# Data Visualization with Python Matplotlib for Beginner - Part 2

## CHAPTER 1

### Pendahuluan

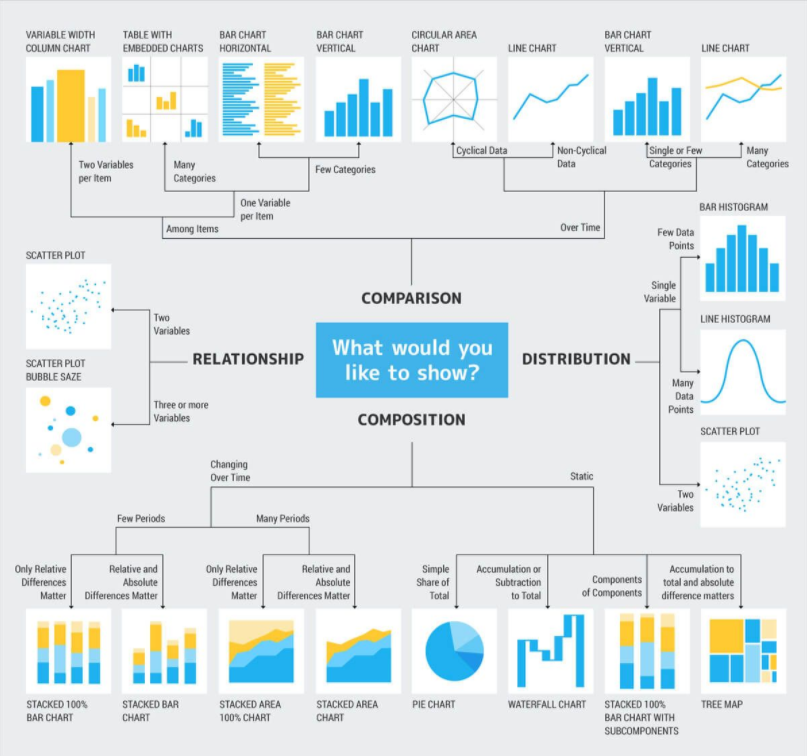
Siang nanti aku akan diikutkan dalam rapat (lagi!) oleh Andra untuk menjelaskan grafik yang kemarin kukerjakan. Sedikit tegang kuakui, karena ini pertama kalinya aku benar-benar diberi kepercayaan untuk bicara dalam rapat, menjelaskan pula! Apalagi Senja juga hadir dalam rapat itu, aku tak boleh mengecewakan hasil bimbingannya.

Aku juga sudah sempat diberi bekal oleh Senja untuk membaca seputar jenis-jenis visualisasi data dan tujuannya. Materi ini membantuku agar lebih memahami tujuan dari hasil kerjaku dalam memvisualisasikan data kemarin.

“Baiklah, tenangkan diri dulu!” kataku pada diri sendiri sambil memulai untuk mempelajari modul ini.

### Mengenal Jenis-Jenis Visualisasi Data dan Tujuannya

Ada banyak sekali jenis visualisasi data, dari yang sederhana hingga yang kompleks. Setiap jenis visualisasi biasanya lebih tepat digunakan untuk suatu tujuan tertentu. Secara garis besar, ada empat tujuan visualisasi yang disampaikan oleh Andrew Abela sesuai gambar berikut.



Berikut penjelasannya:

**1. Perbandingan/Komparasi**

Jika ingin membandingkan suatu sekumpulan nilai dengan nilai lainnya, ingin melihat mana yang lebih besar, mana yang lebih tinggi, mana yang naik, dsb. Untuk data yang terkait dengan waktu, visualisasi yang digunakan biasanya berupa line-chart. Sedangkan, jika yang dibandingkan adalah kategorikal, maka lebih cocok menggunakan bar-chart.

**2. Distribusi**

Bila ingin melihat distribusi/persebaran data di suatu variabel. Visualisasi yang lazim digunakan adalah histogram. Bisa melihat apakah datanya banyak terkumpul di nilai-nilai kecil, berat di kanan, atau simetris, ataupun berdistribusi normal. Untuk melihat distribusi di dua variabel, scatterplot juga bisa digunakan.

**3. Komposisi**

Digunakan untuk melihat komposisi dari suatu variabel, jika di-breakdown terhadap suatu dimensi data. Visualisasi yang biasa digunakan adalah stacked bar-chart untuk data kategorikal, atau stacked line-chart untuk data terkait waktu.

**4. Keterhubungan/Relasi**

Ketika ingin melihat keterhubungan antara suatu variabel dengan variabel lain. Misalnya ingin cek, jika variabel A semakin tinggi, apakah variabel B juga semakin tinggi, atau justru lebih rendah, atau tidak ada keterhubungannya. Visualisasi yang biasa digunakan untuk tujuan ini adalah scatter-plot.

## CHAPTER 2

### Membuat Multi-Line Chart

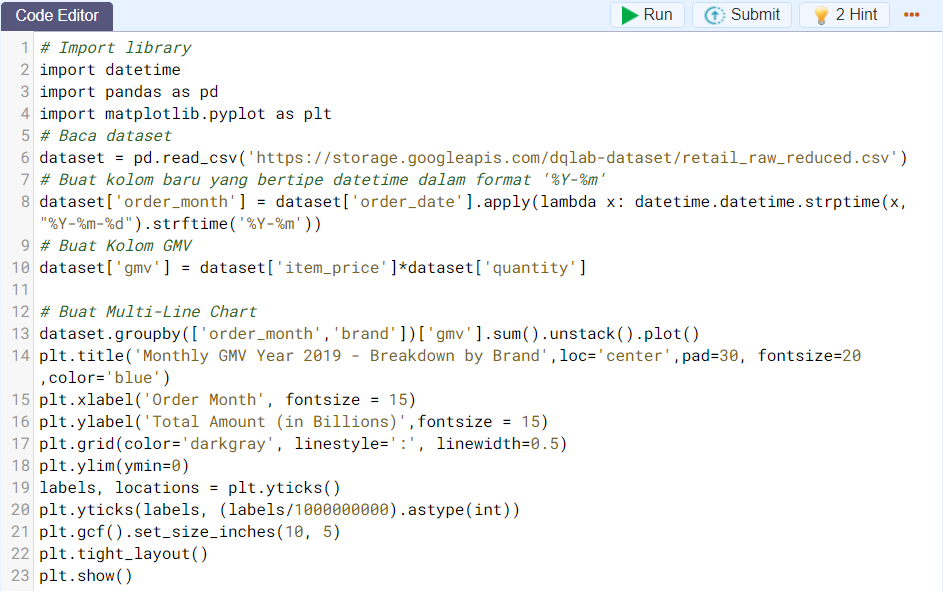
Bagian paling menegangkan dari hari ini berhasil kulewati. Presentasiku berjalan lancar dan Senja masih saja membahasnya.

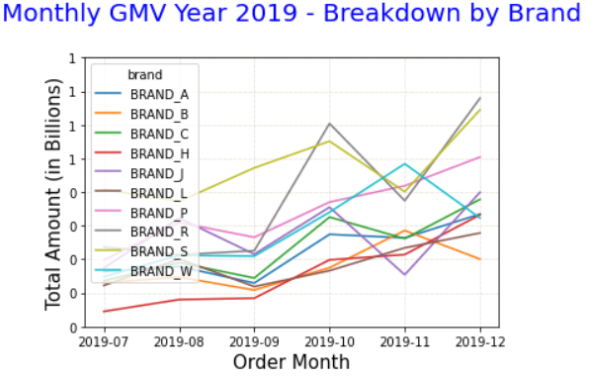
“Kerja bagus, Aksara. Dari grafik ini cabang C jadi bisa lihat GMV mereka di bulan Oktober 2019 yang meningkat dratis."

Aku tersenyum lebar. Apa lagi yang paling membahagiakan selain tahu hasil kerja bisa bermanfaat? Di tengah antara aku dan Senja, Andra ikut menimpali. "“Agar hasilya lebih lengkap lagi, bisa minta tolong gali lebih dalam, apa yang membuat GMV ini meningkat? Jadi kita bisa berikan data ini ke Business Development agar mereka bisa menentukan strategi bisnis ke depannya,” ujar Andra.

“Satu lagi, Untuk pekerjaan ini, kamu akan dibantu Senja karena perlu berkomunikasi langsung dengan kepala cabang.”

“Oke, Ndra. Aku breakdown datanya berdasarkan dimensi yah, misalnya berdasarkan brand atau province. Nanti akan terlihat bagian mana yang paling berperan dalam kenaikan GMV ini,” jawabku mantap sambil mulai menyusun kodenya.





### Kustomisasi Legend

“Kalau saya lihat, kita punya komponen baru ya di sini, namanya legend. Berhubung brand-nya ada banyak, dan defaultnya di tengah makanya jadi menutupi grafiknya. Solusinya adalah kita pakai function khusus ya,” ujar Senja memberi masukan terhadap hasil kerjaku. Senangnya  
Senja masih bersedia membantuku lagi.

Atur komponen legend menggunakan function **plt.legend()** seperti yang ditunjukkan oleh potongan kode berikut ini:

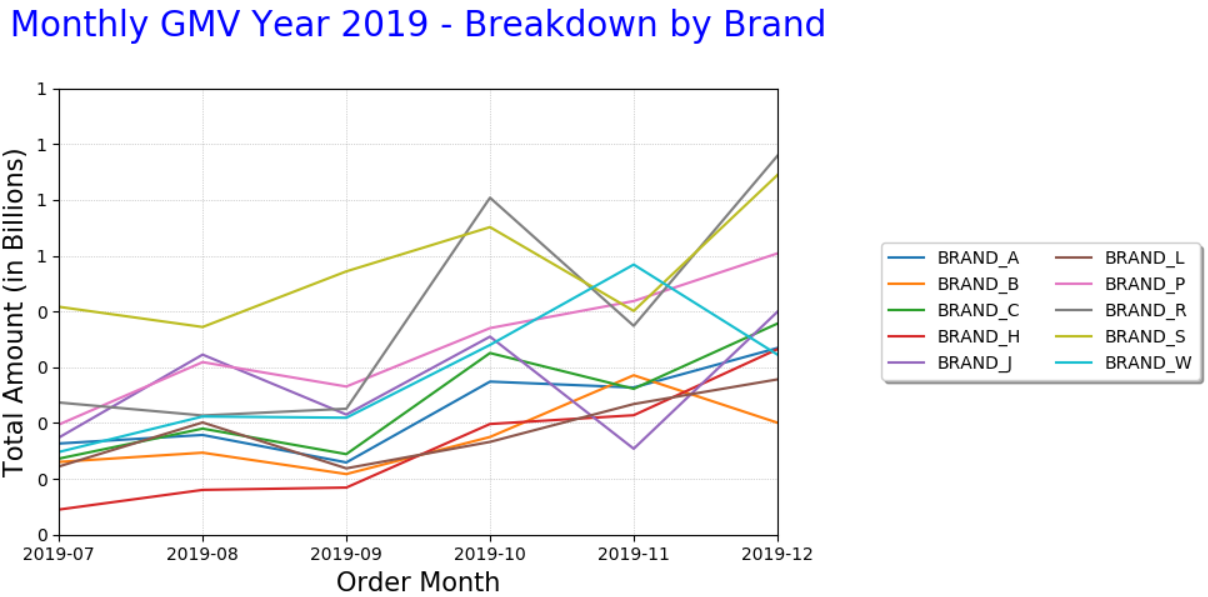


Beberapa parameter yang bisa ditambahkan untuk legend:

* **loc**: untuk menentukan posisi legend, berikut beberapa lokasi legend yang bisa didefinisikan:
  + 'upper left', 'upper right', 'lower left', 'lower right':legend diletakkan di pojok dari axes (atas kiri, atas kanan, bawah kiri, atas kiri)
  + 'upper center', 'lower center', 'center left', 'center right': legend diletakkan di tepi axes (atas tengah, bawah tengah, tengah kiri, tengah kanan)
  + 'center': legend diletakkan di tengah-tengah axes
  + 'best': matplotlib akan memilih satu dari sekian kemungkinan lokasi legend di atas yang paling tidak overlap dengan isi grafik
* **bbox\_to\_anchor**: biasanya digunakan untuk adjust lokasi dari legend. Bisa berisi 2 angka yang menunjukkan koordinat x dan y (misal (1.6,0.5) berarti geser 1.6 ke kanan dan 0.5 ke atas). Bisa juga berisi 4 angka, angka ketiga dan keempat menyatakan width (lebar) dan height (tinggi) dari legend.
* **shadow**: jika diisi True, maka kotak legend akan memiliki bayangan.
* **ncol**: jumlah kolom dari isi legend, defaultnya adalah 1
* **fontsize**: ukuran huruf pada legend
* **title**: memberikan judul pada legend
* **title\_fontsize**: ukuran huruf pada judul legend

Dengan menjalankan keseluruhan kode di code editor dengan menekan , selanjutkan akan diperoleh tampilan grafik multi-line chart yang telah dikustomisasi legend-nya



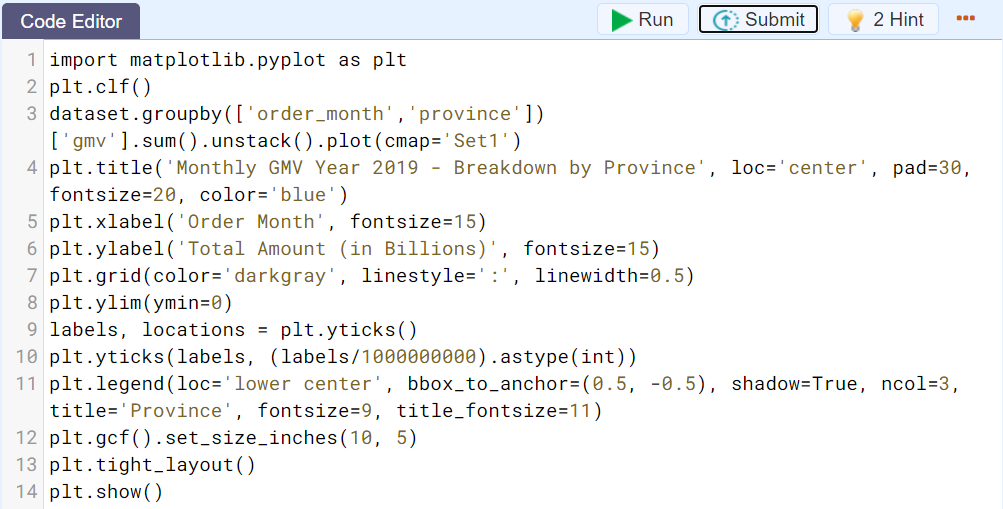


### Kustomisasi Colormap

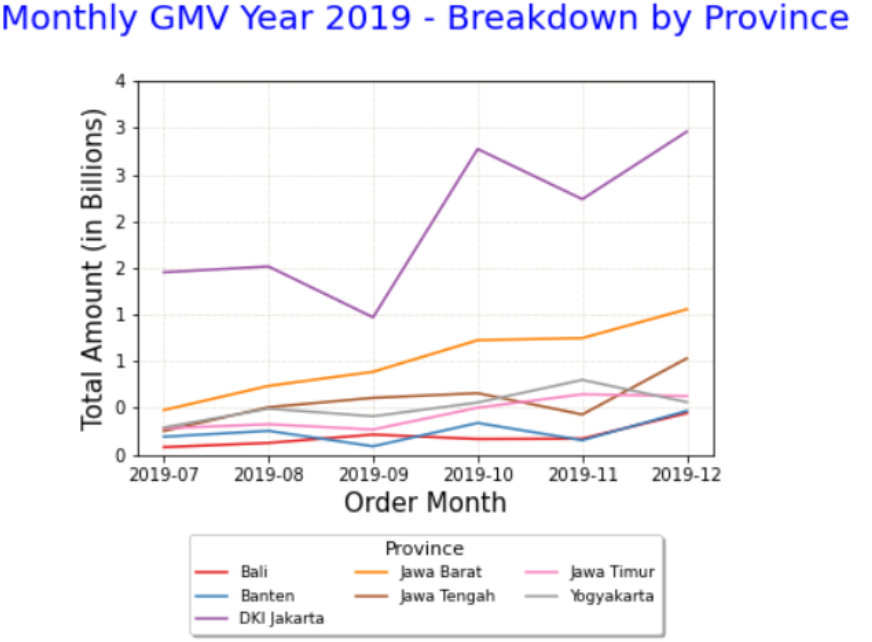
“Kalau sudah di breakdown by brand seperti ini terlihat ya trend sepanjang Oktober-Desember 2019 ini mirip semua. Jadi kemungkinan tidak ada faktor dari brand yang membuat GMV kita naik. Aksara, coba perlihatkan data breakdown by province,” perintah Senja sambil menunjuk layar laptopku.

Kalau tidak pernah dibimbing Senja sebelumnya, aku mungkin akan berpikir Senja tipe yang perfeksionis sekali. Tapi aku tahu ini sudah jadi kebiasaannya untuk lebih teliti. Caranya justru membuat hasil kerjaku lebih maksimal.

“Siap!” Aku pun segera mengganti brand menjadi province. Sekaligus buat legend-nya di bawah dengan judulnya dengan detail seperti ini:



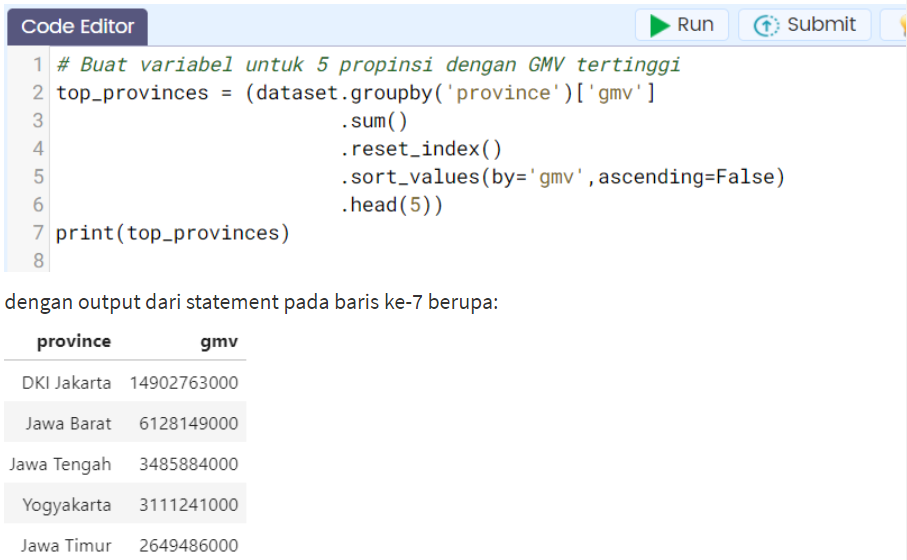
Dengan menjanlankan kode yang telah dibuat dengan , diperoleh tampilan grafik multi-line chart yang baru seperti yang ditunjukkan berikut ini:



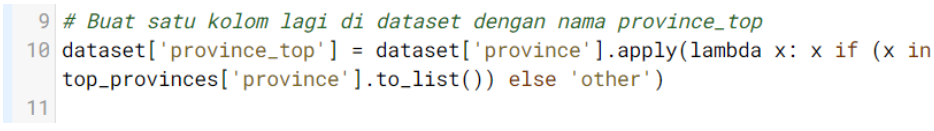
### Membuat Line Chart GMV Breakdown by Top Provinces

“Oke, sudah terlihat cukup jelas jika kenaikan GMW disebabkan oleh provinsi. Berhubung kita fokusnya ke provinsi-provinsi yang GMV-nya besar, kalau begitu kita highlight untuk top 5 provinsi saja, yang lainnya kita gabungkan menjadi ‘other’,” ujar Senja menyimpulkan.

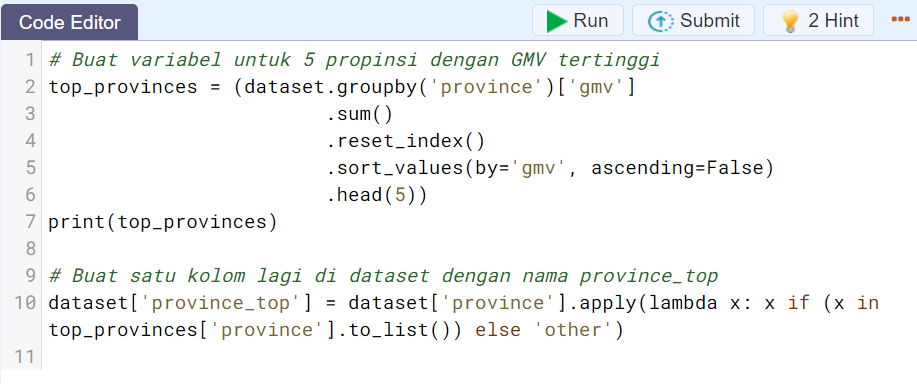
Aku berusaha membantu dengan segera mengecek lima provinsi dengan GMV tertinggi terlebih dulu:

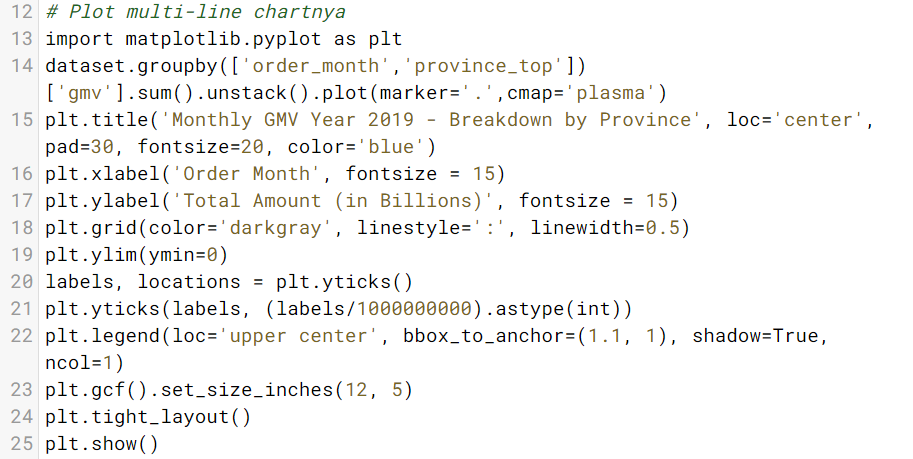


Setelah itu, membuat kolom baru, misalnya ‘province\_top’. Seperti sebelumnya, manfaatkan function apply & lambda.

