# Multivariate Normals and Transformations of Joint Random Variables

Matthew Seguin

## Importing Libraries

```r
library(tidyverse)
library(latex2exp)
```

## 1.

Recall that if $X \sim N_n(\lambda, M)$, $v \in \mathbb{R}^{n \times 1}$, and $S \in \mathbb{R}^{n \times n}$ then we know $SX + v \sim N_n(S\lambda + v, SMS^T)$.

Further recall that if $X \sim N(\lambda, M)$ then marginally $X_i \sim N(\lambda_i, M_{i,i})$ for each $i \in \{1, ..., n\}$.

Questions on following pages.

**a.**

Let $a \in \mathbb{R}^n$ be a fixed vector.

Now let $B$ be an invertible matrix such that the first row of $B$ is just $a^T$, note that such a matrix exists since the only way it doesn't is if $a = 0$ (implying that no matter what the other entries are $B$ is not invertible since $det\, B = 0$) but that is not the case here and we can just choose the rest of the entries of $B$ such that $det\, B \neq 0$ and $B$ is invertible.

Then $BY \sim N_n(B\mu, B\Sigma B^T)$ and marginally $(BY)_1 \sim N\big((B\mu)_1, (B\Sigma B^T)_{1,1}\big)$

In this case we can write $B$ as:

$$B = \begin{bmatrix} a_1 & \dots & a_{n-1} & a_n \\ b_{2,1} & \dots & b_{2,n-1} & b_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ b_{n,1} & \dots & b_{n,n-1} & b_{n,n} \end{bmatrix} \implies B^T = \begin{bmatrix} a_1 & b_{2,1} & \dots & b_{n,1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1} & b_{2,n-1} & \dots & b_{n,n-1} \\ a_n & b_{2,n} & \dots & b_{n,n} \end{bmatrix}$$

For $B\mu$ we only care about the first entry which only depends on the first row of $B$. Namely:

$$(B\mu)_1 = \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} \begin{bmatrix} \mu_1 \\ \dots \\ \mu_n \end{bmatrix} = \sum_{i=1}^{n} a_i \mu_i$$

For $B\Sigma B^T$ we only care about the first entry of the first row which only depends on the first row of $B$ (which is just $a^T$) and the first column of $\Sigma B^T$ (and the first column of $\Sigma B^T$ only depends on the first column of $B^T$ which is just $a$).

The first column of $\Sigma B^T$ is given by:

$$\Sigma a = \begin{bmatrix} \sigma_1^2 & \dots & \rho\sigma_1\sigma_n \\ \vdots & \ddots & \vdots \\ \rho\sigma_n\sigma_1 & \dots & \sigma_n^2 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} a_1\sigma_1^2 + \rho\sigma_1 \sum_{i\neq 1} a_i\sigma_i \\ \vdots \\ a_n\sigma_n^2 + \rho\sigma_n \sum_{i\neq n} a_i\sigma_i \end{bmatrix}$$

Then the first entry of the first row of $B\Sigma B^T$ is given by:

$$a^T\Sigma a = \begin{bmatrix} a_1 & ... & a_n \end{bmatrix} \begin{bmatrix} a_1\sigma_1^2 + \rho\sigma_1 \sum_{i\neq 1} a_i\sigma_i \\ \vdots \\ a_n\sigma_n^2 + \rho\sigma_n \sum_{i\neq n} a_i\sigma_i \end{bmatrix}$$

$$= \sum_{j=1}^{n} a_j \Big( a_j\sigma_j^2 + \rho\sigma_j \sum_{i\neq j} a_i\sigma_i \Big) = \left( \sum_{j=1}^{n} a_j^2\sigma_j^2 \right) + \sum_{j=1}^{n}\sum_{i\neq j} \rho\sigma_j\sigma_i a_j a_i$$

$$= \left( \sum_{j=1}^{n} a_j^2 \mathbb{V}[Y_j] \right) + \sum_{j=1}^{n}\sum_{i\neq j} a_j a_i Cov(Y_j, Y_i) = \left( \sum_{j=1}^{n} \mathbb{V}[a_j Y_j] \right) + \sum_{j=1}^{n}\sum_{i\neq j} Cov(a_j Y_j, a_i Y_i)$$

$$= \sum_{j=1}^{n}\sum_{i=1}^{n} Cov(a_jY_j, a_iY_i)$$

Therefore marginally we know:

$$(BY)_1 \sim N\big((B\mu)_1, (B\Sigma B^T)_{1,1}\big) = N\Big(\sum_{i=1}^{n} a_i\mu_i, \sum_{j=1}^{n}\sum_{i=1}^{n} Cov(a_jY_j, a_iY_i)\Big)$$

Since $(BY)_1$ is just $a^TY$ we know that:

$$a^TY \sim N\Big(\sum_{i=1}^{n} a_i\mu_i, \sum_{j=1}^{n}\sum_{i=1}^{n} Cov(a_jY_j, a_iY_i)\Big)$$

Which implies:

$$a^T(Y-\mu) = a^TY - a^T\mu = a^TY - \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_n \end{bmatrix} = a^TY - \sum_{i=1}^{n} a_i\mu_i \sim N\Big(0, \sum_{j=1}^{n}\sum_{i=1}^{n} Cov(a_jY_j, a_iY_i)\Big)$$

Which gives us the final result that:

$$\frac{a^T(Y-\mu)}{\sqrt{a^T\Sigma a}} = \frac{a^T(Y-\mu)}{\sqrt{\sum_{j=1}^{n}\sum_{i=1}^{n} Cov(a_jY_j, a_iY_i)\Big)}} \sim N(0,1) \ \square$$

**b.**

$$\text{Let } A = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix} \text{ be a random vector that is independent of } Y.$$

Then we can condition to see:

$$\left( \frac{A^T(Y-\mu)}{\sqrt{A^T\Sigma A}} \middle| A = a \right) = \frac{a^T(Y-\mu)}{\sqrt{a^T\Sigma a}} \sim N(0,1)$$

Since $Y|A = a$ is just $Y$ due to the independence of $Y$ and $A$.

Therefore we can write the conditional density of $Z = \frac{A^T(Y-\mu)}{\sqrt{A^T\Sigma A}}$ as:

$f_{Z|A}(z|a) = \phi(z)$ where $\phi$ is just the standard normal density function

Then we can find the unconditional density of $Z$:

$$f_Z(z) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{Z,A_1,\ldots,A_n}(z, a_1, \ldots, a_n)\, da_1 \ldots da_n = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{A_1,\ldots,A_n}(a_1, \ldots, a_n) f_{Z|A}(z|a)\, da_1 \ldots da_n$$

$$= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{A_1,\ldots,A_n}(a_1, \ldots, a_n)\phi(z)\, da_1 \ldots da_n = \phi(z) \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{A_1,\ldots,A_n}(a_1, \ldots, a_n)\, da_1 \ldots da_n = \phi(z)$$

Therefore $Z = \frac{A^T(Y-\mu)}{\sqrt{A^T\Sigma A}} \sim N(0,1)$ and also we know $Z|A \overset{\text{d}}{=} Z$ which implies that $Z$ is independent of $A$.

So $\frac{A^T(Y-\mu)}{\sqrt{A^T\Sigma A}} \sim N(0,1)$ and is independent of $A$ $\square$

Alternatively if $A$ is discrete there will be a sum instead

**c.**

Let $Y \sim N_3(0, I_3)$ then this means $Y_1, Y_2, Y_3 \overset{\text{iid}}{\sim} N(0,1)$

Because of this we know that $Z = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix}$ and $A = \begin{bmatrix} e^{Y_3} \\ \log|Y_3| \end{bmatrix}$ are independent.

Clearly for $Z$ we know $\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ therefore by the results of the previous part $\frac{A^T(Z-\mu)}{\sqrt{A^T\Sigma A}} \sim N(0,1)$

Now quickly note:

$$A^T(Z-\mu) = A^T Z = \begin{bmatrix} e^{Y_3} & \log|Y_3| \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = Y_1 e^{Y_3} + Y_2 \log|Y_3|$$

$$A^T\Sigma A = \begin{bmatrix} e^{Y_3} & \log|Y_3| \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e^{Y_3} \\ \log|Y_3| \end{bmatrix} = \begin{bmatrix} e^{Y_3} & \log|Y_3| \end{bmatrix} \begin{bmatrix} e^{Y_3} \\ \log|Y_3| \end{bmatrix} = (e^{Y_3})^2 + (\log|Y_3|)^2 = e^{2Y_3} + (\log|Y_3|)^2$$

Therefore we know $\dfrac{Y_1 e^{Y_3} + Y_2 \log|Y_3|}{\sqrt{e^{2Y_3} + (\log|Y_3|)^2}} = \dfrac{A^T(Z-\mu)}{\sqrt{A^T \Sigma A}} \sim N(0,1)$ $\square$

## 2.

Let $Y_1 \sim N(0,1)$ and $\mathbb{P}[X=-1]=p$ and $\mathbb{P}[X=1]=1-p$ (where $0<p<1$) with $X$ independent of $Y_1$.

Then let $Y_2 = XY_1$.

Firstly note that:

$$\mathbb{P}[Y_2 \leq y | X=-1] = \mathbb{P}[XY_1 \leq y | X=-1] = \mathbb{P}[-Y_1 \leq y] = \mathbb{P}[Y_1 \geq -y] = 1-\Phi(-y) = \Phi(y)$$

Then:

$$\mathbb{P}[Y_2 \leq y | X=1] = \mathbb{P}[XY_1 \leq y | X=1] = \mathbb{P}[Y_1 \leq y] = \Phi(y)$$

And then we have:

$$\mathbb{P}[Y_2 \leq y] = \mathbb{P}[X=-1]\mathbb{P}[Y_2 \leq y|X=-1] + \mathbb{P}[X=1]\mathbb{P}[Y_2 \leq y | X=1] = p\,\Phi(y) + (1-p)\Phi(y) = \Phi(y)$$

Therefore marginally $Y_2 \sim N(0,1)$.

Quickly:

$$\mathbb{E}[Y_1Y_2] = \mathbb{E}[Y_1Y_2|X] = p\mathbb{E}[Y_1Y_2|X=-1] + (1-p)\mathbb{E}[Y_1Y_2|X=1]$$

$$= p\mathbb{E}[Y_1XY_1|X=-1] + (1-p)\mathbb{E}[Y_1XY_1|X=1] = p\mathbb{E}[-Y_1^2] + (1-p)\mathbb{E}[Y_1^2]$$

$$= (1-2p)\mathbb{E}[Y_1^2] = (1-2p)\Big(\mathbb{V}[Y_1] + (\mathbb{E}[Y_1])^2\Big) = (1-2p)(1+0) = 1-2p$$

Now consider $Y = [Y_1, Y_2]^T$, first we will find the covariance matrix:

$$Cov(Y) = \begin{bmatrix} Cov(Y_1,Y_1) & Cov(Y_1,Y_2) \\ Cov(Y_2,Y_1) & Cov(Y_2,Y_2) \end{bmatrix} = \begin{bmatrix} \mathbb{V}[Y_1] & Cov(Y_1,Y_2) \\ Cov(Y_1,Y_2) & \mathbb{V}[Y_2] \end{bmatrix} = \begin{bmatrix} 1 & 1-2p \\ 1-2p & 1 \end{bmatrix}$$

Notice that:

$$det\,[1] = 1 > 0 \text{ and } det \begin{bmatrix} 1 & 1-2p \\ 1-2p & 1 \end{bmatrix} = 1-(1-2p)^2 = 1-(1-4p+4p^2) = 4p(1-p) > 0$$

Since $0<p<1$ so that $0<1-p<1$ and $p(1-p)>0$.

