Regression Modelling

Ilke Kas

Due: April 12, 2025

Load the `Carseats` data from the `ISLR2` package. Use R to answer the following:

(i) (1 pt)

Set a seed with the numeric part of your CaseID and partition the data into 50-50 training and test sets.

```
set.seed(238)
data(Carseats)

# Example: create partition
 train_id <- createDataPartition(Carseats$Sales, p = 0.5, list = FALSE)
 train_data <- Carseats[train_id, ]
 test_data <- Carseats[-train_id, ]
 # Train verisinin başı
head(train_data)
```

```
##    Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
## 2  11.22       111     48          16        260    83      Good  65        10
## 6  10.81       124    113          13        501    72       Bad  78        16
## 7   6.63       115    105           0         45   108    Medium  71        15
## 8  11.85       136     81          15        425   120      Good  67        10
## 9   6.54       132    110           0        108   124    Medium  76        10
## 14 10.96       115     28          11         29    86      Good  53        18
##    Urban  US
## 2    Yes Yes
## 6     No Yes
## 7    Yes  No
## 8    Yes Yes
## 9     No  No
## 14   Yes Yes
```

```
# Test verisinin başı
head(test_data)
```

```
##    Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
## 1   9.50       138     73          11        276   120       Bad  42        17
## 3  10.06       113     35          10        269    80    Medium  59        12
## 4   7.40       117    100           4        466    97    Medium  55        14
## 5   4.15       141     64           3        340   128       Bad  38        13
## 10  4.69       132    113           0        131   124    Medium  76        17
## 11  9.01       121     78           9        150   100       Bad  26        10
```

```
##    Urban  US
## 1    Yes Yes
## 3    Yes Yes
## 4    Yes Yes
## 5    Yes  No
## 10    No Yes
## 11    No Yes
```

Explanation: In this code segment, ı am splitting the dataset into training and testing sets. Firstly, since my case id is "ixk238", I am setting the seed for reproducibility using set.seed(238).Then, I am using the createDataPartition() function from the caret package to randomly select 50% of the rows as training data. The remaining rows form the test set, and I used the head() function to display the first few rows of each subset.

(ii) (1 pt)

Fit an appropriate linear model to the training data with sales as the response and the remaining variables as predictors.

```r
# Example:
lm_model <- lm(Sales ~ ., data = train_data)
summary(lm_model)
```

```
##
## Call:
## lm(formula = Sales ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7566 -0.7018  0.0009  0.6471  3.3715
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.1456891  0.9256581   5.559 9.15e-08 ***
## CompPrice        0.0943164  0.0060982  15.466  < 2e-16 ***
## Income           0.0150813  0.0027356   5.513 1.15e-07 ***
## Advertising      0.1042784  0.0165621   6.296 2.08e-09 ***
## Population       0.0008845  0.0005354   1.652    0.100
## Price           -0.0938957  0.0038488 -24.396  < 2e-16 ***
## ShelveLocGood    5.1855213  0.2194651  23.628  < 2e-16 ***
## ShelveLocMedium  2.1109116  0.1769269  11.931  < 2e-16 ***
## Age             -0.0440275  0.0047531  -9.263  < 2e-16 ***
## Education       -0.0300864  0.0284645  -1.057    0.292
## UrbanYes         0.0736750  0.1620566   0.455    0.650
## USYes           -0.0055685  0.2149732  -0.026    0.979
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.023 on 189 degrees of freedom
## Multiple R-squared:  0.8711, Adjusted R-squared:  0.8636
## F-statistic: 116.1 on 11 and 189 DF,  p-value: < 2.2e-16
```

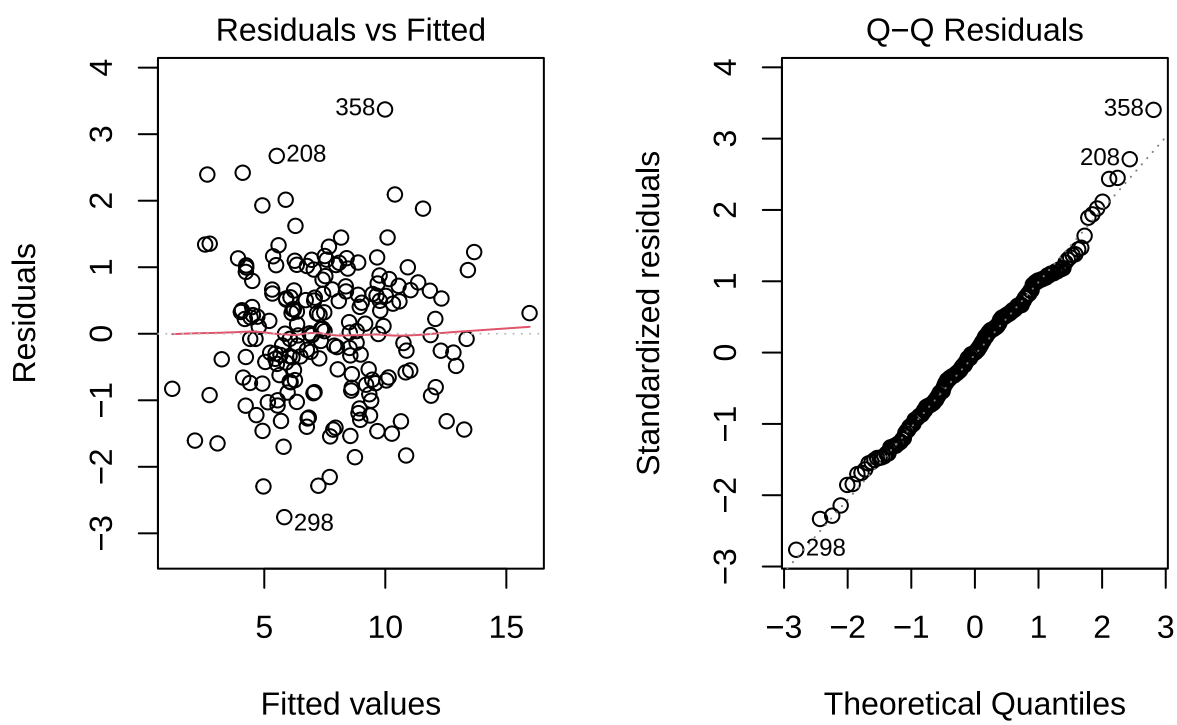
Explanation:

In this code segment, I applied lm() function builds the model with Sales ~ ., where . means "use all other columns as predictors.". The summary(lm_model) function then displays detailed information about the model, including coefficients, statistical significance (p-values), R-squared value, and residual errors. The F-statistic has a very small p-value (< 2.2e-16), which means the model is statistically significant. So, at least one of the predictors is useful for predicting Sales. The adjusted R^2 is 0.8636, which means about 86.4% of the variability in Sales can be explained by the predictors in the model. That's a strong fit. The model shows that CompPrice, Income, Advertising, Price, ShelveLoc, and Age significantly affect Sales, with better shelf location and lower price strongly increasing Sales. On the other hand, Population, Education, UrbanYes, and USYes are not significant predictors based on their p-values. Residuals are fairly balanced around 0, with a min of -2.76 and max of +3.37. This suggests no major outliers, but I should still check diagnostic plots to validate the assumptions.

(iii) (2 pt)

Conduct a residual diagnosis of your model in (ii) and determine whether your model is a good fit. Explain your results.

```r
# Example residual plots:
par(mfrow=c(1,2))
plot(lm_model,1:2)
```



Explanation: Is the model adequate? The diagnostic plots for the first-order linear model look pretty good overall. In the Residuals vs Fitted plot, the points are scattered fairly evenly around the zero line, which means the model captures the linear trend well and there's no clear pattern in the errors. There's a little bit more spread for higher fitted values, but nothing too concerning. The Q-Q plot also looks nice — most of the points follow the straight line, which suggests that the residuals are roughly normally distributed. There are a couple of outliers, like points 358 and 298, but that's pretty normal in real-world data. So, overall, the model seems to be doing a solid job. Let's look at the predictions:

```r
library(pracma)
```

```
##
## Attaching package: 'pracma'

## The following objects are masked from 'package:Matrix':
##
##     expm, lu, tril, triu
```

```r
# Predict Sales using the trained first-order model
sales_pred <- predict(lm_model, newdata = test_data)

# Assess model accuracy (RMSE)
do.call(cbind, pracma::rmserr(test_data$Sales, sales_pred))
```

```
##            mae      mse     rmse mape      nmse      rstd
## [1,] 0.8379974 1.101222 1.049391  Inf 0.1329734 0.1409111
```

The first-order linear model performs well overall, with an RMSE of about 1.05, indicating that predictions deviate from actual Sales values by roughly one unit on average. The low NMSE (0.133) and rSTD (0.141) suggest that the model explains a large portion of the variability in the data and generalizes well. Although the MAPE is infinite due to zero or near-zero values in the test set, the other error metrics confirm that this model is a strong fit.

What if I try second order, will it be better or not:

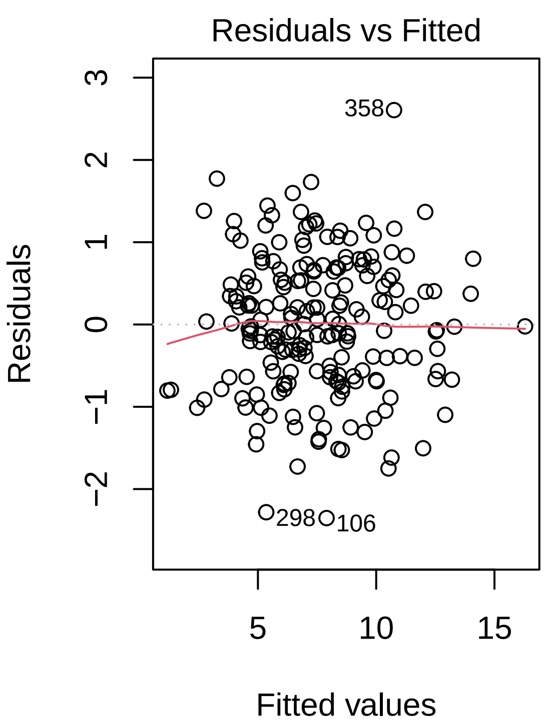
```r
# Second-order linear model with interactions and squared terms
lm_model2 <- lm(Sales ~ (.)^2 + I(CompPrice^2) + I(Income^2) + I(Advertising^2) +
                          I(Population^2) + I(Price^2) + I(Age^2) + I(Education^2),
                data = train_data)

summary(lm_model2)
```

