Simple template for R Markdown

for Advanced Methods for Regression and Classification

Prof. Peter Filzmoser

01.10.2024

```r
data(College,package="ISLR")
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

```r
summary(College)
```

```
##  Private        Apps           Accept          Enroll        Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##            Median : 1558   Median : 1110   Median : 434   Median :23.00
##            Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##            3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##            Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
```

```
##    Room.Board       Books          Personal          PhD        
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00  
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00  
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00  
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66  
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00  
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00  
##     Terminal       S.F.Ratio      perc.alumni        Expend     
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186  
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751  
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377  
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660  
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830  
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233  
##    Grad.Rate     
##  Min.   : 10.00  
##  1st Qu.: 53.00  
##  Median : 65.00  
##  Mean   : 65.46  
##  3rd Qu.: 78.00  
##  Max.   :118.00
```

Our goal is to find a linear regression model which allows to predict the variable Apps, i.e. the number of applications received, using the remaining variables except of the variables Accept and Enroll.

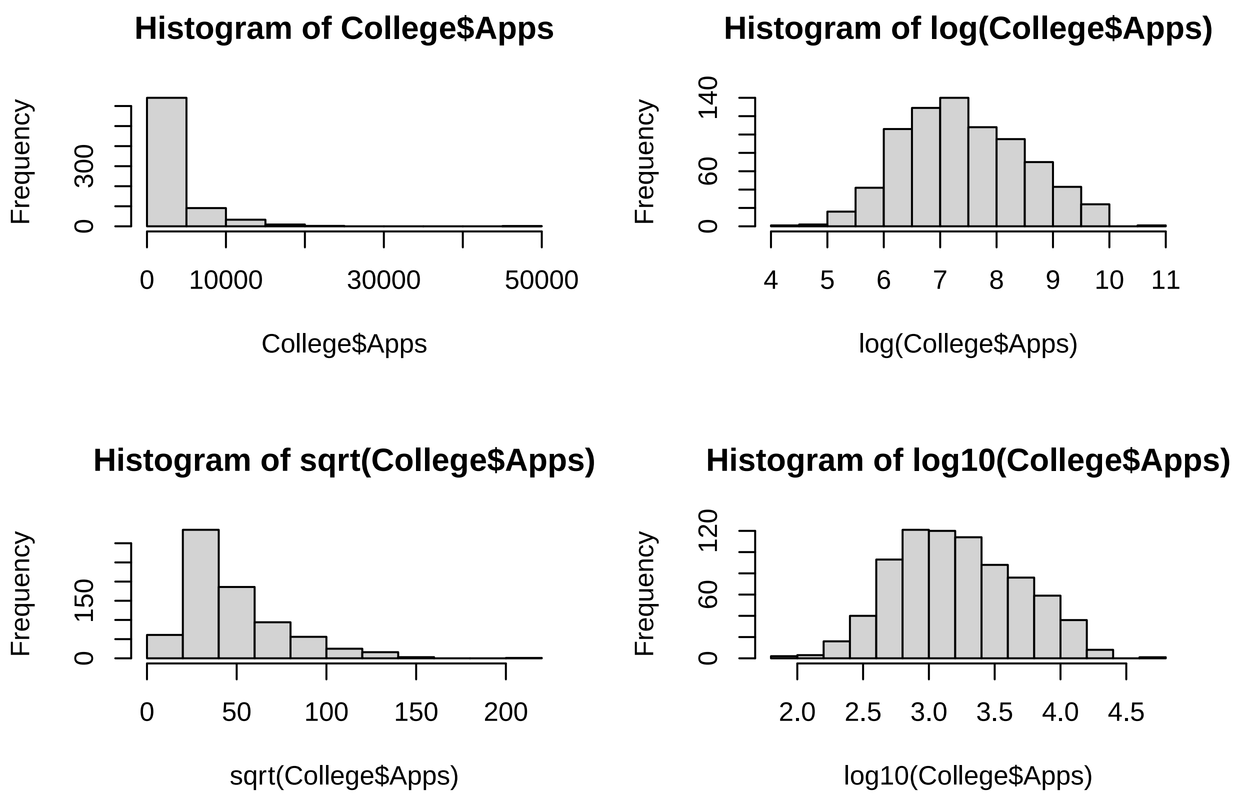
For the following tasks, split the data randomly into training and test data (about 2/3 and 1/3), build the model with the training data, and evaluate the model using the RMSE as a criterion.

split the data into training and test data:

```
set.seed(123)
n <- nrow(College)
train <- sample(1:n, n/3)
test <- -train
train.data <- College[train,]
test.data <- College[test,]
```

1. Look first at your data. Is any preprocessing necessary or useful? Argue why a log-transformation of the response variable can be useful. Continue with log(Apps) as the response.

```
par(mfrow=c(2,2))
hist(College$Apps)
hist(log(College$Apps))
hist(sqrt(College$Apps))
hist(log10(College$Apps))
```




