1. Цель работы

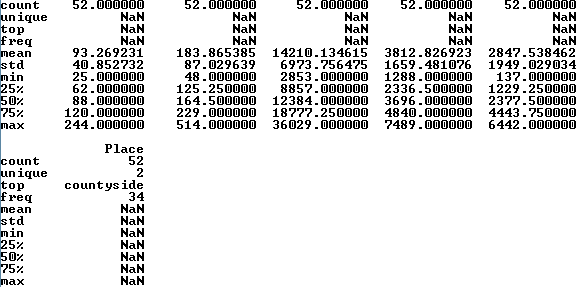
Научиться проводить статистический анализ данных с использованием языка программирования Python. Изучить функции для работы с данными на языке Python, которые являются аналогами используемых ранее функций на языке R.

1. Типы данных

Исходные данные: Hospital.xls

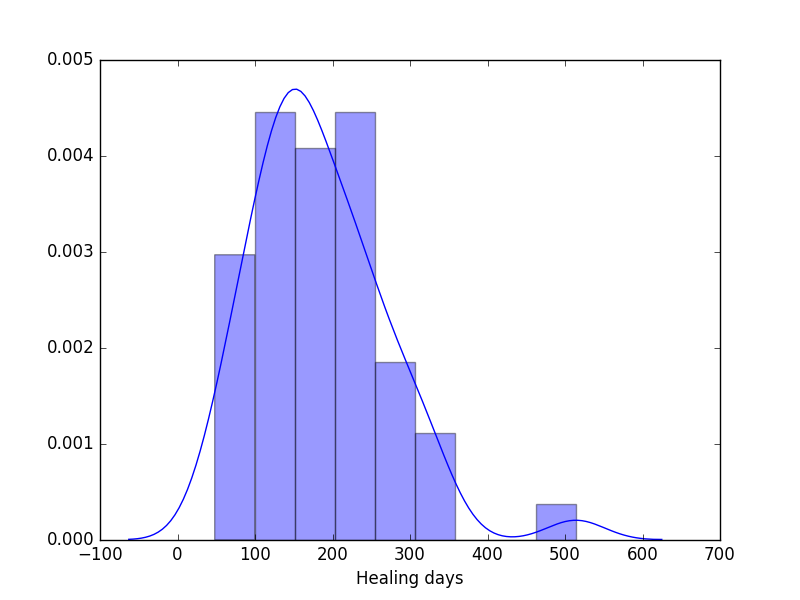
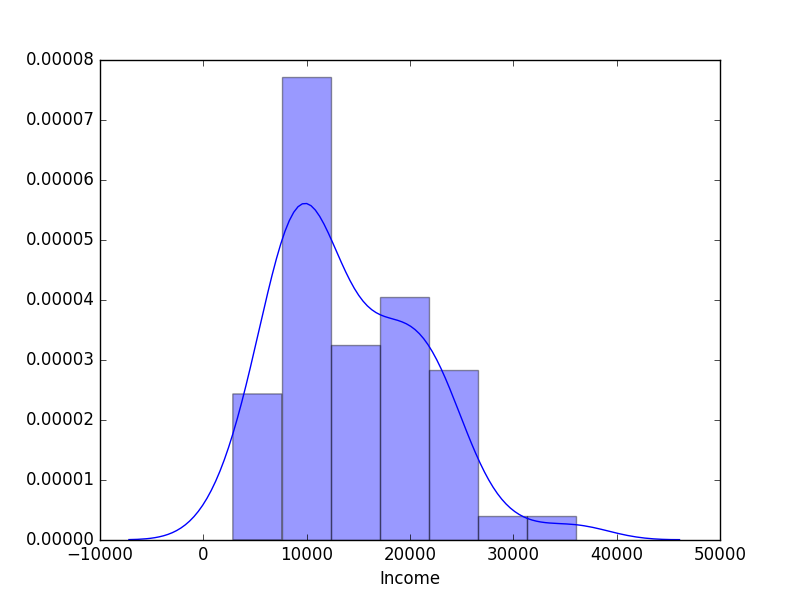
|  |  |
| --- | --- |
| № п/п | Номинальный |
| Койки | Количественный |
| Лечебные дни | Количественный |
| Доход | Количественный |
| Зарплаты | Количественный |
| Расходы | Количественный |
| Расположение | Номинальный |

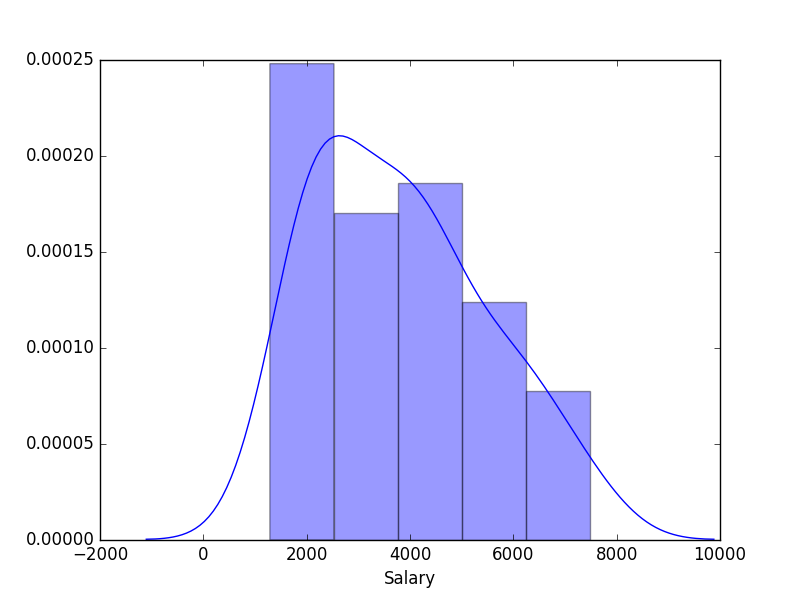
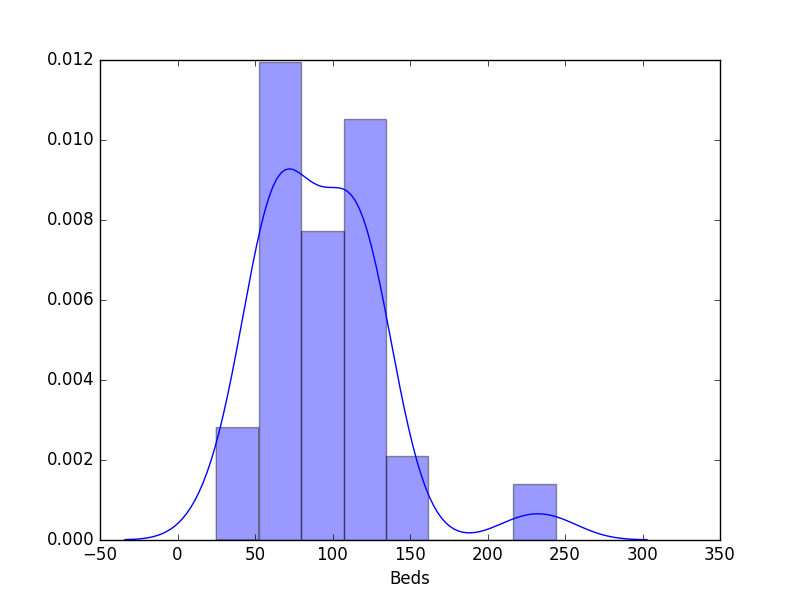
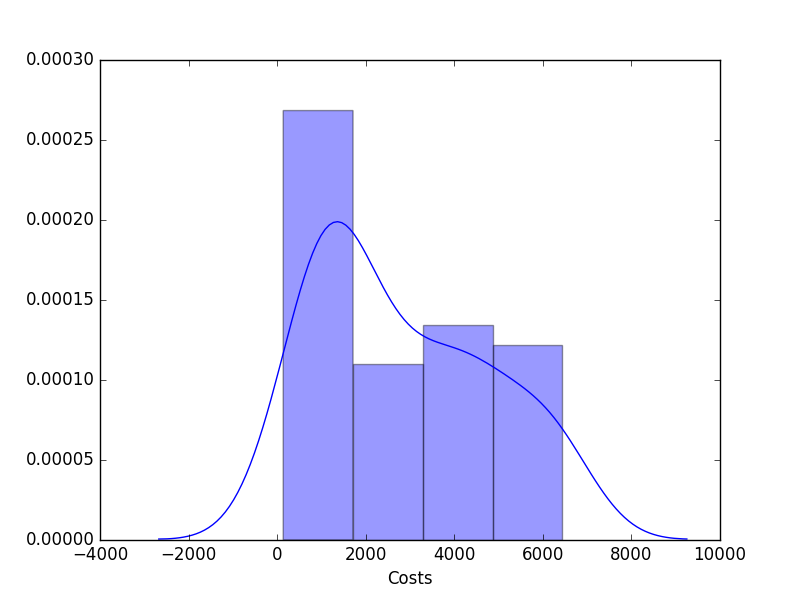
1. Предварительный анализ данных
2. Получить общую информацию по описательным статистикам:



1. По каждому признаку в наборе данных построить гистограммы, для количественных признаков построить «ящики с усами» и графики эмпирических распределений, для пар наиболее перспективных для дальнейшего анализа переменных построить корреляционные поля.

