

A Python Flipbook

With `{flipbookr}` and `xaringan`

Gina Reynolds, March 2020

This is pretty experimental work. The python "parser" (basically no parsing) means there are strong constraints on code. Each new line must yield complete code statement...

```
# import pandas as pd
```

```
# import pandas as pd  
import matplotlib.pyplot as plt
```

```
# import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
# import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
t = np.arange(0, 2, .05)
```

```
# import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
t = np.arange(0, 2, .05)
t
```

```
array([0.   , 0.05, 0.1  , 0.15, 0.2  , 0.25, 0.3  , 0.35, 0.4  , 0.45,
       0.55, 0.6  , 0.65, 0.7  , 0.75, 0.8  , 0.85, 0.9  , 0.95, 1.   ,
       1.1  , 1.15, 1.2  , 1.25, 1.3  , 1.35, 1.4  , 1.45, 1.5  , 1.55,
       1.65, 1.7  , 1.75, 1.8  , 1.85, 1.9  , 1.95])
```

```
# import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
t = np.arange(0, 2, .05)
t
s = np.sin(2*np.pi*t)
```

```
array([0.   , 0.05, 0.1  , 0.15, 0.2  , 0.25, 0.3  , 0.35, 0.4  , 0.45,
       0.55, 0.6  , 0.65, 0.7  , 0.75, 0.8  , 0.85, 0.9  , 0.95, 1.   ,
       1.1  , 1.15, 1.2  , 1.25, 1.3  , 1.35, 1.4  , 1.45, 1.5  , 1.55,
       1.65, 1.7  , 1.75, 1.8  , 1.85, 1.9  , 1.95])
```

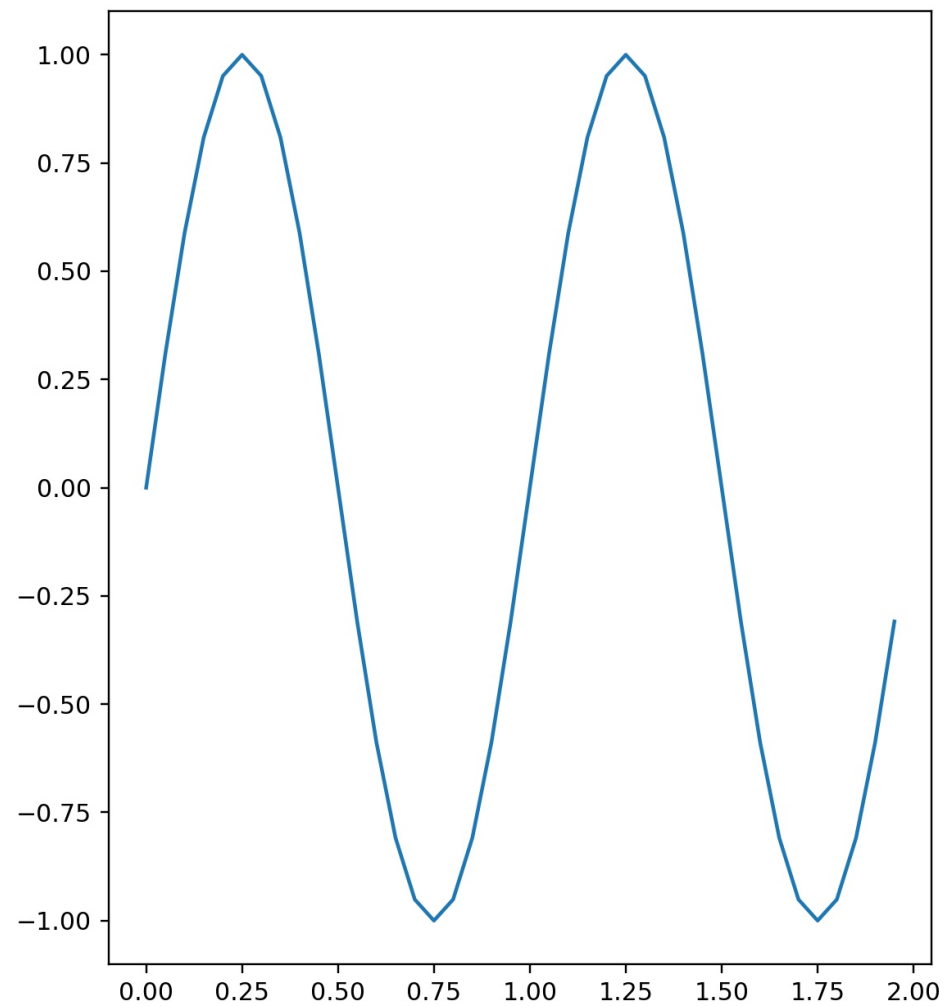


```
# import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
t = np.arange(0, 2, .05)
t
s = np.sin(2*np.pi*t)
s
```

