

Analyzing and Predicting the Relationships between Sesame Street Viewership and Test Scores among School Children

Angela Wang, QiHan Zhou, Matthew Murray, Michelle Mao, Sara Lemus, Sophie Dalldorf

12/9/2021

Introduction

Methodology

Research Question 1

We decided to utilize three different types of models to gain inference and predict the difference in test scores that occurs after subjects watch Sesame Street: (1) least-squares regression model (2) ridge regression model and (3) regression tree model. There are six different test scores in the data set, so we fit three models for each difference in test score variable. Each model took the following variables as covariates: **site**, **sex**, **viewcat**, **setting**, **viewenc**. Before creating these models, we factored those variables to encode them as categoricals. Initially, our least-squares and ridge regression models also took into account all possible two-way interaction terms between the variables. However, we noticed that both the Akaike Information Criterion (AIC) and adjusted- R^2 values were higher for models that did not include any interaction effects. That reason, coupled with the lack of apparent interaction effects in our exploratory data analysis and the fact that we wanted our linear models to be very interpretable, is why we decided not to include any interaction effects in our final linear models.

One problem that we envisioned when evaluating and comparing the different models is that the tests are scored on different scales. For example, the scores for the test on knowledge of body parts (noted by **bodyDiff**) range from 0-32, while those of the test on letters (noted by **letDiff**) range from 0-58. To be able to aptly compare the mean squared error (MSE) between models, we also decided to convert each response variable to the same range. More specifically, we scaled each variable to the arbitrary range [0, 30]. Lastly, we randomly split the data between testing and training, using 70% of the data for training and 30% of the data for testing.

We first decided to use least-squares regression models due to the fact that they provide apt inference into the relationship between the covariates and the response variable. Thereafter, we decided to use a ridge regression model due to the fact that ridge regression often provides performance improvements by shrinking slope coefficients. While shrinkage may introduce bias to a model, it decreases the variance and increases the precision of the slope coefficient estimates. The shrinkage is achieved by applying a shrinkage parameter, λ , to the Euclidean norm of a slope coefficient. Doing so slightly increases the bias of our model (as least-squares regression coefficient estimates are unbiased) but can also significantly decrease the variance (and increase the precision) of our regression coefficients. This slight increase in bias but significant decrease in variance usually decreases the MSE of a model.

We tuned our λ parameter using 10-fold cross validation. More specifically, we computed the cross-validation error rate for our model for a grid of λ values. Thereafter, we selected the λ value for which the cross-validation error is the smallest.

Initially, we were inclined to use least absolute shrinkage and selection operator (LASSO) regression. The main advantage of LASSO regression over ridge regression is that LASSO regression performs variable selection by setting the slope coefficients of inert predictors to 0. The reason why we initially thought that

LASSO regression would work better than ridge regression is that in most of our linear models, only a small subset of variables are significant at a 95% significance level. However, our ridge regression models performed marginally better than our LASSO regression models, so for this reason, we decided to report the MSE's for the ridge regression models.

~Talk about why we chose regression tree models~

In the context of a regression tree, the deviance is simply the sum of the squared errors.

Research Question 2

In order to address our second research question, predicting whether a child came from an disadvantaged background or not based on their pretest scores and demographic information, we utilized a support vector machine (SVM). Our model uses the sex, age, and all pretest score variables to predict our response variable, site. Since our response variable is a categorical variable, a SVM is a valid choice to answer our research question. We also implemented a classification tree and a logistic model to answer this question as well; however, these models did not perform as well on our data (See Appendix for more details). Thus, a SVM was the most appropriate model choice for our research goals. Our full model formula is:

$$svm(site \sim female + male + age + bodyparts_{pretest} + letters_{pretest} + forms_{pretest} + numbers_{pretest} + relationalterms_{pretest} + class$$

We split our data into a 70% training set and 30% testing set and analyzed the performance of our model on the test set. Since we are interested in high predictive power, we implemented a variety of different methods. Standardizing the predictor variables in SVM and encoding categorical variables has been shown to improve performance for SVMs [1]. Thus, we used the standardized forms of the continuous variables in our model and encoded the female and male variables. We did indeed observe a small improvement in prediction accuracy across all models. However, one problem that particularly piqued our interests is that we are making no predictions for classes 4 or 5 for site. This could be due to sites 4 & 5 having a smaller number of observations than the other classes. Thus, we used the class weight formula below to assign weights to each class and specify “one versus one” comparison, which has been suggested to yield better prediction than “one versus all” [2].

$$w_j = \frac{n}{kn_j}, \text{ n is total number of data points, k is number of classes, } n_j \text{ is the number of data in class j}$$

We tested our model using linear, radial, sigmoid, and polynomial kernels and compared the predictive accuracy between these models. After the class weight assignment, the SVM models began to make prediction on class 4&5. However, doing so came at the cost of overall accuracy. Thus, more of the other observations are being misclassified, but the few observations of sites 4&5 are correctly classified. To remedy this issue, we began to experiment with the class weights and increase the sites 4&5 weights by roughly 0.5 until we reached the highest predictive power. The model with the highest accuracy was the linear kernel SVM with weights 0.8 on site 1, 0.87 on site 2, 0.75 on site 3, 1.5 on site 4, and 3 on site 5 and a cross validation selected cost parameter of 0.1. Since this model has the highest predictive accuracy of all others that we had tried, we settled on this model as our final model. There are no explicit tests for SVM model diagnostics; however, there seems to be no major issue with our model fit.

Results

Research Question 1

Table 1: Test Metrics

Response	Least.Squares.Reggression.Test.MSE	Ridge.Reggression.Test.MSE	Regression.Tree.Test.MSE
bodyDiff	24.01	21.61	20.60
letDiff	14.31	14.20	15.41
formDiff	13.26	12.83	14.92
numbDiff	15.51	14.63	15.91
relatDiff	18.94	19.86	19.89
clasfDiff	48.40	44.26	45.53

From the results of the above table, one can see that the ridge regression models have the best performance, although the performance of the regression trees are very similar . For each response variable except **bodyDiff**, the ridge regression model reports the lowest test MSE, although the difference in performance between the ridge regression and the regression tree is very marginal.

Table 2: Classification Model Accuracy Table

Model	Accuracy
Linear Kernel SVM	0.4028
Radial Kernel SVM	0.3611
Sigmoid Kernel SVM	0.2639
Polynomial Kernel SVM	0.2917
Random Forest	0.4444

Research Question 2

As evidenced in the table below, the class weighted linear kernel SVM has a prediction accuracy of 0.403 on our test data with a confidence interval of 0.289 to 0.525. From the confusion matrix, we can also see that our model predicts well for classes 2 and 3 and struggles slightly with classes 1,4, and 5. Specifically, our model has tends to predict class 3 often when the true class is 4, and it tends to predict classes 3 and 5 as commonly as class 1 when the true class is 1.

