Homework 01

Matheus M. S. Velloso

Import packages

```
import numpy as np
import matplotlib.pyplot as mplt
from scipy.stats import beta
```

Problems and solutions

Problem 1) Suppose the globe tossing data (Lecture 2, chapter 2), had turned out to be 3 water and 11 land. Construct the posterior distribution.

Solution:

We write the function to compute the posterior:

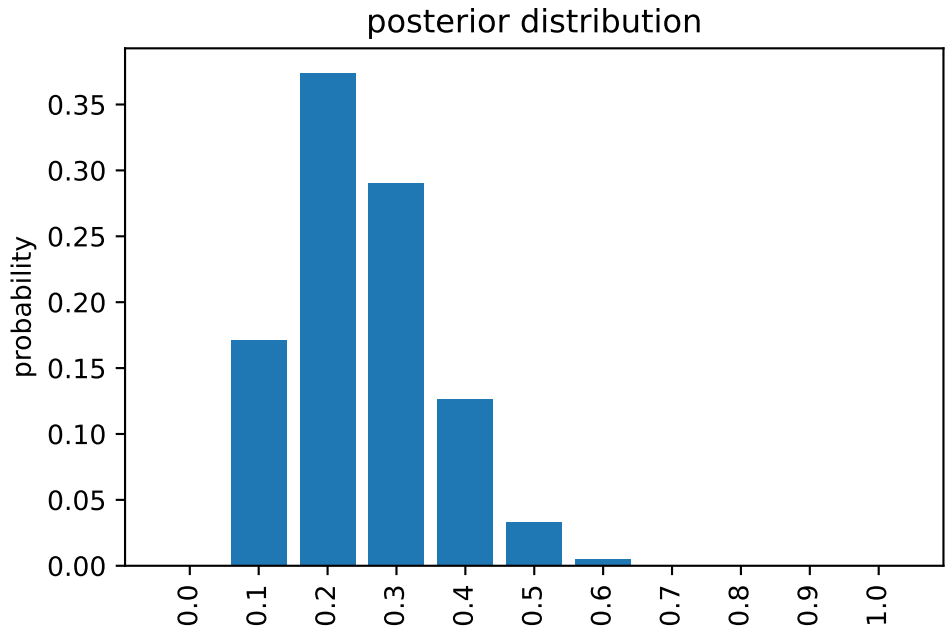
```
def compute_posterior(water, land, proportions):
    factor = proportions.size - 1  # counting factor
    ways = [(factor * p)**water * (factor - factor * p)**land for p in proportions]
    return np.array(ways)/np.sum(ways)  # return probabilities (posterior)
```

Next we compute the posterior for $W = 3$, $L = 11$, using a grid of 11 points (11-sided "globe"):

```
p = np.linspace(0, 1, 11)
posterior = compute_posterior(water=3, land=11, proportions=p)
posterior
```

```
array([0.00000000e+00, 1.70754638e-01, 3.73925212e-01, 2.90500651e-01,
       1.26342258e-01, 3.32112304e-02, 4.92967809e-03, 3.30622821e-04,
       5.70564594e-06, 3.96672811e-09, 0.00000000e+00])
```

```
fig, ax = mplt.subplots()
data = [f'{prop:.1f}' for prop in p]
ax.bar(data, posterior)
ax.set_title('posterior distribution')
ax.set_xticklabels(labels=data, rotation=90)
ax.set_ylabel('probability')
mplt.show()
```

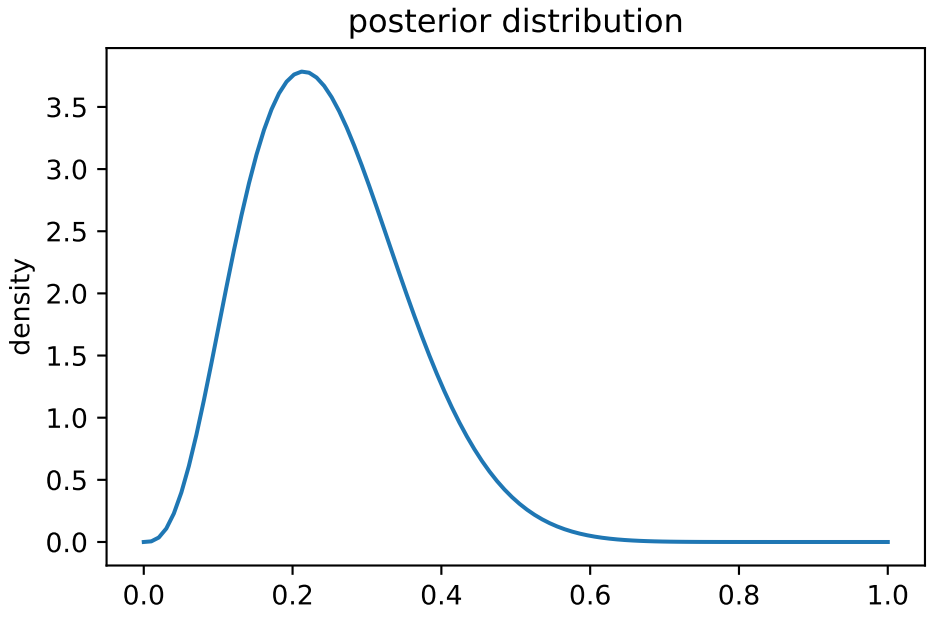


Alternatively, we could use the knoledge that posterior will be the Beta distribution with shape parameters $W + 1$ and $L + 1$ for each p

```
posterior_analytical = beta.pdf(np.linspace(0, 1, 100), 3 + 1, 11 + 1)
```

```
fig, ax = mplt.subplots()
ax.plot(np.linspace(0, 1, 100), posterior_analytical)
ax.set_title('posterior distribution')
ax.set_ylabel('density')
```

```
mplt.show()
```



Problem 2) Using the posterior distribution from 1), compute the posterior predictive distribution for the next 5 tosses of the same globe. Use the sampling method.

Solution:

We'll sample p from the posterior a thousand times, and simulate five globe tosses. Sampling from the beta:

```
posterior_samples = np.random.beta(3 + 1, 11 + 1, 1_000)
```

Next we code the generative model to simulate the tosses

```
def toss_globe(proportion, N):
    return np.random.choice(["W", "L"], size=N, p=[proportion, 1 - proportion])
```

and use it to toss the globe

```
predictive_posterior = [np.sum(toss_globe(p, 5)=='W') for p in posterior_samples]
```

Now we can plot the predictive posterior for the next 5 tosses

```
mplt.hist(predictive_posterior, bins=np.arange(11), alpha=0.3)
mplt.xlabel('water observations')
mplt.ylabel('count')
mplt.xticks(range(11))
mplt.grid(axis='y', alpha=0.75)
mplt.show()
```