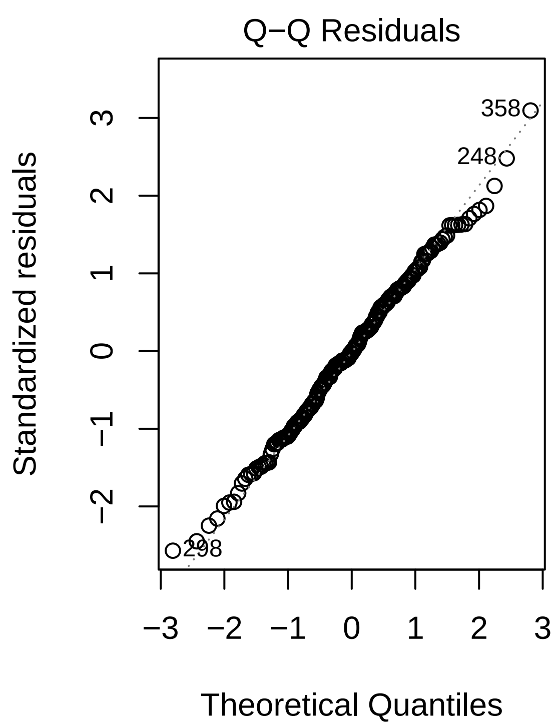
```
##
## Call:
## lm(formula = Sales ~ (.)^2 + I(CompPrice^2) + I(Income^2) + I(Advertising^2) +
##     I(Population^2) + I(Price^2) + I(Age^2) + I(Education^2),
##     data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.35240 -0.62924 -0.02272  0.59581  2.60558
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -7.775e-01  1.128e+01  -0.069   0.9452
## CompPrice                    1.776e-01  1.329e-01   1.337   0.1837
## Income                       2.305e-02  5.111e-02   0.451   0.6527
## Advertising                 -4.630e-01  3.341e-01  -1.386   0.1683
## Population                   2.072e-02  9.018e-03   2.297   0.0232 *
## Price                       -1.088e-01  6.977e-02  -1.559   0.1214
## ShelveLocGood                3.585e+00  3.642e+00   0.984   0.3269
## ShelveLocMedium              6.461e-01  2.978e+00   0.217   0.8286
## Age                         -7.720e-02  9.248e-02  -0.835   0.4054
## Education                    2.076e-01  6.744e-01   0.308   0.7587
## UrbanYes                    -3.285e+00  2.638e+00  -1.245   0.2153
```

```
## USYes                          3.158e+00  3.699e+00   0.854   0.3948
## I(CompPrice^2)                 2.023e-04  5.191e-04   0.390   0.6974
## I(Income^2)                    2.312e-04  1.310e-04   1.765   0.0800 .
## I(Advertising^2)               6.732e-04  3.494e-03   0.193   0.8475
## I(Population^2)               -8.349e-06  5.269e-06  -1.585   0.1155
## I(Price^2)                     7.603e-05  2.090e-04   0.364   0.7167
## I(Age^2)                       1.224e-04  4.255e-04   0.288   0.7740
## I(Education^2)                 8.198e-04  1.755e-02   0.047   0.9628
## CompPrice:Income              -6.642e-04  3.004e-04  -2.211   0.0288 *
## CompPrice:Advertising          1.501e-03  1.909e-03   0.786   0.4332
## CompPrice:Population          -4.672e-05  5.670e-05  -0.824   0.4115
## CompPrice:Price               -6.409e-05  5.125e-04  -0.125   0.9007
## CompPrice:ShelveLocGood        7.322e-04  2.418e-02   0.030   0.9759
## CompPrice:ShelveLocMedium      4.935e-03  1.995e-02   0.247   0.8050
## CompPrice:Age                 -5.138e-04  5.284e-04  -0.972   0.3326
## CompPrice:Education           -3.810e-03  3.010e-03  -1.266   0.2079
## CompPrice:UrbanYes             5.882e-03  1.821e-02   0.323   0.7472
## CompPrice:USYes               -1.191e-03  2.722e-02  -0.044   0.9652
## Income:Advertising             2.265e-04  9.084e-04   0.249   0.8035
## Income:Population              3.005e-06  2.671e-05   0.113   0.9106
## Income:Price                  -1.434e-05  1.897e-04  -0.076   0.9399
## Income:ShelveLocGood           7.150e-03  1.101e-02   0.649   0.5172
## Income:ShelveLocMedium         7.959e-03  8.581e-03   0.928   0.3554
## Income:Age                     1.854e-04  2.458e-04   0.754   0.4521
## Income:Education               1.604e-03  1.312e-03   1.223   0.2237
## Income:UrbanYes                3.327e-03  7.359e-03   0.452   0.6520
## Income:USYes                   1.399e-03  1.043e-02   0.134   0.8935
## Advertising:Population         2.433e-04  1.962e-04   1.240   0.2172
## Advertising:Price              2.137e-03  1.261e-03   1.696   0.0924 .
## Advertising:ShelveLocGood     -2.500e-02  6.623e-02  -0.377   0.7064
## Advertising:ShelveLocMedium    5.377e-02  4.621e-02   1.164   0.2468
## Advertising:Age                6.003e-04  1.398e-03   0.429   0.6685
## Advertising:Education         -9.448e-04  8.346e-03  -0.113   0.9100
## Advertising:UrbanYes          -4.651e-02  5.234e-02  -0.889   0.3758
## Advertising:USYes              3.304e-02  1.360e-01   0.243   0.8084
## Population:Price              -3.329e-05  3.683e-05  -0.904   0.3678
## Population:ShelveLocGood       2.855e-04  2.423e-03   0.118   0.9064
## Population:ShelveLocMedium    -3.598e-03  1.570e-03  -2.291   0.0236 *
## Population:Age                 7.387e-06  4.764e-05   0.155   0.8770
## Population:Education          -2.901e-04  2.558e-04  -1.134   0.2588
## Population:UrbanYes           -8.349e-04  1.450e-03  -0.576   0.5657
## Population:USYes              -2.237e-03  2.344e-03  -0.955   0.3415
## Price:ShelveLocGood           -5.239e-03  1.450e-02  -0.361   0.7184
## Price:ShelveLocMedium         -5.715e-03  1.195e-02  -0.478   0.6334
## Price:Age                      3.914e-04  3.578e-04   1.094   0.2760
## Price:Education                3.889e-04  1.968e-03   0.198   0.8436
## Price:UrbanYes                 6.735e-03  1.112e-02   0.606   0.5458
## Price:USYes                   -3.699e-02  1.665e-02  -2.221   0.0281 *
## ShelveLocGood:Age              5.727e-03  1.712e-02   0.334   0.7386
## ShelveLocMedium:Age            1.246e-02  1.385e-02   0.900   0.3698
## ShelveLocGood:Education        2.719e-02  1.079e-01   0.252   0.8015
## ShelveLocMedium:Education      4.473e-02  9.187e-02   0.487   0.6272
## ShelveLocGood:UrbanYes         1.963e-01  6.957e-01   0.282   0.7783
## ShelveLocMedium:UrbanYes       3.232e-01  5.236e-01   0.617   0.5382
```

```
## ShelveLocGood:USYes           1.415e+00  8.582e-01   1.649   0.1017
## ShelveLocMedium:USYes         2.751e-03  5.875e-01   0.005   0.9963
## Age:Education                 2.498e-04  2.496e-03   0.100   0.9204
## Age:UrbanYes                  2.590e-02  1.467e-02   1.765   0.0800 .
## Age:USYes                    -8.125e-03  1.787e-02  -0.455   0.6501
## Education:UrbanYes            2.111e-02  7.774e-02   0.271   0.7864
## Education:USYes               9.632e-02  1.071e-01   0.899   0.3701
## UrbanYes:USYes                5.630e-01  6.184e-01   0.910   0.3644
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.037 on 128 degrees of freedom
## Multiple R-squared:  0.9102, Adjusted R-squared:  0.8597
## F-statistic: 18.02 on 72 and 128 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(1,2))
plot(lm_model2,1:2)
```