Гистограммы для каждого признака:

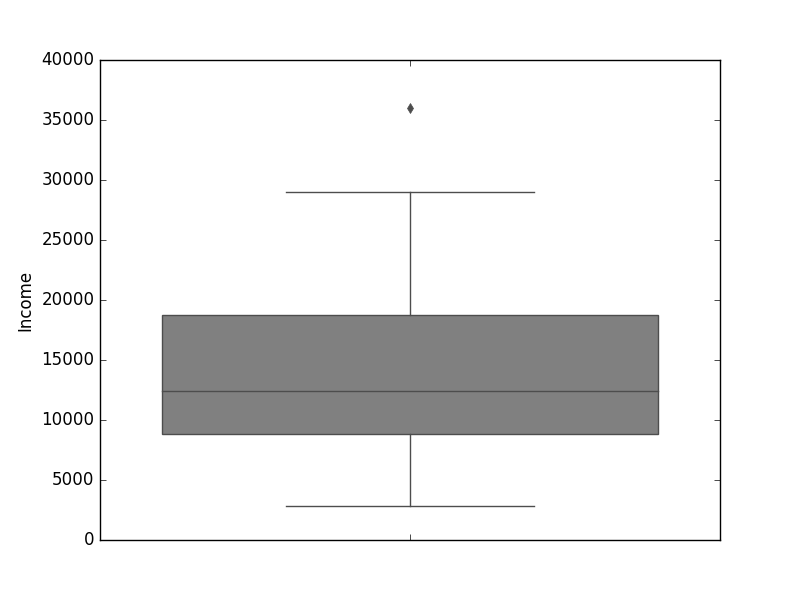
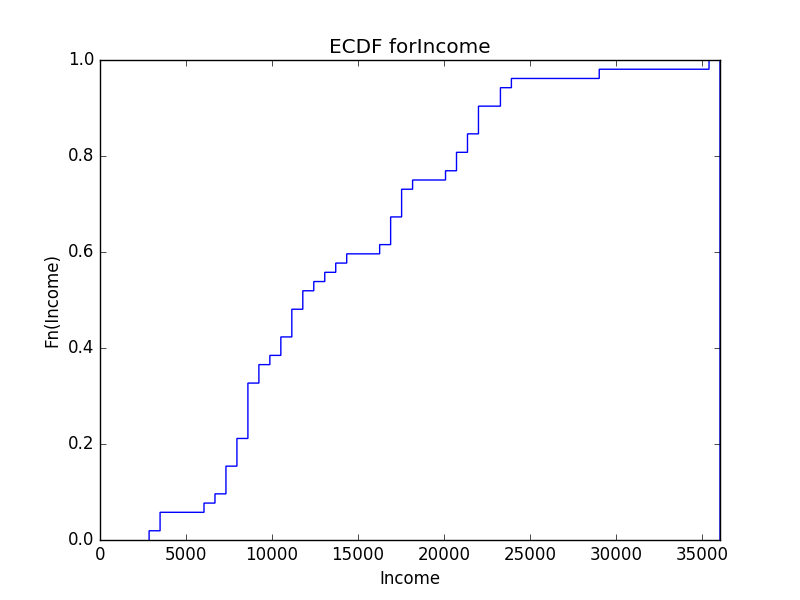


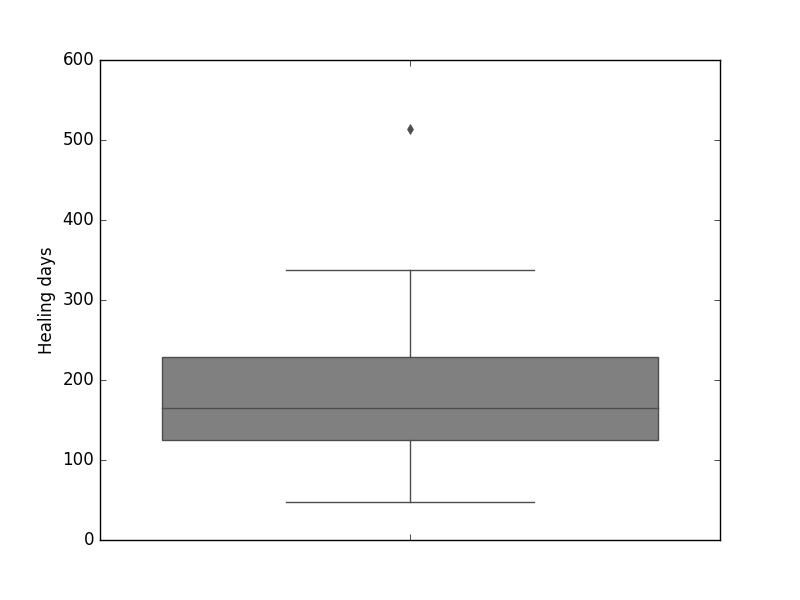
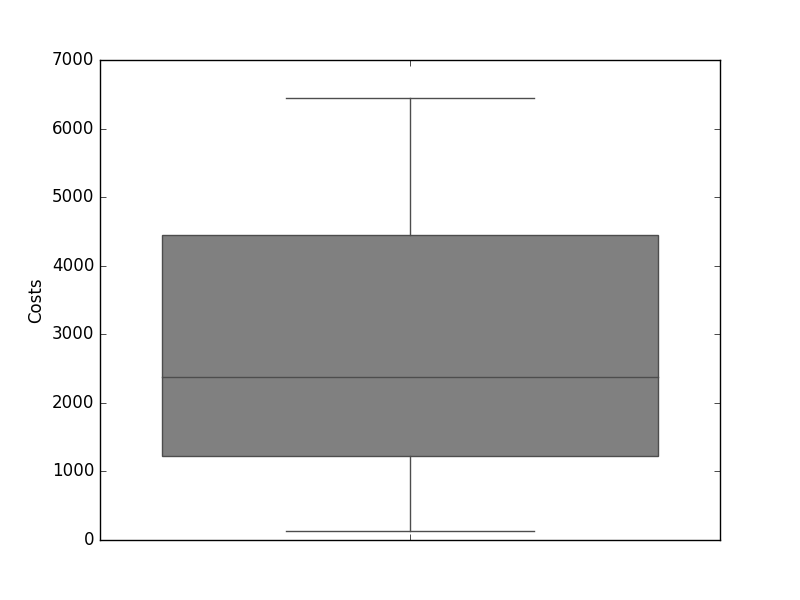
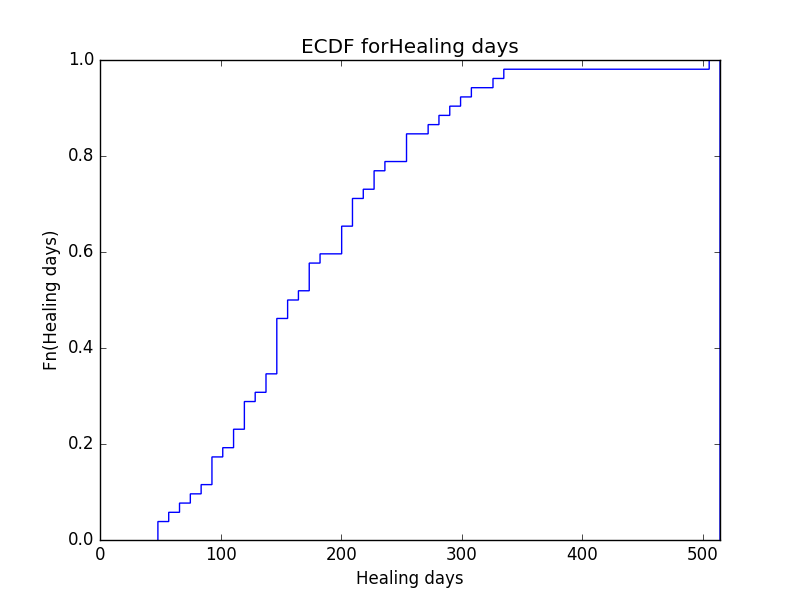
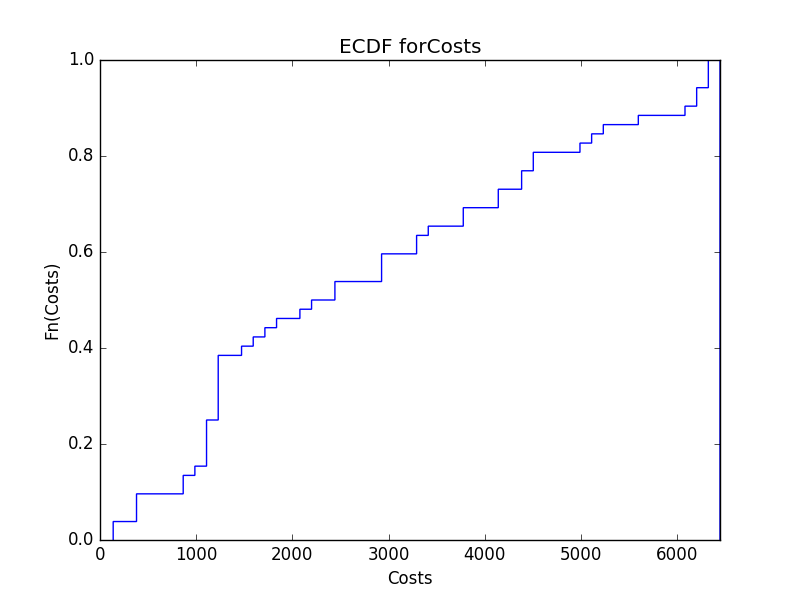


