Summary of first results ADR paper

1. Proposed methodology

This paper is particular because its starting point lays on the field of data analytics, and then it proposes a decision-making framework based on mathematical modelling and risk-adverse optimization. The following Figure 1 summarizes the methodology, we followed the steps below:

1. Exploratory Data Analysis. The US Health Insurance dataset contains a sample from health insurance industry on US. The sample was taken in order to analyze risk underwriting in that industry, which is known of being built over complex rules.
   1.1. Clustering: unsupervised learning. The original dataset has seven features. In order to generate interpretable data analytics, a dimensionality reduction algorithm was used to reduce the feature space to a bidimensional one. The algorithm was t-distributed Stochastic Neighbor Embedding (t-SNE), which comes from manifold learning. It performs dimensionality reduction while it infers clusters from data.
   1.2. Clustering: categorical features. Based on domain knowledge, we hypothesized that some features create natural clusters on the data, this is the case for 'smoker', 'body mass index', and 'age'.

Equation 1.

$$\mathrm{H}_0\text{: categorical feature does not create natural clusters}$$

$$\mathrm{H}_1\text{: categorical feature creates natural clusters}$$

This hypothesis is tested through visualization (after dimensionality reduction) and ANOVA based Turkey HSD test. We reject H_0 if visualization shows that no clusters can be identified and if ANOVA test detects no statistically significant differences in the dependent variable 'charges' between pairwise combinations of clusters created by categories. The main reason behind categorical clustering is to reach simpler rules for stratification of data and produce insights on decision-making that are conditional to each stratum.

2. Re-sampling. Decision-making will be based on a stochastic optimization framework. The Sampling Average Approximation (SAA) method is suitable for two-stage stochastic programming where the uncertainty comes from a sample of stochastic variables. Thus, optimization of mathematical model leads to a decision that is robust to all sampled scenarios. A classical approach is to perform controlled simulations on such variables, but the main challenge lays in finding a statistical distribution that suits real-world variables behavior, specially for high-risk cases where the variable 'charges' can take abnormally high values that can lead insurance companies to bankrupt. In consequence, controlled simulations guarantee convergence in SAA, but leads to loss of valuable information that exist in real data. To overcome such limitations, the following re-sampling algorithms were proposed:
   2.1. Monte Carlo Bootstrap (MCB). This method is widely used in literature to solve mathematical models using real-world data. The challenge produced by this method is that it requires several samples to produce unbiased results in SAA. Under the properties of Theorem of Central Limit (TCL), this method leads to unbiased estimators of sample moments. However, we are not seeking to reproduce moments, but to reproduce distribution properties to get a realistic representation of uncertainty. In this case, the bootstrap method would require several samples to capture atypical values

on the dependent variable. The challenge comes from this fact, small bootstrap sample sizes would lead to different results on stochastic programming SAA method. Unfortunately, bigger samples would lead to exponentially slower converge of SAA. The number of scenarios in SAA is equal to the number of bootstrap samples. For instance, if we take 1000 samples with replacement from original sample, the SAA algorithm will run on 30 minutes for each replication. At least 30 replications are needed to estimate the solution quality, which would lead to expensive computational burden.

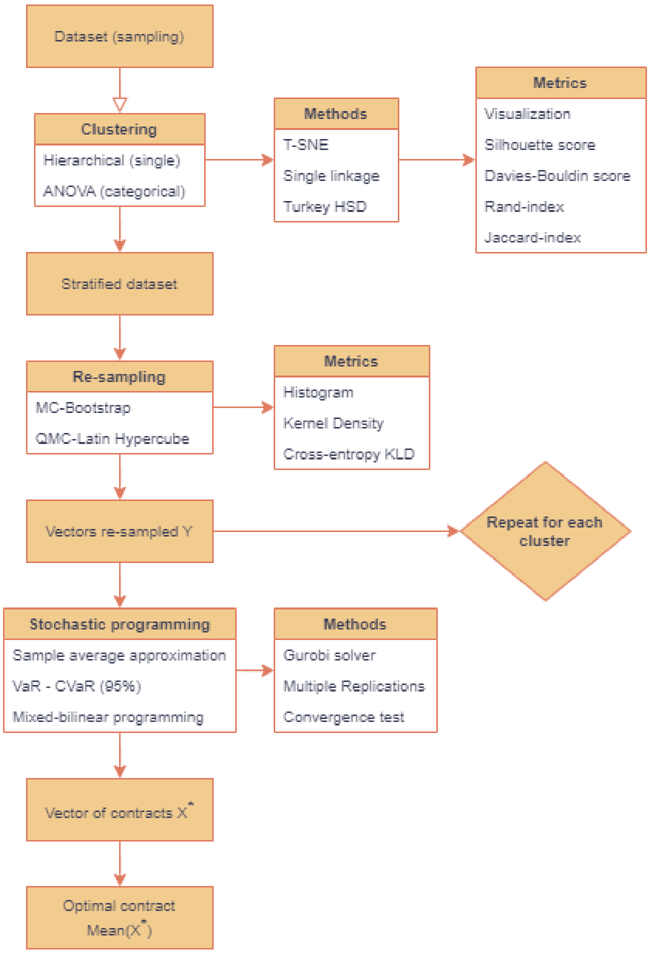
   2.2. Quasi-Monte Carlo Latin Hypercube Sampling (QMC-LHS). To overcome the limitation above, an alternative sampling algorithm was proposed by Budiman (2006). This sampling algorithm is more efficient, as it requires smaller samples to produce unbiased results. Both the MC-Bootstrap and QMC-LHS will be tested against the original distribution of random variable 'charges' that is sampled on original dataset. In order to measure the quality of the re-sampling algorithms the Wasserstein distance, known also as the Earth Mover's distance, of re-sampled distribution from originally sampled distribution of dependent variable will be estimated, as well as its confidence interval. This step is crucial to measure the quality of re-sampling algorithms. The QMC-LHS is used here to reach minimum variance results on SAA algorithm with lower number of scenarios required to model uncertainty and produce decision-making insights.

