

```

In [2]: import os
import pandas as pd
from scipy import stats
from scipy.stats import ttest_ind
import numpy as np

# Загрузка данных
current_dir = os.getcwd()
file_path = os.path.join(current_dir, 'churn.csv')
df = pd.read_csv(file_path)

# Предобработка данных
# Удаление дубликатов
df.drop_duplicates(inplace=True)

# Проверка на неявные пропуски в столбце TotalCharges
df["TotalCharges"] = pd.to_numeric(df["TotalCharges"], errors='coerce')
df['TotalCharges'].fillna(0, inplace=True)

# Изменение типа данных
df['MonthlyCharges'] = df['MonthlyCharges'].astype(float)
df['TotalCharges'] = df['TotalCharges'].astype(float)
df['tenure'] = df['tenure'].astype(int)

# Проверка гипотезы
# Разделение данных по статусу ухода клиента
churn_yes = df[df['Churn'] == 'Yes']['tenure']
churn_no = df[df['Churn'] == 'No']['tenure']

# T-тест
t_stat, p_value = ttest_ind(churn_yes, churn_no)
print(f't-statistic: {t_stat}, p-value: {p_value}')

# Анализ на нормальность распределения числовых признаков
_, p_value_mc = stats.shapiro(df['MonthlyCharges'].sample(500)) # использ
_, p_value_tc = stats.shapiro(df['TotalCharges'].sample(500))
_, p_value_tenure = stats.shapiro(df['tenure'].sample(500))

print(f'P-value для MonthlyCharges: {p_value_mc}')
print(f'P-value для TotalCharges: {p_value_tc}')
print(f'P-value для tenure: {p_value_tenure}')

t-statistic: -31.57955051135377, p-value: 7.99905796059022e-205
P-value для MonthlyCharges: 9.45315027157164e-16
P-value для TotalCharges: 2.4610180649449646e-20
P-value для tenure: 8.111892260789075e-18

```

In []: