Multivariate Analysis for the Behavioral Sciences, Second Edition (Chapman and Hall/CRC, 2019) **Solutions to Exercises of Chapter 3: Simple Linear and Locally Weighted Regression**

Kimmo Vehkalahti and Brian S. Everitt

1 October 2018

Solutions

Exercise 3.1

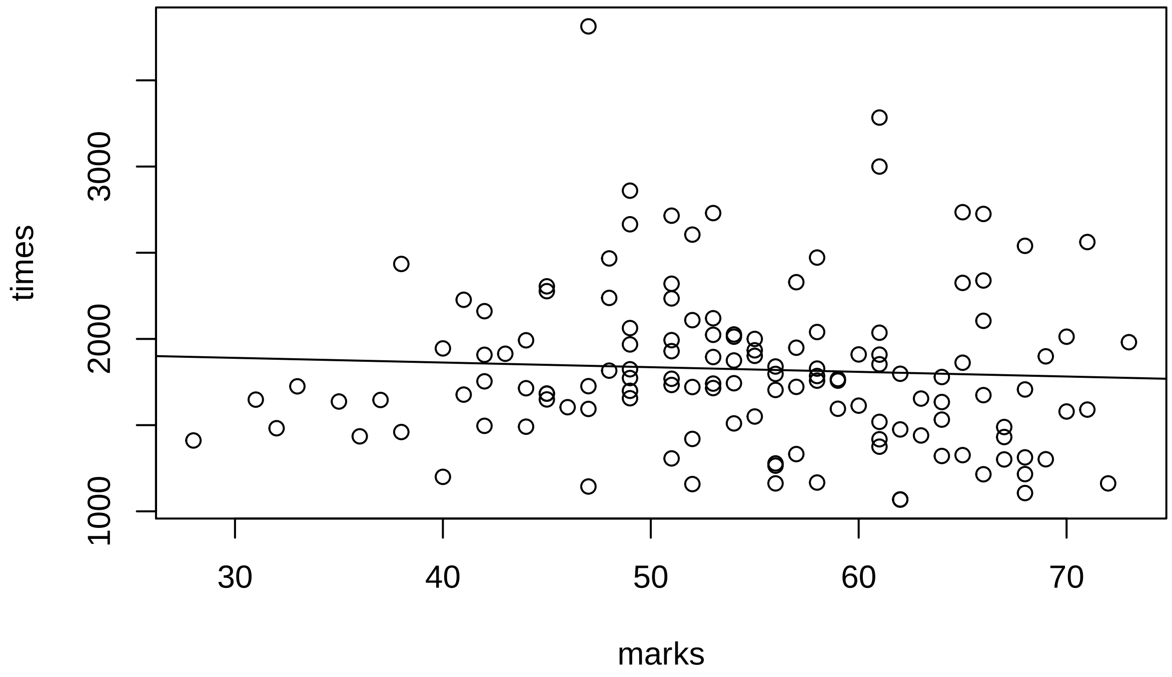
Exercise 3.2

```r
# The final examination scores and corresponding exam completion times:
exam <- structure(list(
    marks = c(49, 49, 70, 55, 52, 55, 61, 65, 57,
        71, 49, 48, 49, 69, 44, 53, 49, 52, 53, 36, 61, 68, 67, 53, 33,
        64, 57, 56, 41, 40, 42, 40, 51, 53, 62, 61, 49, 54, 57, 71, 45,
        70, 58, 62, 28, 72, 37, 67, 51, 55, 68, 58, 61, 43, 60, 53, 51,
        51, 60, 64, 66, 52, 45, 48, 51, 73, 63, 32, 59, 68, 35, 64, 62,
        51, 52, 44, 64, 65, 56, 52, 59, 66, 42, 67, 48, 56, 47, 68, 58,
        59, 45, 31, 47, 56, 38, 47, 65, 61, 45, 63, 66, 44, 57, 56, 56,
        54, 61, 58, 46, 62, 68, 58, 47, 66, 61, 58, 45, 55, 54, 54, 54,
        41, 65, 66, 38, 51, 49, 49, 51, 42, 61, 69, 42, 53),
    times = c(2860, 2063, 2013, 2000, 1420, 1934, 1519, 2735, 2329, 1590,
        1699, 1816, 1824, 1899, 1714, 1741, 1968, 1721, 2120, 1435,
        1909, 1707, 1431, 2024, 1725, 1634, 1949, 1278, 1677, 1945,
        1754, 1200, 1307, 1895, 1798, 1375, 2665, 1743, 1722, 2562,
        2277, 1579, 1785, 1068, 1411, 1162, 1646, 1489, 1769, 1550,
        1313, 2472, 2036, 1914, 1910, 2730, 2235, 1993, 1613, 1532,
        2339, 2109, 1649, 2238, 1733, 1981, 1440, 1482, 1758, 2540,
        1637, 1779, 1069, 1929, 2605, 1491, 1321, 1326, 1797, 1158,
        1595, 2105, 1496, 1301, 2467, 1265, 3813, 1216, 1167, 1767,
        1683, 1648, 1144, 1162, 1460, 1726, 1862, 3284, 1683, 1654,
        2725, 1992, 1332, 1840, 1704, 1510, 3000, 1758, 1604, 1475,
        1106, 2040, 1594, 1215, 1418, 1828, 2305, 1902, 2013, 2026,
        1875, 2227, 2325, 1674, 2435, 2715, 1773, 1656, 2320, 1908,
        1853, 1302, 2161, 1715)),
    .Names = c("marks", "times"), row.names = c(NA, -134L), class = "data.frame")
```

