```r
library(car)
```

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

```r
library(reshape2)
library(ggplot2)
library(MASS)
library(interactions)

source("clean_data.R")


df <- remove_cols(df, c("Color", "Model"))

# Remove columns with only one observation and affected rows
res <- convert_categorical(df, categorical)
design <- as.data.frame(res$dummy)

singles <- c()
bad_idx <- c()
for (col in colnames(design)) {
  if (sum(design[, col] != 0) <= 1) {
    singles <- c(singles, col)
    bad_idx <- c(bad_idx, which(design[, col] != 0))
  }
}
singles
```

```
##  [1] "MakeFiat"            "MakeLexus"            "Fuel.TypePetrol + CNG"
##  [4] "LocationDak. Kannada" "LocationFaizabad"    "LocationGorakhpur"
##  [7] "LocationPurnea"      "LocationRohtak"       "LocationRudrapur"
## [10] "LocationSamastipur"  "LocationValsad"
```
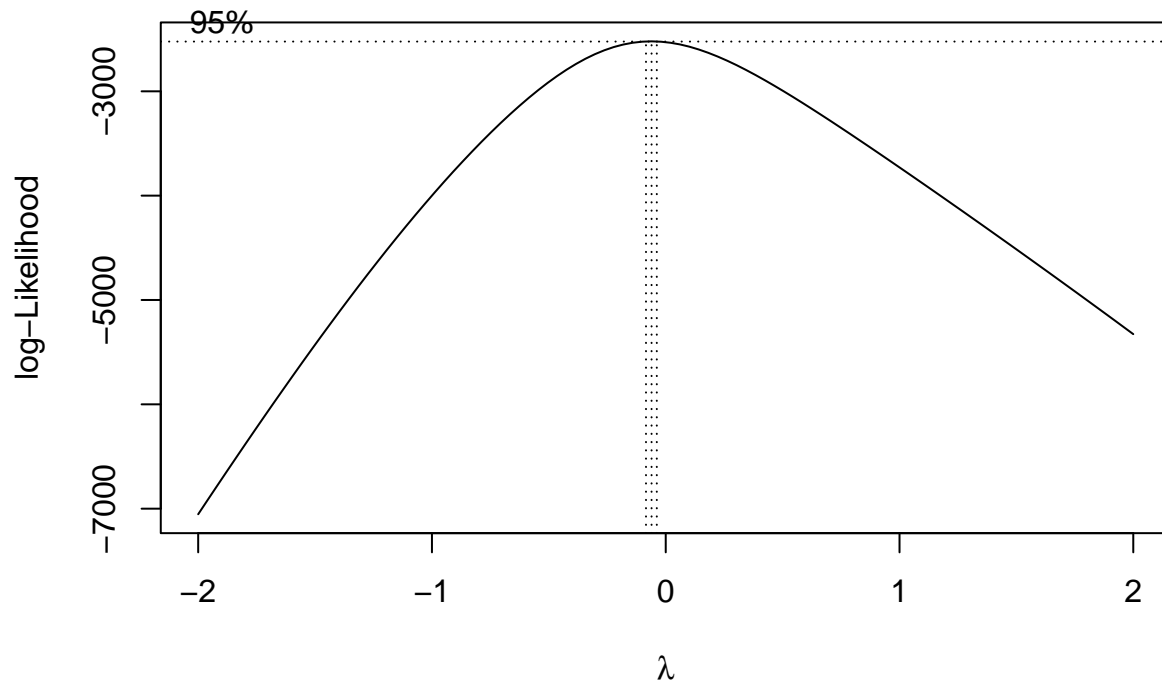
```r
bad_idx
```

```
##  [1]  662 1009 1169 1077  779  569  728  245  510  264  162
```

```r
df <- df[-bad_idx, ]
```

Box Cox

```r
bc <- boxcox(Price ~., data = df)
```

1

```
bc$x[which.max(bc$y)]
```

```
## [1] -0.06060606
```

For the sake of interpretability, we will use a log transformation, since $\lambda \approx 0$.

Iteratively Remove Multicollinear Regressors

```r
x <- df
model <- lm(log(Price) ~., data = x)
removed <- c()

finished <- F
while(!finished) {
  temp <- car::vif(model)[, "GVIF^(1/(2*Df))"]
  worst <- names(which.max(temp))
  if (length(temp) > 0 && temp[worst] > sqrt(5)) {
    x <- remove_cols(x, c(worst))
    model <- lm(log(Price) ~., data = x)

    removed <- c(removed, worst)
  } else {
    finished <- T
  }
}

removed
```

```
## [1] "Max.Torque.Value"  "Max.Power.RPM"     "Max.Power.Value"
## [4] "Engine"            "Fuel.Tank.Capacity" "Height"
```

```r
which(abs(rstudent(model)) > 4)
```

```
##   44  366  492 1536 1681
```

2

```
##    39  268  345 1071 1171
```

After inspecting these data points, we do not find a good reason to remove them (i.e., they are not clerical errors).

```r
vif(model, type = "predictor")
```

```
## GVIFs computed for predictors
```
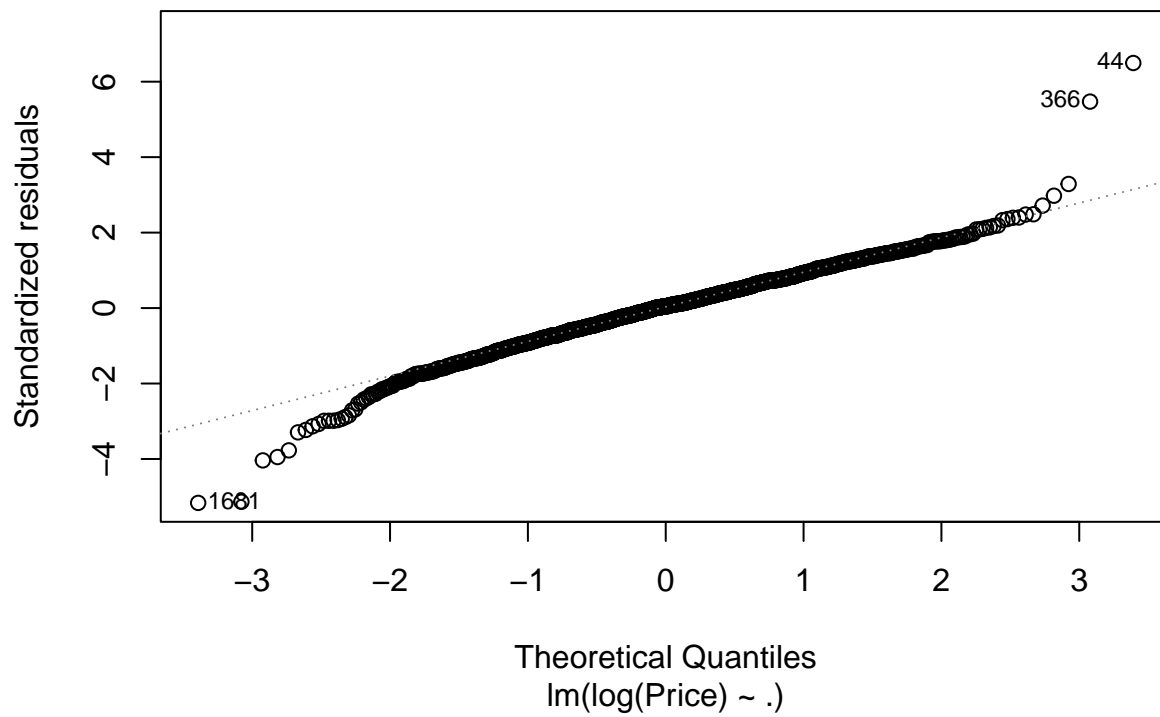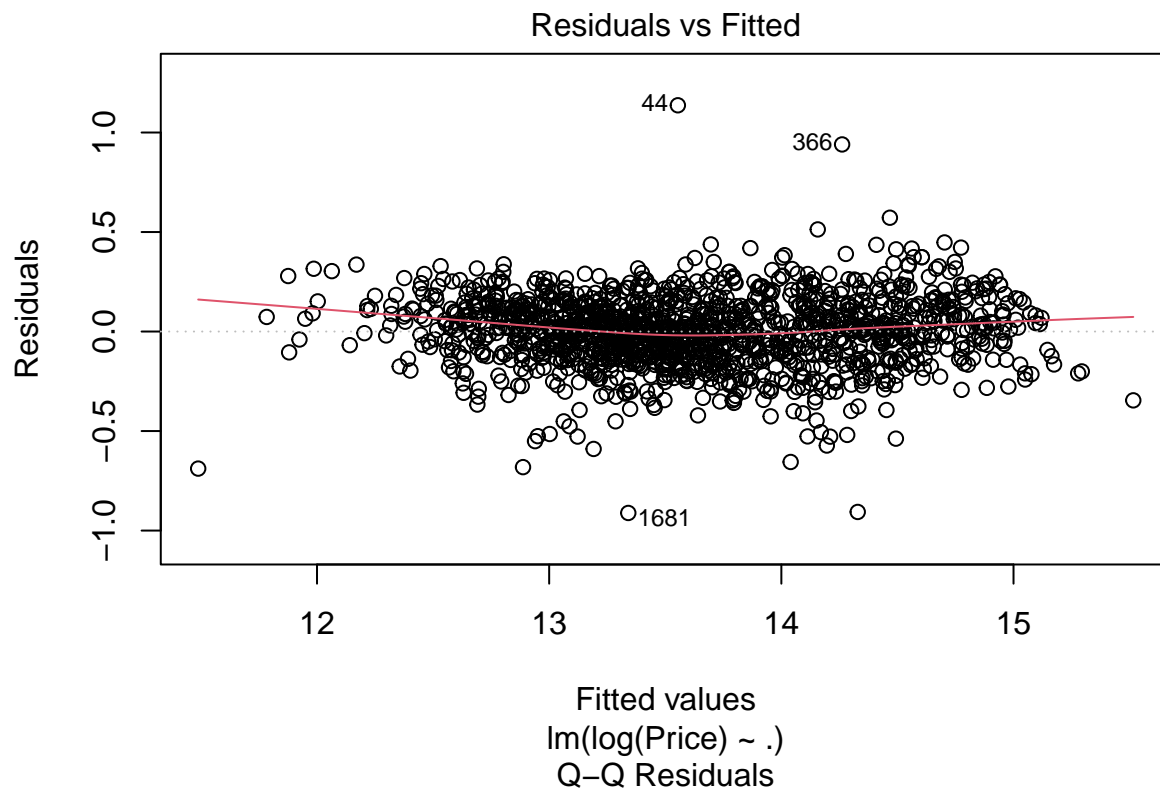
```
##                       GVIF Df GVIF^(1/(2*Df)) Interacts With
## Make             76.706634 25        1.090678             --
## Year              2.038829  1        1.427876             --
## Kilometer         2.089964  1        1.445671             --
## Fuel.Type         5.026586  2        1.497333             --
## Transmission      1.843543  1        1.357771             --
## Location         13.438914 63        1.020834             --
## Owner             1.628105  3        1.084627             --
## Seller.Type       1.329867  2        1.073871             --
## Max.Torque.RPM    4.220839  1        2.054468             --
## Drivetrain        3.872635  2        1.402819             --
## Length            4.511737  1        2.124085             --
## Width             4.596043  1        2.143838             --
## Seating.Capacity  2.506336  1        1.583141             --
##
## Make             Year, Kilometer, Fuel.Type, Transmission, Location, Owner, Seller.Type, Max.Torque.
## Year             Make, Kilometer, Fuel.Type, Transmission, Location, Owner, Seller.Type, Max.Torque.
## Kilometer        Make, Year, Fuel.Type, Transmission, Location, Owner, Seller.Type, Max.Torque.
## Fuel.Type        Make, Year, Kilometer, Transmission, Location, Owner, Seller.Type, Max.Torque.
## Transmission     Make, Year, Kilometer, Fuel.Type, Location, Owner, Seller.Type, Max.Torque.
## Location         Make, Year, Kilometer, Fuel.Type, Transmission, Owner, Seller.Type, Max.Torque.
## Owner            Make, Year, Kilometer, Fuel.Type, Transmission, Location, Seller.Type, Max.Torque.
## Seller.Type      Make, Year, Kilometer, Fuel.Type, Transmission, Location, Owner, Max.Torque.
## Max.Torque.RPM   Make, Year, Kilometer, Fuel.Type, Transmission, Location, Owner, Seller.Ty
## Drivetrain       Make, Year, Kilometer, Fuel.Type, Transmission, Location, Owner, Seller.Type,
## Length           Make, Year, Kilometer, Fuel.Type, Transmission, Location, Owner, Seller.Type, Max
## Width            Make, Year, Kilometer, Fuel.Type, Transmission, Location, Owner, Seller.Type, Max.T
## Seating.Capacity         Make, Year, Kilometer, Fuel.Type, Transmission, Location, Owner, Seller
```
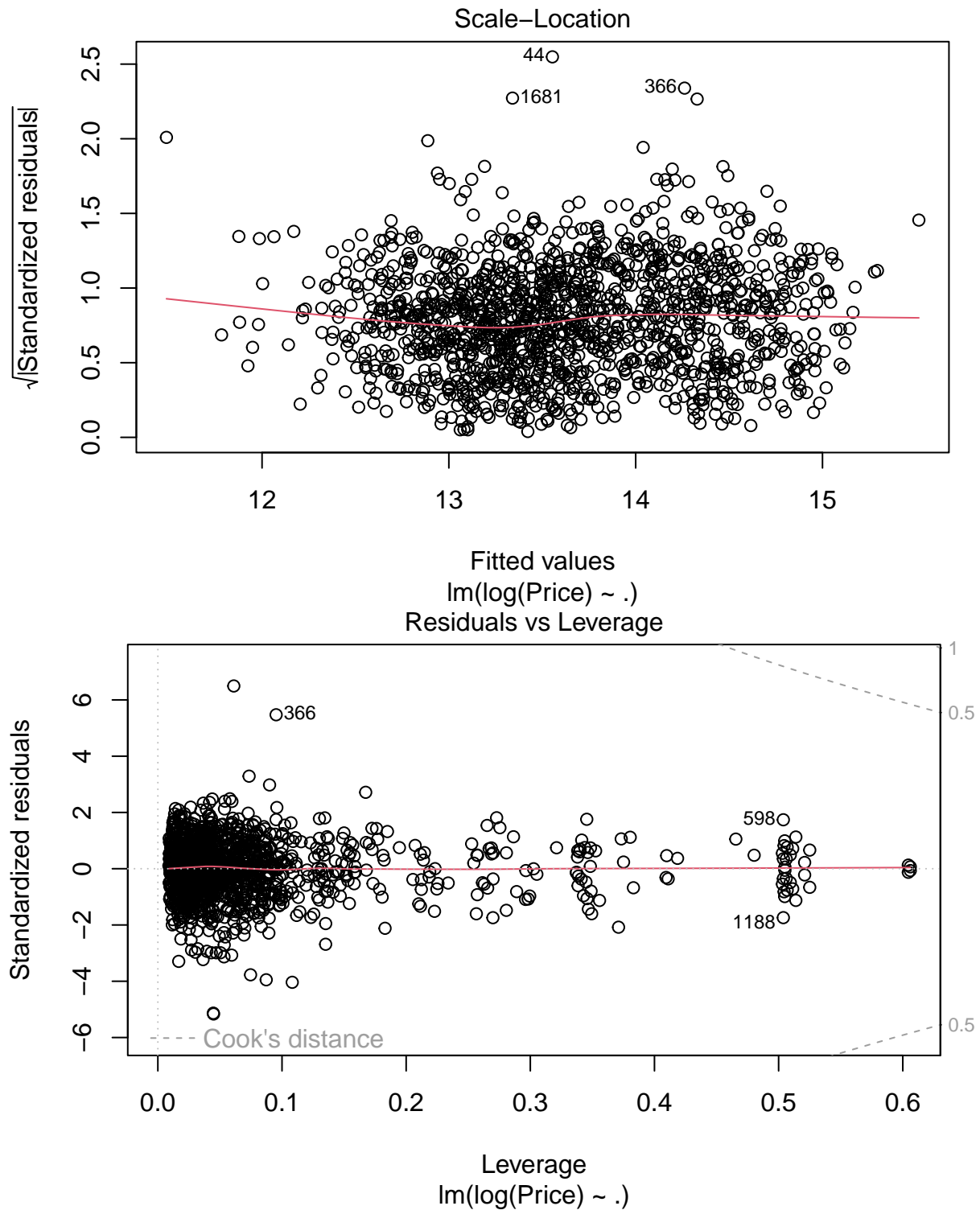
We get that GVIF^(1/(2*Df)) $< 2.236068 \approx \sqrt{5}$ for all regressors, and no interactions are indicated, so there is likely no multicollinearity.

```r
plot(model)
```

Residuals vs Fitted

Residuals

Fitted values
lm(log(Price) ~ .)

Q−Q Residuals

Standardized residuals

Theoretical Quantiles
lm(log(Price) ~ .)

4

Scale–Location

√|Standardized residuals|

44

1681

366

Fitted values
lm(log(Price) ~ .)

Residuals vs Leverage

Standardized residuals

366

598

1188

Cook's distance

Leverage
lm(log(Price) ~ .)

Backwards Stepwise Search

We do a backwards search since our model already conforms to the linear assumptions and is performing well. We simply wish to reduce the model size now.

```
reduced_model <- step(model, direction = "backward", data = x, trace = 0, k = log(nrow(x)))
old <- names(model$coefficients)
new <- names(reduced_model$coefficients)

old[!(old %in% new)]
```

```
##  [1] "LocationAhmedabad"    "LocationAllahabad"    "LocationAmbala Cantt"
##  [4] "LocationAmritsar"     "LocationAurangabad"   "LocationBangalore"
##  [7] "LocationBhopal"       "LocationBhubaneswar"  "LocationBulandshahar"
## [10] "LocationChandigarh"   "LocationChennai"      "LocationCoimbatore"
## [13] "LocationDehradun"     "LocationDelhi"        "LocationDharwad"
## [16] "LocationErnakulam"    "LocationFaridabad"    "LocationGhaziabad"
## [19] "LocationGoa"          "LocationGurgaon"      "LocationGuwahati"
## [22] "LocationHaldwani"     "LocationHyderabad"    "LocationIndore"
## [25] "LocationJaipur"       "LocationJalandhar"    "LocationJamshedpur"
## [28] "LocationKanpur"       "LocationKarnal"       "LocationKharar"
## [31] "LocationKheda"        "LocationKolkata"      "LocationKollam"
## [34] "LocationKota"         "LocationLucknow"      "LocationLudhiana"
## [37] "LocationMangalore"    "LocationMeerut"       "LocationMirzapur"
## [40] "LocationMohali"       "LocationMumbai"       "LocationMuzaffurpur"
## [43] "LocationMysore"       "LocationNagpur"       "LocationNashik"
## [46] "LocationNavi Mumbai"  "LocationNoida"        "LocationPanchkula"
## [49] "LocationPatna"        "LocationPune"         "LocationRaipur"
## [52] "LocationRanchi"       "LocationRanga Reddy"  "LocationRoorkee"
## [55] "LocationSalem"        "LocationSurat"        "LocationThane"
## [58] "LocationUdupi"        "LocationVadodara"     "LocationVaranasi"
## [61] "LocationWarangal"     "LocationYamunanagar"  "LocationZirakpur"
## [64] "OwnerSecond"          "OwnerThird"           "OwnerUnRegistered Car"
## [67] "Seller.TypeCorporate" "Seller.TypeIndividual" "Seating.Capacity"
```

Location, seller type, owner type, and seating capacity were removed by BIC.

