БИБЛИОТЕКИ  
  
import numpy as np

import pandas as pd

import io

import statsmodels.formula.api as smf

import scipy.stats as stats

import scipy

import seaborn as sns

import re

import random

**## Построение OLS модели**

df = pd.read\_csv('https://raw.githubusercontent.com/artamonoff/Econometrica/master/python-notebooks/data-csv/BankPersonalLoanModelling.csv')

df

(ссылку копируем сделав raw)

***Переименовать:***

df.rename(columns = {'ZIP Code':'ZIP\_Code', 'Personal Loan':'Personal\_Loan', 'Securities Account':'Securities\_Account', 'CD Account':'CD\_Account'}, inplace = True)

df

**#### Спецификация**

$$

Income = \beta\_0 + \beta\_1\*Family+\beta\_2\*age+\beta\_3\*CreditCard

$$

bank\_eq = smf.ols(formula='np.log(Income)~Family+Age+CreditCard', data=df).fit()

bank\_eq.params.round(3)

bank\_eq = smf.ols(formula='Income~Family+Age+CreditCard', data=df).fit()

bank\_eq.params.round(3). (другой вариант)

# Интерпретация коэффициентов и построение графиков

### Интерпретация

При увелничении значения переменной family на 1, значение переменной Income ументшается на 7,6% при прочих равных условиях.

При увелничении значения переменной age на 1, значение переменной Income ументшается на 0,4% при прочих равных условиях.

Если пользователь имеет кредитную карту, значение переменной Income ументшается на 0,5% при прочих равных условиях.

sns.regplot(data=df, y='Income', x='Age', ci=None, order = 2, line\_kws={"color": "r"})

sns.regplot(data=df, y='Income', x='Age', ci=None, order = 2, line\_kws={"color": "r"}, marker='s', color='yellow')

df['Age^2'] = df['Age']\*df['Age']

df

**#### Спецификация**

$$

Income = \beta\_0 + \beta\_1\*Family+\beta\_2\*Age^2+\beta\_3\*CreditCard

$$

bank\_eq1 = smf.ols(formula='np.log(Income)~Family+Age^2+CreditCard', data=df).fit()

bank\_eq1.params.round(3)

sns.regplot(data=df, y='Income', x='Age^2', ci=None, order = 2, line\_kws={"color": "g"})

При использовании переменной второго порядка значения наблюдений смещаются ближе к началу координат, увеличивается точность предсказания и изгиб кривой, т.к. изначально зависимость была линейная, а стала квадратической.

**# Вычисление TSS, RSS, ESS, стандартной ошибки, R^2 и R^2adj**

# TSS изначальной регрессии

TSS = bank\_eq.centered\_tss.round(3)

TSS

# ESS изначальной регрессии

ESS = bank\_eq.ess.round(3)

ESS

# RSS изначальной регрессии

RSS = bank\_eq.ssr.round(3)

RSS

#R^2 изначальной регрессии

R2 = RSS/TSS

R2

# Стандартная ошибка изначальной регрессии

s2 = RSS/(5000 - 4)

s2

#R^2adj изначальной регрессии

R2adj = 1 - (1 - R2)\*(5000-1)/(5000 - 4)

R2adj

model = smf.ols(formula='np.log(Income)~Family+Age+CreditCard', data=df).fit()

model.summary(alpha=0.05).tables[1]

df = pd.read\_html(model.summary().tables[1].as\_html(), header=0, index\_col=0)[0]

df

df['tvalues'] = df['coef']/df['std err']

df

alpha = 0.05

print('Уровень значимости alpha: ', alpha)

DF = int(model.df\_resid)

t\_crit = scipy.stats.t.ppf(1-alpha/2, DF)

print('Критическое значение t\_crit: ', t\_crit.round(3))

df['znachimosty'] = df.apply(lambda x : 'znachim' if np.abs(x['tvalues']) > t\_crit else 'ne znachim', axis = 1 )

df

**### Значимость:**

Значимы переменные Family, Age, а переменная CreditCards не значима

**### Фильтрация по бинарной переменной**

df = pd.read\_csv('https://raw.githubusercontent.com/artamonoff/Econometrica/master/python-notebooks/data-csv/BankPersonalLoanModelling.csv')

df.rename(columns = {'ZIP Code':'ZIP\_Code', 'Personal Loan':'Personal\_Loan', 'Securities Account':'Securities\_Account', 'CD Account':'CD\_Account'}, inplace = True)

df

bank\_eq2 = smf.ols(formula='np.log(Income)~Family+Age+CreditCard', data=df[df['CreditCard']==1]).fit()

bank\_eq2.params.round(3)

# TSS

TSS2 = bank\_eq2.centered\_tss.round(3)

TSS2

# ESS

ESS2 = bank\_eq2.ess.round(3)

ESS2

# RSS

RSS2 = bank\_eq2.ssr.round(3)

RSS2

#R^2

R2\_2 = RSS2/TSS2

R2\_2

### вывод

Предсказанная точность регрессии для заданного значения бинарной переменной CreditCard = '1' yb;t, чем предсказанная точность регрессии для не заданного значения бинарной переменной.

**# F-test**

$$

H0: \beta\_{Age}=\beta\_{Experience}=\beta\_{Income}=\beta\_{Family}=0

$$

$$

H1: \beta\_{Age}^2+\beta\_{Experience}^2+\beta\_{Income}^2+\beta\_{Family}^2>0

$$

bank\_eq3 = smf.ols(formula='np.log(Income)~Family+Age+CreditCard+Experience+ZIP\_Code+CCAvg+Education+Mortgage+Personal\_Loan+Securities\_Account+CD\_Account+Online', data=df[df['CreditCard']==1]).fit()

bank\_eq3.params.round(3)

# RSS

RSS3 = bank\_eq3.ssr.round(3)

RSS3

# TSS

TSS3 = bank\_eq3.centered\_tss.round(3)

TSS3

#R^2

R2\_3 = RSS3/TSS3

R2\_3

unrest, rest = np.float\_(re.findall(r'([-+]?\d+.\d+)','R2 0.576 0.98'))

print (unrest, rest)

F\_obs = (unrest - rest)/(1-unrest)\*(5000 - 13)/4

F\_obs

alpha = 0.05

F\_cr = stats.f.ppf(1-alpha, 4, 5000-13)

F\_cr

**### Вывод:**

$$

F\_{obs}<F{cr}

$$

Данные согласуются с гипотезорй Н0

**ДОВЕРИТЕЛЬНЫЕ ИНТЕРВАЛЫ**

model = smf.ols(formula='np.log(Income)~Family+Age+CreditCard', data=df).fit()

model.summary()

df\_table = pd.read\_html(model.summary().tables[1].as\_html(), header=0, index\_col=0)[0]

df\_table

alpha = 0.05 # Уровень значимости 5%

nobs = model.nobs

regr\_cnt = model.df\_model + 1

print(f'Количество наблюдений = {nobs:.0f}')

print(f'Количество регрессоров + 1 = {regr\_cnt:.0f}')

# 5%-критическое значение t-распределения

t\_cr = stats.t.ppf(1-alpha/2, nobs - regr\_cnt)

print(f'Уровень значимости alpha = {alpha:.2f}')

print(f'Критическое значение t\_cr = {t\_cr:.3f}')

df\_table['lower'] = (df\_table['coef'].astype(float) - df\_table['std err'].astype(float)\*t\_cr).round(2)

df\_table['upper'] = (df\_table['coef'].astype(float) + df\_table['std err'].astype(float)\*t\_cr).round(2)

df\_table

**ПРОГНОЗИРОВАНИЕ**

## выбираем 4 случайных строки

df\_ppl = df.sample(n=4)

df\_ppl

res = model.predict(exog=df\_ppl, transform=True)

df\_res = pd.DataFrame(res.round(2), columns=['Прогноз'])

df\_res.reset\_index(drop=True , inplace=True)

df\_res.index += 1

df\_res

# **Мультиколлинеарность**

Для вычисления показателей VIF для коэффициентов Age, Family, Education будем рассматривать следующие регрессии:

Age на Family, ZIP\_Code, Education, Mortgage

Family на Age, ZIP\_Code, Education, Mortgage

Education на Family, ZIP\_Code, Mortgage, Age

model\_\_0 = smf.ols(data = df , formula = 'Income~Age+Family+Education+Mortgage+ZIP\_Code').fit()

model\_\_1 = smf.ols(data = df , formula = 'Age~Family+Education+Mortgage+ZIP\_Code').fit()

model\_\_2 = smf.ols(data = df , formula = 'Family~Age+Education+Mortgage+ZIP\_Code').fit()

model\_\_3 = smf.ols(data = df , formula = 'Education~Age+Family+Mortgage+ZIP\_Code').fit()

model\_\_1.params.round(3)

model\_\_1.rsquared

VIF\_1 = 1/(1 - model\_\_1.rsquared.round(3))

VIF\_1.round(2)

VIF\_2 = 1/(1 - model\_\_2.rsquared.round(3))

VIF\_2.round(2)

VIF\_3 = 1/(1 - model\_\_3.rsquared.round(3))

VIF\_3.round(2)

**# Дамми-переменные**

$$

Н0: \beta\_{Income}=\beta\_{Family}=\beta\_{Age}=\beta{CreditCard}=0

$$

$$

H1: \beta\_{Income}^2+\beta\_{Family}^2+\beta\_{Age}^2+\beta{CreditCard}^2>0

$$

data\_string = '''R2 0.978 0.576 '''

rest, unrest =np.float\_(re.findall(r'([-+]?\d+.\d+)', data\_string))

F\_obs = (unrest - rest)/(1-unrest)\*(935-10)/5

F\_obs.round(2)

alpha = 0.05

F\_cr = stats.f.ppf(1-alpha,4,706-8)

F\_cr

$$

F\_{obs}<F\_{cr}

$$

Нет оснований отвергать гипотезу Н0, структурные сдвиги не значимы

**# Спецификация**

$$

H0:\beta\_{income^2}=\beta\_{income^3}=\beta\_{income^4}=0

$$

model = smf.ols(formula='np.log(Income)~Family+Age+CreditCard', data=df).fit()

model.params

df['inc2'] =model.predict()\*\*2

df['inc3'] =model.predict()\*\*3

df['inc4'] =model.predict()\*\*4

model\_\_\_1 = smf.ols(formula='np.log(Income)~Family+Age+CreditCard+inc2+inc3+inc4', data=df).fit()

model.params

R2=model\_\_\_1.rsquared

R2\_res = model.rsquared

((R2 - R2\_res)/(1-R2))\*((len(df)-7)/3)

alpha = 0.05

stats.f.ppf(1-alpha,3,len(df)-7).round(2)

$$

F\_{cr}<F\_{obs}

$$

Данные согласуются с гипотезой Н0

**# Гетероскедастичность**

# квадраты остатков добавим в датафрейм

df['res2']=model.resid\*\*2

df['res']=model.resid

sub\_model= smf.ols(data=df, formula='res2~Family+Age+CreditCard').fit()

$$

H0:\gamma\_{1}=\gamma\_{2}=\gamma\_{3}=0

$$

$$

H1:\gamma\_{1}^2+\gamma\_{2}^2+\gamma\_{3}^2>0

$$

St = np.round(len(df)\*sub\_model.rsquared,2)

St

Hi2 = stats.chi2.ppf(1-0.05,6)

Hi2

$$

nR\_{0}^2 > X\_{p}^2

$$

Отвергаем Н0. Тест указывает на гетероскедастичность

**# Серийная корреляция**

resSq = sum(model.resid\*\*2)

sum\_resid = 0

for i in range(len(model.resid)-1):

sum\_resid+=(model.resid[i+1]-model.resid[i])\*\*2

sum\_resid/resSq

$$

p ≠ 0

$$

Гипотеза Н0 отвергается, есть серийная когрреляция

**# Тест Чоу, графики структурных сдвигов**

sns.lmplot(x='Age', y='Income', hue='Education', data=df)

## Анализ графика:

-С ростом уровня образования увеличивается доход

-Чем старше человек, тем больше его доход

-У людей, имеющих высшее образование, доход в среднем выше, чем у людей, его не имеющих вне зависимости от возраста.

sns.lmplot(x="Age", y="Income",hue='Family',col='Personal\_Loan', data=df)

## Анализ графиков:

У людей, не имеющих вклады в банке (Personal Loan = 0)

Семьи таких людей в среднем болшьшие, большое количество людей, имебющих более трех детей.

Доход таких семей в среднем меньше, чем доход людкей, имеющих банковские вклады вне зависимости от количества детей.

Чем больше детей, тем меньше доход (Это утверждение не действительно в отношении людей, имеющих 1-2 детей)

У людей, имеющих вклады в банке (Personal Loan = 1)

Семьи таких людей в среднем не болшьшие, распределение детией примерно равномерно.

Доход таких семей в среднем мвыше, чем доход людей, не имеющих банковские вклады вне зависимости от количества детей.

Доход не зхависит от количества детей и возраста.

Матрица

# построим матрицу корелляции:

columns = ['Age', 'Experience', 'ZIP\_Code', 'CCAvg']

subset = df[columns]

corr\_matrix = subset.corr()

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm')

переименовать

df.rename(columns = {'race/ethnicity':'race', 'parental level of education':'level', 'test preparation course':'test', 'math score':'math\_score', 'reading score':'reading', 'writing score':'writing'}, inplace = True)

df

# Создаем точечный график с квадратичной регрессионной линией

ax = sns.regplot(data=df, y='price', x='area', ci=None, order=2, line\_kws={"color": "gold"}, scatter\_kws={"color": "red"})

КОРРЕЛЯЦИЯ

# Строим модель линейной регрессии

house = smf.ols(formula='price~area+furnishingstatus+mainroad', data=df).fit()

# Получаем параметры модели

params = house.params

# Создаем корреляционную матрицу

corr\_matrix = df.corr()

# Выводим корреляционную матрицу в консоль

print(corr\_matrix)

???

ex = smf.ols(formula='np.log(math\_score)~reading+writing', data=df).fit()

ex.params.round(3)

model = smf.ols(formula='np.log(math\_score)~reading+writing', data=df).fit()

model.summary(alpha=0.05).tables[1]

ex = smf.ols(data=df,formula='math\_score~reading+writing'). fit()

ex.params.round(3)

df.rename(columns = {'race/ethnicity':'race', 'parental level of education':'level', 'test preparation course':'test', 'math score':'math\_score', 'reading score':'reading', 'writing score':'writing'}, inplace = True)

df

sns.regplot(data=df, y='math\_score', x='reading', ci=None, order = 2, line\_kws={"color": "r"}, marker='s', color='yellow')

TSS = ex.centered\_tss.round(3)

TSS

ESS = ex.ess.round(3)

ESS

RSS = ex.ssr.round(3)

RSS

R2 = RSS/TSS

R2

Вычитать константу

model = smf.ols('total\_score ~ C(gender) + C(parental\_level\_of\_education)+C(race\_ethnicity)+C(test\_preparation\_course)-1',data = df).fit()

model.summary()

ЦВЕТ жирно/курсив

### Таким образом, можно сделать вывод, что <span style="color:brown">пол, уровень образования родителей, этническая принадлежность и прохождение курсов подготовки к тесту</span> существенно влияют на оценки студентов за три предмета (и общий балл в целом)

### Целью данной работы является $\textit{проанализировать влияние различных факторов на экзаменационный результат студентов}$

Основная статистика

# Получим основную статистику по данным датасета

statistics = df.describe()

print(statistics)

Зависимость чего-то от другого

# Создание двух отдельных DataFrame для оценок мужчин и женщин

male\_scores = df[df['gender'] == 'male'][['math score', 'reading score']]

female\_scores = df[df['gender'] == 'female'][['math score', 'reading score']]

# Построение диаграммы рассеяния

plt.scatter(male\_scores['math score'], male\_scores['reading score'], alpha=0.5, label='male')

plt.scatter(female\_scores['math score'], female\_scores['reading score'], alpha=0.5, label='female', color='pink')

plt.title('Math Scores vs Reading Scores')

plt.xlabel('Math Scores')

plt.ylabel('Reading Scores')

plt.legend()

plt.show()