QCB 508 – Week 9

John D. Storey

Spring 2020

Contents

OLS in R

Example: Davis Data

```
> data("Davis", package="carData")
> htwt <- tbl_df(Davis)
> htwt[12,c(2,3)] <- htwt[12,c(3,2)]
> head(htwt)
# A tibble: 6 x 5
  sex   weight height repwt repht
  <fct>  <int>  <int> <int> <int>
1 M         77    182    77   180
2 F         58    161    51   159
3 F         53    161    54   158
4 M         68    177    70   175
5 F         59    157    59   155
6 M         76    170    76   165
```

R implements OLS of multiple explanatory variables exactly the same as with a single explanatory variable, except we need to show the sum of all explanatory variables that we want to use.

```
> lm(weight ~ height + sex, data=htwt)

Call:
```

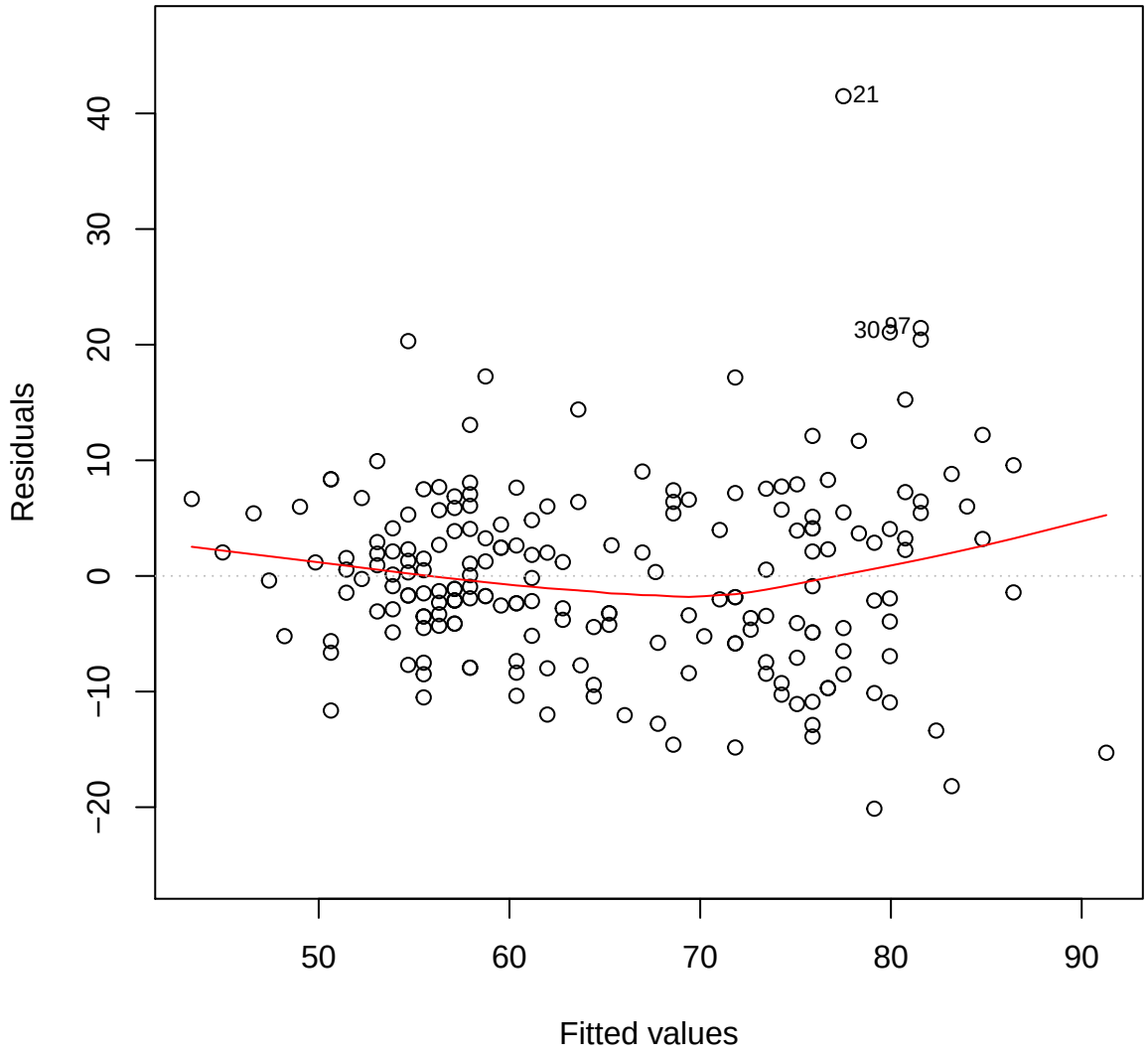
```
lm(formula = weight ~ height + sex, data = htwt)

Coefficients:
(Intercept)       height         sexM
   -76.6167       0.8106       8.2269
```

Weight Regressed on Height + Sex

```
> summary(lm(weight ~ height + sex, data=htwt))

Call:
lm(formula = weight ~ height + sex, data = htwt)

Residuals:
    Min      1Q  Median      3Q     Max
-20.131  -4.884  -0.640   5.160  41.490

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -76.6167    15.7150  -4.875 2.23e-06 ***
height        0.8105     0.0953   8.506 4.50e-15 ***
sexM          8.2269     1.7105   4.810 3.00e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.066 on 197 degrees of freedom
Multiple R-squared:  0.6372,    Adjusted R-squared:  0.6335
F-statistic:   173 on 2 and 197 DF,  p-value: < 2.2e-16
```

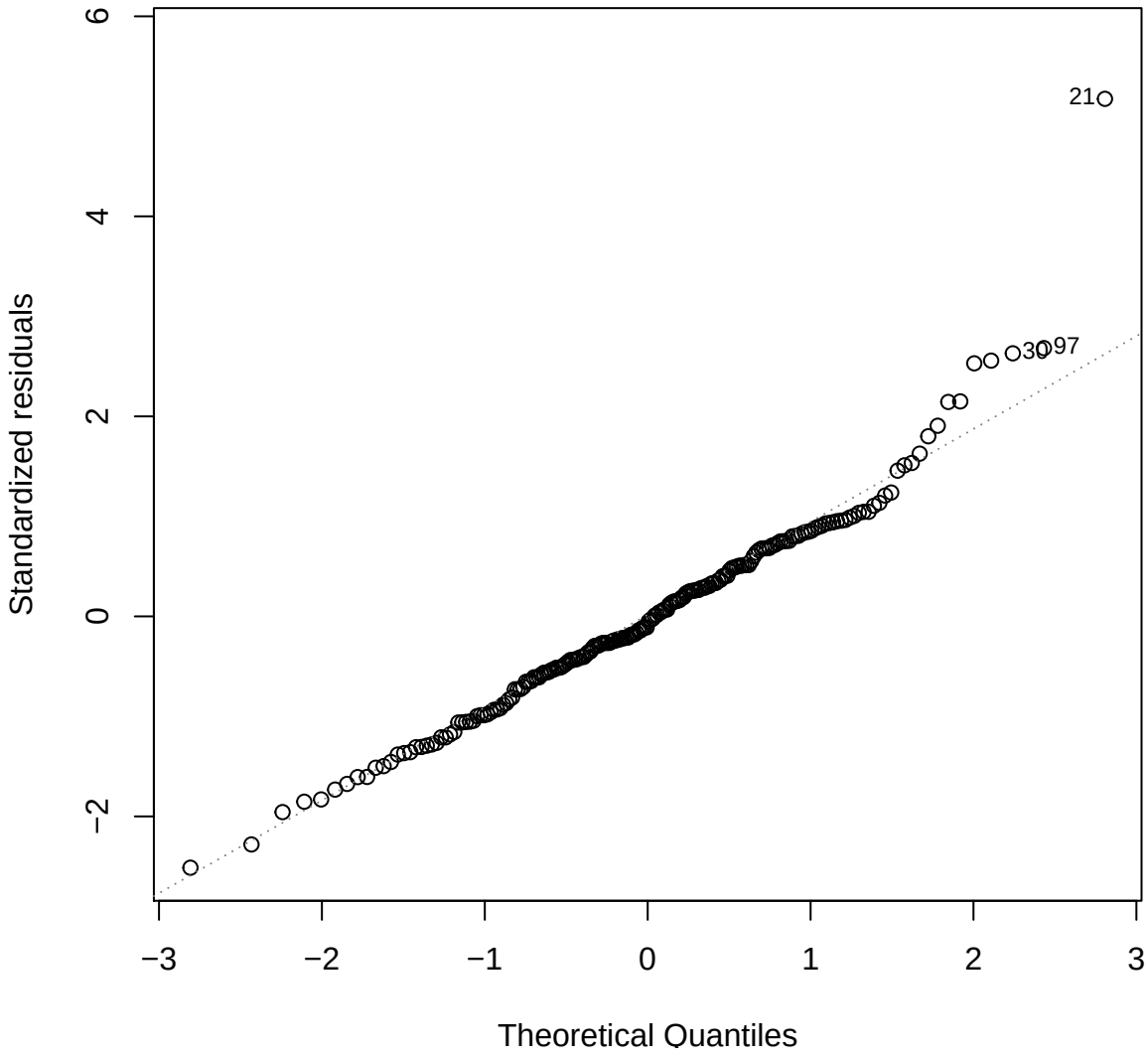
Residual Distribution

```
> myfit <- lm(weight ~ height + sex, data=htwt)
> plot(myfit, which=1)
```



Normal Residuals Check

```
> plot(myfit, which=2)
```



One Variable, Two Scales

We can include a single variable but on two different scales:

```
> htwt <- htwt %>% mutate(height2 = height^2)
> summary(lm(weight ~ height + height2, data=htwt))

Call:
lm(formula = weight ~ height + height2, data = htwt)

Residuals:
    Min      1Q  Median      3Q     Max
-24.265  -5.159  -0.499   4.549  42.965

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 107.117140 175.246872   0.611    0.542
height       -1.632719   2.045524  -0.798    0.426
height2       0.008111   0.005959   1.361    0.175
```

```
Residual standard error: 8.486 on 197 degrees of freedom
Multiple R-squared:  0.5983,    Adjusted R-squared:  0.5943
F-statistic: 146.7 on 2 and 197 DF,  p-value: < 2.2e-16
```

Interactions

It is possible to include products of explanatory variables, which is called an *interaction*.

