳街道康福长者照护之家	电话：021-37657571	地址：松江区岳阳街道乐都路390号荣乐四村5号	松江区
20	上海金山区漕泾镇水库长者照护之家	电话：18202168928	地址：金山区漕泾镇水库村长堰路2076号C区	金山区
21	上海金山区漕泾镇汇安长者照护之家	电话：57250381	地址：金山区漕泾镇金山区漕泾镇漕路539弄48号	金山区

### 3 建模方法分析：

#### 3.1 基本思想

Leslie 矩阵人口模型是一种用于预测人口动态变化的数学模型。该模型基于年龄结构将人口划分为不同的年龄组，并通过考虑生育率和死亡率等因素，建立了一个矩阵方程来描述人口在时间上的演变。

Leslie 矩阵模型的核心思想是将人口按照年龄进行分组，并假设每个年龄组的人口在下一时间步长内将按照一定的比例转移到其他年龄组。这个转移过程可以通过一个矩阵来描述，即 Leslie 矩阵。

在 Leslie 矩阵中，行表示当前时间步长的年龄组，列表示下一个时间步长的年龄组。矩阵的每个元素表示从一个年龄组转移到另一个年龄组的比例。特别地，矩阵的对角线上的元素表示同一年龄组内存活的比例，而非对角线上的元素则表示不同年龄组之间的转移比例。通过将当前时间步长的人口分布向量与 Leslie 矩阵相乘，可以得到下一个时间步长的人口分布向量。通过迭代这个过程，可以预测人口在未来多个时间步长内的动态变化。

Leslie 矩阵模型在人口学、生态学和生物学等领域有广泛的应用。它可以帮助研究人员理解人口结构的变化、预测未来的人口趋势，以及评估不同政策或环境因素对人口动态的影响。

#### 3.2 原理讲解

首先假设预测期各年龄段的男、女人口分布与基期相同。用  $x(i)$  表示第  $k$  年第  $i$  年龄组的人口数， $q_{k-1}(i)$  表示第  $k-1$  年第  $i$  年龄组的死亡率，那么当  $i \geq 1$  时，第  $k$  年第  $i+1$  组的人口数可以表示为

$$x_k(i+1) = x_{k-1}(i)(1 - q_{k-1}(i)), i = 1, 2, \dots, n-1$$

由于  $x(1)$  代表的是由育龄妇女新生育的人口数，不能用上一年龄组的存活人口来计算，故不能适用上述公式。这里我们单独讨论，用每一年龄组女性的生育人口数总和来表示第一年龄组人口数。即

$$x_k(1) = \sum_{i=1}^n c_{k-1}(i)x_{k-1}(i)b_{k-1}(i)$$

于是可以得到人口发展的模型：

$$x_k = \begin{pmatrix} c_{k-1}(1)b_{k-1}(1) & \cdots & \cdots & c_{k-1}(n)b_{k-1}(n) \\ 1 - q_{k-1}(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 - q_{k-1}(n-1) & 0 \\ 0 & \cdots & \cdots & 1 - q_{k-1}(n) \end{pmatrix}$$

其中：

$$x_k = \begin{pmatrix} x_k(1) \\ x_k(2) \\ \vdots \\ x_k(n) \end{pmatrix} \quad x_{k-1} = \begin{pmatrix} x_{k-1}(1) \\ x_{k-1}(2) \\ \vdots \\ x_{k-1}(n) \end{pmatrix}$$

令：

$$M = \begin{pmatrix} c_{k-1}(1)b_{k-1}(1) & \cdots & \cdots & c_{k-1}(n)b_{k-1}(n) \\ 1 - q_{k-1}(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 - q_{k-1}(n-1) & 0 \\ 0 & \cdots & \cdots & 1 - q_{k-1}(n) \end{pmatrix}$$

可得：

$$x_k = Mx_{k-1}$$

### 3.3 建模假设：

为了简化预测模型，使预测过程更简单明了，本节先做以下基本假设：

- (1) 所有预测均以 2020 年为基期，预测所需原始数据均以 2020 年上海市人口普查年鉴数据为基础。
- (2) 本节假设预测期间各年龄组的性别占比与基期保持一致
- (3) 将年龄组分为 21 组，最大可以取 21，年龄组的组间距为 5 周岁。
- (4) 中国目前死亡率已经处于很低的水平，在自然环境比较稳定的情况下变化不大，因此本节为了简化模型，便于预测，假设各年龄组的年龄别死亡率在预测期内与 2020 年上海市人口普查年鉴数据保持一致。
- (5) 模型中使用的基期年龄别生育率和总和生育率用 2020 年上海市人口普查年鉴数据近似代替。

## 4 数据建模：

上文已经详细介绍了 Leslie 矩阵模型，我们可知该模型通常以一年为分割点进行预测，但在本文中，以五年为一段进行预测可能更为合适。因此，代码在 Leslie 模型基础上进行改进，以实现五年段的预测功能。

以下是代码的详细介绍：

## 4.1 数据处理与分析：基于 Pandas 库的数据导入与处理

为了高效、准确地处理数据，我们选用了 Pandas 库，Pandas 是一个功能强大、易于使用的 Python 数据处理库。通过 Pandas，我们能够方便地读取、清洗、转换和可视化各种形式的数据。

