GLM Practical Sessions, Week 3

alexaoh

28.09.21 and 30.09.21

Linear Regression for Cholesterol

```r
data <- read.csv2("COL.csv", header = T)
summary(data)
```

```
#>        A               H               W               C        
#>  Min.   : 9.00   Min.   :103.0   Min.   :37.30   Min.   : 67.5  
#>  1st Qu.:12.00   1st Qu.:130.5   1st Qu.:53.23   1st Qu.:166.5  
#>  Median :15.00   Median :151.5   Median :66.60   Median :217.8  
#>  Mean   :14.71   Mean   :147.4   Mean   :64.57   Mean   :218.2  
#>  3rd Qu.:18.00   3rd Qu.:167.2   3rd Qu.:74.95   3rd Qu.:262.4  
#>  Max.   :20.00   Max.   :187.0   Max.   :89.70   Max.   :438.5  
```

Simple Linear Regression with W - Exercise 1

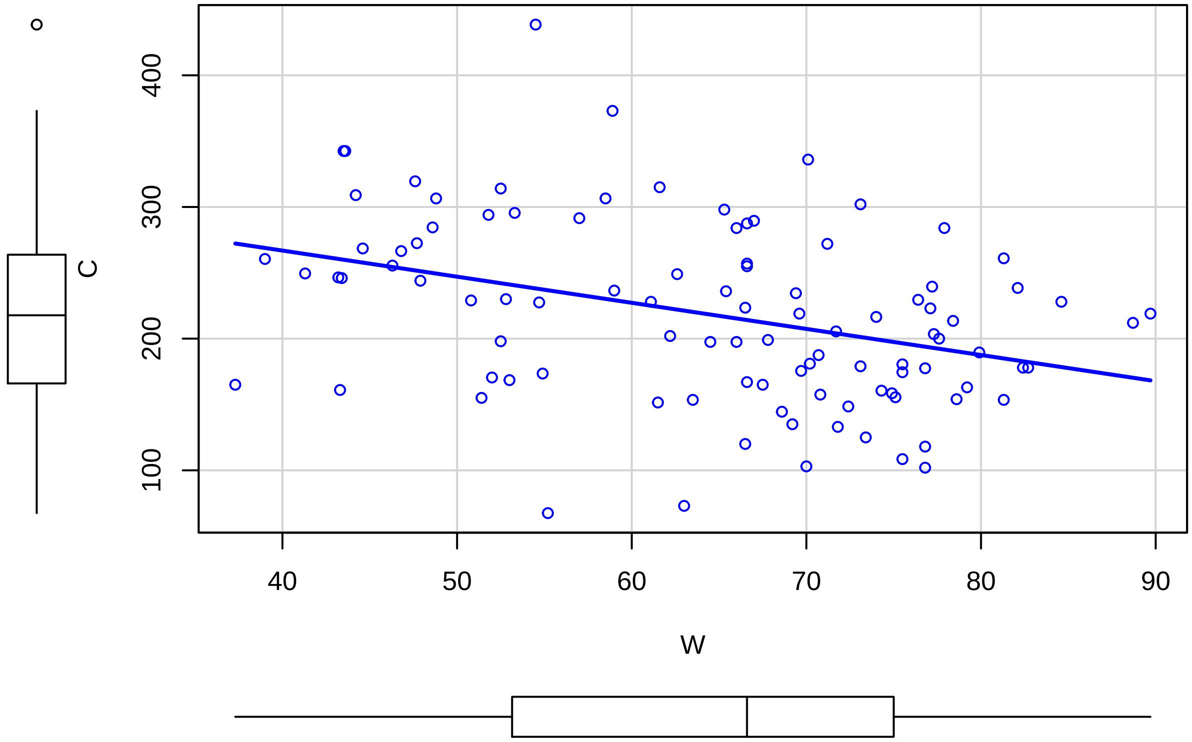
```r
p <- 2
n <- dim(data)[1]

# Fit linear model.
lm.fit <- lm(C~W, data = data)
summary(lm.fit)
```

