R语言相关代码整理

1. 假设检验相关函数

1.1 `var.test(x, y, ratio = 1, alternative = "two.sided", conf.level = 0.95, ...)`

用于检验两组样本方差是否相等（基于 F 分布的检验）。

- **x, y**
  - 两个数值向量，分别代表两个独立样本的观测值。
- **ratio**
  - 指定假设检验中零假设下的方差比值（σ_x^2/σ_y^2）。
  - 默认值为 1，即 $H_0: \sigma_x^2/\sigma_y^2 = 1$。
- **alternative**
  - 指定备择假设方向：
    - `"two.sided"`：双侧检验（默认），$H_1: \sigma_x^2/\sigma_y^2 \neq \text{ratio}$。
    - `"less"`：单侧左检验，$H_1: \sigma_x^2/\sigma_y^2 < \text{ratio}$。
    - `"greater"`：单侧右检验，$H_1: \sigma_x^2/\sigma_y^2 > \text{ratio}$。
- **conf.level**
  - 置信区间的置信水平（0–1 之间），默认为 0.95，对应 95% 置信区间。
- **...**
  - 其他参数

返回值要点

- `statistic`：检验统计量 F 值（即 $\hat{\sigma}_x^2/\hat{\sigma}_y^2$）。
- `parameter`：对应自由度 (df_1, df_2)。
- `p.value`：p 值。
- `conf.int`：基于 F 分布的方差比置信区间（若 `conf.level` < 1）。
- `estimate`：样本方差比 $\hat{\sigma}_x^2/\hat{\sigma}_y^2$。

1.2 `t.test(x, y = NULL, alternative = "two.sided", mu = 0, paired = FALSE, var.equal = FALSE, conf.level = 0.95, ...)`

用于对样本均值进行 t 检验，可处理单样本、两独立样本或成对样本的情形。

- **x**
  - 数值向量，代表第一个样本或（如果 `y=NULL` 时）单样本检验的数据。

- **y**
  - 数值向量，代表第二个样本。如果 `y=NULL`，则执行单样本检验，检验均值是否等于 `mu`；若 `paired=TRUE`，则 `x`、`y` 必须等长度，进行配对样本检验。
- **alternative**
  - 备择假设方向：
    - `"two.sided"`：双侧检验（默认），$H_1：\mu_x - \mu_y \neq \mu$。
    - `"less"`：单侧左检验，$H_1：\mu_x - \mu_y < \mu$。
    - `"greater"`：单侧右检验，$H_1：\mu_x - \mu_y > \mu$。
- **mu**
  - 数值，单样本检验时检测的假设均值；两独立样本检验时可指定检验的差值（默认0），即检验 $\mu_x - \mu_y = \mathrm{mu}$。
- **paired**
  - 逻辑值，是否进行配对样本检验。
    - `FALSE`（默认）：独立样本检验；
    - `TRUE`：`x`、`y` 必须等长度，计算差值 `x - y` 后做单样本t检验，相当于检验均值差是否为 `mu`。
- **var.equal**
  - 逻辑值，仅当 `paired = FALSE` 时生效：
    - `FALSE`（默认）：假定两组样本方差不相等，使用 Welch 校正自由度；
    - `TRUE`：假定两组样本方差相等，使用合并方差估计。
- **conf.level**
  - 置信区间置信水平，默认 0.95。
- **...**
  - 其他参数

返回值要点

- `statistic`：t 统计量值。
- `parameter`：自由度（df）。
- `p.value`：p 值。
- `conf.int`：均值差（或均值）置信区间。
- `estimate`：样本均值（或两组均值的差）。
- `null.value`：用于检验的 `mu`。

1.3 `qt(p, df, lower.tail = TRUE, log.p = FALSE)`

计算 t 分布的分位点（反累积分布函数，quantile）。

- **p**
  - 数值向量，位于 $[0, 1]$ 之间，表示累积分布函数的概率值。

- **df**
  - 自由度（degrees of freedom），正整数。
- **lower.tail**
  - 逻辑值，是否返回下尾概率对应的分位点。
    - `TRUE`（默认）：返回 $P(T \leq t) = p$ 的 t 值；
    - `FALSE`：返回 $P(T > t) = p$ 的分位点，相当于 `qt(1-p, df)`。
- **log.p**
  - 逻辑值，是否将输入的 `p` 看作对数概率（log-scale）。
    - `FALSE`（默认）：`p` 为常规概率值；
    - `TRUE`：`p` 是 $\log(p)$。

示例

```r
# 双侧 95% 区间对应的 t 临界值（df=10）
t_crit <- qt(1 - 0.05/2, df = 10)
```

1.4 `qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)`

计算标准正态分布的分位点（或一般正态分布，当指定 mean、sd 时）。

- **p**
  - 数值向量，位于 $[0, 1]$ 之间，表示累积分布函数的概率值。
- **mean**
  - 正态分布的均值 μ。
  - 默认 `0` 对应标准正态。
- **sd**
  - 正态分布的标准差 σ，必须为正数。
  - 默认 `1` 对应标准正态。
- **lower.tail**
  - 逻辑值：
    - `TRUE`（默认）：返回满足 $P(X \leq x) = p$ 的分位点；
    - `FALSE`：返回满足 $P(X > x) = p$ 的分位点（即 `qnorm(1-p, ...)`）。
- **log.p**
  - 逻辑值：是否将 `p` 看作 $\log(p)$。

1.5 `wilcox.test(x, y = NULL, alternative = "two.sided", mu = 0, paired = FALSE, exact = NULL, correct = TRUE, conf.int = FALSE, conf.level = 0.95, ...)`

用于威尔科克森秩和检验（Wilcoxon rank-sum test 或 Wilcoxon signed-rank test），适用于非正态情形下的中位数比较。

- **x**
  - 数值向量；若 `paired = FALSE`（默认）且 `y=NULL`，则为单样本检验；若 `paired = TRUE` 且 `y` 提供，则为配对检验；若 `paired = FALSE` 且 `y` 非 NULL，则两独立样本检验。
- **y**
  - 数值向量，第二个样本的数据。
- **alternative**
  - 候选备择假设：
    - `"two.sided"`（默认），双侧检验；
    - `"less"`：检验 $\theta_x - \theta_y < \mu$；
    - `"greater"`：检验 $\theta_x - \theta_y > \mu$，其中 θ 表示中位数或分布位置参数。
- **mu**
  - 用于配对检验或单样本检验时，假设的中位数差值，默认 0。
- **paired**
  - 逻辑值，是否进行配对检验。
- **exact**
  - 逻辑值或 `NULL`：是否计算精确 p 值。
    - `TRUE`：强制精确检验；
    - `FALSE`：使用正态近似；
    - `NULL`（默认）：由 R 自动根据样本量（一般当样本量较小时使用精确方法）确定。
- **correct**
  - 逻辑值，在大样本情况下对连续性进行校正。仅当 `exact=FALSE` 时有意义。
- **conf.int**
  - 逻辑值，是否返回中位数差/位置差的置信区间。
- **conf.level**
  - 置信区间水平（仅当 `conf.int=TRUE` 有效），默认 0.95。
- **...**
  - 其他参数

1.6 `wilcoxsign_test(formula, data, alternative = "two.sided", mu = 0, distribution = "exact", ...)`

基于 **coin** 包的威尔科克森符号检验（或签名检验），能处理配对和独立样本的情形，支持精确 p 值计算。

- **formula**
  - 公式形式，如 `response ~ group` 或 `response ~ 1 | block`。
  - 对于配对样本，常用形如 `y - x ~ 1` 或 `Diff ~ 1`。
- **data**
  - 数据框，包含公式中使用的所有变量。
- **alternative**
  - 备择假设方向，同 `wilcox.test`：`"two.sided"`, `"less"`, `"greater"`。
- **mu**
  - 中位数差的假设值，默认 `0`。
- **distribution**
  - p 值的分布计算方式：
    - `"exact"`：精确枚举法；
    - `"approximate"`：近似法（当样本量较大时）；
- **...**
  - 其他参数

1.7 `kruskal.test(x, g, ...)` 或 `kruskal.test(formula, data, ...)`

用于多组独立样本的 Kruskal–Wallis 检验，是非参数等价于单因素方差分析。

- **x**
  - 当使用向量 & 分组向量时：
    - 第一个参数 `x`：数值向量，所有观测值合并。
    - 第二个参数 `g`：一个因子（或可转换为因子）的向量，表示每个观测值对应的组别。
  - 当使用公式时，常见写法 `response ~ group`。
- **g**
  - 因子或可转换为因子的向量，长度与 `x` 相同，表示分组。
- **formula**
  - 类似 `response ~ group`，`response` 为数值型，`group` 为因子。
- **data**
  - 数据框，当使用公式时需指定数据源。
- **...**
  - 其他参数。

