STAT 210
Applied Statistics and Data Analysis:
Homework 8 - Solution

Due on November 23/2025

You cannot use artificial intelligence tools to solve this homework.

Show complete solutions to get full credit. Writing code is not enough to answer a question. Your comments are more important than the code. Do not write comments in chunks. Label your graphs appropriately

For all tests in this homework use a significance level of $\alpha = 0.02$.

Question 1

A labor economist studies weekly wages (`Y`) and how they depend on several worker attributes. The researcher collected a sample of workers and recorded 7 candidate predictors. Your job is to build a regression model. The data is available in the file `HW825FQ1.csv`.

```
dat1 <- read.csv('HW825FQ1.csv')
str(dat1)
```

```
## 'data.frame':    150 obs. of  8 variables:
##  $ Y : num  28.3 17.1 18.9 16.8 18.3 ...
##  $ X1: num  13.28 12.05 10.44 9.74 13.32 ...
##  $ X2: num  24.2 21.8 18.5 19.5 24.7 ...
##  $ X3: num  36.3 34.5 27.9 29.3 38.5 ...
##  $ X4: num  5.3 5.58 3.26 4.92 7.55 5.76 5.13 5.68 5.1 4.77 ...
##  $ X5: num  4.74 4.13 2.98 2.87 6.59 4.89 4.65 5.06 3.3 3.81 ...
##  $ X6: int  1 1 0 0 0 1 1 1 1 1 ...
##  $ X7: num  0.61 0.51 -0.5 -0.14 -0.57 1.83 0.77 -0.05 -0.88 1.33 ...
```

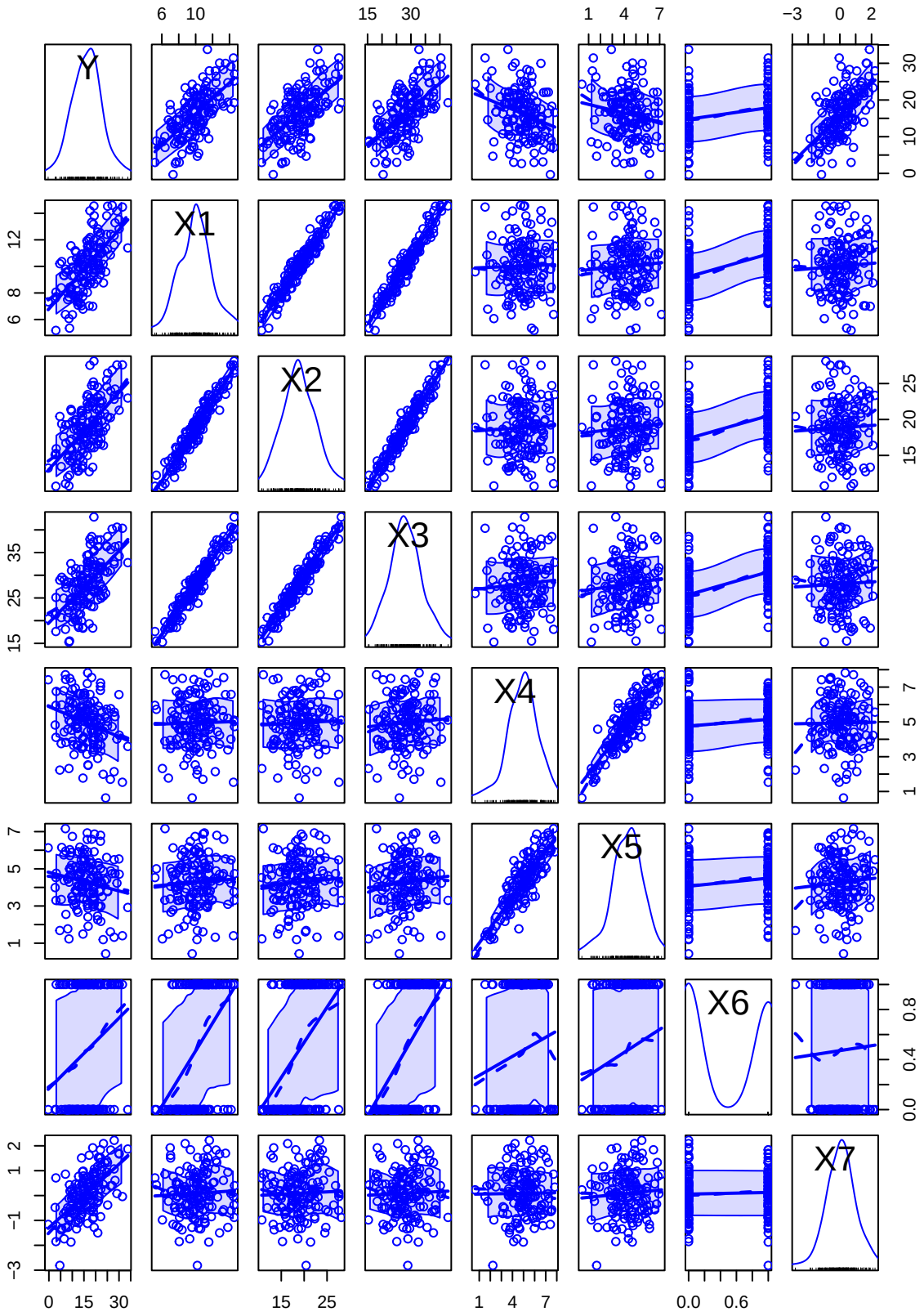
There are 150 observations of eight variables, the dependent variable `Y` and seven independent variables.

