Statistical Computing and Data Visualization in R

Lecture 9

Linear Models

Linear models: an example

- The data frame `whiteside` contains weekly gas consumption and average external temperature at a house in south-east England during two ‘heating seasons’, one before and one after cavity-wall insulation was installed.
- The goal of the exercise was to assess the effect of the insulation on gas consumption.

Linear models: an example

- The dataframe is in the `MASS` package

```
> library(MASS)
> head(whiteside)
   Insul Temp Gas
1 Before -0.8 7.2
2 Before -0.7 6.9
3 Before  0.4 6.4
4 Before  2.5 6.0
5 Before  2.9 5.8
6 Before  3.2 5.8
```

- The response (y) is **Gas**, and you have two predictors (x), one that is continuous (**Temp**) and one that is categorical (**Insul**)

Linear models: exploratory data analysis

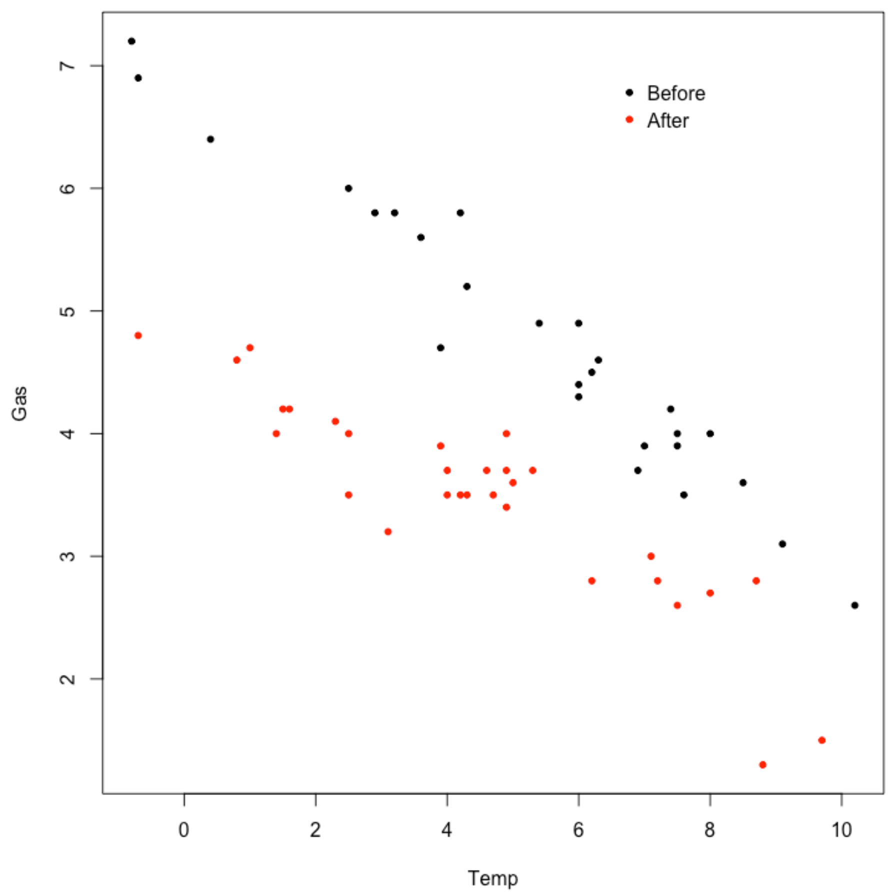
- Note that the variable Gas is continuous (numeric), but the variable Insul is categorical (factor):

```
> is.numeric(Gas)
[1] TRUE
> is.factor(Insul)
[1] TRUE
```

- Let’s plot the data:

```
> attach(whiteside)
> par(mar=c(4,4,1,1)+0.1)
> plot(Temp, Gas, pch=20,
col=as.numeric(Insul))
> legend(6.5, 7, c(”Before”, ”After”),
col=c(”black”,”red”), pch=20, bty=”n”)
```

Linear models: exploratory data analysis



- There is evidence of a linear relationship between Temperature and Gas consumption.
- However, the form of the relationship (both intercept and slope) depends on the type of insulation: after seems much better at low temperatures, but the differences tend to be minor at moderate ones.

Formulating a linear model

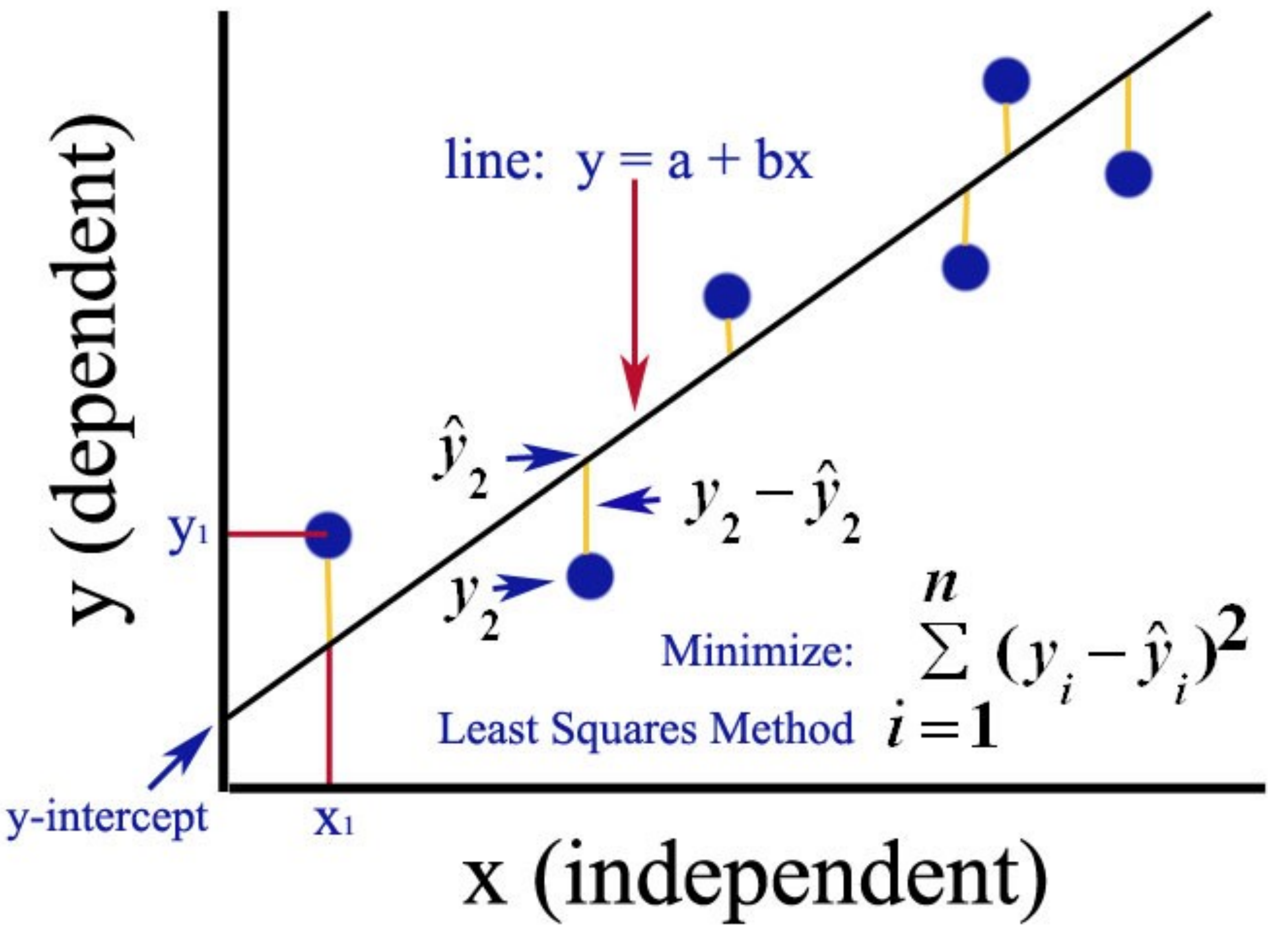
- The scatterplot suggests that fitting two separate simple linear regressions to each of the two groups of observations.

$y_j = \alpha_0 + \beta_0 x_j + \varepsilon_j$ $\quad\quad$ $\varepsilon_j \sim N(0, \sigma_0^2)$ $\quad\quad$ If insulation = “Before”

$y_i = \alpha_1 + \beta_1 x_j + \omega_i$ $\quad\quad$ $\omega_j \sim N(0, \sigma_1^2)$ $\quad\quad$ If insulation = “After”

- Recall that MLEs $(\hat{\alpha}_k, \hat{\beta}_k)$ for (α_k, β_k) under this model correspond to the least square estimators, so that $\hat{y}_j = \hat{\alpha}_k + \hat{\beta}_k x_j$.

Ordinary least squares



A joint model

- In the context of this formulation, the questions of interest translate into: Is $\alpha_0 = \alpha_1$ or $\beta_0 = \beta_1$?
- However, to be able to test these hypotheses we need to setup a joint model for both groups.

A joint model

- Let $y_{i,j}$ represent the *j*-th observation in group *i*, with $i = 0$ for "Before" and $i = 1$ for "After". Define $x_{i,j}$ similarly.
- In this case we have 26 observations in group 0 (Before) and 30 in group 1 (After):

```
> table(Insul)
Insul
Before  After
    26     30
```

- Hence, *i* runs from 0 to 1 and *j* runs from 1 to 26 when $i = 0$ and from 1 to 30 when $i = 1$.

A joint model

- Under this notation, the joint model can be written

$$y_{i,j} = \alpha_i + \beta_i x_{i,j} + \varepsilon_{i,j} \qquad \varepsilon_{i,j} \sim N(0, \sigma^2)$$

- This model also allows for two distinct intercepts and slopes. However, we have now a single variance for the errors
- The model can also be written as

$$y_{i,j} = \mu + \alpha_i^* + (\eta + \beta_i^*) x_{i,j} + \varepsilon_{i,j} \qquad \varepsilon_{i,j} \sim N(0, \sigma^2)$$

wehre $\alpha_i = \mu + \alpha_i^*$, $\beta_i = \mu + \beta_i^*$, $\alpha_0 = 0$ and $\beta_0 = 0$. (α_1^* and β_1^* interpreted as differences with respect to the baseline category values given by μ and η).

Fitting linear models in R

- The function `lm` allows you fit linear models in R. The structure of the function is:

```
lm(formula, data, weights, subset, na.action)
```

- Some of the main arguments are

`formula`	is the model formula (the only required argument).
`data`	in an optional data frame.
`weights`	is a vector of positive weights, if non-uniform weights are needed
`subset`	is an index vector specifying a subset of the data to be used (by default all items are used)
`na.action`	is a function specifying how missing values are to be handled

Fitting simple linear models in R

- For example, to fit the individual models (simple linear regression).

```
> mod.before <- lm(Gas~Temp, data=whiteside,
subset=(Insul=="Before"))
> mod.before
Call:lm(formula = Gas ~ Temp, data = whiteside, subset = (Insul
==      "Before"))

Coefficients:
(Intercept)         Temp
     6.8538      -0.3932
> mod.after <- lm(Gas~Temp, data=whiteside,
subset=(Insul=="After"))
> mod.after
Call:lm(formula = Gas ~ Temp, data = whiteside, subset = (Insul
==      "After"))

Coefficients:
(Intercept)         Temp
     4.7238      -0.2779
```

Fitting simple linear models in R

- Simply printing the object will only give you back the call you used and the coefficients. More detail can be obtained using the function `summary()`

```
> summary(mod.before)
Call:lm(formula = Gas ~ Temp, data = whiteside, subset = (Insul
==      "Before"))

Residuals:     Min       1Q   Median       3Q      Max
          -0.62020 -0.19947  0.06068  0.16770  0.59778

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.85383    0.11842   57.88   <2e-16 ***
Temp        -0.39324    0.01959  -20.08   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2813 on 24 degrees of freedom
Multiple R-squared:  0.9438, Adjusted R-squared:  0.9415
F-statistic: 403.1 on 1 and 24 DF,  p-value: < 2.2e-16
```

H0: $\alpha = 0$ vs Ha: $\alpha \neq 0$

H0: $\beta = 0$ vs Ha: $\beta \neq 0$

`lm` objects

- An `lm` object is a list that contains a number of pieces. Some of the key ones

`coefficents`	a named vector of coefficients
`residuals`	residuals (response minus fitted values)
`fitted.values`	fitted mean values
`df.residual`	degrees of freedom of the residuals

