Log 708 - Chapter 5 Solutions

Halvard Arntzen

5.1

5.2

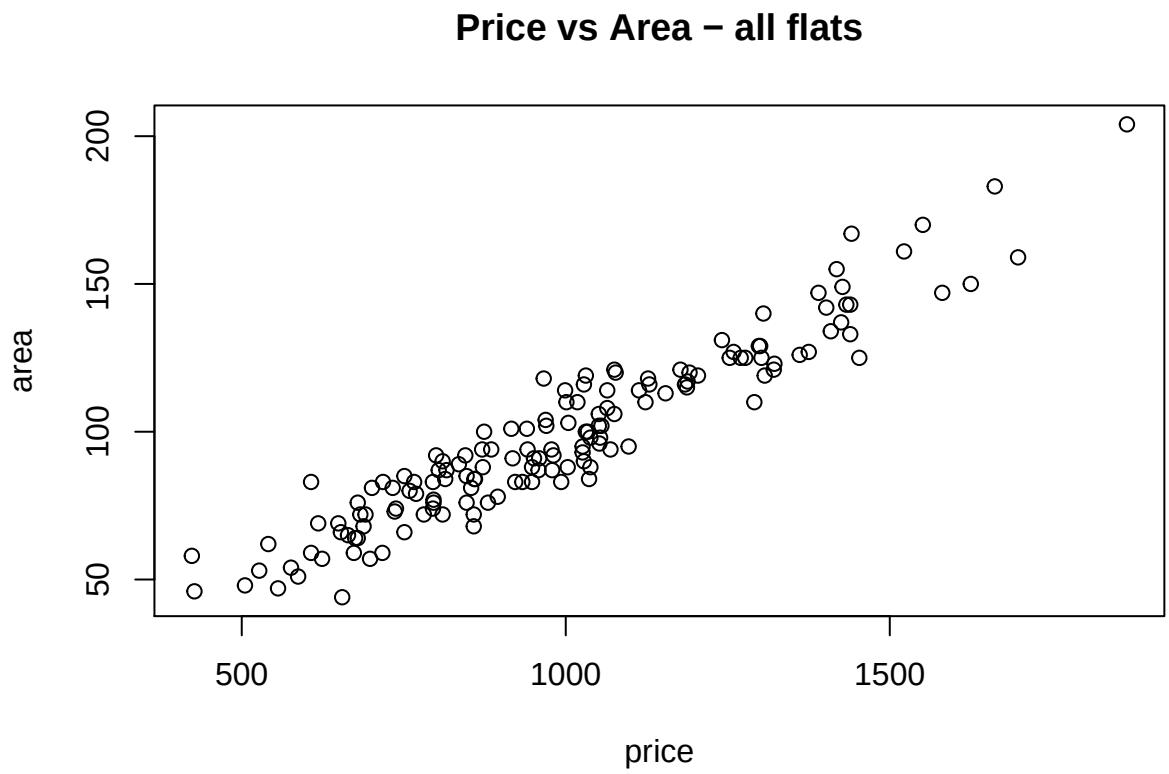
We will work again on the flat prices data, so we start by reading the file:

```
flats <- read.csv("M:/Undervisning/Undervisningh21/Data/flat_prices.csv")
head(flats)
```

```
##   price area rooms standard situated town distcen age rent
## 1  1031  100     3        2        6    1       5  15 2051
## 2  1129  116     3        1        5    1       4  42 2834
## 3  1123  110     3        2        5    1       3  25 2468
## 4   607   59     2        3        5    1       6  25 1940
## 5   858   72     2        3        4    1       1  17 1611
## 6   679   64     2        2        3    1       3  17 2039
```

a. Making a scatterplot.

```
with(flats, plot(price, area,
                 main = "Price vs Area - all flats"))
```



b. The following code should do the job.

```
#run regression
pricereg <- lm(price ~ area, data = flats)

#get coefficients
coef(pricereg)
```