3. Initialize mathematical optimization of stochastic programming model. The model is a mixed integer bi-linear stochastic program type. Expensive computational burden comes from the fact that the number of restrictions increases with the number of scenarios or samples extract from real-world data. This model also includes VaR-CVaR restrictions to account for risk-adverse optimization. The SAA algorithm's solution requires quality assessment, as it lays on the property of convergence of solution of the sample problem to the solution for the true problem for a large number of scenarios. A classic approach to test such convergence is to adopt the Multiple Replications Procedure (MRP) that produces multiple solutions at a fixed number of scenarios to get an estimate of optimality gap that is a metric that assess solution quality and to build a confidence interval for such gap based on TCL and Monte Carlo approaches (Mak and Morton, 1999).
4. Finalize mathematical optimization an obtain the optimal solution as the mean of the solution vector obtained from MRP at a fixed number of scenarios that is sufficiently high to reach convergence to true problem solution. True problem solution is represented by sample problem solution at a very large number of scenarios, this procedure was suggested by Mark and Morton (1999) to create an estimate of optimality gap.
5. Repeat step 2, 3 and 4 for each stratum. This paper will conclude with comparison of optimality for two strategies. Value of objective function will be considered to generate insights that guides decision-making considering uncertainty in health insurance underwriting:

   5.1. Unique optimal contract for the entire sample.

   5.2. Multiple contracts that are drawn for different parameters for each stratum.

Figure 1. Proposed methodology



2. Preliminary results

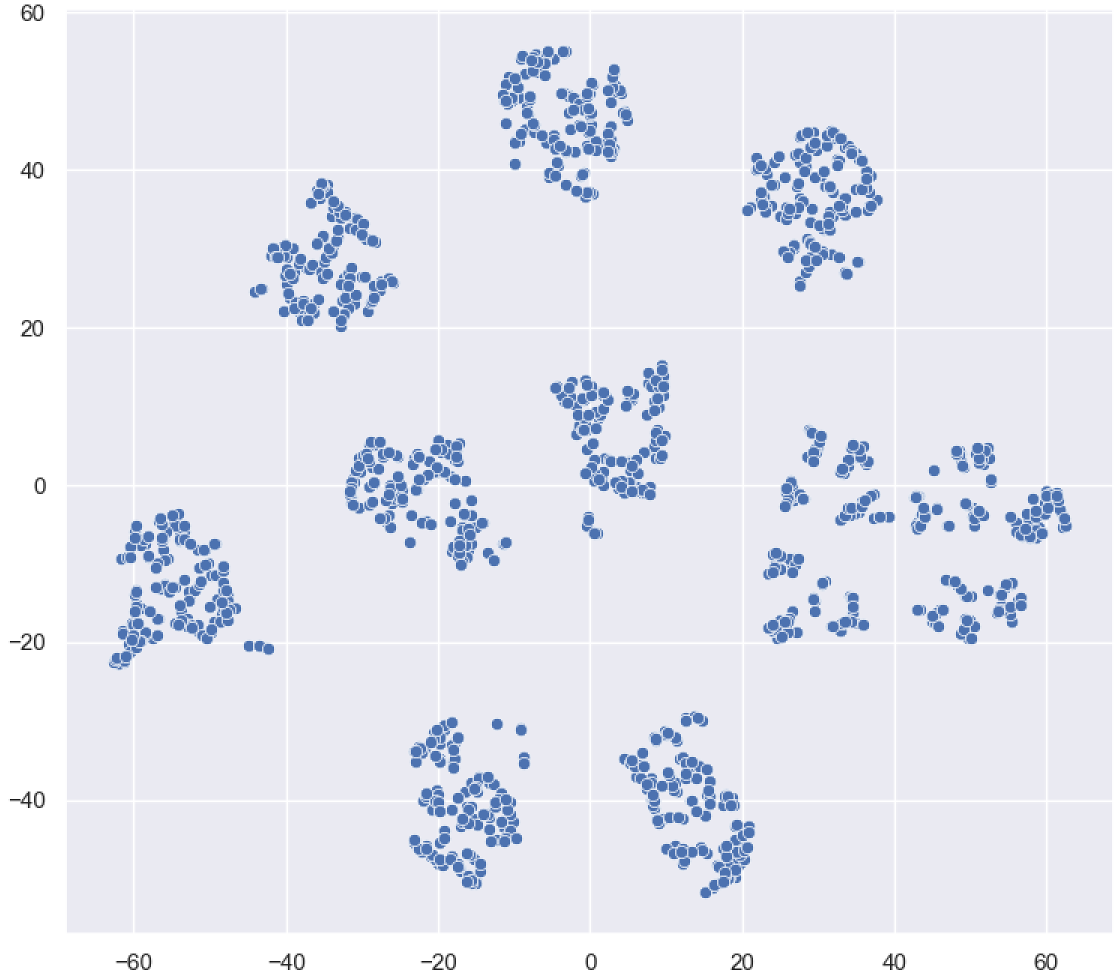
2.1. t-SNE dimensionality reduction

With a robust choice of hyperparameters, the unsupervised learning algorithm identified 9-10 clusters. The algorithm reduced the dimensionality of dataset from 12 features to 2 features. As categorical variables were also included as features and a dummy was created for each category, t-SNE was used as a non-linear dimensionality reduction method. Non-linearity and non-normality of categorical features produced bad visual analytics from classic approach based on Principal Component Analysis (PCA) or Multiple Correspondence Analysis (MCA). The Figure 2 shows the results of t-SNE algorithm.

After the algorithm we proceed to test categorical clustering based on the following variables that passed through a Feature Engineering process:

- Smoker: no feature engineering applied. Smokers are more likely to generate higher charges for health insurance companies.
- Body mass index: discretization. Obesity is defined by WHO where body mass index is above 30kg/m². Obese are more likely to generate higher charges for health insurance companies.
- Age: discretization. Old adults are individuals over 60 years old. Old adults are more likely to generate higher charges for health insurance companies.

Figure 2.



The results of t-SNE does not generate the labels for each identified cluster. To solve this limitation, clusters were identified through Hierarchical clustering algorithm. Figure 3 summarizes the findings, with number of clusters set to 9. At this point we suspect that an additional cluster exists, and it comes from the partition of Cluster 0 into two clusters, specially for the observed differences on charges variable.

To test the hypothesis on Equation 1 for each categorical feature, that stats that a categorical feature creates a natural cluster on data, we will reproduce the visualization on all the categories, including the categories derived from combinations of categories (i.e., smoker-yes obesity-yes). Figure 4 presents the visualization of these results.

According to Figure 4, the categorical clustering coincides with the unsupervised learning for ‘smoker’ and ‘obesity’ variables. However, there are only three clear clusters:

- Cluster 0: non-smoker
- Cluster 1: smoker and non-obese
- Cluster 2: smoker and obese

The rest of groups generated by categories have lower frequency in sample and seem to be uniformly distributed over clusters identified by t-SNE algorithm. For example, obesity does not matter when individuals are non-smokers (see Figure 5).

