Статистика, прикладной поток

Практическое задание 3

В данном задании вы найдете оценки максимального правдоподобия по реальным данным для некоторых вероятностных моделей, изучите bias-variance разложение, а также исследуете оценки в схеме Бернулли.

Правила:

- Дедлайн **20 октября 23:59**. После дедлайна работы не принимаются кроме случаев наличия уважительной причины.
- Выполненную работу нужно отправить на почту `mipt.stats@yandex.ru`, указав тему письма "`[applied] Фамилия Имя - задание 3`". Квадратные скобки обязательны. Если письмо дошло, придет ответ от автоответчика.
- Прислать нужно ноутбук и его pdf-версию (без архивов). Названия файлов должны быть такими: `3.N.ipynb` и `3.N.pdf`, где `N` - ваш номер из таблицы с оценками.
- Решения, размещенные на каких-либо интернет-ресурсах не принимаются. Кроме того, публикация решения в открытом доступе может быть приравнена к предоставлении возможности списать.
- Для выполнения задания используйте этот ноутбук в качестве основы, ничего не удаляя из него.
- Никакой код из данного задания при проверке запускаться не будет.

Баллы за задание:

- Задача 1 - 10 баллов **О2**
- Задача 2a - 5 баллов **О2**
- Задача 2b - 10 баллов **О2**
- Задача 3 - 15 баллов **О2**
- Задача 4 - 15 баллов **О3**

In [56]:

```
import numpy as np
import pandas as pd
import scipy.stats as sps
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(font_scale=1.3)

import warnings
warnings.simplefilter("ignore")

%matplotlib inline
```

Задача 1.

В этой задаче нужно сделать оценку максимального правдоподобия для многомерного нормального распределения по датасету химимического анализа вин трех разных сортов в Италии. Скачайте данные по ссылке https://archive.ics.uci.edu/ml/datasets/wine (https://archive.ics.uci.edu/ml/datasets/wine) и загрузите их с помощью библиотеки pandas.

In [57]:

```
names= ['Cultivar', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium',
'Total phenols', 'Flavanoids',
        'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD3
15 of diluted wines', 'Proline']

sample = pd.read_csv('wine.data', names=names)
```

Пусть выборка $X = (X_1, \ldots, X_n)$ такова, что каждый ее элемент имеет многомерное нормальное распределение со средним вектором $a \in \mathbb{R}^d$ и матрицей ковариаций $\Sigma \in \mathbb{R}^{d \times d}$.

Запишите оценку максимального правдоподобия для параметров a и Σ.

Ответ: $a = \overline{X}$

$Z = \frac{1}{n}(\sum X_i - \overline{X})(\sum X_i - \overline{X})^T$

Рассмотрим колонки "Alcalinity of ash", "Nonflavanoid phenols", "Proanthocyanins", "Hue". Предположим, что данные в них образуют выборку из многомерного нормального распределения с неизвестными параметрами, которые вам нужно оценить.

