Regression in ANOVA

James H. Steiger

Department of Psychology and Human Development
Vanderbilt University

Regression in ANOVA

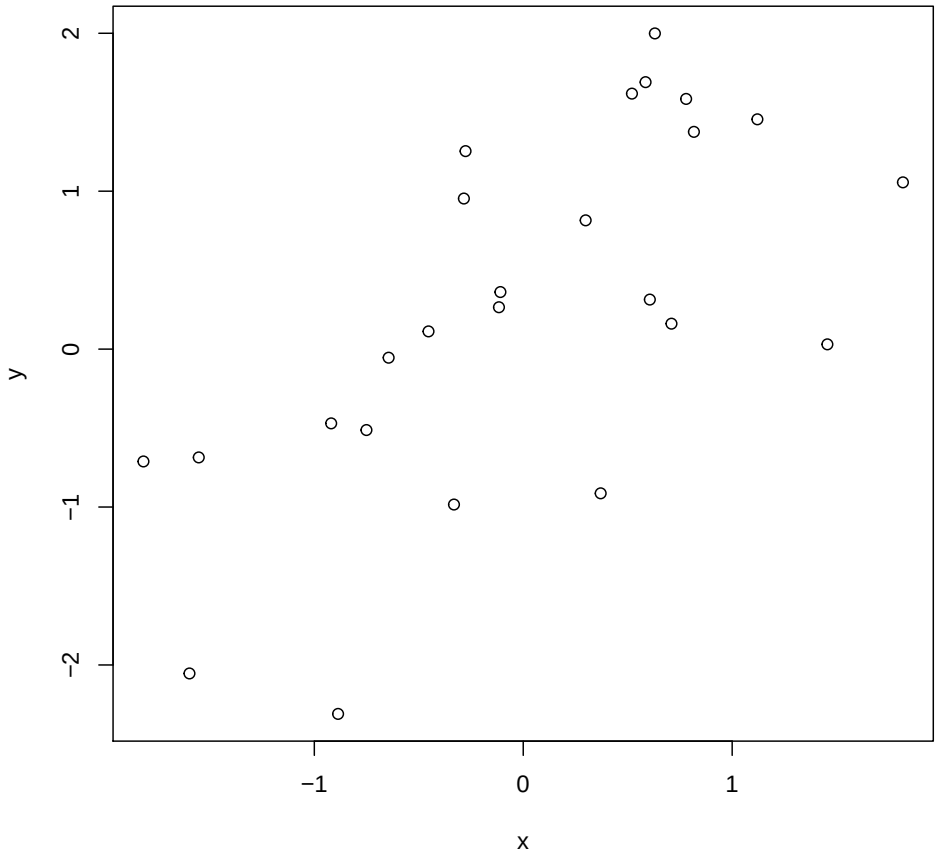
Introduction

- In this module, we begin the study of the classic analysis of variance (ANOVA) designs.
- Since we shall be analyzing these models using R and the regression framework of the General Linear Model, we start by recalling some of the basics of regression modeling.
- We work through linear regression and multiple regression, and include a brief tutorial on the statistical comparison of *nested multiple regression models*.
- We then show how the classic ANOVA model can be (and is) analyzed as a multiple regression model.

Basic Linear Regression in R

- Let's define and plot some artificial data on two variables.

```
> set.seed(12345)
> x <- rnorm(25)
> y <- sqrt(1/2) * x + sqrt(1/2) * rnorm(25)
> plot(x, y)
```



Basic Linear Regression in R

- We want to predict y from x using least squares linear regression.
- We seek to fit a model of the form

$$y_i = \beta_0 + \beta_1 x_i + e_i = \hat{y}_i + e_i$$

  while minimizing the sum of squared errors in the "up-down" plot direction.
- We fit such a model in R by creating a "fit object" and examining its contents.
- We see that the formula for $\hat{y}_i$ is a straight line with slope β_1 and intercept β_0.

Basic Linear Regression in R

- We start by creating the model with a model specification formula.
- This formula corresponds to the model stated on the previous slide in a specific way:
  1. Instead of an equal sign, a " "is used.
  2. The coefficients themselves are not listed, only the predictor variables.
  3. The error term is not listed
  4. The intercept term generally does not need to be listed, but can be listed with a "1".
- So the model on the previous page is translated as `y ~ x`.