Explanation: The second-order model shows some good signs, but it doesn't clearly outperform the first-order model. While we might expect a higher adjusted R^2 from a more complex model, the second-order model actually has a slightly lower adjusted R^2 (0.8597) compared to the first-order model (0.8636), suggesting that the added complexity didn't improve generalization. Looking at the residual plots, the Residuals vs Fitted plot appears fairly random with no clear pattern, which is a good sign, although there's still some spread that could hint at mild heteroscedasticity. The Q-Q plot shows that the residuals mostly follow the diagonal line, indicating they are approximately normally distributed, with just a few outliers. The residual standard error has also slightly increased from 1.023 in the first model to 1.037 in the second. THere are many unnecessary interaction and squared terms, which likely leads to overfitting and weak generalization.

So, instead using a stepwise regression could be better since it selects a subset of predictors.

```r
library(pracma)
# Predict Sales using the trained first-order model
sales_pred <- predict(lm_model2, newdata = test_data)

# Assess model accuracy (RMSE)
do.call(cbind, pracma::rmserr(test_data$Sales, sales_pred))
```

```
##           mae      mse     rmse mape    nmse      rstd
## [1,] 1.049466 1.808188 1.344689  Inf 0.21834 0.1805634
```

The second-order model has a higher RMSE of 1.34 compared to 1.05 from the first-order model, meaning its predictions are less accurate on average. Its NMSE (0.218) and rSTD (0.181) are also worse than the first model's values, indicating a weaker ability to explain the variability in Sales. Overall, despite being more complex, the second-order model performs worse and likely overfits the data, making the first-order model the better choice.

(iv) (2 pt)

Use stepwise regression to select the best model.