返回值要点

- `statistic`：Kruskal–Wallis H 统计量。
- `parameter`：自由度，即组数减 1。
- `p.value`：p 值。
- `method`：检验方法名称。
- `data.name`：调用时使用的对象名称。

1.8 `chisq.test(x, p = rep(1/length(x), length(x)), rescale.p = FALSE, simulate.p.value = FALSE, B = 2000)`

用于单向卡方检验或列联表卡方检验。

- **x**
  - 可以是：
    1. **数值向量**：表示单样本多类别观测频数（即 good-of-fit 检验）。
    2. **矩阵/表格（table 或 matrix）**：表示列联表，行列对应不同分类水平。
- **p**
  - 当 `x` 为向量时，`p` 是一个数值向量，表示在零假设下各类别的理论概率或比例（必须和 `x` 长度一致且相加为1）。默认均匀分布。
  - 当 `x` 为矩阵时，一般忽略此参数。
- **rescale.p**
  - 逻辑值，仅当 `x` 为向量时有效：
    - `TRUE`：将 `p` 重新归一化（使其之和为1）。
    - `FALSE`（默认）：要求 `sum(p) == 1`。
- **simulate.p.value**
  - 逻辑值，是否使用模拟蒙特卡洛方法计算精确 p 值。
    - `FALSE`（默认）：使用渐近 χ^2 分布；
    - `TRUE`：进行模拟，需同时指定 `B`。
- **B**
  - 当 `simulate.p.value = TRUE` 时，指定模拟次数（重采样次数），默认 2000。

返回值要点

- `statistic`：卡方统计量值。
- `parameter`：自由度（单样本检验时 = k-1，列联表时 = (r-1)*(c-1)）。
- `p.value`：p 值（若模拟，则为模拟结果）。
- `observed`：观测频数（列联表）。
- `expected`：理论期望频数。
- `residuals`：Pearson 残差 $\frac{O-E}{\sqrt{E}}$。

- `stdres`：标准化残差 $\frac{O-E}{\sqrt{E(1-p_i)}}$ 等。

1.9 `pchisq(q, df, lower.tail = TRUE, log.p = FALSE)`

卡方分布的累积分布函数（CDF），用于计算上/下尾概率。

- **q**
  - 数值或向量，指定要计算累积概率的卡方统计量值。
- **df**
  - 自由度，正整数。
- **lower.tail**
  - 逻辑值：
    - `TRUE`（默认）：返回 $P(\chi^2 \leq q)$；
    - `FALSE`：返回 $P(\chi^2 > q)$。
- **log.p**
  - 逻辑值，是否将结果以对数形式返回。

1.10 `qchisq(p, df, lower.tail = TRUE, log.p = FALSE)`

卡方分布的分位点函数（反累积分布函数）。

- 参数与 `pchisq` 类似，只是输入输出分别对调：
  - **p**：概率值（$[0, 1]$）。
  - **df**：自由度。
  - **lower.tail**：是否返回下尾分位点。
  - **log.p**：是否将 `p` 视为对数概率。

1.11 `fisher.test(x, y = NULL, alternative = "two.sided", workspace = 2e5, hybrid = FALSE, control = list(), conf.int = TRUE, conf.level = 0.95, simulate.p.value = FALSE, B = 2000)`

对 2×2（或更高维）列联表进行 Fisher 精确检验。

- **x**
  - 可以是 2×2 矩阵，也可以是拟合成 `table(x, y)` 产生的列联表。
- **y**
  - 当 `x` 为向量时，`y` 可为另一个向量，用于构建列联表 `table(x, y)`。否则忽略。
- **alternative**
  - 备择假设方向：

    - `"two.sided"`（默认）、
    - `"less"`（检验 `p_{11} / p_{12} < 1`），
    - `"greater"`。
- **workspace**
    - 数值，内存限制（双整数形式），用于内部算法存储临时表，默认 `2e5`。若出现 “内存不足” 错误，可增大此值。
- **hybrid**
    - 逻辑值，仅当表维度 > 2×2 且 `simulate.p.value = FALSE` 时有效。
- **control**
    - 控制参数列表，用于高级配置，例如 `control = list(maxnp = ...)`。一般无需改动。
- **conf.int**
    - 逻辑值，是否计算比值比（odds ratio）的精确置信区间。
- **conf.level**
    - 置信区间水平，默认 0.95。
- **simulate.p.value**
    - 逻辑值，若列联表维度大于 2×2 或数据量非常大，可以指定模拟方法。
- **B**
    - 当 `simulate.p.value = TRUE` 时，指定模拟次数（重采样次数），默认 2000。

返回值要点

- `estimate`：2×2 表时的优势比（odds ratio）。
- `p.value`：Fisher 精确检验的 p 值（或模拟 p 值）。
- `conf.int`：优势比的置信区间。
- `null.value`：零假设下优势比的值（一般 1）。

1.12 `mcnemar.test(x, y = NULL, correct = TRUE)`

用于配对二项表的 McNemar 检验，检验处理前后、两种条件下的配对个体响应是否有显著变化。

- **x**
    - 2×2 矩阵或 `table` 对象，表示配对结果。例如：

      ```
          B=0  B=1
      A=0  a    b
      A=1  c    d
      ```

      关注的是 `b`（A=0→B=1）与 `c`（A=1→B=0）。
- **y**
    - 当 `x` 为两个向量时，可传入 `x`、`y` 并内部构造表格。一般传 `x` 为 2×2 表。
- **correct**

- 逻辑值，是否使用 Yates 连续性校正。
  - `TRUE`（默认）：使用校正。
  - `FALSE`：不校正。

返回值要点

- `statistic`：χ^2 或校正后的统计量值。
- `parameter`：自由度（始终为 1）。
- `p.value`：p 值。
- `method`、`data.name`：输出方法说明与数据名称。

1.13 `prop.test(x, n, p = NULL, alternative = "two.sided", conf.level = 0.95, correct = TRUE)`

用于两组或多组比例检验（Wald 近似检验），也可用于一个总体比例的区间估计。

- **x**
  - 整数或整数向量，表示“成功”次数。
    - 单个数值：单总体比例检验；
    - 长度 > 1：多总体比例检验，元素数目对应不同样本（或不同组）。
- **n**
  - 整数或整数向量，表示每组总观测次数，长度须与 `x` 一致。
- **p**
  - 单个数值或同长度向量：用于单总体检验时指定零假设比例（或多组比例检验时指定各组比例）。
  - 默认 `NULL`。当 `length(x)>1` 且 `p=NULL` 时，检验 $\pi_1 = \pi_2 = \cdots$。
- **alternative**
  - 备择假设方向：`"two.sided"`, `"less"`, `"greater"`。
- **conf.level**
  - 置信区间水平，默认 0.95。
- **correct**
  - 逻辑值，是否使用 Yates 负偏差校正。仅当 `length(x)=1` 或 `length(x)=2` 时生效。
    - `TRUE`（默认）：校正；
    - `FALSE`：不校正。

返回值要点

- `statistic`：检验统计量值（χ^2 或校正后）。
- `parameter`：自由度（`k-1`，其中 `k = length(x)`）。
- `p.value`：p 值。
- `estimate`：估计的比例（或比例向量）。
- `conf.int`：比例的置信区间（仅当 `length(x)=1` 或调用方法允许）。

- `null.value`：零假设下的比例（或比例向量）。

1.14 `diffscoreci(x1, n1, x2, n2, conf.level = 0.95)`

来自 **PropCIs** 包，用于计算两个比例差的 Agresti–Caffo 风格置信区间。

- **x1, x2**
  - 成功次数（整数）。
- **n1, n2**
  - 总观测次数（整数）。
- **conf.level**
  - 置信区间水平，默认 0.95。

返回值要点

- `diff`：样本比例差 $\hat{p}_1 - \hat{p}_2$。
- `conf.int`：修正后的比例差置信区间。

1.15 `Phi(x, y = NULL, data = NULL, correct = FALSE, ...)`

来自 **DescTools** 包，用于计算 2×2 列联表的 Φ 相关系数。

- **x, y**
  - 当传入两个向量时，函数内部构造 `table(x, y)`。
- **data**
  - 数据框，当 `x, y` 为列名时可指定数据来源。
- **correct**
  - 逻辑值，是否对偏差进行斯莫尔校正（Yates 校正）。
- **...**
  - 其他参数

返回值要点

- `phi`：Φ 相关系数值（范围 $[-1, 1]$）。
- `p.value`：如果要求检验显著性，会输出 p 值。

1.16 `ContCoef(x, y = NULL, data = NULL, adjust = FALSE, ...)`

来自 **DescTools** 包，用于计算列联系数 C（基于卡方统计量的度量，衡量名义变量间强度）。

- **x, y**
  - 类似 `Phi`，可传入两个因子向量或构造前已经是 `table(x, y)`。

- **data**
  - 数据框，当使用列名时需指定。
- **adjust**
  - 逻辑值，是否对列联系数进行偏差校正（基于 (χ^2/n) 的调整）。
- **...**
  - 其他不常见参数。

返回值要点

- `C`：列联系数值（取值范围 $[0, 1]$）。
- `p.value`：卡方检验对应的 p 值。

1.17 `CramerV(x, y = NULL, data = NULL, adjust = FALSE, ...)`

来自 **DescTools** 包，用于计算 Cramér's V（基于卡方统计量的多维列联系数）。

- **x, y**
  - 与上面函数类似，用以构造列联表。
- **data**
  - 数据框，当 `x, y` 为列名形式时指定来源。
- **adjust**
  - 逻辑值，是否对 Cramér's V 进行偏差校正（尤其在小样本时比较重要）。
- **...**
  - 其他参数

返回值要点

- `V`：Cramér's V 值，取值范围 $[0, 1]$。
- `p.value`：卡方检验对应的 p 值。

1.18 `cor.test(x, y = NULL, alternative = "two.sided", method = c("pearson", "kendall", "spearman"), exact = NULL, conf.level = 0.95, ...)`

用于连续变量间的相关性显著性检验，可选皮尔森（Pearson）、斯皮尔曼（Spearman）和 Kendall 等等级相关。

- **x, y**
  - 两个数值向量，长度必须一致。
- **alternative**
  - 备择假设方向：
    - `"two.sided"`（默认）：$H_1 : \rho \neq 0$。

    - `"less"`：$H_1 : \rho < 0$。
    - `"greater"`：$H_1 : \rho > 0$。
- **method**
    - 字符串，指定相关系数类型：
        - `"pearson"`：皮尔森积矩相关系数（需近似正态）。
        - `"spearman"`：Spearman 等级相关系数（对异常值较稳健）。
        - `"kendall"`：Kendall's tau 相关系数。
    - 可以传入字符向量候选，会自动选择第一个匹配项。
- **exact**
    - 逻辑值或 `NULL`：当 `method = "spearman"` 或 `"kendall"` 时，是否使用精确分布计算 p 值。默认 `NULL` 由 R 自动选择。
- **conf.level**
    - 置信区间水平，用于皮尔森相关系数的 Fisher Z 变换计算。
- **...**
    - 其他参数

返回值要点

- `estimate`：相关系数值。
- `statistic`：检验统计量（t 值或 z 值）。
- `parameter`：自由度（对于 Spearman/Kendall 为 NA 或近似值）。
- `p.value`：p 值。
- `conf.int`：相关系数的置信区间（仅对皮尔森相关提供）。

---

2. 方差分析（ANOVA/ANCOVA）相关函数

2.1 `aov(formula, data = NULL, contrasts = NULL, ...)`

基础 R 中用于拟合（可带交互项的）线性模型并做方差分析，主要返回可供 `summary()` 或 `anova()` 调用的 **aov** 对象。

- **formula**
    - 典型形式 `response ~ factor1 + factor2 + factor1:factor2 + ...`，其中 `:` 表示交互项。
    - 例如：`yield ~ fert + block`（单因素 ANCOVA 可写成 `yield ~ fert + covariate`）。
- **data**
    - 数据框，包含公式中所有变量。
- **contrasts**
    - 用于指定因子对比方式（contrast coding），如 `contrasts = list(fert = contr.treatment)`。若不指定，则使用全局 `options("contrasts")` 中的设置。
- **...**

- 其他参数

返回值要点

- 返回一个 **aov** 对象，可对其调用 `summary()` 得到 ANOVA 表（Type I 顺序平方和），或用 `model.tables()` 查看预测平均值、残差等。

2.2 `anova(object, ...)`

对线性模型、aov 对象或多个模型进行方差分析表比较。常见用法：

- **anova(aov_obj)**
  - 当传入单个 **aov** 对象时，返回 Type I 方差分析表。
- **anova(lm_obj)**
  - 当传入线性模型对象（`lm`）时，同样返回 Type I 方差分析表。
- **anova(model1, model2, ..., test = "Chisq" / "F")**
  - 对比嵌套模型（如不含协变量 vs 含协变量、无随机效应 vs 含随机效应）时，计算差异平方和检验：
    - 默认试验统计量取决于模型类型（`Chisq` 常用于 `glm`，`F` 常用于 `lm`/`aov`）。

常见参数：

- **object**
  - 一个或多个模型对象。
- **test**
  - 指定检验类型，如 `"F"`、`"Chisq"`、`"LRT"` 等。一般对 `lm`/`aov` 用 `"F"`，对 `glm` 用 `"Chisq"`。
- **...**
  - 其他参数

2.3 `Anova(mod, type = c("I", "II", "III"), ...)`

来自 **car** 包，用于输出 Type II 或 Type III 方差分析表，适用于不平衡设计或需要特定次序平方和的情形。

- **mod**
  - `lm` 或 `aov` 生成的线性模型对象。
- **type**
  - 指定平方和类型：
    - `"I"`：Type I 顺序平方和（默认，与 `anova()` 返回一致）；
    - `"II"`：Type II 平方和（在不平衡时更常用）；
    - `"III"`：Type III 平方和（要求对比方式为正交或在设计矩阵中包含所有主效应及交互）。
- **...**
  - 其他参数，如 `white.adjust = FALSE`（用于异方差校正检验）。

注意

- 使用 Type III 时，公式中因子需要设置合适的对比编码（如 `options(contrasts = c("contr.sum", "contr.poly"))`），否则结果可能不正确。

2.4 `bartlett.test(formula, data = NULL)` 或 `bartlett.test(x, g, data = NULL)`

用于检验多个正态分布组的方差齐性。

- **formula**
  - 公式形式 `response ~ group`，需满足 `response` 为数值型，`group` 为因子。
- **x, g**
  - 当不使用公式时：
    - `x`：数值向量，表示所有观测值；
    - `g`：因子或可转换为因子的向量，表示每个观测值所属的组。
- **data**
  - 数据框，当使用公式时指定。

返回值要点

- `statistic`：Bartlett 检验统计量。
- `parameter`：自由度（组数 - 1）。
- `p.value`：p 值。
- `method`、`data.name`：输出方法说明与数据名称。

2.5 `fligner.test(formula, data = NULL)` 或 `fligner.test(x, g, data = NULL)`

用于检验多组方差齐性，对数据正态性要求较弱（基于秩）。

- 参数同 `bartlett.test`：
  - **formula**：`response ~ group`。
  - **x**：数值向量。
  - **g**：因子向量。
  - **data**：数据框。

返回值要点

- `statistic`：Fligner–Killeen 检验统计量（基于检验秩）。
- 其他返回值与 `bartlett.test` 类似。

2.6 `leveneTest(formula, data, center = median, ...)`

来自 **car** 包的 Levene 检验，用于方差齐性检验，对正态分布要求较弱，可指定中心化方式。

- **formula**
  - 形如 `response ~ group`。
- **data**
  - 数据框。
- **center**
  - 指定中心化函数，用于计算组内偏差：
    - `median`（默认）：基于中位数计算绝对偏差（更稳健）；
    - `mean`：基于均值计算绝对偏差。
- **...**
  - 其他参数。

返回值要点

- 类似于 `anova()` 的输出，包括群组间平方和、组内平方和、`F` 值和 p 值。

2.7 事后多重比较函数（基于 agricolae 包）

> 以下所有事后比较方法均需先拟合单因素 `aov` 模型（例如 `model <- aov(y ~ group, data = df)`），再将该模型传入对应函数。

2.7.1 `LSD.test(model, trt, alpha = 0.05, group = TRUE, console = TRUE, ...)`

最小显著差（Least Significant Difference, LSD）检验。

- **model**
  - `aov` 对象。
- **trt**
  - 字符串，表示因子名称，例如 `"group"`。
- **alpha**
  - 显著性水平（0–1 之间），默认 0.05。
- **group**
  - 逻辑值，是否对处理按统计显著性分组（输出字母表示法），默认为 `TRUE`。
- **console**
  - 逻辑值，是否在控制台输出结果（`TRUE`）或仅返回列表（`FALSE`）。
- **...**
  - 其他参数，如 `p.adj`（用于 p 值校正，但 LSD 本身一般不做校正）。

