# Trend Analysis

James H. Steiger

Department of Psychology and Human Development
Vanderbilt University

# Trend Analysis

# Introduction

*Trend Analysis* is a name given to a particular technique that typically accompanies an ANOVA when one (or more) of the factors has a sensible quantitative representation.

If the levels of the factor are evenly spaced, and sample sizes are equal, then the trend analysis can be accomplished as a sequence of contrast hypothesis tests, using specially chosen linear weights called "orthogonal polynomials."

However, it turns out that we need not perform "trend analysis" that way. There is a much easier way, which we'll demonstrate in this module.

# Sequential Multiple Regression Modeling

In multiple regression, there is a test of significance that is routinely performed when a term is added to the regression equation.

This test assess whether the $R^2$ value has significantly increased with the addition of the term.

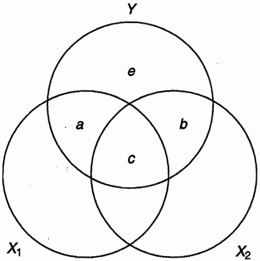
One way to do this is to compute the residual of the new term predicted from the old term(s), and compute the correlation between this residual and the criterion. This correlation is known as the "semipartial" or "part" correlation between the criterion and the last predictor with the earlier predictors "factored out."

The squared semipartial correlation is precisely equal to the change in $R^2$ produced by adding a new predictor.

This is explained graphically by Cohen, Cohen, Aiken and West (2003, p. 72).

# Sequential Multiple Regression Modeling

$$a = sr_1^2 = R^2_{Y.12} - r^2_{Y2},$$
$$b = sr_2^2 = R^2_{Y.12} - r^2_{Y1}.$$




$$r^2_{Y1} = a + c$$
$$r^2_{Y2} = b + c$$
$$R^2_{Y.12} = a + b + c$$

**FIGURE 3.3.1** The ballantine for $X_1$ and $X_2$ with $Y$.

# Sequential Multiple Regression Modeling

When terms are added sequentially, we say the models are a *nested sequence*. Each earlier model may be viewed as a special case of the later model (with some coefficients equal to zero).

When the models are a nested sequence, an $F$-test may be performed to test whether an additional term "adds something significant" to the prediction equation, in the sense of increasing $R^2$ or (equivalently) having a nonzero squared semi-partial correlation.

In standard multiple regression, the very first predictor is, effectively, compared to a model with just an intercept term. The system described above still holds, because an intercept term is a vector with all values equal to 1, so the residuals of the first predictor with this column of 1's factored out is just the set of deviation scores for the first predictor.

The correlation between these deviation scores and $Y$ is, of course, equal to the correlation between $X_1$ and $Y$, so the change in $R^2$ from adding $X_1$ (over and above the intercept) is just the square of the simple Pearson $r$ between $Y$ and $X_1$.

# Sequential Multiple Regression Modeling

There are several aspects of sequential regression modeling that may not be obvious at first glance.

One aspect is that *order matters*!

A variable may not be significant if it is added second, while it might be highly significant if added first.

Let's investigate a couple of aspects of this with a brief numerical example.

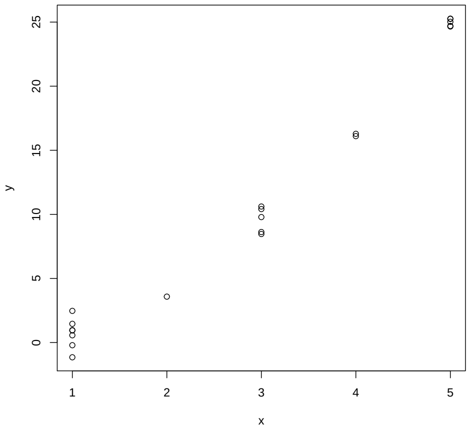
# Sequential Multiple Regression Modeling

The data file *example1.csv* contains 20 observations on an integer predictor $x$ and a criterion $y$.

Suppose, for the sake of argument, we wished to predict $y$ from $x$ and $x^2$, and decided to perform the analysis in a sequential fashion, first analyzing the model with only $x$, then adding $x^2$ to see whether the model improved significantly.

# Sequential Multiple Regression Modeling

```
> example1 <- read.csv("example1.csv")
> attach(example1)
> plot(x,y)
```



# Sequential Multiple Regression Modeling

First we'll compute the model with only the intercept as a predictor, then with $x$ as a predictor, then the model with $x$ and $x^2$ as predictors.