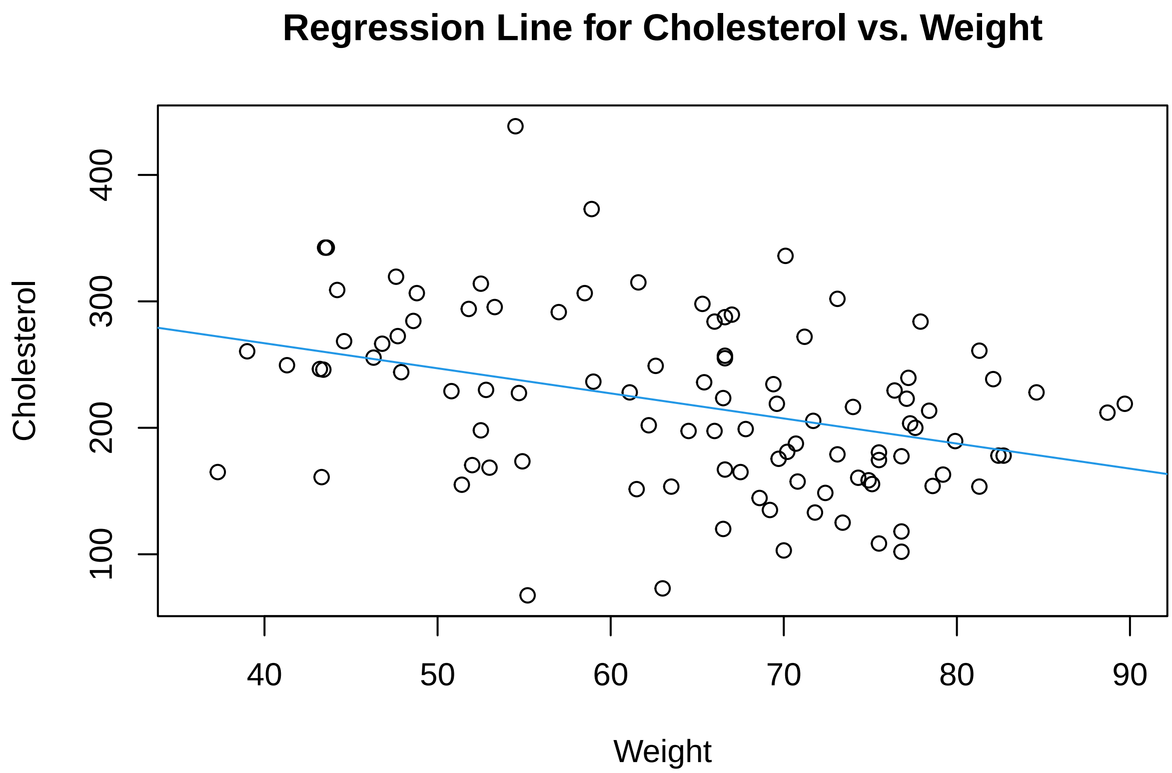
```
#> 
#> Call:
#> lm(formula = C ~ W, data = data)
#> 
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -169.24  -39.81   -4.49   47.19  200.37 
#> 
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)    
#> (Intercept) 346.2251    33.1983   10.43  < 2e-16 ***
#> W            -1.9835     0.5046   -3.93 0.000158 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> Residual standard error: 63.55 on 98 degrees of freedom
#> Multiple R-squared:  0.1362, Adjusted R-squared:  0.1274 
#> F-statistic: 15.45 on 1 and 98 DF,  p-value: 0.0001581
```

Scatterplot of Points and Regression Line.

```
# Can be done manually and with a function.
scatterplot(C~W, smooth = F, data = data)
```



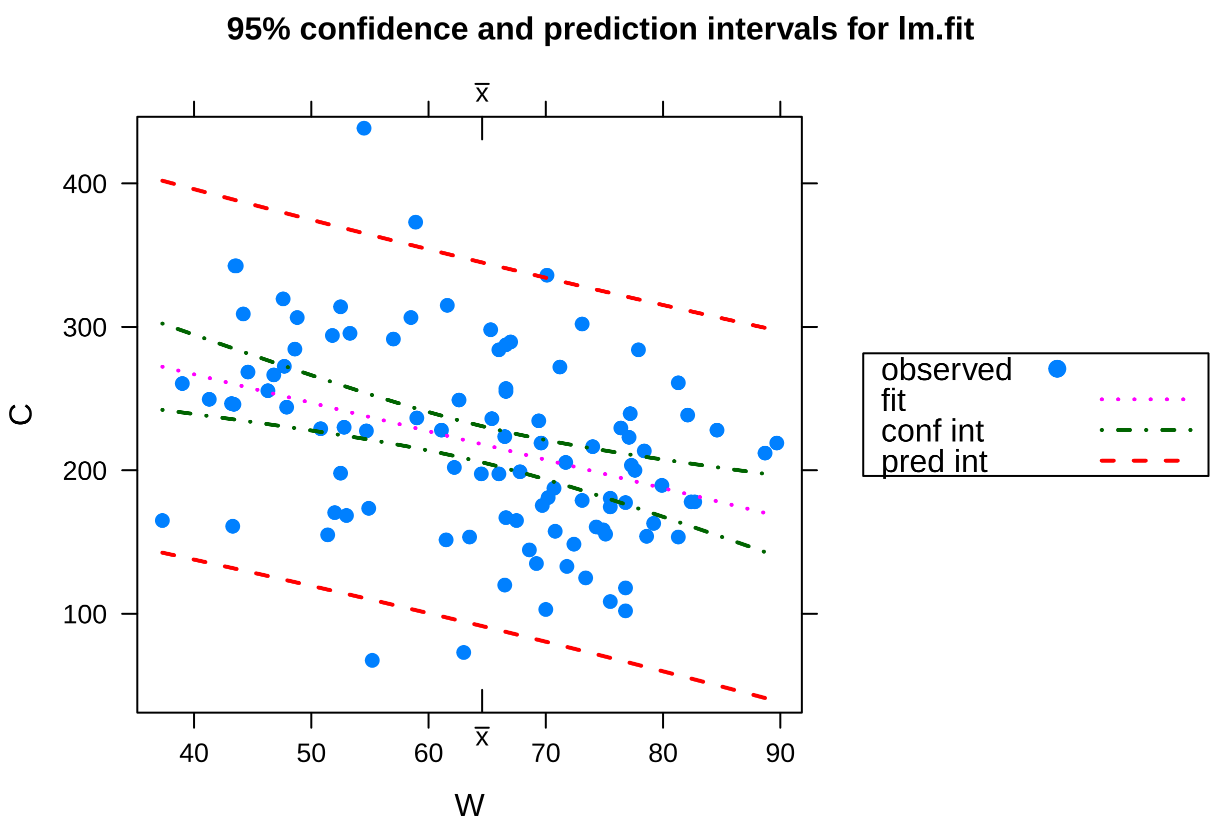

```
plot(data[, "W"], data[, "C"], main = "Regression Line for Cholesterol vs. Weight",
     xlim = c(36, 90), ylim = c(66, 440), xlab = "Weight", ylab = "Cholesterol")
abline(lm.fit, col = 4)
```




```
# Could do the plot from above with the scatterplot function above (comes from 'car' package).
```