Therefore since $Cov(Y)$ is symmetric and the determinants of its leading principal matrices are all positive we know $Cov(Y)$ is positive definite.

Continued on next page

Clearly the distribution of $Y_2|Y_1$ is not normal because $Y_2|Y_1 = y_1$ can only take the values $-y_1$ and $y_1$ and therefore is discrete and so can't have a normal distribution since that is continuous.

If $Y = [Y_1, Y_2]^T$ were jointly normal then $Y_2|Y_1$ should have some kind of normal distribution (this follows from the fact that $X_a|X_b = x_b \sim N_k(\mu_{a|b}, \Sigma_{a|b})$ when $X = [X_a, X_b]^T$ is a multivariate normal random variable), but it doesn't and therefore $Y$ is not a multivariate normal random vector.

So this is an example of a random vector $Y = [Y_1, Y_2]^T$ where $Y_1$ and $Y_2$ are marginally standard normal random variables, $Cov(Y)$ is positive definite, but $Y$ is not bivariate normal $\square$

## 3.

Let $Y_1, Y_2, Y_3 \overset{\text{iid}}{\sim} N(0,1)$ then consider $X_1 = Y_2 + Y_3$, $X_2 = Y_1 + Y_3$, and $X_3 = Y_1 + Y_2$.

Then we know we can write:

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = AY + \mu$$

First note that:

$$det\ A = det \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = 0\ det \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} - 1\ det \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} + 1\ det \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = -1(0-1) + 1(1-0) = 2 \neq 0$$

So $A$ is invertible. Therefore we know $X \sim N_3(0, \Sigma)$ where $\Sigma = AA^T$ and $\Sigma$ is also invertible.

Therefore:

$$\Sigma = AA^T = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

Recall that for multivariate normal distributions if $X = [X_a, X_b]^T$ and $\Sigma = \begin{bmatrix} \Sigma_{a,a} & \Sigma_{a,b} \\ \Sigma_{b,a} & \Sigma_{b,b} \end{bmatrix}$ which is invertible then $X_a | X_b = x_b \sim N_k(\mu_{a|b}, \Sigma_{a|b})$ where:

$$\mu_{a|b} = \mu_a + \Sigma_{a,b}\Sigma_{b,b}^{-1}(x_b - \mu_b) = \mu_a - \Lambda_{a,a}\Lambda_{a,b}(x_b - \mu_b)$$

$$\Sigma_{a|b} = \Sigma_{a,a} - \Sigma_{a,b}\Sigma_{b,b}^{-1}\Sigma_{b,a} = \Lambda_{a,a}^{-1}$$

$$\text{And } \begin{bmatrix} \Lambda_{a,a} & \Lambda_{a,b} \\ \Lambda_{b,a} & \Lambda_{b,b} \end{bmatrix} = \Lambda = \Sigma^{-1}$$

Here $X_a$ is a $k$ dimensional random vector, $X_b$ is therefore $n-k$ dimensional, $\Sigma_{a,a} \in \mathbb{R}^{k \times k}$, $\Sigma_{a,b} \in \mathbb{R}^{k \times n-k}$, $\Sigma_{b,a} \in \mathbb{R}^{n-k \times k}$, and $\Sigma_{b,b} \in \mathbb{R}^{n-k \times n-k}$. There is analogous dimensionality for $\Lambda$.

In this example $X_a = X_1$ and $X_b = [X_2, X_3]^T$ so that we get $\Sigma_{a,a} = [2]$, $\Sigma_{a,b} = \begin{bmatrix} 1 & 1 \end{bmatrix}$, $\Sigma_{b,a} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, and $\Sigma_{b,b} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

Continued on next page

We could compute directly from this or first compute $\Lambda$ here I will just use these directly:

$$\mu_{a|b} = \mu_a + \Sigma_{a,b}\Sigma_{b,b}^{-1}(x_b - \mu_b) = 0 + \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \left( \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right) = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} x_3 \\ x_2 \end{bmatrix} = \frac{x_2 + x_3}{3}$$

$$\Sigma_{a|b} = \Sigma_{a,a} - \Sigma_{a,b}\Sigma_{b,b}^{-1}\Sigma_{b,a} = \begin{bmatrix} 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \end{bmatrix} - \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 2 \end{bmatrix} - \begin{bmatrix} \frac{2}{3} \end{bmatrix} = \begin{bmatrix} \frac{4}{3} \end{bmatrix}$$

Therefore we know $X_1 \Big| [X_2, X_3] = [x_2, x_3] \sim N_k(\mu_{a|b}, \Sigma_{a|b}) = N(\frac{x_2+x_3}{3}, \frac{4}{3})$

Which means that $X_1 \Big| X_2 = X_3 = 0 = X_1 \Big| [X_2, X_3] = [0, 0] \sim N(0, \frac{4}{3})$ $\square$

# 4.

$$\text{Let } X, Y \overset{\text{iid}}{\sim} N(0,1).$$

First I will show a result about normal random variables that I will use in subsequent parts.

Assume $Z \sim N(\mu, \sigma^2)$ then let $c \in \mathbb{R}\backslash\{0\}$.

Then if $c > 0$ then $\mathbb{P}[cZ \leq z] = \mathbb{P}[Z \leq \frac{z}{c}] = \mathbb{P}[\frac{Z-\mu}{\sigma} \leq \frac{z/c-\mu}{\sigma}] = \Phi(\frac{z/c-\mu}{\sigma}) = \Phi(\frac{z-c\mu}{c\sigma})$

Similarly if $c < 0$ then

$\mathbb{P}[cZ \leq z] = \mathbb{P}[Z \geq \frac{z}{c}] = \mathbb{P}[\frac{Z-\mu}{\sigma} \geq \frac{z/c-\mu}{\sigma}] = 1 - \Phi(\frac{z/c-\mu}{\sigma}) = 1 - \Phi(\frac{z-c\mu}{c\sigma}) = 1 - \Phi(-\frac{z-c\mu}{|c|\sigma}) = \Phi(\frac{z-c\mu}{|c|\sigma})$

Therefore the CDF of $cZ$ is just the CDF of a $N(\mu, (c\sigma)^2)$ random variable and so $cZ \sim N(\mu, (c\sigma)^2)$.

For the rest of the problem recall that if $Z_1 \sim N(\mu_1, \sigma_1^2)$ and $Z_2 \sim N(\mu_2, \sigma_2^2)$ are independent then

$$Z_1 + Z_2 \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2).$$

## a.

Let $k \in \mathbb{R}$ be fixed. First trivially if $k = 0$ then $\mathbb{P}[X > kY] = \mathbb{P}[X > 0] = \frac{1}{2}$.

Now assume $k \neq 0$. From before we know $-kZ \sim N(0, k^2)$.

Furthermore clearly $-kY$ is still independent of $X$ since all we did was rescale by a constant.

Therefore we know $X - kY \sim N(0, k^2 + 1)$ which tells us:

$$\mathbb{P}[X > kY] = \mathbb{P}[X - kY > 0] = \frac{1}{2}$$

By the symmetry of the normal distribution.

Therefore $\mathbb{P}[X > kY] = \frac{1}{2}$ for all $k \in \mathbb{R}$.

Next part on next page

**b.**

Now we are letting $U = \sqrt{3}X + Y$ and $V = X - \sqrt{3}Y$ which means we can write:

$$W = \begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} \sqrt{3} & 1 \\ 1 & -\sqrt{3} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \sqrt{3} & 1 \\ 1 & -\sqrt{3} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = AZ + \mu$$

Which means that $W \sim N_2(0, \Sigma)$ where $\Sigma = AA^T$ (shown below):

$$\Sigma = AA^T = \begin{bmatrix} \sqrt{3} & 1 \\ 1 & -\sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 1 \\ 1 & -\sqrt{3} \end{bmatrix}^T = \begin{bmatrix} \sqrt{3} & 1 \\ 1 & -\sqrt{3} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 1 \\ 1 & -\sqrt{3} \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$$

Which implies that $U, V \overset{\text{iid}}{\sim} N(0, 4)$.

Now again let $k \in \mathbb{R}$. First trivially if $k = 0$ then $\mathbb{P}[U > kV] = \mathbb{P}[U > 0] = \frac{1}{2}$.

Now assume $k \neq 0$. From before we know $-kV \sim N(0, 4k^2)$.

Furthermore clearly $-kV$ is still independent of $U$ since all we did was rescale by a constant.

Therefore we know $U - kV \sim N(0, 4(k^2 + 1))$ which tells us:

$$\mathbb{P}[U > kV] = \mathbb{P}[U - kV > 0] = \frac{1}{2}$$

By the symmetry of the normal distribution.

Therefore $\mathbb{P}[U > kV] = \frac{1}{2}$ for all $k \in \mathbb{R}$.

**c.**

First we will look at what $U^2 + V^2$ actually is:

$$U^2 + V^2 = (\sqrt{3}X + Y)^2 + (X - \sqrt{3}Y)^2 = (3X^2 + 2\sqrt{3}XY + Y^2) + (X^2 - 2\sqrt{3}XY + 3Y^2)$$

$$= 4X^2 + 4Y^2 = 4(X^2 + Y^2)$$

Recall from sample work 2 that the sum of the squares of $k$ independent standard normals follows a Gamma$(\frac{k}{2}, \frac{1}{2})$ distribution (this is equivalently a Chi squared distribution with $k$ degrees of freedom).

In this particular example $X^2 + Y^2 \sim$ Gamma$(\frac{2}{2}, \frac{1}{2})$ = Gamma$(1, \frac{1}{2})$ = Exponential$(\frac{1}{2})$ Therefore we know:

$$\mathbb{P}[U^2 + V^2 < 1] = \mathbb{P}[4(X^2 + Y^2) < 1] = \mathbb{P}[X^2 + Y^2 < 1/4] = 1 - e^{-\frac{1}{2}(\frac{1}{4})} = 1 - e^{-\frac{1}{8}}$$

Where we used the fact that $X^2 + Y^2 \sim$ Exponential$(\frac{1}{2})$ and if $W \sim$ Exponential$(\lambda)$ then

$$\mathbb{P}[W < w] = \mathbb{P}[W \leq w] = F_W(w) = 1 - e^{-\lambda w}$$

**d.**

First note that $X = V + \sqrt{3}Y$ then we will show that $X$ and $V$ are independent:

$$W = \begin{bmatrix} X \\ V \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -\sqrt{3} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & -\sqrt{3} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = AZ + \mu$$

Clearly $A$ is invertible. Therefore we know $W \sim N_2(0, \Sigma)$ where $\Sigma = AA^T$ is invertible (shown below):

$$\Sigma = AA^T = \begin{bmatrix} 1 & 0 \\ 1 & -\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & -\sqrt{3} \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \\ 1 & -\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & -\sqrt{3} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix}$$

Recall that for multivariate normal distributions if $X = [X_a, X_b]^T$ and $\Sigma = \begin{bmatrix} \Sigma_{a,a} & \Sigma_{a,b} \\ \Sigma_{b,a} & \Sigma_{b,b} \end{bmatrix}$ which is invertible then $X_a|X_b = x_b \sim N_k(\mu_{a|b}, \Sigma_{a|b})$ where:

$$\mu_{a|b} = \mu_a + \Sigma_{a,b}\Sigma_{b,b}^{-1}(x_b - \mu_b) = \mu_a - \Lambda_{a,a}\Lambda_{a,b}(x_b - \mu_b)$$

$$\Sigma_{a|b} = \Sigma_{a,a} - \Sigma_{a,b}\Sigma_{b,b}^{-1}\Sigma_{b,a} = \Lambda_{a,a}^{-1}$$

And $\begin{bmatrix} \Lambda_{a,a} & \Lambda_{a,b} \\ \Lambda_{b,a} & \Lambda_{b,b} \end{bmatrix} = \Lambda = \Sigma^{-1}$

Here $X_a$ is a $k$ dimensional random vector, $X_b$ is therefore $n-k$ dimensional, $\Sigma_{a,a} \in \mathbb{R}^{k \times k}$, $\Sigma_{a,b} \in \mathbb{R}^{k \times n-k}$, $\Sigma_{b,a} \in \mathbb{R}^{n-k \times k}$, and $\Sigma_{b,b} \in \mathbb{R}^{n-k \times n-k}$. There is analogous dimensionality for $\Lambda$.

In this example $X_a = X$ and $X_b = V$ so that we get $\Sigma_{a,a} = [1]$, $\Sigma_{a,b} = [1]$, $\Sigma_{b,a} = [1]$, and $\Sigma_{b,b} = [4]$

We could compute directly from this or first compute $\Lambda$ here I will just use these directly:

$$\mu_{a|b} = \mu_a + \Sigma_{a,b}\Sigma_{b,b}^{-1}(x_b - \mu_b) = 0 + 1\left(\frac{1}{4}\right)(v - 0) = \frac{v}{4}$$

$$\Sigma_{a|b} = \Sigma_{a,a} - \Sigma_{a,b}\Sigma_{b,b}^{-1}\Sigma_{b,a} = 1 - 1\left(\frac{1}{4}\right)1 = \frac{3}{4}$$

Therefore we know $X|V = v \sim N_k(\mu_{a|b}, \Sigma_{a|b}) = N(\frac{v}{4}, \frac{3}{4})$ $\square$

## 5.

Let $X = \begin{bmatrix} X_1 & X_3 \\ X_3 & X_2 \end{bmatrix}$ where $X_1, X_2 \overset{\text{iid}}{\sim} N(0,1)$ and $X_3 \sim N(0, 1/2)$ is another independent normal random variable. Let $\lambda_1$ and $\lambda_2$ be eigenvalues of $X$ and let $s = |\lambda_1 - \lambda_2|$.

### a.

First we need to find the characteristic polynomial:

$$det(X - \lambda I) = det\left( \begin{bmatrix} X_1 & X_3 \\ X_3 & X_2 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = det \begin{bmatrix} X_1 - \lambda & X_3 \\ X_3 & X_2 - \lambda \end{bmatrix}$$

$$= (X_1 - \lambda)(X_2 - \lambda) - X_3^2 = X_1X_2 - \lambda(X_1 + X_2) + \lambda^2 - X_3^2 = \lambda^2 - \lambda(X_1 + X_2) + X_1X_2 - X_3^2$$

Setting this equal to 0 we get:

$$\lambda^2 - \lambda(X_1 + X_2) + X_1X_2 - X_3^2 = 0$$

Which we can solve using the quadratic formula:

$$\lambda = \frac{X_1 + X_2 \pm \sqrt{(X_1 + X_2)^2 - 4(1)(X_1X_2 - X_3^2)}}{2} = \frac{X_1 + X_2 \pm \sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)}}{2}$$

Or equivalently we know:

$$\lambda_1 = \frac{X_1 + X_2 + \sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)}}{2} \quad \lambda_2 = \frac{X_1 + X_2 - \sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)}}{2}$$

Which implies that:

$$s = |\lambda_1 - \lambda_2| = \left| \frac{X_1 + X_2 + \sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)}}{2} - \frac{X_1 + X_2 - \sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)}}{2} \right|$$

$$= \frac{1}{2} \left| \left( X_1 + X_2 + \sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)} \right) - \left( X_1 + X_2 - \sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)} \right) \right|$$

$$= \frac{1}{2} \left| 2\sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)} \right| = \sqrt{(X_1 + X_2)^2 - 4(X_1X_2 - X_3^2)}$$

$$= \sqrt{X_1^2 + 2X_1X_2 + X_2^2 - 4X_1X_2 + 4X_3^2} = \sqrt{X_1^2 - 2X_1X_2 + X_2^2 + 4X_3^2} = \sqrt{(X_1 - X_2)^2 + 4X_3^2} \quad \square$$

Next part on next page.

**b.**

$$\text{First we will find the distribution of } Z = \begin{bmatrix} \frac{X_1 - X_2}{\sqrt{2}} \\ \sqrt{2}X_3 \end{bmatrix}$$

Note that:

$$\begin{bmatrix} \frac{X_1 - X_2}{\sqrt{2}} \\ \sqrt{2}X_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = AX + \mu$$

Therefore we know $Z \sim N_2(0, \Sigma)$ where $\Sigma = AA^T$ (shown below):

$$\Sigma = AA^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix}^T = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & 0 \\ 0 & \sqrt{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\text{Therefore } Z = \begin{bmatrix} \frac{X_1 - X_2}{\sqrt{2}} \\ \sqrt{2}X_3 \end{bmatrix} \sim N_2(0, I_2) \text{ then let } M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2$$

Clearly $M$ is symmetric with full rank $r = 2$. Furthermore:

$$M^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2^2 = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = M$$

Then we can write:

$$\frac{(X_1 - X_2)^2}{2} + 2X_3^2 = \begin{bmatrix} \frac{X_1 - X_2}{\sqrt{2}} & \sqrt{2}X_3 \end{bmatrix} \begin{bmatrix} \frac{X_1 - X_2}{\sqrt{2}} \\ \sqrt{2}X_3 \end{bmatrix} = \begin{bmatrix} \frac{X_1 - X_2}{\sqrt{2}} & \sqrt{2}X_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{X_1 - X_2}{\sqrt{2}} \\ \sqrt{2}X_3 \end{bmatrix}$$

$$= Z^T M Z = (Z - \mu)^T M (Z - \mu) \sim \chi_r^2 = \chi_2^2$$

Therefore we know $\frac{(X_1 - X_2)^2}{2} + 2X_3^2$ has density $f(w) = \frac{1}{2^{2/2}\Gamma(2/2)} w^{2/2-1} e^{-w/2} = \frac{1}{2} e^{-w/2}$ for $w > 0$ since $\Gamma(1) = 0! = 1$.

More precisely this tells us $\frac{(X_1 - X_2)^2}{2} + 2X_3^2 \sim \text{Exponential}(\frac{1}{2})$

Another way to see this is that $\frac{(X_1 - X_2)^2}{2} + 2X_3^2 \sim \chi_2^2 = \text{Gamma}(\frac{2}{2}, \frac{1}{2}) = \text{Gamma}(1, \frac{1}{2}) = \text{Exponential}(\frac{1}{2})$

The density for $s = \sqrt{(X_1 - X_2)^2 + 4X_3^2} = \sqrt{2(\frac{(X_1 - X_2)^2}{2} + 2X_3^2)}$ is found below:

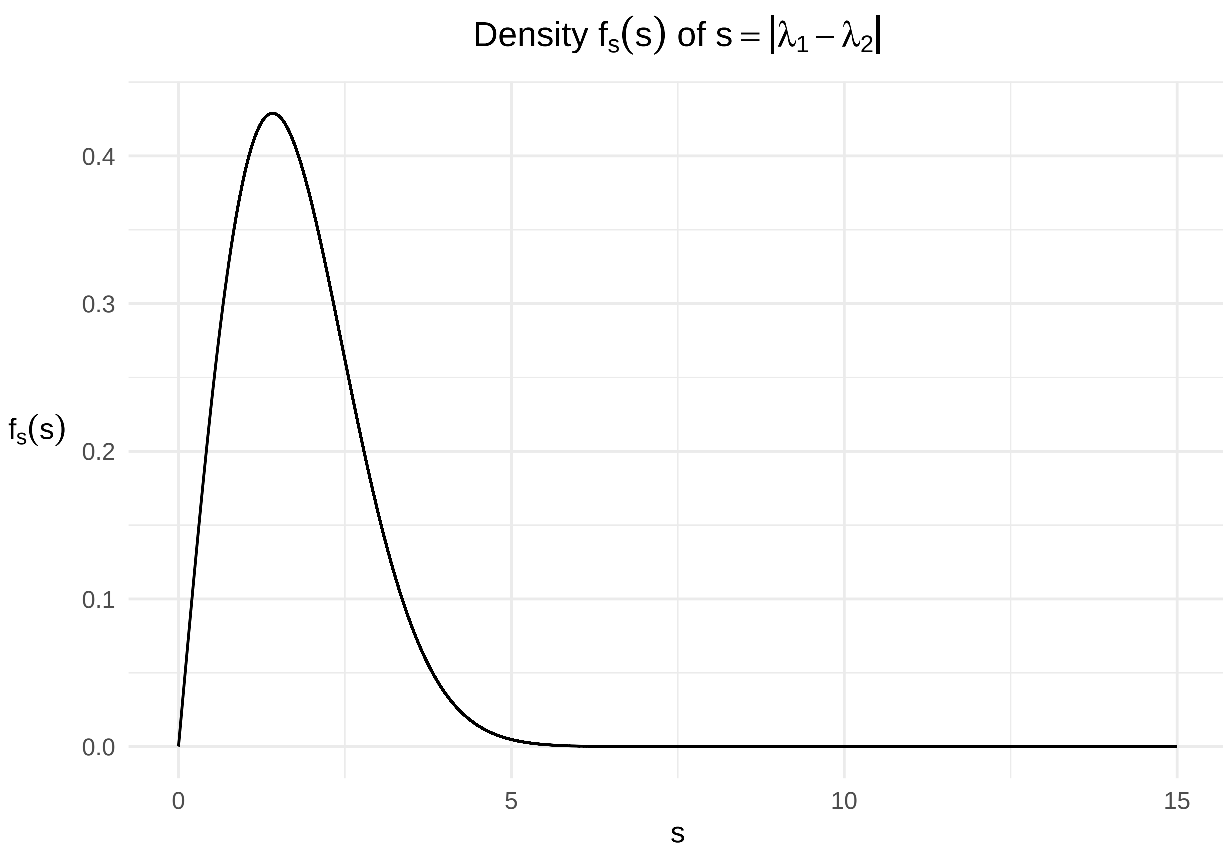
$$\text{First } s = \sqrt{(X_1 - X_2)^2 + 4X_3^2} = g\Big(\tfrac{(X_1-X_2)^2}{2} + 2X_3^2\Big) \text{ where } g(t) = \sqrt{2t} \text{ so } g^{-1}(t) = \tfrac{t^2}{2}$$

$$f_s(s) = f(g^{-1}(s))\Big|\frac{d}{ds}g^{-1}(s)\Big| = f\Big(\frac{s^2}{2}\Big)\Big|\frac{d}{ds}\frac{s^2}{2}\Big| = \frac{s}{2}e^{-\frac{(s^2/2)}{2}} = \frac{s}{2}\exp\Big(-\frac{s^2}{4}\Big) \quad \text{for} \quad s > 0 \ \square$$