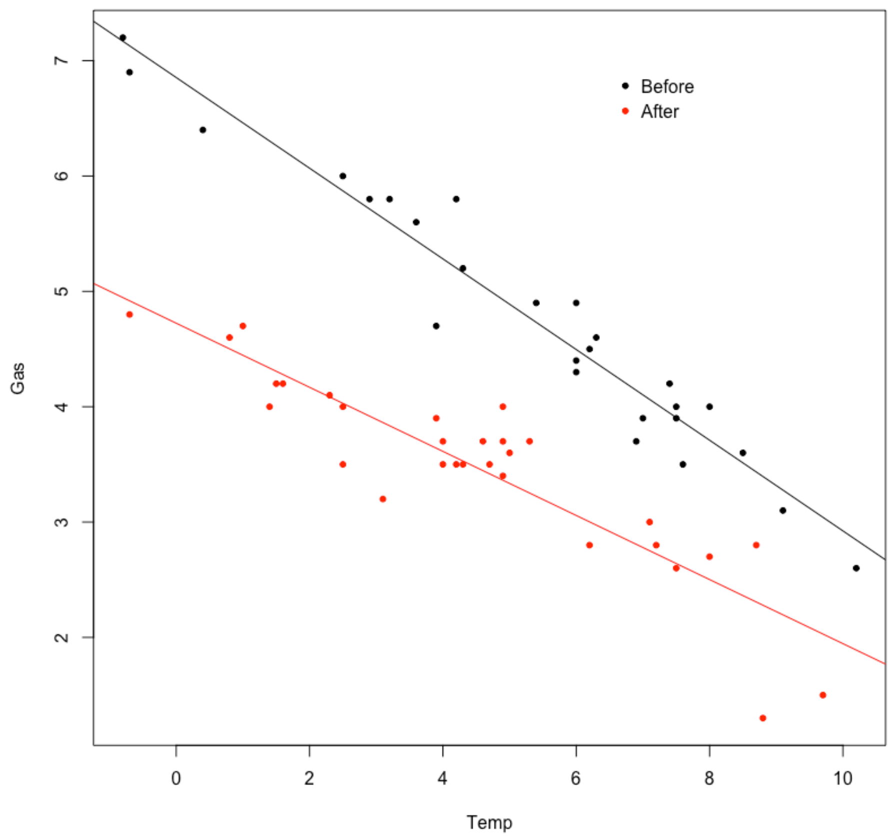
- For example, the estimate of the variance can be computed as

```
> sqrt(sum(mod.before$residuals^2) /
df.residual)
[1] 0.2813337
```

Plotting regression lines

- You can also use the elements of the object to add lines to the graph:

```
> par(mar=c(4,4,1,1)+0.1)
> plot(Temp, Gas, pch=20,
col=as.numeric(Insul))
> legend(6.5, 7, c("Before",
"After"),
col=c("black","red"), pch=20,
bty="n")
>
abline(mod.before$coefficient
s, col="black")
>
abline(mod.after$coefficients
, col="red")
```



Fitting general linear models in R

- Let’s do the joint model now

```
> mod.joint <- lm(Gas~Temp*Insul, data=whiteside)
> summary(mod.joint)

Call:lm(formula = Gas ~ Temp*Insul, data = whiteside)

Residuals:
     Min       1Q   Median       3Q      Max
-0.97802 -0.18011  0.03757  0.20930  0.63803

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
μ    (Intercept)      6.85383    0.13596  50.409  < 2e-16 ***
α1*  Temp            -0.39324    0.02249 -17.487  < 2e-16 ***
η    InsulAfter      -2.12998    0.18009 -11.827 2.32e-16 ***
β1*  Temp:InsulAfter  0.11530    0.03211   3.591 0.000731 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.323 on 52 degrees of freedom
Multiple R-squared:  0.9277, Adjusted R-squared:  0.9235
F-statistic: 222.3 on 3 and 52 DF,  p-value: < 2.2e-16
```

Hypotheses testing for linear models

- Now we can answer the question we started with: what is the effect of insulation on gas consumption?
- This is really two questions:
  - Is the impact of insulation on gas consumption the same no matter what the external temperature is? (i.e., is the slope the same?, i.e., is $\beta_1^* \neq 0$?)
  - If the impact of insulation is the same at all temperatures, is that constant impact different non-negligible? (i.e., is the intercept the same?, i.e., is $\alpha_1^* \neq 0$?)
- Note that we ask these questions sequentially: if the answer to the first is yes the answer to the second is automatically yes too!

Hypotheses testing for linear models

- To answer the first question we can look at the p-value associated with β_1^*

```
> summary(mod.joint)

Call:lm(formula = Gas ~ Temp*Insul, data = whiteside)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.97802 -0.18011  0.03757  0.20930  0.63803 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)      6.85383    0.13596  50.409  < 2e-16 ***
Temp            -0.39324    0.02249 -17.487  < 2e-16 ***
InsulAfter      -2.12998    0.18009 -11.827 2.32e-16 ***
Temp:InsulAfter  0.11530    0.03211   3.591 0.000731 ***
---
Signif. codes:  
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H0: $\beta_1^* = 0$ vs. Ha: $\beta_1^* \neq 0$

Since the p-value is less than 0.05, we reject H0 ➔ The impact of insulation on gas consumption varies with the temperature. No further test is needed.

Hypotheses testing for linear models

- An alternative way to contrast these hypotheses is to run an F test between the model with different slopes and a model with a common slope:

```
> mod.joint2 <- lm(Gas~Temp+Insul, data=whiteside)
> aov(mod.joint2, mod.joint)
Analysis of Variance Table
Model 1: Gas ~ Temp + Insul
Model 2: Gas ~ Temp * Insul
          Res.Df    RSS Df Sum of Sq       F    Pr(>F)
1       53 6.7704
2       52 5.4252  1    1.3451 12.893 0.0007307 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Note in that this is the same p-value as before (in this case the tests are equivalent).
- More generally, F tests allow you to test multiple coefficients simultaneously (unlike t tests).

The general linear model

- The function `lm` allows you to fit any general linear model

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon} \qquad \boldsymbol{\varepsilon} \sim N_n(0, \sigma^2 diag\{w_1, \cdots, w_n\})$$

  Where $\boldsymbol{y}$ is a vector of observations, $\boldsymbol{X}$ is the known design matrix, $\boldsymbol{\beta}$ is a vector of unknown coefficients, σ^2 is the variance and $w_1, \cdots, w_n$ are known weights.

- Examples of the general linear model include:
  - Simple and multiple regression.
  - Analysis of Variance (ANOVA) models.
  - Analysis of Covariance (ANCOVA) models.

