

Дисциплина: «Теория вероятностей и математическая статистика»

Расчетное задание №2 «Реализация и исследование выборки при различных распределениях»

1. Цель работы

Исследовать выборки, проверить гипотезы различными критериями. Сделать выводы по проделанной работе.

2. Реализация выполнения задания №1

Критерий χ^2 - статистический критерий для проверки гипотезы H_0 , что наблюдаемая случайная величина подчиняется некому теоретическому закону распределения.

Гипотеза H_0 : с. в. X подчиняется закону распределения $F(x)$.

В зависимости от значения критерия χ^2 , гипотеза H_0 может приниматься, либо отвергаться:

$\chi^2 < \chi^2_{\alpha}$, гипотеза H_0 выполняется. В случае $\chi^2 > \chi^2_{\alpha}$ – не выполняется.

$$\chi^2 = \frac{10}{N} \sum_{i=0}^9 (v_i - 0,1N)^2$$

№	x	f(x)	f(x)]	число	№	x	f(x)	f(x)]	число
1	0.990000	-2.30582221	22	2	51	50.490025	25.49126421	21	2
2	1.980000	1.06829893	98	9	52	51.480026	0.38785293	93	9
3	2.970000	-0.35732524	25	2	53	52.470028	-1.01249462	62	6
4	3.960000	-15.12527072	70	7	54	53.460030	2.48628403	03	0
5	4.950000	0.51455251	52	5	55	54.450031	0.02679981	81	8
6	5.940000	-0.81925452	54	5	56	55.440033	-2.14641612	12	1
7	6.929999	3.51505156	51	5	57	56.430035	1.12735224	24	2
8	7.919999	0.13280802	08	0	58	57.420036	-0.32740528	28	2
9	8.909999	-1.66357872	78	7	59	58.410038	-10.74566514	14	1
10	9.899999	1.39968883	88	8	60	59.400040	0.54889306	06	0
11	10.889998	-0.21434742	47	4	61	60.390041	-0.77547153	53	5
12	11.879998	-4.98300107	01	0	62	61.380043	3.90971868	68	6
13	12.869998	0.69484042	40	4	63	62.370045	0.16014362	62	6
14	13.859998	-0.61909095	90	9	64	63.360046	-1.56704905	05	0
15	14.849998	6.84168284	82	8	65	64.350044	1.48195107	07	0
16	15.839997	0.27038283	82	8	66	65.340042	-0.18652235	35	3
17	16.829998	-1.25376862	68	6	67	66.330040	-4.37329183	83	8
18	17.819998	1.88241711	17	1	68	67.320038	0.73525291	91	9
19	18.809998	-0.07928283	82	8	69	68.310036	-0.58270062	62	6
20	19.799997	-2.91867370	73	7	70	69.300034	8.40629659	59	5
21	20.789997	0.91178886	88	8	71	70.290031	0.29928199	99	9
22	21.779997	-0.44933836	38	3	72	71.280029	-1.18725083	83	8

Выполнил: студент группы 23656/4 _____ Э.А Гасанов
 (подпись, дата)

Проверил: доцент, к.т.н. _____ Д.С. Лаврова
 (подпись, дата)

23	22.769997	76.33039996	99	9	73	72.270027	2.01027994	94	9
24	23.759996	0.41820703	07	0	74	73.260025	-0.05244911	11	1
25	24.749996	-0.96095652	56	5	75	74.250023	-2.68274752	52	5
26	25.739996	2.68694741	47	4	76	75.240021	0.96193304	04	0
27	26.729996	0.05296668	66	6	77	76.230019	-0.41761564	64	6
28	27.719995	-2.00766461	64	6	78	77.220016	-73.52211091	91	9
29	28.709995	1.18851118	11	1	79	78.210014	0.44993572	72	7
30	29.699995	-0.29870703	07	0	80	79.200012	-0.91088615	15	1
31	30.689995	-8.36842321	23	2	81	80.190010	2.92333343	43	4
32	31.679995	0.58341614	16	1	82	81.180008	0.07977048	48	4
33	32.669994	-0.73442326	23	2	83	82.170006	25.49126451	51	5
34	33.659996	4.38430703	07	0	84	83.160004	0.38785223	23	2
35	34.649998	0.18709592	95	9	85	84.150002	-1.01249490	90	9
36	35.639999	-1.48015857	58	5	86	85.139999	2.48628457	57	5
37	36.630001	1.56901612	16	1	87	86.129997	0.02679903	03	0
38	37.620003	-0.15955252	52	5	88	87.119995	-2.14641692	92	9
39	38.610004	-3.90022426	24	2	89	88.109993	1.12735288	88	8
40	39.600006	0.77641911	19	1	90	89.099991	-0.32740551	51	5
41	40.590008	-0.54811316	13	1	91	90.089989	-10.74566574	74	7
42	41.580009	10.81684149	41	4	92	91.079987	0.54889341	41	4
43	42.570011	0.32808685	86	8	93	92.069984	-0.77547148	48	4
44	43.560013	-1.12594043	40	4	94	93.059982	3.90971832	32	3
45	44.550014	2.14995353	53	5	95	94.049980	0.16014310	10	1
46	45.540016	-0.02616162	61	6	96	95.039978	-1.56704919	19	1
47	46.530018	-2.48165855	58	5	97	96.029976	1.48195132	32	3
48	47.520020	1.01381817	18	1	98	97.019974	-0.18652282	82	8
49	48.510021	-0.38709292	92	9	99	98.009972	-4.37329138	38	3
50	49.500023	-25.06370006	00	0	100	98.999969	0.73525277	77	7

$$\chi^2 = 7.8$$

Табличное значение χ^2 для уровня значимости 0,05 равняется 16,9, а для уровня значимости 0,01 – 21,7.

Наше значение меньше , чем вышесказанные , следовательно , гипотеза принимается.