Table 3: Class Weighted Linear Kernel SVM Prediction Results

Results	Values
Accuracy	0.403
95% CI	(0.289, 0.525)

Table 4: Class Weighted Linear Kernel SVM Confusion Matrix

Truth	Class1.Prediction	Class2.Prediction	class3.Prediction	class4.Prediction	class5.Prediction
Class 1	6	1	6	2	5
Class 2	1	7	1	1	3
Class 3	0	1	11	1	3
Class 4	3	3	9	3	0
Class 5	0	1	1	1	2

Conclusion

Appendix

References

- [1] A Practical Guide to Support Vector Machines. (2003). National Taiwan University. <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- [2] One-vs-Rest and One-vs-One for Mutli-Class Classification. (2020). Machine Learning Mastery. <https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/>

Models

```
##
## Call:
## lm(formula = bodyDiff ~ (site + sex + age + viewcat + setting +
##     viewenc), data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3707  -2.7927   0.0525   2.5545  13.1411
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  16.42668    3.06383   5.361 2.92e-07 ***
## site2         0.58666    0.99737   0.588  0.55724
## site3         2.44827    0.96820   2.529  0.01244 *
## site4         2.96605    1.10488   2.685  0.00805 **
## site5         3.29854    1.45590   2.266  0.02485 *
## sex2        -1.00041    0.66732  -1.499  0.13586
## age          -0.08726    0.05558  -1.570  0.11845
## viewcat2     -0.60564    1.12908  -0.536  0.59244
## viewcat3      1.30568    1.09680   1.190  0.23568
## viewcat4      2.18215    1.12555   1.939  0.05434 .
## setting2     -1.02953    0.77713  -1.325  0.18718
## viewenc2     -0.37832    0.84111  -0.450  0.65349
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.253 on 156 degrees of freedom
## Multiple R-squared:  0.1585, Adjusted R-squared:  0.09917
## F-statistic: 2.671 on 11 and 156 DF,  p-value: 0.00362
```

```
## [1] 976.6991
## [1] 24.01045
## [1] 21.60675

##
## Call:
## lm(formula = letDiff ~ (site + sex + age + viewcat + setting +
##   viewenc), data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.2823  -2.9776  -0.3663   2.9275  11.1426
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.52353    3.32595   3.164  0.00187 **
## site2        3.08586    1.08270   2.850  0.00496 **
## site3       -2.85941    1.05103  -2.721  0.00726 **
## site4       -1.07861    1.19940  -0.899  0.36989
## site5       -0.38501    1.58046  -0.244  0.80786
## sex2         0.37911    0.72441   0.523  0.60148
## age          0.05050    0.06034   0.837  0.40390
## viewcat2     1.59616    1.22568   1.302  0.19474
## viewcat3     4.85135    1.19064   4.075 7.32e-05 ***
## viewcat4     4.77123    1.22184   3.905  0.00014 ***
## setting2     0.70346    0.84362   0.834  0.40563
## viewenc2    -1.56705    0.91307  -1.716  0.08810 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.617 on 156 degrees of freedom
## Multiple R-squared:  0.37, Adjusted R-squared:  0.3256
## F-statistic: 8.33 on 11 and 156 DF, p-value: 2.043e-11

## [1] 1004.281
## [1] 14.31277
## [1] 14.19572

##
## Call:
## lm(formula = formDiff ~ (site + sex + age + viewcat + setting +
##   viewenc), data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.3637  -2.5344   0.2658   2.7054  14.1942
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.04193    3.16388   5.070 1.11e-06 ***
## site2        0.97029    1.02994   0.942  0.3476
## site3        0.73495    0.99982   0.735  0.4634
## site4        0.92962    1.14096   0.815  0.4164
```

```

## site5      1.00499    1.50344    0.668    0.5048
## sex2       0.09439    0.68911    0.137    0.8912
## age       -0.05234    0.05740   -0.912    0.3632
## viewcat2    0.99101    1.16595    0.850    0.3966
## viewcat3    1.65614    1.13262    1.462    0.1457
## viewcat4    2.47768    1.16230    2.132    0.0346 *
## setting2   -0.08994    0.80251   -0.112    0.9109
## viewenc2   -0.42935    0.86858   -0.494    0.6218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.392 on 156 degrees of freedom
## Multiple R-squared:  0.06302,    Adjusted R-squared:  -0.003048
## F-statistic: 0.9539 on 11 and 156 DF,  p-value: 0.4909

## [1] 987.4956
## [1] 13.25639
## [1] 12.82502

##
## Call:
## lm(formula = numbDiff ~ (site + sex + age + viewcat + setting +
##     viewenc), data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.6433  -2.6010   0.1375   2.3541  10.3247
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.388591   3.084422   4.989 1.6e-06 ***
## site2       2.561427   1.004073   2.551 0.01170 *
## site3       0.768616   0.974707   0.789 0.43157
## site4       1.151651   1.112304   1.035 0.30210
## site5       1.946294   1.465685   1.328 0.18615
## sex2        0.013992   0.671801   0.021 0.98341
## age         0.009505   0.055956   0.170 0.86534
## viewcat2    1.841715   1.136668   1.620 0.10719
## viewcat3    3.199614   1.104174   2.898 0.00430 **
## viewcat4    3.636674   1.133111   3.209 0.00161 **
## setting2    0.596707   0.782354   0.763 0.44679
## viewenc2    0.589407   0.846765   0.696 0.48742
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.281 on 156 degrees of freedom
## Multiple R-squared:  0.139,    Adjusted R-squared:  0.07825
## F-statistic: 2.289 on 11 and 156 DF,  p-value: 0.01273

## [1] 978.9495
## [1] 15.5095
## [1] 14.62731
##

```

```
## Call:
## lm(formula = relatDiff ~ (site + sex + age + viewcat + setting +
##     viewenc), data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9246  -2.7114   0.3439   2.2185  14.4415
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.48969    3.17203   5.829 3.1e-08 ***
## site2        -0.05900    1.03259  -0.057  0.9545
## site3         1.66242    1.00239   1.658  0.0992 .
## site4         2.41955    1.14390   2.115  0.0360 *
## site5        -0.40561    1.50731  -0.269  0.7882
## sex2          0.31932    0.69088   0.462  0.6446
## age          -0.09729    0.05755  -1.691  0.0929 .
## viewcat2       1.05190    1.16895   0.900  0.3696
## viewcat3       0.96616    1.13554   0.851  0.3962
## viewcat4       2.95954    1.16529   2.540  0.0121 *
## setting2      -0.60666    0.80457  -0.754  0.4520
## viewenc2       0.19462    0.87082   0.223  0.8234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.403 on 156 degrees of freedom
## Multiple R-squared:  0.1072, Adjusted R-squared:  0.04421
## F-statistic: 1.702 on 11 and 156 DF, p-value: 0.07739
## [1] 988.3598
## [1] 18.94131
## [1] 19.86283
##
## Call:
## lm(formula = clasfDiff ~ (site + sex + age + viewcat + setting +
##     viewenc), data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.1077  -4.0442   0.2665   3.8807  14.5260
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.951998    4.455915   2.458  0.0151 *
## site2         1.799749    1.450536   1.241  0.2166
## site3         0.420798    1.408112   0.299  0.7655
## site4         2.040728    1.606891   1.270  0.2060
## site5         1.497017    2.117404   0.707  0.4806
## sex2          1.500233    0.970519   1.546  0.1242
## age          -0.008915    0.080838  -0.110  0.9123
## viewcat2       3.041609    1.642090   1.852  0.0659 .
## viewcat3       3.072479    1.595147   1.926  0.0559 .
## viewcat4       4.060404    1.636951   2.480  0.0142 *
```

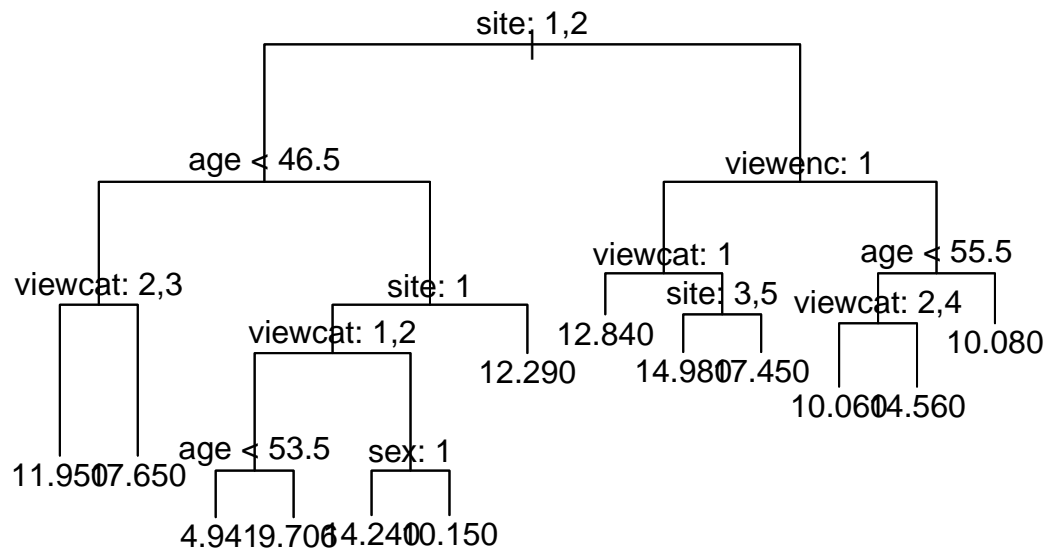
```
## setting2      0.477618   1.130229   0.423   0.6732
## viewenc2     -1.088979   1.223281  -0.890   0.3747
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.185 on 156 degrees of freedom
## Multiple R-squared:  0.09426,    Adjusted R-squared:  0.03039
## F-statistic: 1.476 on 11 and 156 DF,  p-value: 0.1455

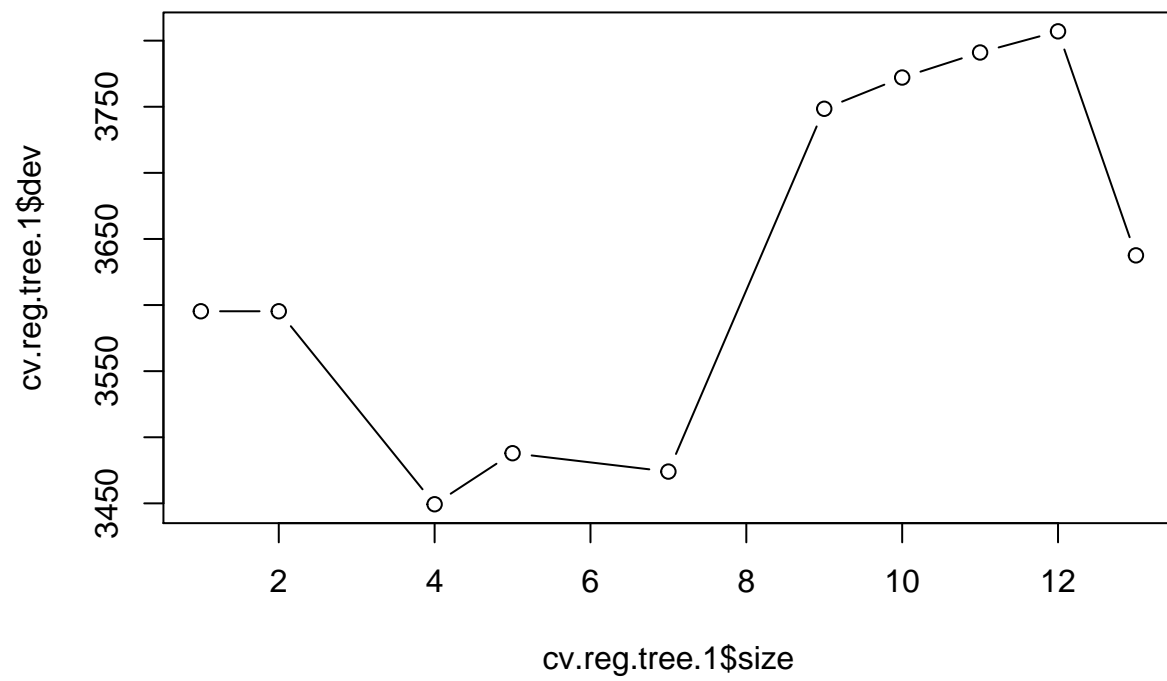
## [1] 1102.553
## [1] 48.398
## [1] 44.26185
```

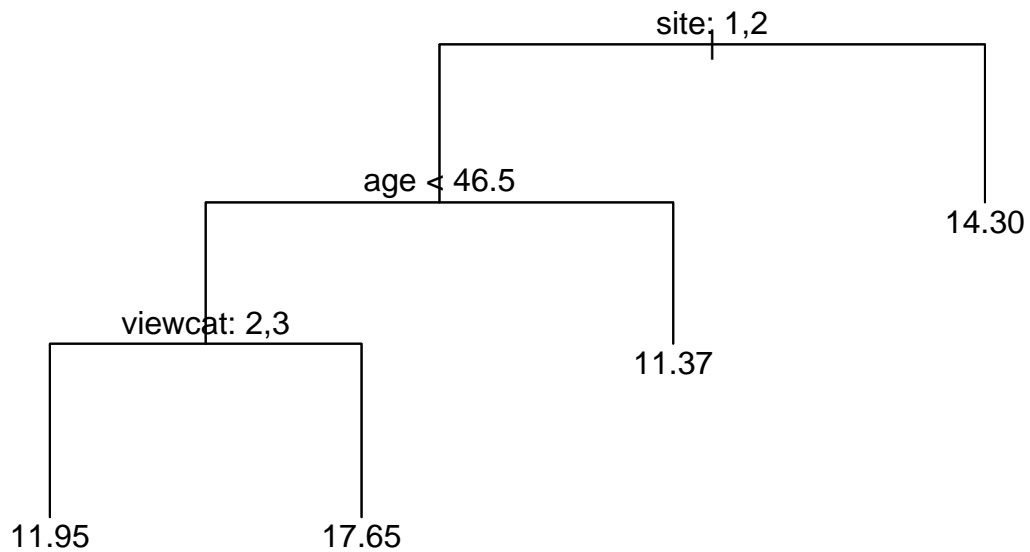
Regression Tree Models

Model 1

```
##
## Regression tree:
## tree(formula = bodyDiff ~ site + sex + age + viewcat + setting +
##       viewenc, data = sesame.q1, subset = train)
## Variables actually used in tree construction:
## [1] "site"    "age"     "viewcat" "sex"     "viewenc"
## Number of terminal nodes: 13
## Residual mean deviance: 14.16 = 2194 / 155
## Distribution of residuals:
##      Min.    1st Qu.    Median      Mean    3rd Qu.      Max.
## -11.08000 -2.45100  -0.06303   0.00000   2.16900  12.55000
```



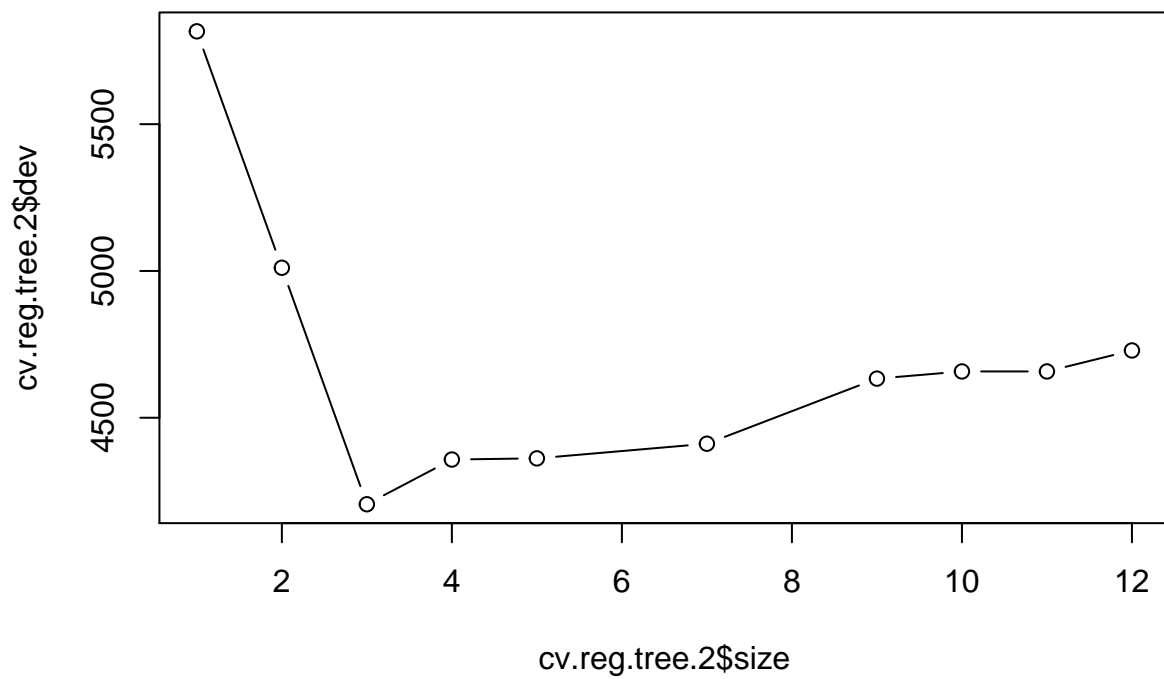


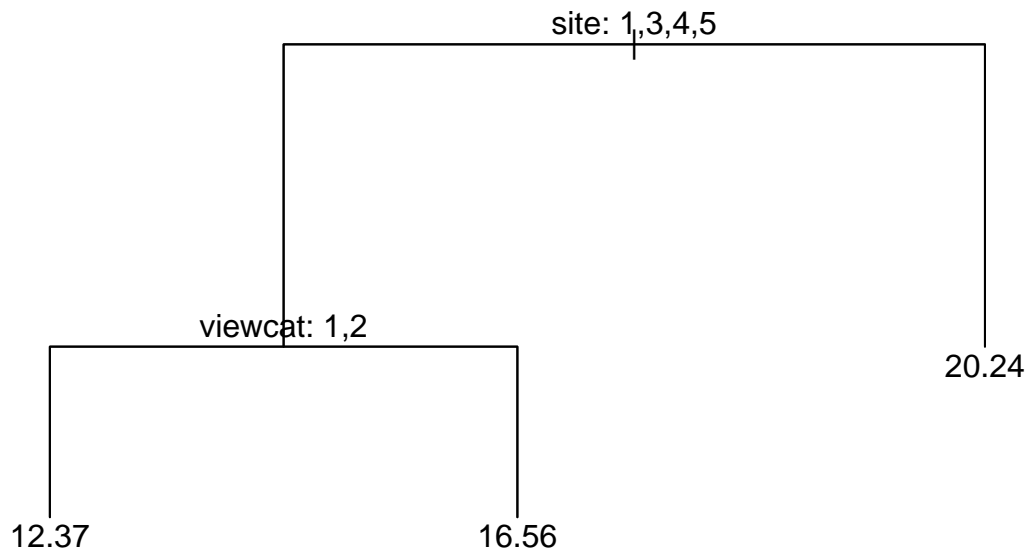
```
## [1] 20.60159
```

Model 2

```
##
## Regression tree:
## tree(formula = letDiff ~ site + sex + age + viewcat + setting +
##       viewenc, data = sesame.q1, subset = train)
## Variables actually used in tree construction:
## [1] "site" "viewcat" "age" "viewenc" "setting"
## Number of terminal nodes: 12
## Residual mean deviance: 18.63 = 2906 / 156
## Distribution of residuals:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -17.560  -2.430   0.000   0.000   2.765   9.587
```



