University of Colorado **Denver**

Department of Mathematical and Statistical Sciences

Analyses of water-meter accuracies with respect to meter manufacturers and flow rates

Sponsor: Water Systems Optimization

Instructor: Dr. Aimé Fournier

Kushmakar Baral

Kevin Klitchman

Laoyi Wang

Wenjuan Zhang

December 14, 2019

Contents

1 Abstract

Are residential water meters accurate enough? Most likely not, but in order to estimate the accuracy for a specific make and model of water meter, adequate testing is required. To perform these tests, it is also important to procure an adequate sample size that is representative of the population, without being too costly. To that end, on Water Systems Optimization's (WSO) behalf, we aspire to review their bootstrapping analysis and determine the minimum sample size that should be acquired. We also analyze the statistics of water-meter performance under different flow rates.

2 Introduction

The water meter's functionality is to measure the quantity of water being used in order to properly track how much money is owed for water services. Without precise measurement it is inevitable that there will be unaccounted mistakes regarding utility revenue generation, appropriate customer billing and water-demand management. According to the water audit for the City of Philadelphia, PA for the year of 2006, customer meters inaccuracy was estimated at 105.8 million gallons, costing the utility more than USD 0.5 million in revenue [1].

Even the slightest error in performance for a single water meter can add up drastically, bit-by-bit over time. As such, evaluating the performance of water meters is crucial for ensuring the resident is not being overcharged or undercharged. WSO could use the information of water meters to also improve upon relevant policies (such as replacing water meters by their model and make). Residents could also be able to know the efficiency of the money they spent, help manage their budget, make the right decision of water company and so on. Studying the performance of water meters is therefore important to determine who is losing money, and by how much, due to imperfect water-meter performance.

Since there are different models and makes of water meters, we have decided it is necessary to detect the differences among the different meters in order to help guide future decisions

---

Kushmakar: Section 2, 3.1, 3.2, 3.4, 3.6.5, 4.4, 4.6.3 and Section 5-6
Kevin: Section 3.3, 4.3 and Section 5
Laoyi: Section 3.6.4, 4.6.2 and Section 5
Wenjuan: Section 1-2, 3.1, 3.5, 3.6.1-3.6.3, 4.1, 4.2, 4.5, 4.6.1, 4.6.4, 4.7 and Section 5

regarding meter selection. In order to detect the differences, sample size determination used to detect the differences is then the next step. However, the main objectives proposed by WSO include:

- Reviewing the bootstrapping analysis they have already completed to ensure that it is statistically rigorous.
- Evaluating additional strategies to pick sample sizes to detect differences in performance between sub-populations of meters.

We will focus on these two topics in the project and also study the performance of water meters under different flow rates to get more insight into their performance.

3 Methods

3.1 Data

In the data set WSO provided, there are 12 different parameters which are `group`, `agency`, `region`, `meterID`, `size`, `make`, `model`, `age`, `throughout`, `low`, `med` and `high`. Of the 12 different parameters in the data set we decided to focus entirely on the water-meter type (`make`) and how accurately each meter was able to gauge water usage at different flow rates (`low`, `med` and `high` denoting the ratios of measured to true flow rates, so that a unit value implies perfect accuracy). One fact neglected by people is that water-meters do not always measure water flowing with a consistent rate. According to the 2016 RUE study, about 12% of water usage is from a 'low' flow rate, 34% from a 'medium' flow rate and the remaining 54% of water usage is from a 'high' flow rate. These percentages of the volume of water at certain flow rates were used to calculate a new column labeled `wgtAve` for each water-meter in the data set by formula `wgtAve`=`low`*0.12+`med`*0.34+`high`*0.54. For this data set, there are 16 different `make`s. To ease the usage of the data set later, define `wgt_A`, `wgt_C`, `wgt_F` …to be the `wgtAve` values for the corresponding `make` A, C, F …which all 12 have larger sample size compared with `make` B, D, E, and M.

3.2 Bootstrapping

WSO having done some bootstrapping analysis with the provided data set, and one of our main focuses will be recreating their bootstrapping analysis in Python code and then determining if it is statistically rigorous. Bootstrapping is a re-sampling technique where a large number of samples of smaller sizes are repeatedly drawn from a data set with replacement. With bootstrapping, we can generate a much larger usable data set from a smaller data set in order to perform more confident analysis than would be possible with just the original data set. Figure 1 shows the basics of bootstrap algorithm, where $\mathbf{X} = (x_1, x_2, ..., x_n)$ is the data set, $\mathbf{X}^{*1}, \mathbf{X}^{*2}, ..., \mathbf{X}^{*B}$ are the bootstrap samples, and bootstrap replicates $s(\mathbf{X}^{*1}), s(\mathbf{X}^{*2}), ..., s(\mathbf{X}^{*B})$ are obtained by calculating the values of the statistic s($\mathbf{X}$) on each bootstrap sample. In

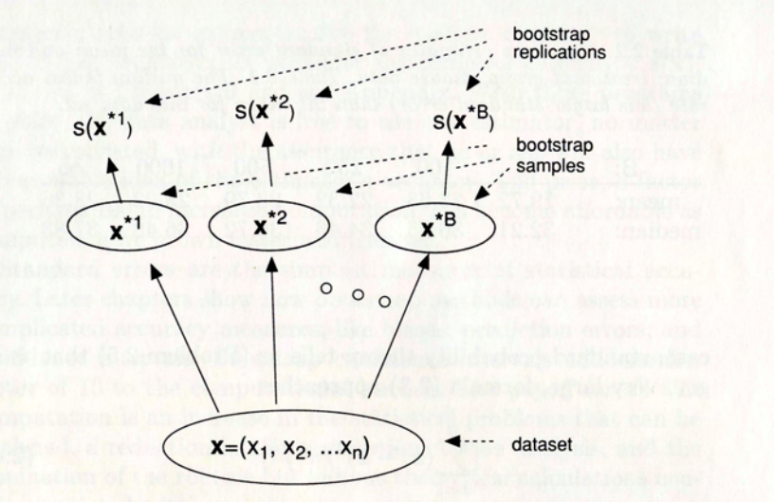

Figure 1: Bootstrap Algorithm [3]

this case, we take our dataset $\mathbf{X}$ and randomly draw samples with replacement which are our bootstrap samples $\mathbf{X}^{*1}, \mathbf{X}^{*2}, ..., \mathbf{X}^{*B}$, then we get our statistics on each of those bootstrap samples which are $s(\mathbf{X}^{*1}), s(\mathbf{X}^{*2}), ..., s(\mathbf{X}^{*B})$. However, it is necessary to understand that each of the B samples has a certain size much less than n. As we increase the size of the sample, the sampling distribution of the mean would become more normal. However, it is not ideal to choose the sample size very large because it will change the distribution of the original data drastically. Finally, we have bootstrap distribution from those statistics of those bootstrap samples.

In the bootstrapping analysis in this report, 95% confidence limits indicate the interval where the left bound is the 2.5% quantile and the right bound is the 97.5% quantile of bootstrap

samples $s(\mathbf{X}^{*1}), s(\mathbf{X}^{*2}), ..., s(\mathbf{X}^{*B})$.

