```{r}
# The data has 234560 observations, or rows, and 28 variables, or columns
dim(data)
```

```
[1] 234560     28
```

```{r}
# Using the na.omit() function to simply remove the rows with missing values. By using this we delete 234560-221520 = 13040 rows, which is 5% of the original data, and I think it's fine.
data <- na.omit(data)
dim(data)
```

Data Cleaning

- The data has 234560 observations, or rows, and 28 variables, or columns. I used the na.omit() function to simply remove the rows with missing values. By using this we delete 234560-221520 = 13040 rows, which is 5% of the original data, and I think it's fine.

Data Cleaning

```
summary(data)
```

```
     hh_id            year          month       zipcode          control          treatment          lusage          luse1
 Min.   :     3   Min.   :2010   Min.   :4   Min.   : 1.00   Min.   :0.0000   Min.   :0.0000   Min.   :3.917   Min.   :-0.2007
 1st Qu.:11853   1st Qu.:2010   1st Qu.:5   1st Qu.:28.00   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:5.967   1st Qu.: 5.7456
 Median :23712   Median :2010   Median :6   Median :42.00   Median :1.0000   Median :0.0000   Median :6.412   Median : 6.1469
 Mean   :23668   Mean   :2010   Mean   :6   Mean   :38.35   Mean   :0.7957   Mean   :0.2043   Mean   :6.369   Mean   : 6.1118
 3rd Qu.:35507   3rd Qu.:2011   3rd Qu.:7   3rd Qu.:54.00   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:6.816   3rd Qu.: 6.5190
 Max.   :47356   Max.   :2011   Max.   :8   Max.   :76.00   Max.   :1.0000   Max.   :1.0000   Max.   :8.318   Max.   : 7.9884
     luse2           luse3           luse4           luse5           luse6          children         hhsize2
 Min.   :0.727   Min.   :1.571   Min.   :1.654   Min.   :1.107   Min.   :1.184   Min.   :0.0000   Min.   :0.000
 1st Qu.:5.745   1st Qu.:5.789   1st Qu.:5.990   1st Qu.:6.136   1st Qu.:5.842   1st Qu.:0.0000   1st Qu.:0.000
 Median :6.159   Median :6.210   Median :6.427   Median :6.570   Median :6.249   Median :0.0000   Median :0.000
 Mean   :6.114   Mean   :6.169   Mean   :6.376   Mean   :6.506   Mean   :6.200   Mean   :0.2991   Mean   :0.313
 3rd Qu.:6.522   3rd Qu.:6.592   3rd Qu.:6.813   3rd Qu.:6.937   3rd Qu.:6.607   3rd Qu.:1.0000   3rd Qu.:1.000
 Max.   :7.883   Max.   :7.809   Max.   :8.027   Max.   :8.232   Max.   :7.852   Max.   :1.0000   Max.   :1.000
    hhsize3          hhsize4          hhsize5         hhsize5plus         income2           income3           income4
 Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
 Median :0.0000   Median :0.0000   Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000   Median :0.0000
 Mean   :0.2411   Mean   :0.1562   Mean   :0.08627   Mean   :0.05548   Mean   :0.05196   Mean   :0.09295   Mean   :0.1128
 3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
    income5          income6          income7          income8          income9            owner
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1.0000
 Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000   Median :0.00000   Median :1.0000
 Mean   :0.1232   Mean   :0.3024   Mean   :0.1361   Mean   :0.0548   Mean   :0.03088   Mean   :0.8429
 3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
```

- This is a summary of the whole dataset, and we need to get rid of the outlier (luse1 < 0), since log is always greater than 0, then we delete 221520-221510 = 10 rows and get the new dataset: data2.

Data Cleaning

- This is the summary of the new dataset, and it looks good.

```
data2 <- subset(data, 0 < luse1)
summary(data2)
dim(data2)
```

```
     hh_id            year          month      zipcode         control         treatment          lusage          luse1
 Min.   :     3   Min.   :2010   Min.   :4   Min.   : 1.00   Min.   :0.0000   Min.   :0.0000   Min.   :3.917   Min.   :0.5785
 1st Qu.:11851   1st Qu.:2010   1st Qu.:5   1st Qu.:28.00   1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:5.967   1st Qu.:5.7456
 Median :23712   Median :2010   Median :6   Median :42.00   Median :1.0000   Median :0.0000   Median :6.412   Median :6.1469
 Mean   :23667   Mean   :2010   Mean   :6   Mean   :38.35   Mean   :0.7957   Mean   :0.2043   Mean   :6.369   Mean   :6.1121
 3rd Qu.:35506   3rd Qu.:2011   3rd Qu.:7   3rd Qu.:54.00   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:6.816   3rd Qu.:6.5190
 Max.   :47356   Max.   :2011   Max.   :8   Max.   :76.00   Max.   :1.0000   Max.   :1.0000   Max.   :8.318   Max.   :7.9884
     luse2           luse3           luse4           luse5           luse6          children         hhsize2
 Min.   :0.8812   Min.   :1.571   Min.   :1.654   Min.   :1.107   Min.   :1.184   Min.   :0.0000   Min.   :0.000
 1st Qu.:5.7451   1st Qu.:5.789   1st Qu.:5.990   1st Qu.:6.136   1st Qu.:5.842   1st Qu.:0.0000   1st Qu.:0.000
 Median :6.1590   Median :6.210   Median :6.427   Median :6.570   Median :6.249   Median :0.0000   Median :0.000
 Mean   :6.1139   Mean   :6.169   Mean   :6.376   Mean   :6.506   Mean   :6.200   Mean   :0.2991   Mean   :0.313
 3rd Qu.:6.5219   3rd Qu.:6.592   3rd Qu.:6.813   3rd Qu.:6.937   3rd Qu.:6.607   3rd Qu.:1.0000   3rd Qu.:1.000
 Max.   :7.8834   Max.   :7.809   Max.   :8.027   Max.   :8.232   Max.   :7.852   Max.   :1.0000   Max.   :1.000
    hhsize3          hhsize4          hhsize5         hhsize5plus         income2           income3           income4
 Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
 Median :0.0000   Median :0.0000   Median :0.00000   Median :0.00000   Median :0.00000   Median :0.00000   Median :0.0000
 Mean   :0.2411   Mean   :0.1562   Mean   :0.08627   Mean   :0.05548   Mean   :0.05196   Mean   :0.09295   Mean   :0.1128
 3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
    income5          income6          income7         income8           income9            owner
 Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:1.0000
 Median :0.0000   Median :0.0000   Median :0.000   Median :0.00000   Median :0.00000   Median :1.0000
 Mean   :0.1232   Mean   :0.3024   Mean   :0.136   Mean   :0.05481   Mean   :0.03088   Mean   :0.8429
 3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:0.000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
[1] 221510     28
```

The correlation between variables

```
cor(data2[sapply(data2, is.numeric)])
```

	hh_id	year	month	zipcode	control	treatment	lusage	luse1
hh_id	1.0000000000	0.00000000	0.0000000	0.0007317851	0.0015116375	-0.0015116375	0.007485496	0.001724315
year	0.0000000000	1.00000000	0.0000000	0.0000000000	0.0000000000	0.0000000000	-0.019973329	0.000000000
month	0.0000000000	0.00000000	1.0000000	0.0000000000	0.0000000000	0.0000000000	0.270313250	0.000000000
zipcode	0.0007317851	0.00000000	0.0000000	1.0000000000	0.0005484504	-0.0005484504	-0.031653221	-0.023634401
control	0.0015116375	0.00000000	0.0000000	0.0005484504	1.0000000000	-1.0000000000	0.006534483	0.004788553
treatment	-0.0015116375	0.00000000	0.0000000	-0.0005484504	-1.0000000000	1.0000000000	-0.006534483	-0.004788553
lusage	0.0074854963	-0.01997333	0.2703132	-0.0316532210	0.0065344831	-0.0065344831	1.000000000	0.728389205
luse1	0.0017243154	0.00000000	0.0000000	-0.0236344014	0.0047885535	-0.0047885535	0.728389205	1.000000000
luse2	0.0063292753	0.00000000	0.0000000	-0.0240713973	0.0061529215	-0.0061529215	0.777279874	0.938130594
luse3	0.0089644561	0.00000000	0.0000000	-0.0441316325	0.0054650247	-0.0054650247	0.795454746	0.867962889
luse4	0.0115393772	0.00000000	0.0000000	-0.0403619122	0.0057604748	-0.0057604748	0.804185667	0.799145526
luse5	0.0118584370	0.00000000	0.0000000	-0.0328202677	0.0074125806	-0.0074125806	0.803073529	0.770741166
luse6	0.0079465119	0.00000000	0.0000000	-0.0348375362	0.0038017485	-0.0038017485	0.806769731	0.854585637
children	0.0131793494	0.00000000	0.0000000	0.0288140386	0.0046109643	-0.0046109643	0.143570514	0.132521575
hhsize2	0.0074017036	0.00000000	0.0000000	-0.0012677892	-0.0128973753	0.0128973753	-0.087193742	-0.080248465
hhsize3	-0.0042172589	0.00000000	0.0000000	0.0166046294	-0.0067270548	0.0067270548	0.024806702	0.021903374
hhsize4	-0.0035455090	0.00000000	0.0000000	-0.0041570896	0.0015309136	-0.0015309136	0.095803625	0.093103717
hhsize5	0.0090556648	0.00000000	0.0000000	-0.0162113470	-0.0034036830	0.0034036830	0.094545949	0.094607691
hhsize5plus	0.0055059518	0.00000000	0.0000000	-0.0087508719	0.0044582329	-0.0044582329	0.112560390	0.110000370
income2	0.0047132627	0.00000000	0.0000000	0.0236412759	-0.0064681910	0.0064681910	-0.085118467	-0.074487569
income3	0.0035476193	0.00000000	0.0000000	0.0524012800	-0.0074401042	0.0074401042	-0.079349384	-0.076783450
income4	-0.0047330176	0.00000000	0.0000000	0.0275804148	0.0001428262	-0.0001428262	-0.080982095	-0.076030751
income5	0.0142484396	0.00000000	0.0000000	0.0072535453	0.0016372570	-0.0016372570	-0.029883784	-0.028281222
income6	-0.0046907376	0.00000000	0.0000000	-0.0516198315	-0.0061496835	0.0061496835	0.053919298	0.049141721
income7	-0.0064427424	0.00000000	0.0000000	-0.0601013276	-0.0059994818	0.0059994818	0.129631007	0.122139556
income8	-0.0064520904	0.00000000	0.0000000	-0.0878965142	-0.0068622615	0.0068622615	0.096067223	0.088604309
income9	-0.0063047394	0.00000000	0.0000000	-0.0478574713	-0.0046870976	0.0046870976	0.092558149	0.086135343
owner	-0.0028588280	0.00000000	0.0000000	-0.1132942997	-0.0286297053	0.0286297053	0.077638038	0.073671353

	hhsize4	hhsize5	hhsize5plus	income2	income3	income4	income5	income6
hh_id	-0.003545509	0.009055665	0.005505952	0.004713263	0.003547619	-0.0047330176	0.014248440	-0.004690738
year	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.0000000000	0.000000000	0.000000000
month	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.0000000000	0.000000000	0.000000000
zipcode	-0.004157090	-0.016211347	-0.008750872	0.023641276	0.052401280	0.0275804148	0.007253545	-0.051619831
control	0.001530914	-0.003403683	0.004458233	-0.006468191	-0.007440104	0.0001428262	0.001637257	-0.006149683
treatment	-0.001530914	0.003403683	-0.004458233	0.006468191	0.007440104	-0.0001428262	-0.001637257	0.006149683
lusage	0.095803625	0.094545949	0.112560390	-0.085118467	-0.079349384	-0.0809820947	-0.029883784	0.053919298
luse1	0.093103717	0.094607691	0.110000370	-0.074487569	-0.076783450	-0.0760307514	-0.028281222	0.049141721
luse2	0.101996977	0.101166095	0.121004867	-0.084837084	-0.086246938	-0.0815579578	-0.031928982	0.053423730
luse3	0.106110166	0.108174337	0.126180064	-0.092188327	-0.094429903	-0.0847333177	-0.032826580	0.060403484
luse4	0.107669248	0.108488947	0.123275981	-0.098045431	-0.094313196	-0.0843971862	-0.027521152	0.064853904
luse5	0.107739417	0.107612124	0.121184162	-0.101414404	-0.089800380	-0.0819063575	-0.025346229	0.067087210
luse6	0.103452176	0.105371283	0.125353164	-0.095850181	-0.092166818	-0.0842699741	-0.029657201	0.051424318
children	0.175172970	0.203103895	0.212063821	-0.018789513	-0.015927241	-0.0611634840	-0.037378961	0.014197033
hhsize2	-0.290404667	-0.207399115	-0.163589774	0.011733869	0.038409399	0.0372895384	0.022669345	-0.012019788
hhsize3	-0.242520923	-0.173201848	-0.136616066	-0.014515395	0.015461103	0.0135753656	0.008587757	0.001551428
hhsize4	1.000000000	-0.132204692	-0.104278824	-0.025652702	-0.021248608	-0.0358485482	-0.025128764	0.005851095
hhsize5	-0.132204692	1.000000000	-0.074473100	-0.012531954	-0.029699574	-0.0353322006	-0.005635574	0.004224425
hhsize5plus	-0.104278824	-0.074473100	1.000000000	-0.016758184	-0.033441552	-0.0196987424	-0.023677997	-0.009308212
income2	-0.025652702	-0.012531954	-0.016758184	1.000000000	-0.074945335	-0.0834660391	-0.087775547	-0.154148979
income3	-0.021248608	-0.029699574	-0.033441552	-0.074945335	1.000000000	-0.1141296219	-0.120022348	-0.210779916
income4	-0.035848548	-0.035332201	-0.019698742	-0.083466039	-0.114129622	1.0000000000	-0.133667960	-0.234743961
income5	-0.025128764	-0.005635574	-0.023677997	-0.087775547	-0.120022348	-0.1336679598	1.000000000	-0.246864231
income6	0.005851095	0.004224425	-0.009308212	-0.154148979	-0.210779916	-0.2347439613	-0.246864231	1.000000000
income7	0.057082934	0.048341126	0.062030973	-0.092892212	-0.127018763	-0.1414598133	-0.148763648	-0.261254590
income8	0.038448118	0.038337357	0.046480329	-0.056374166	-0.077084792	-0.0858487363	-0.090281267	-0.158549456
income9	0.025991957	0.039958983	0.051356385	-0.041789809	-0.057142464	-0.0636391196	-0.066924926	-0.117531698
owner	0.065123653	0.067703863	0.084042468	-0.106297590	-0.151042407	-0.0492430257	0.029396729	0.200801264

	luse2	luse3	luse4	luse5	luse6	children	hhsize2	hhsize3
hh_id	0.006329275	0.008964456	0.011539377	0.011858437	0.007946512	0.013179349	0.007401704	-0.004217259
year	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
month	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000	0.000000000
zipcode	-0.024071397	-0.044131633	-0.040361912	-0.032820268	-0.034837536	0.028814039	-0.001267789	0.016604629
control	0.006152921	0.005465025	0.005760475	0.007412581	0.003801749	0.004610964	-0.012897375	-0.006727055
treatment	-0.006152921	-0.005465025	-0.005760475	-0.007412581	-0.003801749	-0.004610964	0.012897375	0.006727055
lusage	0.777279874	0.795454746	0.804185667	0.803073529	0.806769731	0.143570514	-0.087193742	0.024806702
luse1	0.938130594	0.867962889	0.799145526	0.770741166	0.854585637	0.132521575	-0.080248465	0.021903374
luse2	1.000000000	0.960718766	0.893925078	0.858338552	0.919606310	0.138596309	-0.086571072	0.022763982
luse3	0.960718766	1.000000000	0.957222343	0.912036957	0.933958281	0.141977380	-0.092469515	0.022765654
luse4	0.893925078	0.957222343	1.000000000	0.968279803	0.924061938	0.134832712	-0.090883650	0.021595830
luse5	0.858338552	0.912036957	0.968279803	1.000000000	0.921449334	0.133062954	-0.087174977	0.023860249
luse6	0.919606310	0.933958281	0.924061938	0.921449334	1.000000000	0.146260348	-0.089339421	0.024577137
children	0.138596309	0.141977380	0.134832712	0.133062954	0.146260348	1.000000000	-0.289351449	0.144139266
hhsize2	-0.086571072	-0.092469515	-0.090883650	-0.087174977	-0.089339421	-0.289351449	1.000000000	-0.380460209
hhsize3	0.022763982	0.022765654	0.021595830	0.023860249	0.024577137	0.144139266	-0.380460209	1.000000000
hhsize4	0.101996977	0.106110166	0.107669248	0.107739417	0.103452176	0.175172970	-0.290404667	-0.242520923
hhsize5	0.101166095	0.108174337	0.108488947	0.107612124	0.105371283	0.203103895	-0.207399115	-0.173201848
hhsize5plus	0.121004867	0.126180064	0.123275981	0.121184162	0.125353164	0.212063821	-0.163589774	-0.136616066
income2	-0.084837084	-0.092188327	-0.098045431	-0.101414404	-0.095850181	-0.018789513	0.011733869	-0.014515395
income3	-0.086246938	-0.094429903	-0.094313196	-0.089800380	-0.092166818	-0.015927241	0.038409399	0.015461103
income4	-0.081557958	-0.084733318	-0.084397186	-0.081906358	-0.084269974	-0.061163484	0.037289538	0.013575366
income5	-0.031928982	-0.032826580	-0.027521152	-0.025346229	-0.029657201	-0.037378961	0.022669345	0.008587757
income6	0.053423730	0.060403484	0.064853904	0.067087210	0.051424318	0.014197033	-0.012019788	0.001551428
income7	0.134460099	0.144076285	0.149837941	0.148640733	0.138912552	0.030407626	-0.037780372	0.003544098
income8	0.102475674	0.111650410	0.110413470	0.107137213	0.113112311	0.013367478	-0.022656747	-0.006360487
income9	0.098461384	0.104924931	0.104264769	0.100122675	0.109588355	0.013334571	-0.024246833	-0.007884975
owner	0.078854653	0.090512436	0.105285988	0.108020427	0.086068545	-0.040921134	-0.042484904	0.017137123

	income7	income8	income9	owner
hh_id	-0.006442742	-0.006452090	-0.006304739	-0.002858828
year	0.000000000	0.000000000	0.000000000	0.000000000
month	0.000000000	0.000000000	0.000000000	0.000000000
zipcode	-0.060101328	-0.087896514	-0.047857471	-0.113294300
control	-0.005999482	-0.006862261	-0.004687098	-0.028629705
treatment	0.005999482	0.006862261	0.004687098	0.028629705
lusage	0.129631007	0.096067223	0.092558149	0.077638038
luse1	0.122139556	0.088604309	0.086135343	0.073671353
luse2	0.134460099	0.102475674	0.098461384	0.078854653
luse3	0.144076285	0.111650410	0.104924931	0.090512436
luse4	0.149837941	0.110413470	0.104264769	0.105285988
luse5	0.148640733	0.107137213	0.100122675	0.108020427
luse6	0.138912552	0.113112311	0.109588355	0.086068545
children	0.030407626	0.013367478	0.013334571	-0.040921134
hhsize2	-0.037780372	-0.022656747	-0.024246833	-0.042484904
hhsize3	0.003544098	-0.006360487	-0.007884975	0.017137123
hhsize4	0.057082934	0.038448118	0.025991957	0.065123653
hhsize5	0.048341126	0.038337357	0.039958983	0.067703863
hhsize5plus	0.062030973	0.046480329	0.051356385	0.084042468
income2	-0.092892212	-0.056374166	-0.041789809	-0.106297590
income3	-0.127018763	-0.077084792	-0.057142464	-0.151042407
income4	-0.141459813	-0.085848736	-0.063639120	-0.049243026
income5	-0.148763648	-0.090281267	-0.066924926	0.029396729
income6	-0.261254590	-0.158549456	-0.117531698	0.200801264
income7	1.000000000	-0.095543997	-0.070826155	0.157548228
income8	-0.095543997	1.000000000	-0.042982779	0.090331224
income9	-0.070826155	-0.042982779	1.000000000	0.072043401
owner	0.157548228	0.090331224	0.072043401	1.000000000

- Next, I split the data into 5 different groups according to month. We tried the data in April first.

```
# Split the data into 5 different groups according to month.
```{r}
data_4 <- subset (data2, month == 4)
data_5 <- subset (data2, month == 5)
data_6<- subset (data2, month == 6)
data_7 <- subset (data2, month == 7)
data_8 <- subset (data2, month == 8)
```

Use April first.(I tried to use the entire control group of 2010 to predict at the very end of the code (starts line 263 and we can get the same result).)
```{r}
# Split the data set
library(glmnet)
data_c4 <- subset(data_4, control == 1)
data_c5 <- subset(data_5, control == 1)
data_c6 <- subset(data_6, control == 1)
data_c7 <- subset(data_7, control == 1)
data_c8 <- subset(data_8, control == 1)
x <- model.matrix(lusage ~ .-year-month, data = data_c4)
y <- data_c4$lusage


set.seed(1)
train0 <- subset(data_c4, year == 2010)
test0 <- subset(data_c4, year == 2011)
train <- which(train0$year == 2010)
test <- which(test0$year == 2011)
y.test <- y[test]
```
```

Linear Regression Model and Cross validation

- We can get the MSE = 0.07826786

```
#Cross validation using linear regression model, the result shows the mse is smaller than lasso.
```{r}
lm.fit <- lm(lusage ~ .-year-month-control-treatment, data = data_c4, subset = train)
attach(data_c4)
mean((lusage - predict(lm.fit, data_c4))[test]^2)
```
```

```
The following objects are masked from data_use (pos = 5):

    children, control, hh_id, hhsize2, hhsize3, hhsize4, hhsize5, hhsize5plus, income2, income3, income4, income5,
    income6, income7, income8, income9, lusage, luse1, luse2, luse3, luse4, luse5, luse6, month, owner, treatment,
    year, zipcode

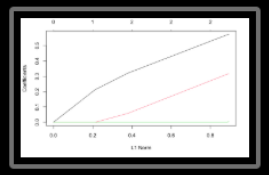
The following objects are masked from data_use (pos = 6):

    children, control, hh_id, hhsize2, hhsize3, hhsize4, hhsize5, hhsize5plus, income2, income3, income4, income5,
    income6, income7, income8, income9, lusage, luse1, luse2, luse3, luse4, luse5, luse6, month, owner, treatment,
    year, zipcode

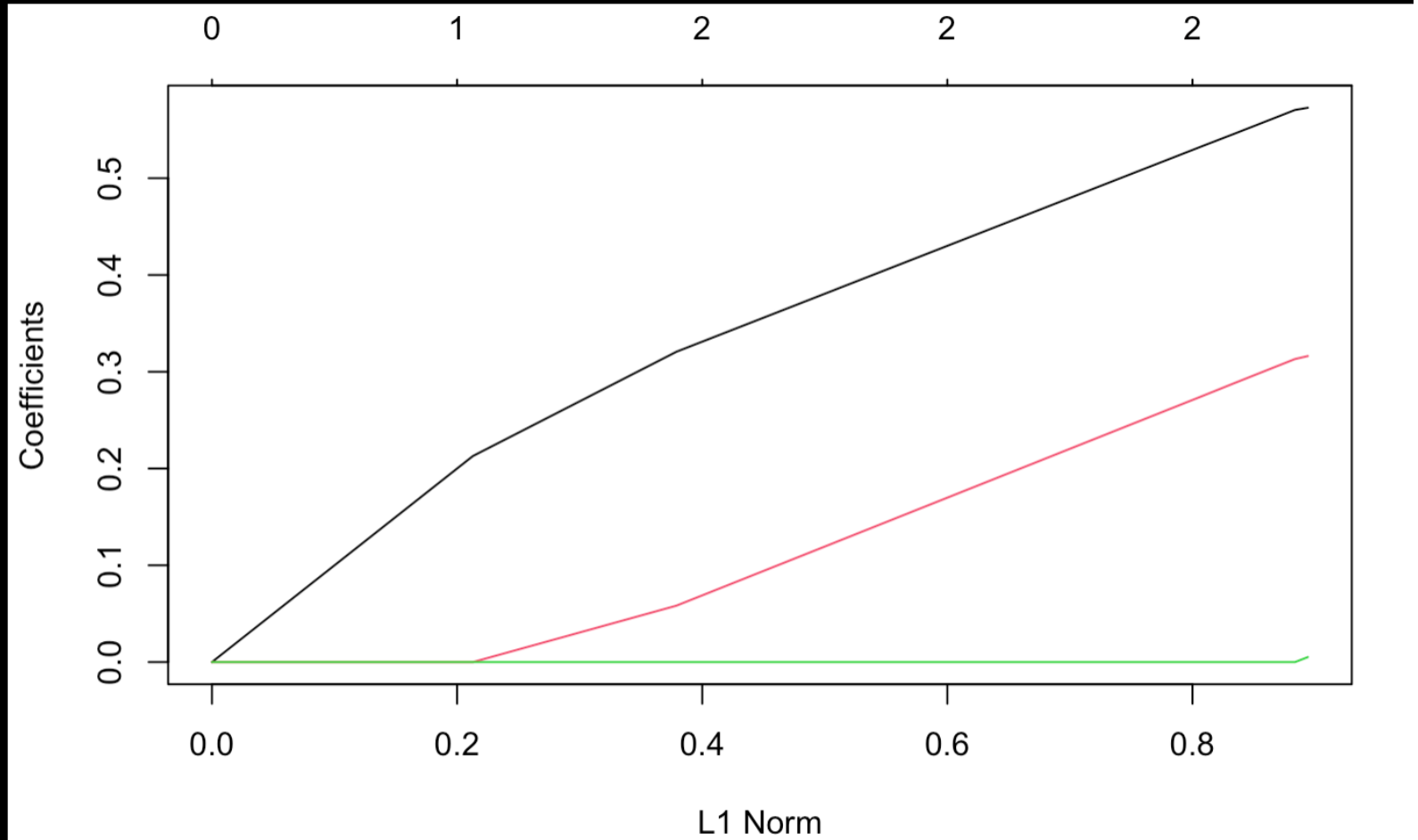
[1] 0.07826786
```

Lasso

```{r}
# Lasso
grid <- 10^seq(10, -2, length = 100)
lasso.mod <- glmnet(x[train, ], y[train], alpha = 1,
    lambda = grid, standardize = TRUE)
plot(lasso.mod)
```



R Console



Lasso

```{r}
bestlam <- cv.out$lambda.min
lasso.pred <- predict(lasso.mod, s = bestlam,
    newx = x[test, ])
mean((lasso.pred - y[test])^2)
```

```
[1] 0.07992893
```

```{r}
out <- glmnet(x, y, alpha = 1, lambda = grid)
lasso.coef <- predict(out, type = "coefficients",
    s = bestlam)[1:27, ]
lasso.coef

```

```
 (Intercept) (Intercept)       hh_id     zipcode     control   treatment       luse1       luse2       luse3       luse4
 0.658256858 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.574710673 0.000000000 0.000000000 0.000000000
       luse5       luse6    children     hhsize2     hhsize3     hhsize4     hhsize5 hhsize5plus     income2     income3
 0.000000000 0.313683386 0.004933622 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
     income4     income5     income6     income7     income8     income9       owner
 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
```

```{r}
lasso.coef[lasso.coef != 0]
```

```
 (Intercept)       luse1       luse6    children
 0.658256858 0.574710673 0.313683386 0.004933622
```

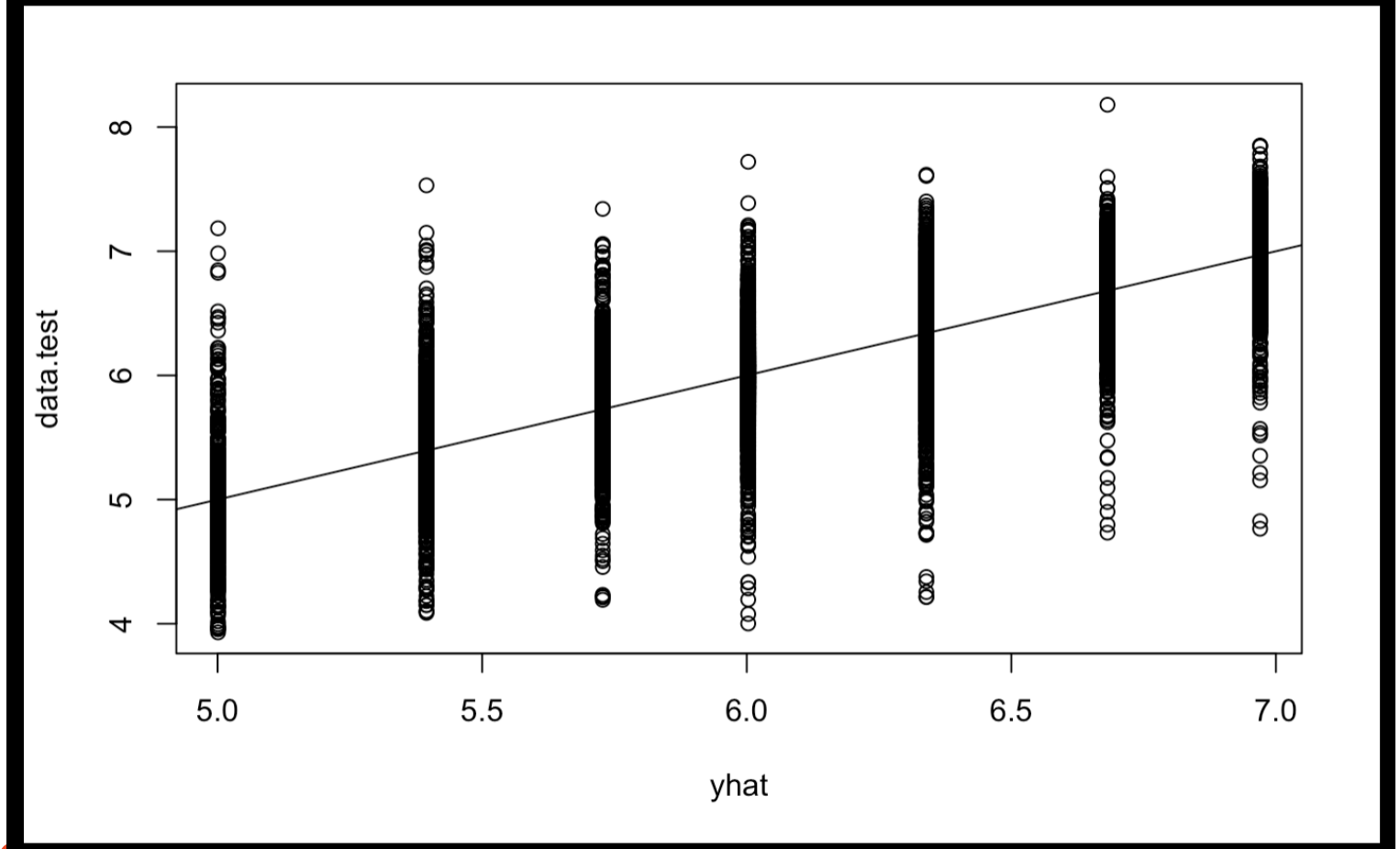
- We can get the MSE = 0.07992893, however, we only get 3 useful variables.