**c.**

Now we will plot the density and comment on what it tells us.

```
df <- data.frame(s = (1:150000)/10000)

df %>%
  mutate(f_s = (s/2)*exp(-(s^2)/4)) %>%
  ggplot(aes(x = s, y = f_s)) +
  geom_line() +
  labs(x = "s",
       y = TeX("$f_s (s)$ "),
       title = TeX("Density $f_s (s)$ of $s = |\ lambda_1 -\ lambda_2|$")
       ) +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.title = element_text(color = "black"),
        axis.title.y = element_text(angle = 0, vjust = 0.5)
        )
```




We can see that the absolute difference of the eigenvalues of $X$ is almost always less than 5 and seem to be most often centered around 1.

# 6.

## a.

Create Python functions for squared exponential and Matérn kernel functions to compute similarity between any pair of inputs.

In [24]:
```
def squared_exponential_kernel(x1, x2, l, sigma_f):
    """
    Computes the squared exponential kernel matrix between inputs x1 and x2.

    :param x1: 1D array of input points.
    :param x2: 1D array of input points.
    :param l: Length scale parameter (float).
    :param sigma_f: Scale factor (float).
    :return: Kernel matrix between x1 and x2.
    """
    # Get length of data
    n = len(x1)
    # Initialize kernel matrix
    k = np.zeros([n, n])
    for i in range(n):
        for j in range(n):
            # Add kernel function for each point (w/o scale factor)
            k[i,j] = np.exp(-((x1[i] - x2[j])**2)/(2*(l**2)))
    # Multiply by the scale factor
    k = (sigma_f**2) * k
    return k


def matern_kernel(x1, x2, nu, l):
    """
    Computes the Matérn kernel matrix between inputs x1 and x2 for arbitrary nu.
    :param x1: 1D array of input points.
    :param x2: 1D array of input points.
    :param nu: Smoothness parameter (float).
    :param l: Length scale parameter (float).
    :return: Kernel matrix between x1 and x2.
    """
    # Get length of data
    n = len(x1)
    # Initialize kernel matrix
    k = np.zeros([n, n])
    # Get coefficient term for kernel function
    coeff = (2**(1-nu))/scipy.special.gamma(nu)
    for i in range(n):
        for j in range(n):
            # Find absolute distance
            r = abs(x1[i] - x2[j])
            # Scale distance
            scaled_r = r*np.sqrt(2*nu)/l
            # If diagonal make value 1
            if i == j:
                k[i,j] = 1
```

```
            # Otherwise add kernel function for each point
            else:
                k[i,j] = coeff*(scaled_r**nu)*scipy.special.kv(nu, scaled_r)
    return k
```

# b.

For a given kernel function make a Python function to predict the posterior mean and variance of test_y in a Gaussian Process regression.

In [25]:
```python
def gp_predict(train_x, train_y, test_x, kernel_func, noise_sigma, **kernel_params):
    """
    Predicts the mean and variance for a set of test points using Gaussian Process regression.

    :param train_x: 1D array of training input points.
    :param train_y: 1D array of training output points.
    :param test_x: 1D array of test input points.
    :param kernel_func: Kernel function to use for prediction.
    :param noise_sigma: Noise standard deviation (float).
    :param kernel_params: Additional parameters for the kernel function.
    :return: mean (1D array of predicted means), variance (1D array of predicted variances).
    """
    # Here I use m(x) = 0
    # Get lengths of data
    m = len(test_x)
    n = len(train_x)
    # Initialize mean and variance lists
    mean = []
    variance = []
    for j in range(m):
        # Add each point to predict one at a time
        vec = np.append(train_x, test_x[j])
        # Add generate kernel for predicted point
        k = kernel_func(vec, vec, **kernel_params) + noise_sigma * np.identity(n+1)
        # Break down kernel matrix
        cT = k[n,:n]
        k_n_inv = np.linalg.inv(k[:n,:n])
        # Find mean and variance
        mean.append(cT @ k_n_inv @ train_y)
        variance.append(k[n,n] - cT @ k_n_inv @ cT.T)
    # Make lists arrays
    mean = np.array(mean)
    variance = np.array(variance)
    return mean, variance
```

# c.

Vary the kernel parameters (e.g., $\sigma_f$, $l$, and $\nu$) and observe prediction changes.

In [26]:
```python
# Simulation function
def generate_training_data(n_points, x_min, x_max, func, noise_sigma, seed=1234):
    rng = np.random.RandomState(seed)
    xs = rng.uniform(x_min, x_max, n_points)
    ys = func(xs) + rng.randn(n_points) * noise_sigma
    return xs, ys
```

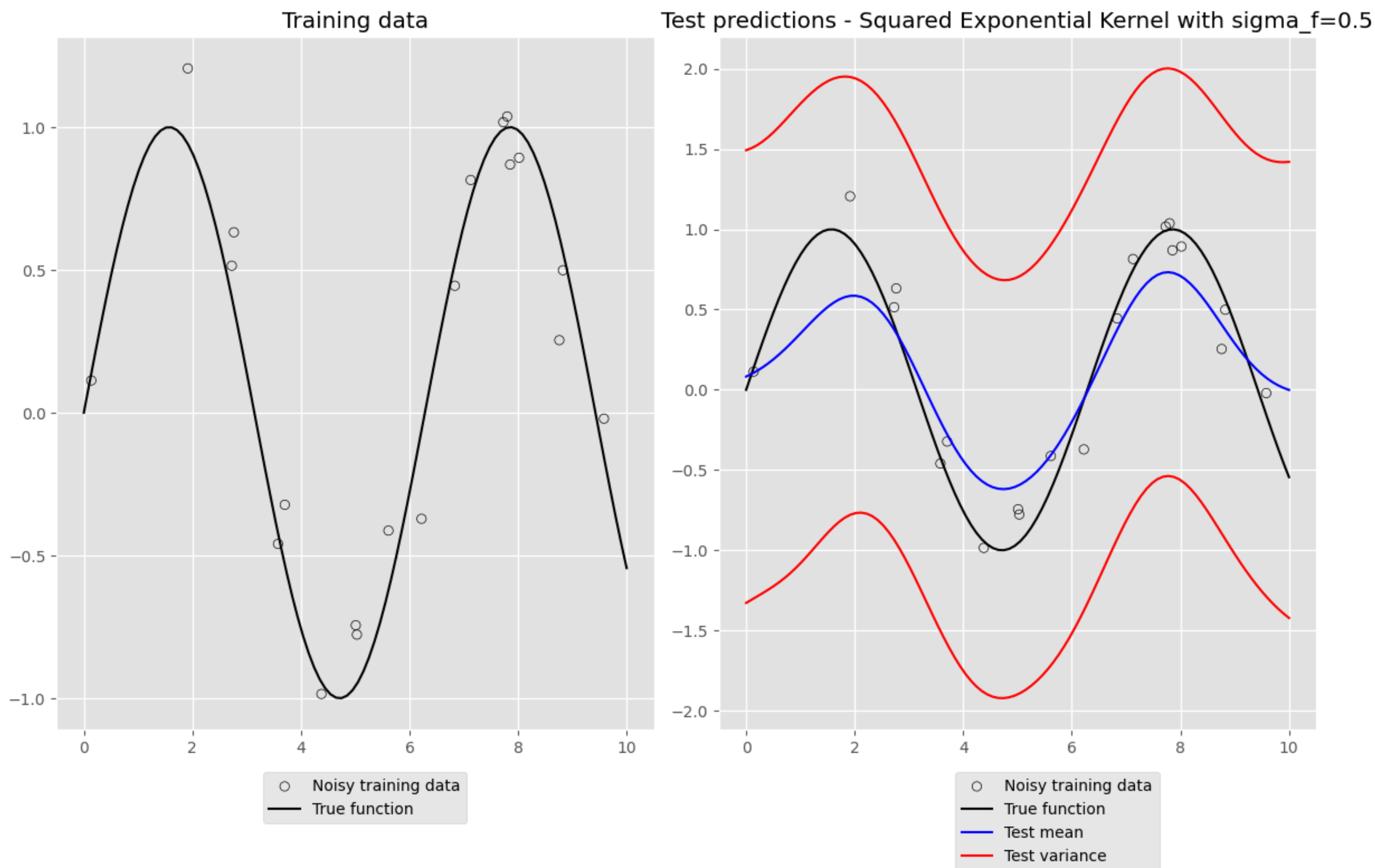
```python
# Plotting function using gp_predict
import numpy as np
import scipy
import matplotlib.pyplot as plt
plt.style.use('ggplot')

def plot(kernel_func, kernel_params, noise_sigma, title):
    predict_y, predict_y_variance = gp_predict(train_x, train_y, test_x, kernel_func, noise_sigma

    fig, axs = plt.subplots(nrows=1, ncols=2, figsize=(12, 8), tight_layout=True)
    axs[0].scatter(train_x, train_y, facecolors='none', edgecolors='k', label='Noisy training da
    axs[0].plot(gt_x, gt_y, color='k', label='True function')
    axs[0].set_title('Training data')
    axs[0].legend(bbox_to_anchor=(0.7, -0.05))

    axs[1].scatter(train_x, train_y, facecolors='none', edgecolors='k', label='Noisy training da
    axs[1].plot(gt_x, gt_y, color='k', label='True function')
    axs[1].plot(test_x, predict_y, color='b', label='Test mean')
    axs[1].plot(test_x, predict_y + np.sqrt(predict_y_variance) * 2.0, color='r', label='Test va
    axs[1].plot(test_x, predict_y - np.sqrt(predict_y_variance) * 2.0, color='r')
    #axs[1].plot(test_x, predict_y + np.sqrt(predict_y_variance) * 2.0 + noise_sigma, color='g',
    #axs[1].plot(test_x, predict_y - np.sqrt(predict_y_variance) * 2.0 - noise_sigma, color='g')
    axs[1].set_title(f'Test predictions - {title}')
    axs[1].legend(bbox_to_anchor=(0.75, -0.05))

    plt.show()

# Generating training data and ground truth for demonstration
f = np.sin
noise_sigma = 0.35
train_x, train_y = generate_training_data(n_points=20, x_min=0.0, x_max=10.0, func=f, noise_sigm

# Ground truth
gt_x = np.linspace(0.0, 10.0, 100)
gt_y = f(gt_x)
test_x = np.linspace(0.0, 10.0, 100)
```