3.3 Jackknife

The Jackknife is a leave-one-out re-sampling technique that is able to calculate the estimate and bias of a parameter. The method works by systematically ignoring only one observation of the data and calculating the parameter of interest with the remaining n-1 values. Then, it takes an average of collection of estimates. Let $(x_1, ..., x_n)$ be the dataset, θ be the mean that we want to estimate from the data, $\hat{\theta}$ be the estimate of the mean based on the entire dataset, $\hat{X}_i$ be the estimate of θ obtained by deleting observation x_i. Then, the mean of the $\hat{\theta}_i$ is calculated by

$$\bar{\theta}\frac{1}{n}\sum_{i=1}^{n}\hat{\theta}_i \,,$$

where $\bar{\theta}$ is the jackknife estimate of the parameter of interest. By doing this method compared with more standard direct calculations of the parameter of interests, $\bar{\theta}$ will be an estimate of the parameter with less bias. By comparing the jackknife estimate of the parameter of interest and the parameter of interest calculated directly over the whole sample, the estimate of the bias of the parameter of interest can be directly calculated by

$$\widehat{\text{bias}}_{(\theta)} = (n-1)(\bar{\theta} - \hat{\theta}) \,.$$

3.4 Cross Validation

Cross-Validation is primarily a way of measuring the predictive performance of a statistical model. There are several different kinds of cross-validation methods such as: Leave-One-Out (LOOCV), K-fold Cross-Validation, and stratified K-fold Cross-Validation. LOOCV is a type of cross-validation where only one data point is tested for validation, while all the other data are used for training the model. It is stable and nice in the sense that you will be testing all the data points individually, but it will take forever if you have a large size of data points. K-fold Cross-Validation is a type of cross-validation where your data are divided into k parts of some equal size and $k-1$ of the parts are used for training the model and 1 for testing. We repeat the procedure k times, rotating the designation of the test set. Then, we determine

an expected performance based on the results across the iterations. Similarly, stratified k-fold cross-validation is almost the same as k-fold cross-validation, but the only difference is that in stratified k-fold, we rearrange the data in a way that each fold has a good representation of the whole dataset. We will be using k-fold cross-validation because the variance of the resulting estimate is reduced as k increases, and we will have reduced bias [13]. We will try to fit the approximated sampling distribution of the mean created by WSO with a Weibull distribution. Then, we will try to cross-validate the parameters of Weibull distribution with the bootstrapping result. Weibull distribution is a continuous probability distribution named after Swedish Mathematician Waloddi Weibull. The standard parameterization of the probability density function of a Weibull random variable is

$$f(x;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k}, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

where $k > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution. This distribution is widely used in reliability engineering and life data analysis due to its versatility.

Then, we will use Kullback-Leibler (KL) divergence to compare the density functions generated from testing and training sets of the `bootResult`. KL divergence, sometimes also called relative entropy, is a measure of how one probability distribution is different from a second [14]. Since we will be using standard kernel-density estimates for testing the sets and Weibull distribution for training the sets, we will then compare these two distributions using KL divergence.

3.5 Decomposition of Total Variance

3.5.1 ANOVA

ANOVA is a test which helps find out if there is a significant difference between testing groups. We will use ANOVA to test if there is a significant difference of the performance of water meters among different `make` groups. Since bootstrapping analysis returns a confidence interval for the performance of the whole population, so we would like to know "Is this confidence interval representative for all the water meters?" Thus, to know whether the confidence interval is representative or not, we would like to test the performance of each subgroup with different

`make`s. The tool to test the differences among subgroups is one-way ANOVA due to the single factor `make` we are interested in. If there is a significant difference, applying bootstrapping analysis to each subgroup to generate its confidence interval would be preferable.

3.5.2 Internal & External Variances

To study the performance of water meters with different `make`s, we decompose the total variance into internal variance and external variance. By analyzing these two variances, we can get more insight of the performance of water meters among different `make` groups.

Let $I = \sum_{j=1}^{J} I_j$ be the total number of data and $y_{i,j}$ be the ith datum out of I_j data in the jth out of J categories e.g., `make`. Then the mean of category j is $m_j = \frac{1}{I_j}\sum_{i=1}^{I_j} y_{ij}$ and the mean over all categories is

$$m = \frac{1}{I}\sum_{j=1}^{J}\sum_{i=1}^{I_j} y_{ij} = \sum_{j=1}^{J}\frac{I_j}{I} m_j \, .$$

The variance of category j is

$$s_j^2 = \frac{1}{I_j - 1}\sum_{i=1}^{I_j}(y_{ij} - m_j)^2 \, ,$$

and the total variance can be decomposed as

$$s^2 = \frac{1}{I-1}\sum_{j=1}^{J}\sum_{i=1}^{I_j}(y_{ij} - m)^2 = \sum_{j=1}^{J}\left(\frac{I_j - 1}{I-1}s_j^2 + \frac{I_j}{I-1}(m_j - m)^2\right) \, , \tag{3.1}$$

where the 1st and 2nd summand terms represent the internal and external variance of category j. *

3.6 Sample-size determination

In this section, we will use α as type-I error, β as type-II error, δ (> 0) as the smallest difference in means that people regard as being important to be able to detect.

*Proof of (3.1) can be found in the Appendix

3.6.1 Power-based sample-size determination - Formula I

We will be connecting sample-size determination with the power of hypothesis testing. Assume there are two independent populations, X and Y, μ_1 and μ_2 being the means of X and Y respectively. In this background, the null hypothesis and alternative hypothesis are:

$$H_0 : \mu_1 - \mu_2 = 0 \, , \qquad H_1 : \mu_1 - \mu_2 = \delta \, .$$

For this hypothesis testing, we choose $\overline{X} - \overline{Y}$ as the statistic where $\overline{X}$ and $\overline{Y}$ are sample means of X and Y. According to the Central Limit Theorem, as sample sizes get larger the distribution of $\overline{X} - \overline{Y}$ will approach a normal distribution.

The power $(1-\beta)$ of a hypothesis testing is the probability of making the correct decision if the alternative hypothesis is true [5]. It can be shown that the power is positively correlated with δ and sample size, and negatively correlated with the standard deviation (σ) and significance level (α). The following formula specifically describes the relations of these several factors [†],

$$Z_{1-\beta} = \frac{\delta}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{rn_1}}} - Z_{1-\alpha/2} \, .$$

where σ_1, σ_2 are the standard deviations of the populations X and Y, n_1 is the size of the smaller sample, r is the ratio of larger sample to smaller sample. After some simple operations, we can get our first formula of sample size:

$$n_1 = \frac{r\sigma_1^2 + \sigma_2^2}{r} \cdot \frac{(Z_{1-\beta} + Z_{1-\alpha/2})^2}{\delta^2} \, . \tag{3.2}$$

3.6.2 Power-based sample-size determination - Formula II

In [7] and [11], the authors summarized the method of sample-size determination based on power for most commonly used α and β values:

$$n = f(\alpha, \beta) \cdot \frac{2\sigma^2}{\delta^2} \tag{3.3}$$

[†] Z_r is the value which makes $P(X < Z_r) = r$ where X follows standard normal distribution.

where $f(\alpha, \beta)$ is a value calculated from α and β, σ is the standard deviation of two populations (with the same standard deviation). Table 1 lists the $f(\alpha, \beta)$ for most commonly used α and β values,

	β			
α	0.05	0.1	0.2	0.5
0.05	13.0	10.5	7.9	3.8
0.01	17.8	14.9	11.7	6.6

Table 1: $f(\alpha, \beta)$ Table

3.6.3 Precision-based sample-size determination - Formula III

Suppose you want to be able to estimate your unknown parameter ($\mu_1 - \mu_2$ here) with a certain degree of precision. What you are essentially saying is that you want your confidence interval to be a certain width [11]. Since $1 - \alpha$ confidence interval of $\mu_1 - \mu_2$ is

$$\left(\bar{X}_1 - \bar{X}_2 - Z_{1-\alpha/2} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}\,, \quad \bar{X}_1 - \bar{X}_2 + Z_{1-\alpha/2} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \right)$$

The corresponding width is

$$Z_{1-\alpha/2} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

where σ_1, σ_2 are the standard deviations of populations X and Y, and n_1, n_2 are the sample sizes of these two populations.