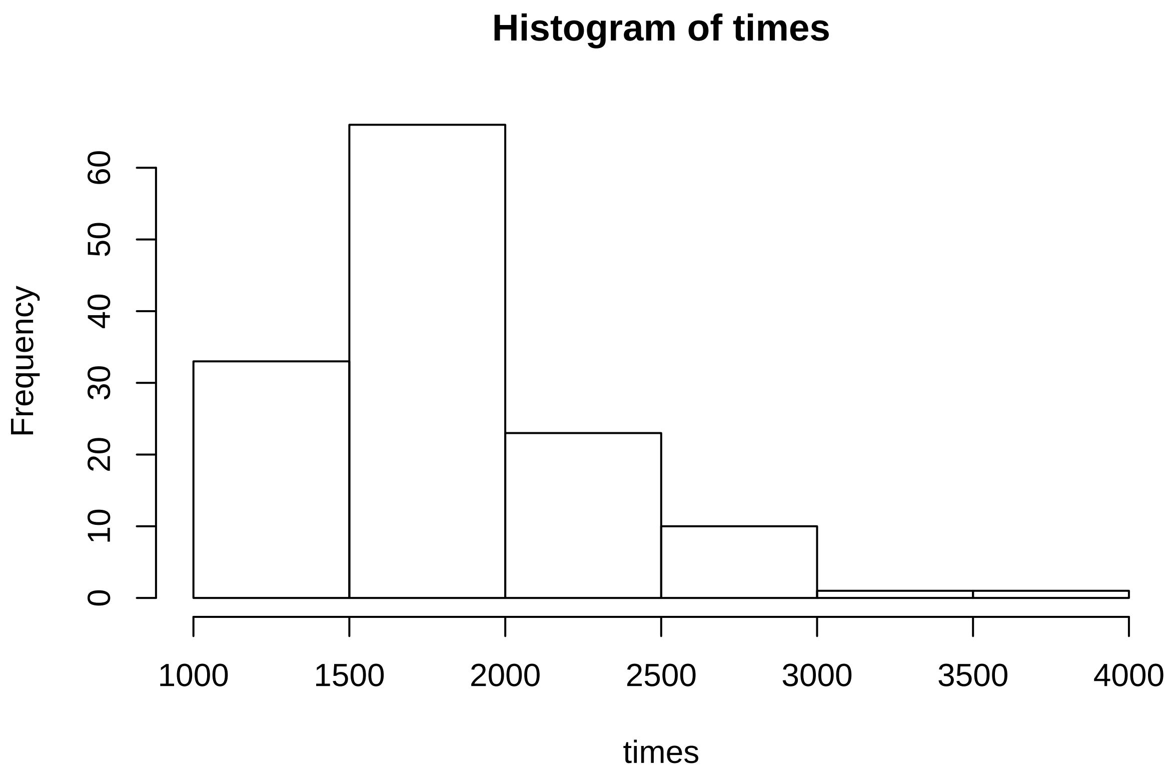
```r
head(exam)
```

```
##   marks times
## 1    49  2860
## 2    49  2063
## 3    70  2013
## 4    55  2000
## 5    52  1420
## 6    55  1934
```