数据导入：通过 Pandas 库的 `read_excel()` 函数，从三个 Excel 文件中分别导入了关于人口统计、死亡率和生育率的数据集。这些数据集存储在 `data_total`、`data_die` 和 `data_born` 这三个变量中，以供后续处理和分析使用。

缺失值处理：为了确保数据的完整性和准确性，代码使用 `fillna()` 函数对数据集中的缺失值进行了填充处理。在本例中，所有缺失值被替换为 0。

时间间隔设置：代码中定义了一个常量 `TIME_GAP`，其值为 5。这个常量用于指定后续计算的时间间隔，即每 5 年作为一个时间段进行数据处理和分析。

初始化数据结构：为了存储按年份组织的数据或统计信息，代码初始化了三个空字典 `t_year_list`、`m_year_list` 和 `w_year_list`。

列名列表获取：为了方便后续引用和处理列名，代码通过 `.columns.tolist()` 方法将 `data_total` 数据集的列名转换为一个列表，并存储在 `col_total_l` 变量中。

```
import pandas as pd
import copy

data_total = pd.read_excel(io: './各区分段统计1.xlsx', sheet_name='Sheet1')
data_die = pd.read_excel(io: './各区死亡率统计1.xlsx', sheet_name='Sheet1')
data_born = pd.read_excel(io: './生育率1.xlsx', sheet_name='Sheet1')
data_total.fillna(0)
data_die.fillna(0)
data_born.fillna(0)

TIME_GAP = 5

t_year_list = {}
m_year_list = {}
w_year_list = {}

col_total_l = data_total.columns.tolist()
```

在导入上海市各区分段人口统计表时，需要将分段总人数、男人数、女人数分别存储进不同的字典中。为了实现这一目的，代码使用了嵌套的循环结构。

外部循环 `for area_index in range(0, 17):` 遍历了从 0 到 16 的索引范围，这个范围代表了上海市的各个区。内部循环 `for col_index in range(1, 64):` 遍历了从 1 到 63 的列索引范围，这些列包含了不同年龄段的人口统计信息。

通过判断 `(col_index - 1) % 3` 的值来确定当前列是属于总人数、男人数还是女人数。如果余数为 0，表示当前列是总人数；如果余数为 1，表示当前列是男人数；如果余数为 2，表示当前列是女人数。

然后使用 `if` 语句来判断相应的年份字典是否已经创建，以及相应的地区字典是否已经存

在。如果不存在，则使用 `{}` 初始化这些字典。

最后，使用 `data\_total.iat[area\_index, col\_index]` 获取当前区域和列的人口统计数据，并将其存储到相应的字典中。字典的层级结构为：年份 → 区域 → 人口统计信息。通过这段代码，我们可以得到三个字典 `t\_year\_list`、`m\_year\_list` 和 `w\_year\_list`，分别包含了上海市各区分段总人数、男人数和女人数的统计信息。这些数据以嵌套字典的形式进行组织，方便后续的数据分析和处理。

```
for area_index in range(0, 17):
    for col_index in range(1, 64):
        if (col_index - 1) % 3 == 0:
            if 2020 not in t_year_list:
                t_year_list[2020] = {}
            if data_total.iat[area_index, 0].strip() not in t_year_list[2020]:
                t_year_list[2020][data_total.iat[area_index, 0].strip()] = {}
            t_year_list[2020][data_total.iat[area_index, 0].strip()][col_total_l[col_index]] = data_total.iat[
                area_index, col_index]
        elif (col_index - 1) % 3 == 1:
            if 2020 not in m_year_list:
                m_year_list[2020] = {}
            if data_total.iat[area_index, 0].strip() not in m_year_list[2020]:
                m_year_list[2020][data_total.iat[area_index, 0].strip()] = {}
            m_year_list[2020][data_total.iat[area_index, 0].strip()][col_total_l[col_index - 1]] = data_total.iat[
                area_index, col_index]
        elif (col_index - 1) % 3 == 2:
            if 2020 not in w_year_list:
                w_year_list[2020] = {}
            if data_total.iat[area_index, 0].strip() not in w_year_list[2020]:
                w_year_list[2020][data_total.iat[area_index, 0].strip()] = {}
            w_year_list[2020][data_total.iat[area_index, 0].strip()][col_total_l[col_index - 2]] = data_total.iat[
                area_index, col_index]
```

接下来使用同样的方法储存死亡率：

```
t_die = {}
m_die = {}
w_die = {}
col_die_l = data_die.columns.tolist()
# print(col_die_l)
for area_index in range(0, 17):
    for col_index in range(1, 64):
        if (col_index - 1) % 3 == 0:
            if data_die.iat[area_index, 0].strip() not in t_die:
                t_die[data_die.iat[area_index, 0].strip()] = {}
            t_die[data_die.iat[area_index, 0].strip()][col_die_l[col_index]] = data_die.iat[area_index, col_index]/1000
        elif (col_index - 1) % 3 == 1:
            if data_die.iat[area_index, 0].strip() not in m_die:
                m_die[data_die.iat[area_index, 0].strip()] = {}
            m_die[data_die.iat[area_index, 0].strip()][col_die_l[col_index - 1]] = data_die.iat[area_index, col_index]/1000
        elif (col_index - 1) % 3 == 2:
            if data_die.iat[area_index, 0].strip() not in w_die:
                w_die[data_die.iat[area_index, 0].strip()] = {}
            w_die[data_die.iat[area_index, 0].strip()][col_die_l[col_index - 2]] = data_die.iat[area_index, col_index]/1000
```