To compare the models, we actually have two options. The "manual" approach involves giving the `anova()` command the models, and asking the function to compare them.

The second approach involves just giving the `anova()` command the full model. In that case, it automatically constructs all sequential tests, starting from the first term. Here we go:

# Sequential Multiple Regression Modeling

## An Example

```
> fit.0 <- lm(y ~ 1)
> fit.1 <- lm(y ~ 1 + x )
> fit.2 <- lm(y ~ 1 + x + I(x^2))
> anova(fit.0,fit.1,fit.2)

Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ 1 + x
Model 3: y ~ 1 + x + I(x^2)
  Res.Df     RSS Df Sum of Sq       F    Pr(>F)
1     19 1846.47
2     18   59.02  1   1787.45 2340.38 < 2.2e-16 ***
3     17   12.98  1     46.04   60.28 5.474e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> anova(fit.2)

Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value    Pr(>F)
x          1 1787.45 1787.45 2340.38 < 2.2e-16 ***
I(x^2)     1   46.04   46.04   60.28 5.474e-07 ***
Residuals 17   12.98    0.76
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Sequential Multiple Regression Modeling

## An Example

Let's look at the summaries of the two models.

We see that both $x$ and $x^2$ are significant predictors, and that the regression coefficient for $x$ in the full model is 0.9763 and the coefficient for $x^2$ is 0.8489.

We can also look at the squared multiple correlations for the two models, and the difference between them.

# Sequential Multiple Regression Modeling

## An Example

```
> summary(fit.1)


Call:
lm(formula = y ~ 1 + x)

Residuals:
    Min      1Q  Median      3Q     Max
-3.1153 -1.2902  0.4886  1.3433  2.7800

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -6.2570     0.8312  -7.528 5.76e-07 ***
x             5.9464     0.2547  23.348 6.56e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.811 on 18 degrees of freedom
Multiple R-squared:  0.968, Adjusted R-squared:  0.9663
F-statistic: 545.1 on 1 and 18 DF,  p-value: 6.556e-15
```

# Sequential Multiple Regression Modeling

## An Example

```
> summary(fit.2)


Call:
lm(formula = y ~ 1 + x + I(x^2))

Residuals:
    Min      1Q  Median      3Q     Max
-1.8435 -0.3942 -0.0105  0.3111  1.7774

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.1332     0.7723  -1.467    0.161
x             0.9763     0.6518   1.498    0.153
I(x^2)        0.8489     0.1093   7.764 5.47e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8739 on 17 degrees of freedom
Multiple R-squared:  0.993, Adjusted R-squared:  0.9921
F-statistic:  1200 on 2 and 17 DF,  p-value: < 2.2e-16
```

# Sequential Multiple Regression Modeling

## An Example

```
> summary(fit.1)$r.squared

[1] 0.9680353

> summary(fit.2)$r.squared

[1] 0.9929684

> summary(fit.2)$r.squared - summary(fit.1)$r.squared

[1] 0.02493308
```

# Sequential Multiple Regression Modeling

## An Example

Next, we demonstrate that the regression coefficient and squared multiple correlation would remain the same if we used the regression residual of $x^2$ with $x$ partialled out, instead of $x^2$, in our model.

Let's try it by first computing the residual, then employing it in the model.

# Sequential Multiple Regression Modeling

## An Example

```
> x2.dot.x <- residuals(lm(I(x^2) ~ x))
> fit.2.b <- lm(y ~ 1 + x + x2.dot.x)
> summary(fit.2.b)


Call:
lm(formula = y ~ 1 + x + x2.dot.x)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8435 -0.3942 -0.0105  0.3111  1.7774

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -6.2570     0.4011 -15.598 1.66e-11 ***
x             5.9464     0.1229  48.378  < 2e-16 ***
x2.dot.x      0.8489     0.1093   7.764 5.47e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8739 on 17 degrees of freedom
Multiple R-squared:  0.993, Adjusted R-squared:  0.9921
F-statistic:  1200 on 2 and 17 DF,  p-value: < 2.2e-16
```

# Sequential Multiple Regression Modeling

## An Example

