به تصویر کشیدن داده‌ها به همراه رسم نمودار در زبان برنامه نویسی پایتون

مصطفی کریمی

فروردین ماه ۱۴۰۱

به نام زیبایی

فهرست مطالب

به تصویر کشیدن داده‌ها

تا به حال شده رییستان از شما بخواهد میزان فروش این ماه (و البته ماه‌های قبل) شرکت را برایش آماده کرده و شرح دهید؟!

در این صورت شما دو راه دارید، یا یک **جدولِ بزرگِ پر از اعداد**، شامل درآمد در ماه‌های مختلف را تقدیم حضور رییستان کنید، که نه به درد ایشان می‌خورد و نه به درد شما، و یا اینکه، پس از بررسی دقیق آن جدول، مفهومی که قرار است از طریق آن جدول منتقل شود را در قالب چند **شکل و نمودارِ** ساده اما جذاب، به ایشان ارایه دهید.

حتما شنیده‌اید که: «**گاهی یک تصویر، گویا تر از هزاران واژه است.**»

واقعا هم همینطور است، ساختار ذهنی انسان طوری شکل گرفته که یک تصویر ساده را بسیار سریع‌تر و راحت‌تر از هر چیز دیگری درک می‌کند. یک مدیر ارشد اجرایی (CEO) وظایف سنگینی دارد که زمانی را برای سر و کله زدن با داده‌ها و جدول‌های عددی و یافتن مفهوم درون آن‌ها باقی نمی‌گذارند. بنابراین، این شخص برای انجام وظایف خود و گرفتن تصمیم‌های سازمانی درست، نیاز به درک سریع و دقیقِ مفهومِ نهفته در این جدول‌های عددی دارد. از همین رو امروزه مدیران شرکت‌های بزرگ، از دانشمندان داده‌ها (Data Scientistها) و مهندسان هوشمندی در کسب و کار (BI Engineerها) برای تحلیل داده‌ها، به تصویر کشیدنِ آن‌ها و تصمیم گیری بر اساس داده‌ها (Data-driven strategy) بهره می‌برند.

شکل ۱-۱ یک Data Scientist داده های خامِ سازمان را پس از پالایش و تمیز کردن، تحلیل کرده و به نمودار های آماری قابل فهم تبدیل می‌کند.

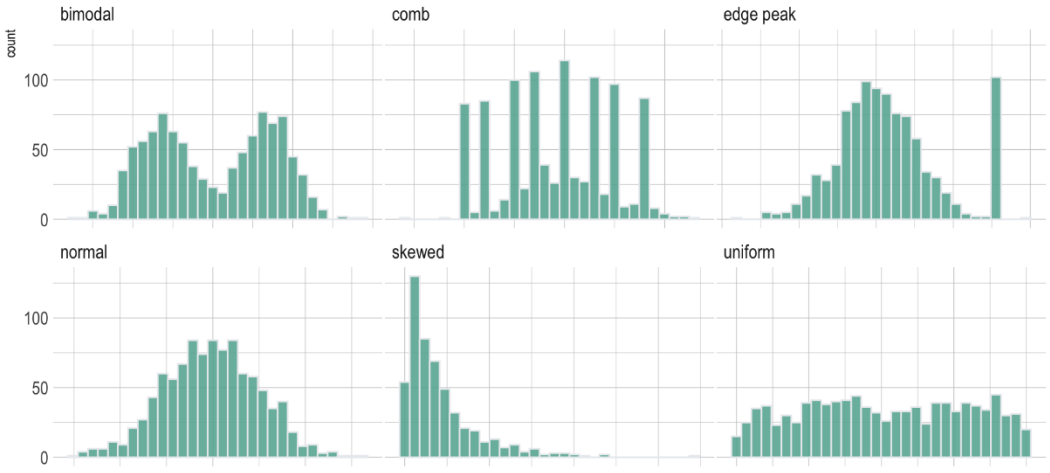
به تصویر کشیدن داده‌ها (Data Visualization) یکی از اصلی‌ترین و مهم‌ترین موضوعات در Data Science است که با نمودارهای آماری، گراف‌ها، پلات‌ها و اشکال سر و کار دارد. با این وجود، هدف از این پُست، مطالعه‌ی کاملِ متدهای Data Visualization نیست بلکه **آشنایی با مهم‌ترین نمودارهای آماریِ** مورد استفاده در Data Science است. ازین رو مطالعه این پست را به تمامی دانشجویان، پژوهشگران، مهندسان، اهالی کسب و کار و مدیران (فارغ از رشته تحصیلی و کاری) پیشنهاد می‌دهیم.

در اینجا فرض می‌شود که شما یک مجموعه داده‌ها (Dataset) شامل چندین سطر (Observation یا Experience یا Data Point) و چندین ستون (Feature یا Variable یا Attribute) دارید و می‌خواهید

با رسم نمودارهای آماری مرتبط (که در زیر معرفی خواهند شد)، به یک بینش در رابطه با مجموعه داده‌ها دست پیدا کنید.

۱ - ۱ - نمودار Histogram

از این نمودار برای بررسی و مطالعه‌ی **توزیعِ آماریِ یک متغیر عددی** (Numerical Variable) استفاده می‌شود. معمولا اولین کاری که در مواجهه با یک Dataset می‌کنیم، بررسی توزیع آماریِ تک‌تکِ متغیرها (Feature ها) ی آن است. درکِ توزیعِ آماریِ متغیرها به ما کمک می‌کند تا اشتباهات موجود در داده‌ها (مانند نویزها و خطاهای اندازه گیری) را کشف کنیم. برای رسم نمودار Histogram، بازه (Range) مقادیرِ مشاهده شده برای متغیر را به تعدادی فاصله‌ی هم اندازه (bin) تقسیم کرده و فراوانی (تعداد) مشاهدات در هر فاصله را به صورتِ ارتفاعِ یک میله نمایش می‌دهیم.



شکل ۱-۲ شش توزیع آماری مرسوم در تحلیل متغیرها

ممکن است هیستوگرامِ مربوط به چند متغیر، برای مقایسه، در یک نمودار و با رنگ‌های مختلف نشان داده شوند، اما معمولا برای مطالعه‌ی همزمانِ توزیعِ آماریِ چند متغیر، از نمودارهای دیگر (مثل نمودار Boxplot، نمودار Violin Plot و یا نمودار Ridgeline Plot) استفاده می‌شود.

برای کسب اطلاعات بیشتر در رابطه با Histogram از لینک زیر استفاده کنید:

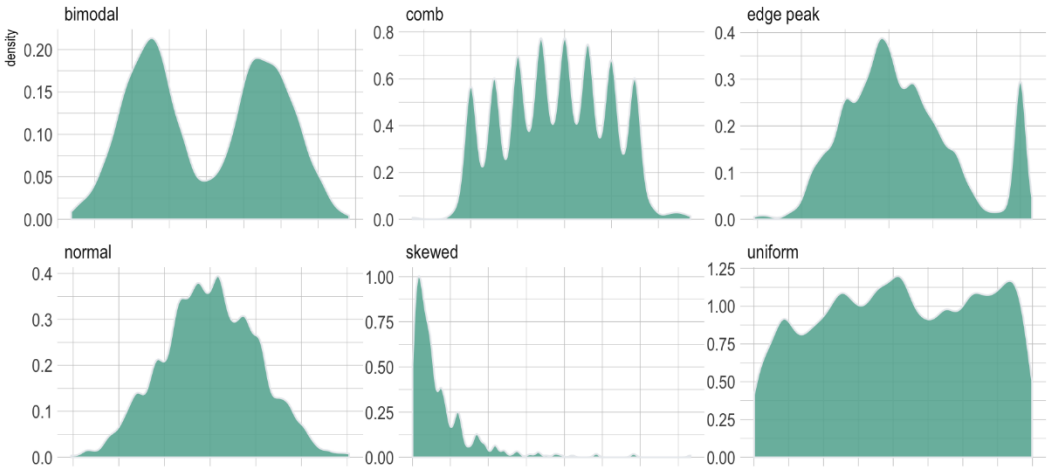
https://www.data-to-viz.com/graph/histogram.html

برای آشنایی با روشِ رسم Histogram در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/histogram/

۱ - ۲ - نمودار Density Plot

این نمودار در واقع شکل صیقل داده شده‌ی نمودار Histogram است و تابعی به نام Probability Density Function را برای یک متغیرِ عددی به تصویر می‌کشد.



شکل ۱- ۳ شش توزیع آماری مرسوم در تحلیل متغیرها

برای کسب اطلاعات بیشتر در رابطه با Density Plot از لینک زیر استفاده کنید:

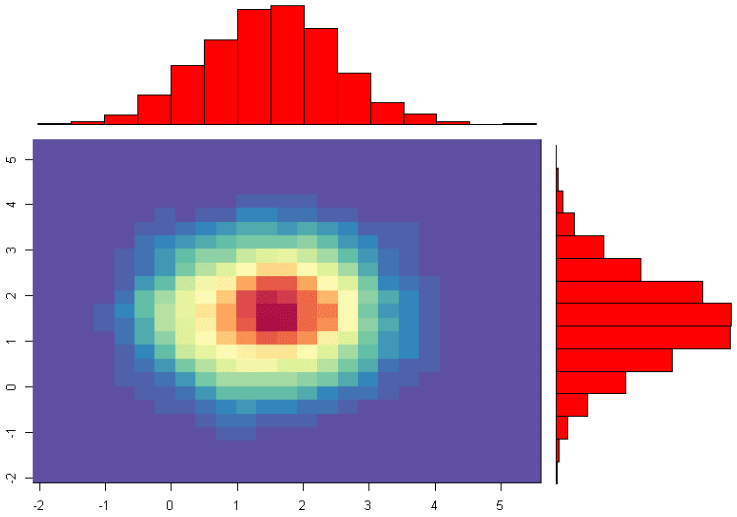
https://www.data-to-viz.com/graph/density.html

برای آشنایی با روشِ رسم Density Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/density-plot/

