STAT 210
Applied Statistics and Data Analysis
Week 11 - Summary

Joaquin Ortega

King Abdullah University of Science and Technology

Reminders

- The project report is due on Sunday, November 30.
- The second exam will be on Saturday, November 29, from 9:00 am to 12:00 noon in Room 2322 (LH1), Building 9.
- The project presentations will be on December 8 and 9.

V 36: Diagnostics

Identifying Unusual Observations

Deleted Residuals

We fit a regression model excluding the i-th case and denote by $\hat{y}_{i(i)}$ the predicted value corresponding to the vector of covariates $\mathbf{x}_i$.

The deleted residual, $\hat{\epsilon}_{(i)}$, is then defined as

$$\hat{\epsilon}_{(i)} = y_i - \hat{y}_{i(i)}.$$

It can be shown that

$$\hat{\epsilon}_{(i)} = y_i - \hat{y}_{i(i)} = \frac{\hat{\epsilon}_i}{1 - h_{ii}}.$$

Identifying Unusual Observations

The estimated variance is

$$Var(\hat{\epsilon}_{(i)}) = \frac{\hat{\sigma}^2_{(i)}}{1 - h_{ii}} = \frac{MSE_{(i)}}{1 - h_{ii}}.$$

The studentized deleted residual or **externally studentized residual** is defined as

$$r_i^* = \frac{\hat{\epsilon}_{(i)}}{\sqrt{Var(\hat{\epsilon}_{(i)})}} = \frac{\hat{\epsilon}_i}{\sqrt{MSE_{(i)}(1 - h_{ii})}}.$$

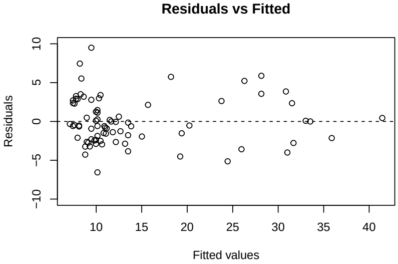
The `R` function `rstudent()` computes the studentized deleted residuals. The function `studres()` in the `MASS` package also computes these residuals.

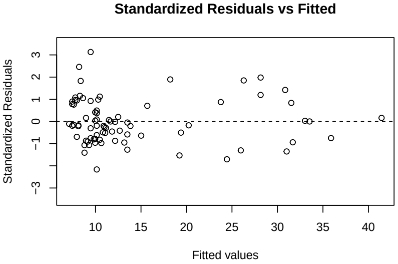
Bonferroni Correction

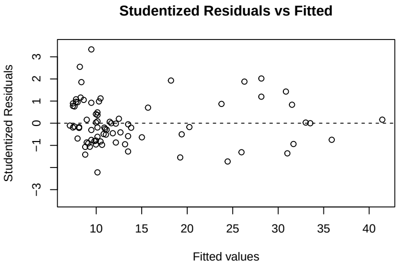
Even though it is very likely only a few ‘large’ r_i^*s will be of interest, by identifying them as large, all cases have implicitly been tested.

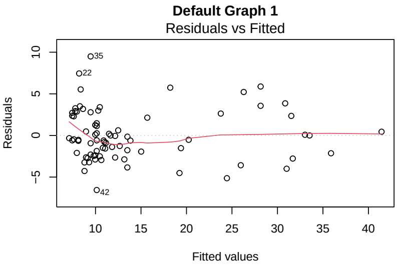
A Bonferroni approach is often used to control the overall significance level, where r_i^* values are declared significant if their absolute value exceeds $t_{1-\alpha/2n;n-p-1}$.

Identifying Unusual Observations









High Leverage Observations

While residuals are used to identify outlying y values, the hat matrix provides an analog for the x values.

The diagonal entry h_{ii} of the hat matrix $\mathbf{H}$ provides a measure of the distance of the i-th case from the centroid of the x observations.

The limits on h_{ii} are

$$1/n \leq h_{ii} \leq 1.$$

In general, a leverage value, h_{ii}, is considered large if it is more than twice as large as the mean leverage value $(2p/n)$.

High Leverage Observations

Observations with large h_{ii} are called **high leverage points**.

When the estimated parameters are substantially different with and without the i-th case, the i-th case is said to be **influential**.

Not all high leverage observations are influential.

Tools

Cook's Distance

Cook's distance D_i is a combined measure of the standardized residual (r_i) and the leverage value (h_{ii}) that produces a number used to assess the impact of removing the *i*-th observation on all the regression coefficients (β).

It is defined as

$$D_i = \frac{r_i^2}{p} \frac{h_{ii}}{1 - h_{ii}} = \frac{\hat{\varepsilon}^2}{pMSE} \frac{h_{ii}}{(1 - h_{ii})^2}.$$

D_i values are generally flagged for further scrutiny when they exceed $F_{0.5;p,n-p}$; however, the exact distribution of D_i is unknown, and the use of $F_{0.5;p,n-p}$ is only a suggestion.

In `R`, `cooks.distance()` will compute the D_is. The package `car` also has the function `cookd()`.

DFFITS

DFFITS, an abbreviation for ‘difference in fits,’ is a standardized measure of the amount by which the predicted value $\hat{y}_i$ changes when the i-th case is deleted from the data.

$$DFFITS_i = \frac{\hat{y}_i - \hat{y}_{(i)}}{\hat{\sigma}_{(i)}\sqrt{h_{ii}}}.$$

DFFITS values whose absolute value exceeds $2(p/n)^{1/2}$ generally require further scrutiny.

To compute DFFITS with `R`, use `dffits()`.

DFBETAS

DFBETAS is a standardized measure of the amount by which the k-th regression coefficient changes when the i-th observation is omitted from the data set.

$$DFBETAS_{k(i)} = \frac{\hat{\beta}_k - \hat{\beta}_{k(i)}}{\sqrt{\hat{\sigma}^2_{(i)} v_{k+1,k+1}}}$$

where $v_{k+1,k+1}$ is the $(k+1)$th diagonal entry $(k = 0, 1, \ldots, p)$ of $\mathbf{V} = (\mathbf{X}'\mathbf{X})^{-1}$.

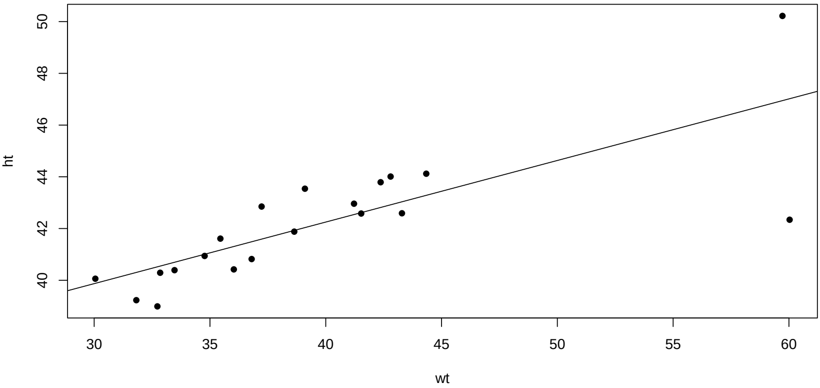
A case is considered to have a large DFBETAS value if its absolute value exceeds $2/\sqrt{n}$.

To compute DFBETAS with `R`, use `dfbetas()`.

Example

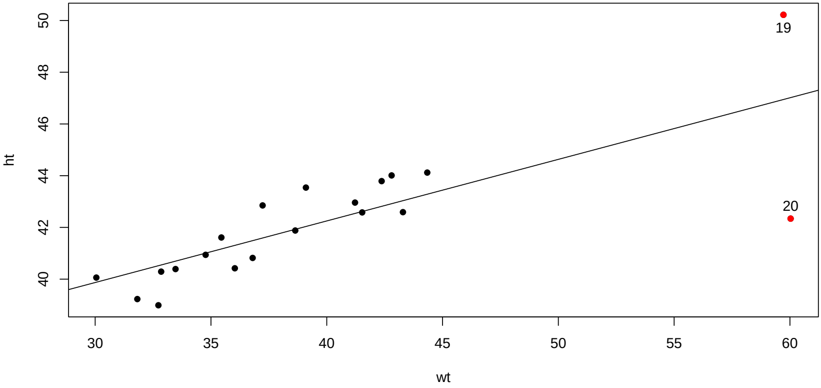
Example

The data set `Kinder` contains the height in inches and weight in pounds of 20 children from a kindergarten class.