If we apply the same notation used above, treat n_1 be the size of the smaller sample and r be the ratio of larger sample to smaller sample, we have

$$Z_{1-\alpha/2} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{rn_1}}$$

If we want the width less than value v, set

$$Z_{1-\alpha/2} \cdot \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{rn_1}} = v$$

Solve for n_1, the smallest sample size would be

$$n_1 = \frac{r\sigma_1^2 + \sigma_2^2}{r} \cdot \frac{Z_{1-\alpha/2}^2}{v^2} \tag{3.4}$$

3.6.4 Cohen's D-based sample-size determination - Formula IV

Cohen's d is a quantitative measure of the difference between two means. The formula is

$$d = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{(n_1-1)\cdot s_1^2 + (n_2-1)\cdot s_2^2}{n_1+n_2-2}}} ,$$

where n_1 and n_2 are the total sizes of two populations, and s_1^2 and s_2^2 are the sample variances of the two populations. The denominator of d is defined to be the pooled standard deviation [12]. Cohen's d is used to determine whether the effect-size value is small, medium or large between the two groups. Applying Cohend's d, the formula for determining the sample size for each group is

$$n = \frac{(Z_{1-\alpha/2} + Z_{1-\beta})^2}{d^2} , \tag{3.5}$$

where d is Cohen's d mentioned above.

3.6.5 Yamane Taro sample-size determination - Formula V

This method for sample-size calculation in this subsection was formulated by the statistician Taro Yamane in 1967 to determine the sample size from a given population. Following is the mathematical formula for the Taro Yamane's method:

$$n = \frac{N}{1 + Ne^2} , \tag{3.6}$$

where n signifies the sample size, N signifies the known population, and e signifies the margin error. Margin of error tells us how many percentage points your results will differ from the real population value. Margin of error here is based on 95% confidence level and the use of the level of maximum variability ($P = 0.5$) in the calculation which generally produce a more conservative sample size, that is, a larger one [16]. Thus, we will use the standard margin of errors that Taro's formulation suggested: 3%, 5%, 7% and 10% to determine the sample sizes

for different `makes`.

4 Results

4.1 Data

By using `values_counts()` command, we can get sample sizes for each `make` from Figure 2.

```
# List sample sizes corresponding to different makes
df['make'].value_counts()
```

executed in 13ms, finished 18:25:45 2019-11-04

```
J          1511
C          1069
N           358
Unknown     332
K           299
O           267
H           137
G           104
F            97
I            93
L            66
A            58
D             8
E             5
B             3
M             1
Name: make, dtype: int64
```

Figure 2: Sample size by make

4.2 Bootstrapping Result

The following Figure 3 and 4 recreate the results using Python.

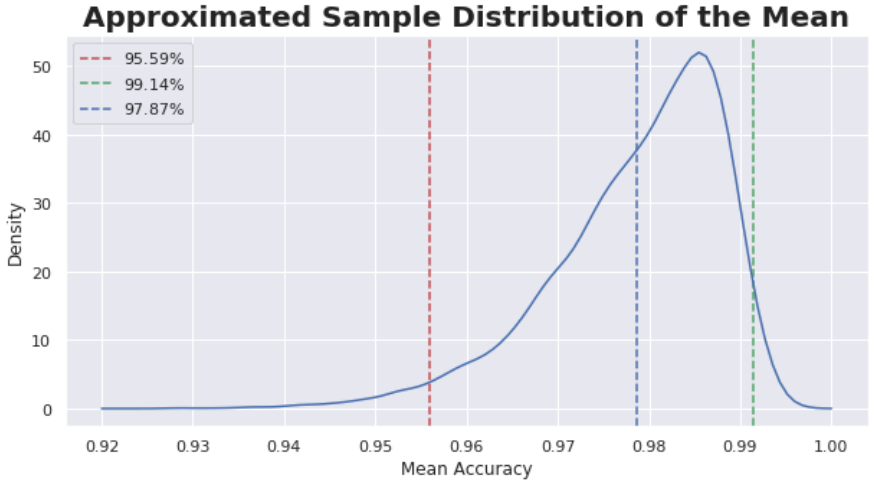

Figure 3: Approximated Mean distribution using Python. Red dashed curve indicates the 2.5% qunantile of bootstrapping means; green dashed curve indicates the 97.5% quantile of bootstrapping means; blue dashed curve indicates the mean of boostrapping means; solid blue curve is the Kernel-Density Estimation of bootstrapping means.

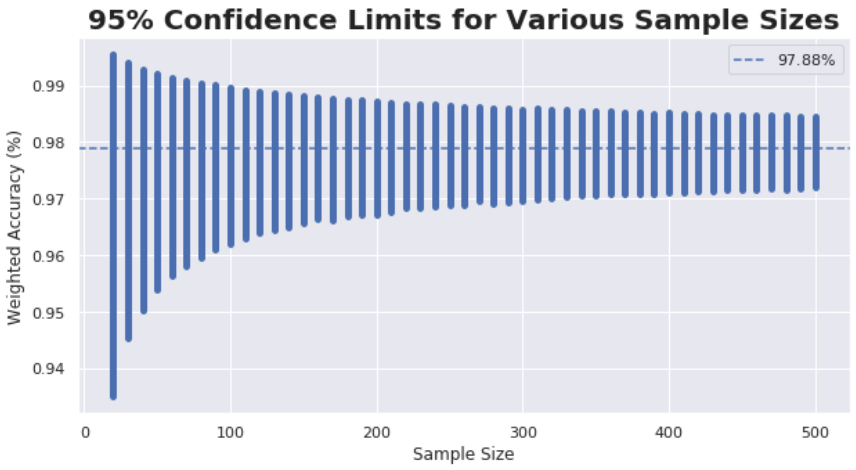

Figure 4: Confidence Limits using Python. Blue vertical bars indicate 2.5% quantile to 97.5% quantile confidence limits using bootstrapping means with the corresponding sample size on the x-axis; blue dashed line indicates the mean of boostrapping means.

4.3 Jackknife Result

Figure 5-8 show the results of applying Jackknife method.

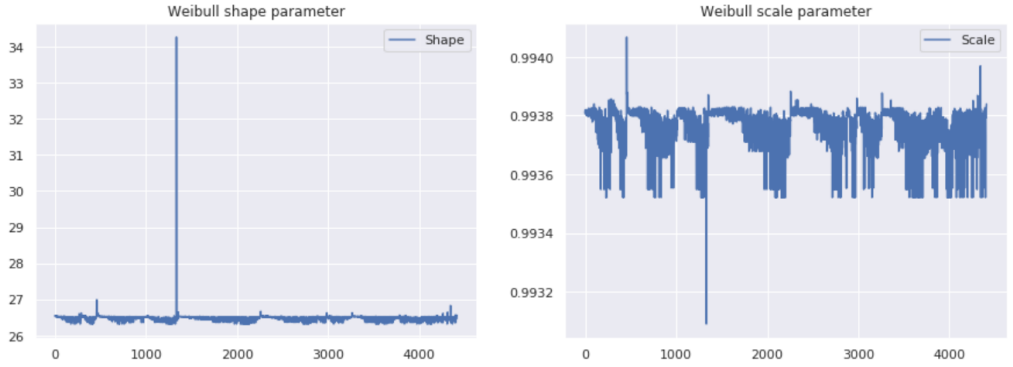

Figure 5: Apply jackknife method to the wso dataset to plot Weibull distribution parameter 'shape' and 'scale' for each sample in the dataset.