Figure 3.

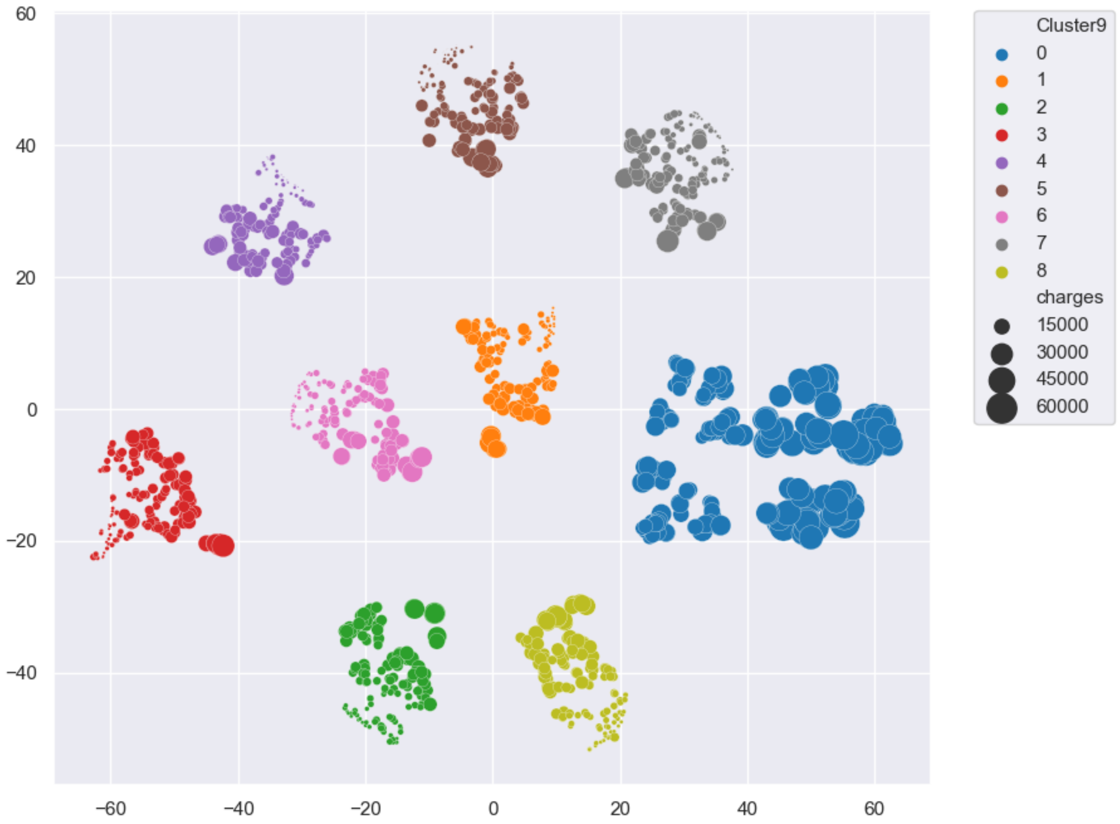
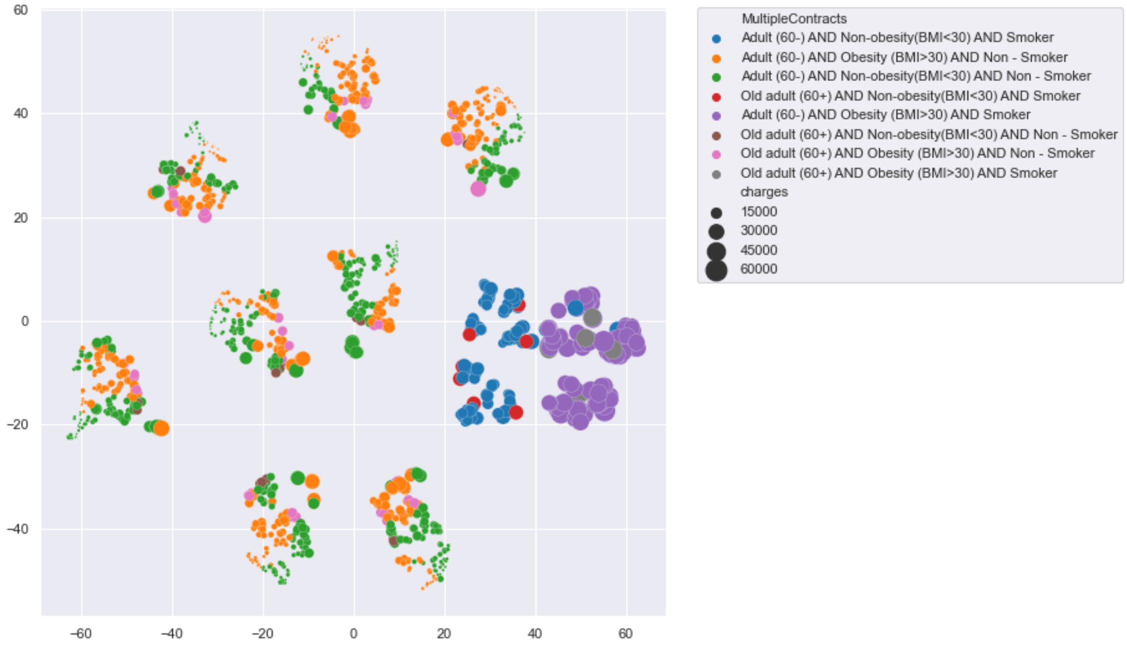


Figure 4.



In this point, we proceed to discard the category Adult and Old adult, generated by the variable ‘age’. We are not negating the importance of age for health insurance, but we are showing evidence against the ability of age to create natural clusters in data. Decision-making will be guided here by stratification of data, although ‘age’ is positively correlated with ‘charges’, it is possible to consider this into each sample passed to SAA algorithm. Every optimal contract for each cluster or stratum would consider that 'charges' are more likely to be high for the old, but the design of contracts is based on stratification rather than on this univariate relationship (?). The Figure 5 shows a boxplot for new categories considering only 'smoker' and 'body mass index' variables. Table 1 shows one-way ANOVA statistical tests' results for pairwise combinations of categories produced by these two variables.

Figure 5.