(a) Do a exploratory analysis of this data, including a scatterplot matrix and a graphical representation of the correlation matrix. Comment on your results.

```
library(car)
```

```
## Loading required package: carData
```

```
scatterplotMatrix(dat1)
```

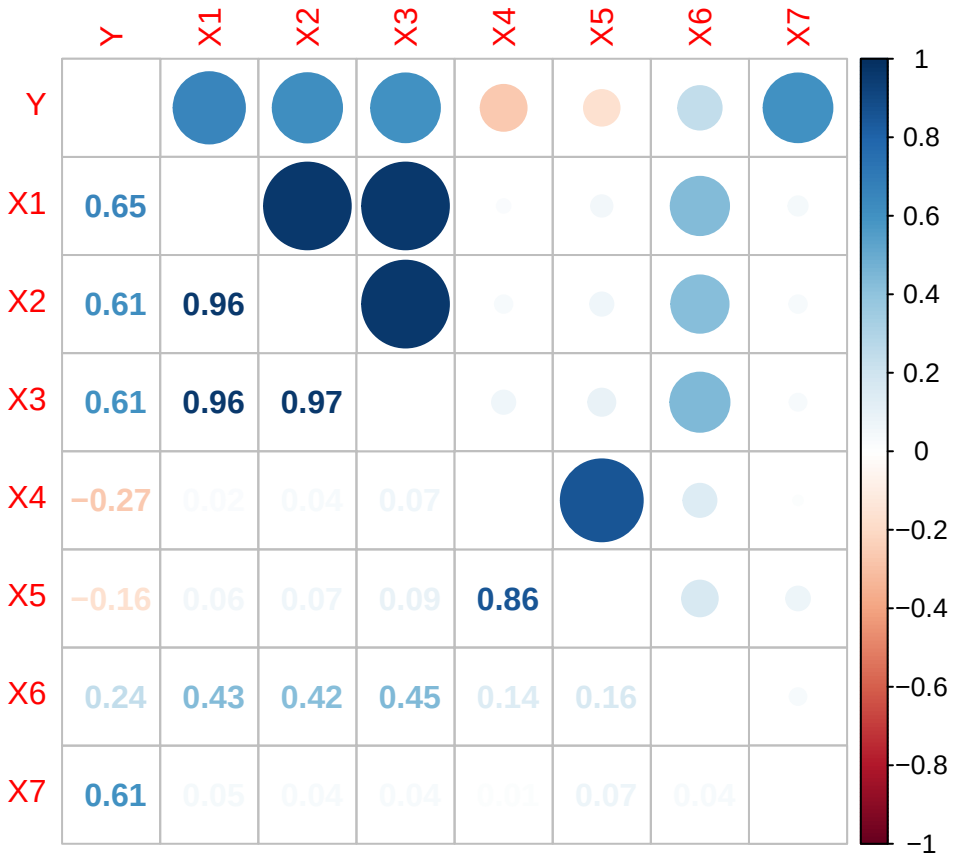


The first row of the scatterplot matrix has the dependent variable `Y` in the y-axis, and shows that it has a positive llinear relation with `X1, X2, X3`, and `X7`, and negative relation with `X4` and `X5`. `X6` is a binary variable and the average value of `Y` is higher for `X6` =1. We also see that there is a strong linear relation

between `X1, X2`, and `X3`, and also between `X4` and `X5`.

The correlation matrix is

```
cor_mat <- cor(dat1)
corrplot::corrplot.mixed(cor_mat, tl.pos = 'lt')
```



It confirms the observations we made above: `X1, X2`, and `X3` have all correlations above 0.95, while `X4` and `X5` have correlation 0.86. Also `Y` has positive correlations with `X1, X2, X3, X6`, and `X7` and negative with `X4` and `X5`.

(b) Fit a complete model for `Y` including all the other variables. Produce a summary table and interpret the t tests in the table. What is the p-value for the overall significance test for the regression?

```
full_model <- lm(Y ~ ., data = dat1)
summary(full_model)
```

```
##
## Call:
## lm(formula = Y ~ ., data = dat1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3504 -1.7645 -0.0845  1.7590  6.3147
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.71368    1.41503   1.211    0.228
## X1           2.26138    0.46965   4.815 3.73e-06 ***
```

```
## X2          -0.08099    0.25043  -0.323    0.747
## X3          -0.01075    0.17219  -0.062    0.950
## X4          -1.49758    0.30950  -4.839 3.37e-06 ***
## X5           0.18949    0.32162   0.589    0.557
## X6          -0.21165    0.48852  -0.433    0.665
## X7           4.09896    0.24099  17.009  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.64 on 142 degrees of freedom
## Multiple R-squared:  0.8399, Adjusted R-squared:  0.832
## F-statistic: 106.4 on 7 and 142 DF,  p-value: < 2.2e-16
```

The summary for the residuals at the top pf the table shows that the values a reasonably symmetric. The t-tests indicate that only three covariates are significant: `X1`, `X4`, and `X7`. The p-value for the overall significance test for the regression is at the bottom of the summary table, and is 0 ($<$ 2e-16). The adjusted R^2 is 0.832.

(c) Check for multicollinearity and drop variables as needed until this problem is resolved. Use a threshold value of 2.

