

Using 3rd party libraries

3rd party knjižnjice/pakete lahko inštaliramo s pomočjo **pip**, ki je python package manager.

Če imamo pip inštaliran lahko preverimo tako, da v konzolo vpišemo `pip --version`

```
$pip --version
```

```
pip 19.1.1 from C:\Users\Anaconda3\lib\site-packages\pip (python 3.7)
```

Če dobimo napako, moramo pip inštalirati:

WINDOWS: Download [get-pip.py \(https://bootstrap.pypa.io/get-pip.py\)](https://bootstrap.pypa.io/get-pip.py) to a folder on your computer. Open a command prompt and navigate to the folder containing get-pip.py. Run the following command: `python get-pip.py`

Debian(Ubuntu/Kali etc) distribution of linux: `$ sudo apt-get install python-pip`

S pomočjo pip lahko inštaliramo knjižnjice s preprostim ukazom

```
pip install <package_name>
```

```
pip install matplotlib
```

Matplotlib

Je eden izmed najbolj uporabljenih Python paketov za grafično prikazovanje podatkov.

```
pip install matplotlib
```

In [1]:

```
#uvoz knjižnice za matplotlib
import matplotlib.pyplot as plt

%matplotlib inline
##matplotlib notebook
# te dve liniji definirata prikazovanje, če uporabljamo jupyter notebook
# to je syntax-a specifična za jupyter notebook
```

`matplotlib.pyplot` is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

Simple plot

Začeli bomo s preprostim primerom:

izrisali bomo funkcijo sinus in kosinus na isti graf.

Za začetek bomo uporabili raznorazne default vrednosti, za velikost grafa, barvo črt, stil črt, itd., nato pa bomo graf nadgrajevali.

In [28]:

```
import matplotlib.pyplot as plt
import math
#x-os
x = [math.radians(x) for x in range(-180, 180)]

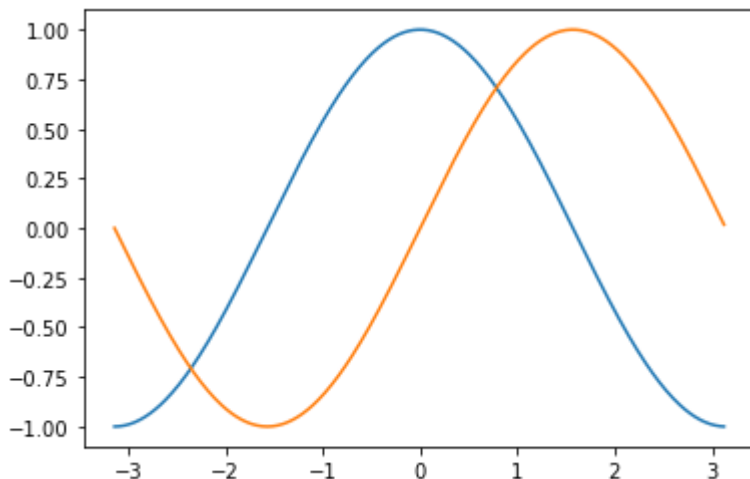
#y-os, potrebujemo vrednosti Cos(x) in sin(x):4.f
cos = [math.cos(i) for i in x]
sin = [math.sin(i) for i in x]

for i, x_ in enumerate(x[:30]):
    print(f"x: {x_:7.4f} \t sin: {sin[i]:7.4f} \t cos: {cos[i]:7.4f}")

plt.plot(x,cos)
plt.plot(x,sin)

plt.show()
```

x: -3.1416	sin: -0.0000	cos: -1.0000
x: -2.6180	sin: -0.0175	cos: -0.9998
x: -2.0944	sin: -0.0349	cos: -0.9994
x: -1.5708	sin: -0.0523	cos: -0.9986
x: -1.0472	sin: -0.0698	cos: -0.9976
x: -0.5236	sin: -0.0872	cos: -0.9962
x: 0.0000	sin: -0.1045	cos: -0.9945
x: 0.5236	sin: -0.1219	cos: -0.9925
x: 1.0472	sin: -0.1392	cos: -0.9903
x: 1.5708	sin: -0.1564	cos: -0.9877
x: 2.0944	sin: -0.1736	cos: -0.9848
x: 2.6180	sin: -0.1908	cos: -0.9816



`plt.plot(x_podatki, y_podatki)` S tem ukazom na naš plot narišemo podatke.

`plt.show()` pokaže naš graf.

Kako Matplotlib izrisuje stvari (Figures, Subplots, Axes, Ticks)

Matplotlib nam bo po defaultu ustvaril `figure`. Po defaultu bo znotraj `figure` ustvaril 1 `subplot`, ki bo zavzel celotno `figuro` in znotraj `subplota` nam bo izrisal graf.

Figure v matplotlib-u pomeni celotno okno. Za naslov imajo `Figure _cifra_`, številčenje pa se začne z 1. Ostali parametri, ki definirajo `figure` so:

- `figsize` - velikost figure v inčih (dolžina, višina)
- `dpi` - resolucija v dots per inch
- `facecolor` - barva risalne podlage
- `edgecolor` - barva obrobe

Subplot

Z uporabo `subplot` lahko našo figuro razdelamo na več razdelkov. Specificiramo število vrstic, število stolpcev in številko `plot-a`.



```
subplot(2,2,1)
```

```
subplot(2,2,2)
```

```
subplot(2,2,3)
```

```
subplot(2,2,4)
```

In [29]:

```
import matplotlib.pyplot as plt
import math

# Create a new figure of size 8x6 points, using 100 dots per inch
plt.figure(figsize=(8,6), dpi=100)

# Create a new subplot from a grid of 1x1
# 2 rows, 2 columns, and the 2nd subplot
a = plt.subplot(2,2,2)

X = [math.radians(x) for x in range(-180, 180)]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]

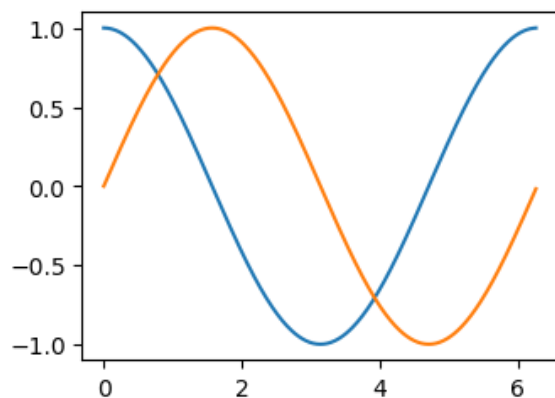
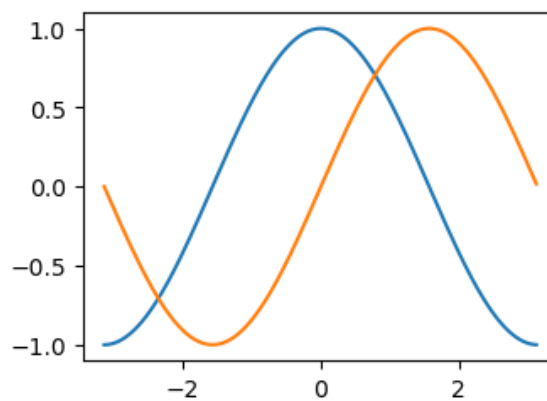
a.plot(X, C)
a.plot(X, S)

b = plt.subplot(2,2,3)

X = [math.radians(x) for x in range(360)]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]

b.plot(X, C)
b.plot(X, S)

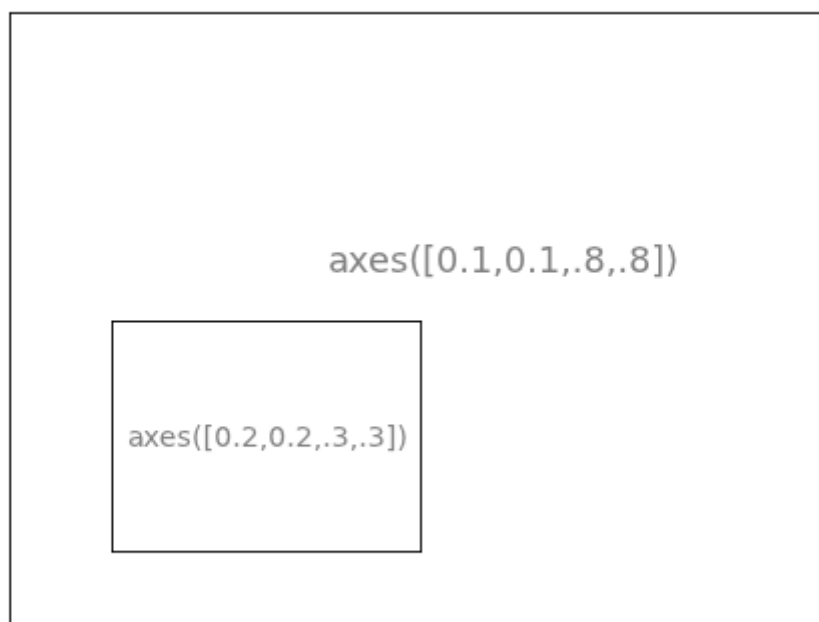
# Show result on screen
plt.show()
```



Axes se obnašajo podobno kot `subplot`, le da oni lahko ležijo kjerkoli v figuri.

```
axes([left, bottom, width, height])
```

Width in height sta normalizirana glede na figuro.



In [38]:

```
#import matplotlib.pyplot as plt
#import math

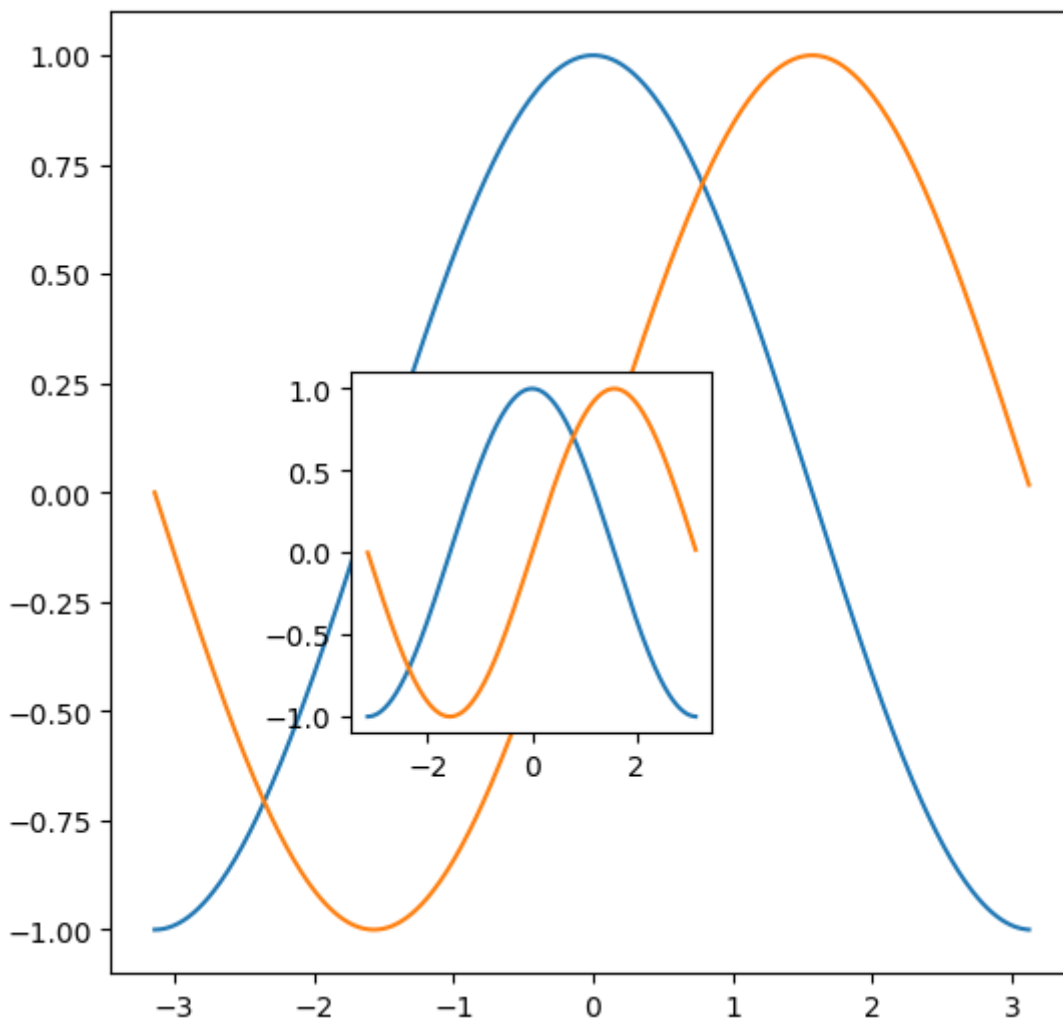
# Create a new figure of size 8x6 points, using 100 dots per inch
plt.figure(figsize=(6,6), dpi=100)

X = [math.radians(x) for x in range(-180, 180)]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]

# Create a new subplot from a grid of 1x1
plt.axes([0,0,0.8, 0.8])
plt.plot(X, C)
plt.plot(X, S)

plt.axes([0.2, 0.2, 0.3,0.3])
plt.plot(X, C)
plt.plot(X, S)

# Show result on screen
plt.show()
```



