ASM - Module 3 Solutions

Michal Siwek

Thursday, November 05, 2015

Task 1

a) Viewing the model output

```
attach(trees)
fit1 <- lm(Volume ~ Girth)
fit2 <- lm(Volume ~ Height)
names(fit1)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
fit1$coefficients
```

```
## (Intercept)       Girth
##  -36.943459    5.065856
```

```
fit1$fitted.values
```

```
##         1         2         3         4         5         6         7
##  5.103149  6.622906  7.636077 16.248033 17.261205 17.767790 18.780962
##         8         9        10        11        12        13        14
## 18.780962 19.287547 19.794133 20.300718 20.807304 20.807304 22.327061
##        15        16        17        18        19        20        21
## 23.846818 28.406089 28.406089 30.432431 32.458774 32.965360 33.978531
##        22        23        24        25        26        27        28
## 34.991702 36.511459 44.110244 45.630001 50.695857 51.709028 53.735371
##        29        30        31
## 54.241956 54.241956 67.413183
```

```
fit1$residuals
```

```
##          1          2          3          4          5          6
##  5.1968508  3.6770939  2.5639226  0.1519667  1.5387954  1.9322098
##          7          8          9         10         11         12
## -3.1809615 -0.5809615  3.3124528  0.1058672  3.8992815  0.1926959
##         13         14         15         16         17         18
##  0.5926959 -1.0270610 -4.7468179 -6.2060887  5.3939113 -3.0324313
##         19         20         21         22         23         24
## -6.7587739 -8.0653595  0.5214692 -3.2917021 -0.2114590 -5.8102436
##         25         26         27         28         29         30
## -3.0300006  4.7041430  3.9909717  4.5646292 -2.7419565 -3.2419565
##         31
##  9.5868168
```

b) Viewing the fitted model summary

```
summary(fit1)
```

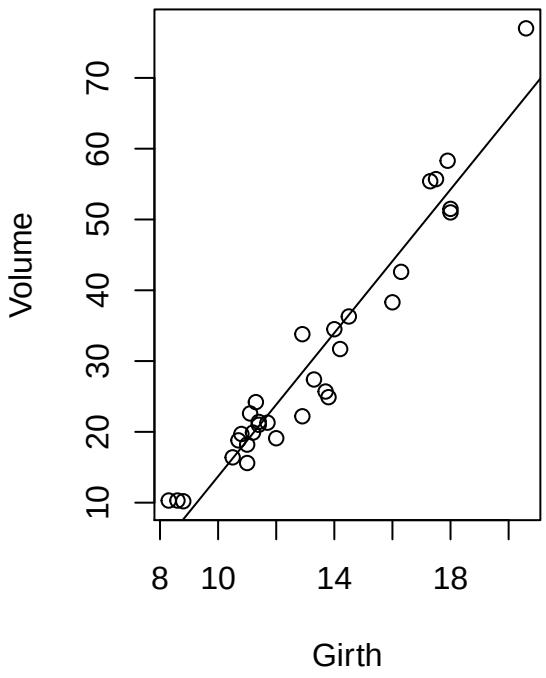
```
##
## Call:
## lm(formula = Volume ~ Girth)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.065 -3.107  0.152  3.495  9.587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
## Girth         5.0659     0.2474   20.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

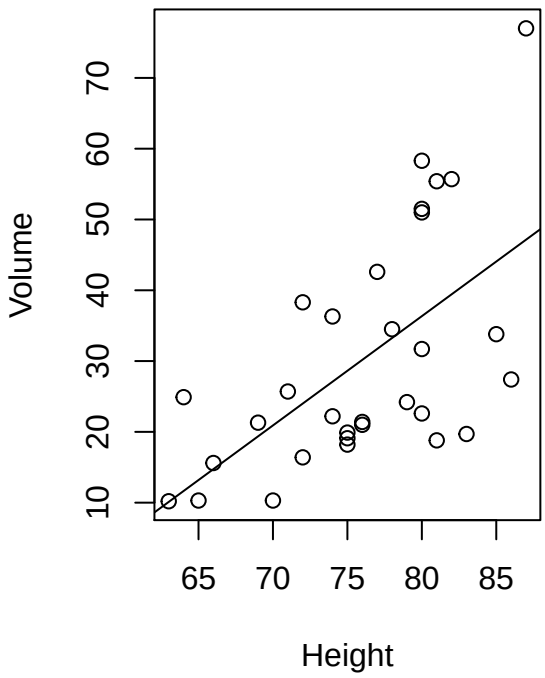
```
names(summary(fit1))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

c) Plots

```
par(mfrow = c(1, 2))
plot(Girth, Volume)
abline(fit1)
plot(Height, Volume)
abline(fit2)
```





d) Comparing R^2 values

```
data.frame(r.squared = c(summary(fit1)$r.squared, summary(fit2)$r.squared),
           row.names = c(fit1$call, fit2$call))
```

```
##                                r.squared
## lm(formula = Volume ~ Girth)  0.9353199
## lm(formula = Volume ~ Height) 0.3579026
```

