Package 'RougeLM'

August 7, 2026

Title Data Accompanying the Book ``The Rogue's Guide to Linear Models''

Version 1.0.0

Description Datasets and utilities for teaching linear models in the context of the BetaBit universe (StatPunk). The package provides simulated datasets based on the fictional LifeCalc algorithmic scoring system, illustrating concepts such as simple regression, ANOVA, ANCOVA, hierarchical models, multicollinearity, model selection (AIC, BIC), and regularisation (LASSO, Ridge). Each dataset is accompanied by a narrative context connecting statistical methodology to questions of algorithmic fairness and social consequence.

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports ggplot2, patchwork

Suggests MASS, effectsize, segmented, glmnet, car, emmeans, lme4, testthat (>= 3.0.0), knitr, rmarkdown, pkgdown

VignetteBuilder knitr

License MIT + file LICENSE

URL https://github.com/BetaAndBit/RougeLM

BugReports https://github.com/BetaAndBit/RougeLM/issues

NeedsCompilation no

Author Przemyslaw 'Prem' Biecek [aut, cre],
Bit Data [aut],
Beta Data [aut]

Maintainer Przemyslaw 'Prem' Biecek <przemyslaw.biecek@gmail.com>

Repository CRAN

Date/Publication 2026-08-07 16:10:17 UTC

Contents

`curiosity` *CuriosityScore before and after exposure to corporate AI agents*

Description

A simulated dataset of 1,200 WaszKrak residents measured on `CuriosityScore` before and after six months of interaction with one of four autonomous AI agents deployed by the major corporations of WaszKrak. The dataset is designed to illustrate one-way analysis of variance (ANOVA), post-hoc comparisons, and the interpretation of group differences in the context of algorithmic behaviour modification.

What is CuriosityScore?:

CuriosityScore is a continuous behavioural index (0–100, approximately Gaussian) measuring the frequency of spontaneous information-seeking outside agent-recommended content: unsolicited queries, searches outside the recommended feed, contacts initiated with people outside the algorithmically suggested network, and clicks on content the agent did not surface. It is derived passively from system logs and updated weekly by LifeCalc.

Higher values indicate greater epistemic autonomy. Lower values indicate greater dependence on agent-curated information. The population baseline in WaszKrak is approximately 51.3 points (SD = 11.4).

The four agents:

Each resident in the dataset was assigned to exactly one of four corporate AI agents for the six-month observation period:

Agent	Corporation	Primary function
`QuantumCorp`	QuantumCorp	Productivity assistant, resource allocation
`NeuroFrame`	NeuroFrame Entertainment	Social companion, content curator
`SynBio`	SynBio	Health advisor, TierCare navigator
`DataSec`	DataSec Industries	Security assistant, privacy manager

The story:

The dataset arrived without a sender. Four clean files, stripped of metadata, dropped into Beta and Bit's onion inbox. Someone on the inside was paying attention.

Beta ran the one-way ANOVA before she made coffee. The F-statistic was significant. All four corporate agents reduced CuriosityScore below the population baseline — but by different amounts, and the differences between corporations were themselves significant after post-hoc adjustment.

QuantumCorp's agent produced the largest reduction: 17 points below baseline. The others ranged from 7 to 11 points below. The corporations had not coordinated. They had simply arrived at the same optimum through independent optimisation: curious users are unpredictable for QuantumCorp, disloyal for NeuroFrame, questioning for SynBio, and evasive for DataSec. Four different problems. One direction.

Neither of them spoke for a while.

"They're not talking to each other," Beta said finally.

"No," Bit agreed.

"You don't need to coordinate when the problem has one solution."

Statistical design:

The one-way ANOVA tests whether mean `CuriosityScore` after six months (`after6msc`) differs across the four agent groups, controlling for baseline (`baseline`). The primary model is:

```
model <- aov(after6msc ~ agent, data = curiosity)
```

Post-hoc Tukey HSD comparisons reveal which specific pairs of corporations differ significantly. The change score `after6msc - baseline` is used to assess the net effect of each agent after controlling for individual differences in starting CuriosityScore.

Group means (approximate):

Agent	Baseline	After 6 months	Change
QuantumCorp	51.1	34.1	-17.0
SynBio	51.4	40.4	-11.0
DataSec	51.2	43.0	-8.2
NeuroFrame	51.3	45.3	-6.0

Usage

```
curiosity
```

Format

A data frame with 1,200 rows and 3 variables:

agent Factor with 4 levels: `"QuantumCorp"`, `"NeuroFrame"`, `"SynBio"`, `"DataSec"`. Indicates which corporate AI agent the resident interacted with during the six-month observation period. Assignment was not random — residents were matched to agents based on their district and LifeContract status — but the dataset is balanced at 300 observations per group.

baseline Numeric (0–100). CuriosityScore measured at the start of the observation period, before any agent interaction. Baseline scores do not differ significantly across agent groups (by design), allowing clean between-group comparisons of post-exposure scores. Population mean = 51.3, SD = 11.4.

after6msc Numeric (0–100). CuriosityScore measured after six months of interaction with the assigned corporate agent. All four groups show a reduction from baseline; the magnitude of reduction differs significantly across agents. The difference `after6msc - baseline` is the net agent effect for each resident.

Source

Simulated dataset generated by `data-raw/generate_curiosity.R`. The data structure is based on the chapter *"The Same Direction"* in *Equations from District 7: A Practical Guide to Linear Models* (BetaBit StatPunk universe). All values are fictional.

See Also

- curiosity_quantum for the piecewise dose-response dataset (minimum effective dose analysis) for QuantumCorp's agent specifically
- lifecalc for the full LifeCalc social scoring dataset
- medical for the simple regression dataset
- `vignette("one-way-anova", package = "RougeLM")` for a worked example

Examples

```
data(curiosity)

# Group means
aggregate(after6msc ~ agent, data = curiosity, FUN = mean)

# Change scores
curiosity$change <- curiosity$after6msc - curiosity$baseline
aggregate(change ~ agent, data = curiosity, FUN = mean)

# One-way ANOVA
model <- aov(after6msc ~ agent, data = curiosity)
summary(model)

# Post-hoc Tukey comparisons
TukeyHSD(model)

# Or with emmeans
```

```
if (requireNamespace("emmeans", quietly = TRUE)) {
  library(emmeans)
  emm <- emmeans(model, ~ agent)
  pairs(emm, adjust = "tukey")
  plot(emm, comparisons = TRUE)
}

# Visualise distributions
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  ggplot(curiosity,
         aes(x = agent, y = after6msc, fill = agent)) +
    geom_boxplot(alpha = 0.7) +
    geom_hline(yintercept = mean(curiosity$baseline),
               linetype = "dashed", colour = "grey40") +
    scale_fill_manual(values = c(
      "QuantumCorp" = "#c4521a",
      "SynBio"      = "#a8d4f5",
      "DataSec"     = "#e8b84b",
      "NeuroFrame"  = "#9FE1CB"
    )) +
    labs(title    = "CuriosityScore after 6 months by agent",
         subtitle = "Dashed line = population baseline (51.3)",
         x = NULL, y = "CuriosityScore") +
    theme_minimal() +
    theme(legend.position = "none")
}
```

`curiosity_quantum` *CuriosityScore dose-response for QuantumCorp's autonomous agent*

Description

A simulated dataset of 1,012 WaszKrak residents who interacted with QuantumCorp's autonomous AI agent over a six-month period. For each resident the dataset records their weekly interaction frequency with the agent (`dose`) and the change in `CuriosityScore` over the observation period (`drop`). The dataset is designed to illustrate piecewise (segmented) regression, minimum effective dose analysis, and the detection of a behavioural threshold below which no effect is observed.

The story:

The dataset arrived from the same anonymous source as the four-corporation ANOVA data — same metadata stripping, same clean headers, same feeling that someone on the inside was paying attention.

This time it was simpler. Two variables. One thousand and twelve users of QuantumCorp's autonomous agent over six months.

Beta ran the scatter plot first. Old habit.