```r
College$logApps <- log(College$Apps)
College<-College[-c(2,3,4)]
train.data <- College[train,]
test.data <- College[test,]
```

2. Full model: Estimate the full regression model and interpret the results.

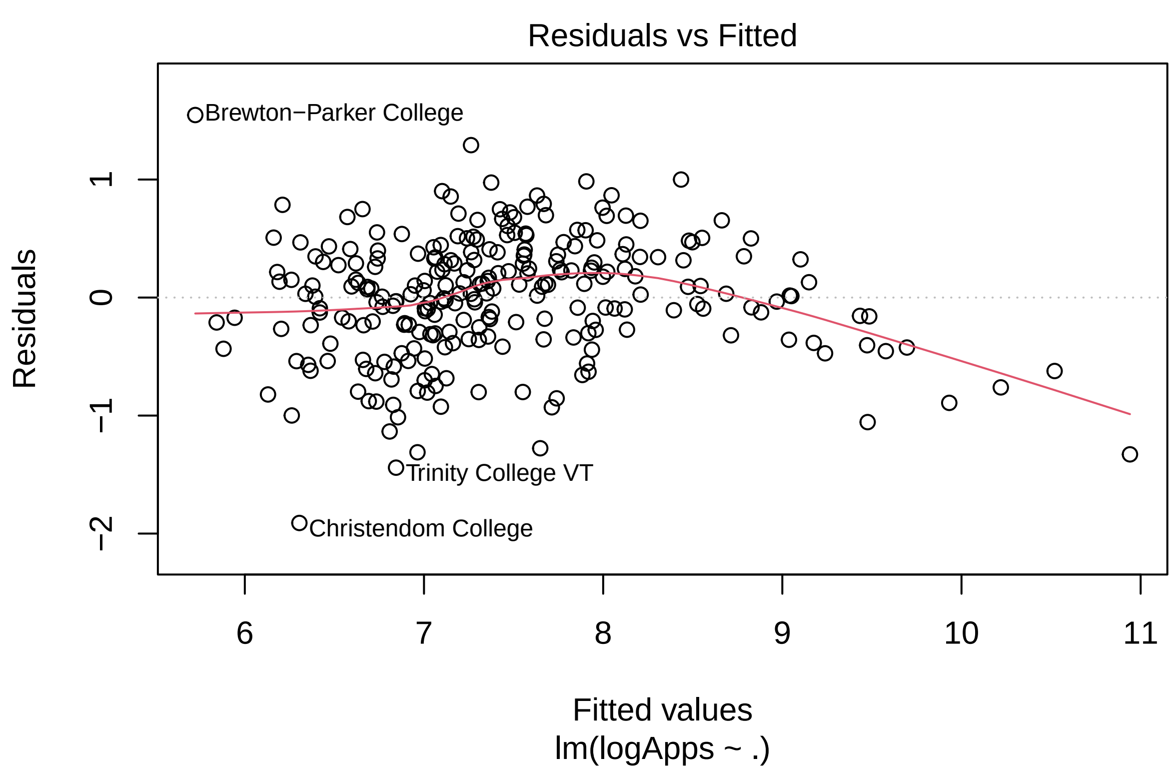
(a) or that purpose, apply the function lm() to compute the estimator – for details see course notes. Interpret the outcome of summary(res), where res is the output from the lm() function. Which variables contribute to explaining the response variable? Look at diagnostics plots with plot(res). Are the model assumptions fulfilled?

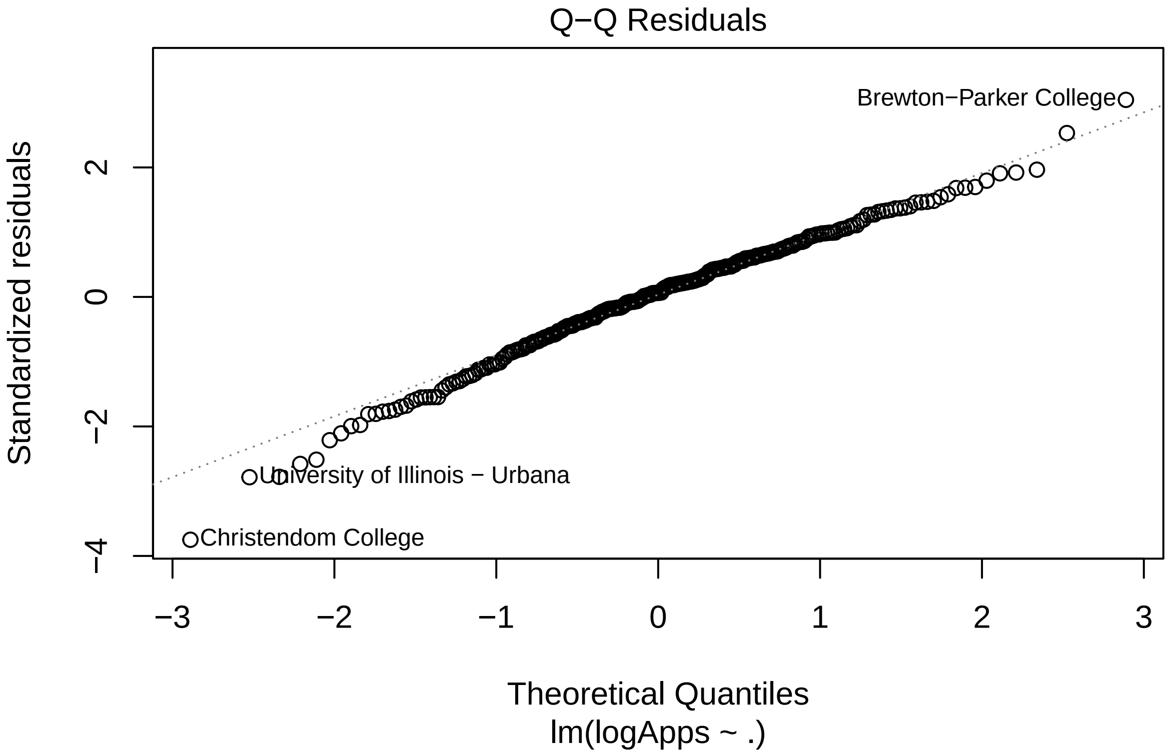
```r
res <- lm(logApps ~ ., data=train.data)
summary(res)
```

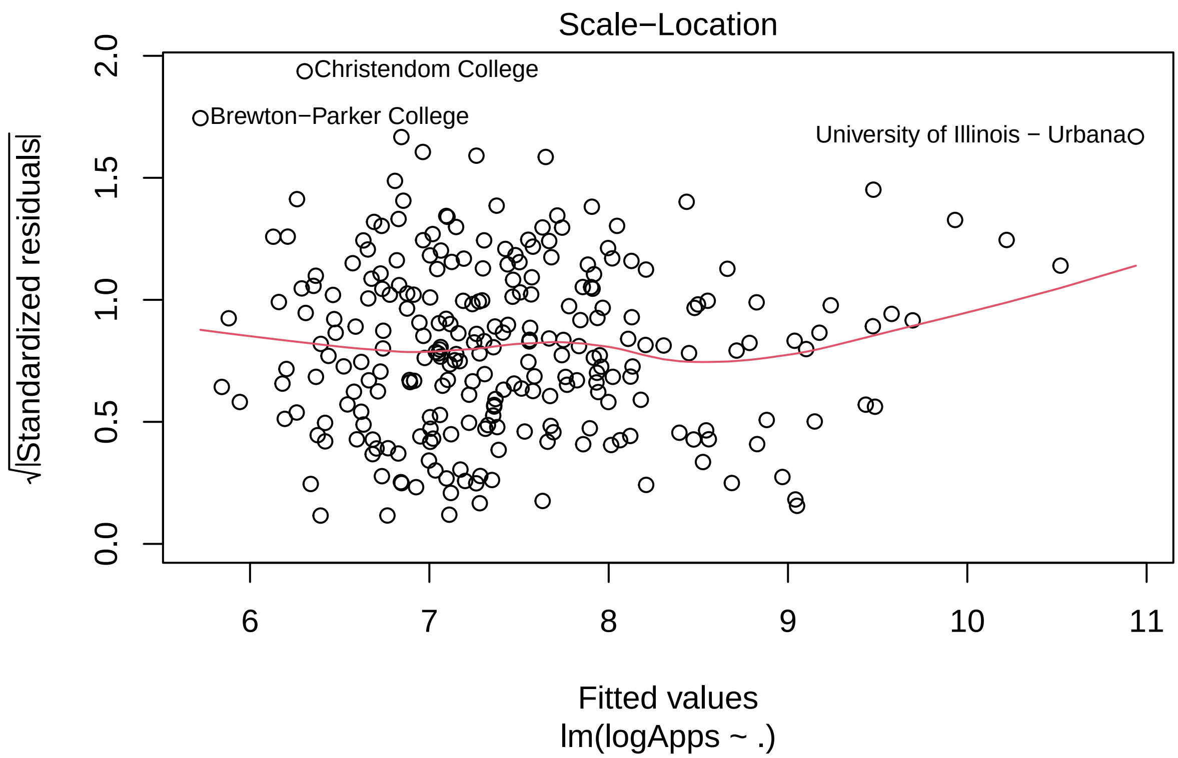
```
##
## Call:
## lm(formula = logApps ~ ., data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.91006 -0.30754  0.03222  0.34620  1.54630
##
```

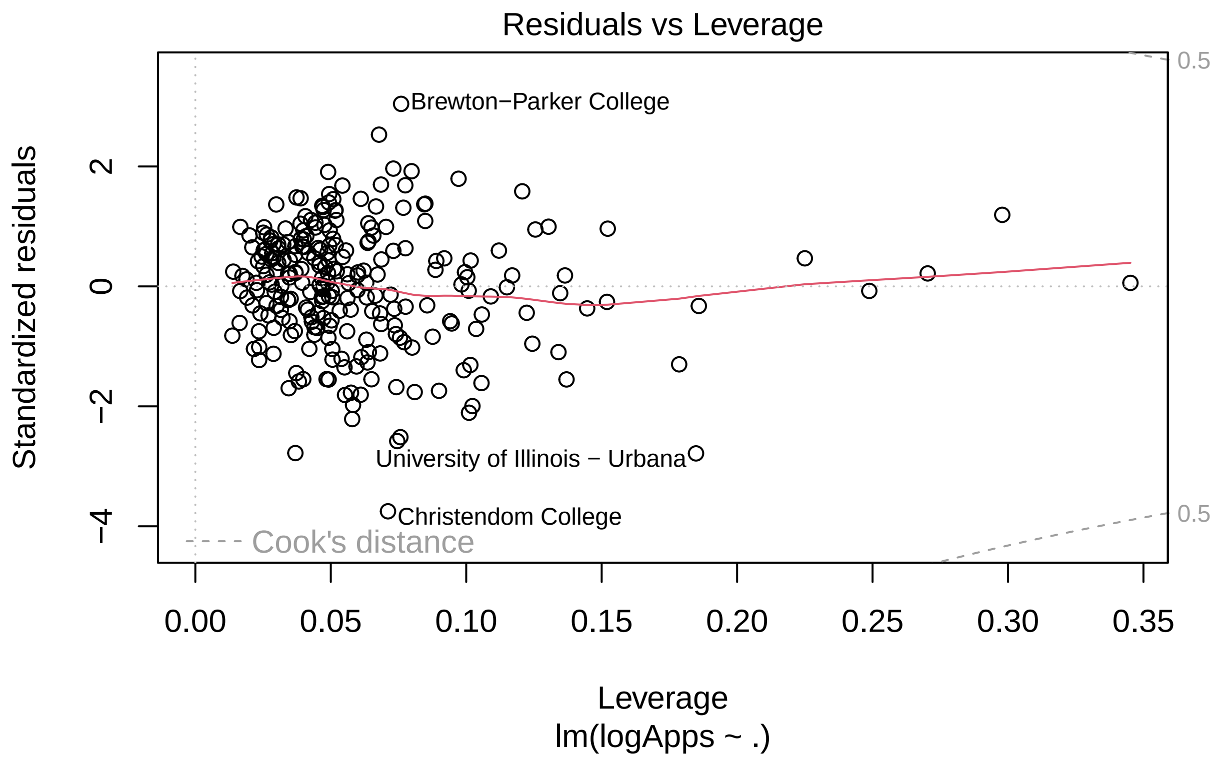
```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.271e+00  3.954e-01  10.803  < 2e-16 ***
