

How to Build a Data Science Cost Estimate

with RRRrrr Studio

Jeremy Eden

Booz | Allen | Hamilton

CONSULTING | ANALYTICS | DIGITAL SOLUTIONS | ENGINEERING | CYBER

**2019 International Cost Estimating and Analysis Association
(ICEAA) Professional Development & Training Workshop**

Tampa, FL

May 14 - 17, 2019

Abstract

ICEAA's effort to expose the community to data science concepts and advantages to cost estimators has been popular. However, you want to know how to apply tools and techniques for creating an estimate NOW. Well prepare to create your very first data science cost estimate. There will be code, there will be data, and they will be for ye to plunder and take as your own because "not all treasure is silver and gold mate."

In this paper I will be providing guidance on how to create a cost estimate using data science techniques and tools. My presentation last year on data science applications for cost estimating received a very positive response. However, there are often some criticisms that many concepts are offered at the conference, but they are never followed up on with information for practical applications. I will be providing a step-by-step guide on how to complete an estimate with a basic real-world example. Included will be the data source for harvesting the cost data, the code used in a popular tool to capture the data, explanation of gap filling and sorting of the data, and finally example code for calculation and display of results in a second popular tool (R Studio).

Table of Contents

Abstract.....	2
Table of Contents.....	3
Executive Summary.....	4
1. Introduction.....	5
2. Cost of Living Estimator Tool.....	6
2.1. Introduction and Scenario.....	6
2.2. Data Collection Source.....	6
2.3. Data Collection Capture.....	7
2.4. Data Filtering and Cleanup.....	8
2.5. Data Analysis.....	9
2.6. User Interface with RStudio and Shiny.....	11
2.7. Data Pipelining.....	16
3. Your Own Data Science Cost Tool and Closing Remarks.....	17
4. Appendix I: Data Science Tool Resources.....	18
5. Appendix II: Sources.....	18

Executive Summary

The cost estimating industry has relied on many of the same analysis tools and strategies for decades. Advances have focused on different ways to conduct the math involved with estimates using innovative formulas and creating customized methodologies for new technology (i.e.. virtual computing). New approaches to creating cost estimates themselves have not been pursued very often and when they are, they are not always received well by those comfortable with tried and true tools and methods. While not a full replacement for these tools and methods, an opportunity now exists for the cost estimating community and ICEAA to benefit from the advantages of data science.

The “data science revolution” has impacted nearly every industry and has often proven to be a more aggressive disruptor of technical fields than those of a non-technical nature.

Data science cost estimating techniques and user interface tools combined with the availability and performance of modern computing power and cost data provides many advantages:

- Data science tool interfaces are easily written by cost estimators and tool developers with little extra training required
- Source data can be provided in continuous pipelines allowing near real-time analysis of fresh data at any time
- With little effort, interfaces can be written to create customized user experiences and collaborative environments for support of individual analyses for the same sets of data

Data science cost estimating can provide an exceptional user experience to nearly all cost analyses with a level of sophistication that has not otherwise been available using traditional methods.

1. Introduction

As data science techniques and tools become a part of everyday life in nearly all professional disciplines, many are frustrated when the cost estimating field is unable to provide the same speed, accuracy, and user experience that has become expected in other industries. For years the effort and focus of cost estimators has been on improving the methodologies of traditional cost estimating techniques for increased precision, instead of pushing the boundaries of the techniques themselves for better overall accuracy. Concerns about the availability of data, its validity, and unfamiliar scripting and statistical tools have caused many cost estimators to avoid data science and miss out on the advantages it provides.

The focus and effort should be on those advantages which enable cost estimators, end users, and decision makers to navigate and leverage more data, faster, more accurately, and more efficiently. Considerable effort is expended year after year to research model design and only to have them increase the precision of a cost estimate by the smallest fractions of a percent. Some have concluded that there is a limit to the speed and accuracy which estimates can be. However, it should be recognized that, aside from the computer revolution, there has never been a greater opportunity to grow the cost estimating profession. I value the effort that ICEAA has put into training cost estimators and understand that resources for new training materials and demonstrations of every new technology are limited. It is my hope that this paper will provide specific actionable insight and resources for implementing new techniques using data science, convince skeptics that such growth is worthwhile to achieve, and serve as a draft curriculum for ICEAA to utilize as data science cost estimating training.

This paper will show how better user interfaces and faster capture/analysis of data sets can be created so they may be used more effectively by end users and decision makers. Some new skills are necessary to capture this data, conduct these analyses, and create these interfaces, but in most cases cost estimators already have the mindset and analytical ability to learn these skills quickly. Presented first will be the rapid capture of raw data from an online database. Next, the organization and analysis of the data will be conducted. Finally, an interface will be built for an end user to access the analyzed data. This exercise will be accomplished using three popular data science tools. You will be provided the code for these tools with the hope that you will attempt your own data science cost estimate soon.

2. Cost of Living Estimator Tool







2.1. Introduction and Scenario

To demonstrate our data science cost estimate, we are going to create a Cost of Living Estimator. This tool, when complete, will allow a normal user to access the analyzed and processed data set and compare how expensive it is to live in one city vs another. The data assembled within the tool is authentic and publicly available. All images throughout this paper are actual screenshots and code, assembled using fully functional and deployed tools.

2.2. Data Collection Source

Although we are creating our cost estimate and tool using data science techniques, we must start just as any ordinary cost estimate would with the data set and structure. Many professionals feel that data is still incredibly difficult to find, but often the challenge is knowing where to search. There are numerous national and international data sites on the internet that provide reliable data for free to the public. Table 1 provides a small sample of popular data resources.

Table 1: Popular Data Sites

Data Site	Address
	https://www.data.gov/
	https://aws.amazon.com/datasets/
 THE WORLD FACTBOOK	https://www.cia.gov/library/publications/the-world-factbook/
 Government of Canada	https://open.canada.ca/en
 EU Open Data Portal	http://data.europa.eu/euodp/en/data/
	https://www.numbeo.com

For the Cost of Living Estimator exercise, we will utilize the Numbeo data portal for data (<https://www.numbeo.com/cost-of-living>) on costs for 55 major cities throughout the world.

Each city data set holds 28 types of costs such as

- Rent costs
- Groceries costs
- Fuel costs
- Dining costs
- Costs of Services (utilities, etc.)

Figure 2-1 provides an example of the display format for data on the Numbeo portal. It is important to note that there are certainly other types of costs that should be also be considered when examining living expenses for a location (e.g. taxes). However, the point of this exercise is to utilize a data science technique for analysis, not to create the most accurate estimate possible, so other expenses will not be considered. The actual data set that we are using for our Cost of Living Estimator Tool can be found here:

<https://github.com/tiwariraj/CostofLivingEstimator/tree/master/Scrapy%20Crawlers/costofliving>.



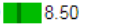


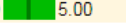











Restaurants	[Edit]	Range
Meal, Inexpensive Restaurant	14.00 \$	10.00  20.00
Meal for 2 People, Mid-range Restaurant, Three-course	50.00 \$	35.00  70.00
McMeal at McDonalds (or Equivalent Combo Meal)	7.00 \$	6.00  8.50
Domestic Beer (1 pint draught)	4.00 \$	3.00  6.00
Imported Beer (12 oz small bottle)	5.50 \$	4.00  7.00
Cappuccino (regular)	4.01 \$	3.00  5.00
Coke/Pepsi (12 oz small bottle)	1.78 \$	1.25  2.50
Water (12 oz small bottle)	1.43 \$	1.00  2.00
Markets	[Edit]	
Milk (regular), (1 gallon)	3.13 \$	2.00  4.00
Loaf of Fresh White Bread (1 lb)	2.34 \$	1.49  3.49
Rice (white), (1 lb)	1.78 \$	1.00  3.00
Eggs (regular) (12)	2.31 \$	1.30  3.50
Local Cheese (1 lb)	4.81 \$	3.00  7.50
Chicken Breasts (Boneless, Skinless), (1 lb)	3.84 \$	2.00  6.65
Beef Round (1 lb) (or Equivalent Back Leg Red Meat)	5.19 \$	3.18  8.00
Apples (1 lb)	1.98 \$	1.00  3.00
Banana (1 lb)	0.68 \$	0.49  1.00

Figure 2-1: Numbeo Data Set

2.3. Data Collection Capture

Since the data we are collecting is served via the Numbeo webpage interface, we will use a popular tool for the capture of the data using “spiders.” The tool, named Scrapy (think “scraping a windshield”) allows us to extract the data off the Numbeo site automatically for all 28 types of data we are interested in and store the data in files located in the cloud, on a stand-

alone computer, or on a server for our analysis. Figure 2-2 below shows the basic code for the Scrapy spider tool. This code is easily modified to accommodate various websites and search parameters. The modified code that we are using for our Cost of Living Estimator Tool can be found here:

https://github.com/tiwariraj/CostofLivingEstimator/blob/master/Scrapy%20Crawlers/costofliving/costofliving/spiders/costofliving_spider.py

A terminal window titled "Terminal" with a dark background and light text. It shows the following code:

```
$ pip install scrapy
$ cat > myspider.py <<EOF
import scrapy

class BlogSpider(scrapy.Spider):
    name = 'blogspider'
    start_urls = ['https://blog.scrapinghub.com']

    def parse(self, response):
        for title in response.css('.post-header>h2'):
            yield {'title': title.css('a ::text').extract_first()}

        for next_page in response.css('div.prev-post > a'):
            yield response.follow(next_page, self.parse)
EOF
$ scrapy runspider myspider.py
```