The plot was not linear. It was not even smoothly curved. It was *broken*. Below a certain frequency threshold, the points scattered randomly around zero — some up, some down, no pattern, no trend. Normal noise. The kind of variation you would expect from life. But somewhere around

one interaction per week, the plot changed character entirely. Above that threshold, every single point was negative.

"There's a breakpoint," Bit said.

"Around 0.9 interactions per week," Beta said. She was already fitting the piecewise model. "Below it: slope not different from zero. Flat. No effect. Above it—"

The estimate came back.

Below threshold: $\hat{\beta} = -0.31$, p = 0.44. Effectively zero.

Above threshold: $\hat{\beta} = -4.2$ per additional weekly interaction, p < 0.001.

"They found the minimum effective dose," Bit said.

Once a week. Casual enough to feel like nothing. Precise enough to work.

Minimum effective dose:

In pharmacology, the minimum effective dose (MED) is the smallest dose required to produce a measurable effect. Here the "dose" is interaction frequency with QuantumCorp's agent, and the "effect" is the reduction in CuriosityScore. The MED is approximately **0.9 interactions per week** — just below once weekly.

The threshold was not set at twice a week or daily use. It was set at the frequency that feels casual and natural — like checking the news, like asking a question you would have looked up anyway. The agent's design was not accidental. You do not find a threshold this clean by accident.

Statistical design:

The piecewise regression model fits two separate linear segments joined at an estimated breakpoint x_0:

$$\text{drop}_i = \begin{cases} \alpha_1 + \beta_1 \cdot \text{dose}_i + \varepsilon_i & \text{if dose}_i < x_0 \\ \alpha_2 + \beta_2 \cdot \text{dose}_i + \varepsilon_i & \text{if dose}_i \geq x_0 \end{cases}$$

The breakpoint x_0 can be estimated by the **segmented** package or by grid search over candidate breakpoints minimising RSS. Below the breakpoint, β_1 is not significantly different from zero (no effect). Above it, $\beta_2 \approx -4.2$ (each additional weekly interaction reduces CuriosityScore by approximately 4.2 points).

Usage

```
curiosity_quantum
```

Format

A data frame with 1,012 rows and 2 variables:

dose Numeric (0–14). Mean number of interactions with QuantumCorp's autonomous agent per week, averaged over the six-month observation period. An "interaction" is defined as any user-initiated query or agent-initiated notification that received a response within 60 seconds. Values below 0.9 show no significant relationship with `drop`; values above 0.9 show a strong negative linear relationship. The critical threshold of approximately 0.9 interactions per week corresponds to slightly less than once weekly — a frequency that feels incidental rather than habitual to most users.

drop Numeric. Change in CuriosityScore over the six-month observation period, defined as `CuriosityScore_after - CuriosityScore_before`. Negative values indicate a reduction in autonomous information-seeking behaviour. Values near zero indicate no measurable effect of agent exposure. Below the dose threshold of 0.9, `drop` is distributed approximately as $N(0, \sigma^2)$ — consistent with natural week-to-week variation. Above the threshold, `drop` is systematically negative with magnitude increasing linearly with `dose`.

Source

Simulated dataset generated by `data-raw/generate_curiosity_quantum.R`. The data structure is based on the chapter *"The Threshold"* in *Equations from District 7: A Practical Guide to Linear Models* (BetaBit StatPunk universe). All values are fictional.

See Also

- curiosity for the one-way ANOVA dataset comparing all four corporate agents
- lifecalc for the full LifeCalc social scoring dataset
- `vignette("piecewise-regression", package = "RougeLM")` for a worked example including segmented regression and MED estimation

Examples

```
data(curiosity_quantum)

# Basic summary
summary(curiosity_quantum)

# Scatter plot — the broken relationship
plot(drop ~ dose, data = curiosity_quantum,
     xlab = "Weekly interactions with QuantumCorp agent",
     ylab = "Change in CuriosityScore (after - before)",
     main = "Dose-response: QuantumCorp agent vs CuriosityScore",
     pch  = 19, col = adjustcolor("steelblue", alpha.f = 0.3))
abline(h = 0, lty = 2, col = "grey50")
abline(v = 0.9, lty = 2, col = "firebrick")

# Simple linear model — misses the threshold structure
model_linear <- lm(drop ~ dose, data = curiosity_quantum)
summary(model_linear)


# Visualise with ggplot2
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  ggplot(curiosity_quantum,
         aes(x = dose, y = drop)) +
    geom_point(alpha = 0.25, colour = "#c4521a") +
    geom_hline(yintercept = 0,   linetype = "dashed", colour = "grey50") +
    geom_vline(xintercept = 0.9, linetype = "dashed", colour = "white") +
    geom_smooth(data = ~ subset(.x, dose <  0.9),
                method = "lm", se = TRUE,
```

```
            colour = "#a8d4f5", linewidth = 1.4) +
  geom_smooth(data = ~ subset(.x, dose >= 0.9),
            method = "lm", se = TRUE,
            colour = "#c4521a", linewidth = 1.4) +
  annotate("text", x = 1.0, y = 8,
           label = "Threshold: 0.9 interactions/week",
           colour = "white", hjust = 0, size = 3.5) +
  labs(title    = "Minimum effective dose — QuantumCorp agent",
     subtitle = "Blue: no effect below threshold. Red: -4.2 pts per interaction above.",
       x = "Weekly interactions with agent (dose)",
       y = "Change in CuriosityScore (drop)") +
  theme_minimal(base_size = 13)
}
```

`employment` *EmploymentScore outcomes from the Second Chance reintegration programme*

Description

A simulated dataset of 431 District 7 and District 11 residents who participated in QuantumCorp's **Second Chance** reintegration programme. For each participant the dataset records their gender, assigned programme track, EducationScore at enrolment, and EmploymentScore after six months. The dataset is designed to illustrate Analysis of Covariance (ANCOVA), the homogeneity of regression slopes assumption and its violation, adjusted means at reference points, and the statistical detection of cream skimming.

What is EmploymentScore?:

EmploymentScore is a continuous index (0–100) computed weekly by QuantumCorp's LifeCalc engine from four layers:

Layer 1 — Contract stability (0–30 pts) Formality and continuity of the employment relationship. Permanent corporate contracts score 30; gig economy contracts are capped at 12 regardless of hours worked. Informal care work scores zero — the system cannot see what it was not trained to count.

Layer 2 — Employer score reflection (0–25 pts) Your score is a partial function of your employer's score. Working for QuantumCorp transfers 25 points; working for a District 7 cooperative transfers 4. Workers in precarious districts are penalised twice: once for their own instability, once for the instability of the only employers available to them.

Layer 3 — Continuity history (0–25 pts) Every gap in employment history is penalised and the loss persists for 24 months after re-employment. Parental leave is a gap. Recovery from illness is a gap. The layer does not model why gaps occur.

Layer 4 — Trajectory score (0–20 pts) The algorithm's forecast, based on the slope of EmploymentScore over the previous 24 months. Resources are allocated to rising trajectories. People with declining trajectories receive less at precisely the moment when more would make the largest difference.

The Second Chance programme:

Second Chance was announced by QuantumCorp in 2046 as a reintegration initiative for economically marginalised residents of Districts 7 and 11. The programme offered three tracks, officially described as equivalent:

`"Track_A"` **— Entrepreneurship** Three months of business planning support, a seed allocation of WaszKrak credits, and access to QuantumCorp's vendor registration system. The Track A viability AI was trained on 14 years of small business survival data in which survival rates were higher for businesses founded by educated men in corporate-adjacent sectors. The model learned the pattern accurately. It did not learn that the pattern was produced by the same allocation system it was now perpetuating.

`"Track_B"` **— Corporate placement** Matching to administrative and coordination roles in mid-tier companies with District 12 and District 23 contracts. The Track B placement AI was trained on eight years of corporate hiring records in which women with high EducationScore were retained in high-stability administrative roles, and men with high EducationScore were redirected toward technical positions with higher turnover.