Print the VIF values:

```
vif(full_model)
```

```
##        X1        X2        X3        X4        X5        X6        X7
## 18.244189 18.874281 19.975250  3.908086  3.911549  1.280474  1.018029
```

Several variables have large values. We eliminate `X3` which has the highest value.

```
model2 <- update(full_model, . ~ . - X3)
vif(model2)
```

```
##        X1        X2        X4        X5        X6        X7
## 13.974219 13.737647  3.876618  3.910763  1.263552  1.017298
```

The second model also has some big VIF values. The highest corresponds to `X1`, so we drop this variable.

```
model3 <- update(model2, . ~ . - X1)
vif(model3)
```

```
##       X2       X4       X5       X6       X7
## 1.217712 3.866653 3.909562 1.242589 1.016281
```

The third model still has large VIF values. Drop `X5`:

```
model4 <- update(model3, . ~ . - X5)
vif(model4)
```

```
##       X2       X4       X6       X7
## 1.216255 1.020561 1.238332 1.002032
```

Now all the values are below the threshold.

(d) Starting with the model obtained in section (c), get a minimal model using a backward selection procedure with a critical α of 0.1. Use the function `drop1` for this.

We use the function `drop1` to get the p-values:

```
drop1(model4, test = 'F')
```

```
## Single term deletions
```

```
##
## Model:
## Y ~ X2 + X4 + X6 + X7
##        Df Sum of Sq    RSS    AIC  F value    Pr(>F)
## <none>              1201.3 322.07
## X2      1   1820.52 3021.8 458.44 219.7494 < 2.2e-16 ***
## X4      1    548.85 1750.1 376.52  66.2499  1.65e-13 ***
## X6      1      0.64 1201.9 320.15   0.0776     0.781
## X7      1   2110.46 3311.7 472.19 254.7464 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Variable X6 has a large p-value, above the critical α. We drop this variable:

```
model5 <- update(model4, . ~ . - X6)
summary(model5)
```

```
##
## Call:
## lm(formula = Y ~ X2 + X4 + X7, data = dat1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1147 -2.0446  0.0755  1.9858  6.7435
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.24369    1.44380   2.247   0.0262 *
## X2           1.03556    0.06274  16.505  < 2e-16 ***
## X4          -1.39703    0.17029  -8.204 1.11e-13 ***
## X7           4.16202    0.25980  16.020  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.869 on 146 degrees of freedom
## Multiple R-squared:  0.8055, Adjusted R-squared:  0.8015
## F-statistic: 201.6 on 3 and 146 DF,  p-value: < 2.2e-16
```

Now all terms in the model are significant. This is the minimal adequate model. We print the summary table using the function S in the car library:

```
S(model5)
```

```
## Call: lm(formula = Y ~ X2 + X4 + X7, data = dat1)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.24369    1.44380   2.247   0.0262 *
## X2           1.03556    0.06274  16.505  < 2e-16 ***
## X4          -1.39703    0.17029  -8.204 1.11e-13 ***
## X7           4.16202    0.25980  16.020  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.869 on 146 degrees of freedom
## Multiple R-squared: 0.8055
```

```
## F-statistic: 201.6 on 3 and 146 DF,  p-value: < 2.2e-16
##    AIC    BIC
## 747.84 762.89
```

(e) Starting with the full model obtained in section (b), fit a model using Akaike's Information Criterion (AIC). You can use the function `stepAIC` in the `MASS` library or the function `step` in the base package. Check the resulting model for multicollinearity. Compare with the model obtained in (d).

We use the function `step`:

```
step_model <- step(full_model, direction = "both")
```

```
## Start:  AIC=299.01
## Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7
##
##        Df Sum of Sq     RSS    AIC
## - X3    1      0.03  989.67 297.01
## - X2    1      0.73  990.37 297.12
## - X6    1      1.31  990.95 297.20
## - X5    1      2.42  992.06 297.37
## <none>               989.64 299.01
## - X1    1    161.58 1151.22 319.69
## - X4    1    163.17 1152.81 319.90
## - X7    1   2016.23 3005.88 463.65
##
## Step:  AIC=297.01
## Y ~ X1 + X2 + X4 + X5 + X6 + X7
##
##        Df Sum of Sq     RSS    AIC
## - X2    1      1.21  990.88 295.19
## - X6    1      1.37  991.04 295.22
## - X5    1      2.43  992.09 295.38
## <none>               989.67 297.01
## + X3    1      0.03  989.64 299.01
## - X4    1    164.87 1154.54 318.12
## - X1    1    208.31 1197.98 323.66
## - X7    1   2018.08 3007.75 461.75
##
## Step:  AIC=295.19
## Y ~ X1 + X4 + X5 + X6 + X7
##
##        Df Sum of Sq     RSS    AIC
## - X6    1      1.37  992.25 293.40
## - X5    1      2.45  993.33 293.56
## <none>               990.88 295.19
## + X2    1      1.21  989.67 297.01
## + X3    1      0.51  990.37 297.12
## - X4    1    166.11 1157.00 316.44
## - X1    1   2020.10 3010.98 459.91
## - X7    1   2021.53 3012.41 459.98
##
## Step:  AIC=293.4
## Y ~ X1 + X4 + X5 + X7
##
##        Df Sum of Sq    RSS    AIC
## - X5    1      2.26  994.5 291.74
```

```
## <none>               992.3 293.40
## + X6    1      1.37  990.9 295.19
## + X2    1      1.21  991.0 295.22
## + X3    1      0.68  991.6 295.30
## - X4    1    167.01 1159.3 314.74
## - X7    1   2020.76 3013.0 458.01
## - X1    1   2427.88 3420.1 477.02
##
## Step:  AIC=291.74
## Y ~ X1 + X4 + X7
##
##        Df Sum of Sq    RSS    AIC
## <none>               994.5 291.74
## + X5    1      2.26  992.3 293.40
## + X2    1      1.23  993.3 293.56
## + X6    1      1.18  993.3 293.56
## + X3    1      0.71  993.8 293.63
## - X4    1    522.97 1517.5 353.12
## - X7    1   2065.90 3060.4 458.35
## - X1    1   2449.87 3444.4 476.08
```

The final model has also three variables: `X1, X4`, and `X7`. We print the summary:

```
S(step_model)
```

```
## Call: lm(formula = Y ~ X1 + X4 + X7, data = dat1)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8711     1.3350   1.401    0.163
## X1            2.0644     0.1089  18.965  < 2e-16 ***
## X4           -1.3567     0.1548  -8.762 4.45e-15 ***
## X7            4.1170     0.2364  17.415  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard deviation: 2.61 on 146 degrees of freedom
## Multiple R-squared: 0.8391
## F-statistic: 253.8 on 3 and 146 DF,  p-value: < 2.2e-16
##    AIC    BIC
## 719.42 734.48
```

```
vif(step_model)
```

```
##       X1       X4       X7
## 1.002760 1.000662 1.002344
```

(f) Starting with the full model obtained in section (b), fit a model by maximizing the adjusted R^2. Check the resulting model for multicollinearity. Compare with the models obtained in (d) and (e). Are all the models the same? If not, which one would you choose and why?

We use the function `regsubsets` in the `leaps` library:

```
library(leaps)
a <- regsubsets(Y ~ ., data = dat1)
summary(a)
```

```
## Subset selection object
```

```
## Call: regsubsets.formula(Y ~ ., data = dat1)
## 7 Variables  (and intercept)
##    Forced in Forced out
## X1     FALSE      FALSE
## X2     FALSE      FALSE
## X3     FALSE      FALSE
## X4     FALSE      FALSE
## X5     FALSE      FALSE
## X6     FALSE      FALSE
## X7     FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: exhaustive
##          X1  X2  X3  X4  X5  X6  X7
## 1  ( 1 ) "*" " " " " " " " " " " " "
## 2  ( 1 ) "*" " " " " " " " " " " "*"
## 3  ( 1 ) "*" " " " " "*" " " " " "*"
## 4  ( 1 ) "*" " " " " "*" "*" " " "*"
## 5  ( 1 ) "*" " " " " "*" "*" "*" "*"
## 6  ( 1 ) "*" "*" " " "*" "*" "*" "*"
## 7  ( 1 ) "*" "*" "*" "*" "*" "*" "*"
```

```
which.max(summary(a)$adjr2)
```

```
## [1] 3
```

We get the same model as in section (e): the variables are `X1, X4`, and `X7`. We have already checked for multicollinearity above. We have two different models, both include `X4` and `X7`, but `model5` uses `X1` while `step_model` has `X2`. Since `X1` and `X2` are highly correlated, they carry essentially the same information, so the models are similar. However, looking at the summary tables, we have the following values:

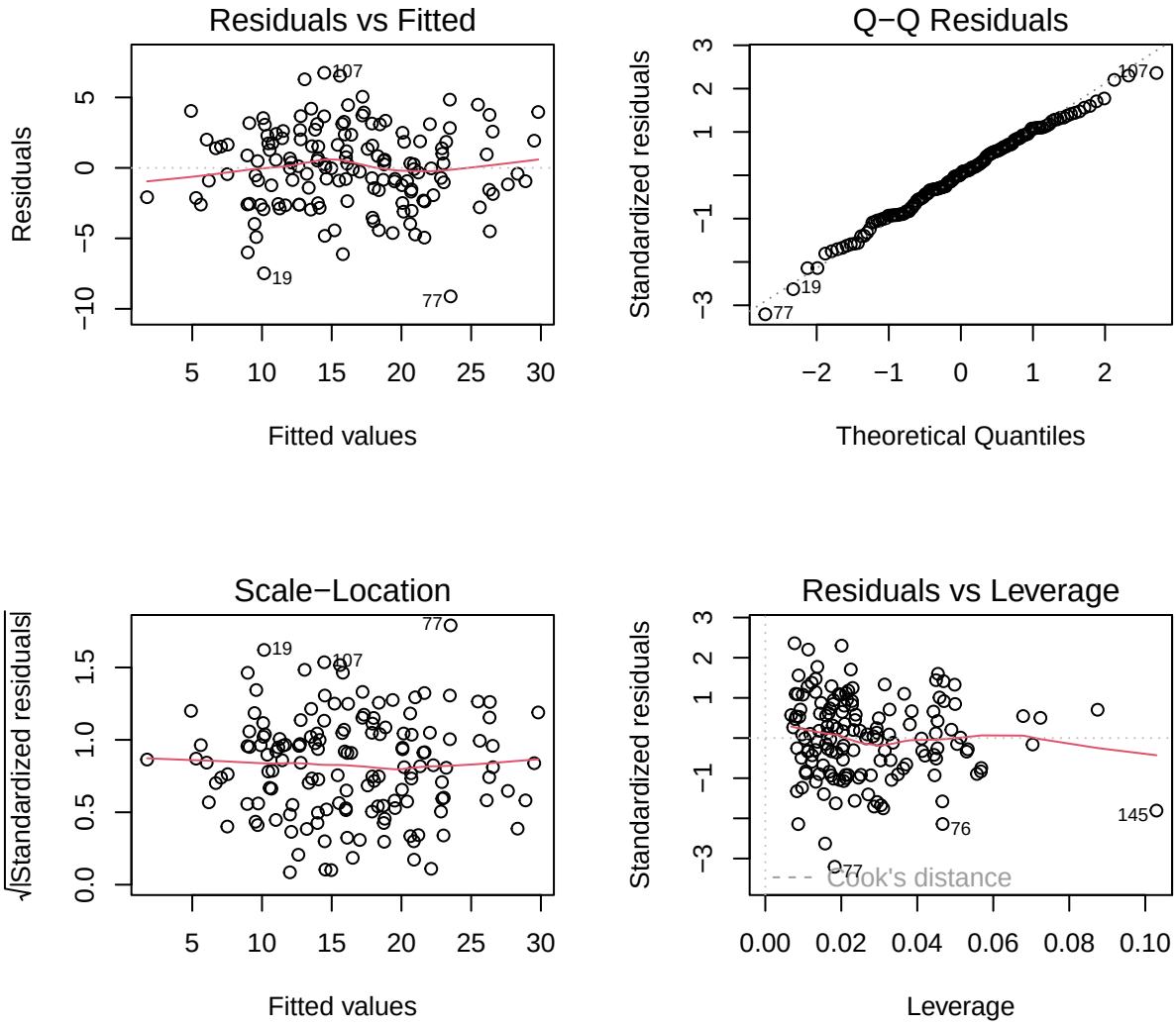
Model	R_a^2	AIC	BIC
`model5`	0.8055	747.84	762.89
`step_model`	0.8391	719.42	734.48

Looking at the table, we would prefer `step_model` since it has better values for all the criteria.

(g) Plot the standard diagnostic graphs for the model that you fitted in (d) and comment on what you observe. Use also the Shapiro-Wilk and ncv tests and comment on the results.

Diagnostic plots:

```
par(mfrow = c(2,2))
plot(model5)
```




```
par(mfrow = c(1,1))
```

All the plots look good. We check with the tests:

```
shapiro.test(rstandard(model5))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(model5)
## W = 0.99279, p-value = 0.6552
```

```
ncvTest(model5)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.04541626, Df = 1, p = 0.83124
```

Both p-values are large, and we do not reject the null hypotheses of normality and homoscedasticity.

(h) Predict the `Y` value for a subject with covariates

(`X1, X2, X3, X4, X5, X6, X7`) = (15.2, 9.6, 10.7, 9.2, 5.33, 1, -0.8)

using the model you fitted in (d). Add a confidence interval at level 98%.

We start by creating a data frame with the values for the covariates. Since the model only depends on `X2`, `X4`, and `X7`, we only include these.

```
newdata <- data.frame(X2 = 9.6, X4 = 9.2, X7 = -0.8)
```

And now we use the `predict` function:

```
predict(model5, newdata, interval = 'c', level = 0.98)
```

```
##         fit      lwr        upr
## 1 -2.997233 -5.34565 -0.6488172
```

```
newdata2 <- data.frame(X1 = 15.2, X4 = 9.2, X7 = -0.8)
```

```
predict(step_model, newdata2, interval = 'c', level = 0.98)
```

```
##        fit      lwr      upr
## 1 17.47491 15.32146 19.62836
```

Question 2

A city transportation department is studying bike rental duration. They believe that the effect of age on rental duration differs between:

- Casual riders (`member` = 0)
- Registered members (`member` = 1)

To study this, they collect data on 30 riders. The data is in the file `HW825FQ2.csv`. Remember to transform `member` into a factor.

Read the data:

```
dat2 <- read.csv('HW825FQ2.csv')
str(dat2)
```

```
## 'data.frame':    30 obs. of  3 variables:
##  $ age     : int  51 24 41 24 39 37 29 31 34 41 ...
##  $ duration: num  101.1 63.8 87.9 61.8 83.1 ...
##  $ member  : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
dat2$member <- factor(dat2$member)
str(dat2)
```

```
## 'data.frame':    30 obs. of  3 variables:
##  $ age     : int  51 24 41 24 39 37 29 31 34 41 ...
##  $ duration: num  101.1 63.8 87.9 61.8 83.1 ...
##  $ member  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

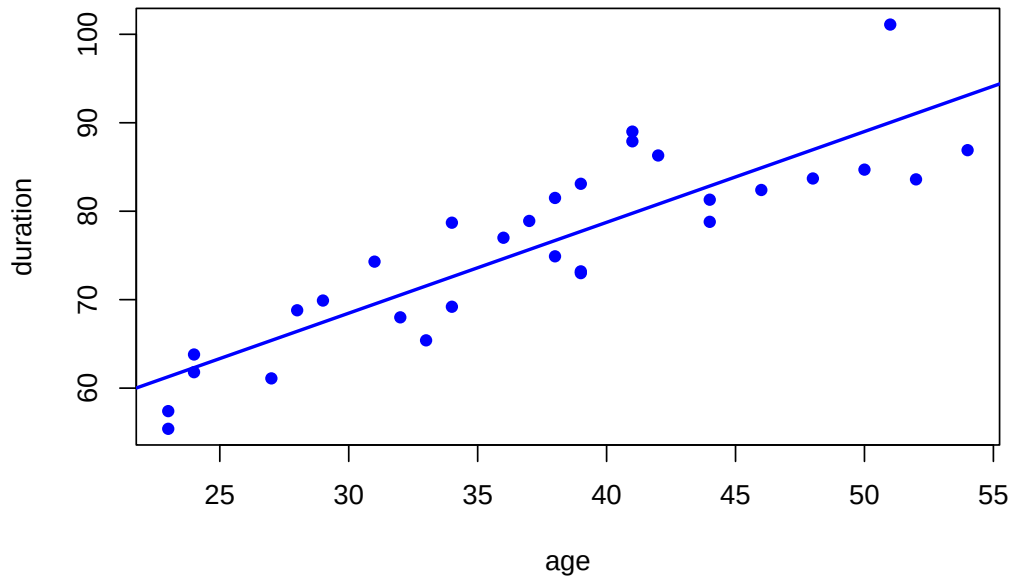
(a) Fit a simple regression model for `duration` in terms of `age`. Print the summary table and comment on the results. Draw a scatterplot and add the regression line. Comment. Using diagnostic plots and test, check whether the assumptions for the model are satisfied. Predict the value of `duration` for a value of `age = 45` with this model and include confidence intervals at the 98% level.

