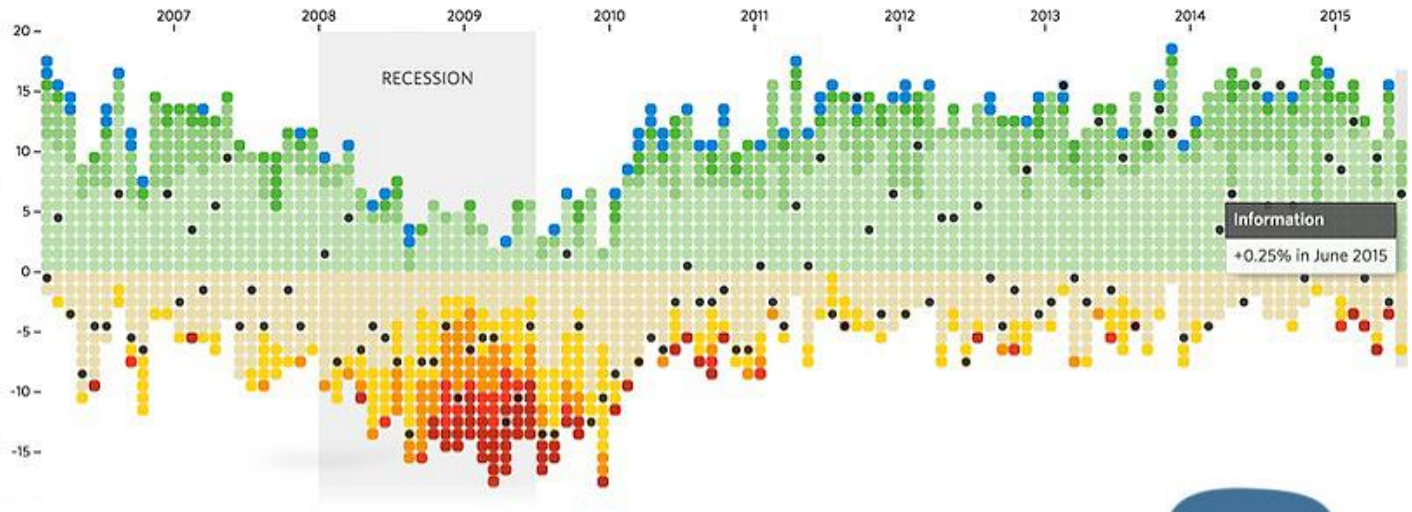


Data Visualization

بصری سازی داده ها در پایتون

علی نظری زاده

LinkedIn linkedin.com/in/ali-nazarizadeh

چطورید رفقا سلام، توی این آموزش میخوایم درمورد بصری سازی داده ها یا data visualization صحبت کنیم. به کمک بصری سازی داده ها یا data visualization، می تونیم یک دید بصری یا گرافیکی نسبت به داده ها داشته باشیم. خیلی از اوقات با نمایش داده ها در قالب نمودارها و چارت های گرافیکی، می توان به دانش خوبی درباره داده ها دست یافت. توی این جزوه قراره با مهم ترین چارت های بصری سازی داده ها در کتابخونه pandas آشنا بشیم !! شاید این جمله برای دوستان تازه کار کمی عجیب باشه چون معمولا pandas رو برای کار با داده ها و پیش پردازش میشناسن که کاملا هم درسته اما باید بگم که pandas رفیق با معرفتیه و علاوه بر این که تو حوزه پیش پردازش و تحلیل داده گل کاشته، تو حوزه بصری سازی داده ها هم خوب عمل کرده و با همین کتابخونه باحال میشه خیلی از چارت ها و شکل های گرافیکی رو کشید. چی از این بهتر، اینجوری دیگه لازم نیست با کتابخونه های دیگه ای کار کنیم و با همین رفیق خوبمون همه کار میکنیم. اما دقت کنید که pandas تو حوزه بصری سازی داده خیلی قوی نیست اما خب در حدی هستش که کارهای ما رو راه بندازه. فکر کنم زیاد حرف زدم درسته ؟ خب بریم وارد داستان بشیم و ببینیم جریان از چه قراره. چارت های مهمی که تو pandas هستش و قراره بررسی کنیم به به شرح زیر هستند :

1 : pie

2 : scatter

3 : hist

4 : bar

5 : barh

6 : area

7 : hexbin

8 : line

9 : box

بریم اولین چارت یعنی pie رو با هم ببینیم اما خب باید داده ای باشه دیگه نه ؟ خالی خالی که نمیشه درسته ؟ برای این کار من دیتاست افراد دیابتی رو به کمک pandas می خونم. Pandas رو بلد نیستید ؟ باشه میگم. ابتدا pandas و اون یکی رفیق ناخلف اش یعنی numpy رو import می کنیم و یه نامی بهشون میدیم که به ترتیب pd و np. حالا numpy چیه ؟ اینم یه لایبرری برای کار با ماتریس ها و داده هستش که این جا خیلی ازش استفاده نمیکنیم اما به رسم ادب import اش میکنیم. پس هرجا خواستم از pandas استفاده کنم فقط کافیه که بنویسم pd. حالا خیلی ساده به کمک تابع read_csv() که تو pandas هست و اینجا ما اسمشو pd گذاشتیم میتونیم داده هایی که فرمتشون csv هست رو بخونیم. خلاصه اینکه این دیتاست شامل 752 سطر در 9 ستون است که سطرها نشون دهنده تعداد افراد هستند و ستون ها نشون دهنده ویژگی افراد. ستون آخر یعنی Outcome نشون دهنده وضعیت دیابت افراده که اگر 1 باشه یعنی اون فرد دیابت داره و اگر 0 باشه یعنی دیابت نداره. پس ما اطلاعات 752 فرد که برای هر فرد 8 ویژگی ثبت شده است رو داریم و البته به همراه وضعیت دیابت داشتن یا نداشتن.

لینک دیتاست افراد دیابتی : (Pima Indians Diabetes Database)

https://www.kaggle.com/uciml/pima-indians-diabetes-database

```
import pandas as pd
import numpy as np
```

```
df_Diabetes = pd.read_csv("E:\\Dataset\\Diabetes.csv")
df_Diabetes
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	0
1	1	85	66	29	0	26.6	0.351	31	1
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	1
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
747	1	81	74	41	57	46.3	1.096	32	0
748	3	187	70	22	200	36.4	0.408	36	1
749	6	162	62	0	0	24.3	0.178	50	1
750	4	136	70	0	0	31.2	1.182	22	1
751	1	121	78	39	74	39.0	0.261	28	0

752 rows × 9 columns

خب بریم سراغ اولین نمودار یا چارت که اسمش pie هست. یه نکته خیلی خیلی خیلی مهم. برای اینکه از چارت های گرافیکی که تو pandas است استفاده کنید باید وارد بخش plot بشید. پس تو بخش plot همه چارت ها هست که ما به ترتیب اونا رو بررسی می کنیم.

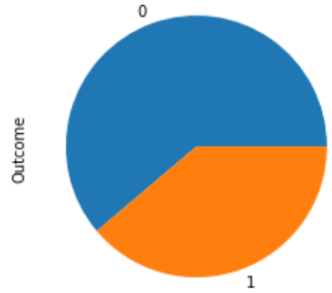
pie

```
df_Diabetes['Outcome'].value_counts()
```

```
0    460
1    292
Name: Outcome, dtype: int64
```

```
df_Diabetes['Outcome'].value_counts().plot.pie();

#x = df_Diabetes['Outcome'].value_counts()
#x.plot.pie()
```



همون طور که میبینید، pie یک چارت دایره ای شکله که من تعداد افراد دیابتی و غیر دیابتی رو در قالب این شکل نشون دادم. به کمک value_counts() تعداد 0 و 1 های ستون Outcoume که به ترتیب 460 و 292 هستش رو حساب کردم بعد بخش plot و بعد pie() رو زدم. پس باز تکررا میکنم این که ما باید اول بریم بخش plot بعد بریم چارت مورد نظرمون رو انتخاب کنیم. اما خب ویژگی های دیگه ای هم میشه براش در نظر گرفت و خوشکلش کرد، مثل دستورات زیر:

Labels = قرار دادن نامی برای هر بخش چارت

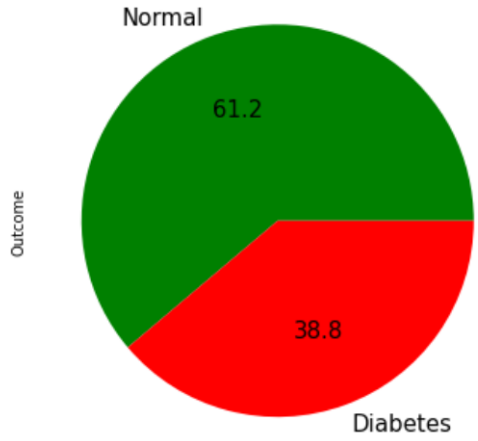
Colors = تغیر رنگ ها

Autopct = تعین اینکه تا چند رقم اعشار، درصد تعلق هر بخش مشخص شود. اینجا تا یک رقم اعشار در نظر گرفته شده است.

Fontsize = مشخص کردن سایز فونت

Figure = مشخص کردن سایز فیگز یا همین اندازه چارت

```
df_Diabetes['Outcome'].value_counts().plot.pie(labels=["Normal", "Diabetes"],
                        colors=["g", "r"], autopct="%.1f", fontsize=15,
                        figsize=(6, 6));
```



اجازه میدید یه دیتاست دیگه رو بخونم ؟ ممنون. این دیتاست خیلی معروفه و همه افرادی که تو حوزه علوم داده هستند اونو میشناسن. اگر کسی ادعای هوش مصنوعی میکنه و این دیتاست رو نمیشناسه، به نظرم باید جمع کنه از این کشور بره. این دیتاست اطلاعات 150 گونه گیاه هستش که برای هر گیاه 4 ویژگی ثبت شده و در ستون آخر یعنی پنجم مشخص شده که گیاه از چه گونه ای است. روی اون ستون آخر value_counts() میزنیم تا ببینیم اولا چند گونه گیاه داریم و دوما از هر گونه چند تا داریم. طبق شکل زیر ما سه گونه گیاه داریم که تعداد هرگونه گیاه هم 50 تاست.

Sepal length : طول کاسبرگ

Sepal width : عرض کاسبرگ

petal length : طول گلبرگ

Petal width : عرض گلبرگ

لینک دیتاست iris :

https://gist.github.com/netj/8836201

```
df_iris = pd.read_csv('E:\\Test\\iris.csv')
df_iris
```

	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	Setosa
1	4.9	3.0	1.4	0.2	Setosa
2	4.7	3.2	1.3	0.2	Setosa
3	4.6	3.1	1.5	0.2	Setosa
4	5.0	3.6	1.4	0.2	Setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	Virginica
146	6.3	2.5	5.0	1.9	Virginica
147	6.5	3.0	5.2	2.0	Virginica
148	6.2	3.4	5.4	2.3	Virginica
149	5.9	3.0	5.1	1.8	Virginica

150 rows × 5 columns

```
df_iris['variety'].value_counts()
```

```
Setosa        50
Versicolor    50
Virginica     50
Name: variety, dtype: int64
```

من میخوام به کمک تابع replace()مقدارهای Setosa و Versicolor و Virginica رو به ترتیب با 1 و 2 و 3 جایگزین کنم که البته اجباری نیست اما خب به حالت عددی تبدیل شون میکنم. اون inpalce = True هم برای اینه که دیتاست من آپدیت بشه و این مقدارهای 1 و 2 و 3 رو جایگزین کنه.

```
df_iris.variety.replace({'Setosa' : 0, 'Versicolor' : 1, 'Virginica' : 2},
                        inplace=True)
df_iris
```

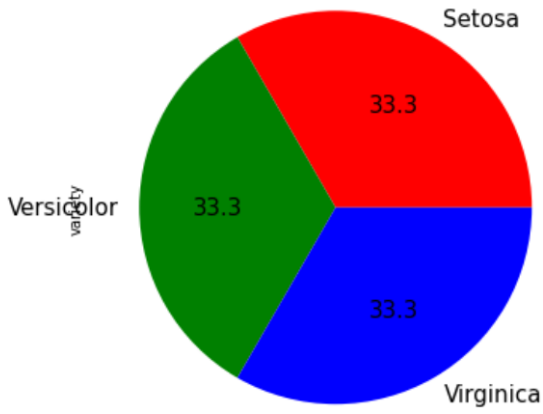
	sepal.length	sepal.width	petal.length	petal.width	variety
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

150 rows × 5 columns

حالا طبق مثال قبل، روی ستون آخر یعنی variety چارت pie رو اعمال میکنیم و همون طور

هم که گفتیم، به تعداد یکسان یعنی 50 تا از هرگونه گیاه وجود دارد.

```
df_iris['variety'].value_counts().plot.pie(labels=["Setosa", "Versicolor",
                                                   "Virginica"],
                                           colors=["r", "g", "b"],
                                           autopct="%.1f",
                                           fontsize=15, figsize=(6, 6));
```



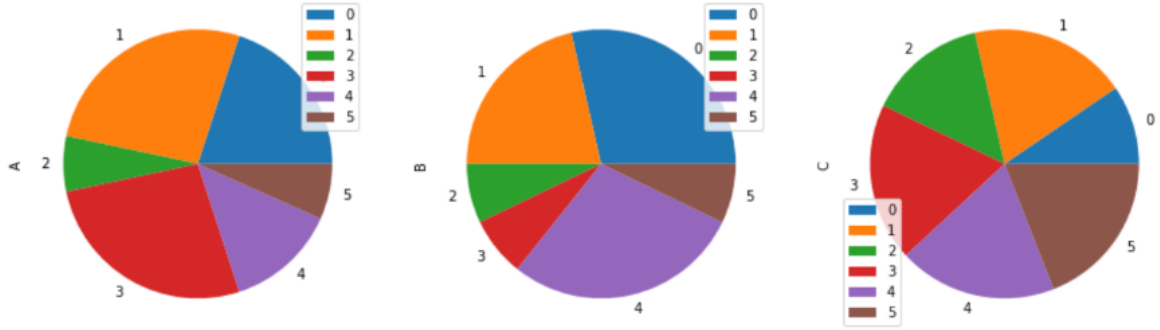
توی مثال آخر از این چارت، 30 تا عدد تصادفی ایجاد و اونا رو توی یک دیتافریم 6 در 3 طبق شکل زیر قرار میدم.

```
df = pd.DataFrame(np.random.randint(1, 5, size=(6,3)),columns=('A', 'B', 'C'))
df
```

	A	B	C
0	3	4	2
1	4	3	4
2	1	1	3
3	4	1	4
4	2	4	4
5	1	1	4

حالا من روی کل دیتافریم که اینجا اسمشو df گذاشتم تابع pie رو اعمال می کنم و چون این دیتافریم یا دیتاست 3 تا ستون A و B و C داره، در نتیجه 3 تا چارت pie هم برای من رسم میکنه.

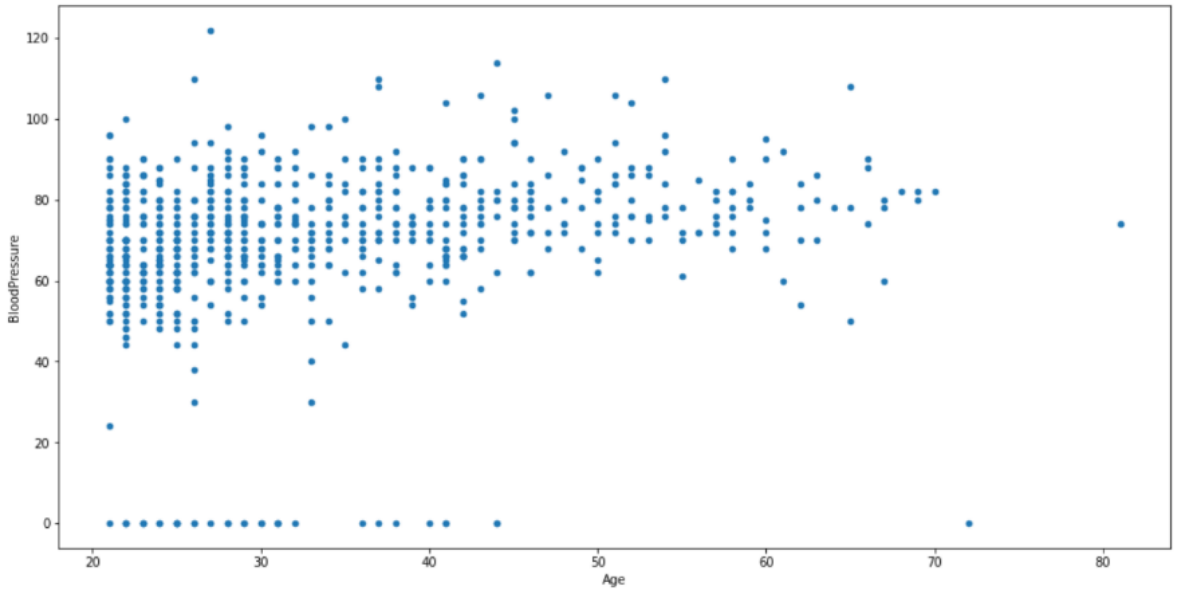
```
df.plot.pie(subplots=True, figsize=(15, 15));
```



خب رفقا بریم سراغ چارت بعدی که اسمش scatter هست و خیلی هم کاربردیه. به کمک این چارت میتونیم داده ها رو توی دو بعد یعنی بر اساس دو ویژگی در صفحه رسم کنیم که من به عنوان نمونه اینجام اومدم بر اساس ویژگی های Age و BloodPressure که تو دیتاست افراد دیابتی بود، چارت scatter رو کشیدم. یعنی رو محور x ها Age و رو محور Y ها BloodPressure.

scatter

```
df_Diabetes.plot.scatter(x='Age', y='BloodPressure', figsize=(16, 8));
```



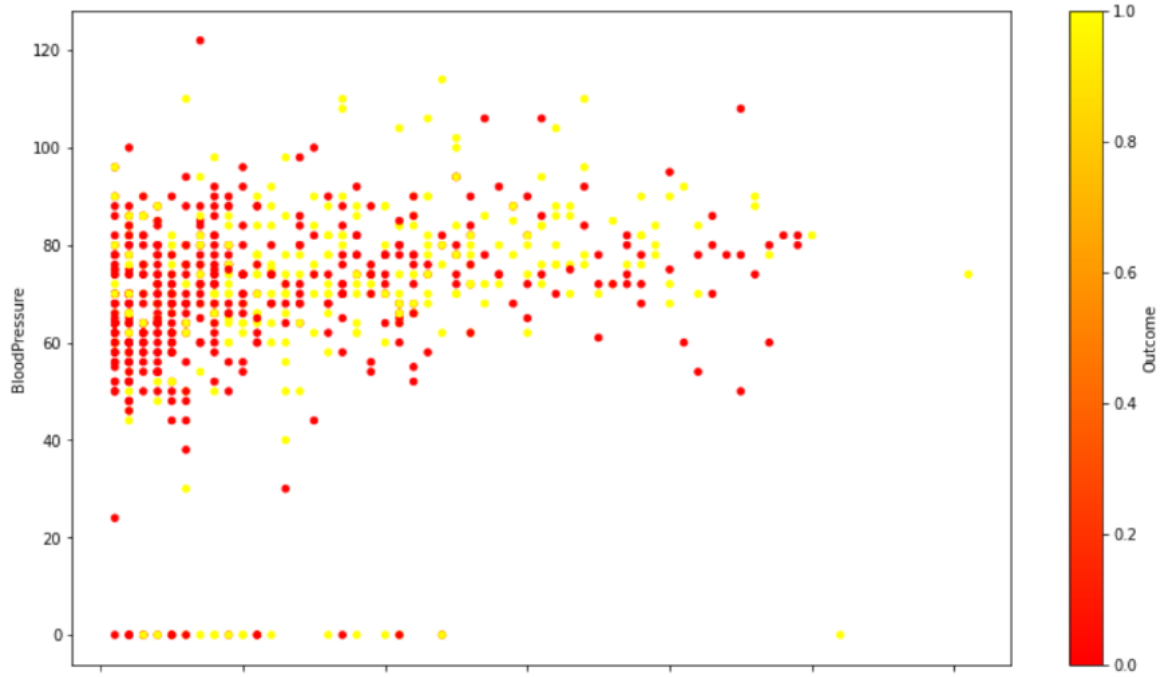
دوستان این چارت خیلی باحاله، چون من اینجا افراد رو بر اساس ویژگی های سن و فشار خون توی صفحه میبینم و با یه نگاه میتونم بفهمم که سن ها و فشار خون افراد در چه بازه ای هست. یه کار خیلی خیلی باحال تری که میتونم انجام بدم اینه که برای هر نقطه که نشون دهنده یک فرد هستش مشخص کنم که آیا دیابت داره یا نه. حالا این چجوری امکان پذیره ؟ ساده ست فقط کافیه که داخل تابع scatter مقدار c رو برابر خروجی یا همون Outcome قرار بدم تا اینجوری افراد دیابتی و غیر دیابتی با دو رنگ، رنگ آمیزی بشن. (دیابتی ها رنگ زرد و غیر دیابتی ها رنگ قرمز). اون colormap چی میگه ؟ هیچی نمیگه فقط به کمک اون میتونیم این ایده رو پیاده سازی کنیم و رنگ بندی ها یا طیف های رنگی مختلفی رو در نظر بگیریم که من اینجا از طیف رنگی autumn استفاده کردم. چه عالی میشد اگر همه طیف های رنگی رو بلد بودم. این که کاری نداره،

من همه رو برات مینویسم. حاجی واقعا رااااست میگی؟؟؟ دمت گرم. آره اگر چشمات ضعیف نباشه، سلول پایین همه رو برات نوشتم و اینکه من حاجی نیستم.

```
# cmap :
'Accent', 'Accent_r', 'Blues', 'Blues_r', 'BrBG', 'BrBG_r', 'BuGn', 'BuGn_r', 'BuPu', 'BuPu_r',
'CMRmap', 'CMRmap_r', 'Dark2', 'Dark2_r', 'GnBu', 'GnBu_r', 'Greens', 'Greens_r', 'Greys',
'Greys_r', 'OrRd', 'OrRd_r', 'Oranges', 'Oranges_r', 'PRGn', 'PRGn_r', 'Paired', 'Paired_r',
'Pastel1', 'Pastel1_r', 'Pastel2', 'Pastel2_r', 'PiYG', 'PiYG_r', 'PuBu', 'PuBuGn', 'PuBuGn_r',
'PuBu_r', 'PuOr', 'PuOr_r', 'PuRd', 'PuRd_r', 'Purples', 'Purples_r', 'RdBu', 'RdBu_r', 'RdGy',
'RdGy_r', 'RdPu', 'RdPu_r', 'RdYlBu', 'RdYlBu_r', 'RdYlGn', 'RdYlGn_r', 'Reds', 'Reds_r',
'Set1','Set1_r', 'Set2', 'Set2_r', 'Set3', 'Set3_r', 'Spectral', 'Spectral_r', 'Wistia',
'Wistia_r','YlGn', 'YlGnBu', 'YlGnBu_r', 'YlGn_r', 'YlOrBr', 'YlOrBr_r', 'YlOrRd', 'YlOrRd_r',
'afmhot','afmhot_r', 'autumn', 'autumn_r', 'binary', 'binary_r', 'bone', 'bone_r', 'brg',
'brg_r', 'bwr','bwr_r', 'cividis', 'cividis_r', 'cool', 'cool_r', 'coolwarm', 'coolwarm_r',
'copper', 'copper_r','cubehelix', 'cubehelix_r', 'flag', 'flag_r', 'gist_earth',
'gist_earth_r', 'gist_gray','gist_gray_r', 'gist_heat', 'gist_heat_r', 'gist_ncar',
'gist_ncar_r', 'gist_rainbow','gist_rainbow_r', 'gist_stern', 'gist_stern_r', 'gist_yarg',
'gist_yarg_r', 'gnuplot','gnuplot2', 'gnuplot2_r', 'gnuplot_r', 'gray', 'gray_r', 'hot',
'hot_r', 'hsv', 'hsv_r','inferno', 'inferno_r', 'jet', 'jet_r', 'magma', 'magma_r',
'nipy_spectral', 'nipy_spectral_r','ocean', 'ocean_r', 'pink', 'pink_r', 'plasma', 'plasma_r',
'prism', 'prism_r', 'rainbow','rainbow_r', 'seismic', 'seismic_r', 'spring', 'spring_r',
'summer', 'summer_r', 'tab10','tab10_r', 'tab20', 'tab20_r', 'tab20b', 'tab20b_r', 'tab20c',
'tab20c_r', 'terrain', 'terrain_r','turbo', 'turbo_r', 'twilight', 'twilight_r',
'twilight_shifted', 'twilight_shifted_r', 'viridis','viridis_r', 'winter', 'winter_r'
```

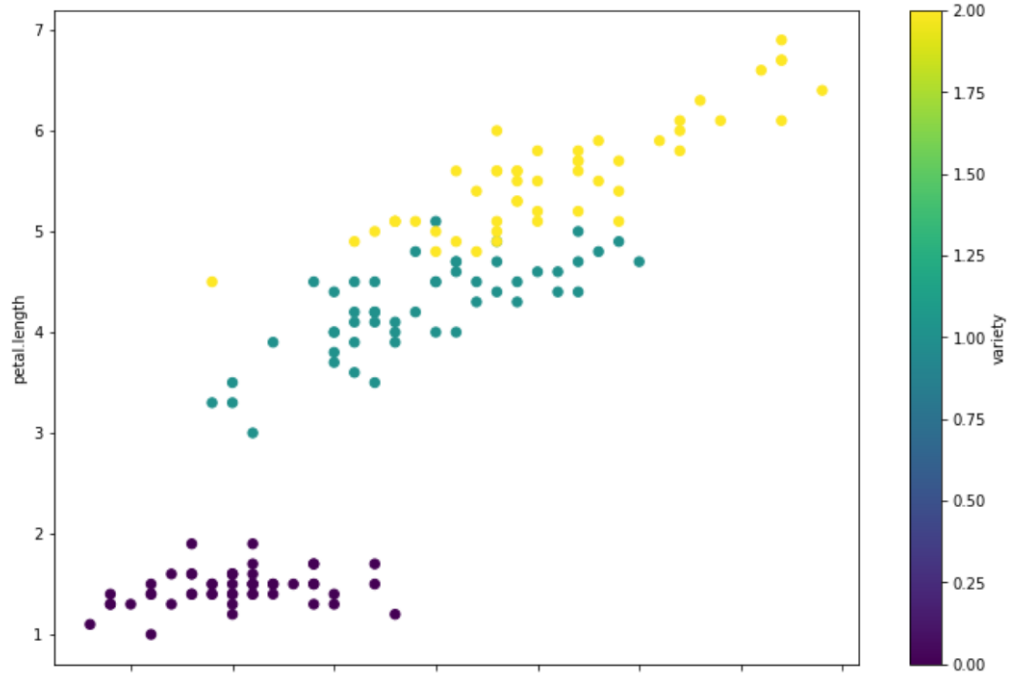
اینجا colormap برابر طیف رنگی autumn هستش که نقاط زرد رنگ نشون دهنده افراد دیابتی و قرمز رنگ نشون دهنده افراد سالم هستند.

```
df_Diabetes.plot.scatter(x='Age', y='BloodPressure', c='Outcome',
                         colormap='autumn', figsize=(14, 8));
```



خب بد نیست که تابع scatter رو روی دیتاست iris هم تست کنیم. اینجا اومدم ویژگی sepal.length رو برای محور افقی یا x و ویژگی petal.length رو برای محور عمودی یا y در نظر گرفتم. اینجا یه مقدار S هم داریم که مقدار 40 بهش دادم. با این مقدار میشه سایز نقاط رو تنظیم کرد که هر چقدر عدد بزرگ تر باشه نقاط روی صفحه بزرگ و هرچقدر عدد کوچک تر باشه، نقاط روی صفحه هم کوچک تر خواهند بود.

```
df_iris.plot.scatter(x='sepal.length', y='petal.length', c='variety', s=40,
                     colormap='viridis', figsize=(12, 8));
```

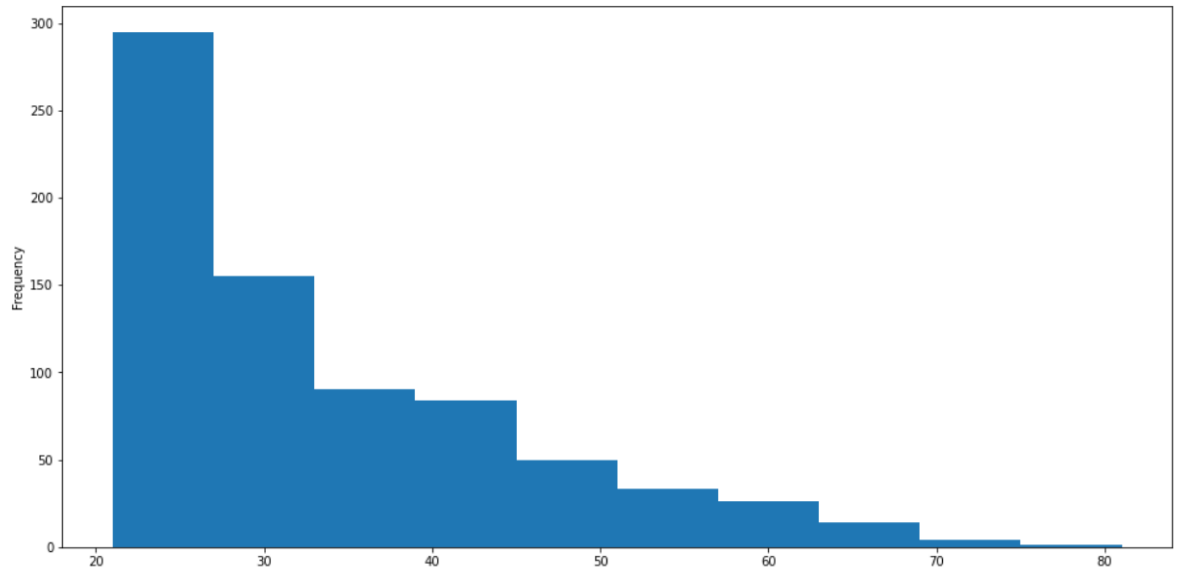


بزن بریم چارت بعدی که اسمش hist است رو بررسی کنیم. این hist حروف اول histogram هست و به کمکش میتونیم فراوانی داده ها رو ببینیم. من اینجا اومدم تابع hist رو روی ویژگی age از دیتاست افراد دیابتی اعمال کردم، یعنی میخوام میزان فراوانی سن افراد رو ببینم. همون طور که مشاهده میکنید، روی محور X ها سن افراد که از 21 تا 81 سال است قرار داره و روی محور

Y تعداد این افراد. با یه نگاه عالمانه و عارفانه میشه فهمید افرادی که بین بازه سنی 21 سال تا 26 سال هستند تعدادشون از همه بیشتره و تقریبا 300 نفر هستند.

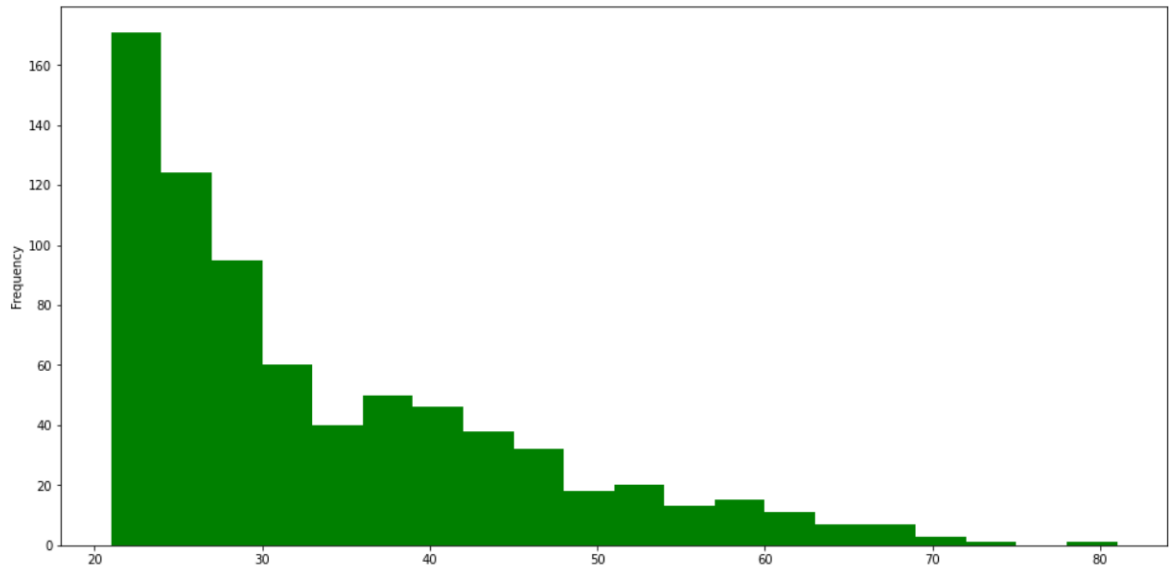
hist

```
df_Diabetes['Age'].plot.hist(figsize=(16, 8));
```



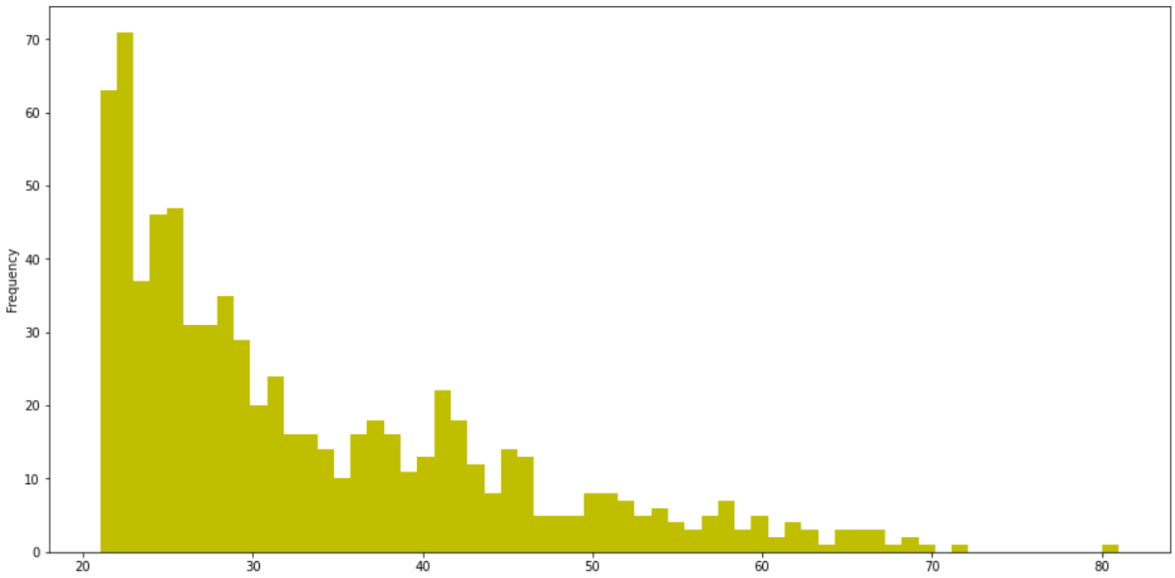
تعداد bins ها یا میله ها به صورت پیش فرض برابر 10 است و سن افراد هم بین بازه 21 تا 81 سال است حالا مشخصه که هر bin, 6 بازه سنی رو شامل میشه. اما من میتونم مقدار bins رو تغییر بدم که در این مثال برابر 20 در نظر گرفتم. یعنی اون بازه سنی 21 تا 81 سال که کلا 61 ساله تقسیم بر تعداد bins ها یا 20 میشه. پس توی نمودار هیستوگرام زیر، هر bins شامل بازه های سنی سه سال سه سال هستش. یعنی اولین bins به بازه سنی 21 تا 23 اشاره میکنه که تقریبا تا عدد 170 بالا رفته. پس میتونم بفهمم افرادی که سنشون بین 21 تا 23 سال هستند، تعدادشون تقریبا 170 نفره.

```
df_Diabetes['Age'].plot.hist(bins=20, figsize=(16, 8), color='g');
```



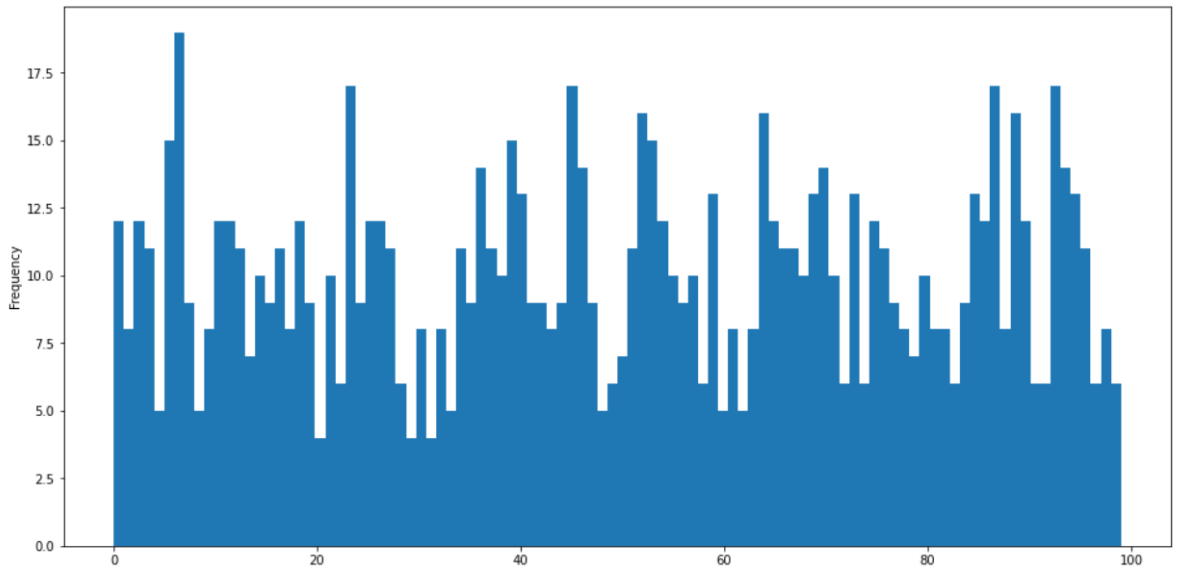
تو مثال زیر تعداد bins رو برابر تعداد بازه سال ها یعنی 61 در نظر گرفتم تا هر bin به یک بازه سنی اشاره کند. مثلا bin اول داره میکه که افرادی که 21 سال هستند تعدادشون 63 نفره.

```
df_Diabetes['Age'].plot.hist(bins=61, figsize=(16, 8), color='y');
```



تو این مثال 1000 تا عدد بین بازه 0 تا 99 ایجاد کردم و فراوانی شون رو در قالب histogram کشیدم.

```
S_random = pd.Series(np.random.randint(0, 100, 1000))
S_random.plot.hist(bins=100, figsize=(16, 8));
```



خب خب میرسیم به پلات یا چارت بعدی که اسم ایشون پلات bar است. قبلش من یه دیتافریم 9 در 4 با اعداد تصادفی 1 تا 9 ایجاد می کنم.

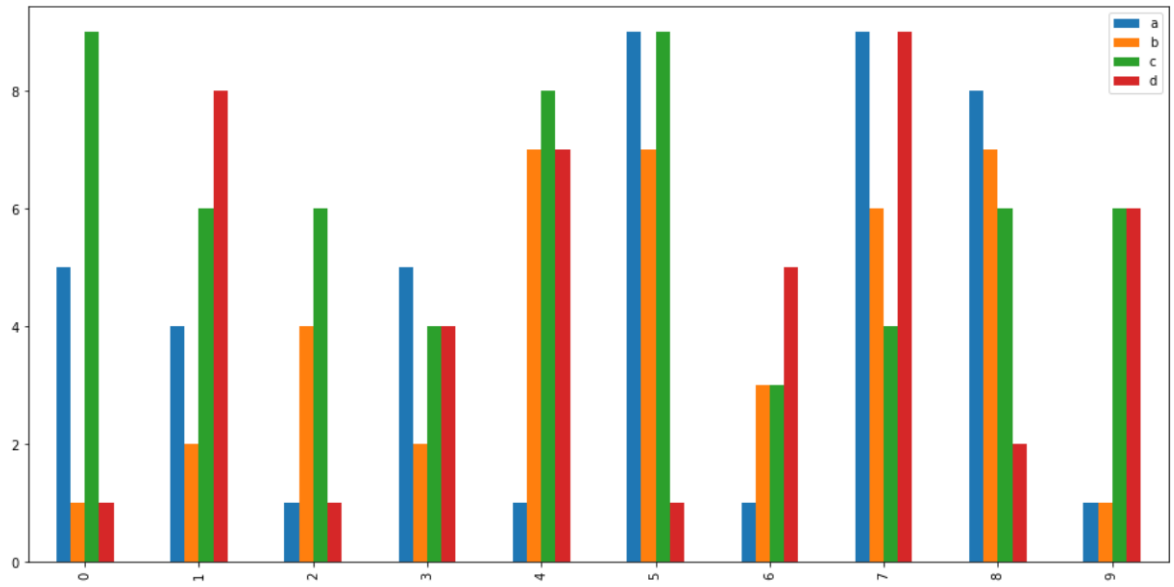
bar

```
df_random = pd.DataFrame(np.random.randint(1, 10, size=(10, 4)),
                         columns=["a", "b", "c", "d"])
df_random
```

	a	b	c	d
0	5	1	9	1
1	4	2	6	8
2	1	4	6	1
3	5	2	4	4
4	1	7	8	7
5	9	7	9	1
6	1	3	3	5
7	9	6	4	9
8	8	7	6	2
9	1	1	6	6

حالا روی کل دیتافریم، تابع bar رو اعمال کردم که چون 4 ستون a, b, c, d داریم، در نتیجه تو هر بخش 4 تا میله مانند وجود داره. میله آبی به ویژگی a، میله زرد به ویژگی b، میله سبز به ویژگی c و میله قرمز هم به ویژگی d توی دیتافریم اشاره میکنه. پس پلات bar هم تقریبا چیزی شبیه پلات histogram هستش.

```
df_random.plot.bar(figsize=(16, 8));
```



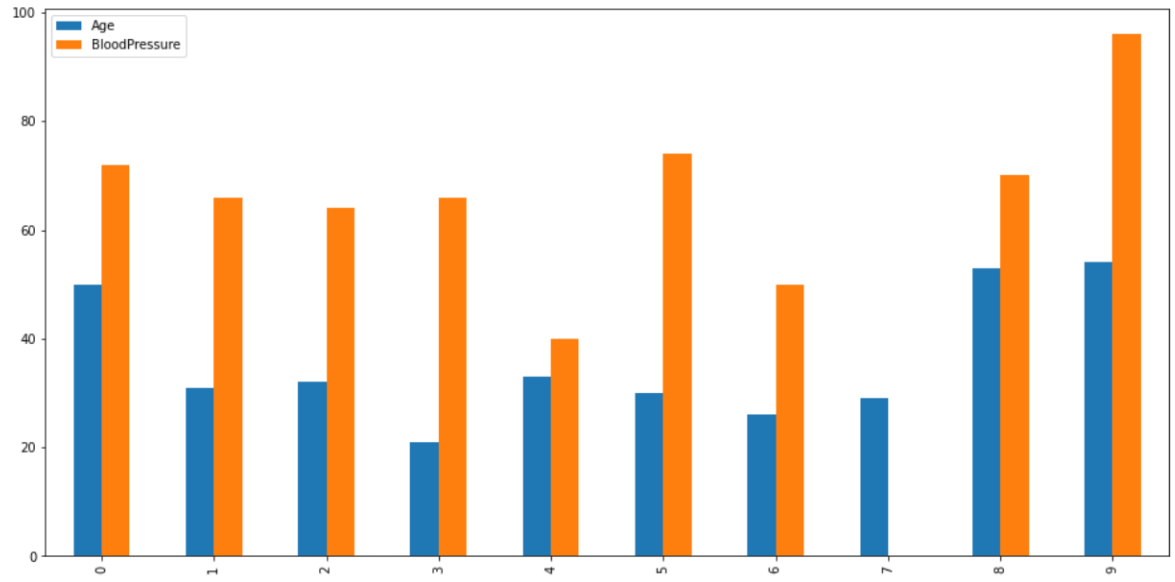
بزارید به کمک این پلات، یه بررسی کوچولویی روی دیتاست افراد دیابتی هم داشته باشیم. اینجا میخوایم پلات bar رو بر اساس دو ویژگی Age و BloodPressure رسم کنیم که قبل از رسم، من 10 تا نمونه اولیه رو توی دیتافریم جدید به اسم df ذخیره و اونو چاپ کردم که ببینیم.

```
df = df_Diabetes[['Age', 'BloodPressure']]
df[0:10]
```

	Age	BloodPressure
0	50	72
1	31	66
2	32	64
3	21	66
4	33	40
5	30	74
6	26	50
7	29	0
8	53	70
9	54	96

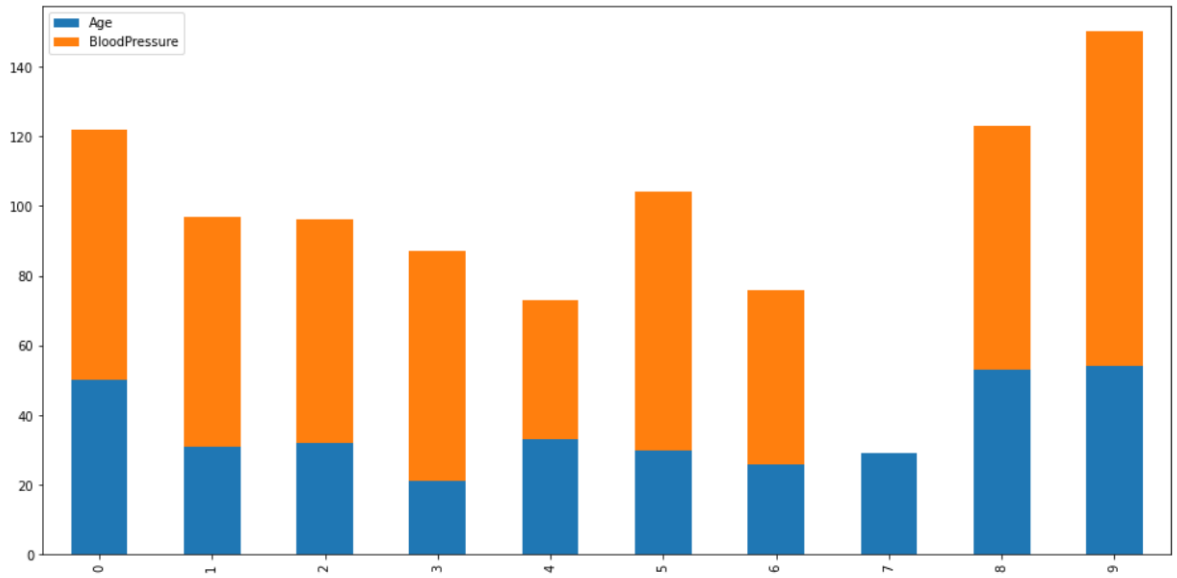
خب حلا پلات bar این دو ویژگی به صورت زیر رسم میشه. پس من اینجا اومدم 10 تا از افراد داخل این دیتاست رو انتخاب کردم و ویژگی های سن و فشار خونشون رو یکجا میبینم. ویژگی سن با رنگ آبی و ویژگی فشار خون با رنگ نارنجی مشخص شده.