## Squared Exponential Kernel

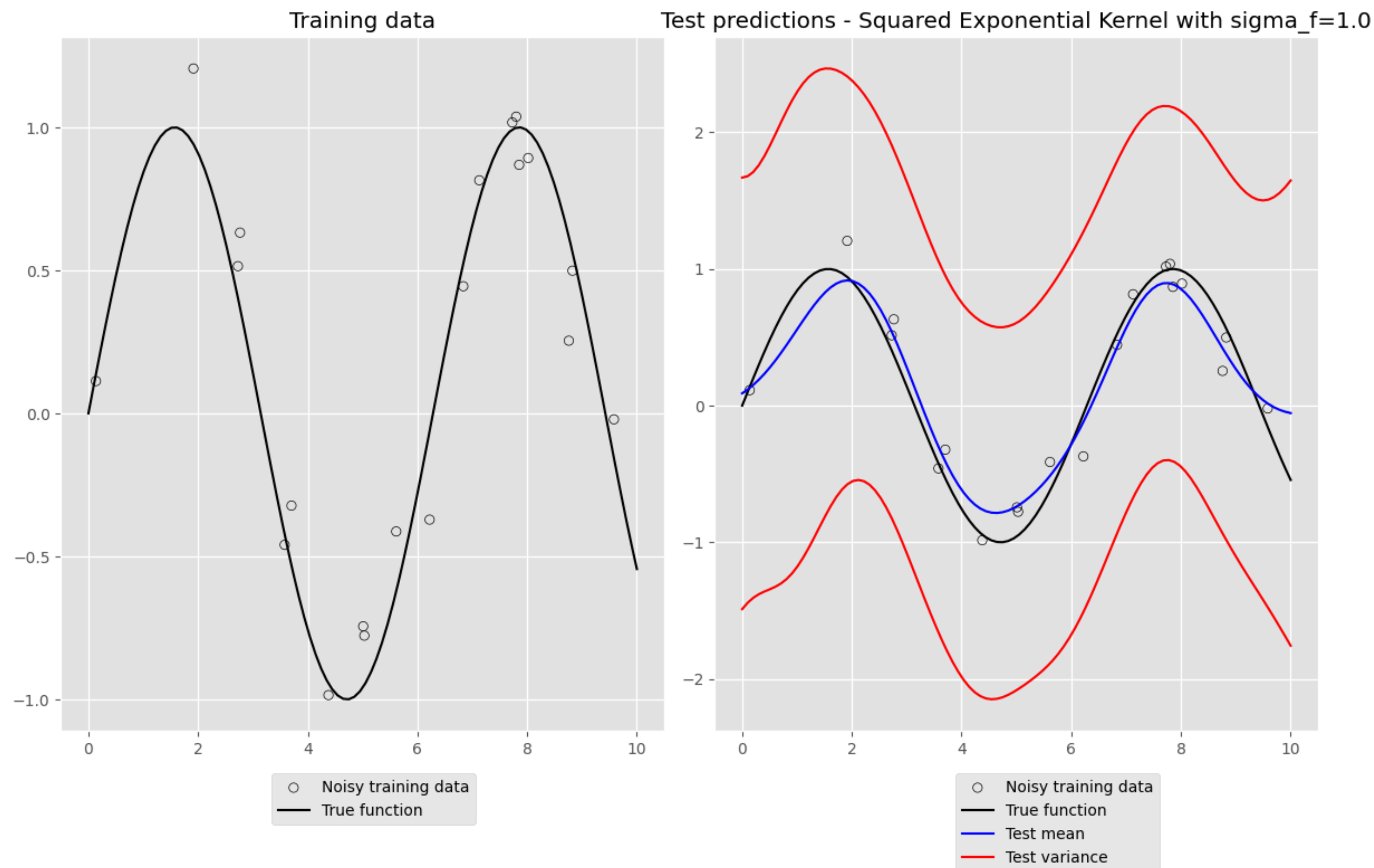
### Varying $\sigma_f$

In [27]:

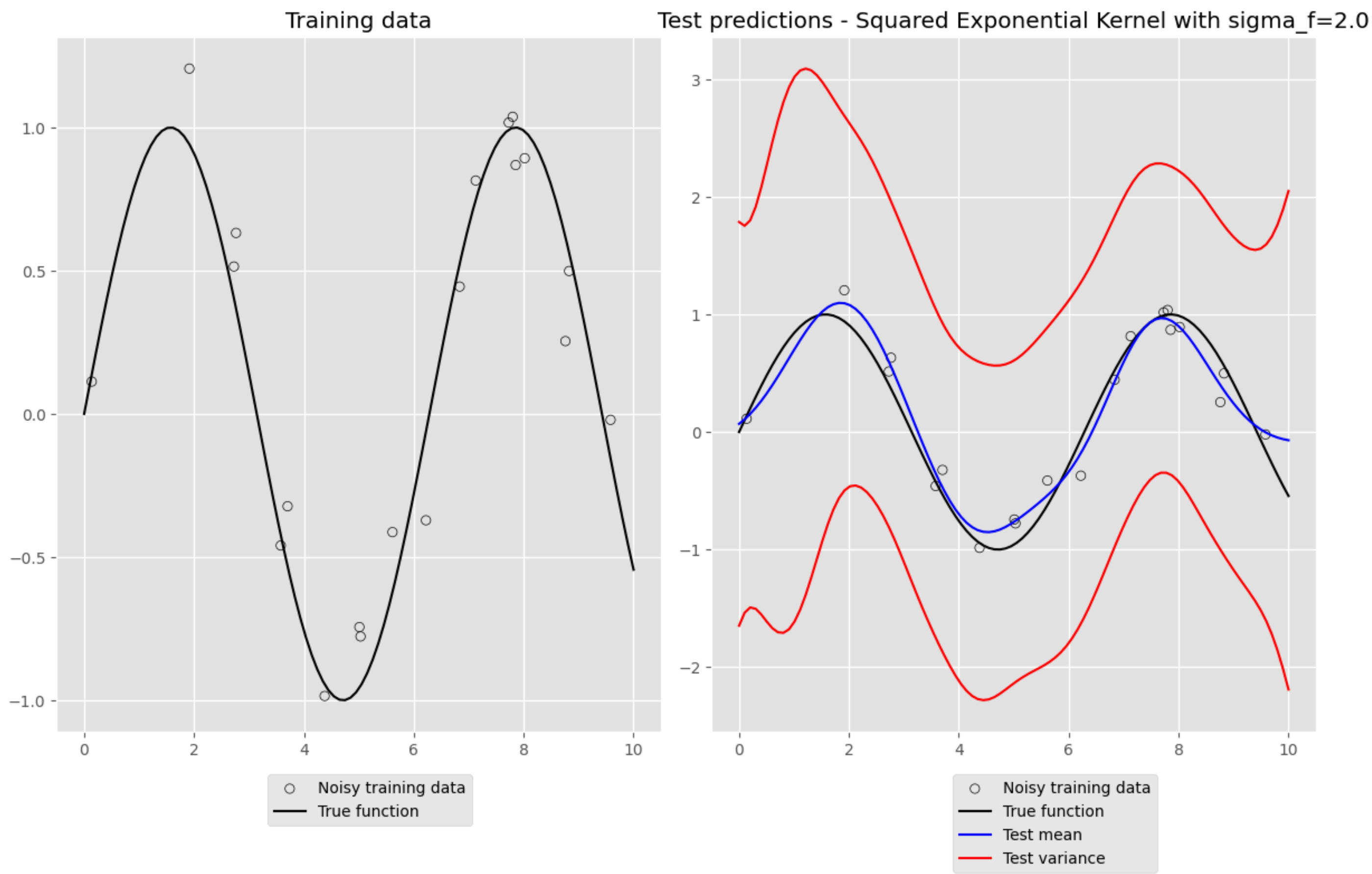
```python
plot(
    kernel_func=squared_exponential_kernel,
    kernel_params={'l': 1.0, 'sigma_f': 0.5},
    noise_sigma=noise_sigma,
    title="Squared Exponential Kernel with sigma_f=0.5"
)
```




In [28]:
```
plot(
    kernel_func=squared_exponential_kernel,
    kernel_params={'l': 1.0, 'sigma_f': 1.0},
    noise_sigma=noise_sigma,
    title="Squared Exponential Kernel with sigma_f=1.0"
)
```



