Leveraging Technology for Compilation of the Consumer Price Index

by Sarah Ng, Lee Jia Wen and Ruth Lee
Prices Division
Singapore Department of Statistics

Introduction

Compiled by the Singapore Department of Statistics (DOS), the Consumer Price Index (CPI) measures the average price changes of a fixed basket of consumption goods and services commonly purchased by resident households over time.

Price data used in the compilation of the CPI are gathered through various modes. This article presents DOS's continual efforts to adopt the latest technology and open-source tools in the areas of price data collection, processing and compilation, which include using web-scrapers, leveraging electronic prices, integrating handheld devices in field operations and compiling reports via Python scripts.

Web-Scraping of Online Prices

Since 2015, DOS has been utilising web-scraping to automate data collection. This not only minimises respondents' survey burden, but also reduces data collection efforts. At the onset, when open-source tools were not prevalent and in-house expertise on web-scraping technology was limited, IT programmers were engaged to develop customised web crawlers to automate the extraction of online prices. However, given the frequent changes in website layouts, the cost of re-developing these customised web crawlers was high. Eventually, these web crawlers were discontinued.

Greater Use of Open-Source Tools

DOS subsequently turned to open-source tools and explored the use of commercial point-and-click web-scrapers, like Import.IO and ParseHub. With the emergence of Python programming language, DOS ventured into the development of in-house web-scraping scripts using Python via the *Requests*, *BeautifulSoup* and *Selenium* libraries.

Detailed price information from websites retailing food, home electronics & appliances, furniture, personal effects, apparels, and medicine & health products were subsequently extracted via these software and Python scripts. Examples of data fields that were extracted directly from selected websites, using Python scripts for food and apparel items are shown in Figure 1.

FIGURE 1 EXAMPLES OF DATA FIELDS EXTRACTED FROM WEBSITES VIA WEB-SCRAPING WITH PYTHON

FOOD

	A	B	C	D	E	F	G	H	I
1	**Link**	**Description**	**Packaging**	**Final Price**	**Usual Price**	**Discount**	**Availability**	**Extract Date**	**Brand**
2	https://w	Golden Phoenix Thai Hom Mali Rice	5kg	13.80		0	ADD TO CART	20230712	Golden Phoenix
3	https://w	Golden Pineapple AAA Thai Premium Fragrant Rice	10kg	22.52	28.17	-20%	ADD TO CART	20230712	Golden Pineapple
4	https://w	Pagoda Pure Corn Flour	400g	1.16		0	ADD TO CART	20230712	Pagoda
5	https://w	Myojo Chicken Abalone Flavoured Instant Noodles	5×79g	2.65		Buy 2 Save 9%	ADD TO CART	20230712	Myojo
6	https://w	CHIPSMORE Original Cookies Biscuits 153G	163.2g	1.95		0	ADD TO CART	20230712	CHIPSMORE
7	https://w	Kellogg's Frosties Breakfast Cereal	300g	5.40	6.00	-10%	ADD TO CART	20230712	Kellogg's
8	https://w	Best Foods Real Mayonnaise	430ml	5.03	5.30	-5%	ADD TO CART	20230712	Best Foods
9	https://w	Woolworths Deliciously Crunchy Peanut Butter	500g	4.00		0	ADD TO WISHLIST	20230712	Woolworths

APPAREL

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**Link**	**Product code**	**Name**	**Color**	**Usual Price**	**Final Price**	**Discount**	**Availability**	**Features**	**Dimensions**	**Description**	**No of colors**	**No of sizes in stock**	**Total no of sizes**	**Extract Date**
2	https://	Product Code: 3611	Organic Loose Fi	white		$24.99	2 FOR $30	Add To Bag	100% organic cot	Size: Chest Circur	Made with sust	7	7	7	20230707
3	https://	Product Code: 3301	Super Skinny Jear	new black		$69.99		Add To Bag	Skinny through tl	Size: Waist Circur	Super skinny, s	8	3	5	20230707
4	https://	Product Code: 2053	Regular Fit Graph	floral boot		$24.99	BUY ONE, GET ONE 50% C	Add To Bag	Standard body le	Size: Bust Circumf	The regular fit	1	4	5	20230707
5	https://	Product Code: 4591	Frill Ribbed Ankl	dusty pink	$9.99	$2.00	(-80%)	Add To Bag	Ribbed Construc	Size: LengthOS - 2	Frill ribbed ank	9	0	0	20230707

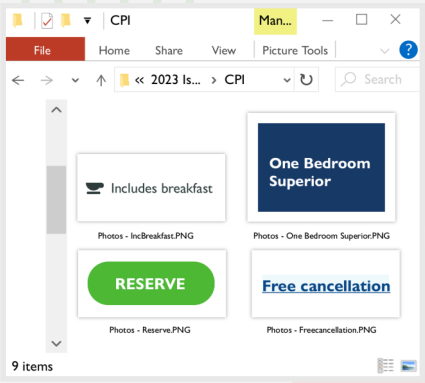
When more advanced Python libraries became available, the extraction of price information from dynamic websites was facilitated using the *Pyautogui* and *Pytesseract* libraries.