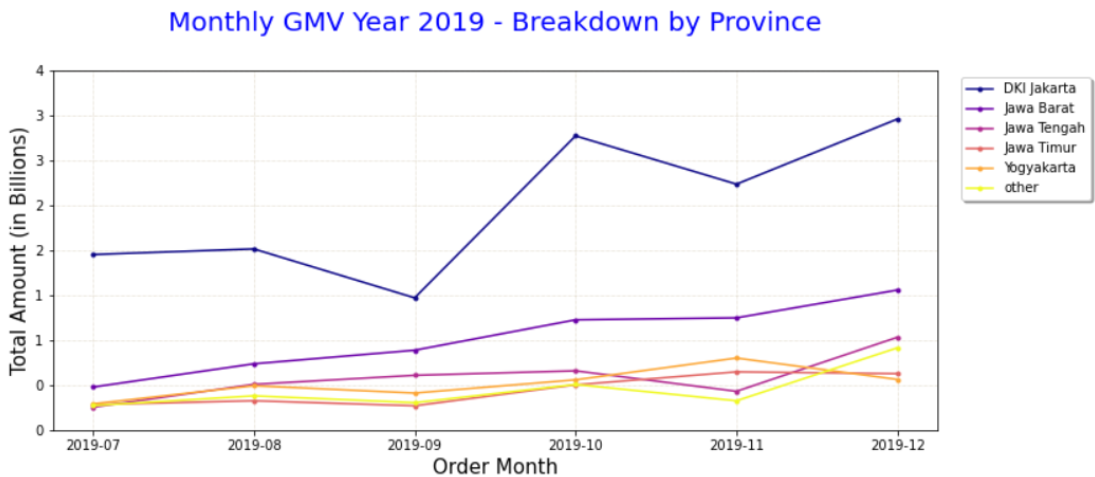


Kemudian, setelah memiliki kolom baru, grafiknya bisa di update. Sebagai catatan, untuk mengubah ukuran figure, juga bisa menggunakan function plt.gcf().set\_size\_inches yang diletakkan di akhir code, sebelum plt.show().





Hasil grafiknya menjadi seperti ini:



### Membuat Anotasi

Setelah mengutak-atik anotasi, aku sedikit bingung. “Nja, ini gimana ya?” tanyaku akhirnya. Senja yang sedang duduk di sebelahku memerhatikan kodeku.

“Begini, Aksara. Anotasi itu mirip seperti teks, hanya saja ada suatu titik tertentu yang kita tunjuk untuk diberikan informasi tambahan. Selain itu, kita bisa menambahkan panah melalui anotasi,” jelas Senja sambil mengambil buku catatan dan menulis di sana.

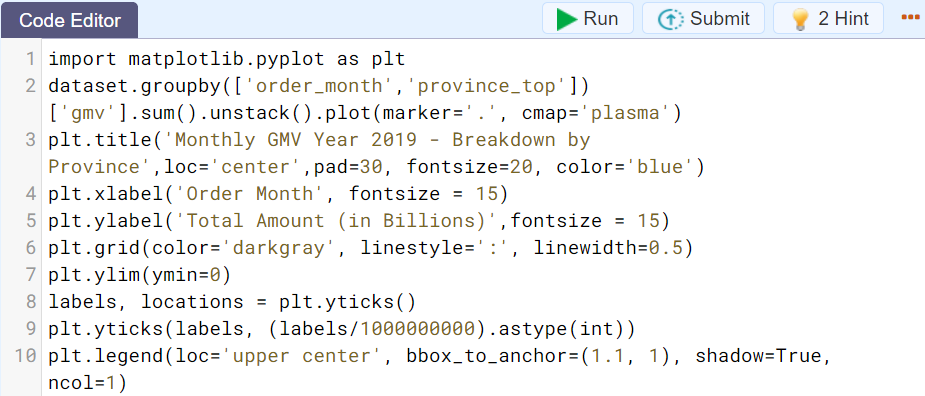
Aku diam memerhatikan:

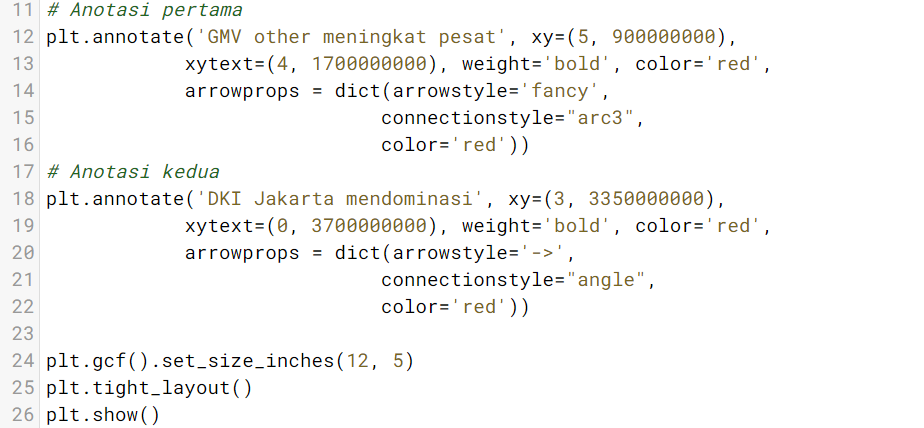
Untuk membuat annotate, gunakan function  plt.annotate. Parameter pertama adalah isi teksnya, dan parameter kedua adalah koordinat dari point yang dianotasi.

Selain itu ada beberapa parameter untuk anotasi yang bisa ditambahkan:

* **xytext**: Koordinat dari teks. Jika tidak diisi,maka teks akan diletakkan di koordinat point yang dianotasi (parameter kedua)
* **arrowprops**:Konfigurasi dari panah yang ditambahkan di anotasi, berupa dictionary. Beberapa hal yang bisa diset:
  + **arrowstyle**: menentukan bentuk dari panah penunjuk. Isinya bisa berupa bentuk seperti '->', '-|>', '<->', '-[', dsb. Bisa juga berupa tulisan seperti 'fancy', 'simple', 'wedge', etc.
  + **connectionstyle**: menentukan bentuk dari garis panahnya. Ada beberapa nilai yang bisa digunakan, misalnya 'arc' dan 'arc3' yang berupa garis lurus, 'angle' untuk garis berbelok siku, 'angle3' untuk garis berbelok lengkung, atau 'bar' untuk berbelok siku dua kali.
  + **color**: menentukan warna dari panah

Berikut adalah kode yang digunakan untuk menganotasi multi-line chart dari 5 propinsi dengan GMV tertinggi





Setelah menuliskan kedua perintah anotasi di atas di code editor, jalankanlah kode program tersebut dengan  dan kemudian akan peroleh tampilan grafik seperti berikut ini



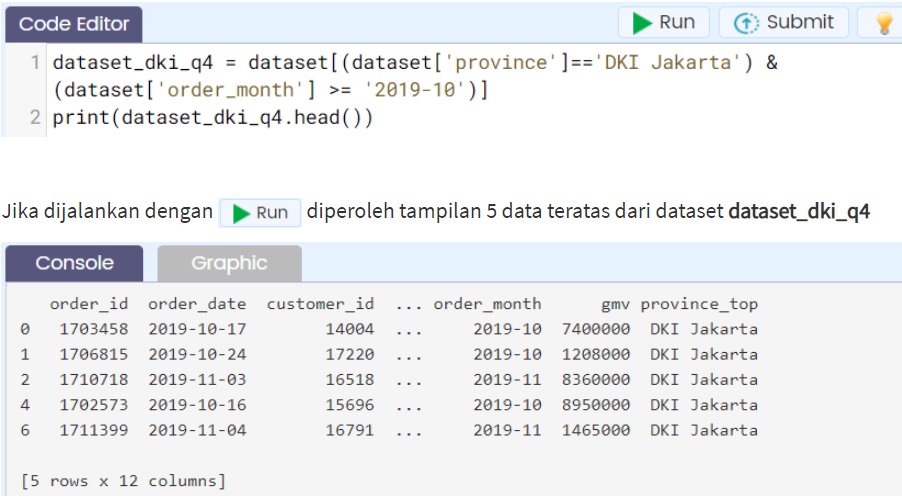
## CHAPTER 3

### Membuat Subset Data

“Sudah aku sampaikan ke kepala cabang. Ini insight yang menarik menurut beliau. Ada rikues kalau provinsi ‘other’ ini, kita analisis lagi di lain waktu saja dan fokuskan untuk analisis behavior transaksi di DKI Jakarta pada bulan Oktober-Desember 2019 ini, karena trend-nya mirip dengan overall GMV. Kita fokuskan ke sana ya,” pinta Senja.

“Oke, aku coba cek dulu perbandingan GMV di kota-kota di DKI Jakarta itu seperti apa,” sahutku segera beraksi.

“Iya, berhubung kita mau fokus ke provinsi DKI Jakarta dan bulan Oktober-Desember 2019, kita filter saja datanya dan disimpan ke dataframe baru", tukas Senja sembari membantu mengarahkan kursorku untuk memberikan contoh menggunakan data frame **dataset\_dki\_q4.**



### Membuat Pie Chart

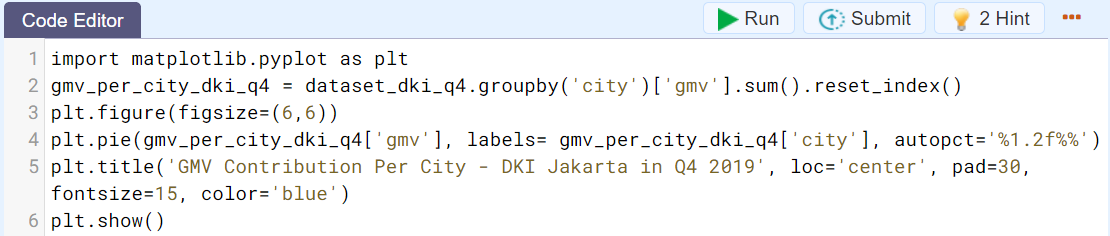
“Ini nanti dibuat pie chartnya juga ya, Aksara. Aku jelaskan sebentar caranya.  Kita perlu membuat dulu dataframe agregat,.” Aku selalu suka mendengar penjelasan Senja karena ia selalu berusaha memaparkan langkah per langkah. Berikut cara yang dipaparkan Senja padaku:

Misalnya  **gmv\_per\_city\_dki\_q4,**  masukkan datanya ke function **plt.pie()**.