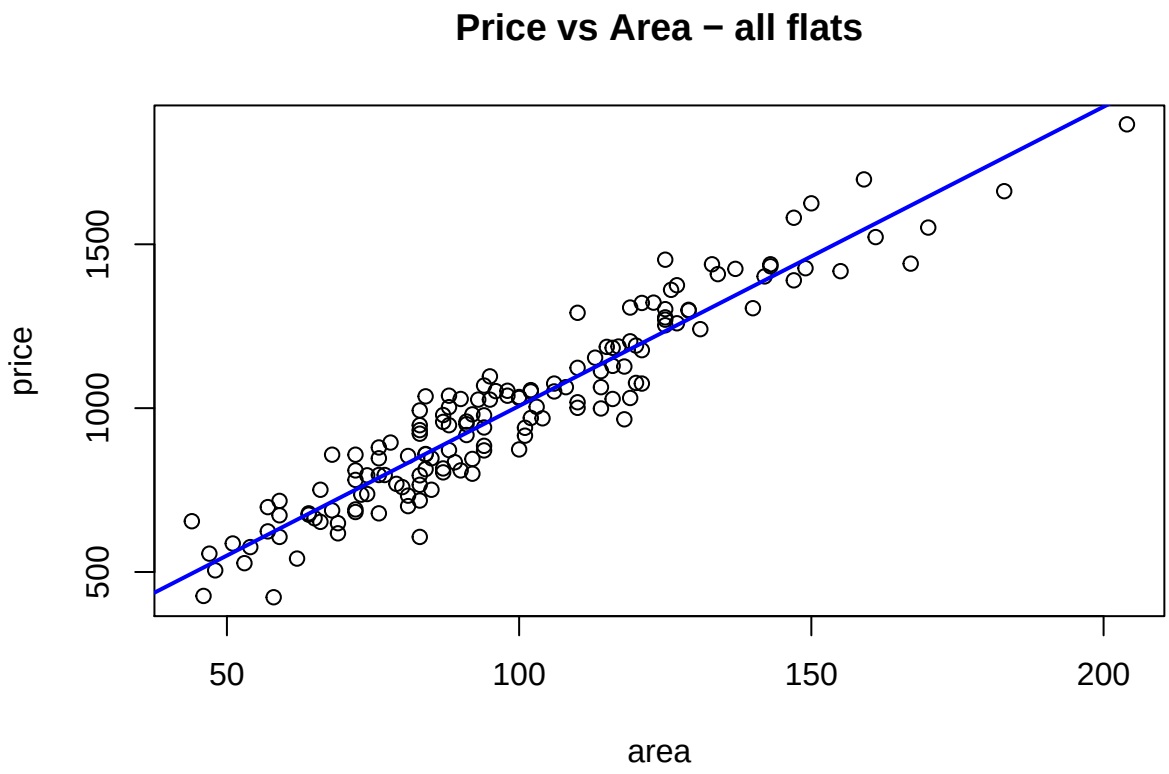
```
## (Intercept)        area
##   93.696532    9.129971
```

```
#get confidence interval
confint(pricereg)
```

```
##                 2.5 %     97.5 %
## (Intercept) 43.637152 143.755912
## area         8.644193   9.615749
```

c. We can do this:

```
with(flats, plot(area, price, main = "Price vs Area - all flats"))
abline(pricereg, col = "blue", lwd = 2)
```



d. The β_1 parameter is generally describing the *marginal effect* on y of the x variable in a regression. That means the average effect on y of increasing x with *one unit*. Here we get the marginal square meter price, i.e. the price for one additional square meter. The overall estimate is $b_1 = 9.13$. Note that this is not identical to what is called "square meter price" in estate markets. Then we talk about the ratio of price to area, and the average in our sample is

```
with(flats, mean(price / area))
```

```
## [1] 10.16392
```

The two values would be equal only if the constant in the regression was 0.