Figure 2-2: The Basic Scrapy Spider Code Template

2.4. Data Filtering and Cleanup

One of the most challenging aspects of both data science and cost estimating is effectively organizing raw data into usable data. To convert raw data into usable data any sorting issues, correlation concerns, gaps, and erroneous data must be addressed. To accomplish this, we will use the Python programming language. Python provides a capability to work with large data sets quickly and effectively with a simple interface. For the Cost of Living Estimator, we will use the Python code in Figure 2-3 below to rapidly update the data set with the total number of data points that were contained in each city's cost data. A data "tag" will also be included for each city that indicates the city type. These city types have been captured separately from the book *24 Hour Cities* by Hugh Kelly and will categorize the cities as 24-hour, 18-hour, 2nd tier market, and small market cities.


```
count = []
marketcount=len(series)
for i in range(marketcount):
    count.append(series[i][13])
obj = pd.Series(count)
Response = pd.DataFrame(obj).rename(columns={0: "Response"})
market = pd.concat([market, Response],axis=1)
```

Figure 2-3: Python Code for Appending Data Set with Data Point Information

2.5. Data Analysis

Once satisfied with the “cleanliness” of our data, we can begin the analysis that will serve our tool and therefore, our end users. First, we will examine the distribution and dispersion of the data and the frequency which it occurs. Figure 2-4 is an example of gasoline prices per city for the fifty markets.

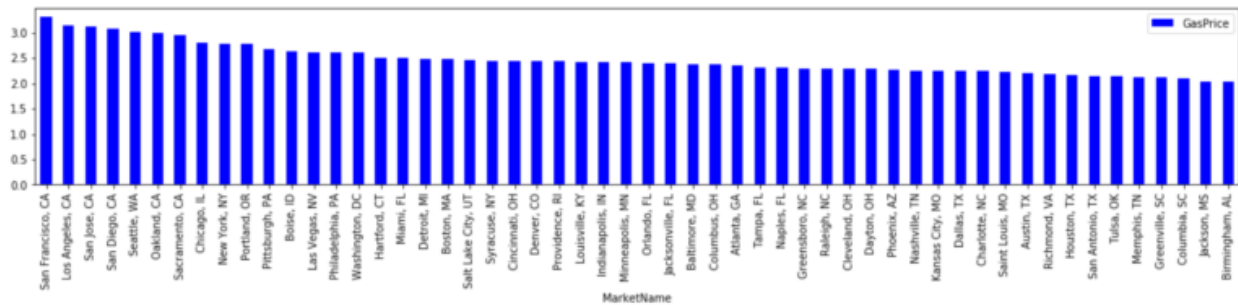


Figure 2-4: Gasoline Prices Per City

Figure 2-5 below shows the frequency that various gasoline prices occur.

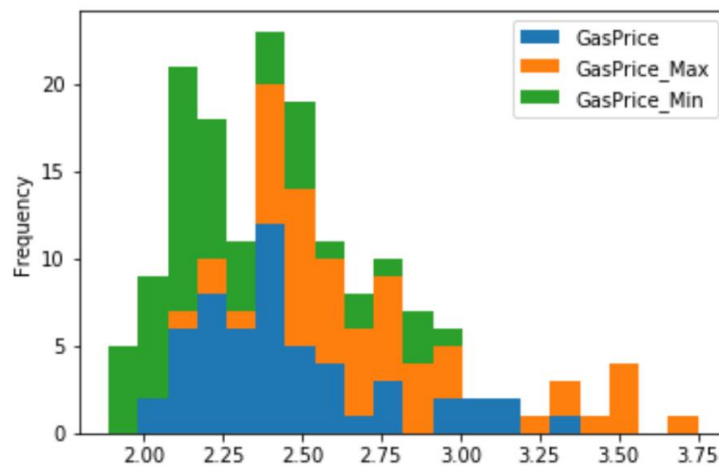


Figure 2-5: Gasoline Price Frequency

Like the prices for gasoline, we will repeat the distribution and frequency analysis for all 28 of the cost types contained in the data in preparation for our next step. The code that we are using for our the gasoline prices above can be found here and is replicated for each cost type: https://github.com/tiwarirai/CostofLivingEstimator/blob/master/Scrapy%20Crawlers/costofliving_gas/gascost.csv

An important part of the data analysis is the association of cost types, the cost themselves, and the city types that were identified earlier. This will be used to reference data in the tool and allow it to be served back to the user. Continuing with the example of gasoline prices above, we will use the code in Figure 2-6 to associate the city type with the prices of gasoline we analyzed.

```
#Anova Test - Gas
twentyfourhr=gascost[gascost['Classification']=='24-hour city']
eighteenhr=gascost[gascost['Classification']=='18-hour city']
secondtier=gascost[gascost['Classification']=='2nd Tier']
smallmarket=gascost[gascost['Classification']=='Small Market']
stats.f_oneway(twentyfourhr["GasPrice"], smallmarket["GasPrice"])
```

Figure 2-6: Code for Associating Gasoline Prices with City Types

As shown in Figure 2-7 below for gasoline prices, the analysis reveals that cost is impacted as a result of the city type and therefore on the cost of living within that city.