Spreminjanje lastnosti

Spreminjanje lastnosti črt

In [39]:

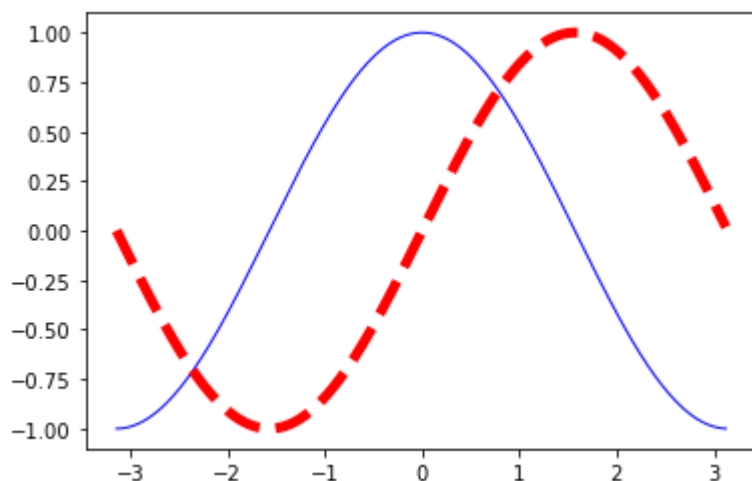
```
#import matplotlib.pyplot as plt
#import math

X = [math.radians(x) for x in range(-180, 180)]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]

# SPREMINJANJE BARVE, DEBELINE IN STIL ČRTE
plt.plot(X, C, color="blue", linewidth=1.0, linestyle="-")
#Plotting cosinus funkcijo
# color="blue" -> črta bo modre barve
# linewidth=1.0 -> debelina črte bo 1pixel
# linestyle="-" -> oblika črte bo neprekinjena

plt.plot(X, S, color="red", linewidth=5.0, linestyle="--")
#Plotting sinus funkcijo
# color="red" -> barva črte bo rdeča
# linewidth=5.0 -> debelina bo 5.0 pixlov
# linestyle="--" -> stil črte bo črtkana črta

plt.show()
```



Spreminjanje lastnosti osi

In [40]:

```

#import matplotlib.pyplot as plt
#import math

X = [math.radians(x) for x in range(-180, 180)]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]

# SPREMINJANJE BARVE, DEBELINE IN STIL ČRTE
plt.plot(X, C, color="blue", linewidth=1.0, linestyle="--")
#Plotting cosinus funkcijo
# color="blue" -> črta bo modre barve
# linewidth=1.0 -> debelina črte bo 1pixel
# linestyle="--" -> oblika črte bo neprekinjena

plt.plot(X, S, color="red", linewidth=5.0, linestyle="--")
#Plotting sinus funkcijo
# color="green" -> barva črte bo zelena
# linewidth=5.0 -> debelina bo 5.0 pixlov
# linestyle="--" -> stil črte bo črtna črta

# OBLIKOVANJE OSI (VELIKOST, VREDNOSTI)
plt.xlim(-5, max(X))
#Postavi meje x osi
# x os bo velika od -5 do max vrednosti X (vključno)

plt.xticks([-3.14, -3.14/2, 0, 3.14/2, 3.14], [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+
# Postavi te "oznake" na x osi
# Prvi argument [] pove katere oznake naj postavi
# Drugi argument [] pove kako naj jih izpiše. V našem primeru izpišemo v formatu La

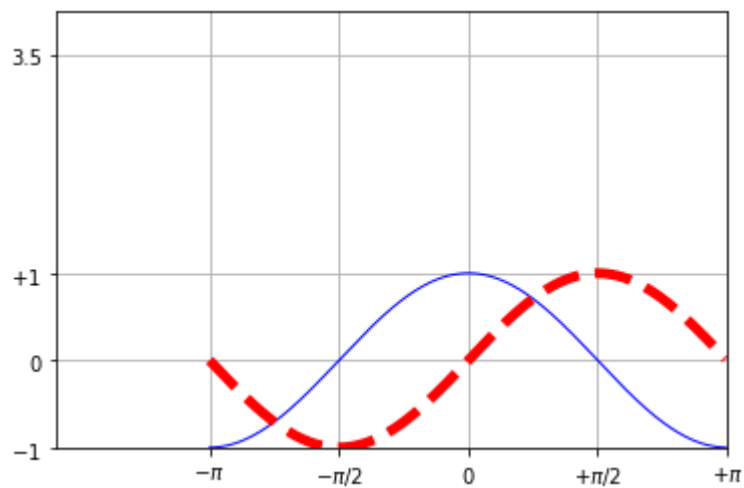
plt.ylim(min(C), 4)
#Postavi meje y osi
# y os bo velika od minimalne vrednosti cos do 4 vključno

plt.yticks([-1, 0, 1, 3.5], [r'$-1$', r'$0$', r'$+1$', '3.5'])
#Postavi te "oznake" na y osi
# Prvi argument [] pove katere oznake naj postavi
# Drugi argument [] pove kako naj jih izpiše. V našem primeru izpišemo v formatu La

plt.grid()
# doda pomožne črte na naš graf
# črte izhajajo iz naših x in y "ticks"

plt.show()

```

Spreminjanje lastnosti okvira

Črte, ki uokvirjajo naš graf se imenujejo **spines**.

