stats

Jane Doe

7/27/2022

Table of contents

Preface

This is a Quarto book.

To learn more about Quarto books visit https://quarto.org/docs/books.

Some more content

1 ! ipython suppress id=b0641b6ae09942babe0c88f69c55482d

```
%pushd book-materials
import numpy as np
np.random.seed(12345)
np.set_printoptions(precision=4, suppress=True)
```

```
#! ipython id=085dfdc5abf744bebb6e84ea614aa99d
import numpy as np
data = [np.random.standard_normal() for i in range(7)]
data
```

```
#! ipython id=1a4e6fcb22eb4f89986abfae4c83b48d
a = [1, 2, 3]
```

```
#! ipython id=64f8a5dfa016429c8d659bf729003401
b = a
b
```

```
#! ipython id=0eced45afa5347248bb2af66fa5c4e2b
a.append(4)
b
```

```
#! ipython id=1018283ab59f4f05bfe84afca4fc9936
def append_element(some_list, element):
    some_list.append(element)
```

```
#! ipython id=2edb843baee44895b71dbd9474407c9c
data = [1, 2, 3]
append_element(data, 4)
data
```

```
#! ipython id=e71481e654d64e649533772a2381a3c6
a = 5
type(a)
a = "foo"
type(a)
```

```
#! ipython allow_exceptions id=c9cf6d998c39413a9fc48646f91dec3e
"5" + 5
```

```
#! ipython id=0003b825a15647e49598d18de47cbd57
a = 4.5
b = 2
# String formatting, to be visited later
print(f"a is {type(a)}, b is {type(b)}")
a / b
```

```
#! ipython id=f3d0ca0ebd384742976ccd83633ac8be
a = 5
isinstance(a, int)
```

```
#! ipython id=5dacb5ef6e394fe7a8b3f724cb9cdf4d
a = 5; b = 4.5
isinstance(a, (int, float))
isinstance(b, (int, float))
```

```
#! ipython suppress id=7bd32461c0bf45ff9bee3de05a7fa5cf
a = "foo"
```

```
#! ipython id=d2aef8895e6f4b67aaa1d6cbd2b6affa
getattr(a, "split")
```

```
#! ipython id=f2617db9765f49009c55756ce9cf87cd
def isiterable(obj):
    try:
        iter(obj)
        return True
    except TypeError: # not iterable
        return False
```

```
#! ipython id=1ea41933d16e460380316fd7a4e0b1d8
isiterable("a string")
isiterable([1, 2, 3])
isiterable(5)
```

```
#! ipython id=84c35fbe613341129d0e94d6fc798e76
5 - 7
12 + 21.5
5 <= 2
```

```
#! ipython id=a91f01a0ee1e43e2a000db5609d04789
a = [1, 2, 3]
b = a
c = list(a)
a is b
a is not c
```

```
#! ipython id=1fca12df22264d9881690fcaa804149d
a == c
```

```
#! ipython id=e6fe568326a94c0d899c9c8b31f0c355
a = None
a is None
```

```
#! ipython id=7db7331aee334850b9b17cb4f86e6d18
a_list = ["foo", 2, [4, 5]]
a_list[2] = (3, 4)
a_list
```

```
#! ipython allow_exceptions id=2536d7cdf32b4faaa753207cbfcc17fe
a_tuple = (3, 5, (4, 5))
a_tuple[1] = "four"
```

```
#! ipython id=51ad7d167b834b85a8eda9bc416c955b
ival = 17239871
ival ** 6
```

```
#! ipython id=3350052c32fc49b599322353c27a4586
fval = 7.243
fval2 = 6.78e-5
```

```
#! ipython id=38bc4370bb6840c5beb557dead1d6998
3 / 2
```

```
#! ipython id=2ccf1adf240348008ed8ce811792dbf8
3 // 2
```

```
#! ipython verbatim id=f1454b6b49ff4e2bba6124576e18cd53
c = """
This is a longer string that
spans multiple lines
"""
```

```
#! ipython id=03cb238d855d42dda40a45ab91254038
c.count("\n")
```

```
#! ipython allow_exceptions id=59788f621484492ba307f10961edbb6c
a = "this is a string"
a[10] = "f"
```

```
#! ipython id=f5c3a08f5aaf4706a5122f3ef51f5e89
b = a.replace("string", "longer string")
b
```

```
#! ipython id=90017a4e11754eac828714ee61b43ac1
a
```

```
#! ipython id=99b192a5b56946cda8707464fb8e2bea
a = 5.6
s = str(a)
print(s)
```

```
#! ipython id=5bce41fece2f47bc90fb813f26cfa081
s = "python"
list(s)
```

```
s[:3]
