1 Pre-analysis

As a first step to our analysis, we have to identify possible features that may change the cost of a tube assembly quoted by a supplier. We also ask questions about the dataset where the answers to these questions will help us to choose the right machine learning algorithm(s) to test. Since we are predicting the cost value, a continuous variable, we will use a regression algorithm.

1.1 Questions

To archieve our goal of predicting the cost with a good accuracy, we need to answer the following questions.

1. What features are used to determine the cost and what features to exclude from the analysis?
2. Is there a unique mathematical model describing the cost in function of the quantity for each supplier?
3. If a mathematical model exists, is it a linear or non-linear model?
4. Are there decisions to take? If yes, what decisions?

2 Possible Dependent Features

In this section, we will answer the question: *What features are used to determine the cost and what features to exclude from the analysis?*

2.1 Tube Physical Properties

As a supplier, we have to think on which tube features the cost will be based. We know that a tube assembly is made with one or more components. Some numerical tube properties may be helpful to check.

- The weight
- The quantity
- The volume
- The number of bends used with the bend radius. Logically, it is more difficult to bend a tube than to keep it linear, so it should be more expansive.
- The number of components to assemble a tube.
- The material used to make the tube. Some type of material can be much expensive than others.

2.2 Supplier Features

- The date when the supplier has quoted the price which is certainly less 20 years ago than today when not adjusted.
- The suppliers may use different mathematical models to quote their price.
- The supplier uses or not a bracket pricing which can influence what features to use in both cases.

3 Preparing & Cleaning the Dataset

In this section, we will explain why we chose to keep and exclude features and how we will clean the dataset.

From the dataset, we note that there are a total of 2048 components. These components are spread amoung the `comp_[type].csv` files uniquely. This means that we can create a single table `Component` by merging

those files together. To avoid to many columns, we will remove some features that we do not want in our analysis.

The training and test sets are merged together where we set the cost to 0 for the test set.

The file `bill_of_materials.csv` gives us the list of components with their respective quantity used to assemble a tube. Thus, to calculate the total weight for each tube, we use the formula

$$W_T = \sum_{i=0}^{n} W_i * Q_i$$

where W_T is the total weight of the tube T, $W = (W_1, \ldots, W_n)$ is the vector of component weights, $Q = (Q_1, \ldots, Q_n)$ the vector of component quantities and $n \leq 8$ the number of possible components used to assemble a tube T.

Let the total volume estimation of a tube assembly be denoted by V_T. The volume is function of the length, the wall thickness and the diameter of the tube and its formula is

$$V_T = \pi L t(d - t)$$

, where t is the wall thickness, d the outside diameter and L the developed length of the tube.

Every ID used (e.g. tube_assembly_id, material, supplier, etc.) as a string in CSV files are converted to an integer without the leading zeros. The quote date is converted to an integer in the format `YYYYMMDD`.

To test and find data efficiently, we create a database `Caterpillar` with tables and views. The script to query this database is given by the file `DatabaseManipulation.R`. The script to insert in batch the data from the CSV files to the database is given by the file `DatabaseInsertions.R`.

4 Mathematical Models

In this section, we will answer the question: *Is there a unique mathematical model describing the cost in function of the quantity for each supplier?* The first objective is to check the existence of a mathematical model representing the cost in function of the quantity. The second objective is to show if the model is applied by a unique supplier. The last objective is to show if each supplier has its own model. If the unicity does not hold, then we have to check if a model is applied by more than one supplier or if a supplier can apply more than one model depending of other features. We will then answer the question: *If a mathematical model exists, is it a linear or non-linear model?*

We denote $C_\beta(Q)$ our cost heuristic function of a tube assembly given by a supplier where β is our learning parameters and Q the vector of quantities.