```
> summary(fit.2.b)$r.squared

[1] 0.9929684

> summary(fit.2.b)$r.squared - summary(fit.1)$r.squared

[1] 0.02493308

> cor(y,x2.dot.x)^2

[1] 0.02493308
```

# Trend Analysis in 1-Way ANOVA

Suppose that, in ANOVA, our factor is actually quantitative, and the levels are evenly spaced.

We wish to test for "significant trend," in the sequential sense. That is, if we plot the cell means versus the values of the quantitative factor, is there a significant linear trend?

And once we factor out the linear trend, is there a significant quadratic trend over and above this linear trend?

Since the values are "stacked" at a handful of points along the $x$-axis, this question amounts to plotting the cell means and computing some sums of squares.

Why? Well, consider *any* prediction you might have for the shape of the plot of cell means. How can you test whether that prediction agrees with the cell means?

# Trend Analysis in 1-Way ANOVA

The answer is surprisingly simple. Simply compute a contrast, using your predictions as contrast weights!

Let's examine this in the case of linear trend. First, let's recall a little algebra.

When sample sizes are equal to $n$, the generalized $t$-statistic we learned about in Pychology 310 has the following form to test the null hypothesis that $\psi = \sum_j w_j \mu_j = 0$.

$$t = \frac{\sum_j w_j \bar{Y}_j}{\sqrt{\left(\sum_j w_j^2 / n\right) MS_{S|A}}} \tag{1}$$

# Trend Analysis in 1-Way ANOVA

Squaring this produces an $F$ with 1 numerator degree of freedom, i.e.,

$$
\begin{aligned}
F &= \frac{n\left(\sum_j w_j^* \bar{Y}_j\right)^2}{MS_{S|A}} && (2)\\
&= \frac{SS_\psi}{MS_{S|A}} && (3)\\
&= \frac{MS_\psi}{MS_{S|A}} && (4)
\end{aligned}
$$

where the $w_j^*$ are weights that sum to zero, and which have a sum of squares equal to 1, and are computed as

$$
w_j^* = \frac{w_j}{\sqrt{\sum_j w_j^2}} \tag{5}
$$

# Trend Analysis in 1-Way ANOVA

So how would we test that there is significant linear trend between the $x$ values for the factor, and the population means of $y$ at the various levels?

This is a test that $\beta_1 = 0$, where $\beta_1$ is the slope of the regression line.

Recall that the formula for $\beta_1$ is

$$\beta_1 = \frac{\rho_{yx}}{\sigma_y \sigma_x} = \frac{\sigma_{yx}}{\sigma_x^2} \tag{6}$$

So $\beta_1$ is zero if and only if $s_{yx} = 0$. Note that the fixed regression model assumes that the $x$ scores are fixed, so that there is no distinction between $\bar{X}$ and $\mu_x$.

# Trend Analysis in 1-Way ANOVA

Moreover, recall that, when computing a covariance, only one of the variables need be converted to deviation score form.

Consequently, a test of $\beta_1 = 0$ is simply a test that

$$\sum_j (\bar{X}_{\bullet j} - \bar{X}_{\bullet\bullet})(\mu_j - \mu) = \sum_j \xi_j \mu_j = 0 \tag{7}$$

Since every observation in the jth group will have the same $x_j$, the $\xi_j$ are contrast weights corresponding to the $x$ values in deviation score form. As we noted in a previous lecture, the test statistic, and $SS_\psi$, are invariant under multiplicative rescaling of the $\xi_j$, and so, for convenience, we can scale them so that they are integers.

So, for example, if there are 5 levels of the (equally spaced) independent variable, the weights will always be $-2, -1, 0, 1, 2$.

# Trend Analysis in 1-Way ANOVA

But what about the quadratic (and cubic, etc.) term?

In a sense, we have already answered that question. We saw in the preceding section that the $\beta_2$ for the quadratic term is actually equivalent to the regression coefficient in a simple regression between $y$ and the $x^2$ values with the $x$ values factored out.

And, the cubic weights are the residuals of the cubed values predicted from the values and their squares, etc.

We can compute *those* values in a few lines of R code. (In some cases, we rescale the results to get integers. It would make no difference in the computations, but I do this so that you can compare these results to those in the table of "orthogonal polynomials" in RDASA3, p.702.) If there are 5 levels of a factor, the weights are. . .

# Trend Analysis in 1-Way ANOVA

