

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Российский государственный университет им. А.Н. Косыгина
(Технологии. Дизайн. Искусство)»

Кафедра автоматизированных систем обработки информации и управления

Отчет по лабораторной работе № 2
по дисциплине «Компьютерное моделирование»

Тема: «Моделирование случайных величин в MATLAB»

Выполнил: Ольховский Н.С., ИТА-123

Проверила: Самойлова Т.А.

Москва 2025

Вариант 13

Условие варианта на рисунке 1.

№	ДСВ		НСВ	
	N	Distr	N	Distr
13	140	Bi(8; 0,7)	570	N(30, 9)

Рис. 1. Условие

1. Сгенерировать N значений дискретной случайной величины с заданным законом распределения.
2. Сгенерировать N значений непрерывной случайной величины с заданным законом распределения.

Программный код частей 1 и 2 отображен на рисунке 2.

```
% 1.  
X_bin = binornd(8, 0.7, [140, 1]);  
  
% 2.  
mu = 30; % математическое ожидание  
sigma2 = 9; % дисперсия  
sigma = sqrt(sigma2); % стандартное отклонение  
X_norm = normrnd(mu, sigma, [570, 1]);
```

Рис. 2. Программный код части 1, 2

3. Для непрерывной случайной величины сделать расчет числовых характеристик: математическое ожидание, медиана, дисперсия, СКО, коэффициент вариации, размах, интердецильный размах, коэффициент асимметрии, коэффициент эксцесса.

Программный код части 3 отображен на рисунке 3. Результат – на рисунке 4.

```

mean_X = mean(X_norm); % математическое ожидание
median_X = median(X_norm); % медиана
var_X = var(X_norm); % дисперсия
std_X = std(X_norm); % стандартное отклонение
cv_X = std_X / mean_X * 100; % коэффициент вариации
range_X = max(X_norm) - min(X_norm); % размах
interdecile_range = quantile(X_norm, 0.9) - quantile(X_norm, 0.1);
% интердецильный размах
skew_X = skewness(X_norm); % коэффициент асимметрии
kurt_X = kurtosis(X_norm) - 3; % коэффициент эксцесса

```

Рис. 3. Программный код части 3

```

Математическое ожидание: 29.7458
Медиана: 29.8162
Дисперсия: 9.2325
СКО: 3.0385
Коэффициент вариации: 0.1021
Размах: 16.2000
Интердецильный размах: 7.9457
Коэффициент асимметрии: -0.0216
Коэффициент эксцесса: -0.3137

```

Рис. 4. Результат выполнения части 3

4. Построить график эмпирической функции распределения непрерывной случайной величины.

Программный код части 4 на рисунке 5. Результат – на рисунке 6.

```

figure;
ecdf(X_norm);
title('Эмпирическая функция распределения');
xlabel('Значение');
ylabel('F(x)');
grid on;

```

Рис. 5. Программный код части 3



Рис. 6. Результат выполнения части 3

5. Определить параметры распределения непрерывной случайной величины по критерию максимального правдоподобия. Проверить по критерию Колмогорова-Смирнова соответствие сгенерированной непрерывной случайной величины заданному распределению (уровень значимости 0.05). Затем вычислить новую гипотетическую ФР для любого другого закона распределения и вновь выполнить проверку по критерию Колмогорова-Смирнова.

Программный код части 5 отображен на рисунке 7. Результат – на рисунке 8.

```
% Оценка наибольшего правдоподобия нормального распределения
phat1 = mle(X_norm, 'distribution', 'normal');
mu_mle = phat1(1); sigma_mle = phat1(2);

% Проверка критерием Колмогорова-Смирнова.
[h, pValue, ksstat] = kstest(X_norm, 'CDF', makedist('Normal', 'mu',
mu_mle, 'sigma', sigma_mle));

if h == 1
    disp('Норм. распр.: Гипотеза отвергнута.');
```

```
else
    disp('Норм. распр.: Гипотеза принята.');
```

```
end

% Проверка логнормального распределения
X_lognorm = lognrnd(mu, sigma, [570, 1]);
phat2 = mle(X_lognorm, 'distribution', 'lognormal');
mu_alt = phat2(1); sigma_alt = phat2(2);

[h_alt, p_ks_alt] = kstest(X_lognorm, 'CDF', makedist('Lognormal',
'mu', mu_alt, 'sigma', sigma_alt));

if h_alt == 1
    disp('ЛогНорм. распр.: Гипотеза отвергнута.');
```

```
else
    disp('ЛогНорм. распр.: Гипотеза принята.');
```

```
end
```

Рис. 7. Программный код части 5

```
Норм. распределение: Гипотеза принята.
ЛогНорм. распределение: Гипотеза принята.
```

Рис. 8. Результат выполнения части 5

6. С помощью функции интерактивного анализа проверить соответствие сгенерированной непрерывной случайной величины заданному распределению. Добавить на гистограмму альтернативный вариант распределения.

