plt.figure( figsize=(12,4) ) # Grafiğin “En”i ve “Boy”u

plt.title( ) #

plt.xlabel( ) #

plt.ylabel( ) #

Not: plt.plot() kullanarak, 2 farklı veriyi aynı grafikte göstermek için, x ve y eksenlerinden en az birisi ortak olmaldır.

plt.plot (

data.Tarih,

data.Enflasyon,

color = “red”, # Grafiğin rengi

linewidth = 3, # Grafik çizgisinin genişliği

linestyle = “--”, # Grafik stilini belirler: “-”, “-.”, “:”

marker = “o”, # Grafikte kırılım noktasının şekli (marker)

markersize = 12, # Marker boyutu

markerfacecolor = “yellow”, # Marker rengi

markeredgewidth = 5, # Marker kalınlığı

markeredgecolor = “blue”, # Marker kenar rengi

alpha = 0.3 # Markerin saydamlıını belirler

)

plt.legend (

loc = 5, # Legend’in lokasyonunu belirler. 0-10 arasında (dahil) değerler alır.

bbox\_to\_anchor = (0.5, 0. , 0.5, 0.5), # Legend’İn manuel lokasyonu

fontsize = ”xx-large”, # Leged’in boyutu, “xx-small, x-small, small, medium, large, x-large, xx-large değerlerini alır.

numpoints = 3, # Legendlerdeki şekillerin sayısı

markerscale = 2, # Legend’in boyutu

markerfirst = False, # Legend’i başa, şekli sona alır.

shadow = True, # Legend’in etrafındaki çizgilere gölge verir.

framealpha = 0.2, # Legend çizgilerinin saydamlığını ayarlar.

facecolor = “r”, # Legend’in arka planına renk verir.

edgecolor = “r”, # Legend’in etrafındaki çizgiyi renklendirir.

mode = “Expand”, # Legend’in kutu boyutu. ”None” ya da “Expand”

title = “Enflasyon”, # Legend’in başlığı

title\_fontsize = 16, # Legend başlığının boyutu

borderpad = 0.8, # Legend çerçevesinin boyutu

handlelength = 6, # Legend kollarının uzunluğu

handletextpad = 5, # Legend noktası ile yazısı arasındaki boşluk

borderaxespad = 2 # Legend ile grafik çizgileri arasındaki boşluk

)

plt.grid (

alpha = 0.2, #Grid çizgilerinin saydamlığını ayarlar.0-1 arasında değer alır. 1'e yaklaştıkça saydamlık artar.0'a yaklaştıkça azalır.

linestyle = “--”, #Grid çizgilerinin biçimini belirler.'-', '--', '-.', ':', '',

linewidth = 3, #Grid çubuklarının kalınlığını belirler.

color = “blue” #Grid çizgilerinin rengini belirler.

dash\_capstyle = “butt”,

dash\_joinstyle = “miter”,

drawstyle = “steps\_mid”,

fillstyle = “bottom”,

…

…

)

Not: <https://matplotlib.org/3.1.0/api/_as_gen/matplotlib.pyplot.grid.html> , grid fonksiyonu dökümantasyonu.

plt.text (

x = 7,

y = 18,

s = “Bu yılda bir kriz olmuştur.”,

fondict = {“color”:b, “fondict”:16}

withdash = True, True ve False değerler alır. Eklediğimiz yazının önüne çizgi çekmemizi sağlar.

dashlength = 55, Ekledğimiz yazının uzunluğunu belirler. Değişiklik yapabilmek için withdash değerinin True olması gerekir.

dashrotation = 45, Çizginin açısını belirler.

dashdirection = 45, 0 ve 1 değerini alır. Yazıya ekleyeceğimiz çizginin yazının sağında mı solunda mı olması gerektiğini ayarlar. 0 sağında 1 ise solunda çizgiyi oluşturur.

dashpad = Metin ile çubuk arasındaki boşluğu ayarlar. İnteger değerler alır.

