# [Data Quality with Python for Beginner](https://academy.dqlab.id/main/package/practice/166)

## CHAPTER 1

### Pengantar

Aku dipanggil ke ruangan Kroma, sang kepala divisi! Gugup? Pastinya. Penasaran? Apalagi. Setelah diam beberapa saat, Kroma menunjukkan lembar kerjaku di layar laptopnya.

“Setelah mengecek hasil analisis kamu, saya rasa ini perlu sedikit revisi. Coba lakukan analisis terhadap datanya secara lebih mendalam menggunakan EDA, termasuk data profiling. Bisa?” pinta Kroma di ruangannya.

Aku memang sudah terbiasa mendengar kata ‘revisi’ dari Senja dan Andra, tapi tetap saja kali ini lebih menegangkan karena langsung dari Kroma, si kepala divisi! Aku hendak mengangguk tapi kuingat kalau aku belum sempat mempelajari mengenai data profiling bersama Senja. Seperti mampu membaca keraguanku, Kroma mencondongkan tubuhnya ke arahku dan berujar, “Sekaligus saya kirimkan referensi lengkap dan contoh bagaimana data profiling dilakukan.”

Aku pun keluar dari ruangan membawa ‘kado’ berupa revisi.

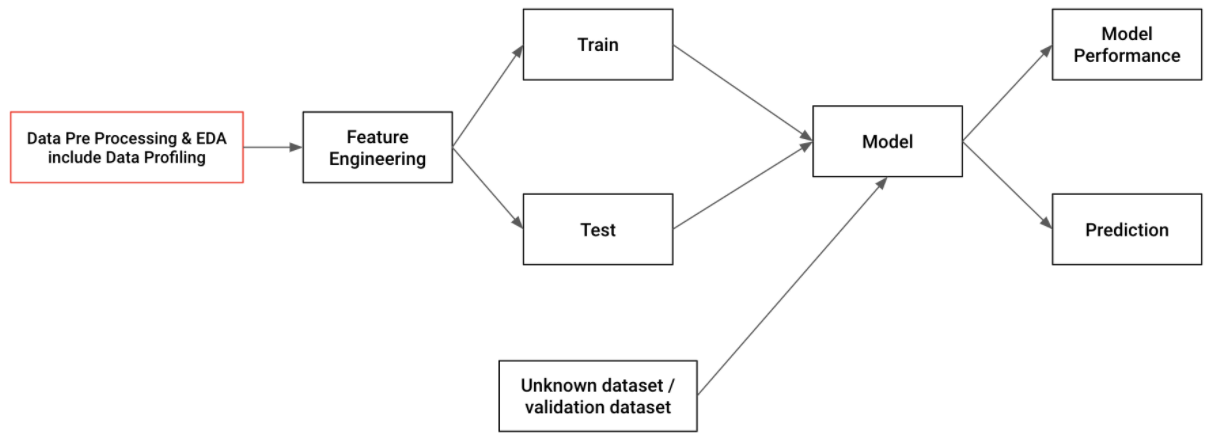
Senangnya kalau punya kepala divisi yang memberikan revisi sekaligus solusi. Setelah kembali ke meja, aku langsung membuka tautan referensi yang diberikan Kroma. Aku memutuskan untuk mempelajarinya dulu sebelum melakukan revisi pekerjaan.

### Apa itu Data Profiling?

Pada bagian sebelumnya, aku sudah mempelajari mengenai exploratory data analysis, dimana Exploratory Data Analysis (EDA) adalah menggunakan pendekatan statistik yang bertujuan untuk menemukan dan meringkas sebuah dataset, mengetahui struktur dan hubungan antar variabel dalam dataset. EDA merupakan proses pre-analysis baik untuk descriptive analysis maupun predictive analysis.

Dalam bab ini, aku akan fokus pada satu aspek EDA, yaitu: **Data Profiling**!

Data profiling adalah kegiatan merangkum dataset menggunakan statistik deskriptif. Tujuan dari pembuatan data profiling adalah untuk memiliki pemahaman yang kuat tentang data sehingga dapat mulai menyusun framework analisis  dan memvisualisasikan data.

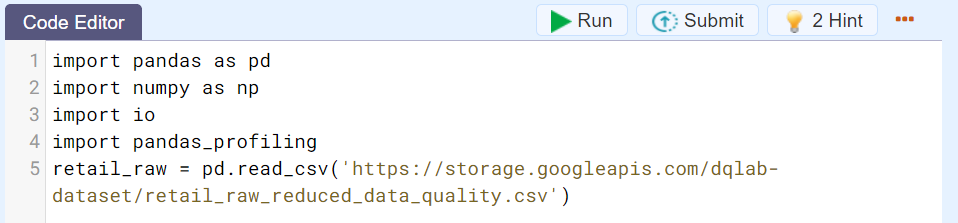


### Importing Data

Sebagai langkah pertama yang harus dilakukan adalah inisialisasi Library dan mengimport dataset tersebut ke dalam Python menggunakan library Pandas dan diassign sebagai **retail\_raw**.

Library yang perlu diimport adalah: (1) **pandas,** (2) **numpy,** (3) **io,** dan (4) **pandas\_profiling.** Untuk dua libray yang pertama importlah sebagai aliasnya.

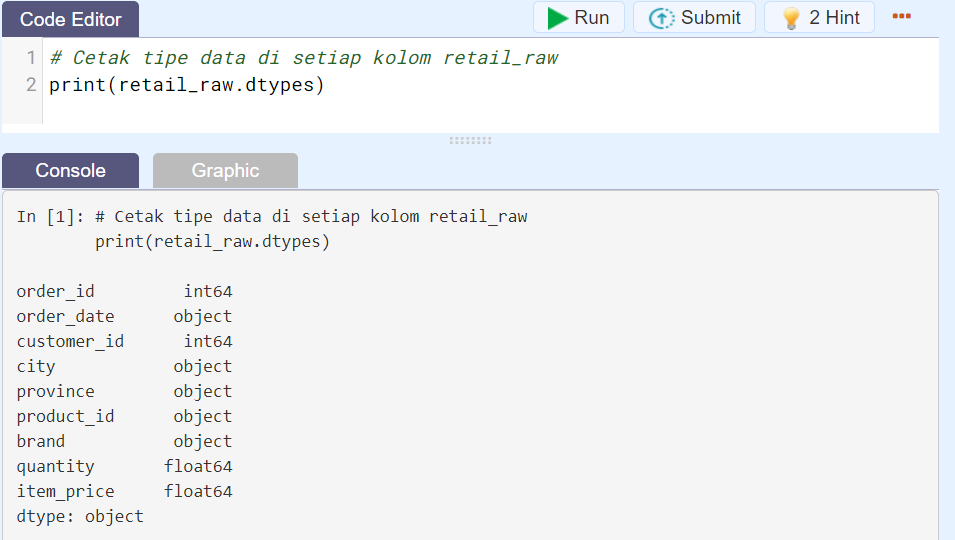
Datasetnya tersedia di: **'https://storage.googleapis.com/dqlab-dataset/retail\_raw\_reduced\_data\_quality.csv'**.



### Inspeksi tipe data

Dengan library Pandas, dapat menjalankan fungsi .**dtypes** untuk melihat data untuk setiap kolom termasuk tipe datanya.

Dengan menekan  untuk mengeksekusi baris program di code editor, diperoleh output di console berupa:



### Descriptive Statistics - Part 1

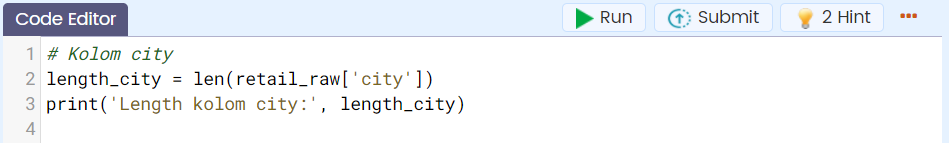
Di bagian ini, aku akan mempelajari berbagai statistik deskriptif yang dapat digunakan untuk lebih memahami struktur data.