`"Track_C"` **— Entrepreneurship support (light)** Two months of AI coaching for micro-enterprises and local trade. The coaching AI directed all participants toward the same market niches regardless of qualifications, producing low and uniform EmploymentScore outcomes across the full range of EducationScore. Track C was the control condition the programme designers did not know they had built.

The story:

The press release arrived on Beta's feed at 7am, pushed by QuantumCorp's PR algorithm with a confidence score of 94.7%.

"Second Chance: Transforming Lives in Districts 7 and 11. After six months, participants show a mean EmploymentScore gain of 18.4 points. QuantumCorp's commitment to algorithmic equity delivers measurable results."

The eighteen-point gain was real. Beta could reproduce it exactly. The numbers in the press release were arithmetically correct.

She sat with that for a moment.

Then she plotted EmploymentScore after programme against EducationScore at enrolment, coloured by track.

Track C: nearly flat. Track B: moderate slope. Track A: steep. Very steep. Every additional point of EducationScore at enrolment translated to roughly half a point of EmploymentScore after the programme.

She checked the EducationScore distribution of participants against the District 7 population. The district median EducationScore was 38.4. The programme participants' median was 61.7.

The programme had recruited from the upper quartile of education in the poorest districts. The people most likely to succeed regardless of intervention. And then it had put the most education-dependent track in front of them, and reported the results as evidence that the programme worked.

It did work. For people who were already positioned to succeed.

Cream skimming:

Cream skimming occurs when a programme selects participants who are more likely to succeed regardless of the intervention, then attributes their success to the intervention. In this dataset, Track A participants arrive with a median EducationScore of 67.1 — 28.7 points above the District 7 population median. The unadjusted mean EmploymentScore gain of 18.4 points describes their

outcomes accurately. It does not describe the outcomes the programme would have produced for a representative sample of District 7 residents.

ANCOVA adjusts for this by estimating group means at a common value of EducationScore. At the District 7 population median (38.4), the adjusted mean EmploymentScore for Track A drops from 67.3 to 42.1 — a difference of 25.2 points. At the District 7 bottom quartile (24.1), Track A produces a predicted EmploymentScore of 31.8, indistinguishable from the pre-programme baseline of 31.4.

Statistical design:

The ANCOVA model includes `education` as a continuous covariate, `program_track` and `gender` as fixed factors, and tests the interaction between the covariate and the grouping factor:

$$\text{score\_after}_i = \mu + \beta_i \cdot \text{education}_i + \alpha_j + \gamma_k + \varepsilon_i$$

The interaction `education * program_track` is significant, indicating that the slope of EducationScore on EmploymentScore differs between tracks — a violation of the homogeneity of regression slopes assumption. Track A has a steep positive slope; Track B a moderate slope; Track C a slope not significantly different from zero.

Usage

```
employment
```

Format

A data frame with 431 rows and 5 variables:

id Integer. Unique participant identifier (1 to 431). No personally identifiable information is retained. The dataset was obtained through an unprotected join key in the city's open data portal; QuantumCorp had forgotten to restrict access.

gender Factor with 2 levels: `"Female"` and `"Male"`. Self-reported gender as recorded in the LifeCalc registration system. Gender interacts with `program_track` in determining EmploymentScore outcomes: the ANCOVA model with a three-way interaction (`education * gender * program_track`) reveals that the slope of EducationScore reverses direction between genders within each track, and that the direction of this reversal itself reverses between Track A and Track B.

program_track Factor with 3 levels: `"Track_A"`, `"Track_B"`, `"Track_C"`. Assigned programme track. Track A (entrepreneurship) has the steepest slope of EducationScore on outcome, making it highly dependent on the education participants bring with them. Track C (light coaching) has a slope not significantly different from zero — outcomes are similar regardless of educational background. Track B (corporate placement) has an intermediate slope. The tracks were officially described as equivalent. They were not.

education Numeric (20–95). EducationScore at the time of programme enrolment, derived from the LifeCalc education cluster (see lifecalc). The population median in Districts 7 and 11 is 38.4; the programme participant median is 61.7, and the Track A participant median is 67.1. This systematic gap between participant and population EducationScore is the statistical signature of cream skimming. ANCOVA adjusts for this gap by estimating track means at a common EducationScore reference point.

score Numeric (0–100). EmploymentScore measured six months after programme completion, derived from the LifeCalc four-layer index described above. The unadjusted group mean for Track A is approximately 67.3. Adjusted to the District 7 population median EducationScore (38.4), the Track A adjusted mean drops to 42.1. Adjusted to the District 7 bottom quartile EducationScore (24.1), the Track A predicted score is 31.8 — indistinguishable from the pre-programme baseline of 31.4. The programme works. It was not designed for the people it was announced to serve.

Source

Simulated dataset generated by `data-raw/generate_employment.R`. The data structure is based on the chapter *"The Second Chance"* in *Equations from District 7: A Practical Guide to Linear Models* (BetaBit StatPunk universe). All values are fictional. The dataset was accessed through an unprotected join key in the WaszKrak open data portal. QuantumCorp has not been informed.

See Also

- mobility for the two-way ANOVA dataset from the same analytical arc
- lifecalc for the full LifeCalc social scoring dataset including `EmploymentScore` as one of 24 predictors
- `vignette("ancova", package = "RougeLM")` for a worked example including homogeneity of slopes testing, adjusted means at reference points, and cream skimming detection

Examples

```
data(employment)

# Unadjusted group means — what the press release reported
aggregate(score ~ program_track, data = employment, FUN = mean)

# EducationScore distributions: participants vs population median
aggregate(education ~ program_track, data = employment, FUN = median)
# District 7 population median for reference:
cat("D7 population median EducationScore: 38.4\n")

# Check homogeneity of regression slopes (should be violated)
model_homogeneity <- aov(score ~ education * program_track,
                         data = employment)
summary(model_homogeneity)
# Significant interaction -> slopes differ between tracks

# Full ANCOVA model
model_ancova <- aov(score ~ education * program_track + gender,
                    data = employment)
summary(model_ancova)

# Track-specific slopes
slopes <- lapply(levels(employment$program_track), function(track) {
  m <- lm(score ~ education,
          data = subset(employment, program_track == track))
  c(track = track, slope = round(coef(m)["education"], 3))
```

```
})
do.call(rbind, slopes)

# Adjusted means at three EducationScore reference points
if (requireNamespace("emmeans", quietly = TRUE)) {
  library(emmeans)

  # At participant median (61.7) — what QuantumCorp reported
  emmeans(model_ancova, ~ program_track,
          at = list(education = 61.7))

  # At D7 population median (38.4) — the relevant comparison
  emmeans(model_ancova, ~ program_track,
          at = list(education = 38.4))

  # At D7 bottom quartile (24.1) — the most vulnerable residents
  emmeans(model_ancova, ~ program_track,
          at = list(education = 24.1))
}

# Visualise the cream skimming effect
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)

  ggplot(employment,
         aes(x     = education,
             y     = score,
             colour = program_track)) +
    geom_point(alpha = 0.2, size = 1.5) +
    geom_smooth(method = "lm", se = TRUE, linewidth = 1.4) +
    geom_vline(xintercept = 38.4,
               linetype = "dashed", colour = "white") +
    geom_vline(xintercept = 61.7,
               linetype = "dashed", colour = "grey60") +
    geom_hline(yintercept = 31.4,
               linetype = "dotted", colour = "grey50") +
    annotate("text", x = 40, y = 90,
             label = "D7 population\nmedian (38.4)",
             colour = "white", size = 3, hjust = 0) +
    annotate("text", x = 63, y = 90,
             label = "Participant\nmedian (61.7)",
             colour = "grey60", size = 3, hjust = 0) +
    annotate("text", x = 21, y = 33.5,
             label = "Pre-programme\nbaseline (31.4)",
             colour = "grey50", size = 3, hjust = 0) +
    scale_colour_manual(values = c(
      "Track_A" = "#c4521a",
      "Track_B" = "#e8b84b",
      "Track_C" = "#a8d4f5"
    )) +
    labs(title    = "ANCOVA: EmploymentScore vs EducationScore by track",
         subtitle = "Slopes differ between tracks (homogeneity assumption violated)",
         x        = "EducationScore at enrolment",
```

```
          y        = "EmploymentScore after programme",
          colour   = "Programme track") +
      theme_minimal(base_size = 13) +
      theme(legend.position = "top")
}
```

`generate_lifecalc` *Generate a LifeCalc-style dataset*

Description

Generates a simulated dataset of WaszKrak residents scored by the fictional LifeCalc algorithm. The function reproduces the same correlation structure as the bundled `lifecalc` dataset but allows changing `n` and `seed` for experiments, teaching, or simulation studies.

