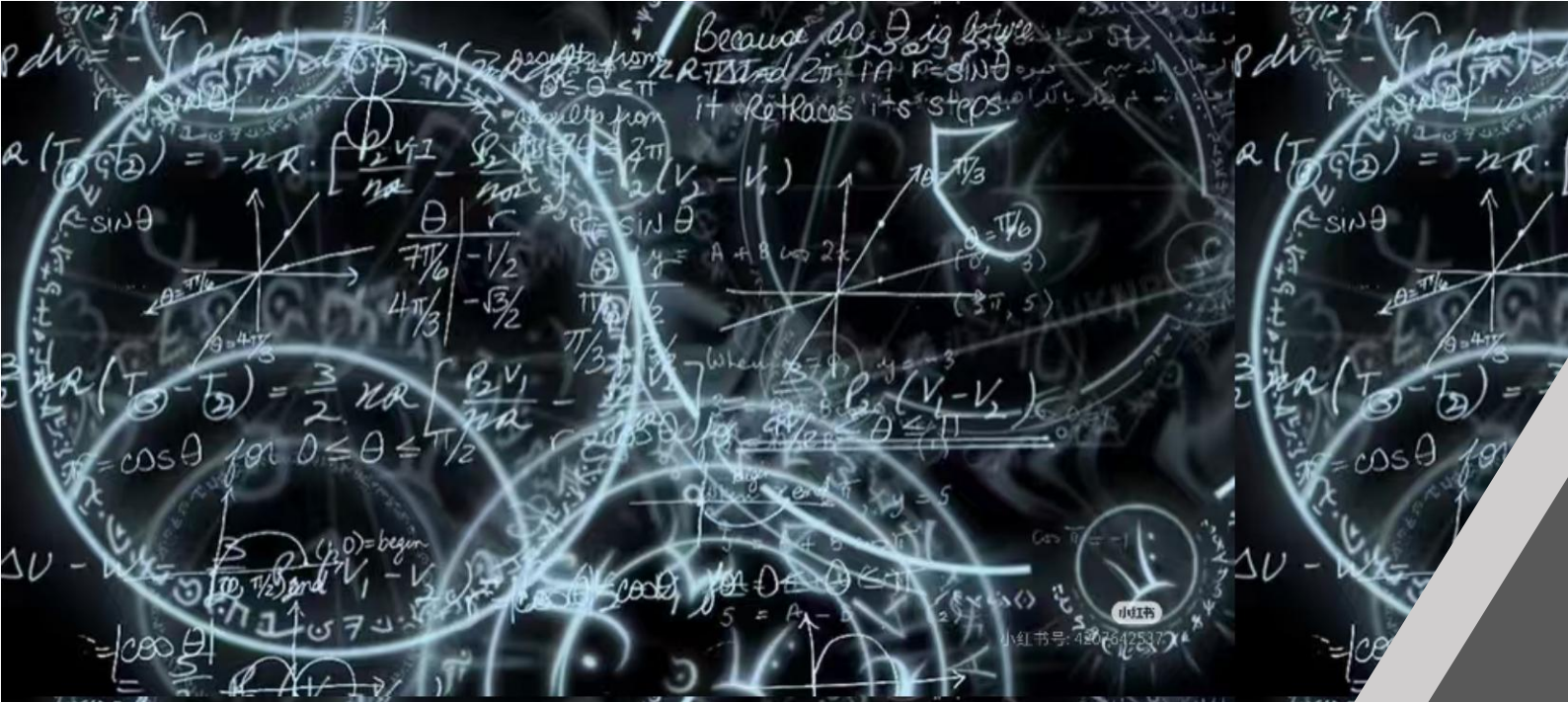

模 型 说

Model simplifies the problem

多 元 回 归 模 型

多元回归模型在经济管理领域中发挥着重要作用，它通过分析多个自变量对一个因变量的影响，帮助决策者理解复杂经济现象的本质。该模型能够量化各类因素对经济结果的综合影响，从而提供更精确的预测和分析。它帮助决策者量化这些影响，优化资源配置，提高决策精度，支持成本控制、投资回报分析和政策评估等。通过多元回归模型，管理者能够更准确地理解经济现象，制定更有效的策略。

