# Week 10
# Model Selection

# F tests

- Two types of F test:
  - Overall F test – test for the usefulness of the model
  - Partial F test – test for linear restrictions
- $S_y^2 = \frac{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}{n-1} = \frac{S_{yy}}{n-1} = \frac{TSS}{n-1}$

  $S_{yy}$ = TSS = (n – 1) $S_y^2$ (total variation in Y)
- TSS = RegSS + RSS

# ANOVA table

| Source of Variation | SS | df | MS | F |
|---|---|---|---|---|
| Regression | RegSS | p – 1 | RegMS | $F = \frac{\mathrm{RegMS}}{\mathrm{MSE}}$ |
| Residual | RSS | n – p | MSE | |
| Total | TSS | n – 1 | | |

RegSS + RSS = TSS

(p – 1) + (n – p) = (n – 1)

RegMS + MSE ≠ $S_y^2$

# F tests

- Overall F test – test for the usefulness of the model

$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \varepsilon_i$

$H_0$: $\beta_1 = \beta_2 = \ldots = \beta_k = 0$ (the model is not useful)

$H_1$: At least one $\beta_j \neq 0$ $(j = 1, 2, 3, \ldots, k)$

$F_{stat} = \frac{\mathrm{RegMS}}{\mathrm{MSE}} \sim F_{p-1,n-p}$ under $H_0$ or $F_{stat} = \frac{\frac{R^2}{p-1}}{\frac{1-R^2}{n-p}} \sim F_{p-1,n-p}$ under $H_0$

Rejecting $H_0$ indicates that the regression is highly significant; i.e., at least one of the predictor variables is contributing significant information for the prediction of the dependent variable.

# F tests

- Partial F test – test for linear restrictions

$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + v_i$
$H_0$: $\beta_2 = \beta_3 = 0$ against $H_1$: At least one $\beta_j \neq 0$ $(j = 2, 3)$

Full model: $Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + v_i$ => $R_f^2$, $RSS_f$
Reduced model: $Y_i = \beta_0 + \beta_1 X_1 + \beta_4 X_4 + \beta_5 X_5 + u_i$ => $R_r^2$, $RSS_r$

$F_{stat} = \frac{(RSS_r - RSS_f)/q}{RSS_f/(n-p)} \sim F_{q, n-p}$ under $H_0$

where q is the number of restrictions. In this example, q = 2

# Example

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon$

```
> reg1 <- lm(y ~ X1+X2+X3+X4, data=hprice)
> summary(reg1)

Call:
lm(formula = y ~ X1 + X2 + X3 + X4, data = hprice)

Residuals:
    Min      1Q  Median      3Q     Max 
-12.700  -1.616   0.984   2.510  11.759 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  18.7633     9.2074   2.038  0.06889 .  
X1            6.2698     0.7252   8.645 5.93e-06 ***
X2          -16.2033     6.2121  -2.608  0.02611 *  
X3           -2.6730     4.4939  -0.595  0.56519    
X4           30.2705     6.8487   4.420  0.00129 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.849 on 10 degrees of freedom
Multiple R-squared:  0.9714,    Adjusted R-squared:  0.9599 
F-statistic:  84.8 on 4 and 10 DF,  p-value: 1.128e-07
```

# Example

```
> anova(reg1)
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq  F value    Pr(>F)    
X1         1 14829.3 14829.3 316.1025 6.76e-09 ***
X2         1     0.9     0.9   0.0184 0.894652    
X3         1   166.4   166.4   3.5472 0.089023 .  
X4         1   916.5   916.5  19.5356 0.001294 ** 
Residuals 10   469.1    46.9                      
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

TSS = 14829.3 + 0.9 + 166.4 + 916.5 + 469.1 = 16382.2

RegSS = 14829.3 + 0.9 + 166.4 + 916.5 = 15913.1

$H_0$: $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$ against $H_1$: at least one $\beta_j \neq 0$ ($j = 1, 2, 3, 4$)

Decision rule based on p-value: reject $H_0$ if p-value < $\alpha$

Decision: reject $H_0$ because p-value = 1.128e-07 < 0.05

Conclusion: There is sufficient evidence to show that the model is useful; i.e., at least one of the independent variables is contributing significant information for the prediction of the dependent variable.

# Example

Suppose we want to test if $X_1$ and $X_3$ are jointly significantly.
$H_0$: $\beta_1 = \beta_3 = 0$ against $H_1$: at least one $\beta_j \neq 0$ ($j = 1, 3$)

Full model: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon$
Reduced model: $Y = \beta_0 + \beta_2 X_2 + \beta_4 X_4 + u$

# Example

```
Call:
lm(formula = y ~ X1 + X2 + X3 + X4, data = hprice)