e. We can look at the P-value for testing $H_0 : \beta_1 = 0$. This is practically 0 as shown by a `summary` of the regression. Another key figure is the R^2 here being about 0.90.

```
summary(pricereg)
```

```
##
## Call:
## lm(formula = price ~ area, data = flats)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -244.484  -67.769    1.346   59.306  218.057
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  93.6965    25.3321   3.699 0.000305 ***
## area          9.1300     0.2458  37.140  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 88.52 on 148 degrees of freedom
## Multiple R-squared:  0.9031, Adjusted R-squared:  0.9024
## F-statistic:  1379 on 1 and 148 DF,  p-value: < 2.2e-16
```

f. As stated, we find $R^2 = 0.90$.

g. We find $S_e = 88.52$. This number describes prediction precision since e.g. $2S_e$ approximates a 95% error margin.

h. We can insert $x = 100$ in the estimated equation. Then we get

```
pred <- 93.7 + 9.13*100
pred
```

```
## [1] 1006.7
```

As said, we can use $2S_e$ for the error margin, so we can get a prediction interval as follows

```
pred + c(-1, 1) * 2 * 88.5
```

```
## [1]  829.7 1183.7
```

Alternatively, using the `predict` function, we can do

```
df <- data.frame(area = 100)
predict(pricereg, df, interval = "prediction")
```

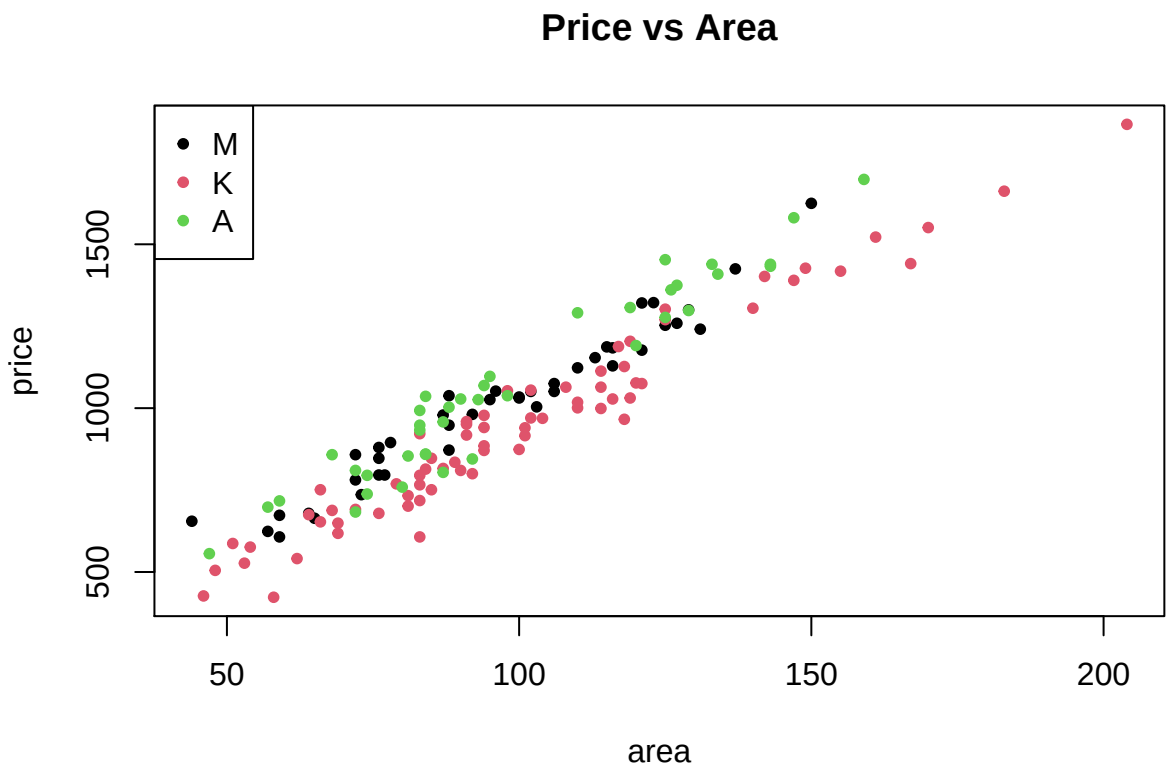
```
##        fit      lwr      upr
## 1 1006.694 831.1867 1182.201
```

We get essentially the same result.