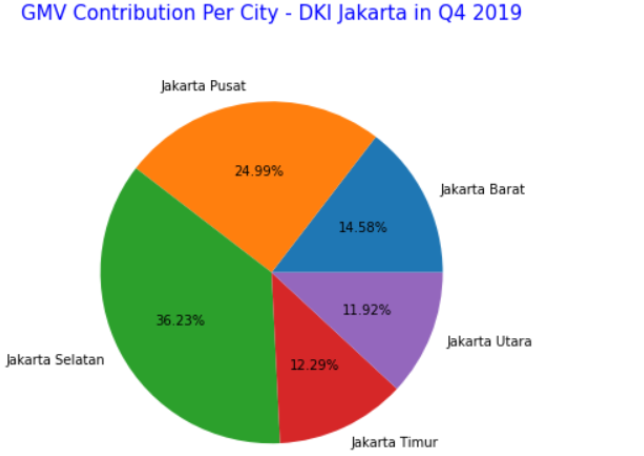
Beberapa parameter yang bisa dimodifikasi:

* **labels**: array yang berisikan label/tulisan yang ditunjukkan untuk masing-masing bagian pie.
* **colors**: array yang berisikan warna untuk masing-masing bagian pie.
* **autopct**: format untuk nilai persentasi yang ditampilkan, bisa berupa string atau function.
* **shadow**: jika diisi True, maka ada bayangan untuk pie chart-nya. Defaultnya adalah False.
* **radius**: jari-jari dari pie-chart

Perhatikan kode berikut:



Jika telah selesai dituliskan ke code editor dengan benar dan dijalankan dengan , maka diperoleh tampilan grafik pie-chart seperti berikut:

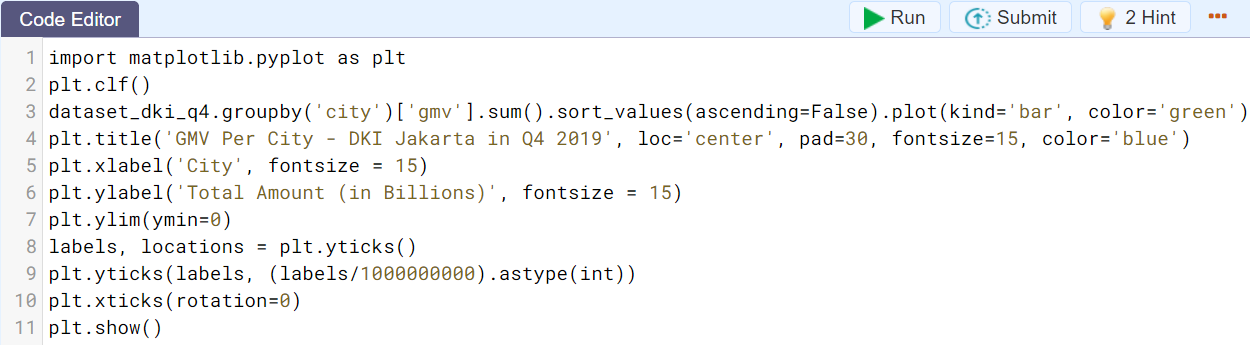


### Membuat Bar Chart

“Selain menggunakan pie chart, kita bisa pakai bar chart, tidak Nja?”

Kulihat matanya berbinar mendengar pertanyaanku. Senja mengangguk cepat dan menjawab, “Alternatif lain untuk membandingkan GMV tiap kota adalah dengan barchart. Saya lebih menyukai visualisasi ini karena lebih mudah terlihat perbandingan antar kota.”

Syntax dalam praktik membuat bar chart mirip dengan line chart, tentukan nilai untuk sumbu-x dan sumbu-y, lalu tambahkan parameter kind='bar' seperti berikut ini:



Setelah menuliskan kode di atas dengan benar dan dijalankan dengan  dapat dipeorleh grafik bar-chart seperti berikut ini:

  
Perhatikan di sini juga ditambahkan parameter rotation di plt.xtick() untuk memutar label di sumbu-x.

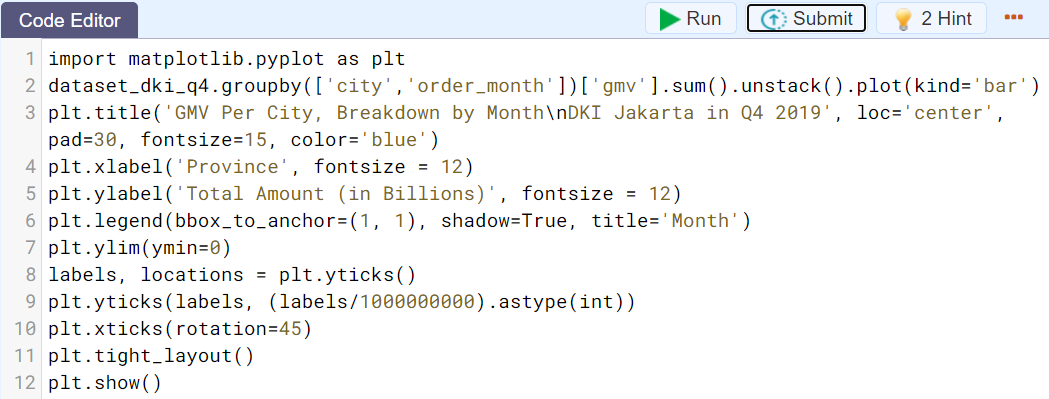
Sebagai catatan, jika ingin barchart-nya horizontal, maka bisa mengganti parameternya menjadi kind='barh', dengan tentu saja tidak lupa juga harus menyesuaikan sumbu x dan y nya.

### Membuat Multi-Bar Chart

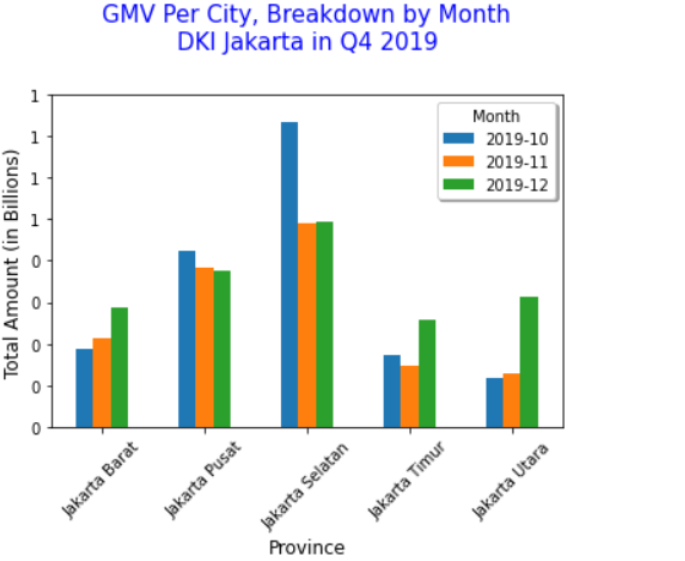
“Aksara, dari grafik ini, kelihatannya Jakarta Selatan dan Jakarta Pusat lumayan mendominasi GMV. Tapi apakah ini konsisten di setiap bulan? Bisa tolong tampilkan?”

“Bisa, Nja. Bentar yah, ini aku buatkan dulu multi-barchart untuk analisisnya,” ujarku.

Ini kode yang kutuliskan untuk membuat multi-barchart:



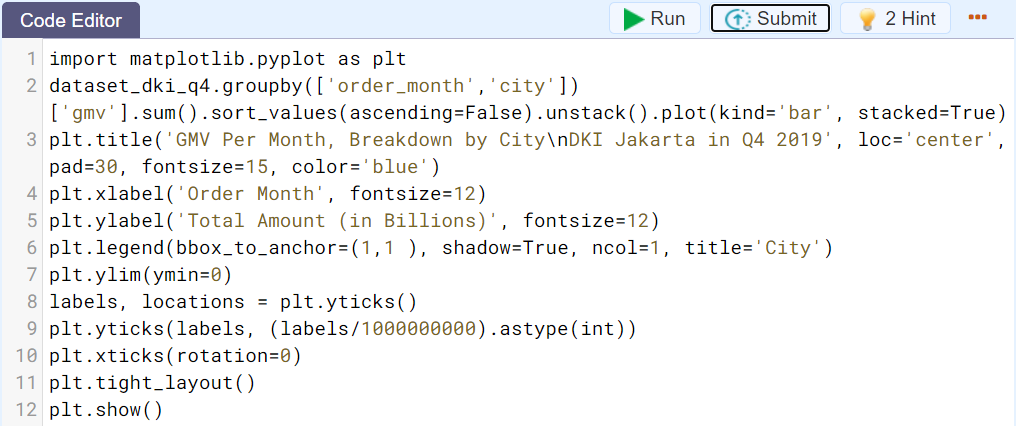
Setelah ditulis dengan benar dan dijalankan dengan  diperoleh tampilan multi-bar chart berikut ini



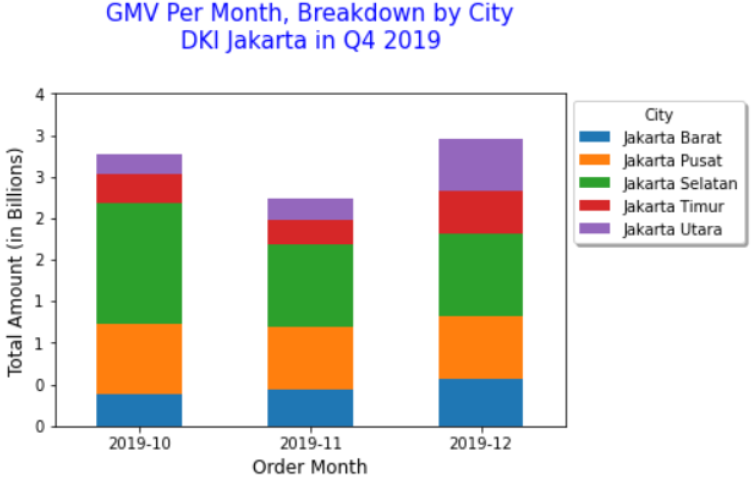
### Membuat Stacked Chart

“Ini udah lebih jelas. Sebagai pembanding saja, bagaimana kalau kita ubah multi-bar chart sebelumnya menjadi stacked barchart?”

Aku mengangguk sembari mengubah sedikit kodenya, seperti ini:



“Begini hasinya, Nja,” aku memutar laptopku ke Senja. Tentunya setelah menekan tombol .



"Dari kedua chart tersebut, kita sama-sama bisa melihat pola bahwa kebanyakan GMV turun dari bulan Oktober ke November, tetapi lalu naik pesat di bulan Desember. Untuk di Jakarta Barat, GMV November justru naik dari Oktober. Sedangkan di Jakarta Selatan, GMV Desember justru lebih kecil daripada GMV Oktober", jelasku.

"Hmm, padahal Jakarta Selatan adalah penyumbang GMV paling besar ya. Berarti kita harus menyarankan tim bisnis untuk genjot lagi penjualan di sana, agar trend-nya naik kembali, seperti kota-kota lainnya. Akan saya bawa kembali hasil temuan kamu ini ke Andra dan kepala cabang. Terima kasih, Aksara.”