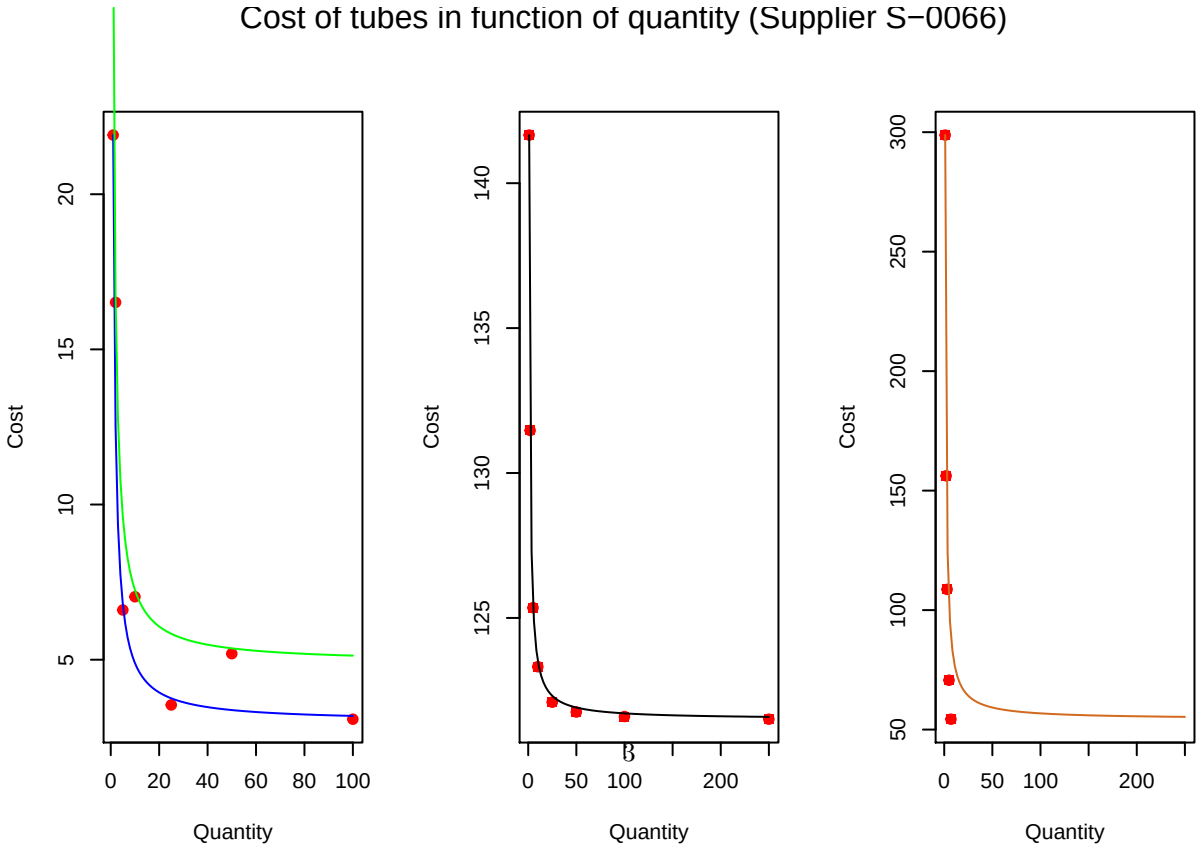
We include librairies for graphs and tables to display. We also connect to the Caterpillar database for the next queries to execute.

4.1 Existence of a Mathematical Model

We start with the few tube assemblies which are quoted by the supplier `S-0066`.