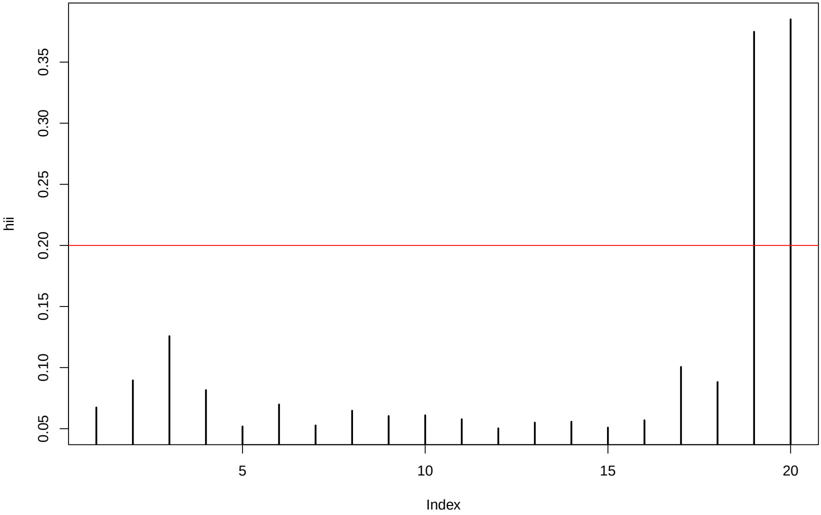
Example

```
plot(wt, ht, pch=16);
points(wt[c(19,20)], ht[c(19,20)], pch=16, col='red')
```



Example

```r
plot(hii, type = 'h', lwd=2)
abline(h=1/5, col='red')
```



Example

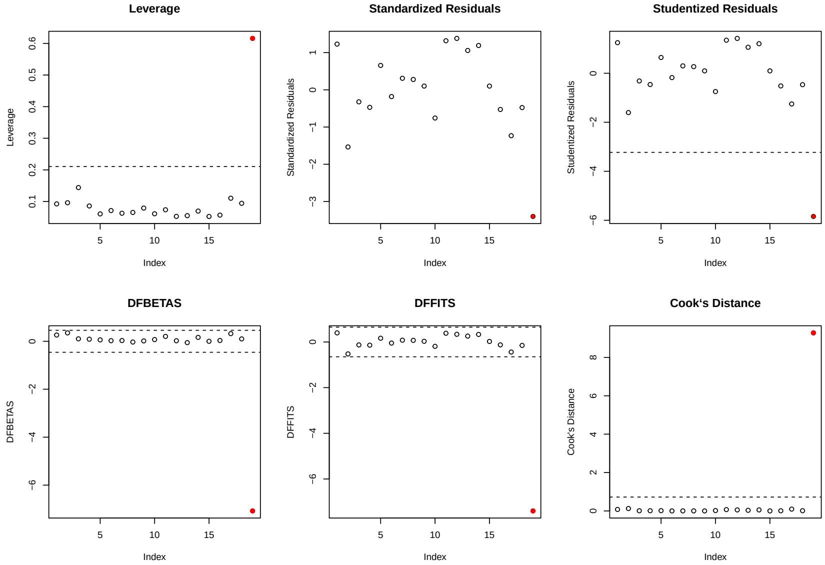
Observations 19 and 20 are removed from consideration and the new model is stored in `modk`.

```r
modk <- lm(ht[-c(19,20)]~wt[-c(19,20)])
```

We now consider a model without observation 19 and store the result in `modk19`.

The 19th observation now (previously 20) corresponds to the ‘overweight’ child.

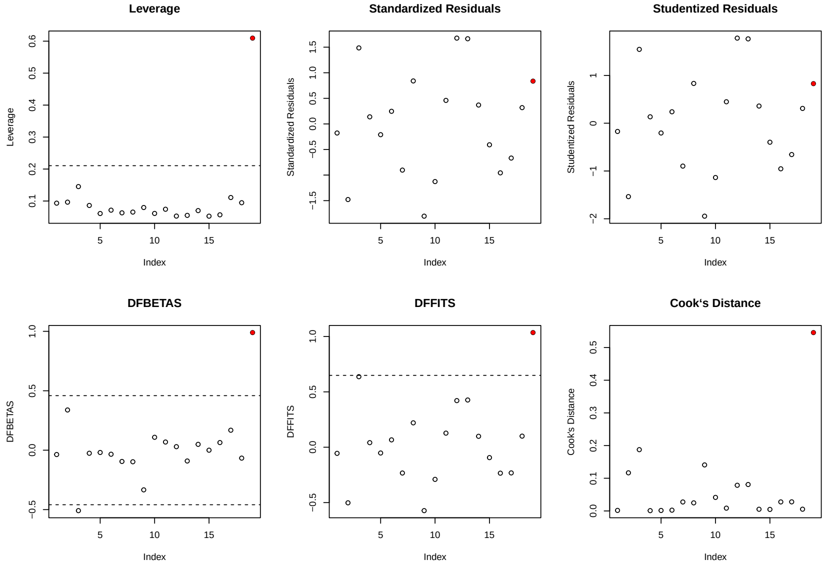
Example



Leverage
Leverage
0.1
0.2
0.3
0.4
0.5
0.6
5
10
15
Index
Standardized Residuals
Standardized Residuals
−3
−2
−1
0
1
5
10
15
Index
Studentized Residuals
Studentized Residuals
−6
−4
−2
0
5
10
15
Index
DFBETAS
DFBETAS
−6
−4
−2
0
5
10
15
Index
DFFITS
DFFITS
−6
−4
−2
0
5
10
15
Index
Cook‘s Distance
Cook's Distance
0
2
4
6
8
5
10
15
Index

Example

Next, we consider a model without observation 20:



Leverage

Standardized Residuals

Studentized Residuals

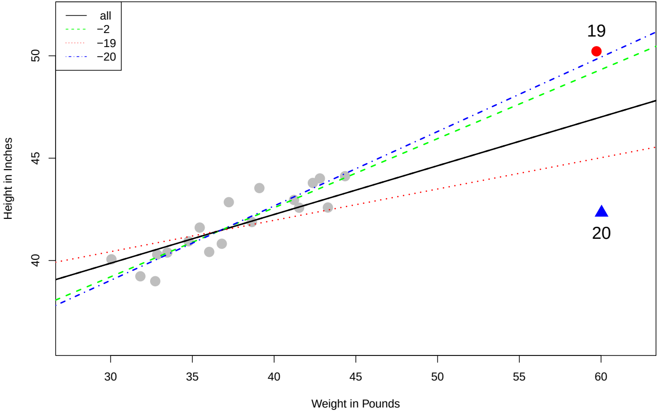
DFBETAS

DFFITS

Cook's Distance

Example

Finally, we plot the three regressions model we have fitted and compare the results.



Example

When all 20 cases are included in the regression, cases 19 (solid circle) and 20 (solid triangle) have large leverage values; however, if case 20 is omitted, case 19 still has a large leverage value, yet it is not very influential.

Consider the differences between the lines `modk20` (dot-dash, blue, case 20 omitted) and `modk` (dash, green, where cases 19 and 20 are omitted). There is very little difference between them.

On the other hand, if case 19 (solid circle) is omitted, the resulting regression `modk19` (dotted, red) is substantially different from modk. In other words, case 20 has high leverage and is influential when case 19 is omitted.