Using formulas with `lm`

Symbol	Example	Meaning
`+`	`+X`	Include this variable
`-`	`-X`	Delete this variable
`:`	`X:Z`	Include the interaction between the variables
`*`	`X*Y`	Include these variables as well as their interactions
`|`	`X|Z`	Conditioning/nesting: include x given z
`^`	`(X+Z+W)^3`	Include all these variables and all interactions up to 3-way
`I`	`I(X*Z)`	Isolate/as is: include a new variable obtained by performing the isolated operation
`1`	`X-1`	Intercept: do not include an intercept (default is to include)

Note that all of the following are equivalent:

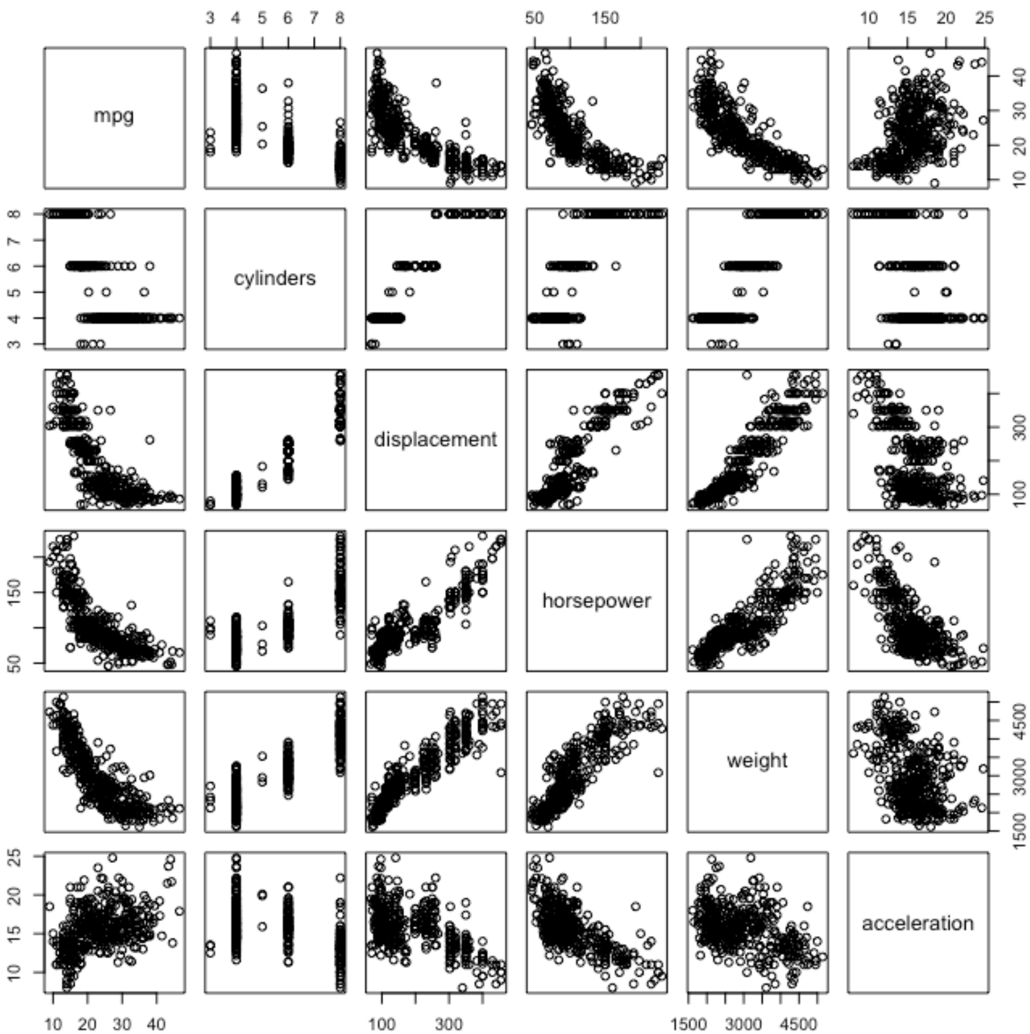
```
Y ~ X + Z + W + X:Z + Z:W + Z:W
Y ~ X*Z*W - X:Z:W
Y ~ (X + Z + W)^2
```

A multiple regression example: Car fuel consumption

- Consider now a second example where we are trying to understand how different factors affect city-cycle fuel consumption in miles per gallon for various car models.
- The dataset was used as the testbed for graphical analysis packages at the 1983 American Statistical Association Exposition
- We have one continuous response variable as well as 5 continuous and 3 categorical predictors. We are going to ignore the 3 categorical ones (model year, origin and car name).

Descriptive analysis

```
> dat = read.table(file="auto-mpg.txt", header=T)
> pairs(~ mpg + cylinders + displacement+ horsepower + weight + acceleration, data=dat)
```



Car fuel consumption

- In this case we are going to fit a model without any interactions:

```
> dat = read.table(file="auto-mpg.txt", header=T)
> names(dat)
> mod = lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration,
data=dat)
> summary(mod)
```

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \beta_3 x_{3,i} + \beta_4 x_{4,i} + \beta_5 x_{5,i} + \varepsilon_i$$

```
Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    acceleration, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5816  -2.8618  -0.3404   2.2438  16.3416

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
β0 (Intercept)  4.626e+01  2.669e+00  17.331   <2e-16 ***
β1 cylinders   -3.979e-01  4.105e-01  -0.969   0.3330
β2 displacement -8.313e-05 9.072e-03  -0.009   0.9927
β3 horsepower  -4.526e-02  1.666e-02  -2.716   0.0069 **
β4 weight      -5.187e-03  8.167e-04  -6.351    6e-10 ***
β5 acceleration -2.910e-02 1.258e-01  -0.231   0.8171
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.247 on 386 degrees of freedom
  (6 observations deleted due to missingness)
Multiple R-squared:  0.7077, Adjusted R-squared:  0.7039
F-statistic: 186.9 on 5 and 386 DF,  p-value: < 2.2e-16
```

Backward selection

- We can start eliminating variables and refitting the model. We first drop `displacement`:

```
> mod.a = lm(mpg ~ cylinders + horsepower + weight + acceleration,
data=dat)
> summary(mod.a)

Call:lm(formula = mpg ~ cylinders + horsepower + weight + acceleration,
data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5807  -2.8628  -0.3409   2.2427  16.3422

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  46.2739915  2.4481591  18.902  < 2e-16 ***
cylinders    -0.4004602  0.3032615  -1.321  0.18744
horsepower   -0.0452970  0.0160604  -2.820  0.00504 **
weight       -0.0051902  0.0007341  -7.070 7.26e-12 ***
acceleration -0.0289828  0.1248944  -0.232  0.81661
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Backward selection

- Next we drop `acceleration`:

```
> mod.a = lm(mpg ~ cylinders + horsepower + weight, data=dat)
> summary(mod.a)

Call:lm(formula = mpg ~ cylinders + horsepower + weight, data =
dat)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5260  -2.7955  -0.3559   2.2567  16.3209

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 45.7368172  0.7959566  57.461  < 2e-16 ***
cylinders   -0.3889745  0.2988302  -1.302  0.193806
horsepower  -0.0427277  0.0116196  -3.677  0.000269 ***
weight      -0.0052723  0.0006424  -8.208  3.37e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Backward selection