Программный код части 6 отображен на рисунке 9. Результат – на рисунке 10.

```
% Параметры для логнормального распределения на основе норм. распределения
param_hat = mle(X_norm, 'distribution', 'lognormal');
mu_logn = param_hat(1); sigma_logn = param_hat(2);

x_vals = linspace(min(X_norm), max(X_norm), 100);

pdf_logn = lognpdf(x_vals, mu_logn, sigma_logn);
pdf_norm = normpdf(x_vals, mean(X_norm), std(X_norm));

% Гистограмма
figure; histogram(X_norm, 'Normalization', 'pdf', 'FaceAlpha', 0.6); hold on;

plot(x_vals, pdf_norm, 'b--', 'LineWidth', 2);
plot(x_vals, pdf_logn, 'r-', 'LineWidth', 2);

legend('Гистограмма данных', 'Логнормальное', 'Нормальное', 'Location', 'best');
title('Нормальное и логнормальное распределения');
xlabel('Значения'); ylabel('Плотность'); hold off;
```

Рис. 9. Программный код части 6

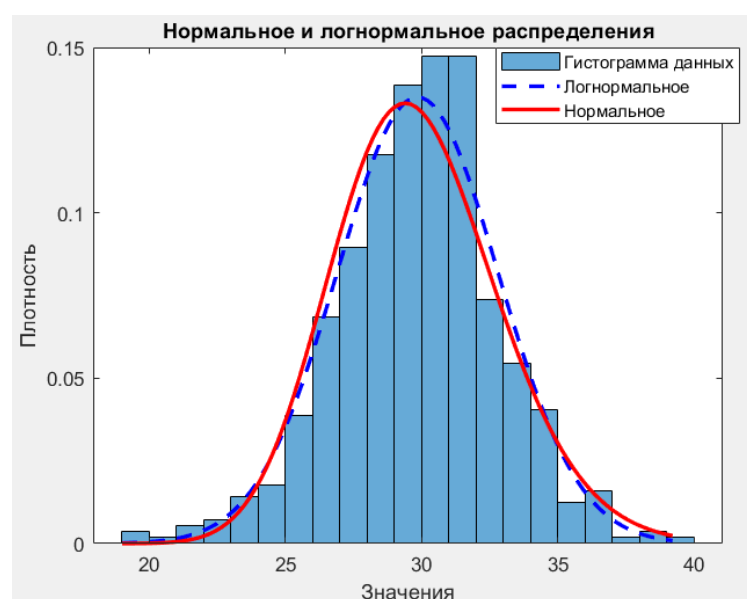


Рис. 10. Результат выполнения части 6

7. Сгенерировать 100, 1 000, 10 000, 100 000 значений непрерывной случайной величины с заданным законом распределения. Построить гистограммы для 7, 17, 117 и 137 классов соответственно.

Программный код части 7 отображен на рисунке 11. Результат – на рисунке 12.

```
N_samples = [100, 1000, 10000, 100000];  
N_bins = [7, 17, 117, 137];  
  
figure;  
for i = 1:length(N_samples)  
    X_large = normrnd(mu, sigma, [N_samples(i), 1]);  
    subplot(2, 2, i);  
    histogram(X_large, N_bins(i));  
    title(sprintf('Знач = %d, Классов = %d', N_samples(i), N_bins(i)));  
    xlabel('Значение'); ylabel('Частота'); grid on;  
end
```

Рис. 11. Программный код части 7

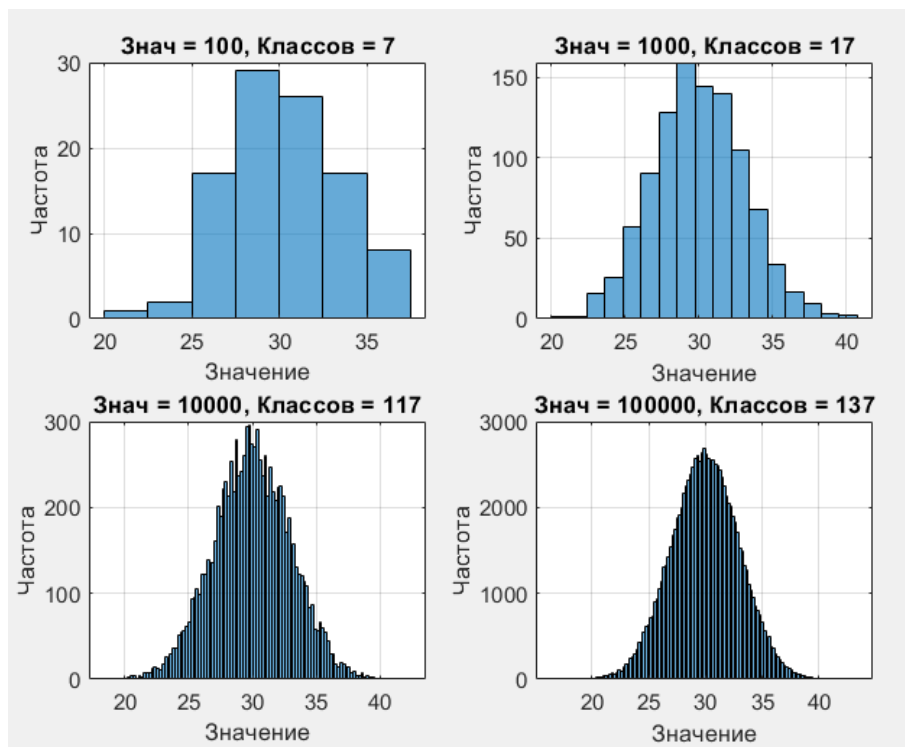


Рис. 12. Результат выполнения части 7