Other Diagnostic Tools

Other Diagnostic Tools

We introduce other graphical diagnostic tools available on the `car` package (Companion to Applied Regression) by Fox, Weisberg, and Price.

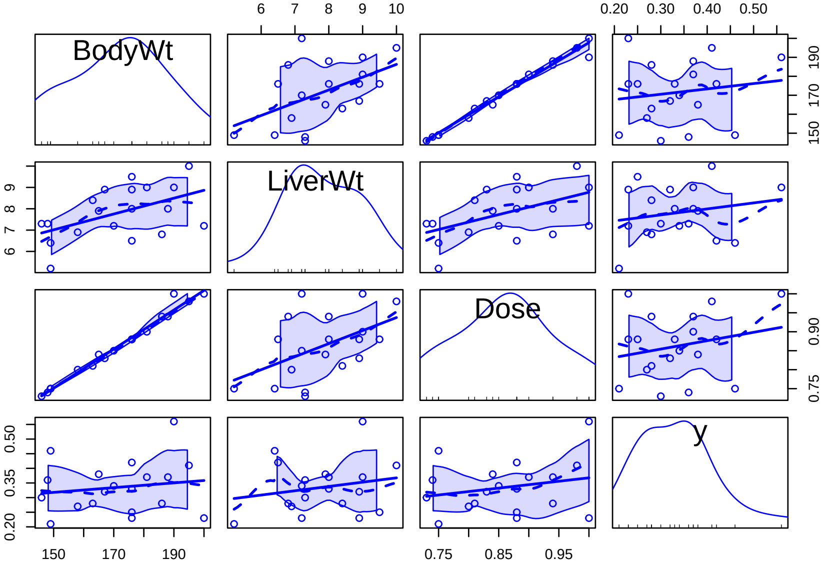
We will consider the data set `rat` from the `alr4` package. This data comes from 'an experiment in which rats were injected with a dose of a drug approximately proportional to body weight.

At the end of the experiment, the animal's liver was weighed, and the fraction of the drug recovered in the liver was recorded.

The experimenter expected the response to be independent of the predictors.'

Other Diagnostic Tools

```
scatterplotMatrix(rat)
```



Other Diagnostic Tools

Fit a full additive model

```
m1 <- lm(y ~ BodyWt + LiverWt + Dose, rat)
S(m1)
```

```
## Call: lm(formula = y ~ BodyWt + LiverWt + Dose, data = rat)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.265922   0.194585   1.367   0.1919
## BodyWt      -0.021246   0.007974  -2.664   0.0177 *
## LiverWt      0.014298   0.017217   0.830   0.4193
## Dose         4.178111   1.522625   2.744   0.0151 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.07729 on 15 degrees of freedom
## Multiple R-squared: 0.3639
## F-statistic:  2.86 on 3 and 15 DF,  p-value: 0.07197
##    AIC    BIC
## -37.86 -33.14
```

Other Diagnostic Tools

Drop LiverWt from the model

```
m2 <- update(m1, ~ . - LiverWt)
S(m2)
```

```
## Call: lm(formula = y ~ BodyWt + Dose, data = rat)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.285517   0.191267   1.493   0.1550
## BodyWt      -0.020444   0.007838  -2.608   0.0190 *
## Dose         4.125330   1.506472   2.738   0.0146 *
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.07654 on 16 degrees of freedom
## Multiple R-squared: 0.3347
## F-statistic: 4.024 on 2 and 16 DF,  p-value: 0.0384
##    AIC    BIC
## -39.00 -35.23
```

Other Diagnostic Tools

Going back to the scatterplot matrix, we see that `Dose` and `BodyWt` are almost perfectly aligned, so they carry the same information.

Let's try to drop one of them from the model

```r
m3 <- update(m2, ~ . - Dose, rat)
S(m3)
```

```
## Call: lm(formula = y ~ BodyWt, data = rat)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.1962346  0.2215825   0.886    0.388
## BodyWt      0.0008105  0.0012862   0.630    0.537
##
## Residual standard deviation: 0.08999 on 17 degrees of freedom
## Multiple R-squared: 0.02283
## F-statistic: 0.3971 on 1 and 17 DF,  p-value: 0.537
##    AIC    BIC
## -33.70 -30.87
```

Other Diagnostic Tools

```r
m4 <- update(m2, ~ . - BodyWt, rat)
S(m4)
```

```
## Call: lm(formula = y ~ Dose, data = rat)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1330     0.2109   0.631    0.537
## Dose          0.2346     0.2435   0.963    0.349
##
## Residual standard deviation: 0.08864 on 17 degrees of freedom
## Multiple R-squared: 0.05178
## F-statistic: 0.9283 on 1 and 17 DF,  p-value: 0.3488
##    AIC    BIC
## -34.27 -31.44
```

It seems that neither Dose nor BodyWt on its own is associated with the response, but when they are together, the story is different.

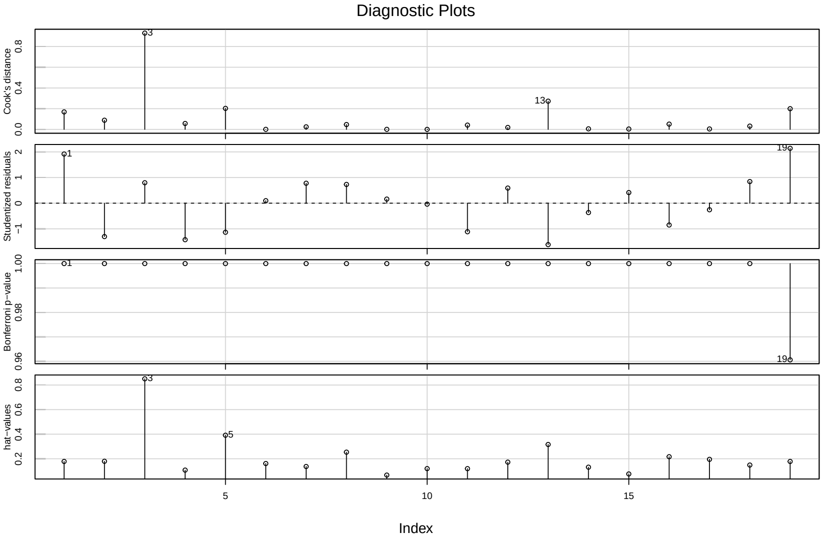
Other Diagnostic Tools: Influence plots

The function `influenceIndexPlot` plots

- Cook's distance,
- Studentized residuals,
- Bonferroni significance levels to testing each observation in turn to be an outlier, and
- Leverage values, or a subset of these, versus observation number

Other Diagnostic Tools: Influence plots

```
influenceIndexPlot(m1)
```



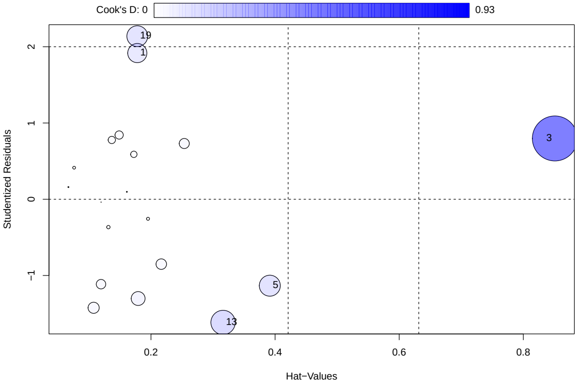
Other Diagnostic Tools: Influence plots

The function `influencePlot` graphs Studentized residuals against leverage values, and includes Cook's distance as the radius of circles.

It also gives information about the flagged points in the plot.