The true model for `SocialScore` is sparse — dominated by `DistrictScore` and `prior_flag` — but the dataset contains 22 additional correlated proxy variables that illustrate multicollinearity and the need for regularisation.

Usage

```
generate_lifecalc(n = 5000, seed = 2047)
```

Arguments

`n`	Integer. Number of observations to generate. Default 5000.
`seed`	Integer. Random seed for reproducibility. Default 2047.

Value

A data frame with `n` rows and 25 columns. See lifecalc for full variable documentation.

Examples

```
# Reproduce the bundled dataset exactly
df <- generate_lifecalc(n = 5000, seed = 2047)
all.equal(df, lifecalc)   # TRUE

# Generate a smaller dataset for quick experiments
df_small <- generate_lifecalc(n = 500, seed = 42)
dim(df_small)

# Demonstrate that LASSO recovers the true predictors
if (requireNamespace("glmnet", quietly = TRUE)) {
  X <- model.matrix(SocialScore ~ ., data = df_small)[, -1]
  y <- df_small$SocialScore
  cv <- glmnet::cv.glmnet(X, y, alpha = 1, nfolds = 10)
  coefs <- coef(cv, s = "lambda.min")
  coefs[coefs[, 1] != 0, , drop = FALSE]
}
```

`lifecalc` *LifeCalc algorithmic scoring dataset*

Description

A simulated dataset of 5,000 WaszKrak residents scored by the fictional LifeCalc algorithm (QuantumCorp, 2047). The dataset is designed to illustrate multicollinearity, model selection, and regularisation in the context of algorithmic social scoring.

Each row represents one resident. The outcome variable `SocialScore` is generated by a sparse true model dominated by `DistrictScore`. The remaining predictors are organised into correlated clusters education, employment, health, behaviour, and personality, that introduce structured multicollinearity, mimicking a real system where every subscore feeds every other subscore in a reinforcing loop.

Correlation clusters:

Cluster	Variables
Education	`EducationScore`, `LiteracyScore`, `QuestioningScore`, `VerificationScore`
Employment	`EmploymentScore`, `NetworkScore`, `ConsumptionScore`, `MobilityScore`
Health	`MedicalScore`, `SleepScore`, `RecoveryScore`, `NutritionScore`, `StressIndex`, `GeneticRisk`
Behaviour	`ComplianceScore`, `NarrativeScore`, `RoutineScore`, `AttentionScore`, `DisplacementScore`
Personality (Big Five)	`OpennessScore`, `ConscientiousnessScore`, `ExtraversionScore`, `AgreeablenessScore`, `Ne`
Personality (facets)	`ImaginationScore`, `IntellectScore`, `OrderlinessScore`, `DutifulnessScore`, `Perfection`
Wellbeing / social cognition	`ResilienceScore`, `EmpathyScore`, `AlexithymiaScore`, `MindfulnessScore`, `SelfEfficacyS`
Dark Triad	`MacchiavelliScore`, `NarcissismScore`, `DarkTriadScore`

Usage

```
lifecalc

lifecalc_future

lifecalc_small
```

Format

A data frame with 5,000 rows and 52 variables:

SocialScore Outcome. LifeCalc master social score (0-100). Determines TierCare tier, credit access, and district reclassification eligibility.

EducationScore Continuous (5-95). Composite education index. Strongly correlated with `DistrictScore`. Anchor of the education cluster.

EmploymentScore Continuous (0-95). Algorithmic employment score (four-layer LifeCalc index). Correlated with `EducationScore` and `DistrictScore`.

DistrictScore Continuous (5-95). Geographic district quality index. Higher values correspond to wealthier districts (District 23 = 85, District 7 = 30). The single strongest predictor of `SocialScore`.

LiteracyScore Continuous (5-95). Capacity to parse algorithmic decisions and contracts. Follows `EducationScore`.

QuestioningScore Continuous (5-95). Propensity to question or challenge algorithmically generated decisions and recommendations, rather than accepting them at face value. Conceptually close to `VerificationScore` and `CuriosityScore`; follows `EducationScore`.

VerificationScore Continuous (5-95). Frequency of cross-checking agent-provided information. Follows `EducationScore`.

NetworkScore Continuous (5-95). Quality-weighted social contact index. Correlated with `EmploymentScore` and `DistrictScore`.

ConsumptionScore Continuous (5-95). Consumption pattern alignment with algorithmic profile. Correlated with `EmploymentScore` and `DistrictScore`.

MobilityScore Continuous (0-95). Algorithmic probability of district reclassification. Correlated with `DistrictScore` and `EmploymentScore`.

GeneticRiskScore Continuous (5-95). Predicted probability of costly medical conditions from genetic profile. Inversely correlated with `DistrictScore`, a selection artefact from years of training data.

NutritionScore Continuous (5-95). Weekly nutritional status index from AI health checks. Follows `DistrictScore`.

SleepScore Continuous (5-95). Sleep regularity and duration from neural implant and smart-device logs. Follows `DistrictScore` and `NutritionScore`.

StressIndex Continuous (5-95). Biomarker-derived stress index. Inversely correlated with `DistrictScore`.

RecoveryScore Continuous (5-95). Speed of return to baseline health after illness or injury. Follows `DistrictScore` and `NutritionScore`; reduced by `StressIndex`.

ChronicLoadScore Continuous (5-95). Accumulated health-system burden over five years. Inversely correlated with `DistrictScore` and `EducationScore`.

MedicalScore Continuous (5-95). Composite health outcome index. Correlated with `SleepScore`, `NutritionScore`, and `RecoveryScore`; reduced by `GeneticRiskScore` and `StressIndex`.

ComplianceScore Continuous (5-95). Alignment of daily behaviour with algorithmic recommendations. Follows `DistrictScore`; inversely related to `QuestioningScore`.

NarrativeScore Continuous (5-95). Internal coherence of beliefs and stated preferences. Follows `DistrictScore` and `LiteracyScore`.

RoutineScore Continuous (5-95). Predictability of daily patterns (routes, purchases, contacts). Follows `DistrictScore` and `ComplianceScore`.

AttentionScore Continuous (5-95). Sustained task-focus duration from productivity logs. Follows `EducationScore` and `SleepScore`; reduced by `StressIndex`.

DisplacementScore Continuous (5-95). Frequency of movement outside predicted daily range. Inversely related to `DistrictScore`; follows `MobilityScore`.

OpennessScore Continuous (5-95). Big Five domain score openness to experience: creativity, curiosity about ideas, and preference for novelty.

ConscientiousnessScore Continuous (5-95). Big Five domain score conscientiousness: self-discipline, organisation, and goal-directed behaviour.

ExtraversionScore Continuous (5-95). Big Five domain score extraversion: sociability, assertiveness, and positive affect.

AgreeablenessScore Continuous (5-95). Big Five domain score agreeableness: cooperativeness, trust, and prosocial orientation.

NeuroticismScore Continuous (5-95). Big Five domain score neuroticism: emotional instability and susceptibility to negative affect.

ImaginationScore Continuous (5-95). Facet of `OpennessScore` vividness of fantasy and imaginative engagement.

IntellectScore Continuous (5-95). Facet of `OpennessScore` intellectual curiosity and engagement with abstract ideas.

OrderlinessScore Continuous (5-95). Facet of `ConscientiousnessScore` preference for structure, tidiness, and planning.