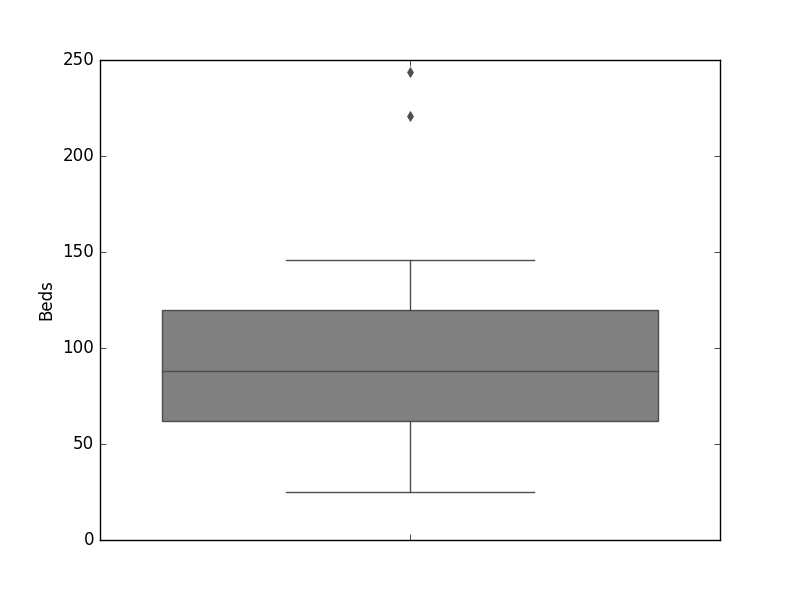
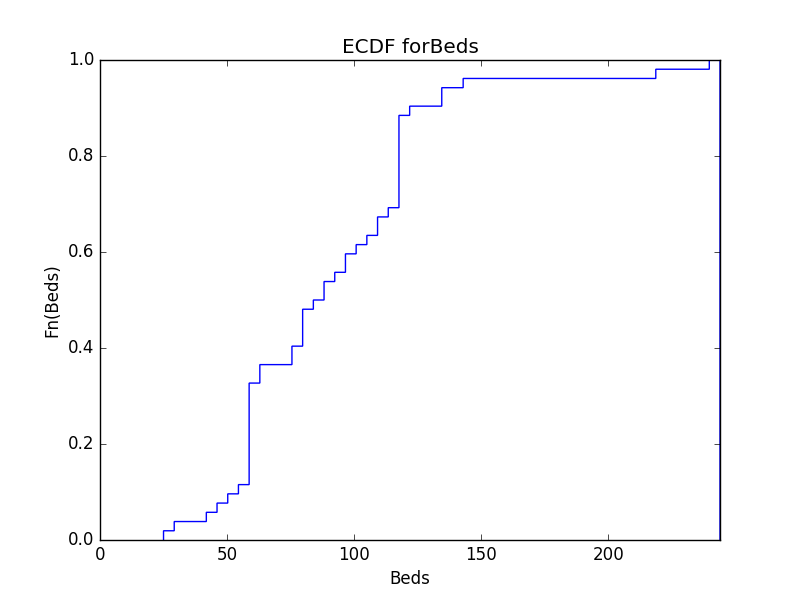
Из полученных графиков можно сделать следующие выводы:

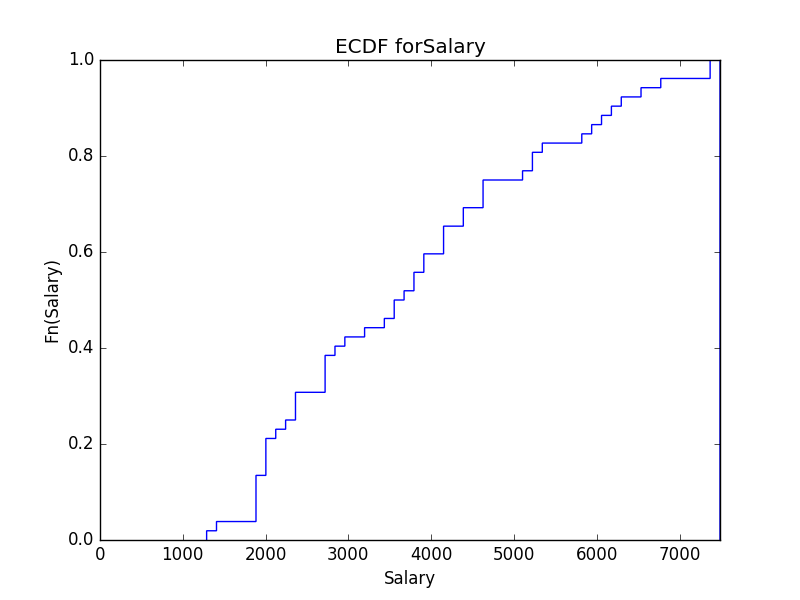
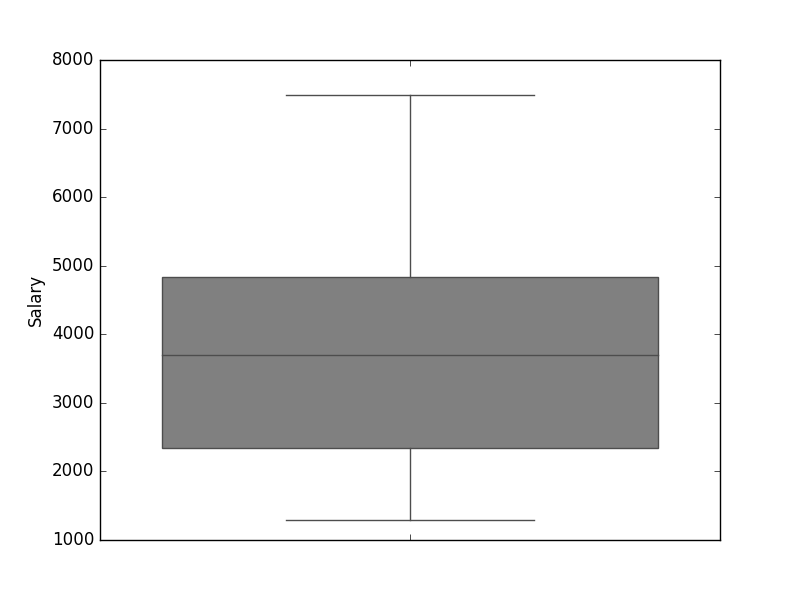
* Переменные лечебные дни, зарплата имеют распределение близкое к нормальному закону;
* Остальные переменные имеют распределение являющееся смесью нормального закона распределения с различными параметрами, либо иное распределение.