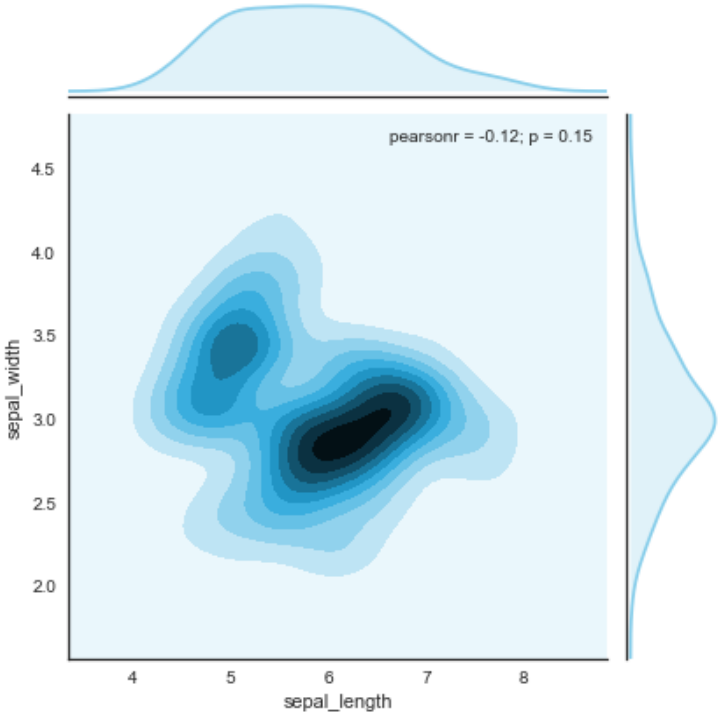
۱ - ۳ - نمودارهای 2D Histogram و 2D Density Plot

این دو نمودار برای مطالعه‌ی **توزیعِ آماریِ دو متغیرِ کمّی (عددی) در ترکیب با هم**، استفاده می‌شوند. برای رسم نمودار 2D Histogram، یکی از متغیر ها را روی محور X و دیگری را روی محور Y در نظر گرفته و با استفاده از bin ها صفحه مختصات دوبعدی را به یک جدول (Grid) تبدیل می‌کنیم. سپس تعداد مشاهدات در هر قسمت از Grid را توسط طیف رنگ ها نمایش می‌دهیم (مثلا برای مقادیر کم از بنفش و برای مقادیر زیاد از قرمز استفاده می‌کنیم)



شکل ۱- ٤ نمودار 2D Histogram

نمودار 2D Density Plot هم صیقل داده شده‌ی 2D Histogram است.



شکل ۱ -۵ نمودار 2D Density Plotشامل خطوط کانتور

این دو نمودار، زمانی که تعداد مشاهدات زیاد است کارایی دارند و برای تعداد مشاهدات کم، نمودارهای دیگر (مثل Scatter Plot) اطلاعات دقیق‌تری را منتقل می‌کنند.

برای کسب اطلاعات بیشتر در رابطه با 2D Histogram و 2D Density Plot از لینک زیر استفاده کنید:

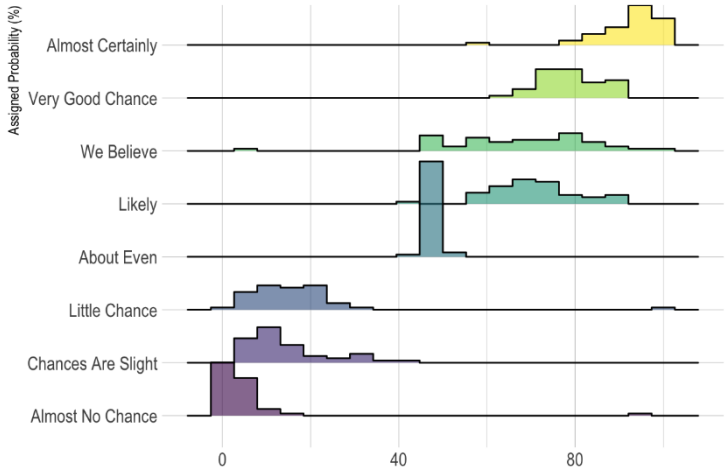
https://www.data-to-viz.com/graph/density2d.html

برای آشنایی با روشِ رسمِ 2D Histogram و 2D Density Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

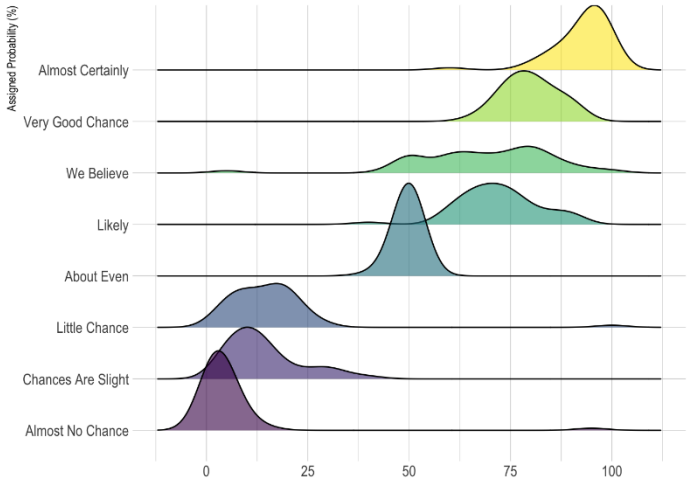
https://www.python-graph-gallery.com/2d-density-plot/

۱ - ۴ - نمودار Ridgeline Plot یا Joyplot

این نمودار شامل چند Histogram یا 2D Density Plot با مقیاس یکسان است که معمولا برای **مقایسه‌ی توزیعِ آماریِ یک متغیرِ عددی برای چندین گروهِ مختلف** استفاده می‌شود. Ridgeline Plot معمولاً وقتی کارایی خود را نشان می‌دهد که الگوی مشخصی بین مقادیرِ گروه‌ها وجود داشته باشد. به عنوان مثال در شکل میبینید که این مقادیر از پایین به بالا زیاد شده اند و نمودارها روی هم نیفتاده‌اند.



شکل ۱- ٦ نمودار Ridgeline Plot با استفاده ازHistogram



شکل ۱- ۷ نمودار Ridgeline Plot با استفاده از Density Plot

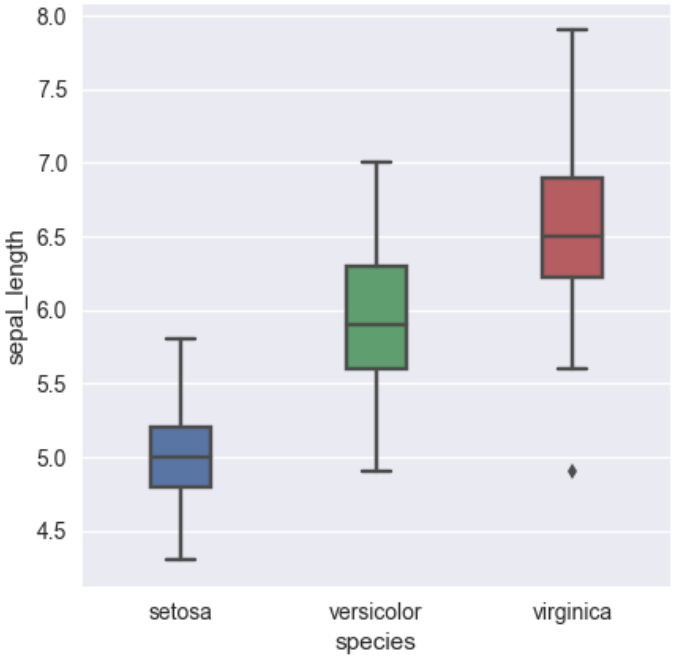
در این نوع نمودار، ترتیب درست گروه‌ها، تاثیر بسزایی در خوانایی نمودار دارد.

برای کسب اطلاعات بیشتر در رابطه با Ridgeline Plot از لینک زیر استفاده کنید:

https://www.data-to-viz.com/graph/ridgeline.html

۱ - ۵ - نمودار Box Plot

این نمودار، **خلاصه‌ای از اطلاعاتِ آماریِ مربوط یک متغیر کمّی، برای چندین گروه مختلف** را به تصویر می‌کشد. به عبارت دیگر، خطی که box ها را به دو نیم تقسیم می‌کند، نشان دهنده‌ی میانه (Median) داده‌های مشاهده شده برای متغیر، و ابتدا و انتهای box ها نشان دهنده‌ی چارک اول (Lower Quartile) و چارک سوم (Upper Quartile) آنها اند. خطوط انتهایی نیز نمایانگر کوچکترین و بزرگترین داده‌ی مشاهده شده (صرف نظر از داده های پَرت یا Outlier[1] ها) اند. Outlier ها (به عنوان نمونه، نویز ها یا داده‌های ناشی از خطای اندازه گیری) نیز با نقطه در نمودار زیر نشان داده شده‌اند.

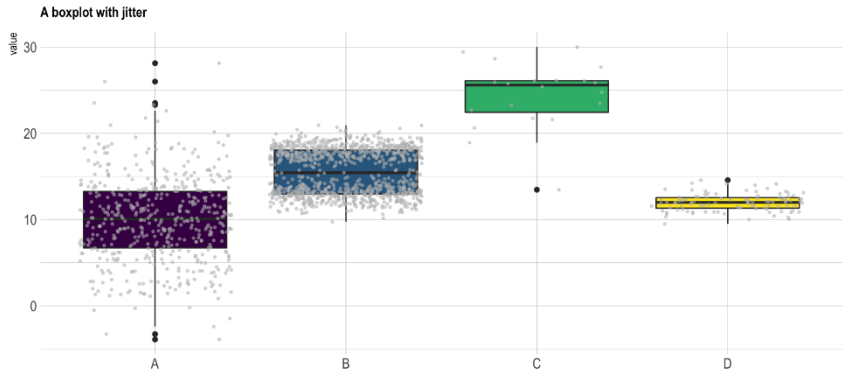


شکل ۱ -۸ نمودار Box Plot برای متغیر sepal_length در دیتاست IRIS

گاهی می‌خواهیم توزیع آماری داده‌های مشاهده شده را نیز در Box Plot نمایش دهیم. اگر تعداد مشاهدات کم است، می‌توانیم آن مشاهدات را با نقطه روی نمودار نشان دهیم تا نحوه توزیع آنها مشخص

[1] دادهٔ پرت یا داده دورافتاده (به انگلیسی: Outlier) در مبحث آمار، به داده‌ای گفته می‌شود که با دیگر داده‌های هم‌گروه فاصلهٔ چشمگیری داشته‌باشد، (یا به‌اصطلاح "نخواند") گرابز دادهٔ پرت را این‌چنین تعریف کرده‌است؛ « دادهٔ پرت داده‌ای است که تفاوت قابل ملاحظه‌ای با بقیهٔ اعضای نمونه‌ای که در آن اتفاق افتاده‌است داشته‌باشد.»