	fkTubeAssembly	supplierID	quantity	cost
1	2	S-0066	1	21.91
2	2	S-0066	2	12.34
3	2	S-0066	5	6.60
4	2	S-0066	10	4.69
5	2	S-0066	25	3.54
6	2	S-0066	50	3.22
7	2	S-0066	100	3.08
8	2	S-0066	250	3.00
9	5	S-0066	1	28.37
10	5	S-0066	2	16.51
11	5	S-0066	5	9.40
12	5	S-0066	10	7.03
13	5	S-0066	25	5.60
14	5	S-0066	50	5.19
15	5	S-0066	100	5.01
16	5	S-0066	250	4.90
17	5000	S-0066	1	141.66
18	5000	S-0066	2	131.47
19	5000	S-0066	5	125.35
20	5000	S-0066	10	123.32
21	5000	S-0066	25	122.09
22	5000	S-0066	50	121.75
23	5000	S-0066	100	121.60
24	5000	S-0066	250	121.51
25	19365	S-0066	1	298.78
26	19365	S-0066	2	156.20
27	19365	S-0066	3	108.67
28	19365	S-0066	5	70.64
29	19365	S-0066	7	54.35

Table 1: Table built from tubes 2, 5, 5000, 19365



From the graphs, we see that the curves estimating the red points are clearly hyperbolas of equation

$$C_T(Q) = \frac{\beta_0 - \beta_1}{Q} + \beta_1$$

where $Q \geq 1$ is the quantity for a tube assembly ID T, β_1 is the cost at the last level of purchase based on quantity and supplier (most of the time $Q = 250$), and β_0 is the cost at the first level of purchase based on quantity and supplier (most of the time $Q = 1$). This equation indicates that if Caterpillar buy more tubes, cheaper will be the cost per tube. This proves the existence of a mathematical model representing the cost in function of the quantity.

