102_hw2

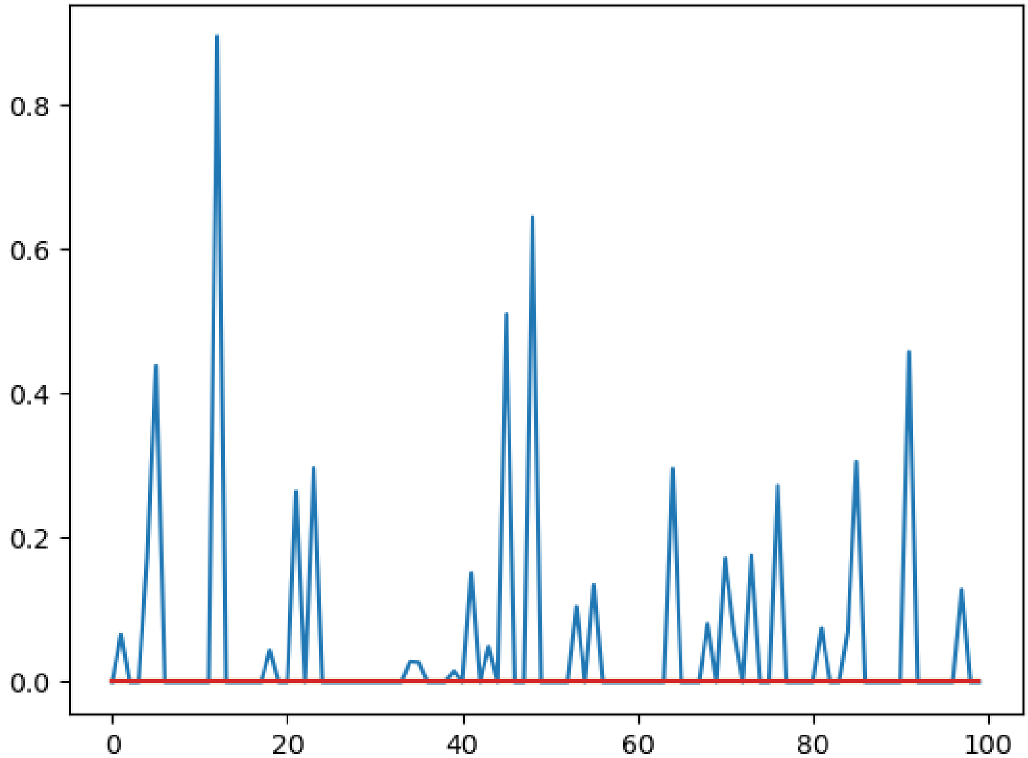
October 4, 2022

[155]:
```
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from scipy.stats import pareto
import pymc3 as pm
from scipy import stats
```

0.1 Question 1 The One with all the Beetles

[156]:
```
w = 15
np.random.seed(10)

y = np.random.uniform(low = 0, high = w , size = 100)
```

[157]:
```
alpha = 1
beta = 10
days = [1, 10, 50, 100]
for i in days:
    curr_x = np.random.uniform(low = 0, high = w, size = 100)
    plt.plot(pareto.pdf(x = curr_x, b = beta, scale = beta**alpha))
    beta = max(beta, max(curr_x))
    alpha = alpha + i
```



[ ]:

```

```

0.2 Question 2 Bayesian Fidget Spinners

$$z_i \sim Bernoulli(\pi)$$

$$q_k \sim Beta(\alpha_k, \beta_k)$$

$$x_i | z_i, q_0, q_1 \sim Geometric(q_{z_i})$$

0.2.1 b)

i) $\pi < \frac{1}{2}$, indicating Nat thinks factory 0 produces more boxes than factory 1

ii) Factory 0 has lower rate of defect comparing to factory 1

0.2.2 c)