- Finally, we drop `cylinders`:

```
> mod.a = lm(mpg ~ cylinders + horsepower + weight, data=dat)
> summary(mod.a)

Call:lm(formula = mpg ~ horsepower + weight, data = dat)

Residuals:
     Min        1Q    Median        3Q       Max 
-11.0762  -2.7340  -0.3312   2.1752  16.2601 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 45.6402108  0.7931958  57.540  < 2e-16 ***
horsepower  -0.0473029  0.0110851  -4.267 2.49e-05 ***
weight      -0.0057942  0.0005023 -11.535  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The variables that seem to explain mpg are horsepoewer and weight.

Continuous vs Categorical

- Note that we are treating cylinders a continuous variable, so `lm` fits a linear regression:

```
> mod1 = lm(mpg ~ cylinders, data=dat)
> summary(mod1)

Call:
lm(formula = mpg ~ cylinders, data = dat)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.2607  -3.3841  -0.6478   2.5538  17.9022 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  42.9493     0.8330   51.56   <2e-16 ***
cylinders    -3.5629     0.1458  -24.43   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.942 on 396 degrees of freedom
Multiple R-squared:  0.6012, Adjusted R-squared:  0.6002 
F-statistic: 597.1 on 1 and 396 DF,  p-value: < 2.2e-16
```

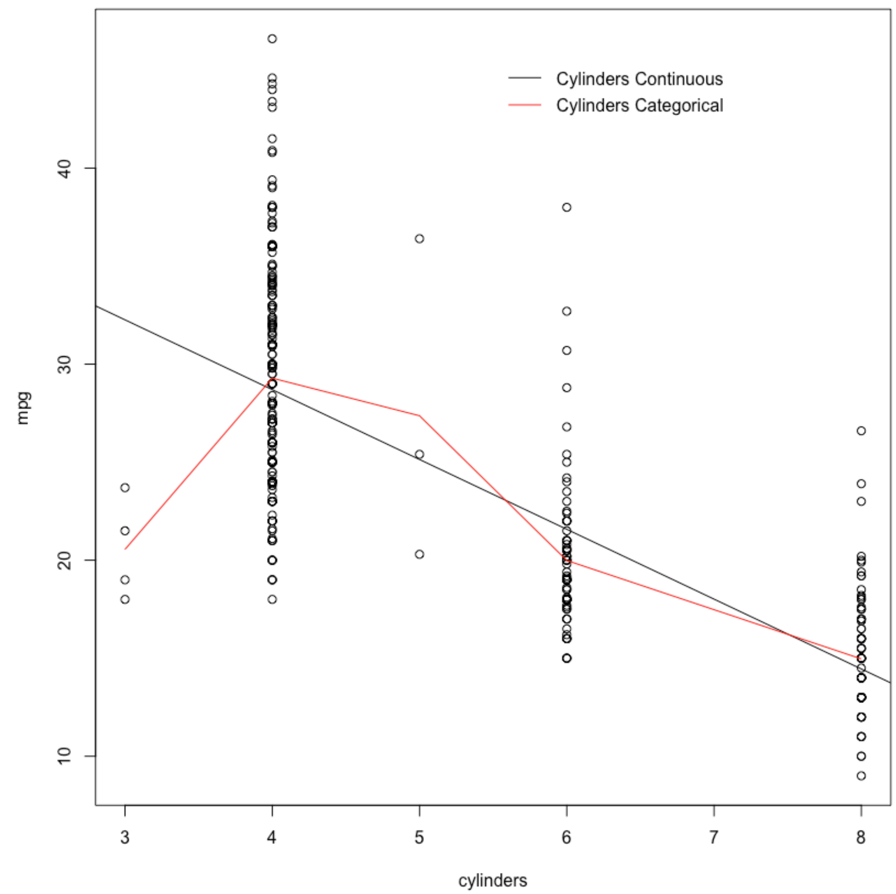
Continuous vs Categorical

- If we turn the number of cylinder into a categorical variable, `lm` fits an ANOVA model instead:

```
> mod2 = lm(mpg ~ factor(cylinders), data=dat)
> summary(mod2)

Call:
lm(formula = mpg ~ factor(cylinders), data = dat)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.2868  -2.9631  -0.9631   2.3890  18.0143 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)         20.5500     2.3657   8.687  < 2e-16 ***
factor(cylinders)4   8.7368     2.3888   3.657 0.000289 ***
factor(cylinders)5   6.8167     3.6137   1.886 0.059985 .  
factor(cylinders)6  -0.5643     2.4214  -0.233 0.815849    
factor(cylinders)8  -5.5869     2.4112  -2.317 0.021014 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.731 on 393 degrees of freedom
Multiple R-squared:  0.6372, Adjusted R-squared:  0.6335 
F-statistic: 172.6 on 4 and 393 DF,  p-value: < 2.2e-16
```

Continuous vs Categorical

- What is more appropriate in the previous example, to treat the number of cylinders as a continuous or a categorical variable?



Continuous vs Categorical

- When you have different classes of predictors, `lm` fits an ANCOVA model (as in our first example):

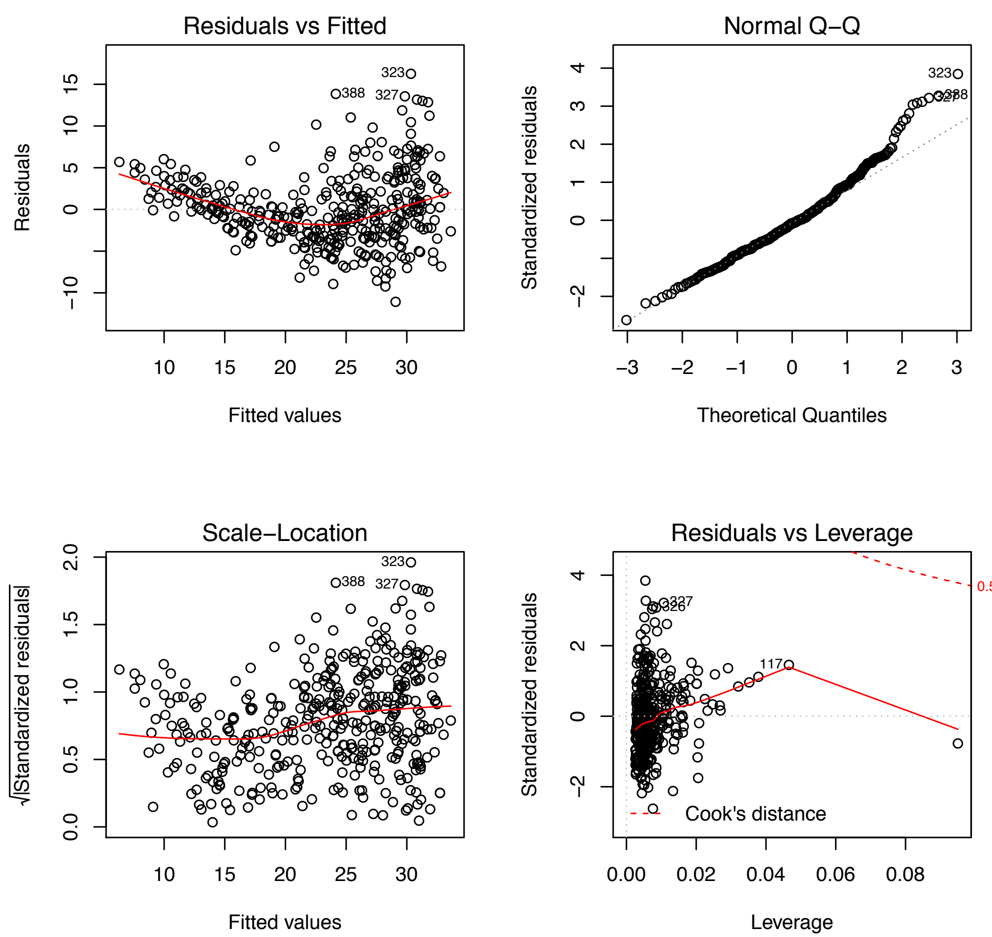
```
> mod3 = lm(mpg ~ weight*factor(cylinders), data=dat)
> summary(mod3)