```
df[0:10].plot.bar(figsize=(16, 8));
```



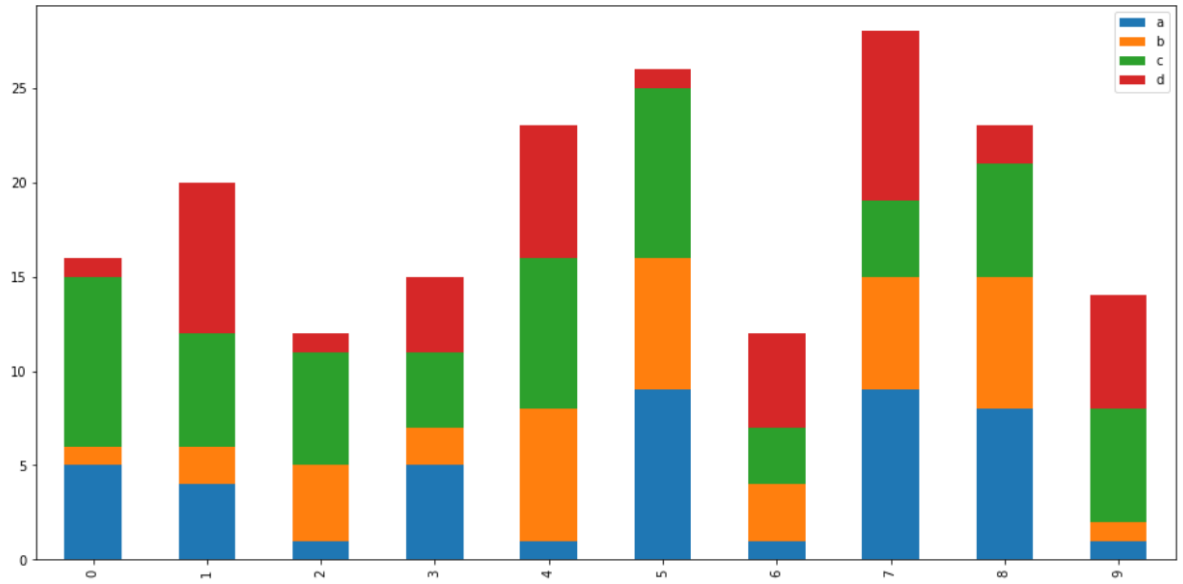
شاید شما نخواید که این دو ویژگی در کنار هم باشند، مثلا دوست داشته باشید که میله ها روی هم بیفتن !!! برای این کار فقط کافیه که stacked رو برابر True قرار بدید، به همین سادگی و البته خوشمزگی.

```
df[0:10].plot.bar(figsize=(16, 8), stacked=True);
```



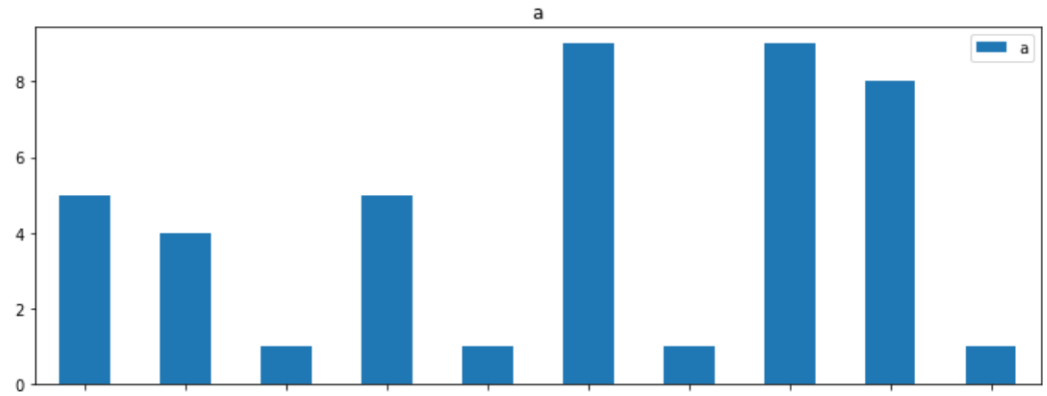
دیتاستی که با اعداد تصادفی پر کردیم و اسمشو df_random گذاشتیم یادتون هست ؟ آره حاجی چطور مگه؟ هیچی خواستم بگم که پلات bar این دیتاست هم به صورت زیره که خب اینجا هم stacked برابر true هست تا همه رو هم بیفتن (استغفر ا...). رنگ های آبی، نارنجی، سبز و قرمز به ترتیب به ویژگی های a و b و c و d اشاره میکنن.

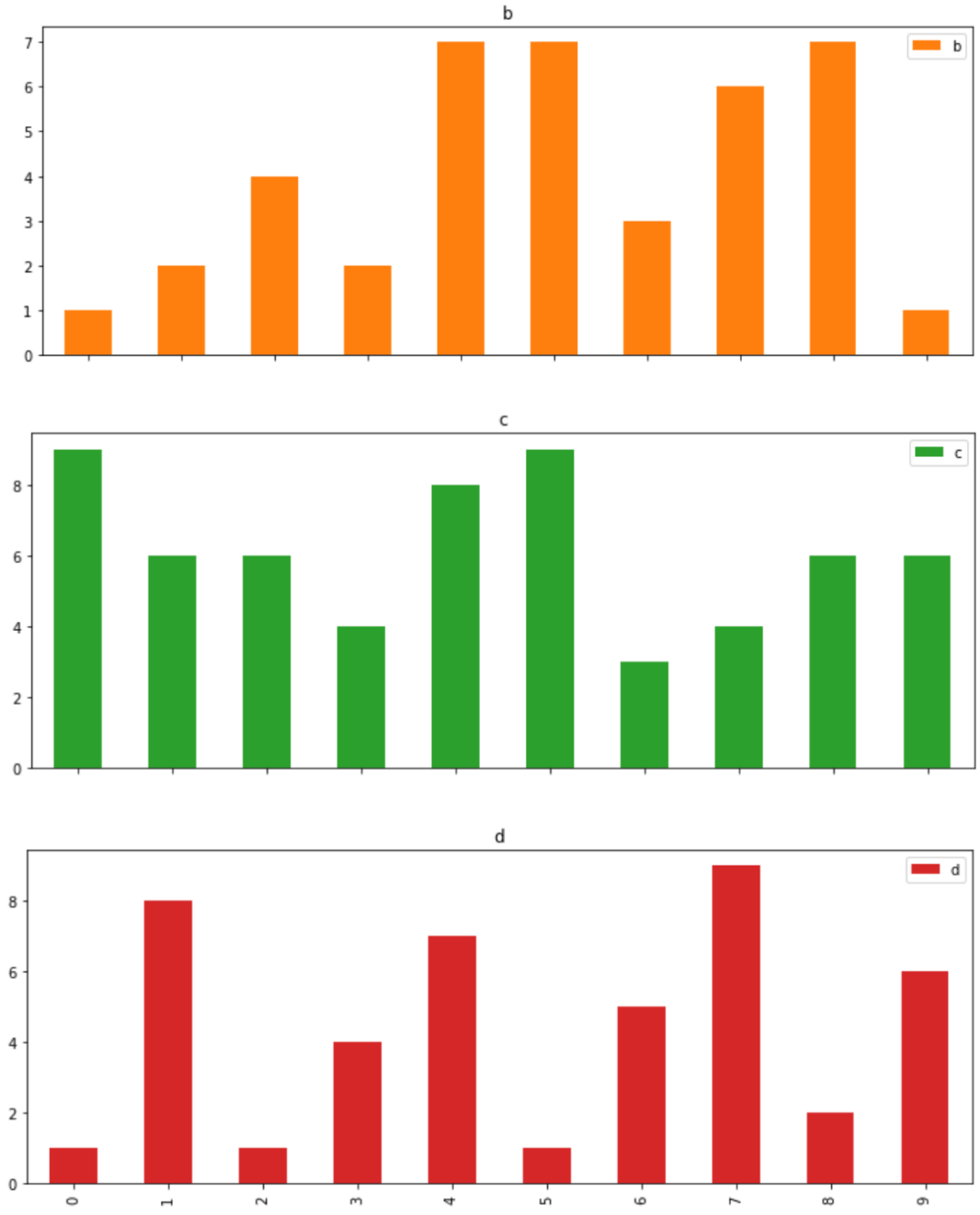
```
df_random.plot.bar(figsize=(16, 8), stacked=True);
```



حالا اگر خواستیم تک تک این میله ها رو توی شکل های جداگانه ببینیم فقط کافیه که subplots رو برابر True کنیم.

