Density Estimation

EN5422/EV4238 | Fall 2023

w06_density_2.pdf

(Week 6 – 2/2)

Contents

1 Bivariate Gaussian/Normal Distribution

A bivariate normal random variable ($\mathbf{X} = X_1, X_2$) has the joint pdf:

$$f(\mathbf{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{p}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{x}-\mu)^{\intercal}\Sigma^{-1}(\mathbf{x}-\mu)\right]$$

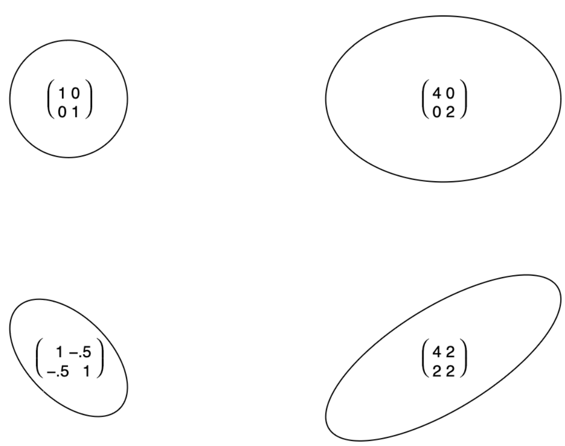
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

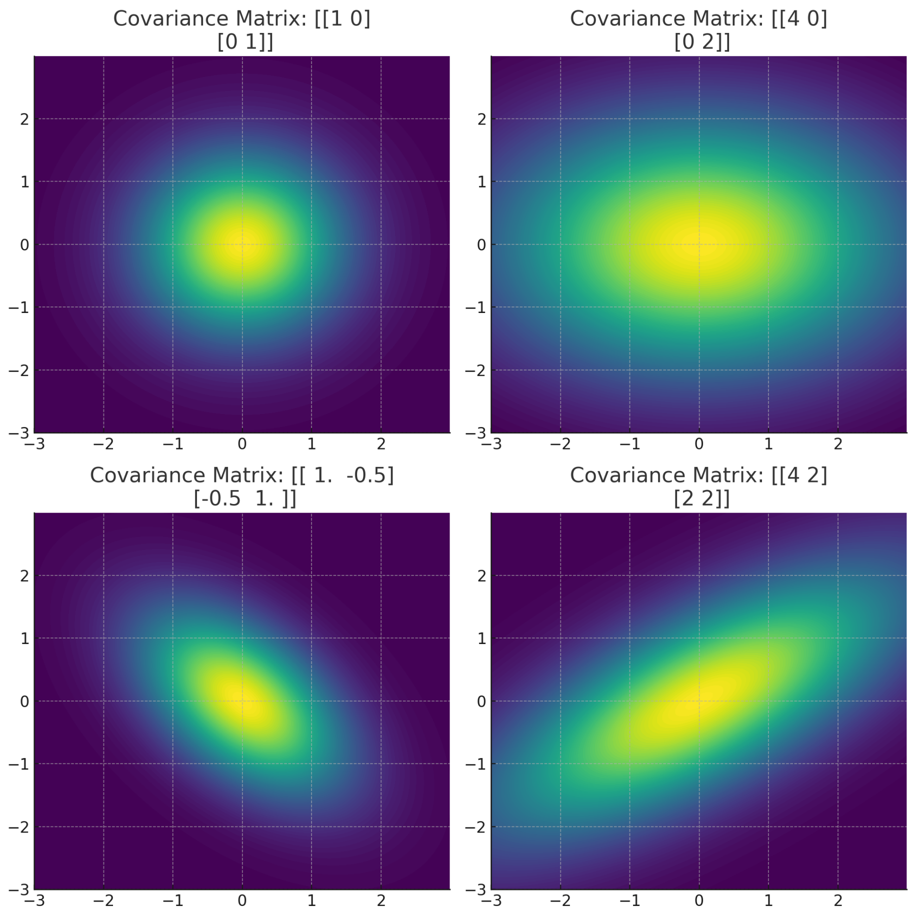
$$\Sigma = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Var(X_2) \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}$$

- $|\Sigma|$ is the *determinant* of the variance-covariance matrix Σ
  - $\det(\Sigma) = \prod_{j=1}^{2} \lambda_i$ where λ are eigenvalues of Σ

$$\left|\begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}\right| = \sigma_1^2\sigma_2^2 - \sigma_{12}^2$$