Inspect Model Coefficients

```
summary(reduced_model)
```

```
##
## Call:
## lm(formula = log(Price) ~ Make + Year + Kilometer + Fuel.Type +
##     Transmission + Max.Torque.RPM + Drivetrain + Length + Width,
##     data = x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04626 -0.11894  0.00895  0.12150  1.12776
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.108e+02  4.959e+00 -42.513  < 2e-16 ***
## MakeBMW         -8.289e-03  4.487e-02  -0.185  0.85345
## MakeChevrolet   -7.614e-01  1.036e-01  -7.353 3.28e-13 ***
## MakeDatsun      -9.462e-01  7.807e-02 -12.120  < 2e-16 ***
## MakeFord        -5.729e-01  4.202e-02 -13.635  < 2e-16 ***
## MakeHonda       -5.122e-01  3.344e-02 -15.318  < 2e-16 ***
```
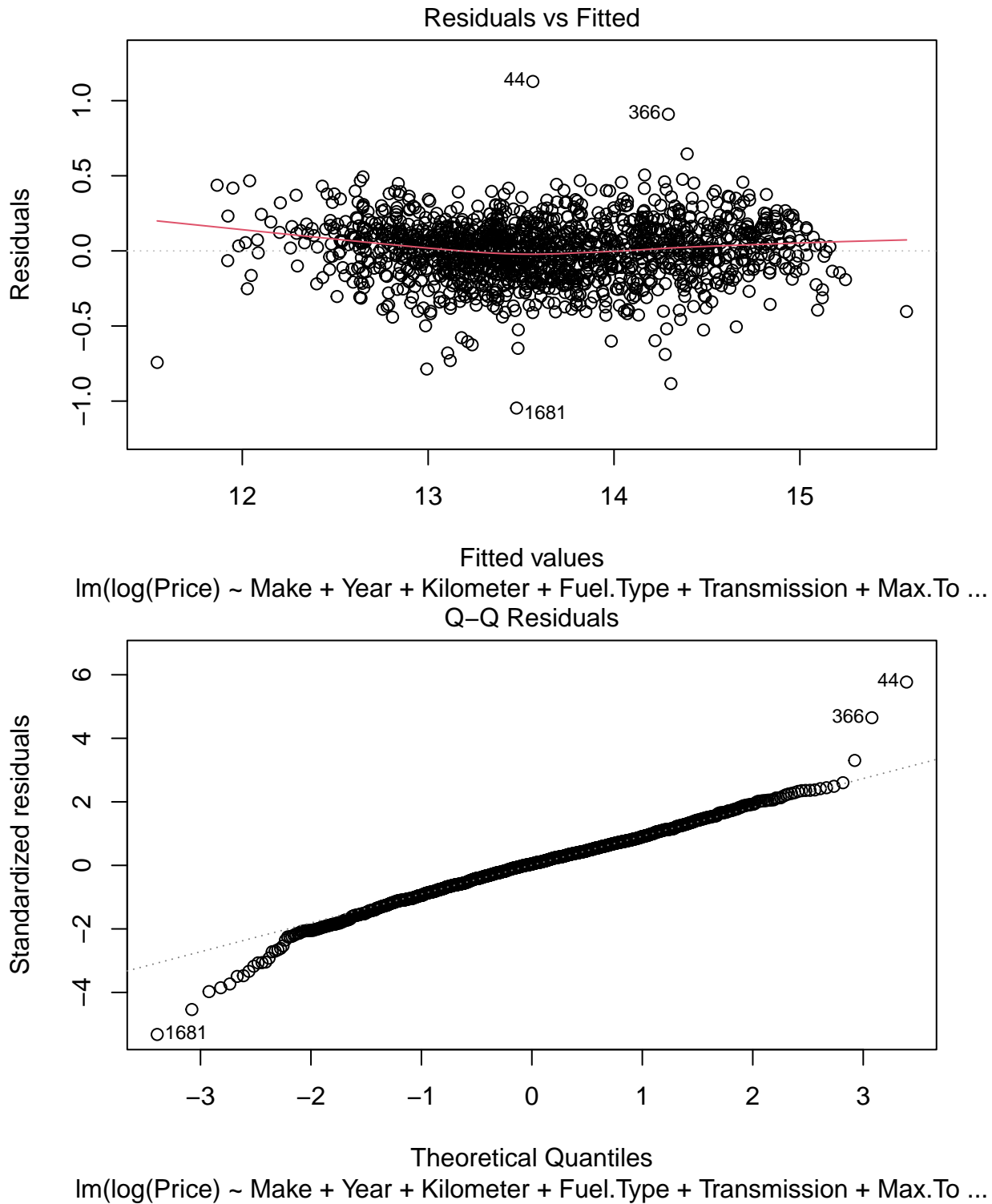
```
## MakeHyundai       -4.787e-01  2.967e-02 -16.136  < 2e-16 ***
## MakeIsuzu         -3.579e-01  1.445e-01  -2.477  0.01336 *
## MakeJaguar         2.577e-02  8.764e-02   0.294  0.76879
## MakeJeep          -3.124e-01  5.672e-02  -5.508 4.31e-08 ***
## MakeKia           -4.186e-01  4.935e-02  -8.482  < 2e-16 ***
## MakeLand Rover     1.135e-01  1.180e-01   0.962  0.33613
## MakeMahindra      -6.140e-01  3.531e-02 -17.389  < 2e-16 ***
## MakeMaruti Suzuki -5.372e-01  3.052e-02 -17.602  < 2e-16 ***
## MakeMercedes-Benz  9.442e-02  3.692e-02   2.558  0.01065 *
## MakeMG            -5.046e-01  6.084e-02  -8.295 2.52e-16 ***
## MakeMINI           6.884e-01  8.562e-02   8.041 1.88e-15 ***
## MakeMitsubishi    -2.069e-01  1.185e-01  -1.746  0.08110 .
## MakeNissan        -7.701e-01  5.774e-02 -13.338  < 2e-16 ***
## MakeRenault       -7.636e-01  4.273e-02 -17.869  < 2e-16 ***
## MakeSkoda         -4.656e-01  4.166e-02 -11.176  < 2e-16 ***
## MakeSsangyong     -8.916e-01  1.188e-01  -7.502 1.11e-13 ***
## MakeTata          -7.320e-01  4.051e-02 -18.068  < 2e-16 ***
## MakeToyota        -2.752e-01  3.299e-02  -8.341  < 2e-16 ***
## MakeVolkswagen    -5.016e-01  4.107e-02 -12.213  < 2e-16 ***
## MakeVolvo          3.521e-02  6.401e-02   0.550  0.58236
## Year               1.094e-01  2.470e-03  44.282  < 2e-16 ***
## Kilometer         -1.624e-06  2.597e-07  -6.253 5.35e-10 ***
## Fuel.TypeDiesel   -1.320e-01  4.157e-02  -3.175  0.00153 **
## Fuel.TypePetrol   -1.518e-01  3.794e-02  -4.000 6.67e-05 ***
## TransmissionManual -1.660e-01 1.395e-02 -11.906  < 2e-16 ***
## Max.Torque.RPM    -4.436e-05  8.320e-06  -5.331 1.14e-07 ***
## DrivetrainFWD     -2.586e-01  2.475e-02 -10.449  < 2e-16 ***
## DrivetrainRWD     -1.703e-01  2.968e-02  -5.738 1.17e-08 ***
## Length             5.894e-04  2.775e-05  21.239  < 2e-16 ***
## Width              1.444e-03  1.208e-04  11.951  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1986 on 1405 degrees of freedom
## Multiple R-squared:  0.9207, Adjusted R-squared:  0.9187
## F-statistic: 465.9 on 35 and 1405 DF,  p-value: < 2.2e-16
```

```r
anova(reduced_model)
```
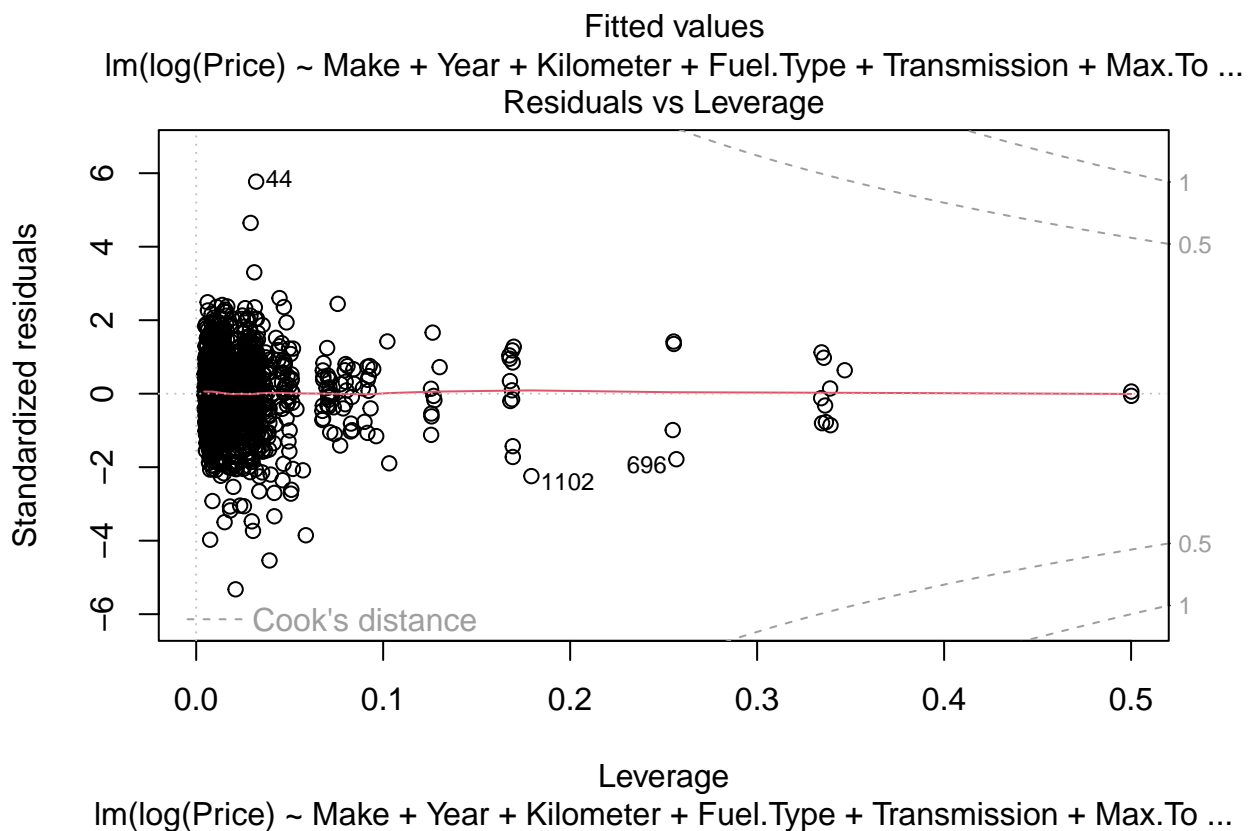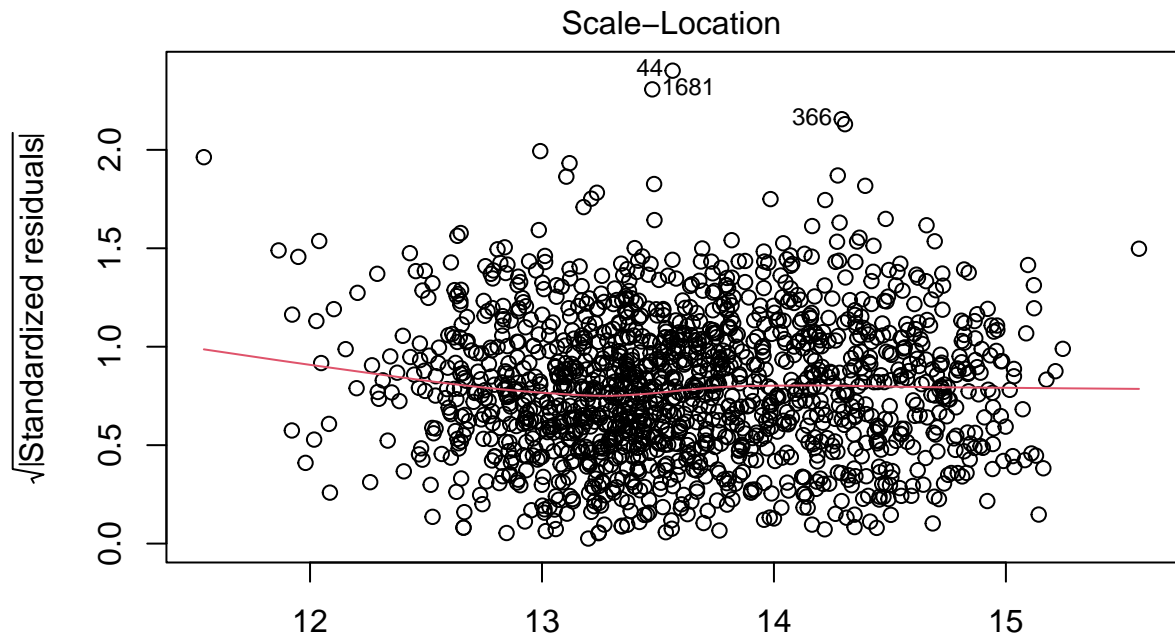
```
## Analysis of Variance Table
##
## Response: log(Price)
##                  Df Sum Sq Mean Sq  F value    Pr(>F)
## Make             25 354.44  14.178  359.289 < 2.2e-16 ***
## Year              1 178.52 178.525 4524.169 < 2.2e-16 ***
## Kilometer         1   0.98   0.982   24.874 6.884e-07 ***
## Fuel.Type         2  19.23   9.615  243.668 < 2.2e-16 ***
## Transmission      1  20.31  20.313  514.760 < 2.2e-16 ***
## Max.Torque.RPM    1   1.14   1.139   28.868 9.060e-08 ***
## Drivetrain        2   6.53   3.267   82.792 < 2.2e-16 ***
## Length            1  56.67  56.670 1436.142 < 2.2e-16 ***
## Width             1   5.64   5.636  142.825 < 2.2e-16 ***
## Residuals      1405  55.44   0.039
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Final Model Verification