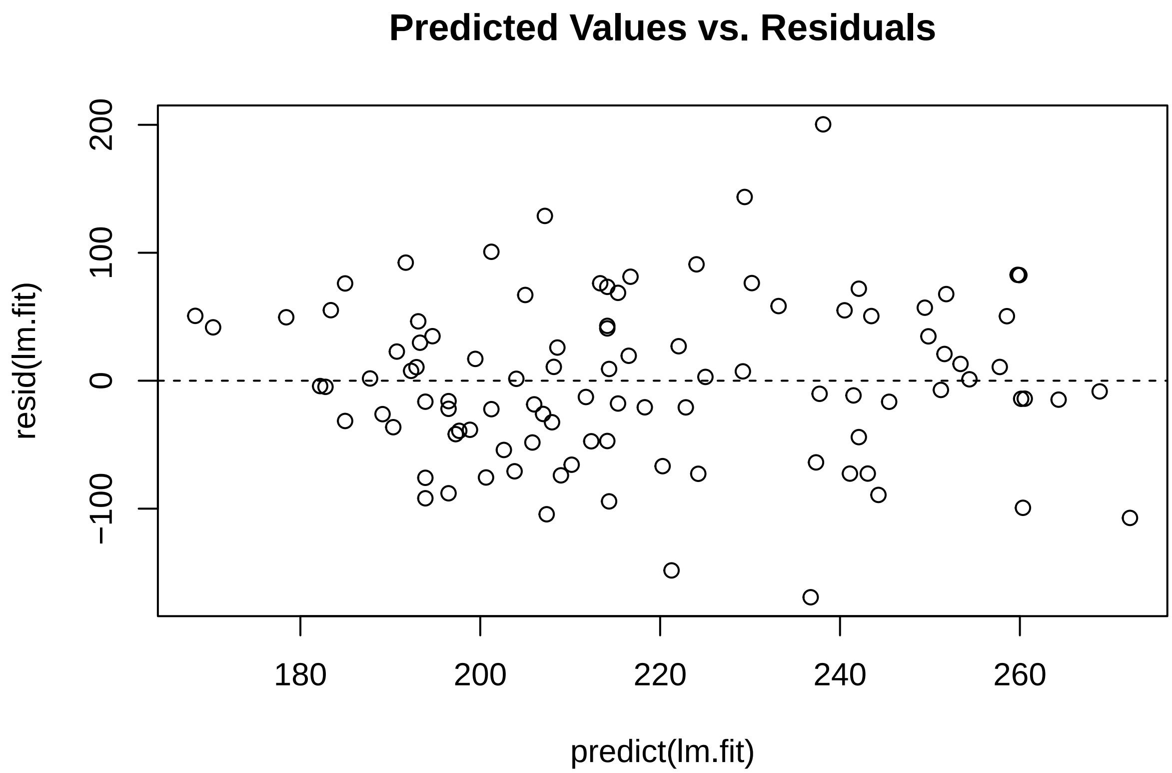
Plot Regression Line with Conf. and Pred. Intervals

```
# Plot confidence and prediction intervals with regression line (From package 'HH').
ci.plot(lm.fit)
```



Plot Predicted Values vs. Residuals

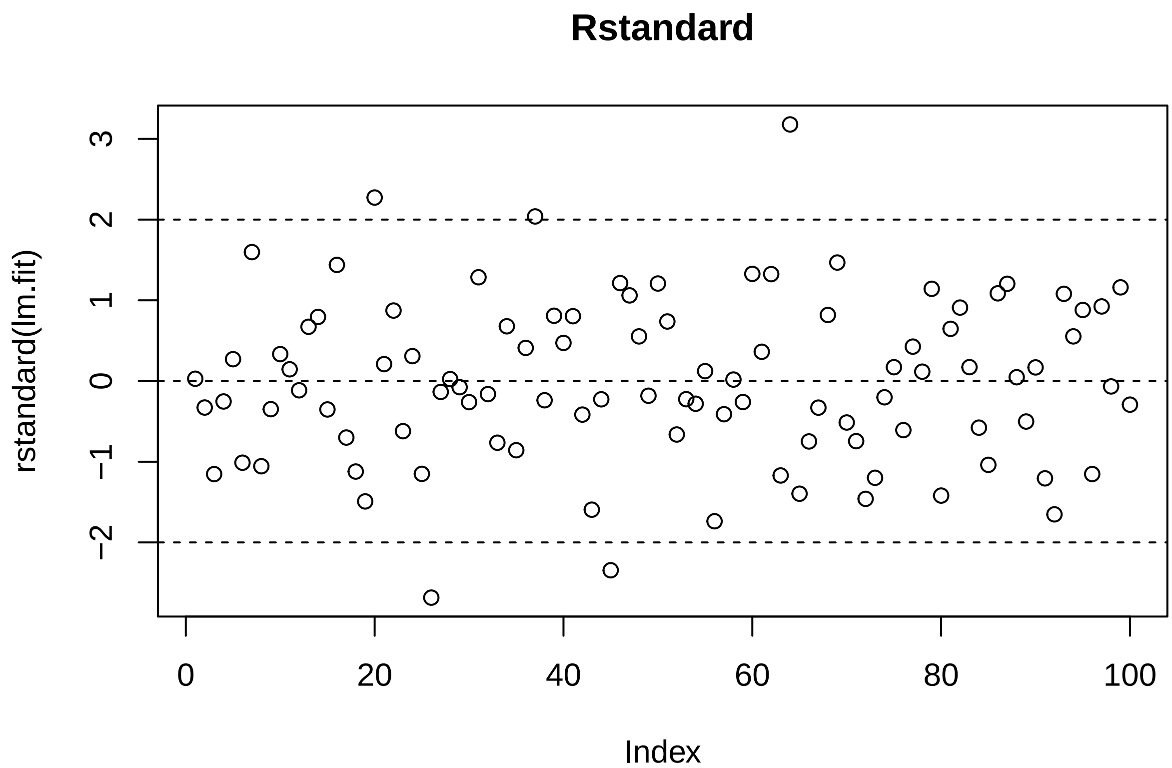
```
# Plot the predicted values vs. residuals.
plot(predict(lm.fit), resid(lm.fit), main = "Predicted Values vs. Residuals")
abline(h=0, lty = 2)
```