dashpush = Çizgi ve yazıyı yukarı veya aşağıya kaydırır. İnteger değerler alır.

alpha =

backgroundcolor = Yazının arka planına renk ekler.

fontfamily = Yazının fontunu ayarlar. ‘serif’, ‘sans-serif’, ‘cursive’, ‘fantasy’, ‘monospace’

style = Yazı tipini ayarlar. İtalik, normal. “Normal”, “Italic”, “Oblique”

weight = Yazının kalınlığını ayarlar. 0-1000 arası değer alır ya da 'ultralight', 'light', 'normal', 'regular', 'book', 'medium', 'roman', 'semibold', 'demibold', 'demi', 'bold', 'heavy', 'extra bold', 'black' değerlerini alır.

size = Yazının büyüklüğünü ayarlar. 'xx-small', 'x-small', 'small', 'medium', 'large', 'x-large', 'xx-large' değerlerini alır.

fontsize = Ya da fontsize şeklinde kullanılarak integer değer alır. “fontsize=12” gibi

)

plt.savefig (

facecolor = “green”, # Grafiğin arka planını renklendirir.

edgecolor = “red”, # Grafiğin kenar rengini ayarlar.

transparent = True, # Grafiğin saydamlığını ayarlar. True ya da False değer alır.

dpi = 600, # Grafiğin çözünürlüğünü arttırmanızı sağlar. İnteger değerler alır.

quality = 95, # Grafiğin kalitesini ayarlamamızı sağlar. 1-95 arası değer alması tavsiye edilir. 95 en iyi değerdir.

optimize = True ya da False değer alır.

)

plt.scatter (

s = 3, #Grafikteki yuvarlakların boyutunu belirler.

c = “r”, #Grafikteki yuvarlakların rengini belirler.

edgecolors = “b”, #Yuvarlağın dışındaki çizginin rengini belirler.

linewidths = 3, #Yuvarlağın dışındaki çizginin kalınlığını belirler.

alpha = “0.5, #Yuvarlağın saydamlığını belirler. 0-1 arasında değer alır.

cmap =

)

plt.step (

data.Tarih,

data.Enflasyon,

marker = “\*”, # Konulacak şekil. “\*”, “o”, vs

c = “r”, # Grafik rengi

linewidth = 3, # Grafik çizgisinin genişliği

where = Küçük yuvarlakların yerleşecekleri yeri belirler. 'pre', 'post', 'mid' değerlerini alır. pre sağa, mid ortaya, post en sola yerleştirir.

)

plt.hist (

data.issizlik,

bins = 12, #Histogramdaki kutu sayısını belirler.

histtype = “barstacked”, #Histogramın tipi. 'bar', 'barstacked', 'step', 'stepfilled'

align = “mi d”, #Histogramın çizimini kontrol eder. 'left', 'mid', 'right'

orientation = “Verical”, #Grafiğin yatay ya da dikey çizilmesini belirler. “Vertical”, “Horizontal”

rwidth = 0.3, #Çubukların genişliğini belirler. 0-1 arasında değer alır.

)

plt.bar (

data.Tarih,

data.Enflasyon,

width = 0.5, #Bar kalınlığını belirler.0-1 aralığındaki değerler daha anlamlı olur.

color = “green”, #Bar rengini belirler.

edgecolor = #Barların kenar renklerini belirler.

linewidth = 2, #Bar kenar çizgilerinin kalınlığını belirler.

tick\_label = #Barları x ekseninde isimlendirir.

xerr = #x ve y eksenlerinden barların ortasına noktalar koyar.

ecolor = “black”, #xerr ve yerr lerin rengini belirler.

capsize = #x ve y ekseninde hata çubukları çizmeye yarar.

orientation = “Vertical”, #vertical ve horizontal değerlerini alır.Yatay ve dikey yapmamızı sağlar.

hatch = “o”, #Barların üzerine şekiller oluşturur. '/', '', '|', '-', '+', 'x', 'o', 'O', '.', '\*'

)