## CHAPTER 4

### Membuat Agregat Data Customer

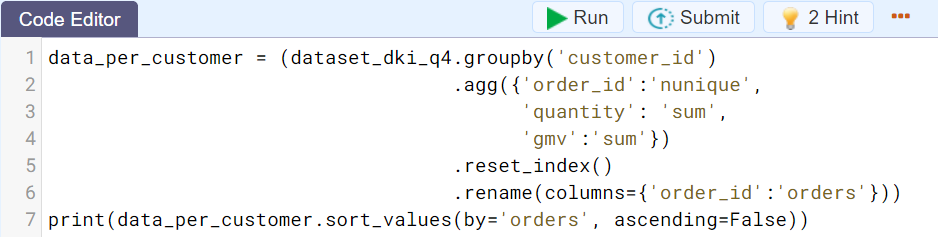
“Oke, data ini sudah bisa jadi dasar bagi kami untuk membuat kampanye yang meningkatkan transaksi. Kalau dilihat, ini di Jakarta Barat potensinya lagi baik ya. Tiap bulan growth-nya positif,” ujar seorang bapak dari video call di ponsel Senja.

Dari Senja, kuketahui kalau beliau adalah kepala cabang A yang menerima hasil dataku kemarin.

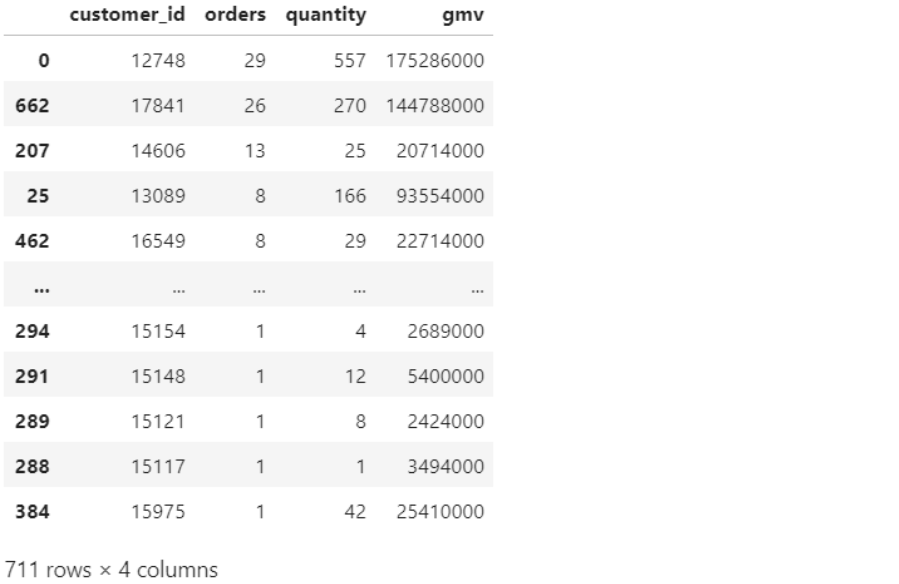
“Bagaimana dengan customer kita di DKI Jakarta? Apakah ada pola-pola menarik, terutama di Q4 2019?” tanya beliau kembali.

“Untuk itu, coba kita lihat dulu summary untuk tiap customer kita ya, Pak,” sahut Senja sembari memberi intruksi padaku.

Aku pun membuat dataframe baru bernama **data\_per\_customer\_dki\_q4.**



Jika dijalankan dengan  menghasilkan

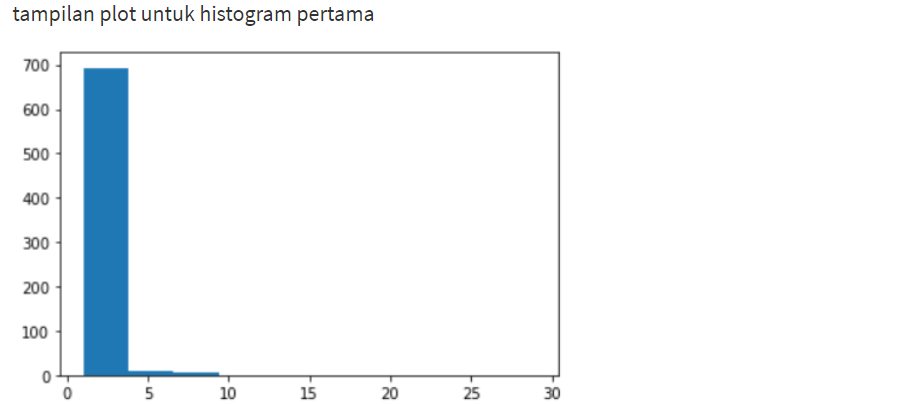


Di sini menggunakan function agg untuk melakukan agregasi data. Data agregat per customer yang diambil yaitu jumlah orders (banyaknya unique order\_id), total quantity, dan total GMV.

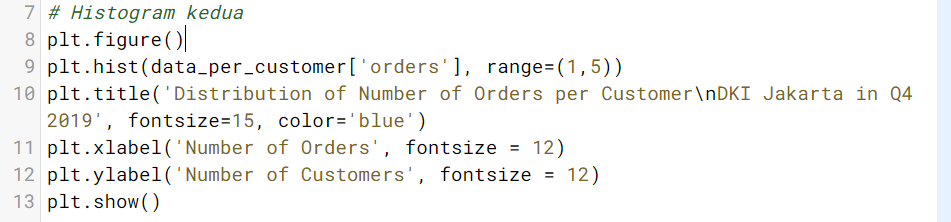
Lalu didapatkan 711 customers yang bertransaksi di DKI Jakarta pada Q4 2020.

Jika diurutkan dari jumlah order terbanyak, bisa dilihat bahwa customer\_id 12748 telah melakukan 29 transaksi dengan jumlah quantity mencapai 557, dan GMV lebih dari 175 Juta! Luar biasa ya. Customer-customer seperti ini harus tetap dijaga agar semakin loyal dengan produk.





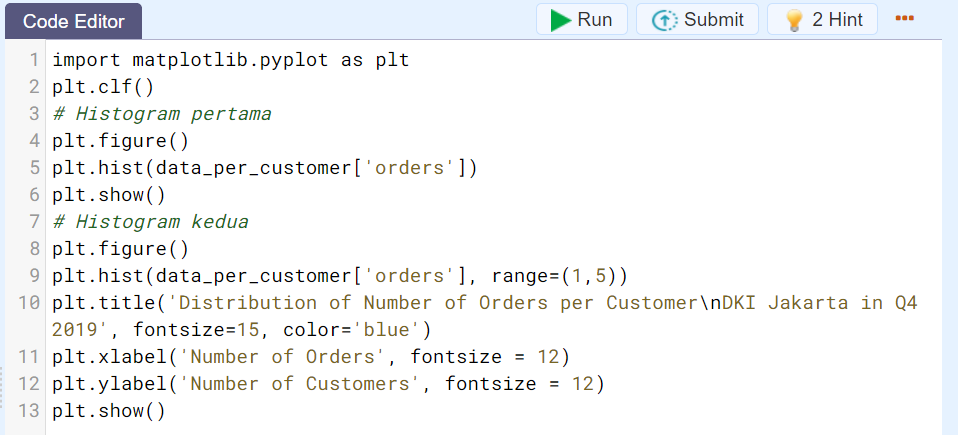
Terlihat bahwa datanya terlalu berat ke kiri, hampir semua customer hanya bertransaksi kurang dari 10, maka dari itu perlu set range-nya, misalnya hanya dari 1 sampai 5.



ampilan plot untuk histogram kedua



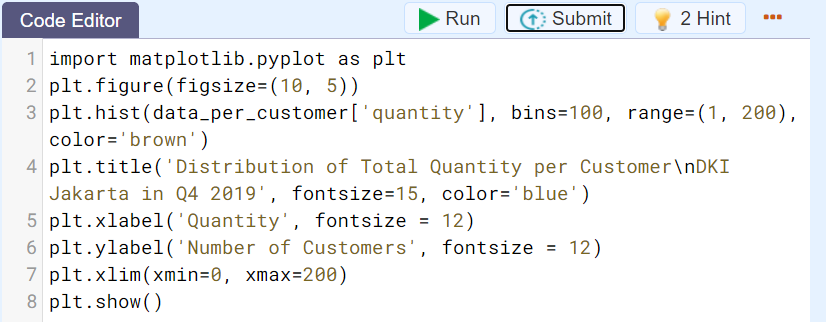
Full Code :



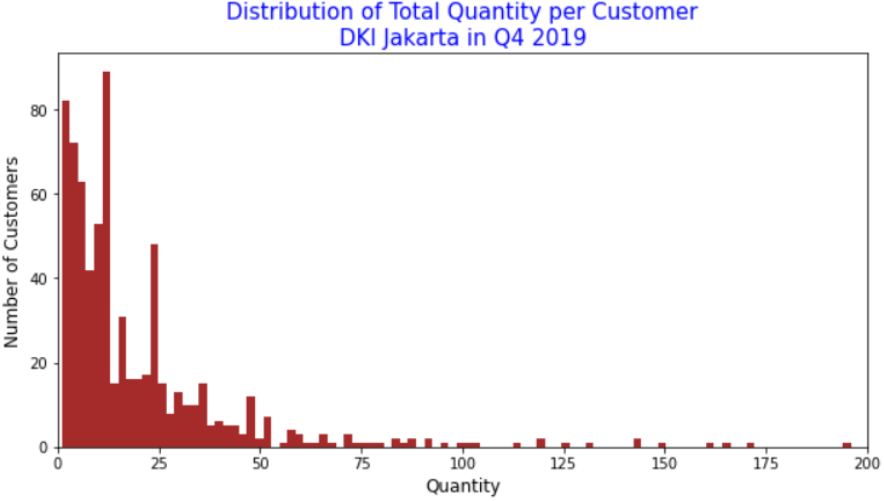
### Membuat Histogram - Part 2

Kalau seperti ini, aku bisa membuat histogram untuk quantity juga.