```r
plot(reduced_model)
```

Residuals vs Fitted

Fitted values
lm(log(Price) ~ Make + Year + Kilometer + Fuel.Type + Transmission + Max.To ...

Q−Q Residuals

Theoretical Quantiles
lm(log(Price) ~ Make + Year + Kilometer + Fuel.Type + Transmission + Max.To ...

Scale–Location

lm(log(Price) ~ Make + Year + Kilometer + Fuel.Type + Transmission + Max.To ...

Residuals vs Leverage

lm(log(Price) ~ Make + Year + Kilometer + Fuel.Type + Transmission + Max.To ...

Normalized Coefficients

```
# removed by BIC. remember multicollinear rows are already removed
normalized <- remove_cols(x, c("Location", "Seating.Capacity", "Seller.Type", "Owner"))
```

```r
for (col in colnames(normalized)) {
  if (col != "Price" && !(col %in% categorical)) {
    normalized[, col] <- (normalized[, col] - mean(normalized[, col])) / sd(normalized[, col])
  }
}

normalized_model <- lm(log(Price) ~ ., data = normalized)
summary(normalized_model)
```

```
##
## Call:
## lm(formula = log(Price) ~ ., data = normalized)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.04626 -0.11894  0.00895  0.12150  1.12776
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         14.543827   0.045612 318.858  < 2e-16 ***
## MakeBMW             -0.008289   0.044869  -0.185  0.85345
## MakeChevrolet       -0.761421   0.103550  -7.353 3.28e-13 ***
## MakeDatsun          -0.946201   0.078066 -12.120  < 2e-16 ***
## MakeFord            -0.572886   0.042016 -13.635  < 2e-16 ***
## MakeHonda           -0.512179   0.033437 -15.318  < 2e-16 ***
## MakeHyundai         -0.478712   0.029667 -16.136  < 2e-16 ***
## MakeIsuzu           -0.357869   0.144462  -2.477  0.01336 *
## MakeJaguar           0.025768   0.087639   0.294  0.76879
## MakeJeep            -0.312446   0.056724  -5.508 4.31e-08 ***
## MakeKia             -0.418627   0.049353  -8.482  < 2e-16 ***
## MakeLand Rover       0.113527   0.117990   0.962  0.33613
## MakeMahindra        -0.613960   0.035308 -17.389  < 2e-16 ***
## MakeMaruti Suzuki   -0.537170   0.030517 -17.602  < 2e-16 ***
## MakeMercedes-Benz    0.094425   0.036921   2.558  0.01065 *
## MakeMG              -0.504646   0.060838  -8.295 2.52e-16 ***
## MakeMINI             0.688415   0.085617   8.041 1.88e-15 ***
## MakeMitsubishi      -0.206925   0.118540  -1.746  0.08110 .
## MakeNissan          -0.770128   0.057739 -13.338  < 2e-16 ***
## MakeRenault         -0.763589   0.042732 -17.869  < 2e-16 ***
## MakeSkoda           -0.465623   0.041663 -11.176  < 2e-16 ***
## MakeSsangyong       -0.891608   0.118844  -7.502 1.11e-13 ***
## MakeTata            -0.732018   0.040515 -18.068  < 2e-16 ***
## MakeToyota          -0.275157   0.032989  -8.341  < 2e-16 ***
## MakeVolkswagen      -0.501595   0.041071 -12.213  < 2e-16 ***
## MakeVolvo            0.035208   0.064006   0.550  0.58236
## Year                 0.311815   0.007042  44.282  < 2e-16 ***
## Kilometer           -0.044526   0.007121  -6.253 5.35e-10 ***
## Fuel.TypeDiesel     -0.132002   0.041572  -3.175  0.00153 **
## Fuel.TypePetrol     -0.151767   0.037944  -4.000 6.67e-05 ***
## TransmissionManual  -0.166038   0.013946 -11.906  < 2e-16 ***
## Max.Torque.RPM      -0.055706   0.010449  -5.331 1.14e-07 ***
## DrivetrainFWD       -0.258640   0.024754 -10.449  < 2e-16 ***
## DrivetrainRWD       -0.170290   0.029677  -5.738 1.17e-08 ***
```

