In [7]:

```
# 导入操作系统库
import os
# 更改工作目录
os.chdir(r"D:\softwares\applied statistics\pythoncodelearning\chap1\sourcecode")
# 导入基础计算库
import numpy as np
# 导入绘图库
import matplotlib.pyplot as plt
# 导入泊松回归模型
from sklearn.linear_model import PoissonRegressor
# 导入管道工具
from sklearn.pipeline import make_pipeline, Pipeline
# 导入数据划分工具
from sklearn.model_selection import train_test_split
# 导入预处理工具
from sklearn.preprocessing import FunctionTransformer, OneHotEncoder
from sklearn.preprocessing import StandardScaler, KBinsDiscretizer
# 导入列变换工具
from sklearn.compose import ColumnTransformer
# 导入数据获取工具
from sklearn.datasets import fetch_openml
# 导入模型评估工具
from sklearn.metrics import mean_squared_error
# 导入绘图库中的字体管理包
from matplotlib import font_manager
# 实现中文字符正常显示
font = font_manager.FontProperties(fname=r"C:\Windows\Fonts\SimKai.ttf")
# 使用seaborn风格绘图
plt.style.use("seaborn-v0_8")
# 获取数据
df = fetch_openml(data_id=41214, as_frame=True, parser="pandas").frame
print(df.head())
# 新增一类
df["Frequency"] = df["ClaimNb"] / df["Exposure"]
# 标准化数据
# 构造管道，取对数变换，并且对其做标准化
log_scale_transformer = make_pipeline(
    FunctionTransformer(np.log, validate=False), StandardScaler()
)
# 列变换
linear_model_preprocessor = ColumnTransformer(
    [
        (
            "passthrough_numeric", # 变换名称
            "passthrough", # 变换方式
            ["BonusMalus"] # 变换对象
        ),
        (
            "binned_numeric", # 变换名称
            KBinsDiscretizer(n_bins=10, subsample=int(2e5), random_state=0), # 
            ["VehAge", "DrivAge"] # 变换对象
        ),
        (
            "log_scaled_numeric", # 变换名称
            log_scale_transformer, # 变换方式
            ["Density"] # 变换对象
        ),
        (
            "onehot_categorical", # 变换名称
            OneHotEncoder(), # 变换方式
            ["VehBrand", "VehPower", "VehGas", "Region", "Area"] # 变换对象
        )
    ],
    remainder="drop" # 剩下的变量的处理方式
)
# 划分数据集，这是对一个dataframe进行划分
df_train, df_test = train_test_split(df, test_size=0.33, random_state=0)
# 训练集的样本量
n_samples = df_train.shape[0]
# 管道工具，建立泊松回归模型
poisson_glm = Pipeline(
    [
        (
            "preprocessor",
            linear_model_preprocessor
        ), # 变量预处理
        (
            "regressor",
            PoissonRegressor(
                alpha=1e-12, solver="newton-cholesky"
            )
        ) # 泊松回归模型
    ]
)
# 模型拟合
poisson_glm.fit(
    df_train, df_train["Frequency"], # X， Y
    regressor__sample_weight=df_train["Exposure"] # 权重
)
# 模型预测
y_pred = poisson_glm.predict(df_test)
# MSE
mse = mean_squared_error(
    df_test["Frequency"],
    y_pred,
    sample_weight=df_test["Exposure"]
)
print("PoissonRegressor evaluation:", mse, sep="\n")
```

```
   IDpol  ClaimNb  Exposure Area  VehPower  VehAge  DrivAge  BonusMalus  \
0    1.0        1      0.10    D         5       0       55          50   
1    3.0        1      0.77    D         5       0       55          50   
2    5.0        1      0.75    B         6       2       52          50   
3   10.0        1      0.09    B         7       0       46          50   
4   11.0        1      0.84    B         7       0       46          50   

  VehBrand     VehGas  Density Region  
0      B12  'Regular'     1217    R82  
1      B12  'Regular'     1217    R82  
2      B12   'Diesel'       54    R22  
3      B12   'Diesel'       76    R72  
4      B12   'Diesel'       76    R72  
PoissonRegressor evaluation:
0.5598099236977848
```