Section 1.2

October 22, 2023

Figures for Section 1.2

[19]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

Figure 1.5

100 simulated values of the cyclical time series (left panel), the stochastic amplitude (middle panel), and the sine part (right panel).

[10]:
```
# plot for Cyclical Time Series seq
def genCTS(R,phi):
    t = np.arange(1, 101)
    lmbda = np.pi / 25

    cyc = R * np.sin(lmbda * t + phi)

    return cyc
```

[17]:
```
R1 = np.random.uniform(-0.5, 1, 100)
phi = np.random.uniform(0, 1, 100)
simulated100 = genCTS(R1,phi)


R2 = np.random.uniform(-1, 1, 100)
stochasticAm = genCTS(R2,phi)


R3 = np.random.uniform(-0.5, 1, 100)
phi3 = np.random.standard_normal(100)
sinePart = genCTS(R3,phi3)
```

[21]:
```
fig, axs = plt.subplots(1, 3, figsize=(12,4),constrained_layout = True)
axs[0].plot(simulated100)
axs[1].plot(stochasticAm)
axs[2].plot(sinePart)
plt.show()
```

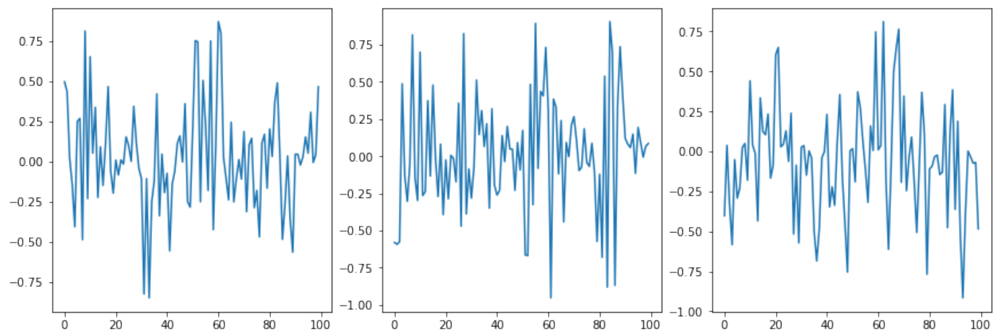

Figure 1: *100 simulated values of the cyclical time series (left panel), the stochastic amplitude (middle panel), and the sine part (right panel).*

Figure 1.6

1000 simulated values of iid $N(0, 1)$ noise (left panel) and a random walk with iid $N(0, 1)$ innovations (right panel).

[31]:
```
# iid noise
z = np.random.standard_normal(1000)
# random walk
rw = np.cumsum(z)
```

[32]:
```
fig, axs = plt.subplots(1, 2, figsize=(9,4),constrained_layout = True)
axs[0].plot(z)
axs[1].plot(rw)
```

[32]:
```
[<matplotlib.lines.Line2D at 0x7f9151f2b750>]
```

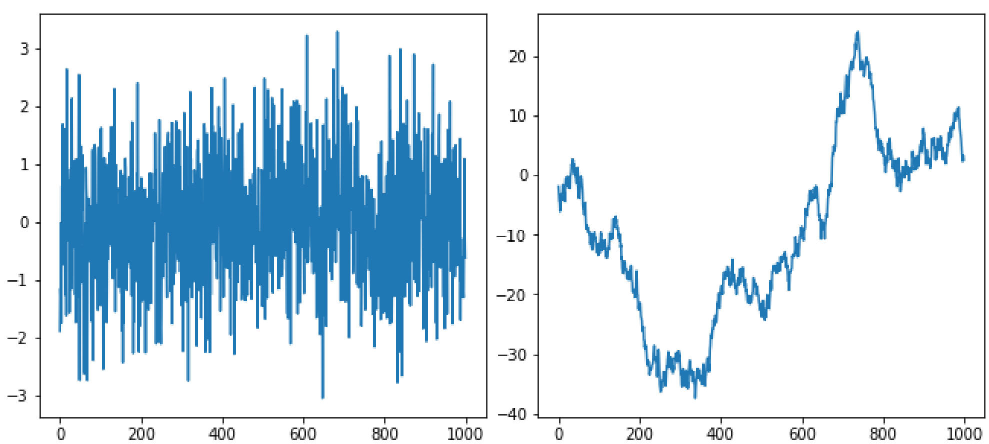

Figure 2: *1000 simulated values of iid N(0,1) noise (left panel) and a random walk with iid N(0,1) innovations (right panel).*