Question 14.1

1. Using the mean/mode imputation method, I determined the mean value to be 3.54465592972182, and the mode value to be 1. These values were then substituted inside of the dataset where the values were initially missing (denoted as “?”). Each of these datasets were then compared to the original dataset in Figure 2, of course with the missing values omitted. The figure caption goes into subsequent detail of the graph, however I did want to note that the mean imputed values will be used for the next portions of the homework assignment as a baseline for regression.

2. Figures 4 and 5 were the culmination of the code written in Figure 3. I attempted to delve a little deeper into the many ways of imputation for the missing values, and this was done in tandem with the R library MICE. This library serves as the medium to which the regressions/statistical methods were performed on the dataset. I will format this section in a way that reflects the ordering of the graph in Figure 4.
    a. Original Data: This portion of the graph serves as the control group for the other computed distributions in Figure 4.
    b. Linear Regression: This distribution utilized linear regression analysis without accounting for the uncertainty of the model parameters to determine the missing values. I believe this to be a poor distribution of the expected missing value substitutions due to the presence of negative numbers (not present in the original dataset), and the significant difference between it and the control group. However, in this method I did not account for possible overfitting or significance of predictors in the dataset, which means that there is still potential for this to be a better method.
    c. CART (Classification and Regression Trees): Given how we recently learned about CART analysis, I wanted to see how it would work with missing value substitution. Out of all these distributions, I found the CART analysis to be the most well balanced out of all of the regression models. Where other distributions have negative numbers or are skewed towards a certain value, the CART distribution found a nice balance that can be attributed towards its inherent ability to find patterns and correlations between predictors.

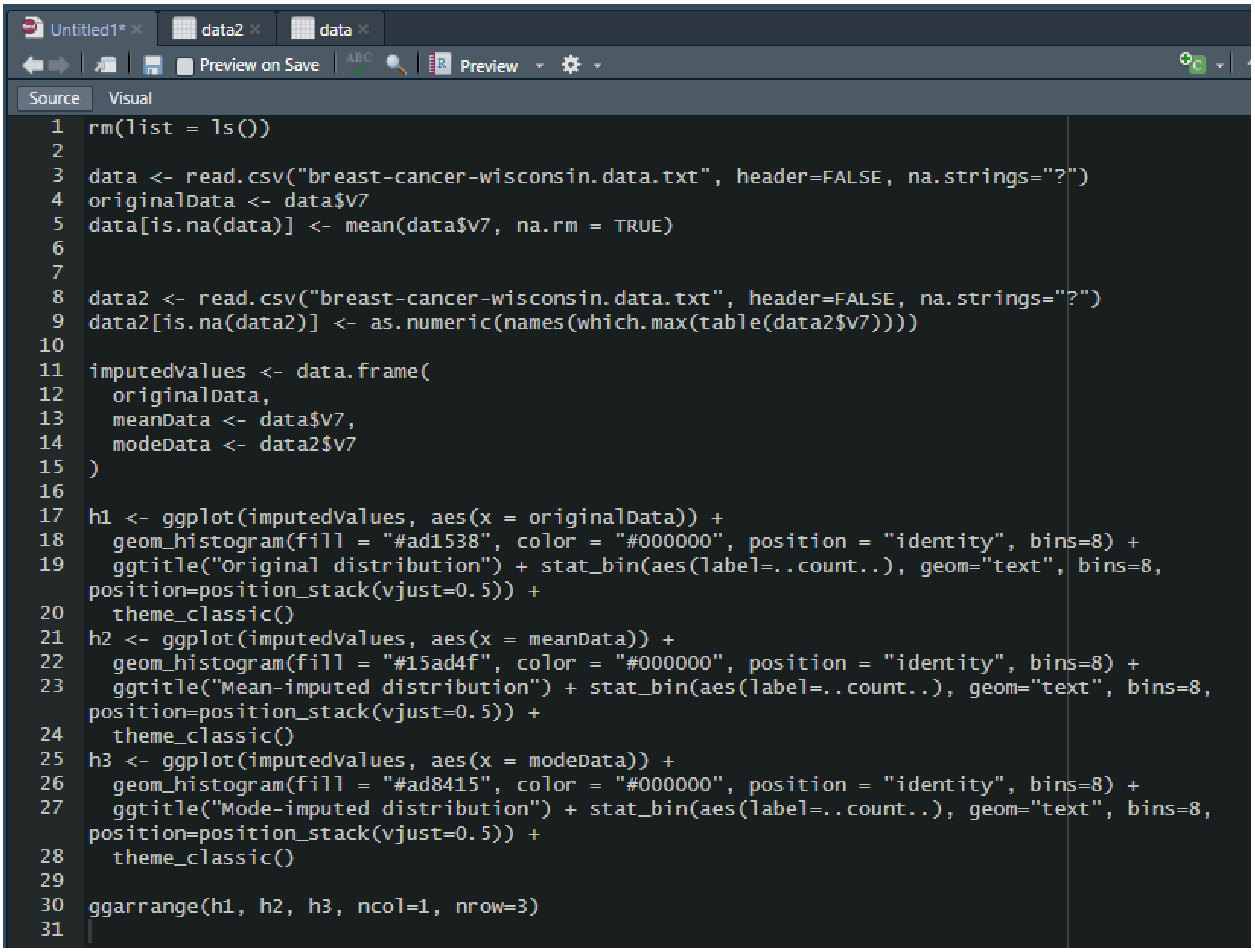
d. LASSO: Based on all of these distributions, I believe the LASSO method to be the worst suited one for substituting missing values. There is a clear bias in values following the zero and preceding the eight value on the x-axis that is not present in other distributions. Because of this I would not recommend the LASSO model for data imputation.