返回值要点

- `statistics`：各组均值及样本量等信息。
- `LSD`：LSD 值。
- `groups`：分组结果，字母表示法。

2.7.2 `SNK.test(model, trt, alpha = 0.05, group = TRUE, console = TRUE, ...)`

Student–Newman–Keuls (SNK) 事后检验。

- **参数**
  - 与 `LSD.test` 类似：
    - `model`：`aov` 对象；
    - `trt`：因子名称；
    - `alpha`：显著性水平；
    - `group`：是否分组；
    - `console`：是否在控制台输出。
  - `...`
    - 其它参数，可能包括 `p.adj` 等参数，用于多重检验校正（SNK 方法一般按序排列进行比较，无全局校正选项）。

返回值要点

- `statistics`：均值排序及分组信息。
- `SNK`：各步临界值及比较结果。
- `groups`：按显著性分组的字母表示法。

2.7.3 `HSD.test(model, trt, alpha = 0.05, group = TRUE, console = TRUE, ...)`

Tukey HSD（Honest Significant Difference）事后检验。

- **参数**
  - 与上两者基本一致：
    - `model`：`aov` 对象；
    - `trt`：因子名称；
    - `alpha`：显著性水平；
    - `group`：分组指示；
    - `console`：是否输出。
  - `...`
    - 其他参数，包括 `mean.protocol = FALSE/TRUE`，是否输出均值排序协议等。

返回值要点

- `statistics`：各组均值及标准误等。
- `HSD`：Tukey HSD 多重比较结果，包括校正后的 p 值和置信区间。
- `groups`：显著性分组结果。

2.7.4 `duncan.test(model, trt, alpha = 0.05, group = TRUE, console = TRUE, ...)`

Duncan 多重检验（基于 Duncan's new multiple range test）。

- **参数**
  - 与前述几乎相同：
    - `model`：`aov` 对象；
    - `trt`：因子名称；
    - `alpha`：显著性水平；
    - `group`：是否分组；
    - `console`：是否输出；
  - `...`
    - 其他参数

返回值要点

- `statistics`：组均值、差异等。
- `Duncan`：各组两两比较结果，对显著性水平按 Duncan 校正进行说明。
- `groups`：显著性分组。

2.8 协方差分析（ANCOVA）相关函数

2.8.1 `aov(y ~ factor(A) + x, data = df)`

将定量协变量 `x` 与分类型自变量 `factor(A)` 同时放入模型，等价于 `lm(y ~ x + factor(A), data = df)`，再使用 `anova()` 比较或直接查看结果。

- **y**
  - 响应变量。
- **factor(A)**
  - 分类协变量（如处理水平）。
- **x**
  - 连续型协变量。
- **data**
  - 数据框，包含上述变量。

后续步骤

- 若需检验协变量与因子之间的交互，可写成 `y ~ factor(A) * x`。
- 事后比较：`TukeyHSD(anova_mod, "factor(A)")`。
- 估计调整后组均值：使用 **emmeans** 包。

2.8.2 `lm(y ~ x + factor(A), data = df)`

与上述 `aov` 等价，将 ANCOVA 当作线性模型拟合，用 `anova()` 比较是否显著。

- 参数同上。

2.8.3 `TukeyHSD(aov_mod, which, ordered = FALSE, conf.level = 0.95, ...)`

对 ANOVA 或 ANCOVA `aov` 对象进行 Tukey HSD 事后多重比较。

- **aov_mod**
  - 拟合的 `aov` 对象。
- **which**
  - 字符串或因子名称，指定对哪个因子水平进行多重比较。
- **ordered**
  - 逻辑值，是否根据均值排序后再进行比较。
- **conf.level**
  - 置信区间水平，默认 0.95。
- **...**
  - 其他参数，如 `console = FALSE`（不输出到控制台）。

返回值要点

- 显示指定因子下所有水平两两比较的均值差、p 值、置信区间等。

2.8.4 `emmeans(model, specs, at = NULL, adjust = "none", from = NULL, cov.reduce = mean, ...)`

来自 **emmeans** 包，用于计算模型（包括 ANCOVA）中不同因子水平的边际或校正后均值（LS-means）。

- **model**
  - `lm`、`aov`、`lmer` 等模型对象。
- **specs**
  - 字符串或公式，指定需要比较的因子。例如 `~ factor(A)`。
- **at**
  - 列表，用于在特定协变量值下计算LS-均值。例如 `at = list(x = mean(df$x))`。
- **adjust**
  - 多重比较校正方法，常见如 `"tukey"`, `"bonferroni"`, `"none"`（默认）。

- **from**
  - 指定从哪个模型对象或哪个分组开始检索。一般无需设置。
- **cov.reduce**
  - 指定如何处理协变量，一般使用 `mean` 表示在协变量取样本平均值时的预测均值。
- **...**
  - 其他参数，例如 `type = "response"`（对二项模型输出概率预测）、`mode` 等。

返回值要点

- 返回一个 **emmGrid** 对象，包含各因子水平的估计均值、标准误、df、置信区间等，可进一步调用 `contrast()` 进行组间差异比较。

---

3. 回归与拟合相关函数

3.1 线性与多元回归

3.1.1 `lm(formula, data, subset, weights, na.action, method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL, offset, ...)`

用于拟合线性回归模型，返回一个 `lm` 对象。常用参数：

- **formula**
  - 线性模型公式，例如 `y ~ x1 + x2 + x1:x2`（交互项）或二次项写成 `y ~ poly(x, 2)`。
- **data**
  - 数据框，包含公式中涉及的所有变量。
- **subset**
  - 向量或逻辑表达式，用于指定使用数据的一个子集。例如 `subset = (group == "A")`。
- **weights**
  - 数值向量，为加权最小二乘提供权重。
- **na.action**
  - 指定处理缺失值的方式，常用 `"na.omit"`（默认）或 `"na.exclude"`。
- **method**
  - 求解方法，通常保持默认 `"qr"`（QR 分解）；也可用 `"model.frame"` 输出数据框而不拟合模型等少见用法。
- **model**
  - 逻辑值，是否将 `model.frame` 存储在返回对象中，默认 `TRUE`。
- **x, y**
  - 逻辑值，是否在返回对象中保留设计矩阵 `x` 与响应向量 `y`，对后续诊断或预测有用。
- **qr**

    - 逻辑值，是否在返回对象中保留 QR 分解结果（默认 `TRUE`）。若设为 `FALSE`，将节省内存，但会丢失某些诊断输出。
- **singular.ok**
    - 逻辑值，是否允许设计矩阵存在共线性（奇异性）。若 `FALSE`，共线性会使函数报错。
- **contrasts**
    - 对因子变量使用的对比编码，可为列表形式，如 `list(group = contr.treatment, block = contr.poly)`。
- **offset**
    - 数值向量，用于偏移变量，例如 `lm(log(y) ~ x + offset(log(n)), data = df)`。
- **...**
    - 其他参数，如 `singular.ok` 等。

常用后续函数

- `summary(model)`：输出回归系数、标准误、t 值、p 值、R^2、F 统计量等。
- `confint(model, level = 0.95)`：计算回归系数置信区间。
- `anova(model)`：返回模型的回归平方和、残差平方和等 ANOVA 表。
- `predict(model, newdata, interval = "confidence"/"prediction", level = 0.95)`：生成预测值及置信/预测区间。
- `plot(model, which = 1:6)`：生成残差诊断图（具体见下方“诊断图”部分）。

3.1.2 `summary(model)`

适用于 `lm`、`glm`、`aov` 等模型对象，输出模型摘要。

- **model**：`lm`、`glm`、`aov` 对象等。
- **...**：其他参数

输出要点（以 `lm` 为例）

- **Call**：原始函数调用。
- **Residuals**：残差的最小值、1/4 分位、中位数、3/4 分位、最大值。
- **Coefficients**：包含系数估计值、标准误（Std. Error）、t 值（或 z 值）、对应的 p 值。
- **Residual standard error**：残差标准差 $\hat{\sigma}$。
- **Degrees of freedom**：自由度（总样本数 - 模型参数个数）。
- **Multiple R-squared**：复相关系数的平方 R^2。
- **Adjusted R-squared**：调整后的 R^2。
- **F-statistic**：回归方程整体显著性检验的 F 值、自由度与 p 值。

3.1.3 `confint(object, parm = NULL, level = 0.95, ...)`

计算回归模型中参数的置信区间。

- **object**
  - `lm`、`glm` 或其他支持 `confint` 方法的模型对象。
- **parm**
  - 字符串或整数向量，指定要计算置信区间的参数名称或索引。例如 `parm = "x1"` 或 `parm = 2:4`。
  - 默认 `NULL`，计算所有参数的置信区间。
- **level**
  - 置信水平，$(0, 1)$ 之间，默认 0.95。
- **...**
  - 其他参数，根据具体模型方法有所不同（如 `glm` 中可指定分布类型等）。

返回值要点

- 一个矩阵，行名为参数名，列分别为置信区间的下限（`2.5 %`）和上限（`97.5 %`）。

3.1.4 `predict(object, newdata, interval = c("none", "confidence", "prediction"), level = 0.95, se.fit = FALSE, type = c("link", "response"), ...)`