生育率处理方法类似，但由于生育率仅跟育龄妇女有关，因此生育率导入更为简单，需要注意的是，统计数据多以%为单位，因此在实际计算时需要除以 1000：

```
born_rate_dict = {}
col_born_l = data_born.columns.tolist()
for area_index in range(0, 17):
    for col_index in range(1, 8):
        if data_born.iat[area_index, 0].strip() not in born_rate_dict:
            born_rate_dict[data_born.iat[area_index, 0].strip()] = {}
        born_rate_dict[data_born.iat[area_index, 0].strip()][col_born_l[col_index]] = data_born.iat[
            area_index, col_index] / 1000
```



## 4.2 人口统计与计算：死亡人口和新出生人口的计算

函数首先计算每年的人口数量`per\_year\_num`，即上一个五年段的总人口数量除以5。这是因为我们假设上一个年龄段各个年龄人数一致。然后，根据每年的存活率和死亡率，计算当前五年段的死亡人口数量，并将结果累加到变量`this\_num`中。最后，函数返回`this\_num`作为当前五年段的死亡人口数量。

```
def this5year_num(last_num, last_die_rate, this_die_rate):
    this_num = 0
    per_year_num = last_num/5
    this_num += per_year_num * ((1 - last_die_rate) ** 1) * ((1 - this_die_rate) ** 4)
    this_num += per_year_num * ((1 - last_die_rate) ** 2) * ((1 - this_die_rate) ** 3)
    this_num += per_year_num * ((1 - last_die_rate) ** 3) * ((1 - this_die_rate) ** 2)
    this_num += per_year_num * ((1 - last_die_rate) ** 4) * ((1 - this_die_rate) ** 1)
    this_num += per_year_num * ((1 - last_die_rate) ** 5) * ((1 - this_die_rate) ** 0)
    return this_num
```

接着，函数`make\_new\_num`计算新出生人口的数量。在循环中，根据不同的育龄段和出生率，计算新出生人口的数量，并根据性别比例将其分为男性和女性。

```
def make_new_num(year):
    for area, w_area_datas in w_year_list[year].items():
        last_born_per5year = -1
        last_born_rate = -1
        girl_rate = 100 / (100 + sex_rate[area])
        boy_rate = sex_rate[area] / (100 + sex_rate[area])
        new_t_num = 0
        new_w_num = 0
        new_m_num = 0
        for born_per5year, born_rate in born_rate_dict[area].items(): # born_per5year: 育龄的每一段(eg:15-19岁)
            if born_per5year == '15-19岁':
                pass
            elif born_per5year == '45-49岁':
                last_peryear_num = w_area_datas[last_born_per5year] / 5.0
                new_t_num += last_peryear_num * (last_born_rate * 15)
                new_w_num += last_peryear_num * (last_born_rate * 15) * girl_rate
                new_m_num += last_peryear_num * (last_born_rate * 15) * boy_rate
            else:
                last_peryear_num = w_area_datas[last_born_per5year] / 5.0
                new_t_num += last_peryear_num * (last_born_rate * 15 + born_rate * 10)
                new_w_num += last_peryear_num * (last_born_rate * 15 + born_rate * 10) * girl_rate
                new_m_num += last_peryear_num * (last_born_rate * 15 + born_rate * 10) * boy_rate
            last_born_per5year = born_per5year
            last_born_rate = born_rate
        t_year_list[year][area]['0-4岁'] = new_t_num
        m_year_list[year][area]['0-4岁'] = new_m_num
        w_year_list[year][area]['0-4岁'] = new_w_num
```

其中，性别比导入如下：



```
sex_rate = {
    '总计': 109.12,
    '黄浦区': 108.14,
    '徐汇区': 116.81,
    '长宁区': 124.82,
    '静安区': 103.55,
    '普陀区': 116.02,
    '虹口区': 119.29,
    '杨浦区': 120.41,
    '闵行区': 110.94,
    '宝山区': 106.46,
    '嘉定区': 109.52,
    '浦东新区': 110.96,
    '金山区': 96.58,
    '松江区': 107.59,
    '青浦区': 97.79,
    '奉贤区': 110.31,
    '崇明区': 83.18
}
```

## 4.3 人口统计与计算：新一阶段人口数量的计算

计算新一阶段的人口数量，包括男性、女性和总体人口，每隔五年进行一次计算。

对于女性人口数量的计算，代码首先将上一个五年段的数据作为初始值。然后，遍历上一个五年段的各个地区和年龄段的人口数量，根据死亡率和出生率计算当前年份的人口数量，将计算得到的人口数量存储到`new\_year\_data[area][per5year]`中。男性人口和总人口同理。这段代码的核心是根据上一个五年段的人口数量、死亡率、出生率和男性比例，计算新一阶段的人口数量。此处仅展示女性人口代码，其余代码均在附录。

```
for year in range(2020, 2051, 5):
    # 女
    last_w_year_list = copy.deepcopy(w_year_list)
    if year in last_w_year_list:
        continue
    last_year = year - 5
    last_year_data = last_w_year_list[last_year]
    new_year_data = {}
    for area, area_datas in last_year_data.items():
        last_per5year_num = -1
        last_die_rate = -1
        new_year_data[area] = {}
        new_num = 0 # 新增人口
        for per5year, num in area_datas.items():
            die_rate = w_die[area][per5year]
            if last_per5year_num == -1: # 0~4岁
                new_year_data[area][per5year] = 0
            else:
                new_year_data[area][per5year] = this5year_num(last_per5year_num, last_die_rate, die_rate)
            last_per5year_num = num
            last_die_rate = die_rate
    w_year_list.update({year: new_year_data})
```