**Length**

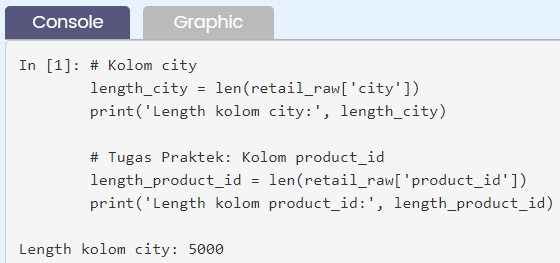
Fungsi **len** menghitung jumlah pengamatan dalam suatu series / column. Fungsi len akan menghitung semua pengamatan, terlepas dari apakah ada null-value atau tidak (include missing value).



Contoh penggunaan untuk mengetahui length dari kolom *city* dari dataframe *retail\_raw*!



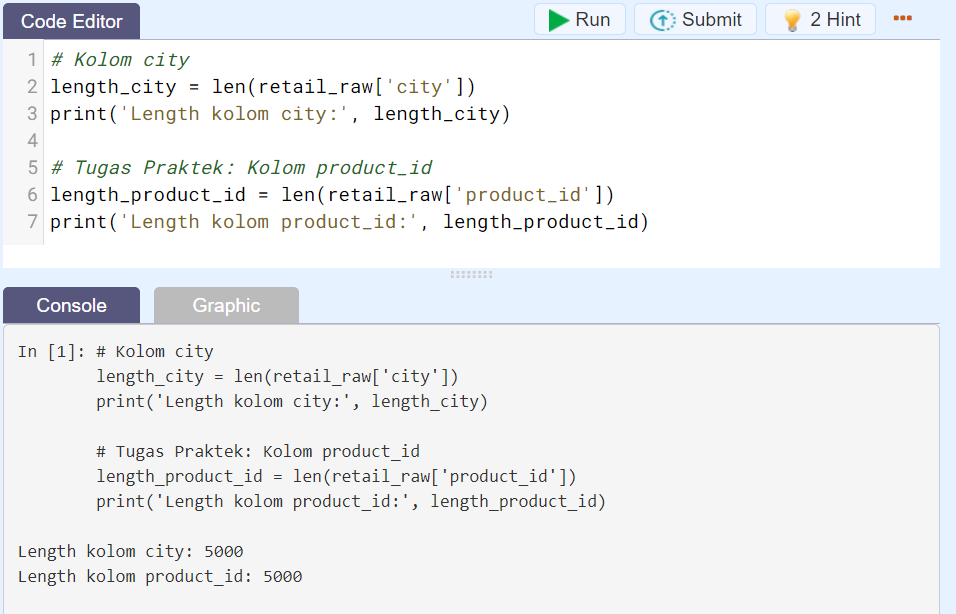
Jika dijalankan kode dengan baris yang ditunjukkan akan menghasilkan output:



**Tugas Praktek**:

Setelah membaca modul referensi Kroma, aku coba memulai  analisis dengan menginspeksi length dari kolom **product\_id** dari dataframe **retail\_raw!** Aku akan membuat syntaks Python untuk mencapai hal tersebut di code editor. Berikut cara yang akan kulakukan:

Jika dijalankan dengan  akan menghasilkan

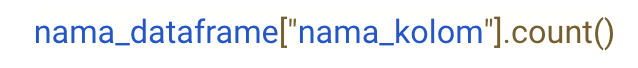


### Descriptive Statistics - Part 2

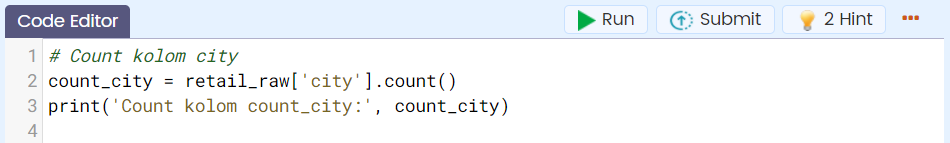
Aku melanjutkan mempelajari ke fungsi berikutnya, yaitu Count.

**Count**

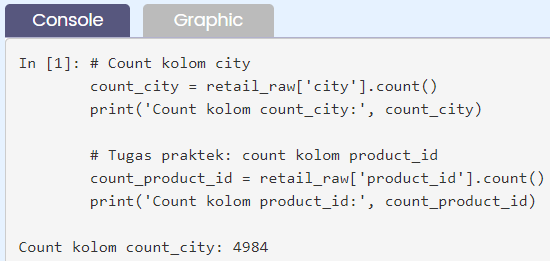
Fungsi count menghitung jumlah pengamatan non-NA / non-null dalam suatu series / column. Fungsi len akan hanya menghitung elemen dari kolom yang mempunyai nilai (exclude missing value).



Contoh penggunaan untuk mengetahui count dari kolom **city** dari dataframe **retail\_raw!**



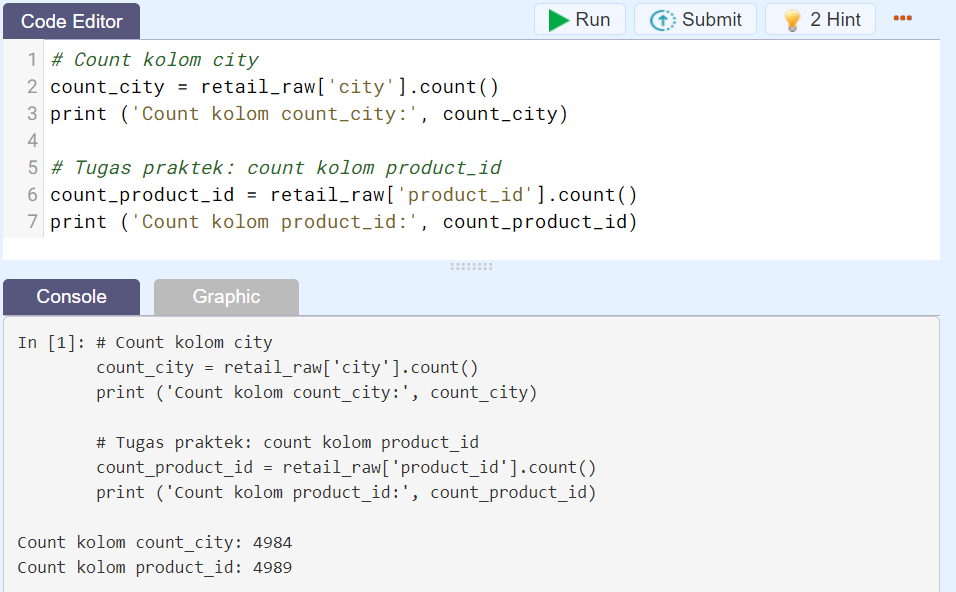
Jika dijalankan dengan  akan menghasilkan



**Tugas Praktek:**

Setelah yang tadi cukup lancar, aku berniat mengetahui jumlah non-null value dari kolom **product\_id** dari dataframe **retail\_raw** agar hasil analisisnya lebih lengkap.

Jika dijalankan dengan menekan  akan diperoleh output di console:

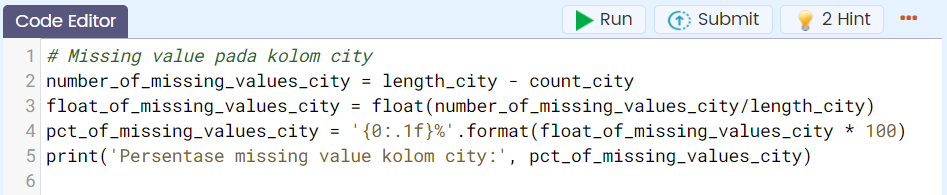


### Descriptive Statistics - Part 3

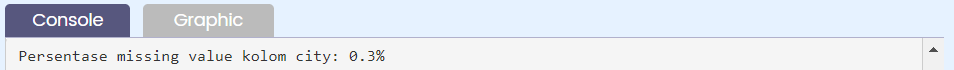
Lanjut ke bagian berikutnya.

**Missing Value**

Dengan Length dan Count, sekarang dapat menghitung jumlah missing-value. Jumlah nilai yang hilang adalah perbedaan antara Length dan Count.