```
df_random.plot.bar(figsize=(12, 20),  subplots=True);
```

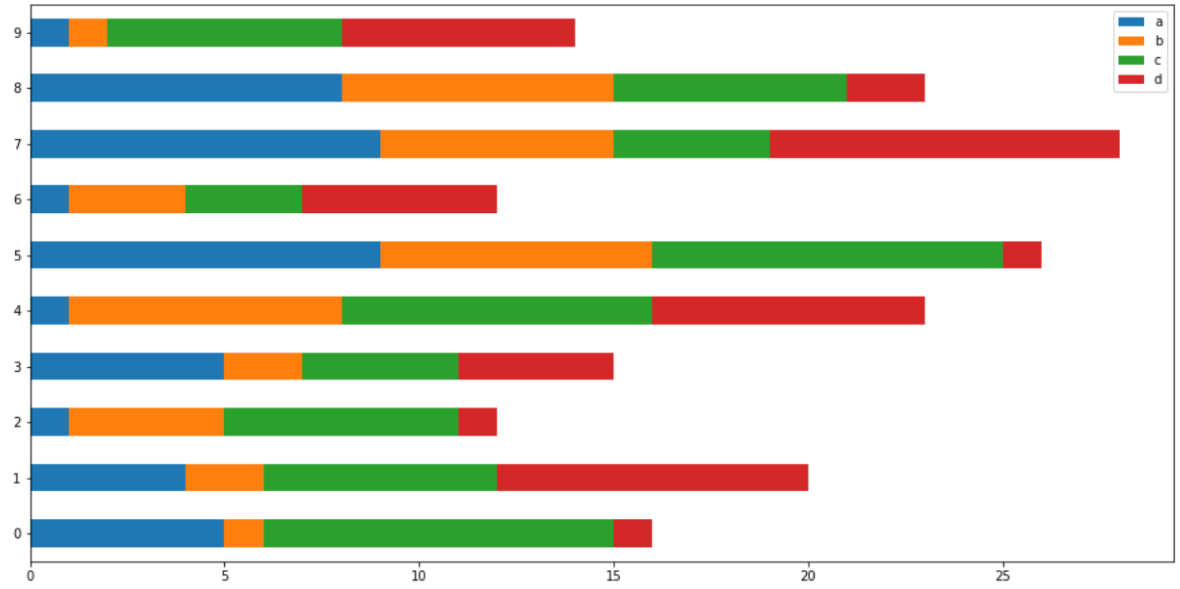




چارت بعدی یعنی barh دقیقا شبیه همین چارت bar است اما خب به صورت افقی یا horizontal رسم میشه. بیا صفحه بعد تا خودت با چشمای خودت ببینی.

barh

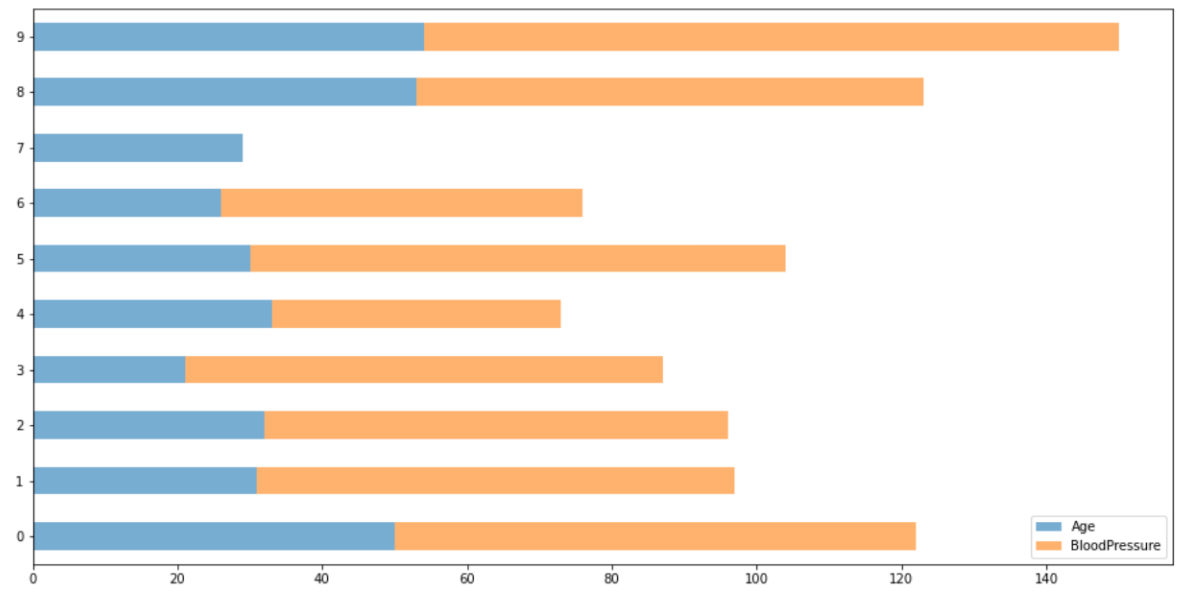
```
df_random.plot.barh(figsize=(16, 8), stacked=True);
```



من دیگه زیاد توضیح نمیدم چون دقیقا شبیه قبله اما بزارید alpha رو بگم که برابر 0.6 گذاشتم.

با این ویژگی میشه یه حالت محو شدگی رو ایجاد کرد که هر چقدر این عدد بزرگ تر باشه، محو شدگی شکل هم بیشتره. نکته مهم اینجاست که از این alpha میشه توی همه پلات ها استفاده کرد و فقط مختص به پلات barh نیست.

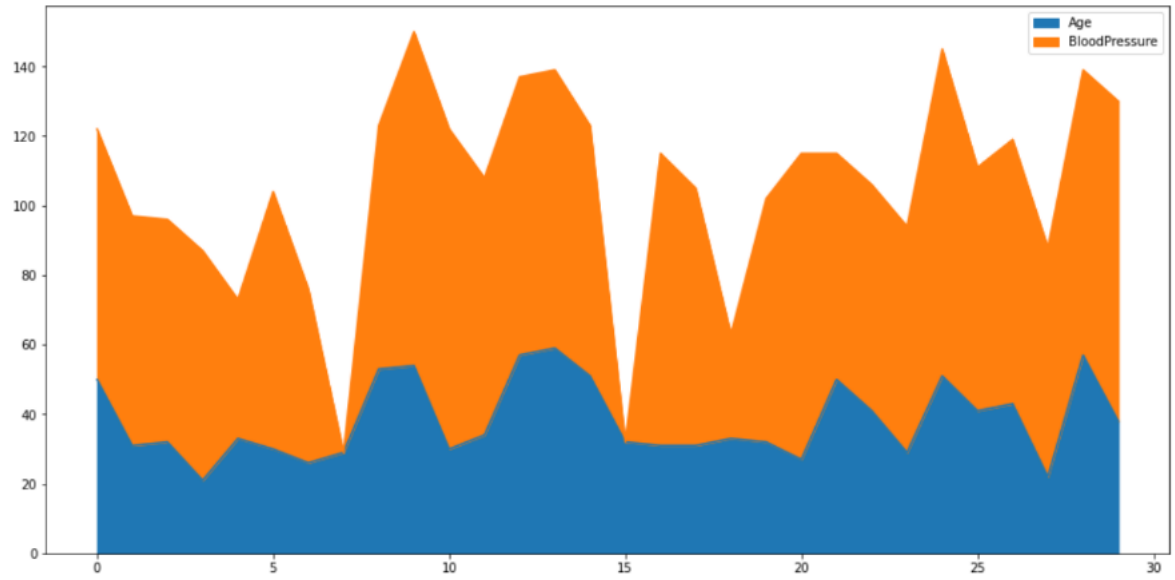
```
df[0:10].plot.barh(figsize=(16, 8), stacked=True, alpha=0.6);
```



تا اینجا که سخت نبود و میبینیم که با نوشتن یه خط کد میشه چارت های خفنی ایجاد کرد. چارت بعدی area هستش که تو سلول زیر اومدم ویژگی های سن و فشار خون 30 نفر اول رو انتخاب و به کمک چارت area بصری سازی شون کردم. سن با رنگ آبی و فشار خون با رنگ نارنجی اما دقت کن که فشار خون روی سن قرار گرفته و از نقطه صفر شروع نشده. (تو این آموزش چقدر "رو هم" داشتیم)

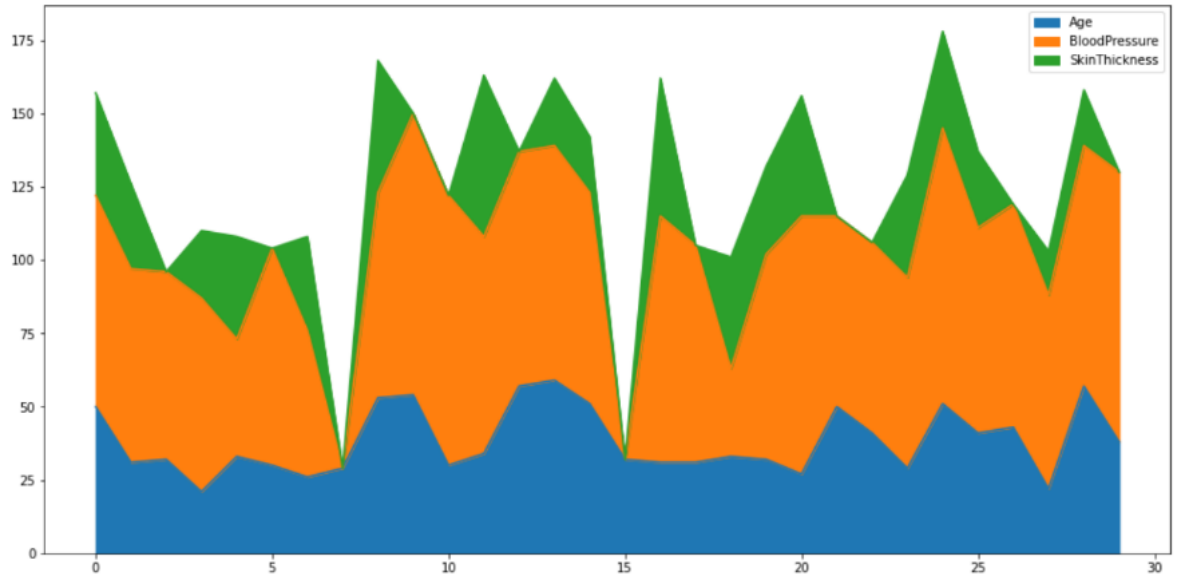
area

```
df_Diabetes[['Age', 'BloodPressure']][0:30].plot.area(figsize=(16, 8));
```



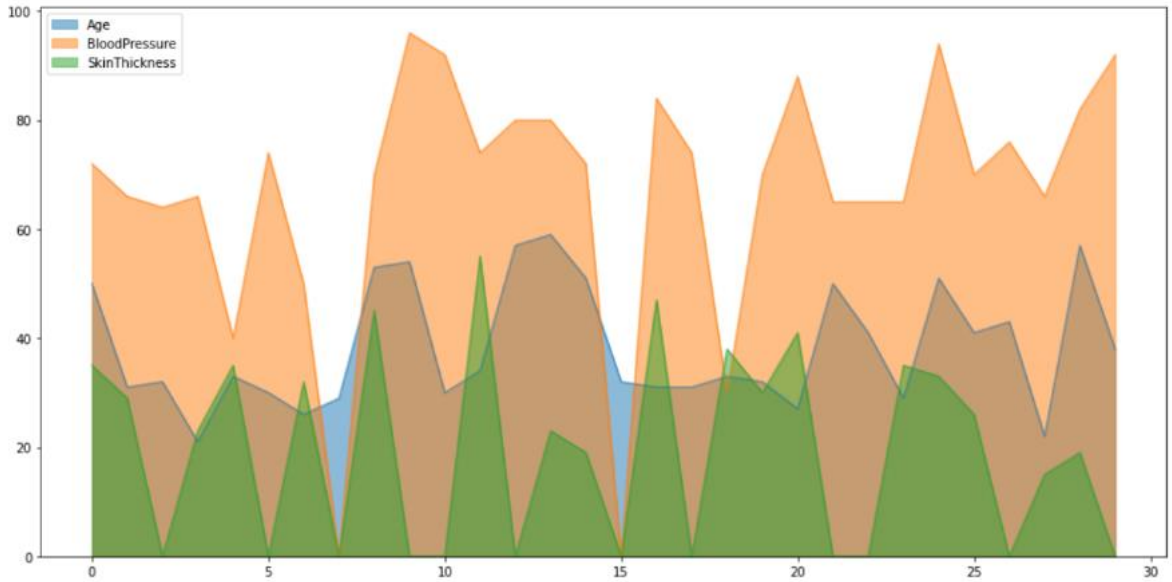
توی کد زیر بر اساس سه ویژگی Age و BloodPressure و SkinThickness مطعلق به 30 نفر اول رو با پلات area رسم کردم.

```
df_Diabetes[['Age', 'BloodPressure', 'SkinThickness']][0:30].plot.area(
                                                        figsize=(16, 8));
```



شاید به دلیل اعتقاداتی که داشته باشید نخواید این ویژگی ها رو هم دیگه بیفتند، برای این کار کافیه که stacked رو برابر False در نظر بگیرید.

```
df_Diabetes[['Age', 'BloodPressure', 'SkinThickness']][0:30].plot.area(
                                                        figsize=(16, 8),
                                                        stacked=False);
```



بدون لحظه ای درنگ بریم سراغ پلات بعدی به اسم hexbin. البته قبلش من یه دیتا فریم ایجاد کردم شامل دو ستون که توی هر ستون 26 تا عدد ذخیره کردم.

hexbin

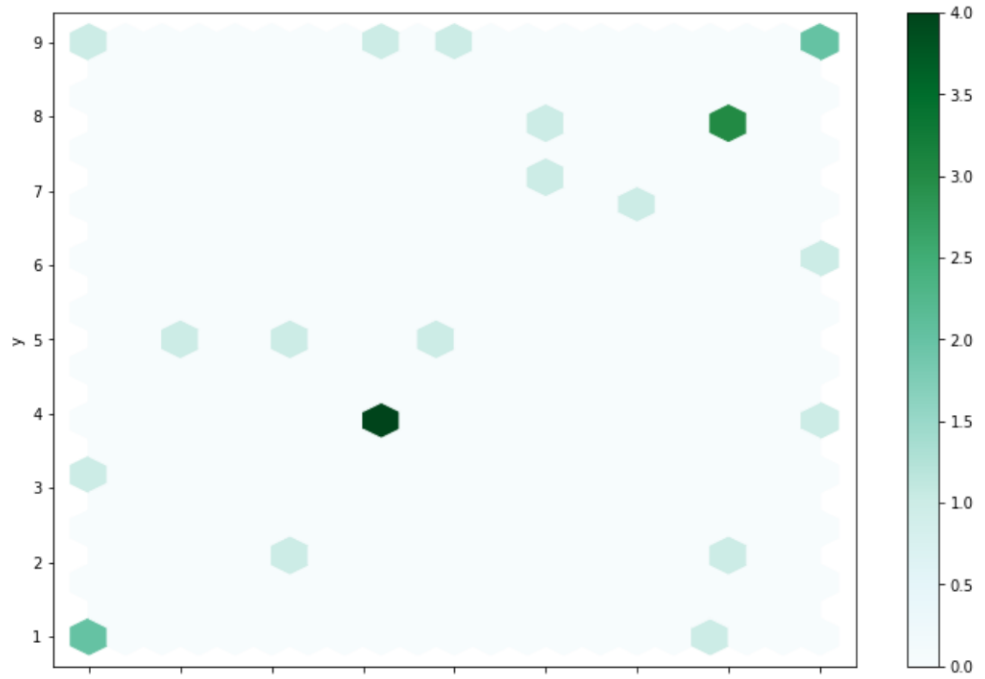
```
np1 = np.array([4, 2, 4, 8, 1, 9, 6, 8, 5, 6, 8, 4, 1, 1, 3, 8, 4, 5, 9, 9,
                3, 7, 9, 8, 1, 4])

np2 = np.array([4, 5, 4, 8, 3, 6, 8, 2, 9, 7, 1, 4, 9, 1, 5, 8, 9, 5, 4, 9,
                2, 7, 9, 8, 1, 4])