5.3

Ok, my dataframe from the previous execise is called `flats` so we go

```
#make plot, col = "color", pch = "point character",
#we want solid dots (pch = 20)
with(flats, plot(area, price, col = town, pch = 20,
              main = "Price vs Area"))
legend("topleft", legend=c("M", "K","A"), col = c(1, 2, 3), pch = 20)
```



It appears that Ålesund and to some extent Molde has a slightly higher general level, while the slope looks very similar for the three markets.

b. We will take the liberty to recode the `town` variable directly into a factor, which makes the plot labels come out as text. This is best done with the help of a `recode` function from the `dplyr` library. (Note that this is not *necessary* for solving the problem, just a cosmetic upgrade :-))

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'
```

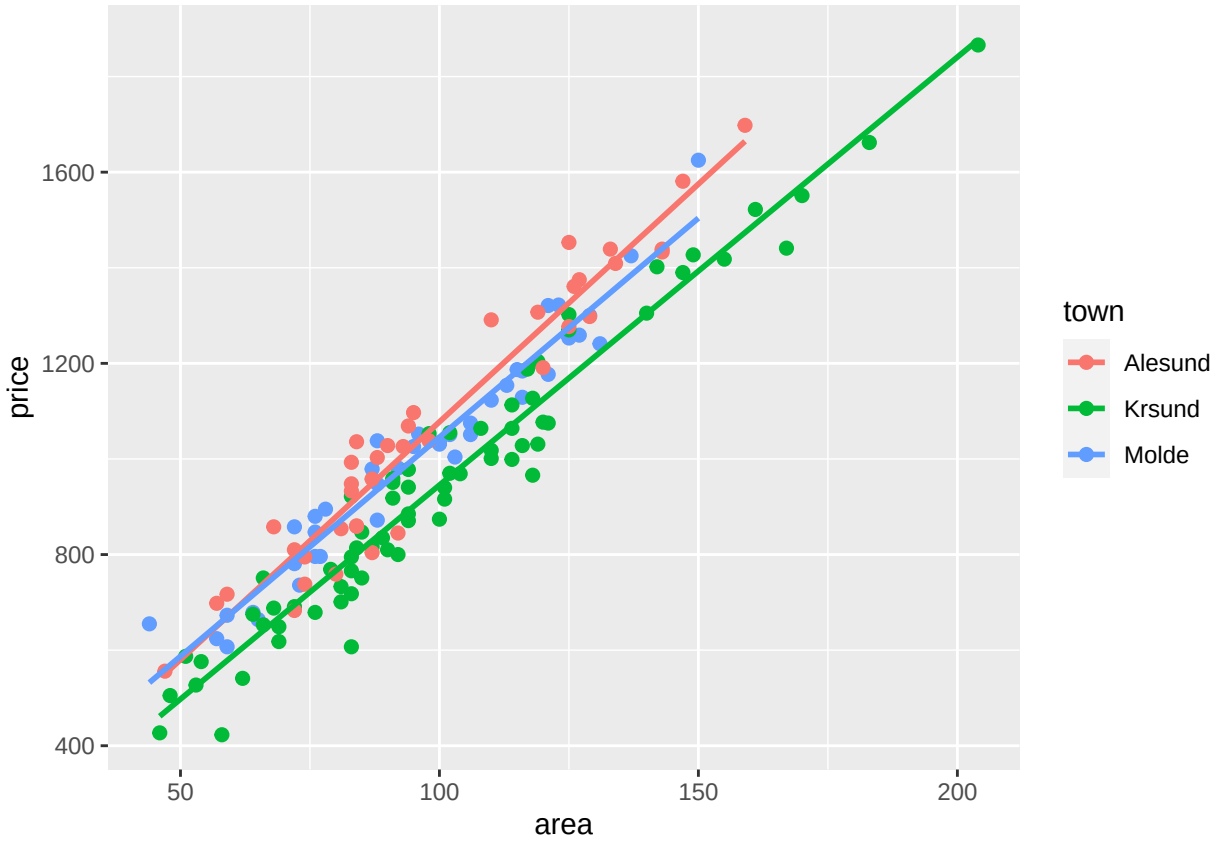
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
```

```
##     intersect, setdiff, setequal, union
```

```
#redefine as character:
flats$town <- as.character(flats$town)
#recode as names
flats$town <- recode(flats$town, "1" = "Molde", "2" = "Krsund", "3" = "Alesund")
#convert to "factor" type
flats$town <- factor(flats$town)

# make plot, (now we do not need the "as.factor..." code suggested in the exercise.)
ggplot(flats, aes(x = area, y = price, color = town)) +
  geom_point(size = 2) +
  geom_smooth(method = lm, se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



The lines are almost parallell, and there is a vertical shift leaving Molde and Ålesund at a generally higher price level, while the marginal square meter price is almost equal in all three markets.

c. We can follow the suggested method. (adjusting for the fact that I made `town` a factor above.) Note: Here it is important to use "==" in the logical expression.

