Библиотечные инструменты языка программирования Python

1. Поиск характеристик и визуализация теоретических раконов распределений

Загрузка основных модулей

```
import numpy as np
import scipy.stats as sts
import scipy.special as sc
import matplotlib.pyplot as plt
%matplotlib inline
```

Вероятностные распределения в модуле scipy.stats

Пусть `R` - обозначение вида закона распределения, `params` - параметры распределения.

Общий вид обращения к распределению:

`sts.R(params)` - закон распределения `R` с параметрами `params`.

Некоторые виды распределений:

`sts.uniform(a1,a2)` - равномерный закон распределения; если $X \sim R(a,b)$, то `a1` совпадает с a, `a2` совпадает с $b-a$.

`sts.norm(a,b)` - нормальный закон распределения с параметрами `a` (математическое ожидание) и `b` (среднее квадратичное отклонение).

`sts.expon(0,m)` - экспоненциальный закон распределения; `m` совпадает с математическим ожиданием, т.е. если $X \sim Exp(\lambda)$, то `m` совпадает с $1/\lambda$.

`sts.chi2(n)` - закон распределения хи-квадрат с `n` степенями свободы.

`sts.t(n)` - закон распределения Стьюдента с `n` степенями свободы.

`sts.f(k1,k2)` - закон распределения Фишера со степенями свободы с `k1` и `k2`.

`sts.binom(n,p)` - биномиальный закон распределения с параметрами `n` (общее число испытаний) и `p` (вероятность успеха в одном испытании).

`sts.poisson (lm)` - закон распределения Пуассона с параметром `lm`, параметр совпадает с традиционным параметром распределения λ (равен математическому ожиданию).

Методы поиска функциональных характеристик вероятностных распределений в модуле scipy.stats

Пусть `R` - обозначение типа закона распределения, `params` - параметры распределения. Тогда:

`R(params).cdf(x)` - значение функции закона распределения `R` с параметрами `params` в точке `x`.

`R(params).pdf(x)` - для непрерывной случайной величины значение плотности распределения в точке `x`.

`R(params).pmf(x)` - для дискретной случайной величины вероятность принять значение `x`.

Методы поиска числовых характеристик случайной величины в модуле scipy.stats

Пусть `R` - обозначение типа закона распределения, `params` - параметры распределения. Тогда:

`R(params).ppf(q)` - квантиль порядка `q`.

`R(params).mean()` - математическое ожидание.

`R(params).var(x)` - дисперсия.

`R(params).std(x)` – среднее квадратичное (стандартное) отклонение.

`R(params).median()` - медиана.

	`R(params).moment(k)` - начальный момент порядка `k`. `R(params).stats('mvsk')` - математическое ожидание, дисперсия, коэффициент асимметрии и коэффициент эксцесса.
Некоторые математические функции в модуле scipy.special	
	`sc.factorial(n)` - $n!$ `sc.comb(n,k)`, `sc.binom(n,k)` - C_n^k, число сочетаний из n по k. `sc.perm(n,k)` - A_n^k, число размещений из n по k. `sc.gamma(x)` - $\Gamma(x)=\int_0^{+\infty} t^{x-1}e^t dt$, гамма-функция.

2. Компьютерное моделирование выборок дискретных случайных величин, первичная обработка выборки

Загрузка основных модулей	

	`import numpy as np`
	`import scipy.stats as sts`
	`import scipy.special as sc`
	`import matplotlib.pyplot as plt`
	`import random`
	`%matplotlib inline`

Генераторы выборок дискретных распределений в библиотеке numpy	
	Функция `np.random.choice(a, size, replace=True, p)` Возвращает выборку заданного объема `size` (если `size` - число) из массива `a`. По умолчанию повторную, если `replace=False`, то бесповторную. Выбор элемента из массива `a` осуществляется с соответствующей вероятностью массива `p`. Если `size` - кортеж, то генерируется массив заданной формы. Другие варианты: если `a` - целое число, то генерируется случайное число из массива `arange(a)`; если параметр `p` не задан, то элементы из `a` равновозможны.
	Функция `np.random.binomial(n, p, size)` Возвращает выборку заданного объема `size` (если `size` - число) биномиального распределения с параметрами `n` и вероятностью `p`. Если `size` - кортеж, то генерируется массив заданной формы.
	Функция `np.random.poisson (lm, size)` Формирует выборку заданного объема `size` (если `size` - число) распределения Пуассона с параметром `lm`. Если `size` - кортеж, то генерируется массив заданной формы.
	В модуле numpy.random также имеются генераторы выборок следующих распределений: геометрического, гипергеометрического, отрицательного биномиального, распределения logser и др.
Средства формирования вариационного и статистического рядов в библиотеке numpy	
	Функция `np.bincount(x)` Вычисляет количество появлений в массиве каждого целого числа от `0` до `max(x)`.
	Функция `np.sort(list(set(x)))` Формирует набор уникальных элементов (вариант) выборки.
Случайное перемешивание	
	Функция `np.random.shuffle(x)` случайным образом перемешивает массив `x`.