R^2 is much lower for the second model.

e) Are the relationships statistically significant?

```
summary(fit1)
```

```
##
## Call:
## lm(formula = Volume ~ Girth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -8.065 -3.107  0.152  3.495  9.587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
## Girth         5.0659     0.2474   20.48  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.252 on 29 degrees of freedom
## Multiple R-squared:  0.9353, Adjusted R-squared:  0.9331
## F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = Volume ~ Height)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.274  -9.894  -2.894  12.068  29.852
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -87.1236    29.2731  -2.976 0.005835 **
## Height        1.5433     0.3839   4.021 0.000378 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.4 on 29 degrees of freedom
## Multiple R-squared:  0.3579, Adjusted R-squared:  0.3358
## F-statistic: 16.16 on 1 and 29 DF,  p-value: 0.0003784
```

Assuming that the linear model is correct (it's likely not, as the residual plot shows a nonlinear pattern) we can say that in the absence of any other predictors Girth and Heights (each separately) turns out to be significant for the Volume. However there may be other relevant predictors that may change this conclusion if included in the models. So it's imporant to decide whether there are any possible confounders that are omitted in each of these two models.

f) Confidence interval for the slope coefficient

Derivation using scalar formulas:

```
n <- nrow(trees)
p <- 2
df <- n - p
b0 <- fit1$coefficients["(Intercept)"]
b1 <- fit1$coefficients["Girth"]
sigSq <- sum(fit1$residuals^2) / (n - p)
var_b1 <- sigSq * (1 / sum((Girth - mean(Girth))^2))
b1 + c(-1, 1) * sqrt(var_b1) * qt(.975, df)
```

```
## [1] 4.559914 5.571799
```

Derivation using matrices (variance-covariance matrix):

```
xMat <- matrix(c(rep(1, length(Girth)), Girth), ncol = 2)
vcm <- (summary(fit1)$sigma)^2 * solve(t(xMat) %*% xMat)
vcm <- matrix(vcm, ncol = 2,
              dimnames = list(c("(Intercept)", "Girth"), c("(Intercept)", "Girth")))
b1 + c(-1, 1) * sqrt(vcm["Girth", "Girth"]) * qt(.975, df)
```

```
## [1] 4.559914 5.571799
```

The same using summary(fit):

```
b1 + c(-1, 1) * summary(fit1)$sigma *
    sqrt(summary(fit1)$cov.unscaled["Girth", "Girth"]) * qt(.975, df)
```

```
## [1] 4.559914 5.571799
```

The same using confint function:

```
confint(fit1)["Girth",]
```

```
##    2.5 %   97.5 %
## 4.559914 5.571799
```

g) Estimated variance of the tree volume

```
summary(fit1)$sigma^2
```

```
## [1] 18.0794
```

h) Predicting $Volume$ for $Girth = 15$ inch

Using estimated coefficients:

```
unname(b0 + b1 * 15)
```

```
## [1] 39.04439
```

Using predict function:

```
unname(predict(fit1, newdata = data.frame(Girth = 15)))
```

```
## [1] 39.04439
```

```
detach(trees)
```

Task 2

Preparing the data:

```
file <- "anscombe_quartet.txt"
data <- read.table(file, header = T)
n <- nrow(data)
m <- ncol(data) / 2  # number of pairs of data (potential relationships)
allX <- data[,grep("^X", colnames(data))]
allY <- data[,grep("^Y", colnames(data))]
```

a) Fitting LS lines

```
fit <- list()
for(i in 1:m)
    fit <- c(fit, list(lm(allY[, i] ~ allX[, i])))
```