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```r
print("backward")
```

```
## [1] "backward"
```

```r
full_model <- lm(Sales ~ ., data = train_data)
backward_model <- stepAIC(full_model, direction = "backward")
```

```
## Start:  AIC=20.68
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education + Urban + US
##
##               Df Sum of Sq    RSS     AIC
## - US           1      0.00 197.71  18.681
## - Urban        1      0.22 197.92  18.900
## - Education    1      1.17 198.88  19.865
## <none>                     197.71  20.680
## - Population   1      2.85 200.56  21.562
## - Income       1     31.79 229.50  48.654
## - Advertising  1     41.47 239.18  56.953
## - Age          1     89.76 287.46  93.916
## - CompPrice    1    250.23 447.94 183.071
## - ShelveLoc    2    586.90 784.61 293.737
## - Price        1    622.58 820.29 304.678
##
## Step:  AIC=18.68
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education + Urban
```

```
##
##               Df Sum of Sq    RSS     AIC
## - Urban        1      0.22 197.92  16.900
## - Education    1      1.18 198.89  17.874
## <none>                     197.71  18.681
## - Population   1      3.12 200.83  19.826
## - Income       1     31.80 229.50  46.656
## - Advertising  1     80.57 278.28  85.388
## - Age          1     89.76 287.47  91.918
## - CompPrice    1    250.24 447.95 181.077
## - ShelveLoc    2    591.23 788.94 292.844
## - Price        1    628.29 826.00 304.072
##
## Step:  AIC=16.9
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education
##
##               Df Sum of Sq    RSS     AIC
## - Education    1      1.21 199.14  16.129
## <none>                     197.92  16.900
## - Population   1      3.07 200.99  17.991
## - Income       1     31.70 229.62  44.757
## - Advertising  1     81.46 279.38  84.183
## - Age          1     90.07 288.00  90.288
## - CompPrice    1    250.12 448.04 179.117
## - ShelveLoc    2    592.75 790.67 291.285
## - Price        1    628.89 826.81 302.269
##
## Step:  AIC=16.13
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age
##
##               Df Sum of Sq    RSS     AIC
## <none>                     199.14  16.129
## - Population   1      3.74 202.88  17.870
## - Income       1     35.19 234.32  46.833
## - Advertising  1     81.24 280.38  82.901
## - Age          1     89.03 288.17  88.406
## - CompPrice    1    250.53 449.67 177.846
## - ShelveLoc    2    600.72 799.86 291.608
## - Price        1    629.40 828.54 300.688
```

```r
summary(backward_model)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income + Advertising + Population +
##     Price + ShelveLoc + Age, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6274 -0.6989 -0.0089  0.6604  3.3643
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      4.6995535  0.7589752   6.192 3.53e-09 ***
## CompPrice        0.0942562  0.0060647  15.542  < 2e-16 ***
## Income           0.0155920  0.0026769   5.825 2.38e-08 ***
## Advertising      0.1041845  0.0117716   8.850 5.73e-16 ***
## Population       0.0009621  0.0005065   1.899    0.059 .
## Price           -0.0938453  0.0038096 -24.634  < 2e-16 ***
## ShelveLocGood    5.1971084  0.2163206  24.025  < 2e-16 ***
## ShelveLocMedium  2.1145656  0.1740942  12.146  < 2e-16 ***
## Age             -0.0437138  0.0047183  -9.265  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.018 on 192 degrees of freedom
## Multiple R-squared:  0.8701, Adjusted R-squared:  0.8647
## F-statistic: 160.8 on 8 and 192 DF,  p-value: < 2.2e-16
```

```r
print("==========================================================================")
```

```
## [1] "=========================================================================="
```

```r
print("forward")
```

```
## [1] "forward"
```

```r
null_model <- lm(Sales ~ 1, data = train_data)
forward_model <- stepAIC(null_model,
                         direction = "forward",
                         scope = list(lower = null_model, upper = full_model))
```

```
## Start:  AIC=410.41
## Sales ~ 1
##
##                 Df Sum of Sq     RSS    AIC
## + ShelveLoc      2    550.61  982.68 324.98
## + Price          1    262.27 1271.02 374.70
## + Advertising    1    118.62 1414.68 396.22
## + Income         1     29.12 1504.17 408.55
## + US             1     27.19 1506.11 408.81
## + Age            1     22.37 1510.93 409.45
## <none>                       1533.30 410.41
## + Education      1     13.55 1519.74 410.62
## + CompPrice      1      3.33 1529.97 411.97
## + Urban          1      2.70 1530.59 412.05
## + Population     1      0.36 1532.94 412.36
##
## Step:  AIC=324.98
## Sales ~ ShelveLoc
##
##                 Df Sum of Sq    RSS    AIC
## + Price          1    349.14 633.55 238.75
## + Advertising    1     70.60 912.08 312.00
## + Income         1     54.01 928.68 315.62
## + Age            1     34.02 948.66 319.90
## + US             1     18.53 964.15 323.16
## <none>                       982.68 324.98
## + Education      1      5.54 977.15 325.85
```