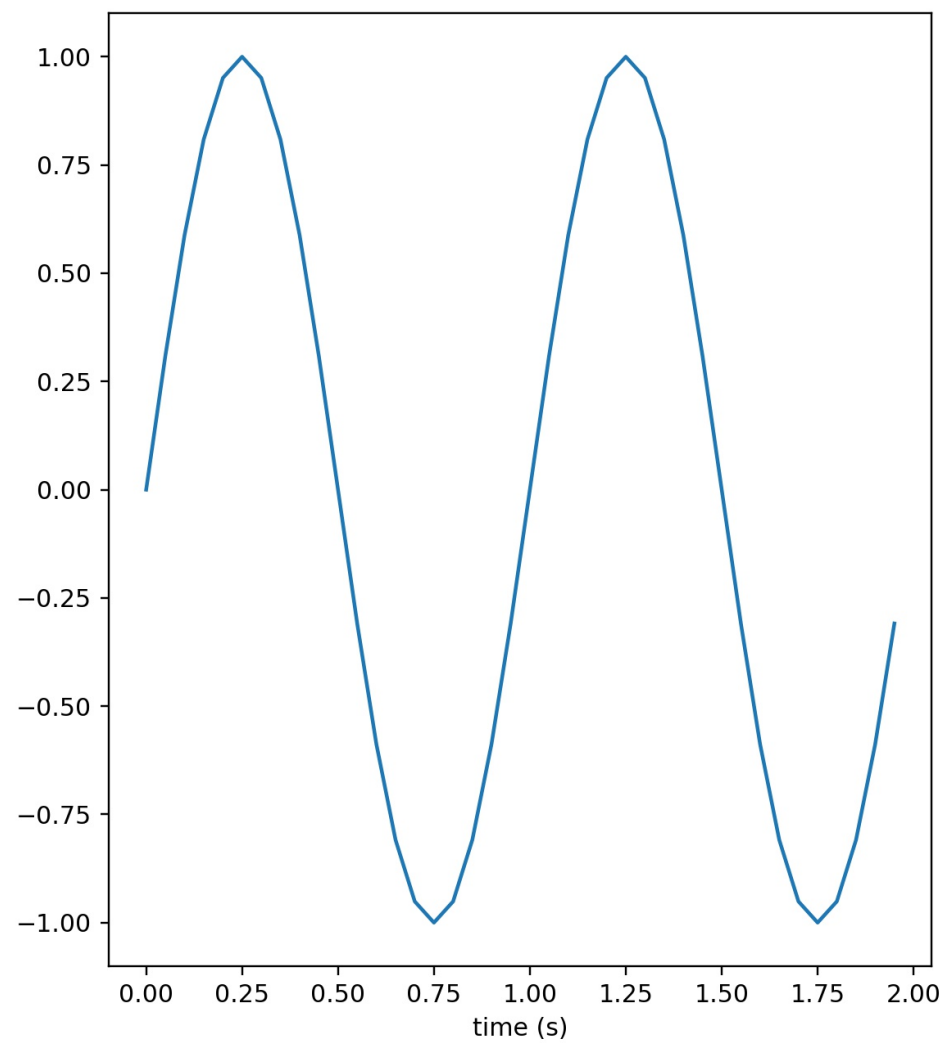
```
array([0.   , 0.05, 0.1 , 0.15, 0.2 , 0.25, 0.3 , 0.35, 0.4 , 0.45,
       0.55, 0.6 , 0.65, 0.7 , 0.75, 0.8 , 0.85, 0.9 , 0.95, 1.   ,
       1.1 , 1.15, 1.2 , 1.25, 1.3 , 1.35, 1.4 , 1.45, 1.5 , 1.55,
       1.65, 1.7 , 1.75, 1.8 , 1.85, 1.9 , 1.95])
```

```
array([ 0.00000000e+00,  3.09016994e-01,  5.87785252e-01,  8.09016
       9.51056516e-01,  1.00000000e+00,  9.51056516e-01,  8.09016
       5.87785252e-01,  3.09016994e-01,  1.22464680e-16, -3.09016
      -5.87785252e-01, -8.09016994e-01, -9.51056516e-01, -1.00000
      -9.51056516e-01, -8.09016994e-01, -5.87785252e-01, -3.09016
      -2.44929360e-16,  3.09016994e-01,  5.87785252e-01,  8.09016
       9.51056516e-01,  1.00000000e+00,  9.51056516e-01,  8.09016
       5.87785252e-01,  3.09016994e-01,  3.67394040e-16, -3.09016
      -5.87785252e-01, -8.09016994e-01, -9.51056516e-01, -1.00000
      -9.51056516e-01, -8.09016994e-01, -5.87785252e-01, -3.09016
```