Caranya seperti berikut ini:



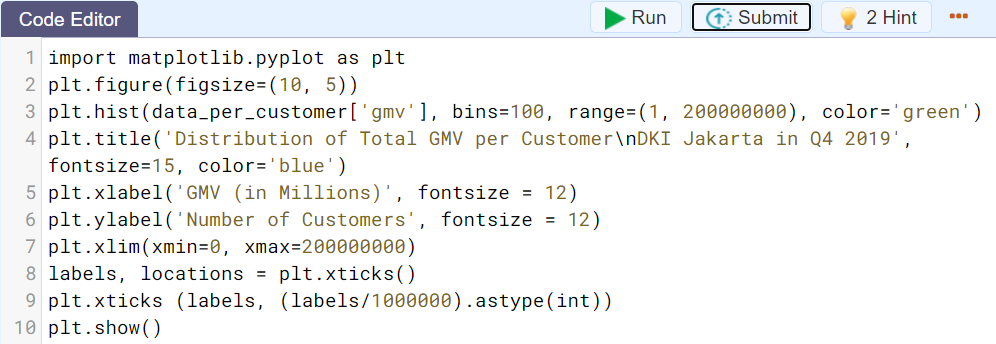
Jika dijalankan dengan  akan menapilkan histogram berupa



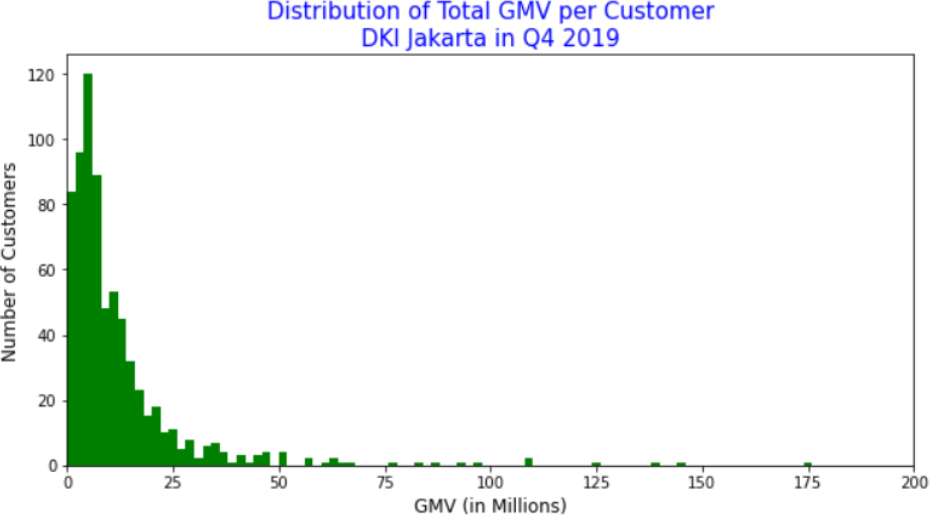
### Membuat Histogram - Part 3

“Aksara, kalau untuk GMV, agak berbeda. Kamu paham maksud saya?” ujar Senja mengingatkan.

Aku mengangguk. Aku harus menormalisasi ticks label di sumbu-x karena angka yang menunjukkan GMV bisa jadi terlalu panjang. Aku menunjukkanya pada Senja:



Jika dijalankan dengan  akan menghasilkan histogram



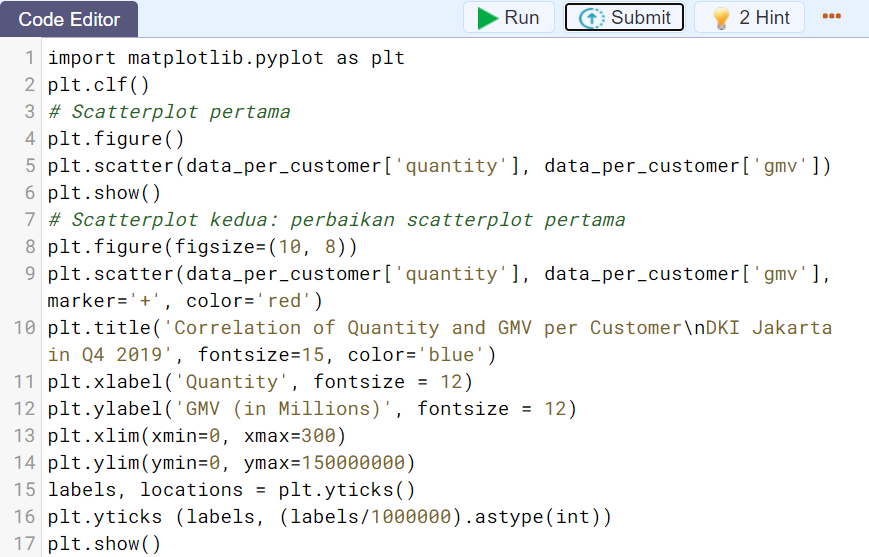
Ternyata ketiga variabel tersebut memiliki distribusi yang mirip, grafiknya condong ke kiri.

### Membuat Scatterplot

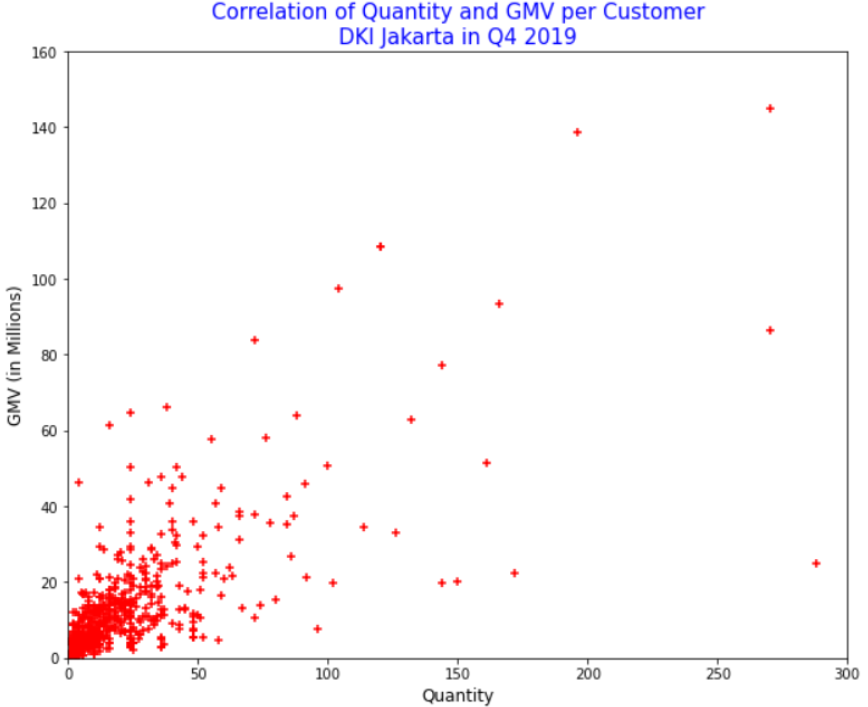
“Aksara, ini udah bagus penyajiannya. Tapi apakah variabel tersebut bisa saling berkorelasi?” tanya Senja menyadarkanku. Aku pun segera melihatnya dengan scatterplot.

Untuk membuat scatterplot, aku perlu menggunakan function plt.scatter dengan memasukkan variabel-variabel yang akan dibuat scatterplot.

Contoh untuk melihat scatterplot quantity vs GMV:



dengan output scatter plot:



## CHAPTER 5

### Pendahuluan

Aku telah selesai belajar bersama Senja mempelajari bagaimana membuat berbagai jenis visualisasi dengan matplotlib, diantaranya:

1. membuat multi line-chart yang dapat digunakan untuk melihat komparasi dari GMV dari berbagai provinsi, dari bulan ke bulan.
2. menghasilkan pie chart atau multi bar-chart untuk melihat komposisi GMV per kota.
3. untuk melihat distribusi GMV atau quantity, kita bisa gunakan histogram.
4. membuat scatterplot yang bisa memperlihatkan relasi antara quantity dan GMV.

Oleh karena itu, saat aku ingin membuat suatu visualisasi dari data yang dimiliki, aku bisa memilih mana jenis yang paling tepat, serta sudah bisa membuatnya dengan matplotlib.

"Sekarang ini untuk menguji pemahaman dan keterampilan kamu. Saya akan memberikanmu sebuah mini project yang harus kamu selesaikan dengan baik."

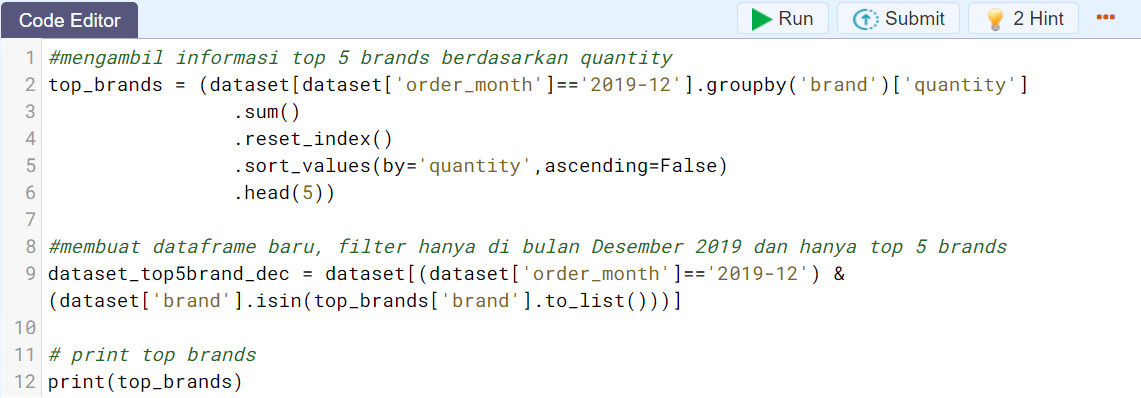
"Oke Nja!"

### Case 1: Menentukan brand top 5

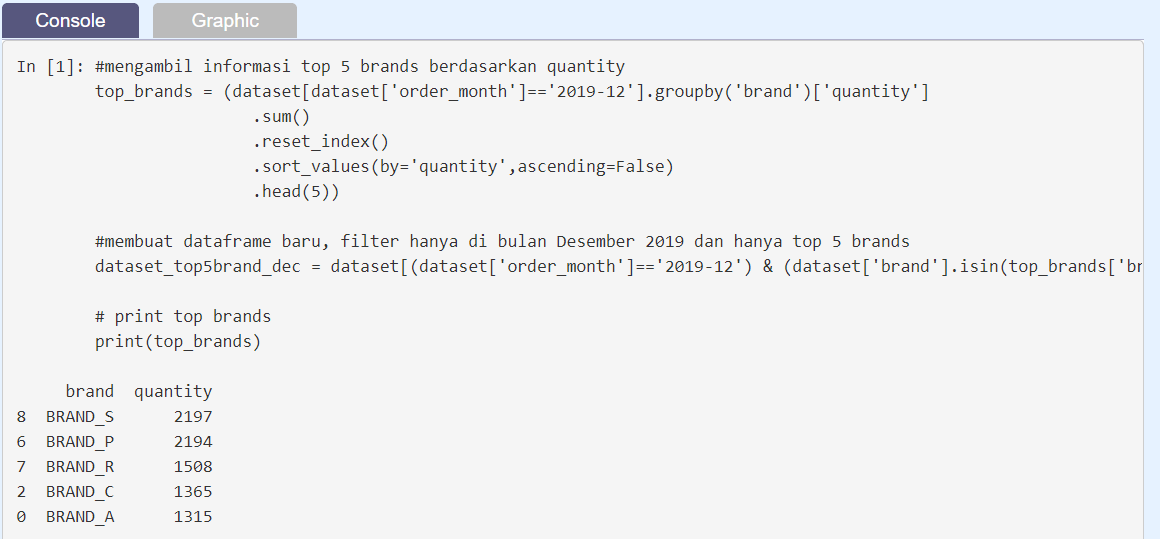
Lakukan analisis untuk top brands di bulan Desember 2019, di bawah ini adalah beberapa hal yang perlu dilakukan. Silakan ditambahkan dengan visualisasi-visualisasi lain yang kira-kira dapat memberikan insight tambahan.

**Case 1**:

Buat variabel baru (e.g. dataset\_top5brand\_dec) yang berisi data penjualan bulan Desember 2019, hanya untuk top 5 brand dengan quantity terjual terbanyak selama bulan Desember 2019. Sebutkan top 5 brands tersebut! Untuk visualisasi-visualisasi selanjutnya, hanya gunakan data frame ini.