3. Компьютерное моделирование выборок непрерывных случайных величин, первичная обработка выборки

Загрузка основных модулей

```
import numpy as np
import scipy.stats as sts
from statsmodels.distributions.empirical_distribution import ECDF
import matplotlib.pyplot as plt
%matplotlib inline
```

Генераторы выборок некоторых непрерывных распределений в библиотеке numpy

Функция `np.random.normal(loc, scale, size)`
Возвращает выборку заданного объема `size` (если `size` - число) нормального распределения $N(m,\sigma)$.
Параметры: `loc=`m ; `scale=`σ. Если `size` - кортеж, то генерируется массив заданной формы.

Функция `np.random.uniform(low=0.0, high=1.0, size=None)`
Возвращает выборку заданного объема `size` (если `size` - число) равномерного распределения $R(a,b)$.
Параметры: `low=`a ; `high=`b. Если `size` - кортеж, то генерируется массив заданной формы.

Функция `np.random.exponential (scale=1.0, size=None)`
Возвращает выборку заданного объема `size` (если `size` - число) экспоненциального распределения $Ex(\lambda)$.
Параметр: `scale =` $1/\lambda$ (равен математическому ожиданию). Если `size` - кортеж, то генерируется массив заданной формы.

Функция `np.random.chisquare (df, size=None)`
Возвращает выборку заданного объема `size` (если `size` - число) распределения хи-квадрат с `df` степенями свободы. Если `size` - кортеж, то генерируется массив заданной формы.

Функция `np.random.f(dfnum, dfden, size)`
Возвращает выборку заданного объема `size` (если `size` - число) распределения Фишера со степенями свободы `dfnum, dfden`. Если `size` - кортеж, то генерируется массив заданной формы.

В модуле numpy.random также имеются генераторы выборок следующих распределений: бета, гамма, Гумбеля, Лапласа, логистического, логнормального, степенного, Рэлея, треугольного, Ломакса (Парето II вида), фон Мизеса, Уайльда, Вейбулла и др.

Построение эмпирической функции распределения в модуле `statsmodels.distributions.empirical_distribution`

Функция `statsmodels.distributions.empirical_distribution.ECDF(x,side=right)`
Возвращает эмпирическую функцию распределения.
Параметры: `x` - массив (выборка); `side` - задает форму интервалов, по которым строятся ступени эмпирической функции: `right` (по умолчанию) - интервалы вида $[...)$, открытые справа, `left` - интервалы вида $(...]$, открытые слева.

Средства визуализации: построение гистограммы и эмпирической функции распределения в модуле matplotlib.pyplot

	Функция `plt.hist(x, bins=None, density=None, weights=None, cumulative=False, histtype='bar', align='mid', orientation='vertical', log=False, color=None)` Строит гистограмму и возвращает два массива: высот столбцов гистограммы и центров интервалов группировки. Параметры: `x` - массив (выборка) ; `bins` - число интервалов группировки или последовательность, задающая границы интервалов (все интервалы, кроме последнего, полуоткрытые вида [...)), или строка из списка, который приводится после перечня параметров; `density` - если `True`, то строится гистограмма относительных частот (суммарная площадь прямоугольников равна 1); `weights` - массив весов той же формы, что и `x`; `cumulative` - если `True`, то в сочетании с признаком `density=True` строит эмпирическую функцию распределения; `histtype` - кроме типа `'bar'` можно указать `'barstacked'` и `'step'`; `align` - задает расположение центров прямоугольников; `orientation` - установив значение `'horizontal'`, можно повернуть график на 90°; `log` - если `True`, для осей используется логарифмическая шкала; `color` - признак, устанавливающий цвет. По умолчанию число интервалов группировки равно 10. Список правил для выбора числа интервалов: `bins='auto'` - максимальное из значений, получаемых по правилу Стерджесса и Фридмана - Диакониса; `bins='fd'` - правило Фридмана - Диакониса; `bins=' sturges'` - правило Стерджесса; `bins= 'doane'` - правило Доэна; `bins=' scott'` - правило Скотта; `bins='stone'` - обобщение правила Скотта; `bins=' rice'` - правило Райса; `bins=' sqrt'` - правило квадратного корня.