DutifulnessScore Continuous (5-95). Facet of `ConscientiousnessScore` adherence to rules, obligations, and ethical standards.

PerfectionismScore Continuous (5-95). Facet of `ConscientiousnessScore` striving for high standards and attention to detail.

AssertivenessScore Continuous (5-95). Facet of `ExtraversionScore` tendency to take charge and express opinions directly.

WarmthScore Continuous (5-95). Facet of `ExtraversionScore` / `AgreeablenessScore` friendliness and interpersonal affection.

AltruismScore Continuous (5-95). Facet of `AgreeablenessScore` concern for others' welfare and willingness to help.

TrustScore Continuous (5-95). Facet of `AgreeablenessScore` belief in the sincerity and good intentions of others.

AnxietyScore Continuous (5-95). Facet of `NeuroticismScore` proneness to worry and nervous tension.

ImpulsivenessScore Continuous (5-95). Facet of `NeuroticismScore` difficulty controlling urges and cravings.

RuminationScore Continuous (5-95). Facet of `NeuroticismScore` tendency toward repetitive, intrusive negative thinking.

ResilienceScore Continuous (5-95). Capacity to recover emotionally from stress or adversity; inversely related to `NeuroticismScore`.

EmpathyScore Continuous (5-95). Capacity to recognise and share the emotional states of others.

AlexithymiaScore Continuous (5-95). Difficulty identifying and describing one's own emotional states.

MindfulnessScore Continuous (5-95). Tendency toward present-moment, non-judgemental awareness.

SelfEfficacyScore Continuous (5-95). Confidence in one's own ability to achieve goals and influence outcomes.

LonelinessScore Continuous (5-95). Subjective experience of social isolation, independent of actual network size.

AmbiguityToleranceScore Continuous (5-95). Comfort with uncertain, contradictory, or incomplete information.

MacchiavelliScore Continuous (5-95). Machiavellianism strategic, manipulative orientation toward others; component of the Dark Triad.

NarcissismScore Continuous (5-95). Grandiosity, entitlement, and need for admiration; component of the Dark Triad.

DarkTriadScore Continuous (5-95). Composite Dark Triad index combining Machiavellianism, narcissism, and psychopathy-related traits.

CuriosityScore Continuous (5-95). Frequency of spontaneous information-seeking outside agent recommendations. Follows `EducationScore`; reduced by agent exposure.

AdaptabilityScore Continuous (5-95). Flexibility in adjusting behaviour and expectations to changing circumstances.

SocialComplianceScore Continuous (5-95). Conformity to broader social norms and expectations, distinct from `ComplianceScore` (which is specific to algorithmic recommendations).

An object of class `data.frame` with 1000 rows and 52 columns.

An object of class `data.frame` with 50 rows and 52 columns.

Source

Simulated dataset generated by `data-raw/generate_lifecalc.R`. All values are fictional. The dataset is designed for pedagogical use in the context of the BetaBit StatPunk universe.

Examples

```
data(lifecalc)

# Basic summary
summary(lifecalc$SocialScore)

# Correlation of all variables with SocialScore
cors <- cor(lifecalc)[, "SocialScore"]
sort(abs(cors), decreasing = TRUE)

# OLS model  all predictors
model_full <- lm(SocialScore ~ ., data = lifecalc)
summary(model_full)

# Check multicollinearity
if (requireNamespace("car", quietly = TRUE)) {
  car::vif(model_full)
}

# LASSO variable selection
if (requireNamespace("glmnet", quietly = TRUE)) {
  X <- model.matrix(SocialScore ~ ., data = lifecalc)[, -1]
  y <- lifecalc$SocialScore
  cv_fit <- glmnet::cv.glmnet(X, y, alpha = 1)
  coef(cv_fit, s = "lambda.min")
}
```

lifecalc_cor_clusters *Correlation cluster summary for a LifeCalc-style dataset*

Description

Returns a tidy data frame of pairwise correlations for the predefined variable clusters in the `lifecalc` dataset. Useful as a quick diagnostic before fitting multivariate models — high within-cluster correlations signal multicollinearity that will inflate VIF and destabilise OLS coefficient estimates.

Usage

```
lifecalc_cor_clusters(data = lifecalc)
```

Arguments

`data`	A data frame with the same column names as lifecalc. Defaults to the bundled `lifecalc` dataset.

Value

A data frame with columns `cluster`, `var1`, `var2`, and `r` (Pearson correlation), sorted by descending `|r|` within each cluster.

Examples

```
data(lifecalc)
clusters <- lifecalc_cor_clusters(lifecalc)
head(clusters, 10)

# High correlations within the education cluster
subset(clusters, cluster == "education")
```

lifecalc_vif *Variance Inflation Factors for the full LifeCalc OLS model*

Description

Fits `SocialScore ~ .` on the supplied data frame and returns Variance Inflation Factors (VIF) for every predictor, together with a severity label. VIF > 10 indicates severe multicollinearity; VIF > 5 indicates moderate multicollinearity.

This is a convenience wrapper around `car::vif()` that adds the severity classification and sorts the output from highest to lowest VIF.

Usage

```
lifecalc_vif(data = lifecalc)
```

Arguments

`data`	A data frame with the same column names as lifecalc. Defaults to the bundled `lifecalc` dataset.

Value

A data frame with columns `variable`, `vif`, and `severity` (`"severe"`, `"moderate"`, or `"acceptable"`), sorted by descending VIF.

Examples

```
data(lifecalc)
vif_df <- lifecalc_vif(lifecalc)
print(vif_df)

# How many variables have severe multicollinearity?
sum(vif_df$severity == "severe")
```

`medical` *medical and healthcare expenditure for LifeContract pairs*

Description

A simulated dataset of 96 LifeContract pairs registered in WaszKrak megapolis, containing the medical of each partner and their joint annual healthcare expenditure in WaszKrak credits. The dataset is designed to illustrate simple linear regression, the asymmetry of the regression model, confidence and prediction intervals, and the Box-Cox transformation for non-linear relationships.

What is a LifeContract?:

In WaszKrak (2047), the legal institution of marriage was replaced by the algorithmically generated **LifeContract** — a resource-sharing agreement between two registered citizens, optimised by QuantumCorp's LifeCalc engine for joint healthcare allocation, credit access, and district reclassification eligibility. A LifeContract can be entered into with any other citizen: a friend, a colleague, a neighbour. Standard durations are 3, 5, or 10 years with an option to renew.

TierCare treats LifeContract Partners as a single allocation unit. Their medicals are combined into a **Shared Health Index** — a weighted average that determines the pair's joint priority in the healthcare queue.

Contract Partners A and B:

The contract designates two roles: **Contract Partner A** and **Contract Partner B**. Officially, the assignment is described as arbitrary. In practice, analysis of registration records shows that Partner A is the individual with the higher medical at time of signing in 94.3% of cases. The Shared Health Index weights Partner A's medical at 0.62 and Partner B's at 0.38 — a fact encoded in the TierCare allocation model but absent from the contract documentation.

The contract is nominally symmetric. The algorithm is not.

What is medical?:

medical is a continuous index (0–200, approximately Gaussian) computed weekly by SynBio's TierCare diagnostic system from neural implant telemetry, biochemical markers, genetic risk profile, and historical healthcare utilisation. Higher values indicate better predicted health outcomes and lower expected healthcare costs. medical determines eligibility for TierCare treatment tiers, access to SynthOrgan transplants, and — through the Shared Health Index — the pair's joint allocation priority.

Statistical design:

The dataset was used to investigate the relationship between the medicals of the two partners within a LifeContract pair. The analysis proceeds in two stages:

Stage 1 — Partner medical correlation and simple regression. The medicals of Partner A and Partner B are positively correlated (r = 0.74): pairs tend to form within the same district, socioeconomic bracket, and statistical future. Simple regression of Partner B's medical on Partner A's medical — or vice versa — illustrates that the choice of response variable is not symmetric, even when the underlying relationship is.