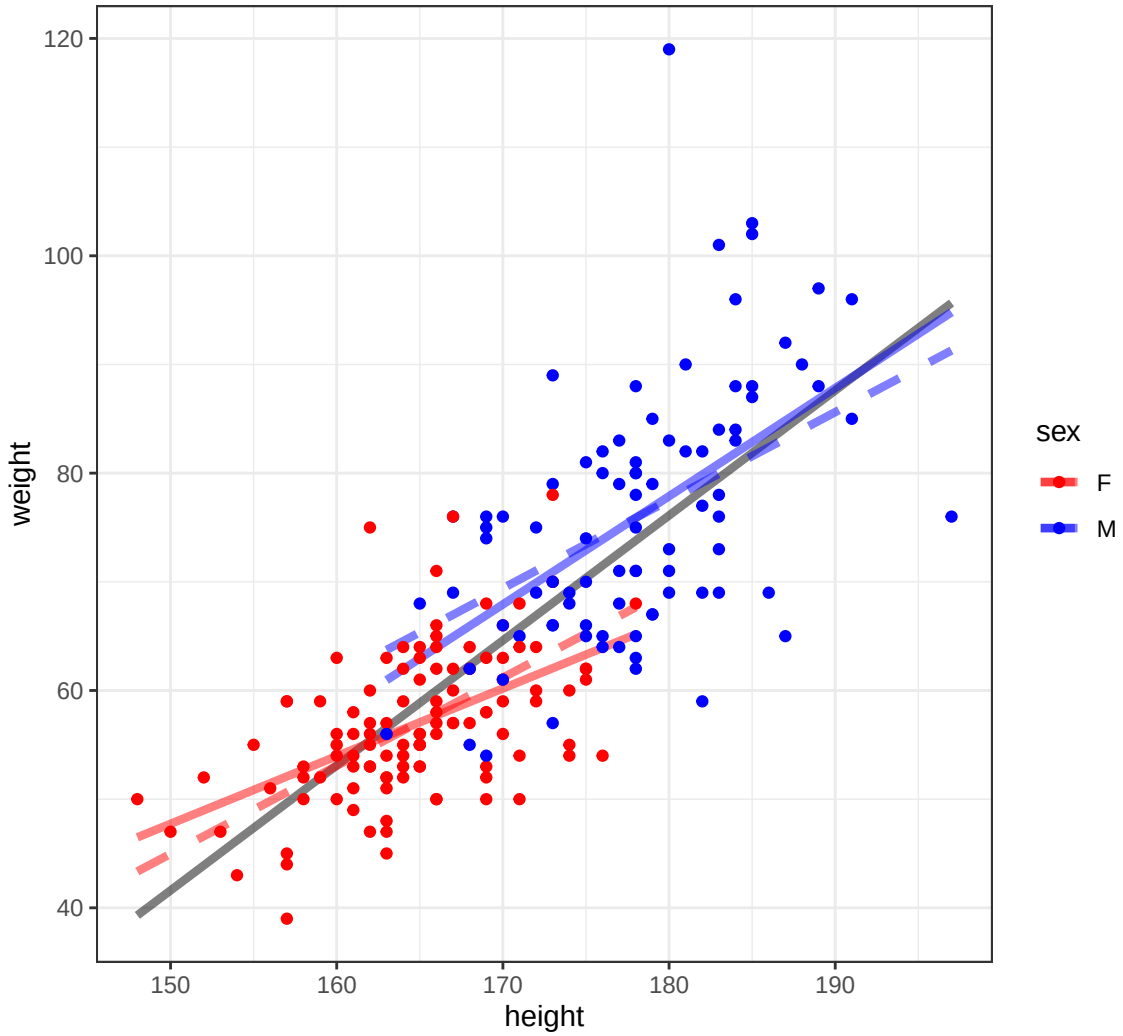
```
> summary(lm(weight ~ height + sex + height:sex, data=htwt))

Call:
lm(formula = weight ~ height + sex + height:sex, data = htwt)

Residuals:
    Min      1Q  Median      3Q     Max
-20.869  -4.835  -0.897   4.429  41.122

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -45.6730    22.1342  -2.063   0.0404 *
height        0.6227     0.1343   4.637 6.46e-06 ***
sexM        -55.6571    32.4597  -1.715   0.0880 .
height:sexM   0.3729     0.1892   1.971   0.0502 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.007 on 196 degrees of freedom
Multiple R-squared:  0.6442,    Adjusted R-squared:  0.6388
F-statistic: 118.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

More on Interactions

What happens when there is an interaction between a quantitative explanatory variable and a factor explanatory variable? In the next plot, we show three models:

- Grey solid: `lm(weight ~ height, data=htwt)`
- Color dashed: `lm(weight ~ height + sex, data=htwt)`
- Color solid: `lm(weight ~ height + sex + height:sex, data=htwt)`

Visualizing Three Different Models



Categorical Explanatory Variables

Example: Chicken Weights

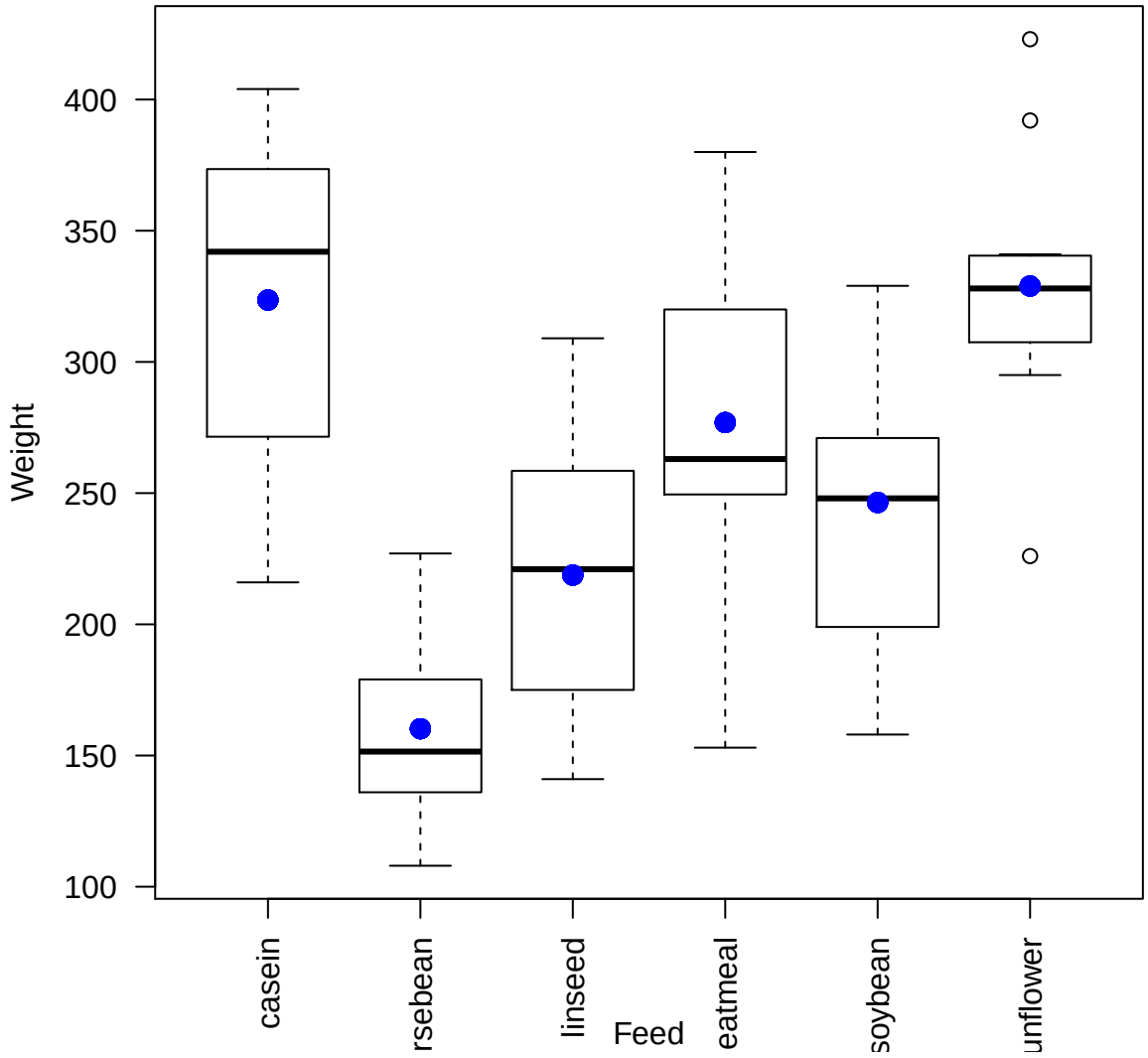
```
> data("chickwts", package="datasets")
> head(chickwts)
  weight      feed
1    179 horsebean
2    160 horsebean
3    136 horsebean
4    227 horsebean
5    217 horsebean
6    168 horsebean
> summary(chickwts$feed)
   casein horsebean   linseed  meatmeal   soybean sunflower
       12        10        12        11        14        12
```

Factor Variables in `lm()`

```
> chick_fit <- lm(weight ~ feed, data=chickwts)
> summary(chick_fit)

Call:
lm(formula = weight ~ feed, data = chickwts)