شود. اما اگر تعداد مشاهدات زیاد است، این روش کارایی ندارد و به جای آن از نمودارهای دیگر (مثل Violin Plot) استفاده می‌شود.



شکل ۱- ۹ نمایش مشاهدات در نمودار Box Plot جهت بررسی توزیع آماری داده ها

برای کسب اطلاعات بیشتر در رابطه با Box Plot از لینک زیر استفاده کنید:

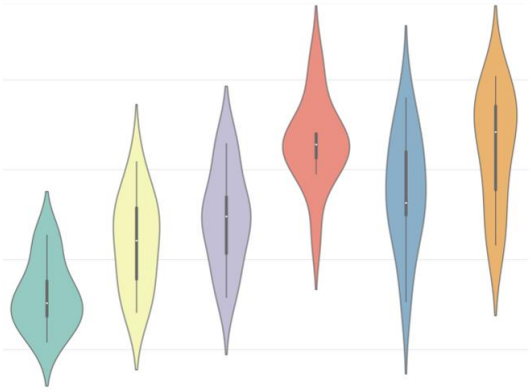
https://www.data-to-viz.com/caveat/boxplot.html

برای آشنایی با روشِ رسمِ Box Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کتید:

https://www.python-graph-gallery.com/boxplot/

۱ - ۶ - نمودار Violin Plot

این نمودار زمانی استفاده می‌شود که **می‌خواهیم توزیع آماری تعداد زیادی داده را در Box Plot داشته باشیم**، اما نمایش همه داده‌ها روی نمودار از خوانایی آن می‌کاهد. Violin Plot با ترکیب نمودارهای Box Plot و Ridgeline Plot این کار را به راحتی انجام می‌دهد.



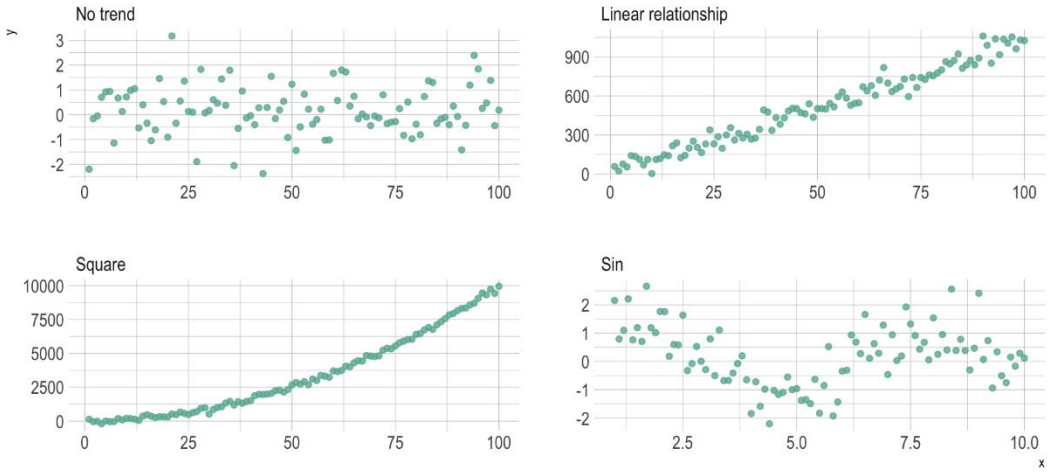
شکل ۱ -۱۰ نمودار Violin Plot از نمایش تابع چگالی احتمال کنار Box Plot تشکیل می‌شود.

برای آشنایی با روشِ رسمِ Violin Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/violin-plot/

۱ - ۷ - نمودار Scatter Plot

از این نمودار برای **نمایشِ رابطه و همبستگیِ میانِ دو متغیرِ کمّی** استفاده می‌شود. متغیرها را توسط محورهای مختصات دکارتی به تصویر کشیده و به ازای هر داده‌ی مشاهده شده در Datset یک نقطه از فضای مختصات را با دایره پُر می‌کنیم. اگر بین متغیرها، رابطه و همبستگی وجود داشته باشد، در نمودار Scatter Plot، قابل تشخیص است. به عنوان مثال، نمودار بالا سمت راست در تصویر زیر، نشان دهنده‌ی یک رابطه خطی بین دو متغیر است چون به نظر می‌رسد نقاط، روی یک خط پراکنده شده‌اند. رابطه خطی به این معنی است که اگر مقدار یکی از متغیرها n برابر شود، مقدار متغیر دیگر نیز n برابر می‌شود. یا نمودار پایین سمت راست، معرف یک رابطه سینوسی بین دو متغیر است.

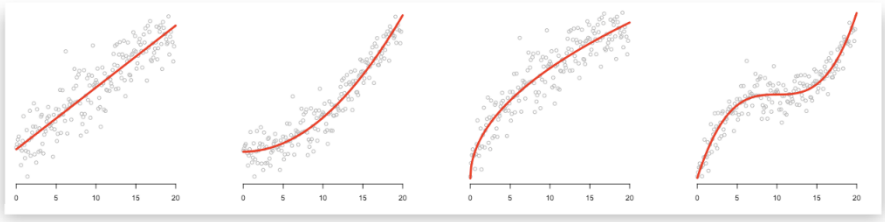


شکل ۱-۱۱ تعدادی از الگو های قابل تشخیص توسط Scatter Plot

بسیاری از اوقات در حل مسایل Data Science، بررسی همه متغیرها (Feature ها) امکان پذیر نیست و مایلیم متغیرهای مهم تر را انتخاب و بررسی کنیم. یافتن همبستگی بین متغیرها به ما کمک می‌کند متغیرهایی که قابل پیش‌بینی از روی سایر متغیرها هستند را برای سادگی بیشتر حذف کنیم. به این کار اصطلاحا Feature Selection[۲] گفته می‌شود.

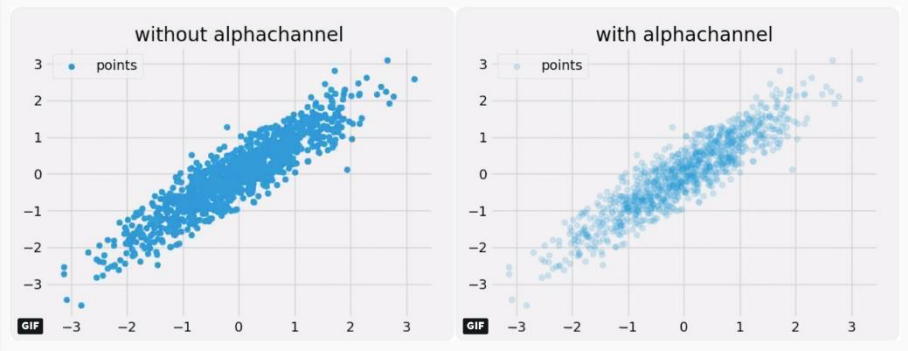
همبستگی بین دو متغیر در Scatter Plot را می‌توان توسط یک **منحنی فیت شده روی مشاهدات** به نام Trend Curve نشان داد.

[۲] در یادگیری ماشینی و آمار، انتخاب ویژگی، همچنین به عنوان انتخاب متغیر، انتخاب ویژگی یا انتخاب زیر مجموعه متغیر شناخته می‌شود، فرآیند انتخاب زیرمجموعه‌ای از ویژگی‌های مرتبط (متغیرها، پیش‌بینی‌کننده‌ها) برای استفاده در ساخت مدل است. تکنیک های انتخاب ویژگی را باید از استخراج ویژگی متمایز کرد. استخراج ویژگی، ویژگی های جدیدی را از توابع ویژگی های اصلی ایجاد می کند، در حالی که انتخاب ویژگی زیر مجموعه ای از ویژگی ها را برمی گرداند. تکنیک‌های انتخاب ویژگی اغلب در حوزه‌هایی استفاده می‌شوند که ویژگی‌های زیادی و نمونه‌های نسبتاً کمی (یا نقاط داده) وجود دارد. موارد کهن الگویی برای استفاده از انتخاب ویژگی شامل تجزیه و تحلیل متون نوشته شده و داده های ریزآرایه DNA است که در آن هزاران ویژگی و چند ده تا صدها نمونه وجود دارد.

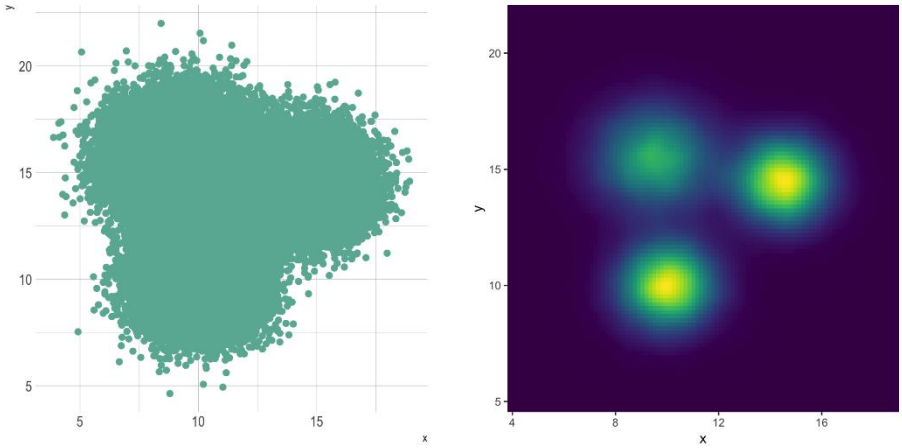


شکل ۱- ۱۲ یافتن بهترین منحنی قابل فیت شدن روی مشاهدات، Regression Analysis نام دارد.