## PrivateYes  -5.474e-01  1.360e-01  -4.024 7.63e-05 ***
## Top10perc   -3.447e-03  4.771e-03  -0.723   0.4706
## Top25perc   -2.501e-03  3.973e-03  -0.630   0.5295
## F.Undergrad  1.406e-04  1.237e-05  11.370  < 2e-16 ***
## P.Undergrad -2.048e-05  3.209e-05  -0.638   0.5239
## Outstate     6.732e-05  1.675e-05   4.019 7.81e-05 ***
## Room.Board   7.726e-05  3.964e-05   1.949   0.0524 .
## Books        4.144e-04  2.107e-04   1.966   0.0504 .
## Personal     1.891e-05  5.460e-05   0.346   0.7294
## PhD          7.245e-03  4.773e-03   1.518   0.1304
## Terminal    -5.793e-03  5.168e-03  -1.121   0.2634
## S.F.Ratio    5.590e-02  1.306e-02   4.280 2.70e-05 ***
## perc.alumni -3.694e-03  3.626e-03  -1.019   0.3094
## Expend       4.859e-05  1.214e-05   4.004 8.29e-05 ***
## Grad.Rate    1.180e-02  2.973e-03   3.970 9.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5285 on 243 degrees of freedom
## Multiple R-squared:  0.7376, Adjusted R-squared:  0.7214
## F-statistic: 45.54 on 15 and 243 DF,  p-value: < 2.2e-16
```