Solution

```
mdl1 <- lm(duration ~ age, data = dat2)
summary(mdl1)
```

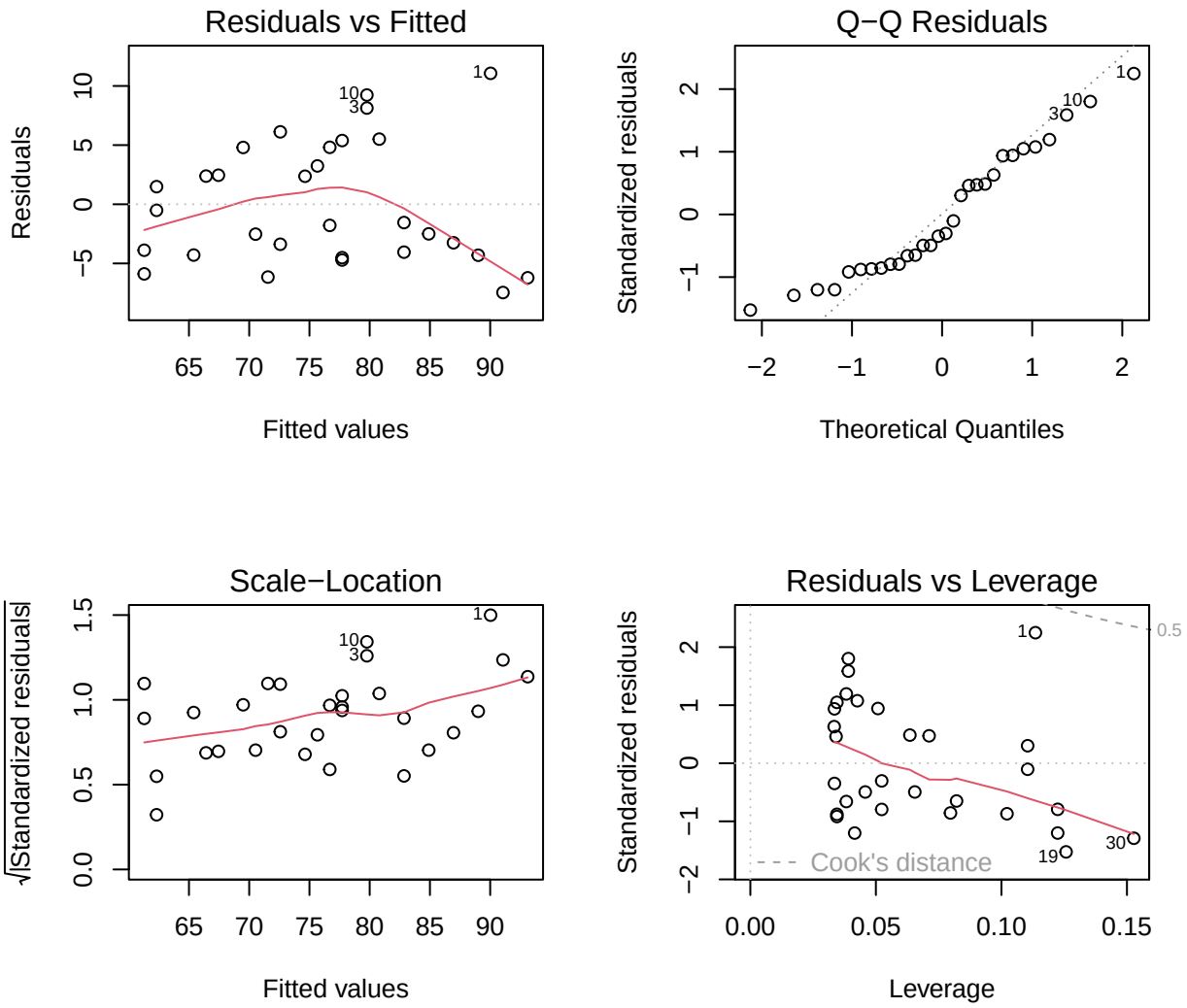
```
##
## Call:
## lm(formula = duration ~ age, data = dat2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.462 -4.231 -1.667  4.410 11.065
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  37.6696     4.1689   9.036 8.59e-10 ***
## age           1.0268     0.1086   9.454 3.27e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.228 on 28 degrees of freedom
## Multiple R-squared:  0.7615, Adjusted R-squared:  0.7529
## F-statistic: 89.38 on 1 and 28 DF,  p-value: 3.273e-10
```

```
plot(duration ~ age, data = dat2, pch = 16, col ='blue')
abline(mdl1, col = 'blue', lwd = 2)
```



Diagnostic plots:

```
par(mfrow = c(2,2))
plot(mdl1)
```




```
par(mfrow = c(1,1))
```