Other Diagnostic Tools: Influence plots

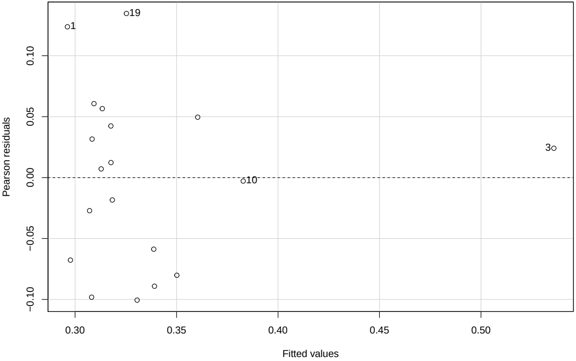
```r
influencePlot(m1)
```



```
##         StudRes       Hat     CookD
## 1    1.9170719 0.1779827 0.1688268
## 3    0.7972915 0.8509146 0.9296160
## 5   -1.1337306 0.3915382 0.2029162
## 13  -1.6160514 0.3161834 0.2726019
## 19   2.1388332 0.1779618 0.1999403
```

Other Diagnostic Tools

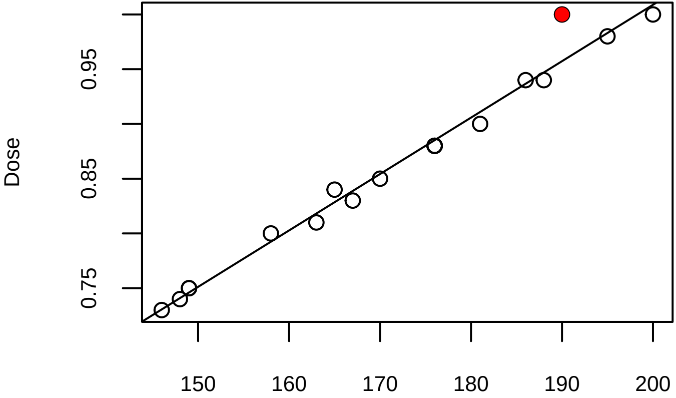
```
residualPlot(m1, id =list(method = list("x", "y"), n=2),
             quadratic = FALSE)
```



Other Diagnostic Tools

The previous residual plot shows that case number 3 has a very large fitted value, compared to the rest of the points.

The cause may be that rat number 3 got a larger dose than it should have received.



Other Diagnostic Tools

What to do next in a case like this depends on the problem.

One alternative that is sometimes advocated is to fit the model without the suspicious point.

However, this throws doubts on the whole experiment, and perhaps there is a need to collect further data, with dose determined with more precision or using a different method.

Other Diagnostic Tools

```r
m1b <- lm(y ~ BodyWt + LiverWt + Dose, rat[-3,])
S(m1b)
```

```
## Call: lm(formula = y ~ BodyWt + LiverWt + Dose, data =
##          rat[-3, ])
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.311427   0.205094   1.518    0.151
## BodyWt      -0.007783   0.018717  -0.416    0.684
## LiverWt      0.008989   0.018659   0.482    0.637
## Dose         1.484877   3.713064   0.400    0.695
##
## Residual standard deviation: 0.07825 on 14 degrees of freedom
## Multiple R-squared: 0.02106
## F-statistic: 0.1004 on 3 and 14 DF,  p-value: 0.9585
##    AIC    BIC
## -35.17 -30.71
```

Other Diagnostic Tools: `residualPlots`

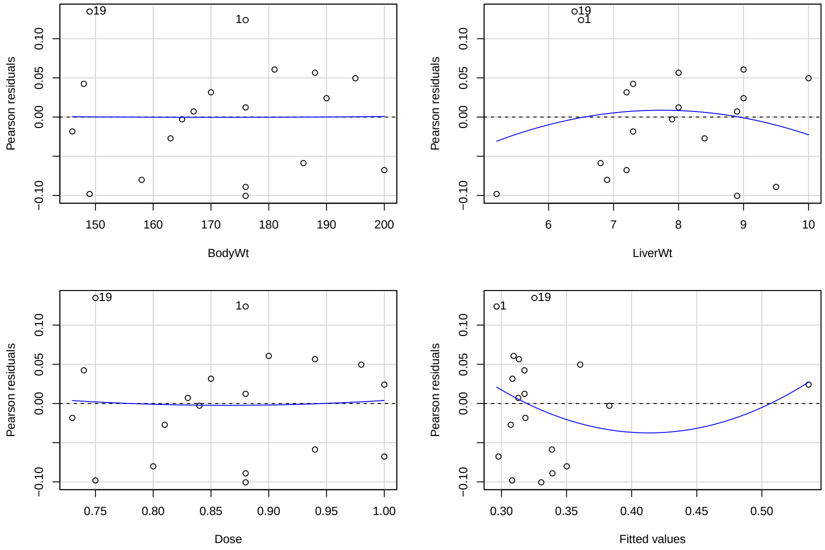
Let's go back to residual plots using the `car` functions.

The command `residualPlots()` produces plots of residuals against all regressors and also against fitted values.

It also, by default, graphs quadratic regression terms and does curvature tests in all cases.

Other Diagnostic Tools: `residualPlots`

```r
residualPlots(m1, id=TRUE)
```



```
##                Test stat Pr(>|Test stat|)
```

Other Diagnostic Tools:residualPlots

```
residualPlots(m1, id=TRUE, plot=FALSE)
```

```
##            Test stat Pr(>|Test stat|)
## BodyWt        0.0185           0.9855
## LiverWt      -0.5760           0.5737
## Dose          0.1444           0.8873
## Tukey test    0.9320           0.3513
```

Other Diagnostic Tools: `residualPlots`

What we see at the bottom are curvature tests.

These tests help decide whether the plots show the need for higher-order terms in the regression.

Suppose we have a plot of residuals against a regressor or a combination of regressors. The test for curvature refits the original model with an additional term for the square of the regressor.

The test for curvature is based on the t-test for the coefficient of the quadratic term under the null hypothesis that it is zero.

If the plot is against fitted values, which depend on the estimated parameters, the t statistic should be compared with the standard normal distribution. This is known as Tukey's test.

None of the tests in the example are significant.

V37: Multicollinearity

Collinearity

Two regressors X_1 and X_2 are **collinear** if they are linearly dependent, i.e. there exist constants c_1, c_2 and c_3, not all equal to zero, such that

$$c_1 X_1 + c_2 X_2 = c_3 \tag{1}$$

Example of Collinearity

```
collin_data = function(num_samples = 100) {
  x1 = rnorm(n = num_samples, mean = 80, sd = 10)
  x2 = rnorm(n = num_samples, mean = 70, sd = 5)
  x3 = 2 * x1 + 4 * x2 + 3
  y = 3 + x1 + x2 + rnorm(n = num_samples,
                          mean = 0, sd = 1)
  data.frame(y, x1, x2, x3)
}
set.seed(123)
collin_exmpl <- collin_data()
```

Collinearity

```r
collin.lm <- lm(y ~ x1 + x2 + x3, data = collin_exmpl)
S(collin.lm)
```

```
## Call: lm(formula = y ~ x1 + x2 + x3, data = collin_exmpl)
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.86708    1.65405   2.338   0.0214 *
## x1           0.98668    0.01049  94.087   <2e-16 ***
## x2           1.00476    0.01980  50.748   <2e-16 ***
## x3                NA         NA      NA       NA
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.9513 on 97 degrees of freedom
## Multiple R-squared: 0.9912
## F-statistic:  5491 on 2 and 97 DF,  p-value: < 2.2e-16
##    AIC    BIC
## 278.76 289.18
```

Collinearity

We see that the third variable has been excluded from the regression.

In this case, the design matrix is

```
X = cbind(1, as.matrix(collin_exmpl[,-1]))
```

and if we try to invert $\mathbf{X}'\mathbf{X}$

```
solve(t(X) %*% X)
```

we get a warning that says

```
Error in solve.default(t(X) %*% X) : system is
computationally singular: reciprocal condition number
= 1.01841e-17 ...
```

When this happens, we have exact collinearity.

Collinearity

The fitted model was $y \sim x1 + x2$ and excluded one of the variables, in this case, x3, but observe that other models would accomplish exactly the same fit

```
fit1 = lm(y ~ x1 + x2, data = collin_exmpl)
fit2 = lm(y ~ x1 + x3, data = collin_exmpl)
fit3 = lm(y ~ x2 + x3, data = collin_exmpl)
```

The fitted values for these three models are exactly the same:

```
all.equal(fitted(fit1), fitted(fit2))
```

```
## [1] TRUE
```

```
all.equal(fitted(fit2), fitted(fit3))
```

```
## [1] TRUE
```

Collinearity

But the estimated coefficients are not

```
coef(fit1); coef(fit2); coef(fit3)
```

```
## (Intercept)          x1          x2
##   3.8670796   0.9866828   1.0047623

## (Intercept)          x1          x3
##   3.1135079   0.4843017   0.2511906

## (Intercept)          x2          x3
##   2.3870554  -0.9686034   0.4933414
```

However, only the first model explains the relationship between the variables.

The other models are able to predict correctly, but the coefficients are meaningless.

Collinearity

Approximate collinearity happens if equation (1) is approximately true

Collinearity between X_1 and X_2 is measured by the square of their sample correlation r_{12}^2.

Exact collinearity corresponds to $r_{12}^2 = 1$ while non-collinearity corresponds to $r_{12}^2 = 0$.

If r_{12}^2 is close to 1, we have approximate collinearity.

Multicollinearity

Multicollinearity

For $p > 2$ regressors, **approximate collinearity** happens if there are constants $c_0, c_1, \ldots, c_p$ not all equal to zero so that

$$c_1 X_1 + c_2 X_2 + \cdots + c_p X_p \approx c_0$$

Observe that if $c_i \neq 0$, then we can write X_i approximately as a linear combination of the other variables.

In this case, instead of the squared correlation, variable X_i is regressed on the other X's, and the R^2 for this regression is considered as the **multiple correlation coefficient** between X_i and the other variables and denoted by R_i^2.

If the largest R_i^2 is close to 1, we have approximate collinearity.

Multicollinearity

When a set of predictors is exactly collinear, one or more predictors must be deleted to be able to estimate the coefficients for the model.

Since the information in the deleted predictor is contained in the other regressors, no information is lost in this process. However, the interpretation of the parameters may be different or more complex.

When approximate collinearity is present, the usual remedy is again to delete variables, with loss of information expected to be small.

The tricky part may be deciding which variable(s) to delete.

Multicollinearity

One important effect of a high correlation between regressors is the increased variance of the estimates.

The sampling variance of $\hat{\beta}_j$ is

$$Var(\hat{\beta}_j) = \frac{1}{1 - R_j^2} \frac{\sigma^2}{(n-1)S_j^2}$$

where

$$S_j^2 = \frac{1}{n-1} \sum_i (x_{ij} - \bar{x}_{\bullet j})^2$$

is the sample variance of X_j.

The term $1/(1 - R_j^2)$, known as the **variance inflation factor (VIF)**, indicates the effect of collinearity on the variance of $\hat{\beta}_j$.

Multicollinearity: Simulated Example

This simulated example is from S. Weisberg *Applied Linear Regression*, Wiley.

We consider two models of the form

$$\begin{aligned} Y &= 1 + X_1 + X_2 + 0 \cdot X_3 + 0 \cdot X_4 + \epsilon \\ &= 1 + X_1 + X_2 + \epsilon \end{aligned}$$

where $\epsilon \sim N(0, 1)$.

In the first model, $\mathbf{X} = (X_1, X_2, X_3, X_4)$ are independent normal random variables while in the second case, the covariance matrix is

$$\Sigma = \begin{pmatrix} 1 & 0 & .95 & 0 \\ 0 & 1 & 0 & -.95 \\ .95 & 0 & 1 & 0 \\ 0 & -.95 & 0 & 1 \end{pmatrix}$$

so that X_1 and X_3 are highly positively correlated while X_2 and X_4 are highly negatively correlated.

Multicollinearity: Simulated Example

We fit linear models in each case

```r
library(mvtnorm)
sigma1 <- diag(4); # Independent variables
sigma2 <- sigma1
sigma2[3,1] <- sigma2[1,3] <- 0.95 # correlation
sigma2[4,2] <- sigma2[2,4] <- -0.95 # correlation
# Sample from independent distribution
sample1 <- data.frame(rmvnorm(100, sigma = sigma1))
colnames(sample1) <- c('X1','X2','X3','X4')
# Sample from correlated distribution
sample2 <- rmvnorm(100, sigma = sigma2)
colnames(sample2) <- c('X1','X2','X3','X4')
# Simulated models
y1 <- 1 + sample1[,1] + sample1[,2] + rnorm(100)
y2 <- 1 + sample2[,1] + sample2[,2] + rnorm(100)
```

Multicollinearity: Simulated Example

First Model

```
col1 <- lm(y1 ~ X1 + X2 + X3 + X4, data = sample1 )
summary(col1)
```

```
##
## Call:
## lm(formula = y1 ~ X1 + X2 + X3 + X4, data = sample1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1377 -0.6255 -0.0358  0.4447  2.8748
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.96891    0.10085   9.607 1.14e-15 ***
## X1           0.93469    0.10538   8.870 4.30e-14 ***
## X2           0.72390    0.11863   6.102 2.26e-08 ***
## X3          -0.01678    0.09660  -0.174    0.862
## X4          -0.07702    0.09004  -0.855    0.395
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9878 on 95 degrees of freedom
## Multiple R-squared:  0.5539, Adjusted R-squared:  0.5351
## F-statistic: 29.49 on 4 and 95 DF,  p-value: 6.09e-16
```

Multicollinearity: Simulated Example

First Model

```
set.seed(92837)
sample1 <- data.frame(rmvnorm(100, sigma = sigma1)); colnames(sample1) <- c('X1','X2','X3','X4')
y1 <- 1 + sample1[,1] + sample1[,2] + rnorm(100)
col1b <- lm(y1 ~ X1 + X2 + X3 + X4, data = sample1); summary(col1b)
```

```
##
## Call:
## lm(formula = y1 ~ X1 + X2 + X3 + X4, data = sample1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6413 -0.5992 -0.0102  0.7065  3.1775
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.94780    0.11096   8.542 2.15e-13 ***
## X1           0.96616    0.11386   8.486 2.82e-13 ***
## X2           0.92582    0.12090   7.658 1.57e-11 ***
## X3          -0.11046    0.10920  -1.011    0.314
## X4           0.05029    0.10927   0.460    0.646
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.095 on 95 degrees of freedom
## Multiple R-squared:  0.5163, Adjusted R-squared:  0.496
## F-statistic: 25.35 on 4 and 95 DF,  p-value: 2.646e-14
```

Multicollinearity: Simulated Example

Second Model

```
set.seed(7364)
sample2 <- data.frame(rmvnorm(100, sigma = sigma2)); colnames(sample2) <- c('X1','X2','X3','X4')
y2 <- 1 + sample2[,1] + sample2[,2] + rnorm(100)
col2a <- lm(y2 ~ X1 + X2 + X3 + X4, data = sample2); summary(col2a)
```

```
##
## Call:
## lm(formula = y2 ~ X1 + X2 + X3 + X4, data = sample2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4231 -0.7921  0.1726  0.8837  2.2925
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0315     0.1111   9.288 5.48e-15 ***
## X1            0.9071     0.2988   3.035   0.0031 **
## X2            0.6506     0.3526   1.846   0.0681 .
## X3            0.1342     0.3084   0.435   0.6645
## X4           -0.1462     0.3597  -0.406   0.6854
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.083 on 95 degrees of freedom
## Multiple R-squared:  0.5786, Adjusted R-squared:  0.5609
## F-statistic: 32.62 on 4 and 95 DF,  p-value: < 2.2e-16
```