Call:
lm(formula = mpg ~ weight * factor(cylinders), data = dat)

Residuals:
     Min       1Q   Median       3Q      Max
-10.2700  -2.4097  -0.4621   1.8307  17.0414

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                5.166320  22.673910   0.228    0.820
weight                     0.006414   0.009416   0.681    0.496
factor(cylinders)4        44.742048  22.753837   1.966    0.050 *
factor(cylinders)5        25.440982  32.858673   0.774    0.439
factor(cylinders)6        31.801776  23.075818   1.378    0.169
factor(cylinders)8        24.277498  22.971639   1.057    0.291
weight:factor(cylinders)4 -0.015348   0.009451  -1.624    0.105
weight:factor(cylinders)5 -0.007458   0.012117  -0.616    0.539
weight:factor(cylinders)6 -0.011724   0.009510  -1.233    0.218
weight:factor(cylinders)8 -0.009933   0.009458  -1.050    0.294
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.037 on 388 degrees of freedom
Multiple R-squared:  0.7392, Adjusted R-squared:  0.7332
F-statistic: 122.2 on 9 and 388 DF,  p-value: < 2.2e-16
```

Goodness of fit in linear models

- You can plot a model object to facilitate the investigation of model fit:

```
> quartz()
> par(mfrow=c(2,2))
> plot(mod)
```

- What should we do about the residual trend?
- How to deal with the heteroskedasticity?
- What should we do about the right tail of the residuals?



Dealing with lack of fit

- It is clear from the descriptive and residual plots that we might want to add quadratic terms to the regression:

```
> mod6 = lm(mpg ~ cylinders + horsepower + I(horsepower^2) + weight +
I(weight^2) + displacement + I(displacement^2) + acceleration, data=dat)
> summary(mod6)

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)        6.256e+01  4.455e+00  14.042  < 2e-16 ***
cylinders          7.847e-01  4.236e-01   1.852 0.064750 .  
horsepower        -2.969e-01  5.328e-02  -5.571 4.78e-08 ***
I(horsepower^2)    7.249e-04  1.819e-04   3.984 8.11e-05 ***
weight            -2.050e-03  3.381e-03  -0.606 0.544653    
I(weight^2)        9.793e-08  4.584e-07   0.214 0.830935    
displacement      -8.150e-02  2.428e-02  -3.356 0.000869 ***
I(displacement^2)  1.157e-04  4.353e-05   2.658 0.008193 ** 
acceleration      -3.866e-01  1.326e-01  -2.916 0.003758 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.731 on 393 degrees of freedom
Multiple R-squared:  0.6372, Adjusted R-squared:  0.6335
F-statistic: 172.6 on 4 and 393 DF,  p-value: < 2.2e-16
```

Dealing with lack of fit

- After doing some stepwise regression

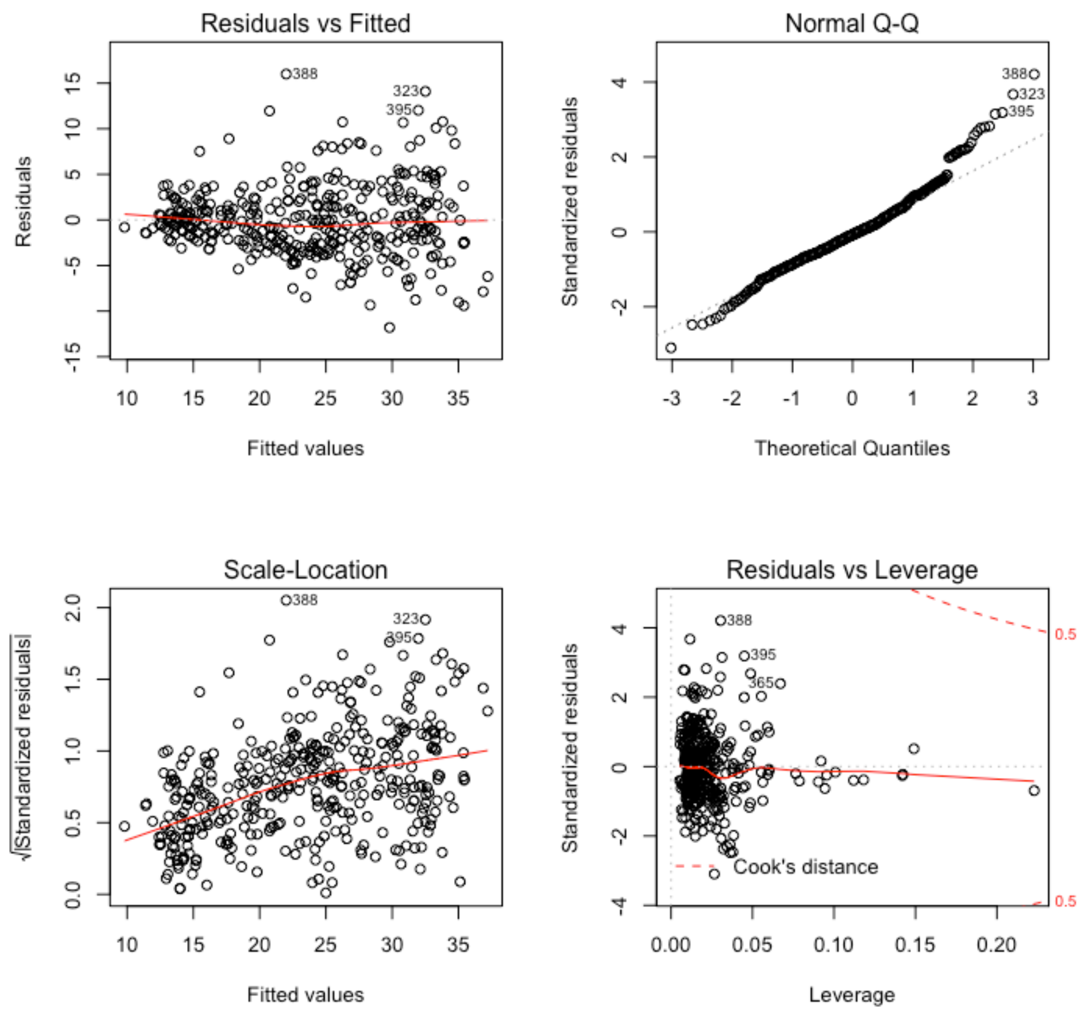
```
> mod7 = lm(mpg ~ cylinders + horsepower + I(horsepower^2) +
displacement + I(displacement^2) + acceleration, data=dat)
> summary(mod7)