Stage 2 — Healthcare expenditure and Box-Cox transformation. Annual healthcare expenditure (`health_expenses`) is regressed on medical. The relationship is not linear: expenditure explodes at the low end of medical and flattens at the high end, producing a curve that pretends to be a line. A Box-Cox profile log-likelihood identifies lambda close to zero, justifying a log transformation of the response. The transformed model is well-behaved; back-transformed predictions reveal the exponential gap between low- and high-medical citizens.

Usage

```
medical
```

Format

A data frame with 96 rows and 3 variables:

partner_a Numeric. medical of Contract Partner A at the time of the most recent TierCare assessment (0–200 scale, approximately Gaussian with mean = 170 and SD = 12). Partner A is, in 94.3% of registered pairs, the individual with the higher medical at contract signing. medical reflects predicted health outcomes and determines the pair's joint TierCare priority through the Shared Health Index (weight 0.62 for Partner A).

partner_b Numeric. medical of Contract Partner B at the time of the most recent TierCare assessment (0-200 scale). Partner B typically has a lower medical than Partner A within the same pair (weight 0.38 in the Shared Health Index). The positive correlation between `partner_a` and `partner_b` (r = 0.74) reflects assortative pairing within districts and socioeconomic brackets, amplified by the algorithmic incentive structure of the LifeContract registration system.

health_expenses Numeric. Joint annual healthcare expenditure for the pair in WaszKrak credits (strictly positive, heavy right tail). Expenditure is driven primarily by Partner B's medical — the lower-scoring partner generates the majority of healthcare costs. The relationship between medical and expenditure is non-linear: a Box-Cox transformation with lambda = 0 (log transformation) is required to satisfy the assumptions of linear regression. After log-transformation, the model reads: $\ln(\text{expenditure}) = 8.14 - 0.043 \times \text{medical}$.

Source

Simulated dataset generated by `data-raw/generate_medical.R`. The data structure is based on the chapters *"Strong Enough"* and *"The Shape of Cost"* in *Equations from District 7: A Practical Guide to Linear Models* (BetaBit StatPunk universe). All values are fictional.

References

Box, G. E. P. and Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society*, Series B, **26**, 211–252.

See Also

- lifecalc for the full LifeCalc social scoring dataset
- nutrition for the nested ANOVA dataset
- `vignette("simple-regression", package = "RougeLM")` for a worked example of simple regression and Box-Cox transformation

Examples

```
data(medical)

# Basic summary
summary(medical)

# Partner medical correlation
cor(medical$partner_a, medical$partner_b)

# Scatter plot of partner medicals
plot(partner_b ~ partner_a, data = medical,
     xlab = "Partner A medical",
     ylab = "Partner B medical",
     main = "LifeContract pair medicals (n = 96)")

# Simple regression: Partner A predicts Partner B
model1 <- lm(partner_b ~ partner_a, data = medical)
summary(model1)

# Prediction interval for Partner B when Partner A = 170
predict(model1,
        newdata  = data.frame(partner_a = 170),
        interval = "prediction",
        level    = 0.95)

# Note: regression is not symmetric
model2 <- lm(partner_a ~ partner_b, data = medical)
coef(model1)  # slope not 1 / coef(model2)["partner_b"]

# Healthcare expenditure: non-linear relationship
plot(health_expenses ~ partner_b, data = medical,
     xlab = "Partner B medical",
     ylab = "Healthcare expenditure (WaszKrak credits)",
```

```
     main = "Expenditure vs medical — curve pretending to be a line")

# Box-Cox transformation
if (requireNamespace("MASS", quietly = TRUE)) {
  model_raw <- lm(health_expenses ~ partner_b, data = medical)
  bc <- MASS::boxcox(model_raw, plotit = FALSE)
  lambda_opt <- bc$x[which.max(bc$y)]
  cat("Optimal lambda:", round(lambda_opt, 3), "\n")

  # Log transformation (lambda = 0)
  model_log <- lm(log(health_expenses) ~ partner_b, data = medical)
  summary(model_log)

  # Back-transformed predictions
  scores <- c(30, 50, 80)
  log_pred <- predict(model_log,
                      newdata = data.frame(partner_b = scores))
  data.frame(partner_b = scores,
             predicted_expenses = round(exp(log_pred), 0))
}
```

`mobility` *MobilityScore by district, gender and age group in WaszKrak*

Description

A simulated dataset of 412 District 7 residents measured on `MobilityScore` — the algorithmic metric determining likelihood of district reclassification — broken down by gender and age group. The dataset is designed to illustrate two-way analysis of variance (ANOVA) with an interaction term, estimated marginal means, pairwise contrasts, and the interpretation of a disordinal interaction.

What is MobilityScore?:

MobilityScore is a continuous index (0–100, approximately Gaussian) computed weekly by QuantumCorp's LifeCalc engine. It represents the algorithmically estimated probability that a resident will change their registered district within the next 12 months. Higher values indicate greater predicted mobility and unlock access to district reclassification applications, retraining programme eligibility, and cross-district employment contracts.

MobilityScore is not a neutral measure. It is trained on 14 years of historical movement data from WaszKrak districts — data in which movement patterns were themselves shaped by economic opportunity, algorithmic resource allocation, and structural inequality. The model does not predict who *will* move. It predicts who moved in the past, under conditions that no longer exist, and projects that pattern forward as if it were destiny.

The story:

The request came from Marek Kowalski — a District 7 resident whose daughter had been approved for mobility reclassification while he had not, despite living in the same building, in the same district, with broadly similar circumstances. He wanted to understand why the same district produces different people.

Three days after Beta wrote back, 412 records arrived — neighbours, friends, relatives, people who had passed Marek's message along the corridor and down the stairs of a building where the elevator had been broken since 2044.

Beta ran the descriptives first. The district mean was 31.4, SD 11.2. Roughly Gaussian, slightly left-skewed. She split by gender. Men: 32.1. Women: 30.8. Difference of 1.3 points. The t-test returned p = 0.26. Not significant.

She split by age group instead. Under 35: 35.7. Thirty-five and over: 27.9. That was significant.

Then she built the two-way ANOVA. Gender and age group as factors. Gender alone: not significant. Age alone: highly significant. But the interaction — gender times age group — was almost as strong as the age effect itself.

She pulled the estimated marginal means.

Young women: 38.9. Young men: 32.4. Older women: 22.7. Older men: 33.1.

The lines didn't just diverge. They crossed.

"Young women are more mobile," Bit said slowly, working through it. "Because the algorithm sees them moving — for work, for partners, for whatever reason the training data said young women from poor districts move. It rewards the pattern it already knows."

"And then they stop moving," Beta said. "Or they move differently. For themselves, not for the patterns the model was trained on. And the algorithm stops seeing it as mobility."

"It's not predicting the future," Beta said. "It's photocopying the past."

The disordinal interaction:

The interaction between `gender` and `age` is **disordinal**: the ranking of gender groups reverses between age categories. Among residents under 35, women have higher MobilityScore than men. Among residents 35 and over, men have higher MobilityScore than women. This reversal is visible as crossing lines in the interaction plot and is the strongest possible form of interaction — it means that no single statement about the effect of gender is true across all age groups simultaneously.

The gender main effect, examined in isolation, is not significant (F(1, 408) = 1.42, p = 0.234). A researcher stopping at main effects would conclude that gender plays no role in MobilityScore in District 7. This conclusion would be precisely wrong.

Key results (approximate):

Group	Estimated marginal mean	95% CI
Female, young	38.9	37.3, 40.5
Male, young	32.4	30.8, 34.0
Female, older	22.7	21.3, 24.1
Male, older	33.1	31.7, 34.5

The contrast female older vs male older: estimate = -10.4, t(408) = -10.30, p < .0001. The system does not freeze men in place as they age. It freezes women.