3. Реализация выполнения задания №2

α	0.005	0.01	0.025	0.05	0.1	0.15	0.2	0.3	0.4
Критерий Фросини	0.779	0.763	0.678	0.589	0.500	0.448	0.400	0.356	0.314
Критерий Фросини (табличные)	0.7428			0.5785	0.4966	0.4456		0.4083	
Андерсон-Дарлинг	4.478	3.766	3.144	2.556	1.918	1.672	1.391	1.133	0.922
Андерсон-Дарлинг(табличные)	3.8781		3.0775	2.4924	1.9330	1.6212			

Напишем процедуры вычисления статистик критерия Фросини:

$$B = \frac{1}{\sqrt{N}} \sum_{i=1}^N \left| U_i - \frac{i - 0.5}{N} \right|$$

и Андерсона – Дарлинга

$$S = -N - 2 \sum_{i=1}^N \left(\frac{2i-1}{2N} \ln U_i + \left(1 - \frac{2i-1}{2n} \right) \ln (1 - U_i) \right)$$

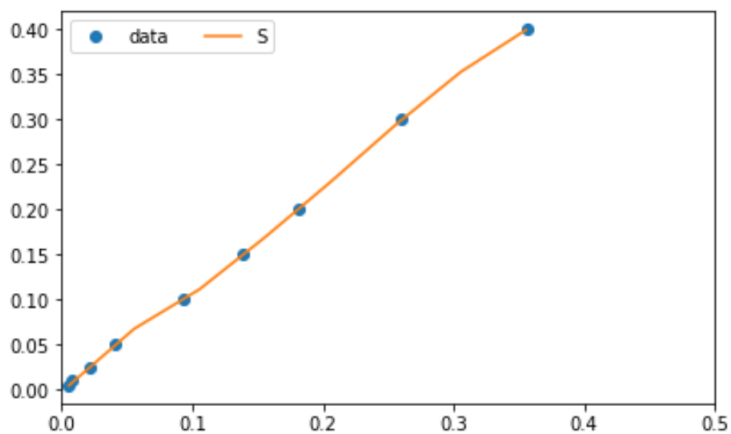
где U_i -элементы вариационного ряда, построенного по выборке X_1, \dots, X_N .

Промоделируем достаточно большое число выборок из равномерного распределения и посчитаем процент, удовлетворяющих критериям О и П на разных уровнях значимости.

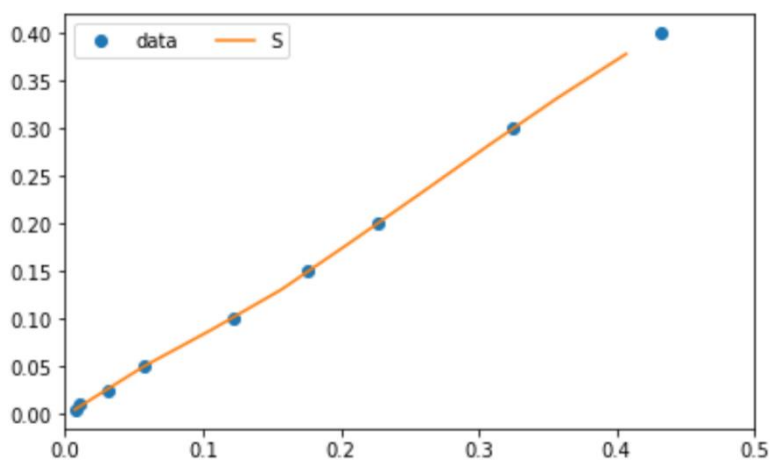
Таблица уровней значимости

α	0.005	0.01	0.025	0.05	0.1	0.15	0.2	0.3	0.4
α О	0.004	0.007	0.016	0.039	0.084	0.120	0.175	0.254	0.357
α П	0.007	0.013	0.023	0.051	0.110	0.158	0.231	0.321	0.433

Интерполируем полученные значения и построим $\alpha = f(a_0)$



$$\alpha = f(a_{\pi})$$



Вычислим значения критериев О и П в точках $\alpha=0.01, 0.05, 0.1, 0.15$ по полученным функциям:

Таблица 1.

$\alpha_{o/\pi}$	0.01	0.05	0.1	0.15
$\alpha = f(a_0)$	0.01406444	0.06010559	0.11502299	0.16745506
$\alpha = f(a_{\pi})$	0.00869123	0.04212751	0.08725126	0.11989893

Проверим полученные значения моделированием. Сначала найдем значения критериев Фросини и Андерсона - Дарлинга в полученных α .

Таблица значений критериев

Таблица 2.

α	0.014	0.060	0.115	0.167	0.009	0.042	0.087	0.120
Критерий Фросини	0.729	0.573	0.484	0.431	0.756	0.599	0.509	0.472
Критерий Андерсона-Дарлинга	3.506	2.326	1.818	1.544	3.946	2.678	2.036	1.837

Теперь рассчитаем значения критериев О и П с этими α

Таблица уровней значимости

Таблица 3

Критерий О	0.008	0.060	0.112	0.159
Критерий П	0.008	0.053	0.113	0.162

Сгенерируем N3 выборки объема 20 из бета-распределения 1-го рода.

Таблица мощности критерия H_1

$\alpha_{o/p}$	0.01	0.05	0.1	0.15
Критерий О	0.035	0.120	0.185	0.235
Критерий П	0.035	0.105	0.190	0.230

Таблица мощности критерия H_2

$\alpha_{o/p}$	0.01	0.05	0.1	0.15
Критерий О	0.440	0.730	0.805	0.870
Критерий П	0.470	0.715	0.815	0.870

Таблица мощности критерия H_3

$\alpha_{o/p}$	0.01	0.05	0.1	0.15
Критерий О	0.130	0.305	0.415	0.500
Критерий П	0.130	0.285	0.385	0.485

Несмотря на невысокую мощность критериев О и П относительно гипотезы H_1 , они показывают довольно высокую мощность относительно гипотез H_2 и H_3 . Также стоит отметить, что критерии Фросини и Андерсона - Дарлинга являются смещенными при объеме выборки 20, критерии О и П лишены этого недостатка.