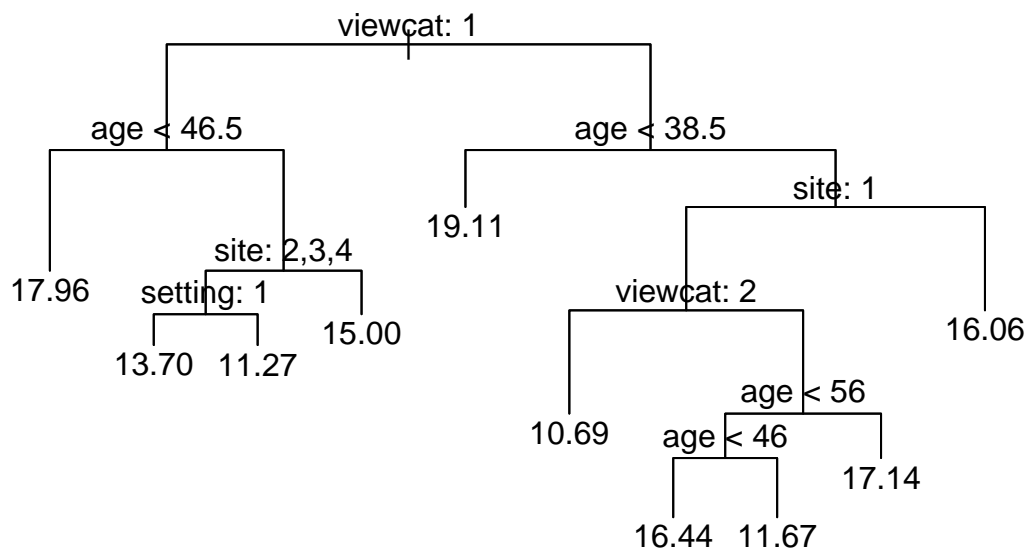


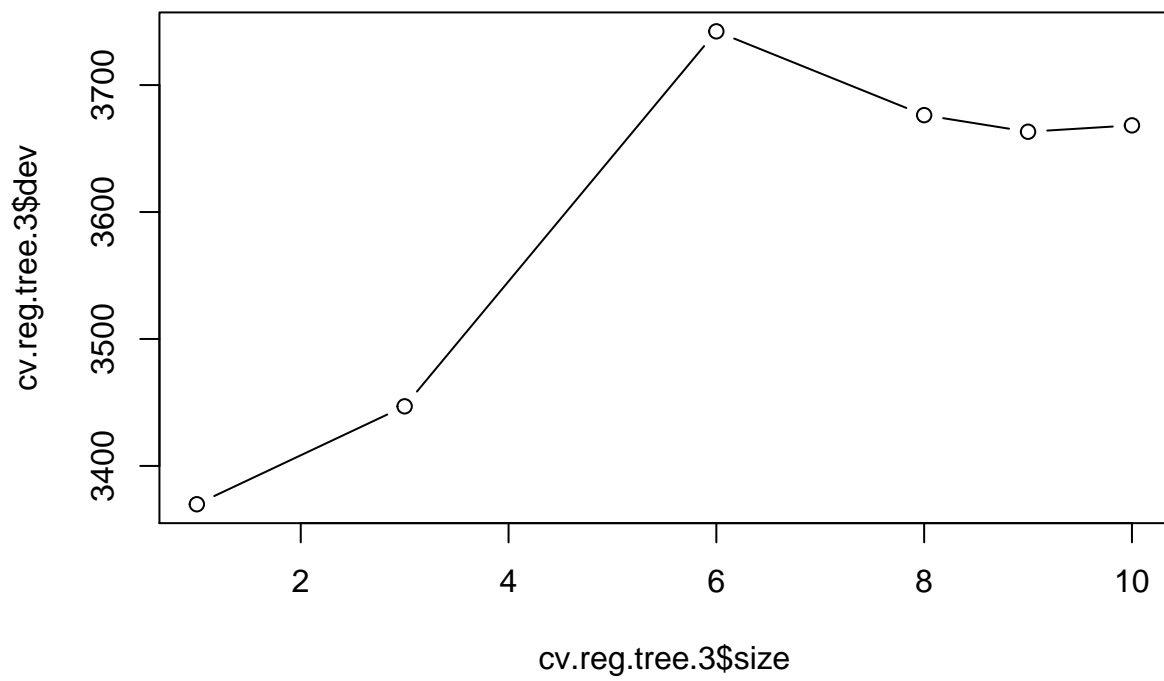


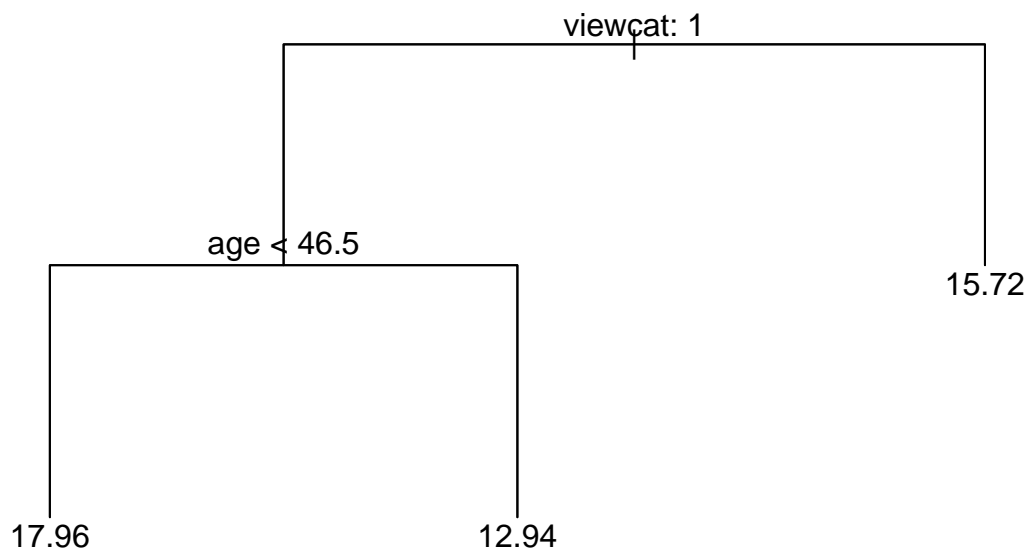
```
## [1] 15.4124
```

Model 3

```
##
## Regression tree:
## tree(formula = formDiff ~ site + sex + age + viewcat + setting +
##       viewenc, data = sesame.q1, subset = train)
## Variables actually used in tree construction:
## [1] "viewcat" "age"      "site"      "setting"
## Number of terminal nodes: 10
## Residual mean deviance: 15.88 = 2509 / 158
## Distribution of residuals:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -11.610  -1.667   0.129   0.000   2.159   13.940
```