Basic Linear Regression in R

- We create the fit object as follows.

```
> fit.1 <- lm(y ~ x)
```

- Once we have created the fit object, we can examine its contents.

```
> summary(fit.1)


Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8459 -0.6692  0.2133  0.5082  1.2330

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2549     0.1754   1.453 0.159709
x             0.8111     0.1894   4.282 0.000279 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8771 on 23 degrees of freedom
Multiple R-squared:  0.4435,Adjusted R-squared:  0.4193
F-statistic: 18.33 on 1 and 23 DF,  p-value: 0.0002791
```

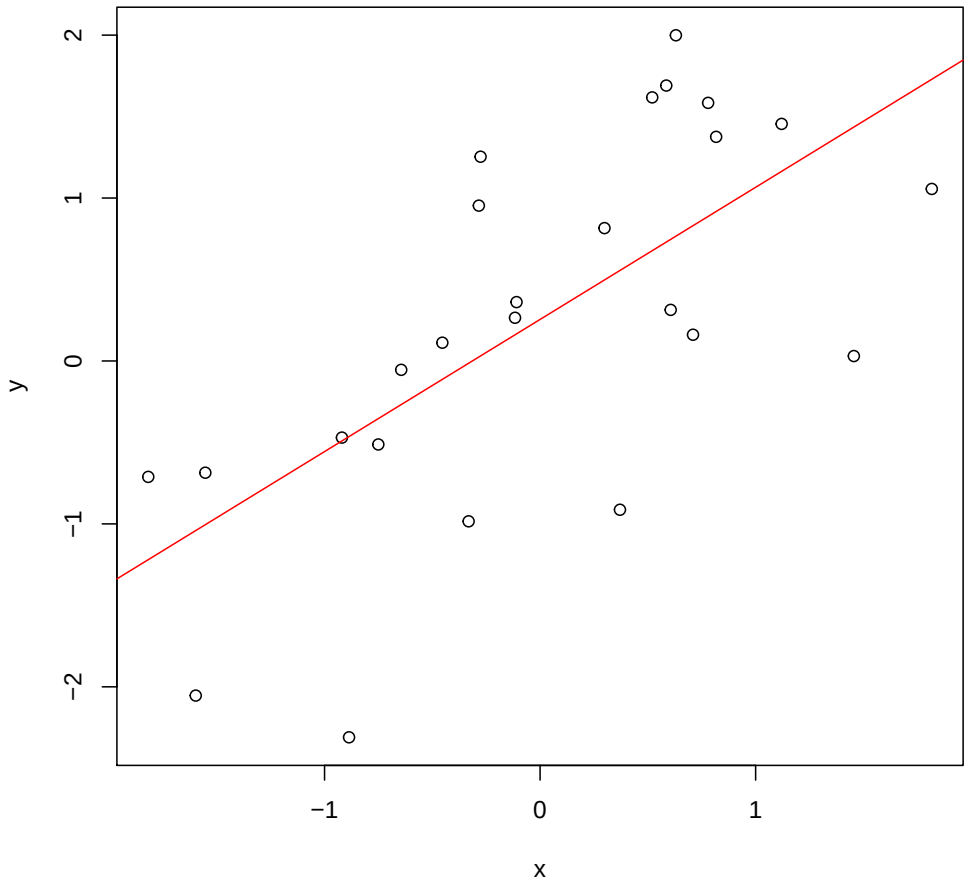
Basic Linear Regression in R

- We see the printed coefficients for the intercept and for x.
- There are statistical t tests for each coefficient. These are tests of the null hypothesis that the coefficient is zero.
- There is also a test of the hypothesis that the squared multiple correlation (the square of the correlation between $\hat{y}$ and y) is zero.
- Standard errors are also printed, so you can compute confidence intervals. (How would you do that quickly "in your head?" (C.P.)
- The slope is not significantly different from zero. Does that surprise you? (C.P.)
- The squared correlation is .4435. What is the correlation in the population? (C.P.)

Basic Linear Regression in R

- If we want, we can, in the case of simple bivariate regression, add a regression line to the plot automatically using the `abline` function.

```
> plot(x, y)
> abline(fit.1, col = "red")
```



Multiple Regression in R

- If we have more than one predictor, we have a multiple regression model.
- Suppose, for example, we add another predictor w to our artificial data set.
- We design this predictor to be completely uncorrelated with the other predictor and the criterion, so this predictor is, in the population, of no value.
- Now our model becomes

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 w_i + e_i$$

```
> w <- rnorm(25)
```

Multiple Regression in R

- How would we set up and fit the model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 w_i + e_i$$

in R?

Multiple Regression in R

- How would we set up and fit the model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 w_i + e_i$$

in R?