4. Реализация выполнения задания №3

Пусть имеется выборка X_1, X_2, \dots, X_n из показательного распределения с параметром θ . Тогда в силу независимости функция максимального правдоподобия будет произведением функций распределения:

$$f(x_1, \dots, x_n; \theta) = \theta e^{-\theta x_1} \cdot \theta e^{-\theta x_2} \cdot \dots \cdot \theta e^{-\theta x_n} = \theta^n e^{-\theta(x_1 + \dots + x_n)}.$$

Ее логарифм равен

$$\ln f = n \ln \theta - \theta(x_1 + \dots + x_n).$$

Дифференцируя и приравнявая нулю, находим оценку параметра θ :

$$\frac{\partial f}{\partial \theta} = \frac{n}{\theta} - \sum_{i=1}^n x_i = 0,$$

$$\hat{\theta} = \frac{1}{n} \sum_{i=1}^n X_i.$$

В силу центральной предельной теоремы

$$\lim_{n \rightarrow \infty} P \left(\frac{X_1 + \dots + X_n - M(X_1 + \dots + X_n)}{\sqrt{D(X_1 + \dots + X_n)}} < x \right) = \Phi(x),$$

где $\Phi(x)$ – функция стандартного нормального распределения. Так как $M(X_i) = \frac{1}{\theta}$ и $D(X_i) = \frac{1}{\theta^2}$,

$$\lim_{n \rightarrow \infty} P \left(\frac{X_1 + \dots + X_n - \frac{n}{\hat{\theta}}}{\frac{\sqrt{n}}{\hat{\theta}}} < x \right) = \lim_{n \rightarrow \infty} P \left(\frac{\hat{\theta} - \frac{1}{\theta}}{\frac{1}{\theta\sqrt{n}}} < x \right) = \Phi(x),$$

$$\lim_{n \rightarrow \infty} P(\hat{\theta} < z) = \Phi \left(\frac{z - \frac{1}{\theta}}{\frac{1}{\theta\sqrt{n}}} \right) = \Phi \left(\theta\sqrt{n} \left(z - \frac{1}{\theta} \right) \right) = \frac{1}{2} - \Phi_0 \left(\theta\sqrt{n} \left(z - \frac{1}{\theta} \right) \right).$$

То есть асимптотически $\hat{\theta} \sim N \left(\theta, \frac{\theta^2}{n} \right)$. При выполнении гипотезы $H_0 = \theta = u$ $\hat{\theta} \sim N \left(u, \frac{u^2}{n} \right)$.

$$P \left(-p < \frac{\hat{\theta} - \frac{1}{u}}{\frac{1}{u\sqrt{n}}} < p \right) = P \left(-p \cdot \frac{1}{u\sqrt{n}} + \frac{1}{u} < \hat{\theta} < p \cdot \frac{1}{u\sqrt{n}} + \frac{1}{u} \right) \approx \Phi(p) - \Phi(-p) = 2\Phi_0(p).$$

Таким образом, гипотеза H_0 принимается на уровне значимости $\alpha = 2\Phi_0(c)$, если выполнено условие

$$-p \cdot \frac{1}{u\sqrt{n}} + \frac{1}{u} < \hat{\theta} < p \cdot \frac{1}{u\sqrt{n}} + \frac{1}{u},$$

где значение p определяется из условия $\Phi_0(p) = \frac{\alpha}{2}$, Φ_0 – функция Лапласа.

Пусть $u=0.418$ при $\alpha=0.05$ $p=0.065$

$$-p \cdot \frac{1}{u\sqrt{100}} + \frac{1}{u} = 2.376 < \hat{\theta} < 2.408 = p \cdot \frac{1}{u\sqrt{100}} + \frac{1}{u}$$

при $\alpha=0.01$ $p=0.015$

$$-p \cdot \frac{1}{u\sqrt{100}} + \frac{1}{u} = 2.389 < \hat{\theta} < 2.396 = p \cdot \frac{1}{u\sqrt{100}} + \frac{1}{u}$$

Пусть $k=3.18$ при $\alpha=0.05$ $p=0.065$:

$$-p \cdot \frac{1}{k\sqrt{100}} + \frac{1}{k} = 0.312 < \hat{\theta} < 0.316 = p \cdot \frac{1}{k\sqrt{100}} + \frac{1}{k}$$

При $\alpha=0.01$ $p=0.015$

$$-p \cdot \frac{1}{k\sqrt{100}} + \frac{1}{k} = 0.314 < \hat{\theta} < 0.315 = p \cdot \frac{1}{k\sqrt{100}} + \frac{1}{k}$$

Пусть имеется выборка X_1, X_2, \dots, X_n из нормального распределения с параметрами θ_1 и θ_2 . Тогда в силу независимости функция максимального правдоподобия будет произведением функций распределения:

$$\begin{aligned}
f(x_1, x_2, \dots, x_n; \theta_1, \theta_2) &= \frac{1}{\sqrt{2\pi\theta_2^2}} e^{-(x_1-\theta_1)^2/2\theta_2^2} \cdot \dots \cdot \frac{1}{\sqrt{2\pi\theta_2^2}} e^{-(x_n-\theta_1)^2/2\theta_2^2} = \\
&= \frac{1}{\sqrt{(2\pi\theta_2^2)^n}} e^{-((x_1-\theta_1)^2 + \dots + (x_n-\theta_1)^2)/2\theta_2^2} = \frac{1}{(2\pi\theta_2^2)^{n/2}} e^{(-\sum(x_i-\bar{x})^2 - n(\bar{x}-\theta_1)^2)/2\theta_2^2} = \\
&= \frac{1}{(2\pi e s^2)^{n/2}} e^{n \ln s} \cdot e^{n/2} \cdot e^{-n \ln \theta_2} \cdot e^{(-n s^2 - n(\bar{x}-\theta_1)^2)/2\theta_2^2} = \\
&= \frac{1}{(2\pi e s^2)^{n/2}} e^{-n((\bar{x}-\theta_1)^2/2\theta_2^2 + 1/2(s^2/\theta_2^2 - 1) - \ln s/\theta_2)}
\end{aligned}$$

где $s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$.

$$\begin{aligned}
\widehat{\theta}_1 &= \bar{x} = \frac{1}{n} \sum x_i, \\
\widehat{\theta}_2 &= s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2.
\end{aligned}$$

Аналогично предыдущему

$$\lim_{n \rightarrow \infty} P\left(\frac{X_1 + \dots + X_n - M(X_1 + \dots + X_n)}{\sqrt{D(X_1 + \dots + X_n)}} < x\right) = \lim_{n \rightarrow \infty} P\left(\frac{\widehat{\theta}_1 - \mu}{\sqrt{\sigma^2/n}} < x\right) = \Phi(x),$$