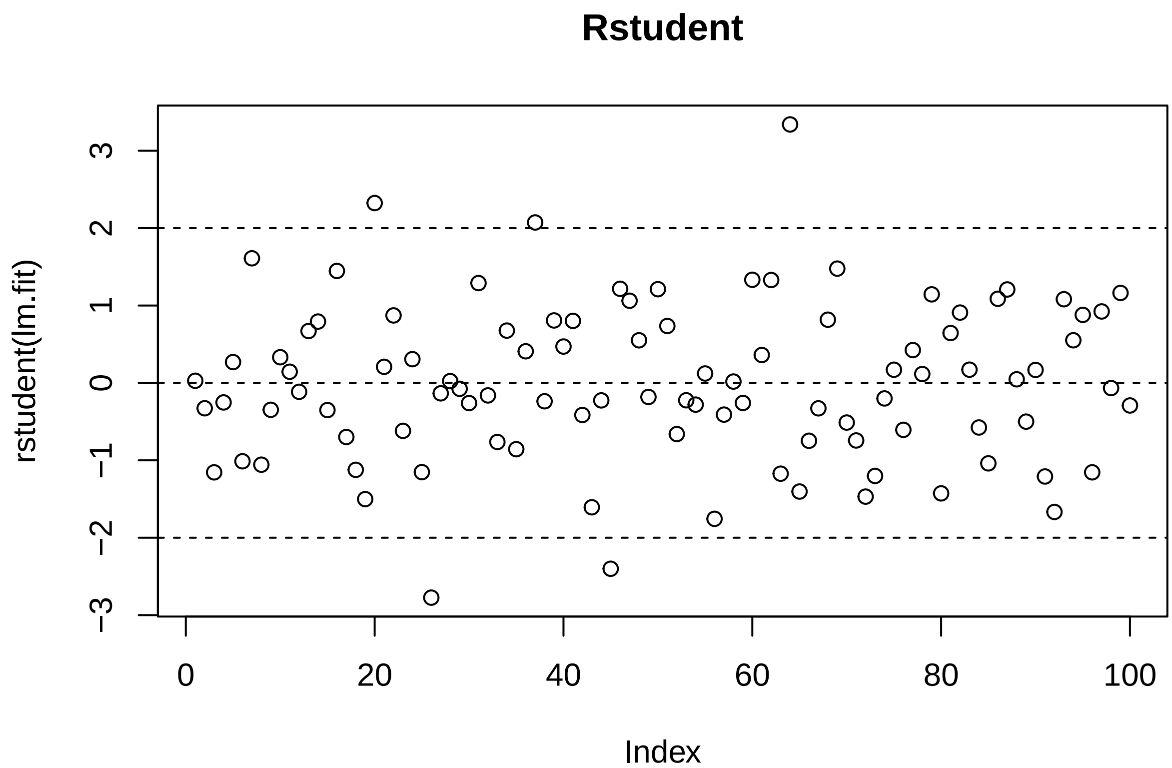
Plot Standardized/Studentized Residuals

Here: 5 of the points should be outside the lines -2 and 2, since we here have 95% confidence intervals (2 approximates 1.96) and we have 100 points in the data. We can see that this is the case.

```
plot(rstandard(lm.fit), main = "Rstandard")
abline(h=c(-2, 0, 2), lty = 2)
```

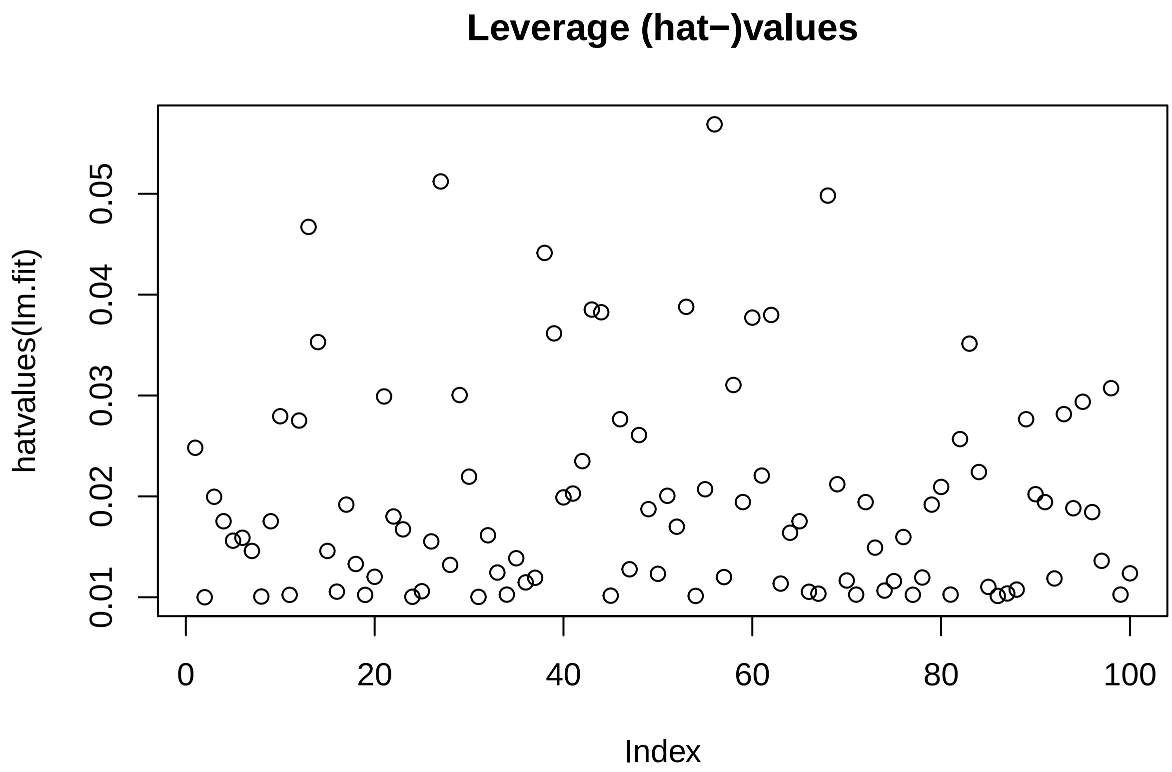



```
plot(rstudent(lm.fit), main = "Rstudent")
abline(h=c(-2, 0, 2), lty = 2)
```



Diagnostic: Leverage

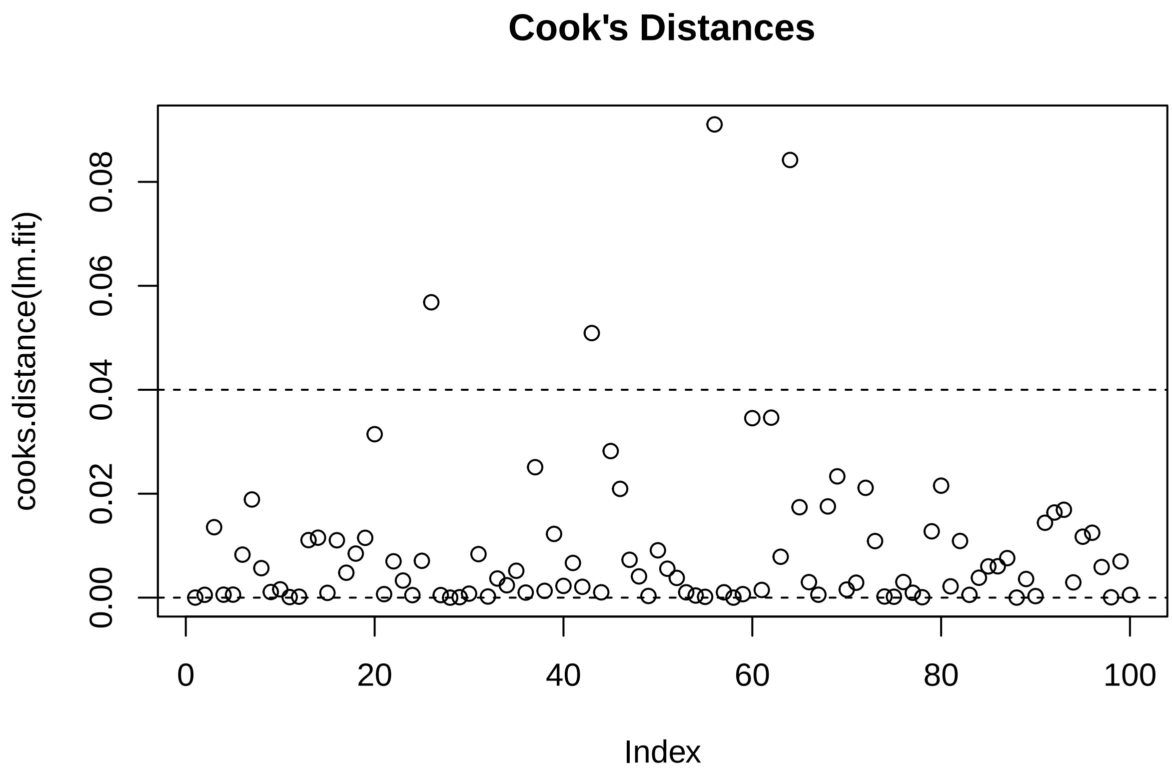
```
# A line at 0.06 for some reason ? Check code after session!
plot(hatvalues(lm.fit), main= "Leverage (hat-)values")
```



Diagnostic: Influential observations (dffits, cooks.distance)

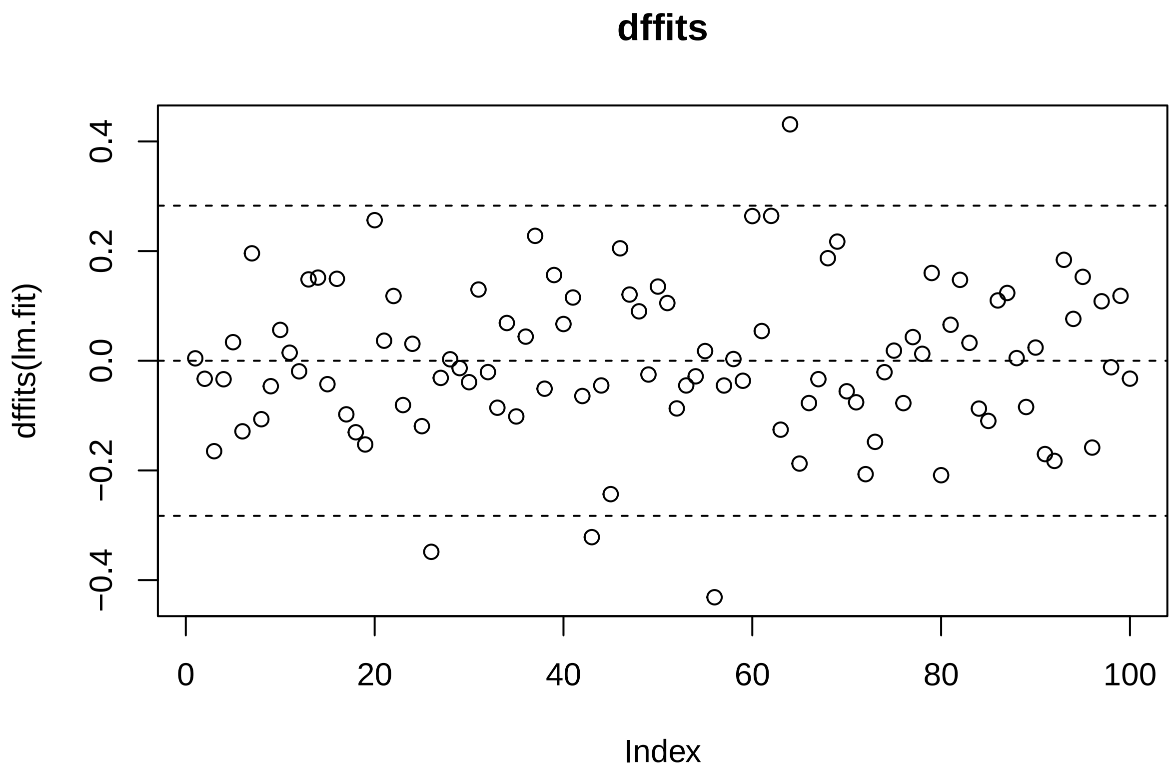
Calculate the Cook's distances.

```
plot(cooks.distance(lm.fit), main = "Cook's Distances")
abline(h=c(0,4/n),lty = 2)
```



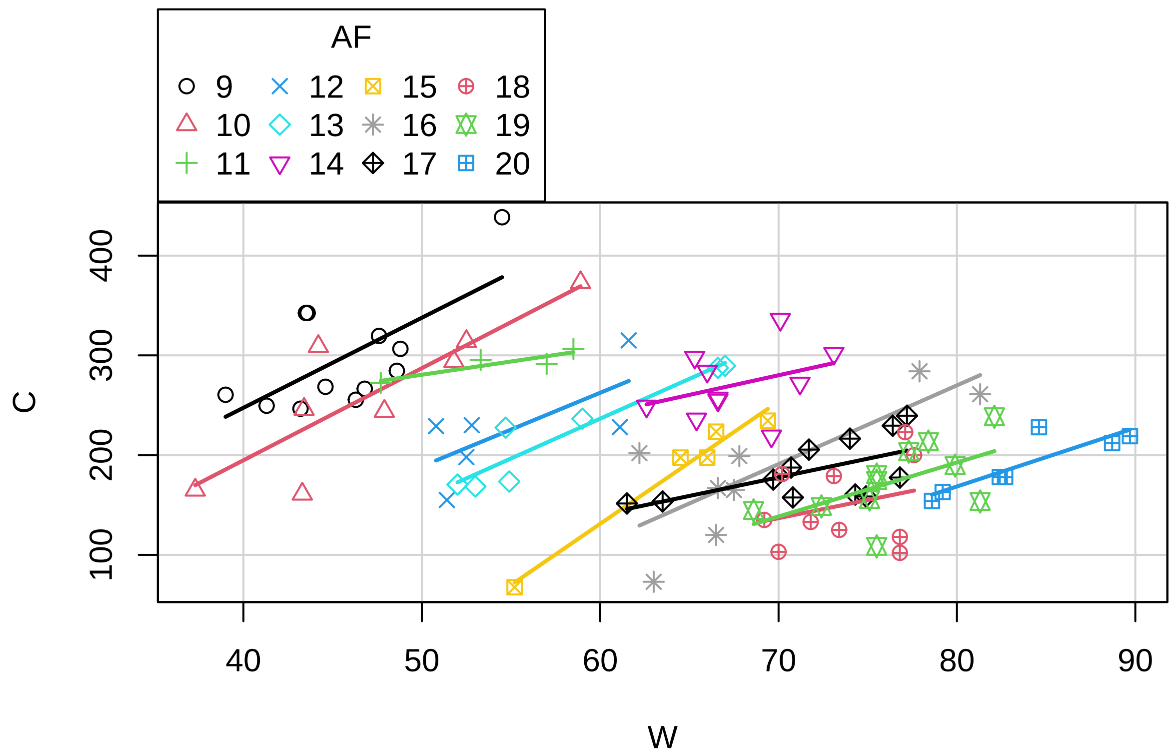
Compute dffits (difference of fits). This is the difference between the fits when a point is in or out of the dataset.

```
plot(dffits(lm.fit), main = "dffits")
abline(h=c(-2*sqrt(p/n), 0, 2*sqrt(p/n)), lty = 2)
```



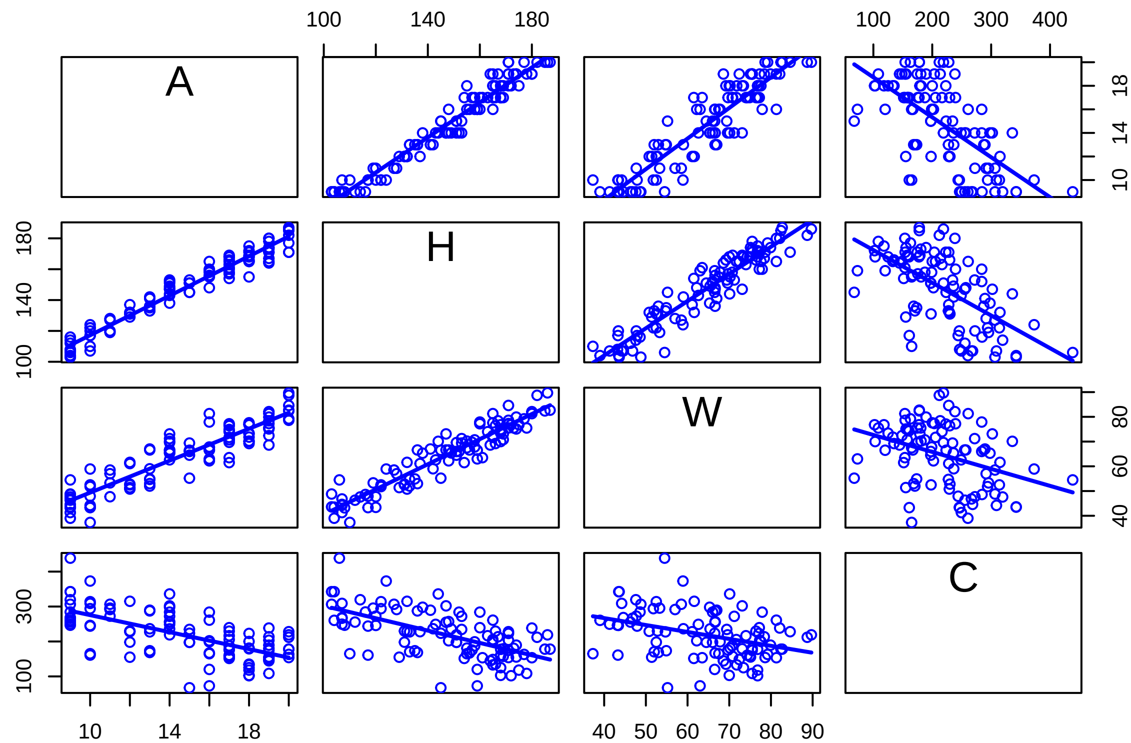
Perform a simple regression for each group of age

```
data$AF <- factor(data$A)
sp(C~W|AF,smooth=F,col=1:20, data=data)
```



Multiple Linear Regression - Exercise 3

```
data <- data[, -5] # Remove AF again.
scatterplotMatrix(data, smooth = F, diagonal = F)
```




```r
lm.fitm <- lm(C~W+A+H, data = data)
summary(lm.fitm)
```

```
#>
#> Call:
#> lm(formula = C ~ W + A + H, data = data)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -74.608 -22.137   1.888  21.156  65.410
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 490.9978    35.0517  14.008  < 2e-16 ***
#> W            10.3773     0.7365  14.090  < 2e-16 ***
#> A           -13.0195     3.8530  -3.379  0.00105 **
#> H            -5.0989     0.7227  -7.055 2.68e-10 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 30.11 on 96 degrees of freedom
#> Multiple R-squared:  0.8101, Adjusted R-squared:  0.8041
#> F-statistic: 136.5 on 3 and 96 DF,  p-value: < 2.2e-16
```

```r
p <- 4
```

$\hat{\sigma}^2 \approx$ Residual standard error$^2 = (30.11)^2$.

Omnibus test (F-test)

Test the null-model (all coefficients are zero, except for the intercept) vs. our model (at least one of the coefficients are zero).

Anova

```r
anova(lm.fitm) # Performs the Type-I test. Order of the variables is important.
```

```
#> Analysis of Variance Table
#>
#> Response: C
#>           Df Sum Sq Mean Sq F value    Pr(>F)
#> W          1  62396   62396  68.826 6.686e-13 ***
#> A          1 263670  263670 290.841 < 2.2e-16 ***
#> H          1  45123   45123  49.773 2.676e-10 ***
#> Residuals 96  87031     907
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(lm.fitm) # Performs the Type-II test. Order of the variables is NOT important.
```

```
#> Anova Table (Type II tests)
#>
#> Response: C
#>           Sum Sq Df F value    Pr(>F)
#> W         179985  1 198.533 < 2.2e-16 ***
#> A          10351  1  11.418  0.001052 **
#> H          45123  1  49.773 2.676e-10 ***
#> Residuals  87031 96
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Confidence Intervals

```r
confint(lm.fitm, level = 0.99)
```

```
#>                  0.5 %     99.5 %
#> (Intercept) 398.881272 583.114304
#> W             8.441792  12.312821
#> A           -23.145228  -2.893732
#> H            -6.998311  -3.199551
```

Prediction

```r
CO <- data.frame(cbind(W = c(65, 75, 65), A = c(15, 15, 12), H = c(150, 150, 150)), row.names = 1:3)
predict(lm.fitm, CO, interval = "confidence", level=0.95, se.fit = T)
```

```
#> $fit
#>        fit      lwr      upr
#> 1 205.3908 199.1668 211.6148
#> 2 309.1639 294.6188 323.7089
#> 3 244.4492 219.8210 269.0774
#>
#> $se.fit
```

```
#>         1          2          3
#>  3.135539  7.327533 12.407261
#>
#> $df
#> [1] 96
#>
#> $residual.scale
#> [1] 30.1094
```

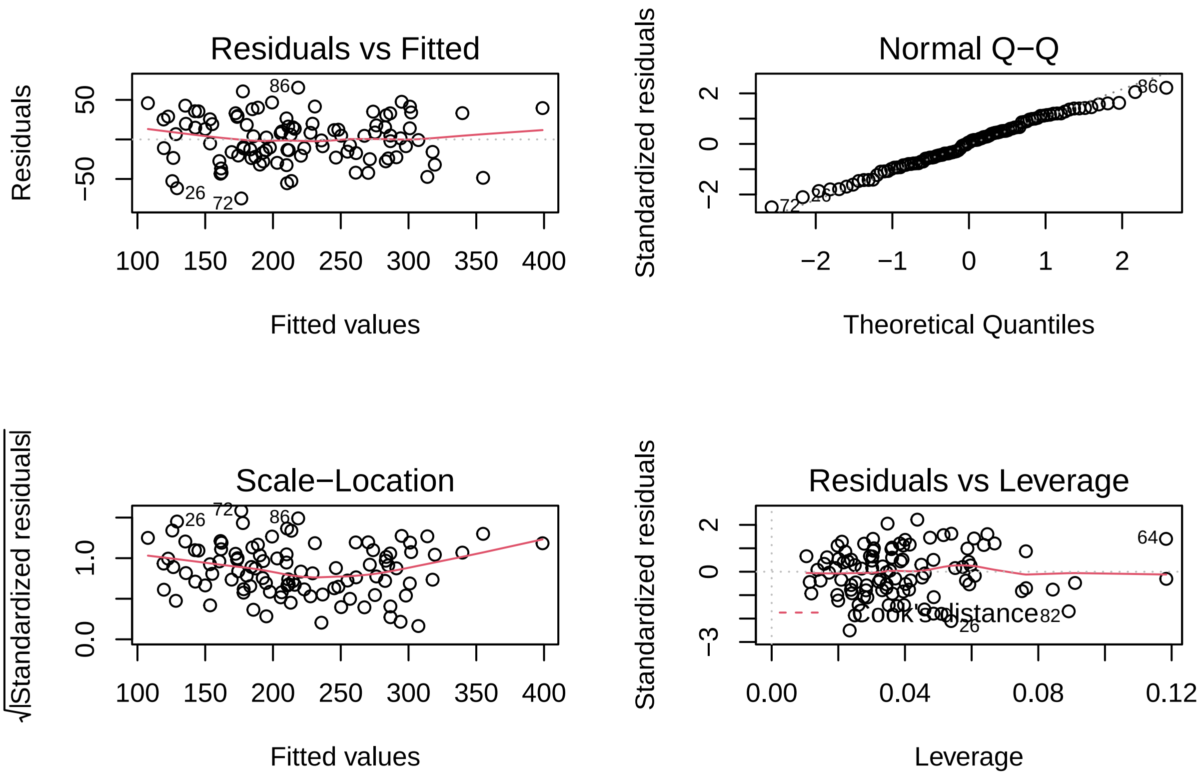
```r
# How can it calculate confidence intervals for new predictions (for the mean)?
```

```r
predict(lm.fitm, C0, interval = "prediction", level=0.95, se.fit = T)
```

```
#> $fit
#>        fit      lwr      upr
#> 1 205.3908 145.3009 265.4807
#> 2 309.1639 247.6528 370.6749
#> 3 244.4492 179.8071 309.0914
#>
#> $se.fit
#>         1          2          3
#>  3.135539  7.327533 12.407261
#>
#> $df
#> [1] 96
#>
#> $residual.scale
#> [1] 30.1094
```

R Diagnostic

```r
par(mfrow=c(2,2))
plot(lm.fitm)
```



Then we did some more diagnostics, similar to the ones dones in the simple linear regression above.

Have a look at the file in Atenea for all of this + explanations regarding all of the work done in these exercises.

DO EXERCISE 2 AND 4 UNTIL NEXT TUESDAY!!!