- That's right,

```
> fit.2 <- lm(y ~ x + w)
```

Multiple Regression in R

```
> summary(fit.2)


Call:
lm(formula = y ~ x + w)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8475 -0.6693  0.2198  0.5108  1.2298

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.254043   0.181833   1.397 0.176312
x           0.812727   0.202128   4.021 0.000573 ***
w           0.004366   0.152239   0.029 0.977380
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8968 on 22 degrees of freedom
Multiple R-squared:  0.4435,Adjusted R-squared:  0.393
F-statistic: 8.768 on 2 and 22 DF,  p-value: 0.001584
```

Nested Models

Introduction

- The situation we examined in the previous sections is a simple example of a *sequence of nested models*.
- One model is *nested within* another if it is a special case of the other in which some model coefficients are constrained to be zero.
- The model with only x as a predictor is a special case of the model with x and w as predictors, with the coefficient β_2 constrained to be zero.

Nested Models

Model Comparison

- When two models are nested multiple regression models, there is a simple procedure for comparing them.
- This procedure tests whether the more complex model is significantly better than the simpler model.
- In the sample, of course, the more complex of two nested models will always fit at least as well as the less complex model.

Nested Models

Partial F-Tests: A General Approach

- Suppose Model A includes Model B as a special case. That is, Model B is a special case of Model A where some terms have coefficients of zero. Then Model B is nested within Model A.
- If we define SS_a to be the sum of squared residuals for Model A, SS_b the sum of squared residuals for Model B.
- Since Model B is a special case of Model A, model A is more complex so SS_b will always be as least as large as SS_a.
- We define df_a to be $n - p_a$, where p_a is the number of terms in Model A including the intercept, and correspondingly $df_b = n - p_b$.
- Then, to compare Model B against Model A, we compute the partial $F-$statistic as follows.

$$F_{df_b - df_a, df_a} = \frac{MS_{comparison}}{MS_{res}} = \frac{(SS_b - SS_a)/(p_a - p_b)}{SS_a/df_a} \quad (1)$$

Nested Models

Partial F-Tests: A General Approach

- R will perform the partial F-test automatically, using the `anova` command.

```
> anova(fit.1, fit.2)

Analysis of Variance Table

Model 1: y ~ x
Model 2: y ~ x + w
  Res.Df    RSS Df  Sum of Sq     F Pr(>F)
1     23 17.694
2     22 17.693  1 0.00066144 8e-04 0.9774
```

- Note that the p value for the model difference test is the same as the p value for the t-test of the significance of the coefficient for w shown previously.

Nested Models

Partial *F*-Tests: A General Approach

- What happens if we call the `anova` command with just a single model?

```
> anova(fit.1)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x          1 14.102 14.1025  18.331 0.0002791 ***
Residuals 23 17.694  0.7693
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Note that the *p*-value for this test is the same as the *p*-value for the overall test of zero squared multiple correlation shown in the output summary for `fit.1`.
- What is going on?

Nested Models

Partial *F*-Tests: A General Approach

- It turns out, if you call the `anova` command with a single fit object, it startes by comparing the first non-intercept term in the model against a baseline model with no predictors (i.e., just an intercept).
- If there is a second predictor, it compares the model with both predictors against the model with just one predictor.
- It produces this sequence of comparisons automatically.
- To demonstrate, let's fit a model with just an intercept.

```
> fit.0 <- lm(y ~ 1)
```

- Recall that the 1 in the model formula stands for the intercept.
- No let's perform a partial *F*-test comparing `fit.0` with `fit.1`.

Nested Models

Partial *F*-Tests: A General Approach

- Here we go.

```
> anova(fit.0, fit.1)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     24 31.796
2     23 17.694  1    14.102 18.331 0.0002791 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Note that we get exactly the same result for the model comparison as we got when we ran `anova` on just the `fit.1` object.

ANOVA as Dummy Variable Regression

- Suppose we have 3 groups, and we want to test the null hypothesis that all 3 come from populations with the same mean. A side assumption is that all groups have the same variance, and that the population distributions are normal.
- The alternative hypothesis is that at least one of the groups has a mean that is different from the others.
- Suppose that we want to test this hypothesis with some artificial data. In Group 1, the scores are 1,2,3. In Group 2, the scores are 4,5,6. In Group 3, they are 7,8,9.
- How can we set up a regression model corresponding to the null model?