Residuals:
    Min      1Q  Median      3Q     Max 
-12.700  -1.616   0.984   2.510  11.759 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  18.7633     9.2074   2.038  0.06889 .  
X1            6.2698     0.7252   8.645 5.93e-06 ***
X2          -16.2033     6.2121  -2.608  0.02611 *  
X3           -2.6730     4.4939  -0.595  0.56519    
X4           30.2705     6.8487   4.420  0.00129 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.849 on 10 degrees of freedom
Multiple R-squared:  0.9714,	Adjusted R-squared:  0.9599 
F-statistic:  84.8 on 4 and 10 DF,  p-value: 1.128e-07
```

$$F_{stat} = \frac{(RSS_r - RSS_f)/q}{RSS_f/(n-p)} = \frac{(4968.1545 - 469.1295)/2}{469.1295/(15-5)} = 47.951$$

p-value = $P(F_{2,10} > 47.951) \approx 0$

```
> reg2 <- lm(y ~ X2+X4, data=hprice)
> summary(reg2)

Call:
lm(formula = y ~ X2 + X4, data = hprice)

Residuals:
    Min      1Q  Median      3Q     Max 
-25.218 -15.291  -1.906  14.027  33.094 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   29.559     22.655   1.305  0.21643   
X2            -4.533     16.491  -0.275  0.78809   
X4            55.890     15.445   3.619  0.00352 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 20.35 on 12 degrees of freedom
Multiple R-squared:  0.6967,	Adjusted R-squared:  0.6462 
F-statistic: 13.78 on 2 and 12 DF,  p-value: 0.0007779
```

```
> fm <- lm(y ~ x1+x2+x3+x4, data=dat)
> rm <- lm(y ~ x2+x4,data=dat)
> fobs <- ((deviance(rm)-deviance(fm))/2)/(deviance(fm)/10)
> c(deviance(rm), deviance(fm), fobs)
[1] 4968.15451  469.12948   47.95078
> pf(fobs,2,10,lower.tail=F)
[1] 7.507376e-06
```

# Example

```
> fm <- lm(y ~ x1+x2+x3+x4, data=dat)
> rm <- lm(y ~ x2+x4,data=dat)
> anova(rm, fm)
Analysis of Variance Table

Model 1: y ~ x2 + x4
Model 2: y ~ x1 + x2 + x3 + x4
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     12 4968.2
2     10  469.1  2      4499 47.951 7.507e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

