prob_probs2

August 30, 2024

[ ]:
```
# 2.16 a & b
import numpy as np
import pandas as pd

boats = pd.read_csv("data/sailboats.csv")
boats["fixed_cost"] = 30000
boats["variable_cost"] = 4800 * boats["nr_sailboats"]
boats["total_cost"] = boats["fixed_cost"] + boats["variable_cost"]

boat_mean = boats["nr_sailboats"].mean()
boat_std = boats["nr_sailboats"].std()
cost_mean = boats["total_cost"].mean()
cost_std = boats["total_cost"].std()

print(f"(a) The mean and standard deviation of the number of boats constructed␣
 ↪are {boat_mean} and {round(boat_std,3)}, respectively.\n(b) The mean and␣
 ↪standard deviation of the monthly cost of constructing the boats are␣
 ↪{cost_mean} and {round(cost_std,3)}, respectively.")
```

(a) The mean and standard deviation of the number of boats constructed are 4.0 and 1.309, respectively.
(b) The mean and standard deviation of the monthly cost of constructing the boats are 49200.0 and 6284.675, respectively.

[ ]:
```
# 2.16 c

boats["fixed_cost"] = 53000
boats["variable_cost"] = 4800 * boats["nr_sailboats"]
boats["total_cost"] = boats["fixed_cost"] + boats["variable_cost"]
cost_mean = boats["total_cost"].mean()
cost_std = boats["total_cost"].std()

print(f"(c) The new mean and standard deviation of the monthly cost of␣
 ↪constructing the boats are {cost_mean} and {round(cost_std,3)}, respectively.
 ↪")
```

(c) The new mean and standard deviation of the monthly cost of constructing the boats are 72200.0 and 6284.675, respectively.

[ ]:
```
# 2.16 d

boats["fixed_cost"] = 30000
boats["variable_cost"] = 7000 * boats["nr_sailboats"]
boats["total_cost"] = boats["fixed_cost"] + boats["variable_cost"]
cost_mean = boats["total_cost"].mean()
cost_std = boats["total_cost"].std()

print(f"(c) The new mean and standard deviation of the monthly cost of␣
 ↪constructing the boats are {cost_mean} and {round(cost_std,3)}, respectively.
 ↪")
```

(c) The new mean and standard deviation of the monthly cost of constructing the boats are 58000.0 and 9165.151, respectively.

[ ]:
```
# 2.22
umb = pd.read_csv("data/umbrella.csv")
umb["dep_sales"] = 17 * umb["department"]
umb["dis_sales"] = 9 * umb["outlet"]

dep_mean = umb["dep_sales"].mean()
dep_std = umb["dep_sales"].std()
dis_mean = umb["dis_sales"].mean()
dis_std = umb["dis_sales"].std()
corr = .7

c_mean = dep_mean + dis_mean
c_var = dep_std**2 + dis_std**2 + (2*corr*dep_std*dis_std)
c_std = np.sqrt(c_var)

print(f"The mean, variance, and standard deviation of the combined sales␣
 ↪revenue are {round(c_mean, 3)}, {round(c_var, 3)}, and {round(c_std, 3)},␣
 ↪respectively.")
```

The mean, variance, and standard deviation of the combined sales revenue are 3081.4, 1266773.4, and 1125.51, respectively.

[ ]:
```
#2.26 a

pJan = .6
pYear = .9
jointw = pJan * pYear
jointl = (1-pJan) * (1-pYear)
disjJ = (pJan * (1-pYear))
disjY = ((1-pJan) * pYear)
jointt = jointw + jointl
```

```
print(f"(a) The probability of the stock market movement in January and the␣
 ↪whole year being the same direction is {round(jointt, 3)}.")
```

(a) The probability of the stock market movement in January and the whole year being the same direction is 0.58.

[ ]:
```
# 2.26 b, c, d
from scipy.stats import binom

prb = binom.pmf(20, 20, jointt)
prc = binom.cdf(20, 20, jointt) - binom.cdf(14, 20, jointt)
prd = binom.cdf(20, 20, jointt) - binom.cdf(16, 20, jointt)

print(f"(b) The probability of the stock market moving in the same direction␣
 ↪for all 20 years is {prb:.10f}. \n(c) The probability of the stock market␣
 ↪moving in the same direction for at least 15 years is {prc}. \n(d) The␣
 ↪probability of the stock market moving in the same direction for at least 17␣
 ↪years is {prd}.")
```

(b) The probability of the stock market moving in the same direction for all 20 years is 0.0000185592.
(c) The probability of the stock market moving in the same direction for at least 15 years is 0.09218964285888731.
(d) The probability of the stock market moving in the same direction for at least 17 years is 0.010170370115304306.

