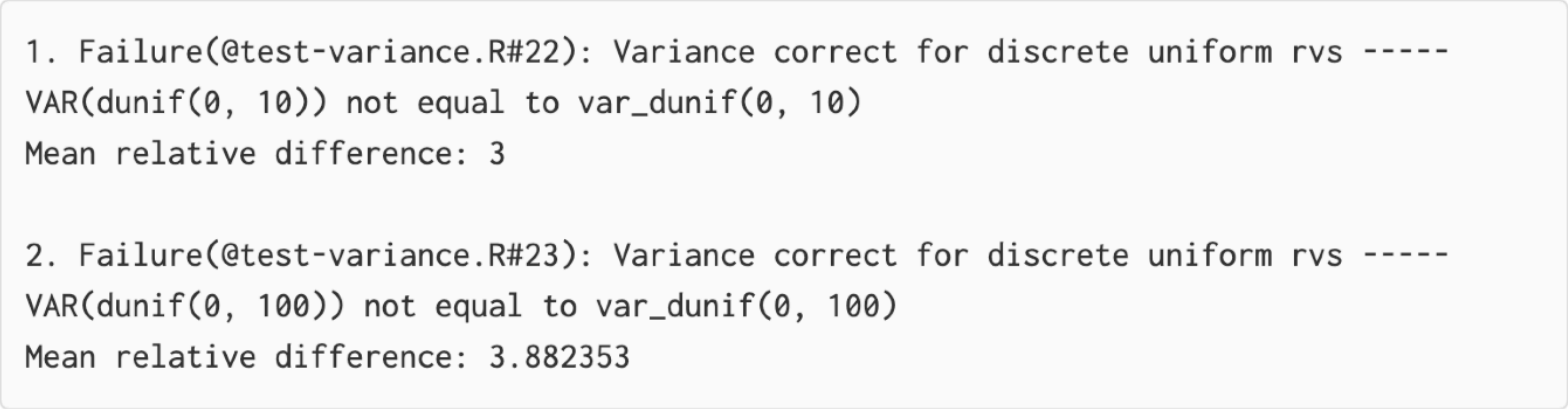
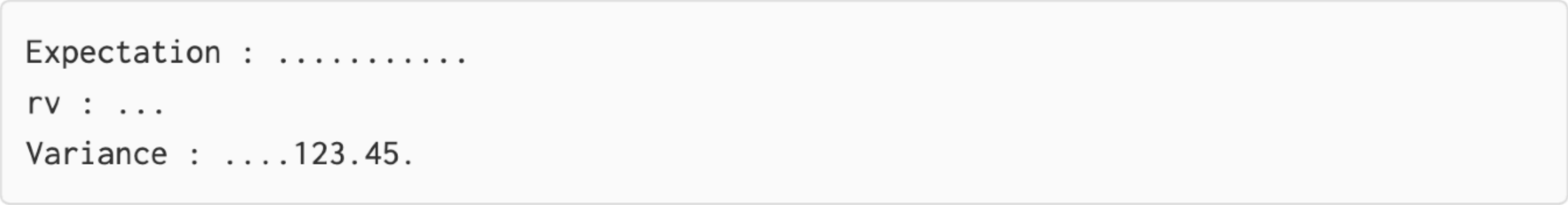
The testthat package helps you execute **unit tests**



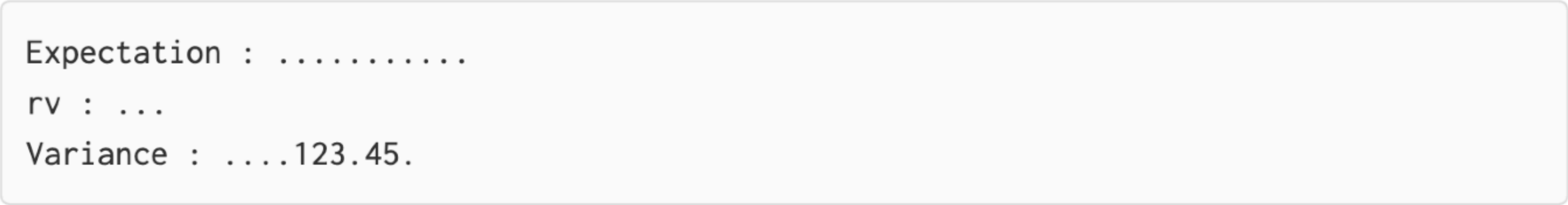
```
1. Failure(@test-variance.R#22): Variance correct for discrete uniform rvs -----
VAR(dunif(0, 10)) not equal to var_dunif(0, 10)
Mean relative difference: 3

2. Failure(@test-variance.R#23): Variance correct for discrete uniform rvs -----
VAR(dunif(0, 100)) not equal to var_dunif(0, 100)
Mean relative difference: 3.882353
```

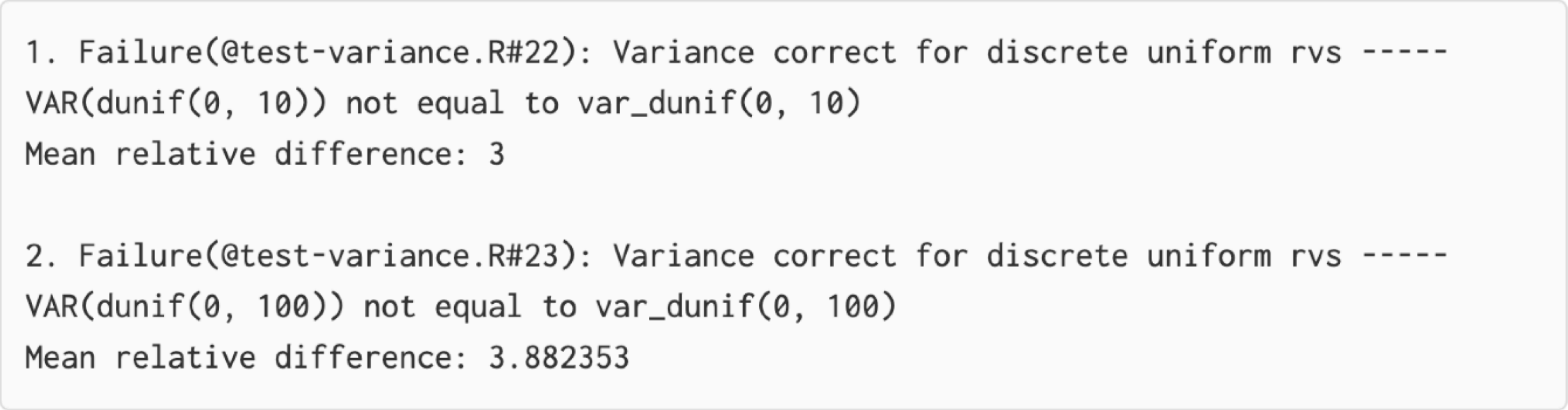


```
Expectation : ...........
rv : ...
Variance : ....123.45.
```

The testthat package helps you execute **unit tests**



```
Expectation : ...........
rv : ...
Variance : ....123.45.
```



```
1. Failure(@test-variance.R#22): Variance correct for discrete uniform rvs -----
VAR(dunif(0, 10)) not equal to var_dunif(0, 10)
Mean relative difference: 3

2. Failure(@test-variance.R#23): Variance correct for discrete uniform rvs -----
VAR(dunif(0, 100)) not equal to var_dunif(0, 100)
Mean relative difference: 3.882353
```

The testthat package helps you execute **unit tests**

tests/testthat/test-evaluate_models.R

```
context("evaluate_models")

cds <- load_a549()
cds <- estimate_size_factors(cds)
test_that("evaluate_models() returns correct output for poisson models",{
  test_cds = cds[rowData(cds)$gene_short_name == "ANGPTL4",]
  fit_m = fit_models(test_cds, model_formula_str = "~log_dose", expression_family = "poisson")
  evaluated_fit = suppressWarnings(evaluate_fits(fit_m))
  expect_equal(evaluated_fit$null_deviance, 1031, tolerance=1e-3)
  expect_equal(evaluated_fit$df_null, 499)
  expect_equal(evaluated_fit$logLik, -775, tolerance=1e-3)
  expect_equal(evaluated_fit$AIC, 1553, tolerance=1e-3)
  expect_equal(evaluated_fit$BIC, NA_real_)
  expect_equal(evaluated_fit$deviance, 983, tolerance=1e-3)
  expect_equal(evaluated_fit$df_residual, 498)
})
```

When we check in changes to the Monocle repo, these tests are automatically run via Travis, a system for "**continuous integration**"