一、模型介绍

1.1 多元回归模型原理

多元回归模型，（multivariable linear regression model ）是一种用于研究一个因变量与多个自变量之间关系的统计方法。

在实际经济问题中，一个变量的变化往往会受到多个变量的影响。回归分析是应用统计原理研究因变量与自变量间函数关系的分析方法。

回归模型的一般形式：

$$y = f(x_1，\ x_2，\ \cdots，\ x_p) + \varepsilon$$

其中，y为因变量，x_1，x_2，…，x_p为自变量，ε称为随机误差，一般要求它的数学期望为 0。

$f(x_1，\ x_2，\ \cdots，\ x_p)$称为$y$对$x_1$，$x_2$，…，$x_p$的（均值）回归函数。

线性回归模型：

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon$$

其中，β_0，β_1，…，β_p为未知参数，β_0为回归常数，β_1，…，β_p为回归系数。通过观测得数据：x_{i1}，x_{i2}，…，x_{ip}；y_i，$i = 1$，2，3，…，n，即有：

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i，\ i = 1，2，3，\cdots，\mathrm{n}$$

再根据这些数据估计β_0，β_1，…，β_p的值。

1.2 多元回归模型假设

多元回归模型需要以下基本假设：

变量x_1，x_2，…，x_p是非随机变量，观测值x_{i1}，x_{i2}，…，x_{ip}是常数

高斯—马尔可夫（Gauss——Markor）条件：G-M 条件（等方差及不相关的假定）

$$\begin{cases} E(\varepsilon_i)=0，\ i=1，\ 2，\ 3，\ \cdots，\ n \\ cov(\varepsilon_i,\varepsilon_j)=\begin{cases} 0，\ i\neq j \\ \sigma^2，\ i=j \end{cases} \end{cases}$$

误差的正态性假设：

$$\begin{cases} \varepsilon_i \sim N(0，\ \sigma^2) \\ \varepsilon_1，\ \varepsilon_2，\ \cdots，\ \varepsilon_n 互相独立 \end{cases}$$

1.3 多元回归模型参数估计

多元线性回归模型的参数估计方法主要采用最小二乘法（Ordinary Least Squares, OLS）。

在模型中，自变量和因变量之间的线性关系通过线性加权组合来建立，以预测因变量的值。参数估计的目标是找到使得误差的平方和最小的回归系数，这通过最小二乘法实现。最小二乘法通过最小化残差的平方和来确定回归系数的值，其中残差是观测值与回归模型预测值之间的差异。

为了进行最小二乘法参数估计，需要计算回归模型的预测值，这可以表示为：

$$\hat{y}=\beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_px_p+\varepsilon$$

其中，$\hat{y}$ 是因变量的预测值。

$$Y=\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}，\ X=\begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}，\ B=\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}，\ E=\begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

参数估计的目标可以表示为最小化观测值与预测值之间的误差平方和。通过对目标函数进行求导，可以得到参数的估计值。具体地，参数向量β可以通过以下公式计算：

$$XB=Y \ \Rightarrow \ B=(X^TX)^{-1}X^TY$$

其中，X 是自变量的矩阵，Y 是因变量的向量，上标 T 表示矩阵的转置。

然而，在实际应用中，数据往往存在噪声和异常值，这可能导致参数估计的不准确性。为了解决这个问题，可以采用正则化方法，如岭回归（Ridge Regression）和 LASSO 回归（Least Absolute Shrinkage and Selection Operator Regression）。这些方法通过在目标函数中引入正则化项，可以降低估计结果对噪声和异常值的敏感性。