Residuals:
     Min       1Q   Median       3Q      Max
-123.909  -34.413    1.571   38.170  103.091

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   323.583     15.834  20.436  < 2e-16 ***
feedhorsebean -163.383     23.485  -6.957 2.07e-09 ***
feedlinseed   -104.833     22.393  -4.682 1.49e-05 ***
feedmeatmeal   -46.674     22.896  -2.039 0.045567 *
feedsoybean    -77.155     21.578  -3.576 0.000665 ***
feedsunflower    5.333     22.393   0.238 0.812495
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 54.85 on 65 degrees of freedom
Multiple R-squared:  0.5417,    Adjusted R-squared:  0.5064
F-statistic: 15.36 on 5 and 65 DF,  p-value: 5.936e-10
```

Plot the Fit

```
> plot(chickwts$feed, chickwts$weight, xlab="Feed", ylab="Weight", las=2)
> points(chickwts$feed, chick_fit$fitted.values, col="blue", pch=20, cex=2)
```



ANOVA (Version 1)

ANOVA (*analysis of variance*) was originally developed as a statistical model and method for comparing differences in mean values between various groups.

ANOVA quantifies and tests for differences in response variables with respect to factor variables.

In doing so, it also partitions the total variance to that due to within and between groups, where groups are defined by the factor variables.

`anova()`

The classic ANOVA table:

```
> anova(chick_fit)
Analysis of Variance Table

Response: weight
          Df Sum Sq Mean Sq F value    Pr(>F)
feed       5 231129   46226  15.365 5.936e-10 ***
Residuals 65 195556    3009
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> n <- length(chick_fit$residuals) # n <- 71
> (n-1)*var(chick_fit$fitted.values)
[1] 231129.2
> (n-1)*var(chick_fit$residuals)
[1] 195556
> (n-1)*var(chickwts$weight) # sum of above two quantities
[1] 426685.2
> (231129/5)/(195556/65) # F-statistic
[1] 15.36479
```

How It Works

```
> levels(chickwts$feed)
[1] "casein"    "horsebean" "linseed"   "meatmeal"  "soybean"
[6] "sunflower"
> head(chickwts, n=3)
  weight      feed
1    179 horsebean
2    160 horsebean
3    136 horsebean
> tail(chickwts, n=3)
   weight   feed
69    222 casein
70    283 casein
71    332 casein
> x <- model.matrix(weight ~ feed, data=chickwts)
> dim(x)
[1] 71  6
```

Top of Design Matrix

```
> head(x)
  (Intercept) feedhorsebean feedlinseed feedmeatmeal
1           1             1           0            0
2           1             1           0            0
3           1             1           0            0
4           1             1           0            0
5           1             1           0            0
6           1             1           0            0
  feedsoybean feedsunflower
1           0             0
2           0             0
3           0             0
4           0             0
5           0             0
6           0             0
```

Bottom of Design Matrix

```
> tail(x)
   (Intercept) feedhorsebean feedlinseed feedmeatmeal
66           1             0           0            0
67           1             0           0            0
68           1             0           0            0
69           1             0           0            0
70           1             0           0            0
71           1             0           0            0
   feedsoybean feedsunflower
66           0             0
67           0             0
68           0             0
69           0             0
70           0             0
71           0             0
```

Model Fits

```
> chick_fit$fitted.values %>% round(digits=4) %>% unique()
[1] 160.2000 218.7500 246.4286 328.9167 276.9091 323.5833
```

```
> chickwts %>% group_by(feed) %>% summarize(mean(weight))
# A tibble: 6 x 2
  feed      `mean(weight)`
  <fct>              <dbl>
1 casein              324.
2 horsebean           160.
3 linseed             219.
4 meatmeal            277.
5 soybean             246.
6 sunflower           329.
```

Another ANOVA Function

```
> aov(weight ~ feed, data=chickwts)
Call:
   aov(formula = weight ~ feed, data = chickwts)

Terms:
                    feed Residuals
Sum of Squares  231129.2  195556.0
Deg. of Freedom        5        65

Residual standard error: 54.85029
Estimated effects may be unbalanced
```

```
> summary(aov(weight ~ feed, data=chickwts))
            Df Sum Sq Mean Sq F value   Pr(>F)
feed         5 231129   46226   15.37 5.94e-10 ***
Residuals   65 195556    3009
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Compare to:

```
> anova(lm(weight ~ feed, data=chickwts))
Analysis of Variance Table

Response: weight
          Df Sum Sq Mean Sq F value    Pr(>F)
feed       5 231129   46226  15.365 5.936e-10 ***
Residuals 65 195556    3009
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Variable Transformations

Rationale

In order to obtain reliable model fits and inference on linear models, the model assumptions described earlier must be satisfied.

Sometimes it is necessary to *transform* the response variable and/or some of the explanatory variables.

This process should involve data visualization and exploration.

Power and Log Transformations

It is often useful to explore power and log transforms of the variables, e.g., $\log(y)$ or y^λ for some λ (and likewise $\log(x)$ or x^λ).

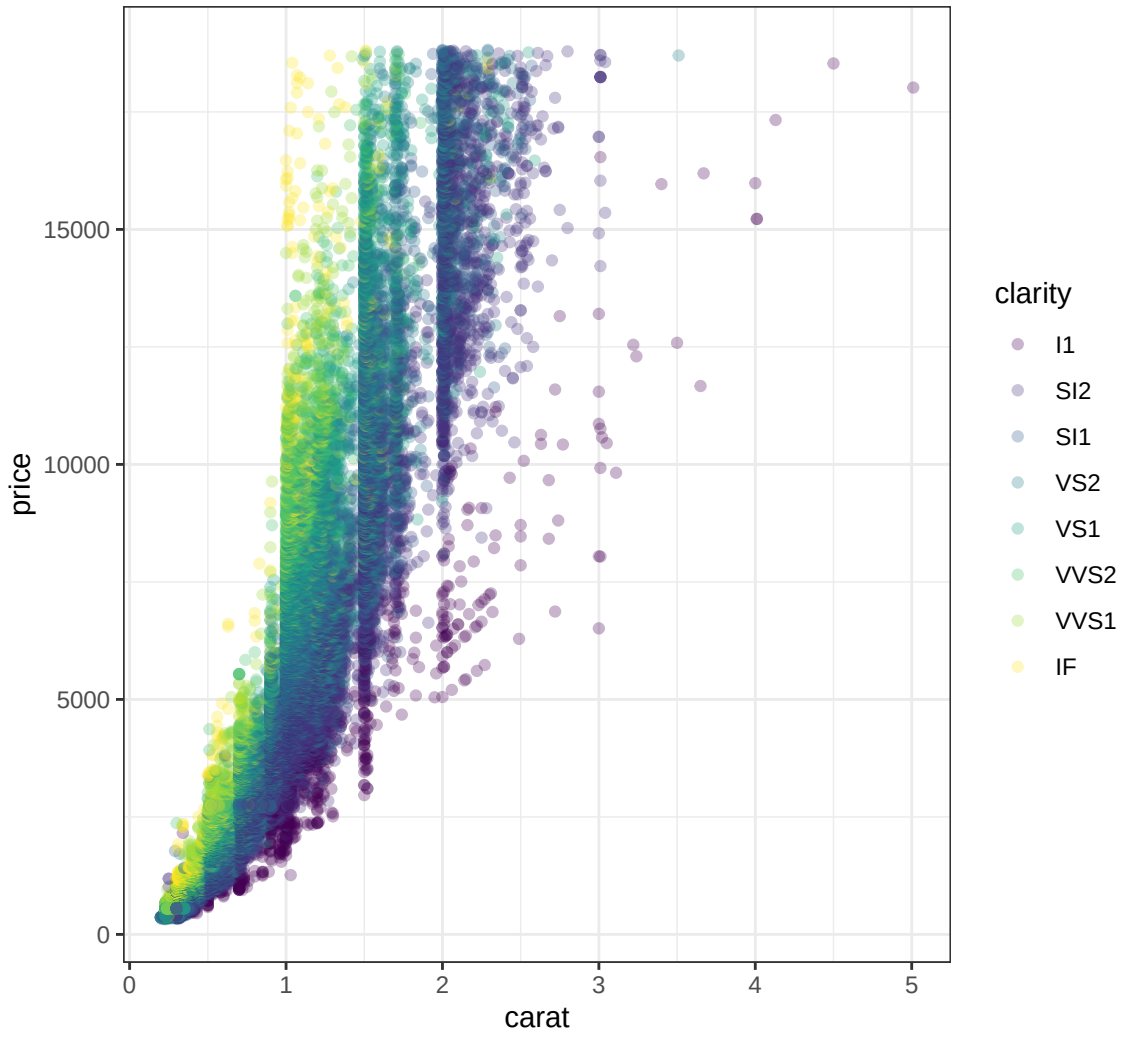
You can read more about the Box-Cox family of power transformations.

`Diamonds` Data

```
> data("diamonds", package="ggplot2")
> head(diamonds)
# A tibble: 6 x 10
  carat cut   color clarity depth table price     x     y     z
  <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1 0.23  Ideal E     SI2      61.5    55   326  3.95  3.98  2.43
2 0.21  Prem~ E     SI1      59.8    61   326  3.89  3.84  2.31
3 0.23  Good  E     VS1      56.9    65   327  4.05  4.07  2.31
4 0.290 Prem~ I     VS2      62.4    58   334  4.2   4.23  2.63
5 0.31  Good  J     SI2      63.3    58   335  4.34  4.35  2.75
6 0.24  Very~ J     VVS2     62.8    57   336  3.94  3.96  2.48
```

Nonlinear Relationship