```
> x.wts <- c(1,2,3,4,5)
> linear.wts <- x.wts - mean(x.wts)
> quadratic.wts <- residuals(lm(linear.wts^2 ~ linear.wts))
> cubic.wts <- residuals(lm(linear.wts^3 ~ linear.wts + quadratic.wts))
> cubic.wts <- zapsmall(cubic.wts)/1.2
> quartic.wts <- residuals(lm(linear.wts^4 ~ linear.wts + quadratic.wts + cubic.wts))
> quartic.wts <- quartic.wts/quartic.wts[1]
> linear.wts

[1] -2 -1  0  1  2

> quadratic.wts

 1  2  3  4  5
 2 -1 -2 -1  2

> cubic.wts

 1  2  3  4  5
-1  2  0 -2  1

> quartic.wts

 1  2  3  4  5
 1 -4  6 -4  1
```

We load in the `gsr` data used in the example shown in Table 11.3. We then run a 1-Way ANOVA to obtain the $SS_A$ and the $SS_{S|A}$ error term.

```
> gsr <- read.csv("GSR data.csv")
> gsr$STIMULUS <- factor(gsr$STIMULUS)
> anova(lm(GSR2 ~STIMULUS,data=gsr))

Analysis of Variance Table

Response: GSR2
          Df  Sum Sq Mean Sq F value   Pr(>F)
STIMULUS   4  35.261  8.8153   3.914 0.008234 **
Residuals 45 101.351  2.2522
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Trend Analysis: A 1-Way Example

Next, we get the cell means, and from them and the weights, it is easy to compute the sum of squares for each trend. In the function `TrendAnalysis` available on the website, I automate the process.

If you study the code, you will notice that the vast majority of effort is involved in setting up the table labels and doing housekeeping.

```
> ybars <-  aggregate(GSR2 ~ STIMULUS, mean,data=gsr)[,2]
> n <- 10
> wts <- rbind(linear.wts,quadratic.wts,cubic.wts,quartic.wts)
> MSerror <-101.351/45
> TrendAnalysis(wts,ybars,n,MSerror,3)

               SS df      MS         F    p.value
Linear    14.8225  1 14.8225 6.5812128 0.01370937
  Remain1 20.4387  3  6.8129 3.0249381 0.03919296
Quadratic 14.7875  1 14.7875 6.5656728 0.01381388
  Remain2  5.6512  2  2.8256 1.2545707 0.29498162
Cubic      3.6100  1  3.6100 1.6028456 0.21201553
  Remain3  2.0412  1  2.0412 0.9062959 0.34618477
```

# Trend Analysis as Regression

Although the contrast-based approach makes it reasonably easy to perform trend analysis, it turns out that most of the effort we just expended is completely unnecessary.

Trend analysis is nothing more than sequential regression, using sequentially "orthogonalized" powers of the independent factor as predictors.

Along the way, we should make a side point. We can easily perform the generalized $t$-test as an $F$ test using linear regression.

Let's see how this works.

# Trend Analysis as Regression

## Contrast Tests via Regression

Suppose we had two means, and we wish to perform the test that $\mu_1 - \mu_2 = 0$. Suppose, for example, the data were as shown below. We can perform the contrast test as a regression test.

```
> Group1 <- c(1,2,3)
> Group2 <- c(6,8,10)
> Y <- c(Group1,Group2)
> Wts <- c(1/3,1/3,1/3,-1/3,-1/3,-1/3)
> summary(lm(Y~Wts))


Call:
lm(formula = Y ~ Wts)

Residuals:
         1          2          3          4          5          6
-1.000e+00  1.234e-15  1.000e+00 -2.000e+00  1.793e-16  2.000e+00

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0000     0.6455   7.746  0.00150 **
Wts          -9.0000     1.9365  -4.648  0.00968 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.581 on 4 degrees of freedom
Multiple R-squared:  0.8438, Adjusted R-squared:  0.8047
F-statistic:  21.6 on 1 and 4 DF,  p-value: 0.009679

> source("http://www.statpower.net/R312/t1.R")
> output <- two.sample.t(mean(Group1),sd(Group1),
+     length(Group1),mean(Group2),sd(Group2),length(Group2))
> output$t^2

[1] 21.6
```

We can see that the square of the *t* statistic is identical to the *F* statistic for the regression analysis.