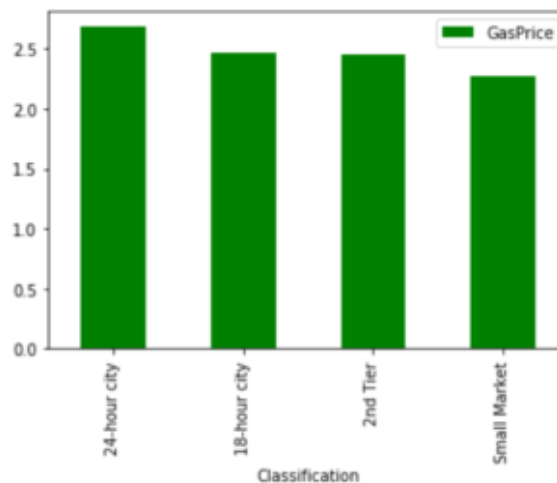


Figure 2-7: Gasoline Prices vs City Type

Using Fisher’s Method for combining P-values ($-2 \sum_{i=1}^k \log(p_i) \sim X_{2k}^2$) we can now determine the impact of a city type on all the cost of living types and place them into a matrix. This matrix will be inserted into the tool that is accessed by the end. Figure 2-8 below provides an example of the final matrix resulting from this analysis. The entire matrix from the analysis can be found

here:

https://github.com/tiwariraj/CostofLivingEstimator/tree/master/Fishers_pmatrix/market_rai

Market	#of tests	#of tests with pv<0.05	mean score(-2log(pv))	#of bootstrap	si	smi	ni.1	nmi	pi	pmi	rri	pv.fisher
New York, NY	28	14	7.813384393	1000	218.7748	3853.833	28	1484	4.27E-21	3.22E-26	7.56E-06	0
Saint Louis, MO	28	11	6.956270103	1000	194.7756	3877.833	28	1484	3.14E-17	1.94E-27	6.17E-11	0.014
San Francisco, CA	28	12	5.792777139	1000	162.1978	3910.41	28	1484	2.87E-12	3.88E-29	1.35E-17	0.15
Jackson, MS	28	6	5.156328516	1000	144.3772	3928.231	28	1484	9.74E-10	4.39E-30	4.50E-21	0.322
Charleston, SC	28	9	5.007953994	1000	140.2227	3932.385	28	1484	3.59E-09	2.63E-30	7.31E-22	0.365
Tulsa, OK	28	6	4.336051087	1000	121.4094	3951.199	28	1484	9.83E-07	2.52E-31	2.57E-25	0.565
San Jose, CA	28	6	4.238865773	1000	118.6882	3953.92	28	1484	2.11E-06	1.79E-31	8.48E-26	0.598
Minneapolis, MN	28	4	4.062658771	1000	113.7544	3958.854	28	1484	8.18E-06	9.63E-32	1.18E-26	0.636
Greenville, SC	28	5	3.730809672	1000	104.4627	3968.145	28	1484	9.20E-05	2.97E-32	3.23E-28	0.714
Birmingham, AL	28	4	3.69121604	1000	103.354	3969.254	28	1484	0.000121	2.58E-32	2.13E-28	0.721
Orlando, FL	28	6	3.494461978	1000	97.84494	3974.763	28	1484	0.000458	1.28E-32	2.79E-29	0.756
Dayton, OH	28	4	3.367414625	1000	94.28761	3978.321	28	1484	0.00104	8.12E-33	7.81E-30	0.771
Boise, ID	28	1	3.072499423	1000	86.02998	3986.578	28	1484	0.006081	2.81E-33	4.63E-31	0.816
Boston, MA	28	5	3.007406716	1000	84.20739	3988.401	28	1484	0.008736	2.23E-33	2.55E-31	0.822
Syracuse, NY	28	3	2.977780787	1000	83.37786	3989.23	28	1484	0.010266	2.00E-33	1.95E-31	0.823
Cincinnati, OH	28	3	2.853879473	1000	79.90863	3992.7	28	1484	0.019662	1.28E-33	6.50E-32	0.835
Las Vegas, NV	28	3	2.792741524	1000	78.19676	3994.411	28	1484	0.026684	1.02E-33	3.84E-32	0.842

Figure 2-8: Matrix with Combined P Values using Fisher’s Method

2.6. User Interface with RStudio and Shiny

RStudio is a programming language that is open-source, free, and efficient at statistical computation and graphics. It has been adopted by many organizations in government and commercial industry after extensive examination and is open source. It continues to become one of the premiere platforms for analysis and has significant advantages over other traditional platforms with the prime advantage being the construction of interfaces for accessing large amounts of data. RStudio can be combined with various add on packages to further extend capability. In this example, we will be using the Shiny package with RStudio to enable the Cost of Living Estimator Tool to be accessed interactively through a web browser. Figure 2-9 shows an example of the RStudio compiler interface.