Следовательно, $\widehat{\theta}_1 \sim N(\mu, \sigma^2/n)$. Подставим в функцию правдоподобия найденную оценку:

$$f(\bar{x}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{(-\sum(x_i-\bar{x})^2 - n(\bar{x}-\theta_1)^2)/2\sigma^2}.$$

И рассмотрим отношение

$$l(x) = \frac{f(\theta_0, \sigma^2)}{f(\theta_1, \sigma^2)} = e^{(-n(\bar{x}-\theta_0)^2 + n(\bar{x}-\theta_1)^2)/2\sigma^2} \geq c$$

для гипотезы $H_0 = \{\mu = \theta_0\}$. Оно эквивалентно следующему:

$$\bar{x} \geq \frac{\sigma^2 \ln c}{n(\theta_1 - \theta_0)} + \frac{\theta_1 + \theta_0}{2},$$

Которое удобно переписать в виде

$$\frac{\sqrt{n}}{\sigma} (\bar{x} - \theta_0) \geq \frac{\sigma}{\sqrt{n}(\theta_1 - \theta_0)} \ln c + \frac{\sqrt{n}}{2\sigma} (\theta_1 - \theta_0).$$

Обозначив правую часть $\gamma(c)$ и учитывая, что $\widehat{\theta}_1 \sim N(\theta_0, \sigma^2/n)$, имеем

$$P\{l(X) \geq c\} = P\left\{\frac{\sqrt{n}}{\sigma} (\bar{x} - \theta_0) \geq \gamma(c)\right\} = \Phi(-\gamma(c)).$$

Это односторонний критерий в предположении, что $\theta_1 > \theta_0$. Обозначим через γ_α число, такое что $\Phi(\gamma_\alpha) = \alpha/2$. Тогда для двустороннего критерия имеем

$$-\frac{\sigma}{\sqrt{n}} \gamma_\alpha + \theta_0 < \bar{x} < \frac{\sigma}{\sqrt{n}} \gamma_\alpha + \theta_0.$$

Если же дисперсия неизвестна, то можно применить отношение правдоподобия. Подставим в функцию правдоподобия найденные оценки:

$$f(\bar{x}, s) = \frac{1}{\left(2\pi e \sum (x_i - \bar{x})^2 / n\right)^{n/2}}.$$

И рассмотрим отношение

$$\frac{f(\mu_0, s)}{f(\bar{x}, s)} = \frac{\left(2\pi e \sum (x_i - \bar{x})^2 / n\right)^{n/2}}{\left(2\pi e \sum (x_i - \mu_0)^2 / n\right)^{n/2}} \leq k_1.$$

После возведения в степень получим

$$\frac{n(\bar{x} - \mu_0)^2}{\sum (x_i - \bar{x})^2} \geq k_2,$$

Деля на $n - 1$ и извлекая корень, имеем

$$\frac{\bar{x} - \mu_0}{s/\sqrt{n}} \geq c \quad \text{или} \quad \frac{\bar{x} - \mu_0}{s/\sqrt{n}} \leq -c.$$

Мы получили распределение Стьюдента, поэтому положим $c = t_{\alpha/2, n-1}$. Окончательно

$$\mu_0 - \frac{s}{\sqrt{n}} t_{\alpha/2, n-1} \leq \widehat{\theta}_1 \leq \mu_0 + \frac{s}{\sqrt{n}} t_{\alpha/2, n-1}.$$

Построим критерий для дисперсии

$$l(x) = \frac{f(\bar{x}, \sigma_0)}{f(\bar{x}, s)} = \frac{(2\pi e s^2)^{n/2}}{(2\pi \sigma_0^2)^{n/2} e^{\left\{\frac{ns^2}{2\sigma_0^2}\right\}}} = \left(\frac{s^2}{\sigma_0^2} e^{\left\{1 - \frac{ns^2}{2\sigma_0^2}\right\}}\right)^{\frac{n}{2}} = (t e^{1-t})^{\frac{n}{2}} \geq c,$$

где $t = \frac{s^2}{\sigma_0^2}$. Критерий имеет вид $t_1 \leq \frac{s^2}{\sigma_0^2} \leq t_2$. Учитывая, что $ns^2/\sigma_0^2 \sim \chi^2_{n-1}$, имеем

$$t_1 = \chi^2_{\alpha/2, n-1}/n, \quad t_2 = \chi^2_{1-\alpha/2, n-1}/n,$$

то есть

$$\frac{\sigma_0^2}{n} \cdot \chi^2_{\alpha/2, n-1} \leq s^2 \leq \frac{\sigma_0^2}{n} \cdot \chi^2_{1-\alpha/2, n-1}.$$

5. Ответы на контрольные вопросы

1. Что такое критерий согласия? Где используется?

Критерий согласия – это критерии проверки гипотез о соответствии эмпирического распределения к теоретическому распределению вероятностей.

2. Какими критериями можно проверить утверждение гипотезы?

- Принятие решения о том, противоречит ли рассматриваемая статистическая гипотеза наблюдаемой выборке данных
- Формулируется нулевая гипотеза H_0 о распределении вероятностей на множестве X . Гипотеза формулируется исходя из требований прикладной задачи.
- Задаётся некоторая статистика (функция выборки)
- Фиксируется уровень значимости — допустимая для данной задачи вероятность ошибки первого рода, то есть того, что гипотеза на самом деле верна, но будет отвергнута процедурой проверки.

3. Чем отличается критерий Колмогорова-Смирнова от критерия Колмогорова?
- a) Критерий согласия Колмогорова предназначен для проверки гипотезы о принадлежности выборки некоторому закону распределения, то есть проверки того, что эмпирическое распределение соответствует предполагаемой модели.
 - b) Критерий Колмогорова-Смирнова позволяет оценить существенность различий между двумя выборками, в том числе возможно его применение для сравнения эмпирического распределения с теоретическим.
4. Для чего используется информация Фишера?
- В математической статистике и теории информации информацией Фишера называется дисперсия. Наиболее часто используемой характеристикой, на основании которой измеряют расстояние между распределениями. Функция вклада выборки.
5. Для чего используется критерий отношения правдоподобия? Где и как он применяется?
- Это метод оценивания неизвестного параметра путём максимизации функции правдоподобия. Основан на предположении о том, что вся информация о статистической выборке содержится в функции правдоподобия. Метод оценки максимального правдоподобия применяется для широкого круга статистических моделей.

6. Выводы по расчетному заданию

1. Для проверки гипотезы о равномерном распределении чисел 0-9, полученных табличным методом, использовался критерий согласия хи-квадрат с девятью степенями свободы. Данный критерий является правосторонним. Квантили заданных уровней значимости определялись из таблицы хи-квадрат. Полученное значение критерия оказалось вне критической области. А это значит, что полученная последовательность является псевдослучайной.
2. В силу того, что не удалось найти таблицу всех уровней значимости, указанных в задании, она была построена численным моделированием в python. Полученные процентные точки практически одинаковы с теми, для которых удалось найти значения в таблице. Так как критерии О и П нужно было рассчитать для 8 уровней значимости, то я посчитал логичным взять большее число выборок, чем указано в задании, чтобы улучшить точность результатов. После нахождения уровней значимости критериев Фросини и Андерсона-Дарлинга по интерполяционному многочлену, они были получены численным моделированием. Численное моделирование показало, что процентные точки, полученные по интерполяционному многочлену близки к требуемым значениям 0,01, 0,05, 0,1 и 0,15 (ср. таблицы 1 и 3). В качестве конкурирующих гипотез были выбраны бета-распределения 1-ого рода с параметрами (1,5;1,5;1;0) (0,8;1;1;0) (1,1;0,9;1;0), которые близки к равномерному распределению. Несмотря на невысокую мощность критериев О и П относительно гипотезы H_1 , они показали довольно высокую мощность относительно гипотез H_2 и H_3 . Также стоит отметить, что критерии Фросини и Андерсона - Дарлинга являются смещенными при объеме выборки 20, критерии О и П лишены этого недостатка.
3. Из условия минимума функции правдоподобия были получены оценки параметров распределений. Из ЦПТ для них были получены законы распределения. Затем были построены отношения правдоподобия (отношения функций правдоподобия, соответствующих проверяемой гипотезе и альтернативной), из которых были найдены границы критериев принятия гипотез на заданных уровнях значимости.

Полученные листинги:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include<math.h>
#include<ctype.h>
#include<string.h>
#include<stdlib.h>
//const short N = 100;
#define N 100
float func_statistics(float*);

void qsortx(int *a, int low, int high) {

    int i = 0, j = 0;
    int tmp = 0, medianofthree = 0;

    i = low;
    j = high;
    int k = 0, l = 0, m = 0;
    k = *(a + 0);
    l = *(a + high);
    m = *(a + (high + low) / 2);
    /*Median-of-three*/
    if (k > l && k < m || k < l && k > m)
        medianofthree = k;
    else
        if (l > k && l < m || l < k && l > m)
            medianofthree = l;
        else
            medianofthree = m;
    /*Median-of-three*/
    //printf("\n%d\n", medianofthree);
    do {
        while (*(a + i) < medianofthree)
        {
            i++;
        }
        while (*(a + j) > medianofthree)
        {
            j--;
        }
        if (i <= j)
        {
            if (*(a + i) > *(a + j))
            {
                tmp = *(a + i);
                *(a + i) = *(a + j);
                *(a + j) = tmp;
            }
            i++;
            if (j > 0)
            {
                j--;
            }
        }
    } while (i <= j);

    if (i < high)
    {
        qsortx(a, i, high);
    }
    if (j > low)
    {
        qsortx(a, low, j);
    }
}
```

```

int main()
{
    int f_x[N];

    float second_znak = 0;
    float x = 0;
    float t = 0;
    char a[20] = { 0 };
    int c;

    char myString[30];

    for (int i = 0; i < N; i++)
    {
        x = x + 0.99; //x
        /*if (i < 50)
            printf("%f\n", x);

        if (i >= 50)
            printf("%f\n", x);*/
        memset(myString, 0, 30);
        t = tan(2.0 * x); //f(x)
        if (i < 50)
            printf("%.8f\n", t);

        if (i >= 50)
            printf("%.8f\n", t);
        sprintf(myString, "%f", t); //f(x)] обрезанный тангенс
        //число второе с конца после обрезания

        f_x[i] = atoi(&myString[strlen(myString) - 2]) / 10;

        /* if (i < 50)

printf("%d\n", f_x[i]);

if (i >= 50)

printf("%d\n", f_x[i]);*/
    }
    for (int i = 0; i < N; i++)
        printf("%d ", f_x[i]);

    float Nu_Num[10];
    qsortx(f_x, 0, N - 1);

    int j = 0;
    short amount = 0;
    int k = 0;
    for (int k = 0; k < 10; k++)
    {
        while (f_x[j] == k)
        {
            amount++;
            j++;
        }
        Nu_Num[k] = amount; // / 100.0;
        amount = 0;
        //j++;
    }
    printf("\n");
    for (int i = 0; i < 10; i++)
        printf("%f ", Nu_Num[i]);

    float Xi = func_statistics(Nu_Num);

```

```

printf("\n%f", Xi);

system("pause");

return 0;

}

float func_statistics(float *Nu_Num)
{
    int i = 0;
    float res_cycle = 0;
    for (; i <= 9; i++)
    {
        res_cycle = res_cycle + (*(Nu_Num + i) - 0.1*N)*(*(Nu_Num + i) - 0.1*N);
    }
    return (10.0 * res_cycle / N);
    //sqrt not sure if i need this
}

```

```

import numpy as np
import matplotlib.pyplot as plt
from statsmodels.distributions.empirical_distribution import ECDF
import math as m
from bisect import bisect
from scipy.interpolate import CubicSpline
from scipy.stats import betaprime

```