Decision Tree

```
# We now build a decision tree
```{r}
library(tree)
tree <- tree(lusage ~ .-year-month-control-treatment, data_c4, subset = train)
summary(tree)
```
```

```
Regression tree:
tree(formula = lusage ~ . - year - month - control - treatment,
    data = data_c4, subset = train)
Variables actually used in tree construction:
[1] "luse1" "luse6"
Number of terminal nodes:  7
Residual mean deviance:  0.09088 = 1601 / 17620
Distribution of residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.205000 -0.148500  0.005755  0.000000  0.157400  2.184000
```

```{r}
# Cross Validation for tree
yhat <- predict(tree, newdata = data_c4[test, ])
data.test <- data_c4[test, "lusage"]
plot(yhat, data.test)
abline(0, 1)
```




```{r}
mean((yhat - data.test)^2)
```

```
[1] 0.09084836
```

- We can get MSE = 0.09084836

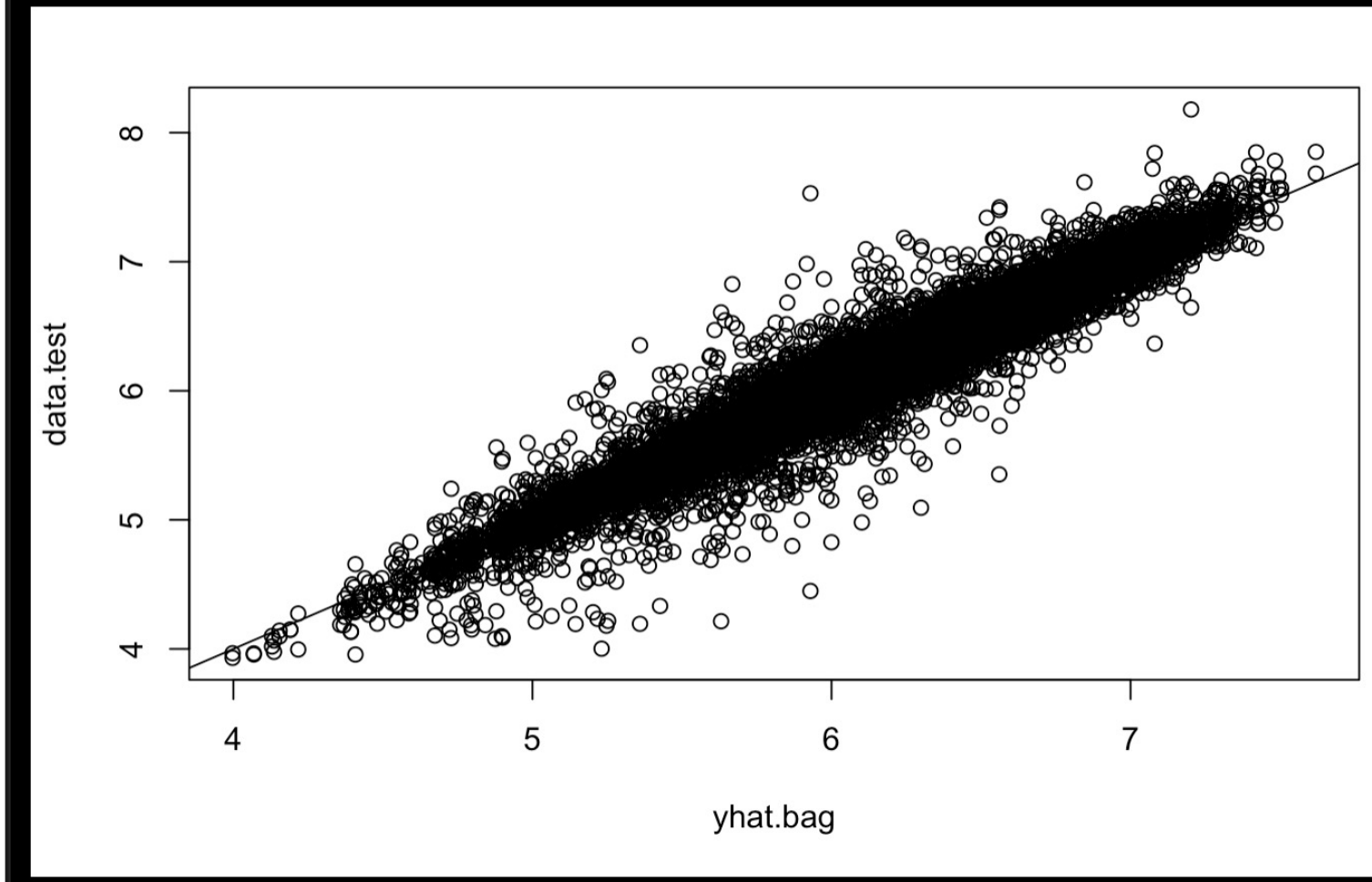
Bagging and Random Forest

```
# Bagging and Random Forest
```{r}
library(randomForest)
set.seed(1)
bag.data <- randomForest(lusage ~ .-year-month-control-treatment, data_c4, subset = train, mtry = 12, importance = TRUE)
bag.data
```
```

```
Call:
 randomForest(formula = lusage ~ . - year - month - control -      treatment, data = data_c4, mtry = 12, importance = TRUE,
subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 12

          Mean of squared residuals: 0.06089205
                    % Var explained: 82.11
```

```{r}
yhat.bag <- predict(bag.data, newdata = data_c4[test, ])
plot(yhat.bag, data.test)
abline(0, 1)
```




```{r}
mean((yhat.bag - data.test)^2)
```

```
[1] 0.02235202
```

- We can get bagging MSE = 0.02235202

```{r}
set.seed(1)
rf.data <- randomForest(lusage ~ .-year-month-control-treatment, data = data_c4,
    subset = train, mtry = 6, importance = TRUE)
yhat.rf <- predict(rf.data, newdata = data_c4[test, ])
mean((yhat.rf - data.test)^2)

```

```
[1] 0.02266756
```

```{r}
importance(rf.data)
```

```
             %IncMSE IncNodePurity
hh_id       84.36335    107.421538
zipcode     64.63504     64.968425
luse1       92.34622   1825.639388
luse2       38.71256   1172.005633
luse3       38.32896    651.622409
luse4       40.45555    435.034959
luse5       52.37938    344.366879
luse6       51.22198    976.589052
children    23.31269     19.097143
hhsize2     25.39623     14.379412
hhsize3     27.56352     12.947605
hhsize4     18.06060     10.461785
hhsize5     22.18583      7.707457
hhsize5plus 13.81591      8.188039
income2     16.97661      9.811333
income3     23.39429      9.418543
income4     21.43231     11.789113
income5     27.31311      9.625733
income6     29.59246     13.612070
income7     16.63371     11.260819
income8     14.43732      7.092076
income9     15.32500      3.909833
owner       28.10200     14.025891
```

- MSE of random forest = 0.02266756 and the importance of variables are showed.

- After comparing the MSE, we can find the MSE of bagging is the smallest, so we apply this method to the control groups for every month to build 5 models.

```{r}
# May
set.seed(1)
bag.data5 <- randomForest(lusage ~ .-year-month-control-treatment, data_c5, subset = train, mtry = 12, importance = TRUE)
bag.data5
```

```
Call:
 randomForest(formula = lusage ~ . - year - month - control -      treatment, data = data_c5, mtry = 12, importance = TRUE,
subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 12

          Mean of squared residuals: 0.04801195
                    % Var explained: 85.57
```

```{r}
# June
set.seed(1)
bag.data6 <- randomForest(lusage ~ .-year-month-control-treatment, data_c6, subset = train, mtry = 12, importance = TRUE)
bag.data6
```

```
Call:
 randomForest(formula = lusage ~ . - year - month - control -      treatment, data = data_c6, mtry = 12, importance = TRUE,
subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 12

          Mean of squared residuals: 0.05629963
                    % Var explained: 84.98
```

```{r}
# July
set.seed(1)
bag.data7 <- randomForest(lusage ~ .-year-month-control-treatment, data_c7, subset = train, mtry = 12, importance = TRUE)
bag.data7
```

```
Call:
 randomForest(formula = lusage ~ . - year - month - control -      treatment, data = data_c7, mtry = 12, importance = TRUE,
subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 12

          Mean of squared residuals: 0.0684023
                    % Var explained: 82.81
```

```{r}
# August
set.seed(1)
bag.data8 <- randomForest(lusage ~ .-year-month-control-treatment, data_c8, subset = train, mtry = 12, importance = TRUE)
bag.data8
```

```
Call:
 randomForest(formula = lusage ~ . - year - month - control -      treatment, data = data_c8, mtry = 12, importance = TRUE,
subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 12

          Mean of squared residuals: 0.05802782
                    % Var explained: 84.87
```

- If we use the whole control group of 2010 instead of just the April data and compare the MSE, we can still get the same result that bagging works better than others because it has the smallest MSE.
- Linear Regression: MSE = 0.127022
- Lasso: MSE = 0.1273782
- Decision tree: MSE = 0.1398303
- Bagging: MSE = 0.08195466

Prediction

- Now, we use the model we built to predict the electricity consumption of the households for 2011 in treatment group.

```
# Apply Random Forest to the treatment groups 2011 for every month.
```{r}
data_t4 <- subset(data_4, control == 0)
data_t5 <- subset(data_5, control == 0)
data_t6 <- subset(data_6, control == 0)
data_t7 <- subset(data_7, control == 0)
data_t8 <- subset(data_8, control == 0)

pred04 <- subset(data_t4, year == 2011)
pred4 <- which(pred04$year == 2011)
#y.test <- y[test]
pred05 <- subset(data_t5, year == 2011)
pred5 <- which(pred05$year == 2011)

pred06 <- subset(data_t6, year == 2011)
pred6 <- which(pred06$year == 2011)

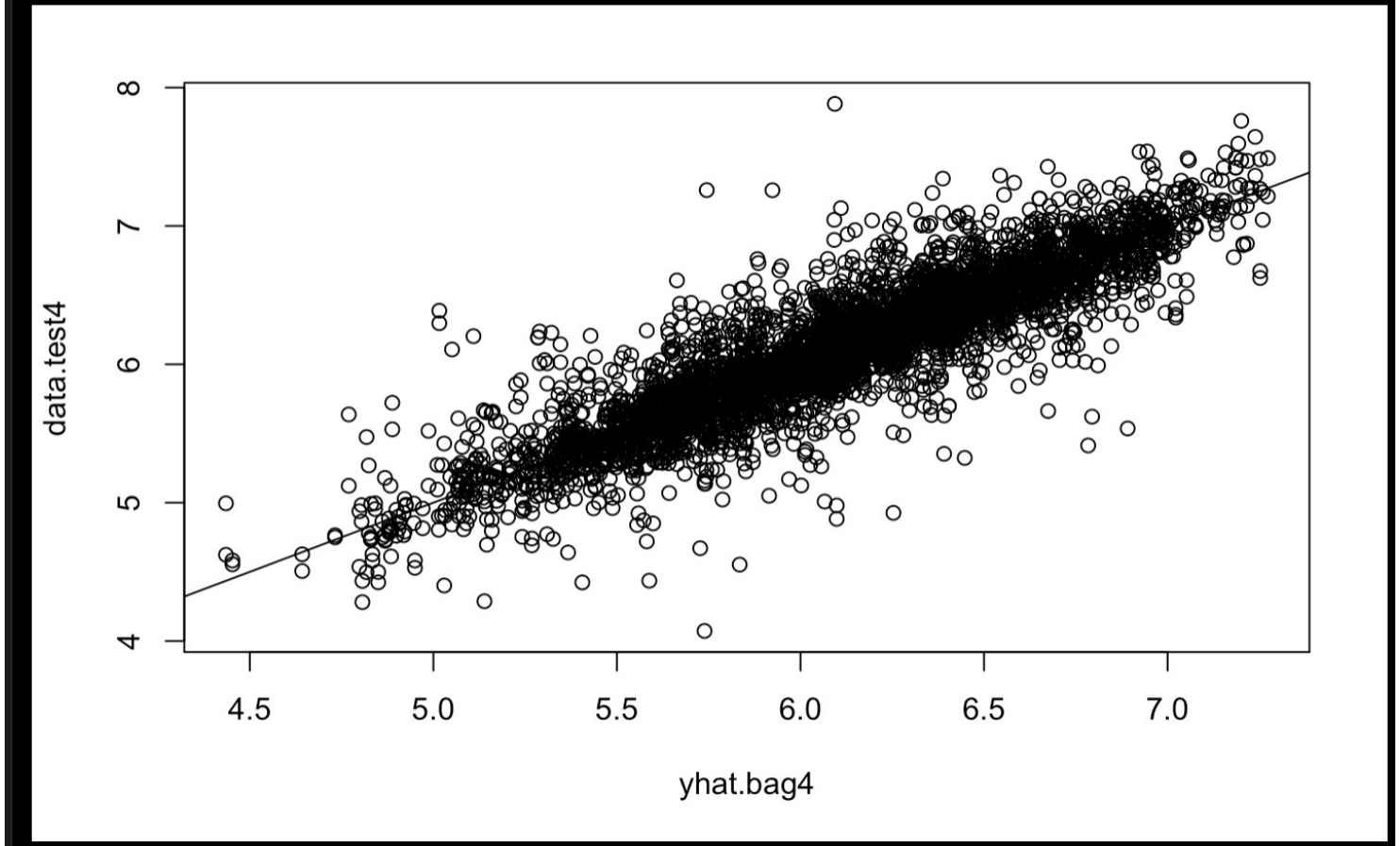
pred07 <- subset(data_t7, year == 2011)
pred7 <- which(pred07$year == 2011)

pred08 <- subset(data_t8, year == 2011)
pred8 <- which(pred08$year == 2011)
```
```

April

- MSE = 0.0607

```{r}
# For April
yhat.bag4 <- predict(bag.data, newdata = data_t4[pred4, ])
data.test4 <- data_t4[pred4, "lusage"]
plot(yhat.bag4, data.test4)
abline(0, 1)
```