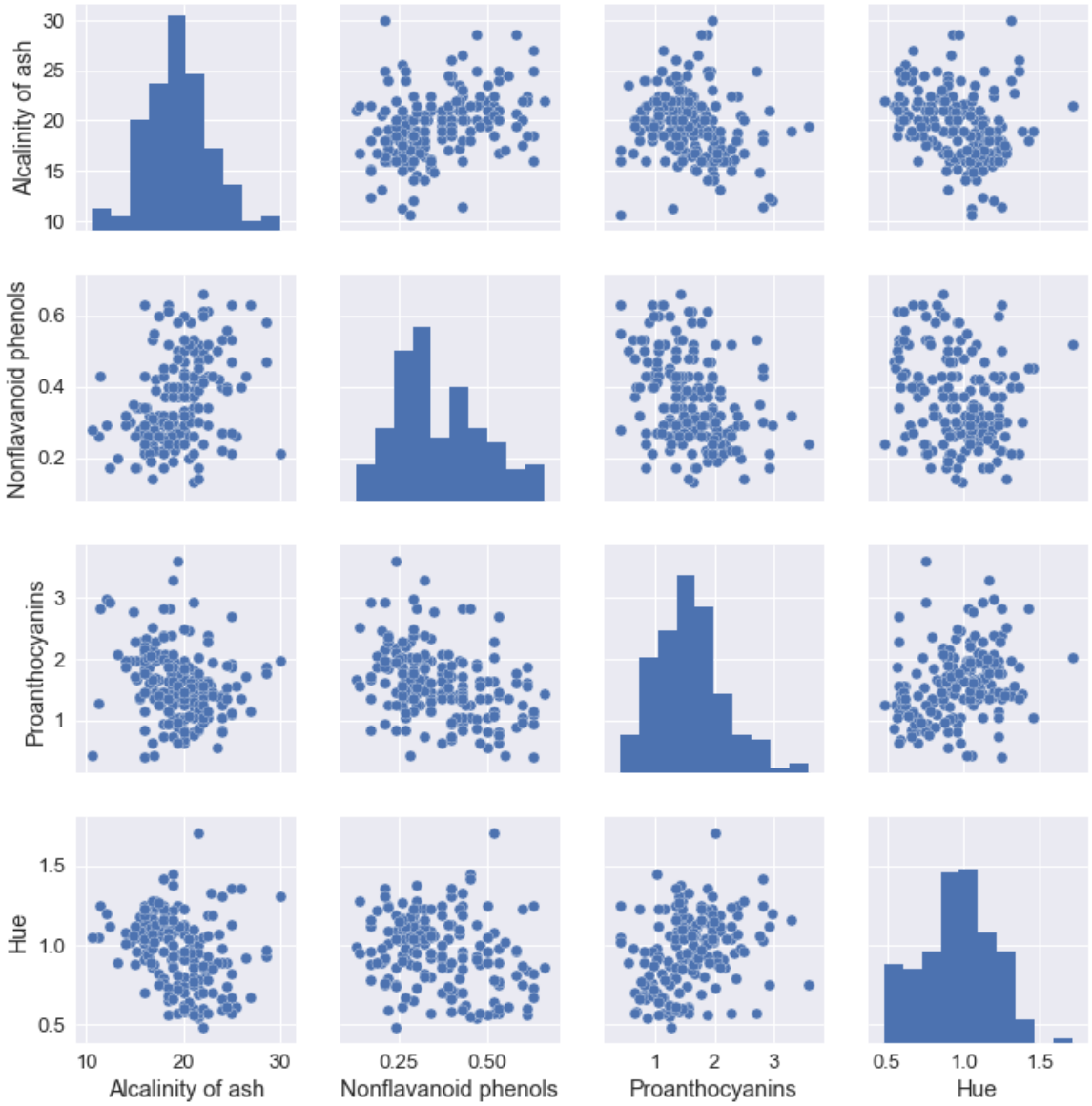
Визуализируйте рассматриваемые данные с помощью `seaborn.pairplot`, чтобы убедиться в том, что данные визуально похожи на нормальное распределение:

In [61]:

```
names = ["Alcalinity of ash", "Nonflavanoid phenols", "Proanthocyanins", "Hue"]
new_sample = sample[names]
sns.pairplot(new_sample)
```

Out[61]:

```
<seaborn.axisgrid.PairGrid at 0x1e427b9b320>
```



Напишите функцию подсчета оценки максимального правдоподобия для вектора средних μ и матрицы ковариаций Σ по выборке:

In [62]:

```
def mle_for_mean(sample):
    """
    :param sample: выборка из многомерного нормального распределения
    :return: ОМП для вектора средних
    """

    return np.mean(sample, axis=0)
```

In [63]:

```
def mle_for_covariance_matrix(sample):
    """
    :param sample: выборка из многомерного нормального распределения
    :return: ОМП для матрицы ковариаций
    """
    mean = mle_for_mean(sample)
    sample = sample - mean
    return np.dot(sample.T, sample) / np.size(sample, axis=0)
```

In [64]:

```
mu = mle_for_mean(new_sample)
covariance = mle_for_covariance_matrix(new_sample)
```

Визуализируйте полученный результат. Для каждой пары признаков постройте график, на котором будут:

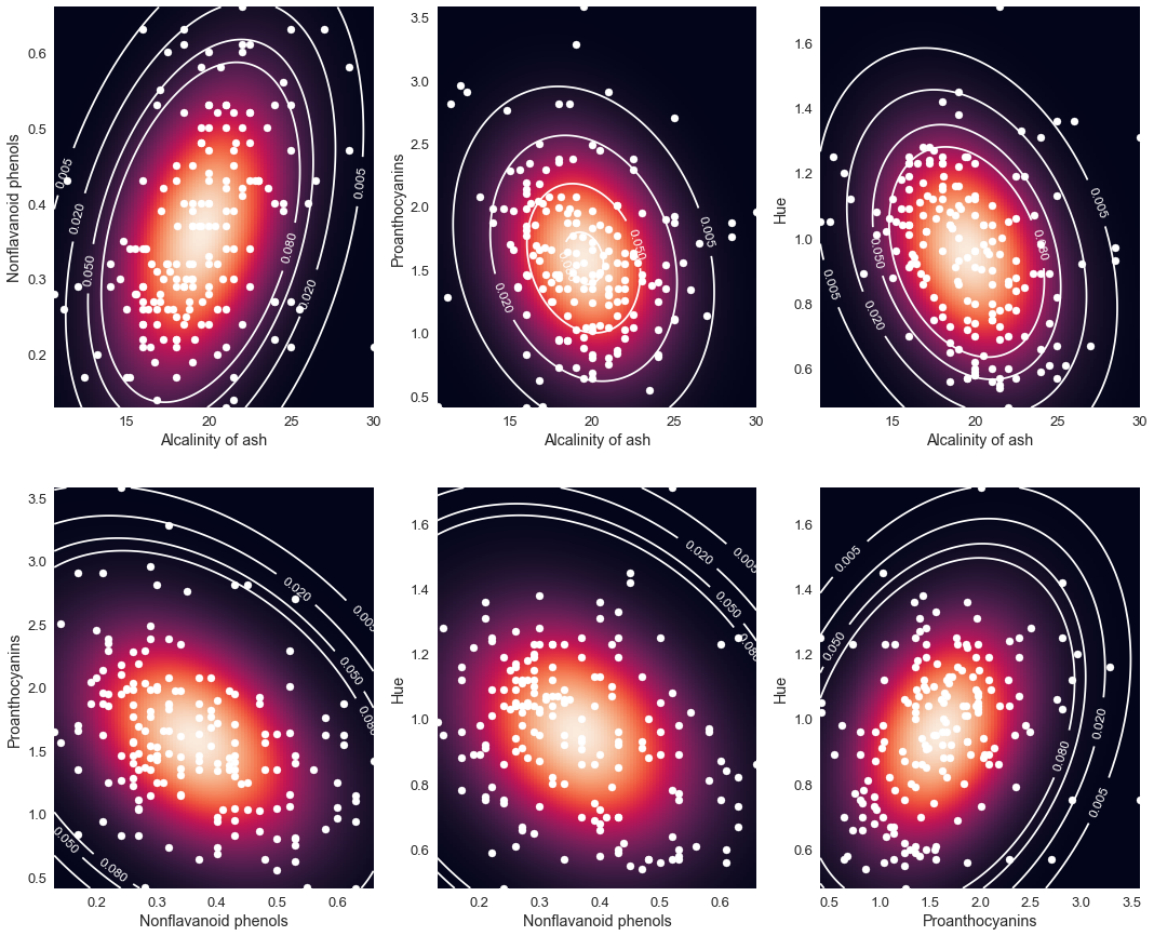
1) Точки выборки.

2) Плотность нормального распредления с оцененными параметрами в виде линий уровня.