基于回归模型生成预测值，并可选置信区间/预测区间。

- **object**
  - `lm`、`glm`、`aov` 等模型对象。
- **newdata**
  - 新的预测数据框，必须包含所有模型中使用的自变量，列名要与建模时一致。
- **interval**
  - 预测区间类型：
    - `"none"`（默认）：仅返回预测值；
    - `"confidence"`：返回置信区间（针对估计的均值）；
    - `"prediction"`：返回预测区间（针对新的单个观测）。
- **level**
  - 区间置信水平，默认 0.95。
- **se.fit**
  - 逻辑值，是否返回预测值的标准误；仅在某些模型（如 `lm`）下可用。
- **type**
  - 对于 `glm`，可选：
    - `"link"`（默认）：返回线性预测器的值；
    - `"response"`：返回经过逆链接函数后的预测值（如概率）。

- **...**
  - 其他参数，如给 `glm` 指定分布族、对数链接等。

返回值要点

- 当 `interval = "confidence"` 或 `"prediction"` 时，返回数据框或矩阵，包含列：`fit`（预测值），`lwr`（下限），`upr`（上限）；若 `se.fit = TRUE`，还包含 `se.fit` 列。

3.1.5 `plot(model, which = 1:6, sub.caption = "", main = "", ask = FALSE, ...)`

生成回归诊断图（基础 R）。常用 `which` 值：

1. **1**：残差 vs 拟合值图（检测非线性趋势和异方差）。
2. **2**：标准化残差正态 Q–Q 图（检查正态性）。
3. **3**：平方根 |残差| vs 拟合值图（同样用于检测异方差）。
4. **4**：Cook's 距离图（检测高影响点）。
5. **5**：残差勒文斯基图（Residuals vs Leverage，检测高杠杆点）。
6. **6**：Cook 值 vs 杠杆值，带等高线（Cook's distance）图。

- **model**
  - `lm` 对象。
- **which**
  - 整数向量，指定要绘制哪些图。
- **sub.caption, main**
  - 标题及副标题，可留空。
- **ask**
  - 逻辑值，若 `TRUE`，在绘制多图时会暂停让用户按键后再显示下一图；若 `FALSE`（默认），一次性绘制所有指定的子图（或在同一设备上连续覆盖）。
- **...**
  - 其他图形参数，如 `pch`, `col` 等。

3.1.6 `avPlots(model, id.n = 2, id.cex = 1, scale = 1, ask = FALSE, main, sub.caption = "", ...)`

来自 **car** 包，用于绘制"添加变量图"(Added-variable plots，又称部分残差图)，可视化每个自变量在控制其他变量之后与响应变量的线性关系。

- **model**
  - `lm` 或 `glm` 对象。
- **id.n**
  - 整数，要标注的最显著观测点数目（基于 Cook's distance）。
- **id.cex**

    - 数值，标注编号的字体大小。
- **scale**
    - 缩放残差与拟合值的比率，默认 `1`。
- **ask**
    - 逻辑值，是否在绘制多张图时暂停以等待用户操作。
- **main**
    - 图形主标题。
- **sub.caption**
    - 副标题，可留空。
- **...**
    - 其他图形参数（如 `pch`, `col` 等）。

返回值要点

- 仅绘图，无返回值；图中点越偏离趋势线，表明在控制其他变量后该观测值对该自变量的系数影响越大。

3.2 非线性回归

3.2.1 `nls(formula, data, start, control = nls.control(), algorithm = "default", trace = FALSE, na.action = na.fail, ...)`

用于拟合非线性最小二乘模型。

- **formula**
    - 形式如 `response ~ f(x, a, b, ...)`，其中响应变量为数值，右侧 `f()` 是包含参数的非线性函数。
    - 例如：`Y ~ a * exp(b * X)`。
- **data**
    - 数据框，包含所有自变量和响应变量。
- **start**
    - 列表，指定所有待估参数的初始值。例如 `start = list(a = 1, b = 0.1)`。
    - 非线性拟合对初始值较敏感，需提供合理猜测。
- **control**
    - `nls.control()` 返回的列表，可指定拟合控制参数，如：
        - `maxiter`：最大迭代次数（默认 50）；
        - `tol`：收敛阈值（拟合参数更新大小）；
        - `minFactor`：最小步长因子；
        - `warnOnly`：收敛失败是否仅发出警告。
- **algorithm**
    - 算法类型，常用 `"default"`（Gauss-Newton 方法）或 `"port"`（利用 `MINPACK.lm` 包的 Levenberg-Marquardt）。

- **trace**
  - 逻辑值，是否打印每次迭代信息。
- **na.action**
  - 指定遇到缺失值时的处理方式，默认 `na.fail`，即存在缺失则报错。
- **...**
  - 其他参数，如指定自定义模型函数等。

常用后续函数

- `coef(nls_obj)`：提取拟合参数估计值。
- `confint(nls_obj)`：计算参数置信区间（基于 Wald 或 profile 方法）。
- `predict(nls_obj, newdata, interval = c("none", "confidence"), level = 0.95)`：生成预测值；
- `nlsResiduals()`（来自 **nlsr** 包）可用于残差诊断。

3.2.2 `coef(nls_obj)`

提取非线性拟合对象中的参数估计值。

- **nls_obj**
  - `nls` 返回的对象。

返回值要点

- 命名数值向量，每个元素对应模型中一个参数的估计值。

3.2.3 `lines(x, y, col = NULL, lwd = NULL, lty = NULL, ...)`

在已有图形上添加折线或曲线。

- **x, y**
  - 数值向量，必须与当前绘图坐标系保持一致，常用于绘制拟合曲线：
    - `x`：按顺序排序的自变量序列；
    - `y`：根据拟合模型计算的预测值。
- **col**
  - 颜色，可为名称或数字。若不指定，则使用当前颜色设置。
- **lwd**
  - 线宽，相对粗细，默认 `1`。
- **lty**
  - 线型，如 `1`（实线）、`2`（虚线）、`3`（点划线）等。
- **...**
  - 其他图形参数，如 `pch`、`type`（点线类型）等。

示例

```r
# 假设已有散点图 plot(x, y_obs)
x_seq <- seq(min(x), max(x), length.out = 100)
y_pred <- predict(nls_model, newdata = data.frame(x = x_seq))
lines(x_seq, y_pred, col = "blue", lwd = 2)
```

3.3 逐步回归与正则化回归

3.3.1 `step(object, scope = NULL, direction = c("both", "backward", "forward"), trace = 1, keep = NULL, steps = 1000, k = 2, ...)`

基于 AIC/BIC 的逐步变量选择。

- **object**
  - 初始模型对象，可以是 `lm`、`glm` 或其他支持 `extractAIC` 方法的模型。
- **scope**
  - 列表或公式，指定可选入/出的变量范围（模型空间）。
  - 例如：`scope = list(lower = ~1, upper = ~ x1 + x2 + x3 + x4)`。
- **direction**
  - 变量移入/移出方向：
    - `"both"`（默认）：双向逐步（可同时进行前向和后向）；
    - `"backward"`：后向剔除，从全模型开始；
    - `"forward"`：前向添加，从常数模型开始。
- **trace**
  - 整数，显示步骤信息：
    - `0`：无输出；
    - `1`（默认）：显示每步 AIC 值及模型变化；
    - `2`：更详细信息。
- **keep**
  - 函数，用于自定义如何保存每一步产生的模型。默认值 `NULL`。
- **steps**
  - 最大迭代步数，防止无限循环，默认 `1000`。
- **k**
  - 信息准则惩罚系数：
    - `k = 2`：表示 AIC；
    - `k = log(n)`：表示 BIC；
- **...**
  - 其他参数，如用于 `update.formula` 的额外信息。

返回值要点

- 一个简化后的模型对象（`lm`、`glm` 等），包含最终选择的变量。

3.3.2 `library(glmnet)` / `glmnet(x, y, family = c("gaussian","binomial","poisson",...), alpha = 1, nlambda = 100, lambda = NULL, standardize = TRUE, intercept = TRUE, thresh = 1e-7, ...)`