In the web-scraping of hotel room rates, the *Pyautogui* library automates the repetitive tasks of searching for the specific room type, breakfast inclusion and cancellation policy (refundable / non-refundable), as well as clicking on the corresponding 'reserve" button. It can also take screenshots, such as the final price on the payment page. The *Pytesseract* library was incorporated into the script to convert the final price embedded in the screenshot into text, before loading into the CPI computerised system for data compilation.

Figure 2 details the programme flow of the Python web-scraper for the extraction of hotel room rates.

FIGURE 2 PROGRAMME FLOW TO EXTRACT HOTEL ROOM RATES

Before running the programme, the following images are stored in a folder, i.e., "One Bedroom Superior", "Free cancellation", "Includes Breakfast", and "Reserve".

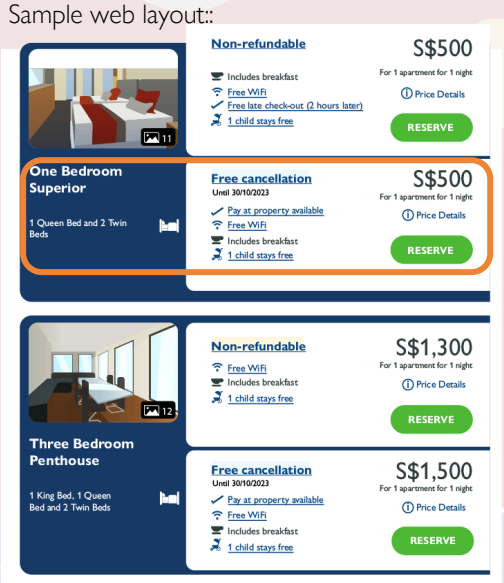


At the start of the programme, the Python script reads in an input file where each row contains the following columns:

- Web link to the hotel
- Room type, breakfast type and cancellation policy

	A	B	C	D	E	F	G
1	No.	City	Hotel	Weblin	Room Type	Breakfast Type	Cancellation Policy
2	1	Country 1	Hotel A	https://ww	One Bedroom Superior	Includes Breakfast	Free Cancellation
3	2	Country 1	Hotel A	https://ww	Three Bedroom Penthouse	Includes Breakfast	Free Cancellation
4	3	Country 1	Hotel A	https://ww	Three Bedroom Penthouse	Includes Breakfast	Non-Refundable
5	4	Country 2	Hotel B	https://ww	One Bedroom Superior	Includes Breakfast	Free Cancellation
6	5	Country 2	Hotel B	https://ww	Three Bedroom Penthouse	Includes Breakfast	Free Cancellation
7	6	Country 2	Hotel B	https://ww	Three Bedroom Penthouse	Includes Breakfast	Non-Refundable
8	7	Country 3	Hotel C	https://ww	One Bedroom Superior	Includes Breakfast	Free Cancellation
9	8	Country 3	Hotel C	https://ww	Three Bedroom Penthouse	Includes Breakfast	Free Cancellation
10	9	Country 3	Hotel C	https://ww	Three Bedroom Penthouse	Includes Breakfast	Non-Refundable

Sample web layout::



Based on the input file, *Pyautogui* keys the web link into the internet browser and begin the search according to the following programme flow:

i. Continue scrolling until the "One Bedroom Superior" image is found.

ii. Search for image of "Free cancellation" in area within 100 to 450 pixels from the top. If not found, scroll down once and repeat (ii).

iii. Search for image of "Includes Breakfast" within 0 to 200 pixels below the coordinates of "Free cancellation". If not found, scroll down once and repeat (ii) & (iii).

iv. If all images have been found, locate image of "Reserve" button of about 1000 pixels to the right and 0 to 180 pixels down of the "Includes Breakfast" image and click on the "Reserve" button (orange box).

v. If unable to find any of the images in (iv), to stop after 20 loops and move on to the next row of the input file.

Pyautogui takes a screenshot of the final payment page for successful iterations.