Hasil :

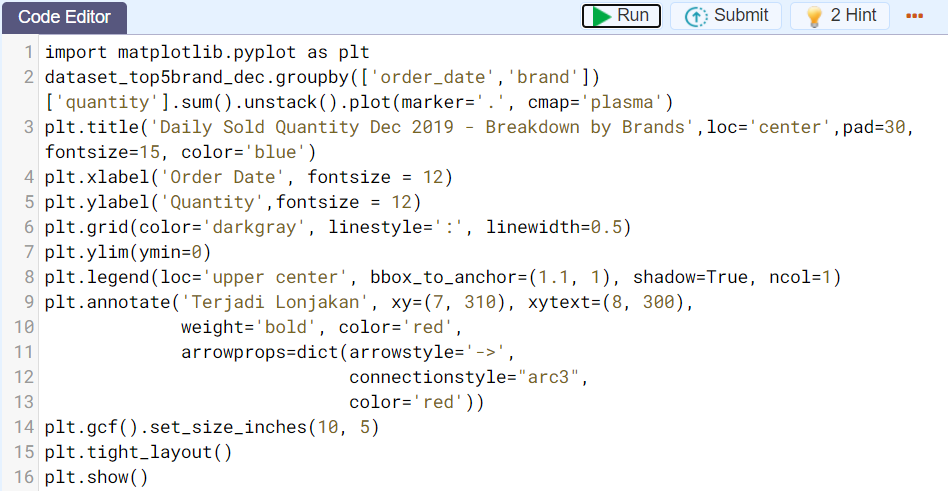


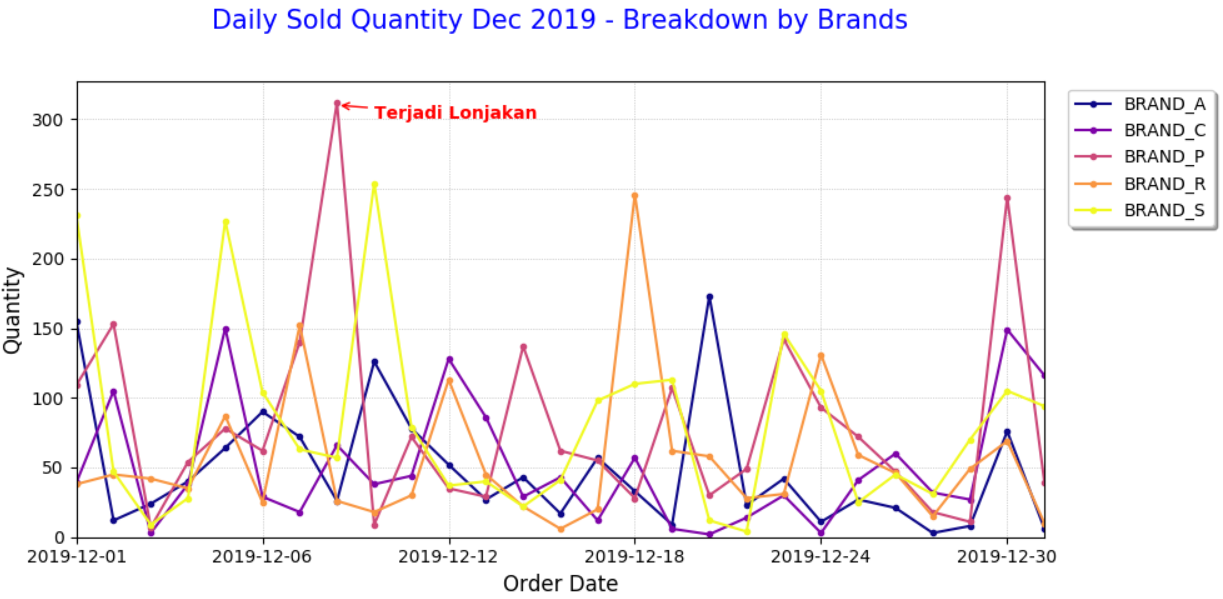
### Case 2: Multi-line chart daily quantity untuk brand top 5

Lakukan analisis untuk top brands di bulan Desember 2019, di bawah ini adalah beberapa hal yang perlu dilakukan. Silakan ditambahkan dengan visualisasi-visualisasi lain yang kira-kira dapat memberikan insight tambahan.

**Case 2**:

Buat visualisasi multi-line chart untuk daily quantity terjualnya, breakdown per brand. Maka, akan terlihat 1 tanggal di mana ada salah satu brand yang mengalami lonjakan (quantity lebih tinggi dari tanggal-tanggal lain). Beri anotasi untuk titik lonjakan tersebut.



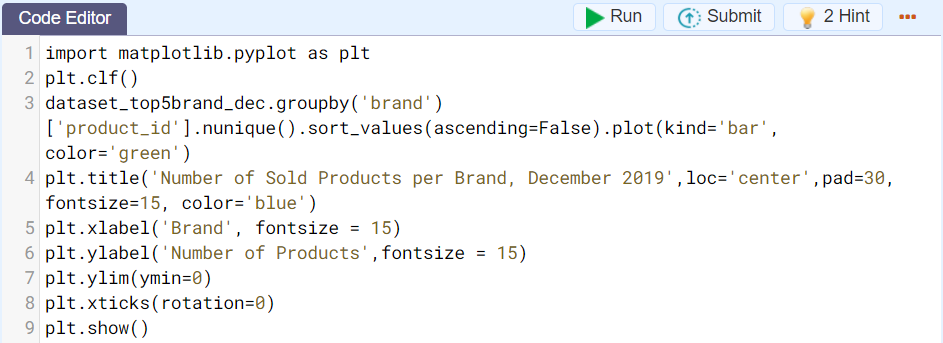


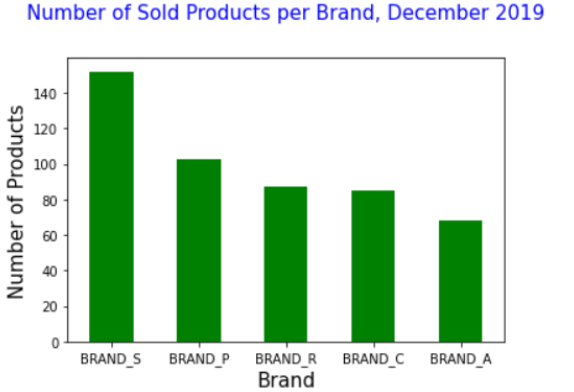
### Case 3: Kuantitas penjualan brand top 5 selama Desember 2019

Lakukan analisis untuk top brands di bulan Desember 2019, di bawah ini adalah beberapa hal yang perlu dilakukan. Silakan ditambahkan dengan visualisasi-visualisasi lain yang kira-kira dapat memberikan insight tambahan.

**Case 3**:

Cari tahu jumlah product untuk masing-masing brand yang laku selama bulan Desember 2019. Gunakan barchart untuk visualisasinya, urutkan dengan yang kiri adalah brand dengan product lebih banyak.





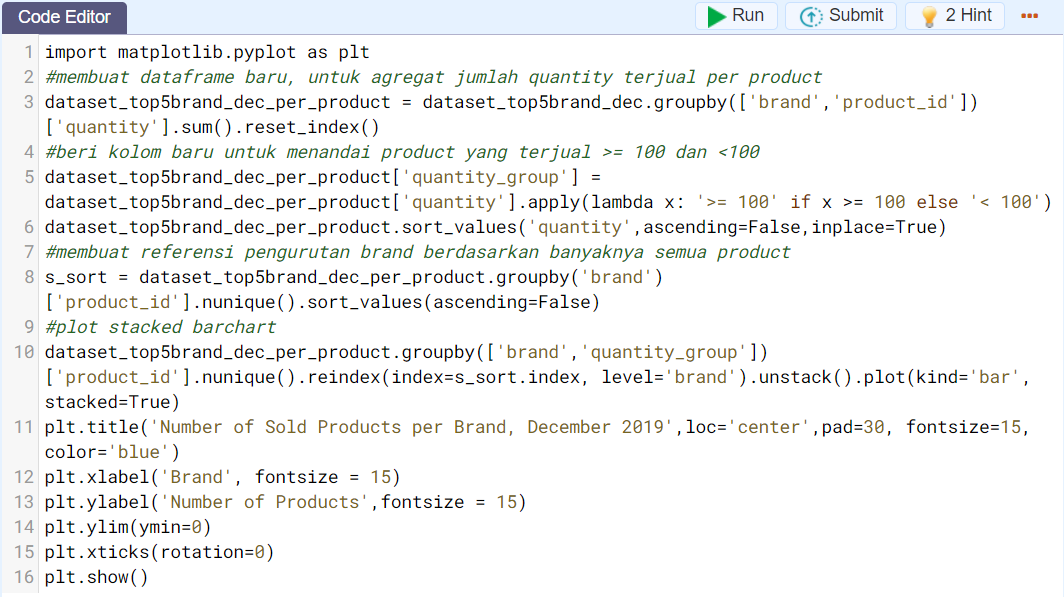
### Case 4: Penjulan produk diatas 100 dan dibawah 100 selama Desember 2019

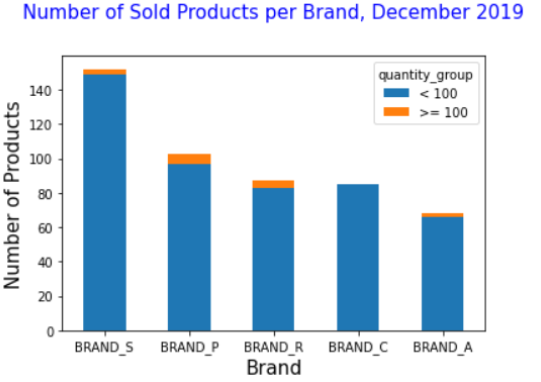
Lakukan analisis untuk top brands di bulan Desember 2019, di bawah ini adalah beberapa hal yang perlu dilakukan. Silakan ditambahkan dengan visualisasi-visualisasi lain yang kira-kira dapat memberikan insight tambahan.

**Case 4**:

Gunakan stacked chart, untuk breakdown barchart yang di Case 3, antara product yang terjual >= 100 dan < 100 di bulan Desember 2019. Apakah ada pola yang menarik?

Case 3: Cari tahu jumlah product untuk masing-masing brand yang laku selama bulan Desember 2019. Gunakan barchart untuk visualisasinya, urutkan dengan yang kiri adalah brand dengan product lebih banyak.