```r
plot(res)
```




Q−Q Residuals
Brewton−Parker College
University of Illinois − Urbana
Christendom College
Standardized residuals
2
0
−2
−4
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(logApps ~ .)


Scale−Location
Christendom College
Brewton−Parker College
University of Illinois − Urbana
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
6
7
8
9
10
11
Fitted values
lm(logApps ~ .)



predict the number of applications for the test data:

```
pred <- predict(res, newdata=test.data)
```

calculate the RMSE:

```
rmse <- sqrt(mean((test.data$logApps - pred)^2))
rmse
```

```
## [1] 0.6259921
```

Now we check what variables are important for the prediction:

```
library(caret)
```

```
## Lade nötiges Paket: ggplot2
```

```
## Warning: Paket 'ggplot2' wurde unter R Version 4.4.1 erstellt
```

```
## Lade nötiges Paket: lattice
```

```
varImp(res)
```

```
##                Overall
## PrivateYes   4.0244074
## Top10perc    0.7226205
## Top25perc    0.6296423
## F.Undergrad 11.3702226
## P.Undergrad  0.6382029
## Outstate     4.0186851
## Room.Board   1.9490630
## Books        1.9660947
## Personal     0.3463857
## PhD          1.5178184
## Terminal     1.1209950
## S.F.Ratio    4.2795262
## perc.alumni  1.0186734
## Expend       4.0037481
## Grad.Rate    3.9696716
```

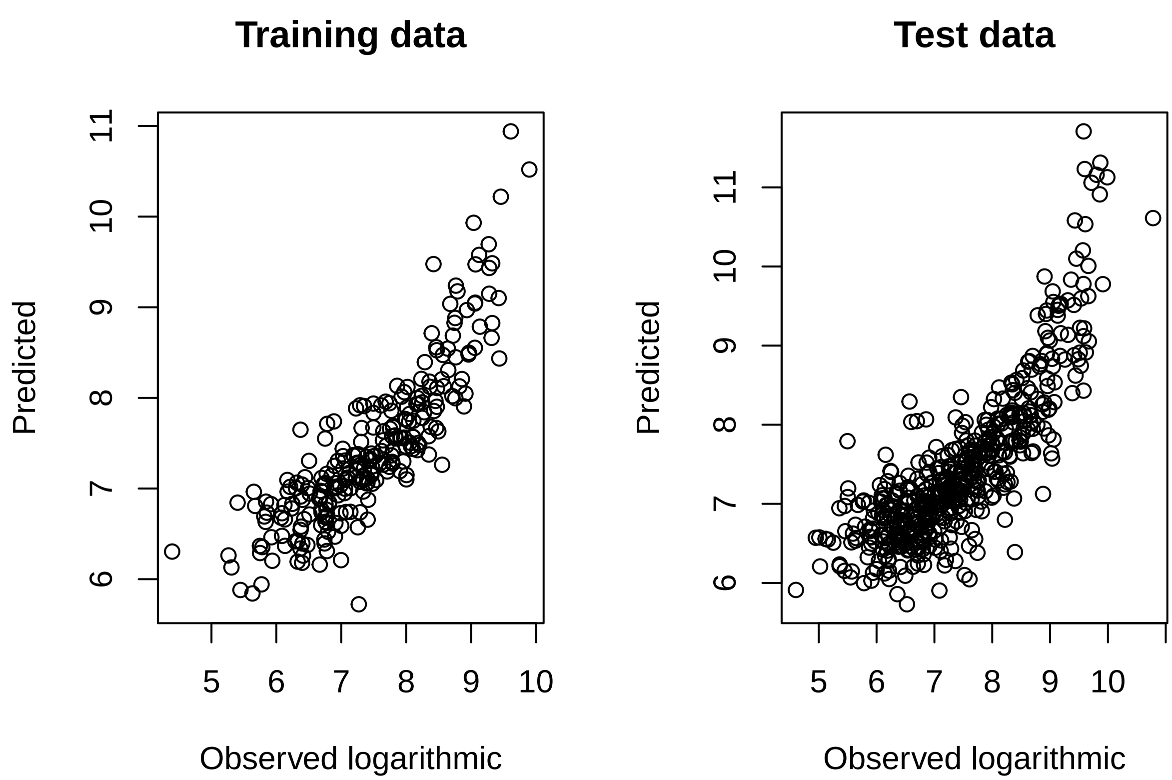
(b) ow we try to manually compute the LS coefficients, in the same way as lm(). Thus, replace from the above command lm() by model.matrix(). This gives you the matrix X as it is used to estimate the regression coefficients. Now apply the formula to compute the LS estimator. You can do matrix multipli- cation in R by %*%, and the inverse of a matrix is computed with solve(). How is R handling binary variables (Private), and how can you interpret the corresponding regression coefficient? Compare the resulting coefficients with those obtained from lm(). Do you get the same result?