Pytesseract library converts the final price in the screenshot into text and exports the output to an excel file.

	A	B	C	D	E	F	G	H
1	No.	City	Hotel	Weblin	Room Type	Breakfast Type	Cancellation Policy	Price
2	1	Country 1	Hotel A	https://ww	One Bedroom Superior	Includes Breakfast	Free Cancellation	500
3	2	Country 1	Hotel A	https://ww	Three Bedroom Penthouse	Includes Breakfast	Free Cancellation	1500
4	3	Country 1	Hotel A	https://ww	Three Bedroom Penthouse	Includes Breakfast	Non-Refundable	1300
5	4	Country 2	Hotel B	https://ww	One Bedroom Superior	Includes Breakfast	Free Cancellation	450
6	5	Country 2	Hotel B	https://ww	Three Bedroom Penthouse	Includes Breakfast	Free Cancellation	1500
7	6	Country 2	Hotel B	https://ww	Three Bedroom Penthouse	Includes Breakfast	Non-Refundable	1250
8	7	Country 3	Hotel C	https://ww	One Bedroom Superior	Includes Breakfast	Free Cancellation	550
9	8	Country 3	Hotel C	https://ww	Three Bedroom Penthouse	Includes Breakfast	Free Cancellation	1550
10	9	Country 3	Hotel C	https://ww	Three Bedroom Penthouse	Includes Breakfast	Non-Refundable	1250
11								

Greater Use of Web-Scraping

Web-scraping is an efficient method to collect online prices. It reduces survey burden while allowing for automated extraction of more data points at a higher frequency. During the COVID-19 pandemic when physical price collection was impeded, the use of web-scraping mitigated some of the price collection challenges.

With more community contributions to open-source libraries, the use of web-scraping techniques can be further enhanced over time and leveraged to extract more online price data for compilation purposes.

Aligned with international best practices, DOS adopts the following principles to ensure that web-scraping is carried out consistently, ethically, and transparently:

i) Minimise burden on the website owners (e.g., by adding idle time between requests; web-scraping at a time of day when the web server is not expected to be under heavy load);

ii) Identify DOS to the website owners as an explicit "declaration of intent" to carry out web-scraping; and,

iii) Web-scrape data for statistical purposes only.

Leveraging Electronics Returns from Major Supermarket Chains

DOS has been collaborating with major supermarket chains to obtain their electronic price data directly for CPI compilation since 2015. These prices are compiled using actual sales transactions of consumer goods obtained via their electronic points of sales (POS). This way, discounts given by the supermarkets are taken into account. Previously, the price data was collected via in-person visits which was significantly more laborious.

The list of items monitored from supermarkets ranged from perishable items to groceries and healthcare products. To facilitate the identification of the required items for CPI compilation, DOS specifies the barcodes for each monitored item in the data file to the supermarkets.

Apart from time savings gained in place of in-person visits, the use of electronic prices also improved the quality of data used for CPI compilation as they are derived from actual purchase transactions which are more reflective of the monthly average prices paid by consumers.

Use of Handheld Devices for Price Collection

Another area which DOS focused on was the improvement in data capturing for field operations through the adoption of handheld devices.

Since 2019, field interviewers have been recording the information collected via web-based survey forms in the handheld devices. Replacing hardcopy survey forms with digital capture and identification via Singpass not only enhance the security of the field data collected, but also improve operational efficiency, through automated data validation and minimal data entry.

Some features of the web-based forms in handheld devices (Figure 3) include:

i) Field interviewers can view information on the outlets to be surveyed for the day and those upcoming, including item descriptions, addresses of establishments, stall numbers, prices from previous period, etc. This helps them plan the routes to conduct field collection more efficiently.

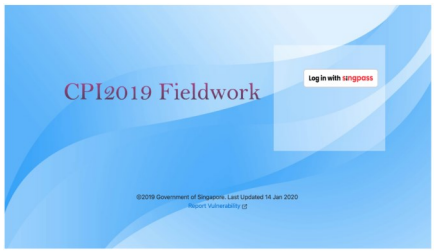
ii) Prices collected in the previous survey period can be copied over to the current survey period with a click of a button for each establishment, doing away with the need to perform data entry for each item.

iii) Real-time computation of month-on-month percentage change in price is integrated in the web-based form. It serves as a validation check so that clarifications could be made with respondents promptly.

iv) Common reasons for price changes such as increase in rental cost are embedded as dropdown options for field interviewers to select from in the web-based form, eliminating the need for them to type in the same reason for applicable items.