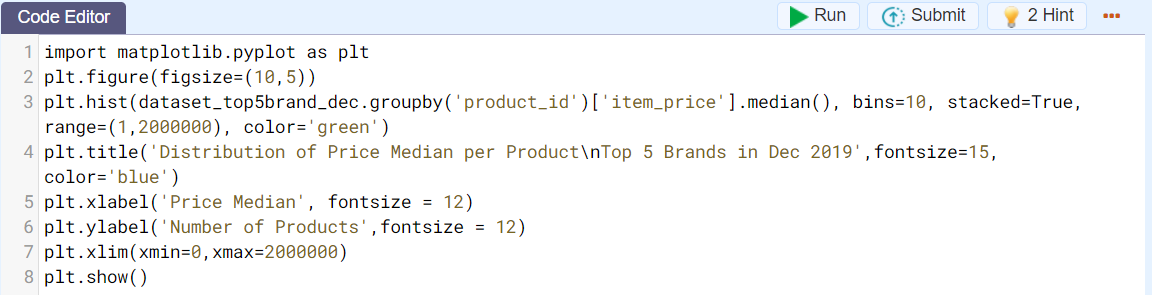


### Case 5: Murah atau mahalkah harga produk brand top 5

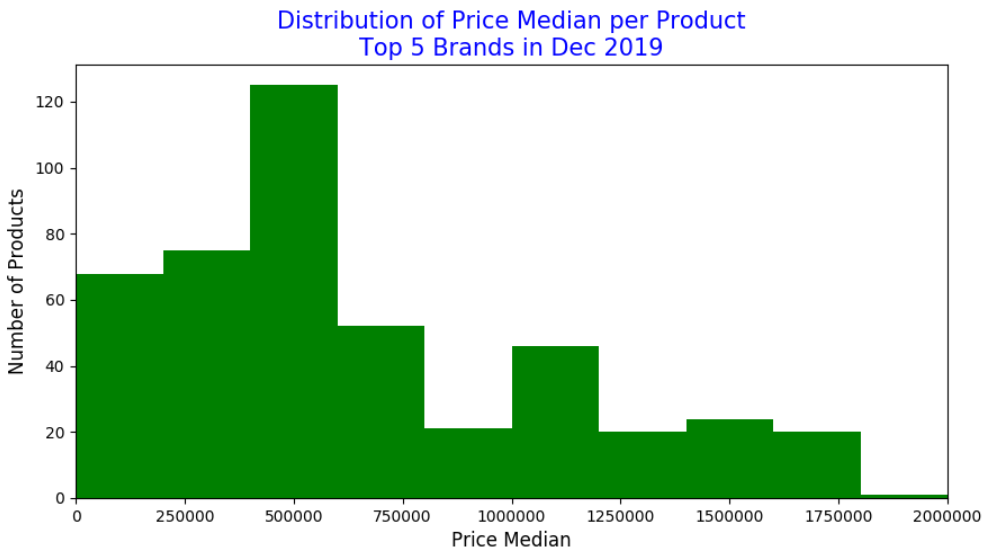
Lakukan analisis untuk top brands di bulan Desember 2019, di bawah ini adalah beberapa hal yang perlu dilakukan. Silakan ditambahkan dengan visualisasi-visualisasi lain yang kira-kira dapat memberikan insight tambahan.

**Case 5**:

Gunakan histogram untuk melihat distribusi harga product-product yang ada di top 5 brand tersebut (untuk tiap product\_id, ambil median harganya). Bagaimana persebaran harga product nya? Cenderung banyak yang murah atau yang mahal?



Hasil :

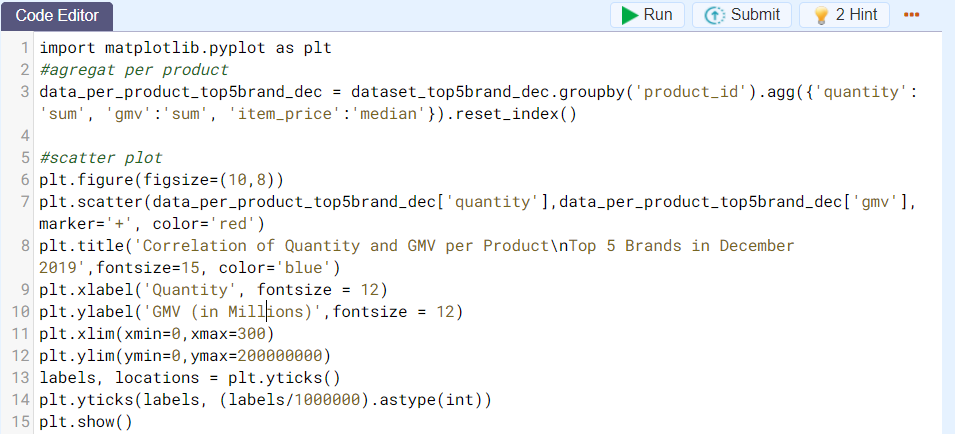


### Case 6a: Korelasi quantity vs GMV

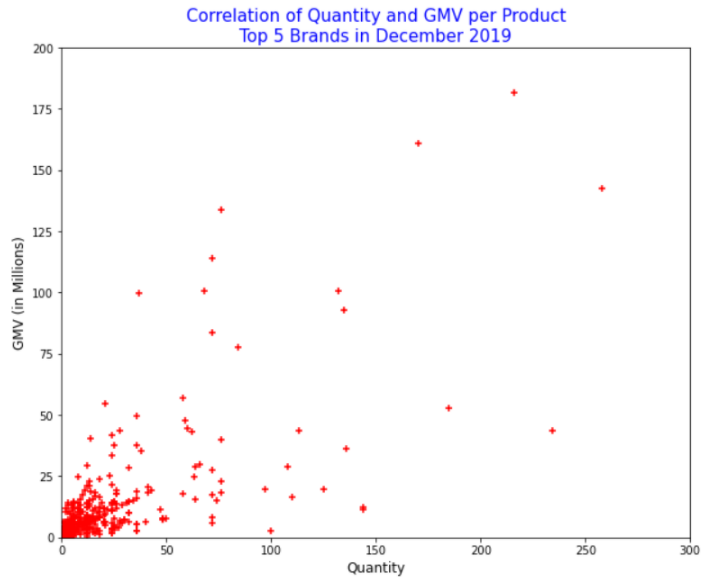
Lakukan analisis untuk top brands di bulan Desember 2019, di bawah ini adalah beberapa hal yang perlu dilakukan. Silakan ditambahkan dengan visualisasi-visualisasi lain yang kira-kira dapat memberikan insight tambahan.

**Case 6a**:

Untuk setiap product\_id, cek scatterplot antara  quantity dan GMV, apakah ada korelasi? Bagaimana dengan median harga vs quantity? Apakah product yang murah cenderung dibeli lebih banyak?



Hasil :

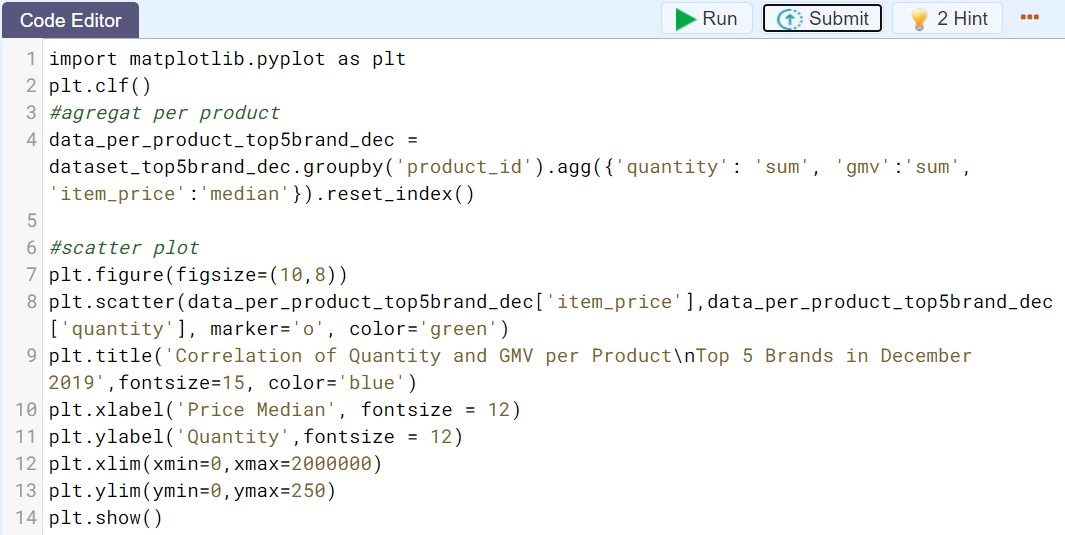


### Case 6b: Korelasi median harga vs quantity

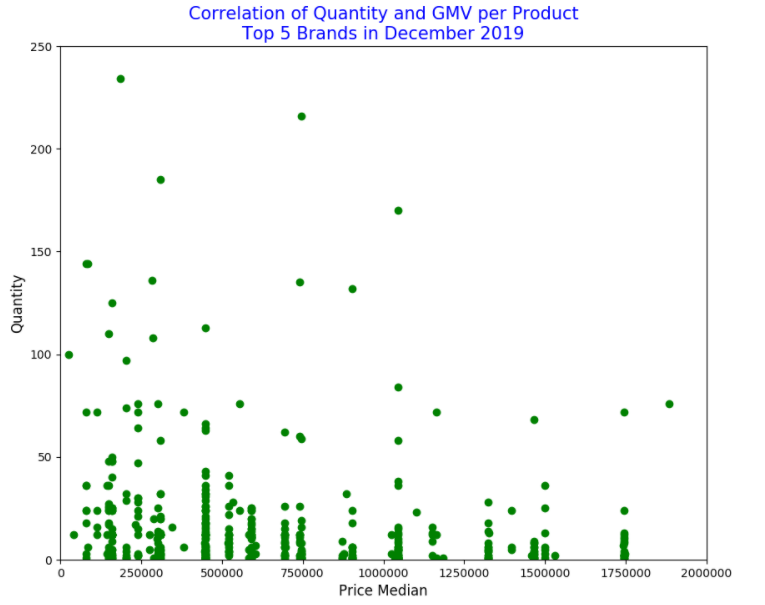
Lakukan analisis untuk top brands di bulan Desember 2019, di bawah ini adalah beberapa hal yang perlu dilakukan. Silakan ditambahkan dengan visualisasi-visualisasi lain yang kira-kira dapat memberikan insight tambahan.

**Case 6b**:

Untuk setiap product\_id, cek scatterplot antara  quantity dan GMV sudah kamu lakukan pada **Case 6a**? Untuk Case 6b ini bagaimanakah dengan median harga vs quantity? Apakah product yang murah cenderung dibeli lebih banyak?



Hasil :



### Penutup/Kesimpulan

**Congratulations!** Akhirnya berhasil lagi aku menyelesaikan modul **Data Visualization with Python Matplotlib for Beginner - Part 2**. Berdasarkan materi-materi yang telah aku pelajari dan praktikkan dalam modul ini, berikut summary materi yang telah kudapatkan pengetahuan (knowledge) dan praktek (skill), diantaranya :

* membuat multi line-chart
* membuat pie chart atau multi bar-chart
* membuat grafik distribusi suatu data dengan menggunakan histogram.
* membuat scatterplot yang dikustomisasi secara langsung untuk menampilkan hasil visualisasi yang menarik dan memberikan insight kepada pembaca. Serta,
* mengerjakan mini project untuk memvisualisasikan data produk retail berdasarkan pengetahuan (knowledge) dan praktek (skill) yangtelah diperoleh pada bab-bab sebelumnya.

**Keep fighting!**