- Σ controls the *orientation*, *shape*, and *volume* of the density contours





Note
1. **Top Left (Subplot 1,1):** This represents independent variables with equal variance. 2. **Top Right (Subplot 1,2):** Here, the variables are still independent but have different variances. 3. **Bottom Left (Subplot 2,1):** This matrix introduces a negative correlation between the variables. 4. **Bottom Right (Subplot 2,2):** This represents variables with a positive correlation and different variances. 5. **Covariance:** $$Cov(X,Y) = E[(X - E[X])(Y - E[Y])]$$ $$Cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})$$

1.1 Independence and EDF

Your Turn #1: Number of Parameters
How may parameters need to be estimated in a bivariate Gaussian model?

- It is a special property of the multi-variate Gaussian distribution that if the random variables are uncorrelated, then they are also independent.
  - So, if the off-diagonal terms are zero ($\sigma_{12} = 0$), then the elements of the random vector are *independent*.

Note: Independence vs. Uncorrelation

Bivariate Uniform Distribution Over a Circle:

- Imagine you have two random variables, X and Y, representing the coordinates of points uniformly distributed inside a circle centered at the origin with radius 1. Intuitively, if you know the value of X (say, X=0.5), you immediately know that Y must lie within a certain range (between -$\sqrt{0.75}$ and $\sqrt{0.75}$ in this case) to still be inside the circle. This dependence is due to the geometric shape of the circle.
- Despite this dependence, the covariance between X and Y can be zero, which means they are uncorrelated. This happens because the positive and negative values of Y for a given X (and vice versa) balance out in the calculation of covariance.

- **Bernoulli Random Variables X and Y=1−X:**

- Consider a simple scenario where X is a Bernoulli random variable that takes the value 1 with probability p and 0 with probability 1−p. Define another variable Y=1−X. Clearly, Y is also a Bernoulli random variable but with opposite outcomes to X.
- If X=1, then Y must be 0; and if X=0, then Y must be 1. This shows a clear dependence between X and Y. However, if p=0.5, the covariance between X and Y is zero, indicating that they are uncorrelated. This is because the covariance formula, which involves the expectation of XY minus the product of the expectations of X and Y, turns out to be zero due to the symmetric nature of the distribution around p=0.5

- For bivariate data, the random vector $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]$ has two elements.
  - Marginal pdf $f_1(X_1), f_2(X_2)$
  - Joint pdf $f_{12}(\mathbf{X}) = \mathbf{f_{2|1}}(\mathbf{X_2}|\mathbf{X_1})\mathbf{f_1}(\mathbf{X_1}) = \mathbf{f_{1|2}}(\mathbf{X_1}|\mathbf{X_2})\mathbf{f_2}(\mathbf{X_2})$
  - If X_1 and X_1 are *independent*, then $f_{12}(\mathbf{X}) = \mathbf{f_1}(\mathbf{X_1})\mathbf{f_2}(\mathbf{X_2})$

- If the data are independent and follow a Gaussian distribution then

$$\mathbf{X} \sim \mathcal{N}(\mu, \Sigma) \quad \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$$

$$f_{12}(\mathbf{X}) = f_1(X_1) f_2(X_2) = \left[\frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{(X_1-\mu_1)}{\sigma_1}\right)^2}\right]\left[\frac{1}{\sigma_2\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{(X_2-\mu_2)}{\sigma_2}\right)^2}\right]$$

$$= \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{1}{2}\left[\left(\frac{(X_1-\mu_1)}{\sigma_1}\right)^2 + \left(\frac{(X_2-\mu_2)}{\sigma_2}\right)^2\right]}$$

- So, if you *choose* to use a Gaussian and *choose* independence, then you can estimate the parameters for each marginal density independently.
  - Will be important knowledge for multivariate KDE and Gaussian Mixture Models (GMMs)
  - And you only have four parameters to estimate: (μ_1, σ_1), (μ_2, σ_2)