## 4.4 人口数量转换和输出

将人口数量转换为整数，并输出特定年份的总体、男性和女性人口数量。

```
def to_int(year_list):
    new_year_list = copy.deepcopy(year_list)
    for year, year_data in year_list.items():
        for area, area_data in year_data.items():
            for per5year, num in area_data.items():
                new_year_list[year][area][per5year] = int(num)
    return new_year_list

print("2025年总: " + str(to_int(t_year_list)[2025]))
print("2025年男: " + str(to_int(m_year_list)[2025]))
print("2025年女: " + str(to_int(w_year_list)[2025]))
print("2030年总: " + str(to_int(t_year_list)[2030]))
print("2030年男: " + str(to_int(m_year_list)[2030]))
print("2030年女: " + str(to_int(w_year_list)[2030]))
print("2035年总: " + str(to_int(t_year_list)[2035]))
print("2035年男: " + str(to_int(m_year_list)[2035]))
print("2035年女: " + str(to_int(w_year_list)[2035]))
print("2040年总: " + str(to_int(t_year_list)[2040]))
print("2040年男: " + str(to_int(m_year_list)[2040]))
print("2040年女: " + str(to_int(w_year_list)[2040]))
print("2045年总: " + str(to_int(t_year_list)[2045]))
print("2045年男: " + str(to_int(m_year_list)[2045]))
print("2045年女: " + str(to_int(w_year_list)[2045]))
print("2050年总: " + str(to_int(t_year_list)[2050]))
print("2050年男: " + str(to_int(m_year_list)[2050]))
print("2050年女: " + str(to_int(w_year_list)[2050]))
```

将得到的数据导入 Excel。（以 2025 年男数据为例）

通过分隔符将总人数与各区人数分开，后进行转换数据。

2025年男.txt

文件原始格式

65001: Unicode (UTF-8)

分隔符

--自定义--

数据类型检测

基于整个数据集

,

Column1	Column2	Column3
2025年男: {'总计': {'0-4岁': 275039, '5-9岁': 445518, '10-14岁': 115039, '15-19岁': 115039, '20-24岁': 115039, '25-29岁': 115039, '30-34岁': 115039, '35-39岁': 115039, '40-44岁': 115039, '45-49岁': 115039, '50-54岁': 115039, '55-59岁': 115039, '60-64岁': 115039, '65-69岁': 115039, '70-74岁': 115039, '75-79岁': 115039, '80-84岁': 115039, '85-89岁': 115039, '90-94岁': 115039, '95-99岁': 115039, '100岁': 115039}}	'黄浦区': {'0-4岁': 4609, '5-9岁': 8709, '10-14岁': 11503, '15-19岁': 11503, '20-24岁': 11503, '25-29岁': 11503, '30-34岁': 11503, '35-39岁': 11503, '40-44岁': 11503, '45-49岁': 11503, '50-54岁': 11503, '55-59岁': 11503, '60-64岁': 11503, '65-69岁': 11503, '70-74岁': 11503, '75-79岁': 11503, '80-84岁': 11503, '85-89岁': 11503, '90-94岁': 11503, '95-99岁': 11503, '100岁': 11503}}	'徐汇区': {'0-4岁': 11090, '5-9岁': 16639, '10-14岁': 11503, '15-19岁': 11503, '20-24岁': 11503, '25-29岁': 11503, '30-34岁': 11503, '35-39岁': 11503, '40-44岁': 11503, '45-49岁': 11503, '50-54岁': 11503, '55-59岁': 11503, '60-64岁': 11503, '65-69岁': 11503, '70-74岁': 11503, '75-79岁': 11503, '80-84岁': 11503, '85-89岁': 11503, '90-94岁': 11503, '95-99岁': 11503, '100岁': 11503}}

通过转置调整数据。



文件

开始

插入

页面布局

公式

数据

审阅

视图

帮助

表设计

查询

共享

表名称:

2025年男

调整表格大小

通过数据透视表汇总

删除重复值

转换为区域

插入切片器

导出

刷新

用浏览器打开

取消链接

属性

外部表数据

☒ 标题行

☐ 第一列

☒ 筛选按钮

☐ 汇总行

☐ 最后一列

☒ 边行

☐ 边列

表格样式选项

表格样式

A1

fx

	A	B	C	D	E	F	G	H	I	J	
1	Column1.1.1	Column1.1.2	Column1.2	Column1.3	Column1.4	Column1.5	Column1.6	Column1.7	Column1.8	Column1.9	Col
2	总计	275039	445518	479981	348278	391571	823975	1179198	1461102	1204569	971102
3	黄浦区	4609	8709	11503	10007	11738	24379	30621	35097	27729	221102
4	徐汇区	11090	16639	22381	18126	15878	35075	45168	49256	46210	371102
5	长宁区	7303	9885	12251	9483	9540	19125	29478	32550	28104	211102
6	静安区	7617	14795	18043	14217	12895	23994	35024	43589	40616	331102
7	普陀区	10711	19385	23308	17702	16008	29923	46318	58097	54141	421102
8	虹口区	6165	9875	12330	9941	10525	19771	27049	34165	30041	231102
9	杨浦区	12033	18173	22419	16463	21255	42350	43965	54863	49904	371102
10	闵行区	32134	51550	57209	41672	42259	93177	132935	158567	136851	111102
11	宝山区	23935	41044	44352	31259	33188	69634	99043	129309	113559	911102
12	嘉定区	21857	37390	34871	22769	28817	66252	100840	136757	106030	811102
13	浦东新区	69912	110797	117221	83363	89623	200767	303469	355408	284284	221102
14	金山区	6859	14095	14442	11205	14985	21980	32059	43795	36270	341102
15	松江区	28674	41307	40478	25994	32717	81371	113864	141769	103777	831102
16	青浦区	14733	23817	21015	14340	19293	44523	72691	92658	68993	581102
17	奉贤区	14559	21179	20283	14531	24518	41282	51848	71777	57151	501102
18	崇明区	3061	6874	7870	7198	8319	10363	14820	23463	20906	211102
19											
20											
21											
22											

查询 & ...

查询

连接

1个查询

2025年男

已加载 17行。

将各年数据处理后导入 Excel 表格，新建每一年度总表格。  
运用数据引用将男、女数据汇总到表格。

F3	fx	= '2025年男'!B2
G3	fx	= '2025年女'!B2

并通过函数计算总人口、总男性人口、总女性人口数。

B3	fx	=C3+D3
C3	fx	=F3+I3+L3+O3+R3+U3+X3+AA3+AD3+AG3+AJ3+AM3+AP3+AS3+AV3+AY3+BB3+BE3+BH3+BK3+BN3
D3	fx	=G3+J3+M3+P3+S3+V3+Y3+AB3+AE3+AH3+AK3+AN3+AQ3+AT3+AW3+AZ3+BC3+BF3+BI3+BL3+BO3

	A	B	C	D	E	F	G	H	I	J	K	L
1	地区	人口数				0-4岁		5-9岁				10-14岁
2		合计	男	女	小计	男	女	小计	男	女	小计	男
3	总计	25465996	12757823	12708173	527091	275039	252052	856798	445518	411280	913677	479981
4	黄浦区	659144	329475	329669	8871	4609	4262	16723	8709	8014	21855	11503

	A	B	C	D	E	F	G	H	I	J	K	L
1	地区	人口数				0-4岁		5-9岁				10-14岁
2		合计	男	女	小计	男	女	小计	男	女	小计	男
3	总计	25465996	12757823	12708173	527091	275039	252052	856798	445518	411280	913677	479981

最后得到各年度总表格数据

2025总	2030总	2035总	2040总	2045总	2050总
-------	-------	-------	-------	-------	-------





按照  $80\text{ 周岁老龄化系数} = \frac{80\text{ 周岁以上人口数}}{\text{总人口数}} \times 100\%$  公式计算各年份数据于 G 列。



按照  $\text{老年抚养比} = \frac{65\text{ 周岁以上人口数}}{15\sim 64\text{ 岁的劳动年龄人口数}} \times 100\%$  公式计算各年份数据于 H 列。

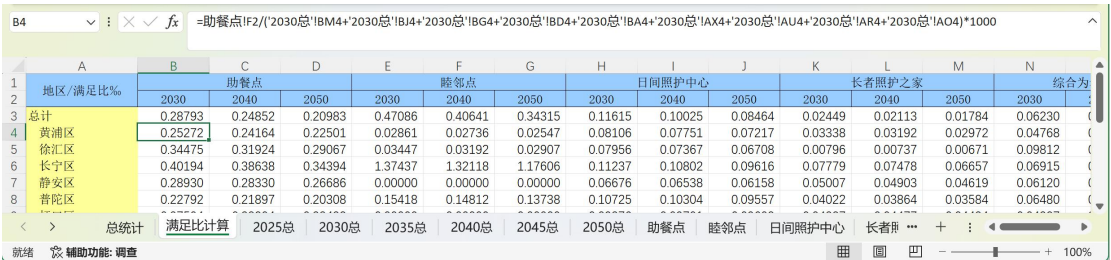


最终结果：

年份	总人数/人	60+周岁人数/人	65+周岁人数/人	60周岁老龄化系数/%	65周岁老龄化系数/%	80周岁老龄化系数/%	老年抚养比/%
2025	25465996	7652726	5949160	30.05	23.36	18.22	34.55
2030	24169429	7800576	6059053	32.27	25.07	16.89	37.03
2035	23375263	8510269	6785031	36.41	29.03	23.57	44.03
2040	22309875	9037624	7238324	40.51	32.44	27.62	51.19
2045	20974461	9741358	7521294	46.44	35.86	27.63	59.56
2050	19420514	10703694	8055884	55.12	41.48	25.81	75.83

5.3 各区老年服务满足比计算

以老年助餐服务场所为例，按照  $\text{满足比} = \frac{\text{服务设施数}}{60\text{ 周岁及以上人口数}} \times 1000\%$  计算得出相应年份数据。