```
## Length                 0.218156   0.010272  21.239  < 2e-16 ***
## Width                  0.130065   0.010883  11.951  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1986 on 1405 degrees of freedom
## Multiple R-squared:  0.9207, Adjusted R-squared:  0.9187
## F-statistic: 465.9 on 35 and 1405 DF,  p-value: < 2.2e-16
```

```r
sort(abs(normalized_model$coefficients), decreasing = T)
```

```
##       (Intercept)        MakeDatsun      MakeSsangyong        MakeNissan
##       14.543826960       0.946201111       0.891608072       0.770128355
##       MakeRenault      MakeChevrolet          MakeTata          MakeMINI
##       0.763589078       0.761420626       0.732018001       0.688415224
##       MakeMahindra          MakeFord  MakeMaruti Suzuki        MakeHonda
##       0.613960112       0.572885621       0.537170191       0.512179318
##             MakeMG    MakeVolkswagen        MakeHyundai         MakeSkoda
##       0.504645860       0.501595422       0.478711619       0.465623344
##            MakeKia         MakeIsuzu          MakeJeep              Year
##       0.418626911       0.357868994       0.312445876       0.311815408
##         MakeToyota      DrivetrainFWD            Length     MakeMitsubishi
##       0.275156539       0.258640046       0.218155692       0.206925140
##      DrivetrainRWD TransmissionManual    Fuel.TypePetrol   Fuel.TypeDiesel
##       0.170289761       0.166038313       0.151767051       0.132001751
##              Width     MakeLand Rover  MakeMercedes-Benz     Max.Torque.RPM
##       0.130064979       0.113527147       0.094424967       0.055705512
##          Kilometer         MakeVolvo        MakeJaguar           MakeBMW
##       0.044525748       0.035207673       0.025767531       0.008289357
```

Confidence Intervals

```r
confint(normalized_model, level = 0.95)
```

```
##                          2.5 %        97.5 %
## (Intercept)        14.45435143  14.63330249
## MakeBMW            -0.09630597   0.07972726
## MakeChevrolet      -0.96455003  -0.55829122
## MakeDatsun         -1.09934049  -0.79306173
## MakeFord           -0.65530709  -0.49046415
## MakeHonda          -0.57777174  -0.44658689
## MakeHyundai        -0.53690767  -0.42051557
## MakeIsuzu          -0.64125440  -0.07448359
## MakeJaguar         -0.14615075   0.19768581
## MakeJeep           -0.42371850  -0.20117325
## MakeKia            -0.51543960  -0.32181422
## MakeLand Rover     -0.11792897   0.34498326
## MakeMahindra       -0.68322180  -0.54469842
## MakeMaruti Suzuki  -0.59703421  -0.47730618
## MakeMercedes-Benz   0.02199928   0.16685065
## MakeMG             -0.62398874  -0.38530298
## MakeMINI            0.52046454   0.85636591
## MakeMitsubishi     -0.43945862   0.02560835
## MakeNissan         -0.88339267  -0.65686404
## MakeRenault        -0.84741387  -0.67976428
```

```
## MakeSkoda          -0.54735131 -0.38389537
## MakeSsangyong      -1.12473817 -0.65847798
## MakeTata           -0.81149433 -0.65254168
## MakeToyota         -0.33986892 -0.21044416
## MakeVolkswagen     -0.58216290 -0.42102794
## MakeVolvo          -0.09034961  0.16076496
## Year                0.29800228  0.32562854
## Kilometer          -0.05849508 -0.03055642
## Fuel.TypeDiesel    -0.21355073 -0.05045278
## Fuel.TypePetrol    -0.22620082 -0.07733328
## TransmissionManual -0.19339565 -0.13868098
## Max.Torque.RPM     -0.07620280 -0.03520823
## DrivetrainFWD      -0.30719809 -0.21008200
## DrivetrainRWD      -0.22850512 -0.11207440
## Length              0.19800639  0.23830499
## Width               0.10871583  0.15141413
```