In [44]:

```

#import matplotlib.pyplot as plt
#import math

X = [math.radians(x) for x in range(-180, 180)]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]

# SPREMINJANJE BARVE, DEBELINE IN STIL ČRTE
plt.plot(X, C, color="blue", linewidth=1.0, linestyle="--")
#Plotting cosinus funkcijo
# color="blue" -> črta bo modre barve
# linewidth=1.0 -> debelina črte bo 1pixel
# linestyle="--" -> oblika črte bo neprekinjena

plt.plot(X, S, color="red", linewidth=5.0, linestyle="--")
#Plotting sinus funkcijo
# color="green" -> barva črte bo zelena
# linewidth=5.0 -> debelina bo 5.0 pixlov
# linestyle="--" -> stil črte bo črtna črta

# OBLIKOVANJE OSI (VELIKOST, VREDNOSTI)
plt.xlim(-5, max(X))
#Postavi meje x osi
# x os bo velika od -5 do max vrednosti X (vključno)

plt.xticks([-3.14, -3.14/2, 0, 3.14/2, 3.14], [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+
# Postavi te "oznake" na x osi
# Prvi argument [] pove katere oznake naj postavi
# Drugi argument [] pove kako naj jih izpiše. V našem primeru izpišemo v formatu La

plt.ylim(min(C), 4)
#Postavi meje y osi
# y os bo velika od minimalne vrednosti cos do 4 vključno

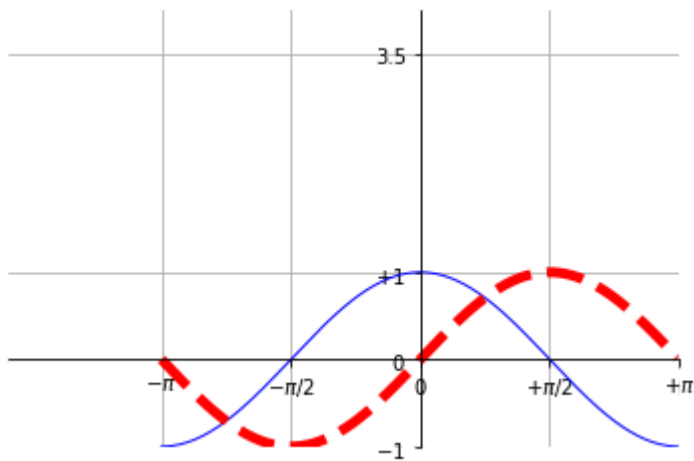
plt.yticks([-1, 0, 1, 3.5], [r'$-1$', r'$0$', r'$+1$', '3.5'])
#Postavi te "oznake" na y osi
# Prvi argument [] pove katere oznake naj postavi
# Drugi argument [] pove kako naj jih izpiše. V našem primeru izpišemo v formatu La

plt.grid()
# doda pomožne črte na naš graf
# črte izhajajo iz naših x in y "ticks"

# OBLIKOVANJE OKVIRA
ax = plt.gca() # GetCurrentAxes -> se prav dobimo nš trenutni graf
ax.spines['right'].set_color('none') # našo desno črto napravimo nevidno
ax.spines['top'].set_color('none') # naši zgornjo črto napravimo nevidno
#ax.xaxis.set_ticks_position('top') # naše x oznake lahko premakno iz spodnje na zg
ax.spines['bottom'].set_position(('data',0)) # ta nič pomen, da gre skoz 0 na y
#ax.yaxis.set_ticks_position('right') # naše y oznake lahko premaknemo iz leve na d
ax.spines['left'].set_position(('data',0))