也可使用极大似然估计计算参数向量β。极大似然估计的应用需要满足一定的条件，例如误差项的分布需要是已知的或者可以合理假设的。在多元线性回归模型中，如果误差项服从正态分布，那么最小二乘估计（Least Squares Estimate，

LSE）实际上是极大似然估计的一个特例。这意味着，在正态假设下，通过最小二乘法得到的参数估计与通过极大似然估计得到的结果是一致的。

二、模型建立

2.1 确定研究问题和选择变量

首先需要明确研究的目的和问题，以及可用数据的类型和范围。这包括确定自变量和因变量，以及收集相关的数据集。

2.2 数据收集与处理

在进行回归分析之前，可能需要对数据进行一些预处理，如缺失值处理、异常值处理、数据标准化等。这是因为多元回归模型要求数据具有一定的规范性和一致性。

根据问题的需求和数据的特性，选择合适的多元回归模型。这可能涉及到选择线性模型、非线性模型等。在多元线性回归中，需要确定自变量的数量和类型，以及它们与因变量之间的关系。

2.3 模型评估与优化

对建立的模型进行评估，包括检查模型的拟合优度、检验模型的假设条件是否满足等。这可以通过各种统计检验和方法来完成，例如 R 方值、F 检验等。

其中，DW 检验，全称 Durbin-Watson 检验，是一种用于检测回归模型中残差项独立性的统计学方法。如果残差项不是独立的，那么回归模型的预测结果就可能不准确。DW 检验通过计算 Durbin-Watson 统计量来检测残差项的独立性。

Durbin-Watson 统计量的计算公式为：

$$DW = \frac{\sum_{i=2}^{n} (e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2}$$

其中 d(i)和 d(i+1)是残差项，r(i)和 r(i+1)是自变量与因变量的相关系数 1。

DW 统计量的取值范围及意义：

（1）DW 接近 0：表示残差项存在完全正自相关。

（2）DW 接近 2：表示残差项是独立的，即无自相关。

（3）DW 接近 4：表示残差项存在完全负自相关。

（4）DW 在 2 到 4 之间：说明存在负的自相关。

在实际应用中，一般认为 DW 值在 1.5 到 2.5 之间即可说明无自相关现象

根据模型评估的结果，可能需要对模型进行优化，例如通过变量选择、模型简化等方法来改进模型的性能。

模型验证和优化后，就可以将其应用于新的数据或实际情况中，进行预测或

解释。

三、代码实现

3.1 python 代码实现

在 Python 中，建立多元回归模型通常使用 scikit-learn 库，它是一个功能强大的机器学习库，提供了丰富的算法和工具，包括线性回归模型。以下是使用 scikit-learn 建立多元回归模型的基本步骤：

1. 导入必要的库：首先，需要导入 scikit-learn 中的 LinearRegression 类，以及用于数据处理的 pandas 库（如果数据是存储在 DataFrame 中的）。

2. 准备数据：准备你的数据集，通常包括自变量（X）和因变量（y）。这些数据可以是 numpy 数组、pandas DataFrame 或 scipy 稀疏矩阵。

3. 划分数据集：使用 scikit-learn 的 train_test_split 函数将数据划分为训练集和测试集。

4. 创建线性回归模型：实例化 LinearRegression 类来创建一个线性回归模型。

5. 训练模型：使用训练集（X_train, y_train）来训练模型，通过调用模型的 fit 方法。

6. 评估模型：使用测试集（X_test, y_test）来评估模型的性能，可以通过计算预测值和实际值之间的误差来实现。

7. 预测新数据：使用训练好的模型对新数据进行预测。

下面是一个简单的示例代码：

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# 加载数据
data = pd.read_csv('your_data.csv')
X = data[['feature1', 'feature2', 'feature3']]  # 自变量
y = data['target']  # 因变量

