Learning to Calculate P-Values from F-Statistics in R

Authored by

Mohammed looti

November 9, 2025

Understanding the F-Statistic and F-Test

The F-test is a fundamental statistical tool derived from the F-distribution, commonly employed to compare variances or to assess the overall significance of a statistical model, such as in Analysis of Variance (ANOVA) or linear regression. When performing an F-test, the result is a calculated test statistic, known as the **F-statistic**. This value summarizes how much the variation between group means (or variation explained by the model) exceeds the variation within the groups (or unexplained residual variation).

To determine if the observed **F-statistic** is statistically significant--meaning it is unlikely to have occurred purely by random chance--we must calculate its corresponding **p-value**. The **p-value** represents the probability of observing an F-statistic as extreme as, or more extreme than, the one calculated, assuming the null hypothesis is true. A small **p-value** typically leads to the rejection of the null hypothesis, suggesting that the model or the differences in variance are statistically significant.

In the programming environment of R, calculating this probability requires referencing the cumulative distribution function (CDF) of the F-distribution. This process necessitates knowledge of not only the F-statistic itself but also the associated numerator and denominator degrees of freedom, which dictate the specific shape of the F-distribution curve being used for the calculation.

The R Function for Calculating P-Values: pf()

The `pf()` function in R is the dedicated tool for interacting with the F-distribution. It stands for "probability of F" and calculates the cumulative probability associated with a given F-statistic value. By default, most statistical tests, including the F-test, are upper-tailed tests, meaning we are interested in the probability of observing values in the upper tail of the distribution.

To find the exact **p-value** associated with an observed **F-statistic** in R, especially when conducting an upper-tailed test, the `pf()` function is invoked using a specific structure. This structure ensures that R correctly identifies the critical region of the distribution that corresponds to the observed test statistic.

The core command used to calculate the upper-tail probability (the **p-value**) for an **F-statistic** is structured as follows:

pf(fstat, df1, df2, lower.tail = FALSE)

Deconstructing the Arguments of the pf() Function

Understanding each argument within the `pf()` function is crucial for accurate calculation. The F-

distribution is characterized by two parameters: the numerator degrees of freedom (df_1) and the denominator degrees of freedom (df_2). These parameters determine the shape of the distribution and, consequently, the resulting **p-value**.

The arguments map directly to the components derived from the statistical test being performed. For instance, in an ANOVA, df_1 relates to the number of groups minus one, while df_2 relates to the total number of observations minus the number of groups. In a multiple regression context, df_1 is the number of predictors, and df_2 is the residual degrees of freedom.

fstat - This argument specifies the calculated value of the **F-statistic** obtained from the statistical test.
df1 - This is the numerator degrees of freedom, often referred to as df_1. This value typically corresponds to the degrees of freedom associated with the variation explained by the model or treatment effects.
df2 - This is the denominator degrees of freedom, or df_2. This value corresponds to the residual or error degrees of freedom, representing the variation not explained by the model.
lower.tail - This logical argument determines which tail of the distribution is used for the probability calculation. By default, this argument is set to **TRUE**, which returns the probability associated with the lower tail (from 0 up to **fstat**). Because the **p-value** for the F-test is almost always calculated using the upper tail (the area to the right of the **fstat**), we must explicitly set **lower.tail = FALSE** to obtain the correct **p-value**.

Practical Application: A Direct Calculation Example

To illustrate the direct calculation of a p-value using the `pf()` function, consider a scenario where an F-test has yielded an **F-statistic** of 5.0. Furthermore, we assume this test had 3 numerator degrees of freedom (df_1) and 14 denominator degrees of freedom (df_2). We are interested in finding the probability of observing an F-statistic of 5.0 or greater.

The calculation requires inputting these values directly into the `pf()` function, ensuring that the `lower.tail` argument is set to **FALSE** to correctly isolate the upper tail probability, which represents the **p-value**.

The R code and the resulting calculation are shown below:

```
pf(5, 3, 14, lower.tail = FALSE)

# 0.01457807
```

The resulting **p-value** is approximately 0.0146. If we were using a standard significance level ($alpha$) of 0.05, we would reject the null hypothesis because $0.0146 < 0.05$. This indicates

strong evidence against the null hypothesis, suggesting that the differences or effects tested by the F-test are statistically significant.

