MAS programmes - Statistical Analysis (Autumn Term)

Exercises 2 - SOLUTIONS

1. The appropriate functional is given by

$$\mathcal{L} = \sum_{i=1}^{n}\{y_i - (\beta_0 + \beta_1(x_i - \bar{x}))\}^2 = \sum_{i=1}^{n}\{y_i^2 - 2y_i(\beta_0 + \beta_1(x_i - \bar{x})) + (\beta_0 + \beta_1(x_i - \bar{x}))^2\}.$$

Hence to find the values of β_0 and β_1 that minimize this quantity, we must find

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \beta_0} &= \sum_{i=1}^{n}\{-2y_i + 2(\beta_0 + \beta_1(x_i - \bar{x}))(1)\} \\ &= -2\sum_{i=1}^{n} y_i + 2\sum_{i=1}^{n}\beta_0 \qquad (1)\end{aligned}$$

since $\sum_{i=1}^{n}(x_i - \bar{x}) = 0$, and

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \beta_1} &= \sum_{i=1}^{n}\{-2y_i(x_i - \bar{x}) + 2(\beta_0 + \beta_1(x_i - \bar{x}))(x_i - \bar{x})\} \\ &= -2\sum_{i=1}^{n} y_i(x_i - \bar{x}) + 2\beta_1\sum_{i=1}^{n}(x_i - \bar{x})^2. \qquad (2)\end{aligned}$$

Setting these partial derivatives equal to zero, we find

$$n\hat{\beta}_0 = \sum_{i=1}^{n} y_i \qquad \text{(from (1))}$$

and

$$\hat{\beta}_1\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n}(x_i - \bar{x})y_i \qquad \text{(from (2))}$$

(these are the *normal equations*).

Now,

$$\sum_{i=1}^{n}(x_i - \bar{x})y_i = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

since $\bar{y}\sum_{i=1}^{n}(x_i - \bar{x}) = 0$. Hence the normal equations may be written

$$\begin{aligned}n\hat{\beta}_0 &= n\bar{y} \\ S_{xx}\hat{\beta}_1 &= S_{xy}\end{aligned}$$

where $S_{xx} = \sum_{i=1}^n (x_i - \bar{x})^2$ and $S_{xy} = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$. It follows that

$$\hat{\beta}_0 = \bar{y} \quad \text{and} \quad \hat{\beta}_1 = S_{xy}/S_{xx}$$

[Note differentiating each of the first derivatives of $\mathcal{L}$ a second time, with respect to each of the parameters, yields a Hessian matrix that is positive semi-definite (which, in this case, follows from it being a diagonal matrix with non-negative components), so that we have indeed found a minimum: see https://en.wikipedia.org/wiki/Hessian_matrix and https://en.wikipedia.org/wiki/Hessian_matrix#Second_derivative_test].

Recovering α and β from the estimates from the reparameterized model, we find

$$\hat{\alpha} = \hat{\beta}_0 - \hat{\beta}_1 \bar{x} = \bar{y} - \hat{\beta}\bar{x}$$

and

$$\hat{\beta} = \hat{\beta}_1 = S_{xy}/S_{xx}$$

as noted in Lecture 1.

2. From the formulation given in the question, we find

$$\mathbf{X}^T\mathbf{X} = \begin{pmatrix} n & 0 \\ 0 & \sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}, \quad \text{so that} \quad (\mathbf{X}^T\mathbf{X})^{-1} = \begin{pmatrix} 1/n & 0 \\ 0 & 1/\sum_{i=1}^n (x_i - \bar{x})^2 \end{pmatrix}$$

Also

$$\mathbf{X}^T\mathbf{y} = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n (x_i - \bar{x}) y_i \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n (x_i - \bar{y})(y_i - \bar{y}) \end{pmatrix}$$

so that

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{pmatrix} (1/n)\sum_{i=1}^n y_i \\ (1/\sum_{i=1}^n (x_i - \bar{x})^2)\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) \end{pmatrix} = \begin{pmatrix} \bar{y} \\ S_{xy}/S_{xx} \end{pmatrix}$$

as found in question 1 above.

3.

(a) Let λ be an eigenvalue of $\mathbf{P}$ and $\mathbf{x}$ a corresponding eigenvector.

$$\begin{aligned} \mathbf{P}\mathbf{x} = \lambda\mathbf{x} &\Rightarrow \mathbf{x}^T\mathbf{P}\mathbf{x} = \lambda\mathbf{x}^T\mathbf{x} \\ &\Rightarrow (\mathbf{P}\mathbf{x})^T\mathbf{P}\mathbf{x} = \lambda\mathbf{x}^T\mathbf{x} \\ &\Rightarrow \lambda^2\mathbf{x}^T\mathbf{x} = \lambda\mathbf{x}^T\mathbf{x} \\ &\Rightarrow \lambda(\lambda - 1) = 0. \end{aligned}$$

Thus all the eigenvalues of $\mathbf{P}$ are either 0 or 1.

