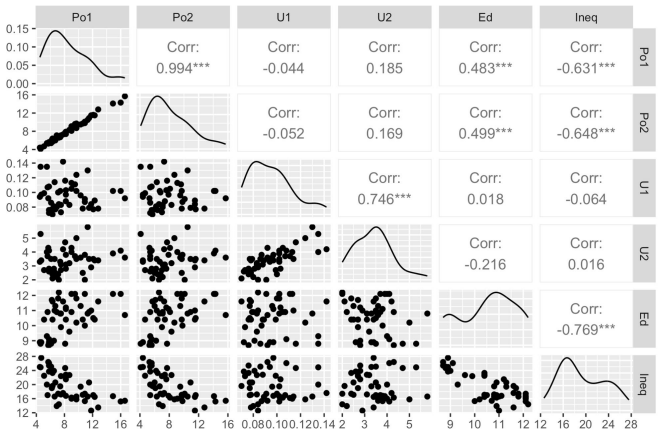
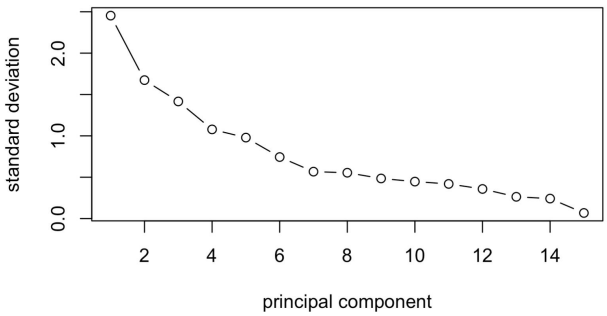
Question 9.1

First, let's examine the uscrime data. From the following plot, it's clear that Po1 and Po2, U1 and U2, and Ed and Ineq have high correlations separately. We can use Principal Component Analysis (PCA) to remove these correlations within the data.



I applied PCA to the predictors in the uscrime data. The standard deviation that the 15 principal components captures is shown below. As we can see, the standard deviations after the 8th principal component are all lower than 0.5. In fact, 95% of the variance is captured by the first 8 principal components. Therefore, I chose the first 8 principal components to set up a regression model.



After performing regression with these principal components, we obtained the results shown below. Compared to my solution to Question 8.2, where the R-squared was 0.8 and the adjusted R-squared was 0.7, the outcome from Question 9.1 has a lower R-squared (0.6899) and adjusted R-squared (0.6246). This is because we experienced

regression overfitting and multicollinearity in Question 8.2 when we included all factors in the regression.

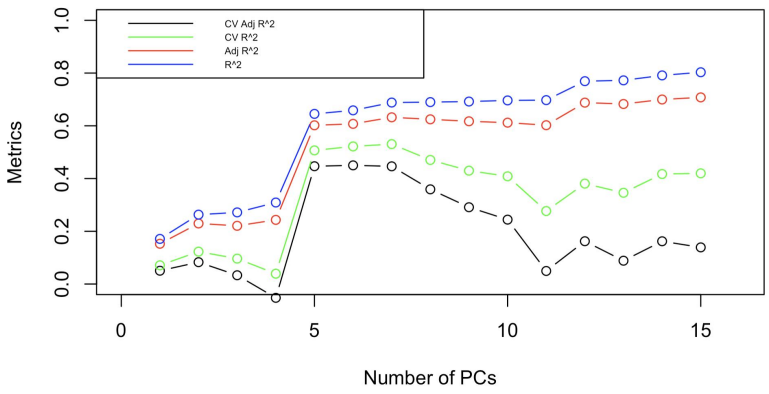
Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	**905.09**	**34.57**	**26.184**	**< 2e-16**	*******
PC1	**65.22**	**14.24**	**4.579**	**4.90e-05**	*******
PC2	**-70.08**	**20.87**	**-3.357**	**0.0018**	******
PC3	**25.19**	**24.68**	**1.021**	**0.3137**	
PC4	**69.45**	**32.41**	**2.143**	**0.0386**	*****
PC5	**-229.04**	**35.69**	**-6.417**	**1.53e-07**	*******
PC6	**-60.21**	**46.98**	**-1.282**	**0.2077**	
PC7	**117.26**	**61.59**	**1.904**	**0.0645**	**.**
PC8	**28.72**	**63.02**	**0.456**	**0.6512**	