```

```
#! ipython id=c826919c945643b8adb2dc48b3fa204b
s = "12\\34"
print(s)
```

```
#! ipython id=f705192174ec4c849203a659260960e0
s = r"this\has\no\special\characters"
s
```

```
#! ipython id=9e52b9bfcf734a899ad5b9deecc6b5f1
a = "this is the first half "
b = "and this is the second half"
a + b
```

```
#! ipython id=8ba0128b1e144dfd87b0a96bbb95e090
template = "{0:.2f} {1:s} are worth US${2:d}"
```

```
#! ipython id=830e8c337251466d8704122c38e4a31d
template.format(88.46, "Argentine Pesos", 1)
```

```
#! ipython id=c5eab29881e7453bbdc2b8d3f1e81924
amount = 10
rate = 88.46
currency = "Pesos"
result = f"{amount} {currency} is worth US${amount / rate}"
```

```
#! ipython id=d4d50688c6c145d5a2d11d846f03dba2
f"{amount} {currency} is worth US${amount / rate:.2f}"
```

```
#! ipython id=fe3d6a3ffd5c4906858baeb46363aee5
val = "español"
val
```

```
#! ipython id=f1369533406f414ea648c7b80c22cba3
val_utf8 = val.encode("utf-8")
val_utf8
```

```
type(val_utf8)
```

```
#! ipython id=6cfcab789284478f9fe920568aad6276
val_utf8.decode("utf-8")
```

```
#! ipython id=9e55ca7c4e2642e5a26f603523e87564
val.encode("latin1")
val.encode("utf-16")
val.encode("utf-16le")
```

```
#! ipython id=422e235bb54842adb436742e6caba89c
True and True
False or True
```

```
#! ipython id=1051ac23010f466c8f2160c865da145f
int(False)
int(True)
```

```
#! ipython id=14dc7378115c43349c92e34020bb3b71
a = True
b = False
not a
not b
```

```
#! ipython id=345c0ce8702b41539a102f07716ff00d
s = "3.14159"
fval = float(s)
type(fval)
int(fval)
bool(fval)
bool(0)
```

```
#! ipython id=163e5f37123741a88e61ef81c85fedc1
a = None
a is None
b = 5
b is not None
```

```
#! ipython id=552d7fff9f2a45a38b54fa44add624c7
from datetime import datetime, date, time
dt = datetime(2011, 10, 29, 20, 30, 21)
dt.day
dt.minute
```

```
#! ipython id=a527d76de7e141568e7dd9e5c9f8d8ce
dt.date()
dt.time()
```

```
#! ipython id=9bd051a88f7b458fae19e8e156fda078
dt.strftime("%Y-%m-%d %H:%M")
```

```
#! ipython id=3fe2b447220b4b8ca70b526d76e31cb5
datetime.strptime("20091031", "%Y%m%d")
```

```
#! ipython id=178fb4d50aff4d68a0a7e0ab28ee71f3
dt_hour = dt.replace(minute=0, second=0)
dt_hour
```

```
#! ipython id=9ad24ebe03e049b2af9a81ec22a90675
dt
```

```
#! ipython id=1a41f2e49eee432cbc4a32f4298e1f3c
dt2 = datetime(2011, 11, 15, 22, 30)
delta = dt2 - dt
delta
type(delta)
```

```
#! ipython id=04baa0c20d534202ad4714e5c68d04a4
dt
dt + delta
```

```
#! ipython id=6551625a4f864b5fb5ce6b81e5ffd81b
a = 5; b = 7
c = 8; d = 4
if a < b or c > d:
    print("Made it")
```

```
#! ipython id=ace8e0c5cbb349cb852d8248394b0f7c
4 > 3 > 2 > 1
```

```
#! ipython id=6e3b4a652a894d25906ad84f8b4248ea
#! blockstart
for i in range(4):
    for j in range(4):
        if j > i:
            break
        print((i, j))
#! blockend
```

```
#! ipython id=60383320960a45bcb83e21d713d5619c
range(10)
list(range(10))
```

```
#! ipython id=71696aba433c43f6b2e05d488837f0f3
list(range(0, 20, 2))
list(range(5, 0, -1))
```

```
#! ipython id=c9e25935441242149e5f4b406032113f
seq = [1, 2, 3, 4]
for i in range(len(seq)):
    print(f"element {i}: {seq[i]}")
```

```
#! ipython id=430c1852ec4847bcad3afc4a3a68c2f3
total = 0
for i in range(100_000):
    # % is the modulo operator
    if i % 3 == 0 or i % 5 == 0:
        total += i
print(total)
```

```
#! ipython suppress id=25d6eae18f4846ed89c817cb487df3b5
%popd
```