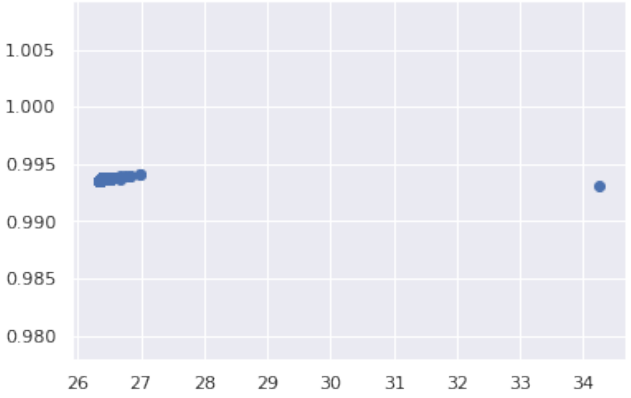

Figure 6: Scatter plot of all the Weibull distribution parameters.

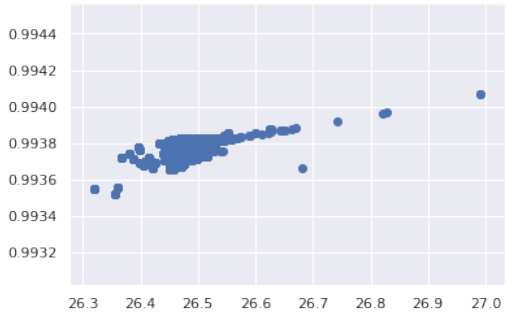

Figure 7: Scatter plot of all the Weibull distribution parameters without the outlier.

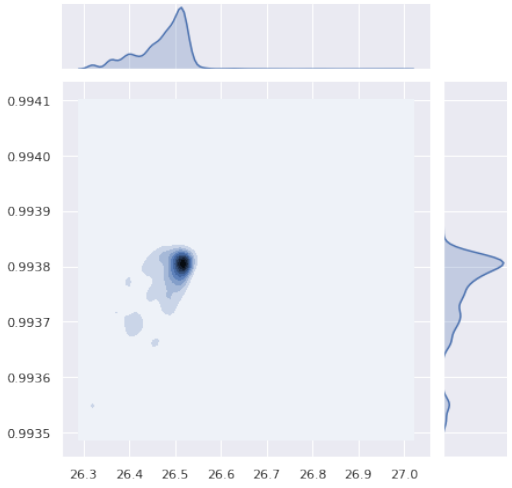

Figure 8: Kernel-Density Estimate plot of the Jackknife Weibull parameters.

4.4 Cross Validation Result

The following Figure 9 and 10 display how Weibull distribution is fitted with `bootResult`.

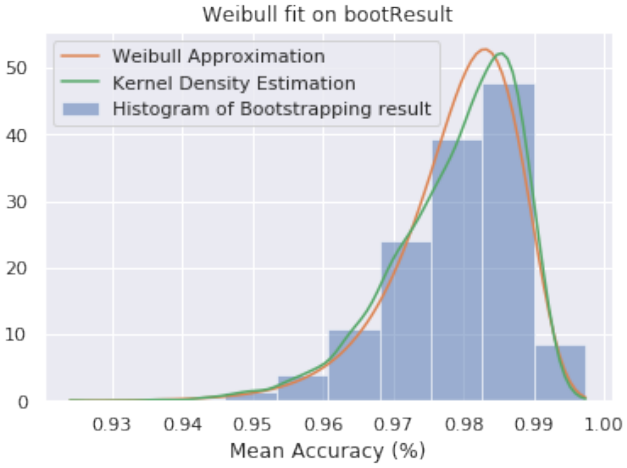

Figure 9: Fitted Weibull distribution on bootResult and (orange curve) with the kernel density estimates of bootstrapping result (green curve).

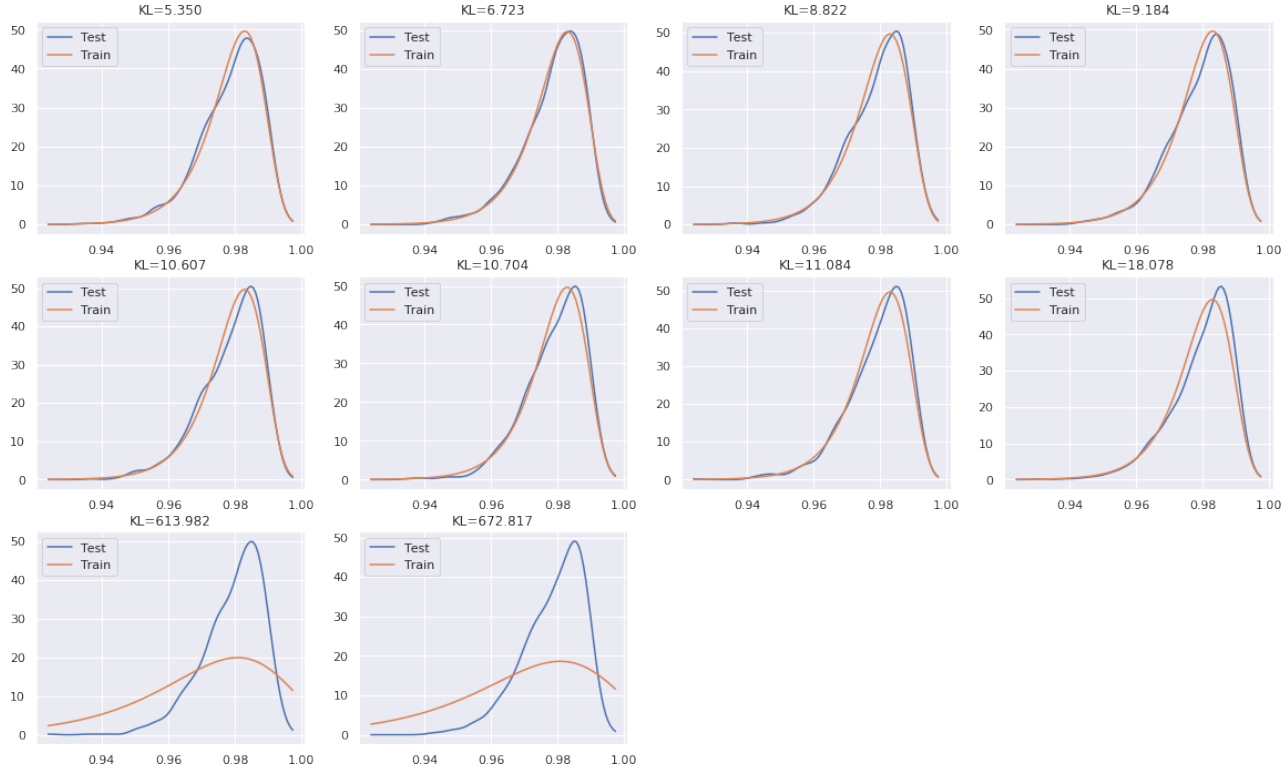

Figure 10: 10-fold cross-validation on the bootResult using kernel-density estimate curve for testing one fold of the data and training nine folds of the data using 2-parameters Weibull distribution.

4.5 Decomposition of Total Variance result

4.5.1 ANOVA result

To understand the data, boxplot of `wgtAve` categorized by `make` is revealed from Figure 11. The test results related to ANOVA are shown in Figure 12-13.