```r
X <- model.matrix(logApps ~ . , data=train.data)
y <- train.data$logApps
beta <- solve(t(X) %*% X) %*% t(X) %*% y
beta
```

```
##                      [,1]
## (Intercept)  4.271060e+00
## PrivateYes  -5.473653e-01
## Top10perc   -3.447438e-03
## Top25perc   -2.501336e-03
## F.Undergrad  1.406177e-04
## P.Undergrad -2.047850e-05
## Outstate     6.731722e-05
## Room.Board   7.726166e-05
## Books        4.143589e-04
## Personal     1.891396e-05
## PhD          7.245154e-03
## Terminal    -5.793214e-03
## S.F.Ratio    5.589919e-02
## perc.alumni -3.693622e-03
## Expend       4.859302e-05
## Grad.Rate    1.180374e-02
```

(c) Compare graphically the observed and the predicted values of the response variable – once only for the training data, and once for the test data. What do you think about the prediction performance of your model?

```
par(mfrow=c(1,2))
plot(train.data$logApps, predict(res), xlab="Observed logarithmic", ylab="Predicted", main="Training da
plot(test.data$logApps, pred, xlab="Observed logarithmic", ylab="Predicted", main="Test data")
```



(d) Compute the RMSE separately for training and test data, and compare the values. What do you conclude?

```
pred.train <- predict(res, newdata=train.data)
rmse.train <- sqrt(mean((train.data$logApps - pred.train)^2))
rmse.train
```

```
## [1] 0.5119145
```

```
rmse
```

```
## [1] 0.6259921
```

3. Reduced model: Exclude all input variables from the model which were not significant in 2(a), and compute the LS-estimator.

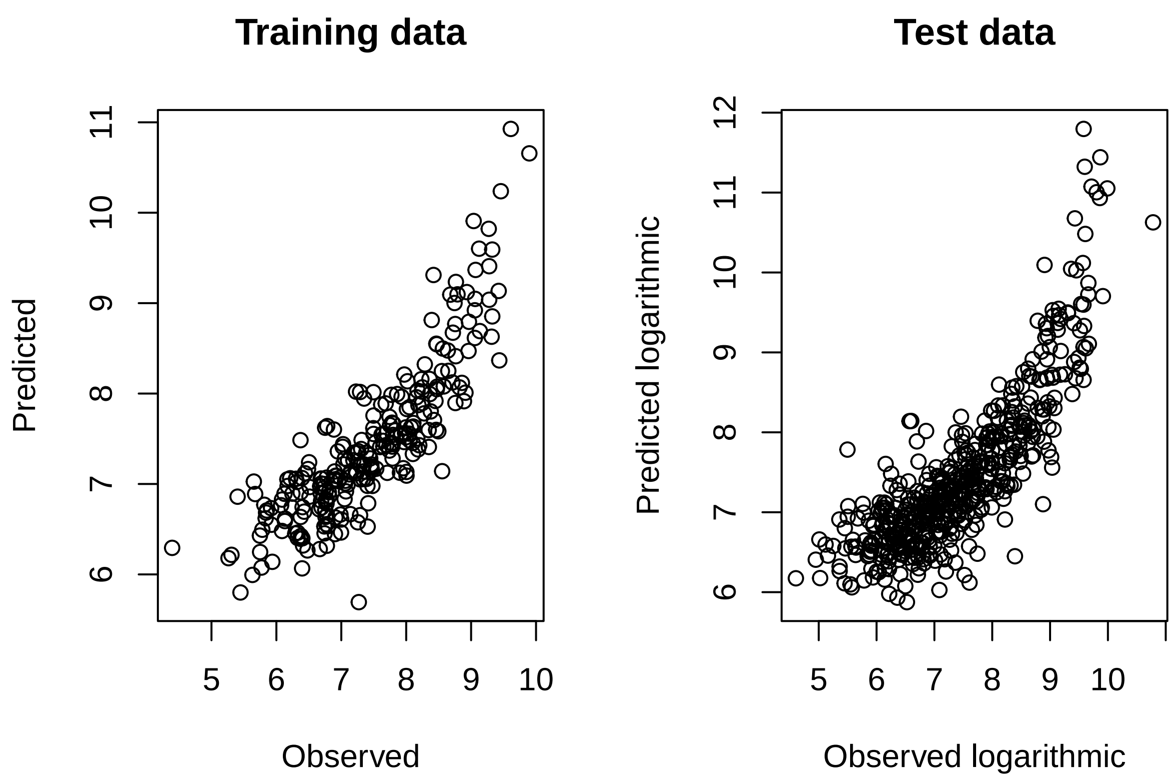
```
reduced.model<- lm(logApps ~ . -Top25perc -Top10perc -P.Undergrad -Personal -PhD -Terminal -perc.alumni
summary(reduced.model)
```

```
##
## Call:
## lm(formula = logApps ~ . - Top25perc - Top10perc - P.Undergrad -
##     Personal - PhD - Terminal - perc.alumni, data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.89921 -0.29947  0.03818  0.35191  1.57624
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.300e+00  3.141e-01  13.689  < 2e-16 ***
## PrivateYes  -5.507e-01  1.310e-01  -4.204 3.66e-05 ***
## F.Undergrad  1.362e-04  1.066e-05  12.778  < 2e-16 ***
## Outstate     5.973e-05  1.527e-05   3.911 0.000118 ***
## Room.Board   7.531e-05  3.882e-05   1.940 0.053518 .
## Books        4.096e-04  2.043e-04   2.005 0.046037 *
## S.F.Ratio    5.855e-02  1.279e-02   4.578 7.42e-06 ***
## Expend       4.263e-05  1.008e-05   4.230 3.28e-05 ***
## Grad.Rate    9.638e-03  2.606e-03   3.699 0.000266 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5296 on 250 degrees of freedom
## Multiple R-squared:  0.7289, Adjusted R-squared:  0.7203
## F-statistic: 84.03 on 8 and 250 DF,  p-value: < 2.2e-16
```

(a) Are now all input variables significant in the model? Why is this not to be expected in general?

Yes. Various Reasons such as Overfitting, colinearity, sample size limitations, noise and bias, etc. ### (b) Visualize the fit and the prediction from the new model, see 2(c).

```
par(mfrow=c(1,2))
pred <- predict(reduced.model, newdata=test.data)
plot(train.data$logApps, predict(reduced.model), xlab="Observed", ylab="Predicted", main="Training data
plot(test.data$logApps, pred, xlab="Observed logarithmic", ylab="Predicted logarithmic", main="Test data
```



(c) Compute the RMSE for the new model, see 2(d). What would we expect?