```

```
plt.show()
```



Dodajanje legende

In [45]:

```

#import matplotlib.pyplot as plt
#import math

X = [math.radians(x) for x in range(-180, 180)]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]

# SPREMINJANJE BARVE, DEBELINE IN STIL ČRTE
plt.plot(X, C, color="blue", linewidth=1.0, linestyle="-", label="cosinus")
#Plotting cosinus funkcijo
# color="blue" -> črta bo modre barve
# linewidth=1.0 -> debelina črte bo 1pixel
# linestyle="-" -> oblika črte bo neprekinjena

plt.plot(X, S, color="red", linewidth=5.0, linestyle="--", label="sinus")
#Plotting sinus funkcijo
# color="green" -> barva črte bo zelena
# linewidth=5.0 -> debelina bo 5.0 pixlov
# linestyle="--" -> stil črte bo črtkana črta

# OBLIKOVANJE OSI (VELIKOST, VREDNOSTI)
plt.xlim(-5, max(X))
#Postavi meje x osi
# x os bo velika od -5 do max vrednosti X (vključno)

plt.xticks([-3.14, -3.14/2, 0, 3.14/2, 3.14], [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+
# Postavi te "oznake" na x osi
# Prvi argument [] pove katere oznake naj postavi
# Drugi argument [] pove kako naj jih izpiše. V našem primeru izpišemo v formatu La

plt.ylim(min(C), 4)
#Postavi meje y osi
# y os bo velika od minimalne vrednosti cos do 4 vključno

plt.yticks([-1, 0, 1, 3.5], [r'$-1$', r'$0$', r'$+1$', '3.5'])
#Postavi te "oznake" na y osi
# Prvi argument [] pove katere oznake naj postavi
# Drugi argument [] pove kako naj jih izpiše. V našem primeru izpišemo v formatu La

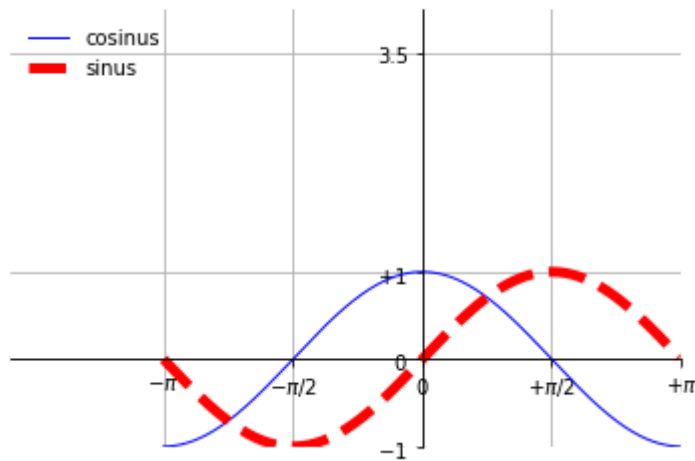
plt.grid()
# doda pomožne črte na naš graf
# črte izhajajo iz naših x in y "ticks"

# OBLIKOVANJE OKVIRA
ax = plt.gca() # GetCurrentAxes -> se prav dobimo nš trenutni graf
ax.spines['right'].set_color('none') # našo desno črto napravimo nevidno
ax.spines['top'].set_color('none') # naši zgornjo črto napravimo nevidno
#ax.xaxis.set_ticks_position('top') # naše x oznake lahko premakno iz spodnje na zg
ax.spines['bottom'].set_position(('data',0)) # ta nič pomen, da gre skozi 0 na y
#ax.yaxis.set_ticks_position('right') # naše y oznake lahko premaknemo iz leve na d
ax.spines['left'].set_position(('data',0))

```

```
# DODAJANJE LEGENDE
plt.legend(loc='upper left', frameon=False)
#doda nam legendo (pred tem mormo našim funkcijam (.plot())) dodat parameter: "label"
# loc -> lokacija
# frameon -> če je uokvirjena legenda ali ne

plt.show()
```



Anotacija

In [47]:

```

#import matplotlib.pyplot as plt
#import math

X = [math.radians(x) for x in range(-180, 180)]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]

# SPREMINJANJE BARVE, DEBELINE IN STIL ČRTE
plt.plot(X, C, color="blue", linewidth=1.0, linestyle="--", label="cosinus")
#Plotting cosinus funkcijo
# color="blue" -> črta bo modre barve
# linewidth=1.0 -> debelina črte bo 1pixel
# linestyle="--" -> oblika črte bo neprekinjena

plt.plot(X, S, color="red", linewidth=5.0, linestyle="--", label="sinus")
#Plotting sinus funkcijo
# color="green" -> barva črte bo zelena
# linewidth=5.0 -> debelina bo 5.0 pixlov
# linestyle="--" -> stil črte bo črtkana črta

# OBLIKOVANJE OSI (VELIKOST, VREDNOSTI)
plt.xlim(-5, max(X))
#Postavi meje x osi
# x os bo velika od -5 do max vrednosti X (vključno)

plt.xticks([-3.14, -3.14/2, 0, 3.14/2, 3.14], [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+
# Postavi te "oznake" na x osi
# Prvi argument [] pove katere oznake naj postavi
# Drugi argument [] pove kako naj jih izpiše. V našem primeru izpišemo v formatu La

plt.ylim(min(C), 4)
#Postavi meje y osi
# y os bo velika od minimalne vrednosti cos do 4 vključno

plt.yticks([-1, 0, 1, 3.5], [r'$-1$', r'$0$', r'$+1$', '3.5'])
#Postavi te "oznake" na y osi
# Prvi argument [] pove katere oznake naj postavi
# Drugi argument [] pove kako naj jih izpiše. V našem primeru izpišemo v formatu La

plt.grid()
# doda pomožne črte na naš graf
# črte izhajajo iz naših x in y "ticks"

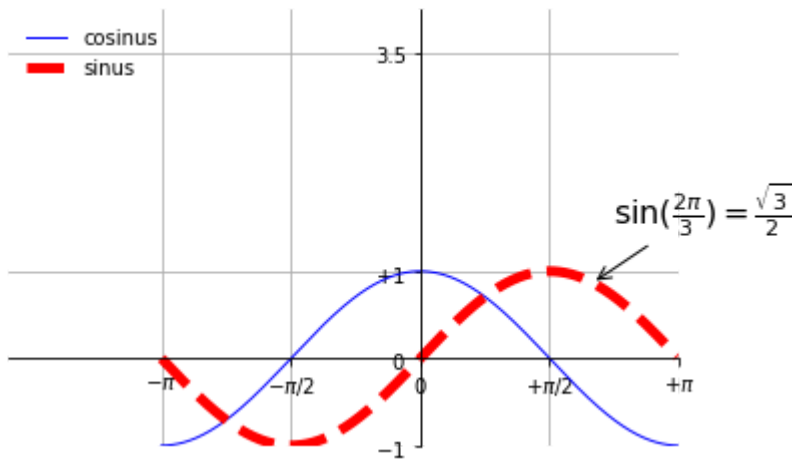
# OBLIKOVANJE OKVIRA
ax = plt.gca() # GetCurrentAxes -> se prav dobimo nš trenutni graf
ax.spines['right'].set_color('none') # našo desno črto napravimo nevidno
ax.spines['top'].set_color('none') # naši zgornjo črto napravimo nevidno
#ax.xaxis.set_ticks_position('top') # naše x oznake lahko premakno iz spodnje na zg
ax.spines['bottom'].set_position(('data',0)) # ta nič pomen, da gre skozi 0 na y
#ax.yaxis.set_ticks_position('right') # naše y oznake lahko premaknemo iz leve na d
ax.spines['left'].set_position(('data',0))