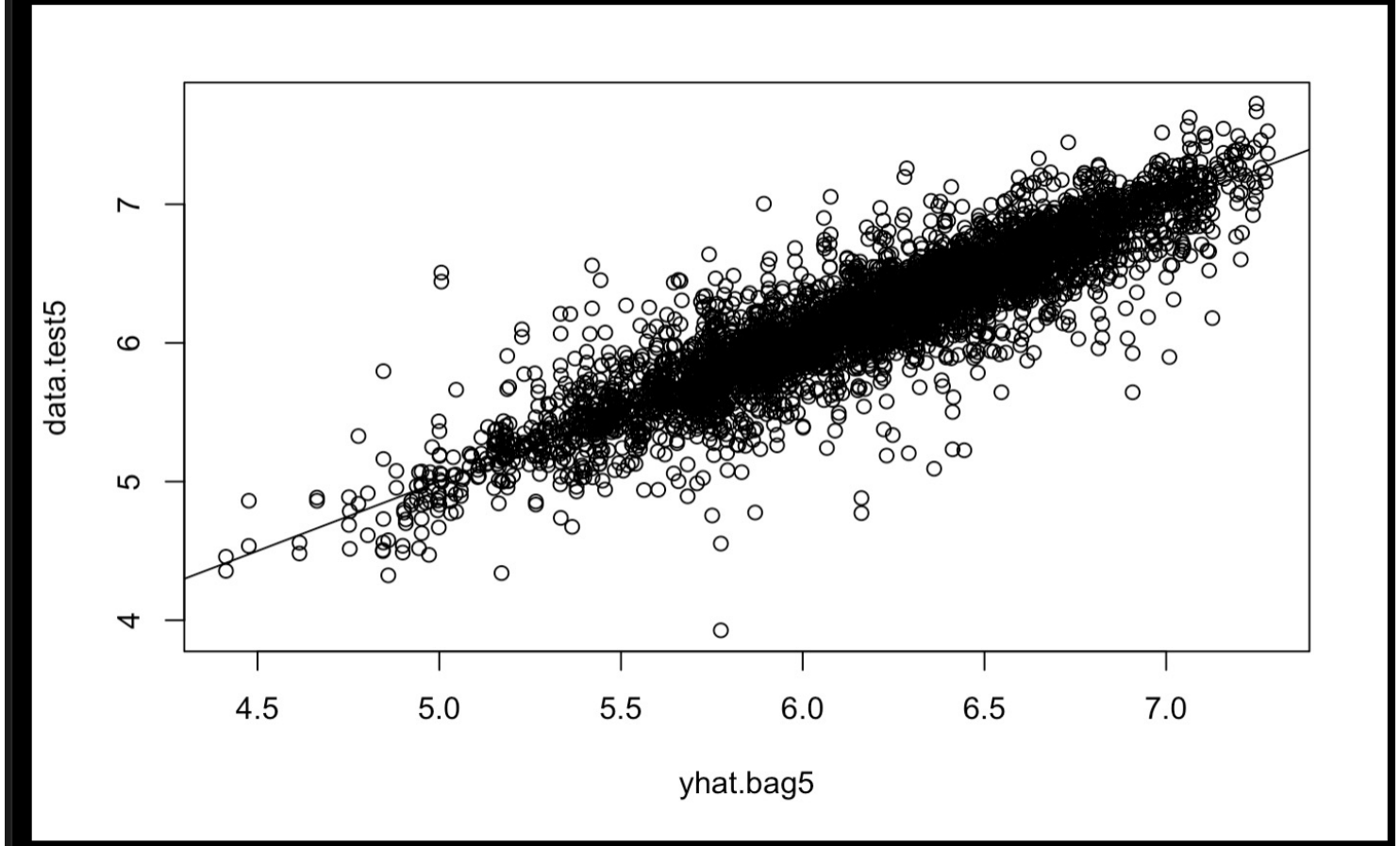

```{r}
mean((yhat.bag4 - data.test4)^2)
```

```
[1] 0.06074779
```

May

- MSE = 0.053501

```
# For May
```{r}
yhat.bag5 <- predict(bag.data5, newdata = data_t5[pred5, ])
data.test5 <- data_t5[pred5, "lusage"]
plot(yhat.bag5, data.test5)
abline(0, 1)
```
```



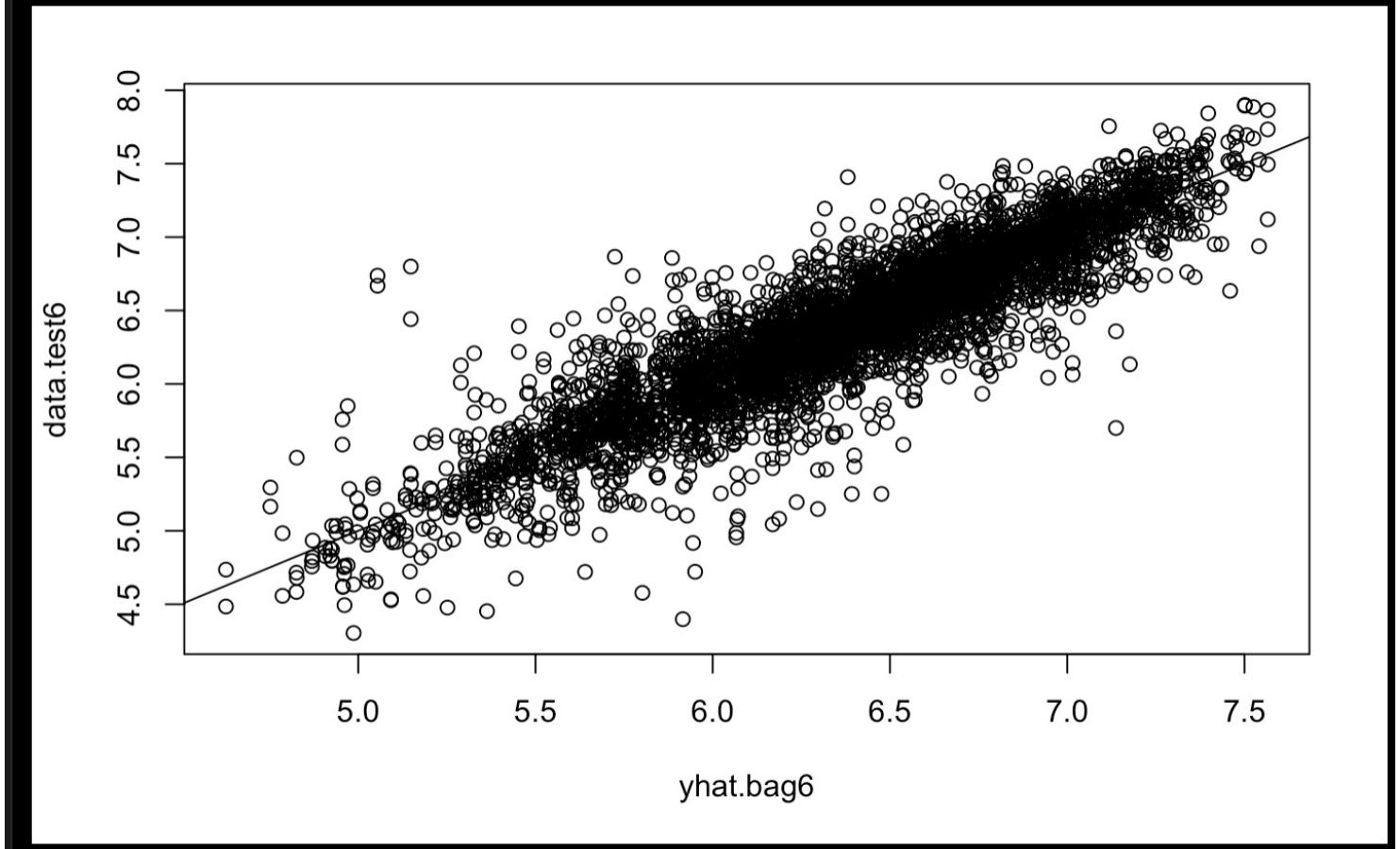

```{r}
mean((yhat.bag5 - data.test5)^2)
```

```
[1] 0.053501
```

June

- MSE = 0.06186554

```
# For June
```{r}
yhat.bag6 <- predict(bag.data6, newdata = data_t6[pred6, ])
data.test6 <- data_t6[pred6, "lusage"]
plot(yhat.bag6, data.test6)
abline(0, 1)
```
```