```
pred.train <- predict(reduced.model, newdata=train.data)
rmse.train <- sqrt(mean((train.data$logApps - pred.train)^2))
rmse.train
```

```
## [1] 0.5203241
```

```
rmse <- sqrt(mean((test.data$logApps - pred)^2))
rmse
```

```
## [1] 0.6124449
```

(d) Compare the two models with anova(). What can you conclude?

```
anova(res,reduced.model)
```

```
## Analysis of Variance Table
##
## Model 1: logApps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + Personal + PhD + Terminal +
```

```
##     S.F.Ratio + perc.alumni + Expend + Grad.Rate
## Model 2: logApps ~ (Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + Personal + PhD + Terminal +
##     S.F.Ratio + perc.alumni + Expend + Grad.Rate) - Top25perc -
##     Top10perc - P.Undergrad - Personal - PhD - Terminal - perc.alumni
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    243 67.873
## 2    250 70.121 -7   -2.2483 1.1499 0.3326
```

4. Perform variable selection based on stepwise regression, using the function step(), see help file and course notes. Perform both, forward selection (start from the empty model) and backward selection (start from the full model). Compare the resulting models with the RMSE, and with plots of response versus predicted values.

```
full_model <- lm(logApps~ .,data=train.data)
empty_model <- lm(logApps ~ 1, data = train.data)
forward_model <- step(empty_model,direction = "forward",scope=full_model)
```

```
## Start:  AIC=1.68
## logApps ~ 1
```

```
backward_model <-step(full_model,direction = "backward")
```

```
## Start:  AIC=-314.85
## logApps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + Personal + PhD + Terminal +
##     S.F.Ratio + perc.alumni + Expend + Grad.Rate
##
##               Df Sum of Sq     RSS     AIC
## - Personal     1     0.034  67.906 -316.72
## - Top25perc    1     0.111  67.983 -316.43
## - P.Undergrad  1     0.114  67.986 -316.42
## - Top10perc    1     0.146  68.018 -316.30
## - perc.alumni  1     0.290  68.162 -315.75
## - Terminal     1     0.351  68.224 -315.52
## <none>                      67.873 -314.85
## - PhD          1     0.643  68.516 -314.41
## - Room.Board   1     1.061  68.934 -312.83
## - Books        1     1.080  68.952 -312.76
## - Grad.Rate    1     4.401  72.274 -300.58
## - Expend       1     4.477  72.350 -300.31
## - Outstate     1     4.511  72.383 -300.19
## - Private      1     4.524  72.396 -300.14
## - S.F.Ratio    1     5.115  72.988 -298.03
## - F.Undergrad  1    36.110 103.983 -206.36
##
## Step:  AIC=-316.72
## logApps ~ Private + Top10perc + Top25perc + F.Undergrad + P.Undergrad +
##     Outstate + Room.Board + Books + PhD + Terminal + S.F.Ratio +
##     perc.alumni + Expend + Grad.Rate
##
```

```
##               Df Sum of Sq     RSS     AIC
## - P.Undergrad  1     0.106  68.012 -318.32
## - Top25perc    1     0.109  68.015 -318.31
## - Top10perc    1     0.142  68.048 -318.18
## - perc.alumni  1     0.315  68.221 -317.53
## - Terminal     1     0.352  68.258 -317.38
## <none>                      67.906 -316.72
## - PhD          1     0.664  68.570 -316.20
## - Room.Board   1     1.039  68.945 -314.79
## - Books        1     1.205  69.111 -314.17
## - Grad.Rate    1     4.371  72.277 -302.57
## - Outstate     1     4.490  72.396 -302.14
## - Private      1     4.492  72.398 -302.13
## - Expend       1     4.498  72.405 -302.11
## - S.F.Ratio    1     5.090  72.996 -300.00
## - F.Undergrad  1    36.733 104.639 -206.74
##
## Step:  AIC=-318.32
## logApps ~ Private + Top10perc + Top25perc + F.Undergrad + Outstate +
##     Room.Board + Books + PhD + Terminal + S.F.Ratio + perc.alumni +
##     Expend + Grad.Rate
##
##               Df Sum of Sq     RSS     AIC
## - Top25perc    1     0.116  68.128 -319.88
## - Top10perc    1     0.122  68.134 -319.85
## - perc.alumni  1     0.276  68.287 -319.27
## - Terminal     1     0.363  68.375 -318.94
## <none>                      68.012 -318.32
## - PhD          1     0.630  68.642 -317.93
## - Room.Board   1     0.979  68.991 -316.62
## - Books        1     1.213  69.224 -315.74
## - Private      1     4.448  72.459 -303.91
## - Expend       1     4.504  72.516 -303.71
## - Outstate     1     4.524  72.536 -303.64
## - Grad.Rate    1     4.824  72.836 -302.57
## - S.F.Ratio    1     5.084  73.096 -301.65
## - F.Undergrad  1    42.081 110.093 -195.58
##
## Step:  AIC=-319.88
## logApps ~ Private + Top10perc + F.Undergrad + Outstate + Room.Board +
##     Books + PhD + Terminal + S.F.Ratio + perc.alumni + Expend +
##     Grad.Rate
##
##               Df Sum of Sq     RSS     AIC
## - perc.alumni  1     0.294  68.422 -320.76
## - Terminal     1     0.381  68.509 -320.43
## <none>                      68.128 -319.88
## - PhD          1     0.583  68.711 -319.67
## - Top10perc    1     0.981  69.109 -318.17
## - Room.Board   1     0.989  69.117 -318.15
## - Books        1     1.198  69.327 -317.36
## - Private      1     4.413  72.541 -305.62
## - Outstate     1     4.482  72.610 -305.38
## - Grad.Rate    1     4.732  72.860 -304.49
```

