表 1. 不同胆固醇降低方法对比分析

trt	n	M ± SD	F	p-value	η_p^2	Sig-Diff Groups
1time	10	5.78 ± 2.88				a
2times	10	9.22 ± 3.48				ab
4times	10	12.37 ± 2.92	32.433	<0.001	0.742	bc
drugD	10	15.36 ± 3.45				c
drugE	10	20.95 ± 3.35				d

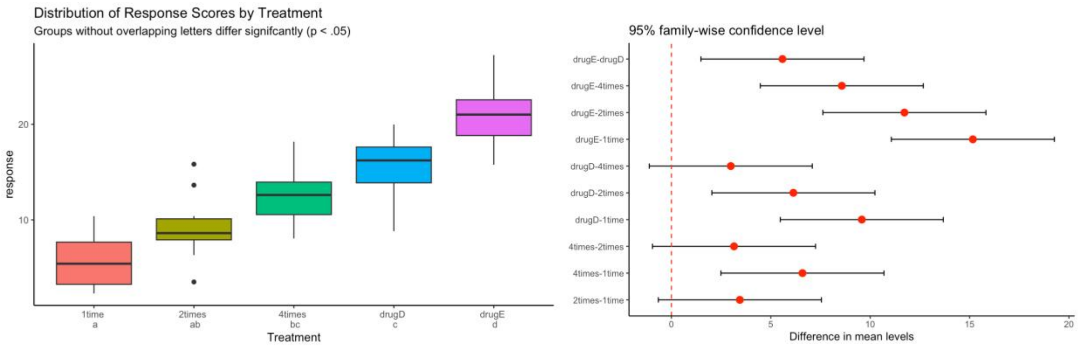

图 1. 不同胆固醇降低方法对比分析

对新研药物在不同给药频次（1time、2times、4times）下与对照药物（drugD、drugE）降低胆固醇效果进行比较，经单因素方差分析结果（表 1 及图 1）显示，五种治疗方式的组间效应显著（F=32.433，p<0.001，η_p^2=0.742），表明治疗方式对胆固醇变化具有较大的效应量并在统计意义之外亦具有实际意义。

经事后多重比较（TukeyHSD）结果显示：1time 给药方案的胆固醇降低幅度显著低于 drugD 与 drugE，表明低频给药方案在疗效上明显弱于两种对照药物；2times 给药方案在胆固醇降低效果上与 drugD 之间差异未达到显著水平，但显著低于 drugE，提示该给药频次在疗效上已接近 drugD，但仍与疗效较强的对照药物存在一定差距；4times 给药方案与 drugD 之间差异不显著，且其与 drugE 之间的差距较低频给药方案明显缩小，说明提高给药频次在一定程度上可增强新研药物的降胆固醇效果，并缩小其与对照药物之间的疗效差异。

综合比较三种给药频次可见，4times 给药方案在统计意义和实际效果上最接近对照药物的疗效水平，其次为 2times 给药方案，而 1time 给药方案与对照药物之间的疗效差距最大。

表 1. 不同胆固醇降低方法对比分析

Group	M ± SD	F	Sig.	Partial-Eta Squared	Multiple Comparisons		
					Between-Group	Cohen's d[1]	95% (LLCI,ULCI)[2]
1time	5.78 ± 2.88				1time-4times***	2.042	(1.060,3,005)
2times	9.22 ± 3.48				1time-drugD***	2.968	(1.888,4.026)
					1time-drugE***	4.700	(3.382,5.994)
4times	12.37 ± 2.92	32.433	<0.001	0.742	2times-drugD***	1.902	(0.933,2.854)
drugD	15.36 ± 3.45				2times-drugE***	3.634	(2.470,4.775)
					4times-drugE***	2.658	(1.614,3.681)
drugE	20.95 ± 3.35				drugD-drugE***	1.732	(0.777,2.670)

Note: [1]: Cohen's d = $(M_i - M_j)$ / Sqrt(MSE), MSE(Mean Square Error) is the pooled unbiased estimator of the common population variance σ^2 across all treatment groups under the assumption of homoscedasticity.; [2]: R Package: MBESS; ***: $p<0.001$.

Appendix:

```
library(MBESS)
mse <- 10.417
df_error <- 45
m_i <- 20.95
m_j <- 15.36
n_i <- 10
n_j <- 10
```

```r
get_d_ci_unbalanced <- function(m_i, m_j, n_i, n_j, mse, df_error, conf.level = 0.95) {
  d_val <- (m_i - m_j) / sqrt(mse)
  t_val <- d_val * sqrt((n_i * n_j) / (n_i + n_j))
  nct_ci <- conf.limits.nct(
    t.value = t_val,
    df = df_error,
    conf.level = conf.level
  )
  conversion_factor <- sqrt((n_i * n_j) / (n_i + n_j))
  d_lower <- nct_ci$Lower.Limit / conversion_factor
  d_upper <- nct_ci$Upper.Limit / conversion_factor
  return(list(
    d = d_val,    ci_lower = d_lower,    ci_upper = d_upper,    t = t_val
  ))
}
result1 <- get_d_ci_unbalanced(m_i, m_j, n_i, n_j, mse, df_error)
result_row <- data.frame(m1 = m_i,m2 = m_j,n1 = n_i,n2 = n_j,
  d = round(result1$d, 3),
  ci_lower = round(result1$ci_lower, 3),
  ci_upper = round(result1$ci_upper, 3),
  ci_text = sprintf("[%.3f, %.3f]", round(result1$ci_lower, 3), round(result1$ci_upper, 3)),
  t_value = round(result1$t, 3),
  stringsAsFactors = FALSE
)
sprintf("d=%.3f, 95%% CI=[%.3f, %.3f]",
                result1$d, result1$ci_lower, result1$ci_upper)
```