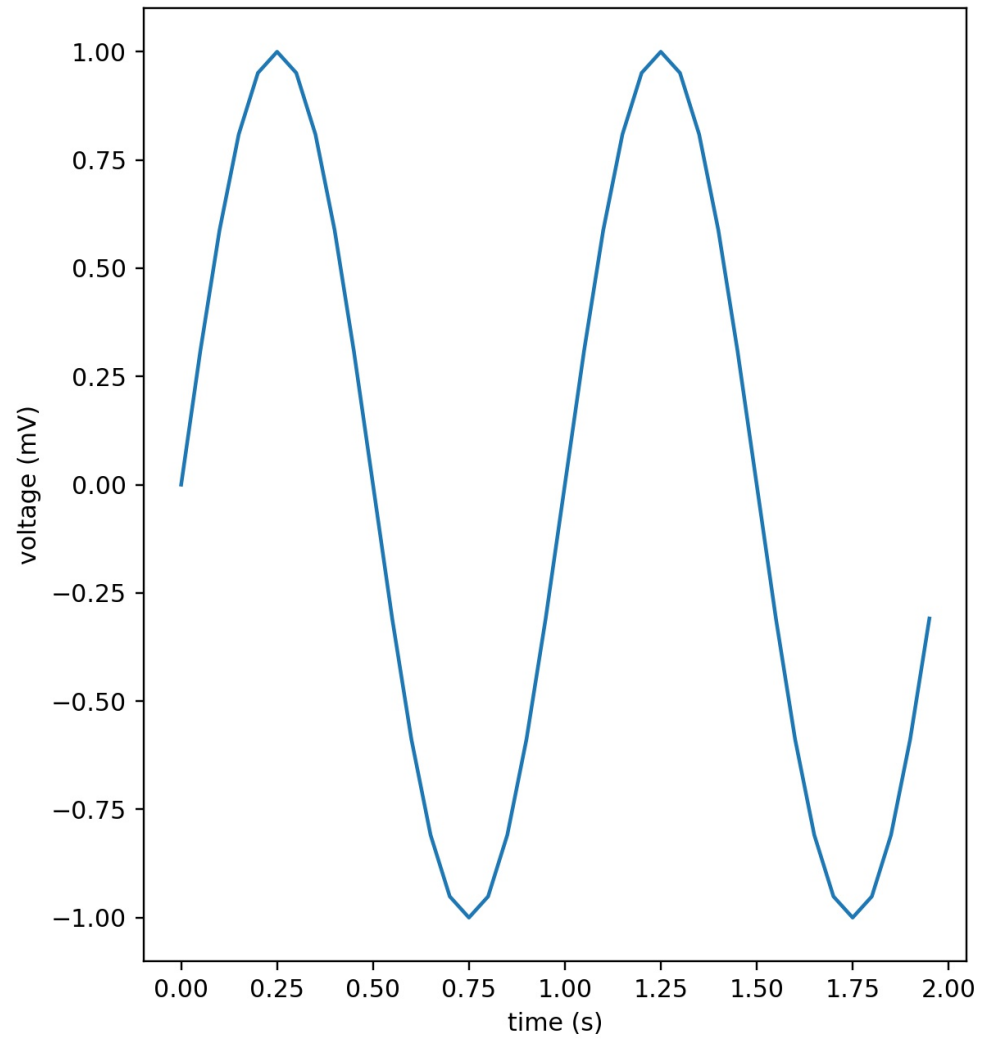
```
plt.plot(t, s)
```



```
plt.plot(t, s)  
plt.xlabel('time (s)')
```



```
plt.plot(t, s)  
plt.xlabel('time (s)')  
plt.ylabel('voltage (mV)')
```



```
plt.plot(t, s)
plt.xlabel('time (s)')
plt.ylabel('voltage (mV)')
plt.grid(True)
```

```
plt.plot(t, s)
plt.xlabel('time (s)')
plt.ylabel('voltage (mV)')
plt.grid(True)
plt.show()
```