```
regMolde <- lm(price ~ area, data = subset(flats, town == "Molde"))
regKrsund <- lm(price ~ area, data = subset(flats, town == "Krsund"))
regAlesund <- lm(price ~ area, data = subset(flats, town == "Alesund"))

coef(regMolde)
```

```
## (Intercept)        area
##  128.686384    9.167775
```

```
coef(regKrsund)
```

```
## (Intercept)        area
##   49.780467    8.954979
```

```
coef(regAlesund)
```

```
## (Intercept)        area
##   82.548319    9.950017
```

d. We get respectively 9.17, 8.95, 9.95 estimated for β_1 in M, K, A.

e. The confidence intervals:

```
confint(regMolde)
```

```
##                 2.5 %     97.5 %
## (Intercept) 60.284118 197.088651
## area         8.477614   9.857936
```

```
confint(regKrsund)
```

```
##                 2.5 %     97.5 %
## (Intercept) -2.337113 101.898048
## area         8.461546   9.448412
```

```
confint(regAlesund)
```

```
##                 2.5 %    97.5 %
## (Intercept) -5.887579 170.98422
## area         9.086685  10.81335
```

The confidence intervals tell us where we are likely to find the *true* parameter values in each town. Since all pairs of intervals do overlap, we can not from this see hard evidence for real differences in β_1 values.

Optional. We can end this section with an example of how one could solve such problem with some modern mechanisms in R. What happens above (and below) is a VERY common task in data analysis. We have a data frame (`flats`). We want to look at subgroups (by `town`). We want to do things to subgroups, i.e. compute mean values of variables, run a regression (on the subsets) and so on. The "%>%" sign is called the "pipe", and it works

by sending the result of calculations to the next step in a sequence. So below we start with `flats`. Send it to `group_by(town)` which makes the subgroups. Then send the subgroups to `summarize` which in this case do a few things: Compute min, mean and max price, the number of flats, and report the b_1 estimate from the above regression. Note that the "%>%" is not part of base R, but will work after activating e.g. `dplyr` (as we did above.)

```
flats %>% group_by(town) %>%
  summarize(minprice = min(price),
            meanprice = mean(price),
            maxprice = max(price),
            N = n(),
            b_1 = coef(lm(price ~ area))[2])
```

```
## # A tibble: 3 x 6
##   town    minprice meanprice maxprice     N   b_1
##   <fct>      <int>     <dbl>    <int> <int> <dbl>
## 1 Alesund      556     1065.     1698    39  9.95
## 2 Krsund       423      950.     1866    70  8.95
## 3 Molde        607     1008.     1625    41  9.17
```

This example is just meant to show how effectively R can be applied to more or less complex data analysis tasks. It is not expected in general that Log 708 students can operate at this level yet.

5.4

Ok, we need to read the data,

```
wdata <- read.csv("M:/Undervisning/Undervisningh21/Data/WaterWorld.csv")
head(wdata)
```

```
##   X Tickets Price
## 1 1    1765   120
## 2 2    2344    90
## 3 3    2352    80
## 4 4    2055    95
## 5 5    1918   110
## 6 6    1311   135
```

the data seems to include a redundant column. We can leave it there or remove it e.g. as follows