e. PMM (Predictive Mean Matching): This final method is not a regression model at all, however I wanted to include it in this section because it is a very widely used statistical method for computing imputation of missing values. This method achieved very similar results to the CART regression model, and retained the distribution pattern found in the control group. This method also has the bonus of accounting for bias within the data, which supports its ability to predict values.

3. Regression with perturbation was used to impute values for the missing data as shown in Figures 6 and 7. Notably, some of these values ended up being negative which does not follow any point in the original dataset (all values in V7 are greater than 0). With that being said, one way to potentially make this model better would be to select the most significant predictors and rerun the linear regression model. In addition, you could also round the values to the nearest non-negative integer to eliminate the negative values, but I would rather do the following than do that procedure. I would strongly recommend using the other regressions outlined in the previous part of this question before trying to use the linear regression model.

Question 15.1

I have found recently that my time management skill has been getting worse and worse. Therefore, I believe that optimizing my schedule based on time spent doing productive tasks (job hunting, homework, exam prep, etc.), pursuing hobbies (sports, games, etc.), and an overall satisfaction score for the day out of 100. Optimizing my time allocation for each of those tasks, whether productive or for hobbies, could help my time management and make me much more effective in my daily life.



```r
rm(list = ls())

data <- read.csv("breast-cancer-wisconsin.data.txt", header=FALSE, na.strings="?")
originalData <- data$V7
data[is.na(data)] <- mean(data$V7, na.rm = TRUE)


data2 <- read.csv("breast-cancer-wisconsin.data.txt", header=FALSE, na.strings="?")
data2[is.na(data2)] <- as.numeric(names(which.max(table(data2$V7))))

imputedValues <- data.frame(
  originalData,
  meanData <- data$V7,
  modeData <- data2$V7
)

h1 <- ggplot(imputedValues, aes(x = originalData)) +
  geom_histogram(fill = "#ad1538", color = "#000000", position = "identity", bins=8) +
  ggtitle("Original distribution") + stat_bin(aes(label=..count..), geom="text", bins=8, position=position_stack(vjust=0.5)) +
  theme_classic()
h2 <- ggplot(imputedValues, aes(x = meanData)) +
  geom_histogram(fill = "#15ad4f", color = "#000000", position = "identity", bins=8) +
  ggtitle("Mean-imputed distribution") + stat_bin(aes(label=..count..), geom="text", bins=8, position=position_stack(vjust=0.5)) +
  theme_classic()
h3 <- ggplot(imputedValues, aes(x = modeData)) +
  geom_histogram(fill = "#ad8415", color = "#000000", position = "identity", bins=8) +
  ggtitle("Mode-imputed distribution") + stat_bin(aes(label=..count..), geom="text", bins=8, position=position_stack(vjust=0.5)) +
  theme_classic()

ggarrange(h1, h2, h3, ncol=1, nrow=3)

```

Figure 1. This figure denotes the code used to run the mean/mode value imputation. Furthermore, there was also a histogram generated of the distributions with respect to the original (note that the missing values in the original distribution are missing from the visualization).

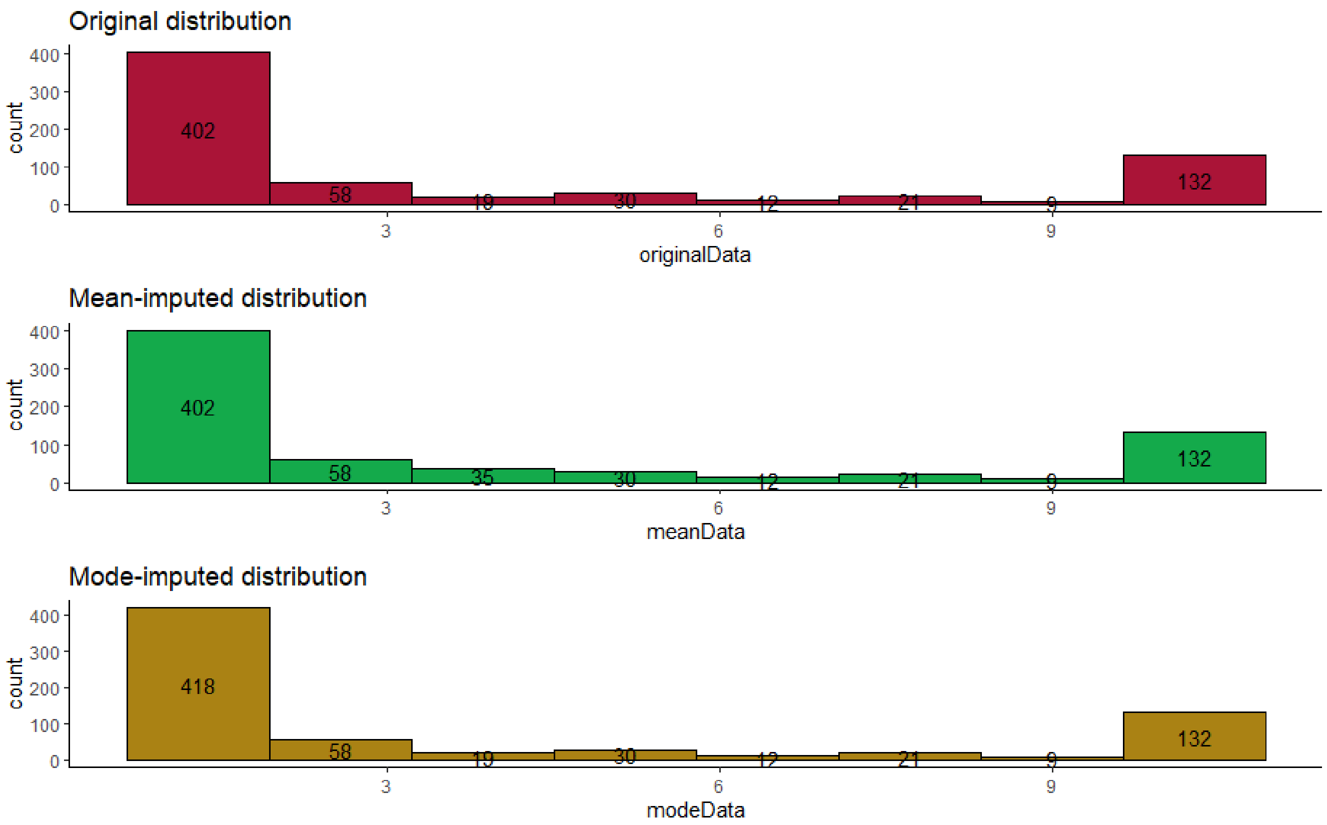
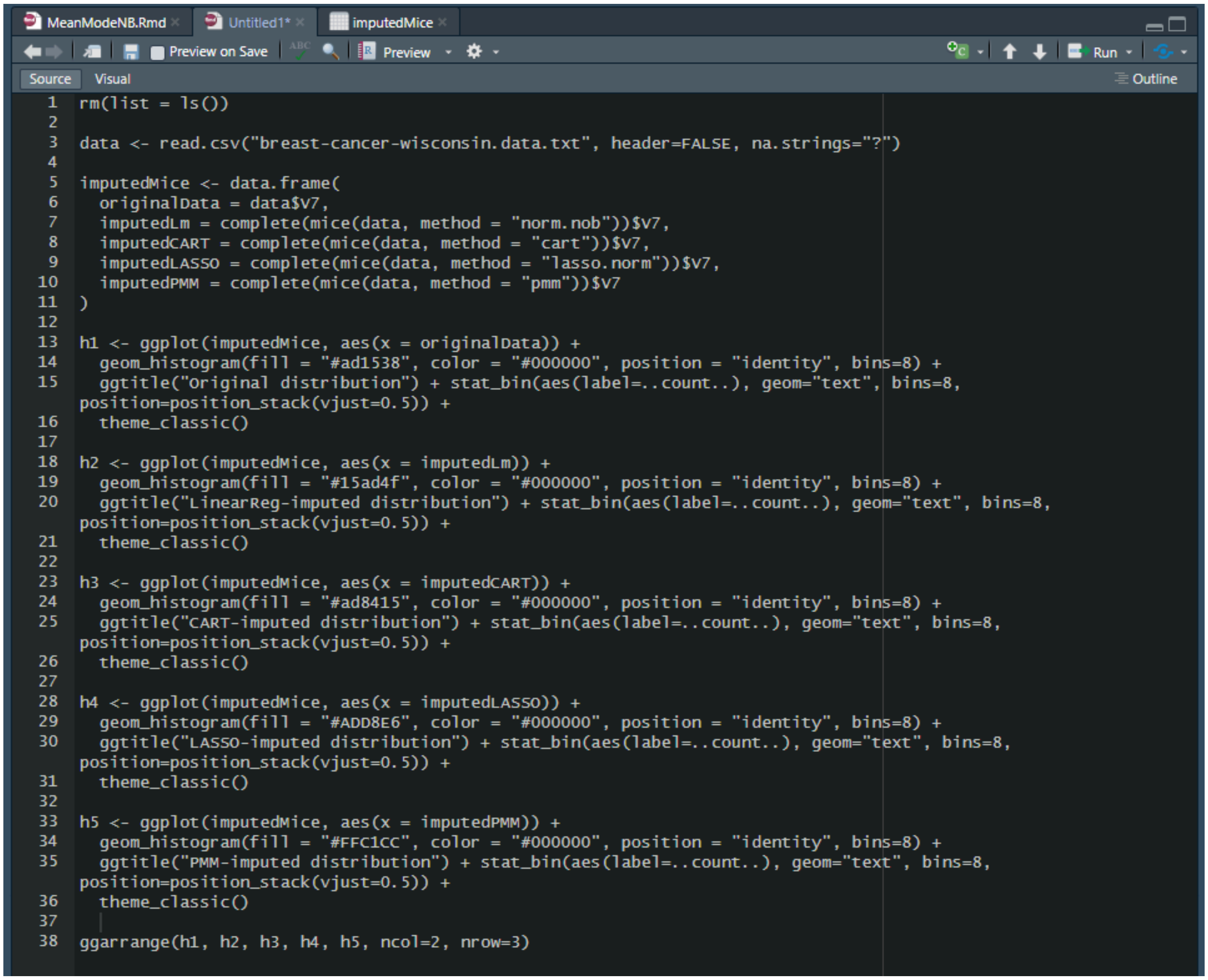