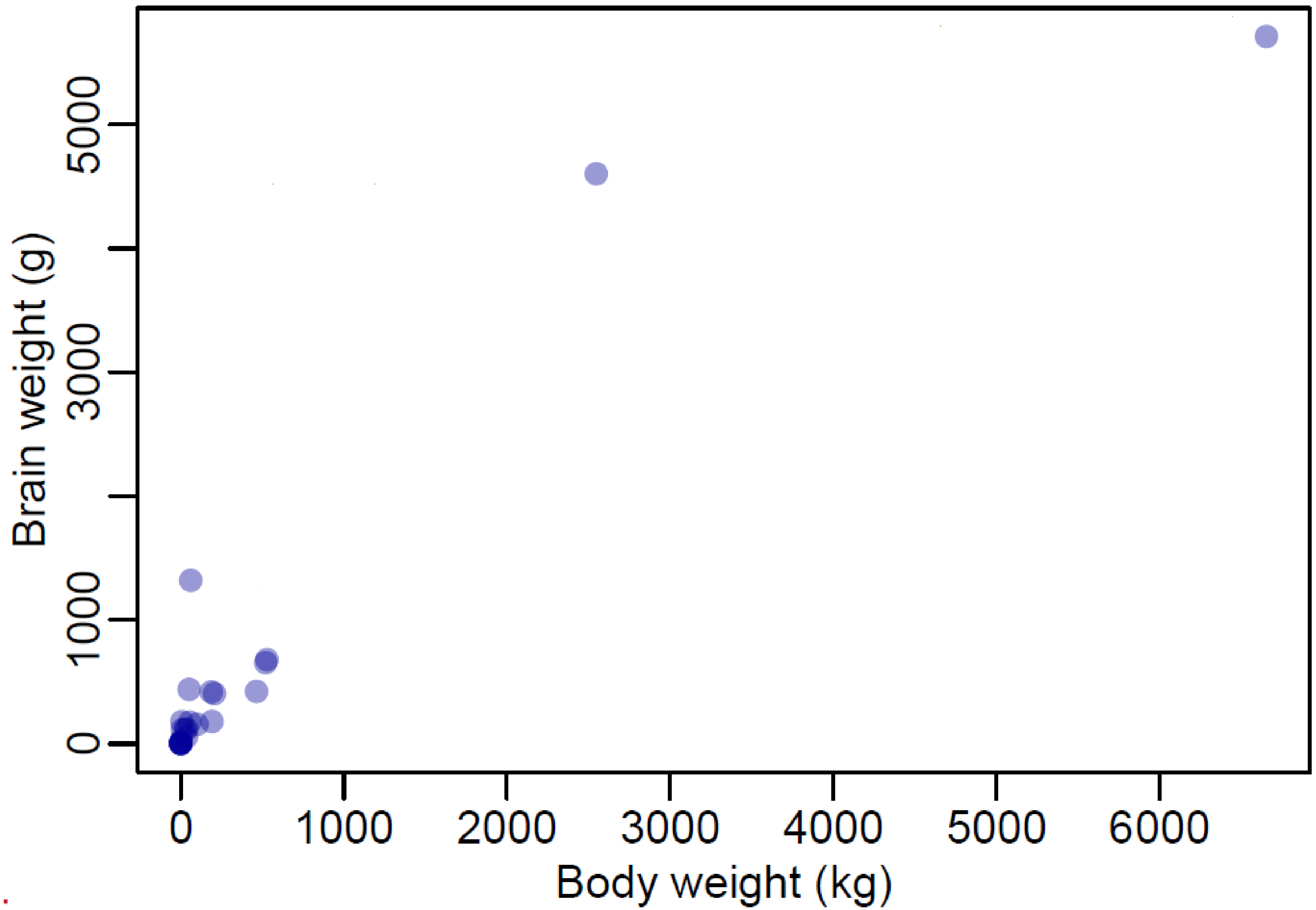
```
> library(MASS)
> library(car)
>
> fm <- lm(y ~ x1+x2+x3+x4, data=dat)
>
> linearHypothesis(fm,c("x1=0","x3=0"))
Linear hypothesis test

Hypothesis:
x1 = 0
x3 = 0

Model 1: restricted model
Model 2: y ~ x1 + x2 + x3 + x4

  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     12 4968.2
2     10  469.1  2      4499 47.951 7.507e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

$H_0$: $\beta_1 = \beta_3 = 0$ against $H_1$: at least one $\beta_j \neq 0$ ($j = 1, 3$)
Decision rule: reject $H_0$ if p-value $< \alpha$
Decision: reject $H_0$ because p-value = 7.507e-06 < 0.05
Conclusion: There is sufficient evidence to show that $X_1$ and $X_3$ are jointly significant.

# Model selection

- If there is no explicit theory to suggest which explanatory variables should be included in the regression model, the set of predictor variables used in the final regression model must be determined by analysis of the data.
- We want to explain the data in the simplest way – redundant predictors should be removed.
- Unnecessary predictors will add noise to the estimation of other quantities that we are interested in. Degrees of freedom will be wasted.
- If the number of potential variable combinations is too high to conduct a manual analysis, we can opt for automatic search procedures.
- With such a procedure, each test is acted upon sequentially to find the best performing model (based on criteria such as $R^2$, AIC, BIC).
- The common selection algorithms are forward, backward and stepwise selection algorithms.
- Prior to variable selection: (i) identify outliers and influential points and (ii) add in any transformations of the variables that seem appropriate.

# Example



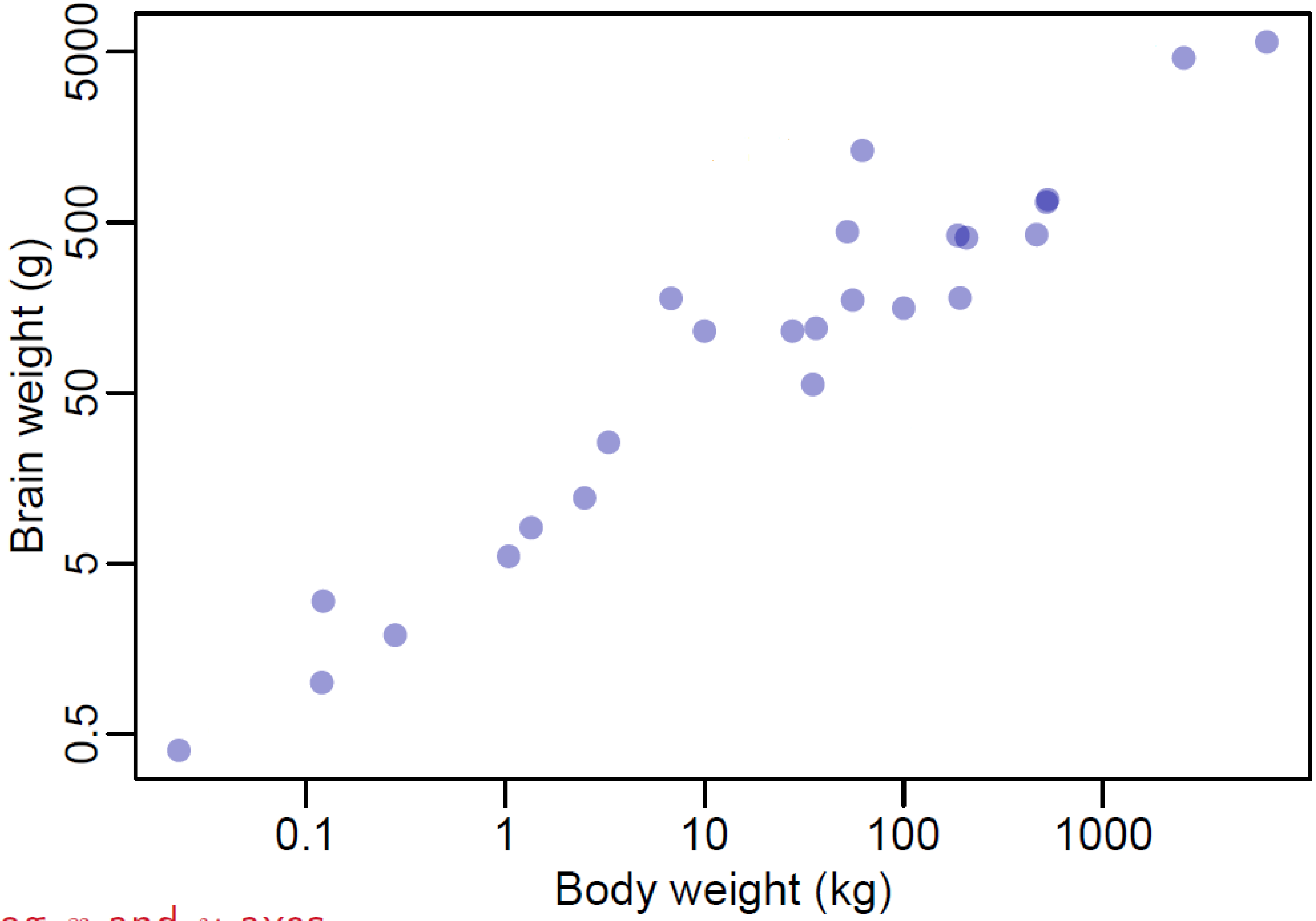

Linear axes

# Example

$n = 24$ mammals

| Animal | Body (kg) | Brain (g) | Animal | Body (kg) | Brain (g) |
|---|---|---|---|---|---|
| Mouse | 0.023 | 0.4 | Chimpanzee | 52.16 | 440 |
| Golden hamster | 0.12 | 1 | Sheep | 55.5 | 175 |
| Mole | 0.122 | 3 | Human | 62 | 1320 |
| Guinea pig | 1.04 | 5.5 | Jaguar | 100 | 157 |
| Mountain beaver | 1.35 | 465 | Donkey | 187.1 | 419 |
| Rabbit | 2.5 | 12.1 | Pig | 192 | 180 |
| Cat | 3.3 | 25.6 | Gorilla | 207 | 406 |
| Rhesus monkey | 6.8 | 179 | Cow | 465 | 423 |
| Potar monkey | 10 | 115 | Horse | 521 | 655 |