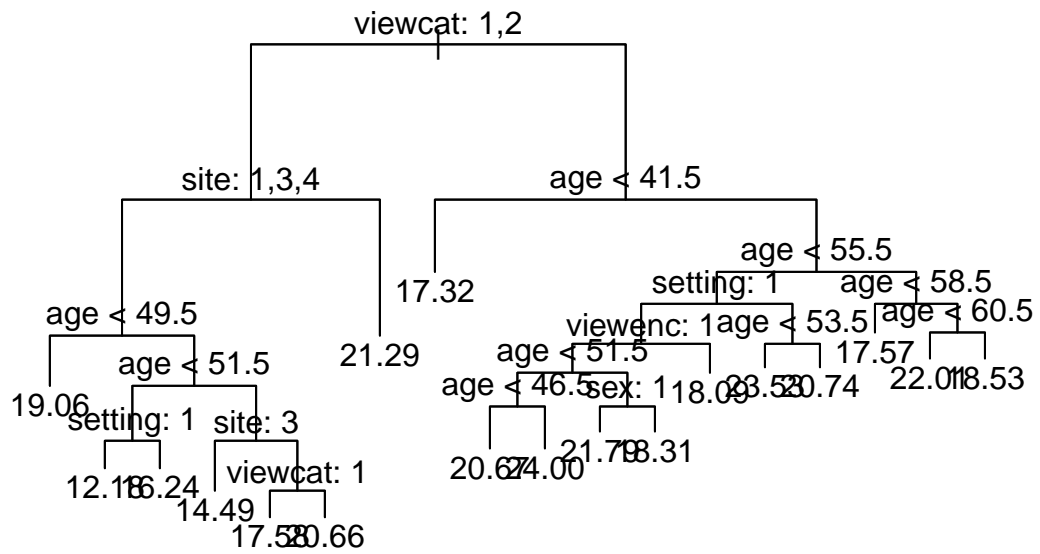


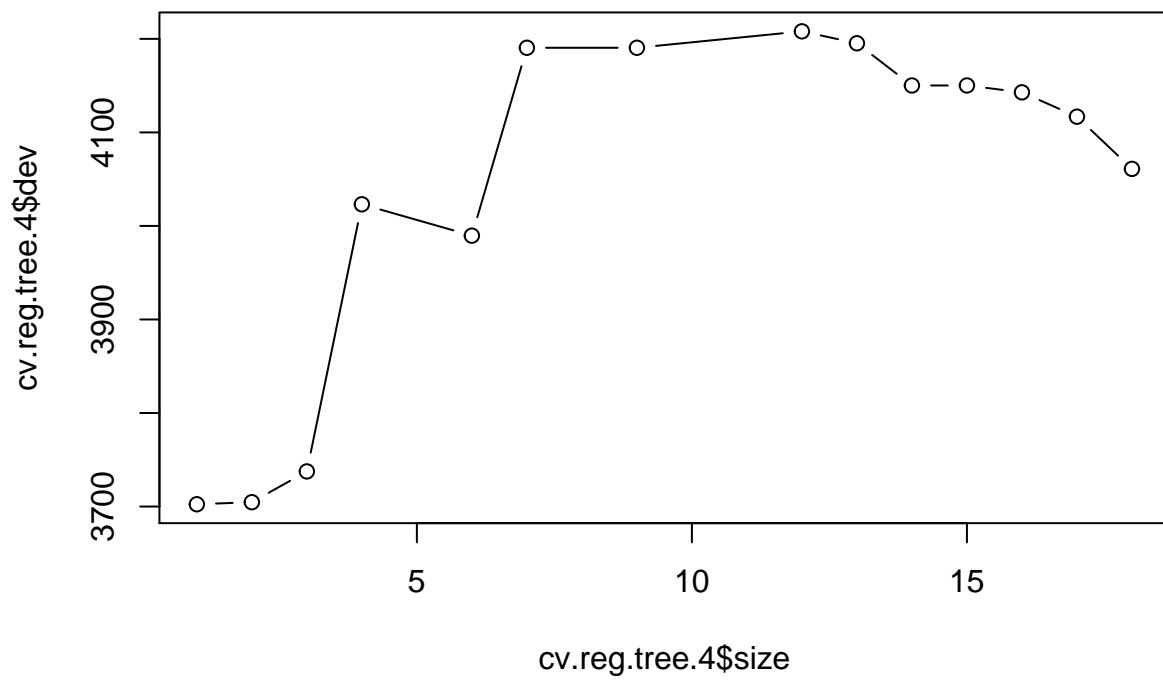


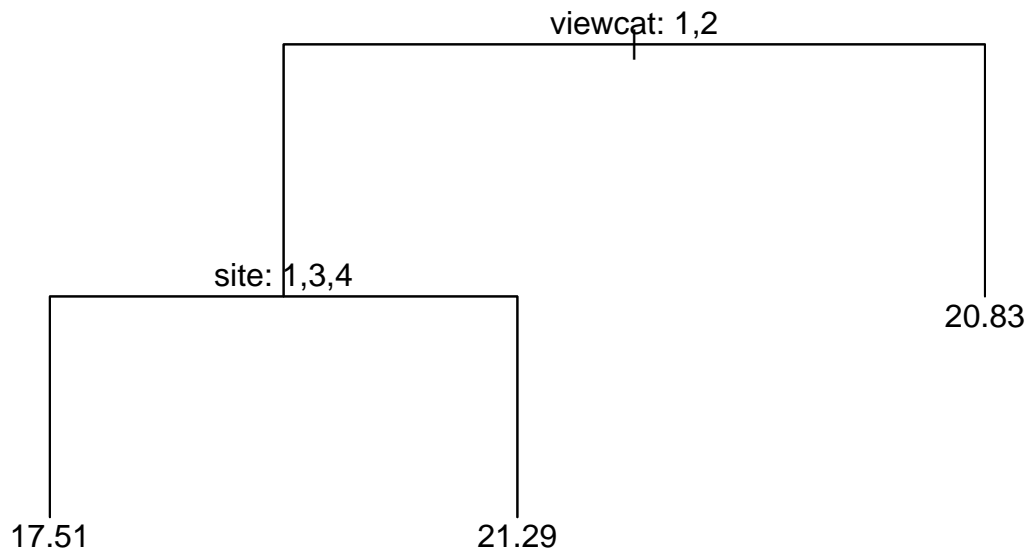
```
## [1] 14.92273
```

Model 4

```
##
## Regression tree:
## tree(formula = numbDiff ~ site + sex + age + viewcat + setting +
##       viewenc, data = sesame.q1, subset = train)
## Number of terminal nodes: 18
## Residual mean deviance: 13.64 = 2046 / 150
## Distribution of residuals:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -18.0900 -2.1210  0.2647  0.0000  2.1180 11.4700
```