# 划分数据集
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 创建线性回归模型
model = LinearRegression()

# 训练模型
model.fit(X_train, y_train)

# 评估模型
y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# 预测新数据
new_data = pd.DataFrame({'feature1': [value1], 'feature2': [value2], 'feature3': [value3]})
new_pred = model.predict(new_data)
print(f'New Prediction: {new_pred}')
```

在这个示例中，你需要将'your_data.csv'替换为你的数据集文件路径，并根据你的数据集调整特征名称和目标变量名称。此外，value1、value2 和 value3 应该替换为你想要预测的新数据的特征值。

通常计算方差膨胀因子（Variance Inflation Factor, VIF）的方法。VIF 是检验多元线性回归模型中变量间是否存在多重共线性的一种常用方法。VIF 值越大，说明共线性问题越严重。

以下是一个使用 Python 计算 VIF 的示例：

```
#  计算 VIF
vif  =  pd.DataFrame()
vif['VIF'] = [variance_inflation_factor(X.values, i) for i in range(X.shape)]
vif['features'] = X.columns

print(vif)
```

VIF 值的标准判断是：

VIF < 5：不存在共线性问题。

5 <= VIF < 10：存在中等程度的共线性问题。

VIF >= 10：存在严重的共线性问题。

如果 VIF 值表明存在共线性问题，你可能需要考虑使用岭回归、主成分回归或其他方法来处理共线性。

3.2 R 语言代码实现

在 R 语言中进行多元回归分析，主要使用 lm()函数。这个函数可以用来拟合线性模型，其中可以包括一个或多个自变量。下面是如何使用 R 语言进行多元回归分析的基本步骤：

1.准备数据：首先，你需要准备包含因变量和自变量的数据集。这些数据可以是 CSV 文件、数据框（data.frame）或其他 R 中的数据结构。

2.读取数据（如果数据在外部文件中）：使用 read.csv()或其他适当的函数来读取数据。

3.拟合多元线性回归模型：使用 lm()函数来拟合模型。函数的第一个参数是一个公式,指定了因变量和自变量,形式为因变量 ~ 自变量 1 + 自变量 2 + ... + 自变量 n。第二个参数是包含这些变量的数据框。

4.查看模型摘要：使用 summary()函数来查看模型的详细摘要，包括系数估计、R 平方值、F 统计量等。

5.进行模型诊断：检查模型的残差、拟合优度等，以确保模型的有效性和假设的满足。

6.预测新数据：使用 predict()函数来根据模型对新数据集进行预测。

下面是一个简单的例子：

```
#  假设你有一个名为 mydata 的数据框，其中包含因变量 y 和自变量 x1、x2
```

```
2.
3. #  拟合多元线性回归模型
4. model  <-  lm(y  ~  x1  +  x2,  data  =  mydata)
5.
6. #  查看模型摘要
7. summary(model)
8.
9. #  对新数据集进行预测
10.#  假设 newdata 是一个包含 x1 和 x2 的新数据框
11.predictions  <-  predict(model,  newdata  =  newdata)
```

向后选择法是一种逐步回归分析方法，用于在多元回归分析中自动选择和删除不重要的自变量。这种方法从包含所有候选自变量的完整模型开始，然后逐步删除最不显著的变量，直到达到某个停止准则为止。以下是如何在 R 语言中进行退步回归分析的基本步骤：

1.评估变量显著性：查看模型摘要，特别是自变量的 p 值，以确定哪些变量在统计上显著。

2.逐步删除不显著变量：使用 R 中的函数（如 step()函数）逐步删除最不显著的变量。step()函数可以自动执行这一过程，但它默认使用的是向前选择法或双向选择法，但你可以通过设置参数来使用后退法。

3.确定停止准则：确定何时停止删除变量的准则。这可以是基于变量的 p 值（例如，只保留 p 值小于某个阈值的变量），或者是基于模型的其他统计指标（如 AIC、BIC 等）。

4.检查最终模型：在删除所有不显著的变量后，检查最终模型的摘要，确保剩余的自变量在统计上显著，并且模型具有良好的拟合优度和预测能力。

5.进行模型诊断：对最终模型进行诊断，以检查其假设的满足情况（如线性关系、同方差性、独立性等）。

需要注意的是，虽然退步回归分析可以帮助减少模型中的变量数量，但它也有一些潜在的问题。特别是，由于它是在已经包含所有变量的完整模型基础上进行的，因此它可能更容易受到多重共线性的影响。此外，删除变量的顺序也可能影响最终模型的选择。

在 R 中，虽然 step()函数可能不直接支持纯粹的后退法，但你可以通过适当设置参数或使用其他包（如 leaps 包中的 regsubsets()函数）来实现类似的功能。然而，这些高级功能可能需要更详细的了解和学习。

以下是 step 函数示例代码：