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        63.9777080  3.1315133  20.430  < 2e-16 ***
cylinders           0.8473024  0.3978732   2.130 0.033840 *
horsepower         -0.3396011  0.0443881  -7.651 1.62e-13 ***
I(horsepower^2)     0.0008323  0.0001660   5.015 8.11e-07 ***
displacement       -0.0933867  0.0178144  -5.242 2.62e-07 ***
I(displacement^2)   0.0001239  0.0000344   3.601 0.000359 ***
acceleration       -0.5072507  0.1074119  -4.722 3.27e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.731 on 393 degrees of freedom
Multiple R-squared:  0.6372, Adjusted R-squared:  0.6335
F-statistic: 172.6 on 4 and 393 DF,  p-value: < 2.2e-16
```

Now weight is not significant, and cylinders, displacement and acceleration are.

Residual analysis

Residuals are not perfect (normal Q-Q plot still shows a heavy right tail and you see some heteroscedasticity), but at least the mean looks better.

You might need to log transform the response to address the remaining issue.



Predictions

- Predictions at new values are easy to obtain:

```
> mod5 = lm(mpg ~ horsepower + weight, data=dat)
> xn  = data.frame(horsepower = c(60,90,120), weight = c(2500,
3250, 4000))
> gpm.pred = predict(mod5, xn, se.fit=TRUE)
> gpm.pred
$fit
       1        2        3
28.31665 22.55194 16.78724

$se.fit
        1         2         3
0.3767431 0.3581450 0.4317328

$df
[1] 389

$residual.scale
[1] 4.240169
```

ANOVA models

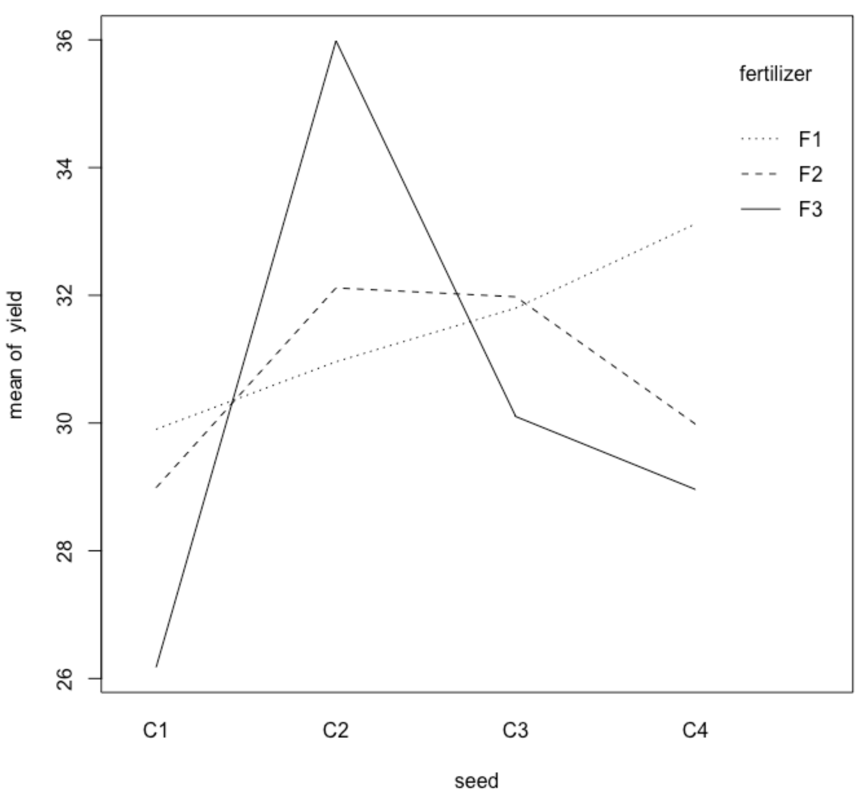
- We want to investigate now how the type of fertilizer and seed used to grow maize affects the yield.
- In this case the response is continuous, but we have two categorical predictors, one with 3 levels (fertilizer) and one with 4 levels (seed).
- Four replicates were taken for each combination of factors.
- Questions of interest:
  - Does the effect of fertilizer depend on the type of seed?
  - If that is not the case, is there an effect from either fertilizer or seed on their own.

Descriptive analysis

- The first thing to do is to construct interaction plots:

```
> yields = read.table(
"yields.csv", header=T)
> attach(yields)
> interaction.plot(seed,
fertilizer, yield)
```

- The graph suggests the presence of an interaction between fertilizer and seed (i.e., the type of seed affects how well the fertilizer works).



Testing for the presence of an interaction

- We can use an F test to determine whether there is actually a significant interaction between the variables:

```
> mod = lm(yield ~ fertilizer*seed, data=yields)
> anova(mod)
Analysis of Variance Table

Response: yield
                Df  Sum Sq Mean Sq F value    Pr(>F)    
fertilizer       2  10.528   5.264  90.986 8.361e-15 ***
seed             3 133.843  44.614 771.155 < 2.2e-16 ***
fertilizer:seed  6 121.206  20.201 349.171 < 2.2e-16 ***
Residuals       36   2.083   0.058                      
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- **The interaction (`fertilizer:seed`) appears to the highly significant in this dataset. Hence, we do no further testing.**
- If the interaction had not been significant, then we refit the model without it and would test for the significance of the main effects (`fertilizer` and `seed`).