[158]:

```
import numpy as np
np.random.seed(5)

N = 100
pi = 0.45
```

```
q0 = 0.05
q1 = 0.18

n1 = int(N * pi)
n0 = N - n1

y_obs = np.zeros(N)
y_obs[:n0] = np.random.geometric(q0, size=n0)
y_obs[n0:] = np.random.geometric(q1, size=n1)

print(y_obs)
```

```
[ 5. 40.  5. 49. 14. 19. 29. 15.  7.  5.  2. 27. 12.  4. 42.  7. 11.  7.
 20. 17. 18.  7.  7.  6.  8.  4.  4. 65. 63.  5.  1.  5. 24. 30.  1. 17.
  1. 15. 20. 83.  6. 32. 40. 50.  1. 13. 78. 10. 33. 16. 29. 13.  1.  2.
  3.  2. 17.  6.  9.  5.  6.  9. 14. 13.  9.  1.  3.  1.  4.  9.  1.  2.
  3.  2.  8.  3.  6.  4.  5. 15. 15.  1. 10.  3.  1.  3.  4. 17.  3.  1.
 10.  1.  4. 10.  4.  3.  9.  1.  5.  3.]
```

[159]:
```
alphas = [1, 5]
betas = [5, 1]
pi = 0.45

with pm.Model() as model:
    z = pm.Bernoulli(p = pi, shape = (100, 2))

    # Hint: you should use the shape= parameter here so that
    # q is a PyMC3 array with both q0 and q1.
    q = pm.Beta("q", alpha = alphas, beta = betas, shape = (100,2))

    # Hint: it may be useful to use "fancy indexing" like we did in class.
    # See below for an example
    X = pm.Geometric("X", p = q)

    trace = ....

# FANCY INDEXING

my_binary_array = np.array([0, 0, 1, 1, 0, 1])
my_real_array = np.array([0.27, 0.34])
print(my_real_array[my_binary_array])
```

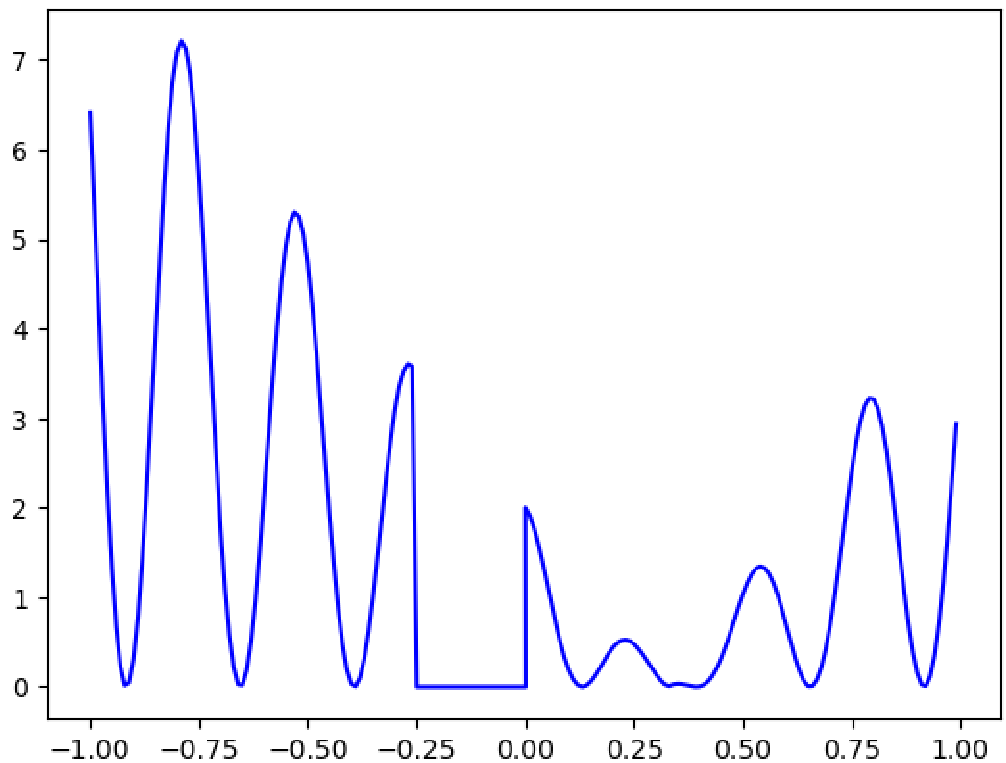
```
  Cell In [159], line 16
    trace = …
            ^
SyntaxError: invalid syntax
```