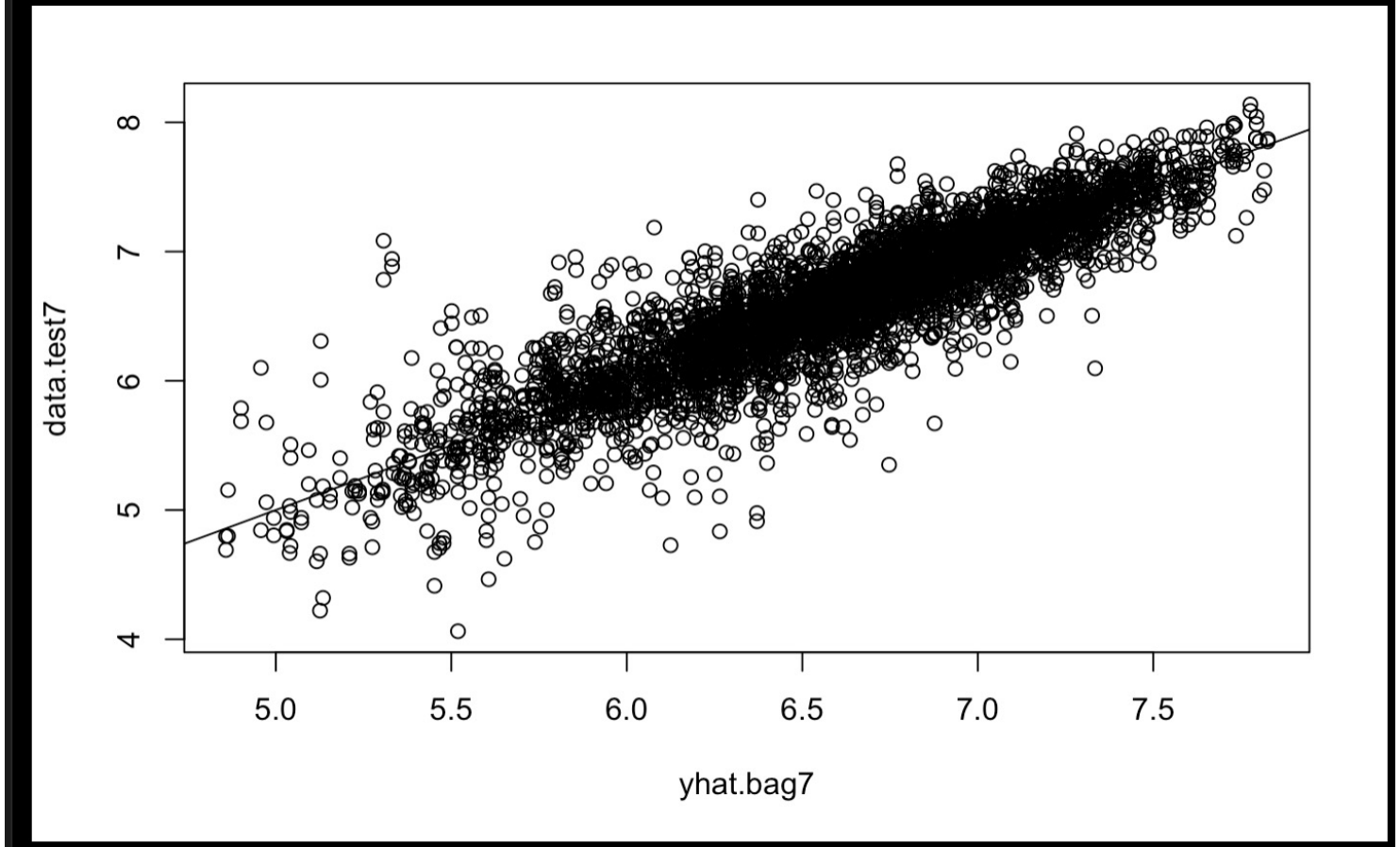

```{r}
mean((yhat.bag6 - data.test6)^2)
```

```
[1] 0.06186554
```

July

- MSE = 0.07299933

```
# For July
```{r}
yhat.bag7 <- predict(bag.data7, newdata = data_t7[pred7, ])
data.test7 <- data_t7[pred7, "lusage"]
plot(yhat.bag7, data.test7)
abline(0, 1)
```
```



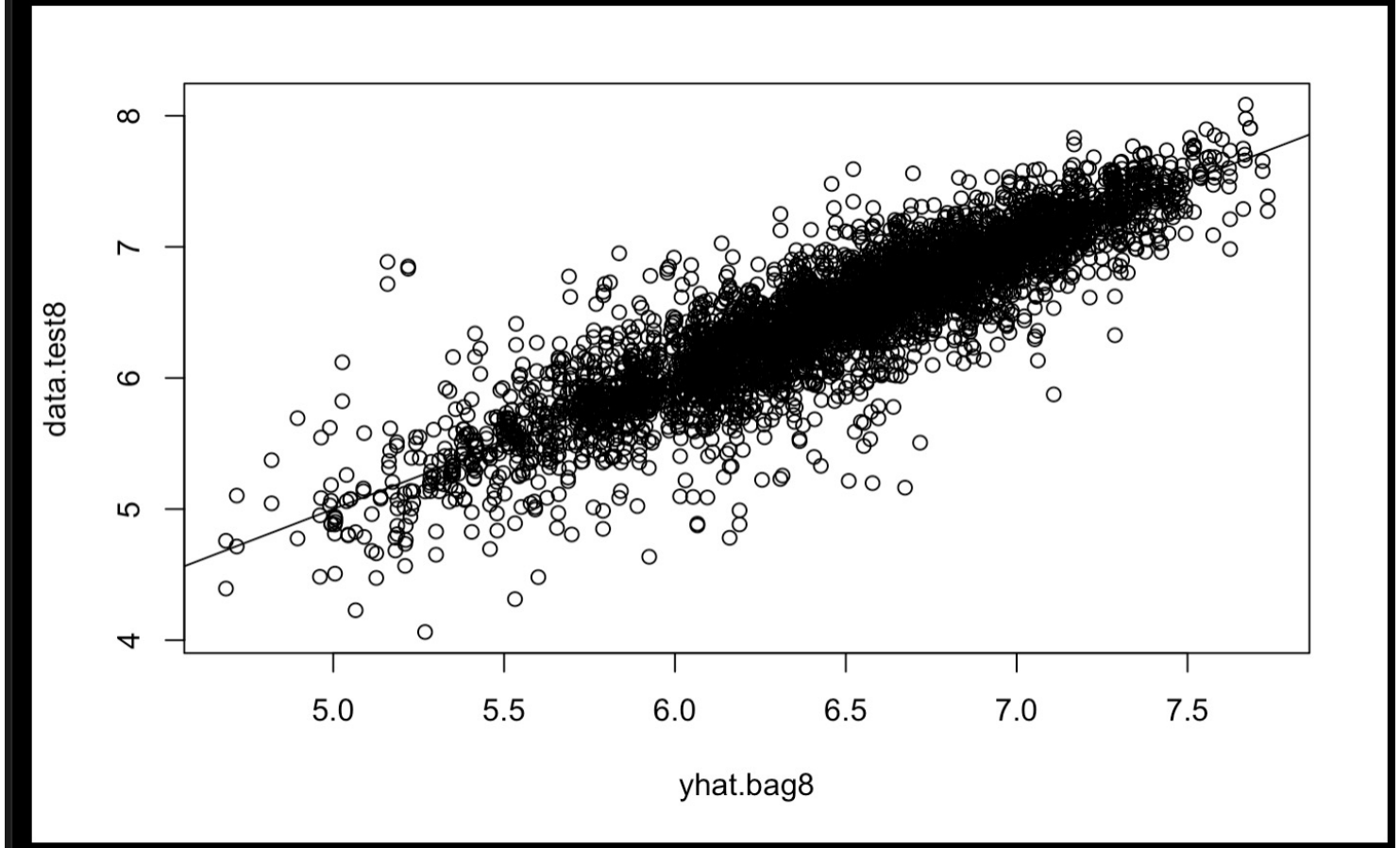

```{r}
mean((yhat.bag7 - data.test7)^2)
```

```
[1] 0.07299933
```

August

- MSE = 0.07107559

```{r}
# For August
yhat.bag8 <- predict(bag.data8, newdata = data_t8[pred8, ])
data.test8 <- data_t8[pred8, "lusage"]
plot(yhat.bag8, data.test8)
abline(0, 1)
```




```{r}
mean((yhat.bag8 - data.test8)^2)
```

```
[1] 0.07107559
```

Evaluation & Extra Credit

- Compare the predicted values for 2011 with the actual values for 2011, we can see the MSE is relatively small, which means our model works well.
- We can see the MSE is the smallest for May, and relatively big for July and August, which means the model can predict the the electricity consumption of the households in May 2011 better than August 2011.
- The same quantity could not have been obtained by comparing the actual valued for 2011 for the treatment group with the values for 2011 for the control group, because the households in the treatment group were encouraged each month in 2011 to conserve energy, and this change could be another variable.