5.4 老年服务设施数据归一化

为了更好的去评估一个区的综合服务水平以及方便可视化展示，将预测 2030、2040、2050 的各年上海市及各辖区老年服务设施满足比进行最大最小值归一化处理，按照  $X_{norm} = \frac{X - Min}{Max - Min}$ ，将结果映射到 [0, 1] 区间，实现原始数据的等比缩放，便于将五项老年服务进行比较和综合。通过五项老年服务设施归一化后结果按辖区求和，得到相应年度该区的老年服务设施满足比。

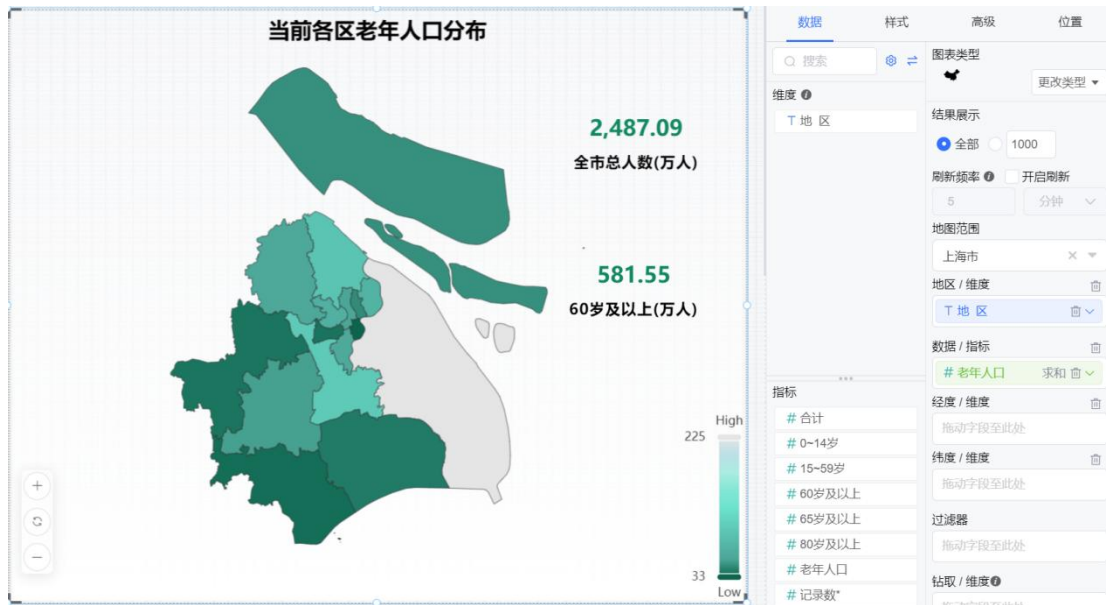


地区	上海市	黄浦区	徐汇区	长宁区	静安区	普陀区	虹口区	杨浦区	闵行区	宝山区	嘉定区	浦东新区	金山区	松江区	青浦区	奉贤区	崇明区
2030	1.209	0.767	1.263	2.594	1.186	1.197	1.092	0.875	1.512	0.427	1.799	0.684	1.498	1.142	2.392	1.260	2.288
2040	1.197	0.918	1.233	2.955	1.339	1.349	1.508	1.090	1.111	0.416	1.580	0.677	1.647	0.861	2.198	1.143	3.017
2050	1.179	1.126	1.277	3.026	1.552	1.506	2.003	1.349	0.863	0.488	1.406	0.636	1.917	0.664	1.908	1.071	4.168

## 6 可视化

### 6.1 上海市老年人口分布现状

根据上海统计局发布的《2022 年上海统计年鉴》，对上海各区老年人口数据进行分析，将老年人口数量分布直观展示在上海市地图上：



再结合前期数据处理得到的老年人口占比：

地 区	崇明区	虹口区	杨浦区	静安区	普陀区	长宁区	徐汇区	黄浦区
老年人口(万人)	48.52	46.84	72.91	57.02	69.46	38.19	60.49	32.76
老年人口占比(%)	76.06	61.83	58.68	58.44	56.03	55.10	54.34	49.49

地 区	金山区	宝山区	浦东新区	闵行区	奉贤区	嘉定区	青浦区	松江区
老年人口(万人)	36.19	91.81	224.63	97.96	40.89	58.96	38.76	54.68
老年人口占比(%)	43.98	41.07	39.54	36.92	35.84	32.14	30.49	28.63