# Trend Analysis as Regression

## Contrast Tests via Regression

This works for unequal $n$ as well.

```
> Group1 <- c(1,2,3)
> Group2 <- c(6,8)
> Score <- c(Group1,Group2)
> Wts <- c(1/3,1/3,1/3,-1/2,-1/2)
> summary(lm(Score~Wts))


Call:
lm(formula = Score ~ Wts)

Residuals:
         1          2          3          4          5 
-1.000e+00 -1.407e-17  1.000e+00 -1.000e+00  1.000e+00 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   4.0000     0.5164   7.746  0.00447 **
Wts          -6.0000     1.2649  -4.743  0.01777 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.155 on 3 degrees of freedom
Multiple R-squared:  0.8824, Adjusted R-squared:  0.8431 
F-statistic:  22.5 on 1 and 3 DF,  p-value: 0.01777

> output <- two.sample.t(mean(Group1),sd(Group1),length(Group1),
+                        mean(Group2),sd(Group2),length(Group2))
> output

$t
[1] -4.743416

$df
[1] 3

$p.value
[1] 0.0177719

> output$t^2

[1] 22.5
```

# Trend Analysis as Regression

## Running the Trend Analysis as Regression

We read the data back in so that STIMULUS is treated as numeric. To obtain the significance tests for linear, quadratic, cubic, and quartic trend, we simply perform the regression analysis.

```
> gsr <- read.csv("gsr data.csv")
> anova(lm(GSR2 ~ STIMULUS + I(STIMULUS^2) +
+   I(STIMULUS^3) + I(STIMULUS^4),data=gsr))

Analysis of Variance Table


Response: GSR2
              Df  Sum Sq Mean Sq F value  Pr(>F)
STIMULUS       1  14.822 14.8225  6.5812 0.01371 *
I(STIMULUS^2)  1  14.787 14.7875  6.5657 0.01381 *
I(STIMULUS^3)  1   3.610  3.6100  1.6028 0.21202
I(STIMULUS^4)  1   2.041  2.0412  0.9063 0.34618
Residuals     45 101.351  2.2522
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Trend Analysis as Regression

## Plotting Trend Analysis Results

Note that the linear and quadratic components are significant, but the cubic and quartic are not.

Moreover, the *p*-values match those in RDASA3 Table 11.3.

We can reanalyze in terms of just the linear and quadratic terms (plus the intercept).

# Trend Analysis as Regression

## Plotting Trend Analysis Results

Now, let's proceed to try to replicate some of the other results in the chapter. Figure 11.1a presents a plot of *trend components*.

In figure 11.1a, the intercept (Grand Mean), linear, and quadratic terms are plotted separately. Summed together, they produce the "predicted" group means that are plotted along with the observed cell means in Figure 11.1b.

MWL derive coefficients for predicting the means as a function of the orthogonal polynomial weights given in their Table C.6.

Once the coefficients are calculated as in Equations 11.16–11.17, they can be applied to the orthogonal polynomial weights to produce predicted values, as shown on p. 284 of RDASA3. (*Note: There are some minor typographical or rounding errors in these calculations.*)

However, using the regression approach, we can bypass these hand calculations.

# Trend Analysis as Regression

## Plotting Trend Analysis Results

As in the book, we shall confine ourselves to the linear and quadratic terms. We start by producing the 3 sequential models involving just the intercept, then the intercept and linear term, and finally the intercept, linear, and quadratic terms.

```
> fit0 <- lm(GSR2 ~ 1,data=gsr)
> fit1 <- lm(GSR2 ~ 1 + I(STIMULUS), data=gsr)
> fit2 <- lm(GSR2 ~ 1 + STIMULUS + I(STIMULUS^2), data=gsr)
```

Next, we set up the *sequentially* predicted trend components.

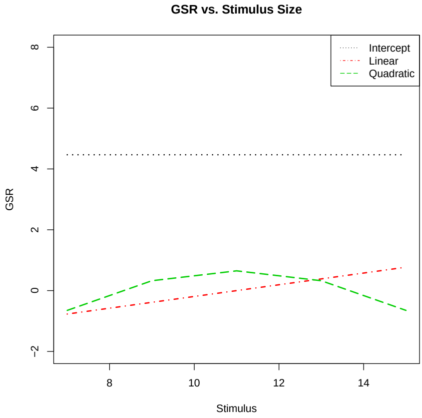
```
> y0 <- predict(fit0)
> y1 <- predict(fit1) - y0
> ## NOTE, y1 also equals predict(fit1) - predict(fit0)
> y2 <- predict(fit2) - y1 -y0
> ## NOTE, y2 also equals predict(fit2) - predict(fit1)
```

We'll plot them on the next slide.

# Trend Analysis as Regression

## Plotting Trend Analysis Results