وقتی تعداد مشاهدات زیاد نیست، نحوه ی توزیعِ آماری داده‌ها در Scatter Plot مشخص است. اما زمانی که تعداد مشاهدات زیاد باشد، پدیده‌ی Overplotting[3] روی نمودار رخ داده و خوانایی آن را از بین می‌برد. در این صورت برای مشخص شدن توزیع آماری، یا از نقطه‌های Transparent در Scatter Plot استفاده می‌کنیم یا از نمودار 2D Density Plot.



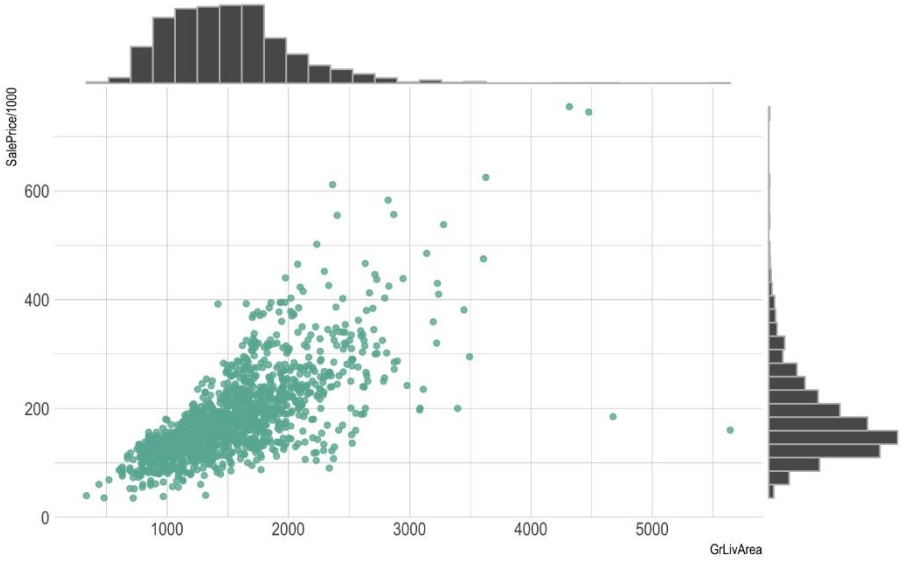
شکل ۱- ۱۳ استفاده از نقاط Transparent برای مشخص تر شدن توزیع آماری



شکل ۱- ۱٤ استفاده از 2D Density Plot به جای Overplot کردن روی Scatter Plot

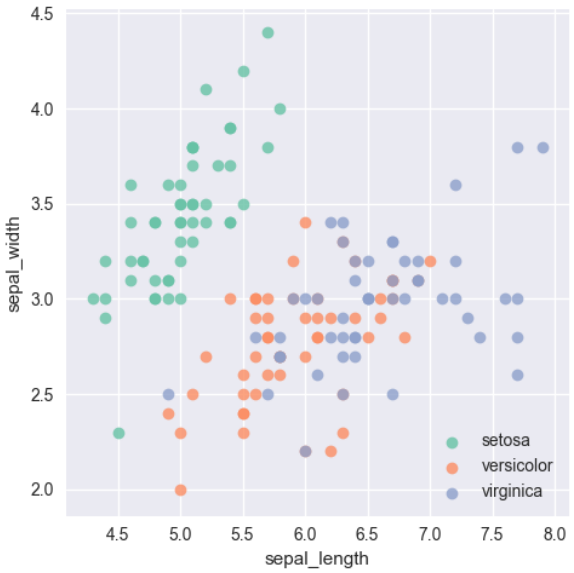
ضمناً گاهی برای مشخص نمودن توزیع آماریِ مشاهدات، از Histogram یا Density Plot در حاشیه‌ی Scatter Plot استفاده می‌شود.

[3] برای اطلاعات بیشتر با Overplotting از این لینک استفاده کنید: «https://www.data-to-viz.com/caveat/overplotting.html»



شکل ۱- ۱۵ استفاده از Histogram در حاشیه ی Scatter Plot

گاهی برای دسته بندی مشاهدات در Scatter Plot (افزودن یک متغیر غیر عددی یا Categorical به نام دسته) از رنگ‌های مختلف برای نمایش نقاط استفاده می‌شود. در این صورت حتما باید از یک راهنما (Legend) در نمودار استفاده کرد.



شکل ۱- ۱۶ در این نمودار از سه رنگ برای دسته بندی مشاهدات استفاده شده است.

اگر بخواهیم یک متغیر عددی (غیر Categorical) را به عنوان متغیر سوم به Scatter Plot اضافه کنیم، به جای رنگ می‌توان سایز نقطه ها را تغییر داد (نمودار Bubble Plot).

برای کسب اطلاعات بیشتر در رابطه با Scatter Plot از لینک زیر استفاده کنید:

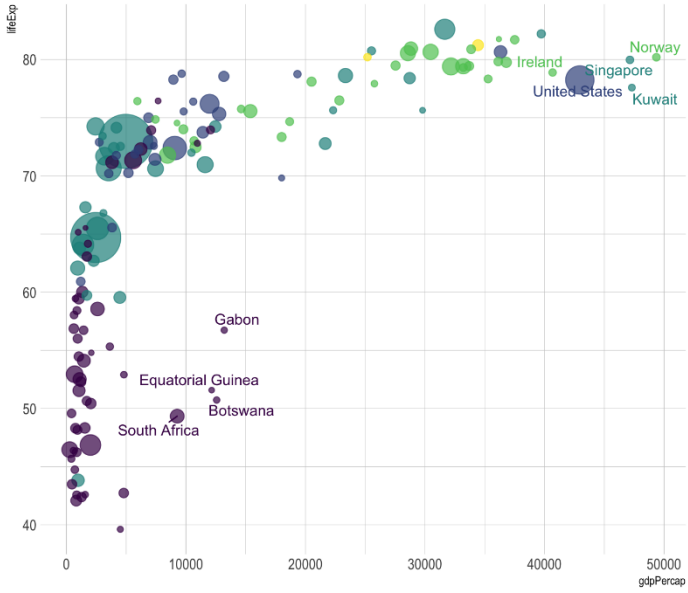
https://www.data-to-viz.com/graph/scatter.html

برای آشنایی با روشِ رسمِ Scatter Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/scatter-plot/

۱ - ۸ - نمودار Bubble Plot

این نمودار، در واقع همان Scatter Plot است که یک **متغیر عددی سوم** توسط اندازه دایره‌ها (حباب‌ها) در آن نشان داده شده است.



شکل ۱- ۱۷ ازبین سه متغیر عددی، متغیری که اهمیت کمتری دارد (در اینجا جمعیت کشور ها) توسط اندازه حباب و دو متغیر مهم تر (در اینجا میزان درآمد ناخالص و میزان امید به زندگی) توسط محور های مختصات نشان داده می‌شوند.

برای کسب اطلاعات بیشتر در رابطه با Bubble Plot از لینک زیر استفاده کنید:

https://www.data-to-viz.com/graph/bubble.html

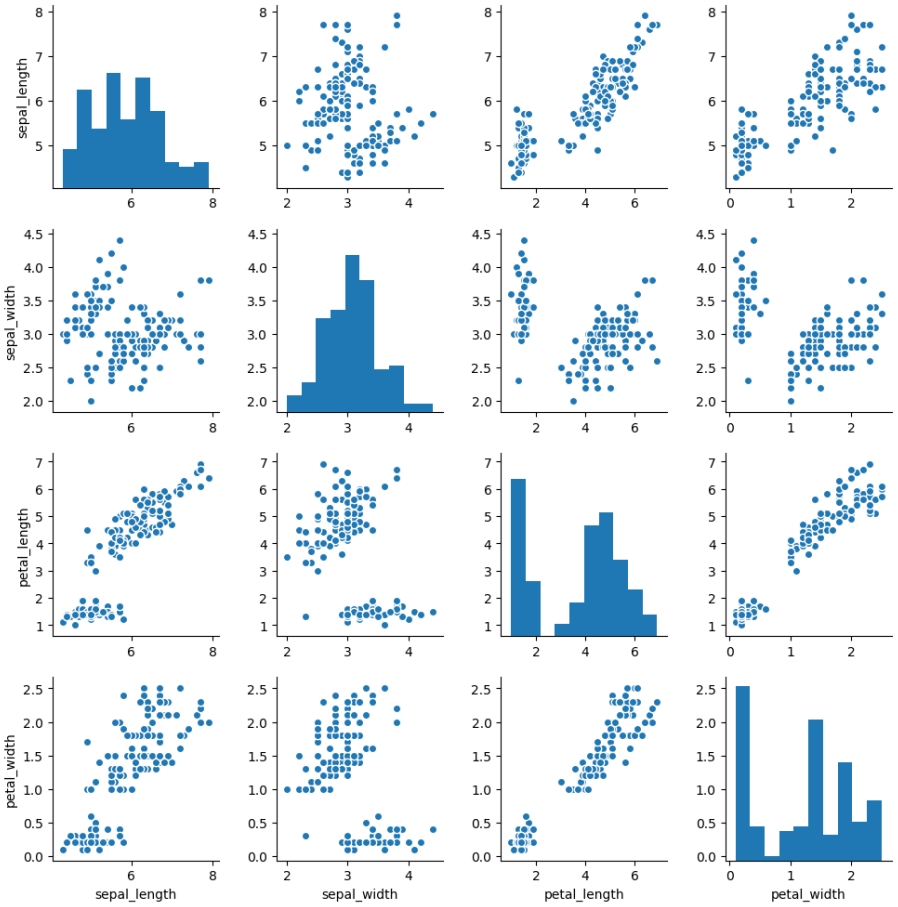
برای آشنایی با روشِ رسمِ Bubble Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/bubble-plot/