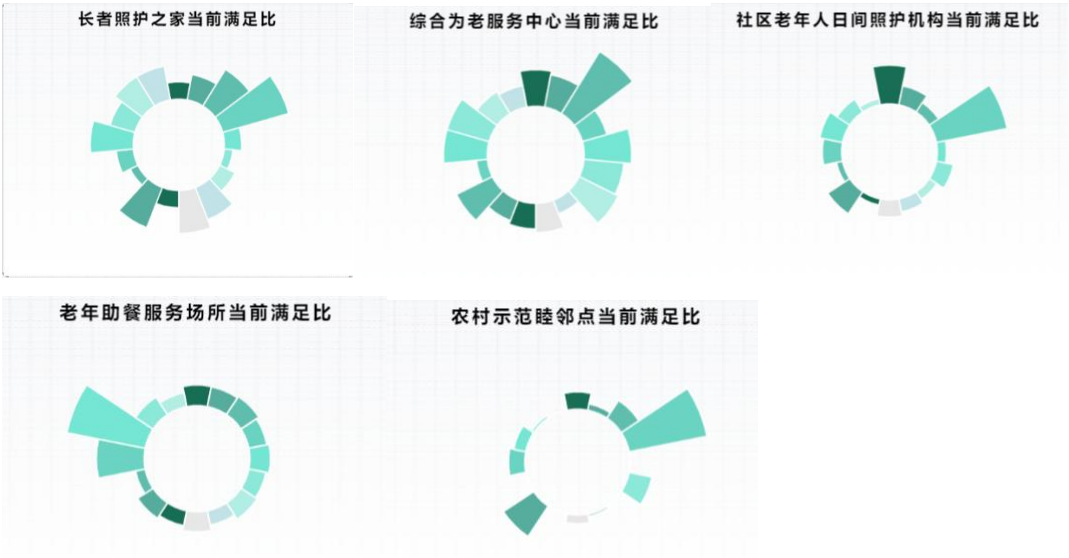
分析发现，中心城区为老年人口密集区域，老龄化趋势严峻，但老年人口的分布有从中心城区向郊区转移的趋势，老年人口向更为均衡的方向发展。

## 6.2 当前各类设施满足比

根据各区老年人口数和各区每种社区养老设施（城镇）和睦邻点（农村）数量，计算得出当前每种设施的满足比如下：

地区/满足比%	助餐点	睦邻点	日间照护中心	长者照护之家	综合为老服务中心
黄浦区	0.18	0.02	0.06	0.02	0.03
徐汇区	0.38	0.04	0.09	0.01	0.11
长宁区	0.40	1.38	0.11	0.08	0.07
静安区	0.28	0.00	0.06	0.05	0.06
普陀区	0.27	0.18	0.13	0.05	0.08
虹口区	0.36	0.00	0.09	0.04	0.04
杨浦区	0.26	0.00	0.06	0.04	0.06
闵行区	0.37	0.41	0.19	0.05	0.20
宝山区	0.39	0.18	0.15	0.04	0.07
嘉定区	1.47	0.81	0.23	0.03	0.12
浦东新区	0.37	0.56	0.17	0.02	0.07
金山区	0.38	1.12	0.38	0.02	0.10
松江区	0.88	1.06	0.19	0.05	0.12
青浦区	0.34	5.15	0.71	0.02	0.13
奉贤区	0.34	2.36	0.26	0.01	0.11
崇明区	0.32	1.51	0.17	0.02	0.12

以南丁格尔玫瑰图展示：



分析可知，上海市各设施满足比达到 0.5 及以上的仅占总体的 12.5%，其中睦邻点在郊区的覆盖度较高，而社区养老设施在整个上海市的覆盖度较低，且空间分布不均。

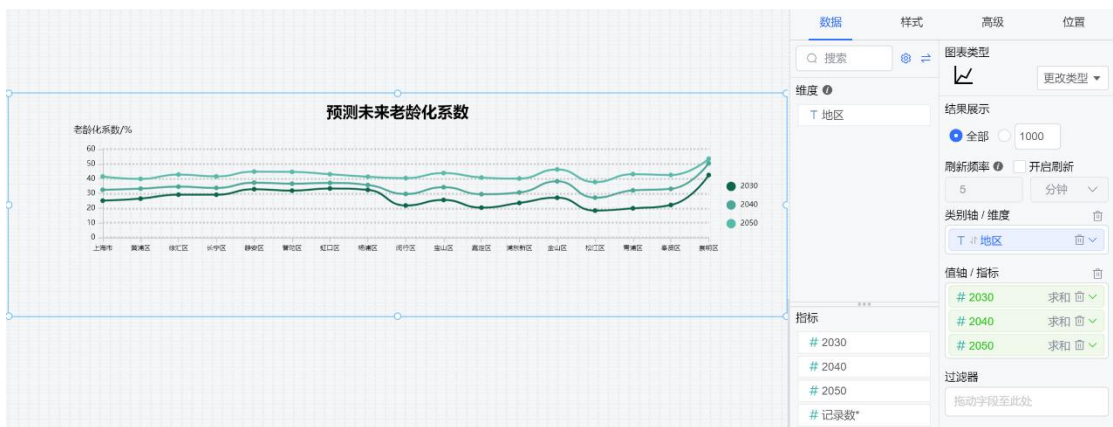
## 6.3 预测未来年龄性别结构

将预测后 2030、2040、2050 年龄性别数据整理后导入 Dataease，通过对称柱状图将男女性别分别呈现，更直观展示各年龄段人数的性别差异。



## 6.4 预测出的未来老龄化系数

综合 2030、2040、2050 年上海市及各辖区的老龄化系数（65 岁以上人口占人口总数百分比），在 Dataease 中通过基础折线图展示各区差距及变化趋势。