(b) It follows from (a) that the diagonalized form of $\mathbf{P}$ has ones and zeros along the diagonal. Let r be the number of ones. Recall that rank and trace are invariant under diagonalization. Now r is the number of independent columns of the diagonalized form. Hence rank($\mathbf{P}$)=r. The sum of the diagonal elements of the diagonalized form is also r. Hence tr($\mathbf{P}$)=r.

4. [The following exercise requires you to use R].

(a) Taking logs, we find

$$\begin{aligned}\log(\text{consump}) &= \log\{A \exp(\alpha \text{ price})\}\\ &= \log(A) + \alpha \text{ price}\end{aligned}$$

which is of the correct form for a simple linear regression with intercept $\beta_0 = \log(A)$ and slope $\beta_1 = \alpha$.

(b) The following code can be run in R to obtain a data frame `sugar` which contains the variables `price`, `consump` and `lconsump`. [This assumes that you have the file `sugar.txt` in your *working directory*].

```
> sugar <- read.table("sugar.txt", header = TRUE)

> sugar$lconsump <- log(sugar$consump)
```

The required plots (shown overleaf) can be obtained as follows:

```
> par(mfrow = c(1, 2))
> plot(sugar$price, sugar$consump, main = "consump vs price")
> plot(sugar$price, sugar$lconsump, main = "lconsump vs price")
```

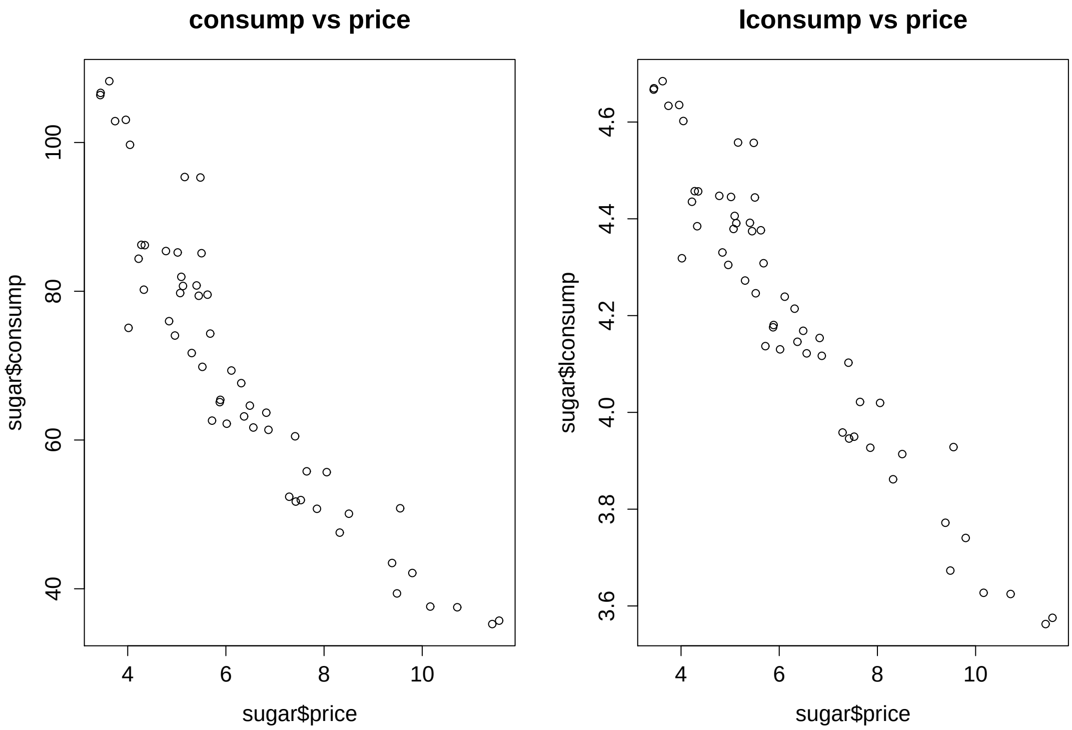
There is clear curvature in the plot of `consump` versus `price`, whereas the plot of `lconsump` versus `price` shows a possibly linear relationship. This confirms that using `lconsump` as the response is more appropriate for a simple linear regression (and conforms with the suggested model in (a)).

(c) The model is obtained and displayed in R as follows:

```
> sugar.lm <- lm(lconsump ~ price, data = sugar)
> summary(sugar.lm)

Call:
lm(formula = lconsump ~ price, data = sugar)

Residuals:
      Min        1Q    Median        3Q       Max
-0.206114 -0.068467  0.004681  0.059175  0.235160

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.081099   0.038202  133.00   <2e-16 ***
price       -0.138536   0.005731  -24.17   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.0877 on 53 degrees of freedom
Multiple R-squared:  0.9168,Adjusted R-squared:  0.9153
F-statistic: 584.3 on 1 and 53 DF,  p-value: < 2.2e-16
```

The fitted model is

$$\widehat{\texttt{lconsump}} = 5.0811 - 0.1385\,\texttt{price}$$



(d) For a price of 6 cents (per pound), we have

$\widehat{\texttt{lconsump}}$ = 5.0811 - 0.1385 $\times$ 6 = 4.2501.