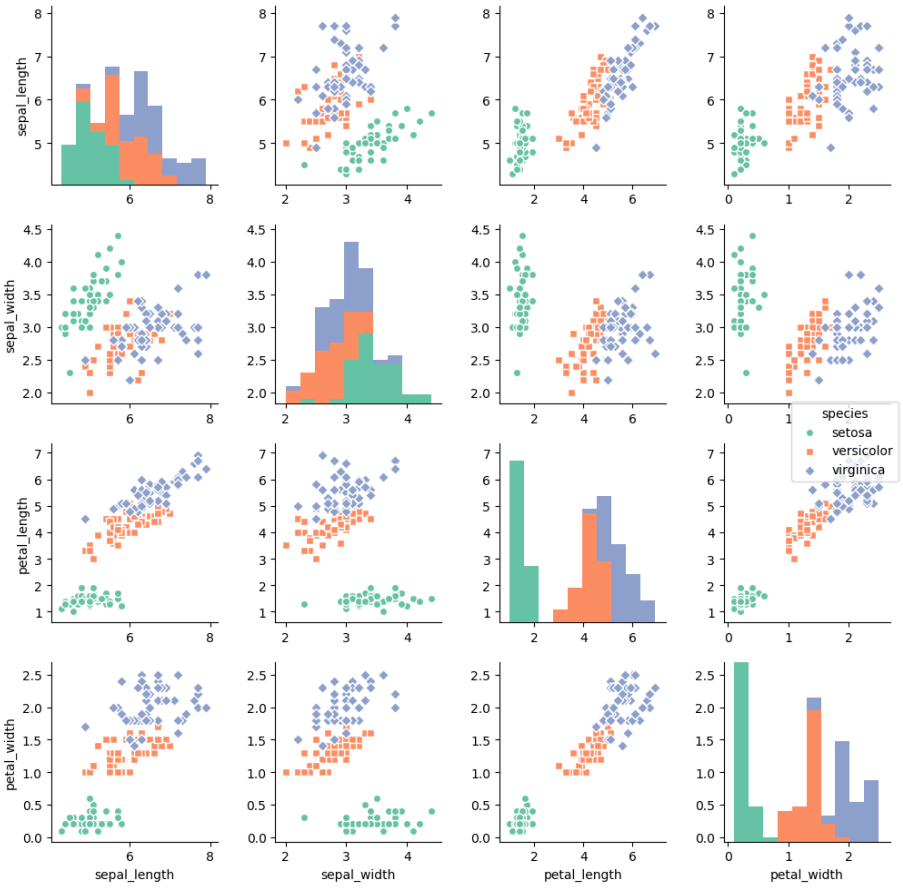
۱ - ۹ - نمودار Correlogram

وقتی **با یک Dataset شامل چند متغیر عددی روبه‌رو می‌شویم**، مطالعه‌ی توزیع آماری تک تک متغیرها با استفاده از Density Plot و همینطور مطالعه‌ی همبستگی بین هر دو متغیر عددی با استفاده از Scatter Plot اولین کاری است که انجام می‌دهیم. نمودار Correlogram یا Correlation Matrix خلاصه‌ی نتایج این فعالیت‌ها را در قالب یک نمودار نمایش می‌دهد.

این نمودار، به شکل یک ماتریس است که متغیرها در سطر و ستون آن تکرار شده‌اند. قطر اصلی این ماتریس شامل Histogram و یا Density Plot هایی برای بررسی توزیع آماری متغیرها و سایر درایه‌های آن شامل Scatter Plot هایی برای بررسی رابطه بین متغیرها هستند.



شکل ۱ -۱۸ در این Correlogram رابطه خطی بین petal_lenght و petal_width به وضوح مشاهده می‌شود.



شکل ۱- ۱۹ در صورت وجود متغیر Categorical می‌توان از رنگ ها به این شکل استفاده کرد.

برای کسب اطلاعات بیشتر در رابطه با Correlogram از لینک زیر استفاده کنید:

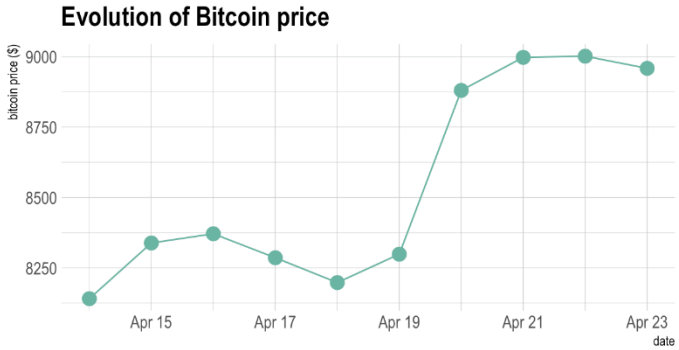
https://www.data-to-viz.com/graph/correlogram.html

برای آشنایی با روشِ رسمِ Correlogram در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/correlogram/

۱ - ۱۰ - نمودار Connected Scatterplot

برای **نمایش رَوَندِ تکاملِ یک متغیرِ کمّی و یا برای نمایش Trend در دیتا طی بازه‌های زمانی مختلف (سریِ زمانی یا Time Series[۴])** مانند نوسانات قیمت Bitcoin در یک بازه زمانی و یا دنباله‌ی سیگنال‌های الکتریکی اندازه گیری شده در یک دستگاه، می‌توان از نمودار Connected Scatterplot استفاده کرد. در این نمودار، دنباله مشاهدات به صورت نقاطی که با خط صاف به هم متصل شده‌اند نمایش داده می‌شود.



شکل ۱- ۲۰ نوسانات قیمت Bitcoin در ماه آپریل سال ۲۰۱۸ (۱۰ روز)

اگر تعداد مشاهدات زیاد باشد (مثلا مطالعه قیمت Bitcoin طی ۵ سال متوالی)، دایره‌ها باعث ناخوانا شدن نمودار می‌شوند. بنابراین در چنین شرایطی، با حذف دایره‌ها نموداری به نام Line Chart رسم می‌کنیم.

برای کسب اطلاعات بیشتر در رابطه با Connected Scatterplot از لینک زیر استفاده کنید:

https://www.data-to-viz.com/graph/connectedscatter.html

برای آشنایی با روشِ رسمِ Connected Scatterplot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

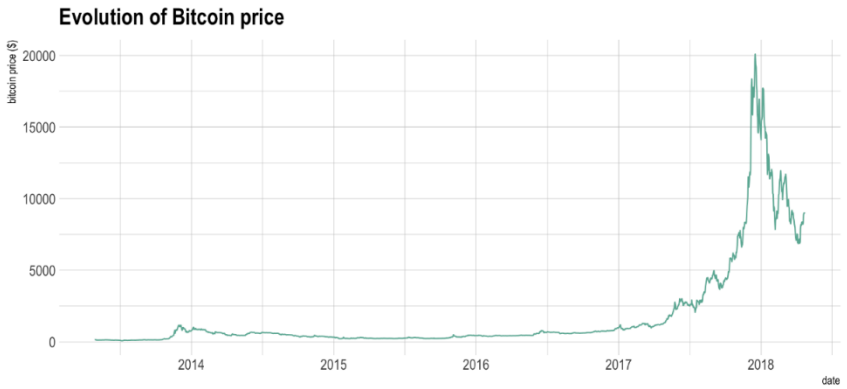
https://www.python-graph-gallery.com/connected-scatter-plot/

۱ - ۱۱ - نمودار Line Chart

با **حذف دایره‌ها از Connected Scatterplot نموداری متشکل از فقط خطوط متصل کننده نقاط** به نام Line Chart پدید می‌آید که برای نمایش نوسانات طولانی مدت (مشاهدات زیاد)، خواناتر است.

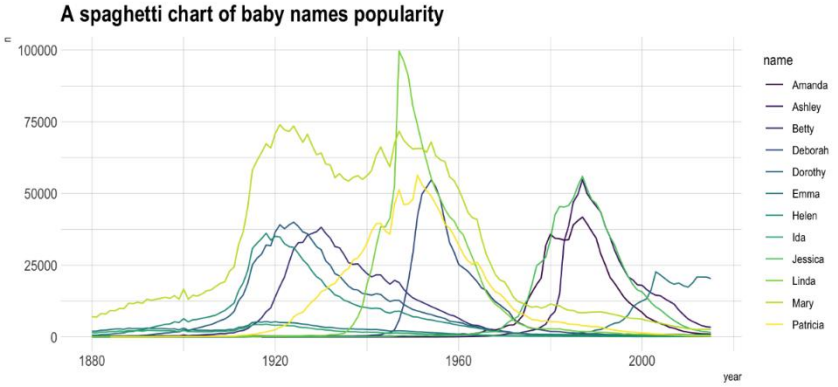
[۴] در علوم مختلف، به یک توالی یا دنباله از متغیرهای تصادفی که در فاصله های زمانی ثابت نمونه برداری شده باشند، اصطلاحاً سری زمانی یا پیشامد تصادفی در مقطع زمان می‌گویند. به عبارت دیگر منظور از یک سری زمانی مجموعه‌ای از داده‌های آماری است که در فواصل زمانی مساوی و منظمی جمع‌آوری شده باشند.

اگر تعداد مشاهدات، محدود است، Connected Scatterplot اطلاعات بیشتری را منتقل می‌کند.



شکل ۱-۲۱ نوسانات قیمت Bitcoin از آپریل ۲۰۱۳ تا آپریل ۲۰۱۸ (۵ سال)

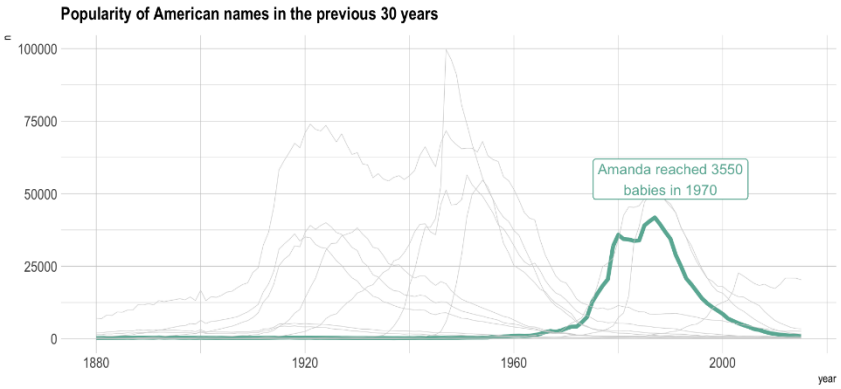
از Line Chart می‌توان برای نمایش نوسانات و تکامل چندین متغیر عددی به‌طور همزمان استفاده کرد. با این حال باید توجه شود که اگر تعداد متغیرهای مورد بررسی زیاد باشد، این نمودار خوانایی خود را از دست داده و به چیزی به نام Spaghetti Chart[5] تبدیل می‌شود.