Диаграммы размахов для количественных переменных и эмпирические функции распределений для количественных признаков:

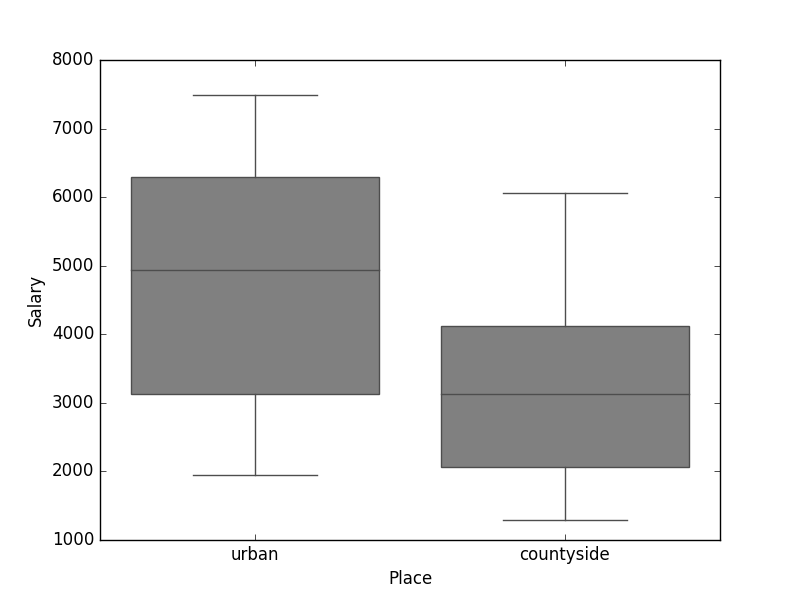
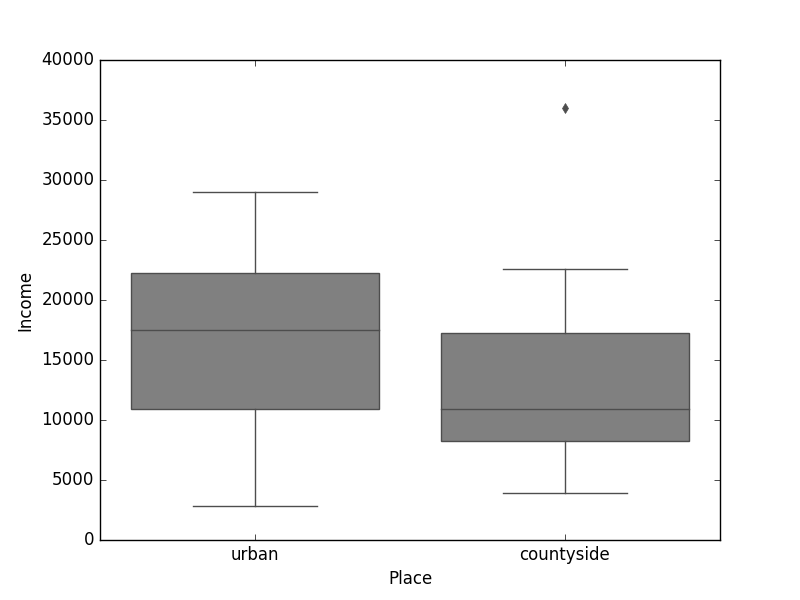
 

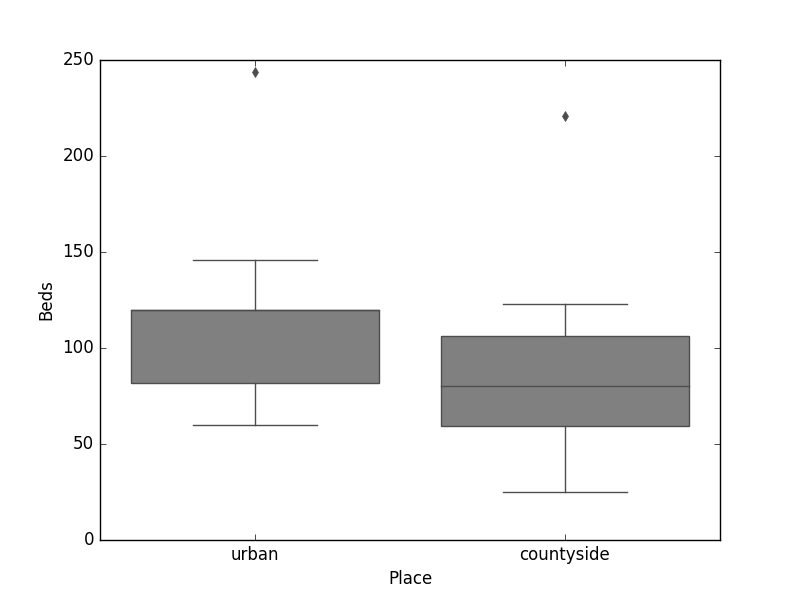
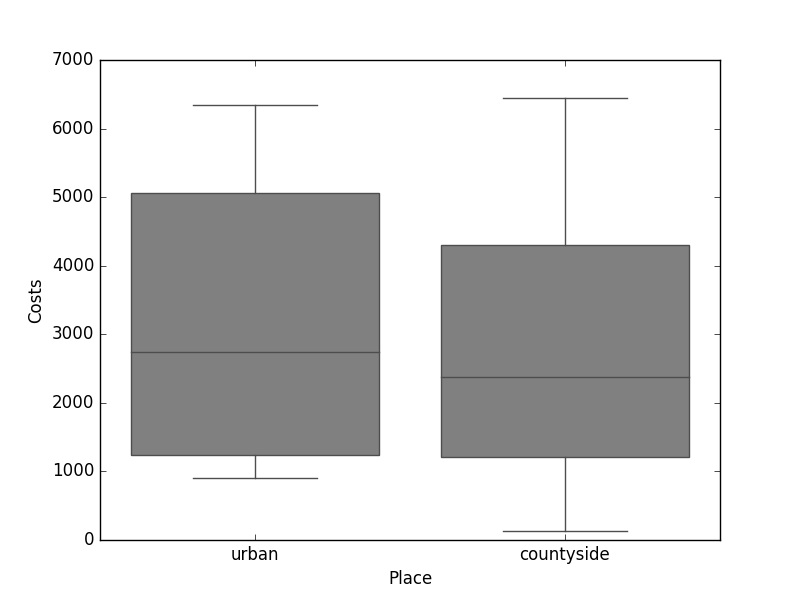
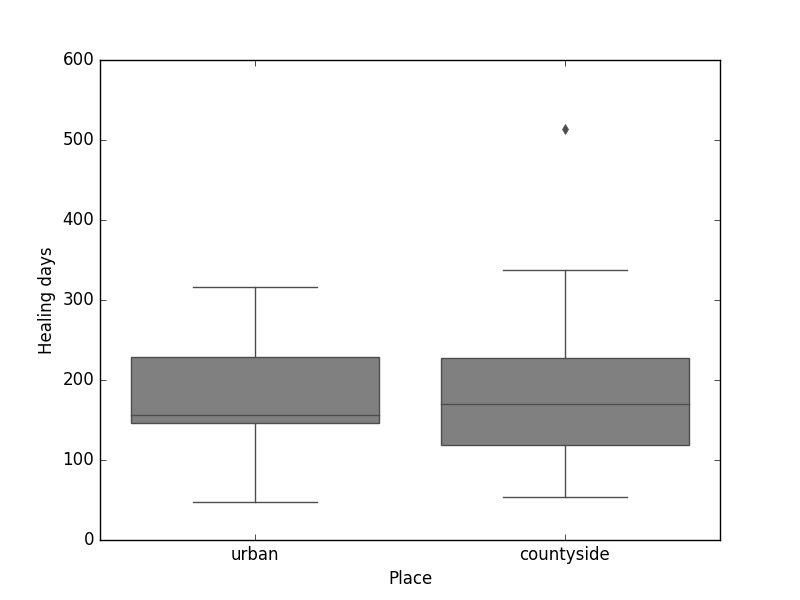
  

