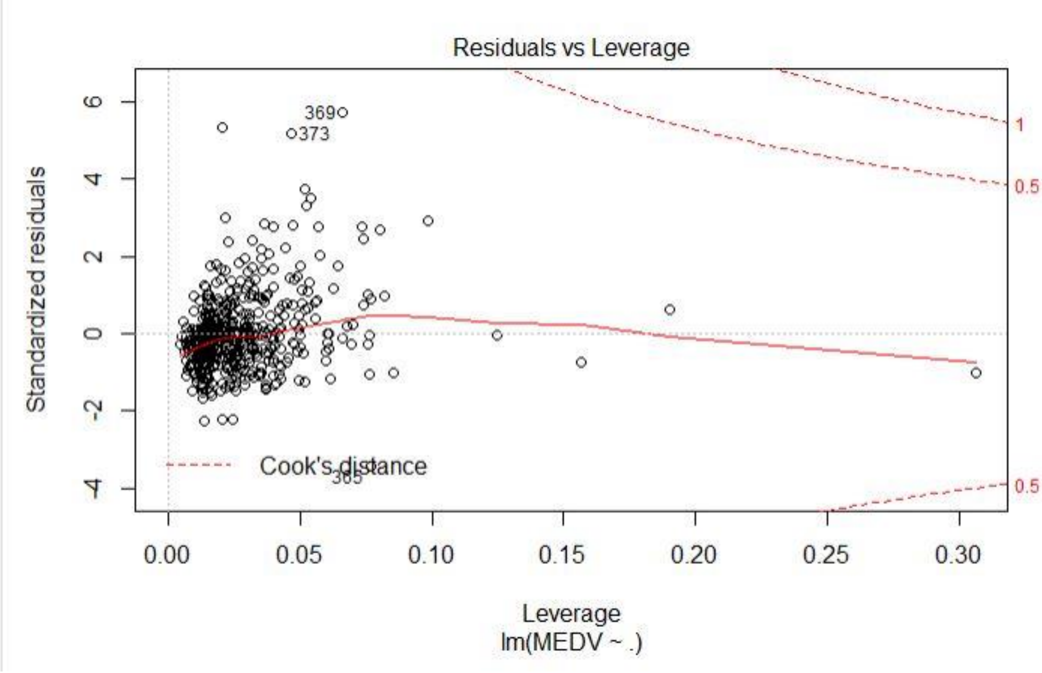
1. Code for regression and resulting model (note houseData.txt contains the data with manually added header):

```
file = "houseData.txt"
houseData = read.table(file, sep = "", header = T)
modelWithOutlier = lm(MEDV ~ ., data = houseData)
```

2. Screenshot of diagnostic plot and explanation for outliers removal:



Indexes of outliers to be removed due to absolute value of standardized residual > 4 are:

```
369 372 373
```

Indexes of outliers to be removed due to cook's distance > 4 * (mean of all cook's distance):

(Note that some convention use 4/(rows of dataset) as threshold, in this case it will remove too many datapoints and thus is not used here)

```
65 142 162 163 164 167 187 215 226 229 254 365 366 368
369 370 371 372 373 375 381 413 415
```

Indexes of outliers to be removed due to leverage > 0.1 are:

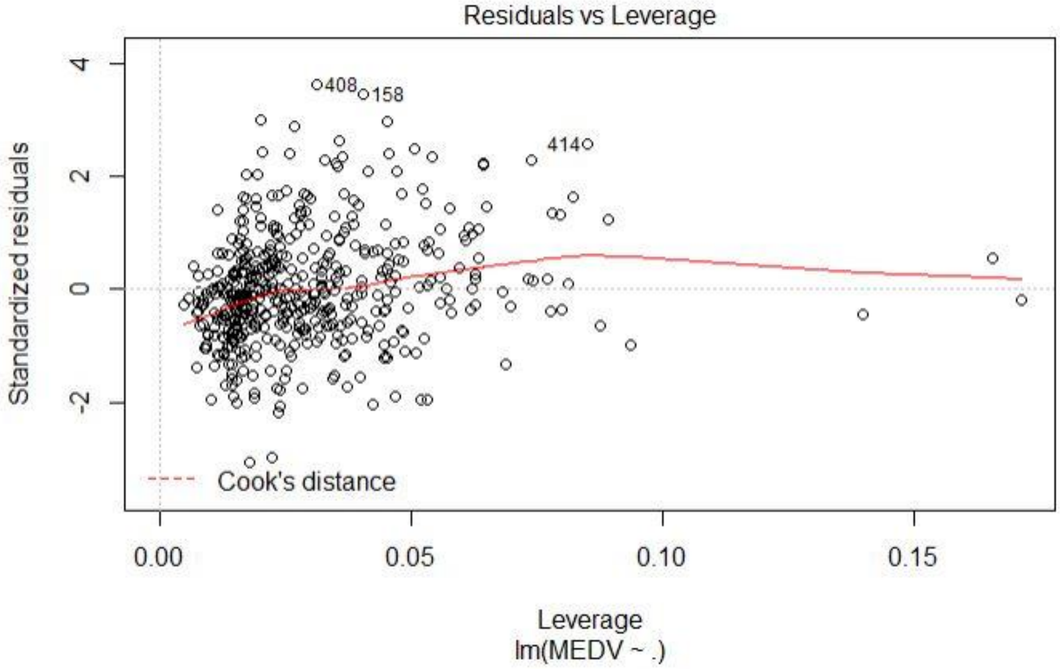
```
381 406 411 419
```

Hence, all indexes of outliers to be removed are (Union of above three):

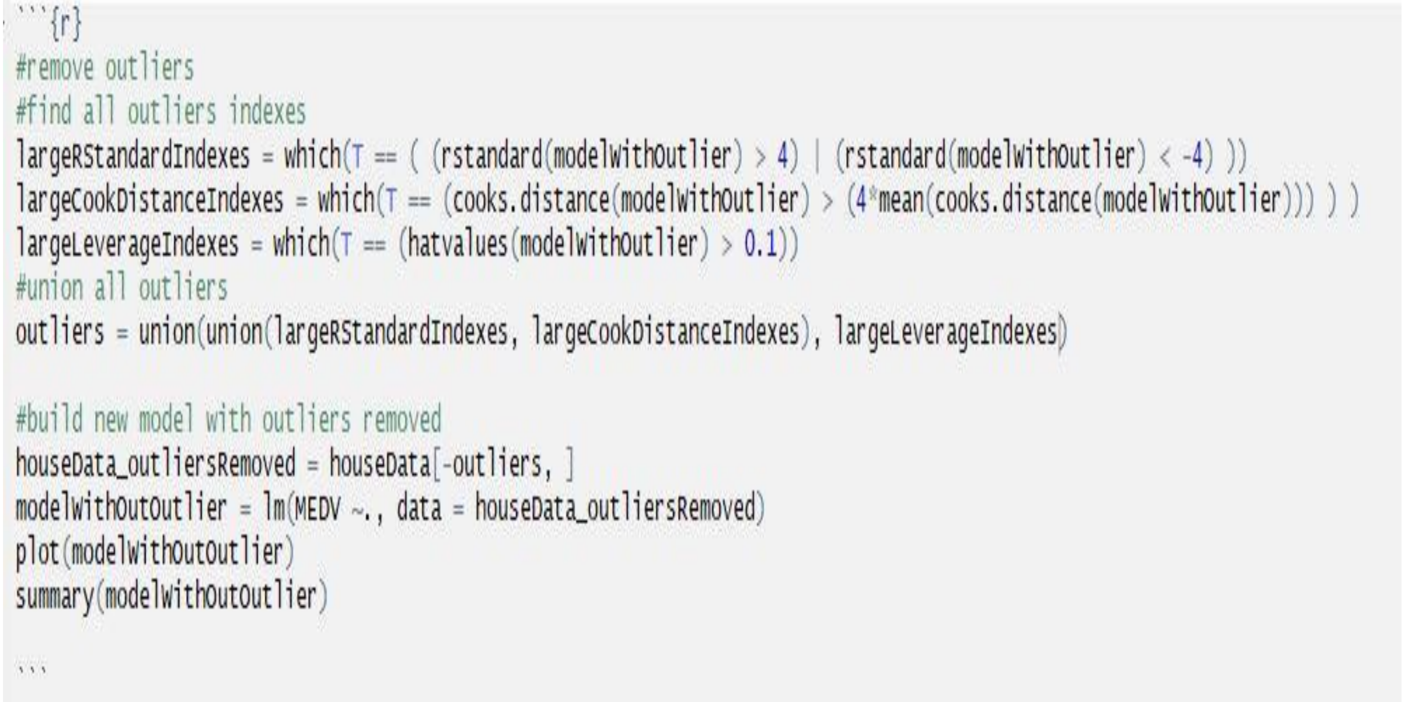
```
369 372 373  65 142 162 163 164 167 187 215 226 229 254
365 366 368 370 371 375 381 413 415 406 411 419
```

Total outliers removed: 26

3. Screenshot of new diagnostic plot:

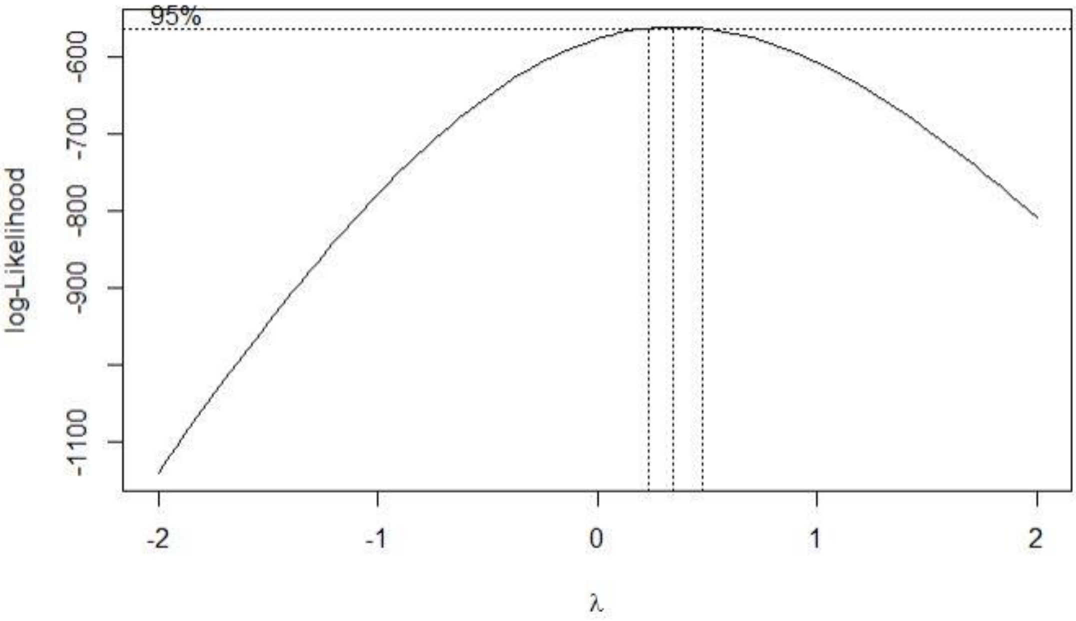


4. Screenshot of code for subproblem 2:



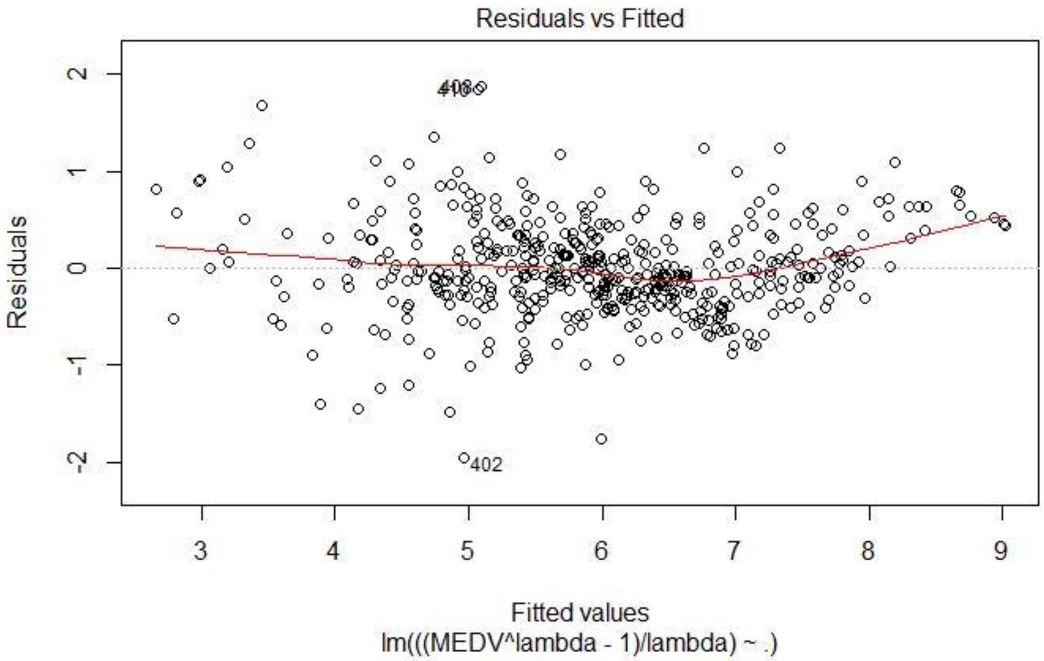
~~~
```{r}
#remove outliers
#find all outliers indexes
largeRStandardIndexes = which(T == ( (rstandard(modelWithOutlier) > 4) | (rstandard(modelWithOutlier) < -4) ))
largeCookDistanceIndexes = which(T == (cooks.distance(modelWithOutlier) > (4*mean(cooks.distance(modelWithOutlier))) ) )
largeLeverageIndexes = which(T == (hatvalues(modelWithOutlier) > 0.1))
#union all outliers
outliers = union(union(largeRStandardIndexes, largeCookDistanceIndexes), largeLeverageIndexes)