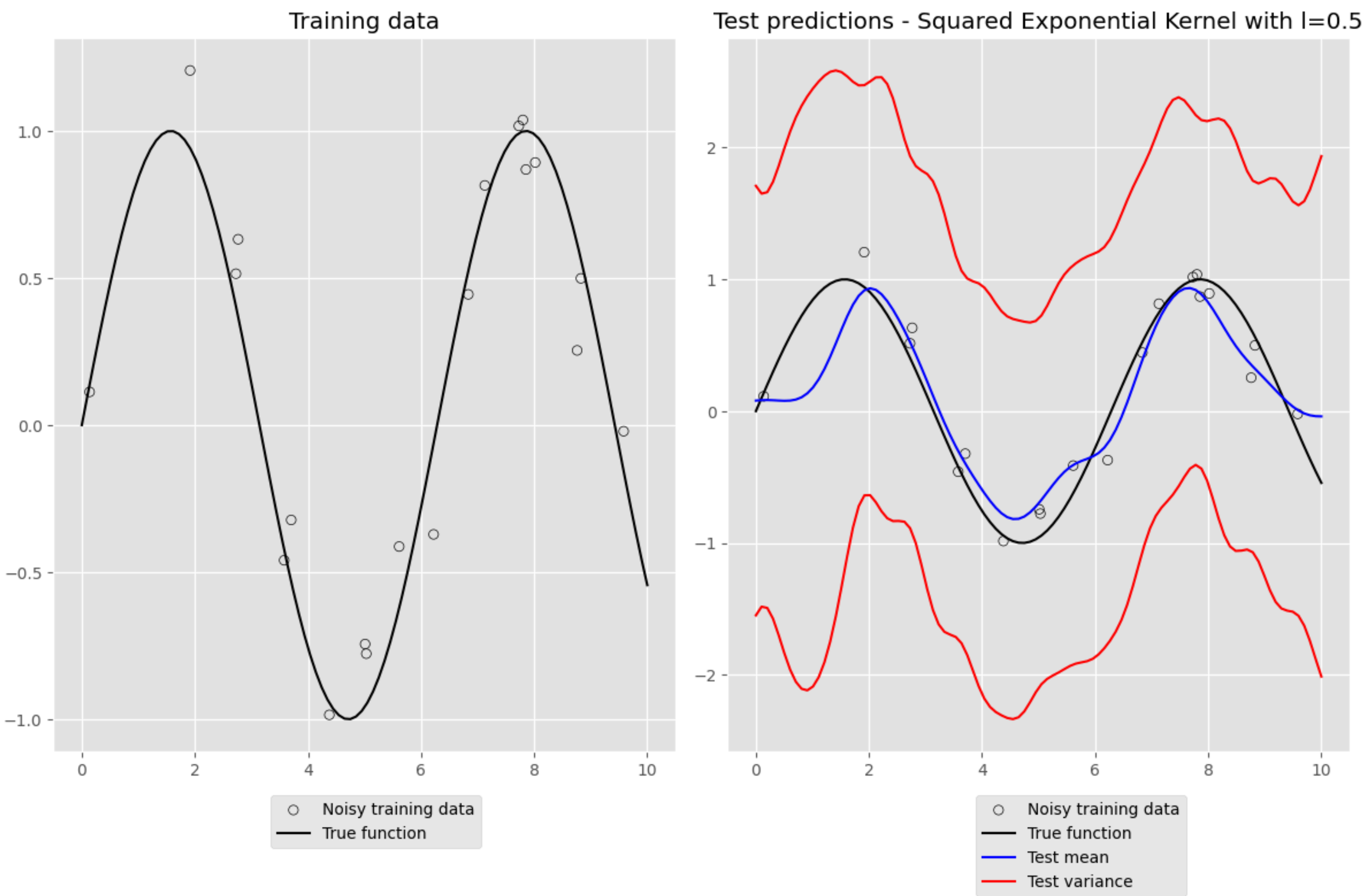

In [29]:
```
plot(
    kernel_func=squared_exponential_kernel,
    kernel_params={'l': 1.0, 'sigma_f': 2.0},
    noise_sigma=noise_sigma,
    title="Squared Exponential Kernel with sigma_f=2.0"
)
```



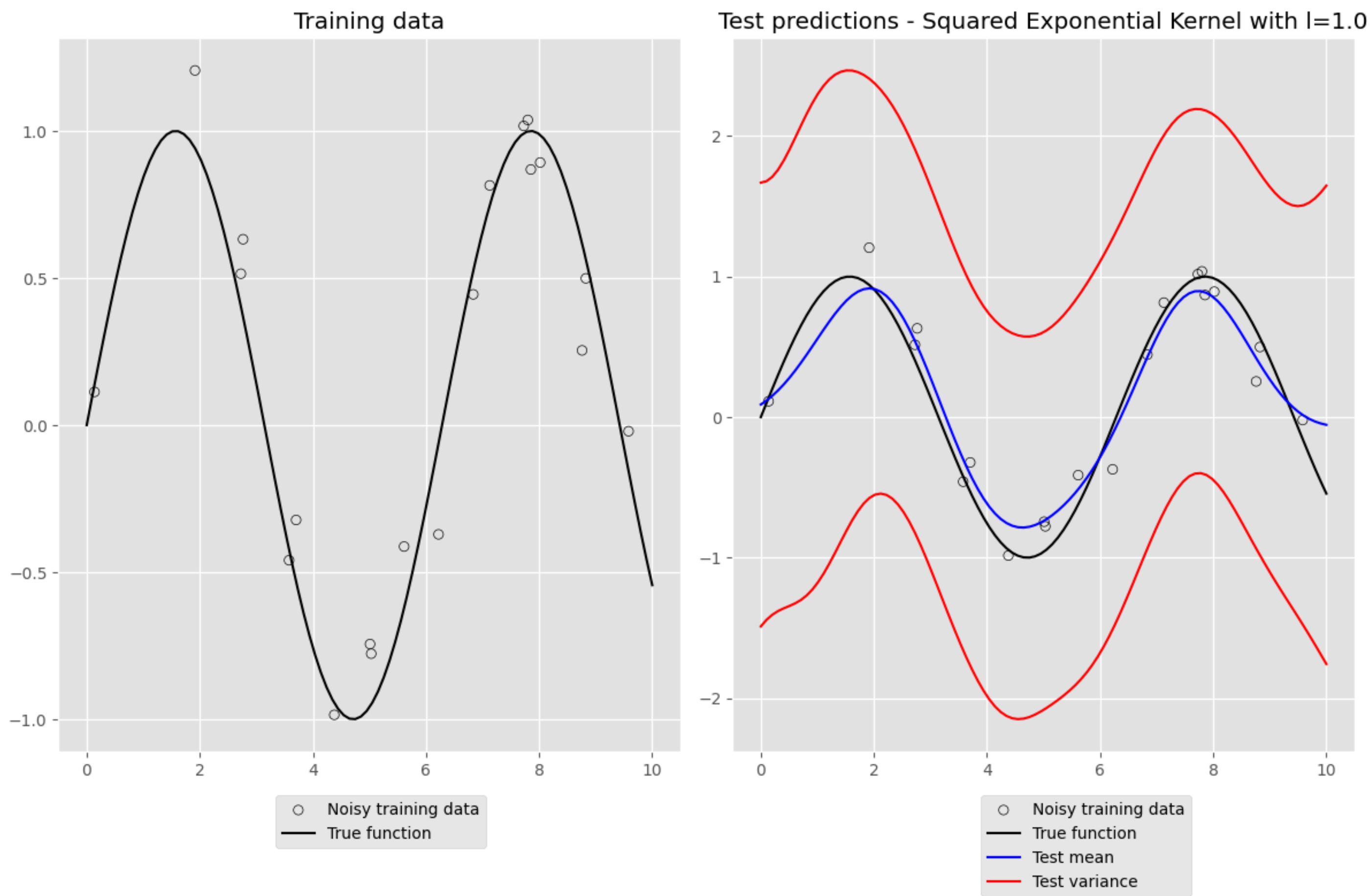

Clearly both the mean and variance line graphs are on a smaller scale (closer to 0) for lower values of $\sigma_f$ but there is not a significant change in the smoothness of the curves as we vary $\sigma_f$. So as $\sigma_f$ increases so does the uncertainty of the GP prediction, but there is not a significant difference in the smoothness of the prediction.

### Varying $l$

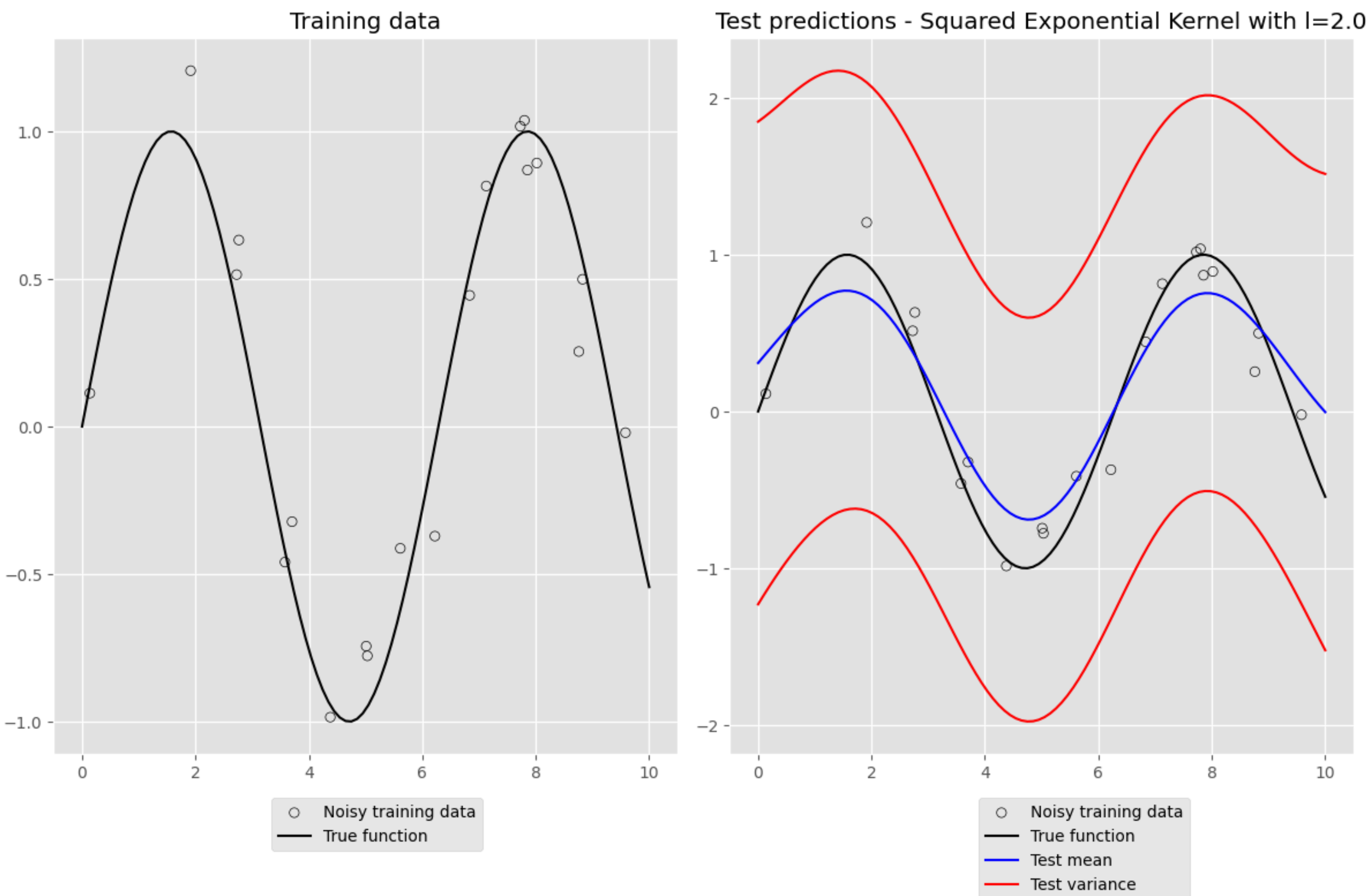
In [30]:
```
plot(
    kernel_func=squared_exponential_kernel,
    kernel_params={'l': 0.5, 'sigma_f': 1.0},
    noise_sigma=noise_sigma,
    title="Squared Exponential Kernel with l=0.5"
)
```




In [31]:
```
plot(
    kernel_func=squared_exponential_kernel,
    kernel_params={'l': 1.0, 'sigma_f': 1.0},
    noise_sigma=noise_sigma,
    title="Squared Exponential Kernel with l=1.0"
)
```




