Agenda

1. Interpreting nested F-tests
2. Model visualization
3. Polynomial regression

Nested F-tests Interpreting nested F-tests.

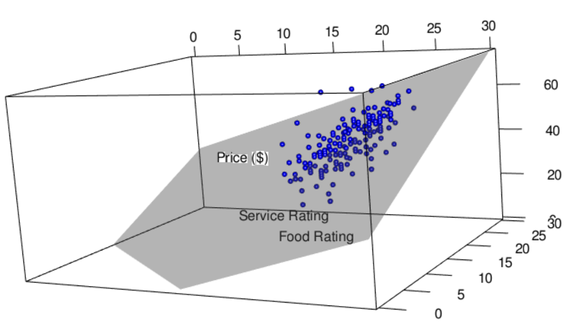
```
> bloodp <- read.csv("http://www.math.smith.edu/~bbaumer/mth247/labs/bloodpress.csv")
> mfull <- lm(BP ~ ., data=bloodp)
> m1 <- lm(BP ~ Weight, data=bloodp)
> m2 <- lm(BP ~ Weight +  Age, data=bloodp)
> m3 <- lm(BP ~ Weight +  Age + Dur + Stress, data=bloodp)
> # Add the models in ascending order of complexity.
> anova(m1, m2, m3, mfull)
```

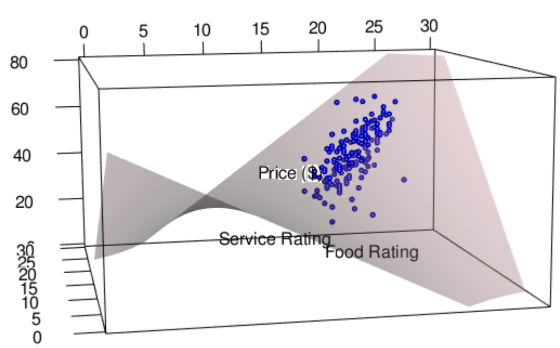
```
Analysis of Variance Table

Model 1: BP ~ Weight
Model 2: BP ~ Weight + Age
Model 3: BP ~ Weight + Age + Dur + Stress
Model 4: BP ~ Age + Weight + BSA + Dur + Pulse + Stress
  Res.Df    RSS Df Sum of Sq        F    Pr(>F)
1     18 54.528
2     17  4.824  1    49.704 299.7198 2.327e-10 ***
3     15  4.545  2     0.279   0.8406  0.453611
4     13  2.156  2     2.389   7.2037  0.007843 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

More model visualization Back to our Italian restaurant data, we have looked at these models in 3D. One was a simple plane in 3D, and the other was a warped plane, because of the interaction between two numeric variables.

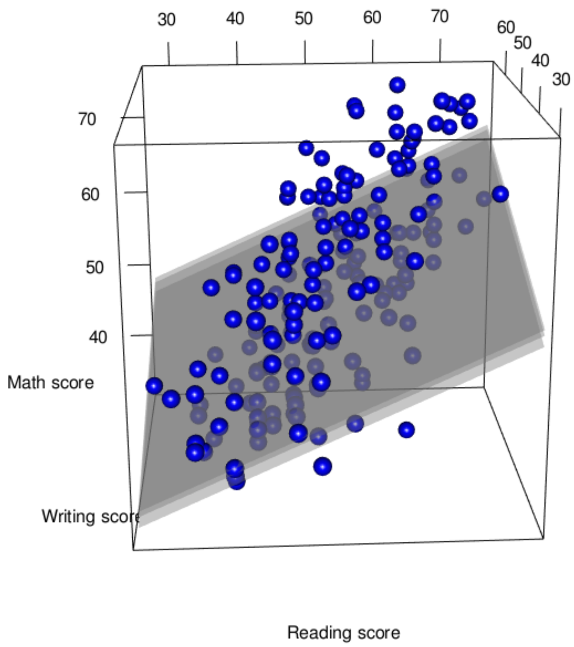
```
> mflat <- lm(Price ~ Food + Service, data=NYC)
> mwarp <- lm(Price ~ Food + Service + Food * Service, data=NYC)
```

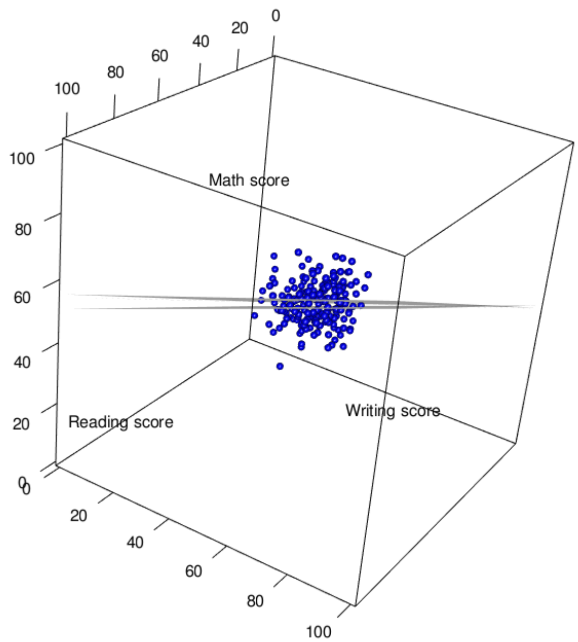




We were also talking about models with parallel planes and those with intersecting planes.

```
> m.parallel <- lm(math~read+write+ses, data=hsb2)
> m.indep <-lm(math~read+write+ses+read*ses+write*ses, data=hsb2)
```





These plots have different shapes, depending on the way we choose to include terms in our model. Including a categorical variable can lead to parallel slopes or parallel planes, and an interaction between a categorical variable and a quantitative variable allows those lines or planes to have differnt slopes. Two quantative variables interacting leads to warped planes. But, what if a variable interacts with itself?

Almost always, we include the constant and linear terms in a model, although we might discover that they are not needed if other terms are added. The question is generally whether to include the quadratic and bilinear terms.

```
> require(mosaic)
> NYC <- read.csv("http://www.math.smith.edu/~bbaumer/mth241/nyc.csv")
> m1 <- lm(Price~Food, data=NYC)
> summary(m1)$adj.r.squared

[1] 0.389528

> mquad <- lm(Price ~ Food + I(Food^2), data=NYC)
> summary(mquad)

Call:
lm(formula = Price ~ Food + I(Food^2), data = NYC)

Residuals:
     Min       1Q   Median       3Q      Max
-21.2196  -4.6185   0.2306   3.9387  27.2306

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  56.9185    53.1993   1.070    0.286
Food         -4.3853     5.1887  -0.845    0.399
I(Food^2)     0.1778     0.1257   1.414    0.159

Residual standard error: 7.239 on 165 degrees of freedom
```

```
Multiple R-squared:  0.4004,	Adjusted R-squared:  0.3932
F-statistic:  55.1 on 2 and 165 DF,  p-value: < 2.2e-16

> # same result, different code
> # lm(Price ~ poly(Food, 2, raw=TRUE), data=NYC)
> plotModel(mquad)
```

You don't want to go too crazy with polynomials, because you can end up overfitting your data.

```
> # xyplot(y~x, data=d1, type=c("p", "r"), xlab="", ylab="")
> # mcube <- lm(y~poly(x, 3, raw=TRUE), data=d1)
> # plotModel(mcube, xlab="", ylab="")
> # mlots <- lm(y~poly(x, 26, raw=TRUE), data=d1)
> # plotModel(mlots, xlab="", ylab="")
> # summary(mlots)$r.squared
```