

# burger king warteschlange sim

April 1, 2022

```
[1]: import numpy as np
kubus = np.arange(1, 7, 1, dtype=int)

from queue import Queue # lets you generate a queue with maxlength

def runsim(customer=1,servants=2,minutes=20): # runs a "Warteschlangen"
    ↪simulation with a queue of length 3
    q = Queue(maxsize=3)
    q.put(1) # at start t=0 there is one customer in the queue
    servings = 0
    for k in range(len(kubus)*minutes):
        throw = np.random.choice(kubus)
        if throw > len(kubus)-customer and q.qsize() < 3: # customer enters queue?
            q.put(1)
        throw = np.random.choice(kubus)
        if throw > len(kubus)-servants and q.qsize() != 0: # customer is served?
            q.get()
            servings += 1
    return servings # how many meals have been served during the simulation
```

```
[2]: customers = 1
servants = 2
minutes = 20

l = []
for k in range(100): # runs the simulation 100 times
    l.append(runsim(customers,servants,minutes))
print(l)
```

```
[22, 19, 17, 16, 22, 15, 25, 26, 21, 17, 25, 16, 16, 21, 17, 20, 24, 23, 20, 16,
21, 17, 21, 16, 20, 17, 25, 24, 22, 19, 25, 23, 17, 22, 18, 18, 18, 21, 17, 15,
22, 17, 18, 24, 17, 12, 23, 21, 16, 22, 21, 30, 24, 18, 25, 20, 24, 27, 20, 18,
19, 17, 17, 15, 23, 17, 21, 24, 18, 22, 14, 20, 20, 23, 23, 20, 17, 15, 20, 16,
18, 19, 19, 17, 21, 18, 19, 16, 22, 25, 21, 16, 18, 23, 24, 18, 27, 18, 20, 19]
```

```
[3]: price = 10
wage = 30
```

```

for k in range(len(l)): # calculates the earnings = sales - wages
    l[k] = price * l[k] - servants * wage * minutes/60
print(l)

```

```

[200.0, 170.0, 150.0, 140.0, 200.0, 130.0, 230.0, 240.0, 190.0, 150.0, 230.0,
140.0, 140.0, 190.0, 150.0, 180.0, 220.0, 210.0, 180.0, 140.0, 190.0, 150.0,
190.0, 140.0, 180.0, 150.0, 230.0, 220.0, 200.0, 170.0, 230.0, 210.0, 150.0,
200.0, 160.0, 160.0, 160.0, 190.0, 150.0, 130.0, 200.0, 150.0, 160.0, 220.0,
150.0, 100.0, 210.0, 190.0, 140.0, 200.0, 190.0, 280.0, 220.0, 160.0, 230.0,
180.0, 220.0, 250.0, 180.0, 160.0, 170.0, 150.0, 150.0, 130.0, 210.0, 150.0,
190.0, 220.0, 160.0, 200.0, 120.0, 180.0, 180.0, 210.0, 210.0, 180.0, 150.0,
130.0, 180.0, 140.0, 160.0, 170.0, 170.0, 150.0, 190.0, 160.0, 170.0, 140.0,
200.0, 230.0, 190.0, 140.0, 160.0, 210.0, 220.0, 160.0, 250.0, 160.0, 180.0,
170.0]

```

```

[4]: sum(l)/len(l) # calculates the mean of the earnings of all simulation runs

```

```

[4]: 179.2

```

```

[5]: customers = 1
servants = 1
minutes = 20

l = []
for k in range(100): # runs the simulation 100 times
    l.append(runsim(customers,servants,minutes))
print(l)

```

```

[18, 16, 18, 17, 17, 13, 17, 14, 15, 15, 20, 11, 15, 10, 16, 17, 14, 18, 16, 19,
12, 17, 20, 21, 21, 13, 12, 13, 17, 14, 13, 22, 18, 19, 15, 17, 18, 12, 12, 10,
14, 15, 14, 19, 12, 13, 16, 16, 16, 14, 16, 14, 15, 8, 16, 15, 19, 16, 12, 13,
18, 14, 16, 15, 13, 15, 16, 12, 13, 14, 14, 20, 11, 18, 14, 14, 15, 14, 20, 17,
11, 14, 15, 17, 14, 13, 20, 11, 13, 16, 14, 14, 16, 13, 18, 14, 13, 14, 19, 15]

```

```

[6]: price = 10
wage = 30

for k in range(len(l)): # calculates the earnings = sales - wages
    l[k] = price * l[k] - servants * wage * minutes/60
print(l)

```

```

[170.0, 150.0, 170.0, 160.0, 160.0, 120.0, 160.0, 130.0, 140.0, 140.0, 190.0,
100.0, 140.0, 90.0, 150.0, 160.0, 130.0, 170.0, 150.0, 180.0, 110.0, 160.0,
190.0, 200.0, 200.0, 120.0, 110.0, 120.0, 160.0, 130.0, 120.0, 210.0, 170.0,
180.0, 140.0, 160.0, 170.0, 110.0, 110.0, 90.0, 130.0, 140.0, 130.0, 180.0,
110.0, 120.0, 150.0, 150.0, 150.0, 130.0, 150.0, 130.0, 140.0, 70.0, 150.0,
140.0, 180.0, 150.0, 110.0, 120.0, 170.0, 130.0, 150.0, 140.0, 120.0, 140.0,
150.0, 110.0, 120.0, 130.0, 130.0, 190.0, 100.0, 170.0, 130.0, 130.0, 140.0,

```

```
130.0, 190.0, 160.0, 100.0, 130.0, 140.0, 160.0, 130.0, 120.0, 190.0, 100.0,  
120.0, 150.0, 130.0, 130.0, 150.0, 120.0, 170.0, 130.0, 120.0, 130.0, 180.0,  
140.0]
```

```
[7]: sum(l)/len(l) # calculates the mean of the earnings of all simulation runs
```

```
[7]: 142.2
```