hint: используйте функции `plt.pcolormesh` и `plt.clabel`

In [94]:

```
from itertools import combinations

plt.figure(figsize=(18, 15))

for i, pair in enumerate(combinations(range(4), 2)):
    pair = list(pair)
    plt.subplot(2, 3, i+1)

    sub_mu = mu[pair]
    sub_cov = covariance[pair][:, pair]

    x_values = new_sample[names[pair[0]]]
    y_values = new_sample[names[pair[1]]]

    xmin = np.min(x_values)
    xmax = np.max(x_values)
    ymin = np.min(y_values)
    ymax = np.max(y_values)

    grid = np.meshgrid(np.linspace(xmin, xmax, 100), np.linspace(ymin, ymax, 500))
    rv = sps.multivariate_normal(mean=sub_mu, cov=sub_cov)
    density = rv.pdf(np.dstack(grid))

    plt.pcolormesh(grid[0], grid[1], density)  # закрасить с интенсивностью density, cm
ap - цветовая схема
    CS = plt.contour(grid[0], grid[1], density, [0.005, 0.02, 0.05, 0.08], colors='w')
 # нарисовать указанные линии уровня
    plt.clabel(CS, fontsize=12, colors='w')

    plt.scatter(x_values, y_values, color='w')


    plt.xlim(xmin, xmax)
    plt.ylim(ymin, ymax)
    plt.xlabel(names[pair[0]])
    plt.ylabel(names[pair[1]])

plt.show()
```



Выводы: На первом графике можно увидеть, что распределение "Alcalinity of ash", "Nonflavanoid phenols", "Proanthocyanins", "Hue" похоже на многомерное нормальное. Значит, его можно оценить ОМП для нормального распределения. На втором графике можно убедиться, что ОМП действительно оценивает данные случайные величины

Задача 2.

На сегодняшний день возобновляемые источники энергии становятся все более востребованными. К таким источникам относятся, например, ветрогенераторы. Однако их мощность очень трудно прогнозировать. В частности, выработка энергии при помощи ветрогенератора сильно зависит от скорости ветра. Поэтому предсказание скорости ветра является очень важной задачей. Скорость ветра часто моделируют с помощью распределения Вейбулла, которое имеет плотность:

$$p_\theta(x) = \frac{kx^{k-1}}{\lambda^k} e^{-(x/\lambda)^k} I(x \geq 0),$$

где $\theta = (k, \lambda)$ — двумерный параметр. К сожалению, найти точную оценку максимального правдоподобия на θ не получится. В данном задании нужно найти оценку максимального правдоподобия приближенно с помощью поиска по сетке.

За распределение Вейбулла отвечает класс `weibull_min` из модуля `scipy.stats`, которое задается так: `weibull_min(c=`k`, scale=`λ`)`.

Выборка: Создайте выборку по значениям среднесуточной скорости ветра на некоторой местности для нескольких лет (не менее трех). Выборку можно получить отсюда (http://www.atlas-yakutia.ru/weather/wind/climate_russia-III_wind_2018.html), используя скрипт `script.py`. Откройте командную строку в той же папке, запустите скрипт (`python3 script.py`) и следуйте инструкциям; на вопрос `Pick data type:` надо ответить `5`, чтобы выбрать данные по ветру. В полученном csv-файле надо выбрать данные (столбец `Mean`) за некоторый промежуток времени.

a). Найдите оценку максимального правдоподобия параметра $\theta = (k, \lambda)$ с точностью 10^{-5} при помощи поиска по двумерной сетке.

Двумерную сетку можно создать с помощью функции `numpy.mgrid[from:to:step, from:to:step]`. Если попробовать сразу создать сетку с шагом 10^{-5}, то может не хватить памяти. Поэтому найдите сначала максимум по сетке с большим шагом, затем сделайте сетку с маленьким шагом в окрестности найденной точки. При вычислении без циклов, возможно, придется создавать трехмерные объекты.

Функция `numpy.argmax` выдает не очень информативный индекс, поэтому пользуйтесь следующей функцией.

In [ ]:

```
def cool_argmax(array):
    return np.unravel_index(np.argmax(array), array.shape)
```

Нарисуйте график плотности с параметрами, соответствующим найденным ОМП, а так же нанесите на график гистограмму.

Решение:

In [ ]:

```
<...>
```

b). Обозначим $\hat{\theta} = \left(\widehat{\lambda}, \hat{k}\right)$ — ОМП. Запишите уравнение правдоподобия (все частные производные в точке экстремума логарифмической функции правдоподобия должны быть равны 0). Используя одно из равенств, можно выразить $\widehat{\lambda}$ через значения $X_1, \ldots, X_n, \hat{k}$; подставив это выражение в другое равенство, получить уравнение на $\hat{k}$. Решите это уравнение приближенно с помощью метода Ньютона, рассказанного в рамках курса методов оптимизации, и получите $\hat{k}$, а значит, и $\widehat{\lambda}$.