Multicollinearity: Simulated Example

Second Model

```r
set.seed(574597)
sample2 <- data.frame(rmvnorm(100, sigma = sigma2)); colnames(sample2) <- c('X1','X2','X3','X4')
y2 <- 1 + sample2[,1] + sample2[,2] + rnorm(100)
col2b <- lm(y2 ~ X1 + X2 + X3 + X4, data = sample2); summary(col2b)
```

```
## 
## Call:
## lm(formula = y2 ~ X1 + X2 + X3 + X4, data = sample2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.15898 -0.63164  0.08673  0.63820  2.40883 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   1.0798     0.1007  10.728   <2e-16 ***
## X1            0.5737     0.3641   1.576    0.118    
## X2            0.4461     0.3069   1.454    0.149    
## X3            0.3938     0.3682   1.070    0.287    
## X4           -0.3111     0.3006  -1.035    0.303    
## ---
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.002 on 95 degrees of freedom
## Multiple R-squared:  0.5494, Adjusted R-squared:  0.5304 
## F-statistic: 28.95 on 4 and 95 DF,  p-value: 9.757e-16
```

Multicollinearity: Simulated Example

Second Model

```r
set.seed(16299125)
sample2 <- data.frame(rmvnorm(100, sigma = sigma2)); colnames(sample2) <- c('X1','X2','X3','X4')
y2 <- 1 + sample2[,1] + sample2[,2] + rnorm(100)
col2c <- lm(y2 ~ X1 + X2 + X3 + X4, data = sample2); summary(col2c)
```

```
##
## Call:
## lm(formula = y2 ~ X1 + X2 + X3 + X4, data = sample2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73719 -0.60276  0.03033  0.62877  2.25573
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0413     0.1069   9.744 5.81e-16 ***
## X1            0.4285     0.3333   1.286 0.201684
## X2            1.5355     0.4166   3.686 0.000379 ***
## X3            0.5874     0.3397   1.729 0.087035 .
## X4            0.6302     0.4048   1.557 0.122821
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.06 on 95 degrees of freedom
## Multiple R-squared:  0.6215, Adjusted R-squared:  0.6056
## F-statistic:    39 on 4 and 95 DF,  p-value: < 2.2e-16
```

Multicollinearity: Simulated Example

```
round(vif(col2a),3); round(vif(col2b),3); round(vif(col2c),3);
```

```
##    X1    X2    X3    X4 
## 7.140 9.568 7.279 9.416
```

```
##     X1     X2     X3     X4 
## 12.010  8.053 12.143  8.051
```

```
##     X1     X2     X3     X4 
## 12.691 12.060 12.753 12.090
```

Multicollinearity

For the `rat` example

```
vif(m1); vif(m1b)
```

```
##    BodyWt   LiverWt      Dose
## 52.101917  1.335679 51.427154
```

```
##     BodyWt    LiverWt       Dose
## 259.449422   1.445674 253.199751
```

V38: Polynomial Regression and Categorical Predictors

Polynomial Regression

In polynomial regression, the regressors associated with a predictor X_i form a polynomial in X_i with degree d.

In the case of only one regressor, the model is

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_d X^d + \epsilon$$

A particular case that occurs frequently is quadratic regression, where the polynomial has degree 2.

We have already seen that the `residualPlots()` function performs a curvature test to determine whether second-order terms should be included in the model.

Polynomial Regression

To illustrate polynomial regression, we follow an example in Fox and Weisberg, *An R Companion to Applied Regression*, SAGE (2019).

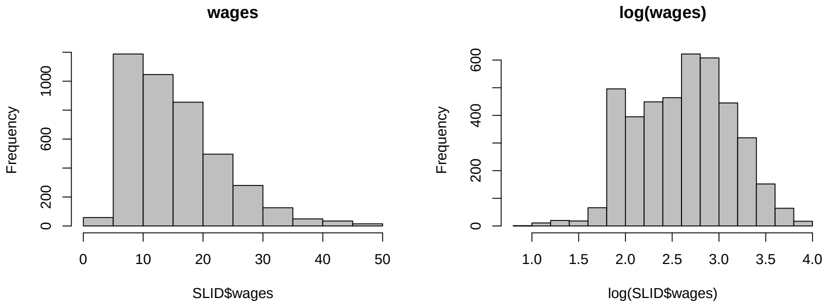
We use the data set SLID in the carData package '*which contains data for the province of Ontario from the 1994 wave of the Survey of Labour and Income Dynamics, a panel study of the Canadian labor force conducted by Statistics Canada.*'

```
str(SLID)
```

```
## 'data.frame':    7425 obs. of  5 variables:
##  $ wages    : num  10.6 11 NA 17.8 NA ...
##  $ education: num  15 13.2 16 14 8 16 12 14.5 15 10 ...
##  $ age      : int  40 19 49 46 71 50 70 42 31 56 ...
##  $ sex      : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 1 1 1 2 1 ...
##  $ language : Factor w/ 3 levels "English","French",..: 1 1 3 3 1 1 1 1 1 1
```

Polynomial Regression

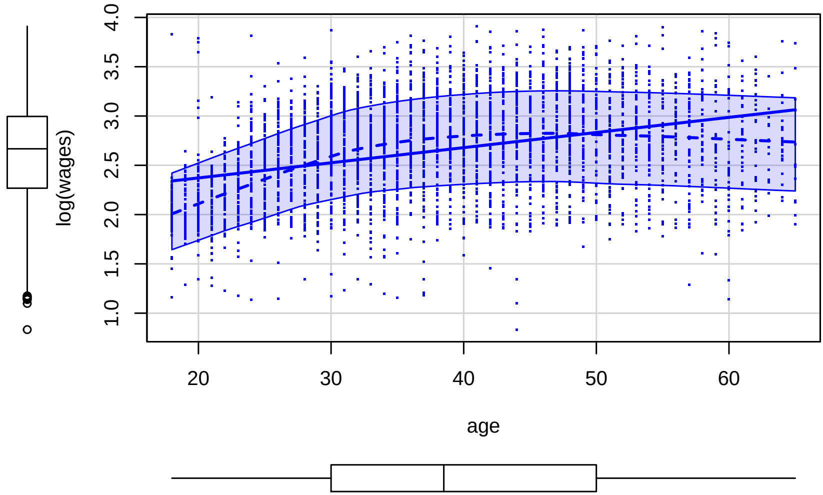
```r
par(mfrow = c(1,2))
hist(SLID$wages, col='gray75', main='wages')
hist(log(SLID$wages), col='gray75', main='log(wages)')
```



We will consider `log(wages)` as the output variable and want to regress it on age

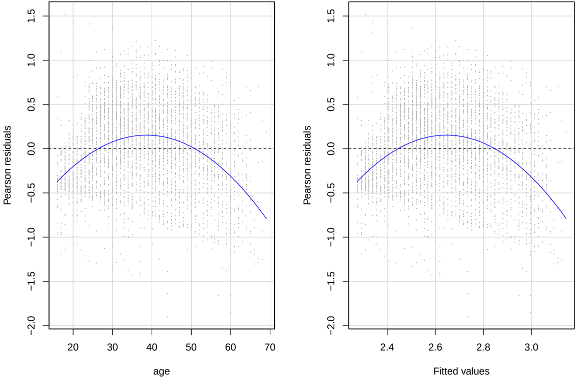
Polynomial Regression

```
slid.m <- lm(log(wages)~age, data = SLID)
scatterplot(log(wages)~age, data = SLID, pch='.',
            subset = age>= 18 & age <= 65)
```



Polynomial Regression

```
residualPlots(slid.m, pch ='.', col=gray(0.75))
```



```
##             Test stat Pr(>|Test stat|)
## age           -24.389        < 2.2e-16 ***
## Tukey test    -24.389        < 2.2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Polynomial Regression

The quadratic model can be fit in at least three equivalent ways:

Adding the square of the regressor with the function `I()`,

```
slid.m1 <- lm(log(wages)~age + I(age^2), data = SLID)
brief(slid.m1)
```

```
##             (Intercept)     age  I(age^2)
## Estimate        0.6394 0.09538 -1.02e-03
## Std. Error      0.0603 0.00329  4.19e-05
##
##  Residual SD = 0.433 on 4144 df, R-squared = 0.262
```

Polynomial Regression

Using the `poly()` function with the option `raw = TRUE`,

```r
slid.m2 <- lm(log(wages) ~ poly(age,2,raw = TRUE), data=SLID)
brief(slid.m2)
```

```
##            (Intercept) poly(age, 2, raw = TRUE)1
## Estimate        0.6394                   0.09538
## Std. Error      0.0603                   0.00329
##            poly(age, 2, raw = TRUE)2
## Estimate                   -1.02e-03
## Std. Error                  4.19e-05
##
##  Residual SD = 0.433 on 4144 df, R-squared = 0.262
```

Polynomial Regression

Using the poly() function with no raw option

```r
slid.m3 <- lm(log(wages) ~ poly(age,2), data=SLID)
brief(slid.m3)
```

```
##            (Intercept) poly(age, 2)1 poly(age, 2)2
## Estimate        2.5380         -3.13         -29.3
## Std. Error      0.0111          1.43           1.2
##
##  Residual SD = 0.433 on 4144 df, R-squared = 0.262
```

Polynomial Regression

The first and second options give the same coefficients. The third fits orthogonal polynomials, and the coefficients are different but fitted values are the same.