شکل ۱- ۲۲ نمایش تعداد زیادی متغیر در یک Line Chart نمودار را به یک Spaghetti Chart تبدیل می‌کند.

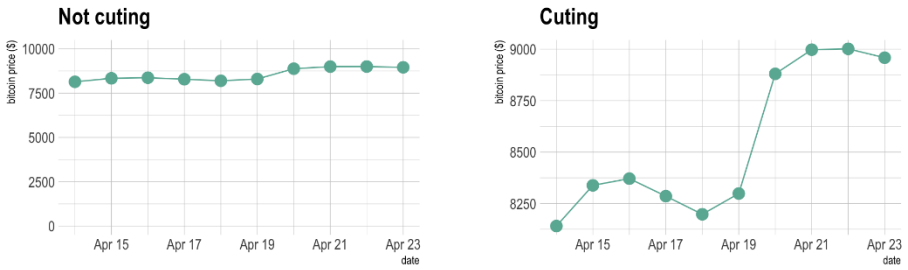
معمولا هدف از نمایش همزمان چند متغیر در یک Line Chart مقایسه یکی از آن‌ها با بقیه است. در چنین حالتی می‌توانید با برجسته کردن نمودار آن متغیر و کمرنگ کردن بقیه از تبدیل شدن نمودار به Spaghetti Chart جلوگیری کنید.

[5] برای اطلاعات بیشتر درمورد Spaghetti Plot به این لینک مراجعه کنید: «https://www.data-to-viz.com/caveat/spaghetti.html»



شکل ۱- ۲۳ برجسته کردن خطوط مربوط به یک متغیر و کمرنگ کردن بقیه خطوط راه حلی برای مقابله با spaghetti Chart است.

نکته دیگری که در رابطه با Line Chart مهم است، این است که **نیازی نیست محور Y در این نمودار حتما از صفر شروع شود.** گاهی نمایش آن از جایی بالاتر از صفر (زوم کردن روی نمودار)، الگوی تغییرات را بسیار بهتر نشان می‌دهد.



شکل ۱- ۲٤ شروع محور Y از صفر در مقایسه با بریدن آن

برای کسب اطلاعات بیشتر در رابطه با Line Chart از لینک زیر استفاده کنید:

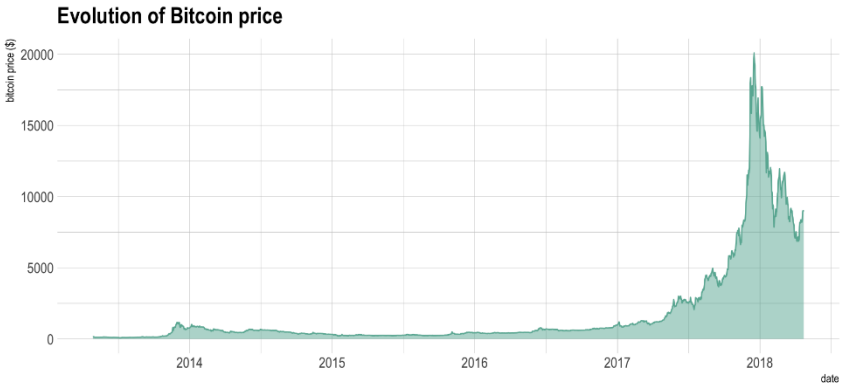
https://www.data-to-viz.com/graph/line.html

برای آشنایی با روشِ رسمِ Line Chart در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/line-chart/

۱ - ۱۲ - نمودار Area Chart

گاهی **با رنگ زدنِ فضای زیرِ منحنیِ** Line Chart **و بررسیِ حجمِ رنگ‌آمیزی شده در طول زمان، نوسانات و الگوی تکاملیِ متغیر، واضح‌تر می‌شود.** معمولا از همان رنگِ استفاده شده برای منحنی با کمی Transparency برای رنگ زدن زیر منحنی استفاده می‌شود.



شکل ۱- ۲۵ تکامل قیمت Bitcoin از آپریل ۲۰۱۳ تا آپریل ۲۰۱۸

برای بررسی نوسانات چند متغیر به طور همزمان در یک Area Chart، مانند Line Chart عمل نمی‌کنیم. بلکه از نمودارهای دیگر مثل Stacked Area Plot استفاده می‌شود.

برای کسب اطلاعات بیشتر در رابطه با Area Chart از لینک زیر استفاده کنید:

https://www.data-to-viz.com/graph/area.html

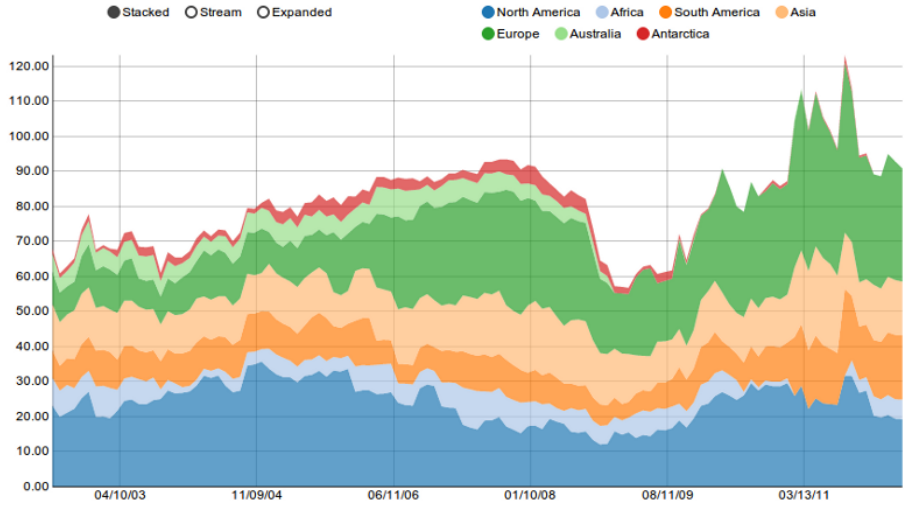
برای آشنایی با روشِ رسمِ Area Chart در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/area-plot/

۱ - ۱۳ - نمودار Stacked Area Plot

برای **نمایشِ روندِ تکاملِ چند متغیرِ عددی و یا نمایشِ روندِ تکاملِ یک متغیرِ عددی برای چندین گروه مختلف و مقایسه اهمیتِ نسبیِ آن‌ها در یک نمودار**، از Stacked Area Plot استفاده می‌شود. در این

نمودار مقدار هر متغیر (یا گروه) در هر زمان، بر بالای منحنیِ مربوط به متغیرِ (گروه) قبلی، به صورت تجمعی رسم می‌شود. بنابراین بالاترین منحنی، معادلِ نمودار Area Plot برای مجموع همه متغیرها (گروه‌ها) است.



شکل ۱-۲۶ نمودار Stacked Area Plot و بررسی الگوی تکاملِ یک متغیر برای قاره های مختلف. بالاترین منحنی، نشان دهنده الگوی تکاملِ متغیر برای کل جهان است. همانطور که در شکل مشخص است، در اواخر نمودار، سهم اروپا از متغیر مربوطه چشمگیر تر از بقیه قاره ها بوده است.

برای کسب اطلاعات بیشتر در رابطه با Stacked Area Plot از لینک زیر استفاده کنید:

https://www.data-to-viz.com/graph/stackedarea.html

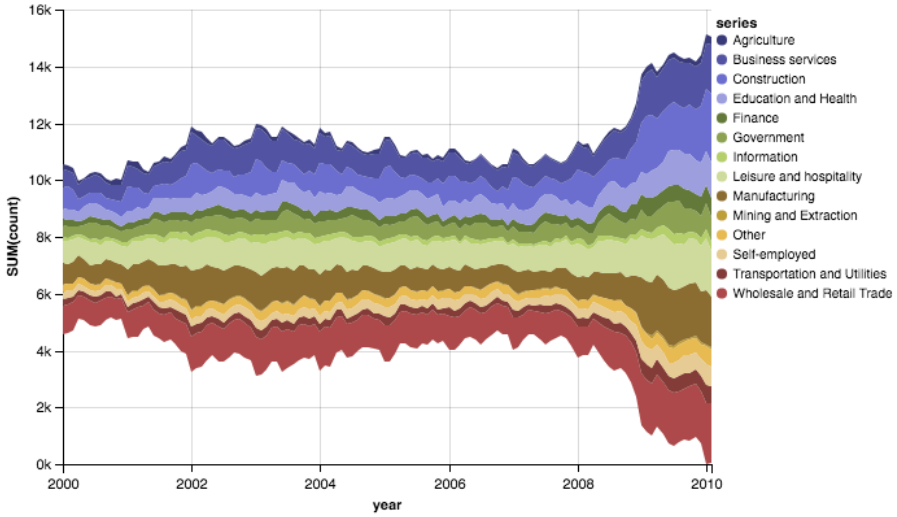
برای آشنایی با روشِ رسمِ Stacked Area Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/stacked-area-plot/

۱ - ۱۴ - نمودار Stream Graph

گاهی تنها چیزی که از یک Stacked Area Plot می‌خواهیم، **نمایشِ نسبیِ سهمِ هر گروه (هر متغیر) از کل گروه‌ها طی یک بازه زمانی** است. در چنین شرایطی می‌توان از یک نوع خاص از Stacked Area Plot به نام Stream Graph استفاده کرد. در Stream Graph تیزی لبه‌ها گرفته شده و منحنی‌ها حول یک محور افقی رسم می‌شوند.

واضح است که چنین نموداری فقط منتقل کننده‌ی نسبتِ سهمِ هر گروه از کل است و برای بررسی الگوی تکاملی یک گروه خاص مناسب نیست.