```
1. #  假设你有一个数据框 df，其中包含因变量 y 和自变量 x1,  x2,  x3
2.
3. #  首先，使用 lm 函数拟合一个包含所有变量的完整模型
4. full_model  <-  lm(y  ~  x1  +  x2  +  x3,  data  =  df)
5.
6. #  然后，使用 step 函数执行后退法逐步回归
```

```
# scope 参数定义了变量选择的范围，这里我们使用默认的~ .，表示考虑所有变量
# direction="backward"指定使用后退法
backward_model <- step(full_model, direction="backward")

#查看后退法逐步回归的结果
summary(backward_model)
```

四、应用案例

4.1 多元回归模型人口预测

人口数量的变化与多因素有关，如出生率、死亡率、房价、性别比等因素。先获取数据、处理数据。

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
import numpy as np

# 读取指定的Sheet
df = pd.read_excel('D://1978-2024.xlsx', sheet_name='中国房价')

# 确保特征变量和目标变量为数值类型
df["人口出生率"] = pd.to_numeric(df["人口出生率"], errors='coerce')
df["人口死亡率"] = pd.to_numeric(df["人口死亡率"], errors='coerce')
df["房价"] = pd.to_numeric(df["房价"], errors='coerce')
df["年末总人口"] = pd.to_numeric(df["年末总人口"], errors='coerce')

# 去除包含缺失值的行
df.dropna(subset=["人口出生率", "人口死亡率", "房价", "年末总人口"], inplace=True)

# 特征变量和目标变量
X_china = df[["年份", "人口出生率", "人口死亡率", "房价"]]
y_china = df["年末总人口"]

# 创建回归模型
model_china = LinearRegression()

# 拟合模型
model_china.fit(X_china, y_china)

# 创建多项式特征转换器
poly = PolynomialFeatures(degree=2, include_bias=False)

```

对于不符合线性规则的数据，可以建立多项式特征转换器，捕捉特征之间的非线性关系。

$$y_{\text{future}} = \beta_0 + \beta_1 x_{\text{future}} + \beta_2 x_{\text{future}}^2$$

然后通过 LinearRegression 函数进行回归模型拟合。

```python
# 创建函数用于预测未来特征变量值
def predict_future_values(df, feature, start_year, end_year):
    # 准备数据
    X = df[["年份"]]
    y = df[feature]

    # 创建多项式特征
    X_poly = poly.fit_transform(X)

    # 创建回归模型
    model = LinearRegression()

    # 拟合模型
    model.fit(X_poly, y)

    # 预测未来值
    future_years = pd.DataFrame({"年份": np.arange(start_year, end_year + 1)})
    future_years_poly = poly.transform(future_years)
    future_values = model.predict(future_years_poly)

    return future_values

# 预测未来的特征变量值
future_years = np.arange(2025, 2031)
future_birth_rate = predict_future_values(df, "人口出生率", 2025, 2030)
future_death_rate = predict_future_values(df, "人口死亡率", 2025, 2030)
future_gender_ratio = predict_future_values(df, "房价", 2025, 2030)