| Goat | 27.66 | 115 | Giraffe | 529 | 680 |
| Kangaroo | 35 | 56 | Asian elephant | 2547 | 4603 |
| Grey wolf | 36.33 | 119.5 | African elephant | 6654 | 5712 |

# Example




Log $x$ and $y$ axes

# Principle of parsimony

- Most practitioners of regression analysis adopt the principle of parsimony
- In situations where 2 competing models are found to have essentially the same predictive power, the model with the fewer number of β's (i.e., the more parsimonious model) is selected.

# Example – Cheese tasting data

- Data on production of cheddar cheese from the LaTrobe Valley of Victoria.
- Taste of the final product is related to the concentration of several chemicals in the cheese.
- $n = 30$ samples of cheese were tasted by experts, and the following four variables recorded:

| | | | |
|---|---|---|---|
| `taste` | Tasters' ratings | `Acetic` | Acetic acid in cheese |
| `H2S` | Hydrogen sulphide in cheese | `Lactic` | Lactic acid in the cheese. |

# Theory – Backward variable selection

1. Start with model containing all possible explanatory variables.
2. For each variable in turn, investigate effect of removing variable from current model.
3. Remove the least informative variable, unless this variable is nonetheless supplying significant information about the response.
4. Go to step 2. Stop only if all variables in the current model are important.

Comments

- Implementation depends on how we assess the importance of variables.
- Possibilities are to use p-values (F, t or $\chi^2$ tests), GoF criteria ($R^2$ or $R_a^2$) or especially designed selection criteria to measure what it means to be informative.

# Example – Cheese data: Backward selection

- Of interest to the manufacturers to relate the cheese's taste to the 'chemical' variables.
- Therefore construct multiple linear regression model of `taste` on other variables.
- Variable selection will allow us to produce a parsimonious model.
- Backwards variable selection starts with the full model (i.e. with all predictors).
- Let us have a look at the data first and then we will run a backward selection based on the F-test with the deletion of the least significant variable as long as $p_{\text{out}} > 5\%$.

# Theory – The drop1 and update command

- For a response variable Y and explanatory variables x.1,...,x.k stored in the data frame dat consider

```
M1 = lm(Y ~ ., data = dat)
```

- Then, the R-command drop1(M1, test = "F") returns a number of information criteria for all variables used in M1 to model the response variable.
- To efficiently delete a variable from regression model M1, say x.1, the update command can be used:

```
M2 = update(M1, . ~ . - x.1, data = dat)
```

- The general syntax for updating models is:

```
update(old.model, new.formula, ...)
```

- Note that full stops in the updated formula stand for 'whatever was in the comparison position in the old formula'.

# Backward Elimination

```
> cheese = read.table("cheese.txt", header = TRUE, row.names = NULL)
> M0 <- lm(taste ~ 1, data=cheese)  # null model
> MF <- lm(taste ~ ., data=cheese)  # full model
> drop1(MF, test="F")
Single term deletions

Model:
taste ~ Acetic + H2S + Lactic
       Df Sum of Sq    RSS    AIC F value   Pr(>F)
<none>               2668.4 142.64
Acetic  1      0.55 2669.0 140.65  0.0054 0.941980
H2S     1   1007.66 3676.1 150.25  9.8182 0.004247 **
Lactic  1    533.32 3201.7 146.11  5.1964 0.031079 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> M1 = update(MF, .~. -Acetic, data=cheese)
> drop1(M1, test="F")
Single term deletions

Model:
taste ~ H2S + Lactic
       Df Sum of Sq    RSS    AIC F value   Pr(>F)
<none>               2669.0 140.65
H2S     1   1193.52 3862.5 149.74 12.0740 0.001743 **
Lactic  1    617.18 3286.1 144.89  6.2435 0.018850 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> summary(M1)

Call:
lm(formula = taste ~ H2S + Lactic, data = cheese)

Residuals:
    Min      1Q  Median      3Q     Max
-17.343  -6.530  -1.164   4.844  25.618

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -27.592      8.982  -3.072  0.00481 **
H2S            3.946      1.136   3.475  0.00174 **
Lactic        19.887      7.959   2.499  0.01885 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.942 on 27 degrees of freedom
Multiple R-squared:  0.6517,    Adjusted R-squared:  0.6259
F-statistic: 25.26 on 2 and 27 DF,  p-value: 6.551e-07
```

# Comments – Cheese data

- First pass through reduction algorithm:
  - Both `H2S` and `Lactic` should not be dropped, otherwise considerably worse fit than the full model (as evidenced by the p-values of $0.004$ and $0.031$).
  - However, deletion of `Acetic` makes little difference (e.g. at the 5% sign. level) in terms of model fit (partial p-value of $0.942$) $\Rightarrow$ omit this variable.
  - If there had been more than one variable with p-value greater than 0.05, then we would have removed the covariate with largest corresponding p-value.
- Second pass through reduction algorithm:
  - Neither of the covariates `H2S` and `Lactic` can be removed from the model without an important loss of fit.
  - Hence, 'best' model for the data (according to backward selection with significance level 5%) is

$$E[\texttt{taste}] = -27.59 + 3.95 \cdot \texttt{H2S} + 19.89 \cdot \texttt{Lactic}.$$

# Theory – Forward variable selection

1. Start with model containing no possible explanatory variables, i.e.

$$\mathbf{m} = \emptyset.$$

2. For each variable in turn, investigate effect of adding variable from current model.
3. Add the most informative/significant variable, unless this variable is not supplying significant information about the response.
4. Go to step 2. Stop only if none of the non-included variables is important.

Comments

- Implementation depends on how we assess the importance of variables.
- Possibilities are to use p-values (F, t or $\chi^2$ tests), GoF criteria ($R^2$ or $R_a^2$) or especially designed selection criteria to measure what it means to be informative.

# Theory – The add1 command

- For a response variable Y and explanatory variables x.1,...,x.k stored in the data frame dat consider M1 = lm(Y ∼ 1, data = dat) with explanatory variables $\mathbf{m}_1$.
- Then, the R-command

```
add1(M1, scope = ~ x.1 + x.2 + ... + x.k, data = dat, test = "F")
```

returns a number of information criteria for all variables specified after the option 'scope = ∼' to model the response variable. Alternatively, list all the variable names by coding up a full model

```
Mf = lm(Y ~ ., data = dat)
add1(M1, scope = Mf, data = dat, test = "F")
```

# Forward selection

```
> # Forward
> MF <- lm(taste ~ ., data=cheese)  # full model
> M0 <- lm(taste ~ 1, data=cheese)  # null model
> add1(M0, scope=MF, data=cheese, test="F")
Single term additions

Model:
taste ~ 1
       Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>              7662.9 168.29
Acetic  1    2314.1 5348.7 159.50  12.114  0.001658 **
H2S     1    4376.7 3286.1 144.89  37.293 1.374e-06 ***
Lactic  1    3800.4 3862.5 149.74  27.550 1.405e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> M1=update(M0, .~. + H2S, data=cheese)
> add1(M1, scope=MF, data=cheese, test="F")
Single term additions

Model:
taste ~ H2S
       Df Sum of Sq    RSS    AIC F value  Pr(>F)
<none>              3286.1 144.89
Acetic  1     84.41 3201.7 146.11  0.7118 0.40625
Lactic  1    617.18 2669.0 140.65  6.2435 0.01885 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> M2=update(M1, .~. + Lactic, data=cheese)
> add1(M2, scope=MF, data=cheese, test="F")
Single term additions

Model:
taste ~ H2S + Lactic
       Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>              2669.0 140.65
Acetic  1   0.55427 2668.4 142.64  0.0054  0.942

> summary(M2)

Call:
lm(formula = taste ~ H2S + Lactic, data = cheese)

Residuals:
    Min      1Q  Median      3Q     Max
-17.343  -6.530  -1.164   4.844  25.618

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -27.592      8.982  -3.072  0.00481 **
H2S            3.946      1.136   3.475  0.00174 **
Lactic        19.887      7.959   2.499  0.01885 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.942 on 27 degrees of freedom
Multiple R-squared:  0.6517,    Adjusted R-squared:  0.6259
F-statistic: 25.26 on 2 and 27 DF,  p-value: 6.551e-07
```

# Theory – Stepwise variable selection

1. Start with some model, typically null model (with no explanatory variables) or full model (with all variables).
2. For each variable in the current model, investigate effect of removing it.
3. Remove the least informative variable, unless this variable is nonetheless supplying significant information about the response.
4. For each variable not in the current model, investigate effect of including it.
5. Include the most statistically significant variable not currently in model (unless no significant variable exists).
6. Go to step 2. Stop only if no change in steps 2–5.

- In R for F-tests: use a combination of `add1()` and `drop1()`.
- In R for using AIC or BIC: use command `step`, which runs an automated search.

# Stepwise regression

First pass through algorithm We will perform stepwise variable selection starting from the ‘null model’ (i.e. the model with no explanatory variables) using the following exclusion/inclusion level of significance:

$$p_{\text{out}} = 0.20 \quad \text{and} \quad p_{\text{in}} = 0.10.$$

Stepwise regression requires two significance levels:
one for adding variables ($p_{in}$) and one for removing variables ($p_{out}$) .

The cutoff probability for adding variables should be less than the cutoff probability for removing variables so that the procedure does not get into an infinite loop ($p_{in} < p_{out}$)

# Stepwise regression

There are no variables to drop from M0. Hence, the algorithm starts at step 4.

```
> # Stepwise
> MF <- lm(taste ~ ., data=cheese)  # full model
> M0 <- lm(taste ~ 1, data=cheese)  # null model
> add1(M0, scope = MF, data = cheese, test = "F")
Single term additions

Model:
taste ~ 1
       Df Sum of Sq    RSS    AIC F value    Pr(>F)    
<none>              7662.9 168.29                      
Acetic  1    2314.1 5348.7 159.50  12.114  0.001658 ** 
H2S     1    4376.7 3286.1 144.89  37.293 1.374e-06 ***
Lactic  1    3800.4 3862.5 149.74  27.550 1.405e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> M1 = update(M0, . ~ . + H2S, data = cheese)
> drop1(M1, test = "F")
Single term deletions

Model:
taste ~ H2S
       Df Sum of Sq    RSS    AIC F value    Pr(>F)    
<none>              3286.1 144.89                      
H2S     1    4376.7 7662.9 168.29  37.293 1.374e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> add1(M1, scope = MF, data = cheese, test = "F")
Single term additions

Model:
taste ~ H2S
       Df Sum of Sq    RSS    AIC F value  Pr(>F)  
<none>              3286.1 144.89                  
Acetic  1     84.41 3201.7 146.11  0.7118 0.40625  
Lactic  1    617.18 2669.0 140.65  6.2435 0.01885 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> M2 = update(M1, . ~ . + Lactic, data = cheese)
> drop1(M2, test = "F")
Single term deletions

Model:
taste ~ H2S + Lactic
       Df Sum of Sq    RSS    AIC F value   Pr(>F)   
<none>              2669.0 140.65                    
H2S     1   1193.52 3862.5 149.74 12.0740 0.001743 **
Lactic  1    617.18 3286.1 144.89  6.2435 0.018850 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# Stepwise regression

```
> add1(M2, scope = MF, data = cheese, test = "F")
Single term additions

Model:
taste ~ H2S + Lactic
       Df Sum of Sq    RSS    AIC F value Pr(>F)
<none>              2669.0 140.65
Acetic  1   0.55427 2668.4 142.64  0.0054  0.942
```

There is no change in the model from steps 2 - 5.

```
> summary(M2)

Call:
lm(formula = taste ~ H2S + Lactic, data = cheese)

Residuals:
    Min      1Q  Median      3Q     Max 
-17.343  -6.530  -1.164   4.844  25.618 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -27.592      8.982  -3.072  0.00481 **
H2S            3.946      1.136   3.475  0.00174 **
Lactic        19.887      7.959   2.499  0.01885 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.942 on 27 degrees of freedom
Multiple R-squared:  0.6517,	Adjusted R-squared:  0.6259 
F-statistic: 25.26 on 2 and 27 DF,  p-value: 6.551e-07
```

# Example – Cheese data: Comments

- No need to add any more terms to `M3`.
- The 'best' model as selected by the stepwise procedure is

$$E[\texttt{taste}] = -27.59 + 3.95 \cdot \texttt{H2S} + 19.89 \cdot \texttt{Lactic}.$$

- This is the same model as was selected by backwards variable selection.
- Stepwise, forward and backward variable selection procedures will sometimes generate the same model, but this will not always be the case.

# Theory – Criticism of stepwise procedures

- Never run automated stepwise procedures on their own!
- Wilkinson (1984, p196, SYSTAT)

  > *"Stepwise regression is probably the most abused computerized statistical technique ever devised. If you think you need stepwise regression to solve a particular problem you have, it is almost certain you do not. Professional statisticians rarely use automated stepwise regression."*

- The main issue is that stepwise procedures potentially identify models that are only locally optimal.

# Theory – More goodness of fit criteria

- Recall for a linear regression model $\mathbf{m} \in \mathcal{M}$

$$R^2(\mathbf{m}) = \frac{S_{yy} - \mathsf{RSS}(\mathbf{m})}{S_{yy}}$$

$$R_a^2(\mathbf{m}) = 1 - \frac{\mathsf{RSS}(\mathbf{m})/(n - p_{\mathbf{m}})}{S_{yy}/(n-1)} \Rightarrow \hat{\mathbf{m}} = \operatorname{argmax}_{\mathbf{m} \in \mathcal{M}} R_a^2(\mathbf{m}) \; .$$

- $S_{yy}$ is independent of $\mathbf{m}$ and $\mathsf{RSS}(\mathbf{m}) = \sum_{i=1}^n R_{\mathbf{m}i}^2$.
- Why not $\sum_{i=1}^n |R_{\mathbf{m}i}|$ or some other loss function (metric)?
- Criteria having the structure

$$\sum_{i=1} \rho(R_i(\mathbf{m})) + \lambda(p_{\mathbf{m}}, n), \tag{1}$$

where $\rho(z)$ is a nonnegative loss function (e.g. $z^2$) & $\lambda$ is a penalty term.

# Theory – Akaike information criterion

- The Akaike information criterion (AIC) is defined by

$$\mathsf{AIC}(\mathbf{m}) = -2\,\mathsf{loglikelihood} + 2p_{\mathbf{m}} \overset{\epsilon_{\mathbf{m}i} \sim NID}{=} n \log\left(\frac{\mathsf{RSS}(\mathbf{m})}{n}\right) + 2p_{\mathbf{m}}$$

- 'Best' model using AIC is $\hat{\mathbf{m}}_{\mathsf{AIC}} = \mathsf{argmin}_{\mathbf{m} \in \mathcal{A}} \mathsf{AIC}(\mathbf{m})$.
- $\hat{\mathbf{m}}_{\mathsf{AIC}}$ is invariant under any strictly non-negative monotone transformation, especially constants.
- To return the AIC value of a regression model in R use `extractAIC`.

```
extractAIC(M3, k = 2)

## [1]   3.0000 140.6475
```

Note that `k = 2` relates to the constant in $2p_{\mathbf{m}}$

# Theory – BIC has tougher penalties

- AIC has a tendency to include too many variables. (*)
- Solution: Penalize more in equation (1)!
- Bayes information criterion (BIC):

$$BIC(\mathbf{m}) = \frac{\mathsf{RSS}(\mathbf{m})}{n\hat{\sigma}^2} + \log(n) \cdot p_{\mathbf{m}}$$

- BIC in R with additional option `k=log(n)` in function `step()`.

(*) BIC has been shown to be consistent if the data are generated by one model with fixed dimension $p_{\mathbf{m}_0}$, whereas AIC tends to overestimate the dimension in this case. On the other hand, BIC tends to better describe the data but AIC is known to be better for prediction purposes.

# Backward elimination

```
> MF <- lm(taste ~ ., data=cheese)  # full model
> M0 <- lm(taste ~ 1, data=cheese)  # null model
>
> M.back <- step(MF, scope=list(lower=M0, upper=MF), direction="backward", k=2)
Start:  AIC=142.64
taste ~ Acetic + H2S + Lactic

         Df Sum of Sq    RSS    AIC
- Acetic  1      0.55 2669.0 140.65
<none>                2668.4 142.64
- Lactic  1    533.32 3201.7 146.11
- H2S     1   1007.66 3676.1 150.25

Step:  AIC=140.65
taste ~ H2S + Lactic

         Df Sum of Sq    RSS    AIC
<none>                2669.0 140.65
- Lactic  1    617.18 3286.1 144.89
- H2S     1   1193.52 3862.5 149.74
```

```
> summary(M.back)

Call:
lm(formula = taste ~ H2S + Lactic, data = cheese)

Residuals:
    Min      1Q  Median      3Q     Max
-17.343  -6.530  -1.164   4.844  25.618

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -27.592      8.982  -3.072  0.00481 **
H2S            3.946      1.136   3.475  0.00174 **
Lactic        19.887      7.959   2.499  0.01885 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.942 on 27 degrees of freedom
Multiple R-squared:  0.6517,	Adjusted R-squared:  0.6259
F-statistic: 25.26 on 2 and 27 DF,  p-value: 6.551e-07

> extractAIC(M.back)
[1]   3.0000 140.6475
```

# Forward selection

```
> MF <- lm(taste ~ ., data=cheese)  # full model
> M0 <- lm(taste ~ 1, data=cheese)  # null model
>
> M.forw <- step(M0, scope=list(lower=M0, upper=MF), direction="forward", k=2)
Start:  AIC=168.29
taste ~ 1

         Df Sum of Sq    RSS    AIC
+ H2S     1    4376.7 3286.1 144.89
+ Lactic  1    3800.4 3862.5 149.74
+ Acetic  1    2314.1 5348.7 159.50
<none>                7662.9 168.29

Step:  AIC=144.89
taste ~ H2S

         Df Sum of Sq    RSS    AIC
+ Lactic  1    617.18 2669.0 140.65
<none>                3286.1 144.89
+ Acetic  1     84.41 3201.7 146.11

Step:  AIC=140.65
taste ~ H2S + Lactic

         Df Sum of Sq    RSS    AIC
<none>                2669.0 140.65
+ Acetic  1   0.55427 2668.4 142.64
```

```
> summary(M.forw)

Call:
lm(formula = taste ~ H2S + Lactic, data = cheese)

Residuals:
    Min      1Q  Median      3Q     Max
-17.343  -6.530  -1.164   4.844  25.618

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -27.592      8.982  -3.072  0.00481 **
H2S            3.946      1.136   3.475  0.00174 **
Lactic        19.887      7.959   2.499  0.01885 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.942 on 27 degrees of freedom
Multiple R-squared:  0.6517,    Adjusted R-squared:  0.6259
F-statistic: 25.26 on 2 and 27 DF,  p-value: 6.551e-07

> extractAIC(M.forw)
[1]   3.0000 140.6475
```

# Stepwise regression

```
> MF <- lm(taste ~ ., data=cheese)  # full model
> M0 <- lm(taste ~ 1, data=cheese)  # null model
>
> M.step <- step(M0, scope=list(lower=M0, upper=MF), direction="both", k=2)
Start:  AIC=168.29
taste ~ 1

         Df Sum of Sq    RSS    AIC
+ H2S     1    4376.7 3286.1 144.89
+ Lactic  1    3800.4 3862.5 149.74
+ Acetic  1    2314.1 5348.7 159.50
<none>                7662.9 168.29

Step:  AIC=144.89
taste ~ H2S

         Df Sum of Sq    RSS    AIC
+ Lactic  1     617.2 2669.0 140.65
<none>                3286.1 144.89
+ Acetic  1      84.4 3201.7 146.11
- H2S     1    4376.7 7662.9 168.29

Step:  AIC=140.65
taste ~ H2S + Lactic

         Df Sum of Sq    RSS    AIC
<none>                2669.0 140.65
+ Acetic  1      0.55 2668.4 142.64
- Lactic  1    617.18 3286.1 144.89
- H2S     1   1193.52 3862.5 149.74
```

```
> summary(M.step)

Call:
lm(formula = taste ~ H2S + Lactic, data = cheese)

Residuals:
    Min      1Q  Median      3Q     Max
-17.343  -6.530  -1.164   4.844  25.618

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -27.592      8.982  -3.072  0.00481 **
H2S            3.946      1.136   3.475  0.00174 **
Lactic        19.887      7.959   2.499  0.01885 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.942 on 27 degrees of freedom
Multiple R-squared:  0.6517,    Adjusted R-squared:  0.6259
F-statistic: 25.26 on 2 and 27 DF,  p-value: 6.551e-07

> extractAIC(M.step)
[1]   3.0000 140.6475
```

# Stepwise regression – BIC or SBC

```
> length(cheese$taste)
[1] 30
> # k = log(n) is sometimes referred to as BIC or SBC.
> M.step <- step(M0, scope=list(lower=M0, upper=MF), direction="both", k=log(30))
Start:  AIC=169.69
taste ~ 1

         Df Sum of Sq    RSS    AIC
+ H2S     1    4376.7 3286.1 147.69
+ Lactic  1    3800.4 3862.5 152.54
+ Acetic  1    2314.1 5348.7 162.31
<none>                7662.9 169.69

Step:  AIC=147.69
taste ~ H2S

         Df Sum of Sq    RSS    AIC
+ Lactic  1     617.2 2669.0 144.85
<none>                3286.1 147.69
+ Acetic  1      84.4 3201.7 150.31
- H2S     1    4376.7 7662.9 169.69

Step:  AIC=144.85
taste ~ H2S + Lactic

         Df Sum of Sq    RSS    AIC
<none>                2669.0 144.85
- Lactic  1    617.18 3286.1 147.69
+ Acetic  1      0.55 2668.4 148.25
- H2S     1   1193.52 3862.5 152.54
```

```
> summary(M.step)

Call:
lm(formula = taste ~ H2S + Lactic, data = cheese)

Residuals:
    Min      1Q  Median      3Q     Max 
-17.343  -6.530  -1.164   4.844  25.618 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -27.592      8.982  -3.072  0.00481 **
H2S            3.946      1.136   3.475  0.00174 **
Lactic        19.887      7.959   2.499  0.01885 * 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.942 on 27 degrees of freedom
Multiple R-squared:  0.6517,	Adjusted R-squared:  0.6259 
F-statistic: 25.26 on 2 and 27 DF,  p-value: 6.551e-07
```