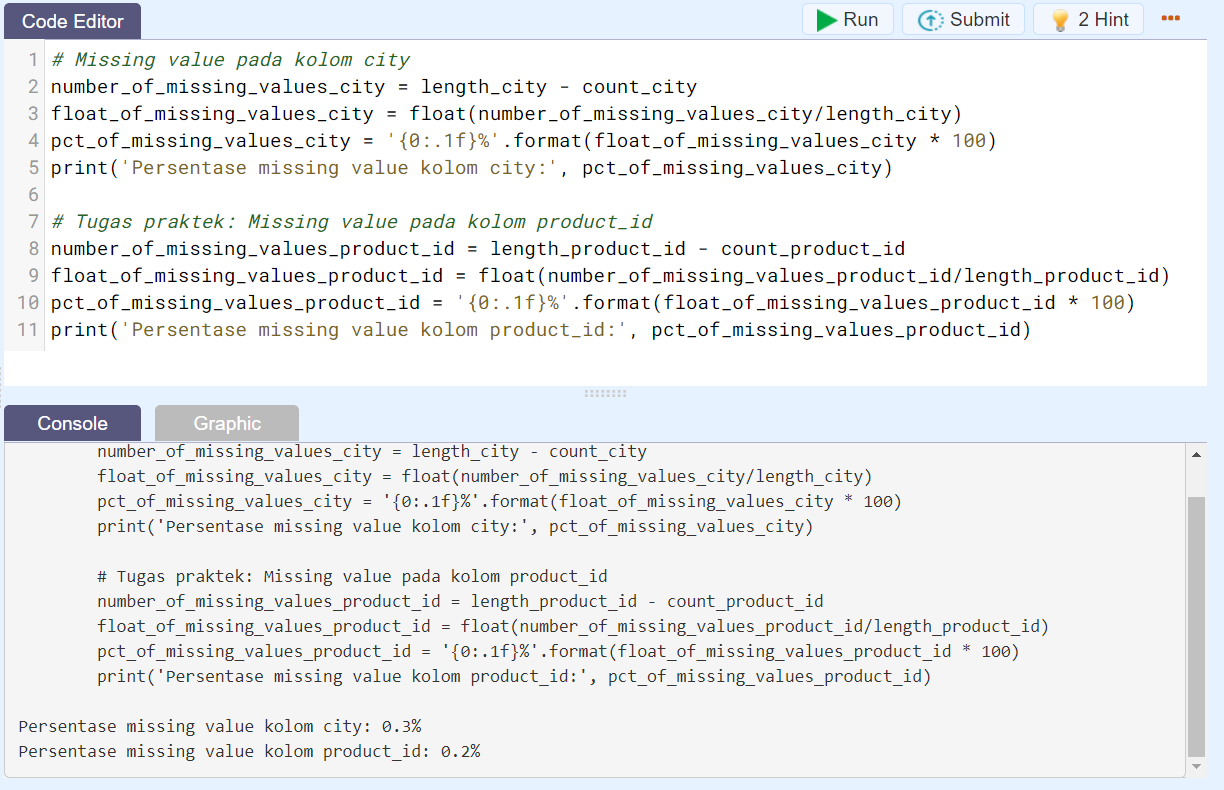
Jika dijalankan dengan  akan menghasilkan persentase data yang missing:



**Tugas Praktek**:

Yes! Aku berhasil mendapatkan length dan count dari **product\_id,** sekarang  aku perlu mengetahui jumlah missing-value dari kolom tersebut. Ini artiya aku perlu membuat syntaks untuk menghitung persentase missing-value dari **product\_id**. Kerjakanlah di code editor pada baris 7 s/d 11.

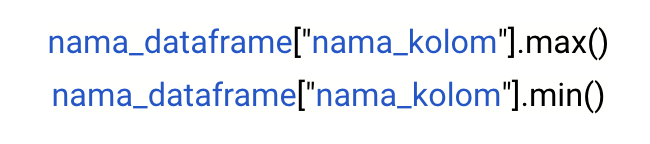
Jika telah ditulis dengan benar dan dijalankan dengan  diperoleh output berupa:



### Descriptive Statistics - Part 4

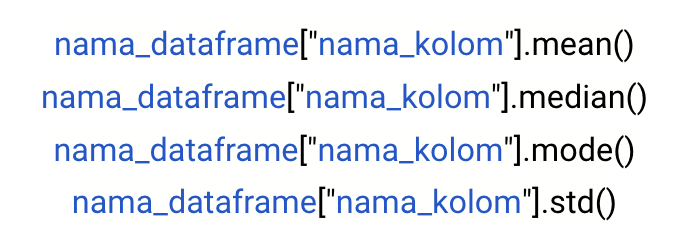
**Maximum dan Minimum**

Fungsi **max** dan **min** digunakan untuk mengetahui element terbesar dan terkecil dari suatu kolom di dataframe.

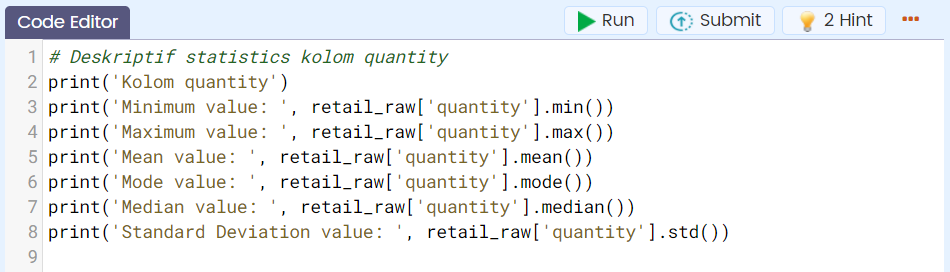


**Mean, Medium, Modus dan Standard Deviasi**

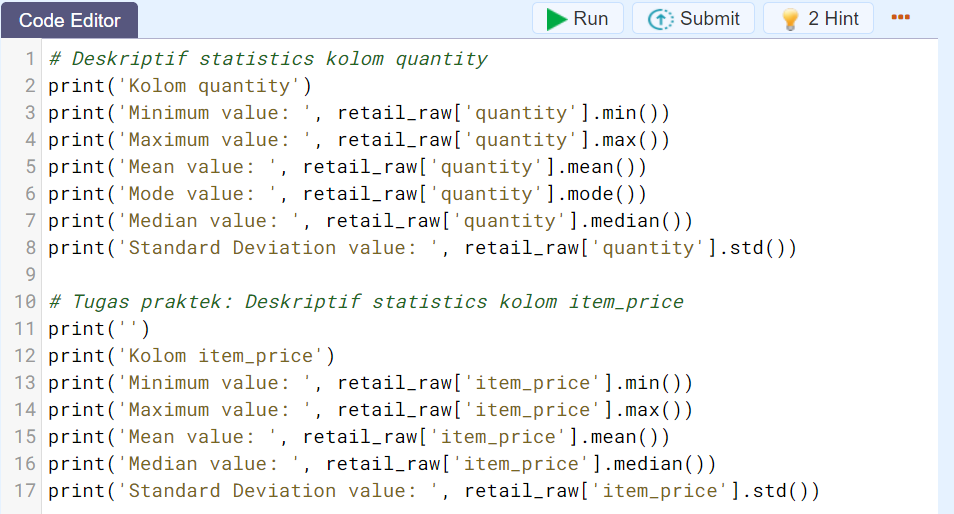
Fungsi **mean, medium, modus** dan **standard deviasi** digunakan untuk mengetahui pemusatan data dan persebarannya.

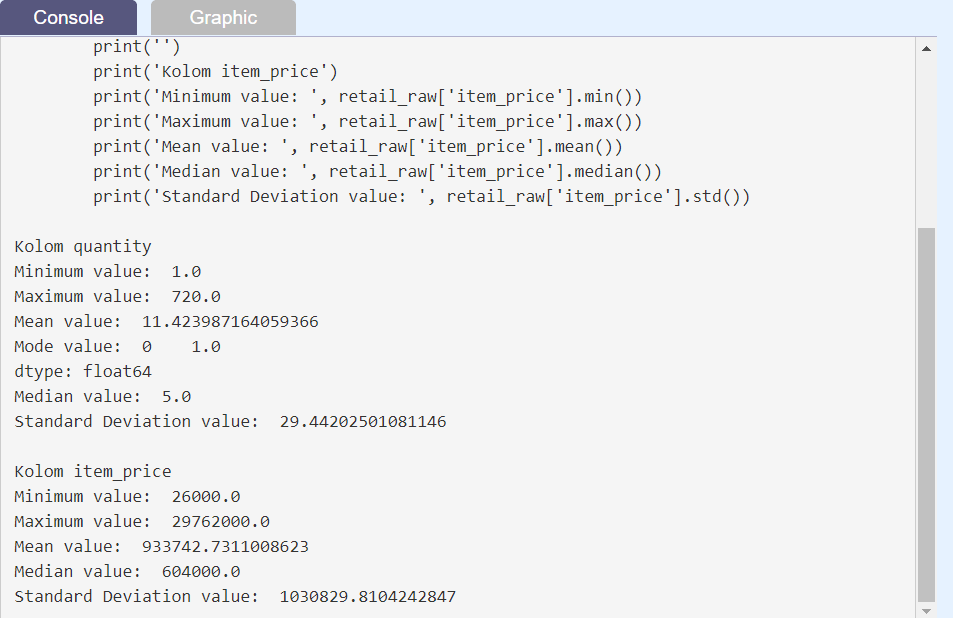


Contohnya:



Jika dijalankan dengan  akan menghasilkan:

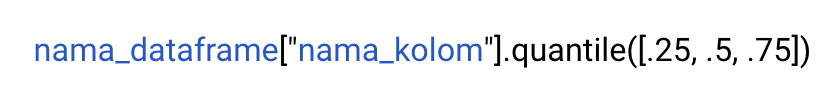




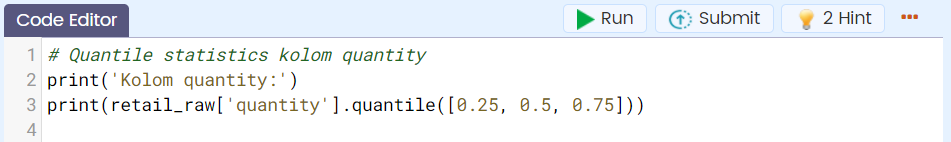
### Descriptive Statistics - Part 5

**Quantile Statistics**

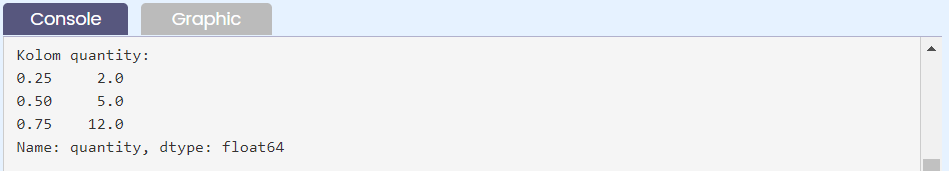
Quantiles adalah titik potong yang membagi distribusi dalam ukuran yang sama. Jika akan membagi distribusi menjadi empat grup yang sama, kuantil yang dibuat dinamai quartile. Jika dibagi kedalam 10 sepuluh group yang sama dinamakan percentile. Dalam kasus di bawah ini, ingin membagi distribusi menjadi empat grup atau quartile.



Contoh:

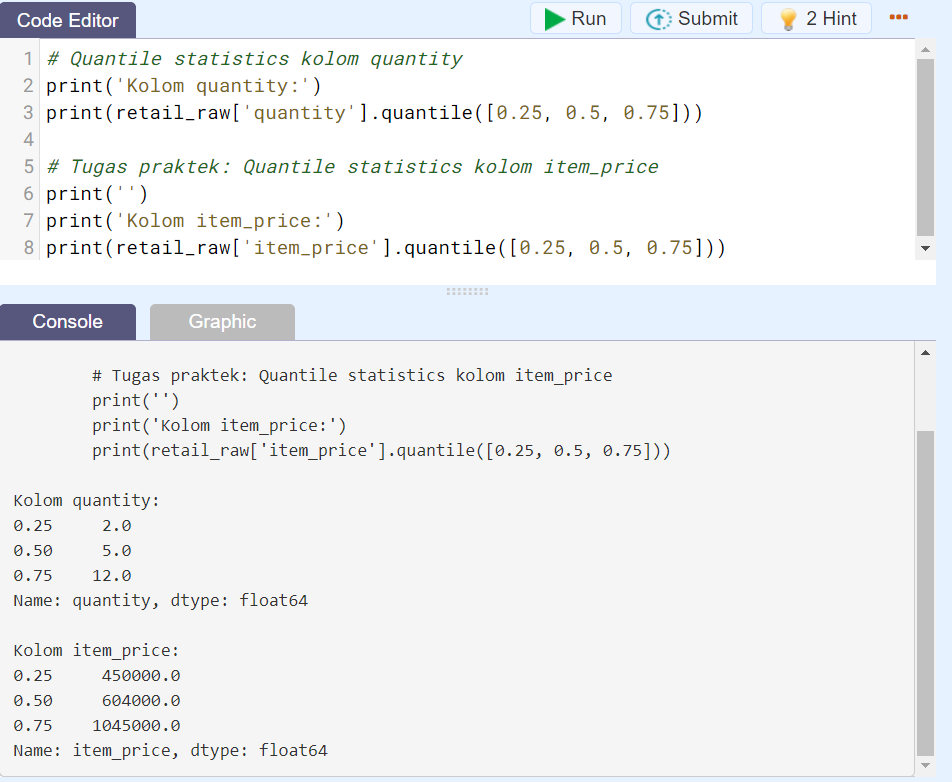


Jika dijalankan dengan  akan menghasilkan:



**Tugas Praktek:**

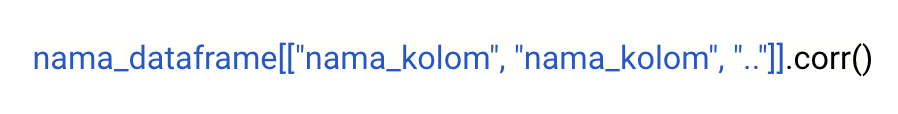
Baiklah, sekarang saatnya lanjut untuk membuat distribusi quartile dari **item\_price** dari dataframe **retail\_raw**. Kalau ini sih seharusnya mudah, aku mulai menyusun kodenya di code editor.



### Descriptive Statistics - Part 6

**Correlation**

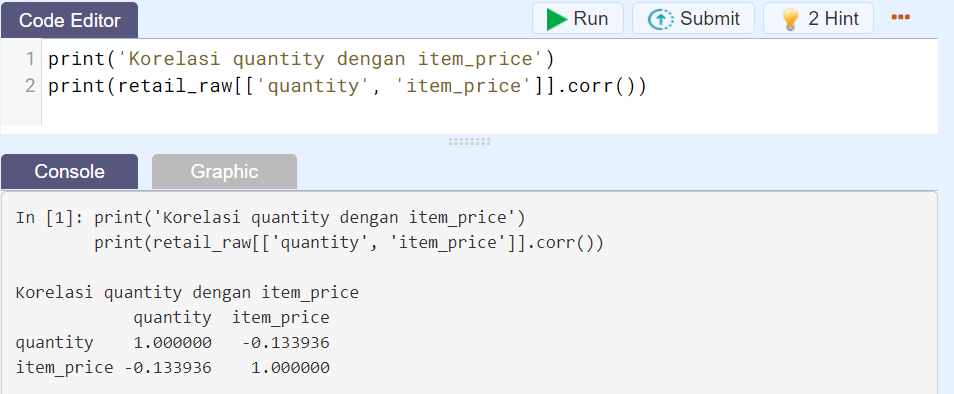
Korelasi adalah cara yang tepat untuk menemukan hubungan antara variabel numerik. Koefisien korelasi berkisar antara -1 hingga 1. Korelasi 1 adalah korelasi positif total, korelasi -1 adalah korelasi negatif total dan korelasi 0 adalah korelasi non-linear.



**Tugas Praktek:**

“Hmmm, korelasi antara **quantity** dan **item\_price** apa ya?” batinku. Aku perlu tahu untuk melengkapi syntaks. Aku coba cari di modul yang Kroma berikan tadi deh. Aku mengetikkan kode yang kupelajari sebelumnya di code editor.

Jika dengan benar ditulis dan dijalankan dengan  maka akupun memperoleh hasil berikut:



### Penggunaan Profiling Libraries

Seperti yang terlihat di atas, mengumpulkan statistik deskriptif dapat menjadi proses yang panjang. Pandas Profiling library memiliki function yang dapat membuat profiling data secara otomatis.

Untuk dapat menggunakannya, cukup dengan memanggil library:



Syntax:



Contoh penggunaan untuk dataset **retail\_raw**:



Mengingat output yang ditampilkan kaya akan penggunaan javascript, maka silakan klik tautan berikut ini untuk dibuka di tab baru browser kamu:

***\_\_link\_\_ProfileReport\_retail\_raw\_\_\_.html***

untuk melihat profiling report dari data frame retail\_raw yang telah DQLab buatkan.

Akan dapat memahami proses profiling data yang menerapkan EDA (exploratory data analysis) dari report yang di-generate menggunakan library pandas\_profiling melalui method ProfileReport.

## CHAPTER 2

### Apa itu Data Cleansing?

Fiuh! Materi tadi benar-benar penuh dengan latihan-latihan. Walaupun latihannya singkat, kalau jumlahnya banyak, ternyata menguras energi dan pikiranku juga. Apa mungkin Kroma sengaja memberikan referensi yang penuh latihan agar revisian aku nantinya bisa lebih lancar? Bisa juga caranya.

Aku mengecek sisa materi referensi yang diberikan, masih terdapat pembahasan seputar “Data Cleansing” lengkap dengan latihan-latihan singkat seperti sebelumnya. Baiklah, akan kujalani! Sisa ini saja kok, aku pasti bisa, batinku percaya diri.

Data Cleansing berarti proses mengidentifikasi bagian data yang salah, tidak lengkap, tidak akurat, tidak relevan atau hilang dan kemudian memodifikasi, mengganti atau menghapusnya sesuai dengan kebutuhan. Data Cleansing dianggap sebagai elemen dasar dari Data Science

Pada bagian ini, akan membahas data cleansing dari treatment terhadap missing data, treatment outliers, sampai deduplikasi data.

### Missing Data

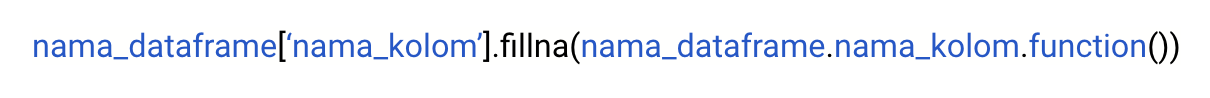
Dataset yang ditemui di real-world biasanya akan memiliki banyak missing value. Kemampuan untuk treatment missing value sangat penting karena jika membiarkan missing value itu dapat memengaruhi analisis dan machine learning model. Sehingga jika menemukan nilai yang hilang dalam dataset, harus melakukan treatment sedemikian rupa. Cara check kolom yang mempunyai missing value:



Cara treatment terhadap missing-value antara lain:

1. Leave as it is (dibiarkan)
2. Filling the missing value (imputasi)
3. Drop them (hapus row yang mengandung missing value)

**Imputasi** merupakan suatu metode treatment terhadap missing value dengan mengisinya menggunakan teknik tertentu. Bisa menggunakan mean, modus ataupun menggunakan predictive modelling. Pada modul ini akan membahas mengenai pemanfaatan function **fillna** dari Pandas untuk imputasi ini, yaitu



**.function()** yang dimaksud pada syntax di atas adalah penggunaa fungsi **.mean()** atau **.mode()**. Penggunaan fungsi **.mean()** atau **.mode()**ini bergantung pada kondisi yang mengharuskan menggunakan nilai rata - rata atau modus dari kolom yang akan diimputasi, seperti

nama\_dataframe['nama\_kolom'].fillna(nama\_dataframe.nama\_kolom**.mean()**)

atau

nama\_dataframe['nama\_kolom'].fillna(nama\_dataframe.nama\_kolom**.modus()**)

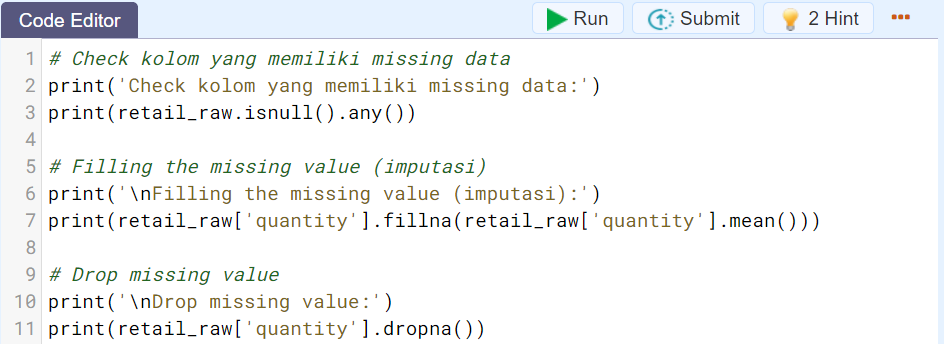
**Drop** row yang mengandung missing value. Dapat menggunakan function **dropna** dari Pandas.

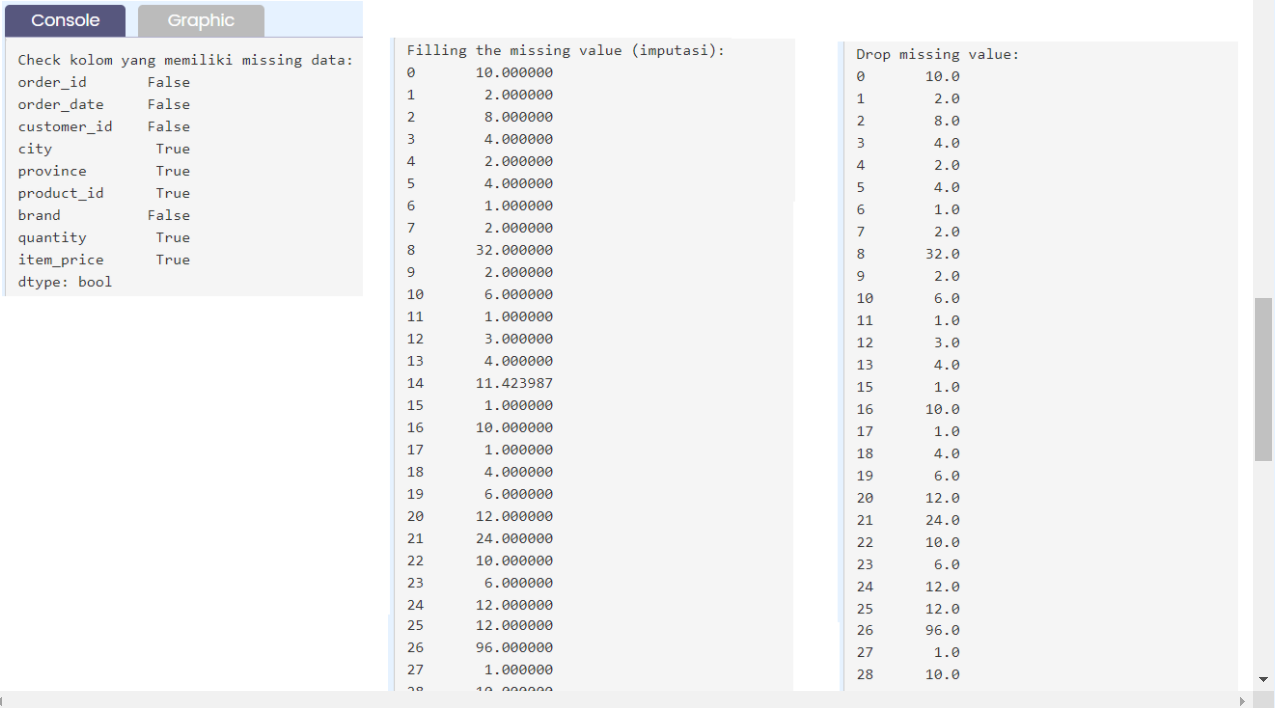


Untuk menangani missing data pada retail\_raw,

* Ceklah jika terdapat missing value pada variabel dataframe, dan kemudian cetak ke console
* Imputasi missing value pada kolom quantity dengan menggunaan nilai rataan (mean), dan kemudian cetak ke console
* Drop-lah missing value pada kolom quantity, dan kemudian cetak ke console

Jika ditulis dengan benar dan dijalankan dengan menekan  diperoleh output berikut:





### Tugas Praktek

Bagaimana, Aksara? Sudah dicoba dari modul yang saya kasih?” sapa Kroma. Kulihat ia baru keluar dari ruang meeting. Sedikit terkejut karena Kroma langsung menyapa dan menanyai prosesku, aku butuh beberapa detik sebelum menjawab.