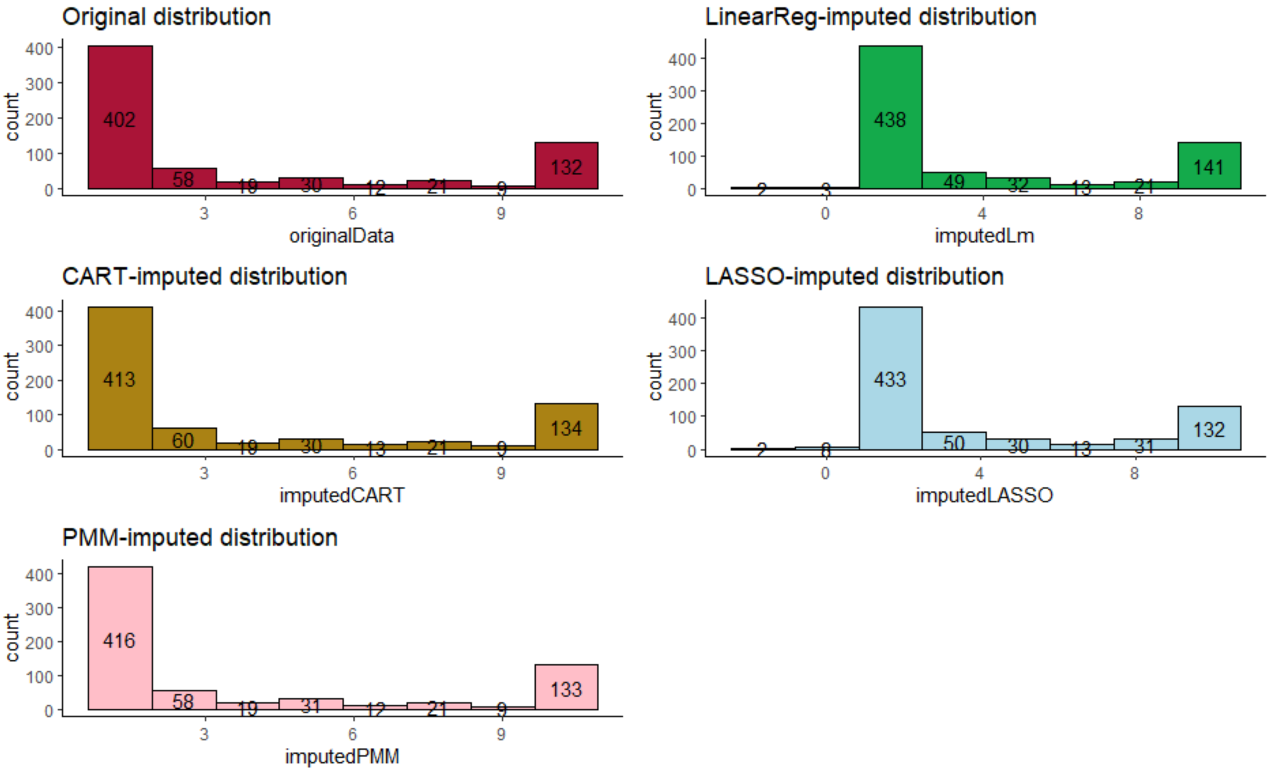


Figure 2. These are the histograms of the original/mean/mode distribution. You can see how the mode has a greater count in the beginning of the distribution, whereas the mean-imputed dataset has a greater magnitude in the section after 3. This is due to the average being calculated as 3.54465592972182 and thus making up more of the dataset. The mode distribution had the same effect but seen in the earlier part of the distribution because of the mode equaling 1.



```r
rm(list = ls())

data <- read.csv("breast-cancer-wisconsin.data.txt", header=FALSE, na.strings="?")

imputedMice <- data.frame(
  originalData = data$V7,
  imputedLm = complete(mice(data, method = "norm.nob"))$V7,
  imputedCART = complete(mice(data, method = "cart"))$V7,
  imputedLASSO = complete(mice(data, method = "lasso.norm"))$V7,
  imputedPMM = complete(mice(data, method = "pmm"))$V7
)

h1 <- ggplot(imputedMice, aes(x = originalData)) +
  geom_histogram(fill = "#ad1538", color = "#000000", position = "identity", bins=8) +
  ggtitle("Original distribution") + stat_bin(aes(label=..count..), geom="text", bins=8,
position=position_stack(vjust=0.5)) +
  theme_classic()

h2 <- ggplot(imputedMice, aes(x = imputedLm)) +
  geom_histogram(fill = "#15ad4f", color = "#000000", position = "identity", bins=8) +
  ggtitle("LinearReg-imputed distribution") + stat_bin(aes(label=..count..), geom="text", bins=8,
position=position_stack(vjust=0.5)) +
  theme_classic()

h3 <- ggplot(imputedMice, aes(x = imputedCART)) +
  geom_histogram(fill = "#ad8415", color = "#000000", position = "identity", bins=8) +
  ggtitle("CART-imputed distribution") + stat_bin(aes(label=..count..), geom="text", bins=8,
position=position_stack(vjust=0.5)) +
  theme_classic()

h4 <- ggplot(imputedMice, aes(x = imputedLASSO)) +
  geom_histogram(fill = "#ADD8E6", color = "#000000", position = "identity", bins=8) +
  ggtitle("LASSO-imputed distribution") + stat_bin(aes(label=..count..), geom="text", bins=8,
position=position_stack(vjust=0.5)) +
  theme_classic()

h5 <- ggplot(imputedMice, aes(x = imputedPMM)) +
  geom_histogram(fill = "#FFC1CC", color = "#000000", position = "identity", bins=8) +
  ggtitle("PMM-imputed distribution") + stat_bin(aes(label=..count..), geom="text", bins=8,
position=position_stack(vjust=0.5)) +
  theme_classic()

ggarrange(h1, h2, h3, h4, h5, ncol=2, nrow=3)
```

Figure 3. The above code utilized the mice library to show how various regression models and a statistical method of predictive mean matching (PMM) was used to impute missing values in the given dataset.


Original distribution

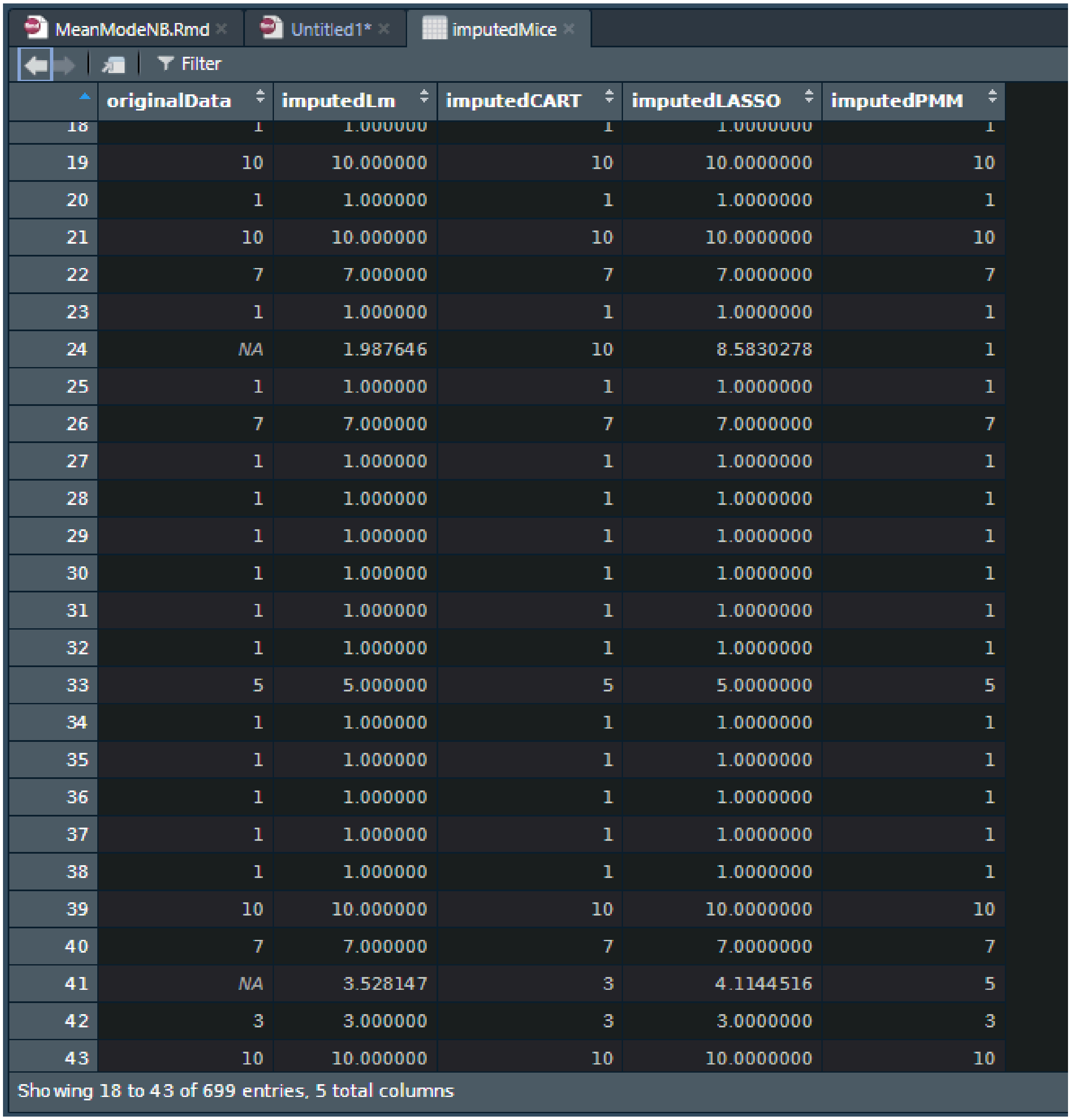
LinearReg-imputed distribution

CART-imputed distribution

LASSO-imputed distribution

PMM-imputed distribution

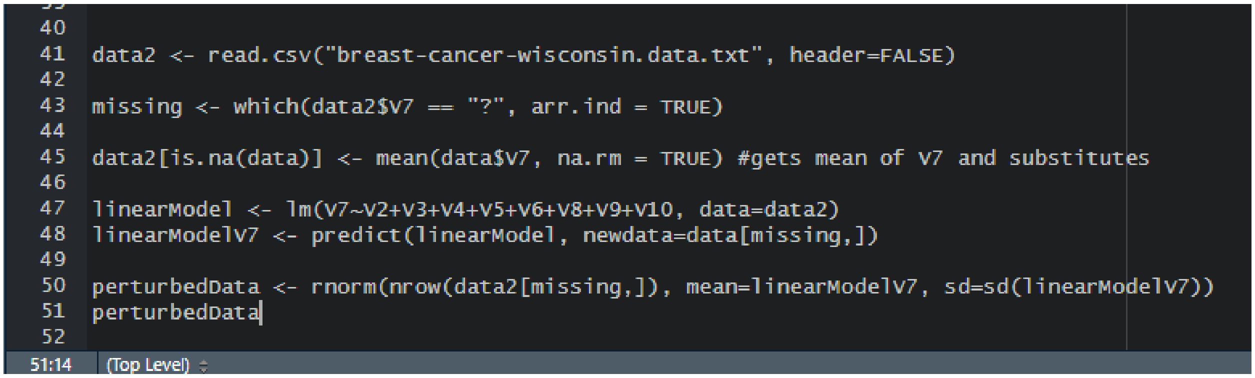
Figure 4. This figure is a collection of the distributions visualized by histograms to see the change with respect to the other distributions.



	originalData	imputedLm	imputedCART	imputedLASSO	imputedPMM
18	1	1.000000	1	1.0000000	1
19	10	10.000000	10	10.0000000	10
20	1	1.000000	1	1.0000000	1
21	10	10.000000	10	10.0000000	10
22	7	7.000000	7	7.0000000	7
23	1	1.000000	1	1.0000000	1
24	*NA*	1.987646	10	8.5830278	1
25	1	1.000000	1	1.0000000	1
26	7	7.000000	7	7.0000000	7
27	1	1.000000	1	1.0000000	1
28	1	1.000000	1	1.0000000	1
29	1	1.000000	1	1.0000000	1
30	1	1.000000	1	1.0000000	1
31	1	1.000000	1	1.0000000	1
32	1	1.000000	1	1.0000000	1
33	5	5.000000	5	5.0000000	5
34	1	1.000000	1	1.0000000	1
35	1	1.000000	1	1.0000000	1
36	1	1.000000	1	1.0000000	1
37	1	1.000000	1	1.0000000	1
38	1	1.000000	1	1.0000000	1
39	10	10.000000	10	10.0000000	10
40	7	7.000000	7	7.0000000	7
41	*NA*	3.528147	3	4.1144516	5
42	3	3.000000	3	3.0000000	3
43	10	10.000000	10	10.0000000	10