```
## + Population   1      3.23 979.46 326.32
## + Urban        1      0.06 982.62 326.97
## + CompPrice    1      0.04 982.64 326.97
##
## Step:  AIC=238.75
## Sales ~ ShelveLoc + Price
##
##                Df Sum of Sq    RSS    AIC
## + CompPrice    1   207.936 425.61 160.79
## + Age          1    91.045 542.50 209.57
## + Advertising  1    65.069 568.48 218.97
## + US           1    33.971 599.57 229.68
## + Income       1    30.463 603.08 230.85
## <none>                     633.55 238.75
## + Education    1     2.424 631.12 239.98
## + Population   1     0.439 633.11 240.61
## + Urban        1     0.217 633.33 240.69
##
## Step:  AIC=160.79
## Sales ~ ShelveLoc + Price + CompPrice
##
##                Df Sum of Sq    RSS    AIC
## + Advertising  1   100.447 325.16 108.69
## + Age          1    85.112 340.50 117.95
## + US           1    44.269 381.34 140.72
## + Income       1    42.900 382.71 141.44
## + Population   1    14.164 411.45 155.99
## <none>                     425.61 160.79
## + Education    1     4.170 421.44 160.81
## + Urban        1     1.399 424.21 162.13
##
## Step:  AIC=108.69
## Sales ~ ShelveLoc + Price + CompPrice + Advertising
##
##               Df Sum of Sq    RSS     AIC
## + Age         1    86.474 238.69  48.543
## + Income      1    35.038 290.12  87.768
## <none>                    325.16 108.685
## + Education   1     2.895 322.27 108.888
## + Population  1     2.419 322.74 109.184
## + Urban       1     0.356 324.81 110.465
## + US          1     0.010 325.15 110.679
##
## Step:  AIC=48.54
## Sales ~ ShelveLoc + Price + CompPrice + Advertising + Age
##
##               Df Sum of Sq    RSS    AIC
## + Income      1    35.810 202.88 17.870
## + Education   1     6.035 232.65 45.396
## + Population  1     4.365 234.32 46.833
## <none>                    238.69 48.543
## + Urban       1     0.119 238.57 50.443
## + US          1     0.058 238.63 50.494
##
```

```
## Step:  AIC=17.87
## Sales ~ ShelveLoc + Price + CompPrice + Advertising + Age + Income
##
##               Df Sum of Sq    RSS    AIC
## + Population  1    3.7416 199.14 16.129
## <none>                    202.88 17.870
## + Education   1    1.8886 200.99 17.991
## + Urban       1    0.1994 202.68 19.673
## + US          1    0.1620 202.72 19.710
##
## Step:  AIC=16.13
## Sales ~ ShelveLoc + Price + CompPrice + Advertising + Age + Income +
##     Population
##
##              Df Sum of Sq    RSS    AIC
## <none>                    199.14 16.129
## + Education  1   1.21362 197.92 16.900
## + Urban      1   0.25205 198.89 17.874
## + US         1   0.01208 199.13 18.117
```

```r
summary(forward_model)
```

```
##
## Call:
## lm(formula = Sales ~ ShelveLoc + Price + CompPrice + Advertising +
##     Age + Income + Population, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6274 -0.6989 -0.0089  0.6604  3.3643
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.6995535  0.7589752   6.192 3.53e-09 ***
## ShelveLocGood    5.1971084  0.2163206  24.025  < 2e-16 ***
## ShelveLocMedium  2.1145656  0.1740942  12.146  < 2e-16 ***
## Price           -0.0938453  0.0038096 -24.634  < 2e-16 ***
## CompPrice        0.0942562  0.0060647  15.542  < 2e-16 ***
## Advertising      0.1041845  0.0117716   8.850 5.73e-16 ***
## Age             -0.0437138  0.0047183  -9.265  < 2e-16 ***
## Income           0.0155920  0.0026769   5.825 2.38e-08 ***
## Population       0.0009621  0.0005065   1.899    0.059 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.018 on 192 degrees of freedom
## Multiple R-squared:  0.8701, Adjusted R-squared:  0.8647
## F-statistic: 160.8 on 8 and 192 DF,  p-value: < 2.2e-16
```

```r
print("=========================================================================")
```

```
## [1] "========================================================================="
```

```r
print("both")
```

```
## [1] "both"
```

```r
stepwise_model <- stepAIC(full_model, direction = "both")
```

```
## Start:  AIC=20.68
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education + Urban + US
##
##               Df Sum of Sq    RSS     AIC
## - US           1      0.00 197.71  18.681
## - Urban        1      0.22 197.92  18.900
## - Education    1      1.17 198.88  19.865
## <none>                     197.71  20.680
## - Population   1      2.85 200.56  21.562
## - Income       1     31.79 229.50  48.654
## - Advertising  1     41.47 239.18  56.953
## - Age          1     89.76 287.46  93.916
## - CompPrice    1    250.23 447.94 183.071
## - ShelveLoc    2    586.90 784.61 293.737
## - Price        1    622.58 820.29 304.678
##
## Step:  AIC=18.68
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education + Urban
##
##               Df Sum of Sq    RSS     AIC
## - Urban        1      0.22 197.92  16.900
## - Education    1      1.18 198.89  17.874
## <none>                     197.71  18.681
## - Population   1      3.12 200.83  19.826
## + US           1      0.00 197.71  20.680
## - Income       1     31.80 229.50  46.656
## - Advertising  1     80.57 278.28  85.388
## - Age          1     89.76 287.47  91.918
## - CompPrice    1    250.24 447.95 181.077
## - ShelveLoc    2    591.23 788.94 292.844
## - Price        1    628.29 826.00 304.072
##
## Step:  AIC=16.9
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age + Education
##
##               Df Sum of Sq    RSS     AIC
## - Education    1      1.21 199.14  16.129
## <none>                     197.92  16.900
## - Population   1      3.07 200.99  17.991
## + Urban        1      0.22 197.71  18.681
## + US           1      0.00 197.92  18.900
## - Income       1     31.70 229.62  44.757
## - Advertising  1     81.46 279.38  84.183
## - Age          1     90.07 288.00  90.288
## - CompPrice    1    250.12 448.04 179.117
## - ShelveLoc    2    592.75 790.67 291.285
## - Price        1    628.89 826.81 302.269
##
## Step:  AIC=16.13
```