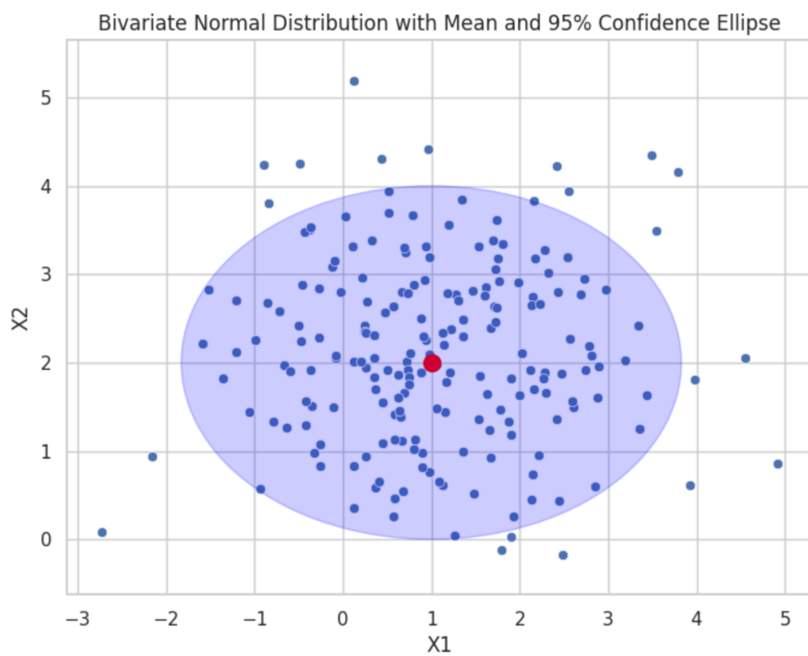
1.2 Parameter Estimation

Just like a one-dimensional Gaussian, we need to estimate the equivalent to the *mean* and *variance.*

- For bivariate data, the mean has two elements (i.e., two element vector)
- For bivariate data, the variance has four elements (i.e., two by two symmetric matrix)
  - Note there are only 3 *unique* values since the off-diagonal are equal

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.patches import Ellipse

# Function to create an ellipse representing the confidence interval
def plot_cov_ellipse(cov, pos, nstd=2, ax=None, **kwargs):
    if ax is None:
        ax = plt.gca()

    vals, vecs = np.linalg.eigh(cov)
    order = vals.argsort()[::-1]
    vals, vecs = vals[order], vecs[:, order]
    theta = np.degrees(np.arctan2(*vecs[:, 0][::-1]))
    width, height = 2 * nstd * np.sqrt(vals)
    ellipse = Ellipse(xy=pos, width=width, height=height, angle=theta,
**kwargs)

    ax.add_patch(ellipse)
    return ellipse

# Sample size and parameters
n = 200
```

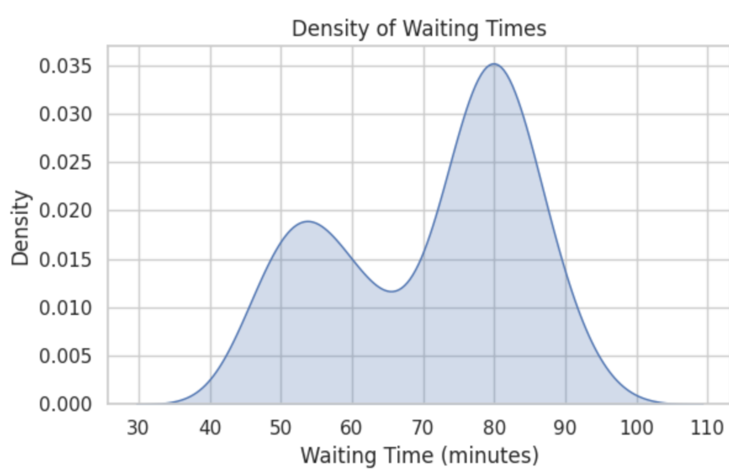
```
Sigma = np.array([[2, 0], [0, 1]])
mu = np.array([1, 2])

# Generate samples
X = np.random.multivariate_normal(mu, Sigma, n)
df = pd.DataFrame(X, columns=['X1', 'X2'])

# Plot
sns.set(style="whitegrid")
plt.figure(figsize=(8, 6))
sns.scatterplot(x='X1', y='X2', data=df)
plt.scatter(mu[0], mu[1], color="red", s=100) # plot the mean (corrected
index)
plot_cov_ellipse(Sigma, mu, nstd=2, alpha=0.2, color='blue')
plt.axis('equal')
plt.title("Bivariate Normal Distribution with Mean and 95% Confidence
Ellipse")
plt.show()
```



2 Multivariate Kernel Density Estimation

Your Turn #2: The Return to Old Faithful

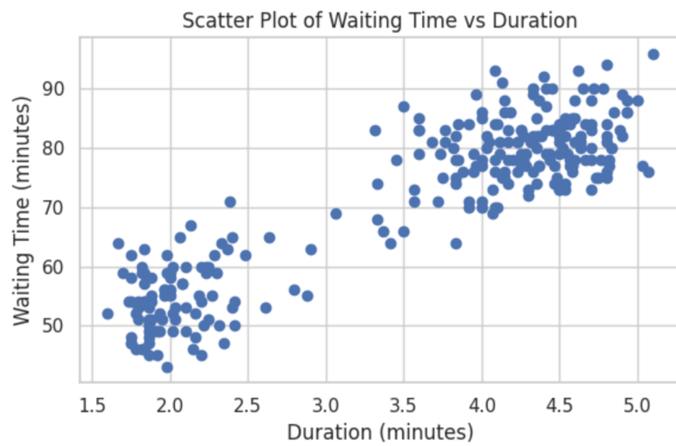
```
import statsmodels.api as sm
import matplotlib.pyplot as plt
import seaborn as sns

# Load the Old Faithful data
faithful_dataset = sm.datasets.get_rdataset("faithful", "datasets")
faithful_data = faithful_dataset.data
wait = faithful_data['waiting'].values
duration = faithful_data['eruptions'].values

# Create a figure with two subplots
fig, ax = plt.subplots(1, 2, figsize=(12, 6))

# Density plot for waiting times
sns.kdeplot(wait, ax=ax[0], fill=True)
ax[0].set_title('Density of Waiting Times')
ax[0].set_xlabel('Waiting Time (minutes)')
ax[0].set_ylabel('Density')
```



Did I forget to mention what we have additional information about old faithful eruption? It turns out that we also have information on the *duration of the previous eruption*.

```
# Scatter plot of waiting time vs duration
ax[1].scatter(duration, wait)
ax[1].set_title('Scatter Plot of Waiting Time vs Duration')
ax[1].set_xlabel('Duration (minutes)')
ax[1].set_ylabel('Waiting Time (minutes)')
```