```

def fros(alpha,n,sim):
    x1=np.random.rand(sim,n)
    s1=[]
    B1=[]
    #print(x1)
    for i in range(0,sim):
        x1[i,:].sort()
        #print(x1[i])
        s1.append([])
        for k in range(0,n):
            s1[i].append(abs(x1[i,k]-(k+0.5)/n))
        #print(s1[i])
        B1.append(pow(n,-0.5)*sum(s1[i]))
    return np.percentile(B1, alpha)

def andersond(alpha,n,sim):
    x1=np.random.rand(sim,n)
    s1=[]
    B1=[]
    #print(x1)
    for i in range(0,sim):
        x1[i,:].sort()
        #print(x1[i])
        s1.append([])
        for k in range(0,n):

```

```

        s1[i].append(((2*(k+1)-1)*m.log(x1[i,k],m.e)/(2*n)+(1-(2*k+1)/(2*n))*m.log(1-
x1[i,k],m.e))
        #print(s1[i])
        B1.append(-n-2*sum(s1[i]))
    return np.percentile(B1, alpha)
N2=4000
f005=fros(99.5,20,N2)
f01=fros(99,20,N2)
f025=fros(97.5,20,N2)
f05=fros(95,20,N2)
f1=fros(90,20,N2)
f15=fros(85,20,N2)
f2=fros(80,20,N2)
f3=fros(70,20,N2)
f4=fros(60,20,N2)
a005=andersond(99.5,20,N2)
a01=andersond(99,20,N2)
a025=andersond(97.5,20,N2)
a05=andersond(95,20,N2)
a1=andersond(90,20,N2)
a15=andersond(85,20,N2)
a2=andersond(80,20,N2)
a3=andersond(70,20,N2)
a4=andersond(60,20,N2)
print('          Table of the values of criterias ')
print('\nalpha  0.005  0.01  0.025  0.05  0.1   0.15  0.2  0.3  0.4\n')
print('Frosini % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f'%(f005, f01,
f025, f05, f1, f15, f2, f3, f4))
print('Frosini      (0.7428)      (0.5785) (0.4966) (0.4456) (0.4083)')
print('And-Dar % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f % 1.3f'%(a005,
a01, a025, a05, a1, a15, a2, a3, a4))
print('And-Dar      (3.8781) (3.0775) (2.4924) (1.9330) (1.6212)')
#ecdf = ECDF(fros(20,1000))
#plt.step(ecdf.x, ecdf.y)
def frosEst(x):
    nl=len(x)
    nr=len(x[0])
    u=np.array(x)
    s=[]
    B=[]
    for i in range(0,nl):
        u[i,:].sort()
        s.append([])
        for k in range(0,nr):
            s[i].append(abs(u[i,k]-(k+0.5)/nr))

```

```

        B.append(pow(nr,-0.5)*sum(s[i]))
    return B
def andersondEst(x):
    nl=len(x)
    nr=len(x[0])
    u=np.array(x)
    s=[]
    B=[]
    for i in range(0,nl):
        u[i,:].sort()
        s.append([])
        for k in range(0,nr):
            s[i].append((2*(k+1)-1)*m.log(u[i,k],m.e)/(2*nr)+(1-(2*k+1)/(2*nr))*m.log(1-
u[i,k],m.e))
        B.append(-nr-2*sum(s[i]))
    return B
N1=4000
x=np.random.rand(N1,20)
fx=frosEst(x)
ax=andersondEst(x)
Cr005=0
CrP005=0
Cr01=0
CrP01=0
Cr025=0
CrP025=0
Cr05=0
CrP05=0
Cr1=0
CrP1=0
Cr15=0
CrP15=0
Cr2=0
CrP2=0
Cr3=0
CrP3=0
Cr4=0
CrP4=0
for i in range(0,len(fx)):
    if (fx[i]<f005) or (ax[i]<a005):
        Cr005=Cr005+1
    if (fx[i]<f005) and (ax[i]<a005):
        CrP005=CrP005+1
    if (fx[i]<f01) or (ax[i]<a01):
        Cr01=Cr01+1

```

```

if (fx[i]<f01) and (ax[i]<a01):
    CrP01=CrP01+1
if (fx[i]<f025) or (ax[i]<a025):
    Cr025=Cr025+1
if (fx[i]<f025) and (ax[i]<a025):
    CrP025=CrP025+1
if (fx[i]<f05) or (ax[i]<a05):
    Cr05=Cr05+1
if (fx[i]<f05) and (ax[i]<a05):
    CrP05=CrP05+1
if (fx[i]<f1) or (ax[i]<a1):
    Cr1=Cr1+1
if (fx[i]<f1) and (ax[i]<a1):
    CrP1=CrP1+1
if (fx[i]<f15) or (ax[i]<a15):
    Cr15=Cr15+1
if (fx[i]<f15) and (ax[i]<a15):
    CrP15=CrP15+1
if (fx[i]<f2) or (ax[i]<a2):
    Cr2=Cr2+1
if (fx[i]<f2) and (ax[i]<a2):
    CrP2=CrP2+1
if (fx[i]<f3) or (ax[i]<a3):
    Cr3=Cr3+1
if (fx[i]<f3) and (ax[i]<a3):
    CrP3=CrP3+1
if (fx[i]<f4) or (ax[i]<a4):
    Cr4=Cr4+1
if (fx[i]<f4) and (ax[i]<a4):
    CrP4=CrP4+1
Cr005=1-Cr005/N1
Cr01=1-Cr01/N1
Cr025=1-Cr025/N1
Cr05=1-Cr05/N1
Cr1=1-Cr1/N1
Cr15=1-Cr15/N1
Cr2=1-Cr2/N1
Cr3=1-Cr3/N1
Cr4=1-Cr4/N1
CrP005=1-CrP005/N1
CrP01=1-CrP01/N1
CrP025=1-CrP025/N1
CrP05=1-CrP05/N1
CrP1=1-CrP1/N1
CrP15=1-CrP15/N1

```