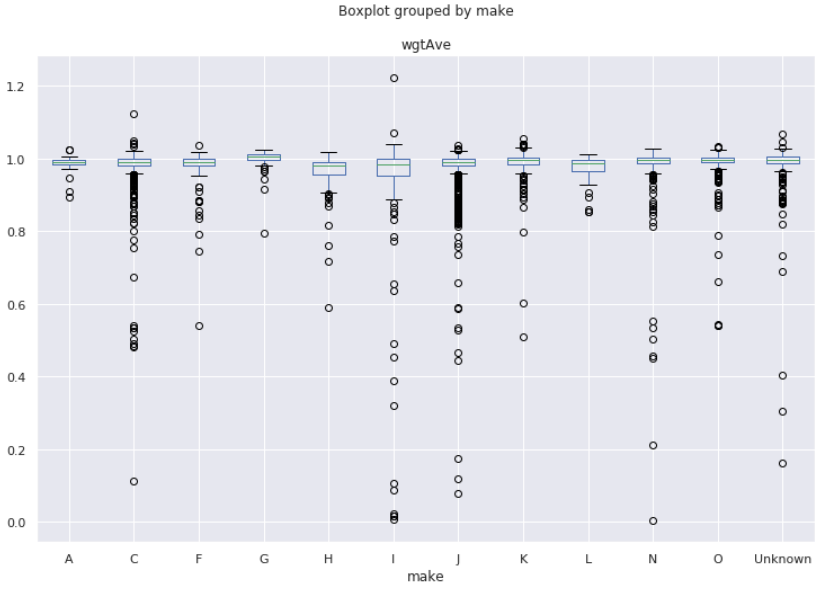

Figure 11: Boxplot of wgtAve by make

```
# One-way ANOVA
statistic0, pvalue0 = stats.f_oneway(wgt_A, wgt_C, wgt_F, wgt_G, wgt_H, wgt_I,\
                                     wgt_J, wgt_K, wgt_L, wgt_N, wgt_O, wgt_U)
print('{:.10f}'.format(statistic0))
print('{:.10e}'.format(pvalue0))
```

executed in 10ms, finished 20:06:41 2019-12-10

```
17.3049066366
4.6022996378e-34
```

Figure 12: ANOVA Result

```
statistic1, pvalue1 = stats.levene(wgt_A, wgt_C, wgt_F, wgt_G, wgt_H, wgt_I,\
                                   wgt_J, wgt_K, wgt_L, wgt_N, wgt_O, wgt_U)
print('{:.10f}'.format(statistic1), '{:.10e}'.format(pvalue1))
```

executed in 11ms, finished 20:06:41 2019-12-10

```
19.3366685315 1.6927420477e-38
```

```
statistic2, pvalue2 = stats.shapiro(results.resid)
print('{:.10f}'.format(statistic2), '{:.10e}'.format(pvalue2))
```

executed in 16ms, finished 20:06:41 2019-12-10

```
0.3626574278 0.0000000000e+00
```

Figure 13: Assumption Check for ANOVA

4.5.2 Internal & External Variances result

To show the relation of internal, external variances with respect to different `makes`, Figure 14 shows the plots of $\frac{I_j-1}{I-1}\frac{s_j^2}{s^2}$ and $\frac{I_j}{I-1}\frac{(m_j-m)^2}{s^2}$ with an abscissa label for each `make`.

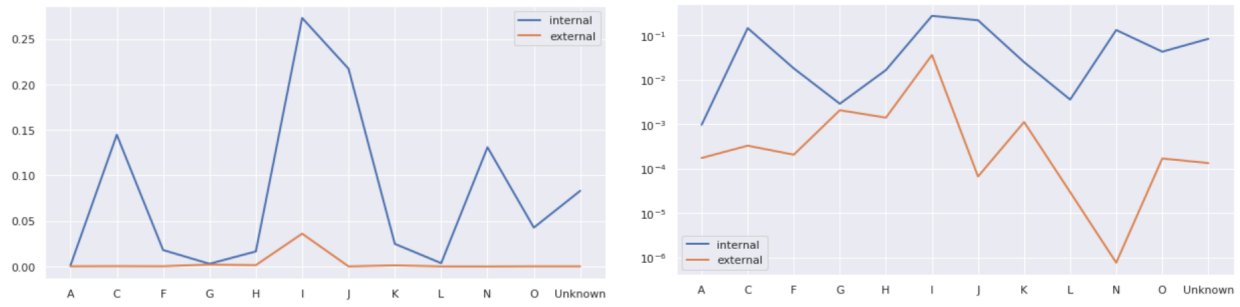

Figure 14: Internal & external Variances

4.5.3 Bootstrapping for each group

Instead of applying bootstrapping method on the whole data set to generate 95% confidence limits, Figure 15 shows 95% confidence limits for each `make`.

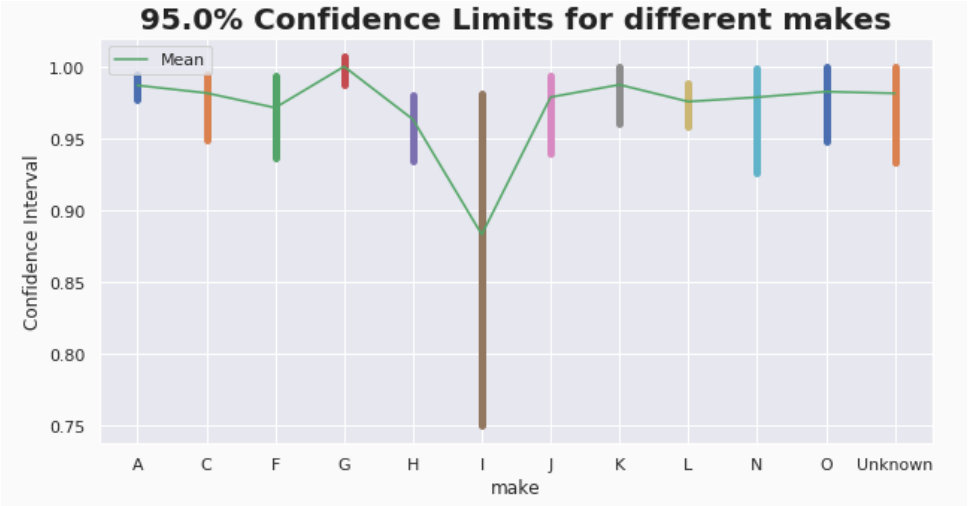

Figure 15: Confidence intervals for each group

4.6 Sample-size determination result

4.6.1 Formula I

The quantity of δ is highly dependent on the problem. Based on the type of problem, the sponsor can decide the smallest difference in performance of water meters that they regard as being important to be detected. Figure 16 demonstrates the relation between sample size and factors mentioned above.

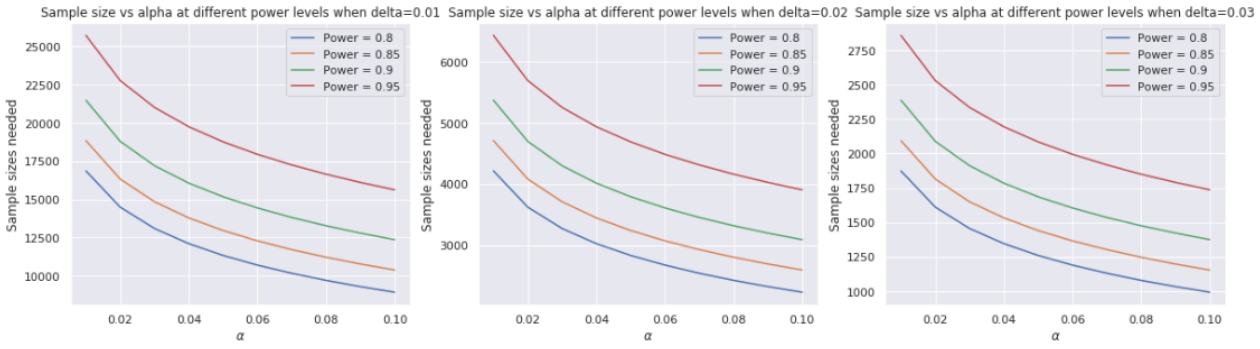

Figure 16: Sample Size using Formula I

4.6.2 Formula IV

Figures 17-19 show results using Formula IV.