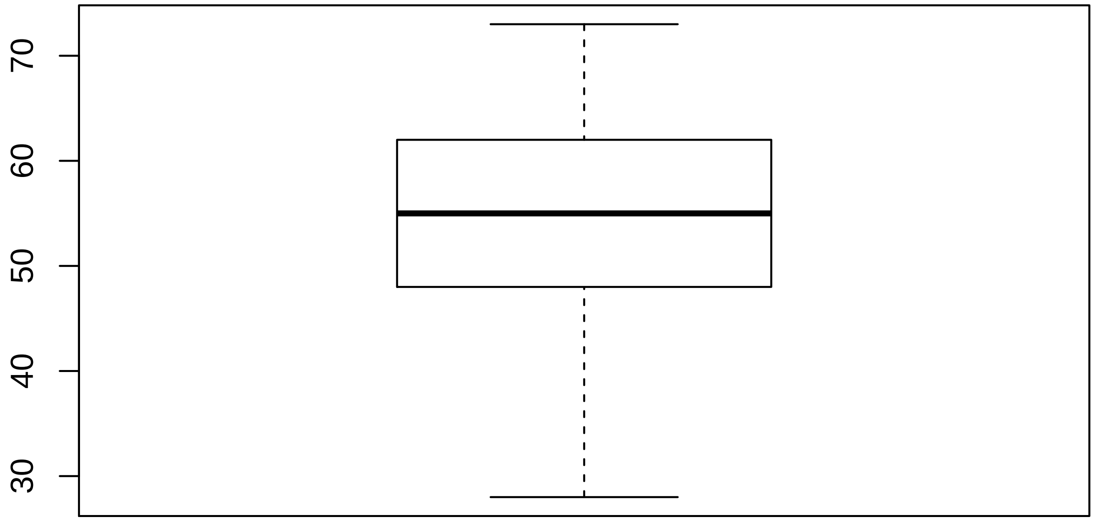
```r
attach(exam)
#plot data
# layout(matrix(c(2,0,1,3), 2, 2, byrow=TRUE), c(2,1), c(1,2), TRUE)
plot(marks, times)
abline(lm(times ~ marks))
```



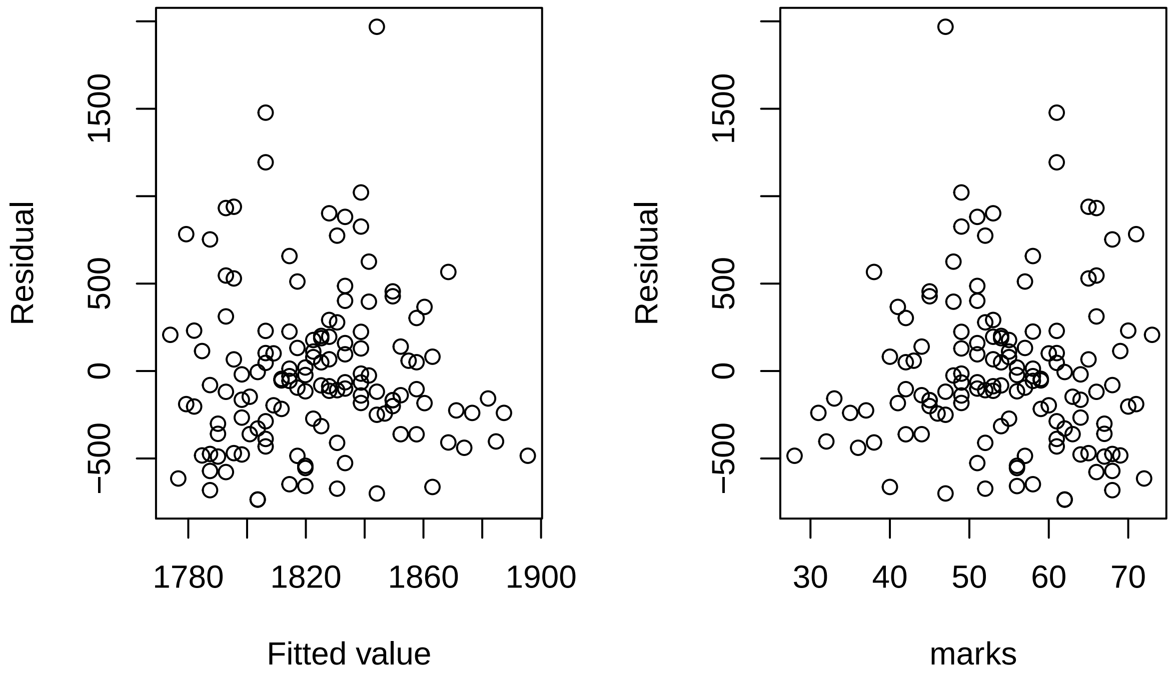
```r
hist(times)
```


Histogram of times
Frequency
0
10
20
30
40
50
60
1000
1500
2000
2500
3000
3500
4000
times

```
boxplot(marks)
```



```
exam_reg <- lm(times ~ marks)
pred <- predict(exam_reg)
resd <- residuals(exam_reg)
par(mfrow = c(1,2))
plot(pred, resd, xlab = "Fitted value", ylab = "Residual")
plot(marks, resd, ylab = "Residual")
```



```
detach(exam)
```