4. Точечное оценивание параметров распределения по выборке

Загрузка основных модулей	

	`import numpy as np`
	`import scipy.stats as sts`
	`import matplotlib.pyplot as plt`
	`import seaborn`
	`%matplotlib inline`

Точечные оценки параметров распределения в пакете numpy	
	Функция `np.mean(a, axis)` Возвращает выборочное среднее. Параметры: `a` - массив; в случае многомерного массива `a` можно указать ось (`axis`), вдоль которой вычисляется среднее. Функция `np.nan mean(a, axis)` при вычислении игнорирует пропущенные данные `nan` (важно при обработке реальных данных). Для вычисления выборочных начальных моментов порядка k можно использовать функцию `mean` применительно к k-й степени массива `a`.
	Функция `np.var(a, axis, ddof)` Возвращает оценку дисперсии по выборке `a`.

	Параметры: a - массив; в случае многомерного массива a можно указать ось (axis), вдоль которой вычисляется дисперсия; ddof по умолчанию равен 0 (вычисляется выборочная дисперсия), если задать ddof=1, то функция возвращает исправленную выборочную дисперсию. Функция np.nanvar(a, axis, ddof) при вычислении игнорирует пропущенные данные nan.
	Функция np.std(a, axis, ddof) Возвращает корень из выборочной (или исправленной выборочной) дисперсии. Параметры: a - массив; в случае многомерного массива a можно указать ось, вдоль которой вычисляется дисперсия; ddof по умолчанию равен 0 (вычисляется выборочная дисперсия), если задать ddof=1, то функция возвращает исправленную выборочную дисперсию.
	Функция np.median(a, axis=None, out=None) Возвращает выборочную медиану (вычисляется как центральный элемент $a_{\frac{n-1}{2}}$ отсортированного по неубыванию массива a при нечетном n и как среднее арифметическое двух центральных значений при четном n). Параметры: a - массив; в случае многомерного массива a можно указать ось (axis), вдоль которой вычисляется среднее; out - массив, если он указан, в него помещаются вычисленные значения медиан.
	Функция np.quantile (a, q, axis=None, out=None, interpolation='linear') Возвращает квантиль порядка q (указывается число из интервала $(0,1)$). Параметры: a – массив; в случае многомерного массива a параметр axis - ось (кортеж осей), вдоль которой производятся вычисления; out - массив, если он указан, в него помещаются вычисленные значения квантилей; interpolation - признак, определяющий метод интерполяции в ситуации, когда квантиль расположена между двумя значениями массива ('linear' по умолчанию, есть другие варианты). Первый квартиль Q_1 выборки X вычисляется с помощью функции np.quantile(X,0.25). Третий квартиль Q_3 выборки X вычисляется с помощью функции np.quantile(X,0.75).
Точечные оценки параметров распределения в модуле scipy.stasts	
	Функция sts.moment(x, moment=k, axis=0, nan_policy='propogate') Возвращает выборочный центральный момент порядка k. Параметры: x – выборка; axis - ось, вдоль которой вычисляется оценка; nan_policy - определяет способ обработки пропущенных значений ('propogate' - возвращает nan, 'raise' - генерирует ошибку, 'omit' - игнорирует пропущенные данные).
	Функция sts.skew(x, axis=0, bias=True, nan_policy='propogate') Возвращает выборочный коэффициент асимметрии. Параметры: x – выборка; axis - ось, вдоль которой вычисляется оценка; bias - признак (если False - применяется коррекция для устранения смещенности); nan_policy - определяет способ обработки пропущенных значений.
	Функция sts.kurtosis(x, axis=0, fisher=True, bias=True, nan_policy='propogate') Возвращает выборочный коэффициент асимметрии.