Signif. codes: 0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1**

Residual standard error: 237 on 38 degrees of freedom
Multiple R-squared: 0.6899, Adjusted R-squared: 0.6246
F-statistic: 10.57 on 8 and 38 DF, p-value: 1.182e-07

In general, the more principal components included in the regression, the higher the R-squared value, and vice versa. When all principal components are used in the regression, the R-squared result is the same as that of Question 8.2. We could also use cross-validation to check the quality of the model. The following plot shows that the model achieves the highest R-squared value with around 7 principal components. With 7 principal components in the regression, the cross-validation R-squared value is 0.53 and the adjusted R-squared value is 0.45, which are higher than the cross-validation R-squared value of 0.41 in Question 8.2.

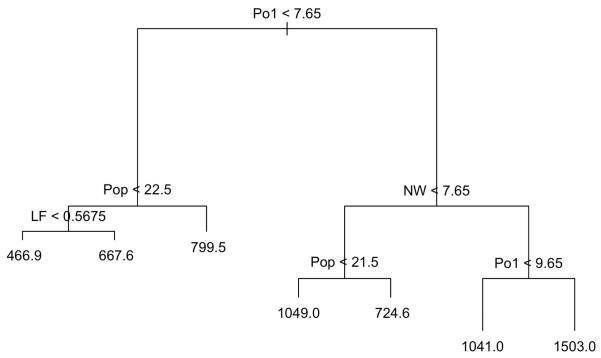


Finally, we must use the coefficients of the 8 principal components to find the coefficients of the 15 original variables. The related scaled and unscaled coefficients for each original variable are shown below. Using the coefficients of the unscaled original predictors and the data point from Question 8.2, we can make a prediction of 1190.455 for the independent variable Crime. This prediction makes more sense than 155.4349 in Question 8.2, as the mean of Crime in the original dataset is 905.

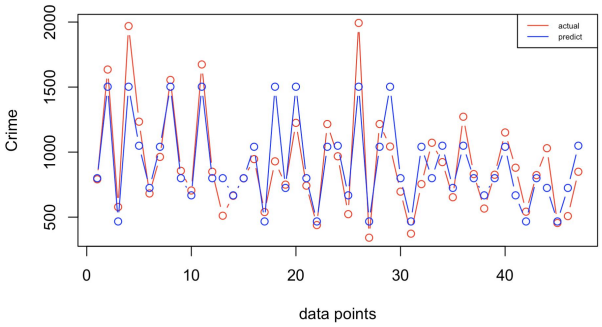
|regression coefficient | (Intercept) -5369.778
variable	scale	unscale
M	53.52058	42.5860470
So	73.46865	153.3871891
Ed	-11.81690	-10.5630639
Po1	133.83089	45.0321369
Po2	130.71847	46.7497510
LF	34.53922	854.6813283
M.F	129.26225	43.8662381
Pop	19.57959	0.5142890
NW	64.32517	6.2555583
U1	-24.33136	-1349.5844084
U2	27.61844	32.7021567
Wealth	30.30222	0.0314042
Ineq	51.73812	12.9682287
Prob	-120.03240	-5279.1724361
Time	-10.09618	-1.4246262

Question 10.1

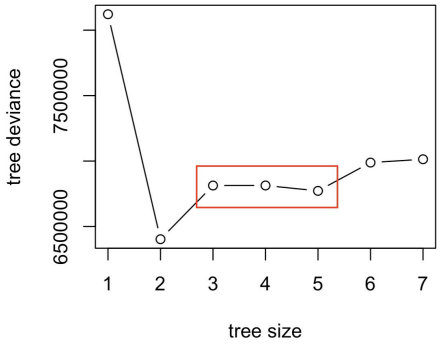
(a) First, I trained a tree model using the uscrime data by splitting predictors based on decreasing deviance and using the average of responses as the leaf value. The resulting tree split the data into four factors. In this tree, each leaf has at least 5 data points, which is more than 5% of the whole dataset. However, I noticed that leaf 13, where Pop >= 21.5, contains 5 data points, 4 of which are potential outliers, as identified in HW2. This raises the possibility that the tree may be overfitting.

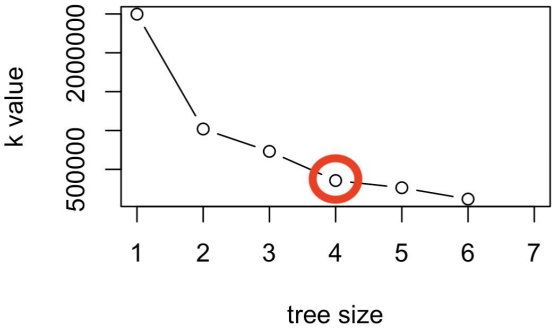


The plot below compares the predicted crime values from our tree model with the actual values in the original data. The predicted values fit well and even smooth out the original data because we used the same dataset and an overfitted model with average responses as the outcome to predict. The R-squared value of our tree model is 0.724, which supports our assumption of overfitting.

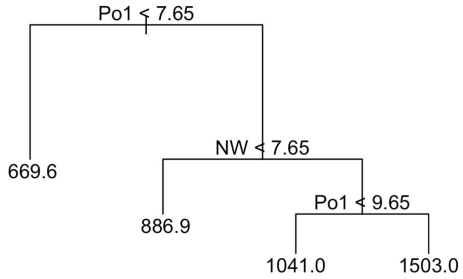


To address overfitting, we need to prune the tree. Using cross-validation, we determine the optimal tree size. The first plot below shows that having between 3 to 5 leaves results in relatively low deviance. A smaller tree may lose information, while a larger tree may overfit, so we aim for a balance. The second plot below indicates an elbow point at 4 leaves, suggesting this as the best tree size for pruning.





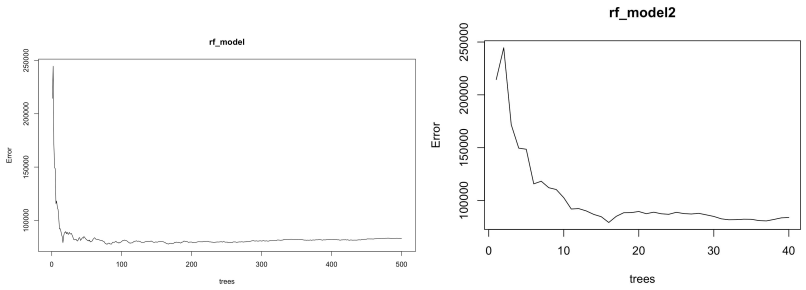
After pruning, we obtain a new, simplified tree model with an R-squared value of 0.62. This pruned tree no longer includes the Pop variable split. Based on this tree, we can build a regression model in the leaf on the far left of the tree where Po1 is less than 7.65.



The linear regression model obtained is shown below. It has a high R-squared value of 0.8794, but a relatively low adjusted R-squared value of 0.6209, likely due to the 23 data points versus 15 factors. The F-statistic indicates that the regression model is not very significant. Adding a regression model to every leaf could lead to overfitting, so it is not the best approach.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -48.5477  2044.9766  -0.024   0.9817
M             45.8622    58.6256   0.782   0.4597
So           380.4815   223.1072   1.705   0.1319
Ed           187.9074    89.5799   2.098   0.0741 .
Po1           -3.5138   157.7513  -0.022   0.9829
Po2           44.6382   148.5528   0.300   0.7725
LF          1059.3652  1187.9722   0.892   0.4021
M.F          -22.5521    21.4677  -1.051   0.3284
Pop           10.6413     5.0929   2.089   0.0750 .
NW             0.1010     7.9019   0.013   0.9902
U1          4878.2802  4874.8165   1.001   0.3503
U2            -5.5126   133.5094  -0.041   0.9682
Wealth        -0.1022     0.1752  -0.583   0.5779
Ineq           4.7779    35.5290   0.134   0.8968
Prob       -7317.4407  3280.7511  -2.230   0.0609 .
Time          -20.0603     7.7287  -2.596   0.0357 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 115.9 on 7 degrees of freedom
Multiple R-squared:  0.8794,          Adjusted R-squared:  0.6209
G-statistic: 3.403 on 15 and 7 DF,  p-value: 0.0541
```

(b)First, I grew a random forest with 500 trees, each split by 5 factors. The random forest model indicates that Po1 is a crucial factor for splitting the tree, consistent with the single tree model. In this random forest model, the R-squared value is 0.432, which is lower than the R-squared value of 0.62 for the single tree model, indicating a lower fit. However, the error decreases rapidly at the beginning and stabilizes around 40 trees. Therefore, we can reduce the number of trees in the forest to 40. This adjustment yields an R-squared value of 0.427. By reducing the number of trees, we save considerable computing time while maintaining a similar level of model performance.



Question 10.2

Logistic regression model may help to make a decision on whether to buy a car or not. The predictors might be family income, the price of car, distance to workplace, availability of public transportation, and cost of car maintenance. By analyzing the relationship between these predictors and the binary outcome (buy a car or not), the logistic regression model can estimate the probability of deciding to buy a car based on specific circumstances. Furthermore, since deciding to buy a car when not buying is actually the better choice can be costly and potentially lead to a financial crisis, I would set a higher cutoff value for the decision. If buying a car is represented by 1, I would set my cutoff value higher than 0.5, for example, 0.7. This way, I could avoid false positives.

Question 10.3

1. First, I separated the German credit dataset into training data (80%) and testing data (20%). The logistic regression model outcome with all factors involved is displayed below. Each categorical variable is automatically converted to several dummy variables. By using the testing data, we can predict the probability of the target binary variable. If we set the threshold at 0.5, we achieve a prediction accuracy of 76%. However, the model below is too complex and hard to explain. Therefore, I will select a few specific variables based on my knowledge to create a new and more explainable model.

```
Call:
glm(formula = V21 ~ ., family = "binomial", data = train_data)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.142e-01  1.180e+00  -0.690 0.490076
V1A12        1.598e-01  2.479e-01   0.645 0.519134
V1A13        1.084e+00  4.381e-01   2.475 0.013309 *
V1A14        1.681e+00  2.670e-01   6.295 3.07e-10 ***
V2          -3.915e-02  1.078e-02  -3.632 0.000281 ***
V3A31       -4.611e-01  6.074e-01  -0.759 0.447760
V3A32        3.667e-01  4.698e-01   0.781 0.435074
V3A33        1.032e+00  5.417e-01   1.905 0.056788 .
V3A34        1.100e+00  4.778e-01   2.302 0.021361 *
V4A41        1.576e+00  4.034e-01   3.908 9.32e-05 ***
V4A410       1.671e+00  8.923e-01   1.873 0.061089 .
V4A42        6.130e-01  2.933e-01   2.090 0.036625 *
V4A43        8.175e-01  2.878e-01   2.841 0.004498 **
V4A44       -5.643e-01  9.886e-01  -0.571 0.568154
V4A45        1.883e-01  6.240e-01   0.302 0.762785
V4A46       -1.607e-01  4.461e-01  -0.360 0.718587
V4A48        1.997e+00  1.229e+00   1.625 0.104097
V4A49        7.518e-01  3.922e-01   1.917 0.055242 .
V5          -1.058e-04  5.123e-05  -2.065 0.038969 *
V6A62        5.022e-01  3.223e-01   1.558 0.119222
V6A63        3.768e-01  4.425e-01   0.852 0.394423
V6A64        1.637e+00  6.001e-01   2.727 0.006390 **
V6A65        1.010e+00  3.082e-01   3.276 0.001053 **
V7A72        5.095e-04  4.715e-01   0.001 0.999138
V7A73        3.343e-02  4.557e-01   0.073 0.941514
V7A74        7.435e-01  4.962e-01   1.498 0.134014
V7A75        2.030e-01  4.544e-01   0.447 0.654986
V8          -3.163e-01  1.006e-01  -3.143 0.001672 **
```

```
V9A92        3.207e-01  4.354e-01   0.737 0.461352
V9A93        8.637e-01  4.301e-01   2.008 0.044602 *
V9A94        5.048e-01  5.272e-01   0.957 0.338355
V10A102     -7.067e-01  4.643e-01  -1.522 0.128041
V10A103      8.903e-01  4.588e-01   1.941 0.052314 .
V11          8.624e-03  9.721e-02   0.089 0.929305
V12A122     -3.449e-01  2.855e-01  -1.208 0.227070
V12A123     -6.343e-02  2.767e-01  -0.229 0.818711
V12A124     -8.471e-01  5.164e-01  -1.640 0.100906
V13          1.560e-02  1.061e-02   1.470 0.141622
V14A142      6.849e-02  4.591e-01   0.149 0.881419
V14A143      7.430e-01  2.750e-01   2.702 0.006898 **
V15A152      6.566e-01  2.694e-01   2.437 0.014796 *
V15A153      1.027e+00  5.632e-01   1.824 0.068153 .
V16         -2.697e-01  2.143e-01  -1.259 0.208049
V17A172     -1.564e-01  7.355e-01  -0.213 0.831577
V17A173      6.609e-04  7.043e-01   0.001 0.999251
V17A174     -4.988e-02  7.098e-01  -0.070 0.943979
V18         -2.339e-01  2.896e-01  -0.808 0.419226
V19A192      3.661e-01  2.271e-01   1.612 0.106895
V20A202      1.351e+00  6.989e-01   1.933 0.053182 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 982.41  on 799  degrees of freedom
Residual deviance: 700.97  on 751  degrees of freedom
AIC: 798.97