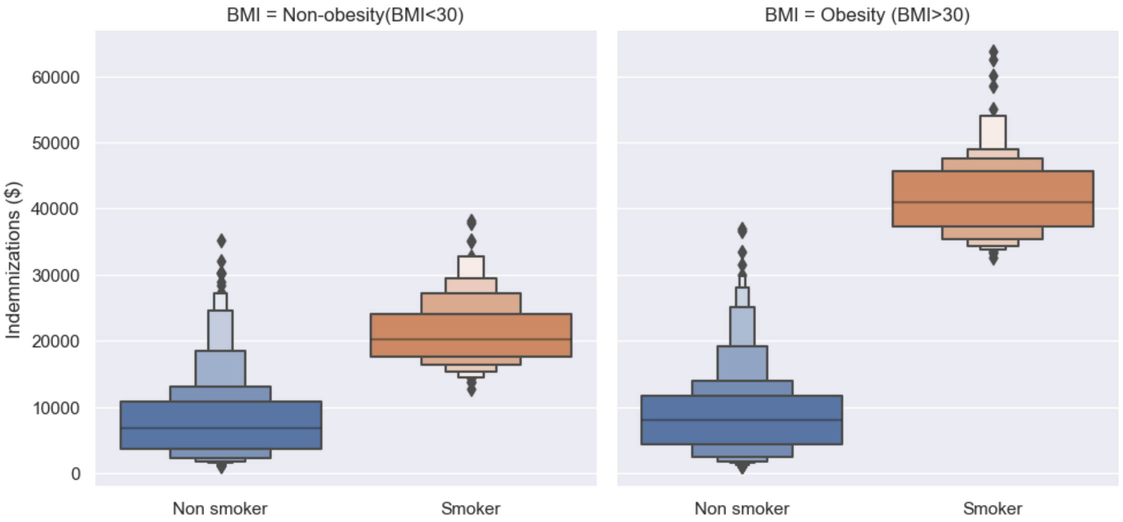


Table 1.

Group1	Group2	Diff(charges)	Q	p-value
Non-obesity (BMI<30) AND Smoker	Obesity (BMI>30) AND Non-smoker	12515.9	30.9*	0.0
Non-obesity (BMI<30) AND Smoker	Non-obesity (BMI<30) AND Non-smoker	13402.3	32.8*	0.0
Non-obesity (BMI<30) AND Smoker	Obesity (BMI>30) AND Smoker	20323.6	40.4*	0.0
Obesity (BMI>30) AND Non-smoker	Non-obesity (BMI<30) AND Non-smoker	886.3	3.5	0.1
Obesity (BMI>30) AND Non-smoker	Obesity (BMI>30) AND Smoker	32839.5	84.5*	0.0
Non-obesity (BMI<30) AND Non-smoker	Obesity (BMI>30) AND Smoker	33725.9	85.8*	0.0

(*) statistically significant at 95% of confidence.

As expected, there are no statistically significant differences between the categorical cluster 'Obesity (BMI>30) AND Non-smoker' and 'Non-obesity (BMI<30) AND Non-smoker'. Within smokers, obesity makes difference for 'charges', and between smokers and non-smoker, every pairwise combinations are statistically significant. However, within non-smokers, obese people are not related to higher or lower charges. This supports the proposition of the existence of three clusters:

- Cluster 0: non-smoker
- Cluster 1: smoker and non-obese
- Cluster 2: smoker and obese

3. Re-sampling algorithms

3.1. Monte Carlo Bootstrap

The classical bootstrap follows the next steps:

1. Define a desired sample size equal to parameter ‘$sample\_size$’
2. Draw a random sample with replacement with the chosen size
    2.1. Generate a vector of uniformly distributed random numbers in the interval [0,1]
    2.2. Concatenate the vector to the data ‘y’
    2.3. Order the vector ascending and keep the maximum value to sample the ‘ith’ value
    2.4. Repeat the process until reach the desired ‘$sample\_size$’.

3.2. Quasi Monte Carlo Latin Hypercube Sampling

The algorithm for this re-sampling method was developed by Budiman (2006) and follows the next steps:

1. Generate a vector r_u of uniformly distributed random numbers in the interval [0,1], the shape of vector is equal to the desired sample size, measured by the parameter ‘$sample\_size$’.
2. Permute the index of the vector that is going to be re-sampled, which enters the function as ‘y’, create a vector with shuffled indexes called ‘idx’.
3. Divide the distribution of the vector ‘y’ into ‘sample_size’ intervals. Thus, creating a fully stratified scheme for sampling (Budiman, 2006).
    3.1. For the $'ith'$ interval, the sampled cumulative probability can be written as:
    $\mathrm{prob}_i = \frac{idx_i\text{-}\mathrm{r_{iu}}}{\mathrm{sample\_size}}, i{=}1,\ldots,\mathrm{sample\_size}, r_{iu} \sim Uniform(0{,}1)$
4. Sample values from the vector ‘y’ calculating the percentiles for each value in the vector of sampled cumulative probabilities ‘prob’. For instance, percentile 20 would generate a corresponding value which is the sampled value[1].

3.3. Performance evaluation of re-sampling algorithms

As both MC-Bootstrap and QMC-LHS algorithms do not make distributional assumptions over the data, a good metric to estimate the quality of re-sampling is the distance between distributions. This distance is measured with the Wasserstein Distance metric as described in the following Equation 2:

Equation 2. Wasserstein distance

$$W(u,v) = \int_{-\infty}^{\infty} |U - V|\, dy$$

For two vectors u and v that are samples for the random variable y, U and V are the Cumulative Density Functions of such vectors of samples. The empirical computation of Wasserstein distance allows for a point-by-point comparison of two distributions, and it overcomes the limitations arising from distributional assumptions of other comparison metrics such as Kolmogorov-Smirnov or ANOVA based metrics. The procedure for performance evaluation of re-sampling algorithms is the following:

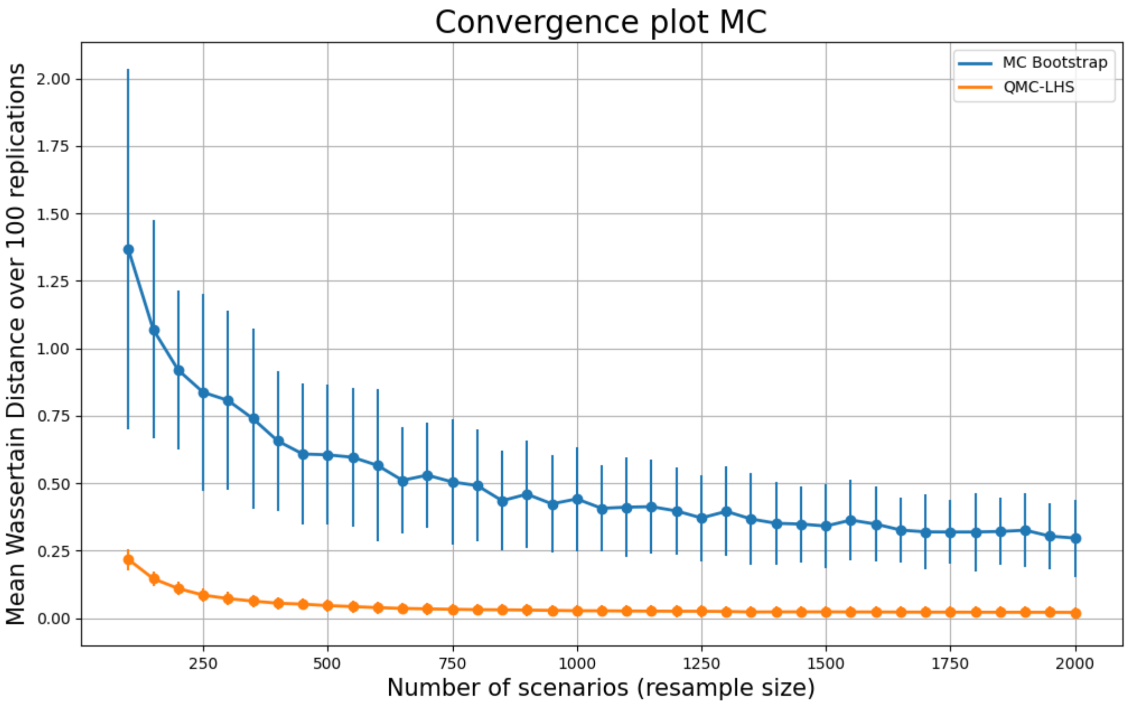
1. Choose a grid for parameter ‘$sample\_size$’ for re-sampling procedure.
2. Compute $W(u,v)$ between original vectors and re-sampled vectors for a fixed ‘$sample\_size$’.
3. Repeat the process $'R'$ times equal to the desired number of replications, as re-sampling is performed at random.
4. Estimate the mean and standard deviation of $W(u,v)$ over the $'R'$ replications

[1] This step performs the fully stratified sampling and overcomes the limitations of making additional distribution assumptions to sample from inverse of distribution function $F^{-1}(prob)$.

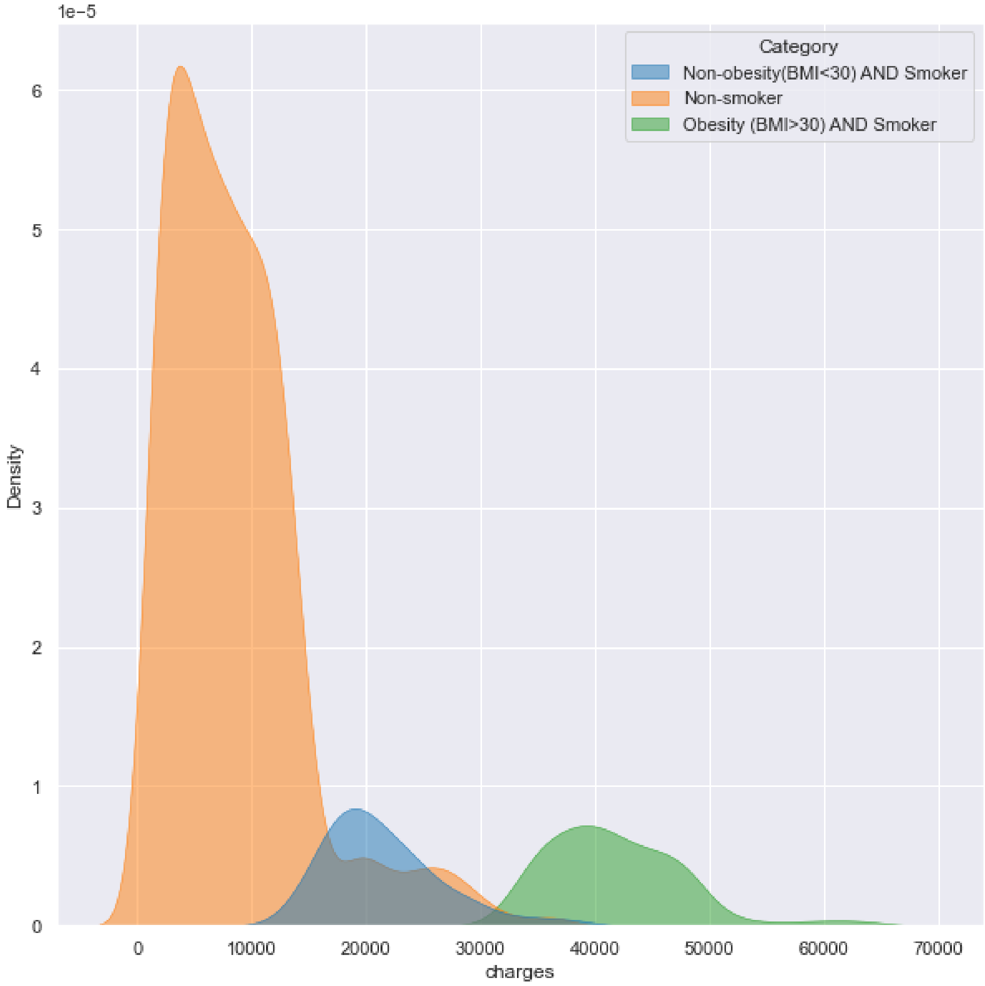
5. Repeat steps 2, 3 and 4 for each value of parameter '$sample\_size$' within the grid defined in step 1.
6. Collect and analyze the experimental results in terms of convergence $W(u, v)$ and its variability among the 'R' replications for each point of the grid defined in step 1.

In Figure 6 we show the results of these experiments. As expected, QMC-LHS outperform classical MC-Bootstrap in both convergence and variability among replications.