شکل ۱- ۲۷ نمودار Stream Graph

برای کسب اطلاعات بیشتر در رابطه با Stream Graph از لینک زیر استفاده کنید:

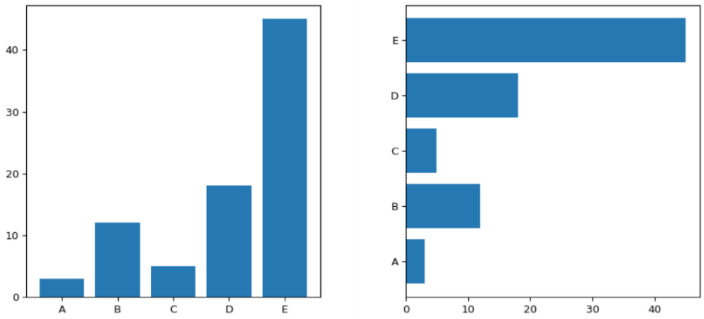
https://www.data-to-viz.com/graph/streamgraph.html

برای آشنایی با روشِ رسمِ Stream Graph در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/streamchart/

۱ - ۱۵ - نمودار Barplot

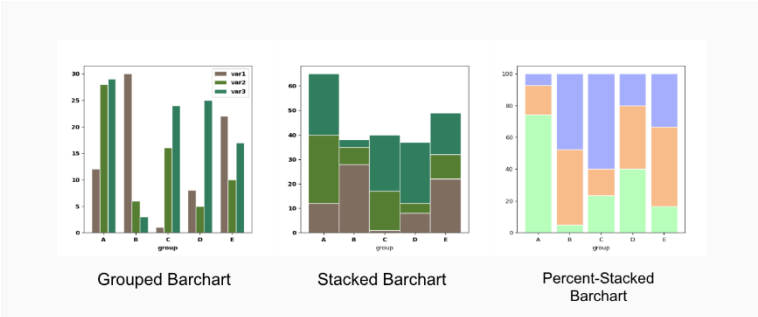
این نمودار برای **نمایش رابطه‌ی بین یک متغیر دسته‌ای (Categorical) و یک متغیر عددی (Numerical)** بکار می‌رود. طول Bar ها نشان دهنده اندازه متغیر عددی به ازای هر متغیر دسته‌ای است. توجه کنید که این نمودار را با Histogram اشتباه نگیرید. در Histogram همیشه یک متغیر عددی داریم که می‌خواهیم توزیع آماری آن را نمایش دهیم اما در Barplot یک متغیر دسته‌ای و یک متغیر عددی داریم و می‌خواهیم رابطه بین آن‌ها را به تصویر بکشیم. اگر ترتیب مقادیر متغیر دسته‌ای اهمیتی ندارد، بهتر است آن‌ها را به ترتیب مقدار عددی مرتب کنید تا نمودار خواناتری داشته باشید. زیرا در این صورت نمودار Barplot نه تنها نمایشگر مقدار عددی هر دسته است، بلکه ترتیب و رتبه ی هر دسته را نیز نشان می‌دهد.



شکل ۱- ۲۸ نمودار Barplot که ترتیب مقادیر متغیر دسته ای در آن مهم بوده (مثلا ماه های سال) و به همین دلیل بر اساس مقدار عددی مرتب نشده

اگر دو متغیر دسته‌ای موجود باشند می‌توان از انواع زیر استفاده کرد:

۱. نمودار Grouped Barplot: مقادیر مربوط به متغیر دسته‌ای دوم کنار هم قرار می‌گیرند.
۲. نمودار Stacked Barplot: مقادیر مربوط به متغیر دسته‌ای دوم به صورت تجمعی روی هم قرار می‌گیرند.
۳. نمودار Percent-stacked Barplot: مقادیر مربوط به متغیر دسته‌ای دوم به صورت نسبی (درصدی) روی هم قرار می‌گیرند. به این معنی که مجموع آن‌ها باید ۱۰۰ شود.



شکل ۱- ۲۹ نمایش داده‌های دو متغیر دسته‌ای به سه نوع نمودار Groupec، Stacked و Percent-Stacked

برای کسب اطلاعات بیشتر در رابطه با Barplot از لینک زیر استفاده کنید:

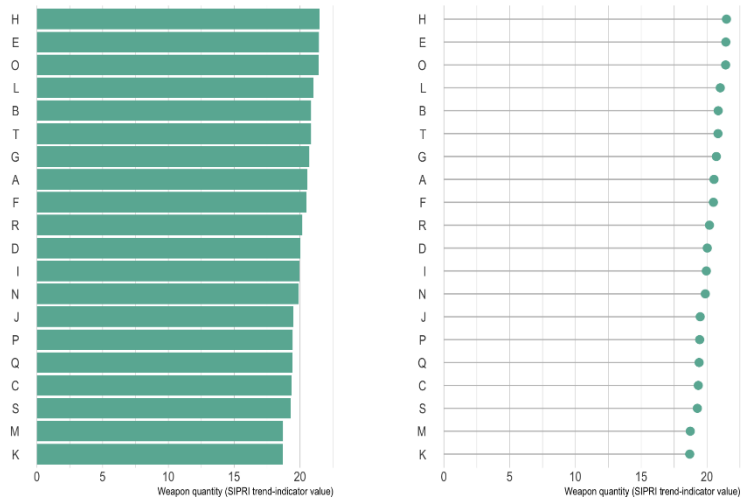
https://www.data-to-viz.com/graph/barplot.html

برای آشنایی با روشِ رسمِ Barplot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/barplot/

۱ - ۱۶ - نمودار Lollipop Plot

همانطور که در شکل زیر مشخص است، زمانی که **طول تعدادی از Bar های کنار هم در Barplot هم اندازه است**، شکل ناهنجاری شبیه به کرکره ایجاد می‌شود. در این حالت، برای زیبایی بیشتر می‌توان از Lollipop Plot استفاده کرد.



شکل ۱ -۳۰ استفاده از Lollipop Plot به جای Barplot

برای کسب اطلاعات بیشتر در رابطه با Lollipop Plot از لینک زیر استفاده کنید:

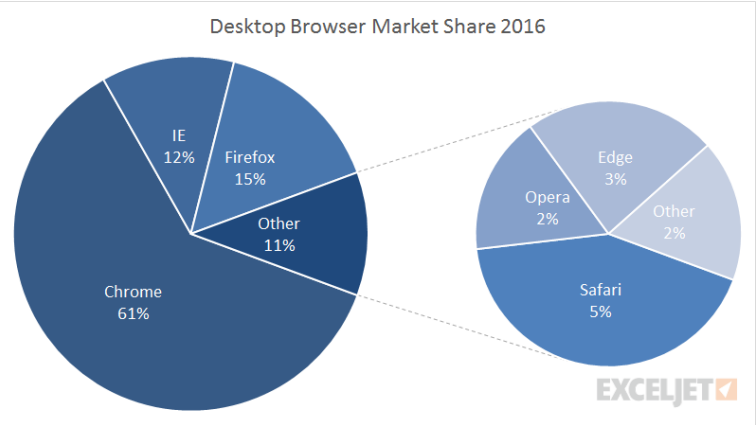
https://www.data-to-viz.com/graph/lollipop.html

برای آشنایی با روشِ رسمِ Lollipop Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/lollipop-plot/

۱ - ۱۷ - نمودار Pie Plot

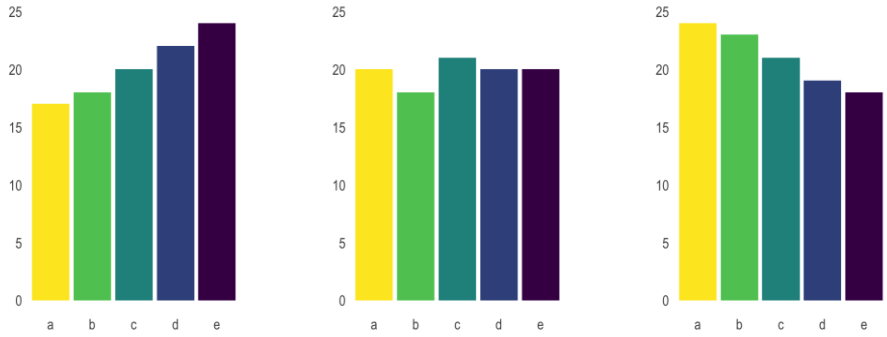
برای **نمایش دادنِ سهم مقادیر یک متغیر دسته‌ای از کل مقادیر**، می‌توان از Pie Plot استفاده کرد. در این نمودار به ازای هر متغیر دسته‌ای، یک برش از دایره در نظر گرفته می‌شود که زاویه‌ی آن بر اساس درصد سهم آن گروه از کل گروه‌ها تعیین می‌شود.



شکل ۱- ۳۱ نمودار دایره‌ای مرورگر های نسخه رومیزی

از آنجا که ذهن انسان در تشخیص و مقایسه سریع مفهوم زاویه خیلی خوب عمل نمی‌کند، **بیشتر وقت‌ها، استفاده از این نمودار بدترین انتخاب ممکن است و بهتر است به جای آن از Barplot استفاده شود.** برای نمونه سعی کنید در Pie Plotهای زیر، برش‌ها را از کوچکترین به بزرگترین پیدا و مرتب کنید.

حال نتیجه را با تصویر زیر مقایسه کنید!



برای کسب اطلاعات بیشتر در رابطه با Pie Plot از لینک زیر استفاده کنید:

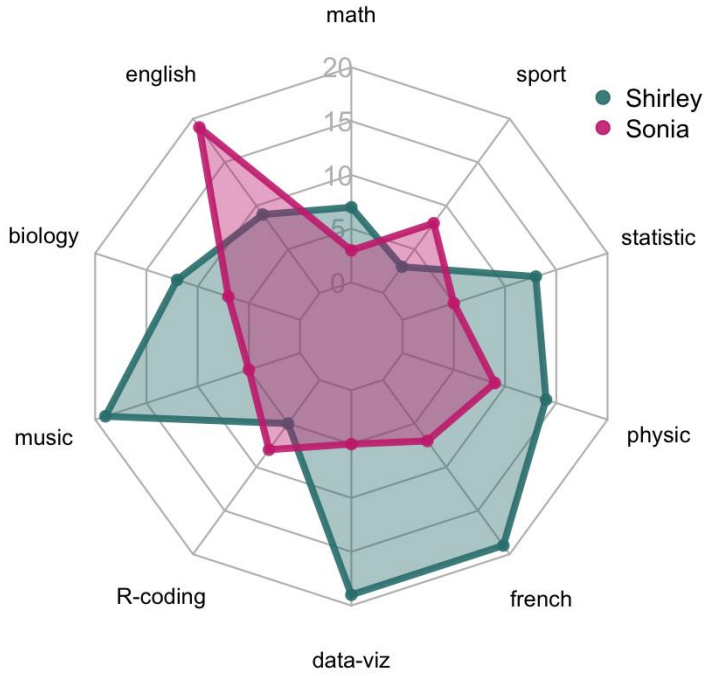
https://www.data-to-viz.com/caveat/pie.html

برای آشنایی با روشِ رسمِ Pie Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/pie-plot/

۱ - ۱۸ - نمودار Radar Chart

زمانی که **تعدادی متغیر کمّی وجود دارد و می‌خواهیم تعداد اندکی از مشاهدات مربوط به این متغیرها را نشان دهیم**، می‌توانیم از Radar Chart (نام‌های دیگر: Spider Chart و Web Chart) استفاده کنیم. در این نمودار، هر مشاهده شکل ظاهری مختص به خود را دارد.