ANOVA as Dummy Variable Regression

The Null Model

- Actually, such a model is very simple to specify, *providing we learn a couple of simple tricks*.
- First, instead of conceptualizing our scores as 3 columns with 3 numbers in each column, imagine them as stacked in a single vector of 9 scores, representing 9 observations from the variable y.
- Our null model is simply

$$y_i = \beta_0 + e_i \tag{2}$$

- Think about it. If all 3 population means are equal to a common value, then all 9 scores represent random variation around a single value β_0.

ANOVA as Dummy Variable Regression

The Null Model

- So in R, we could fit the model as follows.

```
> y <- 1:9
> model.0 <- lm(y ~ 1)
```

- We want to compare this model against a model that allows each group to have its own mean.
- How do we do that? The answer is to create *dummy predictors*.
- Let's see how that is done.

ANOVA as Dummy Variable Regression

The Alternative Model

- Remember, our baseline model includes an intercept.
- Let's consider why R signifies and intercept with a 1.
- The model

$$y_i = \beta_0 + e_i$$

can be rewritten as

$$y_i = \beta_0 One + e_i$$

where One is a "dummy variable" that always takes on the value 1.

- Since every variable has a 1 for the (implicit) intercept "variable," we need to allow Groups 1 and 2 to vary from the value β_0 in order for them to be modeled as having different means from Group 3.

ANOVA as Dummy Variable Regression

The Alternative Model

- That is easy to do. Simply create two more dummy variables called Group1 and Group2. Group1 takes on the value 1 for any observation in Group 1, but it takes on the value 0 otherwise. Group2 takes on the value 2 for any observation in Group 2, but takes on the value 0 otherwise.

```
> Group1 <- c(1, 1, 1, 0, 0, 0, 0, 0, 0)
> Group2 <- c(0, 0, 0, 1, 1, 1, 0, 0, 0)
```

- Our non-null model is then

$$y_i = \beta_0 + \beta_1 Group_1 + \beta_2 Group_2 + e_i$$

- We fit this model in R as

```
> model.1 <- lm(y ~ 1 + Group1 + Group2)
```

ANOVA as Dummy Variable Regression

The Alternative Model

- To test the null hypothesis of equal means against the alternative that the means may not be equal, we compare the two models.

```
> anova(model.0, model.1)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ 1 + Group1 + Group2
  Res.Df RSS Df Sum of Sq  F Pr(>F)
1      8  60
2      6   6  2        54 27  0.001 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Notice that the value of the F-statistic is 27.00, so the model that allows the group means to each be different is significantly better than the model that forces them all to be the same.
- Since our null model only had an intercept, we would get the identical result running the anova

Using Factor Variables

- A couple of observations are in order. First, we would get the same F-statistic had we chosen our dummy variables to be Group2 and Group3 (or Group1 and Group3) instead of Group1 and Group2.
- Second, although this is straightforward, it is tedious.
- R developers have automated the whole process through the use of `factor` variables.
- A factor variable contains codes for the various groups. If you include a factor variable in a regression formula, R automatically substitutes dummy variables for it.
- Let's create a factor variable called Group.

```
> Group <- factor(c(1, 1, 1, 2, 2, 2, 3, 3, 3))
```

Using Factor Variables

- Now, we'll fit a model with the Group variable as the only predictor.

```
> anova.model <- lm(y ~ Group)
> anova(anova.model)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value Pr(>F)
Group      2     54      27      27  0.001 ***
Residuals  6      6       1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Since the factor variable includes two dummy predictors (implicitly), the anova command compares a model with both predictors against a model with just the intercept.

Using Factor Variables

-
```
> summary(anova.model)


Call:
lm(formula = y ~ Group)

Residuals:
   Min     1Q Median     3Q    Max
    -1     -1      0      1      1

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.0000     0.5774   3.464 0.013400 *
Group2        3.0000     0.8165   3.674 0.010402 *
Group3        6.0000     0.8165   7.348 0.000325 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1 on 6 degrees of freedom
Multiple R-squared:    0.9,Adjusted R-squared:  0.8667
F-statistic:    27 on 2 and 6 DF,  p-value: 0.001
```

Using Factor Variables

- Note that the factor variable "decided" to create dummy variables for Group2 and Group3, rather than Group1 and Group2.
- In our data, the cell means were 2,5, and 8. Note how the intercept and the coefficients for Groups 2 and 3 reproduce those means.
- Scores in Group 1 are reproduced only with the intercept (2) and error, so they have a mean of 2.
- Scores in Group 2 are reproduced with the intercept (2) plus the coefficient for Group 2 (i.e., 3) plus error, so they are estimated to have a population mean of 5, and so on.