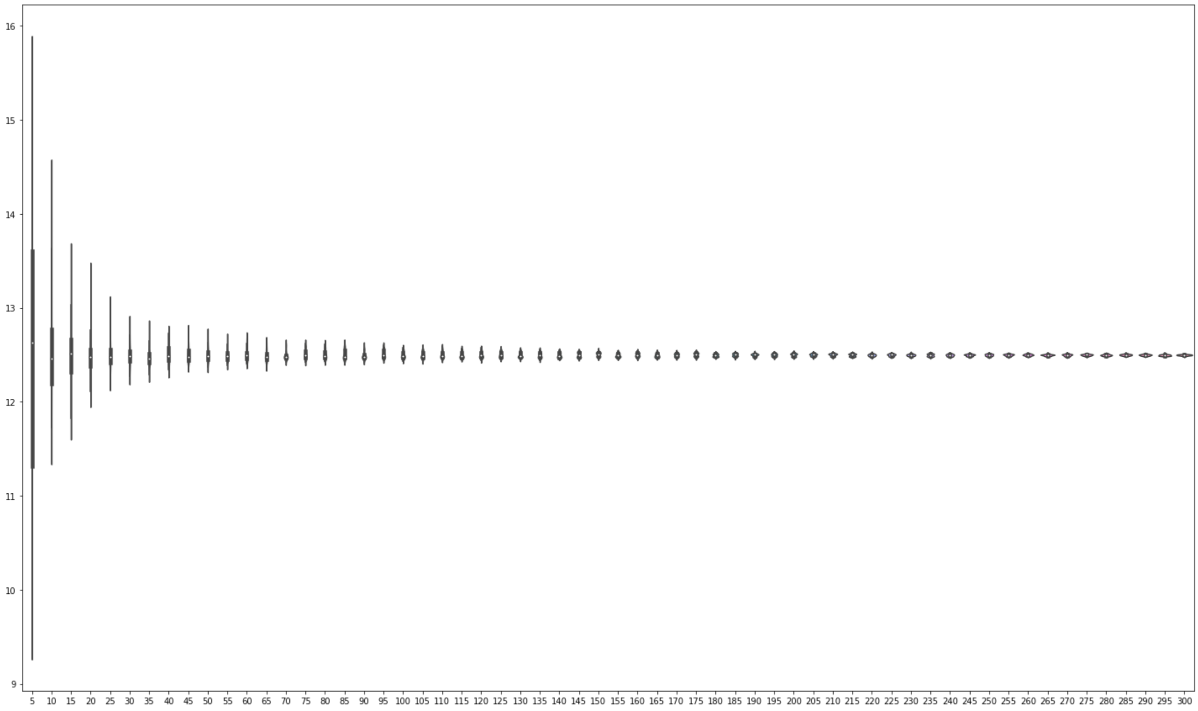
Figure 6.