```
## Sales ~ CompPrice + Income + Advertising + Population + Price +
##     ShelveLoc + Age
##
##               Df Sum of Sq    RSS     AIC
## <none>                     199.14  16.129
## + Education    1      1.21 197.92  16.900
## - Population   1      3.74 202.88  17.870
## + Urban        1      0.25 198.89  17.874
## + US           1      0.01 199.13  18.117
## - Income       1     35.19 234.32  46.833
## - Advertising  1     81.24 280.38  82.901
## - Age          1     89.03 288.17  88.406
## - CompPrice    1    250.53 449.67 177.846
## - ShelveLoc    2    600.72 799.86 291.608
## - Price        1    629.40 828.54 300.688
```

```r
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income + Advertising + Population +
##     Price + ShelveLoc + Age, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6274 -0.6989 -0.0089  0.6604  3.3643
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.6995535  0.7589752   6.192 3.53e-09 ***
## CompPrice        0.0942562  0.0060647  15.542  < 2e-16 ***
## Income           0.0155920  0.0026769   5.825 2.38e-08 ***
## Advertising      0.1041845  0.0117716   8.850 5.73e-16 ***
## Population       0.0009621  0.0005065   1.899    0.059 .
## Price           -0.0938453  0.0038096 -24.634  < 2e-16 ***
## ShelveLocGood    5.1971084  0.2163206  24.025  < 2e-16 ***
## ShelveLocMedium  2.1145656  0.1740942  12.146  < 2e-16 ***
## Age             -0.0437138  0.0047183  -9.265  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.018 on 192 degrees of freedom
## Multiple R-squared:  0.8701, Adjusted R-squared:  0.8647
## F-statistic: 160.8 on 8 and 192 DF,  p-value: < 2.2e-16
```

Explanation

The backward model started with all variables and removed the least useful ones. It ended with 8 predictors, including CompPrice, Income, Advertising, Population, Price, ShelveLoc, and Age. It has an adjusted R^2 of 0.8647 and residual standard error of 1.018, which are both very solid. The model is simple and excludes irrelevant variables like Urban, US, and Education.

The forward selection model started with no variables and added the most significant ones step by step. It also selected the same final set of 8 variables as backward elimination. The adjusted R^2 and residuals are identical to the backward model, meaning both procedures converged to the same solution.

The both directions method combines forward and backward steps. It also ended with the exact same model as forward and backward selection. All three methods selected the same 8 predictors and achieved the same fit and accuracy.

So, all three methods produced the same final model, so they are equally good in terms of performance. The final model with 8 predictors is the best because it balances accuracy (adjusted $R^2 = 0.8647$) and simplicity, removing non-significant variables while retaining the most important ones.

```
# Predict on test set
pred_backward <- predict(backward_model, newdata = test_data)
pred_forward <- predict(forward_model, newdata = test_data)
pred_stepwise <- predict(stepwise_model, newdata = test_data)

# Evaluate accuracy (using pracma::rmserr)
print("backward")
```

```
## [1] "backward"
```

```
do.call(cbind, rmserr(test_data$Sales, pred_backward))
```

```
##            mae      mse     rmse mape      nmse      rstd
## [1,] 0.8411262 1.105686 1.051516  Inf 0.1335124 0.1411964
```

```
print("forward")
```

```
## [1] "forward"
```

```
do.call(cbind, rmserr(test_data$Sales, pred_forward))
```

```
##            mae      mse     rmse mape      nmse      rstd
## [1,] 0.8411262 1.105686 1.051516  Inf 0.1335124 0.1411964
```