```
## - S.F.Ratio    1     5.320  73.448 -302.41
## - Expend       1     5.367  73.496 -302.24
## - F.Undergrad  1    41.965 110.093 -197.57
##
## Step:  AIC=-320.76
## logApps ~ Private + Top10perc + F.Undergrad + Outstate + Room.Board +
##     Books + PhD + Terminal + S.F.Ratio + Expend + Grad.Rate
##
##               Df Sum of Sq     RSS     AIC
## - Terminal     1     0.515  68.937 -320.82
## <none>                      68.422 -320.76
## - PhD          1     0.694  69.117 -320.15
## - Room.Board   1     1.109  69.531 -318.60
## - Top10perc    1     1.147  69.569 -318.46
## - Books        1     1.341  69.763 -317.74
## - Outstate     1     4.244  72.666 -307.18
## - Private      1     4.416  72.838 -306.56
## - Grad.Rate    1     4.449  72.871 -306.45
## - Expend       1     5.389  73.811 -303.13
## - S.F.Ratio    1     5.604  74.026 -302.38
## - F.Undergrad  1    44.122 112.544 -193.87
##
## Step:  AIC=-320.82
## logApps ~ Private + Top10perc + F.Undergrad + Outstate + Room.Board +
##     Books + PhD + S.F.Ratio + Expend + Grad.Rate
##
##               Df Sum of Sq     RSS     AIC
## - PhD          1     0.196  69.133 -322.08
## <none>                      68.937 -320.82
## - Room.Board   1     0.966  69.903 -319.22
## - Top10perc    1     1.097  70.034 -318.73
## - Books        1     1.311  70.248 -317.94
## - Outstate     1     3.860  72.797 -308.71
## - Private      1     4.242  73.179 -307.35
## - Grad.Rate    1     4.630  73.567 -305.99
## - Expend       1     5.554  74.491 -302.75
## - S.F.Ratio    1     6.004  74.941 -301.19
## - F.Undergrad  1    43.609 112.546 -195.87
##
## Step:  AIC=-322.08
## logApps ~ Private + Top10perc + F.Undergrad + Outstate + Room.Board +
##     Books + S.F.Ratio + Expend + Grad.Rate
##
##               Df Sum of Sq     RSS     AIC
## <none>                      69.133 -322.08
## - Top10perc    1     0.988  70.121 -320.41
## - Room.Board   1     1.045  70.178 -320.20
## - Books        1     1.356  70.489 -319.05
## - Grad.Rate    1     4.623  73.757 -307.32
## - Outstate     1     4.825  73.958 -306.61
## - Private      1     5.058  74.191 -305.80
## - Expend       1     5.958  75.092 -302.67
## - S.F.Ratio    1     6.015  75.148 -302.48
## - F.Undergrad  1    46.737 115.871 -190.33
```

```r
summary(backward_model)
```

```
##
## Call:
## lm(formula = logApps ~ Private + Top10perc + F.Undergrad + Outstate +
##     Room.Board + Books + S.F.Ratio + Expend + Grad.Rate, data = train.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.84046 -0.30379  0.03197  0.36108  1.63316
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.175e+00  3.194e-01  13.071  < 2e-16 ***
## PrivateYes  -5.565e-01  1.304e-01  -4.268 2.80e-05 ***
## Top10perc   -5.455e-03  2.892e-03  -1.886   0.0605 .
## F.Undergrad  1.398e-04  1.077e-05  12.974  < 2e-16 ***
## Outstate     6.406e-05  1.537e-05   4.169 4.23e-05 ***
## Room.Board   7.494e-05  3.863e-05   1.940   0.0535 .
## Books        4.519e-04  2.045e-04   2.210   0.0280 *
## S.F.Ratio    5.926e-02  1.273e-02   4.654 5.29e-06 ***
## Expend       5.313e-05  1.147e-05   4.633 5.83e-06 ***
## Grad.Rate    1.096e-02  2.686e-03   4.081 6.05e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5269 on 249 degrees of freedom
## Multiple R-squared:  0.7327, Adjusted R-squared:  0.7231
## F-statistic: 75.86 on 9 and 249 DF,  p-value: < 2.2e-16
```

```r
# Function to calculate RMSE
rmse <- function(model) {
  predictions <- predict(model, newdata = test.data)
  sqrt(mean((test.data$logApps - predictions)^2)) }
rmse_forward <- rmse(forward_model)
rmse_backward <- rmse(backward_model)