0.2.3 Question 3 Rejection Sampling

a) Plot $g(x)$

[160]:
```
fig, ax = plt.subplots()
x_1 = np.arange(-1, -0.25, 0.01)
x_2 = np.arange(0,1,0.01)
x_3 = np.arange(-0.249, 0.01, 0.01)
g_1 = np.cos(12*x_1)**2 * abs(x_1**3 + 6*x_1 -2)
g_2 = np.cos(12*x_2)**2 * abs(x_2**3 + 6*x_2 -2)
g = np.concatenate([g_1,[0]*len(x_3), g_2])
x = np.concatenate([x_1, x_3, x_2])
ax.plot(x,g, 'b')
```

[160]:
```
[<matplotlib.lines.Line2D at 0x7fbd82b88ca0>]
```



[161]:
```
max(g)
```

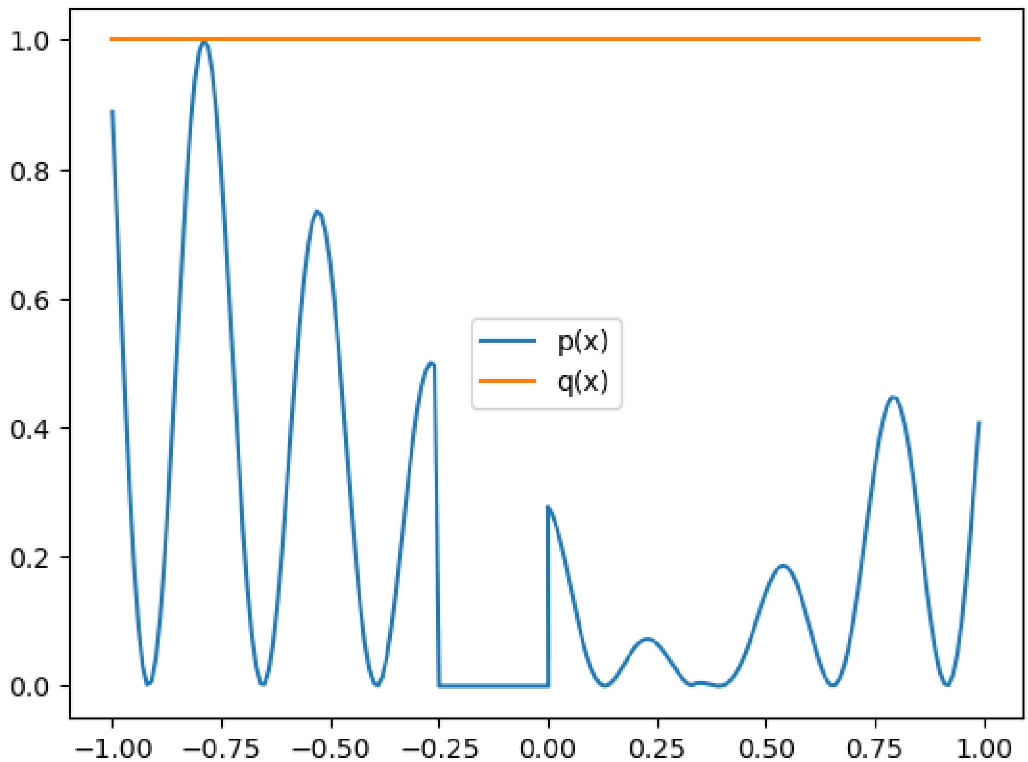
[161]:
```
7.2110044499935935
```

$q \sim Uniform(-1, 1)$, $M \leq \frac{1}{max(g(x))} = \frac{1}{7.211}$

b) plot proposal $q(x)$ and $p(x)$, estimate $\hat{c}$

[162]:
```
fig, ax = plt.subplots()
plt.plot(x, (1/7.211)*g, label = 'p(x)')
plt.plot(np.arange(-1, 1,0.01), len(np.arange(-1, 1,0.01))*[1], label = 'q(x)')
plt.legend()
```

[162]: <matplotlib.legend.Legend at 0x7fbd848c7e50>



$$\frac{n}{N} \approx \frac{\int_{-1}^{1} p(x)dx}{\int_{-1}^{1} q(x)dx} = M \int_{-1}^{1} g(x)dx = \frac{M}{c} \int f(x)dx = \frac{M}{c}$$