```

```
# DODAJANJE LEGENDE
plt.legend(loc='upper left', frameon=False)
#doda nam legendo (pred tem mormo našim funkcijam (.plot())) dodat parameter: "label"
# loc -> lokacija
# frameon -> če je uokvirjena legenda ali ne

# ANOTACIJA -> OPOMBA
t = 2*3.14/3 # vrednost naše točke, na katero hočemo pokazati
plt.annotate(r'$\sin(\frac{2\pi}{3})=\frac{\sqrt{3}}{2}$',
             xy=(t, math.sin(t)),
             xycoords='data',
             xytext=(10, +30),
             textcoords='offset points',
             fontsize=16,
             arrowprops=dict(arrowstyle="->"))
#r'' -> izpis v latex formatu
#xy=(t, math.sin(t)) -> specificira kero pozicijo anotiramo (točka na kero nej poka
#xycoords="data" -> koordinatni sistem v katerem sta xy=
#xytext -> specificiram kam naj vstavi text
#textcoords -> koordinatni sistem za text -> 'offset points'    offset (in points)
#arrowprops
# arrowstyle -> kakšnega stila je naša puščica

plt.show()
```



Animation

FuncAnimation(fig, update, interval=10)

- fig - figura katero animiramo
- update - funkcija katero se kliče vsak nov frame. Frame spremenljivka je vedno podana kot prva, ostalo so naši poljubni parametri
- interval - delay med frami v milisekundah. Default je 200

In [48]:

```

import math
import matplotlib
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
%matplotlib notebook
# zgornja vrstica je potrebna, če delamo na jupyter notebook.
# - če je "inline" se stvari prikazujejo kot statične slike
# - če je "notebook" bo normalno prikazoval animacijo

fig = plt.figure(figsize=(6,6), facecolor='white')

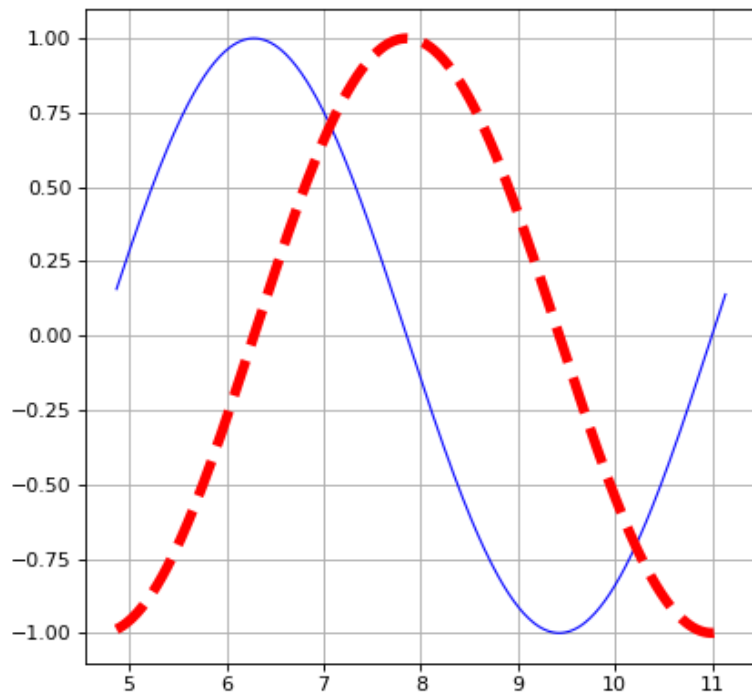
ax = plt.subplot(1,1,1)

X = [math.radians(x) for x in range(-180, 180)]
X_korak = X[-1] - X[-2]
C = [math.cos(i) for i in X]
S = [math.sin(i) for i in X]
plt.plot(X, C, color="blue", linewidth=1.0, linestyle="-", label="cosine")
plt.plot(X, S, color="red", linewidth=5.0, linestyle="--", label="sine")

def update(frame):
    global X, X_korak, ax, C, S
    del X[0]
    X.append(X[-1] + X_korak)
    C = [math.cos(i) for i in X]
    S = [math.sin(i) for i in X]
    ax.clear()
    ax.plot(X, C, color="blue", linewidth=1.0, linestyle="-", label="cosine")
    ax.plot(X, S, color="red", linewidth=5.0, linestyle="--", label="sine")
    ax.grid()

animation = FuncAnimation(fig, update, interval = 40) # 40 milisekund je približno
plt.show()

```

PRIMERI OSTALIH GRAFOV

Scatter Plot

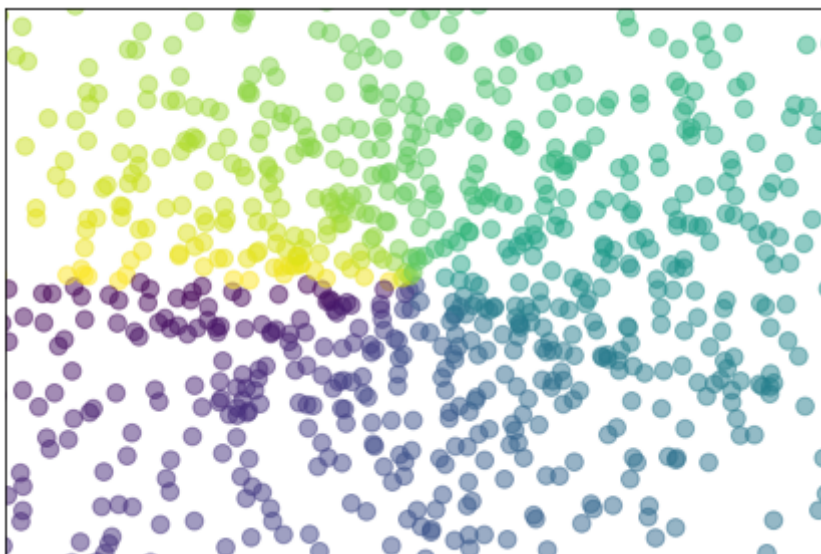
In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

n = 1024
X = np.random.normal(0,1,n)
Y = np.random.normal(0,1,n)
T = np.arctan2(Y,X)

plt.axes([0.025,0.025,0.95,0.95])
plt.scatter(X,Y, s=75, c=T, alpha=.5)

plt.xlim(-1.5,1.5), plt.xticks([])
plt.ylim(-1.5,1.5), plt.yticks([])
# savefig('../figures/scatter_ex.png',dpi=48)
plt.show()
```



Bar plot

In [2]:

```

import numpy as np
import matplotlib.pyplot as plt
##matplotlib notebook

n = 12
X = np.arange(n)
Y1 = (1-X/float(n)) * np.random.uniform(0.5,1.0,n)
Y2 = (1-X/float(n)) * np.random.uniform(0.5,1.0,n)

plt.axes([0.025,0.025,0.95,0.95])
plt.bar(X, +Y1, facecolor='#9999ff', edgecolor='white')
plt.bar(X, -Y2, facecolor='#ff9999', edgecolor='white')

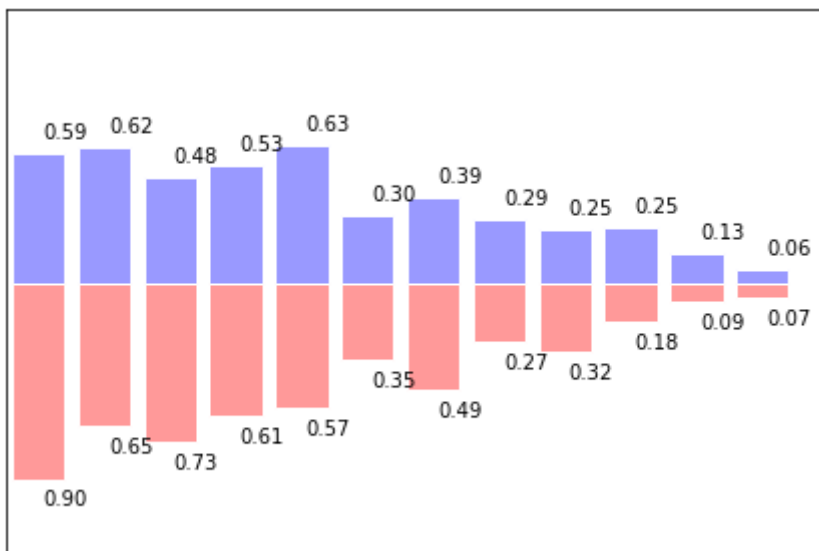
for x,y in zip(X,Y1):
    plt.text(x+0.4, y+0.05, '%.2f' % y, ha='center', va= 'bottom')

for x,y in zip(X,Y2):
    plt.text(x+0.4, -y-0.05, '%.2f' % y, ha='center', va= 'top')

plt.xlim(-.5,n), plt.xticks([])
plt.ylim(-1.25,+1.25), plt.yticks([])

plt.show()

```



Pie chart

In [3]:

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib notebook

n = 20
Z = np.ones(n)
Z[-1] *= 2

plt.axes([0.025,0.025,0.95,0.95])

plt.pie(Z, explode=Z*.05, colors = ['%f' % (i/float(n)) for i in range(n)])
plt.gca().set_aspect('equal')
plt.xticks([], plt.yticks([]))

plt.show()
```



In []:

Vaja - Matplotlib

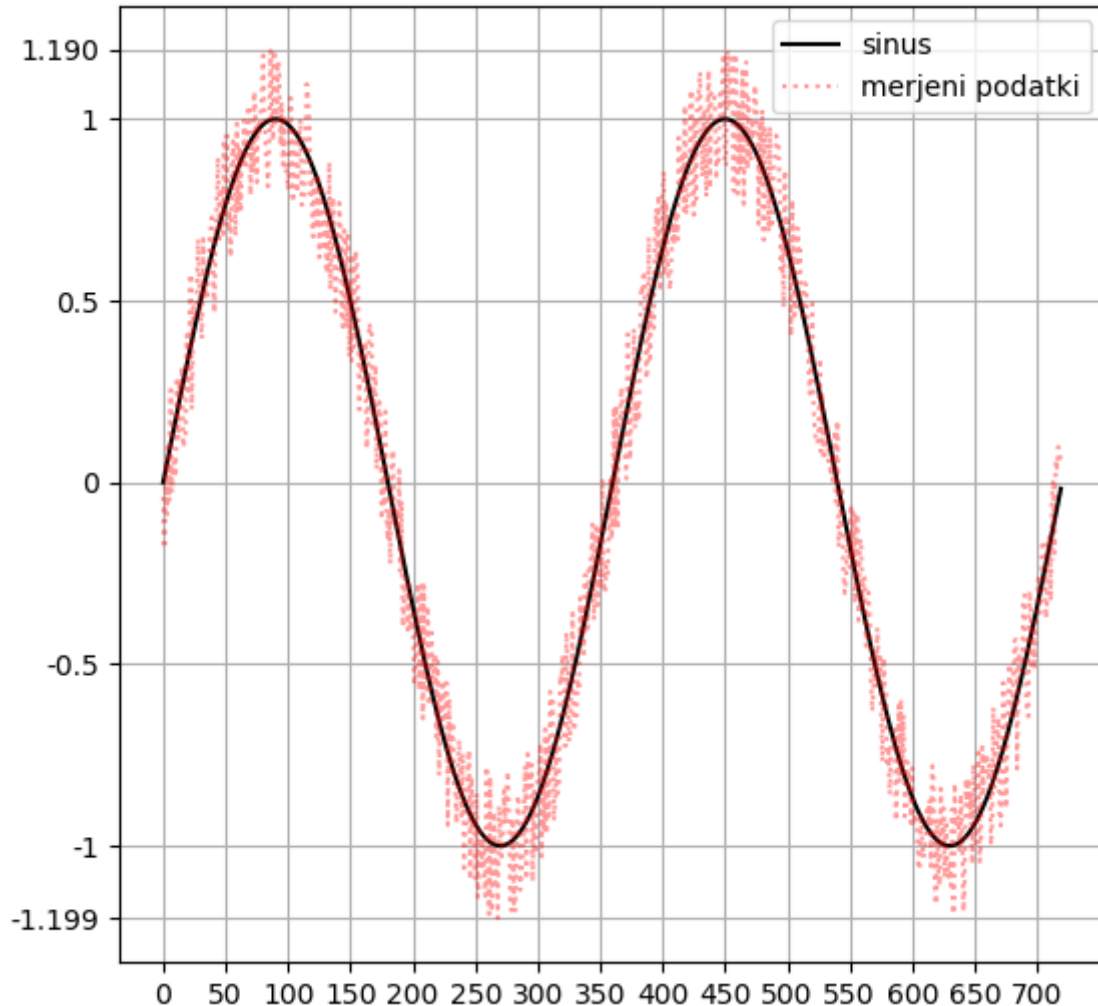
<https://pastebin.com/tU3PD8KJ> (<https://pastebin.com/tU3PD8KJ>)