```
> ggplot(data = diamonds) +
+   geom_point(mapping=aes(x=carat, y=price, color=clarity), alpha=0.3)
```

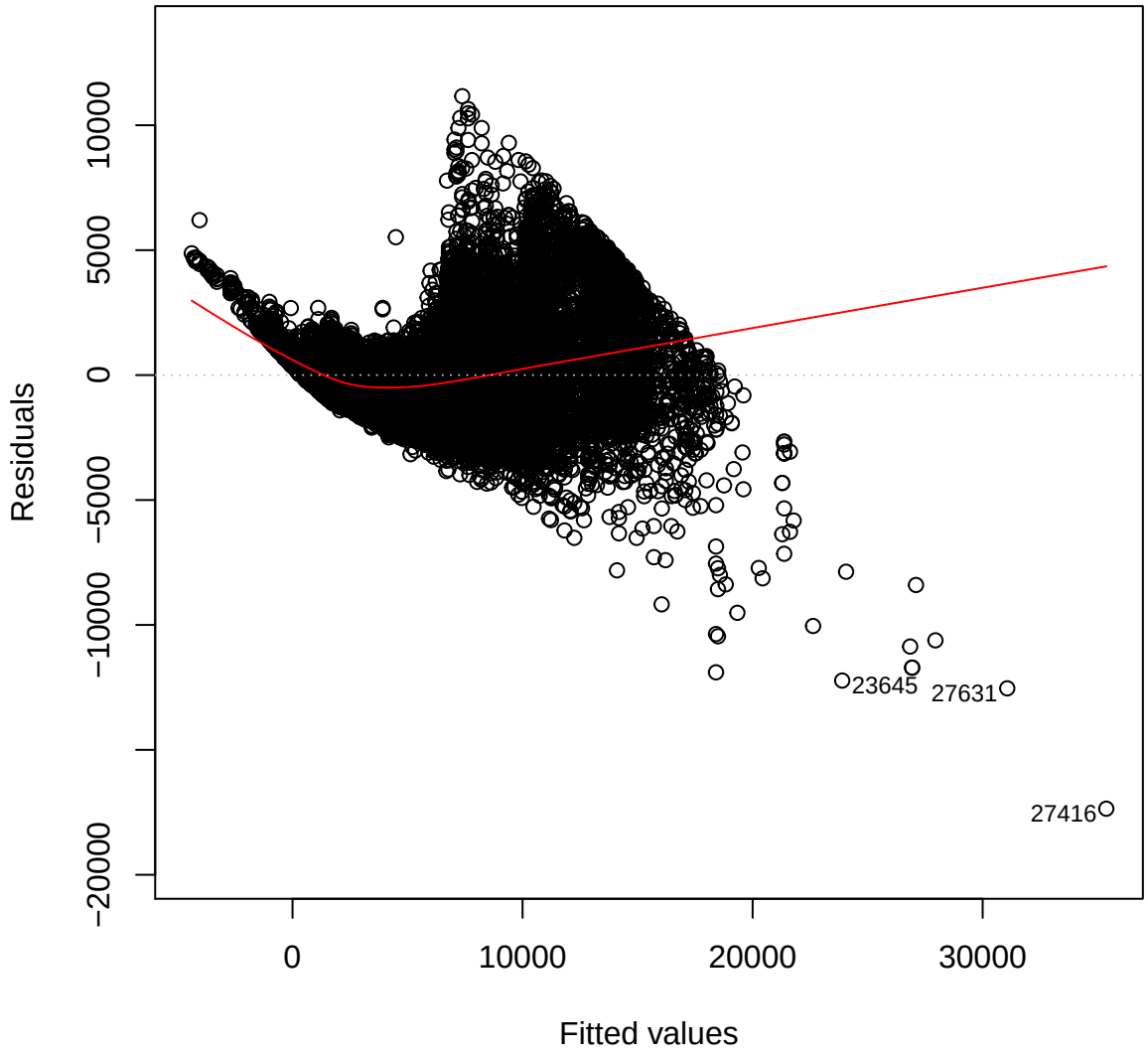


Regression with Nonlinear Relationship

```
> diam_fit <- lm(price ~ carat + clarity, data=diamonds)
> anova(diam_fit)
Analysis of Variance Table

Response: price
             Df     Sum Sq    Mean Sq  F value    Pr(>F)
carat         1 7.2913e+11 7.2913e+11 435639.9 < 2.2e-16 ***
clarity       7 3.9082e+10 5.5831e+09   3335.8 < 2.2e-16 ***
Residuals 53931 9.0264e+10 1.6737e+06
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

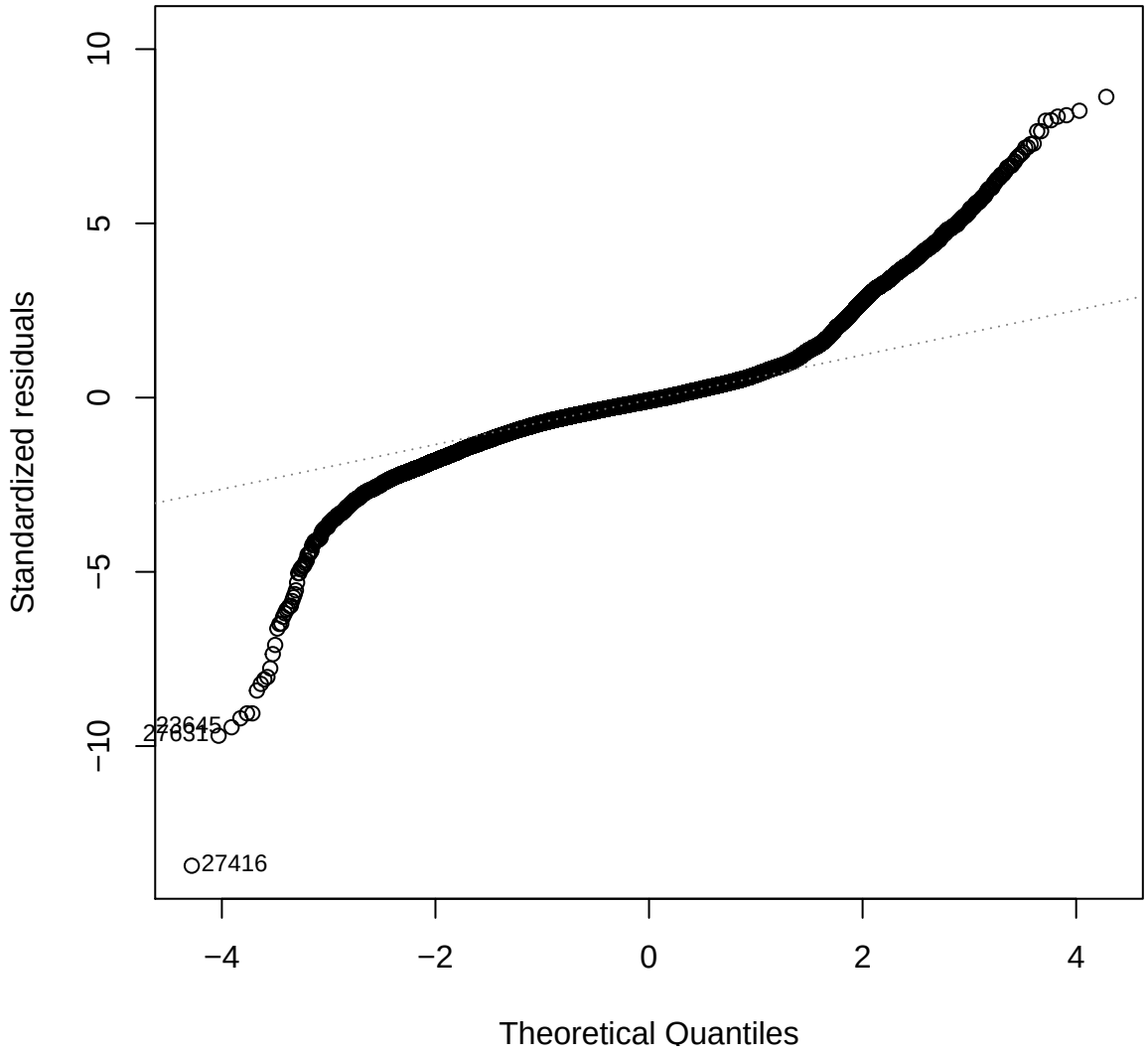
Residual Distribution

```
> plot(diam_fit, which=1)
```



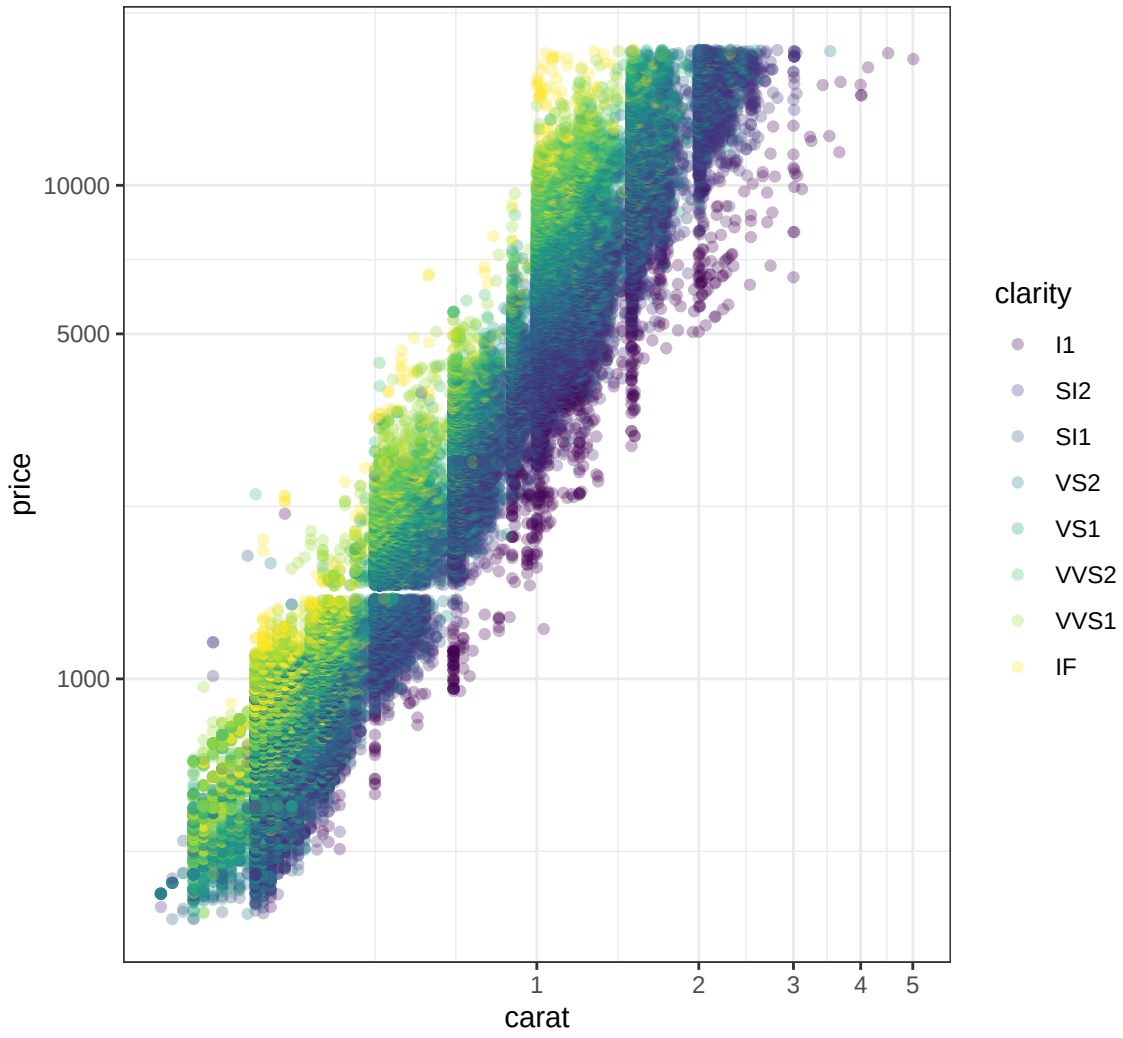
Normal Residuals Check

```
> plot(diam_fit, which=2)
```



Log-Transformation

```
> ggplot(data = diamonds) +
+   geom_point(aes(x=carat, y=price, color=clarity), alpha=0.3) +
+   scale_y_log10(breaks=c(1000,5000,10000)) +
+   scale_x_log10(breaks=1:5)
```

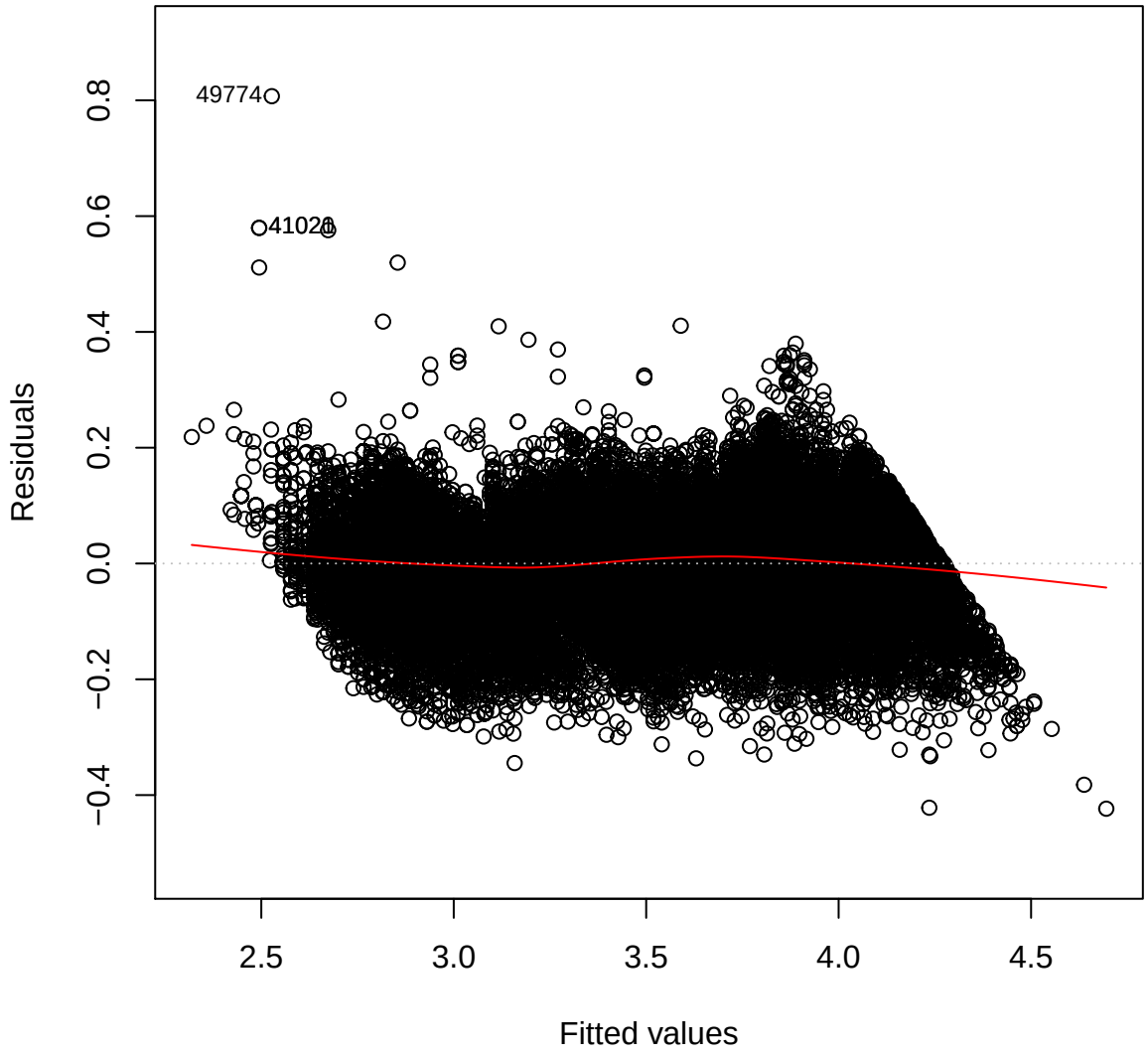


OLS on Log-Transformed Data

```
> diamonds <- mutate(diamonds, log_price = log(price, base=10),
+                    log_carat = log(carat, base=10))
> ldiam_fit <- lm(log_price ~ log_carat + clarity, data=diamonds)
> anova(ldiam_fit)
Analysis of Variance Table

Response: log_price
             Df Sum Sq Mean Sq   F value    Pr(>F)
log_carat     1 9771.9  9771.9 1452922.6 < 2.2e-16 ***
clarity       7  339.1    48.4    7203.3 < 2.2e-16 ***
Residuals 53931  362.7     0.0
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

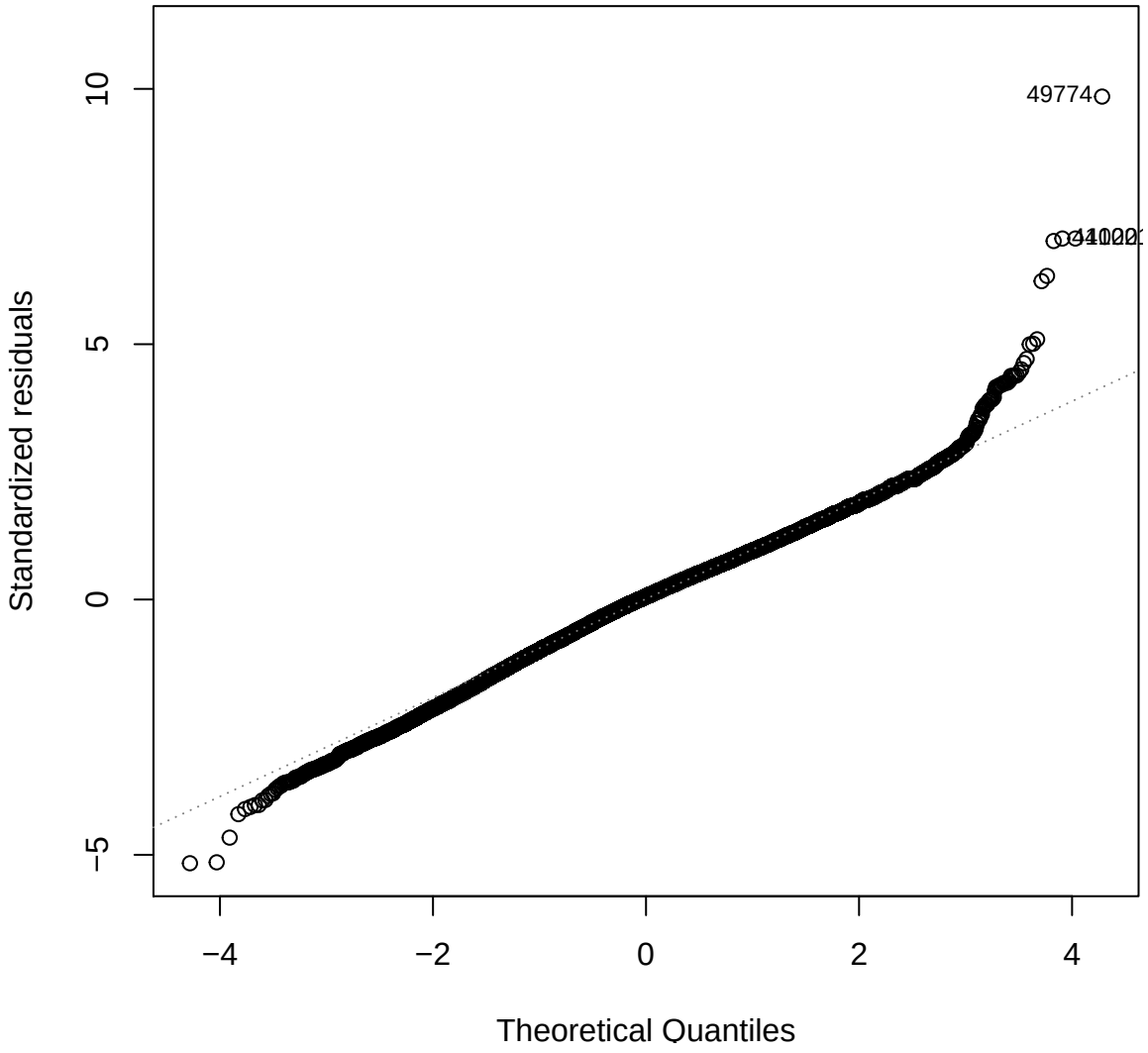
Residual Distribution

```
> plot(ldiam_fit, which=1)
```



Normal Residuals Check

```
> plot(ldiam_fit, which=2)
```



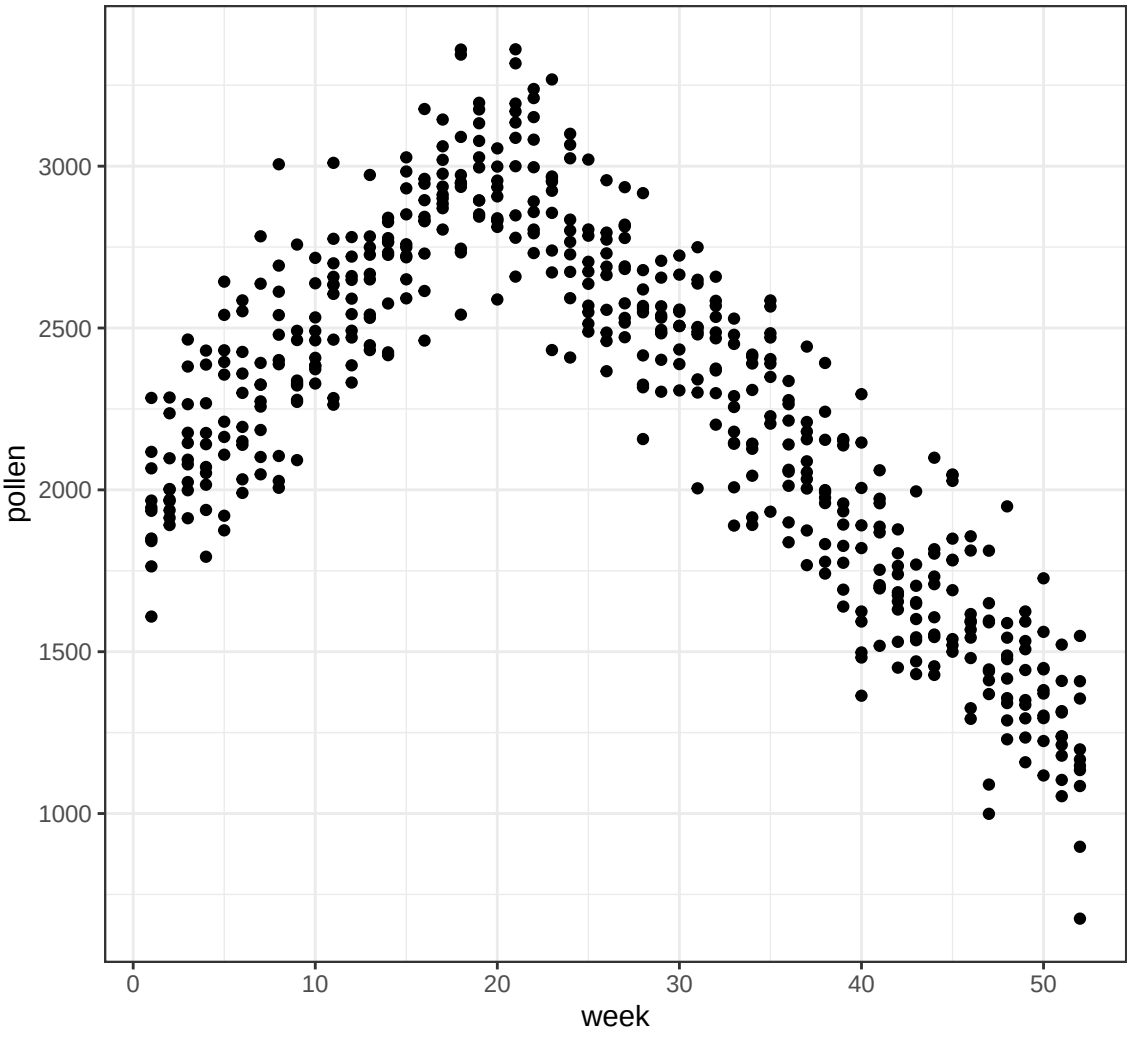
Tree Pollen Study

Suppose that we have a study where tree pollen measurements are averaged every week, and these data are recorded for 10 years. These data are simulated:

```
> pollen_study
# A tibble: 520 x 3
    week  year pollen
   <int> <int>  <dbl>
 1     1  2001  1842.
 2     2  2001  1966.
 3     3  2001  2381.
 4     4  2001  2141.
 5     5  2001  2210.
 6     6  2001  2585.
 7     7  2001  2392.
 8     8  2001  2105.
 9     9  2001  2278.
10    10  2001  2384.
# ... with 510 more rows
```

Tree Pollen Count by Week

```
> ggplot(pollen_study) + geom_point(aes(x=week, y=pollen))
```



A Clever Transformation

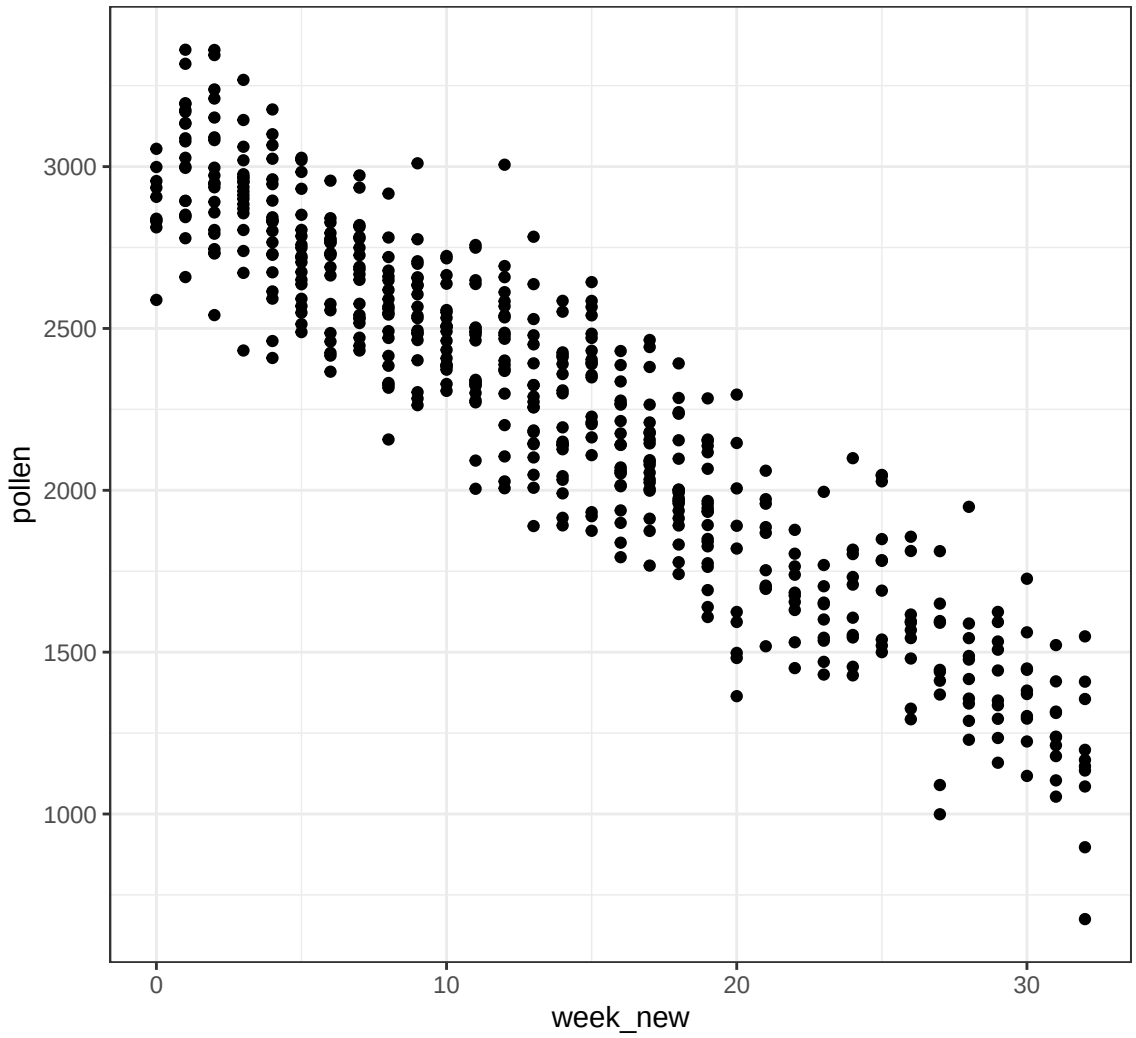
We can see there is a linear relationship between `pollen` and `week` if we transform `week` to be number of weeks from the peak week.

```
> pollen_study <- pollen_study %>%
+                         mutate(week_new = abs(week-20))
```

Note that this is a very different transformation from taking a log or power transformation.

`week` Transformed

```
> ggplot(pollen_study) + geom_point(aes(x=week_new, y=pollen))
```



OLS Goodness of Fit: Theory

Pythagorean Theorem

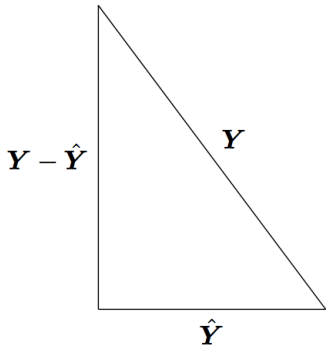


Figure 1: PythMod

Least squares model fitting can be understood through the Pythagorean theorem: $a^2 + b^2 = c^2$. However, here we have:

$$\sum_{i=1}^{n} Y_i^2 = \sum_{i=1}^{n} \hat{Y}_i^2 + \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

where the $\hat{Y}_i$ are the result of a **linear projection** of the Y_i.

OLS Normal Model

In this section, let's assume that $(\boldsymbol{X}_1, Y_1), \ldots, (\boldsymbol{X}_n, Y_n)$ are distributed so that

$$\begin{aligned} Y_i &= \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip} + E_i \\ &= \boldsymbol{X}_i \boldsymbol{\beta} + E_i \end{aligned}$$

where $\boldsymbol{E}|\boldsymbol{X} \sim \text{MVN}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. Note that we haven't specified the distribution of the $\boldsymbol{X}_i$ rv's.

Projection Matrices

In the OLS framework we have:

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}(\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{Y}.$$

The matrix $\boldsymbol{P}_{n \times n} = \boldsymbol{X}(\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T$ is a projection matrix. The vector $\boldsymbol{Y}$ is projected into the space spanned by the column space of $\boldsymbol{X}$.

Project matrices have the following properties:

- $\boldsymbol{P}$ is symmetric
- $\boldsymbol{P}$ is idempotent so that $\boldsymbol{PP} = \boldsymbol{P}$
- If $\boldsymbol{X}$ has column rank p, then $\boldsymbol{P}$ has rank p
- The eigenvalues of $\boldsymbol{P}$ are p 1's and $n - p$ 0's
- The trace (sum of diagonal entries) is $\text{tr}(\boldsymbol{P}) = p$
- $\boldsymbol{I} - \boldsymbol{P}$ is also a projection matrix with rank $n - p$

Decomposition

Note that $\boldsymbol{P}(\boldsymbol{I} - \boldsymbol{P}) = \boldsymbol{P} - \boldsymbol{PP} = \boldsymbol{P} - \boldsymbol{P} = \boldsymbol{0}$.

We have

$$\begin{aligned} \|\boldsymbol{Y}\|_2^2 = \boldsymbol{Y}^T \boldsymbol{Y} &= (\boldsymbol{PY} + (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y})^T (\boldsymbol{PY} + (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y}) \\ &= (\boldsymbol{PY})^T (\boldsymbol{PY}) + ((\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y})^T ((\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y}) \\ &= \|\boldsymbol{PY}\|_2^2 + \|(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y}\|_2^2 \end{aligned}$$

where the cross terms disappear because $\boldsymbol{P}(\boldsymbol{I} - \boldsymbol{P}) = \boldsymbol{0}$.

Note: The ℓ_p norm of an n-vector $\boldsymbol{w}$ is defined as

$$\|\boldsymbol{w}\|_p = \left(\sum_{i=1}^{n} |w_i|^p \right)^{1/p}.$$

Above we calculated

$$\|\boldsymbol{w}\|_2^2 = \sum_{i=1}^{n} w_i^2.$$

Distribution of Projection

Suppose that $Y_1, Y_2, \ldots, Y_n \overset{\text{iid}}{\sim} \text{Normal}(0, \sigma^2)$. This can also be written as $\boldsymbol{Y} \sim \text{MVN}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. It follows that

$$\boldsymbol{PY} \sim \text{MVN}_n(\boldsymbol{0}, \sigma^2 \boldsymbol{PIP}^T).$$

where $\boldsymbol{PIP}^T = \boldsymbol{PP}^T = \boldsymbol{PP} = \boldsymbol{P}$.

Also, $(\boldsymbol{PY})^T(\boldsymbol{PY}) = \boldsymbol{Y}^T\boldsymbol{P}^T\boldsymbol{PY} = \boldsymbol{Y}^T\boldsymbol{PY}$, a **quadratic form**. Given the eigenvalues of $\boldsymbol{P}$, $\boldsymbol{Y}^T\boldsymbol{PY}$ is equivalent in distribution to p squared iid Normal(0,1) rv's, so

$$\frac{\boldsymbol{Y}^T\boldsymbol{PY}}{\sigma^2} \sim \chi^2_p.$$

Distribution of Residuals

If $\boldsymbol{PY} = \hat{\boldsymbol{Y}}$ are the fitted OLS values, then $(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y} = \boldsymbol{Y} - \hat{\boldsymbol{Y}}$ are the residuals.

It follows by the same argument as above that

$$\frac{\boldsymbol{Y}^T(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y}}{\sigma^2} \sim \chi^2_{n-p}.$$

It's also straightforward to show that $(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y} \sim \text{MVN}_n(\boldsymbol{0}, \sigma^2(\boldsymbol{I} - \boldsymbol{P}))$ and $\text{Cov}(\boldsymbol{PY}, (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{Y}) = \boldsymbol{0}$.

Degrees of Freedom

The degrees of freedom, p, of a linear projection model fit is equal to

- The number of linearly dependent columns of $\boldsymbol{X}$
- The number of nonzero eigenvalues of $\boldsymbol{P}$ (where nonzero eigenvalues are equal to 1)
- The trace of the projection matrix, $\text{tr}(\boldsymbol{P})$.

The reason why we divide estimates of variance by $n - p$ is because this is the number of effective independent sources of variation remaining after the model is fit by projecting the n observations into a p dimensional linear space.

Submodels

Consider the OLS model $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{E}$ where there are p columns of $\boldsymbol{X}$ and $\boldsymbol{\beta}$ is a p-vector.

Let $\boldsymbol{X}_0$ be a subset of p_0 columns of $\boldsymbol{X}$ and let $\boldsymbol{X}_1$ be a subset of p_1 columns, where $1 \leq p_0 < p_1 \leq p$. Also, assume that the columns of $\boldsymbol{X}_0$ are a subset of $\boldsymbol{X}_1$.

We can form $\hat{\boldsymbol{Y}}_0 = \boldsymbol{P}_0\boldsymbol{Y}$ where $\boldsymbol{P}_0$ is the projection matrix built from $\boldsymbol{X}_0$. We can analogously form $\hat{\boldsymbol{Y}}_1 = \boldsymbol{P}_1\boldsymbol{Y}$.

Hypothesis Testing

Without loss of generality, suppose that $\boldsymbol{\beta}_0 = (\beta_1, \beta_2, \ldots, \beta_{p_0})^T$ and $\boldsymbol{\beta}_1 = (\beta_1, \beta_2, \ldots, \beta_{p_1})^T$.

How do we compare these models, specifically to test $H_0 : (\beta_{p_0+1}, \beta_{p_0+2}, \ldots, \beta_{p_1}) = \boldsymbol{0}$ vs $H_1 : (\beta_{p_0+1}, \beta_{p_0+2}, \ldots, \beta_{p_1}) \neq \boldsymbol{0}$?

The basic idea to perform this test is to compare the goodness of fits of each model via a pivotal statistic. We will discuss the generalized LRT and ANOVA approaches.

Generalized LRT

Under the OLS Normal model, it follows that $\hat{\boldsymbol{\beta}}_0 = (\boldsymbol{X}_0^T\boldsymbol{X}_0)^{-1}\boldsymbol{X}_0^T\boldsymbol{Y}$ is the MLE under the null hypothesis and $\hat{\boldsymbol{\beta}}_1 = (\boldsymbol{X}_1^T\boldsymbol{X}_1)^{-1}\boldsymbol{X}_1^T\boldsymbol{Y}$ is the unconstrained MLE. Also, the respective MLEs of σ^2 are

$$\hat{\sigma}_0^2 = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_{0,i})^2}{n}$$

$$\hat{\sigma}_1^2 = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_{1,i})^2}{n}$$

where $\hat{\boldsymbol{Y}}_0 = \boldsymbol{X}_0\hat{\boldsymbol{\beta}}_0$ and $\hat{\boldsymbol{Y}}_1 = \boldsymbol{X}_1\hat{\boldsymbol{\beta}}_1$.

The generalized LRT statistic is

$$\lambda(\boldsymbol{X}, \boldsymbol{Y}) = \frac{L\left(\hat{\boldsymbol{\beta}}_1, \hat{\sigma}_1^2; \boldsymbol{X}, \boldsymbol{Y}\right)}{L\left(\hat{\boldsymbol{\beta}}_0, \hat{\sigma}_0^2; \boldsymbol{X}, \boldsymbol{Y}\right)}$$

where $2\log\lambda(\boldsymbol{X}, \boldsymbol{Y})$ has a $\chi^2_{p_1-p_0}$ null distribution.

Nested Projections

We can apply the Pythagorean theorem we saw earlier to linear subspaces to get:

$$\begin{aligned}\|\boldsymbol{Y}\|_2^2 &= \|(\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{Y}\|_2^2 + \|\boldsymbol{P}_1\boldsymbol{Y}\|_2^2 \\ &= \|(\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{Y}\|_2^2 + \|(\boldsymbol{P}_1 - \boldsymbol{P}_0)\boldsymbol{Y}\|_2^2 + \|\boldsymbol{P}_0\boldsymbol{Y}\|_2^2\end{aligned}$$

We can also use the Pythagorean theorem to decompose the residuals from the smaller projection $\boldsymbol{P}_0$:

$$\|(\boldsymbol{I} - \boldsymbol{P}_0)\boldsymbol{Y}\|_2^2 = \|(\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{Y}\|_2^2 + \|(\boldsymbol{P}_1 - \boldsymbol{P}_0)\boldsymbol{Y}\|_2^2$$

$\boldsymbol{F}$ Statistic

The F statistic compares the improvement of goodness in fit of the larger model to that of the smaller model in terms of sums of squared residuals, and it scales this improvement by an estimate of σ^2:

$$\begin{aligned}F &= \frac{\left[\|(\boldsymbol{I} - \boldsymbol{P}_0)\boldsymbol{Y}\|_2^2 - \|(\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{Y}\|_2^2\right]/(p_1 - p_0)}{\|(\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{Y}\|_2^2/(n - p_1)} \\ &= \frac{\left[\sum_{i=1}^n (Y_i - \hat{Y}_{0,i})^2 - \sum_{i=1}^n (Y_i - \hat{Y}_{1,i})^2\right]/(p_1 - p_0)}{\sum_{i=1}^n (Y_i - \hat{Y}_{1,i})^2/(n - p_1)}\end{aligned}$$

Since $\|(\boldsymbol{I} - \boldsymbol{P}_0)\boldsymbol{Y}\|_2^2 - \|(\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{Y}\|_2^2 = \|(\boldsymbol{P}_1 - \boldsymbol{P}_0)\boldsymbol{Y}\|_2^2$, we can equivalently write the F statistic as:

$$\begin{aligned}F &= \frac{\|(\boldsymbol{P}_1 - \boldsymbol{P}_0)\boldsymbol{Y}\|_2^2/(p_1 - p_0)}{\|(\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{Y}\|_2^2/(n - p_1)} \\ &= \frac{\sum_{i=1}^n (\hat{Y}_{1,i} - \hat{Y}_{0,i})^2/(p_1 - p_0)}{\sum_{i=1}^n (Y_i - \hat{Y}_{1,i})^2/(n - p_1)}\end{aligned}$$

F Distribution

Suppose we have independent random variables $V \sim \chi^2_a$ and $W \sim \chi^2_b$. It follows that

$$\frac{V/a}{W/b} \sim F_{a,b}$$

where $F_{a,b}$ is the F distribution with (a, b) degrees of freedom.

By arguments similar to those given above, we have

$$\frac{\|(\boldsymbol{P}_1 - \boldsymbol{P}_0)\boldsymbol{Y}\|_2^2}{\sigma^2} \sim \chi^2_{p_1-p_0}$$

$$\frac{\|(\boldsymbol{I} - \boldsymbol{P}_1)\boldsymbol{Y}\|_2^2}{\sigma^2} \sim \chi^2_{n-p_1}$$

and these two rv's are independent.

F Test

Suppose that the OLS model holds where $\boldsymbol{E}|\boldsymbol{X} \sim \text{MVN}_n(\boldsymbol{0}, \sigma^2\boldsymbol{I})$.

In order to test $H_0 : (\beta_{p_0+1}, \beta_{p_0+2}, \ldots, \beta_{p_1}) = \boldsymbol{0}$ vs $H_1 : (\beta_{p_0+1}, \beta_{p_0+2}, \ldots, \beta_{p_1}) \neq \boldsymbol{0}$, we can form the F statistic as given above, which has null distribution $F_{p_1-p_0,n-p_1}$. The p-value is calculated as $\Pr(F^* \geq F)$ where F is the observed F statistic and $F^* \sim F_{p_1-p_0,n-p_1}$.

If the above assumption on the distribution of $\boldsymbol{E}|\boldsymbol{X}$ only approximately holds, then the F test p-value is also an approximation.

OLS Goodness of Fit: R

Example: Davis Data