	Параметры: x – выборка; axis - ось, вдоль которой вычисляется оценка; fisher - признак, если равен True (по умолчанию), то в формуле для эксцесса из отношения моментов вычисляется число 3; bias - признак (если False - применяется коррекция для устранения смещенности); nan_policy - определяет способ обработки пропущенных значений.
	Функция sts.igr(x) Вычисляет межквартильный размах - разность между третьим и первым квартилями.
	Функция sts.describe(a, axis, ddof, bians, nan_policy) Возвращает набор оценок основных параметров случайной величины: nobs - объем выборки; minmax - кортеж, содержащий максимальное и минимальное значение выборки; mean - выборочное среднее; variance - исправленная выборочная дисперсия s^2 (в случае задания ddof=1 или по умолчанию) либо выборочная дисперсия (в случае задания ddof=0); skewness - коэффициент асимметрии; kurtosis - коэффициент эксцесса (в случае задания bias=False, коэффициенты асимметрии и эксцесса корректируются на величину смещения). Параметры: a - выборка; axis - задание оси (для многомерной выборки); ddof - признак смещенности (только для дисперсии); bians - признак коррекции (только для асимметрии и эксцесса); nan_policy - задает способ обработки пропущенных данных.
Средства визуализации: построение гистограммы и боксплота в пакете seaborn	
	Функция boxplot(x=None, y=None, hue=None, data=None, order=None, hue_order=None, orient=None, color=None) Строит боксплот. Параметры: x,y, hue - наименование признаков в наборе data; data - датафрейм, или массив numpy, или список; order и hue_order - строки, с помощью которых можно изменить порядок вывода признаков на график; orient - вертикальная или горизонтальная ориентация («v» или «u»); color - задание цвета. Построение боксплота в пакете удобно сочетать с построением гистограммы с помощью функции histplot .

5. Интервальное оценивание параметров распределения по выборке

Загрузка основных модулей	
	import numpy **as** np **import** scipy.stats **as** sts **import** scipy.special **as** sc **import** matplotlib.pyplot **as** plt **import** statsmodels.api **as** sm **import** statsmodels.stats.weightstats **import** statsmodels.stats.proportion **%matplotlib** inline
Построение доверительных интервалов в предположении нормального распределения генеральной совокупности	
	Функция _zconfint_generic (mean, std_mean, alpha, alternative) модуля statsmodels.stats.weightstats Возвращает границы доверительного интервала (1) (см. приложение 1).

	Параметры: `mean` - выборочное среднее, `std_mean` = $\frac{\sigma}{\sqrt{n}}$; `alpha` - уровень значимости; `alternative` - вид доверительного интервала («`two-sided`» - двусторонний, по умолчанию, «`smaller`» - левосторонний, «`larger`» - правосторонний).
	Функция `zconfint(x, alpha=0.05, alternative=«two-sided»)` модуля `statsmodels.api.stats` Возвращает границы доверительного интервала (2). Параметры: `x` - выборка; `alpha` - уровень значимости; `alternative` - вид доверительного интервала («`two-sided`» - двусторонний (по умолчанию), «`smaller`» - левосторонний, «`larger`» - правосторонний).
Построение доверительных интервалов в предположении биномиального распределения генеральной совокупности	
	Функция `proportion_confint(count=m, nobs=n, alpha, method='normal')` модуля `statsmodels.stats.proportion` Возвращает границы доверительного интервала (3) или (4). Параметры: `count` - число успехов; `nobs` - число испытаний; `alpha` - уровень значимости; `method` - вид доверительного интервала (если `'normal'`, строится интервал (6), если `'wilson'`, строится интервал (3)).
	Функция `samplesize_confit_proportion(proportion, half_length, alpha, method='normal')` модуля `statsmodels.stats.proportion` Возвращает минимальный объем выборки, необходимый для достижения желаемой точности при интервальном оценивании вероятности события. Параметры: `proportion` - вероятность успеха; `half_length` - половина ширины требуемого интервала; `alpha` - уровень значимости (по умолчанию 0,05).

6. Проверка гипотез о значениях параметров распределения

Загрузка основных модулей	

	`import numpy as np`
	`import scipy.stats as sts`
	`import matplotlib.pyplot as plt`
	`%matplotlib inline`

Проверка гипотезы о значении математического ожидания нормально распределенной генеральной совокупности при неизвестной дисперсии при двусторонней альтернативе	
	Функция `ttest_1samp(a, popmean, axis = 0, nan_policy = 'propagate')` модуля `scipy.stats` Возвращает: выборочное значение статистики $W = \frac{\overline{X} - m_0}{S/\sqrt{n}}$; достигаемый уровень значимости - p - значение (см. приложение 2). Параметры: `a` – выборка; `popmean` - гипотетическое значение математического ожидания; `nan_policy` - задает способ обработки пропущенных значений.