```r
Anova(slid.m1)
```

```
## Anova Table (Type II tests)
##
## Response: log(wages)
##           Sum Sq   Df F value    Pr(>F)
## age       158.14    1  842.39 < 2.2e-16 ***
## I(age^2)  111.66    1  594.81 < 2.2e-16 ***
## Residuals 777.95 4144
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Polynomial Regression

```
Anova(slid.m2)
```

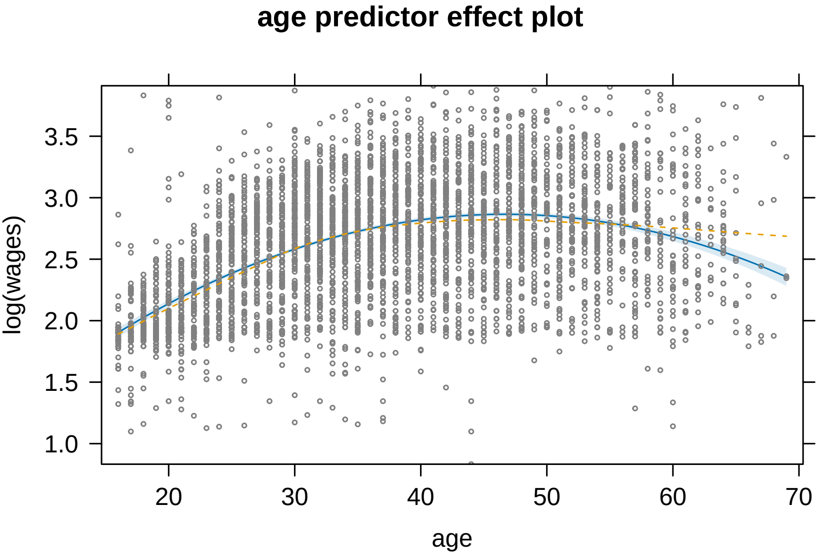
```
## Anova Table (Type II tests)
##
## Response: log(wages)
##                          Sum Sq   Df F value    Pr(>F)
## poly(age, 2, raw = TRUE) 275.95    2  734.97 < 2.2e-16 ***
## Residuals                777.95 4144
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(slid.m3)
```

```
## Anova Table (Type II tests)
##
## Response: log(wages)
##              Sum Sq   Df F value    Pr(>F)
## poly(age, 2) 275.95    2  734.97 < 2.2e-16 ***
## Residuals    777.95 4144
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

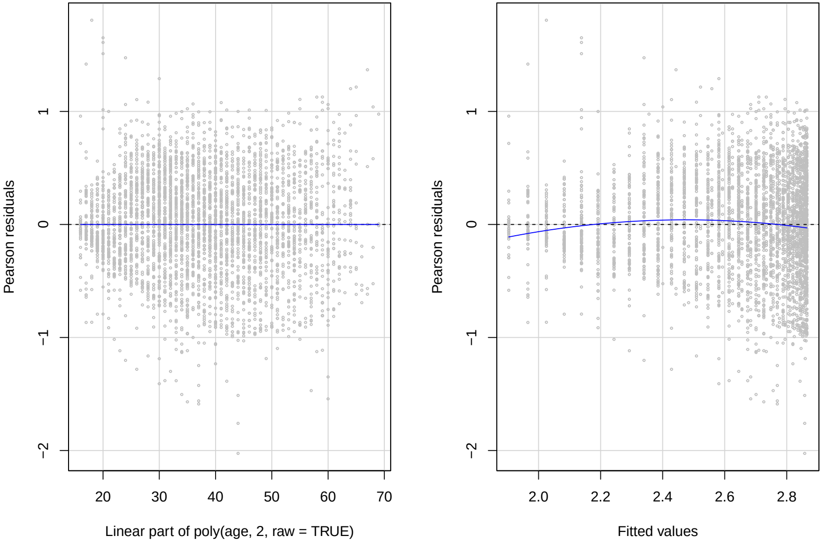
Polynomial Regression

```
plot(predictorEffects(slid.m2, residuals = TRUE),
     partial.residuals = list(cex=0.35, col=gray(0.5), lty = 2))
```



Polynomial Regression

```
residualPlots(slid.m2, cex=0.35, col=gray(0.75), tests = FALSE)
```



Qualitative Predictors

Qualitative Predictors

Up to this point, only quantitative predictor variables have been used in regression models.

Regression using quantitative variables can be generalized to qualitative variables with the use of **dummy** variables.

A dummy variable is any variable in a regression model that takes on a finite number of values to identify different categories of a nominal variable.

Provided the regression model has an intercept, one must define $k-1$ dummy variables with values 0 and 1 to define a qualitative variable with k categories.

Qualitative Predictors

The simplest situation where dummy variables might be used in a regression model is when the qualitative predictor has only two levels.

The regression model for a single quantitative predictor (X) and a dummy variable (D) is written

$$Y = \beta_0 + \beta_1 X + \beta_2 D + \beta_3 DX + \epsilon \tag{2}$$

where

$$D = \begin{cases} 0 & \text{for the first level} \\ 1 & \text{for the second level} \end{cases}$$

The model in (2) when D has two levels will yield one of four possible scenarios, as shown in the figure on the next slide.

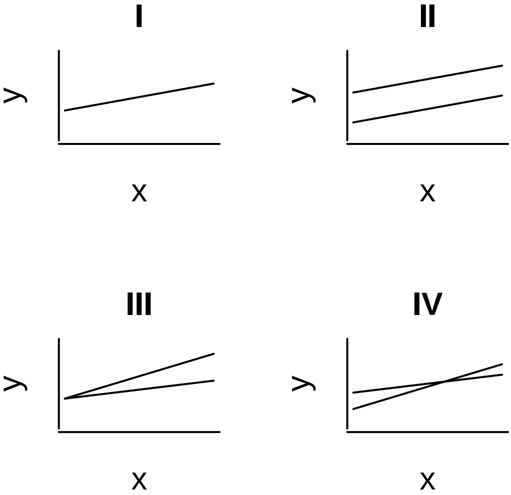


Figure 1: Effect of categorical variables in simple linear regression

Qualitative Predictors

This type of model requires the user to answer three basic questions:

1. Are the lines the same?
2. Are the slopes the same?
3. Are the intercepts the same?

To address (1), the null hypothesis $H_0 : \beta_2 = \beta_3 = 0$ must be tested.

To answer (2), the null hypothesis $H_0 : \beta_3 = 0$ must be tested.

To answer (3), the null hypothesis $H_0 : \beta_2 = 0$ must be tested.

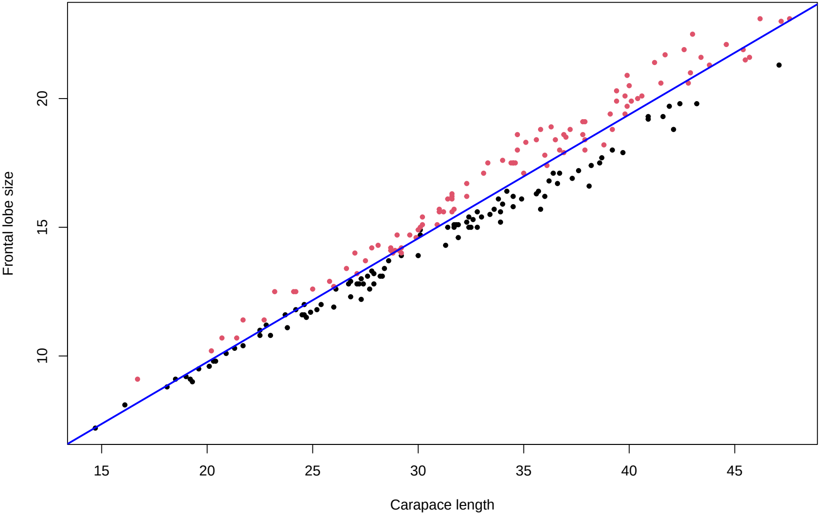
Example: Crabs