If we take a look at the right most graph, we see that our curve doesn't seem to fit the points. However, the maximum quantity is 7 (not 250) for this tube which make the model less accurate assuming the same model is used. This assumption makes sense since

$$\lim_{Q \to \infty} C_T(Q) = \beta_1$$

which means that we need to find the right β_1 to match with any quantity. We also have to find the cost of one tube which is β_0.

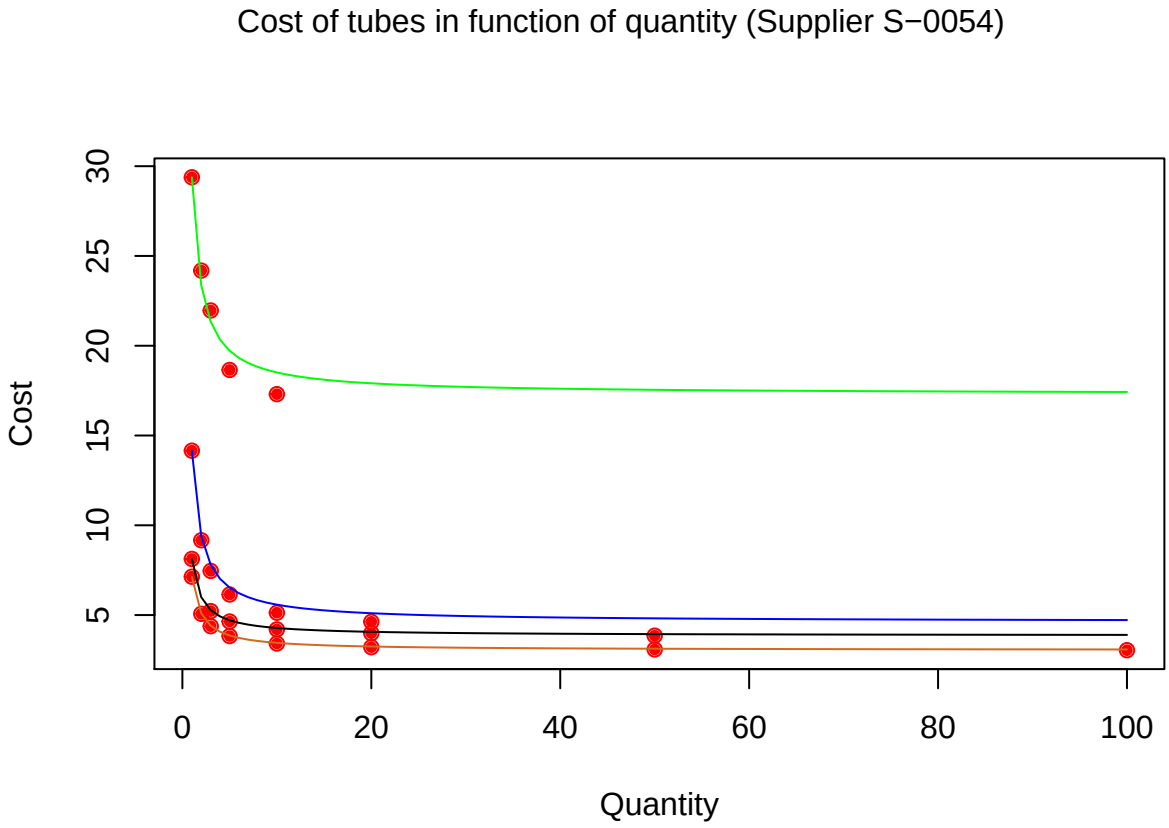
For example, if we take the tube `TA-19365`, we have $C_T(1) = \beta_0 = 298.7820145446$. We know that $C_T(2) = \frac{\beta_0+\beta_1}{2} = 156.1959237271 \Leftrightarrow \beta_1 = 13.60983291$. Therefore, the model for the tube `TA-19365` is $C_T(Q) = \frac{285.172181635}{Q} + 13.60983291$. With $Q = 7$, we obtain $C_T(7) = 54.348716001$ which has a square error of 0.000001422 from the original cost. With our estimated, i.e. $C_T(Q) = (244.434490855/Q) + 54.3475236892$, we have $C_T(7) = 89.266736668$ which has a square error of 1219.351435073. Thus, if Q is small (say $Q < 25$), the model may underfit.

4.2 Unicity of the Model per Supplier

We verify with few tube assemblies which are quoted by the supplier `S-0054` if the same model used for the supplier `S-0066` applies.

	fkTubeAssembly	supplierID	quantity	cost
1	130	S-0054	1	14.16
2	130	S-0054	2	9.17
3	130	S-0054	3	7.46
4	130	S-0054	5	6.15
5	130	S-0054	10	5.13
6	130	S-0054	20	4.63
7	280	S-0054	1	29.38
8	280	S-0054	2	24.18
9	280	S-0054	3	21.96
10	280	S-0054	5	18.65
11	280	S-0054	10	17.30
12	1892	S-0054	1	8.13
13	1892	S-0054	3	5.22
14	1892	S-0054	5	4.64
15	1892	S-0054	10	4.20
16	1892	S-0054	20	3.99
17	1892	S-0054	50	3.85
18	5013	S-0054	1	7.14
19	5013	S-0054	2	5.07
20	5013	S-0054	3	4.38
21	5013	S-0054	5	3.83
22	5013	S-0054	10	3.42
23	5013	S-0054	20	3.21
24	5013	S-0054	50	3.09
25	5013	S-0054	100	3.04

Table 2: Table built from Tube 130, 280, 1892, 5013

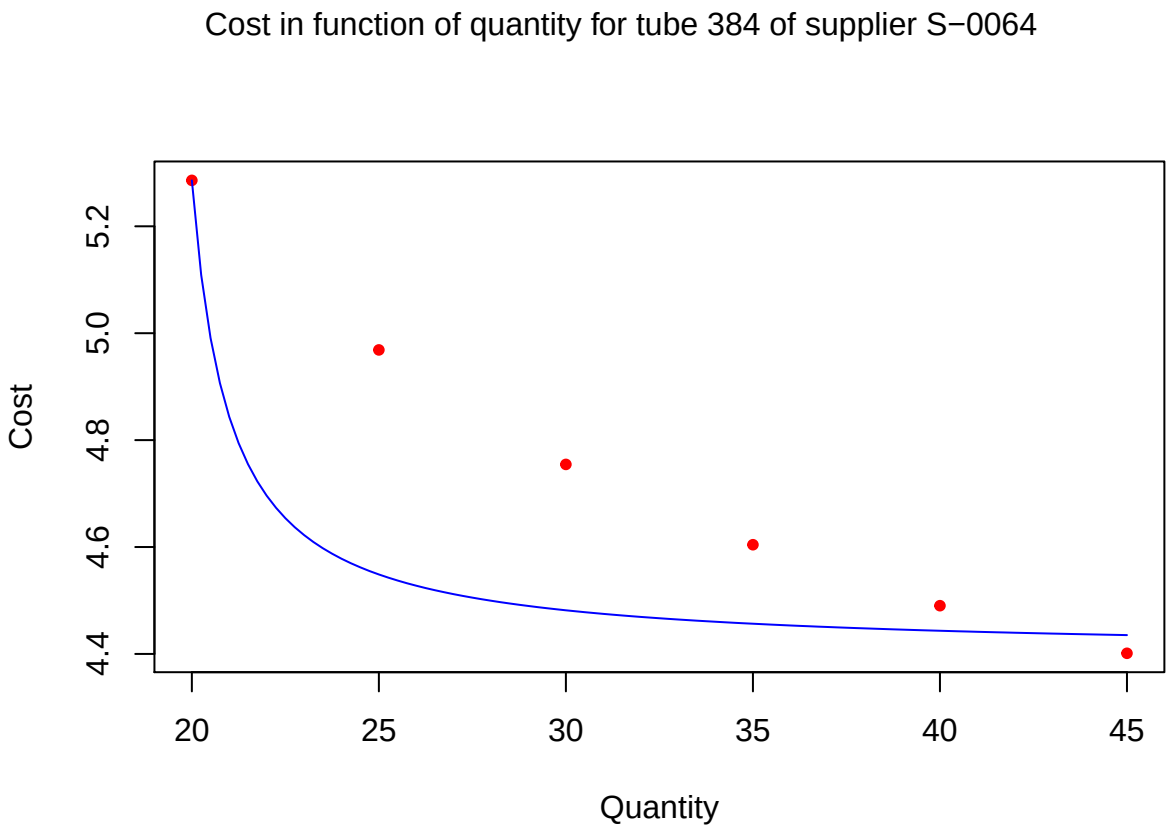


The model used by the supplier `S-0054` seems to be the same as the one used by the supplier `S-0066`, but if we look carefully the curves, we see that greater is the quantity, more accurate is the estimate. This means that the model follows the same behaviour as the model used by the supplier `S-0066`.

This doesn't seem to be the case for the tube `TA-00384` from the supplier `S-0064`. This supplier provides 6 levels of purchase where the highest quantity is $Q = 45$. We use the same model as before but this time, the model doesn't fit the points.

	fkTubeAssembly	supplierID	minOrderQuantity	anualUsage	quantity	cost
1	384	S-0064	0	0	20	5.29
2	384	S-0064	0	0	25	4.97
3	384	S-0064	0	0	30	4.75
4	384	S-0064	0	0	35	4.60
5	384	S-0064	0	0	40	4.49
6	384	S-0064	0	0	45	4.40

Table 3: Tube 384



Since the first quantity level is 20, we need to translate the model by subtracting x by $Q_0 - 1 = 19$. This gives the following model

$$C_T(Q) = \frac{\beta_0 - \beta_1}{Q - Q_0 - 1} + \beta_1$$

if $Q > 1$. However, the model still underfits the data because the cost decreases much slower than the model used for our previous tests. Therefore, we can assume that a model can be used to estimate the cost by one or many suppliers but not all.

5 Decision Tree(s)

In this section, we will identify conditional paths which will tell us if decision trees will be useful or not. We will answer the question: *Are there decisions to take? If yes, what decisions?* In the previous section, we have seen that the model can underfit if there are not enough quantity purchase levels and if the quantity is small. Otherwise, we can use the model to estimate the cost given a quantity and a tube assembly. Here are few points that identify some conditions.