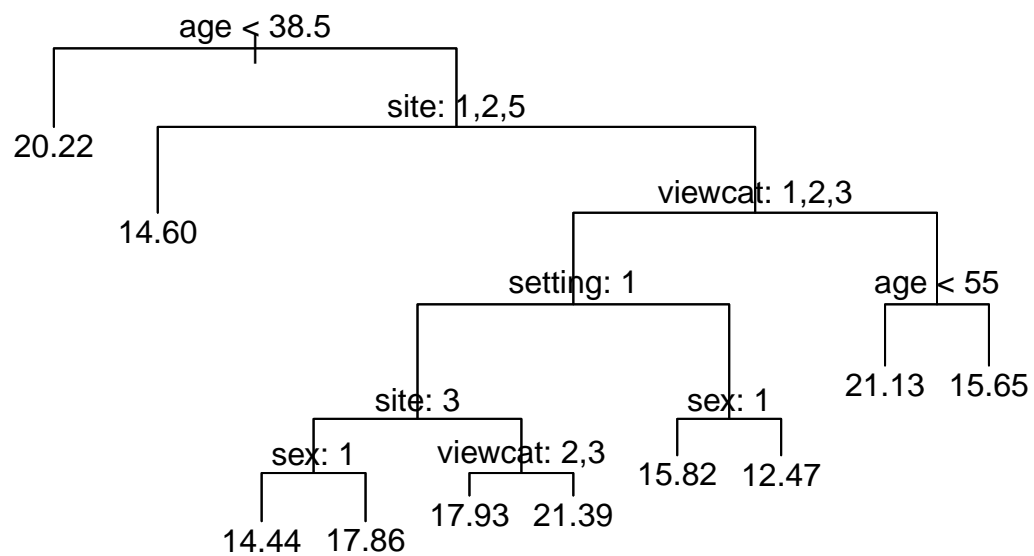


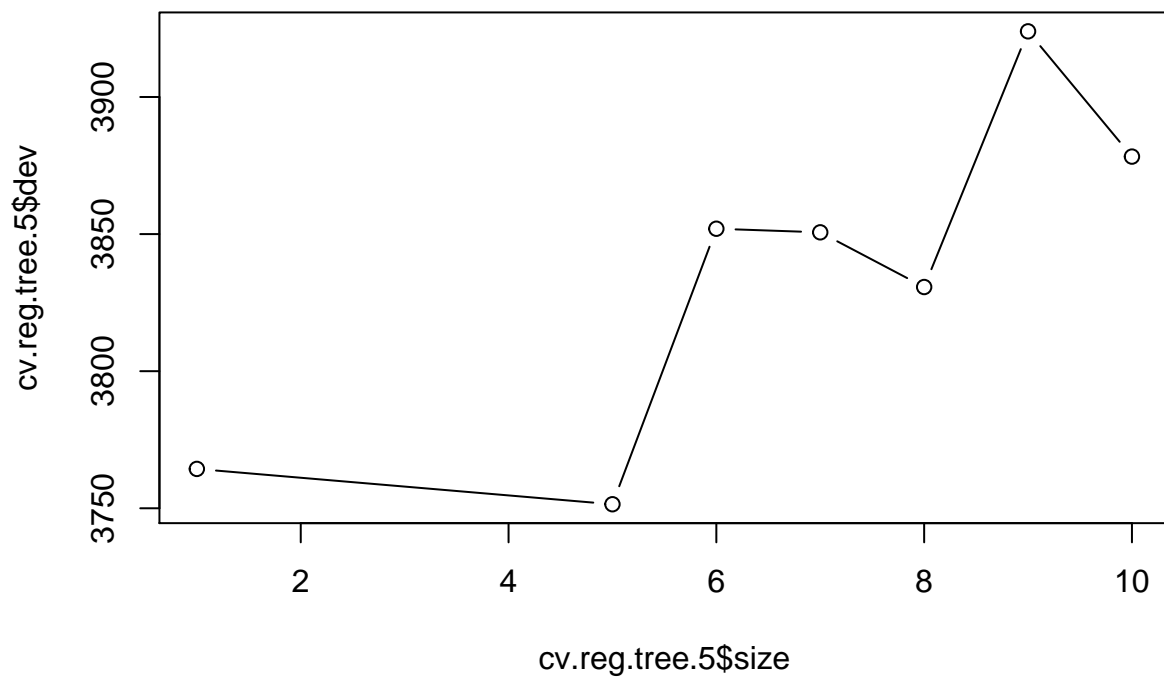


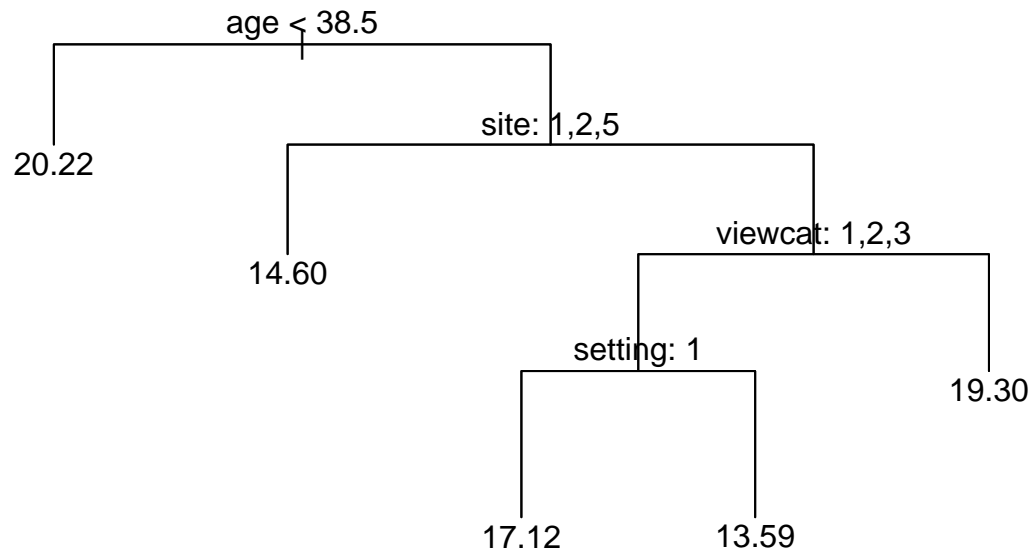
```
## [1] 15.90771
```

Model 5

```
##
## Regression tree:
## tree(formula = relatDiff ~ site + sex + age + viewcat + setting +
##       viewenc, data = sesame.q1, subset = train)
## Variables actually used in tree construction:
## [1] "age"      "site"     "viewcat"  "setting"  "sex"
## Number of terminal nodes: 10
## Residual mean deviance: 15.3 = 2418 / 158
## Distribution of residuals:
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -10.6900 -1.5650 -0.2514  0.0000  1.8950  11.4900
```