Решение:

In [ ]:

```
<...>
```

Вывод: <...>

Задача 3.

Пусть $\hat{\theta}$ — оценка параметра θ и $MSE_{\hat{\theta}}(\theta) = \mathsf{E}_\theta\Big(\hat{\theta} - \theta\Big)^2$ — среднеквадратичная ошибка оценки $\hat{\theta}$.

Тогда справедливо bias-variance разложение:

$$MSE_{\hat{\theta}}(\theta) = \mathrm{bias}^2_{\hat{\theta}}(\theta) + \mathrm{var}_{\hat{\theta}}(\theta);$$

$$\mathrm{bias}_{\hat{\theta}}(\theta) = \mathsf{E}_\theta\hat{\theta} - \theta;$$

$$\mathrm{var}_{\hat{\theta}}(\theta) = \mathsf{D}_\theta\hat{\theta}.$$

а). Пусть $X = (X_1, \ldots, X_n)$ — выборка из распределения $U[0, \theta]$. Рассмотрим класс оценок $\mathcal{K} = \big\{cX_{(n)}, c \in \mathbb{R}\big\}$. Выпишите формулы bias-variance разложения для таких оценок.

Ответ:

$$\mathrm{bias}_{\hat{\theta}}(\theta) = \mathsf{E}_\theta\hat{\theta} - \theta = c\frac{n}{n+1}\theta - \theta;$$

$$\mathrm{var}_{\hat{\theta}}(\theta) = \mathsf{D}_\theta cX_{(n)} = c^2\mathsf{D}_\theta X_{(n)} = c^2\frac{n}{(n+1)^2(n+2)}\theta^2;$$

$$MSE_{\hat{\theta}}(\theta) = \theta^2\left(c\frac{n}{(n+1)} - 1\right)^2 + \theta^2c^2\frac{n}{(n+1)^2(n+2)}$$

Заметим, что каждая компонента bias-variance разложения пропорциональна θ^2. Это означает, достаточно рассмотреть поведение компонент при изменении c только для одного значения θ.

Постройте график зависимости компонент bias-variance разложения от c для $n = 5$ и $\theta = 1$. С помощью функций `plt.xlim` и `plt.ylim` настройте видимую область графика так, чтобы четко была отображена информативная часть графика (по оси *x* примерно от `0.9` до `1.4`). Не забудьте добавить сетку и легенду, а также подписать оси.

На графике проведите вертикальные линии с координатами c, соответствующими минимуму функции риска, несмещенной оценке и ОМП.

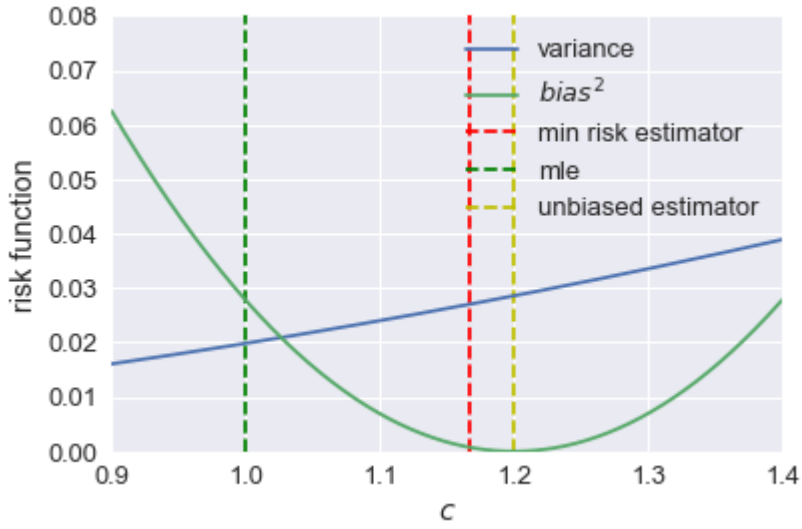
Сделайте выводы. Какое c дает минимум функции риска? Каково поведение компонент разложения? Как соотносятся полученные оценки?