As a result, $\hat{c} = \frac{NM}{n}$

c) Rejection Sampling

[181]:
```
def target (x):
    return np.cos(12*x)**2 * abs(x**3 + 6*x -2)
```

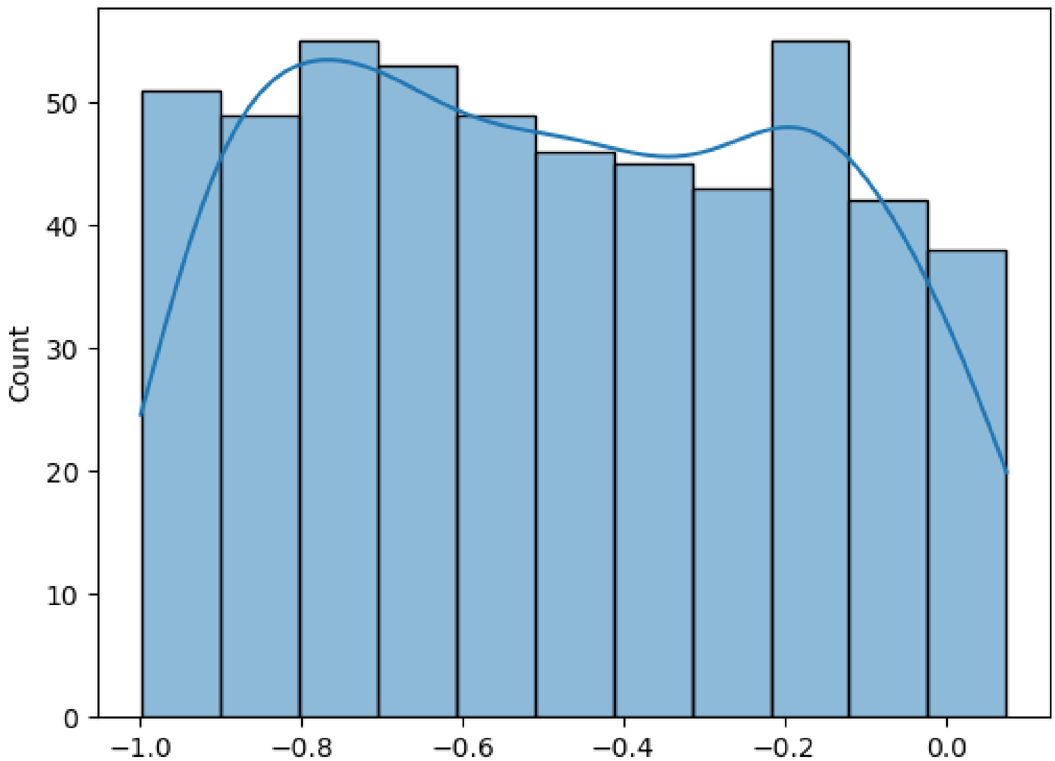
[174]:
```
def uniform_sampling_dist(t):
    return stats.uniform.pdf(t, -1, 1)
```

[190]:
```python
def rejection_sample_uniform(num_samples=1000):
    proposals = np.random.uniform(low=-1, high=1, size=num_samples)
    target  = [target(i)*(1/7.211) for i in proposals]

    num_accept = np.sum(accept)
    print('Accepted %d out of %d proposals' % (num_accept, num_samples))
    return proposals[accept]
```

[191]:
```python
samples = rejection_sample_uniform(num_samples=1000)

# Plot a true histogram (comparable with density functions) using density=True
sns.histplot(x = samples, kde=True)
```

```
Accepted 526 out of 1000 proposals
```