来自 **glmnet** 包，用于拟合 Ridge、Lasso、Elastic Net 以及广义线性模型（带 L1/L2 正则化）。

- **x**
  - 自变量矩阵，一般为数值矩阵（可由 `model.matrix()` 生成），**不包含截距列**。
  - 行数 = 样本数，列数 = 自变量个数。
- **y**
  - 响应变量：当 `family = "gaussian"` 时为数值向量；
  - 当 `family = "binomial"` 时，为二分类响应（0/1）或因子。
- **family**
  - 字符串，指定模型类型：
    - `"gaussian"`（回归，默认），
    - `"binomial"`（逻辑回归），
    - `"poisson"`（泊松回归），
    - `"multinomial"`（多分类），
    - `"cox"`（Cox 比例风险模型），
    - 还有 `"mgaussian"`, `"ordinal"`, `"cloglog"` 等。
- **alpha**
  - 数值，$\alpha \in [0, 1]$：
    - $\alpha = 1$：Lasso 回归（L1 正则化）；
    - $\alpha = 0$：Ridge 回归（L2 正则化）；
    - $0 < \alpha < 1$：Elastic Net。
- **nlambda**
  - 整数，要自动生成的 λ 值数量，默认 `100`。
- **lambda**
  - 数值向量，用户自行指定的 λ 值序列。若不指定，则按某规则自动生成。
- **standardize**
  - 逻辑值，是否对自变量进行标准化（均值中心化、除以标准差），默认 `TRUE`。
- **intercept**
  - 逻辑值，是否拟合截距项，默认 `TRUE`。若 `FALSE`，则不包含截距。
- **thresh**
  - 收敛阈值，迭代停止的标准，默认极小 `1e-7`。

- **...**
  - 其他参数，如 `maxit`（最大迭代次数）、`type.multinomial`（多分类类型）、`penalty.factor`（对不同变量给予不同惩罚系数）等。

常用后续函数

- `cv.glmnet(x, y, family, alpha, nfolds, ...)`：交叉验证选择最优 λ。
- `coef(fit, s = "lambda.min")`：提取指定 λ 下的回归系数。
- `predict(fit, newx, s = "lambda.min", type = c("link","response","coef","class"), ...)`：基于模型生成预测结果。

3.3.3 `cv.glmnet(x, y, family = "gaussian", nfolds = 10, alpha = 1, type.measure = c("mse","deviance","class",...), nlambda = 100, lambda.min.ratio = ifelse(nobs<nvars, 0.01, 1e-4), standardize = TRUE, intercept = TRUE, ...)`

基于 K 折交叉验证自动选择最优 λ。

- **x, y, family, alpha, nlambda, standardize, intercept**
  - 与 `glmnet()` 中含义相同。
- **nfolds**
  - 整数，指定交叉验证的折数，默认 `10`。
- **type.measure**
  - 指定用于评估误差的度量：
    - `"mse"`：均方误差（回归）；
    - `"deviance"`：偏差（广义线性模型）；
    - `"class"`：分类错误率（分类）；
    - `"auc"`：ROC AUC（分类）。
- **lambda.min.ratio**
  - 数值，在自动生成 λ 序列时，最小 λ 与最大 λ 的比值。默认：如果样本数 < 变量数，则为 0.01；否则为 1e-4。
- **...**
  - 其他高级参数，如 `weights`、`foldid`（指定各样本所属折的索引向量）等。

返回值要点

- `lambda.min`：使交叉验证误差最小的 λ。
- `lambda.1se`：在最优误差加一倍标准误范围内最大的 λ（较保守）。
- `cvm`：各 λ 下的平均误差。
- `cvsd`：各 λ 下误差标准差。

3.3.4 `coef(fit, s = c("lambda.min","lambda.1se"), exact = FALSE, ...)`

提取 `glmnet`/`cv.glmnet` 模型在指定 λ 下的回归系数。

- **fit**
  - `glmnet` 或 `cv.glmnet` 返回的对象。
- **s**
  - 指定 λ 值，可使用字符 `"lambda.min"` 或 `"lambda.1se"`（仅对 `cv.glmnet` 对象有效），也可指定实际数值。
- **exact**
  - 逻辑值，是否严格在 `lambda` 序列上查找，或在两端线性插值，默认 `FALSE`（插值）。
- **...**
  - 其他参数。

返回值要点

- “稀疏”矩阵或向量，包含截距项 `(Intercept)` 及自变量系数。

3.4 标准化回归系数

3.4.1 `lm.beta(model, use = c("complete.obs","pairwise.complete.obs"), verbose = FALSE)`

来自 **lm.beta** 包，用于计算线性模型的标准化回归系数（β 系数）。

- **model**
  - 拟合的 `lm` 对象。
- **use**
  - 缺失值处理方式：
    - `"complete.obs"`（默认）：仅使用完整观测。
    - `"pairwise.complete.obs"`：计算各对变量相关系数时允许成对删除缺失。
- **verbose**
  - 逻辑值，是否输出中间计算信息。

返回值要点

- 在原始系数上乘以自变量标准差与因变量标准差之比，得到标准化系数，可用于比较不同量纲自变量的重要性。

3.4.2 `scale(x, center = TRUE, scale = TRUE)`

对向量或矩阵进行中心化和标准化。

- **x**
  - 数值向量、矩阵或数据框。

- **center**
  - 布尔值或数值向量：
    - `TRUE`（默认）：按均值中心化（减去均值）；
    - `FALSE`：不中心化；
    - 数值向量：指定减去的数值（长度与列数匹配）。
- **scale**
  - 布尔值或数值向量：
    - `TRUE`（默认）：除以样本标准差（$\sqrt{\frac{1}{n-1}\sum(x-\bar{x})^2}$）；
    - `FALSE`：不缩放；
    - 数值向量：指定除以的数值（长度与列数匹配）。

返回值要点

- 返回同维度的矩阵或数据框，列中心化并标准化后的结果。

3.5 偏相关分析

3.5.1 `library(ppcor)`

载入 **ppcor** 包，该包用于计算偏相关系数。

- 无参数，调用后可使用 `pcor()` 系列函数。

3.5.2 `pcor(x, method = c("pearson","kendall","spearman"))`

计算多变量数据框/矩阵的偏相关系数矩阵及其对应 p 值。

- **x**
  - 数据框或矩阵，列为不同变量，行对应观测。必须为数值型。
- **method**
  - 指定相关系数类型：
    - `"pearson"`（默认）、`"spearman"`、`"kendall"`。

返回值要点

- 列表，包含：
  - `$estimate`：偏相关系数矩阵（对角为1）；
  - `$p.value`：对应的 p 值矩阵；
  - `$statistic`：Z 统计量或 t 统计量矩阵；
  - `$gp`：对角线元素数据信息；
  - `$n`：样本量。

---

4. 混合效应与随机效应模型

4.1 `lmer(formula, data, REML = TRUE, control = lmerControl(), ...)`

来自 **lme4** 包，用于拟合线性混合效应模型。

- **formula**
  - 典型形式 `response ~ fixed1 + fixed2 + (random1 | group1) + (random2 | group2) + ...`，
  - 其中 `(random | group)` 表示随机截距或随机斜率。例如 `(1 | Subject)` 表示对 `Subject` 作为随机截距；`(x | Subject)` 表示随机截距和随机斜率。
- **data**
  - 数据框，包含所有固定效应和随机效应的变量。
- **REML**
  - 逻辑值，是否使用 REML（限制最大似然）估计，默认 `TRUE`。
  - 若需比较嵌套模型，通常设 `REML = FALSE`（使用 ML）。
- **control**
  - `lmerControl()` 的结果，用于调整拟合算法（如迭代次数、优化方法等）。
- **...**
  - 其他参数，如 `start`（初始值）、`verbose` 等，不常用。

返回值要点

- 返回 `lmerMod` 对象，可调用 `summary()` 输出固定效应估计、随机效应方差分量、AIC/BIC、对数似然等。

4.2 `summary(model_random)`

适用于 `lmerMod` 对象，输出混合模型详细摘要。

- **model_random**
  - `lmer` 返回的对象。
- 无额外参数（可指定 `correlation = FALSE`、`ddf = "Satterthwaite"` 用于不同 df 计算方式）。

输出要点

- **Fixed effects**：固定效应估计值、标准误、t 值。
- **Random effects**：每个随机因素（分组变量）的方差成分及标准差。
- **AIC / BIC / logLik**：模型信息准则与对数似然。
- 如果指定了 `REML = FALSE`，则输出 ML 方法估计。

4.3 `VarCorr(model_random)`

提取 `lmerMod` 对象中各随机效应的方差和协方差阵。

- **model_random**
  - `lmer` 返回的对象。

返回值要点

- 一个包含各随机效应组（grouping factor）的列表，每个元素是一个包含方差与协方差的矩阵。

4.4 `anova(model_null, model_random, refit = FALSE, test = c("Chisq", "F"))`

对两个嵌套的混合模型进行模型比较，通常以似然比检验（LRT）为主。

- **model_null**
  - 被比较的简化模型（如无随机效应）。
- **model_random**
  - 完整模型（含随机效应）。
- **refit**
  - 逻辑值，是否在比较时重新拟合模型。默认 `FALSE`，即使用已拟合的对象。
- **test**
  - 指定检验统计量类型：
    - `"Chisq"`（默认）：似然比检验 χ^2；
    - `"F"`：基于 Satterthwaite 或 Kenward–Roger 方法近似 F 分布（需其他包支持）。