- If there is only one quantity purchase level, we cannot estimate the cost. We need al least another feature on which the cost depends.
- If there are many quantity purchase levels but with $Q_0 > 1$, then we need to translate the model found at section 4.
- If the quantities are too low, then the model underfits. Thus, we need to add other cost-dependent features.
- Depending of the supplier, the model may be different. Since we have 57 suppliers in the train set, we can have at most 57 possible models.

Only with those conditions, we can build many decision trees to help us to estimate the cost.

6 Machine Learning Algorithms and Results

In this section, we present the machine learning algorithms we will try after the analysis given by the four previous sections. Per the section 4 (Decisions), we have seen that many decision trees can be built. This leaves us two possible algorithms: Random Forest or Boosting Trees for Regression (XGBoost).

6.1 Random Forest Algorithm

We use only the year and month parts of the date since the day in the same month and year will not influance the cost.

```
##          used (Mb) gc trigger (Mb) max used (Mb)
## Ncells 527716 28.2     940480 50.3   750400 40.1
## Vcells 983543  7.6    1650153 12.6  1307238 10.0
```

```
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##    2 |   0.1243    18.35 |
##    4 |   0.1038    15.31 |
##    6 |  0.09158    13.51 |
##    8 |  0.08353    12.33 |
##   10 |  0.07659    11.30 |
##   12 |   0.0703    10.37 |
##   14 |  0.06677     9.85 |
##   16 |  0.06343     9.36 |
##   18 |  0.06156     9.08 |
##   20 |  0.06024     8.89 |
##   22 |  0.05887     8.69 |
##   24 |  0.05788     8.54 |
##   26 |  0.05702     8.41 |
##   28 |  0.05625     8.30 |
```

```
##   30 |  0.05562     8.21 |
##   32 |  0.05527     8.16 |
##   34 |  0.05481     8.09 |
##   36 |  0.05437     8.02 |
##   38 |  0.05384     7.94 |
##   40 |  0.05314     7.84 |
##   42 |  0.05281     7.79 |
##   44 |  0.05254     7.75 |
##   46 |  0.05249     7.74 |
##   48 |  0.05216     7.70 |
##   50 |  0.05188     7.66 |
##   52 |   0.0518     7.64 |
##   54 |  0.05168     7.63 |
##   56 |  0.05147     7.59 |
##   58 |  0.05118     7.55 |
##   60 |  0.05117     7.55 |
##   62 |  0.05097     7.52 |
##   64 |  0.05089     7.51 |
##   66 |  0.05075     7.49 |
##   68 |  0.05059     7.47 |
##   70 |  0.05069     7.48 |
##   72 |  0.05054     7.46 |
##   74 |  0.05046     7.44 |
##   76 |  0.05051     7.45 |
##   78 |  0.05046     7.45 |
##   80 |  0.05024     7.41 |
##   82 |  0.05014     7.40 |
##   84 |  0.05002     7.38 |
##   86 |  0.04995     7.37 |
##   88 |  0.04981     7.35 |
##   90 |  0.04974     7.34 |
##   92 |  0.04966     7.33 |
##   94 |   0.0496     7.32 |
##   96 |  0.04964     7.32 |
##   98 |  0.04949     7.30 |
##  100 |  0.04947     7.30 |
##  102 |  0.04933     7.28 |
##  104 |  0.04931     7.28 |
##  106 |  0.04927     7.27 |
##  108 |  0.04928     7.27 |
##  110 |  0.04922     7.26 |
##  112 |  0.04918     7.26 |
##  114 |  0.04911     7.25 |
##  116 |   0.0491     7.24 |
##  118 |  0.04913     7.25 |
##  120 |  0.04914     7.25 |
##  122 |  0.04911     7.25 |
##  124 |  0.04908     7.24 |
##  126 |   0.0491     7.25 |
##  128 |  0.04907     7.24 |
##  130 |  0.04909     7.24 |
##  132 |  0.04906     7.24 |
##  134 |  0.04908     7.24 |
##  136 |  0.04904     7.24 |
```

```
##  138 |  0.04902     7.23 |
##  140 |    0.049     7.23 |
##  142 |  0.04898     7.23 |
##  144 |  0.04899     7.23 |
##  146 |  0.04896     7.22 |
##  148 |  0.04896     7.22 |
##  150 |  0.04894     7.22 |
##  152 |  0.04889     7.21 |
##  154 |   0.0489     7.22 |
##  156 |  0.04887     7.21 |
##  158 |  0.04886     7.21 |
##  160 |  0.04882     7.20 |
##  162 |  0.04891     7.22 |
##  164 |  0.04894     7.22 |
##  166 |  0.04894     7.22 |
##  168 |  0.04896     7.22 |
##  170 |  0.04895     7.22 |
##  172 |  0.04899     7.23 |
##  174 |  0.04897     7.23 |
##  176 |  0.04896     7.22 |
##  178 |  0.04891     7.22 |
##  180 |  0.04891     7.22 |
##  182 |  0.04886     7.21 |
##  184 |  0.04883     7.21 |
##  186 |  0.04883     7.20 |
##  188 |  0.04882     7.20 |
##  190 |  0.04882     7.20 |
##  192 |  0.04878     7.20 |
##  194 |  0.04878     7.20 |
##  196 |   0.0488     7.20 |
##  198 |  0.04879     7.20 |
##  200 |  0.04879     7.20 |
##  202 |   0.0488     7.20 |
##  204 |  0.04876     7.20 |
##  206 |   0.0488     7.20 |
##  208 |  0.04877     7.20 |
##  210 |  0.04877     7.20 |
##  212 |  0.04879     7.20 |
##  214 |  0.04882     7.20 |
##  216 |  0.04879     7.20 |
##  218 |  0.04876     7.19 |
##  220 |  0.04871     7.19 |
##  222 |  0.04869     7.18 |
##  224 |  0.04865     7.18 |
##  226 |  0.04865     7.18 |
##  228 |  0.04863     7.18 |
##  230 |  0.04861     7.17 |
##  232 |  0.04858     7.17 |
##  234 |  0.04858     7.17 |
##  236 |  0.04857     7.17 |
##  238 |  0.04859     7.17 |
##  240 |  0.04859     7.17 |
##  242 |  0.04856     7.17 |
##  244 |  0.04858     7.17 |
```

```
##  246 |  0.04856     7.17 |
##  248 |  0.04853     7.16 |
##  250 |  0.04855     7.16 |
##  252 |  0.04855     7.16 |
##  254 |  0.04852     7.16 |
##  256 |  0.04851     7.16 |
##  258 |  0.04853     7.16 |
##  260 |  0.04853     7.16 |
##  262 |  0.04852     7.16 |
##  264 |  0.04854     7.16 |
##  266 |  0.04854     7.16 |
##  268 |  0.04853     7.16 |
##  270 |   0.0485     7.16 |
##  272 |  0.04852     7.16 |
##  274 |  0.04852     7.16 |
##  276 |  0.04847     7.15 |
##  278 |  0.04846     7.15 |
##  280 |  0.04843     7.15 |
##  282 |  0.04843     7.15 |
##  284 |  0.04843     7.15 |
##  286 |  0.04841     7.14 |
##  288 |  0.04837     7.14 |
##  290 |  0.04836     7.14 |
##  292 |  0.04836     7.14 |
##  294 |  0.04836     7.14 |
##  296 |  0.04835     7.13 |
##  298 |  0.04835     7.13 |
##  300 |  0.04833     7.13 |
##  302 |  0.04833     7.13 |
##  304 |  0.04835     7.13 |