7. Проверка гипотез о законе распределения, проблема нормализации выборки

Загрузка основных модулей

```
import numpy as np
import scipy.stats as sts
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
```

Экспорт данных из csv или Excell- файла в объект DataFrame библиотеки Pandas

Функция `pd.read_csv('Data.xlsx', sep=',' header = 'infer', index_col=None)`

Создает объект `DataFrame`.

Параметры: `Data.xlsx` – строка с указанием пути к файлу; `sep` - разделитель (по умолчанию `,`); `header` - строка, содержащая имена столбцов (по умолчанию `'infer'` - в качестве имен используется первая строка данных); `index_col` - указывает, какой столбец в файле использовать в качестве индекса, если установить `index_col= False`, первый столбец данных в качестве индекса использоваться не будет.

Функция `pd.read_excel()` имеет аналогичные параметры.

Средства группировки выборки библиотеки numpy

Функция `np.histogram(a, bins=10, range=None, weights=None, density= None)`

Возвращает два массива: `hist` - массив высот столбцов гистограммы; `bin_edges` - массив границ интервалов.

Параметры: `a` - одномерный массив (выборка); `bins` - число интервалов группировки (по умолчанию - 10) или последовательность, задающая границы интервалов; если `bins='auto'` - число интервалов выбирается как максимальное из величин, получаемых по правилу Стерджесса и Фридмана-Диакониса; `range` - начальная и конечная границы интервалов (если параметр не определен, то в качестве границ берутся минимальный и максимальный элементы выборки), элементы выборки вне области `range` игнорируются; `density` - если `True` - строится гистограмма относительных частот (суммарная площадь прямоугольников равна 1); `weights` - массив весов той же формы, что и `a`.

Реализация критерия хи-квадрат проверки гипотезы о законе распределения в модуле scipy.stats

Функция `sts.chisquare (f_obs, f_exp = None, ddof = 0, axis = 0)`

Возвращает наблюдаемое значение статистики и p - значение (т.е. максимальное значение уровня значимости, при котором основная гипотеза принимается).

Параметры: `f_obs` - наблюдаемые частоты (n_i); `f_exp` - частоты гипотетического (согласно основной гипотезе) распределения (np_i) (по умолчанию равные между собой); `ddof` - число параметров гипотетического распределения, оцениваемых по выборке. На вход функции можно подавать многомерный массив. Критерий будет применяться к каждому столбцу массива (если `axis = 1`, то к строке).

Условие использования: все наблюдаемые и гипотетические частоты должны быть не менее 5 ($n_i \geq 5, np_i \geq 5$).

Реализация критерия Шапиро-Уилка в модуле scipy.stats

Функция `sts.shapiro (x, a=None, reta=False)`

Возвращает наблюдаемое значение статистики p - значение; массив параметров (присутствует, если `reta=True`).

	Параметры: x - одномерный массив (выборка); a - массив внутренних параметров (если не заданы, вычисляется самой функцией); reta - признак, нужно ли возвращать вычисленные параметры).
Преобразование Бокса-Кокса модуля scipy.stats	
	Функция sts.boxcox(x, lmbda=None, alpha=None) Возвращает: (1) boxcox - массив, результат преобразования Бокса-Кокса; (2) если параметр lmbda=None , то второй возвращаемый параметр maxlog - значение lmbda, максимизирующее логарифм функции правдоподобия; (3) если lmbda=None и alpha не None, возвращается кортеж, содержащий границы доверительного интервала. Параметры: x - входной одномерный массив положительных чисел (выборка); lmbd - если не None, преобразование выполняется для этого значения; alpha - если не None, то функция возвращает в качестве третьего аргумента $100(1-\alpha)\%$ -й доверительный интервал для параметра lmbda.