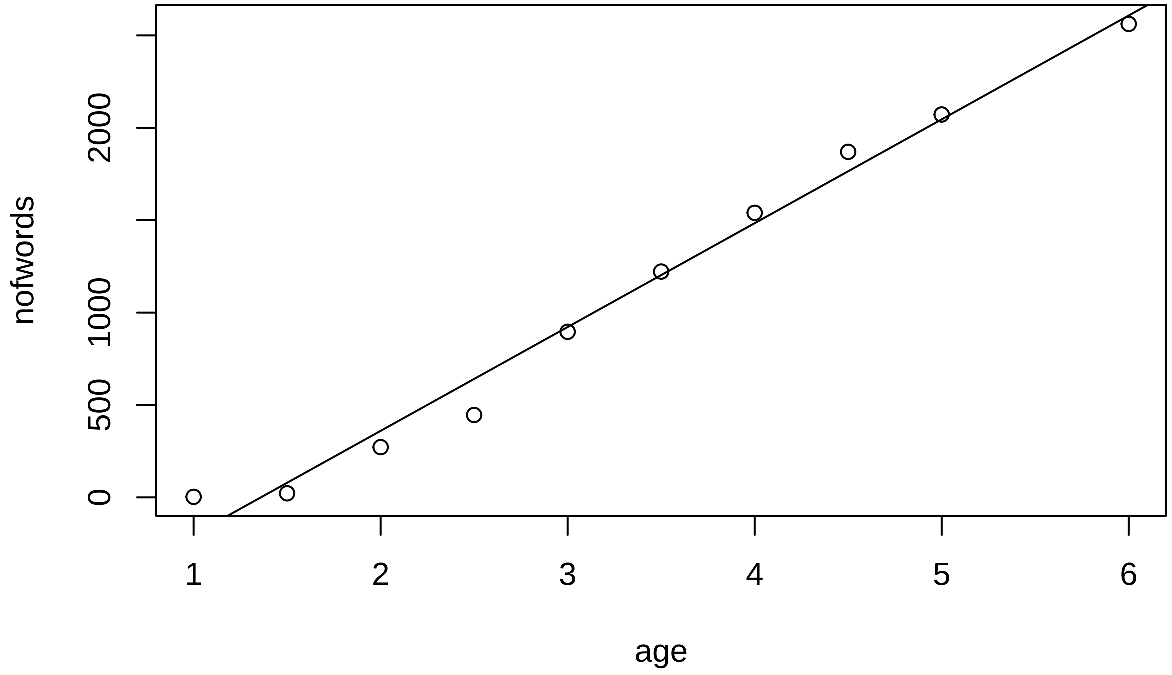
The residual plots show some large positive residuals. A probability plot of residuals may be helpful, and then, perhaps a log transform of the response might be worth investigating.

Exercise 3.3

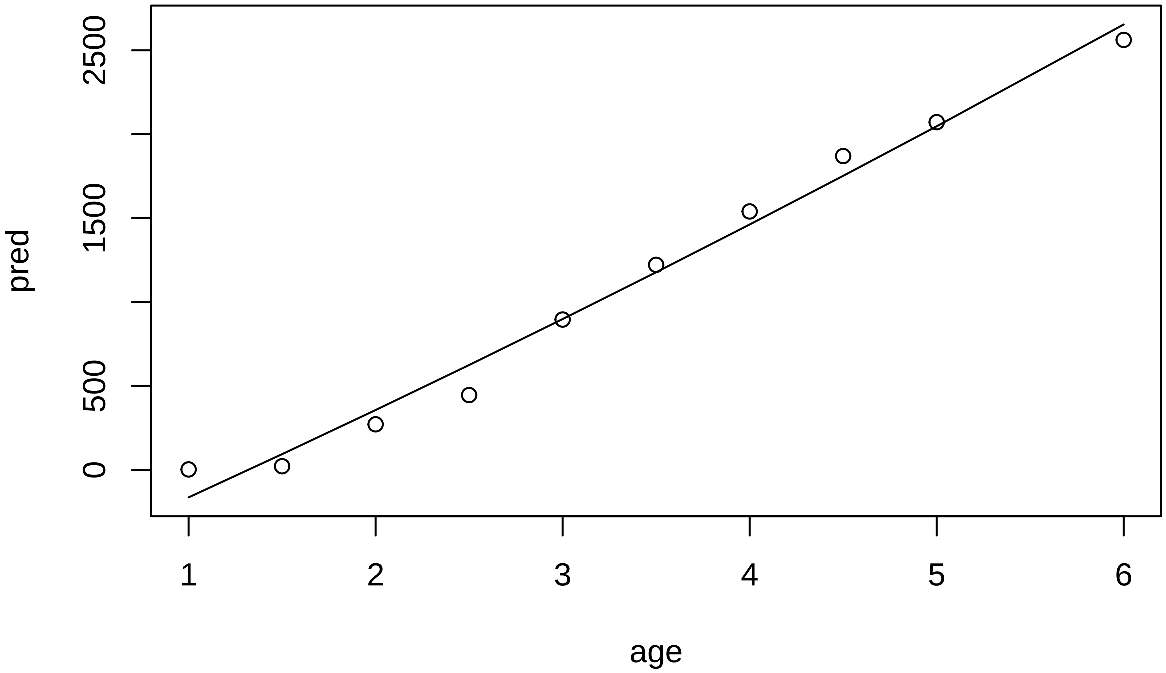
```
# Average vocabulary size of children at various ages:
vocab <- structure(
    list(age = c(1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 6),
    nofwords = c(3, 22, 272, 446, 896, 1222, 1540, 1870, 2072, 2562)),
   .Names = c("age", "nofwords"), row.names = c(NA, -10L), class = "data.frame")

vocab
```

```
##    age nofwords
## 1  1.0        3
## 2  1.5       22
## 3  2.0      272
## 4  2.5      446
## 5  3.0      896
## 6  3.5     1222
## 7  4.0     1540
## 8  4.5     1870
## 9  5.0     2072
## 10 6.0     2562
```

```
attach(vocab)
#scatterplot
plot(age, nofwords)
abline(lm(nofwords ~ age))
```



```
# Linear regression with quadratic term for age:
vocab_reg <- lm(nofwords ~ age + I(age*age))
pred <- predict(vocab_reg)
#
plot(age, pred, type = "l")
points(age, nofwords)
```



```
detach(vocab)
```

Exercise 3.4

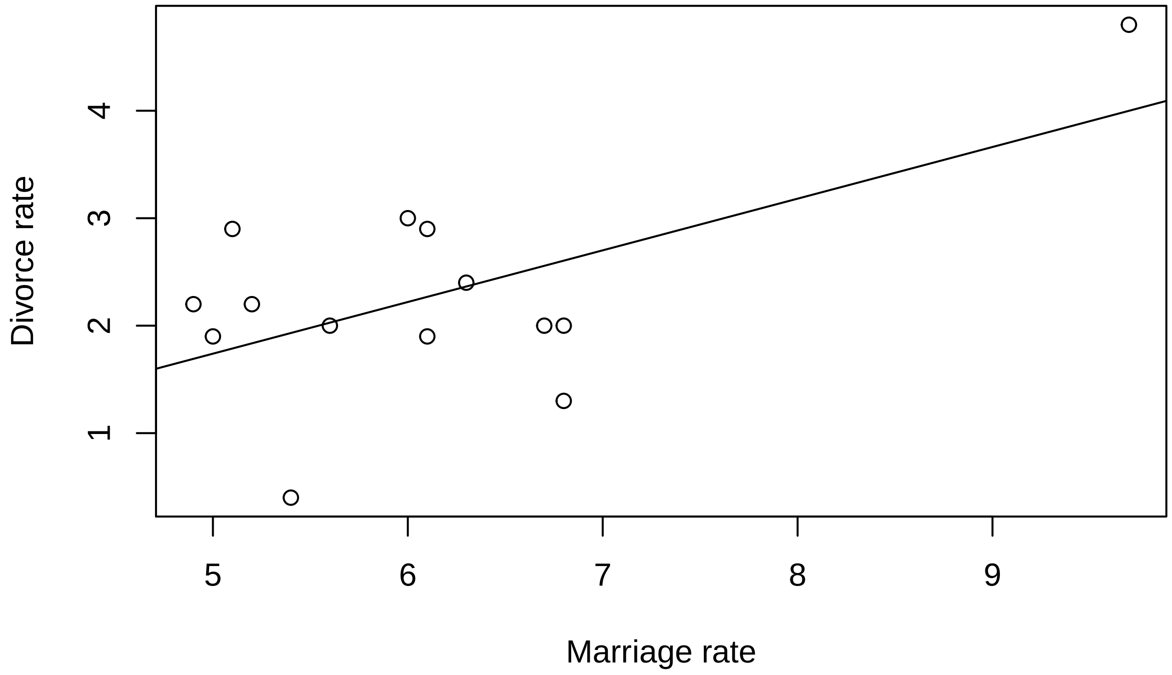
```
# The marriage and divorce rates for 14 countries:
mardiv_rates <- structure(list(
    marrate = c(5.6, 6, 5.1, 5, 6.7, 6.3, 5.4,
       6.1, 4.9, 6.8, 5.2, 6.8, 6.1, 9.7),
    divrate = c(2, 3, 2.9, 1.9, 2, 2.4, 0.4, 1.9,
       2.2, 1.3, 2.2, 2, 2.9, 4.8)),
   .Names = c("marrate", "divrate"), row.names = c(NA, -14L), class = "data.frame")

mardiv_rates
```

```
##    marrate divrate
## 1      5.6     2.0
## 2      6.0     3.0
## 3      5.1     2.9
## 4      5.0     1.9
## 5      6.7     2.0
## 6      6.3     2.4
## 7      5.4     0.4
## 8      6.1     1.9
## 9      4.9     2.2
## 10     6.8     1.3
## 11     5.2     2.2
## 12     6.8     2.0
## 13     6.1     2.9
## 14     9.7     4.8
```

```
attach(mardiv_rates)
mardiv_reg <- lm(divrate ~ marrate)
summary(mardiv_reg)
```

```
##
## Call:
## lm(formula = divrate ~ marrate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.53171 -0.50963  0.09809  0.60097  1.11253
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.6646     1.1633  -0.571   0.5784
## marrate       0.4808     0.1866   2.577   0.0242 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8249 on 12 degrees of freedom
## Multiple R-squared:  0.3562, Adjusted R-squared:  0.3025
## F-statistic: 6.639 on 1 and 12 DF,  p-value: 0.02425
```

```
plot(divrate ~ marrate, xlab = "Marriage rate", ylab = "Divorce rate")
abline(mardiv_reg)
```



```
divpred8 <- -0.6646 + 0.4808 * 8
divpred14 <- -0.6646 + 0.4808 * 14

divpred8; divpred14
```

```
## [1] 3.1818
```

```
## [1] 6.0666
```

```
detach(mardiv_rates)
```

The prediction for a marriage rate of 14 is extrapolating outside the observed range of marriage rates—a procedure fraught with danger! Find the standard errors of both predictions.

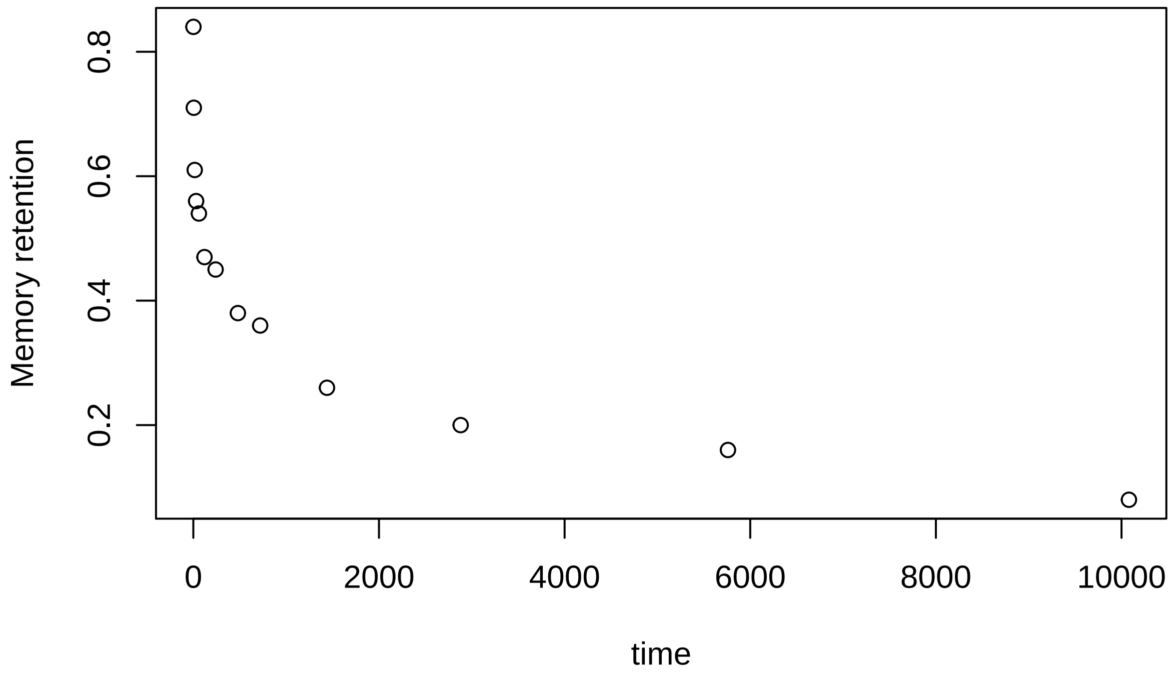
Exercise 3.5

```
# Average percentage memory retention against passing time:
memory <- structure(list(
    time = c(1, 5, 15, 30, 60, 120, 240, 480,
        720, 1440, 2880, 5760, 10080),
  memret = c(0.84, 0.71, 0.61, 0.56, 0.54, 0.47, 0.45,
        0.38, 0.36, 0.26, 0.2, 0.16, 0.08)),
 .Names = c("time", "memret"), row.names = c(NA, -13L), class = "data.frame")

memory
```

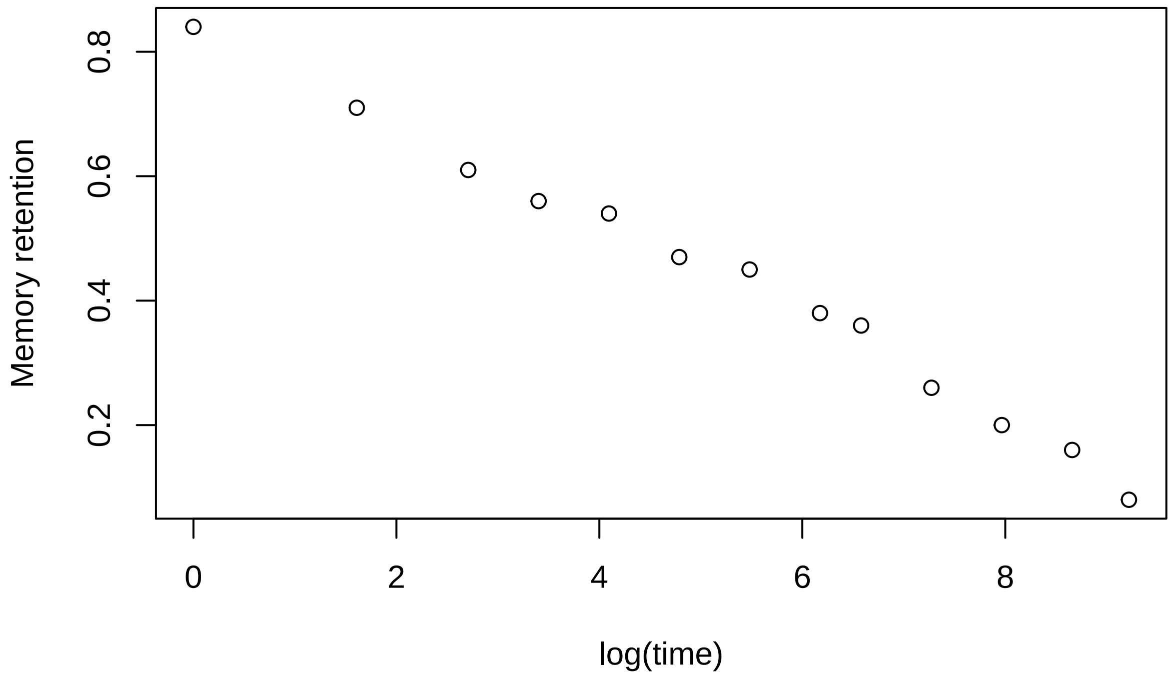
```
##     time memret
## 1      1   0.84
## 2      5   0.71
## 3     15   0.61
## 4     30   0.56
## 5     60   0.54
## 6    120   0.47
## 7    240   0.45
## 8    480   0.38
## 9    720   0.36
## 10  1440   0.26
## 11  2880   0.20
## 12  5760   0.16
## 13 10080   0.08
```

```
attach(memory)
```

```r
plot(time, memret, ylab = "Memory retention")
```



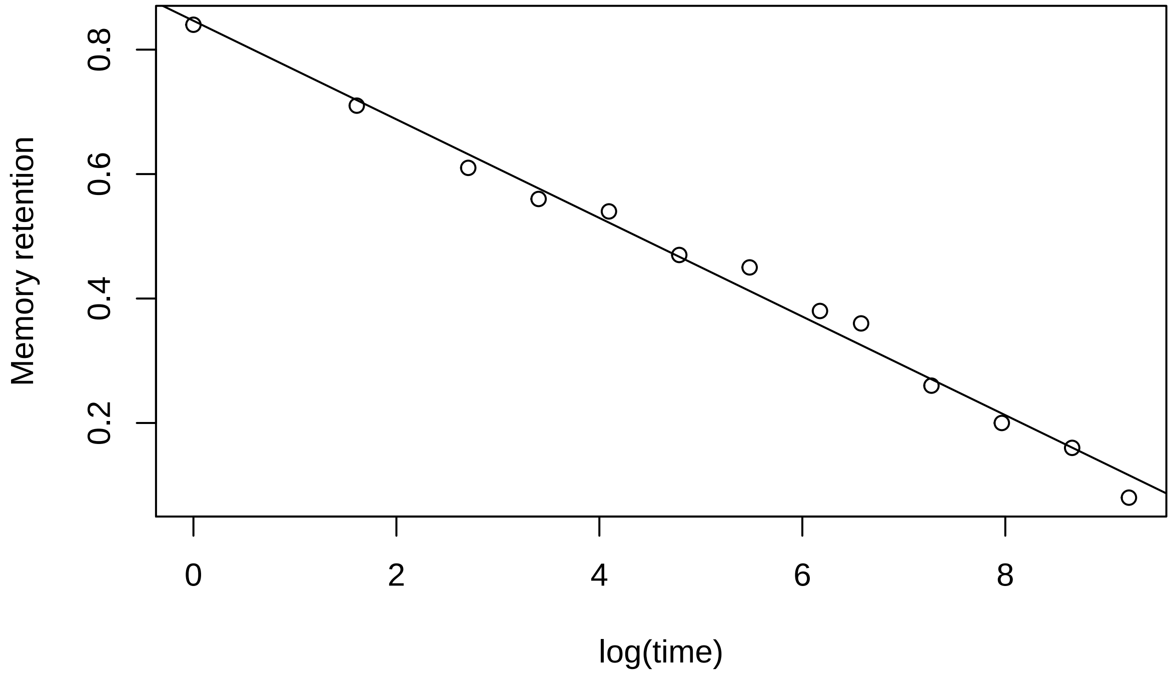
```r
plot(log(time), memret, ylab = "Memory retention")
```



```r
memory_reg <- lm(memret ~ log(time))
summary(memory_reg)
```

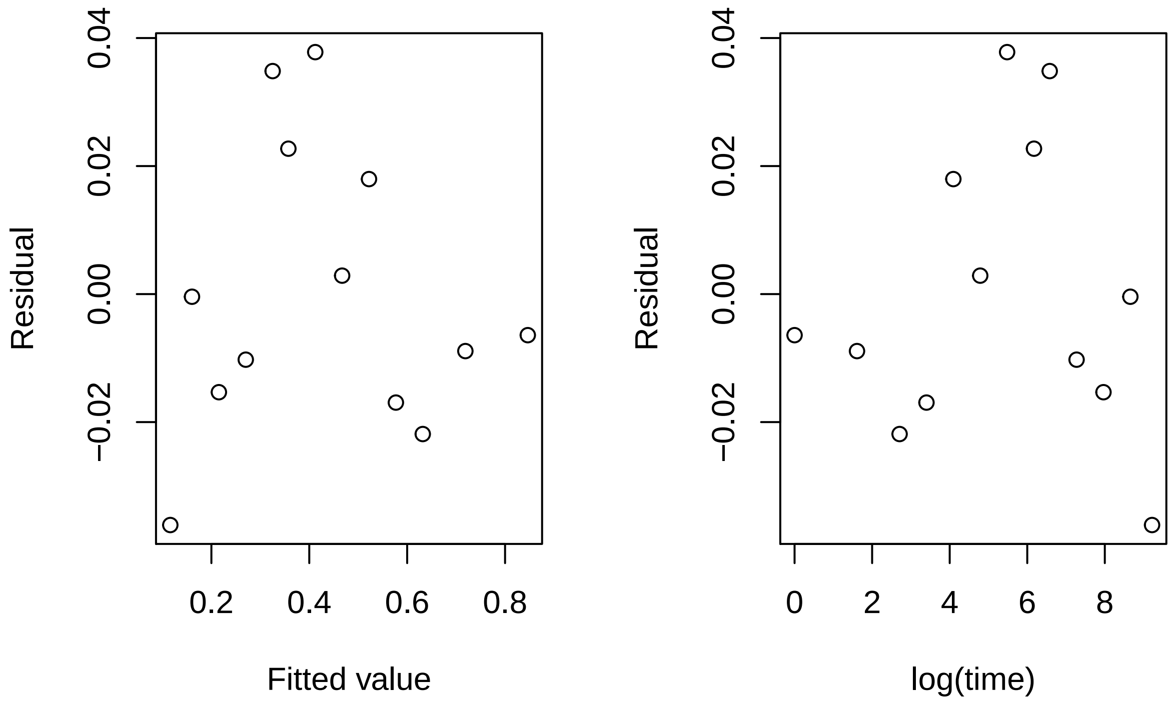
```
##
## Call:
## lm(formula = memret ~ log(time))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.036077 -0.015330 -0.006415  0.017967  0.037799
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.846415   0.014195   59.63 3.65e-15 ***
## log(time)   -0.079227   0.002416  -32.80 2.53e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02339 on 11 degrees of freedom
## Multiple R-squared:  0.9899, Adjusted R-squared:  0.989
## F-statistic:  1076 on 1 and 11 DF,  p-value: 2.525e-12
```

```
plot(memret ~ log(time), xlab = "log(time)", ylab = "Memory retention")
abline(memory_reg)
```



```
pred <- predict(memory_reg)
resd <- residuals(memory_reg)
```

```r
par(mfrow = c(1,2))
plot(pred, resd, xlab = "Fitted value", ylab = "Residual")
plot(log(time), resd, ylab = "Residual")
```



```r
detach(memory)
```