In [32]:
```
plot(
    kernel_func=squared_exponential_kernel,
    kernel_params={'l': 2.0, 'sigma_f': 1.0},
    noise_sigma=noise_sigma,
    title="Squared Exponential Kernel with l=2.0"
)
```

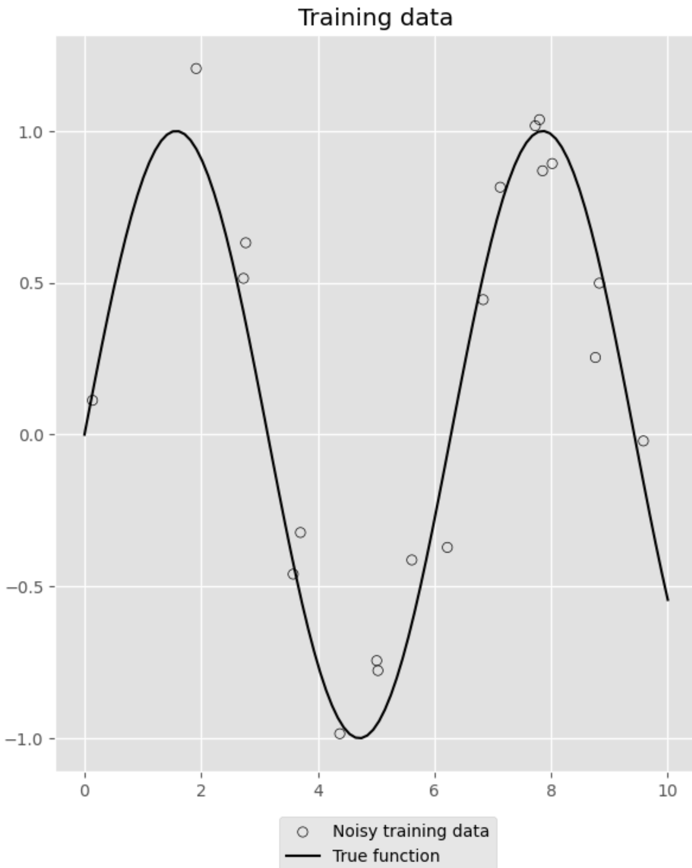


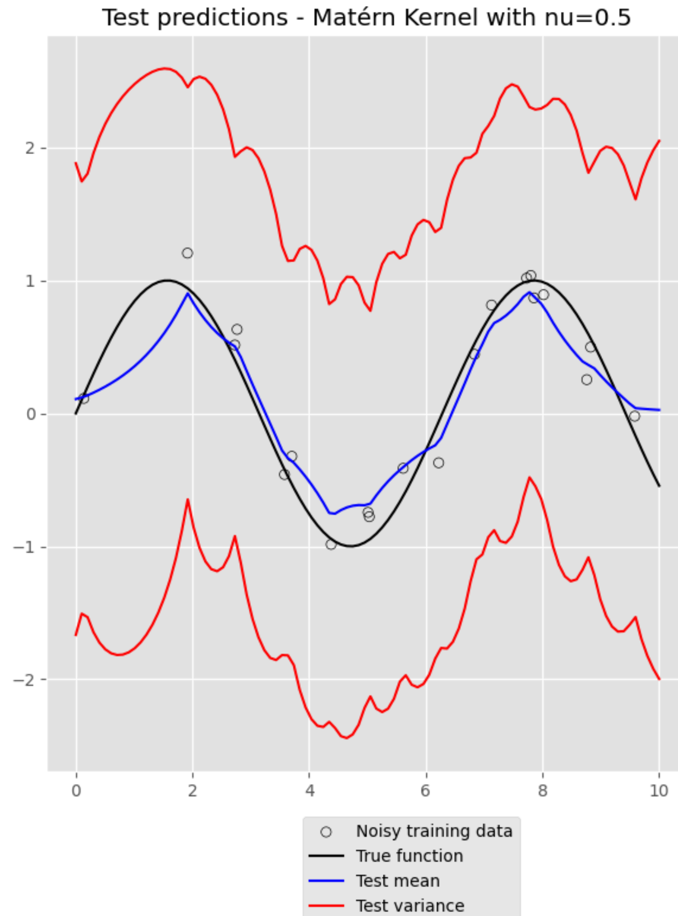

Clearly both the mean and variance line graphs are less smooth for lower values of $l$ but there is not a significant change in the scale of the curves (distance from 0) as we vary $l$. So as $l$ increases so does the smoothness of the GP prediction, but there is not a significant difference in the uncertainty of the prediction.

## Matérn Kernel

### Varying $\nu$

In [33]:
```
plot(
    kernel_func=matern_kernel,
    kernel_params={'nu': 0.5, 'l': 1.0},
    noise_sigma=noise_sigma,
    title="Matérn Kernel with nu=0.5"
)
```

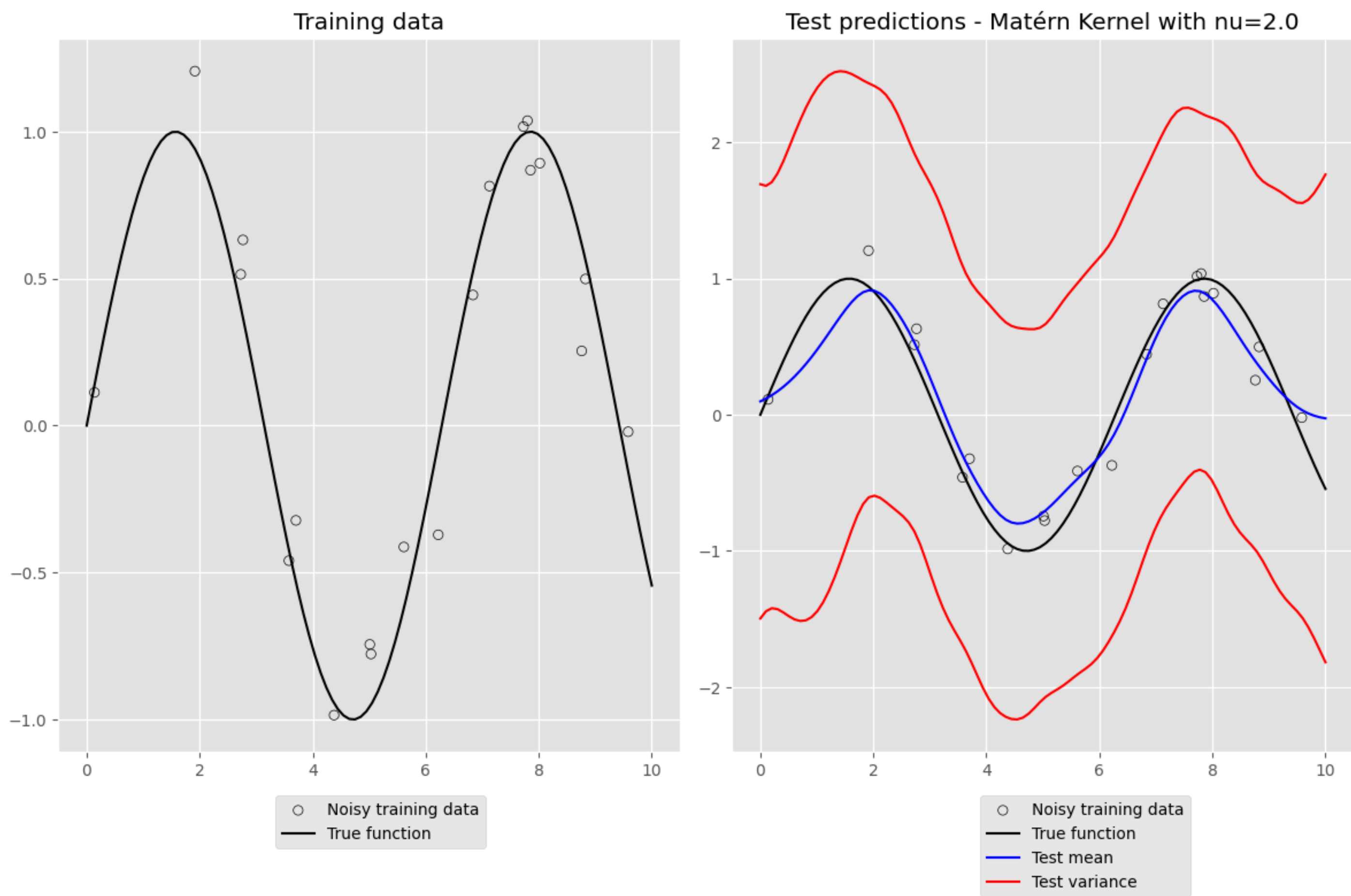



In [34]:
```
plot(
    kernel_func=matern_kernel,
    kernel_params={'nu': 1, 'l': 1.0},
    noise_sigma=noise_sigma,
    title="Matérn Kernel with nu=1.0"
)
```


In [35]:

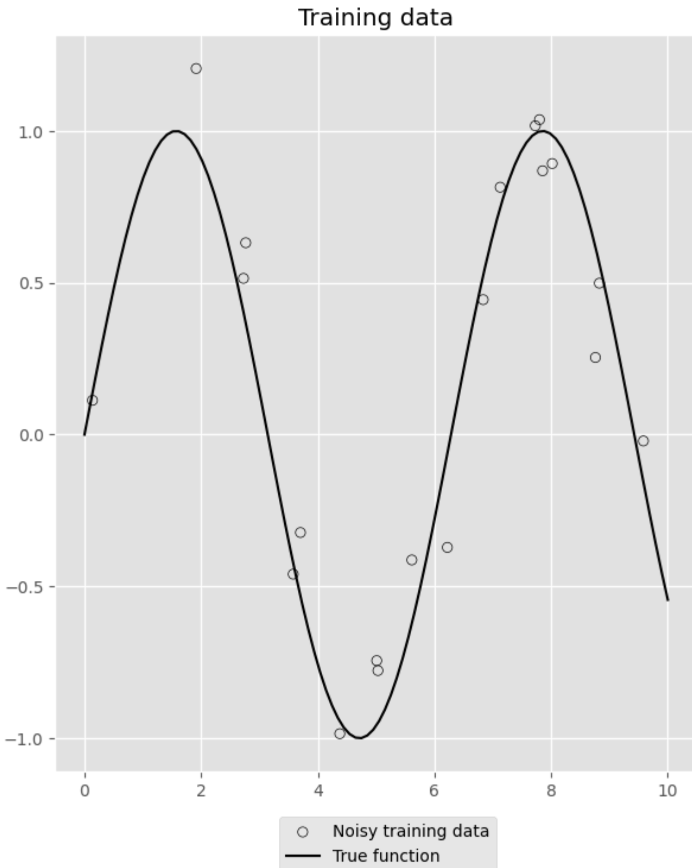
```
plot(
    kernel_func=matern_kernel,
    kernel_params={'nu': 2, 'l': 1.0},
    noise_sigma=noise_sigma,
    title="Matérn Kernel with nu=2.0"
)
```