Recalling that the model is for `lconsump` which is the *log* of consumption, the predicted value for `consump` (consumption) is

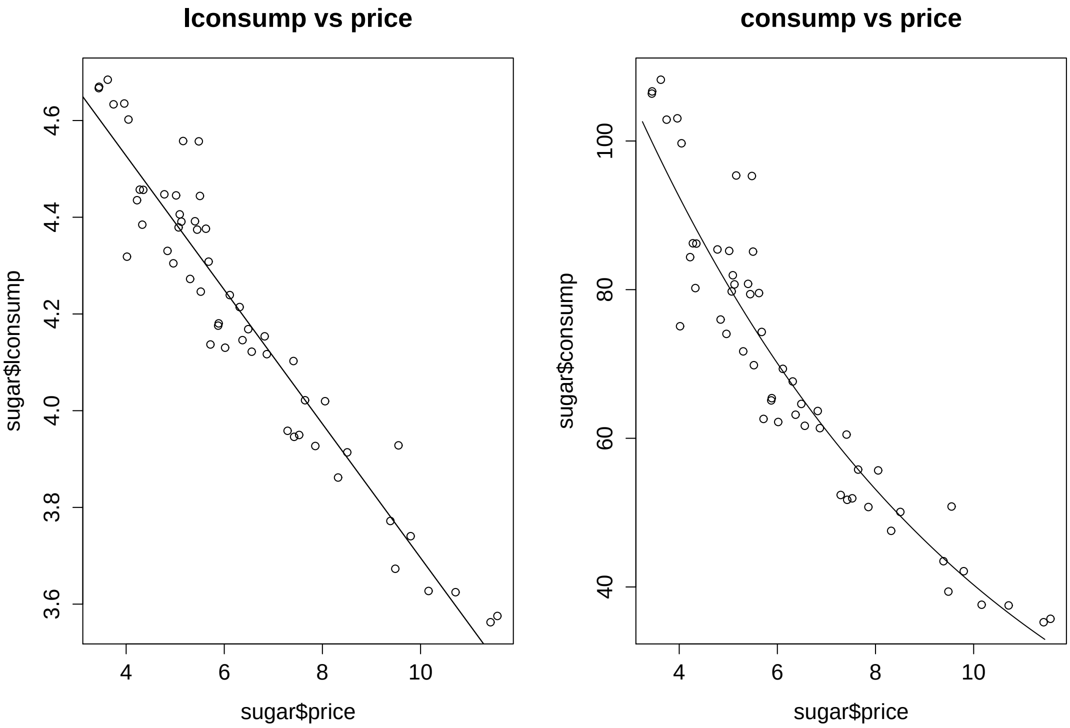
$$\exp(4.2501) = 70.112 \quad \text{(pounds per capita)}$$

(You will see how to make predictions directly using R in Lecture 3).

(e) Plots of the fitted model(s) for `lconsump` and `consump` can be produced as follows:

```
# Plotting the fitted regressions
# lconsump vs price
> par(mfrow = c(1, 2))
> plot(sugar$price, sugar$lconsump, main = "lconsump vs price")
> a <- coef(sugar.lm)[1]
> b <- coef(sugar.lm)[2]
> abline(a, b)  # alternatively > abline(sugar.lm)
```

```
# consump vs price
> plot(sugar$price, sugar$consump, main = "consump vs price")
> x <- seq(3.25, 11.5, 0.1)
> y <- exp(a + b * x)
> lines(x, y)
```



Reading from the second graph: for a price of 6 cents (per pound) we see that the predicted consumption is about 70 pounds per capita as was suggested by the calculation above.

Extension: The estimated regression coefficients are easily obtained 'by-hand' using matrix calculations in R.

Recalling that $\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, to achieve this we must define the design matrix $\mathbf{X}$, which consists of a column of ones and a column containing the values of the single explanatory variable `price`, and the vector $\mathbf{y}$ of values for the response variable `lconsump`:

```
> X <- model.matrix(sugar.lm)
> y <- sugar$lconsump
```

Using `%*%` for matrix multiplication in R we obtain our vector of estimates as follows:

```
> bvec <- solve(t(X) %*% X) %*% t(X) %*% y
> bvec
                 [,1]
(Intercept)  5.0810987
price       -0.1385355
```

which match those given by the `lm` output.

• Try to obtain estimates for the regression parameters in this way for the *full regression* (all four explanatory variables) of the oil example of the Practical Computing Session Notes 2.