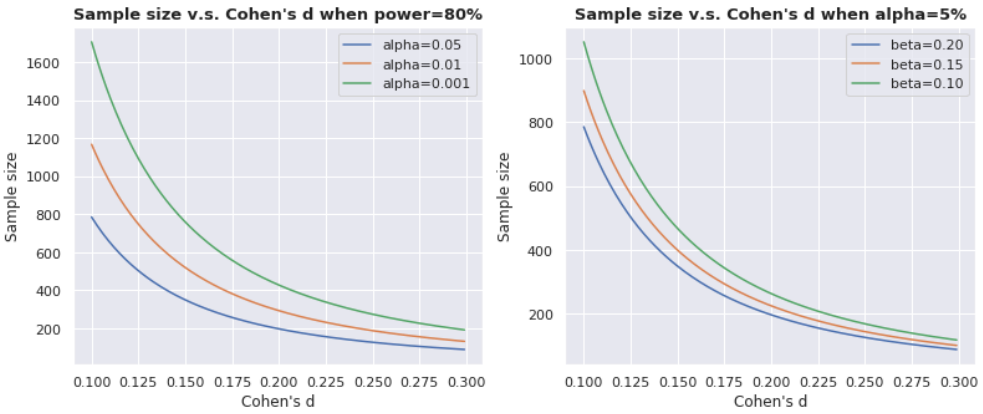

Figure 17: Sample size determine using Cohen's d

	A	C	F	G	H	I	J	K	L	N	O	Unknown
A		0.103	0.286	0.555	0.518	0.499	0.128	0.024	0.393	0.099	0.08	0.074
C			0.17	0.347	0.324	1.066	0.029	0.124	0.107	0.043	0.02	0.006
F				0.576	0.143	0.463	0.137	0.325	0.068	0.078	0.174	0.136
G					0.849	0.641	0.362	0.303	0.826	0.263	0.328	0.271
H						0.462	0.286	0.526	0.253	0.182	0.329	0.262
I							1.136	0.782	0.453	0.654	0.689	0.7
J								0.149	0.074	0.019	0.048	0.033
K									0.294	0.128	0.104	0.099
L										0.035	0.124	0.088
N											0.05	0.038
O												0.011
Unknown												

Figure 18: Cohen's d values among subgroups

	A	C	F	G	H	I	J	K	L	N	O	Unknown
A		739	97	26	30	32	476	13685	51	797	1224	1423
C			273	66	75	7	9337	514	690	4233	18862	201966
F				24	385	37	417	75	1702	1304	261	428
G					11	20	60	86	12	114	73	108
H						37	97	29	123	237	73	115
I							7	13	39	19	17	17
J								356	1431	22451	3354	7154
K									91	479	729	800
L										6473	513	1016
N											3095	5304
O												62830
Unknown												

Figure 19: Cohen's d sample sizes among subgroups

4.6.3 Formula V

Figures 20-21 represent the sample-size determination using Yamane Taro's Formula for each makes with different levels of margin of errors and for the whole population without considering the `make` types.

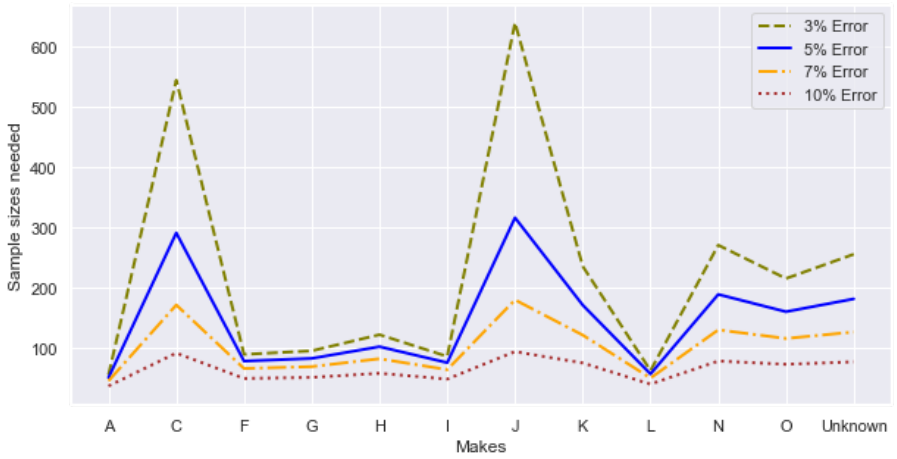

Figure 20: Sample sizes needed for different margin of errors (3%, 5%, 7%, and 10%) for 12 different makes.

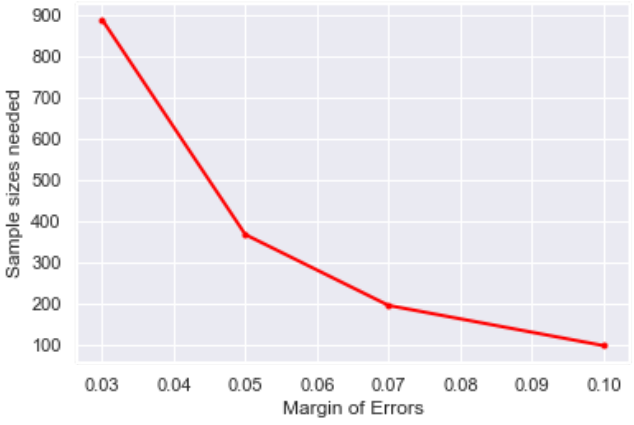

Figure 21: Sample sizes needed for different margin of errors (3%, 5%, 7%, and 10%) for the whole population where $N = 4408$.

4.6.4 Test sample size formulas

Functions `comparison(i,j,Size,delta,replace)` and `comparison_percent(i,j,Size, delta,n_trial,replace)` are defined in the notebook to test sample-size determination formulas mentioned above. Here we will take Formula 1 as an example and apply it to two populations both following Weibull distribution.

```
group1 = np.random.weibull(5,int(1e6))
group2 = np.random.weibull(10,int(1e6))

var1 = np.var(group1)
var2 = np.var(group2)

n_w = size1(1, var1, var2, delta=0.03)
print(n_w)
```

executed in 231ms, finished 17:01:45 2019-12-11

```
500
```

Figure 22: Formula I is applied to Weibull data

```
comparison_percent(group1, group2, n_w, 0.03, 1000, replace=False)
```

executed in 44.1s, finished 17:06:47 2019-12-11

```
0.617
```

```
comparison_percent(group1, group2, n_w, 0.02, 1000, replace=False)
```

executed in 43.6s, finished 17:07:31 2019-12-11

```
0.899
```

```
comparison_percent(group1, group2, n_w, 0.06, 1000, replace=False)
```

executed in 43.9s, finished 17:08:15 2019-12-11

```
0.003
```

Figure 23: Verification of Formula I when using Weibull data

Then we test Formula I on our dataset. `comp_percent_make(i,j,Size,delta,n_trial=1000)` is defined similarly to `comparison_percent(i,j,Size,delta,n_trial,replace)`. The only difference is that `comp_percent_make` is defined to test `make` i and `make` j.

```
comp_percent_make('J', 'C', ntest, 0.03, 1000)
#high means they are different by the value we proposed
```

executed in 400ms, finished 10:34:41 2019-12-05

```
0.005
```

```
comp_percent_make('H', 'I', ntest, 0.03, 1000)
#high means they are different by the value we proposed
```

executed in 422ms, finished 10:34:46 2019-12-05

```
0.98
```

Figure 24: Verification of Formula I when using WSO data with size=91

4.7 Explore flow rate result

Boxplot categorized by make under different flow rates is shown in Figure 25,[‡]