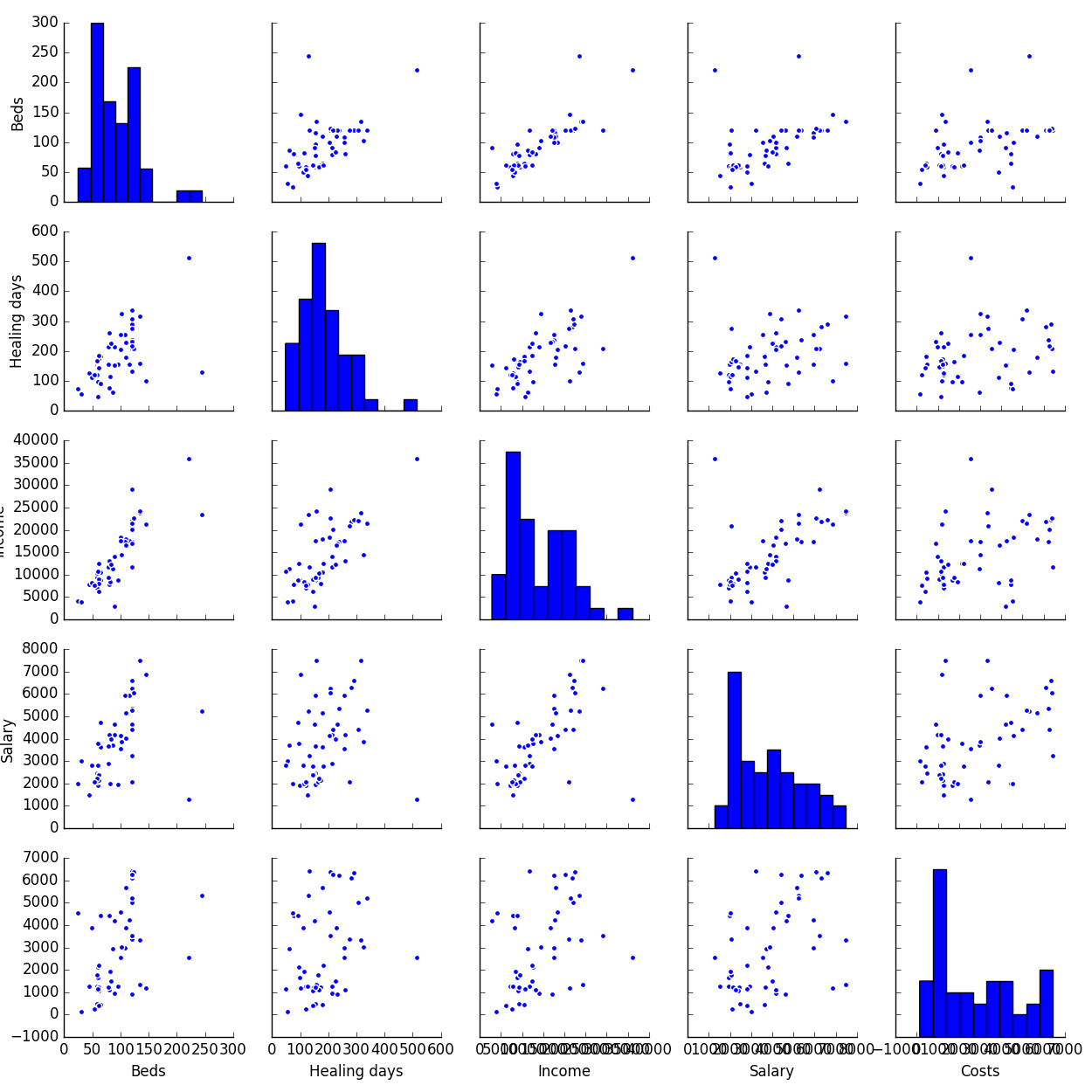


Диаграммы для поиска зависимостей:





По данным графикам можно сделать вывод, что доходы и зарплаты в городе выше, чем загородом, при том, что затраты схожи.



По данным графикам можно проследить линейную зависимость между доходами и зарплатами, койками и доходами. В остальных случаях явная зависимость не наблюдается.

1. Проверка статистических гипотез

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1. Койки | 1. Лечебные дни | 1. Расположение |
| 1. Доход | 1. H0:{q = 0} 2. cor 0.8437752 3. p-value = 4.048e-15 4. Гипотеза о равенстве коэффициента корреляции нулю отвергается. | 1. H0:{q = 0} 2. cor 0.7070754 3. p-value = 4.656e-09 4. Гипотеза о равенстве коэффициента корреляции нулю отвергается. | 1. H0:{нормальное распределение доходов в городе} 2. statistic=0.96513 3. pvalue=0.61720 4. Гипотеза **не отвергается.** |
| 1. H0:{доход и кол-во коек независимы} 3. S = 24.18301 4. p-value = 5.60695e-06 5. chi2: p < 0.05 6. Гипотеза об отсутствии зависимости отвергается 7. Наблюдается зависимость между переменными | 1. H0:{доход и кол-во лечебных дней не зависимы} 3. S = 21.76471 4. p-value = 1.87869e-05 5. chi2: p < 0.05 6. Гипотеза об отсутствии зависимости отвергается 7. Наблюдается неявная линейная зависимость | 1. H0:{нормальное распределение доходов за городом} 2. statistic=16.81343 3. pvalue=0.00022 4. Гипотеза отвергается |
| 1. H0:{однородность доходов в зависимости от расположения} 2. Манн-Уитни: 3. statistic=190.0 4. pvalue=0.01316 5. Гипотеза отвергается |
| 1. Зарплаты | 1. H0:{q = 0} 2. cor 0.5094241 3. p-value = 0.0001148 4. Гипотеза о равенстве коэффициента корреляции нулю отвергается. | 1. H0:{q = 0} 2. cor 0.2541355 3. p-value = 0.06906 4. Гипотеза о равенстве коэффициента корреляции нулю **не отвергается** | 1. H0:{нормальное распределение зарплат в городе} 2. statistic=3.02814 3. pvalue=0.22001 4. Гипотеза **не отвергается** |
| 1. H0:{зарплаты и кол-во коек независимы} 3. S = 34.74053, 4. p-value = 5.25177e-07 5. chi2: p < 0.05 6. Гипотеза об отсутствии зависимости отвергается | 1. H0:{нормальное распределение лечебных дней} 2. statistic=16.72868 3. pvalue=0.00023 4. Гипотеза о нормальном распределении лечебных дней отвергается. 5. H0:{нормальное распределение зарплат } 6. statistic=4.14984 7. pvalue=0.12557 8. Гипотеза о нормальном распределении зарплат **не отвергается.** 9. H0:{зарплаты и лечебные дни независимы} 11. S = 7.34218 12. p-value = 0.11887 13. Гипотеза об отсутствии зависимости **не отвергается** | 1. H0:{нормальное распределение зарплат за городом} 2. statistic=2.15686 3. pvalue=0.34013 4. Гипотеза **не отвергается** |
| 1. H0:{однородность зарплат в зависимости от расположения} 2. Стьюдент: 3. statistic=3.69768 4. pvalue=0.00054 5. Гипотеза отвергается |
| 1. Расходы | 1. H0:{q = 0} 2. cor 0.4602559 3. p-value = 0.0005967 4. Гипотеза о равенстве коэффициента корреляции нулю отвергается. | 1. H0:{q = 0} 2. cor 0.2583959 3. p-value = 0.06438 4. Гипотеза о равенстве коэффициента корреляции нулю **не отвергается**. | 1. H0:{нормальное распределение расходов в городе} 2. statistic=7.06219 3. pvalue=0.02927 4. Гипотеза отвергается |
| 1. H0:{расходы и кол-во коек независимы} 3. S = 11.11128198556111 4. p-value = 0.02534141934810448 5. chi2: p < 0.05 6. Гипотеза об отсутствии зависимости отвергается 7. Зависимость явно не наблюдается | 1. H0:{нормальное распределение лечебных дней} 2. statistic=16.72868 3. pvalue=0.00023 4. Гипотеза о нормальном распределении лечебных дней отвергается. 5. H0:{нормальное распределение расходов} 6. statistic=10.23611 7. pvalue=0.00599 8. Гипотеза о нормальном распределении расходов отвергается. 9. H0:{расходы и лечебные дни независимы} 11. S = 8.99346 12. p-value = 0.17395 13. Гипотеза об отсутствии зависимости **не отвергается** | 1. H0:{нормальное распределение расходов за городом} 2. statistic=3.62990 3. pvalue=0.16285 4. Гипотеза **не отвергается** |
| 1. H0:{однородность расходов в зависимости от расположения} 2. Маннa-Уитни: 3. statistic=273.0 4. pvalue=0.26595 5. Гипотеза **не отвергается** |