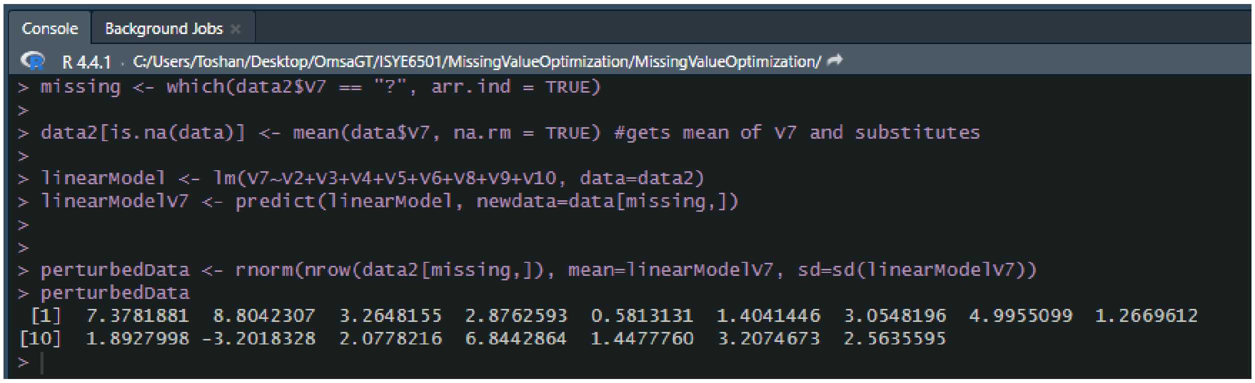
Showing 18 to 43 of 699 entries, 5 total columns

Figure 5. This is a small subset of the imputed mice dataset that shows how some of the *NA* values were substituted with numbers from the respective regression or statistical method.



```
data2 <- read.csv("breast-cancer-wisconsin.data.txt", header=FALSE)

missing <- which(data2$V7 == "?", arr.ind = TRUE)

data2[is.na(data)] <- mean(data$V7, na.rm = TRUE) #gets mean of V7 and substitutes

linearModel <- lm(V7~V2+V3+V4+V5+V6+V8+V9+V10, data=data2)
linearModelV7 <- predict(linearModel, newdata=data[missing,])

perturbedData <- rnorm(nrow(data2[missing,]), mean=linearModelV7, sd=sd(linearModelV7))
perturbedData
```

Figure 6. This is the code used to find the imputed values via perturbation with linear regression. These values were calculated using the predict function of the linear regression model.



Console
Background Jobs
R 4.4.1 · C:/Users/Toshan/Desktop/OmsaGT/ISYE6501/MissingValueOptimization/MissingValueOptimization/

```
> missing <- which(data2$V7 == "?", arr.ind = TRUE)
>
> data2[is.na(data)] <- mean(data$V7, na.rm = TRUE) #gets mean of V7 and substitutes
>
> linearModel <- lm(V7~V2+V3+V4+V5+V6+V8+V9+V10, data=data2)
> linearModelV7 <- predict(linearModel, newdata=data[missing,])
>
>
> perturbedData <- rnorm(nrow(data2[missing,]), mean=linearModelV7, sd=sd(linearModelV7))
> perturbedData
 [1]  7.3781881  8.8042307  3.2648155  2.8762593  0.5813131  1.4041446  3.0548196  4.9955099  1.2669612
[10]  1.8927998 -3.2018328  2.0778216  6.8442864  1.4477760  3.2074673  2.5635595
>
```

Figure 7. This figure is the final output of the missing filled-in values using perturbation via linear regression, with the noise being the standard deviation of the predicted regression model. In order of the perturbed values, they correspond with the first appearing index from the dataset (i.e first value of 7.3781881 is row 24 of V7, 8.8042307 is row 41, etc.).