Pairwise multiple comparisons

- Since there is an interaction, we might want to know what fertilizer is most effective for each type of seed.
- We can get the mean yields for each combination of factor using `aggregate()`.

```
> aggregate(yield, by=list(seed,fertilizer), mean)
   Group.1 Group.2        x
1       C1      F1 29.90229
2       C2      F1 30.95948
3       C3      F1 31.79963
4       C4      F1 33.11874
5       C1      F2 28.98773
6       C2      F2 32.11534
7       C3      F2 31.97647
8       C4      F2 29.98227
9       C1      F3 26.17516
10      C2      F3 35.98370
11      C3      F3 30.10192
12      C4      F3 28.95868
```

Pairwise multiple comparisons

- Consider for example seed C1.
- In that case the highest yield seems to be associated with fertilizer F3. But are the differences statistically significant?
- To answer that question we need to run pairwise tests

```
> pairwise.t.test(yield[seed=="C1"], fertilizer[seed=="C1"],
p.adjust.method="bonferroni")

        Pairwise comparisons using t tests with pooled SD

data:  yield[seed == "C1"] and fertilizer[seed == "C1"]

       F1      F2
F2 0.0041  -
F3 5.3e-08 6.2e-07

P value adjustment method: bonferroni
```

All p-values are < 0.05, so all differences in mean are statistically significant!

Simplest possible correction for multiple comparisons

Contrasts

- When categorical variables (factors) are included in a linear model, computation requires that they be encoded using dummy variables.
- There is an infinite number of potential encodings, each corresponding to one different set of constraints that ensure that the parameters of the model are estimable.

Contrasts

- Consider a simple example in which we are trying to explain writing skill as a function of race. The ANOVA model takes the form

$$y_{i,j} = \mu + \alpha_i + \varepsilon_{i,j} \qquad j = 1,\ldots,J_i \qquad i = 1,\ldots,I$$

- For example:

```
> hsb2 = read.table("hsb2.csv", sep=",", header=T)
> hsb2$race.f = factor(hsb2$race,
labels=c("Hispanic", "Asian", "African-Am",
"Caucasian"))
> tapply(hsb2$write, hsb2$race.f, mean)
  Hispanic      Asian African-Am  Caucasian
  46.45833   58.00000   48.20000   54.05517
```

- Here we have $I = 4$ groups and 5 parameters (μ plus 4 αs), so one constraint needs to be introduced!

Contrasts

- One popular options is for the first category to be the baseline level and the regression coefficients to represent differences with respect to the baseline level, i.e., $\alpha_1 = 0$.

```
> contrasts(hsb2$race.f) = contr.treatment(nlevels(hsb2$race.f))
> mod = lm(write ~ race.f, hsb2)
> summary(mod)
Call:
lm(formula = write ~ race.f, data = hsb2)

Residuals:
     Min       1Q   Median       3Q      Max 
-23.0552  -5.4583   0.9724   7.0000  18.8000 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   46.458      1.842  25.218  < 2e-16 ***
race.f2       11.542      3.286   3.512 0.000552 ***
race.f3        1.742      2.732   0.637 0.524613    
race.f4        7.597      1.989   3.820 0.000179 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.025 on 196 degrees of freedom
Multiple R-squared:  0.1071,	Adjusted R-squared:  0.0934 
F-statistic: 7.833 on 3 and 196 DF,  p-value: 5.785e-05
```

Contrasts

- Note that the estimate of the intercept is the mean for Hispanics (see previous slide), the one for `race.f2` is the difference between the mean of Hispanics and the mean for Asians, 58.000 - 46.458 = 11.542, and so on ...
- It is illustrative to look at the matrix of contrasts:

```
> contr.treatment(nlevels(hsb2$race.f))
  2 3 4
1 0 0 0
2 1 0 0
3 0 1 0
4 0 0 1
```

Mean of the second factor is the intercept plus the coefficient of the first dummy variable

- This type of contrast is the default in R.

Contrasts

- You can change the baseline level for the contrast (now African-Americans are the baseline group):

```
> contrasts(hsb2$race.f) = contr.treatment(4)[c(3,2,1,4),]
> mod = lm(write ~ race.f, hsb2)
> summary(mod)
Call:
lm(formula = write ~ race.f, data = hsb2)

Residuals:
     Min       1Q   Median       3Q      Max 
-23.0552  -5.4583   0.9724   7.0000  18.8000 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   48.200      2.018  23.884  < 2e-16 ***
race.f2        9.800      3.388   2.893  0.00425 ** 
race.f3       -1.742      2.732  -0.637  0.52461    
race.f4        5.855      2.153   2.720  0.00712 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.025 on 196 degrees of freedom
Multiple R-squared:  0.1071,	Adjusted R-squared:  0.0934 
F-statistic: 7.833 on 3 and 196 DF,  p-value: 5.785e-05
```

Contrasts

- Alternatively, we can interpret the intercept as the grand mean (mean of group means):

```
> contrasts(hsb2$race.f) = contr.sum(nlevels(hsb2$race.f))
> mod = lm(write ~ race.f, hsb2)
> summary(mod)
Call:
lm(formula = write ~ race.f, data = hsb2)

Residuals:
     Min       1Q   Median       3Q      Max
-23.0552  -5.4583   0.9724   7.0000  18.8000

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  51.6784     0.9821  52.619  < 2e-16 ***
race.f1      -5.2200     1.6314  -3.200  0.00160 **
race.f2       6.3216     2.1603   2.926  0.00384 **
race.f3      -3.4784     1.7323  -2.008  0.04602 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.025 on 196 degrees of freedom
Multiple R-squared:  0.1071,   Adjusted R-squared:  0.0934
F-statistic: 7.833 on 3 and 196 DF,  p-value: 5.785e-05
```

Contrasts

- Note that:
  - The intercept is indeed the grand mean (46.45833 + 58.00000 + 48.20000 + 54.05517)/4 = 51.67837
  - The coefficient for race.f1 is the difference between the mean of Hispanics and the grand mean and 46.45833 – 51.67837 = -5.2200 (“Hispanic effect”).
  - The coefficient for race.f2 is the difference between the mean of Asians and the grand mean and 58 – 51.67837 = 6.32163 (“Asianeffect”).
  - The coefficient for race.f3 is the difference between the mean of African-Americans and the grand mean and 48.2 – 51.67837 = -3.47837 (“African-American effect”).
  - The “Caucassian” effect is simply 3×51.67837 – 46.45833 – 58.00000 – 48.2 = 2.37678.

Contrasts

- The matrix of contrasts in this case is

```
> contr.sum(nlevels(hsb2$race.f))
  [,1] [,2] [,3]
1    1    0    0
2    0    1    0
3    0    0    1
4   -1   -1   -1
```

- Other options include `contr.poly` (for equally-space ordinal variables, useful to asses trends) and `contr.helmert` (comparing each level to the mean of all subsequent ones).
- You can create your own encoding by providing an appropriate matrix of contrasts!