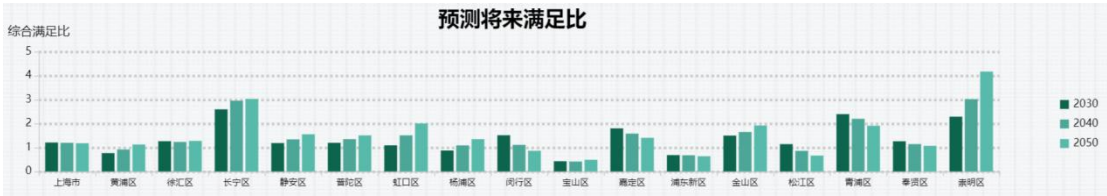


## 6.5 预测未来老年服务设施满足比

将预测 2030、2040、2050 的各年上海市及各辖区老年服务设施满足比进行最大最小值归一化处理，按照  $X_{norm} = \frac{X - Min}{Max - Min}$ ，将结果映射到 [0, 1] 区间，实现原始数据的等比缩放，便于将五项老年服务进行比较和综合。通过五项老年服务设施归一化后结果按辖区求和，得到相应年度该区的老年服务设施满足比。

地区	上海市	黄浦区	徐汇区	长宁区	静安区	普陀区	虹口区	杨浦区	闵行区	宝山区	嘉定区	浦东新区	金山区	松江区	青浦区	奉贤区	崇明区
2030	1.209	0.767	1.263	2.594	1.186	1.197	1.092	0.875	1.512	0.427	1.799	0.684	1.498	1.142	2.392	1.260	2.288
2040	1.197	0.918	1.233	2.955	1.339	1.349	1.508	1.090	1.111	0.416	1.580	0.677	1.647	0.861	2.198	1.143	3.017
2050	1.179	1.126	1.277	3.026	1.552	1.506	2.003	1.349	0.863	0.488	1.406	0.636	1.917	0.664	1.908	1.071	4.168

将数据导入 Dataease，绘制基础柱状图，展示预测各年度满足比、各区满足比差距及上海市及各辖区满足比变化。



## 7 结论

根据上述数据分析，未来上海的人口结构中老年人口比例持续攀升，独生子女父母进入老龄阶段，纯老家庭及独居老年人群体不断壮大，再加上目前我国生育率较低，家庭养老功能弱化，社会养老负担加重。

但目前上海社区养老方面还有许多问题亟待解决，由于本项目仅做了数量及分布方面的研究，此处主要分析这两方面存在的问题：

1. 养老设施不足且分布不平衡：社区养老设施整体数量不高，全市覆盖度较低，供给不足；在市中心地区较为齐备，但在郊区或新城镇相对滞后，空间分布不均衡；此外种类不均的问题也逐渐凸显，且存在质量参差不齐的情况。
2. 养老设施规划不足：未来上海老年人口将继续增长，但养老设施规划和建设未能跟上老年人口的增长速度，且缺少区域间的统筹和总体的布局规划，仅以单一指标为建设目标，造成养老设施供需矛盾和空间失衡交错并存。

虽然目前上海社区养老设施发展态势良好，但由于老年人口基数不断增加和老年人口分布不均衡，如何处理供需关系，仍然是今后上海养老服务工作面临的突出问题。

## 8 建议

随着上海养老事业的发展，社会养老需综合考虑老年群体多样化的养老需求以及城乡老年群体的养老需求差异以构建布局均衡、功能丰富、设施齐全的各类养老服务设施体系，通过统筹规划，避免设施布局的随意性，提高设施空间布局的均衡性，做到公平及效率最大化。基于本文的研究结果，提出三点建议：

### 1. 提高社区养老设施覆盖率

政府应加大对社区养老设施的投入，提高养老设施的数量和覆盖率，制定明确的覆盖率目标，例如未来五年内在市区内以及郊区中老年人口密集的地区做到全面覆盖，按照服务均等化的要求，扫除服务盲区，确保所有老年人能够享受基本养老服务。同时，可以鼓励社会力量参与养老设施的建设和管理，提高设施的

运营效率和服务质量。此外还可以挖掘和利用社区中原有的公共设施资源和闲置物业资源，提高资源使用效率，增加社区养老设施的可获得性与便捷性。

## 2. 建立合理指标体系，加强区域统筹和布局规划

政府应将养老设施规划纳入城市总体规划，实现养老设施供需的长期平衡。通过建立一套指标体系，考虑老年人口分布、空间布局、用地特点、养老设施数量和质量等多种因素，以指导养老设施的规划和建设，实现规划的多维度平衡，确保养老设施的供需平衡和空间均衡。此外，还应建立跨区域养老设施共建共享机制，促进各区之间的养老设施规划协调和合作，优化资源配置，解决跨区域供需矛盾。

在规划和建设过程中，需要注重养老设施的质量和功能多样性，满足老年人的不同需求，提高养老服务的专业化和个性化水平。

## 3. 有效整合资源，融合机构养老和社区养老

社区养老还可以考虑与机构养老合作，做到互为补充支撑，适度打破社区养老设施与机构养老设施的界限，融合二者的功能。例如，将机构养老设施同时作为老年人助餐点，为周围老年人提供配餐、送餐服务，节省空间以及人力资源；在入住率较低或是规模较大的机构养老设施中划分一小部分的床位用于收住临时需要短期照护的老人；将养老机构中的图书室、阅览室、活动中心等向社区老人开放等。此外，社区居家养老设施可与社区内的其它公共设施进行综合设置，可以共享设施及空间资源，降低建设成本和运营成本。