```r
library(MASS); attach(crabs);lmSimple <- lm(FL~CL); S(lmSimple)
```

```
## Call: lm(formula = FL ~ CL)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.15316    0.23477   0.652    0.515
## CL           0.48060    0.00714  67.313   <2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.717 on 198 degrees of freedom
## Multiple R-squared: 0.9581
## F-statistic:  4531 on 1 and 198 DF,  p-value: < 2.2e-16
##   AIC   BIC
## 438.5 448.4
```

Example: Crabs

```
plot(CL,FL, pch=20, xlab='Carapace length', ylab='Frontal lobe size', col=sp)
abline(lmSimple, lw=2, col='blue')
```



Example: Crabs

This corresponds to fitting the simple model

$$FL = \beta_0 + \beta_1 CL + \epsilon$$

We now consider a second model including a dummy variable D for species, which gives

$$FL = \beta_0 + \beta_1 CL + \beta_2 D + \beta_3 CL \cdot D + \epsilon \tag{3}$$

```
fsp <- as.factor(sp)
contrasts(fsp)
```

```
##   O
## B 0
## O 1
```

The factor `fsp` has values 0 for the blue (`B`) and 1 for the orange (`O`) species.

Example: Crabs

We now fit the complete model (3).

```r
lmComplete <-  lm(FL~CL+fsp+CL:fsp)
S(lmComplete)
```

```
## Call: lm(formula = FL ~ CL + fsp + CL:fsp)
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.971315   0.184593   5.262 3.72e-07 ***
## CL           0.435315   0.005987  72.711  < 2e-16 ***
## fspO        -0.209274   0.281608  -0.743    0.458    
## CL:fspO      0.043354   0.008554   5.068 9.25e-07 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 0.4112 on 196 degrees of freedom
## Multiple R-squared: 0.9864
## F-statistic:  4728 on 3 and 196 DF,  p-value: < 2.2e-16
##    AIC    BIC
## 218.05 234.55
```

Example: Crabs

To compare these models, we use an anova with the two models

```
anova(lmSimple,lmComplete)
```

```
## Analysis of Variance Table
##
## Model 1: FL ~ CL
## Model 2: FL ~ CL + fsp + CL:fsp
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    198 101.793
## 2    196  33.139  2    68.654 203.03 < 2.2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example: Crabs

The small *p*-value says that at least one of the two parameters β_2, β_3 is not zero.

To see if the lines have different slopes, we want to test

$$H_0 : \beta_3 = 0 \qquad vs. \qquad H_1 : \beta_3 \neq 0.$$

Looking at the `summary()` for the `lmComplete` model

Example: Crabs

```r
summary(lmComplete)
```

```
##
## Call:
## lm(formula = FL ~ CL + fsp + CL:fsp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.13437 -0.23131 -0.01476  0.23612  1.22817
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.971315   0.184593   5.262 3.72e-07 ***
## CL           0.435315   0.005987  72.711  < 2e-16 ***
## fspO        -0.209274   0.281608  -0.743    0.458
## CL:fspO      0.043354   0.008554   5.068 9.25e-07 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4112 on 196 degrees of freedom
## Multiple R-squared:  0.9864, Adjusted R-squared:  0.9862
## F-statistic:  4728 on 3 and 196 DF,  p-value: < 2.2e-16
```

Example: Crabs

We see that the term `CL:fsp0` has a small *p*-value and therefore is statistically significant at the usual levels.

This means that the slopes are different: when $D = 0$ (blue species) the slope is $\beta_1 = 0.435315$ and when $D = 1$ (orange species) the slope is $\beta_1 + \beta_3 = 0.435315 + 0.043354 = 0.478669$.

Finally, the variable `fsp0` is the dummy variable, and the *p* value associated with it is large (0.458), which means it is not significant, and therefore there is no difference in the intercepts. The final model, then, is an intermediate model:

Example: Crabs

```r
lmInter <-  lm(FL~CL+CL:fsp)
summary(lmInter)
```

```
##
## Call:
## lm(formula = FL ~ CL + CL:fsp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1232 -0.2509 -0.0102  0.2441  1.2255
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.881395   0.139246    6.33 1.62e-09 ***
## CL          0.438158   0.004600   95.26  < 2e-16 ***
## CL:fsp0     0.037147   0.001843   20.16  < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4107 on 197 degrees of freedom
## Multiple R-squared:  0.9863, Adjusted R-squared:  0.9862
## F-statistic:  7108 on 2 and 197 DF,  p-value: < 2.2e-16
```

Example: Crabs

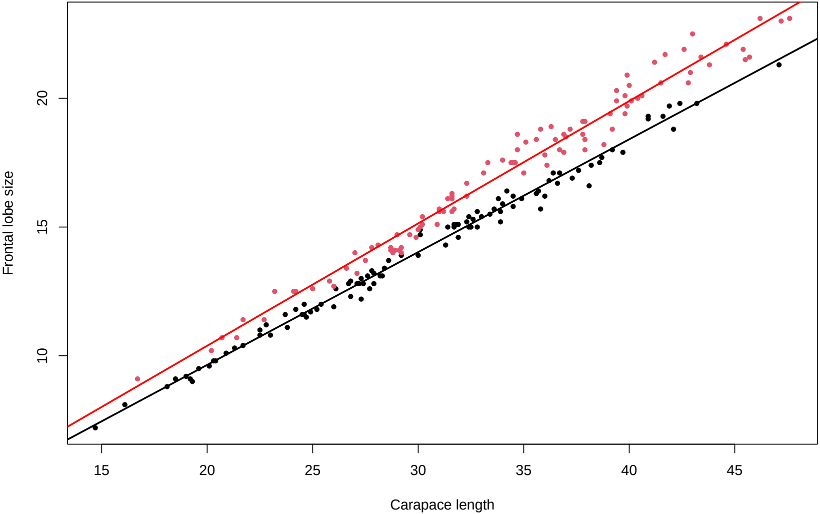
Observe that if we compare this model with the complete model through an anova table, we get the same test as in the summary:

```r
anova(lmInter,lmComplete)
```

```
## Analysis of Variance Table
##
## Model 1: FL ~ CL + CL:fsp
## Model 2: FL ~ CL + fsp + CL:fsp
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    197 33.232
## 2    196 33.139  1  0.093374 0.5523 0.4583
```

Example: Crabs

```
plot(CL,FL, pch=20, xlab='Carapace length', ylab='Frontal lobe size', col=sp)
beta <- coef(lmInter)
abline(beta[1], beta[2],lwd=2)
abline(beta[1], sum(beta[-1]),lwd=2, col='red')
```



Example: Crabs

The final model is

$$FL = 0.881395 + 0.438158 \times CL + 0.037147 \times CL \times fsp.$$

Problem List 10

Problem List 10

1

2