```
> plot(gsr$STIMULUS,y0,type="l",ylim=c(-2,8),
+   lty=21,col=1,xlab="Stimulus",ylab="GSR",
+   main="GSR vs. Stimulus Size",lwd=2)
> lines(gsr$STIMULUS,y1,lty=22,col=2,lwd=2)
> lines(gsr$STIMULUS,y2,lty=23,col=3,lwd=2)
> legend("topright",c("Intercept","Linear","Quadratic"),
+  lty=c(21,22,23),col=c(1,2,3))
```



# Trend Analysis as Regression

## Plotting Trend Analysis Results

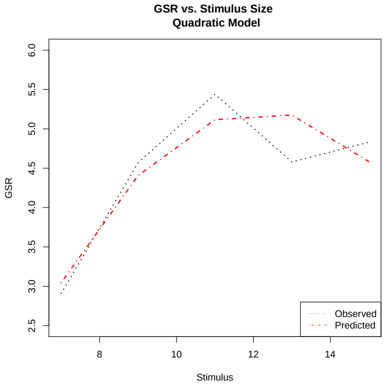
We can also reproduce Figure 11.1b. This figure plots both the predicted group means (from the full quadratic model), and the observed group means. We can grab the group means easily in several ways.

Note that, when we plot the group means, we use all the `STIMULUS` data, but when we plot the group means, we just use the five distinct values of 7,9,11,13,15.

# Trend Analysis as Regression

## Plotting Trend Analysis Results

```
> group.data <- aggregate(GSR2 ~ STIMULUS,mean,data=gsr)
> stimulus.values <- group.data[,1]
> group.means <- group.data[,2]
> plot(stimulus.values,group.means,type="l",ylim=c(2.5,6),
+   lty=21,col=1,xlab="Stimulus",ylab="GSR",
+   main="GSR vs. Stimulus Size \n Quadratic Model",
+   lwd=2
+   )
> lines(gsr$STIMULUS,predict(fit2),lty=22,col=2,lwd=2)
> legend("bottomright",c("Observed","Predicted"),
+   lty=c(21,22),col=c(1,2))
```



# Interactions in Trend Analysis

In Section 11.5 of RDASA3, the authors discuss extending trend analysis to the case of multi-factor designs.

They present data for GSR responses to 3 categories of patients (MS = *Mildly Schizophrenic*, SS = *Severely Schizophrenic*, C = *Control*). Again, the subjects responded to stimulus heights of 7, 9, 11, 13, and 15 inches.

In the context of this experiment, we might ask whether the linear or quadratic components of trend differ across various participant categories, and, if so, precisely how. One way of getting at this is to compute interaction effect tests separately for trend components.

On the next slide, we show how to reproduce the results in Table 11.5 of RDASA3. We see that, overall, there are significant linear and quadratic trends in the *Stimulus* variable, but that there is an interaction between the quadratic trend component and the `CATEGORY` variable that suggests this trend varies across diagnostic categories.

# Interactions in Trend Analysis

```
> gsr <- read.csv("GSR2 data.csv")
> fit.1 <- lm(GSR ~ Category * factor(Stimulus), data = gsr)
> fit.2 <- lm(GSR ~ Category *
+  (Stimulus + I(Stimulus^2) + I(Stimulus^3) + I(Stimulus^4))
+ ,data=gsr)
> anova(fit.1)