Number of Fisher Scoring iterations: 5
```

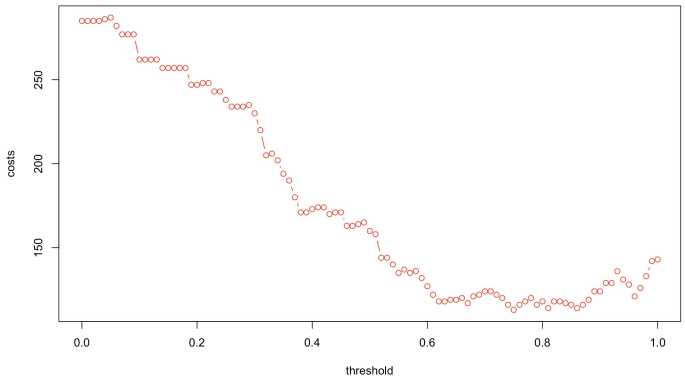
For the second model, I selected 6 factors, which include 4 categorical variables and 2 numerical variables. I transformed the categorical variables into dummy variables and reduced the number of categories within each variable. The 6 variables are the status of the existing checking account, credit history, credit amount, savings account/bonds, installment rate in percentage of disposable income, and property. The response variable is whether a customer is good or not. The output of this regression is shown below.

customerQuality = 1.826 + 3.668e-01checking - 1.140 creditHistory - 1.062e-04creditAmount + 3.155e-01savings - 2.487e-01installmentRate + 3.473e-01property

```
Call:
glm(formula = target ~ ., family = "binomial", data = train_data2)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       1.826e+00  3.122e-01   5.849 4.94e-09 ***
checking          3.668e-01  3.767e-01   0.974  0.33019
`credit history`  -1.140e+00  3.681e-01  -3.098  0.00195 **
`credit amount`   -1.062e-04  2.951e-05  -3.598  0.00032 ***
savings           3.155e-01  1.997e-01   1.580  0.11410
`installment rate` -2.487e-01  7.722e-02  -3.221  0.00128 **
Property          3.473e-01  1.932e-01   1.797  0.07230 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 982.41  on 799  degrees of freedom
Residual deviance: 937.55  on 793  degrees of freedom
AIC: 951.55

Number of Fisher Scoring iterations: 4
```

It is clear that the new model is more explainable. However, since I selected part of the factors and manipulated them based on need and prior knowledge, it may not perform as well as the previous model. The second model has a higher residual deviance (937.55) and AIC (951.55) compared to the first one (residual deviance of 700.97 and AIC of 798.97). Additionally, the prediction accuracy is 73.5%, indicating that it is not an ideal model compared to the first one. Therefore, I will stick with the first model to complete the second part of the question.

2. Since incorrectly identifying a bad customer as good is 5 times worse than incorrectly classifying a good customer as bad, I calculated all possible total costs with thresholds from 0 to 1, incrementing by 0.01. Assuming the unit cost of incorrectly classifying a good customer as bad is $1, the unit cost of incorrectly identifying a bad customer as good is $5.



As shown in the plot above, total costs continually decrease from the smallest threshold and stabilize around a threshold of 0.6. This makes sense because identifying a bad customer as good costs more, so we may decrease the probability of identifying a customer as good. The threshold should be at least greater than 0.5. In fact, the best threshold is 0.75, and the total cost is $113. The accuracy of the model with the best threshold is 0.675, which is lower than the previous setting. However, it results in lower costs in practice.

The confusion matrix and ROC plot are displayed below. The area under the curve (AUC) is 0.77.

		prediction		
		0	1	sum
actual	0	45	12	57
	1	53	90	143
	sum	98	102	200