Tests

```
shapiro.test(rstandard(mdl1))
```

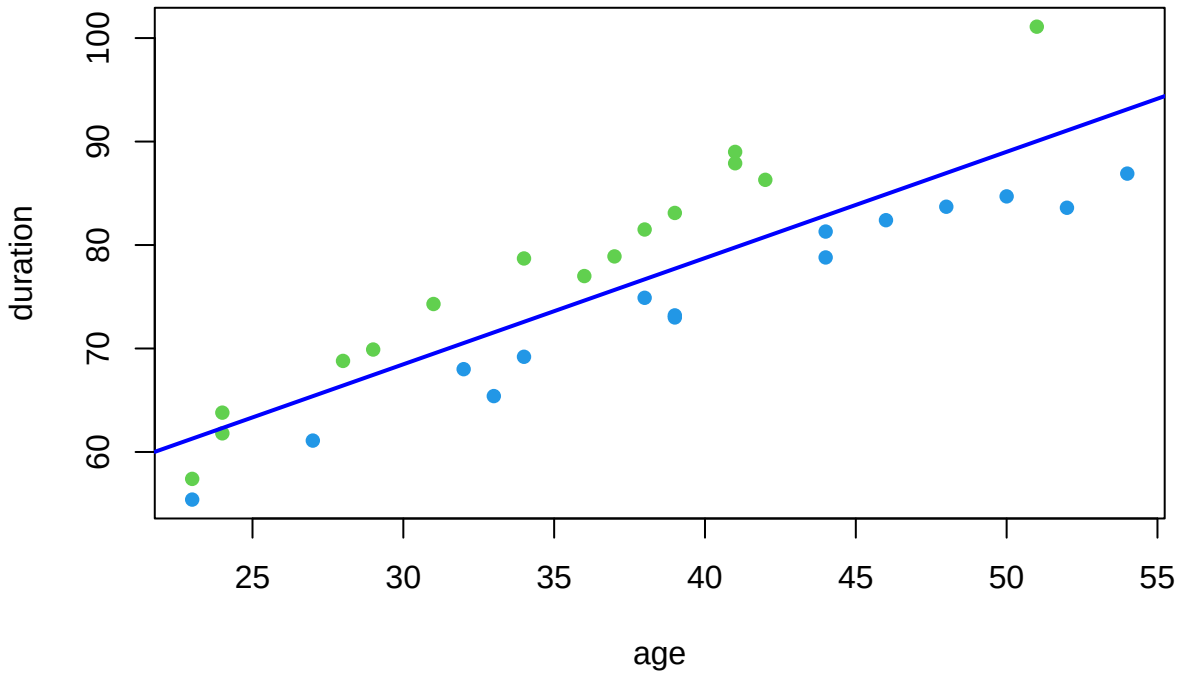
```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(mdl1)
## W = 0.93962, p-value = 0.08885
```

```
library(car)
ncvTest(mdl1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 2.812606, Df = 1, p = 0.093526
```

(b) Draw a new scatterplot and color the points according to the value of `member`. Comment on what you observe.

```
plot(duration ~ age, data = dat2, pch = 16, col = as.numeric(member) + 2)
abline(mdl1, col = 'blue', lwd = 2)
```



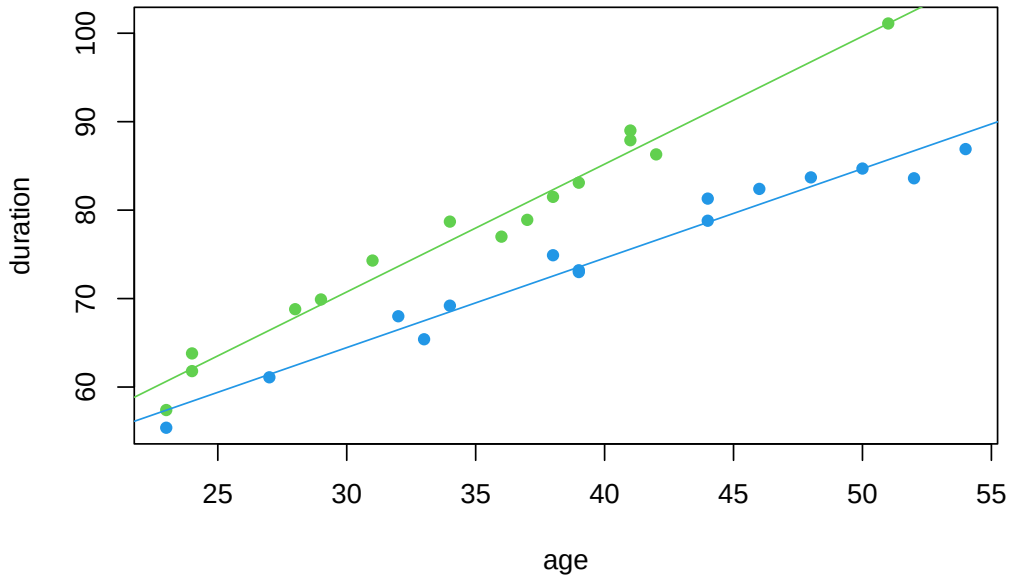
(c) Fit a new model for `duration` as a function of `age` and `member`, including an interaction term. Print an anova table for this model and interpret the p-values in the table. If necessary, fit a new model dropping the terms that have a non-significant p-value. Print a summary table for the final model and interpret the coefficients. What is the value for the estimated variance of the errors? What is the R^2, how do they compare with the previous model?

```
mdl2 <- lm(duration ~ age*member, data = dat2)
summary(mdl2)
```

```
##
## Call:
## lm(formula = duration ~ age * member, data = dat2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2337 -1.5447  0.0104  1.4533  2.6837
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.39758    2.16233  12.670 1.24e-12 ***
## age          1.44505    0.06111  23.646  < 2e-16 ***
## member1      6.72162    3.07271   2.188   0.0379 *
## age:member1 -0.43375    0.08091  -5.361 1.30e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.825 on 26 degrees of freedom
## Multiple R-squared:  0.973,  Adjusted R-squared:  0.9699
## F-statistic: 312.4 on 3 and 26 DF,  p-value: < 2.2e-16
```