[ ]:
```
# 2.26 e

pYear_given_Jan = jointw / pJan
pYear_given_nJan = disjY / (1-pJan)

print(f"(e) The probability of the stock market being up for the year given␣
 ↪that it moved up in January is {round(pYear_given_Jan, 3)}. \nThe␣
 ↪probability of the stock market being up for the year given that it moved␣
 ↪down in January is {round(pYear_given_nJan, 3)}.")
print("\nI would further want to test the dependence of annual performance on␣
 ↪January performance by collecting \nhistoric data samples for the two␣
 ↪January conditions (up or down) and performing a T-test on the mean \nannual␣
 ↪performance (yield) between the two samples.")
```

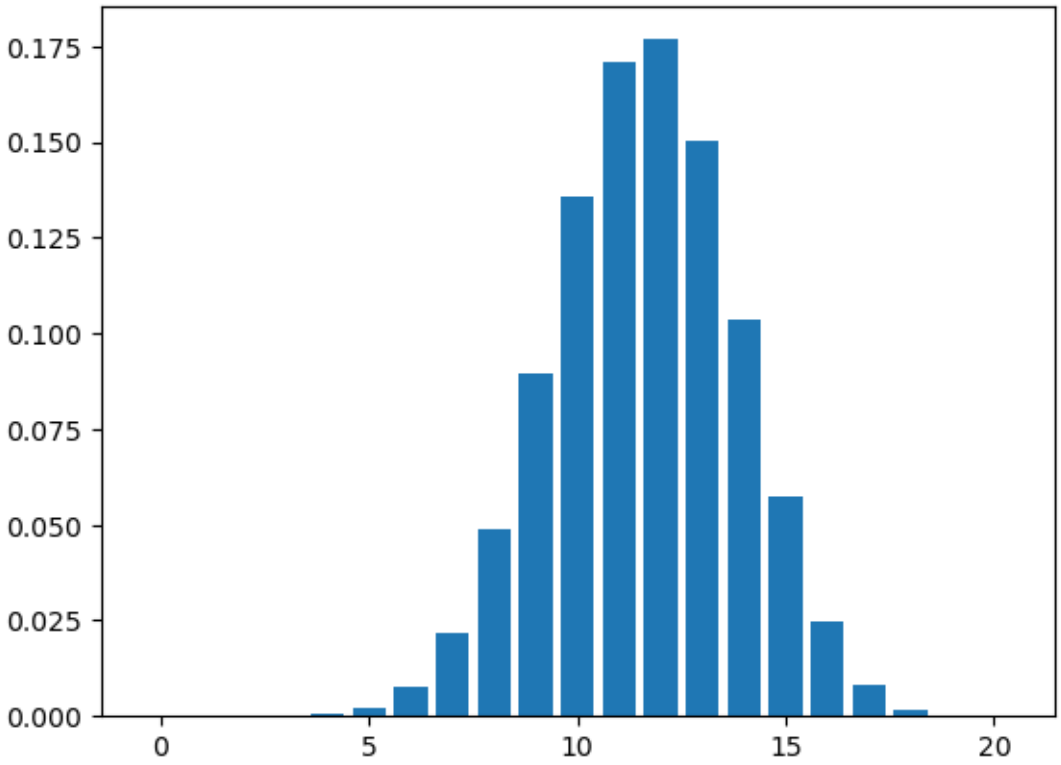
(e) The probability of the stock market being up for the year given that it moved up in January is 0.9.
The probability of the stock market being up for the year given that it moved down in January is 0.9.

I would further want to test the dependence of annual performance on January performance by collecting
historic data samples for the two January conditions (up or down) and performing

a T-test on the mean
annual performance (yield) between the two samples.

[ ]:
```
import matplotlib.pyplot as plt

n = 20
p = jointt
x = np.arange(0, 21)
y = binom.pmf(x, n, p)
plt.bar(x, y)
plt.show()
```



[ ]:
```
# AI Footnote: I used chatGPT to assist with this. See the chat link below:
https://chatgpt.com/share/e/f15b6341-8af6-4649-b54f-5b47a7ad44b1
```