شکل ۱- ۳۲ در حالت کلی عملکرد Shirly بهتر از Sonia بوده، اما در درس های sports و english و R-coding نمره ی Sonia بهتر است.

برای کسب اطلاعات بیشتر در رابطه با Radar Chart از لینک زیر استفاده کنید:

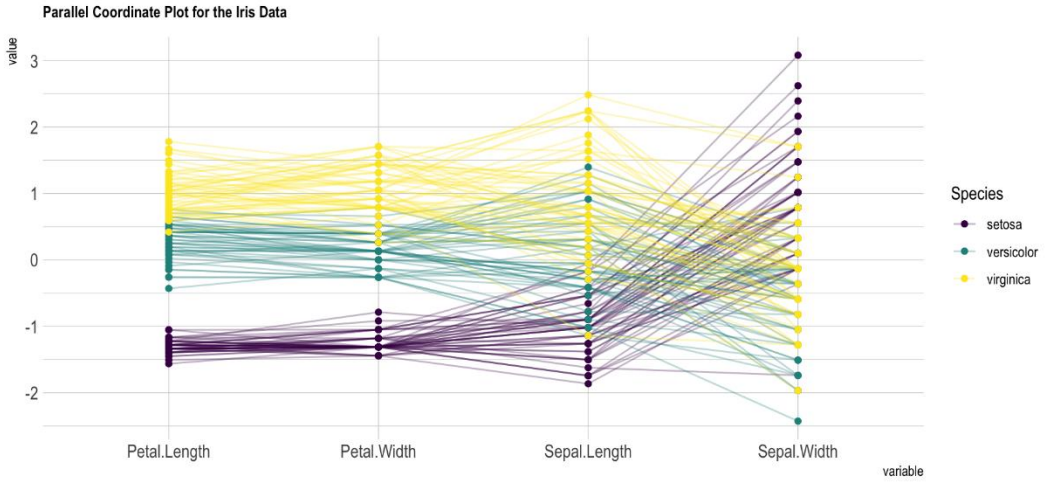
https://www.data-to-viz.com/caveat/spider.html

برای آشنایی با روشِ رسمِ Radar Chart در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/radar-chart/

۱ - ۱۹ - نمودار Parallel Plot

زمانی که **تعدادی متغیر کمّی وجود دارد و می‌خواهیم تعداد زیادی از مشاهدات مربوط به این متغیرها را در یک شکل، نشان دهیم**، می‌توانیم از Parallel Plot (نام کاملتر: Parallel Coordinates Plot) استفاده کنیم. در این نمودار، هر محورِ عمودی، معرف یک Feature (متغیر) است که می‌تواند واحد خود را داشته باشد و مشاهدات توسط خطوطی که نقاطِ روی محور های عمودی را قطع می‌کنند نشان داده می‌شوند. همانطور که در شکل دیده می‌شود، می‌توان از خطوط رنگی برای دسته‌بندی مشاهدات استفاده کرد.



شکل ۱ -۳۳ همانطور که در شکل مشخص است، مشاهدات دسته ی setosa دارای Petal های کوچکتر و Sepal های عریض تر هستند.

برای کسب اطلاعات بیشتر در رابطه با Parallel Plot از لینک زیر استفاده کنید:

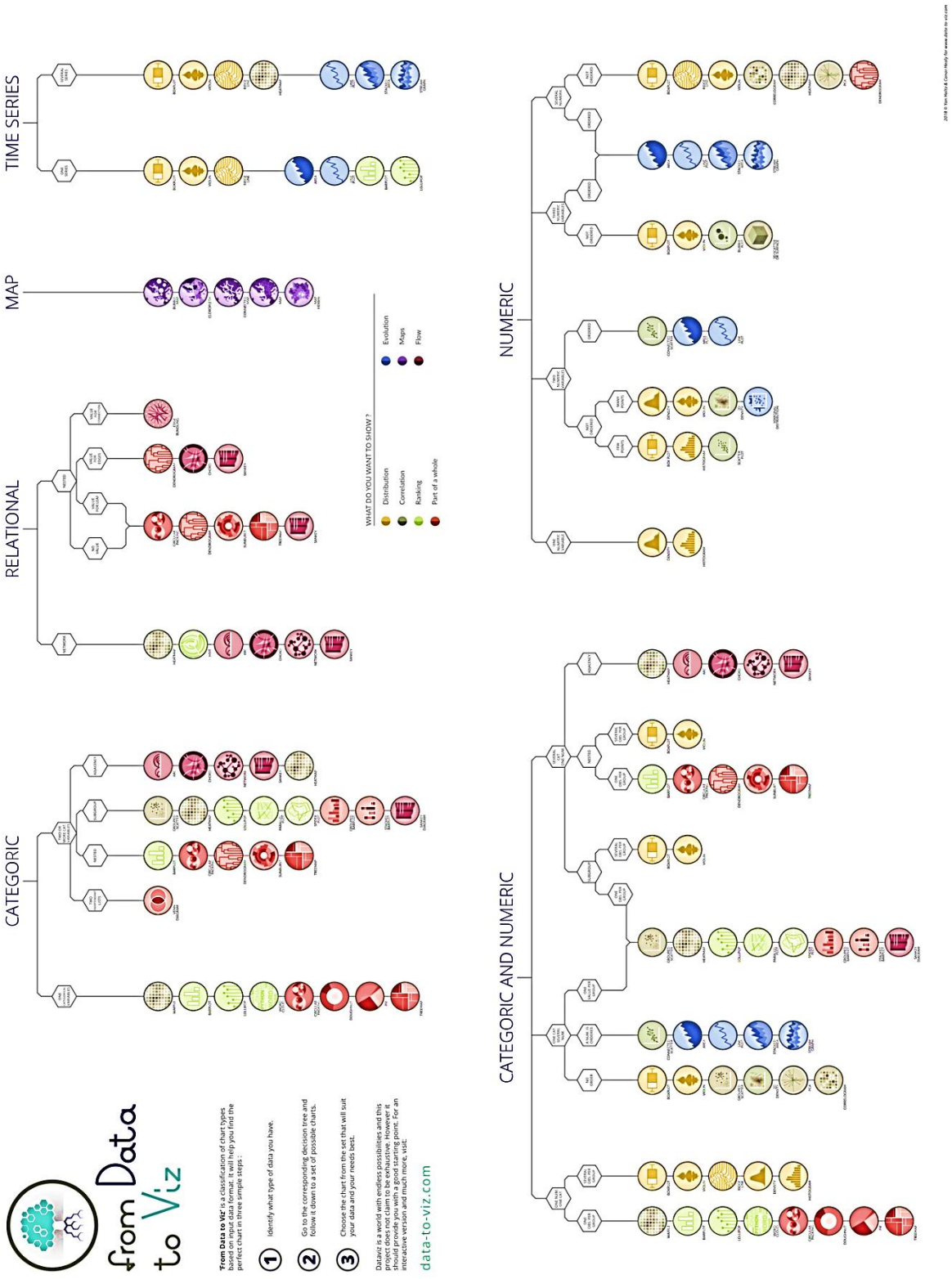
https://www.data-to-viz.com/graph/parallel.html

برای آشنایی با روشِ رسمِ Parallel Plot در زبان برنامه‌نویسی Python از لینک زیر استفاده کنید:

https://www.python-graph-gallery.com/parallel-plot/

۱ - ۲۰ - سایر نمودارها

تا اینجا با ۱۹ عدد از پرکاربردترین نمودارها در Data Visualization آشنا شدید، اما همانطور که قبلا هم اشاره شد، اینها فقط تعدادی از نمودارهای آماری موجود برای به تصویر کشیدن داده‌ها اند و می‌توانید با مراجعه به سایت data-to-viz.com با سایر نمودارها نیز آشنا شوید.



شکل ۱ -۳٤ پوستر انواع نمودار های قابل استفاده برای به تصویر کشیدن داده‌ها بر اساس نوع داده ها را نشان می‌دهد.

حال که با مهم‌ترین نمودارهای آماریِ مورد استفاده در Data Visualization و کاربردهای آنها آشنا شدید، اکیداً توصیه می‌کنیم برای جمع‌بندی مطالب، لینک‌های زیر را به ترتیب مطالعه کنید. در این لینک‌ها، Dataset هایی (به ترتیب از موارد ساده تر تا پیچیده تر) معرفی شده و با استفاده از نمودارهای معرفی شده در این پست مورد مطالعه و تحلیل قرار گرفته‌اند.

https://www.data-to-viz.com/story/OneNum.html

https://www.data-to-viz.com/story/TwoNum.html

https://www.data-to-viz.com/story/TwoNumOrdered.html

https://www.data-to-viz.com/story/ThreeNum.html

https://www.data-to-viz.com/story/OneCatSevOrderedNum.html

https://www.data-to-viz.com/story/SeveralNum.html

https://www.data-to-viz.com/story/SevCatOneNumNestedOneObsPerGroup.html

https://www.data-to-viz.com/story/OneNumOneCat.html

https://www.data-to-viz.com/story/OneNumOneCatSeveralObs.html

منابع

برگرفته از نوشتار جناب **حمیدرضا حسین خانی** از کانال مدرسه هوش مصنوعی «School of AI» به نشانی؛

https://vrgl.ir/iqWnF

https://t.me/schoolofai