```

CrP2=1-CrP2/N1
CrP3=1-CrP3/N1
CrP4=1-CrP4/N1
print('\n          Table of significance levels')
print('\n\nalpha  0.005  0.01  0.025  0.05  0.1  0.15  0.2  0.3  0.4\n')
print('alpha O  %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f%(Cr005, Cr01,
Cr025, Cr05, Cr1, Cr15, Cr2, Cr3, Cr4))
print('alpha P  %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f %1.3f%(CrP005,
CrP01, CrP025, CrP05, CrP1, CrP15, CrP2, CrP3, CrP4))
Alpha=[0.005, 0.01, 0.025, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4]
CrArr=[Cr005, Cr01, Cr025, Cr05, Cr1, Cr15, Cr2, Cr3, Cr4]
cs = CubicSpline(CrArr, Alpha)
xs = np.arange(Cr005, Cr4, 0.05)
fig, ax = plt.subplots(figsize=(6.5, 4))
ax.plot(CrArr, Alpha, 'o', label='data')
ax.plot(xs, cs(xs), label="S")
ax.set_xlim(0, 0.5)
ax.legend(loc='upper left', ncol=2)
plt.show()
CrPArr=[CrP005, CrP01, CrP025, CrP05, CrP1, CrP15, CrP2, CrP3, CrP4]
csP = CubicSpline(CrPArr, Alpha)
xsP = np.arange(CrP005, CrP4, 0.05)
fig, ax = plt.subplots(figsize=(6.5, 4))
ax.plot(CrPArr, Alpha, 'o', label='data')
ax.plot(xsP, csP(xsP), label="S")
ax.set_xlim(0, 0.5)
ax.legend(loc='upper left', ncol=2)
plt.show()
AlphaO=cs([0.01, 0.05, 0.1, 0.15])
AlphaP=csP([0.01, 0.05, 0.1, 0.15])
print('alpha_O/P [0.01, 0.05, 0.1, 0.15]')
print('f(alpha_O)', AlphaO)
print('f(alpha_P)', AlphaP)
N2=4000
fAO1=fros(100*(1-AlphaO[0]),20,N2)
fAO2=fros(100*(1-AlphaO[1]),20,N2)
fAO3=fros(100*(1-AlphaO[2]),20,N2)
fAO4=fros(100*(1-AlphaO[3]),20,N2)
fAP1=fros(100*(1-AlphaP[0]),20,N2)
fAP2=fros(100*(1-AlphaP[1]),20,N2)
fAP3=fros(100*(1-AlphaP[2]),20,N2)
fAP4=fros(100*(1-AlphaP[3]),20,N2)
aAP1=andersond(100*(1-AlphaP[0]),20,N2)
aAP2=andersond(100*(1-AlphaP[1]),20,N2)
aAP3=andersond(100*(1-AlphaP[2]),20,N2)

```



```

aAP4=andersond(100*(1-AlphaP[3]),20,N2)
aAO1=andersond(100*(1-AlphaO[0]),20,N2)
aAO2=andersond(100*(1-AlphaO[1]),20,N2)
aAO3=andersond(100*(1-AlphaO[2]),20,N2)
aAO4=andersond(100*(1-AlphaO[3]),20,N2)
print('          Table of the values of criterias ')
print('\nalpha      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f
% 1.3f\n'%(AlphaO[0], AlphaO[1], AlphaO[2], AlphaO[3], AlphaP[0], AlphaP[1], AlphaP[2],
AlphaP[3]))
print('Frosini      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f
% 1.3f'%(fAO1,fAO2,fAO3,fAO4,fAP1,fAP2,fAP3,fAP4))
print('And-Dar      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f      % 1.3f
% 1.3f'%(aAO1,aAO2,aAO3,aAO4,aAP1,aAP2,aAP3,aAP4))
N1=4000
x=np.random.rand(N1,20)
fx=frosEst(x)
ax=andersondEst(x)
CrAO1=0
CrPO1=0
CrAO2=0
CrPO2=0
CrAO3=0
CrPO3=0
CrAO4=0
CrPO4=0
for i in range(0,len(fx)):
    if (fx[i]<fAO1) or (ax[i]<aAO1):
        CrAO1=CrAO1+1
    if (fx[i]<fAP1) and (ax[i]<aAP1):
        CrPO1=CrPO1+1
    if (fx[i]<fAO2) or (ax[i]<aAO2):
        CrAO2=CrAO2+1
    if (fx[i]<fAP2) and (ax[i]<aAP2):
        CrPO2=CrPO2+1
    if (fx[i]<fAO3) or (ax[i]<aAO3):
        CrAO3=CrAO3+1
    if (fx[i]<fAP3) and (ax[i]<aAP3):
        CrPO3=CrPO3+1
    if (fx[i]<fAO4) or (ax[i]<aAO4):
        CrAO4=CrAO4+1
    if (fx[i]<fAP4) and (ax[i]<aAP4):
        CrPO4=CrPO4+1
CrAO1=1-CrAO1/N1
CrAO2=1-CrAO2/N1
CrAO3=1-CrAO3/N1

```

```

CrAO4=1-CrAO4/N1
CrPO1=1-CrPO1/N1
CrPO2=1-CrPO2/N1
CrPO3=1-CrPO3/N1
CrPO4=1-CrPO4/N1
print('\n          Table of significance levels')
#print('\n\nalpha    0.01    0.05    0.1    0.15 \n')
print('Crit O    %1.3f %1.3f %1.3f %1.3f'%(CrAO1, CrAO2, CrAO3, CrAO4))
print('Crit P    %1.3f %1.3f %1.3f %1.3f'%(CrPO1, CrPO2, CrPO3, CrPO4))
ah1=1.5
bh1=1.5
ch1=1.0
dh1=0.0
N3=200
k=0
sampleh1=[]
for i in range(0,N3):
    sample1 = betaprime.rvs(ah1, bh1, loc=dh1, scale=ch1, size=500)
    sample2=[]
    k=0
    j=0
    while (k<20):
        if sample1[j]<=1:
            sample2.append(sample1[j])
            k=k+1
        j=j+1
    sampleh1.append(sample2)
fx=frosEst(sampleh1)
ax=andersondEst(sampleh1)
CrAO1=0
CrPO1=0
CrAO2=0
CrPO2=0
CrAO3=0
CrPO3=0
CrAO4=0
CrPO4=0
for i in range(0,len(fx)):
    if (fx[i]<fAO1) or (ax[i]<aAO1):
        CrAO1=CrAO1+1
    if (fx[i]<fAP1) and (ax[i]<aAP1):
        CrPO1=CrPO1+1
    if (fx[i]<fAO2) or (ax[i]<aAO2):
        CrAO2=CrAO2+1
    if (fx[i]<fAP2) and (ax[i]<aAP2):

```