Podatke imate podane:

```
import random
import math
import matplotlib.pyplot as plt

x = [(deg) for deg in range(0, 2*360)]
sin1 = [math.sin(math.radians(i)) for i in x]
sin2 = [i+(random.randint(-20,20)/100) for i in sin1]
```

Vaša naloga je, da napravite sledeč graf:



Velikost grafa naj bo 8x8.

Podatke **sin1** ("normalna sinus krivulja") prikažite z navadno črto, ki naj bo črne barve. Njeno ime naj bo **sinus**.

Podatke **sin2** ("podatki z vključenimi motnjami") prikažite kot točkovne vrednosti. Naj bodo rdeče barve. Zmanjšajte njihovo transparentnost (alpha). Ime podatkov naj bo "merjeni podatki"

x_ticks naj bodo vsakih 50 enot na x osi.

y_ticks naj bodo pri minimalni vrednosti **sin2** podatkov, -1, -0.5, 0, 0.5, 1, maksimalni vrednosti **sin2** podatkov. Minimalna vrednost in maksimalna vrednost naj bosta podani na 3 decimalna mesta natančno.

Prikažite legendo in pomožne črte.

Za pomoč naj vam bo matplotlib dokumentacija.

https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html
https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.plot.html

In [52]:

```
import random
import math
import matplotlib.pyplot as plt

x = [(deg) for deg in range(0, 2*360)]
sin1 = [math.sin(math.radians(i)) for i in x]
sin2 = [i+(random.randint(-20,20)/100) for i in sin1]

fig = plt.figure(figsize=(8,8))

plt.plot(x, sin1, color="black", label="sinus")
plt.plot(x, sin2, linestyle="dotted", color="red", label="merjeni podatki", alpha=0.5)

x_ticks = x[::50]
x_ticks_notation = x_ticks
plt.xticks(x_ticks, x_ticks_notation)

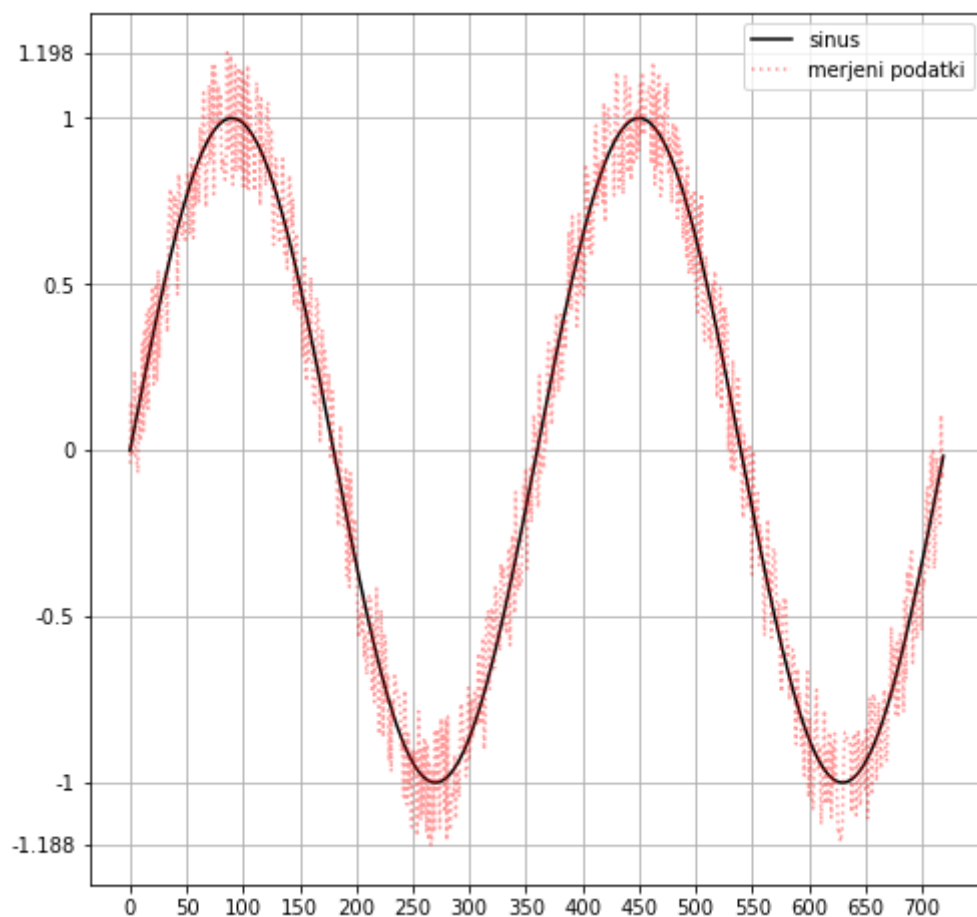
y_ticks = [-1, -0.5, 0, 0.5, 1]
y_ticks.append(max(sin2))
y_ticks.append(min(sin2))

y_ticks_notation = ["-1", "-0.5", "0", "0.5", "1", f"{max(sin2):.3f}", f"{min(sin2):.3f}"]
plt.yticks(y_ticks, y_ticks_notation)

plt.legend()

plt.grid()

plt.show()
```



In []:

In []:

In []:

In []: