# **概率论与数理统计项目1-产品质量管理**

## **date：2025/4/29 21:30 Writer：王子诺 20235459（Interstellar～兮儿）**

**本项目数据来源为随机生成的正态数据，按照以下项目要求进行编写：**

· 首先收集实际数据或者用软件生成模拟数据，要求25组以上，每组至少5个以上样本；

· 数据描述性统计分析：均值，方差，极差，直方图等等指标；

· 数据正态性检验，以及总体均值检验；

· 数据工序能力指数计算和评估；

· 均值控制图和方差控制图描绘。

· 作结论：工艺水平如何；生产过程是否处于统计受控状态。

## 一、生成数据

生成25组正态分布数据，每组300个样本，使用均值设置为80，方差设置为1（以上参数均可调整）

num\_groups = 25; % 组数

samples\_per\_group = 300; % 每组样本数

mu = 80; % 正态分布均值

sigma = 1; % 正态分布标准差

% 初始化数据存储矩阵

data = zeros(num\_groups, samples\_per\_group);

% 生成正态分布数据

for i = 1:num\_groups

data(i, :) = normrnd(mu, sigma, [1, samples\_per\_group]);

end

二、计算各样本的均值、方差、极差，并绘制条形图、箱线图

% 初始化结果存储

means = zeros(1, size(data, 1)); % 每组均值

variances = zeros(1, size(data, 1)); % 每组方差

ranges = zeros(1, size(data, 1)); % 每组极差

% 计算每组的均值、方差和极差

for i = 1:size(data, 1)

means(i) = mean(data(i, :)); % 均值

variances(i) = var(data(i, :)); % 方差

ranges(i) = max(data(i, :)) - min(data(i, :)); % 极差

end

% 创建表格显示结果

group\_numbers = (1:size(data, 1))'; % 组号

stats\_table = table(group\_numbers, means', variances', ranges', ...

'VariableNames', {'组号', '均值', '方差', '极差'});

% 显示表格

disp('每组的统计结果：');

disp(stats\_table);

% 随机抽取一组

num\_groups = size(data, 1);

random\_group\_idx = randi(num\_groups); % 随机选择一组的索引

selected\_data = data(random\_group\_idx, :); % 提取该组数据

% 绘制直方图

histogram(selected\_data, 'Normalization', 'pdf', 'BinWidth', 0.5);

title(sprintf('组 %d 的直方图', random\_group\_idx));

xlabel('值');

ylabel('概率密度');

grid on;

% 绘制所有组的箱线图（右侧）

boxplot(data', 'Labels', string(1:num\_groups));

title('所有25组数据的箱线图');

xlabel('组号');

ylabel('值');

grid on;

set(gca, 'XTickLabelRotation', 45); % 旋转X轴标签以避免重叠

## 三、正态检验

由于数据本身是正态分布随机生成，先添加噪声再进行检验

以下提供三种噪声供选择性添加

注：若添加高斯噪声，可能对数据分布类型没有很大的改变

% % 添加高斯噪声

% noise\_mu = 0;

% noise\_sigma = 25;

% noise = normrnd(noise\_mu, noise\_sigma, size(data));

% noisy\_data = data + noise;

%添加均匀分布噪声

noise\_min = -10; % 噪声范围下界

noise\_max = 10; % 噪声范围上界

noise = noise\_min + (noise\_max - noise\_min) \* rand(size(data));

noisy\_data = data + noise;

%添加泊松分布噪声

% lambda = 10; % 泊松分布的均值

% noise = poissrnd(lambda, size(data));

% noisy\_data = data + noise;

然后进行K-s检验（非假设参数检验）

% 初始化参数

num\_groups = size(noisy\_data, 1);

test\_mean = mu; % 检验的理论均值（与数据生成过程的 mu 一致）

alpha = 0.05; % 显著性水平

% 初始化结果存储

p\_values\_ttest = zeros(num\_groups, 1); % 均值检验 p 值

h\_values\_ttest = zeros(num\_groups, 1); % 均值检验结果（1=拒绝均值等于 test\_mean，0=接受）

sample\_means = zeros(num\_groups, 1); % 每组样本均值

p\_values\_ks = zeros(num\_groups, 1); % KS 正态性检验 p 值

h\_values\_ks = zeros(num\_groups, 1); % KS 正态性检验结果（1=拒绝正态性，0=接受）

% 对每组进行 KS 正态性检验和单样本 t 检验

for i = 1:num\_groups

% KS 正态性检验

data\_standardized = (noisy\_data(i, :) - mean(noisy\_data(i, :))) / std(noisy\_data(i, :));