```

    CrPO2=CrPO2+1
    if (fx[i]<fAO3) or (ax[i]<aAO3):
        CrAO3=CrAO3+1
    if (fx[i]<fAP3) and (ax[i]<aAP3):
        CrPO3=CrPO3+1
    if (fx[i]<fAO4) or (ax[i]<aAO4):
        CrAO4=CrAO4+1
    if (fx[i]<fAP4) and (ax[i]<aAP4):
        CrPO4=CrPO4+1
CrAO1=1-CrAO1/N3
CrAO2=1-CrAO2/N3
CrAO3=1-CrAO3/N3
CrAO4=1-CrAO4/N3
CrPO1=1-CrPO1/N3
CrPO2=1-CrPO2/N3
CrPO3=1-CrPO3/N3
CrPO4=1-CrPO4/N3
print("\n          Table of capacities for H1')
#print("\n\nalpha   0.01   0.05   0.1   0.15 \n')
print('alpha_O/P [0.01, 0.05, 0.1, 0.15]')
print('Crit O   %1.3f %1.3f %1.3f %1.3f'%(CrAO1, CrAO2, CrAO3, CrAO4))
print('Crit P   %1.3f %1.3f %1.3f %1.3f'%(CrPO1, CrPO2, CrPO3, CrPO4))
ah1=0.8
bh1=1.0
ch1=1.0
dh1=0.0
N3=200
k=0
sampleh2=[]
for i in range(0,N3):
    sample1 = betaprime.rvs(ah1, bh1, loc=dh1, scale=ch1, size=500)
    sample2=[]
    k=0
    j=0
    while (k<20):
        if sample1[j]<=1:
            sample2.append(sample1[j])
            k=k+1
        j=j+1
    sampleh2.append(sample2)
fx=frosEst(sampleh2)
ax=andersondEst(sampleh2)
CrAO1=0
CrPO1=0
CrAO2=0

```

```

CrPO2=0
CrAO3=0
CrPO3=0
CrAO4=0
CrPO4=0
for i in range(0,len(fx)):
    if (fx[i]<fAO1) or (ax[i]<aAO1):
        CrAO1=CrAO1+1
    if (fx[i]<fAP1) and (ax[i]<aAP1):
        CrPO1=CrPO1+1
    if (fx[i]<fAO2) or (ax[i]<aAO2):
        CrAO2=CrAO2+1
    if (fx[i]<fAP2) and (ax[i]<aAP2):
        CrPO2=CrPO2+1
    if (fx[i]<fAO3) or (ax[i]<aAO3):
        CrAO3=CrAO3+1
    if (fx[i]<fAP3) and (ax[i]<aAP3):
        CrPO3=CrPO3+1
    if (fx[i]<fAO4) or (ax[i]<aAO4):
        CrAO4=CrAO4+1
    if (fx[i]<fAP4) and (ax[i]<aAP4):
        CrPO4=CrPO4+1
CrAO1=1-CrAO1/N3
CrAO2=1-CrAO2/N3
CrAO3=1-CrAO3/N3
CrAO4=1-CrAO4/N3
CrPO1=1-CrPO1/N3
CrPO2=1-CrPO2/N3
CrPO3=1-CrPO3/N3
CrPO4=1-CrPO4/N3
print('\n          Table of capacities for H2')
#print('\n\nalpha   0.01  0.05  0.1  0.15 \n')
print('alpha_O/P [0.01, 0.05, 0.1, 0.15]')
print('Crit O   %1.3f %1.3f %1.3f %1.3f'%(CrAO1, CrAO2, CrAO3, CrAO4))
print('Crit P   %1.3f %1.3f %1.3f %1.3f'%(CrPO1, CrPO2, CrPO3, CrPO4))
ah1=1.1
bh1=0.9
ch1=1.0
dh1=0.0
N3=200
k=0
sampleh3=[]
for i in range(0,N3):
    sample1 = betaprime.rvs(ah1, bh1, loc=dh1, scale=ch1, size=500)
    sample2=[]

```

```

k=0
j=0
while (k<20):
    if sample1[j]<=1:
        sample2.append(sample1[j])
        k=k+1
    j=j+1
sampleh3.append(sample2)
fx=frosEst(sampleh3)
ax=andersondEst(sampleh3)
CrAO1=0
CrPO1=0
CrAO2=0
CrPO2=0
CrAO3=0
CrPO3=0
CrAO4=0
CrPO4=0
for i in range(0,len(fx)):
    if (fx[i]<fAO1) or (ax[i]<aAO1):
        CrAO1=CrAO1+1
    if (fx[i]<fAP1) and (ax[i]<aAP1):
        CrPO1=CrPO1+1
    if (fx[i]<fAO2) or (ax[i]<aAO2):
        CrAO2=CrAO2+1
    if (fx[i]<fAP2) and (ax[i]<aAP2):
        CrPO2=CrPO2+1
    if (fx[i]<fAO3) or (ax[i]<aAO3):
        CrAO3=CrAO3+1
    if (fx[i]<fAP3) and (ax[i]<aAP3):
        CrPO3=CrPO3+1
    if (fx[i]<fAO4) or (ax[i]<aAO4):
        CrAO4=CrAO4+1
    if (fx[i]<fAP4) and (ax[i]<aAP4):
        CrPO4=CrPO4+1
CrAO1=1-CrAO1/N3
CrAO2=1-CrAO2/N3
CrAO3=1-CrAO3/N3
CrAO4=1-CrAO4/N3
CrPO1=1-CrPO1/N3
CrPO2=1-CrPO2/N3
CrPO3=1-CrPO3/N3
CrPO4=1-CrPO4/N3
print("\n          Table of capacities for H3')
#print("\n\nalpha   0.01   0.05   0.1   0.15 \n')

```

```
print('alpha_O/P [0.01, 0.05, 0.1, 0.15]')  
print('Crit O   %1.3f %1.3f %1.3f %1.3f'%(CrAO1, CrAO2, CrAO3, CrAO4))  
print('Crit P   %1.3f %1.3f %1.3f %1.3f'%(CrPO1, CrPO2, CrPO3, CrPO4))
```