“Sudah, sejauh ini paham sih. apalagi tiap subbab ada latihan singkatnya, banyak banget lagi! Cuma sedikit kendala saja soal missing value,” ujarku. Missing value memang bagian yang paling membuat penasaran karena sedari tadi aku berkutat dengan persoalan ini.

“Oke, bagus kalau memang lancar. Bisa temui saya di ruangan kalau dibutuhkan ya, Aksara.”

Aku mengangguk mantap. Aku tak mungkin bertanya pada Kroma dan memintanya mengajariku soal letak missing value-ku. Dan, tidak enak rasanya kalau merepotkan Senja lagi. Aku harus bisa mengandalkan diriku sendiri kali ini. Aku pun mengutak-atik dataku kembali selama hampir setengah jam dan menemukan jika missing value aku ada pada kolom item\_price.

“Ah, akhirnya!” Dengan cepat aku melengkapi missing value tersebut dengan **mean** dari **item\_price.** Berikut caraya:



### Outliers

Outliers merupakan data observasi yang muncul dengan nilai-nilai ekstrim. Yang dimaksud dengan nilai-nilai ekstrim dalam observasi adalah nilai yang jauh atau beda sama sekali dengan sebagian besar nilai lain dalam kelompoknya.

Cara treatment terhadap outliers antara lain:

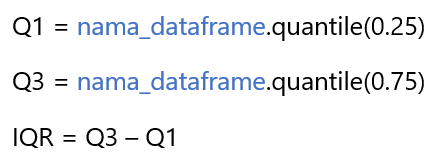
1. Remove the outliers (dibuang)
2. Filling the missing value (imputasi)
3. Capping
4. Prediction

Pada umumnya, outliers dapat ditentukan dengan metric **IQR** (interquartile range).

Rumus dasar dari IQR: Q3 - Q1, dan data suatu observasi dapat dikatakan outliers jika memenuhi kedua syarat dibawah ini:

* < Q1 - 1.5 \* IQR
* > Q3 + 1.5 \* IQR

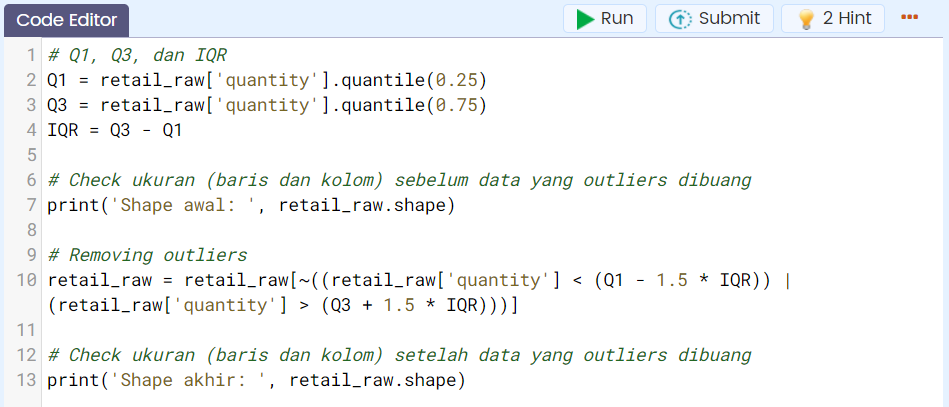
Syntax di Python:



Kemudian untuk membuang outliersnya:



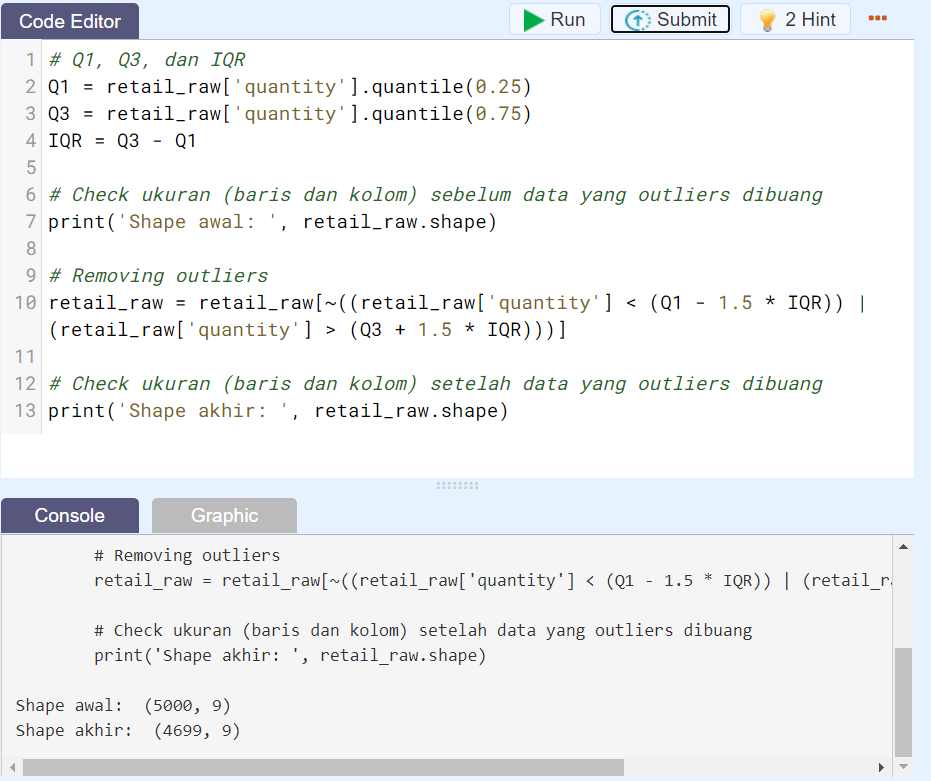
Mari melihat penggunaannya pada dataframe retail\_raw untuk kolom quantity:



Ukuran dataframe retail\_raw sebelum dan setelah dibuang outliers pada kolom quantity yaitu:



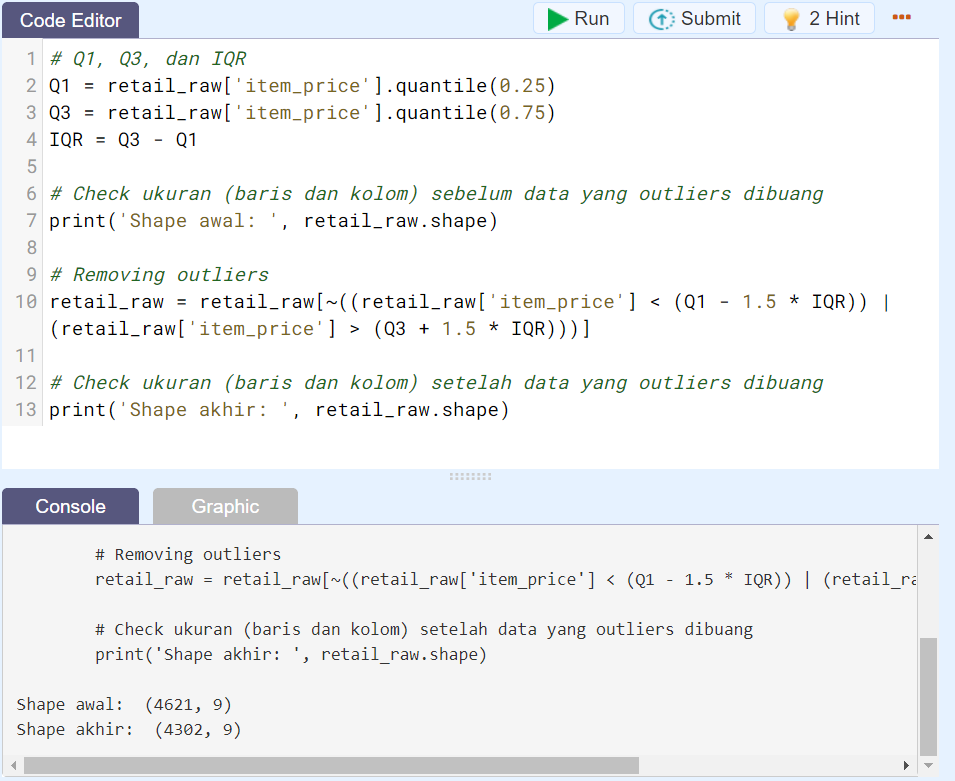
***Note***: Langsung di-submit saja ya tanpa di-run :)



### Tugas Praktek

Setelah berhasil mengatasi missing value tadi, aku memutuskan untuk mencoba menemukan sejumlah outliers menggunakan IQR. Dengan begitu, aku bisa mengetahui berapa IQR dari variabel **item\_price**.