[191]: <AxesSubplot:ylabel='Count'>

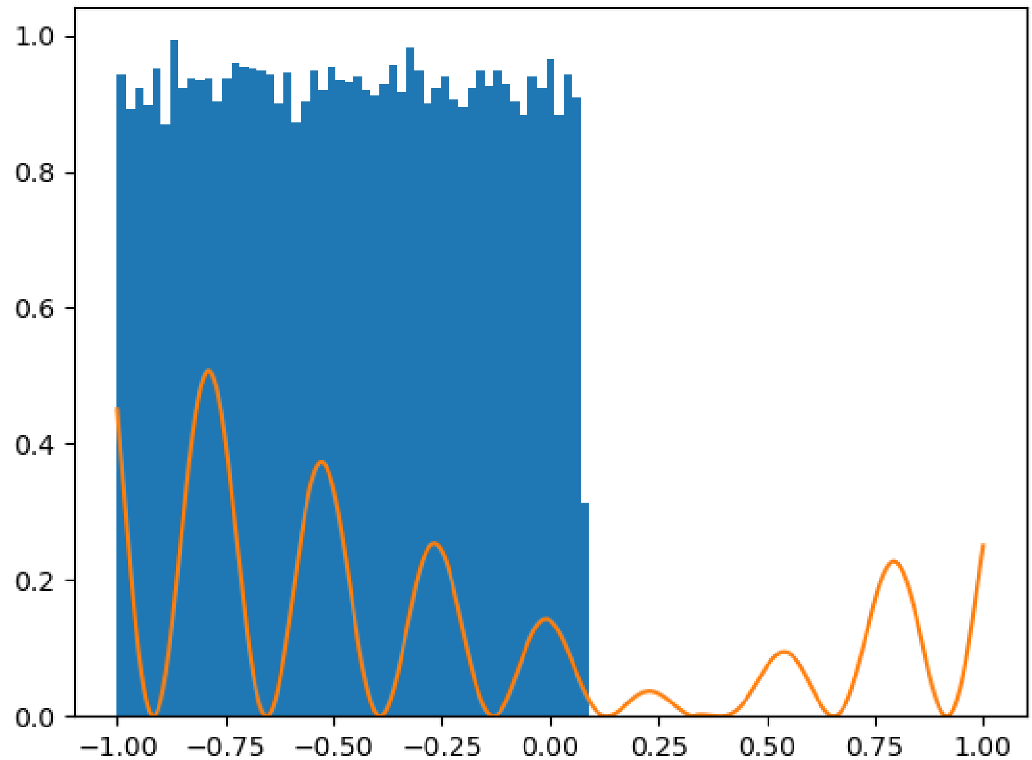


[194]:
```python
samples = rejection_sample_uniform(num_samples=100000)

# Plot a true histogram (comparable with density functions) using density=True
plt.hist(samples, bins=np.linspace(-1, 1, 100), density=True)
t = np.linspace(-1, 1, 1000)
```

```
# Where did this magic number 0.36 come from? What happens if you change it?
plt.plot(t, target(t)*(1/14.211))
```

```
Accepted 54069 out of 100000 proposals
```

[194]: [<matplotlib.lines.Line2D at 0x7fbd80c46ac0>]



[ ]:

[189]:
```
proposals = np.random.uniform(low=-1, high=1, size=1000)
target_proposals  = target(proposals)*(1/7.211)
```

[ ]:

[ ]:

[109]:
```
target(proposals)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In [109], line 1
----> 1 target(proposals)
```

HW2 Q1

a) Given $x \sim \text{Uniform}(0, w)$:

$$f_1(x|w) = \frac{1}{w} \mathbb{1}[x \le w]$$

$$f_n(x_1, \dots x_n | w) = \frac{1}{w^n} \prod_{i=1}^{n} \mathbb{1}[x_i \le w]$$

$$= \frac{1}{w^n} \mathbb{1}[\max_i x_i \le w]$$

b) $\text{Lik}(w) = \frac{1}{w^n} \mathbb{1}[\max_i x_i \le w]$

$\max_i x_i > w$ $\quad$ $\text{Lik}(x_1 \dots x_n; w) = 0$

$\max_i x_i \le w$ $\quad$ $\text{Lik}(x_1 \dots x_n; w) = \frac{1}{w^n}$ $\Rightarrow$ Mono-decreasing function

So maximum $w = \max_i x_i$

c) $$p'(w | x_1 \dots x_n) = \frac{f_n(x_1 \dots x_n | w) p(w)}{\int_{w'} f_n(x_1 \dots x_n | w') p(w')}$$

Numerator: $$f_n(x_1 \dots x_n | w) p(w) = \frac{1}{w^n} \mathbb{1}[\max_i x_i \le w] \cdot \frac{\alpha \beta^\alpha}{w^{\alpha+1}} \mathbb{1}(w > \beta)$$

$$= \frac{\alpha \beta^\alpha}{w^{n+\alpha+1}} \mathbb{1}[\max_i x_i \le w] \mathbb{1}(w > \beta)$$

$$= \frac{\alpha \beta^\alpha}{w^{n+\alpha+1}} \mathbb{1}[\max_i w \ge \max(x_i, \beta)]$$

$$\int_{w'} f_n(x_1 \dots x_n | w') p(w') = \int_0^\infty \frac{\alpha \beta^\alpha}{w'^{n+\alpha+1}} \mathbb{1}[w' \ge \max[\max_i x_i, \beta]] dw'$$