[‡]I did the Weibull fit for each make under 3 flow rates, but I do not think it is interpretable. So I choose not to put it here. But you can find it in the notebook

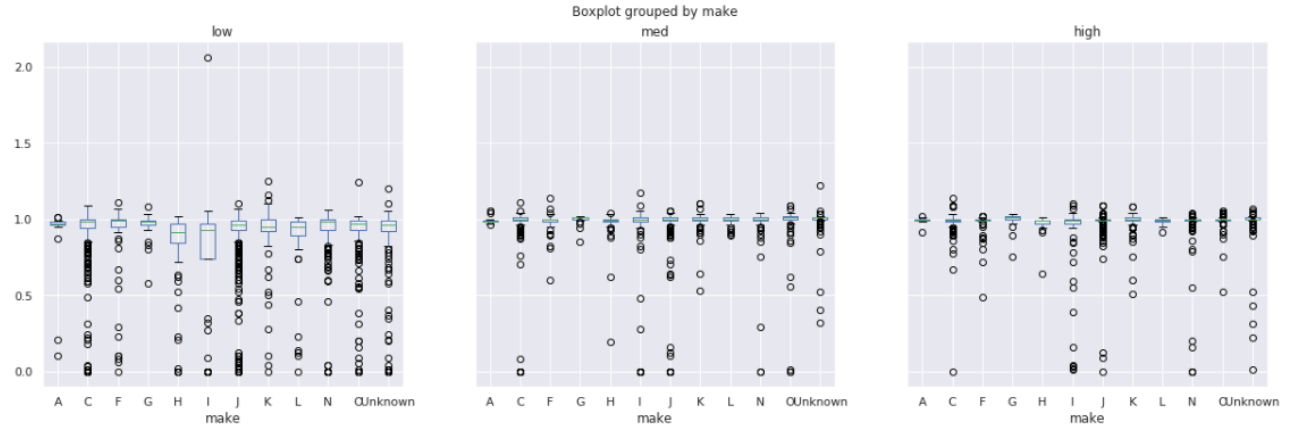

Figure 25: Boxplot categorized by make under different flow rates

5 Discussion

In Figure 2, `make`s D, E, B, and M have comparatively smaller sample sizes. Hence, when we perform the hypothesis testing we would like to exclude the `make` with small sample sizes because very small sample sizes are not representative and ideal for data analysis.

Figures 3 and 4 are the plots similar to the plots our sponsor have already generated but using Python.

Figure 5 shows the shape and scale parameter values over each iteration of the jackknife calculation for the Weibull parameter. The sharp spike indicates that one of the water meters is a potential outlier. The outlier water meter is index 1334, make I and has a low flow rate accuracy of 2.06 causing the noticeable spike in the plots.

Figure 6 is a scatter plot of the Weibull distribution parameters (shape vs scale), removing the aforementioned outlier we see in figure 7 that there is a positive linear relationship between shape and scale.

Figure 8 is a kernel density estimate plot of the parameters of the Weibull distribution shape and scale calculated from a jackknife calculation from the weighted average of the entire data set. While, the majority of the points sit clustered together, there are some points tapering behind implying that as shape decreases so would scale. The Y axis 'scale' has less variability then the X axis 'shape. Because the data set is quite large, there isn't much unexpected variability.However, any unusual observations from the jackknife calculation is interesting since it is only a point calculated from the original data set with just one datum removed. Using this method with smaller sample sizes might be able to visualize more unusual observations since

the weight of each point removed from the jackknife calculation would be more noticeable.

Figure 9 is plot of the fitted Weibull-distribution of the `bootResult` with the kernel density estimate (kde) of the `bootResult`. This figure indicates that fitting a Weibull distribution through the data would give us some information about the data and help us test with the kde distribution. This leads us to use k-fold cross-validation of the `bootResult` comparing kde and Weibull distributions.

Figure 10 is the 10-fold cross-validation plots with the KL-divergence values for each folds. We are comparing the kde and Weibull distributions using KL-divergence in each training and testing sets of the data. Eight out of the 10-fold results looks almost identical with KL-divergence value of lower than 20. But, two of the 10-fold results look totally off and have KL-divergence values almost close to 700. It is probably because these two folds most likely had the most values of mean accuracy not well distributed in the training set which blew off the Weibull distribution. This tells us that we will have to look for any influential points in the data which is probably not giving a good Weibull fit on two folds.

Figure 11 returns the boxplot catergorized by `make`. Boxplot displays the five-number summary of a set of data. The five-number summary is the "minimum", first quartile (Q1), median, third quartile (Q3), and "maximum". It shows that the medians of different `make`s are all close to 1. From the length of IQR (interquartile range), there are lots of variations among `make` B, E and I by eyeball test.

Figure 12 is the result from one-way ANOVA test for different sub-groups categoried by `make`. Here we only display 10 decimal places of the statistic and p-velue. Since the p-value is small, we need to reject the null hypothesis. That is, there is a significant difference among different `make`s. By checking the assumptions for one-way ANOVA (shown below), it turns out no assumption is satisfied. So we would need to find alternative way to test differences among different `make`s.

Figure 13 returns the result of the Levene test and the Shapiro-Wilk test where the Levene test is to check the homogeneity of variances and the Shapiro-Wilk test is to check the normality of residuals. From Figure 13, we can see that the data here satisfy neither the Levene test nor the Shapiro-Wilk test. This also implies that the result from one-way ANOVA is not trustable. The statistic from ANOVA is the ratio of MST (variance between treatment) and MSE (variance

within treatment) [15], due to the fact that the assumptions of ANOVA are not satisfied, we would like to explore another way of decomposing the total variance to know if there is a significant difference of the mean performance among all `make`s.

The left plot in Figure 14 shows that the plot of external ratios and internal ratios among different `make`s. Since the external ratios are so close to 0 from the left plot, we then make semilogy plot on the right to display more information of the tiny changes of external variances which are not reflected from the left plot. From Figure 14, it can be seen that external variances are comparatively small among different `make`s, while the majority of variability concentrates within each sub-population. This demonstrates that there is no much significant difference among different `make`s in regard to their mean values, if not their internal variability.

Figure 15 shows the 95% confidence limits for each `make`. This also shows that there is a probability of 5% or less that the true mean value of each `make` will lie outside of the above confidence limits. Instead of returning the confidence limit for all the water meters, Figure 15 returns more information of the variability among different `make`s. As we can see from Figure 15, `make` I is the sub-population with the largest variability among all the `make`s.

As we can see from Figure 16

- Sample size will increase as we decrease the significance level. This is reasonable, we always need to generate more samples if we want to make smaller type-I error.
- Sample size will increase as we increase the power level which is also reasonable.
- Sample size will increase as we decease the δ level. Student A thinks 0.03 means different, student B think 0.01 is different, then more samples will be needed to test a difference of 0.01 than a difference of 0.03.

In Figure 17, the left plot shows that when setting $\beta = 20\%$, under the same α value, the sample size needed using Formula IV is negatively related to Cohen's d; under the same Cohen's d value, as we increase α value, sample size needed using Formula IV is decreasing. The right plot shows that when setting $\alpha = 5\%$, under the same β value, the sample size needed is negatively related to Cohen's d; under the same Cohen's d value, the sample size needed is negatively related to β.