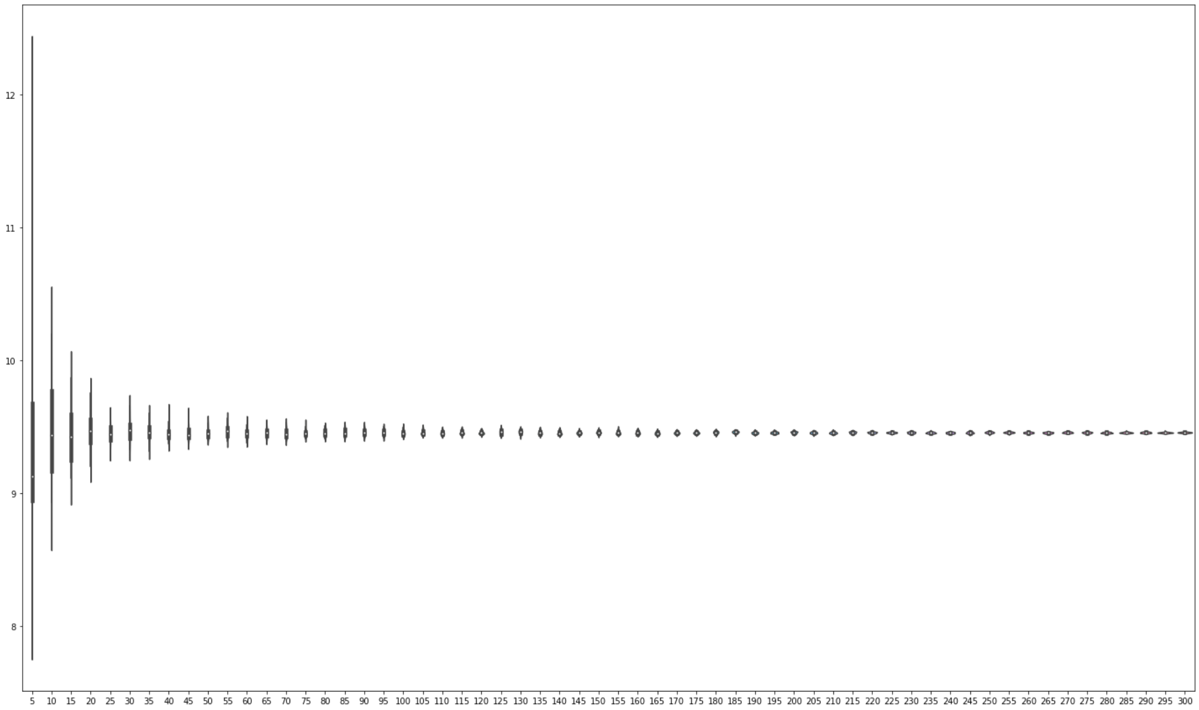
Sampling Average Approximation

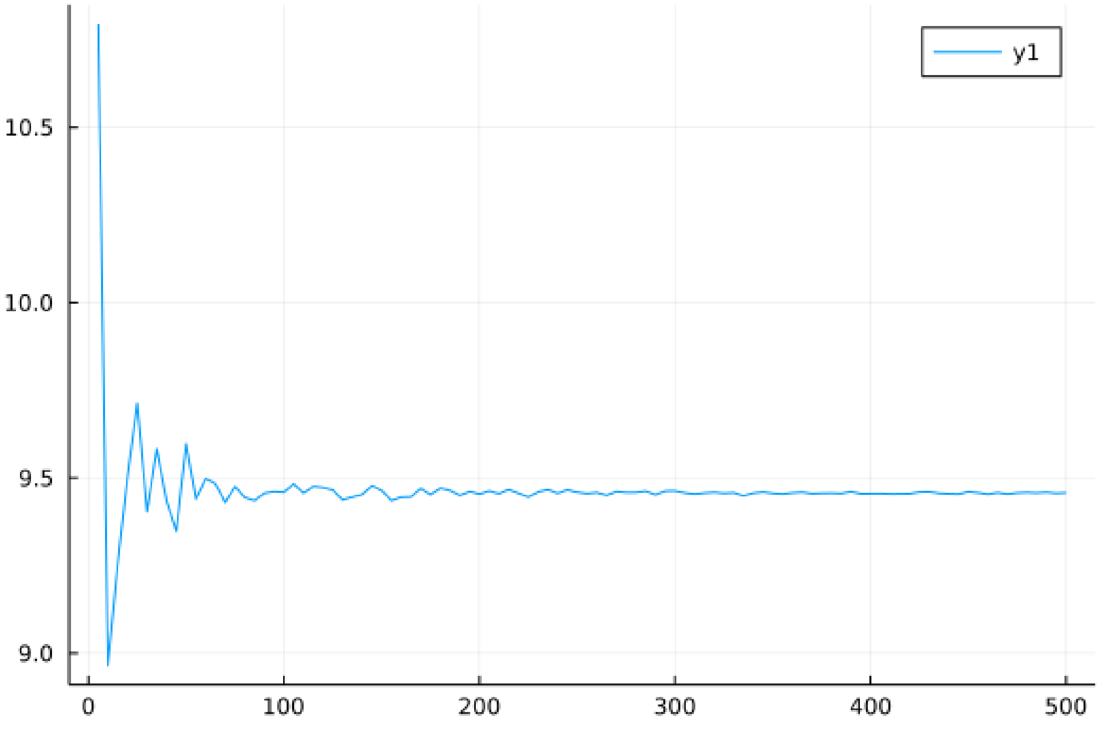


All sample

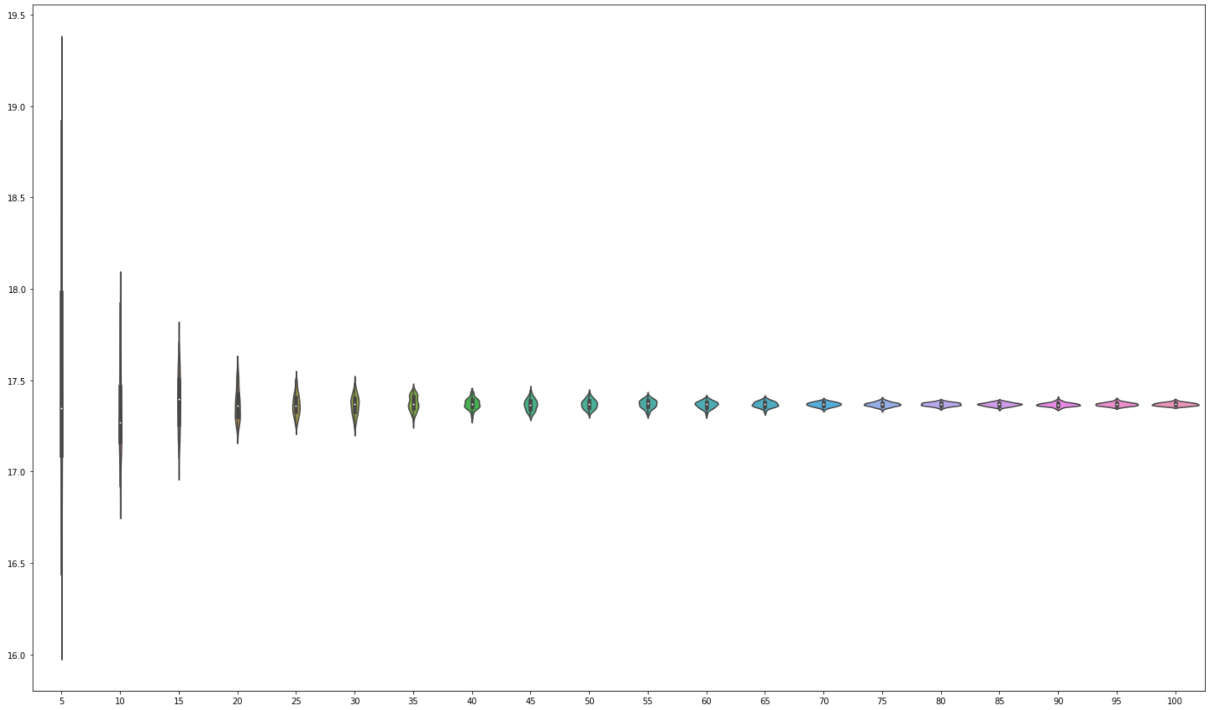


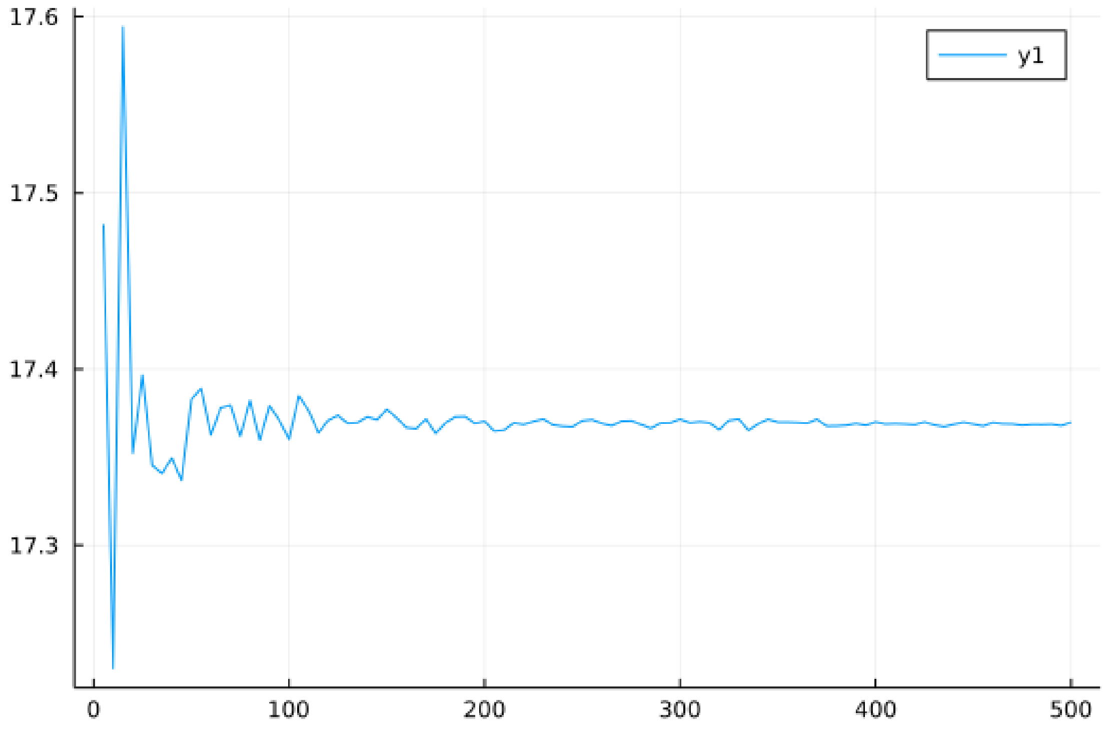
Non-smoker


12
11
10
9
8
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300