[h\_values\_ks(i), p\_values\_ks(i)] = kstest(data\_standardized);

if p\_values\_ks(i) >= alpha

h\_values\_ks(i) = 0; % 接受正态性假设

else

h\_values\_ks(i) = 1; % 拒绝正态性假设

end

% 单样本 t 检验

[h\_values\_ttest(i), p\_values\_ttest(i)] = ttest(noisy\_data(i, :), test\_mean, 'Alpha', alpha);

sample\_means(i) = mean(noisy\_data(i, :));

% 输出每组正态性检验结果

result = '接受';

if h\_values\_ks(i) == 1

result = '拒绝';

end

fprintf('组 %d: 正态性 p值 = %.4f, %s正态分布 (H=%d); 均值检验 p值 = %.4f, H=%d\n', ...

i, p\_values\_ks(i), result, h\_values\_ks(i), p\_values\_ttest(i), h\_values\_ttest(i));

end

% 总结正态性检验结果

num\_accepted = sum(h\_values\_ks == 0);

fprintf('\n总结：在显著性水平 %.2f 下：\n', alpha);

fprintf('%d 组接受正态分布假设，%d 组拒绝正态分布假设。\n', ...

num\_accepted, num\_groups - num\_accepted);

*Question1：为什么正态检验总是高于显著性水平？*

*Answer1:因为添加的高斯噪声也是正态分布，因此合成后数据任服从正态分布。*

4、总体均值检验（对通过正态检验的组进行总体均值检验，即进行假设参数检验）

% 对通过正态性检验的组进行总体均值检验

normal\_groups = find(h\_values\_ks == 0);

if isempty(normal\_groups)

fprintf('没有组通过正态性检验，无法进行总体均值检验。\n');

else

% 合并通过正态检验的组的数据

normal\_data = noisy\_data(normal\_groups, :);

normal\_data = normal\_data(:); % 展平数据

% 总体均值检验

[h\_normal\_all, p\_normal\_all] = ttest(normal\_data, test\_mean, 'Alpha', alpha);

mean\_normal\_all = mean(normal\_data);

% 显示总体均值检验结果

fprintf('\n通过正态检验的组（组号：%s）：\n', num2str(normal\_groups'));

fprintf('样本均值 = %.4f, p值 = %.4f, H = %d\n', mean\_normal\_all, p\_normal\_all, h\_normal\_all);

if h\_normal\_all == 0

fprintf('接受总体均值等于 %.2f 的假设\n', test\_mean);

else

fprintf('拒绝总体均值等于 %.2f 的假设\n', test\_mean);

end

end

% 对所有数据进行总体均值检验

all\_data = noisy\_data(:); % 展平所有数据

[h\_all, p\_all] = ttest(all\_data, test\_mean, 'Alpha', alpha);

mean\_all = mean(all\_data);

fprintf('\n所有数据的总体均值检验结果：\n');

fprintf('样本均值 = %.4f, p值 = %.4f, H = %d\n', mean\_all, p\_all, h\_all);

if h\_all == 0

fprintf('接受总体均值等于 %.2f 的假设\n', test\_mean);

else

fprintf('拒绝总体均值等于 %.2f 的假设\n', test\_mean);

end

% 创建表格展示每组结果

group\_numbers = (1:num\_groups)';

stats\_table = table(group\_numbers, sample\_means, p\_values\_ttest, h\_values\_ttest, p\_values\_ks, h\_values\_ks, ...

'VariableNames', {'组号', '样本均值', '均值检验p值', '均值检验H值', '正态性p值', '正态性H值'});

disp('每组的检验结果：');

disp(stats\_table);

% 可视化 p 值分布

figure('Position', [100, 100, 1200, 400]);

% 正态性检验 p 值

subplot(1, 2, 1);

stem(1:num\_groups, p\_values\_ks, 'filled', 'MarkerFaceColor', 'g');

hold on;

plot([1, num\_groups], [alpha, alpha], 'r--', 'LineWidth', 1.5);

title('各组正态性检验的 p 值 (KS 检验)');

xlabel('组号');

ylabel('p 值');

grid on;

legend('p 值', sprintf('显著性水平 \\alpha = %.2f', alpha));

hold off;

% 均值检验 p 值

subplot(1, 2, 2);

stem(1:num\_groups, p\_values\_ttest, 'filled');

hold on;

plot([1, num\_groups], [alpha, alpha], 'r--', 'LineWidth', 1.5);

title('各组均值检验的 p 值');

xlabel('组号');

ylabel('p 值');

grid on;

legend('p 值', sprintf('显著性水平 \\alpha = %.2f', alpha));

hold off;

## 四、产品工序能力指数（使用6sigma)

上下规范值T\_U,T\_L是已知的产品加工要求，在代码中表现为USL和LSL，需要根据实际行业规定进行调整

% 初始化参数

num\_groups = size(noisy\_data, 1);

USL = 150; % 规格上限

LSL = -150; % 规格下限

cpk\_values = zeros(num\_groups, 1); % 存储Cpk值

sample\_means = zeros(num\_groups, 1); % 样本均值

sample\_stds = zeros(num\_groups, 1); % 样本标准差

% 计算每组的Cpk

for i = 1:num\_groups

sample\_means(i) = mean(noisy\_data(i, :));

sample\_stds(i) = std(noisy\_data(i, :));

% 计算Cpk

cpk\_upper = (USL - sample\_means(i)) / (3 \* sample\_stds(i)); % 上限能力

cpk\_lower = (sample\_means(i) - LSL) / (3 \* sample\_stds(i)); % 下限能力

cpk\_values(i) = min(cpk\_upper, cpk\_lower); % 取最小值

end

% 创建表格展示结果

group\_numbers = (1:num\_groups)';

stats\_table = table(group\_numbers, sample\_means, sample\_stds, cpk\_values, ...

'VariableNames', {'组号', '样本均值', '样本标准差', 'Cpk'});

% 显示结果

disp('每组的Cpk计算结果：');

disp(stats\_table);

% 评估Cpk

disp('Cpk评估：');

for i = 1:num\_groups

if cpk\_values(i) >= 1.33

assessment = '优秀';

elseif cpk\_values(i) >= 1.0

assessment = '合格';

else

assessment = '不足';

end

fprintf('组 %d: Cpk = %.4f (%s)\n', i, cpk\_values(i), assessment);

end

% 统计Cpk分布

num\_excellent = sum(cpk\_values >= 1.33);

num\_acceptable = sum(cpk\_values >= 1.0 & cpk\_values < 1.33);

num\_insufficient = sum(cpk\_values < 1.0);

fprintf('\n总结：\n');

fprintf('Cpk ≥ 1.33 (优秀): %d 组\n', num\_excellent);

fprintf('1.0 ≤ Cpk < 1.33 (合格): %d 组\n', num\_acceptable);

fprintf('Cpk < 1.0 (不足): %d 组\n', num\_insufficient);

% 可视化Cpk分布

figure('Position', [100, 100, 800, 400]);

stem(1:num\_groups, cpk\_values, 'filled');

hold on;

plot([1, num\_groups], [1.33, 1.33], 'g--', 'LineWidth', 1.5);

plot([1, num\_groups], [1.0, 1.0], 'r--', 'LineWidth', 1.5);

title('各组工序能力指数 (Cpk)');

xlabel('组号');

ylabel('Cpk值');

grid on;

legend('Cpk值', 'Cpk = 1.33 (优秀)', 'Cpk = 1.0 (合格)');

hold off;

## 五、产品过程控制（使用Shewhart控制图进行SPC过程）

计算均值控制图和标准差控制图，并使用八大规则进行判断产品生产过程是否可控：

*规则1：控制图上有一个点(对应某个批次数据)位于控制限以外；*

*规则2：连续9个点落在中心线同一侧；*

*规则3：连续6个点递增或者递减；*

*规则4：连续14个点交替上下；*

*规则5：连续3个点中有2个点落在中心线同侧的B区以外；*

*规则6：连续5个点中有4个点落在中心线同侧的C区以外；*

*规则7：连续15个点落在中心线两侧的C区；*

*规则8：连续8个点落在中心线两侧且无一点在C区以内。*

上述规则适应于均值控制图和方差控制图，最终目的是识别产品生产过程是否，仅仅受到了随机因素的影响。

% 初始化参数

num\_groups = size(noisy\_data, 1);

samples\_per\_group = size(noisy\_data, 2);

sample\_means = zeros(num\_groups, 1); % 样本均值

sample\_stds = zeros(num\_groups, 1); % 样本标准差

% 计算每组均值和标准差

for i = 1:num\_groups

sample\_means(i) = mean(noisy\_data(i, :));

sample\_stds(i) = std(noisy\_data(i, :));

end

% 计算控制图参数

grand\_mean = mean(sample\_means); % 总均值

avg\_std = mean(sample\_stds); % 平均标准差

% 控制图常数（n=10）

A3 = 0.975; % X-bar chart constant

B3 = 0.284; % S chart lower constant

B4 = 1.716; % S chart upper constant

% X-bar chart 控制限

UCL\_xbar = grand\_mean + A3 \* avg\_std;