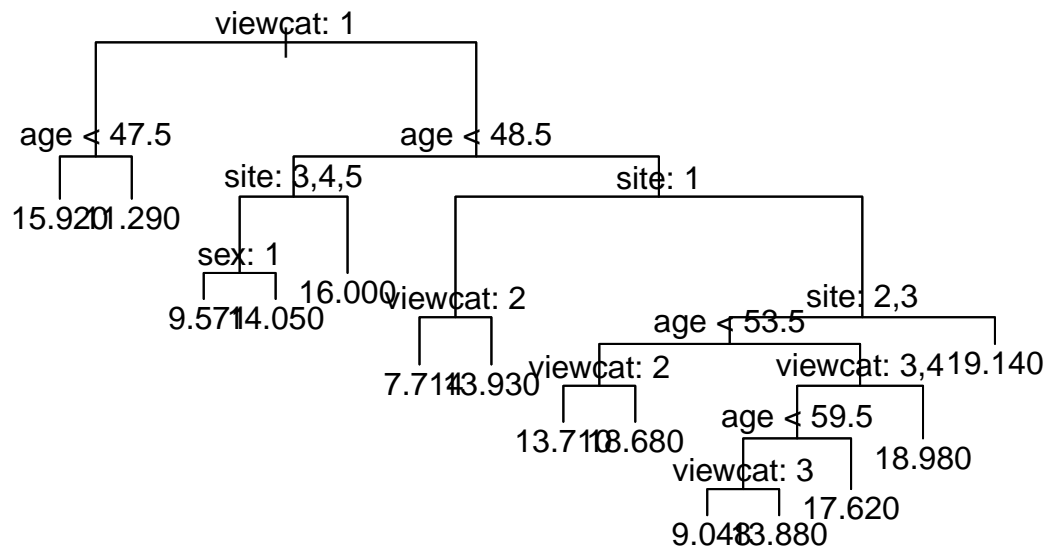


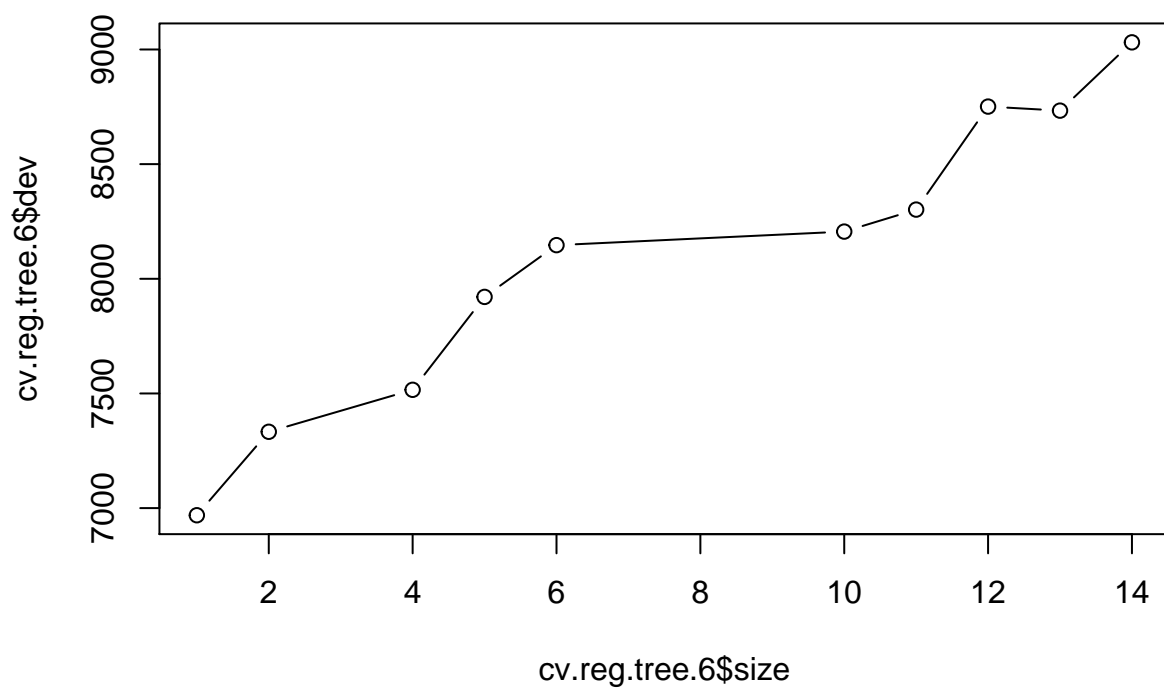


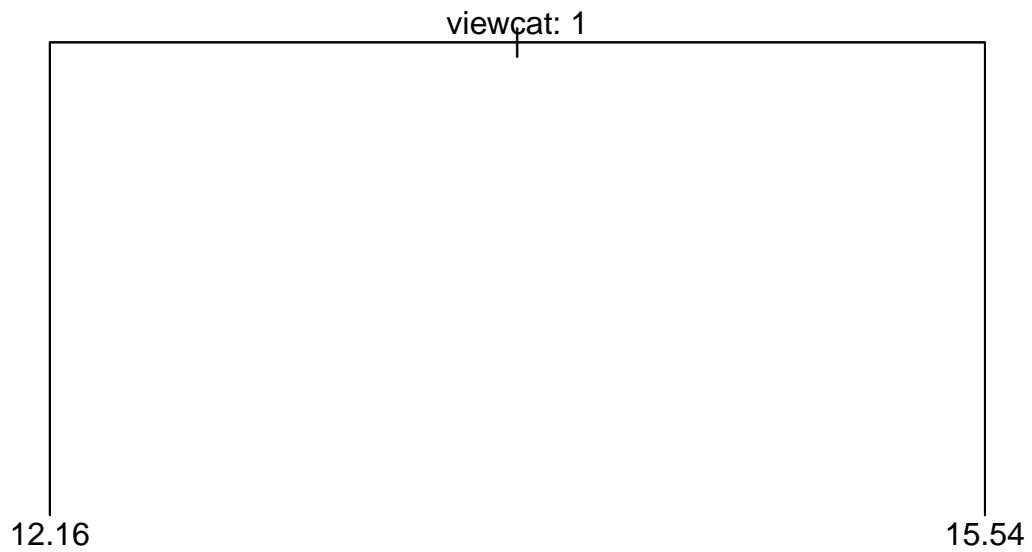
```
## [1] 19.88506
```

Model 6

```
##
## Regression tree:
## tree(formula = clasfDiff ~ site + sex + age + viewcat + setting +
##       viewenc, data = sesame.q1, subset = train)
## Variables actually used in tree construction:
## [1] "viewcat" "age"      "site"      "sex"
## Number of terminal nodes: 14
## Residual mean deviance: 29.66 = 4568 / 154
## Distribution of residuals:
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## -16.00000  -3.17300   0.01852   0.00000   3.71400  13.10000
```







[1] 45.52784

Q.2 Classification Question: Can we use the pre-test scores and other demographic variables to predict which region the children came from?

Extra SVM models