2 Quarto Basics 2

For a demonstration of a line plot on a polar axis, see Figure 2.1.

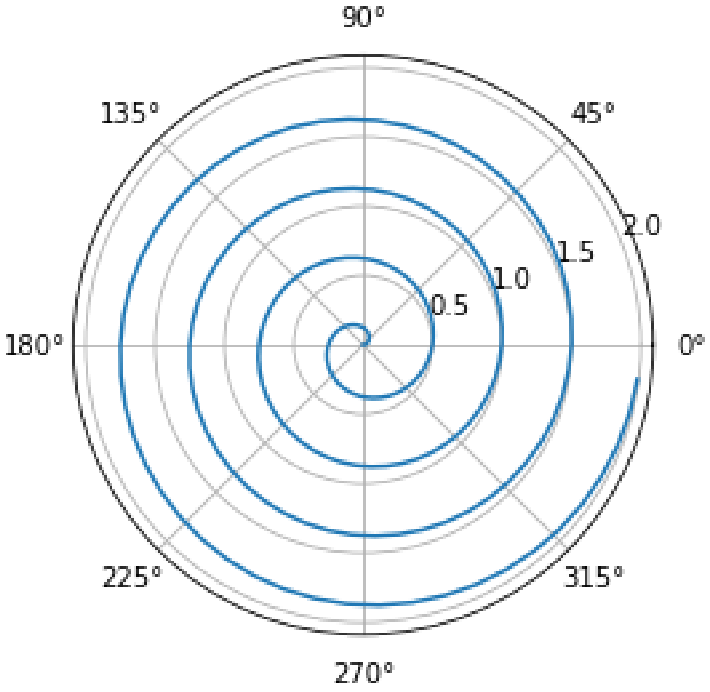

Figure 2.1: A line plot on a polar axis

3 Introduction

This is a book created from markdown and executable code.

See Knuth (1984) for additional discussion of literate programming.

#| label: fig-polar #| fig-cap: “A line plot on a polar axis”

import numpy as np import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01) theta = 4 * np.pi * r fig, ax = plt.subplots(subplot_kw = {‘projection’: ‘polar’}) ax.plot(theta, r) ax.set_rticks([0.5, 1, 1.5, 2]) ax.grid(True) plt.show()

4 Summary

In summary, this book has no content whatsoever.

References

Knuth, Donald E. 1984. “Literate Programming.” *Comput. J.* 27 (2): 97–111. https://doi.org/10.1093/comjnl/27.2.97.