LCL\_xbar = grand\_mean - A3 \* avg\_std;

% S chart 控制限

UCL\_s = B4 \* avg\_std;

LCL\_s = B3 \* avg\_std;

% 应用八大规则

rule\_violations\_xbar = cell(num\_groups, 1);

rule\_violations\_s = cell(num\_groups, 1);

sigma\_xbar = (UCL\_xbar - grand\_mean) / 3; % 1-sigma for X-bar chart

sigma\_s = (UCL\_s - avg\_std) / 3; % 1-sigma for S chart

for i = 1:num\_groups

violations\_xbar = '';

violations\_s = '';

% Rule 1: One point beyond 3-sigma limits

if sample\_means(i) > UCL\_xbar || sample\_means(i) < LCL\_xbar

violations\_xbar = [violations\_xbar 'Rule 1; '];

end

if sample\_stds(i) > UCL\_s || sample\_stds(i) < LCL\_s

violations\_s = [violations\_s 'Rule 1; '];

end

% Rule 2: Nine points in a row on same side of centerline

if i >= 9

if all(sample\_means(i-8:i) > grand\_mean) || all(sample\_means(i-8:i) < grand\_mean)

violations\_xbar = [violations\_xbar 'Rule 2; '];

end

if all(sample\_stds(i-8:i) > avg\_std) || all(sample\_stds(i-8:i) < avg\_std)

violations\_s = [violations\_s 'Rule 2; '];

end

end

% Rule 3: Six points in a row steadily increasing or decreasing

if i >= 6

diffs = diff(sample\_means(i-5:i));

if all(diffs > 0) || all(diffs < 0)

violations\_xbar = [violations\_xbar 'Rule 3; '];

end

diffs\_s = diff(sample\_stds(i-5:i));

if all(diffs\_s > 0) || all(diffs\_s < 0)

violations\_s = [violations\_s 'Rule 3; '];

end

end

% Rule 4: Fourteen points in a row alternating up and down

if i >= 14

signs = sign(diff(sample\_means(i-13:i)));

if all(abs(diff(signs)) == 2)

violations\_xbar = [violations\_xbar 'Rule 4; '];

end

signs\_s = sign(diff(sample\_stds(i-13:i)));

if all(abs(diff(signs\_s)) == 2)

violations\_s = [violations\_s 'Rule 4; '];

end

end

% Rule 5: Two out of three points beyond 2-sigma

if i >= 3

recent\_means = sample\_means(i-2:i);

if sum(recent\_means > grand\_mean + 2\*sigma\_xbar | recent\_means < grand\_mean - 2\*sigma\_xbar) >= 2

violations\_xbar = [violations\_xbar 'Rule 5; '];

end

recent\_stds = sample\_stds(i-2:i);

if sum(recent\_stds > avg\_std + 2\*sigma\_s | recent\_stds < avg\_std - 2\*sigma\_s) >= 2

violations\_s = [violations\_s 'Rule 5; '];

end

end

% Rule 6: Four out of five points beyond 1-sigma

if i >= 5

recent\_means = sample\_means(i-4:i);

if sum(recent\_means > grand\_mean + sigma\_xbar | recent\_means < grand\_mean - sigma\_xbar) >= 4

violations\_xbar = [violations\_xbar 'Rule 6; '];

end

recent\_stds = sample\_stds(i-4:i);

if sum(recent\_stds > avg\_std + sigma\_s | recent\_stds < avg\_std - sigma\_s) >= 4

violations\_s = [violations\_s 'Rule 6; '];

end

end

% Rule 7: Fifteen points in a row within 1-sigma

if i >= 15

if all(abs(sample\_means(i-14:i) - grand\_mean) < sigma\_xbar)

violations\_xbar = [violations\_xbar 'Rule 7; '];

end

if all(abs(sample\_stds(i-14:i) - avg\_std) < sigma\_s)

violations\_s = [violations\_s 'Rule 7; '];

end

end

% Rule 8: Eight points in a row beyond 1-sigma (on both sides)

if i >= 8

if all(abs(sample\_means(i-7:i) - grand\_mean) > sigma\_xbar)

violations\_xbar = [violations\_xbar 'Rule 8; '];

end

if all(abs(sample\_stds(i-7:i) - avg\_std) > sigma\_s)

violations\_s = [violations\_s 'Rule 8; '];

end

end

rule\_violations\_xbar{i} = violations\_xbar;

rule\_violations\_s{i} = violations\_s;

end

% 创建结果表格

stats\_table = table((1:num\_groups)', sample\_means, sample\_stds, rule\_violations\_xbar, rule\_violations\_s, ...