```
# Show the plots
plt.tight_layout()
plt.show()
```



1. What patterns do you see?
2. Think about how to estimate this *bivariate* density.
3. Would a 2D Gaussian be appropriate?

There are three primary approaches to multivariate (d dimensional) KDE:

1. Multivariate kernels
   - (e.g., $K(u) = N(\mathbf{0},\ \Sigma)$)

$$\hat{f}(x) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}n}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}(x-x_i)^{\intercal}\Sigma^{-1}(x-x_i)\right)$$

   - Let $\Sigma = h^2 A$ where $|A| = 1$, thus $|\Sigma| = h^{2d}$

$$\hat{f}(x) = \frac{1}{(2\pi)^{\frac{d}{2}}h^d n}\sum_{i=1}^{n}\exp\left(-\frac{1}{2}(x-x_i)^{\intercal}\Sigma^{-1}(x-x_i)\right)$$

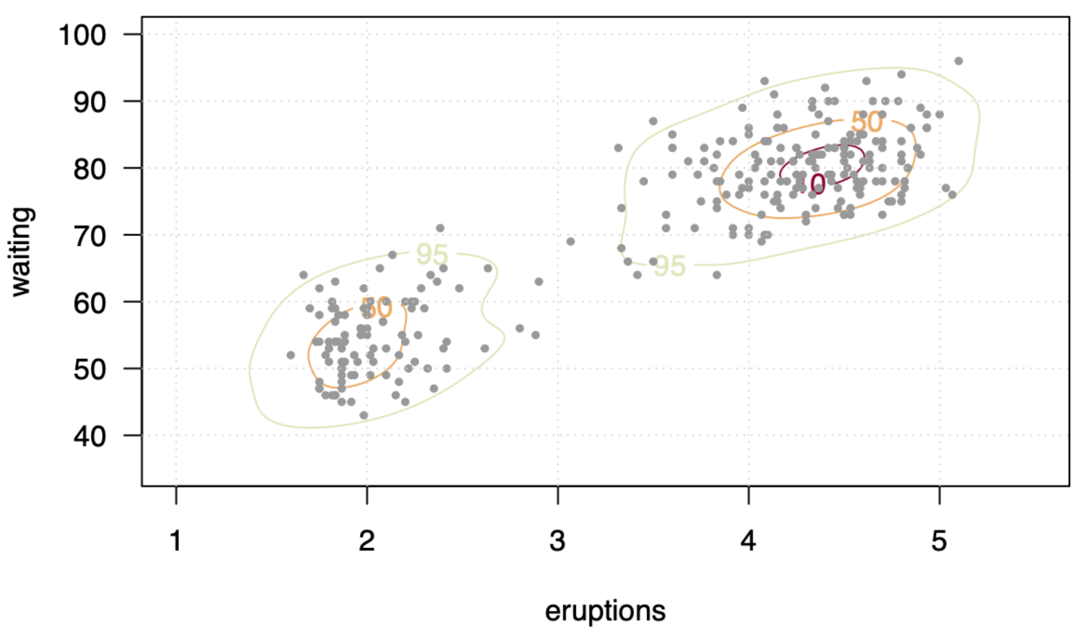
2. Product Kernels
    - Specifies $\Sigma = h^2 I_d$.
    - The kernel is independent Gaussian.
    - Estimate bandwidth for each dimension.

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\left(\prod_{j=1}^{d} K_{h_j}\left(x_j - x_{ij}\right)\right)$$

3. Independence
    - Estimate a KDE separately for each dimension.
    - Multiple the resulting density estimates.
    - This has strongest constraints (least complex).

$$\hat{f}(x) = \prod_{j=1}^{d} \hat{f}_j(x) = \prod_{j=1}^{d}\left(\frac{1}{n}\sum_{i=1}^{n} K_{h_j}(x_j - x_{ij})\right)$$

2.1 Multivariate KDE



$$H = \begin{bmatrix} 0.0601 & 0.511 \\ 0.5110 & 12.408 \end{bmatrix}$$

Above is the *unconstrained* Gaussian kernel.

- The Kernel is a MV Normal, $K(X; H)$, with variance-covariance matrix

$$\Sigma = H = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix}$$

- The diagonal terms correspond to the **variances** in each dimension.
  - Note: take square root to compare with univariate bandwidth
- The off-diagonal term corresponds to the *correlation*: $H_{12} = \rho h_1 h_2$, where $h_i = \sqrt{H_{ii}}$
  - $\rho = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$