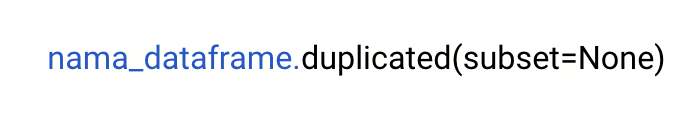
Caranya dengan mengetikkan bagian yang kosong pada live code editor.



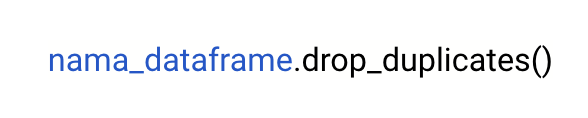
### Deduplikasi Data

**Duplikasi data** merupakan data dengan kondisi pada row-row tertentu memiliki kesamaan data di seluruh kolomnya. Tentunya ada data yang duplikat dalam dataset yang dimiliki. Kondisi duplikasi harus diatasi dengan jalan mengeliminir baris yang mengalami duplikasi, sehingga proses ini dikenal dengan **deduplikasi**.

**Untuk mencek duplikasi data**:



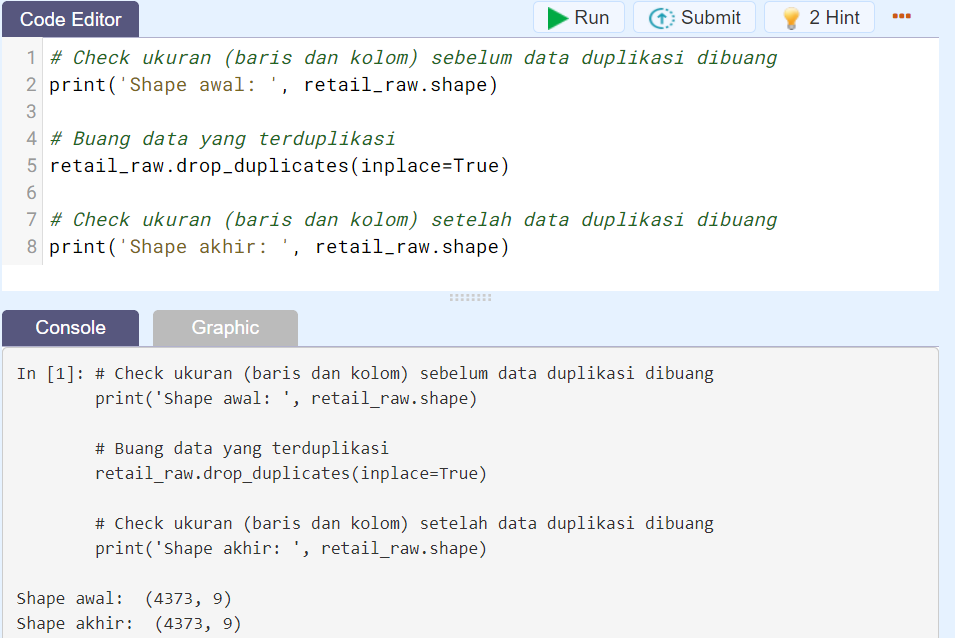
**Syntax untuk membuang duplikasi**:



### Tugas Praktek

Terakhir lagi menuju selesai! Aku tinggal membuang duplikasi data dari dataset **retail\_raw**.

Aku akan melengkapi kode pada live code editor berikut.



## CHAPTER 3

### Pendahuluan

“Revisianmu sudah saya pelajari kembali. Hasilnya cukup memuaskan. Kamu berhasil menganalisisnya dengan tepat dan mendalam. Saya jadi tertarik untuk memberikanmu sebuah proyek,” ujar Kroma setelah mengecek satu per satu hasil revisi pekerjaanku.

Aku tidak tahu harus senang atau gugup saat ini. Proyek yang dipercayakan langsung dari Kroma? Jujur saja, aku cukup gelisah. Beragam kemungkinan menggantung di dadaku, apakah aku mampu? Aku hanya anak baru yang selama ini banyak dibimbing Senja dan dibantu Andra.

Tanpa sadar aku terdiam lama. “Jangan khawatir. Saya yakin proyek ini sesuai dengan program belajarmu selama ini bersama Senja dan Andra. Jika ada penyesuaian dan pengembangan, hanya beberapa. Saya yakin kamu bisa melakukan improvisasi dengan baik,” lanjut Kroma.

Aku curiga atasanku ini memang bisa mendengar suara hati! Tapi mengetahui Kroma menaruh harapan dan kepercayaan padaku, aku seharusnya bangga. Dengan mengumpulkan kepercayaan diriku, aku mengiyakan proyek tersebut.

“Baik. Lima menit lagi saya akan mengirimkan data dari cabang kita di daerah G. Tolong bersihkan saja dulu datanya, detailnya yang harus kamu lakukan akan saya arahkan juga di email.”

“Baik.”

Aku pun keluar dari ruangan. Setelah mengatur napas sejenak, aku bergegas kembali ke meja. Apalagi setelah aku mendengar notifikasi email masuk di ponselku. Dan tahu kalau itu pasti dari Kroma.

### Case Studi: Data Profiling

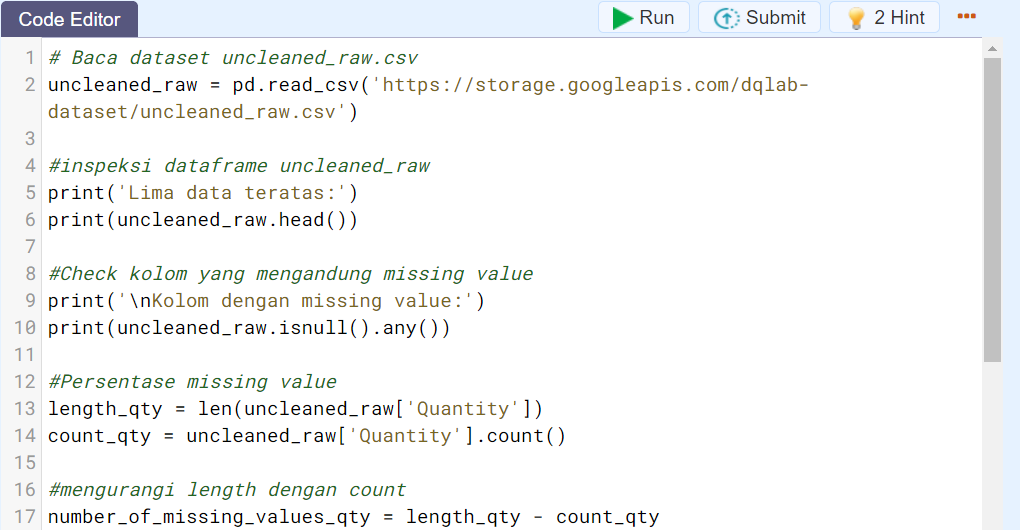
Dear Aksara,

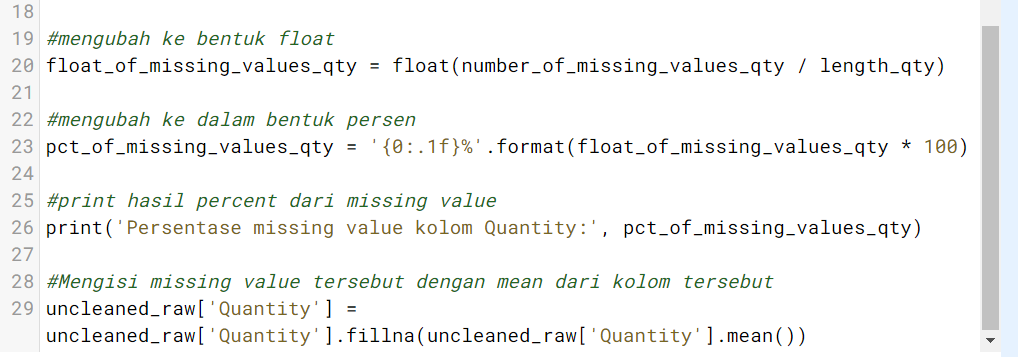
Tolong proses dataset terlampir yang  disimpan dalam bentuk csv bernama '**https://storage.googleapis.com/dqlab-dataset/uncleaned\_raw.csv**'.