Решение:

In [96]:

```
n = 5
theta = 1
xmin, xmax = 0.9, 1.4
ymin, ymax = 0, 0.08
grid = np.linspace(xmin, xmax, 1000)
bias = grid*n/(n+1)*theta-theta
var = theta**2*grid**2*n/(n+1)**2/(n+2)
mse = bias**2+var
#plt.plot(grid, mse, label='risk_function')
plt.plot(grid, var, label='variance')
plt.plot(grid, bias**2, label='$bias^2$')

# вычисление различных значений c
min_risk = grid[np.argmin(mse)]
mle = theta
unbiased = theta*(n+1)/n
# вертикальные линии
plt.vlines(min_risk, ymin, ymax, color='r', linestyles='dashed', label='min risk estima
tor')
plt.vlines(mle, ymin, ymax, color='g', linestyles='dashed', label='mle')
plt.vlines(unbiased, ymin, ymax, color='y', linestyles='dashed', label='unbiased estima
tor')

print("c=%f дает минимум функции риска, ОМП=%f, несмещенная оценка равна %f" % (min_ris
k, mle, unbiased))

plt.xlabel("$c$")
plt.ylabel("risk function")
plt.xlim(xmin, xmax)
plt.ylim(ymin, ymax)
plt.legend()
plt.show()
```

```
c=1.166767 дает минимум функции риска, ОМП=1.000000, несмещенная оценка ра
вна 1.200000
```



Вывод: минимум функции риска дает $c = 1.17$. $bias^2 = 0$ достигается при $c = 1.2$, что соответствует несмещенной оценке. $Variance$ имеет линейную зависимиоть от c, а $bias$ -- квадратичную.

b) Пусть $X = (X_1, \ldots, X_n)$ — выборка из распределения $\mathcal{N}(0, \sigma^2)$. Рассмотрим класс оценок $\mathcal{K} = \left\{ \frac{1}{c} \sum\limits_{i=1}^{n} \left(X_i - \overline{X} \right)^2,\ c \in \mathbb{R} \right\}$. Выпишите формулы bias-variance разложения для таких оценок.

Ответ:

$$\theta = \sigma^2$$

$$\mathrm{bias}_{\hat{\theta}}(\theta) = \mathsf{E}_\theta \hat{\theta} - \theta = \frac{n-1}{c}\theta - \theta;$$

$$\mathrm{var}_{\hat{\theta}}(\theta) = \mathsf{D}_\theta \frac{1}{c} \sum_{i=1}^{n} \left(X_i - \overline{X} \right)^2 = \frac{1}{c^2} \mathsf{D}_\theta \sum_{i=1}^{n} \left(X_i - \overline{X} \right)^2 = \frac{2(n-1)}{c^2}\theta^2.$$