#build new model with outliers removed
houseData_outliersRemoved = houseData[-outliers, ]
modelWithOutOutlier = lm(MEDV ~., data = houseData_outliersRemoved)
plot(modelWithOutOutlier)
summary(modelWithOutOutlier)

```
~~~

5. Screenshot of Box-Cox transformation plot and best value for lambda

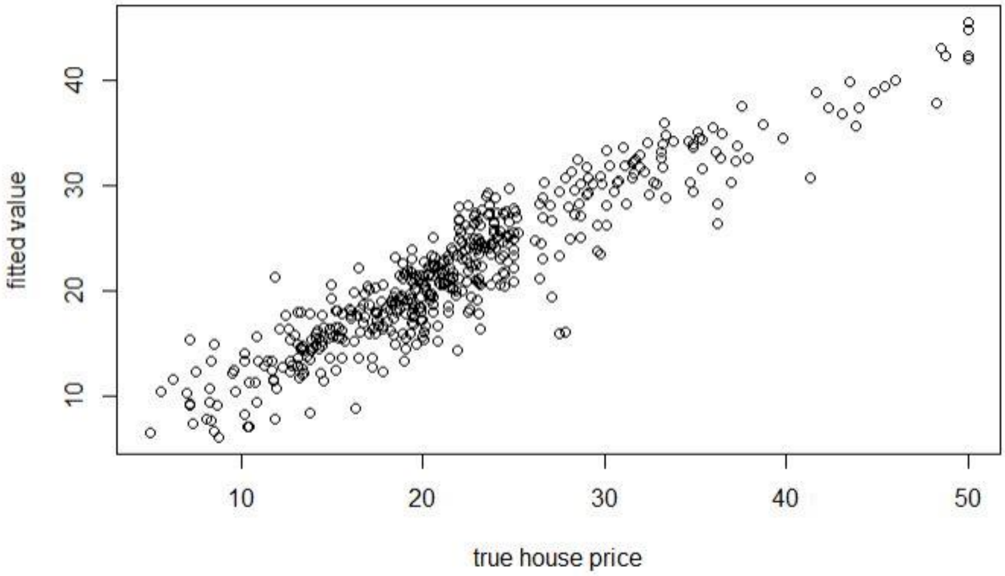


The best value for lambda is **0.4**, it is calculated by find the lambda maximize log-likelihood, R code to achieved this is attached at the end.

6. Result of the standardized residuals of the regression after Box-Cox transformation:



a plot of fitted house price against true house price:



Code for part 3:

```
  #apply boxcox transformation
       library(MASS)
  #plot the box-cox transformation
       boxcox(modelWithOutOutlier)
  #find the best lambda value:
       lambda = with(boxcox(modelWithOutOutlier, plotit = F), x[which.max(y)])
```

Code for part 4:

```
   #apply transformation to dependant variable and fit the model again
     modelWithoutOutlierTransformed = lm( ((MEDV^lambda - 1) / lambda) ~ ., data = houseData_outliersRemoved)
   plot(modelWithoutOutlierTransformed)

   #plot fitted house price against true house price
   fittedValue = (predict(modelWithoutOutlierTransformed, newdata = houseData_outliersRemoved) * lambda + 1) ^ (1/lambda)
 plot(fittedValue ~ houseData_outliersRemoved$MEDV, xlab = "true house price", ylab = "fitted value")
```