FIGURE 3 WEB-BASED SURVEY FORM INTERFACE

Login Screen



Establishment Listing Screen

Item Listing Screen

Data Entry Screen

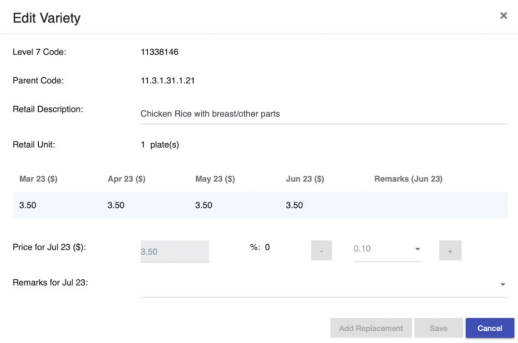


FIGURE 4 PROGRAMME FLOW FOR THE GENERATION OF MULTI-DIMENSIONAL DATA CUBES

- The Python programme reads in the annual and monthly average retail prices tab sheets 'T1' and 'T2' in the excel file for "Tables on CPI and percentage changes up to class level and average retail prices of selected consumer items" using the *Pandas* library.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Subject: Prices and Price Indices															
2	Topic: Consumer Price Index															
3	Title: M212891-Average Retail Prices Of Selected Consumer Items, Monthly[1]															
4																
5																
6	**Variables**	**2010 Jan**	**2010 Feb**	**2010 Mar**	**2010 Apr**	**2010 May**	**2010 Jun**	**2010 Jul**	**2010 Aug**	**2010 Sep**	**2010 Oct**	**2010 Nov**	**2010 Dec**	**2011 Jan**	**2011 Feb**	**2011 Mar**	**2011**
7	Premium Thai Rice (Per 5 Kilogram)	11.99	11.84	12.33	12.55	12.58	12.67	12.57	12.35	12.30	12.67	12.71	12.63	12.60	12.41	12.68	12
8	Ordinary White Bread (Per 400 Gram)	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.40	1.39	1.39	1.40	1.40	1.44	1.44	1.
9	Vitamin Enriched Bread (Per 400 Gram)	1.64	1.64	1.58	1.58	1.64	1.64	1.64	1.64	1.64	1.64	1.64	1.64	1.64	1.62	1.56	1.
10	Wholemeal Bread (Per 400 Gram)	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	n
11	Instant Noodles (Per 5 Packets)	2.12	2.13	2.15	2.18	2.18	2.15	2.17	2.13	2.17	2.20	2.19	2.18	2.18	2.20	2.22	2.
12	Lean Pork, Chilled (Per Kilogram)	12.63	12.89	12.68	12.70	12.69	12.53	12.65	12.74	12.63	12.65	12.55	12.58	12.96	13.27	12.65	12
13	Streaky Pork, Chilled (Per Kilogram)	13.25	13.80	13.61	13.61	13.63	13.60	13.59	13.62	13.66	13.67	13.67	13.72	13.90	14.14	13.77	13
14	Pork Rib Bones, Chilled (Per Kilogram)	15.43	15.93	15.84	15.84	15.95	15.87	15.89	15.73	15.92	15.91	15.86	15.91	16.09	16.52	16.07	16
15	Beef, Chilled (Per Kilogram)	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	n
16	Mutton, Chilled (Per Kilogram)	14.09	14.09	14.09	14.09	14.59	15.33	15.41	15.31	15.88	15.98	16.26	16.26	16.26	16.99	17.79	17
17	Whole Chicken, Chilled (Per Kilogram)	5.40	5.47	5.42	5.37	5.40	5.49	5.58	5.63	5.64	5.67	5.71	5.65	5.73	5.83	5.79	5.
18	Chicken Wing, Chilled (Per Kilogram)	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	n
19	Duck, Chilled (Per Kilogram)	6.56	6.59	6.53	6.49	6.50	6.55	6.62	6.70	6.64	6.63	6.64	6.65	6.79	6.93	6.80	6.
20	Lean Pork, Frozen (Per 500 Gram)	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	n
21	Pork Rib Bones, Frozen (Per 500 Gram)	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	n
22	Beef Cube, Frozen (Per 500 Gram)	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	n
23	Whole Chicken, Frozen (Each)	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	n
24	Chicken Wing, Frozen (Per 2 Kilogram)	na	na	na	na	na	na	na	na	na	na	na	na	na	na	na	n
25	Cod Fish (Per Kilogram)	37.58	38.50	38.12	39.07	39.28	38.64	39.81	40.74	40.93	41.31	41.30	42.25	43.67	43.75	45.42	46
26	Gold Banded Scad (Kuning) (Per Kilogram)	5.58	5.66	5.59	5.72	5.68	5.74	5.74	5.76	5.78	5.81	5.82	5.68	5.85	5.67	5.85	5.
27	Flowery Grouper (Per Kilogram)	13.02	14.44	13.10	13.15	12.83	12.46	12.81	13.01	13.25	13.00	13.06	12.89	13.89	14.37	13.93	14
28	White Pomfret (Per Kilogram)	20.86	24.92	19.94	20.85	20.70	21.17	22.13	22.57	22.51	21.93	22.22	22.45	26.91	26.81	23.37	22
29	Salmon (Per Kilogram)	23.42	24.31	24.19	24.91	25.37	25.29	25.66	25.52	25.58	25.53	25.61	25.81	26.27	26.75	25.91	26