Figure 18 displays the Cohen's d values for each pair of `make`s. Figure 19 shows the sample size needed for each pair of `make`s using Formula IV.

In Figure 20, we have the sample sizes for different margin of errors for different makes. Using the Yamane's formula, we computed the samples sizes for different margin of errors for the 12 different makes where four out of sixteen makes were excluded because those four makes had data sizes of less than 10. Now, we can see in the Figure 20 that as margin of error increases, the number of sample size needed for specific makes decreases. If we accept 10% margin of error for all the makes, the sample sizes needed for each make is lower than 100. However, if we accept 7% margin of error for all the makes, the sample sizes needed for each make is lower than or equal 180. If we accept 5% margin of error for all the makes, the sample sizes needed for each make is lower than or equal to 317. Moreover, if we accept 3% margin of error for the all the makes, the sample sizes needed for each make is lower than or equal to 641.

In Figure 21, we can see that at 3% margin of error, with our population ($N = 4408$), we will need sample size of approximately 900 to predict the true mean of the population. For 5% margin of error with our population ($N = 4408$), we will need sample size of approximately 370 to predict the true mean of the population. For 7% margin of error with our population ($N = 4408$), we will need sample size of approximately 200 to predict the true mean of the population. For 10% margin of error with our population ($N = 4408$), we will need sample size of approximately 100 to predict the true mean of the population. If we accept the sampling error (margin of error) that is higher or equal to 10%, we get sample size of less than 100 from which we can say something about the whole population. However, if we want to choose sampling error (margin of error) lower or equal to 3%, we will need a sample size of 1000 or more to say something about the whole population.

`comparison(i,j,Size,delta,replace)` is defined in the notebook to get the conclusion if the means of populations i and j are different by `delta` when `Size` samples are generated from these two populations with Boolean `replace` (`replace` is either True or False). Returning True indicates that the means of two populations are different by `delta` proposed in the above function. `comparison_percent(i,j,Size,delta, n_trial, replace)` returns the percentage when running above function `n_trial` times. In Figure 22, two Weibull populations are generated, and Formula I is applied to data to get the sample size being 500. The top plot in

Figure 23, it shows that 617/1000 times we are able to detect a difference 0.03 using the sample size from the above step. As we can see, the result is a little conservative since $0.617 < 0.80$. However, it shows from the middle plot that 899/1000 times we are able to detect a difference 0.02 using the same sample size. This shows that the probability of detecting a difference δ_1 can be improved by using sample size of detecting a larger difference δ_2 ($> \delta_1$). For the bottom plot, 3/1000 times we get the conclusion that the means of these two populations are different by 0.06. This can be explained since the means are only different by 0.05 in fact.

By setting $\delta = 0.03$, $\beta = 20\%$, $\alpha = 0.05$, $r = 1$, the corresponding sample size using Formula 3.2 is 91. Take this sample size 91 as an example to detect the difference of two groups with different `make` names. If there is no significant difference more than 0.03 between two groups, using 91 samples from two groups is not enough to detect the difference since there is no significant difference in fact. The above statement can also be verified in Figure 24. Function `comp_percent_make` in Figure 24 returns the ratio of numbers of accepting the null hypothesis and the number of trials (1000 here). For the top plot in Figure 24, if `make` J and C are different by 0.03, we should have a power of 80% to detect it. Since 5/1000 (this varies caused by randomness) times we accept they are different by 0.03, our conclusion is that there is no significant difference between `make` J and C when $\delta = 0.03$. Similarly for the lower plot in Figure 24, if `make` H and I are different by 0.03, we should have a power of 80% to detect it. Since 980/1000 (this varies caused by randomness) times we think they are different by 0.03, our conclusion is that there is significant difference (at least 0.03) between `make` H and `make` I.

Besides the topics our sponsor proposed, we are also interested in studying the performance of water-meters under different flow rates. Comparing the three sub-boxplots in Figure 25, we can see that the average performance of `make`s under median and high flow rates are all close to 1. There is a large variability of the performances for different `make`s under low flow rate. And there are more outliers in low flow rate than in median and high flow rates.

To detect if there is a significant difference of the performance for each `make` under different flow rates, we apply the similar idea mentioned in Section 3.5.2. Under low flow rate, there is a lot of variability among all the `make`s. And the external variances are also higher under low flow rate than median- and high- flow rates.

6 Conclusions & Future Work

We recreated the plots from bootstrapping analysis and used 10-fold cross-validation to check the bootstrapping analysis that WSO did. We also detected the differences among different `make`s. We also proposed five different ways for sample size determination. For future work, we can probably use different mathematical techniques such as marginal error to create a model to determine the number of sample sizes needed to represent the population. That can also control the type I and type II error for better control of the sample size determination in hypothesis testing.

Appendix

Proof of equation (3.1)

Proof.

$$
\begin{aligned}
s^2 &= \frac{1}{I-1}\sum_{j=1}^{J}\sum_{i=1}^{I_j}(y_{ij}-m)^2 \\
&= \sum_{j=1}^{J}\left(\frac{1}{I-1}\sum_{i=1}^{I_j}(y_{ij}-m_j+m_j-m)^2\right) \\
&= \sum_{j=1}^{J}\left(\frac{1}{I-1}\sum_{i=1}^{I_j}(y_{ij}-m_j)^2+\frac{2}{I-1}\sum_{i=1}^{I_j}(y_{ij}-m_j)(m_j-m)+\frac{1}{I-1}\sum_{i=1}^{I_j}(m_j-m)^2\right) \\
&= \sum_{j=1}^{J}\left(\frac{I_j-1}{I-1}s_j^2+\frac{2}{I-1}(m_j-m)\sum_{i=1}^{I_j}(y_{ij}-m_j)+\frac{I_j}{I-1}(m_j-m)^2\right) \\
&= \sum_{j=1}^{J}\left(\frac{I_j-1}{I-1}s_j^2+\frac{I_j}{I-1}(m_j-m)^2\right).
\end{aligned}
$$

The cross term in the second to last step vanishes because $\sum_{i=1}^{I_j}(y_{ij}-m_j)=0$ by the definition of m_j. □

References

[1] Harrison E. Mutikanga *Residential water-meter selection using the analytical hierarchy process*. National Water and Sewerge Corp., Kampala, Uganda, 2014.

[2] Bootstrapping Wikipedia

[3] Bradley Efron, Robert J. Tibshirani. *Introduction to the Bootstrap*. CRC Press LLC, 1994.

[4] Jason Brownlee. *A Gentle Introduction to k-fold Cross-Validation*

[5] Definition of Power

[6] Ralph E. Steve H. Rajasekhar R. *Sample Size Determination*

[7] Nick Fox, Amanda Hunn, Nigel Mathers. *Sampling and sample size calculation*

[8] Theresa A Scott. *Sample Size Planning, Calculation, and Justification*

[9] William D. Dupont, Walton D. Plummer, Jr. *Power and Sample Size Calculation*

[10] Ayumi Shintani. *Sample Size Estimation and Power Analysis*

[11] Rosie Cornish. *Statistics: An introduction to sample size calculations*

[12] Rosnow, Ralph L, Rosenthal, Robert. *Computing contrasts, effect sizes, and counternulls on other people's published data: General procedures for research consumers.*, 1996.

[13] Prashant Gupta. *Cross-Validation in Machine Learning*

[14] Kullback-Leibler Divergence

[15] ANOVA Wikipedia

[16] Glenn D. Israel. *Determing Sample Size*