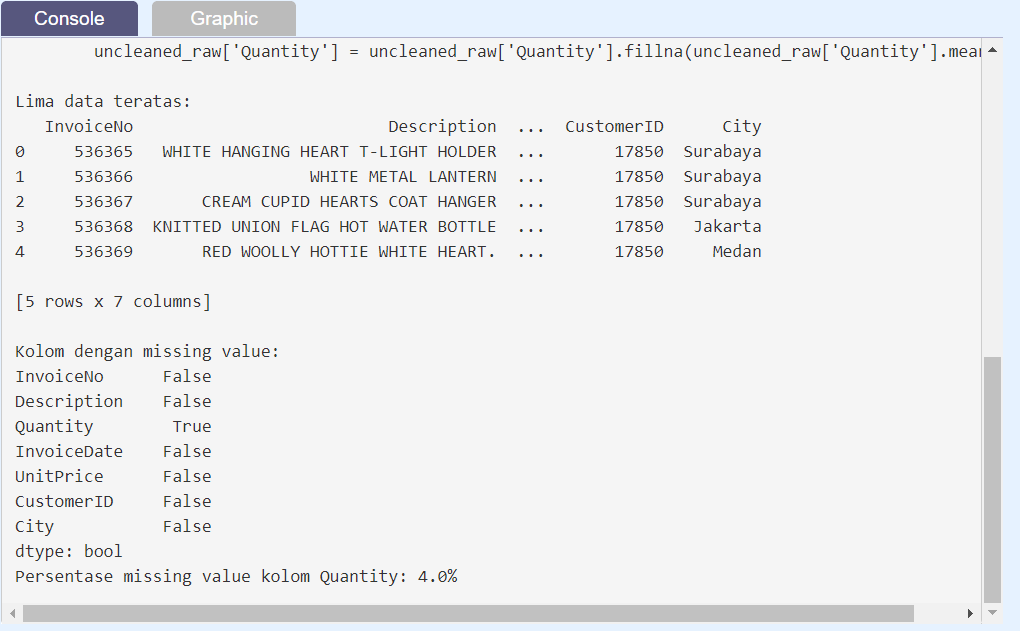
Kamu bisa memprosesnya dengan cara berikut:

1. Import dataset csv ke variable bernama **uncleaned\_raw**
2. Inspeksi dataframe **uncleaned\_raw**
3. Check kolom yang mengandung missing value, jika ada, kolom apakah itu dan berapa persen missing value pada kolom tersebut?
4. Mengisi missing value tersebut dengan mean dari kolom tersebut!

Setelah membaca email tersebut, aku pun memulai kode programnya di code editor.



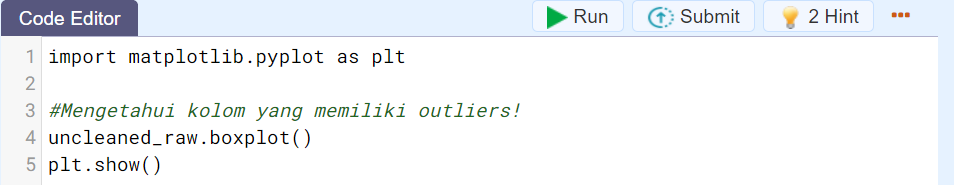


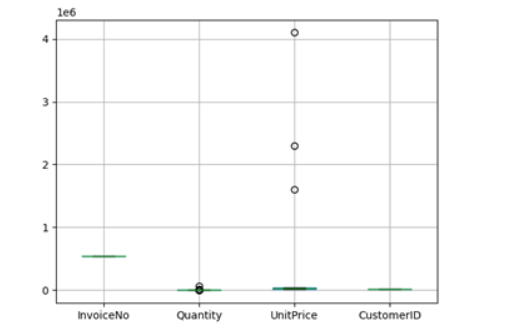


### Case Study: Data Cleansing - Part 1

Untuk memprosesnya bisa dilakukan dengan cara berikut:

1. Mengetahui kolom yang memiliki outliers! Gunakan visualisasi dengan boxplot pada dataframe uncleaned\_raw.

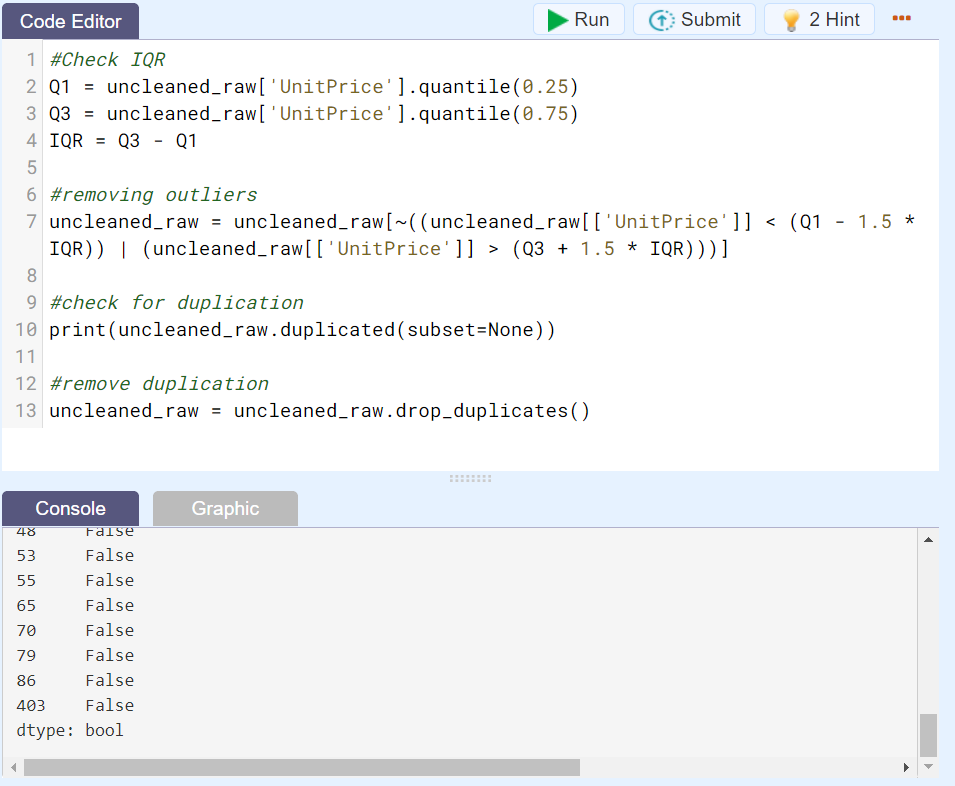




### Case Study: Data Cleansing - Part 2

Langkah selanjutnya bisa dilakukan dengan cara berikut:

1. Melakukan proses removing outliers pada kolom UnitPrice.
2. Checking duplikasi and melakukan deduplikasi dataset tersebut!



### Penutup/Kesimpulan

**Congratulations!** Aku telah menyelesaikan modul **Data Quality with Python for Beginner**. Berdasarkan materi-materi yang diberikan Kroma yang telah kupelajari dan praktekkan dalam modul ini, aku telah mendapatkan pengetahuan (knowledge) dan praktek (skill) yang diantaranya :

* Mampu melakukan profiling data baik untuk data angka maupun teks
* Mampu membuat rule untuk data cleansing (standarisasi data, value lookup, dan treatment duplikasi data)