8. Корреляционный анализ

Загрузка основных модулей	
	import numpy **as** np **import** scipy.stats **as** sts **import** matplotlib.pyplot **as** plt import pandas as pd import seaborn **%matplotlib** inline
Генерация многомерного нормального распределения в библиотеке numpy	
	Функция np.random.multivariate_normal(mean, cov, n) Возвращает выборку объема n для многомерного нормального распределения с заданным вектором математических ожиданий mean и ковариационной матрицей cov.
Средства визуализации диаграммы рассеивания	
	Функция plt.scatter(x, y) модуля matplotlib.pyplot Строит диаграмму рассеивания признаков x и y. Параметры: x, y - два массива одинаковой длины.
	Функция seaborn.pairplot(data, vars=None, kind='scatter', diag_kind='hist', height=4) пакета seaborn Строит диаграммы рассеивания пар признаков из vars, а также визуализирует распределение отдельных признаков. Параметры: - датафрейм; vars - список имен переменных из vars, которые будут использованы для вывода диаграммы (если не задан, используются все числовые колонки data); kind - тип диаграммы рассеяния (обычная 'scatter' или с линией регрессии 'reg'); diag_kind - тип диагональных графиков ('auto', 'hist', 'kde'); height - высота каждой фацеты (в дюймах).
Расчет выборочных характеристик двумерной выборки	
	Функция np.cov(x, y=None, rowvar=True, bias=False, ddof=None) библиотеки numpy Вычисляет выборочную ковариационную матрицу. Параметры: x - одномерный или двумерный массив. Если одномерный - вычисляется ковариация между x и y. Если двумерный - при значении rowvar=True (по

	умолчанию) вычисляется ковариация между строками массива x, при значении rowvar=False - между столбцами массива x; bias - признак, определяющий способ нормализации, по умолчанию (False) - производится деление на $n-1$, иначе на n; ddof - выполняет функцию, аналогичную признаку bias: при значении ddof=1 производится деление на $n-1$, при значении ddof=0 - деление на n .
	Функция np.corrcoef(x, y=None, rowvar=True) библиотеки numpy Вычисляет выборочную корреляционную матрицу. Параметры: x - одномерный или двумерный массив. Если одномерный - вычисляется коэффициент корреляции между x и y. Если двумерный: при значении rowvar=True (по умолчанию) вычисляется коэффициент корреляции между строками массива x, при значении rowvar=False - между столбцами массива x.
	Метод data.corr(method='pearson') библиотеки pandas Параметры: data - объект DataFrame, method - задает вид коэффициента корреляции 'pearson', 'spearman', ' kendall' (по умолчанию 'pearson').
Средства визуализация корреляционной матрицы	
	Функция seaborn.heatmap(data, annot=None, fmt='.2g', linewidth=0, linecolor='white', cbar=True, cbar_kws=None, cbar_ax=None) пакета seaborn Принимает на вход прямоугольный массив данных и отображает данные с помощью цвета. Цветовая панель показывает соответствие цвета числовым значениям переменной. Параметры: data - объект DataFrame, annot - признак: если True, в каждую ячейку карты выводится значение признака; fmt - строка: задает формат для случая annot= True, linewidth, linecolor - размер и цвет линий, разделяющих ячейки; cbar, cbar_kws, cbar_ax - информация о необходимости вывода, цвете и расположении цветовой панели.
Критерии значимости коэффициента корреляции Спирмена модуля scipy.stats	
	Функция sts.spearmanr(a,b=None, axis=0, nan_policy='propogate') Проверяет гипотезу об отсутствии значимой монотонной связи. Возвращает r – выборочный коэффициент корреляции, p-value - достигаемый уровень значимости. Параметры: a, b - два одномерных или двумерных массива одинакового размера, nan_policy ('propagate', 'raise', 'omit') - задает способ обработки пропущенных (NaN) значений.

9. Регрессионный анализ: парная линейная регрессия

Загрузка основных модулей

```
import numpy as np
import scipy.stats as sts
import matplotlib.pyplot as plt
import pandas as pd
imporseaborn
from sklearn.linear_model import LinearRegression
%matplotlib inline
```

Средства получения оценок линейной регрессии

Вначале надо создать экземпляр класса `LinearRegression`, который будет представлять модель регрессии: `linreg = LinearRegression()`.

Функция `linreg.fit(x, y)`
Вычисляет оценки коэффициентов регрессии b_0, b_1.
Ее параметры: `x` - двумерный массив, размера $n \times 1$ (независимая переменная); `y` - одномерный массив (вектор-строка) длины n (зависимая переменная).