|  |  |  |
| --- | --- | --- |
| 1. Койки/Месторасположение | 1. Койки/Зарплата | 1. Лечебные дни (нормальное распределение) |
| 1. H0:{нормальное распределение коек в городе} 2. statistic=13.21356 3. pvalue=0.00135 4. Гипотеза отвергается 5. H0:{нормальное распределение коек за городом} 6. statistic=20.59152 7. pvalue=3.37759e-05 8. Гипотеза отвергается 9. H0:{однородность коек в зависимости от расположения} 10. Манн-Уитни: 11. statistic=169.5 12. pvalue=0.00431 13. Гипотеза отвергается | 1. H0:{нормальное распределение коек в более престижных больницах(зп>3000)} 2. statistic=20.89003 3. pvalue=2.90929e-05 4. Гипотеза отвергается 5. H0:{нормальное распределение коек в менее престижных больницах(зп<3000)} 6. statistic=35.00929 7. pvalue=2.49936e-08 8. Гипотеза отвергается 9. H0:{однородность коек в зависимости от престижности} 10. Манн-Уитни: 11. statistic=98.0 12. pvalue=1.06223e-05 13. Гипотеза отвергается | 1. H0:{нормальное распределение лечебных дней} 2. statistic=16.72868 3. pvalue=0.00023 4. Гипотеза отвергается |

1. Таким образом, зависимость отсутствует между следующими парами: Зарплаты:Лечебные дни, Расходы: Лечебные дни. Гипотеза об однородности отвергается для доход-месторасположение, зарплата-месторасположение, койки-месторасположение, койки-зарплата, что свидетельствует о статистически значимых отличиях в размере дохода и зарплаты

при различном местоположении. Гипотеза об однородности не отвергается для расходы-месторасположение, что свидетельствует о том, что средние расходы у жителей города и загорода статистически не отличаются. Гипотеза о нормальном распределении зарплат не отвергается.

1. Построение моделей регрессии

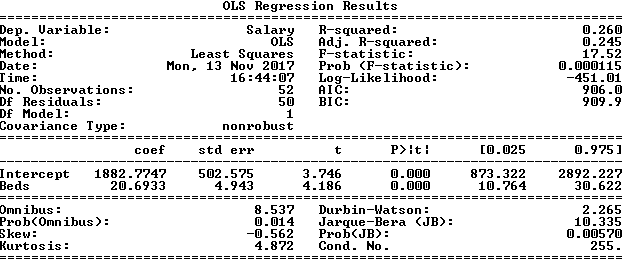
Исключим «Месторасположение» из набора.

Построим модель регрессии, которая поможет предсказать увеличение заработной платы по количеству коек больницы:

smm = sm.OLS.from\_formula("Salary ~ Beds", data=data)

res = smm.fit()

print (res.summary())



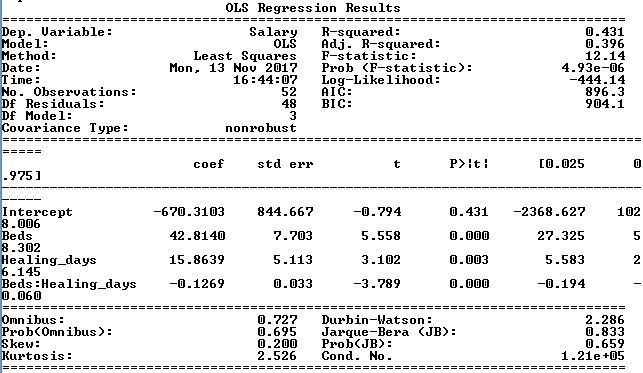
Значение Pr(>|t|) показывает, что коэффициент регрессии (20.693) значимо отличается от нуля (p<0.05).

Допустим нас интересует влияние «Коек» и «Лечебных дней» на заработную плату, для этого построим следующую регрессионную модель:

smm1 = sm.OLS.from\_formula(formula="Salary ~ Beds + Healing\_days + Beds:Healing\_days", data=data)

res1 = smm1.fit()

print (res1.summary())



Из полученных результатов видно, что взаимодействие между «Заработной платой» и «Расположением» значимо. Коэффициент детерминации (R2 = 0.431) близок к 0, это означает, что данная модель плохо описывает имеющиеся данные.

Сравним две модели регрессии:

anova\_lm(res, res1)



Полученное значение Pr(>F) = 0.001765 < 0.05, это означает, что разница между двумя моделями статистически значима. Т.е. переменная «Лечебные дни» вносит статистическую значимость.

Построим «наилучшую» модель. Для этого прибегнем к пошаговой регрессии с использованием точного критерия AIC (информационный критерий Акаике). Наилучшая модель будет обладать наименьшим значением AIC:

def forward\_selected(data, response):

"""Linear model designed by forward selection.

Parameters:

-----------

data : pandas DataFrame with all possible predictors and response

response: string, name of response column in data

Returns:

--------

model: an "optimal" fitted statsmodels linear model

with an intercept

selected by forward selection

evaluated by AIC

"""

remaining = set({ 'Healing\_days:Income',\

'Healing\_days:Beds', 'Healing\_days:Costs', 'Income:Beds', 'Income:Costs', 'Beds:Costs'})

selected = []

current\_score, best\_new\_score = 1000., 1000.0

while remaining and current\_score == best\_new\_score:

scores\_with\_candidates = []

for candidate in remaining:

formula = "{} ~ Healing\_days+Income+Beds+Costs+{}".format(response,

' + '.join(selected + [candidate]))

score = sm.OLS.from\_formula(formula=formula, data=data).fit().aic

scores\_with\_candidates.append((score, candidate))

scores\_with\_candidates.sort()

best\_new\_score, best\_candidate = scores\_with\_candidates[0]

if current\_score > best\_new\_score:

remaining.remove(best\_candidate)

selected.append(best\_candidate)

current\_score = best\_new\_score

formula = "{} ~ Healing\_days+Income+Beds+Costs+{}".format(response,

' + '.join(selected))

model = sm.OLS.from\_formula(formula=formula, data=data).fit()

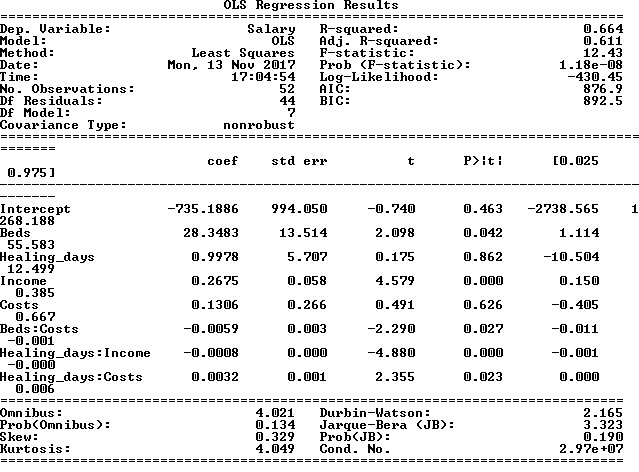
return model

model = forward\_selected(data, 'Salary')

print(model.model.formula)

print(model.summary())

Salary ~ Healing\_days+Income+Beds+Costs+Healing\_days:Income + Healing\_days:Costs + Beds:Costs

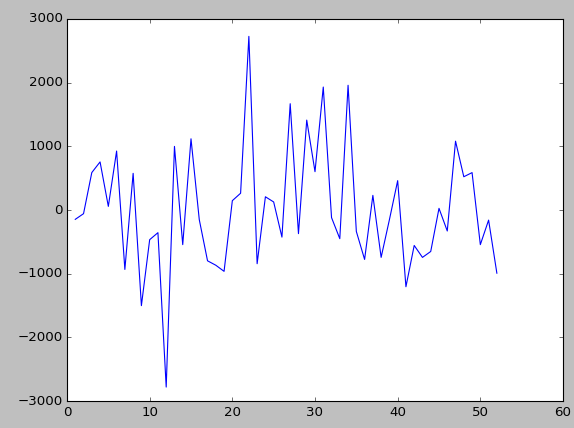


Полученный коэффициент детерминации R2=0.664 достаточно близок к 0.75, а значит, данная модель хорошо описывает данные.