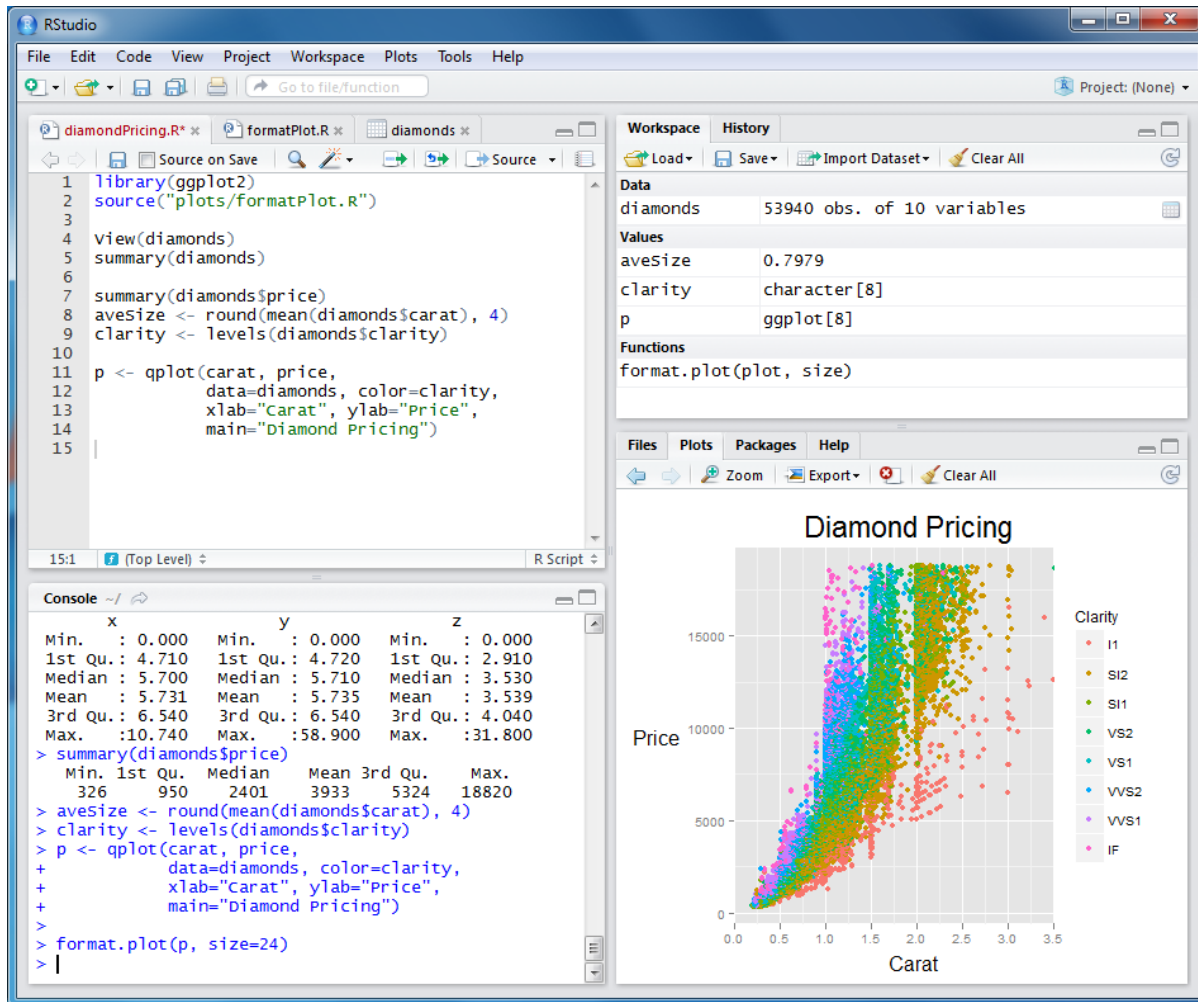


Figure 2-9: RStudio Compiler Interface

The Cost of Living Estimator Tool interface has seven major inputs for the user to populate as seen in Figure 2-10:

- Married status
- Number of children
- Urban or suburban location preference
- Transportation method preferred
- Number of meals purchased
- Preferred region
- Household monthly take-home income