print(paste("RMSE of Forward Model:", rmse_forward))
```

```
## [1] "RMSE of Forward Model: 1.10803589059827"
```

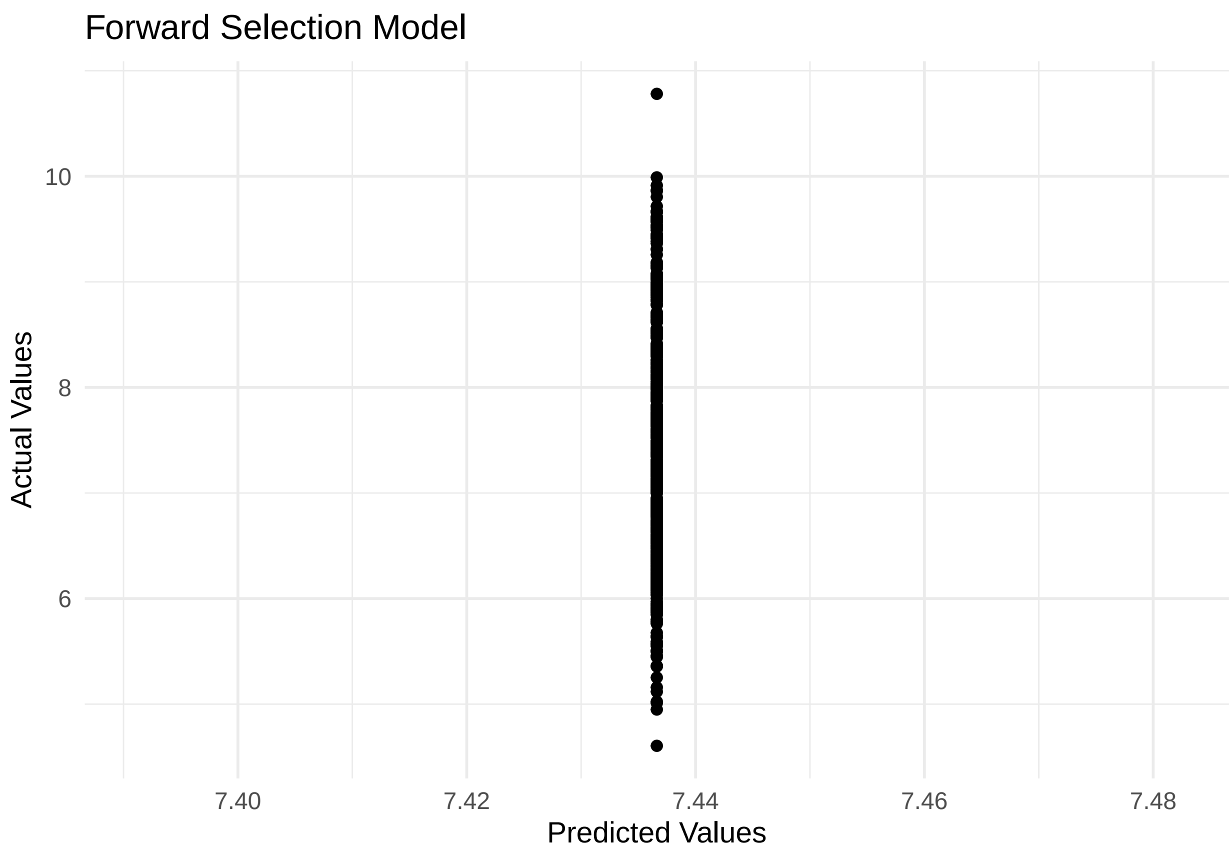
```r
print(paste("RMSE of Backward Model:", rmse_backward))
```

```
## [1] "RMSE of Backward Model: 0.63030470488214"
```

```r
plot_model <- function(model, title) {
  predictions <- predict(model, newdata = test.data)
  ggplot(test.data, aes(x = predictions, y = logApps)) +
    geom_point() +
    geom_smooth(method = "lm", color = "blue") +
    labs(title = title, x = "Predicted Values", y = "Actual Values") +
```

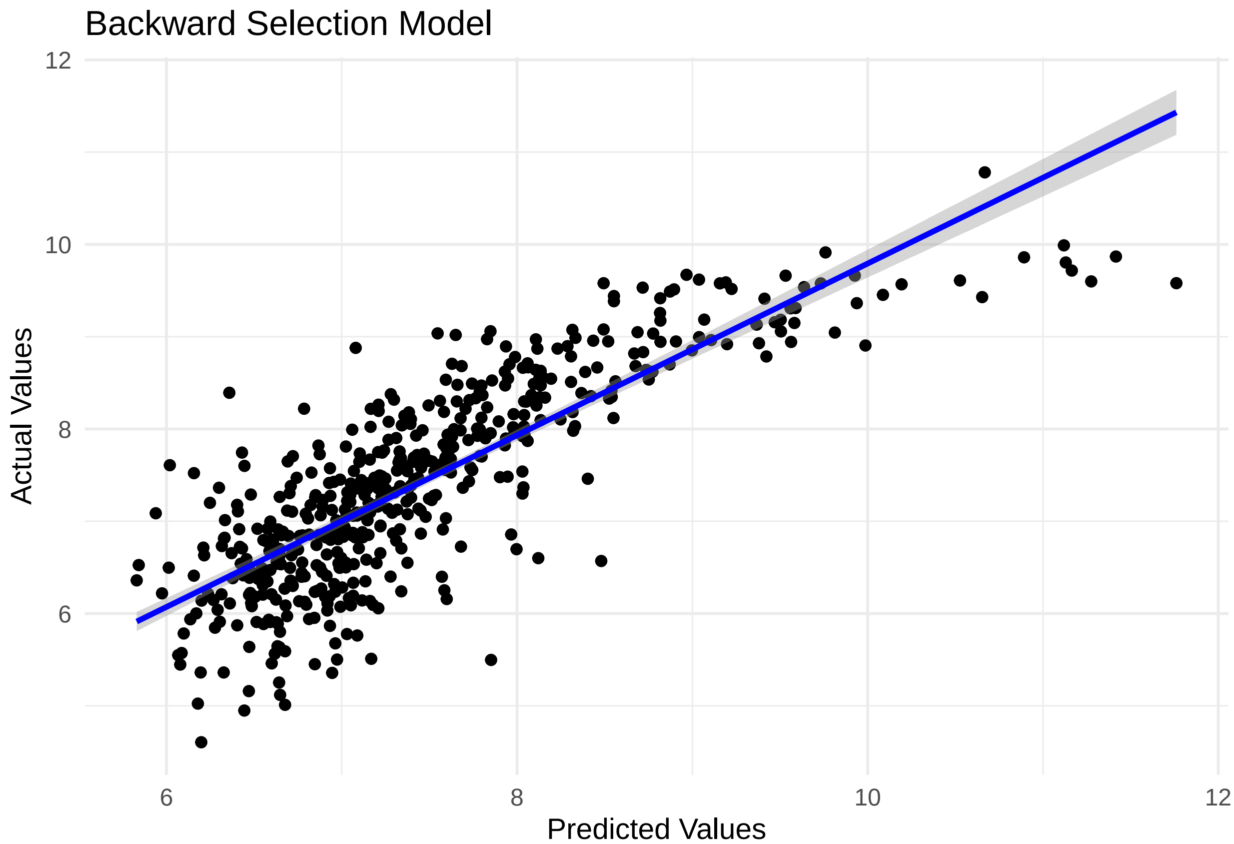
```
    theme_minimal() } # Plotting both models
plot1 <- plot_model(forward_model, "Forward Selection Model")
plot2 <- plot_model(backward_model, "Backward Selection Model") # Print plots
print(plot1)
```

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




```
print(plot2)
```

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



Backward Selection Model
Actual Values
Predicted Values
12
10
8
6
6
8
10
12