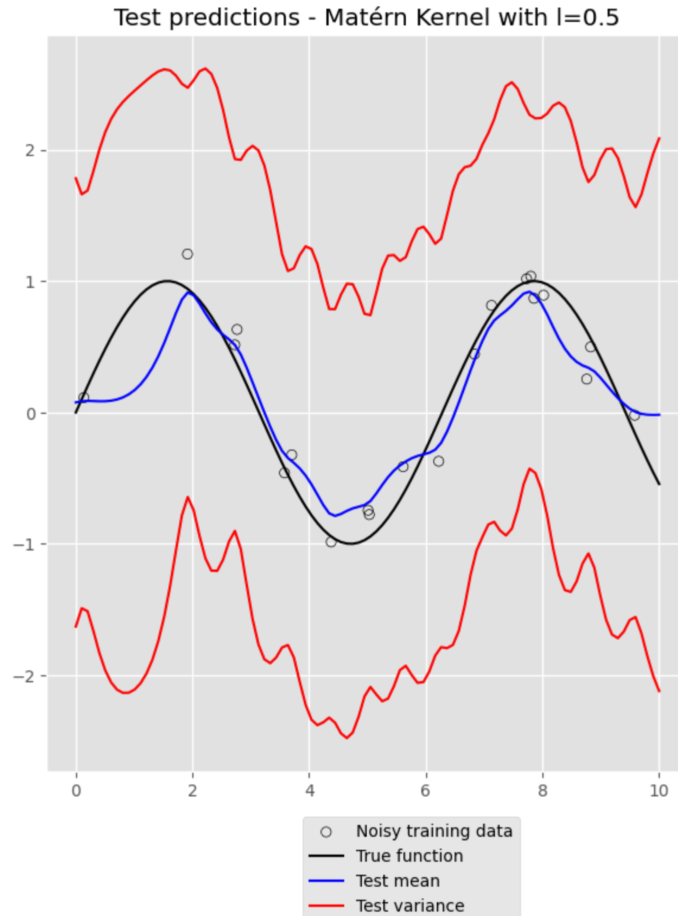



Clearly both the mean and variance line graphs are less smooth for lower values of $\nu$ but there is not a significant change in the scale of the curves (distance from 0) as we vary $\nu$. So as $\nu$ increases so does the smoothness of the GP prediction, but there is not a significant difference in the uncertainty of the prediction.

## Varying $l$

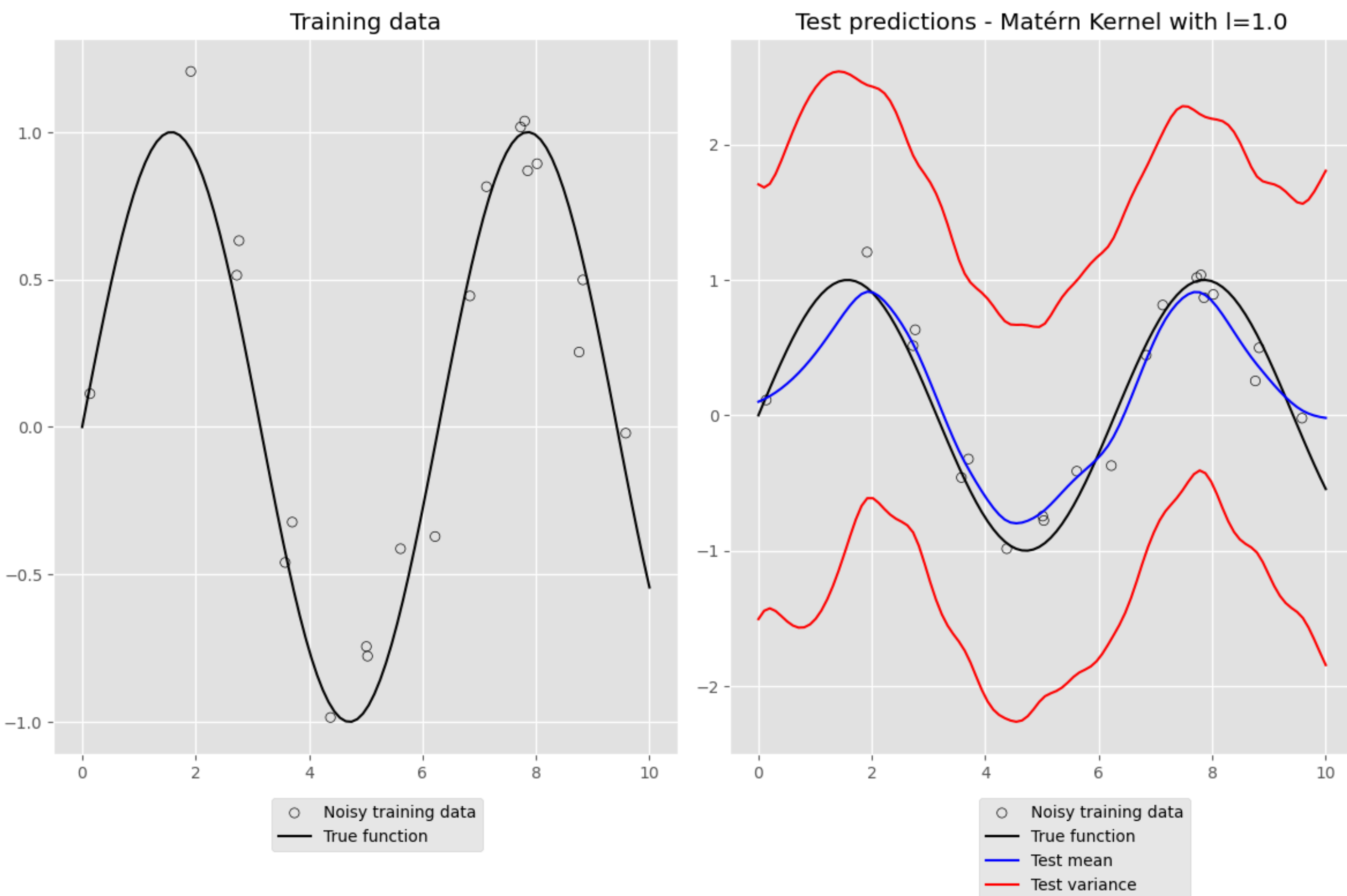
In [36]:
```
# Using the Matérn kernel
# Varying l
plot(
    kernel_func=matern_kernel,
    kernel_params={'nu': 1.5, 'l': 0.5},
    noise_sigma=noise_sigma,
    title="Matérn Kernel with l=0.5"
)
```



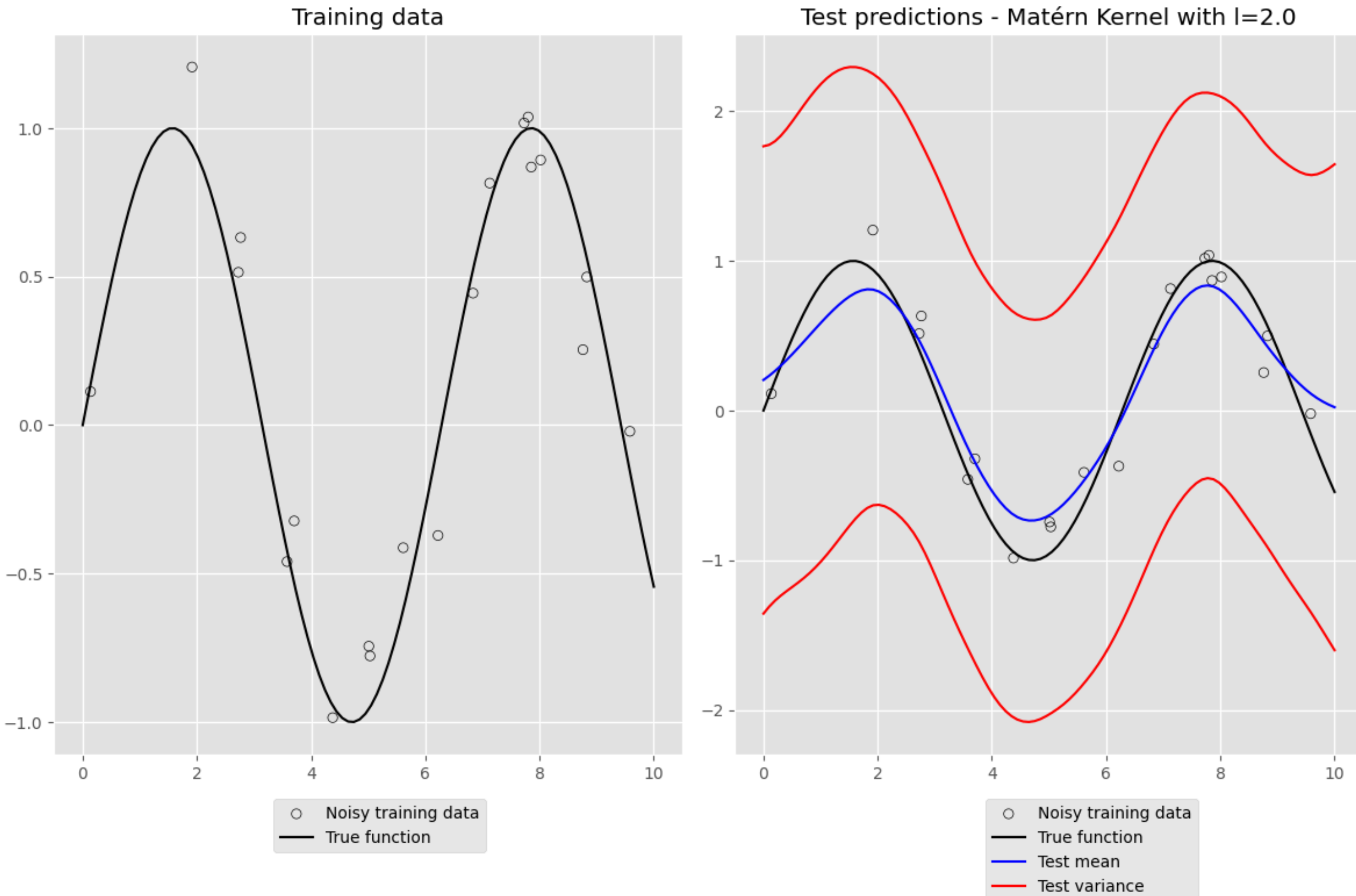

In [37]:

```
plot(
    kernel_func=matern_kernel,
    kernel_params={'nu': 1.5, 'l': 1},
    noise_sigma=noise_sigma,
    title="Matérn Kernel with l=1.0"
)
```




In [38]:

```
plot(
    kernel_func=matern_kernel,
    kernel_params={'nu': 1.5, 'l': 2.0},
    noise_sigma=noise_sigma,
    title="Matérn Kernel with l=2.0"
)
```




Clearly both the mean and variance line graphs are less smooth and the scale of the curves are larger (further from 0) for lower values of $l$. So as $l$ increases so does the smoothness of the GP prediction while the uncertainty of the GP prediction decreases.