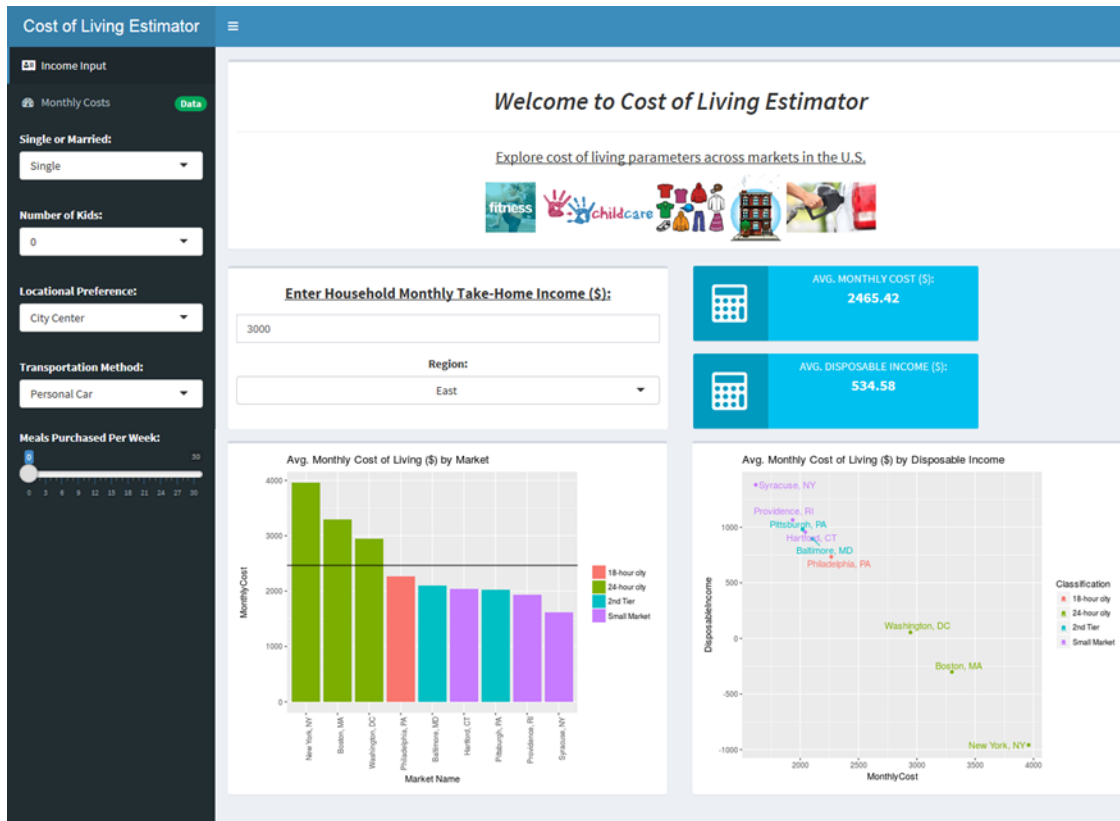


Figure 2-10: Cost of Living Estimator Tool Main Interface

Stepping through the tool in sequence we will first select the marital status from the drop-down menu as seen in Figure 2-11:

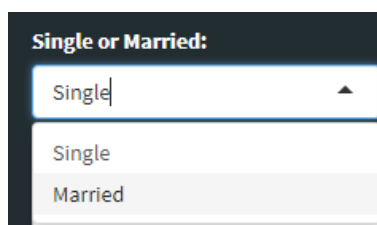


Figure 2-11: Marital Status Selection

Once the first input is selected the tool immediately updates the average monthly cost without any delay for calculation cycles. Since we have not yet entered a household monthly take-home income the average disposable income is calculated based on the default of \$3000 as seen in Figure 2-12:

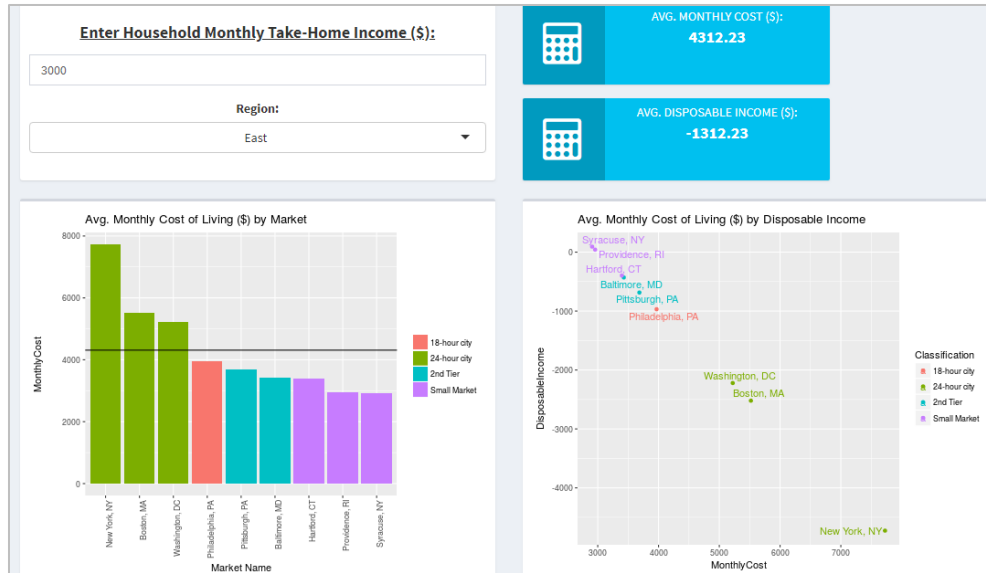


Figure 2-12: Cost of Living Estimator Output After Marital Status Selection

Like the input made for marital status, the remaining inputs can be selected/entered in any order the user desires as shown in the following figures with an immediate change in results:

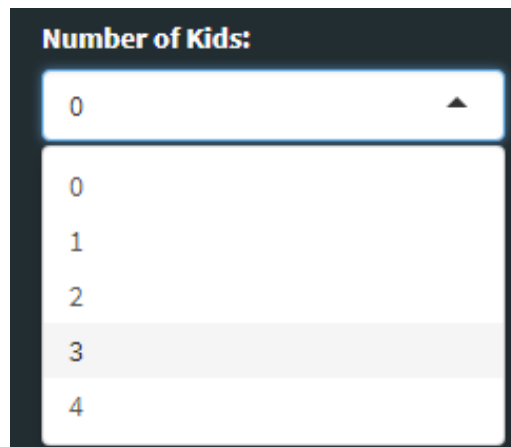
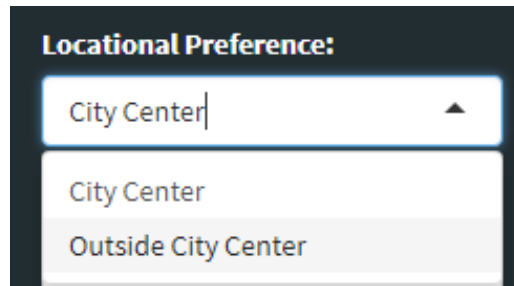


Figure 2-13: Number of Kids Selection



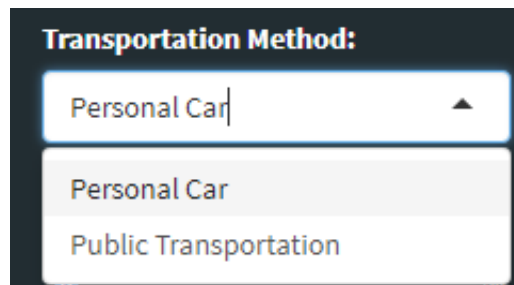
Locational Preference:

City Center

City Center

Outside City Center

Figure 2-14: Location Preference Selection



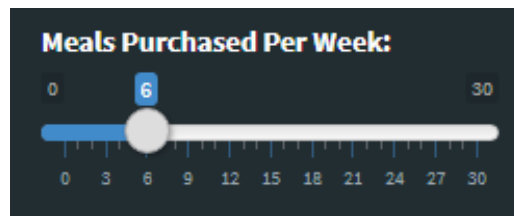
Transportation Method:

Personal Car

Personal Car

Public Transportation

Figure 2-15 Transportation Method Selection

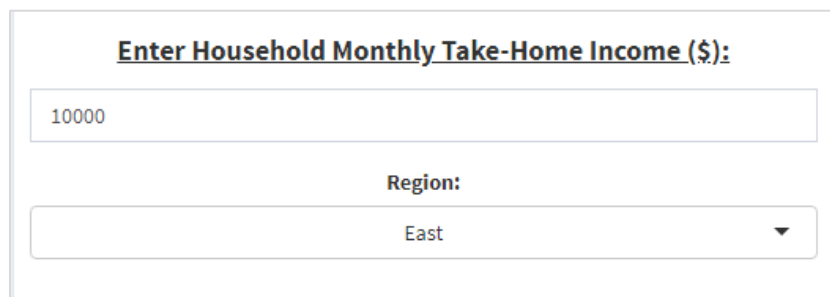


Meals Purchased Per Week:

0 6 30

0 3 6 9 12 15 18 21 24 27 30

Figure 2-16: Meals Purchased Per Week Selection



Enter Household Monthly Take-Home Income (\$):

10000

Region:

East

Figure 2-17: Household Monthly Take-Home Income and Region Input

When all inputs have been addressed, the estimator provides an accurate display, shown in Figure 2-18, of the average monthly cost and average disposable income based on the analysis discussed earlier. Additionally, the user is provided with charting options showing the comparison of the scenario to the average monthly cost of living by market and the average monthly cost of living by disposable income.