Коэффициент b_0 можно получить, вызвав `linreg.intercept_`.

Коэффициент b_1 можно получить, вызвав `linreg.coef_`.

Функция `linreg.score(x, y)` вычисляет коэффициент детерминации.

Приложение 1

Параметр	Условия	Доверительный интервал	Номер формулы
m	$X \sim N(m,\sigma)$ σ известно	$\overline{X} - u_{\frac{1+\beta}{2}} \cdot \frac{\sigma}{\sqrt{n}} < m < \overline{X} + u_{\frac{1+\beta}{2}} \cdot \frac{\sigma}{\sqrt{n}}$	1
m	$X \sim N(m,\sigma)$ σ неизвестно	$\overline{X} - t_{\frac{1+\beta}{2}}(n-1) \cdot \frac{S}{\sqrt{n}} < m < \overline{X} + t_{\frac{1+\beta}{2}}(n-1) \cdot \frac{S}{\sqrt{n}}$	2
p	$X \sim B(n,p)$ $\sqrt{npq} \gg 1$	$\frac{P^* + \frac{1}{2}\frac{u^2_{\frac{1+\beta}{2}}}{n} - u_{\frac{1+\beta}{2}} \cdot \sqrt{\frac{P^*(1-P^*)}{n} + \frac{1}{4}\frac{u^2_{\frac{1+\beta}{2}}}{n^2}}}{1 + \frac{u^2_{\frac{1+\beta}{2}}}{n}} < p < \frac{P^* + \frac{1}{2}\frac{u^2_{\frac{1+\beta}{2}}}{n} + u_{\frac{1+\beta}{2}} \cdot \sqrt{\frac{P^*(1-P^*)}{n} + \frac{1}{4}\frac{u^2_{\frac{1+\beta}{2}}}{n^2}}}{1 + \frac{u^2_{\frac{1+\beta}{2}}}{n}}$	3
p	$X \sim B(n,p)$ $\sqrt{npq} \gg 1$	$P^* - u_{\frac{1+\beta}{2}} \cdot \sqrt{\frac{P^*(1-P^*)}{n}} < p < P^* + u_{\frac{1+\beta}{2}} \cdot \sqrt{\frac{P^*(1-P^*)}{n}}$	4

Приложение 2

Модификация схемы проверки статистических гипотез с использованием p - значения.

Метод, основанный на использовании так называемого p - значения критерия, позволяет решить для всех уровней значимости одновременно, принять или отклонить основную гипотезу.

Определение. p - *значением* $p(x_1, x_2, ..., x_n)$ нулевой гипотезы, проверяемой по выборке с помощью статистики критерия Z и критической области G_α, называется наименьший уровень значимости, при котором основная гипотеза при имеющейся выборке отклоняется:

$$p(x_1, x_2, ..., x_n) = \min\{\alpha \mid z_{\text{выб}} \in G_\alpha\}.$$

Здесь $z_{\text{выб}} = Z(x_1, x_2, ..., x_n)$ – выборочное значение статистики.

Для всех значений уровня значимости, таких, что $\alpha \le p(x_1, x_2, ..., x_n)$, основная гипотеза принимается, при всех $\alpha > p(x_1, x_2, ..., x_n)$ - отклоняется. Чем меньше p - значение, тем больше оснований отклонить нулевую гипотезу.

Вид формул, по которым вычисляются p - значения, зависит от вида критической области.

Справедливо следующее утверждение:

1. Если критическая область правосторонняя, т.е. имеет вид $(z_{1-\alpha}; +\infty)$, где $z_{1-\alpha}$ - квантиль порядка $1-\alpha$, то p - значение находится по формуле $p(x_1, x_2, ..., x_n) = P\{Z > z_{\text{выб}} | H_0\}$.

2. Если критическая область левосторонняя, т.е. имеет вид $(-\infty; z_\alpha)$, где z_α - квантиль порядка α, то p - значение находится по формуле $p(x_1, x_2, ..., x_n) = P\{Z < z_{\text{выб}} | H_0\}$.

3. Если критическая область двусторонняя, т.е. имеет вид $(-\infty; z_{\alpha/2}) \cup (z_{1-\alpha/2}; +\infty)$, то p - значение находится по формуле $p(x_1, x_2, ..., x_n) = 2 \cdot \min\{p, 1-p\}$, где $p = P\{Z < z_{\text{выб}} | H_0\}$.