```
> data("Davis", package="carData")
> htwt <- tbl_df(Davis)
> htwt[12,c(2,3)] <- htwt[12,c(3,2)]
> head(htwt)
# A tibble: 6 x 5
  sex   weight height repwt repht
  <fct>  <int>  <int> <int> <int>
1 M         77    182    77   180
2 F         58    161    51   159
3 F         53    161    54   158
4 M         68    177    70   175
5 F         59    157    59   155
6 M         76    170    76   165
```

Comparing Linear Models in R

Example: Davis Data

Suppose we are considering the three following models:

```
> f1 <- lm(weight ~ height, data=htwt)
> f2 <- lm(weight ~ height + sex, data=htwt)
> f3 <- lm(weight ~ height + sex + height:sex, data=htwt)
```

How do we determine if the additional terms in models `f2` and `f3` are needed?

ANOVA (Version 2)

A generalization of ANOVA exists that allows us to compare two nested models, quantifying their differences in terms of goodness of fit and performing a hypothesis test of whether this difference is statistically significant.

A model is *nested* within another model if their difference is simply the absence of certain terms in the smaller model.

The null hypothesis is that the additional terms have coefficients equal to zero, and the alternative hypothesis is that at least one coefficient is nonzero.

Both versions of ANOVA can be described in a single, elegant mathematical framework.

Comparing Two Models with `anova()`

This provides a comparison of the improvement in fit from model `f2` compared to model `f1`:

```
> anova(f1, f2)
Analysis of Variance Table

Model 1: weight ~ height
Model 2: weight ~ height + sex
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    198 14321
2    197 12816  1    1504.9 23.133 2.999e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

When There's a Single Variable Difference

Compare above `anova(f1, f2)` p-value to that for the `sex` term from the `f2` model:

```
> library(broom)
> tidy(f2)
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  -76.6     15.7        -4.88 2.23e- 6
2 height         0.811    0.0953      8.51 4.50e-15
3 sexM           8.23     1.71        4.81 3.00e- 6
```

Calculating the F-statistic

```
> anova(f1, f2)
Analysis of Variance Table

Model 1: weight ~ height
Model 2: weight ~ height + sex
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    198 14321
```

```
2    197 12816  1    1504.9 23.133 2.999e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How the F-statistic is calculated:

```
> n <- nrow(htwt)
> ss1 <- (n-1)*var(f1$residuals)
> ss1
[1] 14321.11
> ss2 <- (n-1)*var(f2$residuals)
> ss2
[1] 12816.18
> ((ss1 - ss2)/anova(f1, f2)$Df[2])/(ss2/f2$df.residual)
[1] 23.13253
```

Calculating the Generalized LRT

```
> anova(f1, f2, test="LRT")
Analysis of Variance Table

Model 1: weight ~ height
Model 2: weight ~ height + sex
  Res.Df   RSS Df Sum of Sq  Pr(>Chi)
1    198 14321
2    197 12816  1    1504.9 1.512e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> library(lmtest)
> lrtest(f1, f2)
Likelihood ratio test

Model 1: weight ~ height
Model 2: weight ~ height + sex
  #Df LogLik Df  Chisq Pr(>Chisq)
1   3 -710.9
2   4 -699.8  1 22.205   2.45e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These tests produce slightly different answers because `anova()` adjusts for degrees of freedom when estimating the variance, whereas `lrtest()` is the strict generalized LRT. See here.

ANOVA on More Distant Models

We can compare models with multiple differences in terms:

```
> anova(f1, f3)
Analysis of Variance Table

Model 1: weight ~ height
Model 2: weight ~ height + sex + height:sex
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    198 14321
```

```
2    196 12567  2      1754 13.678 2.751e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Compare Multiple Models at Once

We can compare multiple models at once:

```
> anova(f1, f2, f3)
Analysis of Variance Table

Model 1: weight ~ height
Model 2: weight ~ height + sex
Model 3: weight ~ height + sex + height:sex
  Res.Df   RSS Df Sum of Sq       F    Pr(>F)
1    198 14321
2    197 12816  1   1504.93 23.4712 2.571e-06 ***
3    196 12567  1    249.04  3.8841   0.05015 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Extras

Source

License

Source Code

Session Information

```
> sessionInfo()
R version 3.6.0 (2019-04-26)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS  10.15.3

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
 [1] lmtest_0.9-37   zoo_1.8-7       broom_0.5.2
 [4] carData_3.0-3   forcats_0.5.0   stringr_1.4.0
 [7] dplyr_0.8.4     purrr_0.3.3     readr_1.3.1
[10] tidyr_1.0.2     tibble_2.1.3    ggplot2_3.2.1
[13] tidyverse_1.3.0 knitr_1.28
```

```
loaded via a namespace (and not attached):
 [1] tidyselect_1.0.0 xfun_0.12        haven_2.2.0
 [4] lattice_0.20-40  colorspace_1.4-1 vctrs_0.2.3
 [7] generics_0.0.2   htmltools_0.4.0  yaml_2.2.1
[10] utf8_1.1.4       rlang_0.4.5      pillar_1.4.3
[13] withr_2.1.2      glue_1.3.1       DBI_1.1.0
[16] dbplyr_1.4.2     modelr_0.1.6     readxl_1.3.1
[19] lifecycle_0.1.0  munsell_0.5.0    gtable_0.3.0
[22] cellranger_1.1.0 rvest_0.3.5      evaluate_0.14
[25] labeling_0.3     fansi_0.4.1      Rcpp_1.0.3
[28] scales_1.1.0     backports_1.1.5  jsonlite_1.6.1
[31] farver_2.0.3     fs_1.3.1         hms_0.5.3
[34] digest_0.6.25    stringi_1.4.6    grid_3.6.0
[37] cli_2.0.2        tools_3.6.0      magrittr_1.5
[40] lazyeval_0.2.2   crayon_1.3.4     pkgconfig_2.0.3
[43] xml2_1.2.2       reprex_0.3.0     lubridate_1.7.4
[46] assertthat_0.2.1 rmarkdown_2.1    httr_1.4.1
[49] rstudioapi_0.11  R6_2.4.1         nlme_3.1-144
[52] compiler_3.6.0
```