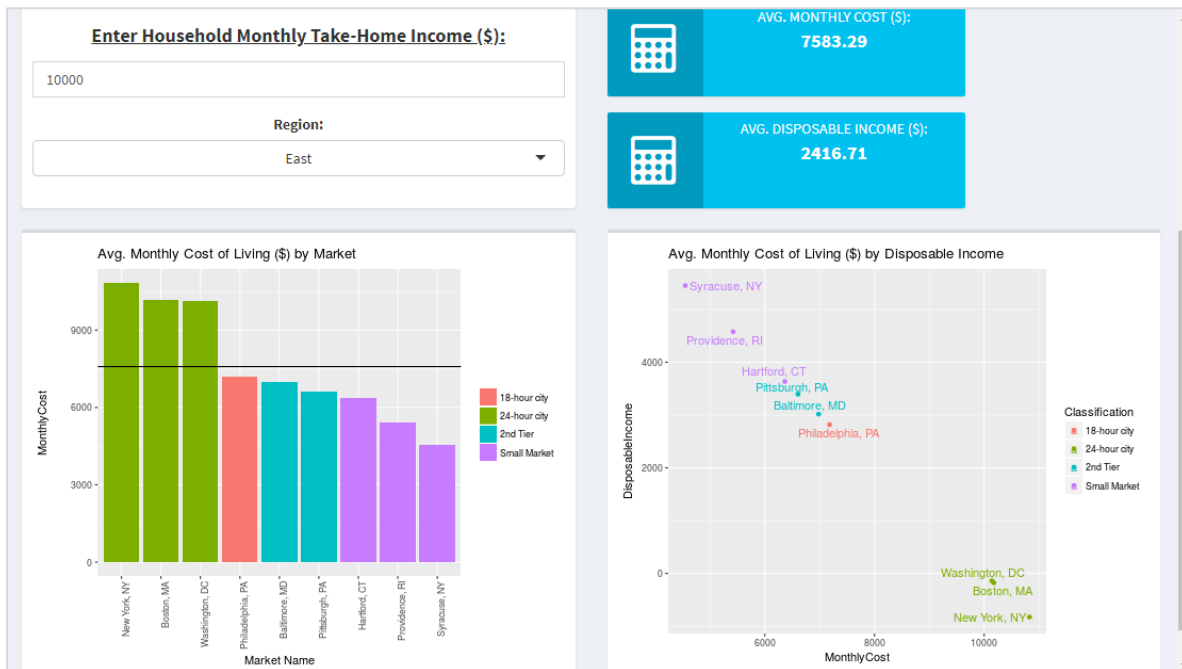


Figure 2-18: Cost of Living Estimator Output After All Selections Made

The complete Cost of Living Estimator is available at the following address:

<https://tiwariraj.shinyapps.io/CostofLivingEstimator/>

2.7. Data Pipelining

Although the example shown above utilizes an analysis of a static data set, all the tools used provide the capability of modifying the tool to perform analysis of data sets refreshed via a data pipeline at any interval desired including continuous real-time data. This feature provides a powerful advantage over other tools and analyses that suffer from data expiration the moment deployment occurs. However, one should be cautious to recognize that since the data portal is under the control of a third party, there is the possibility the tool could stop functioning at any time because of data formatting changes, portal configuration changes, and interruptions in the operation of the data portal beyond the control of the tool author and end user.

3. Your Own Data Science Cost Tool and Closing Remarks

The tools and methods shown in the Cost Estimating Tool example above are easily applied to other scenarios. If you are constructing a data science-based cost estimate or tool, it is important to apply the following approach:

- Plan the analysis just as you would with any other cost estimating approach
- Collect well documented data as you would with any other estimate
- Use Scrapy or a similar data capture tool to collect the data
- Use Python or a similar tool to clean, sort, gap fill, and analyze the data with well documented assumptions
- If an interface is required, use a tool like RStudio or R Shiny to create the interface and populate a tool with the end user experience in mind

It is not necessary to do use all the tools and methods shown here at once. Integrate individual techniques and tools into the methods you use now and then add others as your become more comfortable with the approach.

Analysts and ICEAA have been clear that data science is a good fit in the cost estimating community and provides distinct advantages over traditional cost estimating techniques. An opportunity exists for ICEAA to reinforce its standing, grow recognition as the leading authority in the cost estimating industry, and increase membership through the training, and recruitment of data science professionals for cost estimating. It is essential that the opportunity be pursued as aggressively as possible to ensure success. It's my intention to provide this paper for leadership as a basis for future cost estimating training curriculum and practices. I have further expectations that this paper will help analysts and end users expand their abilities and apply data science techniques in their current estimates.

I would like to add a special thanks to Raj Tiwari for giving me permission to use his code and tool for the discussion in this paper.

4. Appendix I: Data Science Tool Resources

Scrapy

<https://scrapy.org/>

Numbeo Cost of Living

<https://www.numbeo.com/cost-of-living/>

Python

<https://www.python.org/>

R Studio

<https://www.rstudio.com/>

5. Appendix II: Sources

Cost of Living Estimator, Raj Tiwari

<https://tiwariraj.shinyapps.io/CostofLivingEstimator/>

“Big Data Not a Cure All In Medicine,” 2015.

<https://www.npr.org/2015/01/05/375201444/big-data-not-a-cure-all-in-medicine>

“The American Statistician Vol. 2, No.5,” 1948, Frederick Mosteller and R.A. Fisher.

<https://www.npr.org/2015/01/05/375201444/big-data-not-a-cure-all-in-medicine>

“24 Hour Cities,” 2016, Hugh Kelly.

https://www.amazon.com/dp/B01IHQ23V2/ref=cm_sw_em_r_mt_dp_U_htbCCbFDV1ZPY