```
print("both")
```

```
## [1] "both"
```

```
do.call(cbind, rmserr(test_data$Sales, pred_stepwise))
```

```
##            mae      mse     rmse mape      nmse      rstd
## [1,] 0.8411262 1.105686 1.051516  Inf 0.1335124 0.1411964
```

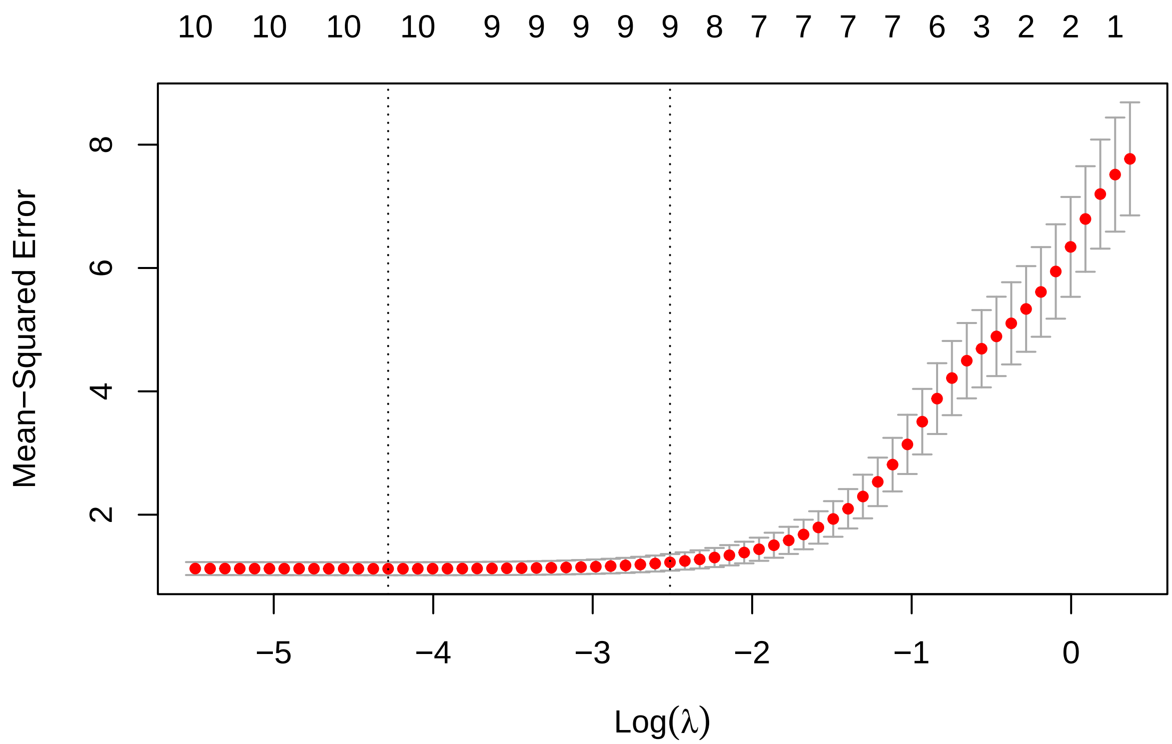
Explanations All three models — backward, forward, and stepwise (both directions) — resulted in exactly the same model, both in terms of predictors selected and performance metrics. Since they yield the same predictive accuracy and variable set, none is better than the others in this case — they all converged to the same optimal model. So, we can confidently choose any of them, but typically stepwise (both) is preferred because it checks additions and deletions at each step, offering more flexibility in general. —

(v) (2 pt)

Perform variable selection using the LASSO and select the best model.

```
library(glmnet)

# Prepare the data
x_train <- model.matrix(Sales ~ ., data = train_data)[, -1]  # Remove intercept
y_train <- train_data$Sales

x_test <- model.matrix(Sales ~ ., data = test_data)[, -1]
y_test <- test_data$Sales

# Fit LASSO using cross-validation
```

```r
cv_lasso <- cv.glmnet(x_train, y_train, alpha = 1)

# Plot the cross-validated MSE
plot(cv_lasso)
```




```r
# Best lambda (minimizes error)
best_lambda <- cv_lasso$lambda.min
best_lambda
```

```
## [1] 0.01380456
```

```r
# Fit LASSO with the best lambda
lasso_model <- glmnet(x_train, y_train, alpha = 1, lambda = best_lambda)

# Coefficients of selected features
coef(lasso_model)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)      5.3402544923
## CompPrice        0.0914803371
## Income           0.0145536334
## Advertising      0.1024687968
## Population       0.0007640299
## Price           -0.0920818142
## ShelveLocGood    5.0991716276
## ShelveLocMedium  2.0391071357
```

```
## Age               -0.0427011852
## Education         -0.0271602851
## UrbanYes           0.0328146654
## USYes              .
```

```r
# Predict on test data
lasso_pred <- predict(lasso_model, s = best_lambda, newx = x_test)

# Calculate prediction accuracy
library(pracma)
do.call(cbind, rmserr(y_test, lasso_pred))
```

```
##            mae      mse    rmse mape      nmse      rstd
## [1,] 0.8339726 1.095896 1.04685  Inf 0.1323302 0.1405699
```

The LASSO cross-validation plot shows how the mean squared error changes with different values of the regularization parameter log(lambda). The curve reaches its minimum around log(lambda) is approximately -4.3, which corresponds to the lambda value you used (lambda = 0.0138), indicated by the left vertical dotted line. This choice minimizes the prediction error while keeping 11 variables in the model, excluding only USYes. The right vertical line represents a larger(lambda.1se), which would give a simpler model with fewer variables but slightly higher error. Overall, your selected lambda provides the best balance between model accuracy and complexity, and the earlier results confirm it performs slightly better than stepwise regression.

(vi) (2 pt)

Which of the two models yield the best prediction based on your test set? ### Model Comparison: LASSO vs. Stepwise Regression

Metric	LASSO	Stepwise Regression
MAE	0.8340	0.8411
MSE	1.0959	1.1057
RMSE	1.0469	1.0515
MAPE	Inf	Inf
NMSE	0.1323	0.1335
rSTD	0.1406	0.1412
# Variables	11	12

Conclusion: LASSO performs slightly better in terms of error metrics and produces a simpler model by excluding irrelevant variables.

Explanation: Based on the test set results, the LASSO model yields the best prediction. It has a slightly lower RMSE (1.0469 vs. 1.0515) and NMSE (0.1323 vs. 0.1335) compared to the stepwise regression models, indicating better accuracy and generalization, while also simplifying the model by eliminating an irrelevant variable (USYes).