返回值要点

- 比较表格，包含：两个模型的对数似然、AIC、BIC、χ^2/F 统计量、p 值等，用于判断模型拟合是否显著改进。

---

5. 数据结构与绘图函数

5.1 数据结构与转换

5.1.1 `matrix(data, nrow, ncol, byrow = FALSE, dimnames = NULL)`

用于创建矩阵。

- **data**
  - 数值向量，用于填充矩阵元素，长度应为 `nrow * ncol` 或能被重复。
- **nrow**
  - 矩阵行数。
- **ncol**
  - 矩阵列数。

- **byrow**
  - 逻辑值：
    - `FALSE`（默认）：按列优先填充；
    - `TRUE`：按行优先填充。
- **dimnames**
  - 列表形式，包含行名与列名。例如 `list(row = c("r1","r2"), col = c("c1","c2"))`。

示例

```
# 构造 2×3 矩阵
M <- matrix(1:6, nrow = 2, ncol = 3, byrow = TRUE,
            dimnames = list(c("R1","R2"), c("C1","C2","C3")))
```

5.1.2 `data.frame(..., row.names = NULL, check.rows = FALSE, check.names = TRUE, stringsAsFactors = default.stringsAsFactors())`

用于创建数据框。

- **... 如x,y等**
  - 多个向量或矩阵，长度必须一致，也可包含因子或列表。同名列会自动重命名或报错（取决于 `check.names`）。
- **row.names**
  - 向量，指定行名。可用序号或字符向量。
- **check.rows**
  - 逻辑值，是否检验行名的重复与长度。
- **check.names**
  - 逻辑值，是否检查并修正列名使其合法（如含空格、数字开头等）。默认 `TRUE`。
- **stringsAsFactors**
  - 逻辑值，是否将字符向量自动转为因子（在 R 4.0.0 及以后版本，默认值为 `FALSE`；之前版本为 `TRUE`）。

5.1.3 `factor(x, levels = NULL, labels = NULL, exclude = NA, ordered = FALSE, ...)`

将向量转换为因子（分类变量）。

- **x**
  - 向量，可为数值、字符等。
- **levels**
  - 指定因子水平（类别）顺序；若不指定，默认取 `sort(unique(x))`。
- **labels**
  - 与 `levels` 相对应的标签，长度需与 `levels` 一致。

- **exclude**
  - 指定哪些值不纳入因子水平，默认 `NA`。
- **ordered**
  - 逻辑值，是否生成有序因子（若 `TRUE`，可进行有序检验）。
- **...**
  - 其他不常用参数。

5.1.4 `reshape(data, idvar, timevar, direction = c("wide","long"), varying = NULL, v.names = NULL, times = NULL, sep = ".", ...)`

基础 R 中用于长宽格式转换。

- **data**
  - 数据框，包含所有需要转化的变量。
- **idvar**
  - 标识行的标识符变量，单列名称或向量形式（如 `c("Subject","Trial")`）。
- **timevar**
  - 用于识别不同时间点或重复测量的变量名，例如 `"Time"`。
- **direction**
  - `"wide"` 或 `"long"`，指定从长格式到宽格式，或反之。
- **varying**
  - 当 `direction = "wide"` 时，可选：指定一组列名或位置，用于在多个列之间跨行聚合；
  - 当 `direction = "long"` 时，系统会自动按 `idvar`、`timevar` 检索，若不合适可手动指定。
- **v.names**
  - 当 `direction = "long"` 时，指定汇总后的变量名称。
- **times**
  - 向量，指定 `timevar` 对应的具体值序列。
- **sep**
  - 当 `direction = "wide"` 时，指定展开后列名的分隔符，默认 `"."`。
- **...**
  - 其他高级参数，如 `drop`（是否删除 NA 行）、`new.row.names`（新行名）等。

示例

```r
# 将长格式 data 长表转换为宽表
wide_df <- reshape(long_df, idvar = "Subject", timevar = "Time", direction = "wide")
```

5.1.5 `pivot_longer(data, cols, names_to, values_to, names_pattern = NULL, values_drop_na = FALSE, ...)`

来自 **tidyr** 包，将宽格式数据转换成长格式。

- **data**
  - 数据框。
- **cols**
  - 指定要“拉长”的列，可以是通配选择（如 `starts_with("t")`）或列名向量。
- **names_to**
  - 字符串或字符串向量，表示将原列名拆分后放入新列的名称。
- **values_to**
  - 字符串，指定将原列对应值放入哪个新列。
- **names_pattern**
  - 正则表达式，用于从原列名中提取信息并分别填充到 `names_to` 指定的新列中；捕获组会按顺序对应 `names_to`。
- **values_drop_na**
  - 逻辑值，若 `TRUE`，则删除 `NA` 值对应的行。
- **...**
  - 其他常见参数，如 `names_sep`（使用分隔符拆分列名）等。

5.2 绘图函数

5.2.1 `plot(x, y = NULL, type = c("p","l","b","c","o","h","s","S","n"), main = NULL, sub = NULL, xlab = NULL, ylab = NULL, xlim = NULL, ylim = NULL, pch = NULL, col = NULL, lwd = NULL, lty = NULL, ...)`

基础 R 中通用绘图函数，用于散点图、折线图等。

- **x, y**
  - 数值向量；若只提供 `x`，则默认以索引为 x 轴绘制 `x` 的数值；若同时提供 `x, y`，绘制二维坐标图。
- **type**
  - 绘图类型：
    - `"p"`：散点（points）；
    - `"l"`：折线（lines）；
    - `"b"`：散点+折线；
    - `"c"`：仅线段（相当于折线但不绘制点）；
    - `"o"`：点在折线上；
    - `"h"`：直方条（像直方图）；

    - `"s","S"`：阶梯图；
    - `"n"`：仅坐标轴，不绘制数据。
- **main**
    - 图形主标题。
- **sub**
    - 子标题。
- **xlab, ylab**
    - x 轴、y 轴标签。
- **xlim, ylim**
    - 数值向量，指定坐标轴范围，如 `xlim = c(0, 10)`。
- **pch**
    - 点的符号/形状，整数或字符，如 `pch = 1`（空心圆），`pch = 16`（实心圆）。
- **col**
    - 颜色，可为名称或数字。
- **lwd**
    - 线宽，数值。
- **lty**
    - 线型：整数或字符，如 `1`（实线）、`2`（虚线）等。
- **...**
    - 其他图形参数，例如 `cex`（字符放大倍数）、`cex.axis`（坐标轴刻度放大倍数）等。

5.2.2 `lines(x, y, col = NULL, lwd = NULL, lty = NULL, ...)`

详见 **3.2.3**。

5.2.3 `polygon(x, y, density = NULL, angle = 45, border = NULL, col = NULL, lty = NULL, ...)`

在图形中绘制多边形，可用于绘制回归置信带。

- **x, y**
    - 数值向量，指定多边形顶点的坐标，通常形成一个闭合路径（自动按顺序连接最后一个点到第一个点）。
- **density**
    - 数值，填充线条的密度（单位线/英寸），若 `NULL`（默认），则绘制实心颜色。
- **angle**
    - 数值，线条填充的倾斜角度（以度为单位）。
- **border**
    - 颜色或 `NA`，指定边框线的颜色。若 `NA`，不绘制边框。
- **col**

    - 填充颜色。
- **lty**
    - 边框线的线型。
- **...**
    - 其他图形参数。

示例（回归置信带）

```
# 已有拟合直线 plot，x_seq，y_fit，lwr，upr：
x_poly <- c(x_seq，rev(x_seq))
y_poly <- c(lwr，rev(upr))
polygon(x_poly，y_poly，col = rgb(0，0，1，alpha = 0.2)，border = NA)
```

5.2.4 `legend(x, y = NULL, legend, col = NULL, lty = NULL, lwd = NULL, pch = NULL, pt.cex = 1, pt.bg = NULL, title = NULL, bty = "o", ...)`

向图形中添加图例。

- **x, y**
    - 图例位置：
        - 如果为单个字符串（如 `"topright"`、`"bottomleft"`、`"topleft"` 等），则自动放置；
        - 也可为坐标值数值向量，如 `x = 1.2, y = 3.4`。
- **legend**
    - 字符向量，表示每个图例项的标签。
- **col**
    - 颜色向量，与 `legend` 长度一致，用于显示各组颜色。
- **lty**
    - 线型向量，与 `col` 长度一致，若为空则不绘制线，仅绘制点。
- **lwd**
    - 线宽向量，与 `col` 一致。
- **pch**
    - 点形状向量，与 `legend` 一致，用于显示散点符号。
- **pt.cex**
    - 数值或数值向量，点的放大倍数。
- **pt.bg**
    - 点的背景色，对于某些符号（如 `pch = 21`）有效。
- **title**
    - 图例标题。
- **bty**