Analysis of Variance Table

Response: GSR
                          Df Sum Sq Mean Sq F value   Pr(>F)
Category                   2  7.145  3.5723  3.0123 0.055164 .
factor(Stimulus)           4 21.628  5.4071  4.5594 0.002377 **
Category:factor(Stimulus)  8 19.240  2.4050  2.0280 0.054309 .
Residuals                 75 88.943  1.1859
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Interactions in Trend Analysis

```
> anova(fit.2)

Analysis of Variance Table

Response: GSR
                        Df Sum Sq Mean Sq F value    Pr(>F)
Category                 2  7.145  3.5723  3.0123 0.0551644 .
Stimulus                 1  5.408  5.4080  4.5602 0.0359879 *
I(Stimulus^2)            1 15.750 15.7500 13.2809 0.0004915 ***
I(Stimulus^3)            1  0.072  0.0720  0.0607 0.8060460
I(Stimulus^4)            1  0.398  0.3982  0.3358 0.5640035
Category:Stimulus        2  1.992  0.9962  0.8400 0.4357305
Category:I(Stimulus^2)  2 11.756  5.8780  4.9565 0.0095118 **
Category:I(Stimulus^3)  2  1.369  0.6847  0.5773 0.5638682
Category:I(Stimulus^4)  2  4.122  2.0611  1.7380 0.1828846
Residuals               75 88.943  1.1859
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Interactions in Trend Analysis

Since only the linear and quadratic terms are significant, and the only significant interaction is between `CATEGORY` and the quadratic trend component, a reasonable approach to a follow-up analysis would be to construct plots of the quadratic trend for each of the three categories, to try to pinpoint where the differences lie.

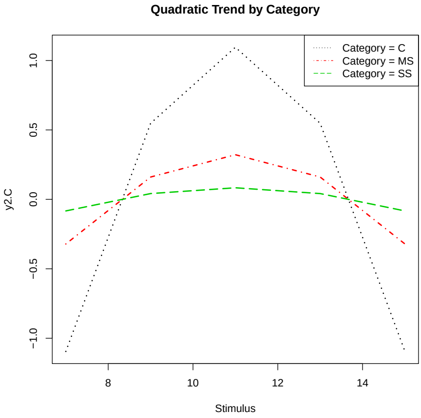
To do that, we simply extract the data by category, and perform an analysis similar to the one above for each category.

# Interactions in Trend Analysis

```
> Stimulus <- c(7,9,11,13,15)
> S <- data.frame(Stimulus)
> fit1.C <- lm(GSR ~ 1 + Stimulus, data=subset(gsr,Category=="C"))
> fit2.C <- lm(GSR ~ 1 + Stimulus + I(Stimulus^2),
+   data=subset(gsr,Category=="C"))
> y2.C <- predict(fit2.C,newdata=S)-predict(fit1.C,newdata=S)
> fit1.MS <- lm(GSR ~ 1 + Stimulus, data=subset(gsr,Category=="MS"))
> fit2.MS <- lm(GSR ~ 1 + Stimulus + I(Stimulus^2),
+   data=subset(gsr,Category=="MS"))
> y2.MS <- predict(fit2.MS,newdata=S)-predict(fit1.MS,newdata=S)
> fit1.SS <- lm(GSR ~ 1 + Stimulus, data=subset(gsr,Category=="SS"))
> fit2.SS <- lm(GSR ~ 1 + Stimulus + I(Stimulus^2),
+   data=subset(gsr,Category=="SS"))
> y2.SS <- predict(fit2.SS,newdata=S)-predict(fit1.SS,newdata=S)
```

# Interactions in Trend Analysis

```
> plot(Stimulus,y2.C,type="l",lty=21,col=1,
+  main="Quadratic Trend by Category",lwd=2)
> lines(Stimulus,y2.MS,lty=22,col=2,lwd=2)
> lines(Stimulus,y2.SS,lty=23,col=3,lwd=2)
> legend("topright",
+ c("Category = C","Category = MS","Category = SS"),
+ lty=21:23,col=1:3)
```



We have seen how to perform sophisticated trend analysis calculations in just a few lines of R code.

One significant payoff of the "regression approach" to trend analysis is that it allows us to recognize that trend analysis is simply another kind of sequential regression analysis.

Another potential payoff is that, should we inherit a data set in which the quantitative levels of an independent variable are not equally spaced, we don't have to change anything! We simply do the analysis.

Other approaches require, at the least, recomputation of the orthogonal polynomial weights. This can be done exactly as we computed the standard weights early in this module.

Suppose we have 5 levels, and they are 7,8,9,11,15. In deviation score form, these are $-3, -2, -1, 1, 5$. We can now derive quadratic, cubic and quartic weights using the same sequential regression approach shown earlier.

However, there is simply no need to do this. As we have shown, all the calculations can be done directly, using the original level values.