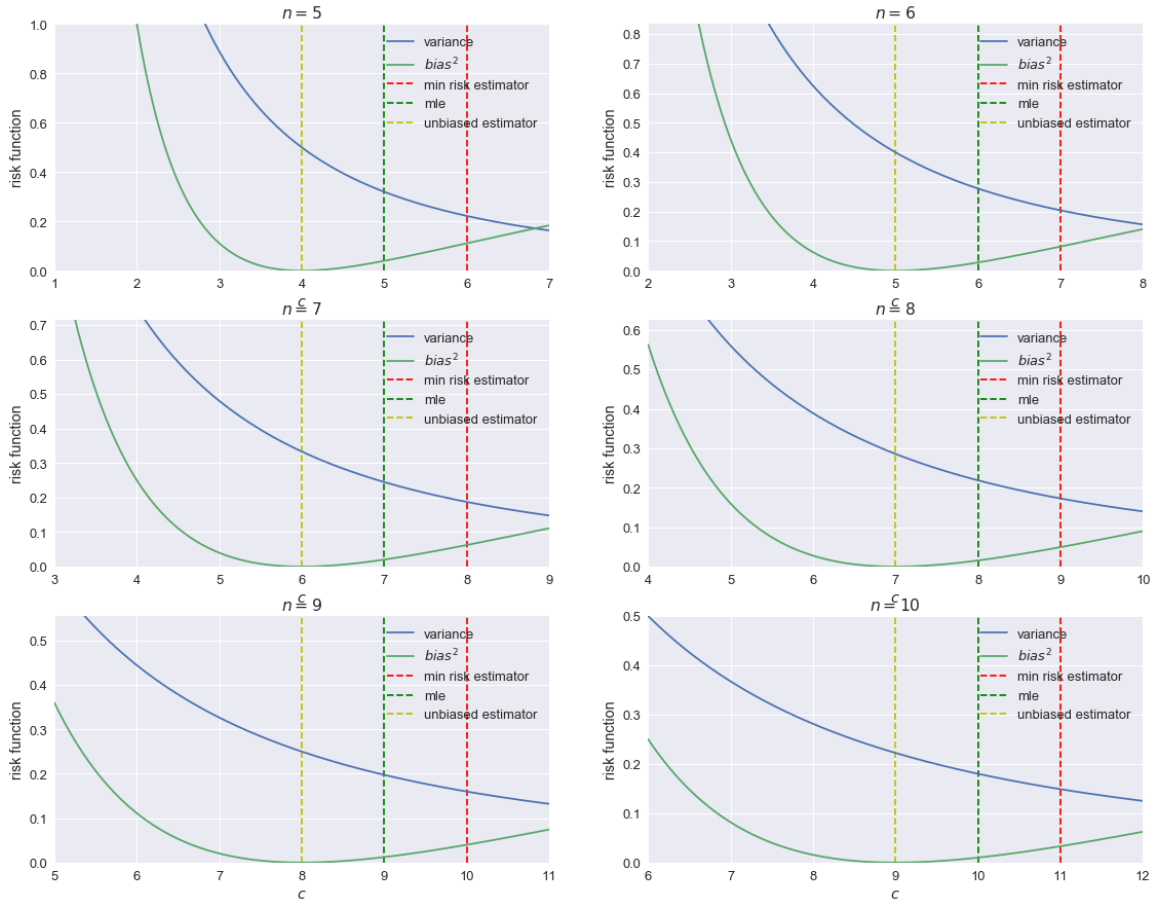
Повторите исследование, аналогичное пункту *a)* для $\sigma^2 = 1$ и $n \in \{5, 10\}$. Для экономии места нарисуйте два графика в строчку. Не забудьте сделать выводы.

Решение:

In [112]:

```
theta = 1
plt.figure(figsize=(19,15))
for n in range(5, 11):
    plt.subplot(3,2,n-4)

    xmin, xmax = n-4, n+2
    ymin, ymax = 0, 5/n
    grid = np.linspace(xmin, xmax, 1000)
    bias = (n-1)/grid*theta-theta
    var = theta**2*2*(n-1)/grid**2
    mse = bias**2+var

    plt.plot(grid, var, label='variance')
    plt.plot(grid, bias**2, label='$bias^2$')

    # вычисление различных значений c
    min_risk = grid[np.argmin(mse)]
    mle = n
    unbiased = n-1
    # вертикальные линии
    plt.vlines(min_risk, ymin, ymax, color='r', linestyles='dashed', label='min risk es
timator')
    plt.vlines(mle, ymin, ymax, color='g', linestyles='dashed', label='mle')
    plt.vlines(unbiased, ymin, ymax, color='y', linestyles='dashed', label='unbiased es
timator')

    print("c=%f дает минимум функции риска, ОМП=%f, несмещенная оценка равна %f" % (min
_risk, mle, unbiased))

    plt.xlabel("$c$")
    plt.ylabel("risk function")
    plt.xlim(xmin, xmax)
    plt.ylim(ymin, ymax)
    plt.title("$n=%d$" % n)
    plt.legend()
plt.show()
```

```
c=6.003003 дает минимум функции риска, ОМП=5.000000, несмещенная оценка ра
вна 4.000000
c=7.003003 дает минимум функции риска, ОМП=6.000000, несмещенная оценка ра
вна 5.000000
c=8.003003 дает минимум функции риска, ОМП=7.000000, несмещенная оценка ра
вна 6.000000
c=9.003003 дает минимум функции риска, ОМП=8.000000, несмещенная оценка ра
вна 7.000000
c=10.003003 дает минимум функции риска, ОМП=9.000000, несмещенная оценка р
авна 8.000000
c=11.003003 дает минимум функции риска, ОМП=10.000000, несмещенная оценка
равна 9.000000
```



Вывод: Можно заметить, что минимум фунуции риска достигается при $c = n + 1$. Минимум $bias^2$ достигается при $c = n - 1$, что соответствуеь несмещенной оценке. $variance$ монотонно убывает.

c) Пусть $X = (X_1, \dots, X_n)$ — выборка из распределения $Exp(\theta)$. Рассмотрим класс оценок $\mathcal{K} = \left\{ \frac{c}{X_1 + \dots + X_n},\ c \in \mathbb{R} \right\}$. Выпишите формулы bias-variance разложения для таких оценок.