    - 边框类型：`"o"`（默认，矩形边框），`"n"`（无边框），`"l"`,`"7"`,`"c"`,`"u"` 等。
- **...**
    - 其他图形参数，如 `cex`（文本放大倍数）、`text.col`（文本颜色）等。

5.2.5 `barplot(height, beside = FALSE, names.arg = NULL, col = NULL, main = NULL, xlab = NULL, ylab = NULL, legend.text = NULL, args.legend = list(), ...)`

绘制柱状图，可用于分组柱状图对比或绘制观测与理论频数对比。

- **height**
    - 数值向量或矩阵：
        - 向量时，绘制单组柱状图；
        - 矩阵时，如果 `beside = FALSE`（默认），每列叠加；若 `beside = TRUE`，列代表组，将并排绘制分组柱。
- **beside**
    - 逻辑值，是否并排绘制分组柱。
- **names.arg**
    - 字符向量，指定每个柱的标签。
- **col**
    - 颜色向量，为每条柱或每组柱指定颜色。
- **main, xlab, ylab**
    - 图形主标题、x 轴标签、y 轴标签。
- **legend.text**
    - 字符向量，若 `height` 为矩阵并且 `beside=TRUE`，可设置图例文本。
- **args.legend**
    - 列表，传递给 `legend()` 函数的其他参数，如 `bty="n"`, `cex=0.8` 等。
- **...**
    - 其他图形参数，如 `ylim`（y 轴范围）、`border`（柱边框颜色）等。

5.2.6 `interaction.plot(x.factor, trace.factor, response, fun = mean, type = c("b","p","l","o","s","S","h"), pch = NULL, leg.bty = "o", xlab = NULL, ylab = NULL, main = NULL, lty = NULL, col = NULL, trace.label = deparse(substitute(trace.factor)), ...)`

用于可视化多因素 ANOVA 中因子间交互作用。

- **x.factor**
    - 因子或因子向量，将映射到 x 轴的水平。
- **trace.factor**

    - 第二个因子，将映射到不同曲线或点集。
- **response**
    - 响应变量向量，数值型。
- **fun**
    - 聚合函数，默认 `mean`，可为 `median` 等。
- **type**
    - 绘图类型，参见 `plot()` 中的 `type` 参数。
- **pch**
    - 点形状，向量或单值；若 `NULL`，系统根据因子水平自动选择。
- **leg.bty**
    - 图例边框类型，传给 `legend()`，如 `"o"`、`"n"` 等。
- **xlab, ylab, main**
    - x 轴标签、y 轴标签、主标题。
- **lty**
    - 线型，可为向量。
- **col**
    - 颜色，可为向量。
- **trace.label**
    - 字符串，用于图例标题，默认为 `trace.factor` 的名称。
- **...**
    - 其他图形参数，例如 `ylim`。

5.3 高级可视化

5.3.1 `ggplot(data, mapping = aes(...)) + geom_point(size = , color = ) + geom_smooth(method = "lm", se = TRUE, level = 0.95, ...) + labs(title = , x = , y = ) + theme_minimal() + ...`

来自 **ggplot2** 包，用于创建分层绘图。

- **ggplot(data, mapping)**
    - **data**：数据框；
    - **mapping**：由 `aes()` 生成的映射关系，例如 `aes(x = xvar, y = yvar, color = group)`。
- **geom_point(...)**
    - 绘制散点：
        - `size`：点大小；
        - `color`：点颜色；
        - `shape`：点形状；
        - `alpha`：透明度；

- **geom_smooth(method = "lm", se = TRUE, level = 0.95)**
  - 绘制拟合线和置信带：
    - `method`：拟合方法，如 `"lm"`、`"loess"`；
    - `se`：逻辑值，是否绘制置信区间；
    - `level`：置信水平。
- **labs(title, x, y, subtitle, caption)**
  - 添加标题、轴标签、子标题、图说等。
- **theme_minimal()**
  - 选择一个主题，可选的还有 `theme_classic()`、`theme_bw()` 等。
- **其他可选层**
  - `facet_wrap(~ group)`：分面绘图；
  - `scale_color_manual(values = c("red","blue"))`：自定义配色；
- **...**
  - 其他图层、缩放、坐标变换等。

5.3.2 `plot_ly() %>% add_markers(data = , x = ~ , y = ~ , z = ~ , marker = list(color =), name =) %>% add_surface(x = , y = , z = , showscale = FALSE, opacity = , colorscale = , name =) %>% layout(scene = list(xaxis = list(title =), yaxis = list(title =), zaxis = list(title =), camera = list(eye = list(x = , y = , z =))), title =)`

来自 **plotly** 包，用于交互式 3D 绘图。

- **plot_ly()**
  - 初始化绘图对象，可不带参数，后续通过 `%>%` 添加图层。
- **add_markers(...)**
  - 绘制三维散点：
    - `data`：数据框；
    - `x = ~var1, y = ~var2, z = ~var3`：公式指定三维坐标；
    - `marker = list(color = )`：指定点颜色或映射；
    - `name`：图例标签。
- **add_surface(...)**
  - 绘制三维表面（曲面）：
    - `x`, `y`：数值向量，指定网格点的 x、y 范围；
    - `z`：数值矩阵，维度为 `length(y) × length(x)`（注意行/列顺序）；
    - `showscale`：逻辑值，是否显示颜色刻度条；
    - `opacity`：透明度，范围 $[0, 1]$；

    - `colorscale`：颜色映射，可为 `"Viridis"`, `"Jet"` 等；
    - `name`：图例标签。
- **layout(...)**
    - 配置坐标轴与相机视角：
        - `scene = list(xaxis = list(title = "..."), yaxis = list(title = "..."), zaxis = list(title = "..."), camera = list(eye = list(x = , y = , z = )))`。
        - `title`：图形标题。
- **...**
    - 其他高级参数，如 `hoverinfo`、`colorbar` 等。

5.4 其他函数

5.4.1 `seq(from, to, by = NULL, length.out = NULL, along.with = NULL)`

生成数值序列。

- **from**
    - 起始值。
- **to**
    - 终止值。
- **by**
    - 步长；若指定 `by`，则 `length.out` 不可指定。
- **length.out**
    - 序列长度；与 `by` 二选一。
- **along.with**
    - 向量，其长度决定序列生成的长度（相当于 `length.out = length(along.with)` 并忽略 `from`, `to`）。

示例

```
seq(0, 1, by = 0.1)        # 0.0, 0.1, 0.2, ..., 1.0
seq(min(x), max(x), length.out = 100)  # 100 个等间隔点
```

5.4.2 `runif(n, min = 0, max = 1)` 与 `rnorm(n, mean = 0, sd = 1)`

在模拟或引入随机噪声时使用。

- **runif(n, min, max)**
    - **n**：生成随机数数量。
    - **min, max**：均匀分布上下界，生成 $[\min, \max]$ 区间内均匀随机数。
- **rnorm(n, mean, sd)**
    - **n**：生成随机数数量。

- **mean, sd**：正态分布参数，生成 $\mathcal{N}(\text{mean}, \text{sd}^2)$ 分布随机数。

5.4.3 `sum(...), abs(x), length(x), round(x, digits = 0)`

常见基本数值运算。

- **sum(...)**
  - 对数值向量求和，可含 `na.rm = TRUE`。
- **abs(x)**
  - 取绝对值。
- **length(x)**
  - 向量或列表长度。
- **round(x, digits)**
  - 四舍五入：`digits` 指定保留小数位数（正值为小数位，负值可用于十位、百位等）。

5.4.4 `cat(..., sep = " ", fill = FALSE, labels = NULL, append = FALSE)` 与 `print(x, ...)`

输出文本或变量值到控制台。

- **cat(...)**
  - 将多个对象拼接输出，可自定义分隔符 `sep`；`fill` 参数可控制自动换行；
  - 不会自动换行，需手动添加 `"\n"`。
- **print(x, ...)**
  - 将对象按其 `print` 方法输出，常用于调试或查看数值/因子/数据框等。

5.4.5 `options(contrasts = c("contr.treatment", "contr.poly"))`

设置因子对比方式，用于线性模型中对交互效应或 Type III 方差分析的正确性至关重要。

- `contr.treatment`：治疗对比（dummy coding），参考组效应为 0。
- `contr.poly`：多项式对比，用于有序因子。
- 例如：

  ```
  options(contrasts = c("contr.sum", "contr.poly"))
  ```

  设置对比方式为和为零的对比，对于 Type III 方差分析更适合。