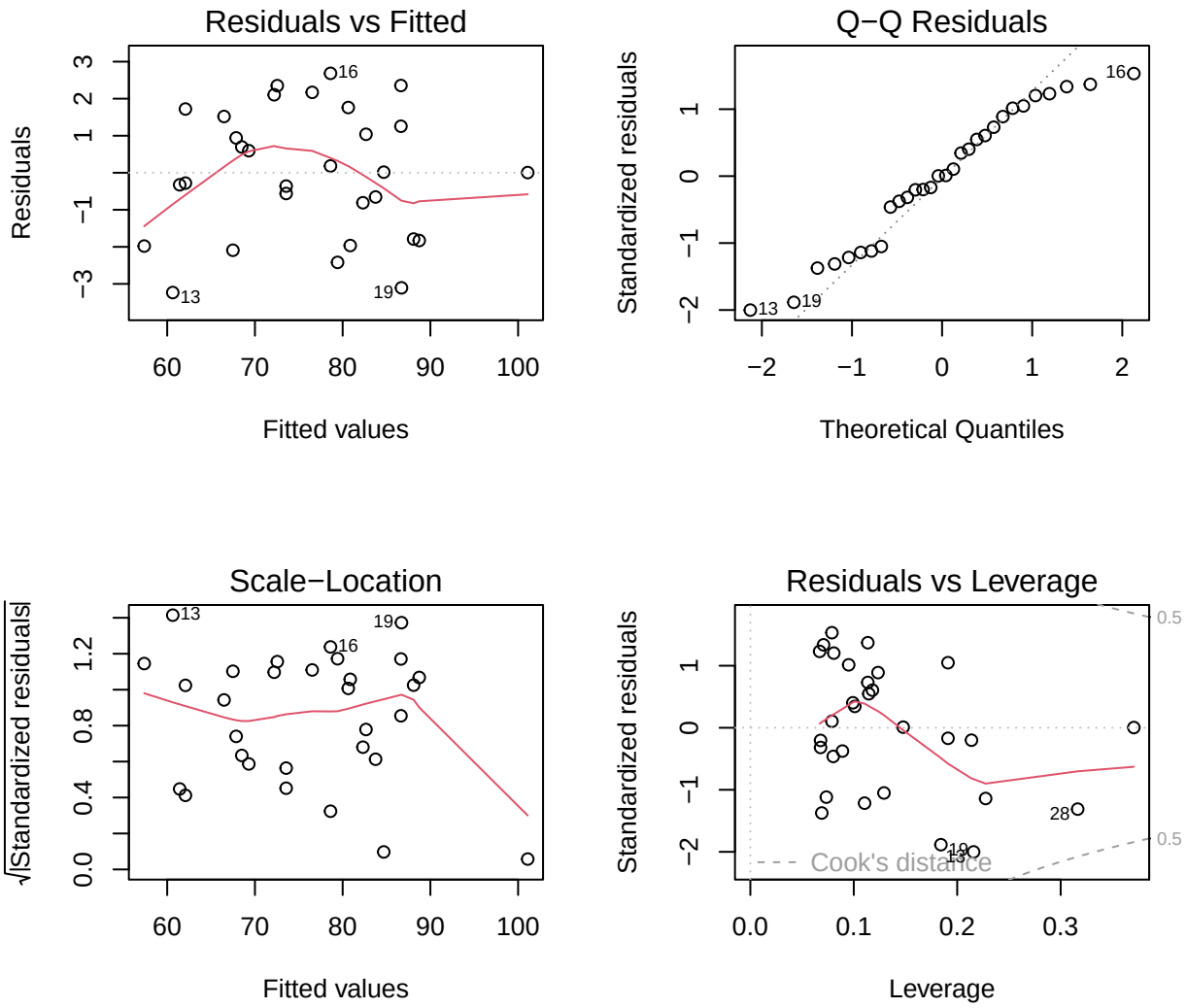
(d) Draw a scatterplot and color the points according to the value of type. Add the regression lines corresponding to the model you fitted in (c). Write down an equation for this model.

```
cf2 <- coef(md12)
plot(duration ~ age, data = dat2, pch = 16, col = as.numeric(member) + 2)
abline(a = cf2[1], b =cf2[2], col = 3 )
abline(a = cf2[1]+cf2[3], b =cf2[2]+cf2[4], col = 4 )
```



(e) Plot the standard diagnostic graphs for the model that you fitted in (c) and comment on what you observe. Use also the Shapiro-Wilk and ncv tests and comment on the results.

```
par(mfrow = c(2,2))
plot(md12)
```




```
par(mfrow = c(1,1))
shapiro.test(rstandard(mdl2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(mdl2)
## W = 0.95335, p-value = 0.2076
```

```
ncvTest(mdl2)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.03139018, Df = 1, p = 0.85937
```

(f) Predict the value of `duration` for a value of `age = 45` and for the two levels of `member`. Include confidence intervals at the 98% level. Compare with the prediction in (a).

```
predict(mdl2, data.frame(age = 45, member = '0'), interval = 'c', level = 0.98)
```

```
##        fit      lwr      upr
## 1 92.42487 90.45574 94.39399
```

```r
predict(mdl2, data.frame(age = 45, member = '1'), interval = 'c', level = 0.98)
```

```
##        fit      lwr      upr
## 1 79.62756 78.30017 80.95495
```

Compare with

```r
predict(mdl1, data.frame(age = 45), interval = 'c', level = 0.98)
```

```
##        fit      lwr     upr
## 1 83.87436 80.75542 86.9933
```