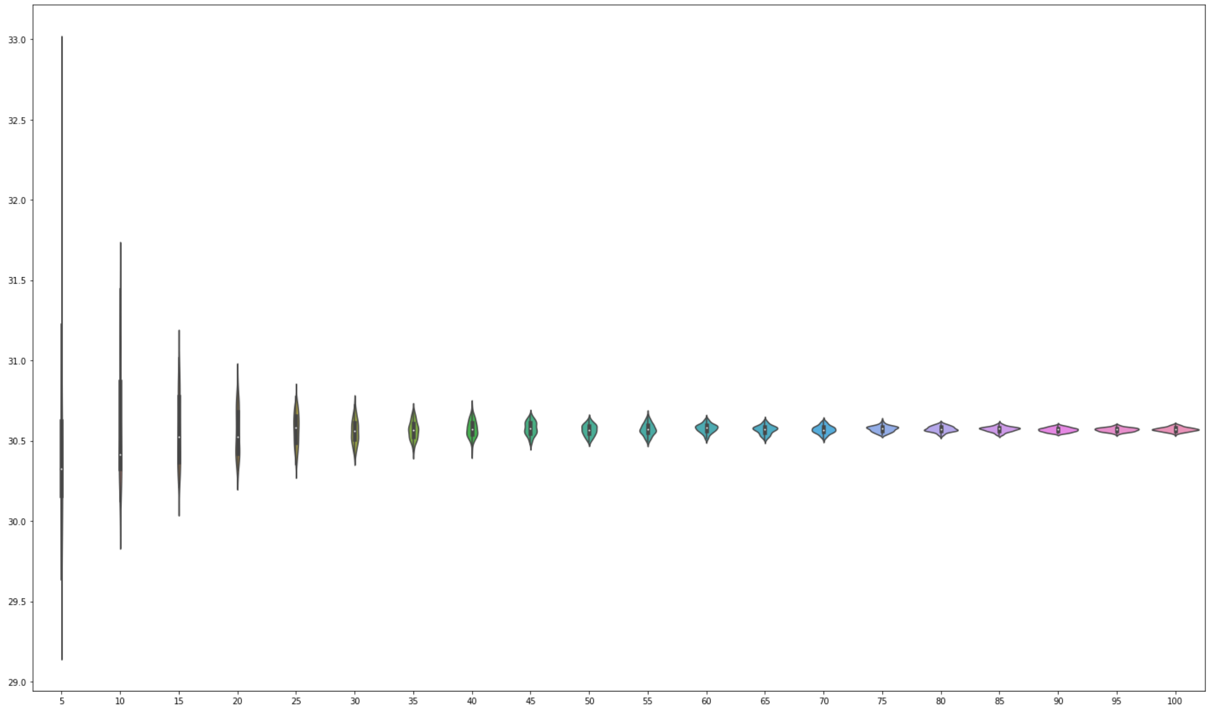

y1
10.5
10.0
9.5
9.0
0
100
200
300
400
500

Smoker non-obese

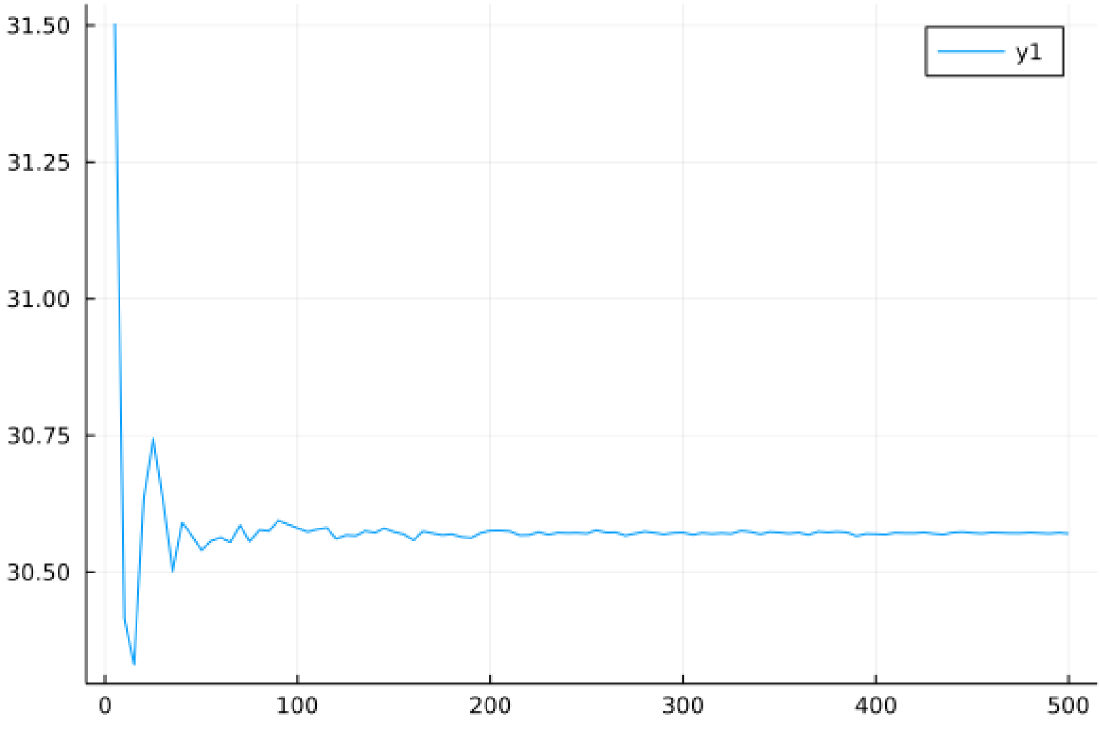




Smoker obese



33.0
32.5
32.0
31.5
31.0
30.5
30.0
29.5
29.0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100



31.50
31.25
31.00
30.75
30.50
0
100
200
300
400
500
y1