Using the F-Test in Linear Regression Analysis

One of the most critical applications of the F-test is assessing the overall performance of a linear regression model. When multiple predictor variables are used, the F-test evaluates the null hypothesis that all regression coefficients (except the intercept) are simultaneously equal to zero. In simpler terms, it tests whether the entire model, collectively, provides a statistically significant fit to the data compared to a model with no predictors.

The output of a typical regression analysis in R will automatically provide the **F-statistic** and its corresponding **p-value** for the overall model fit. This calculation is derived by comparing the Mean Square Regression (MSR) to the Mean Square Error (MSE). The MSR represents the variation explained by the predictors, while the MSE represents the unexplained variation (residuals).

The F-statistic in this context is defined as the ratio $F = frac{MSR}{MSE}$. If this ratio is significantly greater than 1, it suggests that the predictors explain more variation than is left unexplained, leading to a small **p-value** and a conclusion that the model is statistically significant. The following example demonstrates how to extract and verify the **p-value** from an F-statistic generated by a multiple regression analysis.

Step-by-Step R Implementation for Regression F-Test

Consider a scenario involving student performance data. We want to model the final exam score based on two predictor variables: the total number of hours studied and the total number of preparatory exams taken. We start by creating a simple dataset for 12 students in R.

The dataset includes three variables: `study_hours`, `prep_exams`, and the `final_score`. Defining this structure is the necessary first step before fitting any statistical model.

```
#create dataset
data <- data.frame(study_hours = c(3, 7, 16, 14, 12, 7, 4, 19, 4, 8, 8, 3),
prep_exams = c(2, 6, 5, 2, 7, 4, 4, 2, 8, 4, 1, 3),
final_score = c(76, 88, 96, 90, 98, 80, 86, 89, 68, 75, 72, 76))

#view first six rows of dataset
head(data)
```

study_hours prep_exams final_score
#1 3 2 76

#2 7 6 88
#3 16 5 96
#4 14 2 90
#5 12 7 98
#6 7 4 80

Once the data is structured, we utilize the `lm()` function in R to fit the multiple linear regression model. Here, `final_score` is the response variable, and both `study_hours` and `prep_exams` are the predictor variables. The `summary()` function is then used to generate a detailed output that includes the coefficients, standard errors, R-squared values, and, crucially, the overall **F-statistic** and its associated **p-value**.

#fit regression model
model <- lm(final_score ~ study_hours + prep_exams, data = data)

#view output of the model
summary(model)

#Call:
#lm(formula = final_score ~ study_hours + prep_exams, data = data)
#
#Residuals:
Min 1Q Median 3Q Max
#-13.128 -5.319 2.168 3.458 9.341
#
#Coefficients:
Estimate Std. Error t value Pr(>|t|)
#(Intercept) 66.990 6.211 10.785 1.9e-06 ***
#study_hours 1.300 0.417 3.117 0.0124 *
#prep_exams 1.117 1.025 1.090 0.3041
#---
#Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
#Residual standard error: 7.327 on 9 degrees of freedom
#Multiple R-squared: 0.5308, Adjusted R-squared: 0.4265
#F-statistic: 5.091 on 2 and 9 DF, p-value: 0.0332

Interpreting the Final P-Value and Model Significance

The crucial information regarding the overall model significance is located on the final line of the regression output summary. We observe that the **F-statistic** for the overall regression model is reported as **5.091**. This test statistic is calculated using two sets of degrees of freedom: 2 for the numerator (corresponding to the two predictor variables, `study_hours` and `prep_exams`) and 9 for the denominator (the residual degrees of freedom, calculated as $N - k - 1$, where $N=12$ and $k=2$).

R automatically computes the **p-value** corresponding to this **F-statistic**, which is reported as **0.0332**. Since 0.0332 is less than the conventional significance level of 0.05, we reject the null hypothesis that both regression coefficients are zero. This conclusion implies that the model, incorporating study hours and prep exams, is a statistically significant predictor of the final score.

We can verify R's automated calculation by manually inputting these parameters into the `pf()` function, confirming our understanding of the F-distribution calculation. We use the observed **F-statistic** of 5.091, the numerator degrees of freedom (2), and the denominator degrees of freedom (9), ensuring **lower.tail = FALSE** is set for the upper-tail probability (the p-value).

pf(5.091, 2, 9, lower.tail = FALSE)

0.0331947

The manual calculation yields 0.0331947, which confirms the **p-value** provided by the linear regression output (0.0332). This exercise demonstrates the reliability of R's statistical functions and provides confidence in interpreting the results of complex statistical models.