```
#remove column X
wdata <- subset(wdata, select = -X)
head(wdata)
```

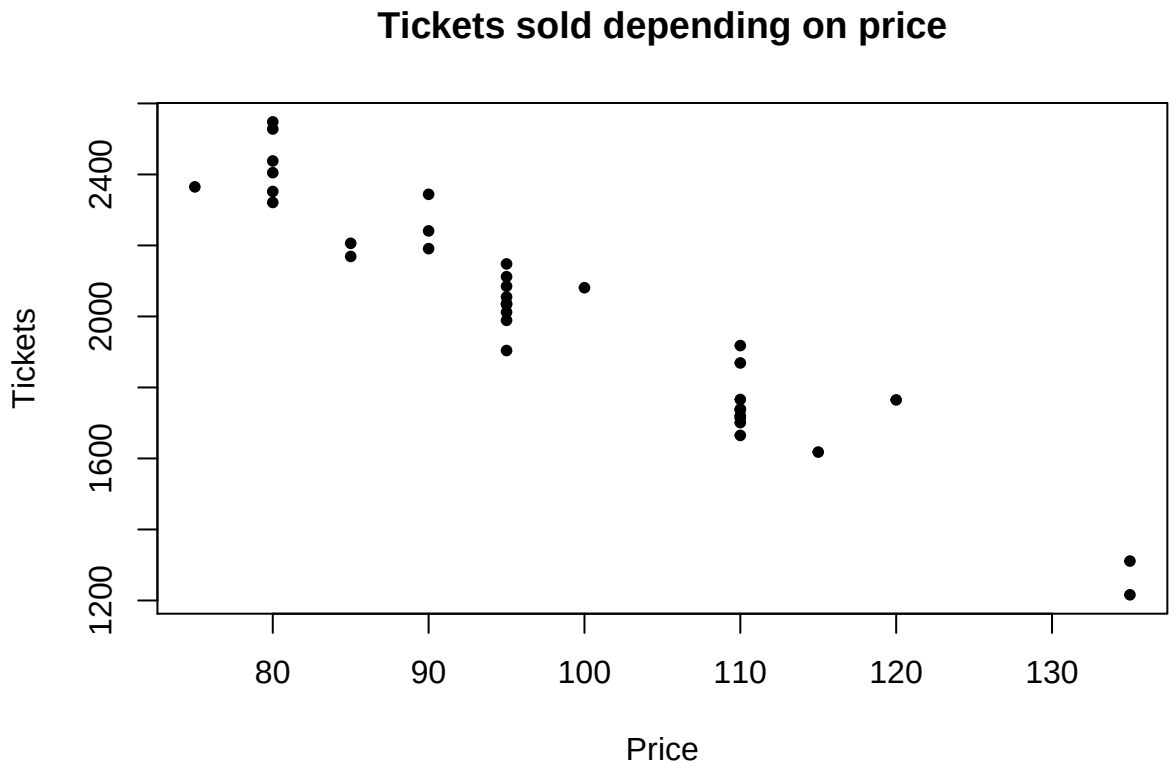
```
##   Tickets Price
## 1    1765   120
```

```
## 2    2344    90
## 3    2352    80
## 4    2055    95
## 5    1918   110
## 6    1311   135
```

Another way to remove a column would be `wdata <- wdata[ , -1]` which selects everything except the first column.

a. We are getting used to this now, so:

```
with(wdata, plot(Price, Tickets,
                 main = "Tickets sold depending on price",
                 pch = 20))
```



From the scatterplot, a linear model looks reasonable. Economic theory usually suggest a non-linear relation of the form $y = Cx^{-}\alpha$, where C, α are constants. (See chap. 8 in compendium for more about this.) In many applications a linear function is a sufficiently good approximation.

b. We estimate a linear model, and check the coefficients.