Ответ:

Повторите исследование, аналогичное пункту *c)* для $\theta = 1$ и $n = 7$. Не забудьте сделать выводы.

Решение:

In [ ]:

```
<...>
```

Вывод: <...>

Сделайте вывод по результатам пунктов *a)*, *b)*, *c)*.

Общий вывод: <...>

Задача 4.

Пусть $X = (X_1, \ldots, X_n), n = 9$ — выборка из распределения $Bern(\theta), \theta \in [0, 1]$. При сравнении оценок будем рассматривать среднеквадратичный риск $MSE_{\hat{\theta}}(\theta) = \mathsf{E}_\theta\left(\hat{\theta} - \theta\right)^2$.

Известно, что оценка $\overline{X}$ параметра сдвига θ является наилучшей оценкой в среднеквадратичном подходе среди всех несмещенных оценок.

В минимаксном подходе среди всех оценок наилучшей является оценка Ходжеса-Лемана:
$\tilde{\theta} = \overline{X} + \frac{1}{1+\sqrt{n}}\left(\frac{1}{2} - \overline{X}\right)$.

Сравним точность оценок $\overline{X}$ и $\tilde{\theta}$.

1. Нанесите на один график функции риска $MSE_{\overline{X}}(\theta)$ и $MSE_{\tilde{\theta}}(\theta)$. Вычислите долю тех θ, при которых $MSE_{\tilde{\theta}}(\theta) < MSE_{\overline{X}}(\theta)$.

Решение:

In [ ]:

```
n = 9
grid = np.linspace(0, 1, 100)
mse1 = n*grid*(1-grid)
mse2 =
```

Ответ: <...>

2. Проведите эксперимент. Сгенерируйте параметры $\theta = (\theta_1, \ldots, \theta_{1000})$ из распределения $U[0, 1]$ независимо, после чего сгенерируйте выборки
$X_k = (X_{k1}, \ldots, X_{kn}) \sim Bern(\theta_k),\ 1 \leqslant k \leqslant 1000,\ n = 9.$

По каждой из выборок X_k вычислите оценки $\overline{X_k}$ и $\tilde{\theta}_k$ и определите, какая из них ближе к θ_k. В какой доле случаев оценка Ходжеса-Лемана оказалась лучше? Похож ли результат на ответ в прошлом пункте? Почему?

Решение:

In [ ]:

```
<...>
```

Вывод: <...>

3. Рассмотрим функцию $p(\theta) = \mathsf{P}_\theta \left(\left| \tilde{\theta} - \theta \right| < \left| \overline{X} - \theta \right| \right)$ — вероятность того, что оценка $\tilde{\theta}$ оказалась ближе к θ, чем $\overline{X}$. Можно показать, что при $\theta \leqslant 1/2$ верно равенство $p(\theta) = \mathsf{P}_\theta \left(\frac{\theta - d_n}{1 - 2d_n} \leqslant \overline{X} \leqslant \frac{1}{2} \right)$, где $d_n = \frac{1}{4(1+\sqrt{n})}$, причем функция $p(\theta)$ симметрично относительно $1/2$. Такую вероятность можно вычислить, используя функцию распределения (`cdf`) биномиального распределения.

Если параметр θ случаен, то вероятность того, что оценка $\tilde{\theta}$ оказажется ближе к θ, равна $p_* = \int\limits_0^1 p(t)dt$, что соответствует площади под кривой графика функции $p(\theta)$.

Постройте график функции $p(\theta)$. Посчитайте вероятность p_* с помощью метода прямоугольников. Сделайте выводы.

Решение:

In [ ]:

```
<...>
```

Ответ: <...>

4. Исследуйте, как зависит вероятность p_* от размера выборки, постройте график этой зависимости. Сделайте выводы.

Решение:

In [ ]:

```
<...>
```

Ответ: <...>