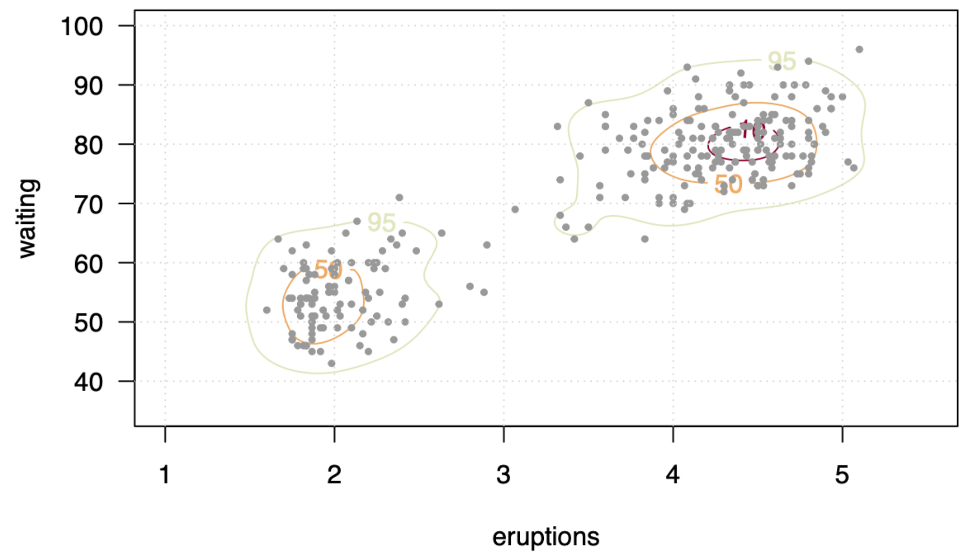
Note
• When we say the Gaussian kernel is "unconstrained," it means there are no restrictions on the shape or orientation of this multivariate normal distribution. This flexibility is determined by the variance-covariance matrix (denoted as H or Σ in your description). • The matrix $H = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{22} \end{bmatrix}$ defines how the kernel is shaped and oriented in the data space. Here, H_{11} and H_{22} are the variances in each dimension, and H_{12} (equal to H_{21} for a symmetric matrix) represents the covariance between the dimensions. • An "unconstrained" Gaussian kernel means that all these elements can vary freely. There are no specific constraints like forcing it to be a circular shape (which would be the case if $H_{11} = H_{22} = 0$) • This flexibility allows the kernel to adapt to the data's structure. For example, if the data points are more spread out along one axis than another, or if there's a correlation between variables, an unconstrained Gaussian kernel can accommodate this by adjusting its shape and orientation accordingly. • The choice of H (or Σ) determines how broad or narrow, and in which direction, the smoothing occurs at each data point.

2.1.1 Multivariate KDE

If the off-diagonal/correlation is zero, then kernel reduces to the product of two univariate kernels:

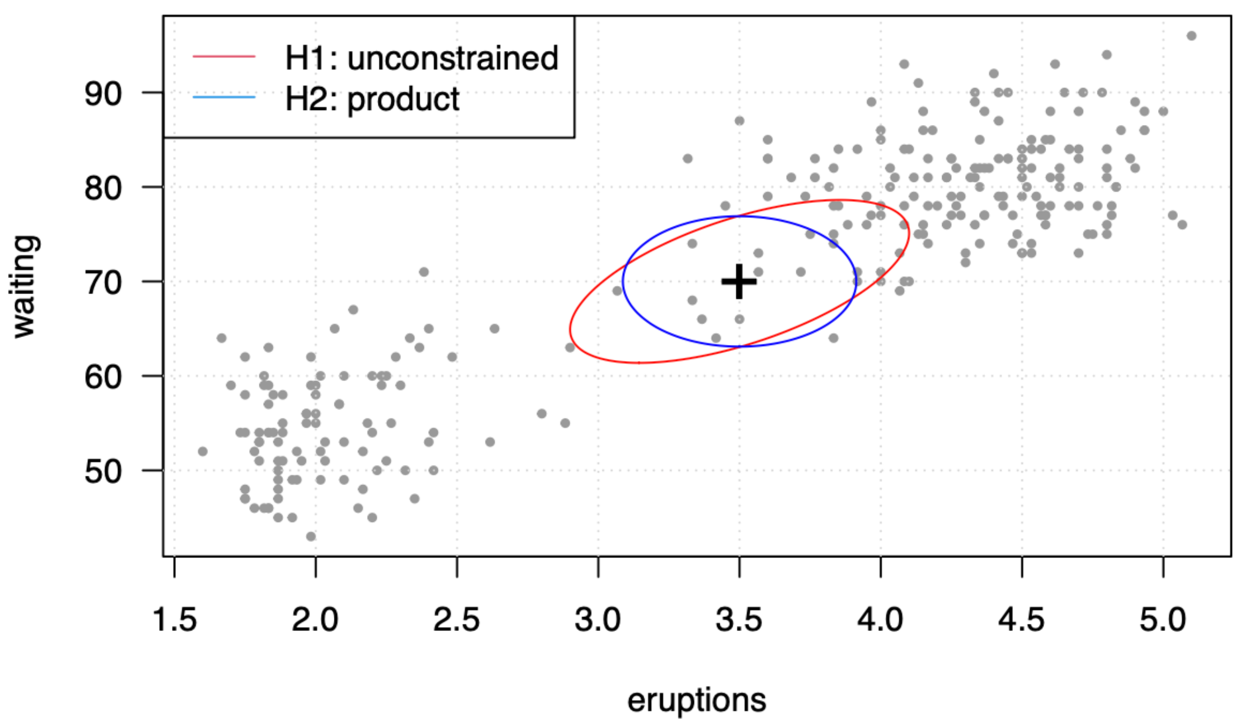
$$K\big((x_1, x_2); H\big) = K_1(x_1; h_1) K_2(x_2; h_2)$$

Where $h_1 = \sqrt{H_{11}}$.



$$H = \begin{bmatrix} 0.0285 & 0 \\ 0 & 7.949 \end{bmatrix}$$

Here is the kernel at location (3.5, 70).



2.1.2 Independent

The least complex model is one based on independence. Just like Naïve Bayes.

- Estimate the KDE for each dimension separately.
- The joint density estimate is the product of the p KDEs