b) Models' outputs comparison

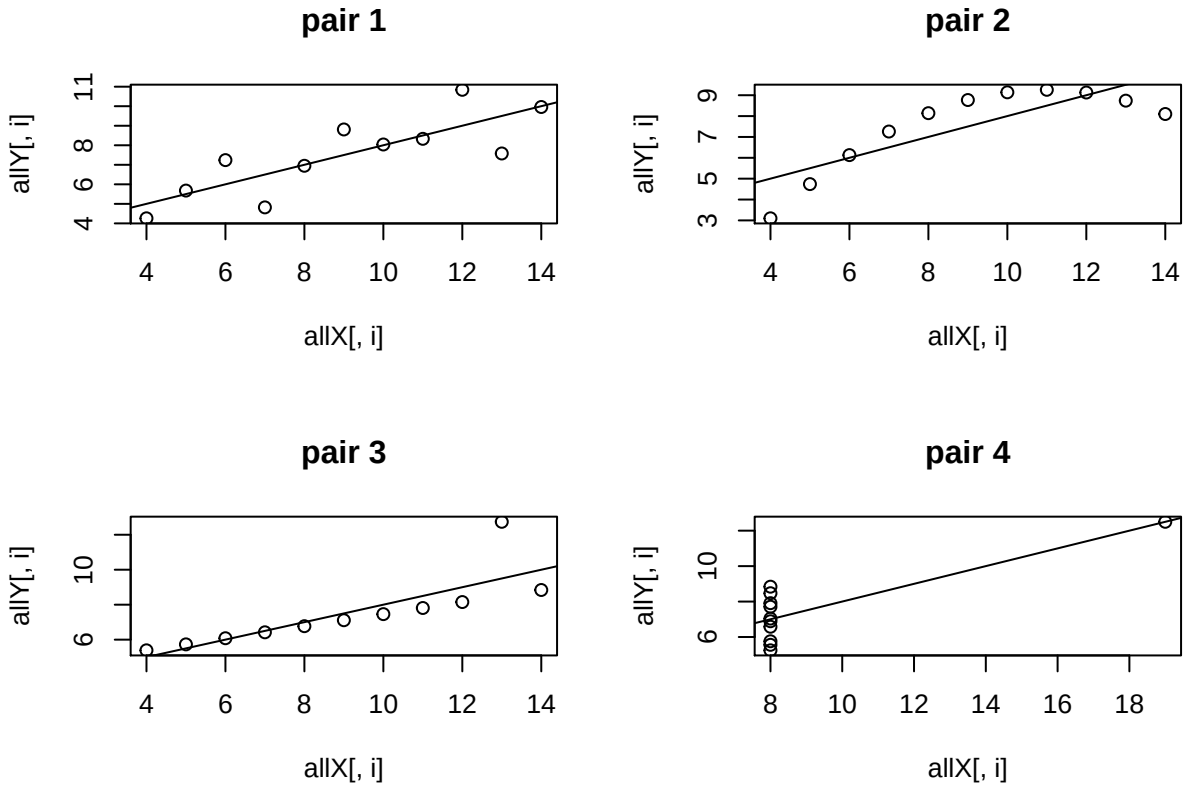
```
sapply(1:m, function(i) {
        setNames(c(fit[[i]]$coefficients,
                   summary(fit[[i]])$r.squared,
                   cor(allX[, i], allY[, i])),
                 c("b0", "b1", "R^2", "cor"))
    })
```

```
##          [,1]      [,2]      [,3]      [,4]
## b0  3.0000909 3.0009091 3.0024545 3.0017273
## b1  0.5000909 0.5000000 0.4997273 0.4999091
## R^2 0.6665425 0.6662420 0.6663240 0.6667073
## cor 0.8164205 0.8162365 0.8162867 0.8165214
```

All the respective values are very close to each other.

c) Plots

```
par(mfrow = c(2, 2))
for(i in 1:m) {
    plot(allX[, i], allY[, i], main = paste("pair", i))
    abline(fit[[i]])
}
```



Fitting the linear model is reasonable in the **case #1**. In **case #3** there's one high-leverage outlier - this one data point has a big impact on the slope estimation. Linear model is a reasonable solution in this case if this one point is an error. Then we should remove it before fitting the model. **Case #2** shows nonlinear relationship. In **case #4** almost all data lay at one value of X. Putting aside the one outlaying point there's no variation in X's so we can't infere anything about the impact of X on Y. Here the regression hinges on just one outlaying data point and shouldn't be considered reliable.

Task 3

Preparing the data and fitting the model with all avaliable predictors:

```
file <- "realest.txt"
data <- read.table(file, header = T)
fit <- lm(Price ~ ., data)
```

a) Impact of the number of bedrooms on the house price

```
summary(fit)
```

```
##
## Call:
## lm(formula = Price ~ ., data = data)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.7630  -4.0514   0.5389   2.3899  12.9855
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.712572   9.514111   1.441   0.1677
## Bedroom     -7.756208   3.109374  -2.494   0.0232 *
## Space        0.011626   0.008981   1.295   0.2128
## Room         5.097706   2.764303   1.844   0.0827 .
## Lot          0.228063   0.195434   1.167   0.2593
## Tax          0.003374   0.006859   0.492   0.6291
## Bathroom     5.718372   4.276867   1.337   0.1988
## Garage       3.613603   2.064997   1.750   0.0982 .
## Condition   -2.162027   4.137400  -0.523   0.6080
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.337 on 17 degrees of freedom
## Multiple R-squared:  0.7688, Adjusted R-squared:   0.66
## F-statistic: 7.065 on 8 and 17 DF,  p-value: 0.0003757
```

In the full model, adding one bedroom decreases the expected house price by 7.76.

```
fit1 <- lm(Price ~ Bedroom, data)
summary(fit1)
```

```
##
## Call:
## lm(formula = Price ~ Bedroom, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.170  -7.769   1.211   8.731  22.830
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   43.487      6.245   6.964 3.35e-07 ***
## Bedroom        3.921      1.797   2.182   0.0391 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.73 on 24 degrees of freedom
## Multiple R-squared:  0.1655, Adjusted R-squared:  0.1308
## F-statistic: 4.761 on 1 and 24 DF,  p-value: 0.03914
```

In the minimal model, adding one bedroom increases the expected house price by 3.92.
These two conclusions are contradictory. The reason behind this might be that the size of the house is cought by four variables ($Bedroom, Space, Room, Bathroom$) that are positively correlated with each other:

```
cor(data[,c("Bedroom", "Space", "Room", "Bathroom")])
```

```
##             Bedroom     Space      Room  Bathroom
```

```
## Bedroom  1.0000000 0.6753029 0.9175251 0.6327913
## Space    0.6753029 1.0000000 0.7400716 0.5251595
## Room     0.9175251 0.7400716 1.0000000 0.6874336
## Bathroom 0.6327913 0.5251595 0.6874336 1.0000000
```

My guess is that the the price increases with the size of the house, so if bathroom is the only predictor then it serves as the measure for the house size. When all predictors are included, the the other ones give better meausure of the house size and hence are better predictors of the impact of the house size on the price. In the full model the bathroom coefficient measures the impact of additional bathroom keeping the size of the house (i.e. number of rooms, bedrooms, the total space) intact. The sign of the estimated coefficient would suggest that there are more houses having too many than too few bathrooms considering market demand and thus on average additional bathroom have a negative impact on the price.

b) Prediction

```
new_house <- data.frame(Condition = 1,
                        Bedroom = 3,
                        Room = 8,
                        Bathroom = 2,
                        Garage = 1,
                        Space = 1500,
                        Lot = 40,
                        Tax = 1000)
```

Mean value response prediction:

```
predict(fit, new_house, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 74.05021 64.33342 83.76699
```

Value of the response prediction:

```
predict(fit, new_house, interval = "prediction")
```

```
##        fit      lwr     upr
## 1 74.05021 55.77411 92.3263
```

Task 4

Preparing the data and fitting the models:

```
file <- "cheese.txt"
data <- read.table(file, header = T)
fit1 <- lm(taste ~ Acetic, data)
fit2 <- lm(taste ~ Acetic + Lactic + H2S, data)
summary(fit1)
```

```
##
## Call:
## lm(formula = taste ~ Acetic, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.642  -7.443   2.082   6.597  26.581
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -61.499     24.846  -2.475  0.01964 *
## Acetic        15.648      4.496   3.481  0.00166 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.82 on 28 degrees of freedom
## Multiple R-squared:  0.302,  Adjusted R-squared:  0.2771
## F-statistic: 12.11 on 1 and 28 DF,  p-value: 0.001658
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = taste ~ Acetic + Lactic + H2S, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.390  -6.612  -1.009   4.908  25.449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.8768    19.7354  -1.463  0.15540
## Acetic        0.3277     4.4598   0.073  0.94198
## Lactic       19.6705     8.6291   2.280  0.03108 *
## H2S           3.9118     1.2484   3.133  0.00425 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.13 on 26 degrees of freedom
## Multiple R-squared:  0.6518, Adjusted R-squared:  0.6116
## F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

F-test for testing hypothesis that the smaller model fits better the data than the large one

Step by step derivation of the p-value:

```
n <- nrow(data)
q <- 2
p <- 4
ESS1 <- sum(fit1$residuals^2)
ESS2 <- sum(fit2$residuals^2)
F_stat <- ((ESS1 - ESS2) / (p - q)) / (ESS2 / (n - p))
pf(F_stat, p - q, n - p, lower.tail = F)
```

```
## [1] 0.0001185798
```

Obtaining the p-value using anova function:

```
anova(fit1, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: taste ~ Acetic
## Model 2: taste ~ Acetic + Lactic + H2S
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     28 5348.7
## 2     26 2668.4  2    2680.3 13.058 0.0001186 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

At the significance level of 0.05 we reject the null hypothesis that the smaller model is better.