$\alpha \beta^\alpha$ is constant

$$\hookrightarrow \alpha \beta^\alpha \int \frac{\mathbb{1}[w' \ge \max[\max_i x_i, \beta]]}{w'^{n+\alpha+1}} dw'$$

Let $m = \max(\max_i x_i, \beta)$ $\Rightarrow$ $$\alpha \beta^\alpha \left[\frac{w'^{-\alpha-n}}{-\alpha-n} \right] \Bigg|_m^\infty = \alpha \beta^\alpha \frac{m^{-\alpha-n}}{\alpha+n}$$

$$P'(w \mid x_1 \dots x_n) = \frac{\dfrac{\alpha\beta^\alpha}{w^{n+\alpha+1}} m}{\alpha\beta^\alpha \dfrac{m^{-\alpha-n}}{\alpha+n}} = \frac{(\alpha+n)\, m^{(\alpha+n)}}{w^{(\alpha+n+1)}} \mathbb{1}[w > m]$$

set $\alpha + n = \delta \Rightarrow \dfrac{\delta m^\delta}{w^{\delta+1}} \mathbb{1}[w > m]$

$\hookrightarrow$ Pareto(δ, m)

d) prior : $w \sim$ Pareto(α, β)

posterior $\sim$ Pareto(δ, m)

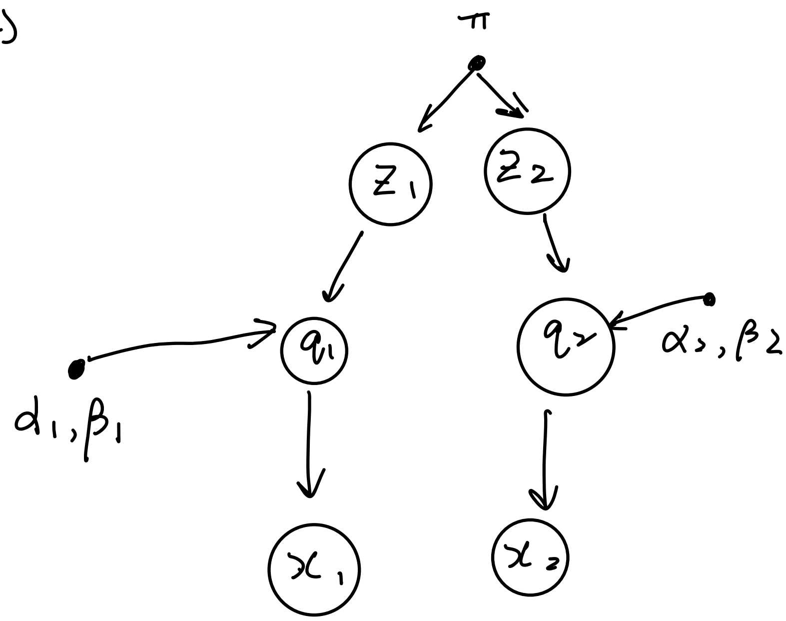
where $\delta = \alpha + n$ $\quad m = \max(\max_i x_i, \beta)$

α : track # of samples seen

m : record/update maximum length seen

Question 2

a)



c) i)

$$P[\text{Bin}(50, 0.45) \leq 50]$$

$$\Rightarrow \sum_{i=0}^{50} \binom{100}{i} (0.45)^{i} (0.55)^{100-i} = 0.87$$

d)

$$q_0 \sim \text{Beta}\left(\alpha_0 + n_0,\ \beta_0 - n_0 + \sum_{i: z_i = 0} x_i\right)$$

$$q_1 \sim \text{Beta}\left(\alpha_1 + n_1,\ \beta_1 - n_1 + \sum_{i: z_i = 1} x_i\right)$$

$$n_0 = \sum_{i=0}^{n} \mathbb{1}_{z_i = 0}$$

$$n_1 = \sum_{i=0}^{N} \mathbb{1}_{z_i = 1}$$