'VariableNames', {'组号', '样本均值', '样本标准差', 'Xbar\_违规', 'S\_违规'});

% 显示结果

disp('控制图分析结果：');

disp(stats\_table);

% 总结违规情况

out\_of\_control\_xbar = sum(~cellfun(@isempty, rule\_violations\_xbar));

out\_of\_control\_s = sum(~cellfun(@isempty, rule\_violations\_s));

fprintf('\n总结：\n');

fprintf('X-bar图：%d 组违反控制规则，表明过程均值可能失控。\n', out\_of\_control\_xbar);

fprintf('S图：%d 组违反控制规则，表明过程变异性可能失控。\n', out\_of\_control\_s);

if out\_of\_control\_xbar == 0 && out\_of\_control\_s == 0

fprintf('过程受控，无显著非随机模式。\n');

else

fprintf('过程可能失控，需调查违规原因（如设备、操作或环境变化）。\n');

end

% 可视化控制图

figure('Position', [100, 100, 1200, 800]);

% X-bar Chart

subplot(2, 1, 1);

plot(1:num\_groups, sample\_means, 'b.-', 'LineWidth', 1.5, 'MarkerSize', 15);

hold on;

plot([1, num\_groups], [grand\_mean, grand\_mean], 'k-', 'LineWidth', 1.5);

plot([1, num\_groups], [UCL\_xbar, UCL\_xbar], 'r--', 'LineWidth', 1.5);

plot([1, num\_groups], [LCL\_xbar, LCL\_xbar], 'r--', 'LineWidth', 1.5);

% 1-sigma and 2-sigma lines

plot([1, num\_groups], [grand\_mean + sigma\_xbar, grand\_mean + sigma\_xbar], 'g--', 'LineWidth', 1);

plot([1, num\_groups], [grand\_mean - sigma\_xbar, grand\_mean - sigma\_xbar], 'g--', 'LineWidth', 1);

plot([1, num\_groups], [grand\_mean + 2\*sigma\_xbar, grand\_mean + 2\*sigma\_xbar], 'm--', 'LineWidth', 1);

plot([1, num\_groups], [grand\_mean - 2\*sigma\_xbar, grand\_mean - 2\*sigma\_xbar], 'm--', 'LineWidth', 1);

title('X-bar Control Chart');

xlabel('组号');

ylabel('样本均值');

grid on;

legend('样本均值', '中心线', 'UCL/LCL (±3σ)', '±1σ', '±2σ', 'Location', 'Best');

hold off;

% S Chart

subplot(2, 1, 2);

plot(1:num\_groups, sample\_stds, 'b.-', 'LineWidth', 1.5, 'MarkerSize', 15);

hold on;

plot([1, num\_groups], [avg\_std, avg\_std], 'k-', 'LineWidth', 1.5);

plot([1, num\_groups], [UCL\_s, UCL\_s], 'r--', 'LineWidth', 1.5);

plot([1, num\_groups], [LCL\_s, LCL\_s], 'r--', 'LineWidth', 1.5);

% 1-sigma and 2-sigma lines

plot([1, num\_groups], [avg\_std + sigma\_s, avg\_std + sigma\_s], 'g--', 'LineWidth', 1);

plot([1, num\_groups], [avg\_std - sigma\_s, avg\_std - sigma\_s], 'g--', 'LineWidth', 1);

plot([1, num\_groups], [avg\_std + 2\*sigma\_s, avg\_std + 2\*sigma\_s], 'm--', 'LineWidth', 1);

plot([1, num\_groups], [avg\_std - 2\*sigma\_s, avg\_std - 2\*sigma\_s], 'm--', 'LineWidth', 1);

title('S Control Chart');

xlabel('组号');

ylabel('样本标准差');

grid on;

legend('样本标准差', '中心线', 'UCL/LCL', '±1σ', '±2σ', 'Location', 'Best');

hold off;