```
wreg <- lm(Tickets ~ Price, data = wdata)
coef(wreg)
```

```
## (Intercept)       Price
##  3998.09607   -20.21201
```

So in the form $y = a - bx$ we use $a = 3998, b = 20.21$.

c. This is an estimated marginal effect of `Price` on `Tickets` so an additional NOK in price is estimated to lead to about 20 less tickets sold.

d. "Mathematically" the value $x = 0$ leads to a prediction $\hat{y} = a$. However, all our data have x values very far from 0, so a prediction so far outside of the observed range is not valid. We can not use the estimated model at $x = 0$. Economical reasoning would point to other "costs" when the price is 0, like waiting times for access, the facility being overfull etc., where such generalized costs would limit the demand.

e. We can ask our regression object about it. Either a full `summary`:

```
summary(wreg)
```

```
##
## Call:
## lm(formula = Tickets ~ Price, data = wdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -173.96  -60.45  -29.14   59.42  192.34
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3998.096    104.214   38.36   <2e-16 ***
## Price        -20.212      1.047  -19.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 92.96 on 33 degrees of freedom
## Multiple R-squared:  0.9187, Adjusted R-squared:  0.9162
## F-statistic: 372.9 on 1 and 33 DF,  p-value: < 2.2e-16
```

Or being more selective:

```
results <- summary(wreg)
results$r.squared
```

```
## [1] 0.9187001
```

```
results$sigma
```

```
## [1] 92.96295
```

Either way, we get $R^2 = 0.92$, and $S_e = 92.7$.

f. We have the estimated equation, so we can just calculate

$$\hat{y} = 3998 - 20.21 \cdot 110 = 1775 \ .$$

The approximate error margin would be $2S_e = 2 \cdot 92.7 = 185.4$. We can also apply the `predict` function to get the prediction along with a prediction interval.

```
testprice <- data.frame(Price = 110)
testpredict <- predict(wreg, testprice, interval = "prediction")
testpredict
```