my_df = pd.DataFrame({'x': np1,
                      'y': np2})
```

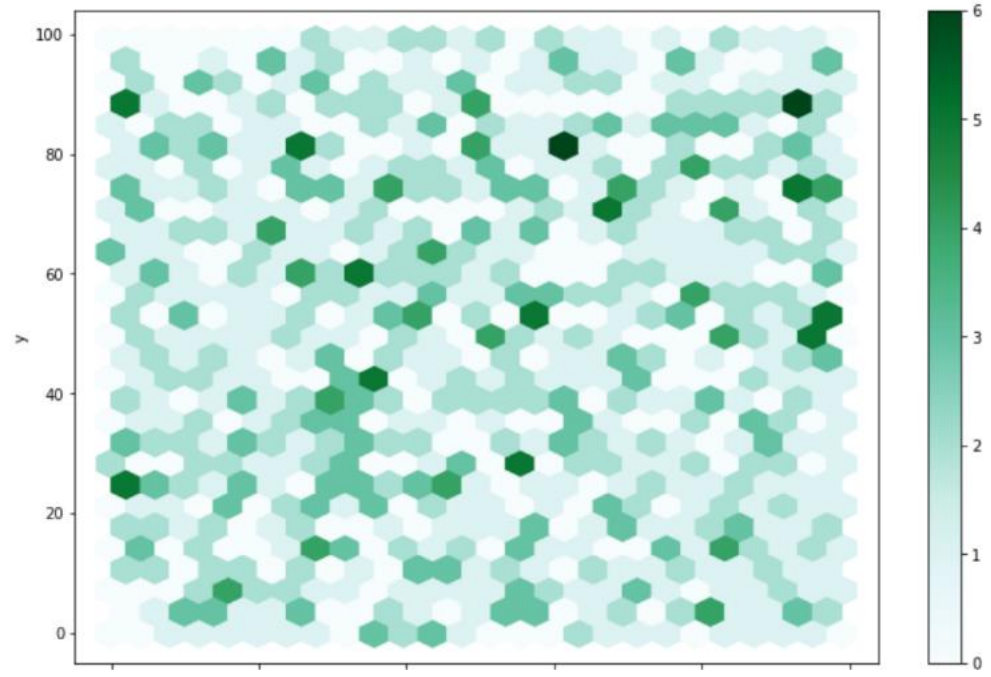
حالا میخوایم چارت hexbin رو روی این دیتاست اعمال کنیم. به کمک hexbin می تونیم داده ها رو در دو بعد یا بر اساس دو ویژگی در صفحه رسم کنیم. دقت کنید که هر داده ای که تعدادش بیشتر باشه با سبز پر رنگ نشون داده میشه و هر داده ای هم که مقدارش کم باشه با سبز کم رنگ در صفحه نشون داده میشه. مثلا توی همین دیتافریم my_df داده ای که مقدار x و y اش هر دو 4 هستند، 4 بار تکرار شدند پس تو صفحه، مختصات 4 و 4 با سبز پر رنگ نشون داده شده. اما مثلا سطری که x اش 1 و y اش 3 است تو مختصات 1 و 3 با سبز خیلی کم رنگ نشون داده شده. پس با شدت رنگ ها میشه فهمید که تعداد هر کدوم از داده ها چقدر هستند.

```
my_df.plot.hexbin(x='x', y='y', gridsize=20, figsize=(12, 8));
```



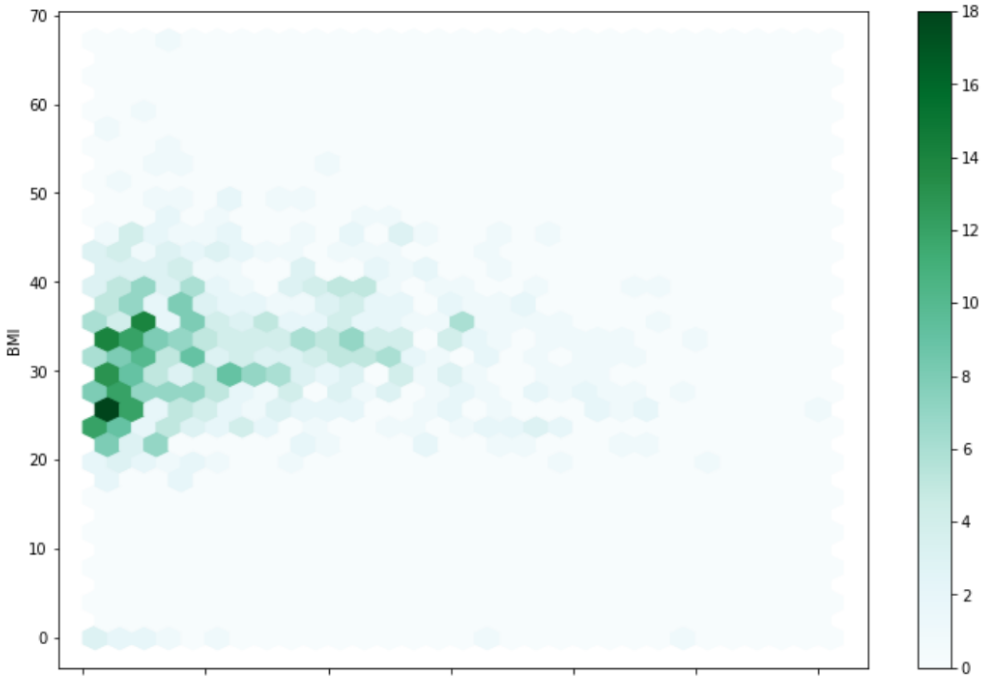
تو مثال زیر یه دیتافریم به اسم df ایجاد کردم شامل 1000 سطر و دو ستون x و y هستش. تو هر کدوم از ستون های x و y، 1000 تا عدد تصادفی بین 0 تا 99 ریختم و hexbin شو رسم کردم.

```
df = pd.DataFrame({'x': np.random.randint(0, 100, 1000),
                   'y': np.random.randint(0, 100, 1000)})
df.plot.hexbin(x='x', y='y', gridsize=25, figsize=(12, 8));
```



تو مثال زیر ویژگی های سن و فشار خون رو به ترتیب روی محور های x و y نشون دادم.

```
df_Diabetes.plot.hexbin(x='Age', y='BMI', gridsize=30, figsize=(12, 8));
```



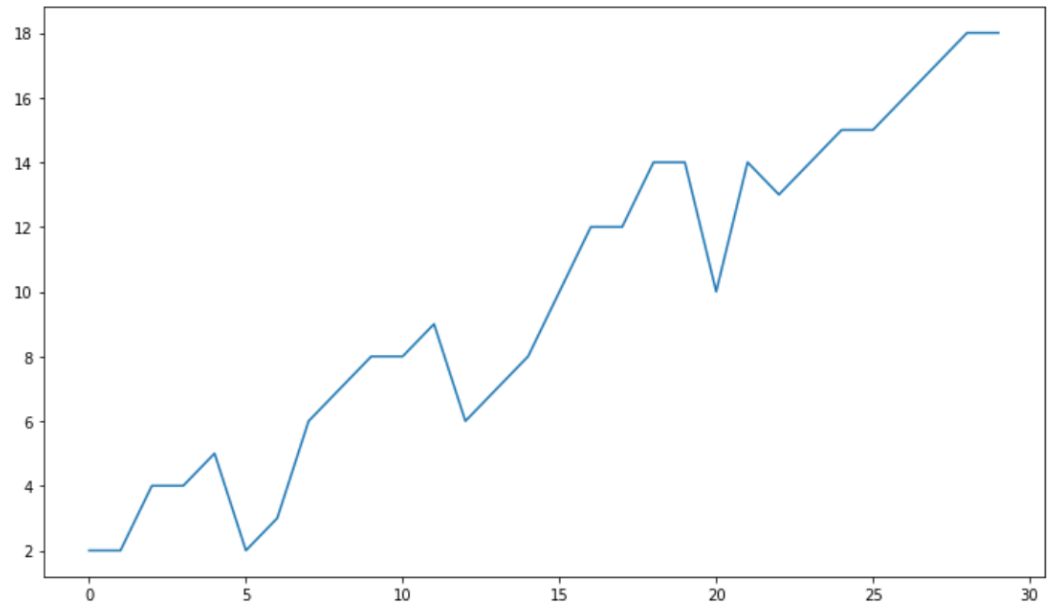
چارت بعدی line هستش که خیلی مهمه و البته خیلی هم ساده. با این چارت میشه داده ها رو بر اساس دو ویژگی در صفحه رسم کرد. برای نشون دادن میزان رشد یا روند یک فعالیت، نوسانات بازارهای مالی و در کل هر فرآیندی که یا در طول زمان یا در طول واحدهای مختلف در حال تغیر باشه، میشه از پلات line استفاده کرد. تو سلول زیر یه دیتافریم از اطلاعات یک دانشجو ایجاد کردم که ستون Day روز های ماه رو نشون میده و مقدارهاش از 1 تا 29 هستش. ستون hour نشون دهنده تعداد ساعاتی است که این دانشجو در هر روز مطالعه داشته. مثلا این دانشجو روز اول 2 ساعت، روز دوم 2 ساعت، روز سوم 4 ساعت و ... درس خونده.

line

```
df = pd.DataFrame({
    'Day': np.linspace(1, 30, 30),
    'hour': [2, 2, 4, 4, 5, 2, 3, 6, 7, 8, 8, 9, 6, 7, 8, 10, 12,
             12, 14, 14, 10, 14, 13, 14, 15, 15, 16, 17, 18, 18]
})
df
```

من میخوام بدونم که روند مطالعه این دانشجو در طول این یک ماه چقدر بوده پس کافیه که خیلی ساده پلات line رو رسم کنم تا شکلی شبیه زیر داشته باشم. البته اگر فقط روی ویژگی hour این پلات رو رسم کنم هم کافیه. طبق این شکل میفهمم دانشجو قصه ما، دانشجو زرنگ و درس خونی هستش که ساعات مطالعه درس هاش به صورت صعودی است.

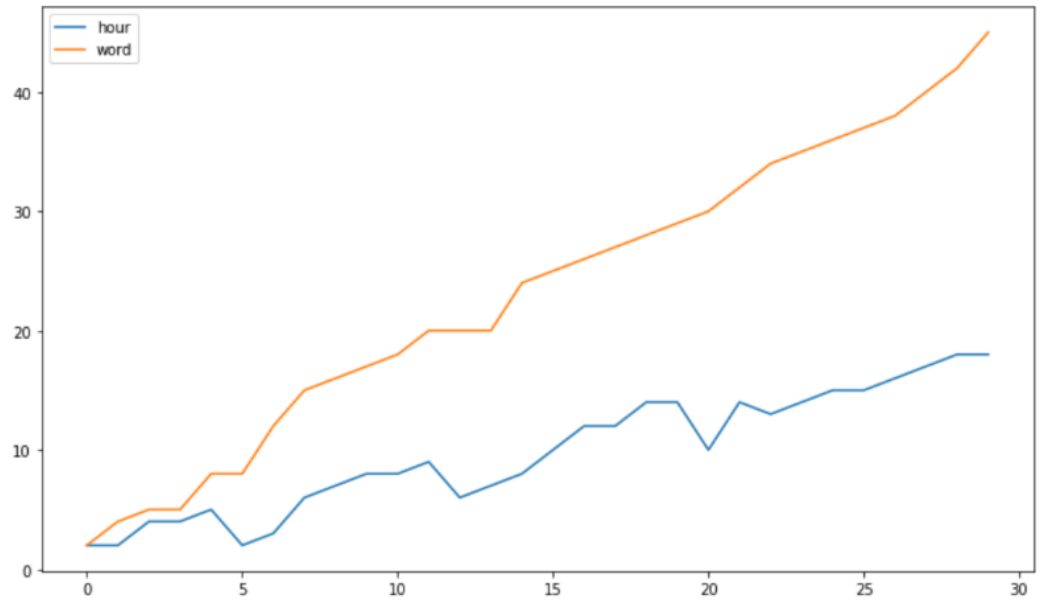
```
df['hour'].plot.line( figsize=(12, 7));
```



دیتاست بعدی هم اطلاعات یک ماه یک داشنجو هست. ستون hour نشون دهنده تعداد ساعات مطالعه یک درس و ستون word هم نشون دهنده تعداد لغات انگلیسی است که این دانشجو در طول هر روز خونده. وقتی که تابع line رو رو این دیتاست اعمال کنم، به ازای هر ستون یک نمودار

line برای من رسم میشه. ویژگی hour با رنگ آبی و ویژگی word با رنگ نارنجی نشون داده شده و اینجا هم میبینیم که تعداد ساعت مطالعه روزانه و تعداد لغات انگلیسی که این دانشجو در طول یک ماه خونده صعودی است.