Content T1 T2 T3 T4 T5 T6 T7 T8 T9 T10 T11 T12 T13 T14 T15

- The programme extracts the required rows and stores them as a data frame.
- The *melt()* function is then used to transform the data frame from the existing wide form to the tidy format, i.e., the data is unpivoted into rows. It also renames the column label 'Variables' to 'Consumer Item', the column label for average price to 'M1', and the column label for the months to 'Month'.

```
.melt(id_vars=['Consumer Item'], var_name='Month', value_name='M1')
```

Sample output in Python:

Month	Consumer Item	M1
2010 Jan	Premium Thai Ri	11.99
2010 Jan	Ordinary White	1.4
2010 Jan	Vitamin Enriche	1.64
2010 Jan	Wholemeal Brea	na
	...	
2023 Jun	Premium Thai Ri	13.67
	...	

FIGURE 4 PROGRAMME FLOW FOR THE GENERATION OF MULTI-DIMENSIONAL DATA CUBES (cont'd)

- The *NumPy* library's *.nan* function converts "na" text value to a special floating-point value "NaN" and sets the 'Flag_M1' values for each data point as 0. A *for* loop is also used to amend 'Flag_M1' values as "3" if a data point is not available.
- A *lambda* function is added to convert the date format from 'YYYY Mmm' to 'Mmm-YY'.

```
|.apply(lambda x: datetime.strptime(x, '%Y %b').strftime('%b-%y'))
```

- The data frame is further sorted by a descending order for 'Month'.

Sample output in Python:

Month	Consumer Item	M1	Flag_M1
Jun-23	Premium Thai Ri	13.67	0
	...		
Jan-10	Premium Thai Ri	11.99	0
Jan-10	Ordinary White	1.4	0
Jan-10	Vitamin Enriche	1.64	0
Jan-10	Wholemeal Brea	NaN	3

- The programme repeats the above steps for the annual prices found in the 'T1' tab sheet, and exports the transformed data frames to excel.
- The *Openpyxl* library applies some formatting features to the exported file such as setting cell alignment to center, adding thin borders, setting number formats to 2 decimal places and auto-fitting column width.

Sample output in Excel:

Month	Consumer Item	M1	Flag_M1
Jun-23	Premium Thai Rice (Per 5 Kilogram)	13.67	0
	...		
Jan-10	Premium Thai Rice (Per 5 Kilogram)	11.99	0
Jan-10	Ordinary White Bread (Per 400 Gram)	1.40	0
Jan-10	Vitamin Enriched Bread (Per 400 Gram)	1.64	0
Jan-10	Wholemeal Bread (Per 400 Gram)		3

* Multi-dimensional data cubes for the CPI are available on the SingStat Website.

Compilation of Reports via Python

Besides data collection, Python libraries (*Pandas*, *NumPy* and *Openpyxl*) are used to automate the monthly generation of multi-dimensional data cubes for the CPI* (Figure 4). These libraries facilitate the extraction and merging of data from different file types and format them into the required structure.

Previously, the generation of multi-dimensional data cubes involved manual extraction of existing datasets in different frequencies and their consolidation into a single file in the required format. These tasks were repetitive and time consuming.

With the use of Python, the multi-dimensional data cubes are now generated in a timely, consistent and structured manner each month. Human error is reduced and the savings in man-hours led to greater productivity and efficiency.

Conclusion

The COVID-19 pandemic has accelerated the adoption of technology worldwide. DOS keeps pace with evolving technologies by benchmarking against best practices in the private sector and its international counterparts, and tapping on new software and open-source tools to ensure that data collection, processing and compilation methods remain relevant and efficient.

For an animated introduction to how price data are gathered for the compilation of the CPI, check out the video on "How are Prices Collected for the Compilation of Consumer Price Index".