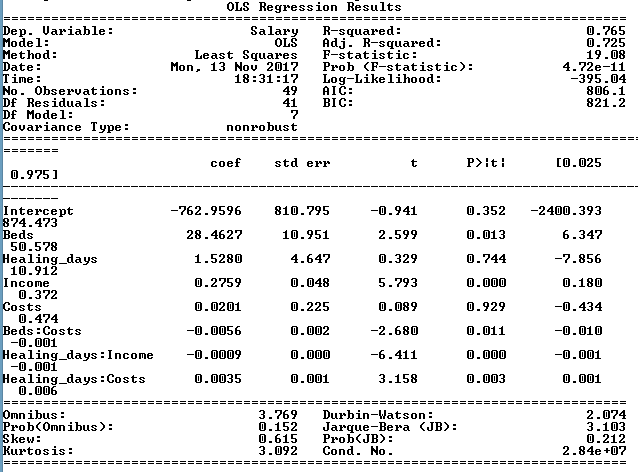
Проверим нашу модель на наличие выбросов:

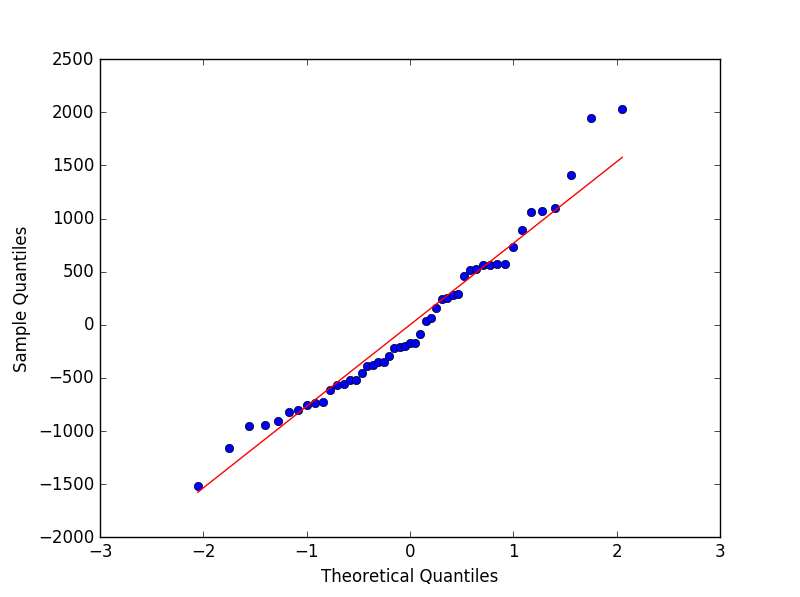
plt.plot(model.resid)

plt.show()



Как видно из полученного графика, в нашей модели присутствуют выбросы, чему соответствует большие колебания графика. Выбросы находятся на 12 и 22 позиции в данных. Исключим эти данные из набора и перестроим модель. 27 тоже был исключен, так как после первого исключения он оказался выбросом. Полученный коэффициент детерминации - означает, что полученная модель хорошо описывает данные.





Данная модель является адекватной, так как практически все значения лежат на прямой, сильных выбросов не выявлено.

Проверим модель на гомоскедастичность:

test = sms.het\_breushpagan(res11.resid, res11.model.exog)

name = ['Lagrange multiplier statistic', 'p-value', 'f-value', 'f p-value']

print(lzip(name, test))

('Lagrange multiplier statistic', 9.856872035419963), ('p-value', 0.19683054481

619247), ('f-value', 1.4749232020566403), ('f p-value', 0.20320640829686315)]

name = ['F statistic', 'p-value']

test = sms.het\_goldfeldquandt(res11.resid, res11.model.exog)

print(lzip(name, test))

[('F statistic', 1.6165789992118276), ('p-value', 0.16771726909456247)]

По результатам теста можно сказать, что данные гомоскедастичны.

Тест Дарбина- Уотсона

DW = 2.07443350404

Т.к статистика близка к 2, то автокорреляция отсутствует.

Кросс-валидация:

import patsy

y, X = patsy.dmatrices(model.model.formula, data, return\_type='dataframe')

model\_n = lm.LinearRegression().fit(X,y)

k\_fold = KFold(n\_splits=10)

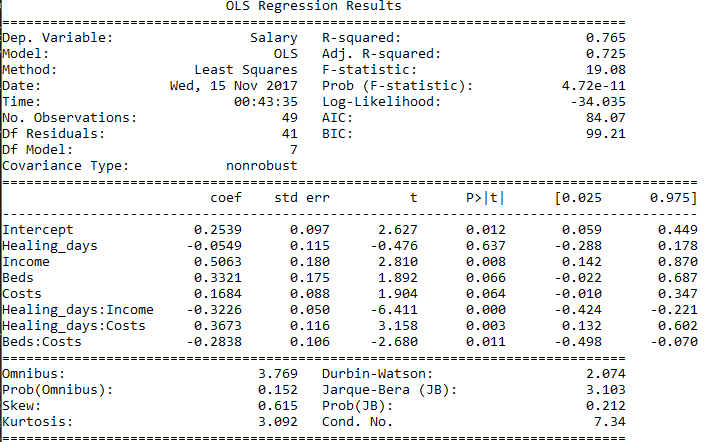
print(cross\_val\_score(model\_n, X, y, cv=k\_fold, scoring='r2'))

array([ 0.52849098, -0.54979584, 0.69399684, 0.73771069, -2.98877512, 0.21141756, 0.09851911, 0.72850001, 0.74379325, 0.40437979])

cross\_val\_score(model\_n, X, y, cv=k\_fold, scoring='r2').mean() 0.060823726151792379

.

Сравним стандартизированные коэффициенты регрессии (Бета-коэффициенты). Они характеризуют ожидаемое изменение зависимой переменной при изменении отдельной независимой переменной на одно стандартное отклонение при постоянных значениях прочих независимых переменных.



зарплата = 0.2539 + 0.3321\*Койки - 0.0549\*Лечебные дни + 0.5063\*доход + 0.1684\*расходы – 0.2838\*Койки:расходы – 0.3226\*Лечебные дни:доход + 0.3673\*Лечебные дни:расходы

1. Вывод

В результате выполнения данной лабораторной работы были сформулированы следующие выводы по поводу используемых программных систем:

Язык R предназначен для проведения статистического анализа данных и работы с графикой. R обычно применяется в тех случаях, когда для анализа данных требуются выделенные вычислительные мощности или отдельные сервера. R отлично подходит для исследовательской работы, удобен практически при любом варианте анализа данных, поскольку в языке R существует масса пакетов и готовые тесты, обеспечивающие нужный инструментарий для быстрого старта. R даже может быть элементом решения в области больших данных.

Язык python – язык общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Для проведения статистического анализа данных используются специальные библиотеки, такие как NumPy, SciPy, Pandas, предоставляющие большой набор инструментов для проведения статистического анализа данных. За графическую составляющую отвечают такие библиотеки как Mapplotlib и Seaborn, обладающие большим набором графического представления данных. Python пригодится в случаях, когда задачи, связанные с анализом данных, вплетаются в работу веб-приложений, или если статистический код требуется инкорпорировать в рабочую базу данных. Python, будучи полнофункциональным языком программирования, отлично подходит для реализации алгоритмов с их последующим практическим использованием. Еще недавно пакеты для анализа данных на Python находились в зачаточном состоянии, что представляло определенную проблему, но в последние годы ситуация значительно улучшилась.

Сравнивая их между собой можно сказать, что R содержит в себе больше пакетов и предоставляет больше возможностей для анализа данных, нежели python, а также огромным плюсом при выборе среды для решения статистических задач является то, что R ориентирован именно для решения таких проблем, однако в последнее время для python появляется все больше новых библиотек, в частности и для анализа данных, что делает его конкурентом относительно языка R.

1. Код программы

import pandas as pd

import numpy as np

import math as mth

import scipy.stats as st

import sklearn.linear\_model as lm

import statsmodels.api as sm

import statsmodels.stats.api as sms

import sklearn.preprocessing as pp

from statsmodels.stats.api import anova\_lm

from statsmodels.compat import lzip

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import cross\_val\_score, KFold, cross\_val\_predict

data = pd.read\_csv('Hospitals.csv', index\_col='No')

#Графики

==============================================================

Гистограммы для каждого признака

for i in data.columns.values[:-1]:

sns.distplot(data[i])

plt.savefig('hist/'+str(i)+'.png', format='png')

plt.clf()

==============================================================

ящики с усами

for i in data.columns.values[:-1]:

sns.boxplot(data[i], orient='v', color='grey')

plt.savefig('box/'+str(i)+'.png', format='png')

plt.clf()

#Эмпирические функции распределений для количественных признаков

for i in data.columns.values[:-1]:

plt.title('ECDF for'+str(i))

plt.ylim(0, 1)

plt.xlim(0, np.max(data[i]))

plt.xlabel(str(i))

plt.ylabel('Fn('+str(i)+')')

x = np.sort(data[i])

num\_bins = len(x)

n, bins, patches = plt.hist(x, num\_bins, normed=1, histtype='step', cumulative=True, label="Emperical cdf")

plt.savefig('ecdf/'+str(i)+'.png', format='png')

plt.clf()

ящики с усами

for i in data.columns.values[0:5]:

sns.boxplot(data['Place'], data[i], orient='v', color='grey')

plt.savefig('box/Plase\_'+str(i)+'.png', format='png')

plt.clf()

sns.pairplot(data)

plt.savefig('pairpl/P.png', format='png')

#Проверка гипотез

ct = pd.crosstab(pd.qcut(data.Income, 2), pd.qcut(data.Beds,3))

res = st.chi2\_contingency(ct)

ct = pd.crosstab(pd.qcut(data.Income, 2), pd.qcut(data.Healing\_days,3))

res = st.chi2\_contingency(ct)

ct = pd.crosstab(pd.qcut(data.Salary, 3), pd.qcut(data.Beds,3))

res = st.chi2\_contingency(ct)

ct2 = pd.crosstab(pd.qcut(data.Salary, 3), pd.qcut(data.Healing\_days,3))

res2 = st.chi2\_contingency(ct2)

ct = pd.crosstab(pd.qcut(data.Costs, 3), pd.qcut(data.Beds,3))

res = st.chi2\_contingency(ct)

ct3 = pd.crosstab(pd.qcut(data.Costs, 4), pd.qcut(data.Healing\_days,3))

res3 = st.chi2\_contingency(ct3)

print(ct)

print(ct2)

print(ct3)

dv1 = data[['Income', 'Place']].loc[lambda x: x.Place == 'urban', :]

dv2 = data[['Income', 'Place']].loc[lambda x: x.Place != 'urban', :]

res1 = st.mstats.normaltest(dv1.Income)

res2 = st.mstats.normaltest(dv2.Income)

res = st.mannwhitneyu(dv1.Income, dv2.Income)

dv1 = data[['Salary', 'Place']].loc[lambda x: x.Place == 'urban', :]

dv2 = data[['Salary', 'Place']].loc[lambda x: x.Place != 'urban', :]

res1 = st.mstats.normaltest(dv1.Salary)

res2 = st.mstats.normaltest(dv2.Salary)

res = st.ttest\_ind(dv1.Salary, dv2.Salary)

dv1 = data[['Costs', 'Place']].loc[lambda x: x.Place == 'urban', :]

dv2 = data[['Costs', 'Place']].loc[lambda x: x.Place != 'urban', :]

res1 = st.mstats.normaltest(dv1.Costs)

res2 = st.mstats.normaltest(dv2.Costs)

res = st.mannwhitneyu(dv1.Costs, dv2.Costs)

dv1 = data[['Beds', 'Salary']].loc[lambda x: x.Salary > 3000, :]

dv2 = data[['Beds', 'Salary']].loc[lambda x: x.Salary < 3000, :]

res = st.mstats.normaltest(dw.Beds)

dv1 = data[['Beds', 'Place']].loc[lambda x: x.Place == 'urban', :]

dv2 = data[['Beds', 'Place']].loc[lambda x: x.Place != 'urban', :]

res = st.mstats.normaltest(dv1.Beds)

res2 = st.mstats.normaltest(dv2.Beds)

res = st.mannwhitneyu(dv2.Beds, dv1.Beds)

print(res)

res = st.mstats.normaltest(data['Healing\_days'])

print(res)

res = st.pearsonr(data.Beds, data.Income)

res1 = st.pearsonr(data.Beds, data.Salary)

res2 = st.pearsonr(data.Beds, data.Costs)

res = st.pearsonr(data.Healing\_days, data.Income)

res1 = st.pearsonr(data.Healing\_days, data.Salary)

res2 = st.pearsonr(data.Healing\_days, data.Costs)

#Выбор множественной регрессии

def forward\_selected(data, response):

"""Linear model designed by forward selection.

Parameters:

-----------

data : pandas DataFrame with all possible predictors and response

response: string, name of response column in data

Returns:

--------

model: an "optimal" fitted statsmodels linear model

with an intercept

selected by forward selection

evaluated by AIC

"""

remaining = set({ 'Healing\_days:Income',\

'Healing\_days:Beds', 'Healing\_days:Costs', 'Income:Beds', 'Income:Costs', 'Beds:Costs'})

selected = []

current\_score, best\_new\_score = 1000., 1000.0

while remaining and current\_score == best\_new\_score:

scores\_with\_candidates = []

for candidate in remaining:

formula = "{} ~ Healing\_days+Income+Beds+Costs+{}".format(response,

' + '.join(selected + [candidate]))

score = sm.OLS.from\_formula(formula=formula, data=data).fit().aic

scores\_with\_candidates.append((score, candidate))

scores\_with\_candidates.sort()

best\_new\_score, best\_candidate = scores\_with\_candidates[0]

if current\_score > best\_new\_score:

remaining.remove(best\_candidate)

selected.append(best\_candidate)

current\_score = best\_new\_score

formula = "{} ~ Healing\_days+Income+Beds+Costs+{}".format(response,

' + '.join(selected))

model = sm.OLS.from\_formula(formula=formula, data=data).fit()

return model

model = forward\_selected(data, 'Salary')

print(model.model.formula)

print(model.summary())

data = data.drop(12)

data = data.drop(22)

data = data.drop(27)

model\_new = sm.OLS.from\_formula(formula=model.model.formula, data=data)

data\_new = pd.DataFrame(pp.scale(data.values[:,:-1]), columns=['Beds','Healing\_days','Income','Salary','Costs'])

model\_new1 = sm.OLS.from\_formula(formula=model.model.formula, data=data\_new).fit()

print(model\_new1.summary())

import patsy

y, X = patsy.dmatrices(model.model.formula, data, return\_type='dataframe')

model\_n = lm.LinearRegression()

#Кросс-Валидация

k\_fold = KFold(n\_splits=10)

scores = cross\_val\_score(model\_n, X, y, cv=k\_fold, scoring='r2')

predicted = cross\_val\_predict(model\_n,X,y,cv=k\_fold)

slope, intercept, r\_value, p\_value, std\_err = st.linregress(y.values[:,0],predicted[:,0])

print(r\_value\*r\_value)

#Гомоскедастичность (Бреуш-Паган, Голдфильд-Квандт)

test = sms.het\_breushpagan(res11.resid, res11.model.exog)

name = ['Lagrange multiplier statistic', 'p-value',

'f-value', 'f p-value']

print(lzip(name, test))

name = ['F statistic', 'p-value']

test = sms.het\_goldfeldquandt(res11.resid, res11.model.exog)

print(lzip(name, test))

#Q-Q

st.probplot(res4.resid,plot=plt)

sm.qqplot(res11.resid, line='s')

plt.show()

#Дарбин-Уотсон

dw = sms.stattools.durbin\_watson(res11.resid)

print(dw)