##  306 |  0.04833     7.13 |
##  308 |  0.04833     7.13 |
##  310 |  0.04834     7.13 |
##  312 |  0.04832     7.13 |
##  314 |  0.04831     7.13 |
##  316 |   0.0483     7.13 |
##  318 |  0.04832     7.13 |
##  320 |  0.04829     7.13 |
##  322 |  0.04829     7.13 |
##  324 |  0.04829     7.12 |
##  326 |  0.04826     7.12 |
##  328 |  0.04823     7.12 |
##  330 |  0.04825     7.12 |
##  332 |  0.04823     7.12 |
##  334 |  0.04822     7.12 |
##  336 |  0.04821     7.11 |
##  338 |   0.0482     7.11 |
##  340 |  0.04821     7.11 |
##  342 |   0.0482     7.11 |
##  344 |  0.04817     7.11 |
##  346 |  0.04815     7.11 |
##  348 |  0.04814     7.10 |
##  350 |  0.04814     7.10 |
##  352 |  0.04816     7.11 |
```

```
##  354 |  0.04815     7.11 |
##  356 |  0.04815     7.10 |
##  358 |  0.04813     7.10 |
##  360 |  0.04811     7.10 |
##  362 |  0.04811     7.10 |
##  364 |  0.04809     7.10 |
##  366 |  0.04808     7.09 |
##  368 |  0.04809     7.10 |
##  370 |  0.04808     7.09 |
##  372 |  0.04808     7.09 |
##  374 |  0.04806     7.09 |
##  376 |  0.04805     7.09 |
##  378 |  0.04805     7.09 |
##  380 |  0.04803     7.09 |
##  382 |  0.04803     7.09 |
##  384 |  0.04804     7.09 |
##  386 |  0.04804     7.09 |
##  388 |  0.04806     7.09 |
##  390 |  0.04807     7.09 |
##  392 |  0.04807     7.09 |
##  394 |  0.04805     7.09 |
##  396 |  0.04806     7.09 |
##  398 |  0.04803     7.09 |
##  400 |  0.04803     7.09 |
##  402 |  0.04805     7.09 |
##  404 |  0.04805     7.09 |
##  406 |  0.04807     7.09 |
##  408 |  0.04807     7.09 |
##  410 |  0.04803     7.09 |
##  412 |  0.04806     7.09 |
##  414 |  0.04809     7.10 |
##  416 |   0.0481     7.10 |
##  418 |  0.04807     7.09 |
##  420 |  0.04807     7.09 |
##  422 |  0.04806     7.09 |
##  424 |  0.04807     7.09 |
##  426 |  0.04807     7.09 |
##  428 |  0.04809     7.10 |
##  430 |  0.04807     7.09 |
##  432 |  0.04806     7.09 |
##  434 |  0.04806     7.09 |
##  436 |  0.04807     7.09 |
##  438 |  0.04808     7.09 |
##  440 |  0.04807     7.09 |
##  442 |  0.04806     7.09 |
##  444 |  0.04806     7.09 |
##  446 |  0.04809     7.10 |
##  448 |  0.04808     7.09 |
##  450 |  0.04809     7.10 |
##  452 |  0.04809     7.10 |
##  454 |  0.04808     7.09 |
##  456 |  0.04806     7.09 |
##  458 |  0.04807     7.09 |
##  460 |  0.04806     7.09 |
```

```
##  462 |  0.04806     7.09 |
##  464 |  0.04806     7.09 |
##  466 |  0.04808     7.09 |
##  468 |  0.04808     7.09 |
##  470 |  0.04808     7.09 |
##  472 |  0.04805     7.09 |
##  474 |  0.04803     7.09 |
##  476 |  0.04802     7.09 |
##  478 |    0.048     7.08 |
##  480 |    0.048     7.08 |
##  482 |    0.048     7.08 |
##  484 |  0.04798     7.08 |
##  486 |    0.048     7.08 |
##  488 |  0.04799     7.08 |
##  490 |    0.048     7.08 |
##  492 |    0.048     7.08 |
##  494 |    0.048     7.08 |
##  496 |  0.04799     7.08 |
##  498 |    0.048     7.08 |
##  500 |  0.04801     7.08 |
```