Usage

```
mobility
```

Format

A data frame with 412 rows and 4 variables:

score Numeric (0–100). MobilityScore at the time of measurement, derived from QuantumCorp's LifeCalc engine. Represents the algorithmically estimated probability of district change within 12 months, expressed as a 0–100 index. The District 7 population mean is approximately 31.4 (SD = 11.2). Values above 48 are required for TierCare Tier 2 reclassification eligibility. The distribution is approximately Gaussian with a slight left skew, reflecting the structural floor imposed by prior_flag and district assignment.

district Character. District identifier. All observations in this dataset are from `"D7"` (District 7, formerly Grójec — the district where Beta and Bit live and work). The variable is retained for compatibility with multi-district datasets and to allow merging with lifecalc.

gender Factor with 2 levels: `"female"` and `"male"`. Self-reported gender as recorded in the LifeCalc registration system. The gender main effect on MobilityScore is not significant when examined alone ($p = 0.234$), but is strongly moderated by age group — the interaction term is highly significant ($F(1, 408) = 69.18$, $p < 0.001$), producing a disordinal crossing of group means.

age Factor with 2 levels: `"young"` (under 35) and `"older"` (35 and over). The cut point of 35 corresponds to the approximate age at which the LifeCalc model's treatment of gender reverses direction, as identified through the interaction plot. Below 35, young women have higher MobilityScore than young men — because the training data shows young women from poor districts moving for work and partnership, patterns the algorithm rewards. Above 35, older women have substantially lower MobilityScore than older men — because their movement history is dominated by care work and gig employment, both of which the algorithm either cannot see or classifies as instability rather than mobility.

Source

Simulated dataset generated by `data-raw/generate_mobility.R`. The data structure is based on the chapter *"The Crossing"* in *Equations from District 7: A Practical Guide to Linear Models* (BetaBit StatPunk universe). All values are fictional. The dataset was constructed from 412 records shared by District 7 residents at the request of Marek Kowalski, mathematics teacher, father, and the man whose building elevator has been broken since 2044.

See Also

- curiosity for the one-way ANOVA dataset
- lifecalc for the full LifeCalc social scoring dataset including `MobilityScore` as one of 24 predictors
- `vignette("two-way-anova", package = "RougeLM")` for a worked example including interaction plots, emmeans, and disordinal interaction interpretation

Examples

```
data(mobility)

# District mean and SD
mean(mobility$score)
```

```
sd(mobility$score)

# Main effect of gender alone — not significant
t.test(score ~ gender, data = mobility)

# Two-way ANOVA with interaction
model <- aov(score ~ gender * age, data = mobility)
summary(model)

# Estimated marginal means — the crossing is visible here
if (requireNamespace("emmeans", quietly = TRUE)) {
  library(emmeans)
  emm <- emmeans(model, ~ gender * age)
  emm

  # Interaction plot
  emmip(model, gender ~ age, CIs = TRUE)

  # Pairwise contrasts
  contrast(emm, interaction = "pairwise")
}

# Effect sizes
if (requireNamespace("effectsize", quietly = TRUE)) {
  effectsize::eta_squared(model, partial = TRUE)
}

# Visualise the disordinal interaction
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)

  # Raw data with group means
  group_means <- aggregate(score ~ gender + age,
                           data = mobility, FUN = mean)

  ggplot(mobility,
         aes(x = age, y = score,
             colour = gender, group = gender)) +
    geom_jitter(alpha = 0.2, width = 0.08) +
    stat_summary(fun = mean, geom = "line",  linewidth = 1.4) +
    stat_summary(fun = mean, geom = "point", size = 4) +
    geom_hline(yintercept = 31.4,
               linetype = "dotted", colour = "grey50") +
    scale_colour_manual(values = c(
      "female" = "#c4521a",
      "male"   = "#a8d4f5"
    )) +
    scale_x_discrete(labels = c("young" = "Under 35",
                                "older" = "35 and over")) +
    annotate("text", x = 1.5, y = 29,
             label = "District 7 mean (31.4)",
             colour = "grey50", size = 3.5) +
    labs(title    = "MobilityScore: Gender \u00d7 Age Group",
```

```
        subtitle = "District 7, n = 412. Lines cross \u2014 disordinal interaction.",
        x        = "Age group",
        y        = "MobilityScore",
        colour   = "Gender") +
    theme_minimal(base_size = 13) +
    theme(legend.position = "top")
}
```

`nutrition` *NutritionScore monitoring across WaszKrak schools*

Description

A simulated dataset of weekly NutritionScore measurements collected from 41 schools across three districts of WaszKrak megapolis (Districts 7, 12, and 23) over 26 weeks. The dataset is designed to illustrate nested (hierarchical) ANOVA with fixed effects, post-hoc comparisons, and the detection of a single outlier school.

NutritionScore is a composite index (0–100) derived from weekly AI health checks performed on students — measuring weight indicators, biochemical markers, and energy levels. It is updated every Friday by the SynBio TierCare diagnostic system.

The story:

The dataset originates from a tip sent by Tomasz Bernat, a mathematics teacher at School 4 in District 12, who noticed that his students were unable to concentrate and were falling asleep by 10am. The school nutritionist reported that NutriFirst program scores were within normal range. Bernat did not believe the scores.

Beta and Bit obtained a leaked export of the school monitoring system covering all 41 schools. The nested ANOVA revealed that School 4 — the only school enrolled in the **NutriFirst** program (SynBio batch `SB-2046-NF-07`) — scored more than 20 points below every other school in District 12, and below the District 7 mean, despite District 12 being a substantially wealthier district.

A data entry error by a procurement clerk had assigned School 4 to a cartridge batch intended exclusively for districts with MedScore below 55. The batch had reduced protein bioavailability. The error was statistically visible. The students' fatigue was not a mystery. It was a data point.

Statistical design:

Schools are nested within districts — School 4 in District 12 is a different entity from School 4 in District 7. Both district and school are treated as **fixed effects** (not random), because the analysis concerns these specific schools and districts, not schools and districts in general.

The nested model is:

```
NutritionScore ~ district + district:school
```

which in R notation is equivalent to `aov(Score ~ district/school)`.

The key finding is a significant `school(district)` term, driven entirely by School 4 in District 12. Post-hoc Tukey comparisons confirm that School 4 differs significantly from every other District 12 school (all adjusted $p < 0.001$).

District means (approximate):

District	n schools	Mean Score	SD
D7	11	54.1	3.2
D12	19	66.8	9.1
D23	11	79.3	3.4

The anomalously large SD for District 12 is the first signal that something is wrong — it is more than twice the SD of either other district.

Usage

```
nutrition
```

Format

A data frame with 1,066 rows and 5 variables:

school Character. School identifier, e.g. `"D12_S4"`. Format: district prefix + underscore + `S` + school number within district. Schools are uniquely identified within districts — `D7_S1` and `D12_S1` are different schools.

week Integer (1–26). Observation week within the 26-week monitoring period. Week 1 corresponds to the start of the academic term. NutritionScore is recorded once per week per school.

district Factor with 3 levels: `"D7"`, `"D12"`, `"D23"`. Ordered by socioeconomic status: D7 (lowest) < D12 (middle) < D23 (highest). District determines baseline NutritionScore through access to food infrastructure, healthcare, and algorithmic resource allocation.

nutrifirst Logical. `TRUE` if the school is enrolled in SynBio's NutriFirst synthetic food programme (cartridge batch `SB-2046-NF-07`). Only one school in the dataset has `nutrifirst = TRUE`: School 4 in District 12 (`"D12_S4"`). This school was assigned to the batch by a data entry error — the clerk mistyped the district code. The batch is otherwise distributed exclusively to schools in districts with MedScore below 55.

score Numeric (0–100). Weekly NutritionScore for the school, averaged across all students in that school for that week. Derived from SynBio TierCare AI health checks. Higher values indicate better nutritional status. The true school mean is stable across weeks; within-school week-to-week variation reflects natural measurement noise (SD = 4 points).

Source

Simulated dataset generated by `data-raw/generate_nutrition.R`. The data structure is based on the chapter *"The Outlier"* in *Equations from District 7: A Practical Guide to Linear Models* (BetaBit StatPunk universe). All values are fictional.

