BFF

```
## Package 'BFF' version 4.2.1 for Bayesian hypothesis testing.
## Type 'citation("BFF")' for citing this R package in publications

## Loading required package: lattice

##
## Attaching package: 'BSDA'

## The following object is masked from 'package:datasets':
##
##     Orange
```

Bayes factors are an alternative to p-values for evaluating hypothesis tests. However, unlike p-values, Bayes factors are able to provide evidence for a null hypothesis. This allows a researcher to conclude that data provides evidence for the null hypothesis (instead of just "failing to reject" a null hypothesis). Bayes factors also have a clear interpretation: a larger Bayes factor shows more evidence for a hypothesis, as opposed to p-values (can anyone tell the difference between 0.05 and 0.06?). Bayes factors have in the past had limited acceptance due to computational issues and difficulty in selecting a prior. Recent work (see 'Bayes factor functions for reporting outcomes of hypothesis tests,' 2023 and 'On the use of non-local prior densities in Bayesian hypothesis tests,' 2010) introduced the idea of using non-local priors to calculate Bayes factors. This package implements "Bayes Factor Functions" (or BFFS). In contrast to a single Bayes factor, BFFs express Bayes factors as a function of the prior densities used to define the alternative hypotheses.

Interpreting Bayes factors is usually done on the log scale (also called the weight of evidence, or WoE) On this scale, a positive Bayes factor represents evidence for the alternative hypothesis. A negative Bayes factor represents evidence for the null hypothesis. As a rule of thumb, the following table can be used to interpret a Bayes factor. However, these are just guidelines and some fields may require higher or lower thresholds of evidence.

WoE	Interpretation
(-1, 1)	No strong evidence for either H_0 or H_1
(1, 3)	Positive evidence for H_1
(-1, -3)	Positive evidence for H_0
(3, 5)	Strong evidence for H_1
(-3, -5)	Strong evidence for H_0
(5, ∞)	Very strong evidence for H_1
(-5, -∞)	Very strong evidence for H_0

Table 1: Common interpretations of the Weight of Evidence

This package provides the bayes factor values for different effect sizes from 0 to 1. A small effect size is usually considered from 0.2 to 0.5„ medium effect sizes from 0.5 to 0.8, and large effect sizes as greater than 0.8. However, we use the cutoffs in "Statistical Power Analysis For the Behavioral Sciences" (Cohen) for specific tests.

Using this package is very similar to using the familiar t, z, chi^2, and F tests in R. You will need the same information - the test statistic, degrees of freedom, and sample size. A graph is produced that shows the BFF curve over the different effect sizes.

For evaluating evidence from multiple studies, the parameter 'r' can also be set. The default value for r is 1, but 'r' can be set to place more of the prior mass around a suggested point. The higher the r, the more

peaked the prior is around the desired effect size. The graph below shows different options.

```r
# Define the density function
density_function <- function(lambda, tau2, r) {
  coef <- (lambda^2)^r / ((2 * tau2)^(r + 0.5) * gamma(r + 0.5))
  exp_term <- exp(-lambda^2 / (2 * tau2))
  return(coef * exp_term)
}

# Define omega and n
omega <- 0.5
n <- 50

# Define the range for lambda
lambda_range <- seq(-10, 10, length.out = 1000)

# Define the values of r to plot
r_values <- c(1, 2, 3, 5, 10)

# Define colors for the different r values
colors <- rainbow(length(r_values))

# Plot the densities using base R plotting functions
plot(NULL, xlim = c(-10, 10), xlab = expression(lambda), ylab = "Density", main = "Density for varying :

# Add lines for each value of r
for (i in seq_along(r_values)) {
  r <- r_values[i]
  tau2 <- (n * omega^2) / (2 * r)
  densities <- sapply(lambda_range, density_function, tau2 = tau2, r = r)
  lines(lambda_range, densities, col = colors[i], lwd = 2)
}

# Add a legend
legend("topright", legend = paste("r =", r_values), col = colors, lwd = 2)
```
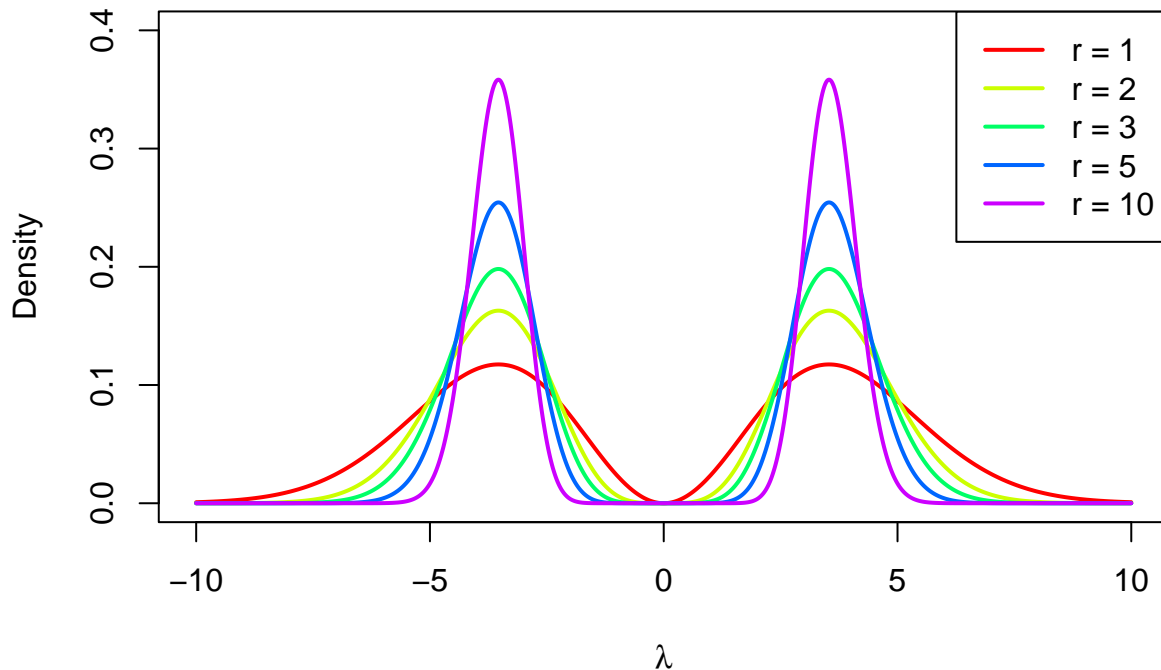
Density for varying r

Examples

In the example below, you can see how the test returns different results for the same t statistic in a one sided verses two sample case. In the one sample case, the Bayes factor is maximized at a Cohen's d value of 0.28, and in the two sample test the Bayes factor is maximized at the Cohen's d value of 0.59.

The second two examples show the t statistic calculated with the same one and two sample cases, but with a higher r value. This higher r value leads to a smaller Cohen's d value that maxizes the Bayes factor.

```
tBFF = t_test_BFF(t_stat = 2.5, n = 50, one_sample = TRUE)
tBFF # plot the results
```

```
##  Bayesian non-local one-sample t test
##
## maximized log Bayes factor = 2.34
## maximized omega = 0.28 (Cohen's d)
## alternative = two.sided
```
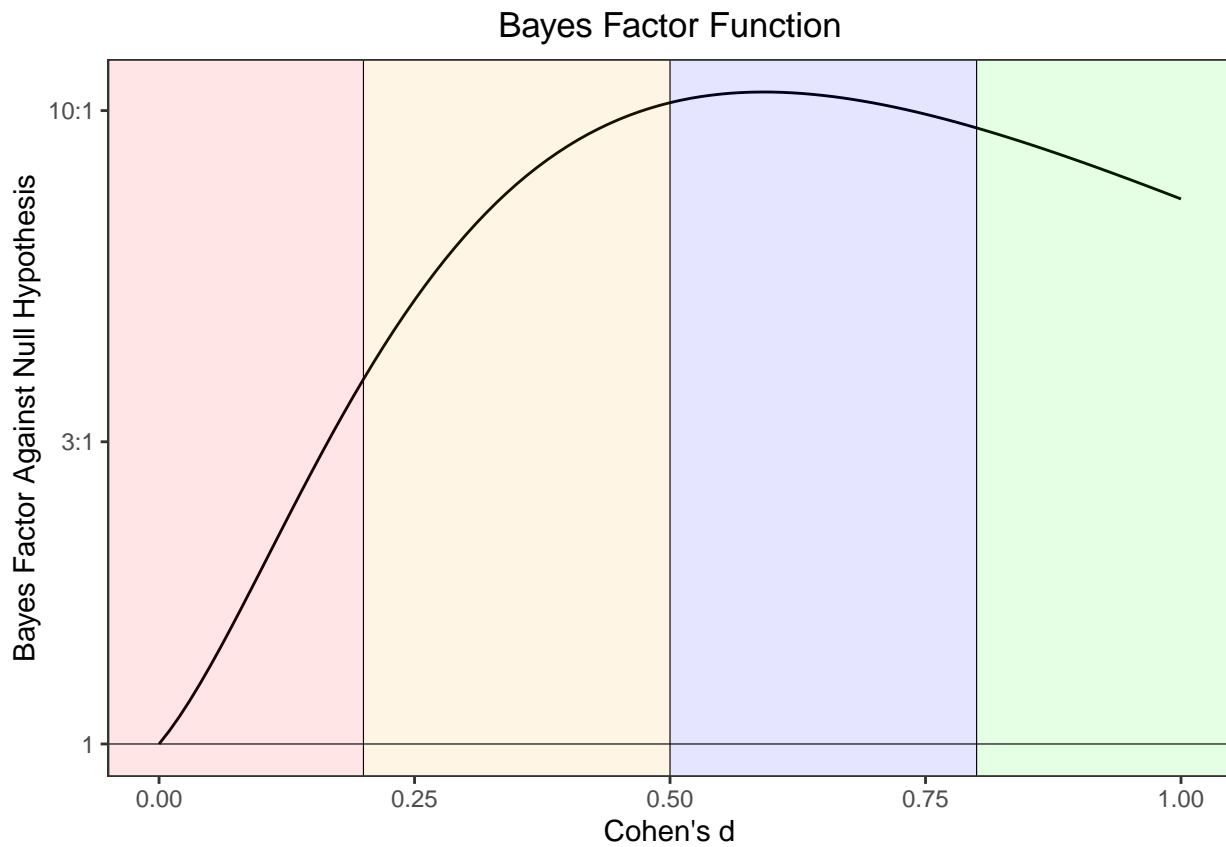
```
plot(tBFF)
```

Bayes Factor Function
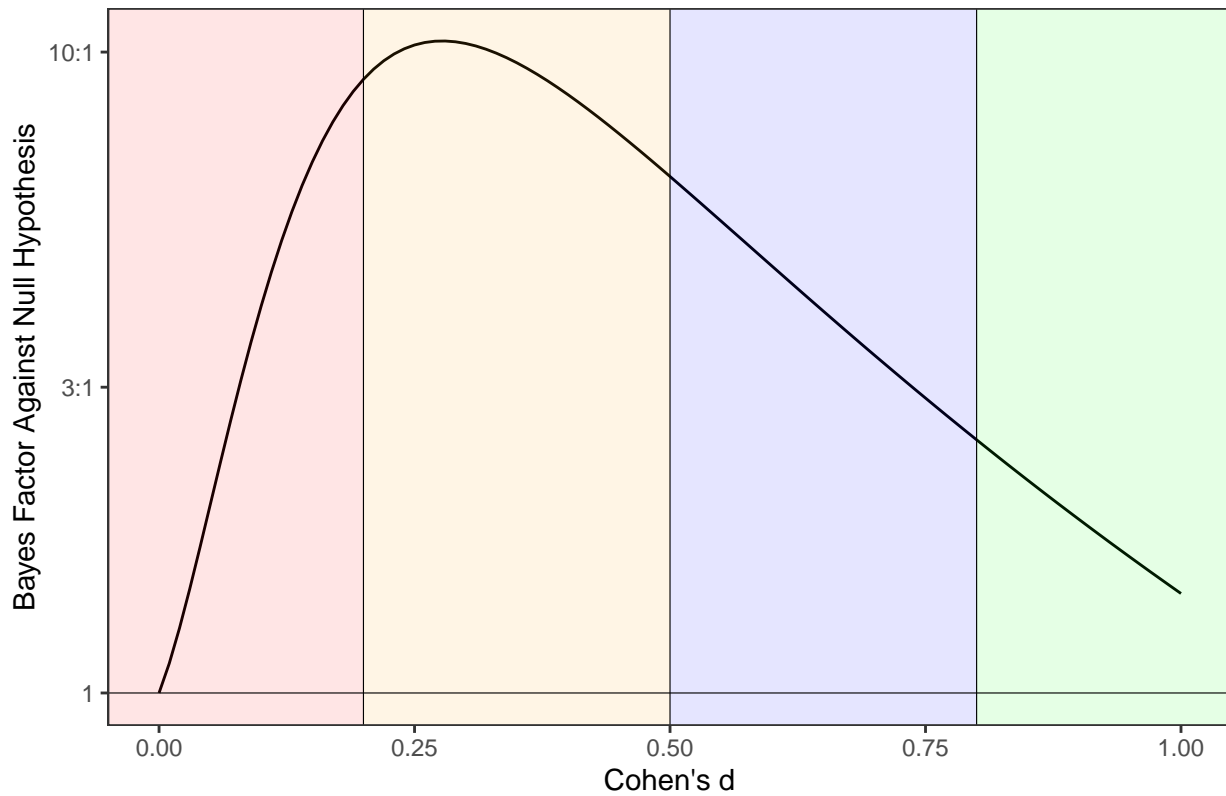

```r
# now an example with different types of settings
tBFF_twosample = t_test_BFF(t_stat = 2.5, n1 = 50, n2 = 14, one_sample = FALSE)
tBFF_twosample # plot the results
```

```
##  Bayesian non-local two-sample t test
##
## maximized log Bayes factor = 2.37
## maximized omega = 0.59 (Cohen's d)
## alternative = two.sided
```

```r
plot(tBFF_twosample)
```
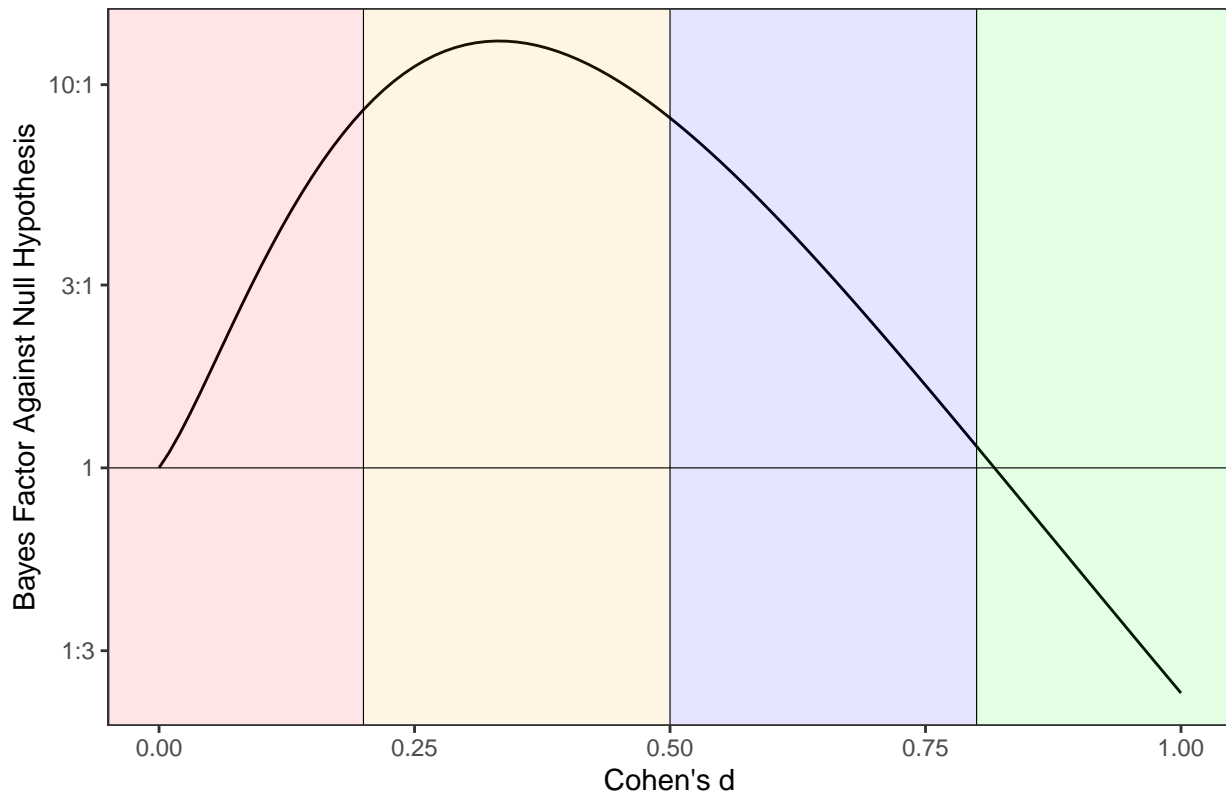
Bayes Factor Function


```
# repeat the above with r = 5
tBFF_r5 = t_test_BFF(t_stat = 2.5, n = 50, one_sample = TRUE, r = 5)
tBFF_r5 # plot the results
```

```
##  Bayesian non-local one-sample t test
##
## maximized log Bayes factor = 2.56
## maximized omega = 0.33 (Cohen's d)
## alternative = two.sided
```

```
plot(tBFF_r5)
```

Bayes Factor Function


```
# now an example with different types of settings
tBFF_twosample_r5 = t_test_BFF(t_stat = 2.5, n1 = 50, n2 = 14, one_sample = FALSE, r = 5)
tBFF_twosample_r5 # plot the results
```

```
##  Bayesian non-local two-sample t test
##
## maximized log Bayes factor = 2.59
## maximized omega = 0.71 (Cohen's d)
## alternative = two.sided
```

```
plot(tBFF_twosample_r5)
```

Bayes Factor Function