```
df2 = pd.DataFrame({
    'hour': [2, 2, 4, 4, 5, 2, 3, 6, 7, 8, 8, 9, 6, 7, 8, 10, 12,
             12, 14, 14, 10, 14, 13, 14, 15, 15, 16, 17, 18, 18],
    'word': [2, 4, 5, 5, 8, 8, 12, 15, 16, 17, 18, 20, 20, 20, 24,
             25, 26, 27, 28, 29, 30, 32, 34, 35, 36, 37, 38, 40, 42, 45]
})
df2.plot.line( figsize=(12, 7));
```

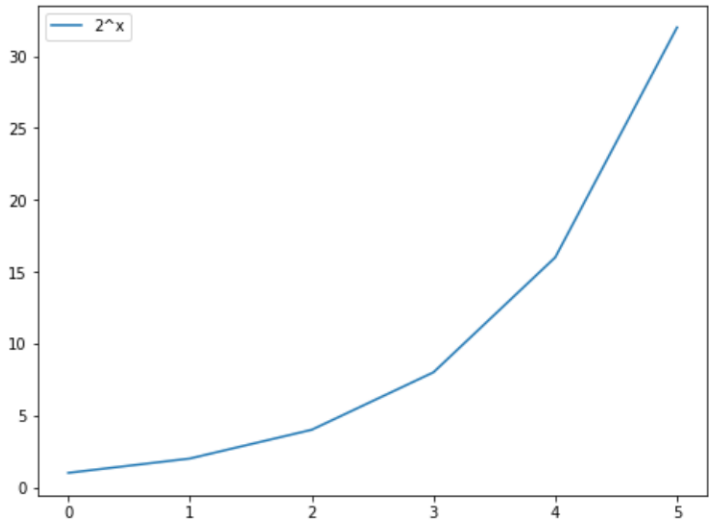


تو مثال های بعدی میخوایم کمی ریاضی چاشنی کار کنیم چجوری ؟ بهتون میگم. تو مثال زیر من نمودار 2 به توان اعداد 0 تا 5 رو رسم کردم. خب برای انجام این کار خیلی ساده عدد 2 به توان 0 تا 5 رو حساب و به عنوان تنها ویژگی تو دیتافریم df قرار دادم. سپس تابع line رو رسم کردم و همون طور که میبنید این نمودار به صورت نمایی رشد میکنه. درسته دیگه شد توان نمایی هستش.

```
x = np.linspace(0, 5, 6)
y = np.power(2, x)
df = pd.DataFrame(y, columns=['2^x'])
df
```

	2^x
0	1.0
1	2.0
2	4.0
3	8.0
4	16.0
5	32.0

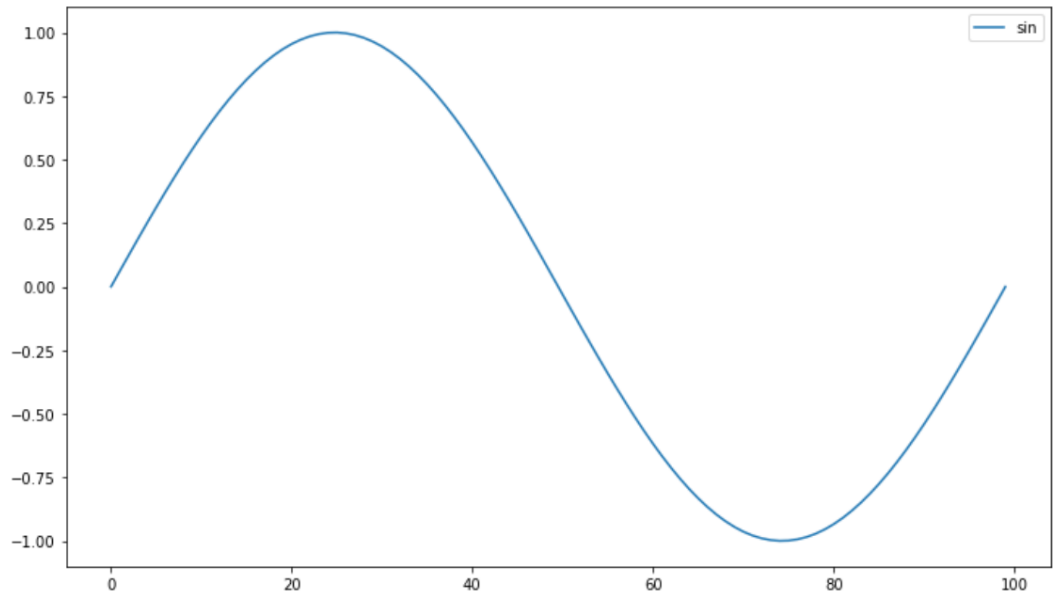
```
df.plot.line( figsize=(8, 6));
```



مثال بعدی هم خیلی باحاله، اینجا اول اومدم بین 0 تا عدد 2پی، 100 عدد ایجاد و سپس سینوس این 100 عدد رو محاسبه و در نهایت تو صفحه رسم شون کردم. پس نموارد صفحه بعد، شکل سینوس 0 تا عدد 2پی هستش. حالا دیگه خیلی ساده میتونید نمودار هر تابع ریاضی که دوست داشتید رو به همین سادگی و خوشمزگی بکشید.

```
x = np.linspace(0, 2*np.pi, 100)
y = np.sin(x)
df2 = pd.DataFrame(y, columns=['sin'])
# df2
```

```
df2.plot.line(figsize=(12, 7));
```



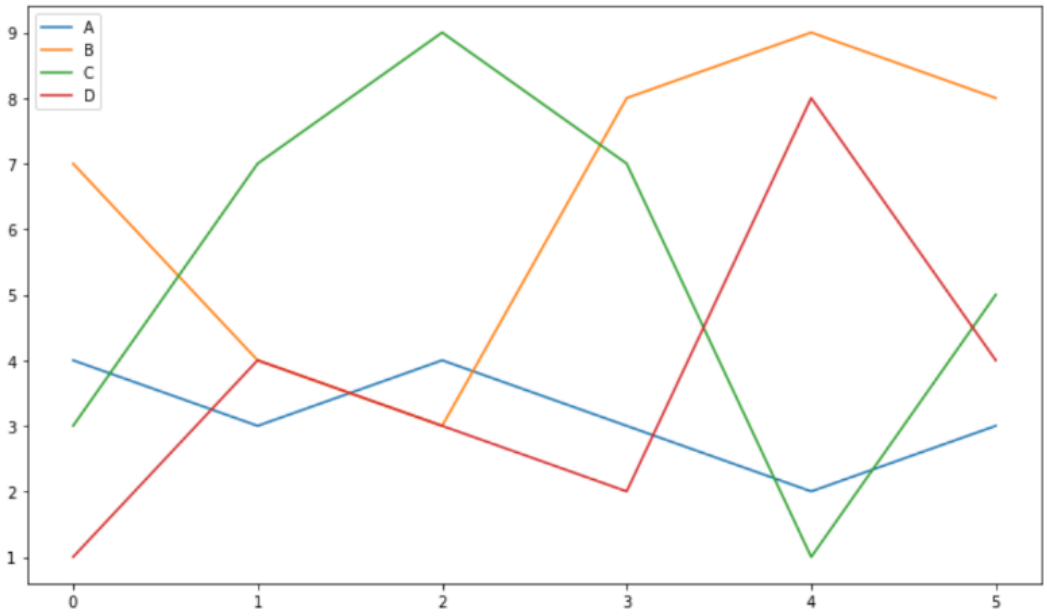
تو مثال بعدی یه دیتا فریم شامل 4 ویژگی با اعداد تصادفی بین 1 تا 9 پر کردم.

```
df3 = pd.DataFrame(np.random.randint(1, 10, (6,4)),
                   columns=('A', 'B', 'C', 'D'))
df3
```

	A	B	C	D
0	4	7	3	1
1	3	4	7	4
2	4	3	9	3
3	3	8	7	2
4	2	9	1	8
5	3	8	5	4

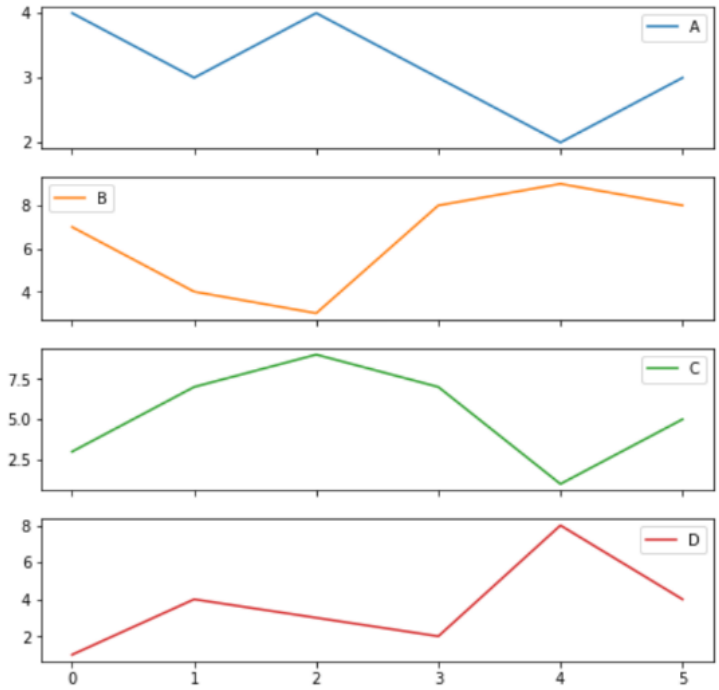
حالا اگر پلات line رو روی این دیتافریم ایجاد کنم، به ازای هر ستون یک نمودار رسم میشه.

```
df3.plot.line(figsize=(12, 7));
```



اگر هم خواستید هر نمودار که توصیف کننده یه ستون یا ویژگی از دیتاست است رو توی یه شکل جداگانه ببینید، فقط کافیه که subplots رو True کنید.

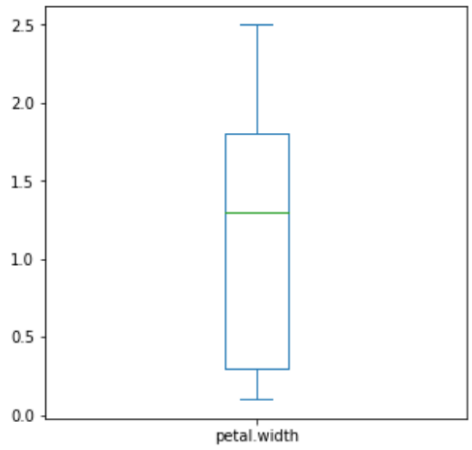
```
df3.plot.line(subplots=True, figsize=(8, 8));
```



خب بچه ها میرسیم به آخرین پلات یعنی box. با این پلات میشه اطلاعات خوبی رو درباره داده ها بدست آورد. توی سلول زیر یک شکل جعبه مانند میبینید که اطلاعات مهمی رو تو خودش داره. من روی ویژگی petal.width از دیتاست iris این پلات رو اعمال کردم. حالا ین خط ها و جعبه داخل شکل چی میگه ؟ ببینید اون پایین پایین که خط روی عدد 0.1 است در واقع همون min یا کم ترین مقدار petal.width رو نشون میده و بالاترش یعنی عدد 0.3 به چارک اول اشاره میکنه. خطی که وسط مستطیل است داره به چارک دوم که 1.3 است اشاره می کنه و خط بالایی اش به چارک سوم که مقدارش 1.8 است اشاره میکنه. اون خط بالا بالا هم max یا ماکزیمم این ویژگی است که مقداش 2.5 است.

box

```
df_iris['petal.width'].plot.box( figsize=(5, 5) );
```



حالا همین اطلاعات هم به سادگی و به کمک تابع describe میشه بدست آورد و لی خب با تابع box یک نمای بصری از این داستان داریم. بیا صفحه بعد تا تابع describe رو ببینی.

```
df_iris['petal.width'].describe()
```

```
count    150.000000
mean       1.199333
std        0.762238
min        0.100000
25%        0.300000
50%        1.300000
75%        1.800000
max        2.500000
Name: petal.width, dtype: float64
```

```
my_name = "Ali Nazarizadeh"
whatsapp = "09331367233"
print(my_name + " : " + whatsapp)
```

```
Ali Nazarizadeh : 09331367233
```

https://www.linkedin.com/in/ali-nazarizadeh/

این یکی از 20 جلسه دوره تحلیل داده است که لیست کامل این دوره تو لینک زیر در دسترسه :

https://bigdataworld.ir/product/pandas_numpy_matplotlib/