```

通过拟合的回归模型进行预测：

```python
# 创建包含未来年份和预测特征变量值的数据框
future_data = pd.DataFrame({
    "年份": future_years,
    "人口出生率": future_birth_rate,
    "人口死亡率": future_death_rate,
    "房价": future_gender_ratio
})

# 打印预测的出生率和死亡率
print("未来的出生率和死亡率预测：")
print(future_data)

# 预测未来人口
predictions_china = model_china.predict(future_data)

# 输出预测结果
predictions = pd.DataFrame({
    "Year": future_years,
    "Predicted_China_Population": predictions_china
})

# 显示预测结果
print("未来的总人口预测：")
print(predictions)

```

最后可进行检验：

```python
from sklearn.metrics import r2_score
import statsmodels.api as sm

# 计算R平方值
r2 = r2_score(y_china, model_china.predict(X_china))
print(f"R平方值: {r2}")

# 使用statsmodels进行详细回归分析
X_china_with_const = sm.add_constant(X_china)
china_sm = sm.OLS(y_china, X_china_with_const).fit()

# 打印回归结果，包括F检验和t检验
print(china_sm.summary())
```

可以运行获得结果：

```
R平方值: 0.9952932014755441
                            OLS Regression Results
==============================================================================
Dep. Variable:                  年末总人口   R-squared:                       0.995
Model:                            OLS   Adj. R-squared:                  0.994
Method:                 Least Squares   F-statistic:                     687.2
Date:                Sun, 02 Jun 2024   Prob (F-statistic):           5.57e-15
Time:                        01:41:43   Log-Likelihood:                -124.79
No. Observations:                  18   AIC:                             259.6
Df Residuals:                      13   BIC:                             264.0
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -1.374e+06   2.86e+05     -4.796      0.000   -1.99e+06   -7.55e+05
年份          753.5639    145.036      5.196      0.000     440.232    1066.895
人口出生率       303.4769     64.620      4.696      0.000     163.874     443.080
人口死亡率     -1544.7317    685.420     -2.254      0.042   -3025.493     -63.971
房价           0.0933      0.307      0.304      0.766      -0.571       0.757
==============================================================================
Omnibus:                        1.475   Durbin-Watson:                   1.204
Prob(Omnibus):                  0.478   Jarque-Bera (JB):                1.233
Skew:                          -0.493   Prob(JB):                        0.540
Kurtosis:                       2.181   Cond. No.                     2.98e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.98e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

得到预测结果：

出生率的系数为 303.4769，表示出生率每增加一个单位，年末总人口平均增加 303476.9 人，t 值为 4.696，p 值为 0.000。

死亡率的系数为-1544.7317，表示死亡率每增加一个单位，年末总人口平均减少 1544731.7 人，t 值为-2.254，p 值为 0.042。

房价的系数为 0.0933，表示房价对年末总人口的影响较小，p 值为 0.766，表明这一系数在统计上不显著。

五、模型优缺点

优点

1、解释多个自变量的影响：

多元回归分析可以同时考虑多个自变量的影响，提供更全面的分析结果。能够揭示各个自变量对因变量的独立贡献，帮助理解复杂的因果关系。

2、灵活性高：

多元回归分析可以灵活去掉显著性不强的因素，增加新的因素进行分析。

3、统计显著性检验：

通过统计显著性检验（如 t 检验和 F 检验），可以帮助评估自变量的显著性及模型的整体拟合度。

缺点

1、线性假设限制：

多元回归假设变量之间的关系是线性的，这在很多实际情况下不一定成立。对非线性关系的适用性较差，需要扩展模型，如使用非线性回归。

2、解释性有限：

虽然可以解释自变量对因变量的线性影响，但无法揭示更深层次的因果关系。

参考书目

《应用回归分析》（第二版），王黎明等编著，复旦大学出版社，2018 .