Problem 3

Code

```
import numpy as np

from scipy.stats import norm


with open("LAozone.data", "r") as file:

    lines = file.readlines()


#Parse header and data lines

header = lines[0].strip().split(",")

data = []

for line in lines[1:]:

    split_line = line.strip().split(",")

    if "" not in split_line:  # skip incomplete rows

        data.append([float(val) for val in split_line])


data = np.array(data)


#Identify columns

col_idx = {col: i for i, col in enumerate(header)}

ozone = data[:, col_idx["ozone"]]

wind = data[:, col_idx["wind"]]

humidity = data[:, col_idx["humidity"]]

temp = data[:, col_idx["temp"]]

vis = data[:, col_idx["vis"]]
```

```
#X matrix and y vector
n = len(ozone)
X = np.column_stack((np.ones(n), wind, humidity, temp, vis))  # Intercept term + 4 features
y = ozone
p = X.shape[1]

#Prior info
mu = np.full(p, 0.2)
S2 = np.full((p, p), 0.1)
np.fill_diagonal(S2, 1)

sigma2 = 0.3

#Compute posterior
S2_inv = np.linalg.inv(S2)
Sigma_post = np.linalg.inv((1 / sigma2) * X.T @ X + S2_inv)
w_post = Sigma_post @ ((1 / sigma2) * X.T @ y + S2_inv @ mu)

#Print Bayes estimator
print("Bayes point estimate for w:")
for i, val in enumerate(w_post):
   print(f"w{i} = {val:.4f}")

#95% credible interval for 'wind' (index 1)
wind_idx = 1
wind_mean = w_post[wind_idx]
```

```
wind_std = np.sqrt(Sigma_post[wind_idx, wind_idx])
z = norm.ppf(0.975)

ci_lower = wind_mean - z * wind_std
ci_upper = wind_mean + z * wind_std

print(f"\n95% Bayes Confidence Interval for wind coefficient: ({ci_lower:.4f}, {ci_upper:.4f})")
```

A. Bayes Point Estimate of w:

Bayes point estimates for the regression coefficients:

w_0 (Intercept): -13.5953

w_1 (Wind): -0.0554

w_2 (Humidity): 0.0706

w_3 (Temperature): 0.3716

w_4 (Visibility): -0.0112

B. 95% Bayes Confidence Interval for Wind Coefficient:

(-0.0827, -0.0280)