```
##        fit      lwr      upr
## 1 1774.775 1581.382 1968.169
```

From the difference between `lwr` and `upr` we find that R uses an error margin of $0.5 \cdot (1968 - 1581) = 193.5$. So, our approximation above is slightly too low, but still acceptable.

g. This is fairly straightforward, as each customer generates expected income $x + s$ and expected cost c. Multiply this by expected number of customers (as given by $y = a - bx$) We get expected operational income $R(x)$ and operational cost $C(x)$ as follows.

$$R(x) = y(x + s) = (a - bx)(x + s) = ax + as - bx^2 - bsx \ ,$$

$$C(x) = yc + f = (a - bx)c + f = ac - bcx + f \ .$$

Then the operational profit is as follows

$$\begin{aligned} P(x) &= R(x) - C(x) \\ &= ax + as - bx^2 - bsx - (ac - bcx + f) \\ &= -bx^2 + (a - bs + bc)x + as - ac - f \end{aligned}$$

h. The profit is a quadratic function of x. We maximize it simply by solving the equation $P'(x) = 0$. This becomes

$$P'(x) = -2bx + (a - bs + bc) = 0 \ ,$$

for which the only solution is $x = x^*$ as given in the exercise.

i. Using the expression for x^* with the given values for s, c and f as well as previously found estimates for a, b, we find the optimal price at $x^* = 96.40$. The expected number of customers with this price is 2050, and the expected profit is $P(x^*) = 57826$. Putting the price higher, at $x = 110$ would lead to an expected loss of about 3740 NOK per week.

j. The prices we consider as realistic is then rounding down to 95 or up to 100. Compared to the optimum, we can expect to lose only 40 NOK per week at price 95, while we lose 260 NOK per week at price 100. That means $x = 95$ would be the suggested price.

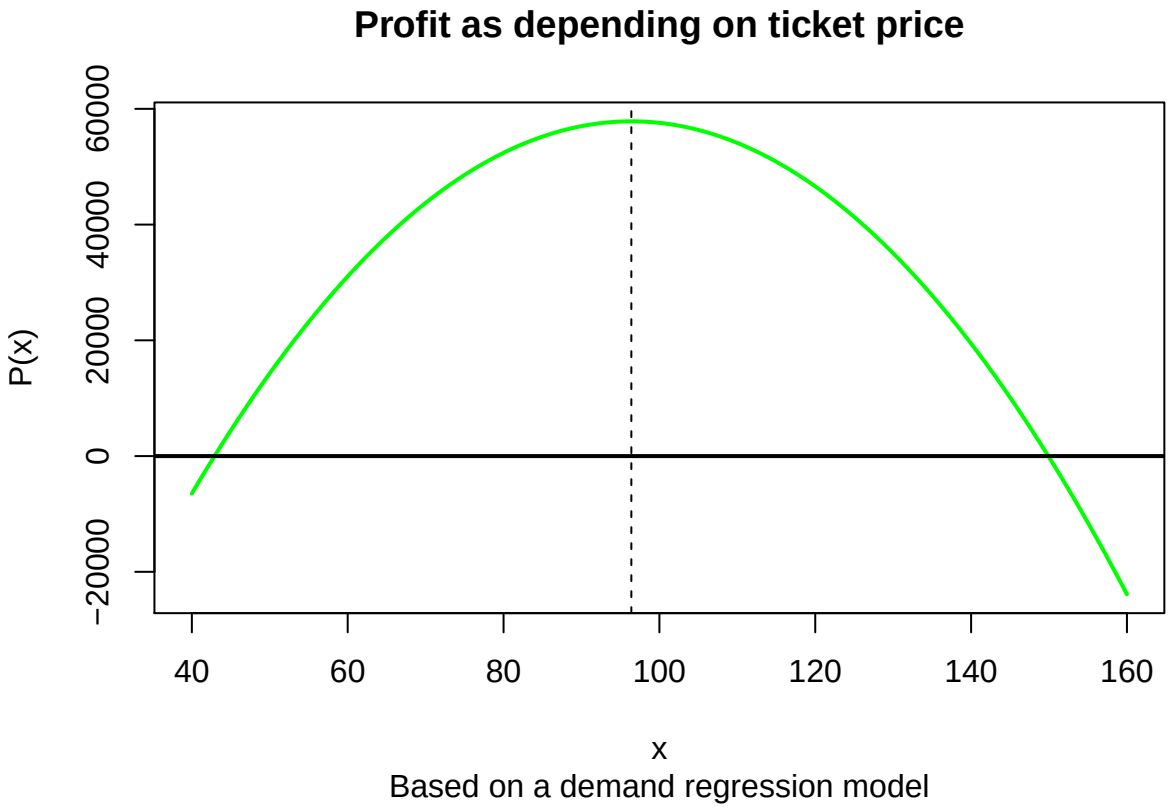
We can round off here with an example of how we might use R to assist in such calculations as in i. and j. We can use the `function` construction to define the profit as a function of a variable `x`. Then this function can be used with various input, and we can make a plot. We will use the general parameters $a, b, ...$ as part of the function, to make it more general.

```
#set parameter values.
a <- 3998
b <- 20.21
s <- 10
c <- 5
f <- 150000
#define function "P(x)" as deduced in g.

P <- function(x) { -b*(x^2) + (a - b*s + b*c)*x + a*s - a*c - f }

#compute a sample of values for an x vector:
x <- seq(from = 85, to = 115, by = 5)
data.frame(price = x, tickets = a - b*x, profit = P(x))
```

```
##   price tickets  profit
## 1    85 2280.15 55213.5
## 2    90 2179.10 57014.5
## 3    95 2078.05 57805.0
## 4   100 1977.00 57585.0
## 5   105 1875.95 56354.5
## 6   110 1774.90 54113.5
## 7   115 1673.85 50862.0
```

So at the (close to) optimal price 95, we expect to sell 2078 tickets earning a profit of 57800 NOK.We can plot the function, like

```
curve(P(x), from = 40, to = 160, n = 100,
      lwd = 2, col = "green",
      main = "Profit as depending on ticket price",
      sub = "Based on a demand regression model")

abline(h = 0, lwd = 2)
abline(v = 96.4, lty = 2)
```



So, here we see the "whole story" in a picture, profits occur in a fairly wide price range, with a theoretical optimum at 96.4 NOK, (dashed line)