See Also

- lifecalc for the full LifeCalc social scoring dataset
- `vignette("nested-anova", package = "RougeLM")` for a worked example

Examples

```
data(nutrition)

# District-level summary
aggregate(score ~ district, data = nutrition, FUN = mean)
aggregate(score ~ district, data = nutrition, FUN = sd)

# School-level means
school_means <- aggregate(score ~ district + school + nutrifirst,
                          data = nutrition, FUN = mean)
school_means[order(school_means$score), ]

# Identify the outlier school
school_means[school_means$nutrifirst == TRUE, ]

# Nested ANOVA: schools nested within districts, both fixed effects
model <- aov(score ~ district + district:school,
             data = school_means)
summary(model)

# Visualise: school means coloured by district and NutriFirst status
if (requireNamespace("ggplot2", quietly = TRUE)) {
  library(ggplot2)
  ggplot(school_means,
         aes(x = district, y = score,
             colour = district, shape = nutrifirst)) +
    geom_jitter(width = 0.15, size = 3) +
    scale_shape_manual(values = c("FALSE" = 16, "TRUE" = 8)) +
    labs(title = "NutritionScore by district and school",
         subtitle = "Star = NutriFirst batch SB-2046-NF-07") +
    theme_minimal()
}
```

---

scale_colour_district *District colour scale (D7 / D12 / D23)*

Description

District colour scale (D7 / D12 / D23)

Usage

```
scale_colour_district(...)
```

Arguments

`...`	Arguments passed to `scale_colour_manual`.

Value

Returns an object that allows you to customize the appearance of plots.

`scale_colour_statpunk` *StatPunk discrete colour scale*

Description

StatPunk discrete colour scale

Usage

```
scale_colour_statpunk(...)
```

Arguments

`...`	Arguments passed to `scale_colour_manual`.

Value

Returns an object that allows you to customize the appearance of plots.

`scale_colour_statpunk_continuous`
StatPunk continuous colour scale

Description

Diverging scale from teal (low) through dark (mid) to rust (high). Use for variables where both extremes are meaningful.

Usage

```
scale_colour_statpunk_continuous(
  low = SP_TEAL,
  high = SP_RUST,
  mid = SP_BG_3,
  midpoint = 0,
  ...
)
```

Arguments

low	Colour for low values. Default teal.
high	Colour for high values. Default rust.
mid	Colour for midpoint. Default dark background.
midpoint	Numeric midpoint. Default 0.
...	Arguments passed to scale_colour_gradient2.

Value

Returns an object that allows you to customize the appearance of plots.

---

scale_fill_district *District fill scale (D7 / D12 / D23)*

Description

District fill scale (D7 / D12 / D23)

Usage

```
scale_fill_district(...)
```

Arguments

...	Arguments passed to scale_fill_manual.

Value

Returns an object that allows you to customize the appearance of plots.

---

scale_fill_statpunk *StatPunk discrete fill scale*

Description

StatPunk discrete fill scale

Usage

```
scale_fill_statpunk(...)
```

Arguments

...	Arguments passed to scale_fill_manual.

Value

Returns an object that allows you to customize the appearance of plots.

scale_fill_statpunk_continuous

StatPunk continuous fill scale

Description

StatPunk continuous fill scale

Usage

```
scale_fill_statpunk_continuous(
  low = SP_TEAL,
  high = SP_RUST,
  mid = SP_BG_3,
  midpoint = 0,
  ...
)
```

Arguments

`low`	Colour for low values. Default teal.
`high`	Colour for high values. Default rust.
`mid`	Colour for midpoint. Default dark background.
`midpoint`	Numeric midpoint. Default 0.
`...`	Arguments passed to `scale_colour_gradient2`.

Value

Returns an object that allows you to customize the appearance of plots.

scale_fill_statpunk_seq

StatPunk sequential fill scale

Description

Single-hue scale from dark background to rust. Use for variables that only go in one direction (counts, probabilities, scores).

Usage

```
scale_fill_statpunk_seq(...)

scale_colour_statpunk_seq(...)
```

Arguments

...	Arguments passed to `scale_fill_gradient`.

Value

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

---

statpunk_demo *Show a gallery of StatPunk theme examples*

Description

Produces three demonstration plots using built-in R datasets. Requires ggplot2 and patchwork.

Usage

```
statpunk_demo()
```

Value

Returns an object that allows you to customize the appearance of plots.

---

theme_statpunk *StatPunk ggplot2 theme*

Description

A dark, monospace, noir theme inspired by the BetaBit StatPunk universe (WaszKrak, 2047). Designed for datasets from the RougeLM package.

Usage

```
theme_statpunk(
  base_size = 13,
  base_family = "sans",
  grid = "both",
  axis_ticks = TRUE,
  border = FALSE
)
```

Arguments

base_size	Base font size in points. Default 13.
base_family	Base font family. Default "IBM Plex Mono". Falls back gracefully to system monospace if not installed.
grid	One of "both" (default), "x", "y", or "none". Controls which major grid lines are shown.
axis_ticks	Logical. Show axis tick marks. Default TRUE.
border	Logical. Draw a border around the plot panel. Default FALSE.

Value

A ggplot2 theme object.

Examples

```
library(ggplot2)
ggplot(mtcars, aes(wt, mpg, colour = factor(cyl))) +
  geom_point(size = 2.5) +
  theme_statpunk()
```

---

theme_statpunk_light *StatPunk light ggplot2 theme*

Description

A clean, high-contrast theme on white background for the BetaBit StatPunk universe. Near-black data elements on white ensure strong legibility in print and on-screen. Rust orange (#c4521a) is used as the primary accent.

Usage

```
theme_statpunk_light(
  base_size = 13,
  base_family = "sans",
  grid = "both",
  axis_ticks = TRUE,
  border = FALSE
)

theme_statpunk_light_void(base_size = 13, base_family = "IBM Plex Mono")

scale_colour_statpunk_light(...)

scale_fill_statpunk_light(...)
```

```
scale_colour_district_light(...)

scale_fill_district_light(...)

scale_colour_statpunk_light_continuous(
  low = SL_TEAL,
  high = SL_RUST,
  mid = "#f0ede8",
  midpoint = 0,
  ...
)

scale_fill_statpunk_light_continuous(
  low = SL_TEAL,
  high = SL_RUST,
  mid = "#f0ede8",
  midpoint = 0,
  ...
)

scale_fill_statpunk_light_seq(...)

scale_colour_statpunk_light_seq(...)

use_statpunk_light(base_size = 13)

statpunk_light_demo()
```

Arguments

base_size	Base font size in points. Default 13.
base_family	Base font family. Default "IBM Plex Mono".
grid	One of "both" (default), "x", "y", or "none".
axis_ticks	Logical. Show axis ticks. Default TRUE.
border	Logical. Draw panel border. Default FALSE.
...	Just passed to the internal function.
low	Just passed to the internal function.
high	Just passed to the internal function.
mid	Just passed to the internal function.
midpoint	Just passed to the internal function.

Value

A ggplot2 theme object.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Returns an object that allows you to customize the appearance of plots.

Examples

```
library(ggplot2)
ggplot(mtcars, aes(wt, mpg, colour = factor(cyl))) +
  geom_point(size = 2.5) +
  scale_colour_statpunk_light() +
  theme_statpunk_light()
```

`theme_statpunk_void` *Minimal StatPunk theme*

Description

A stripped-down version of `theme_statpunk()` with no grid lines and no axis ticks. Good for scatter plots and maps.

Usage

```
theme_statpunk_void(base_size = 13, base_family = "IBM Plex Mono")
```

Arguments

`base_size`	Base font size in points. Default 13.
`base_family`	Base font family. Default "IBM Plex Mono". Falls back gracefully to system monospace if not installed.

Value

Returns an object that allows you to customize the appearance of plots.

use_statpunk *Set theme_statpunk as the session default*

Description

Runs `theme_set(theme_statpunk())` and updates default geom colours so that `geom_point()`, `geom_line()` etc. use the palette automatically when no explicit colour is mapped.

Usage

```
use_statpunk(base_size = 13)
```

Arguments

`base_size`	Passed to `theme_statpunk()`.

Value

Returns an object that allows you to customize the appearance of plots.

Index