In [3]:

```python
import numpy as np
from matplotlib import cm
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d  # because we will be using projection='3d'
from ipywidgets import interact_manual, FloatSlider
```

executed in 6ms, finished 17:12:29 2019-08-15

In [4]:

```python
def graph3d_wrapper(func, xlabel:str, ylabel:str, zlabel:str, title:str, xlower:float, xupper:float,xstep:float,
                    ylower:float, yupper:float, ystep:float, proj:bool):
    """
    Wrapper function for making the plotting function easily usable for 'interact_manual'.
    Set values to all arguments except the one manipulated by 'FloatSlider'.
    """

    def graph3d(cov):

        fig = plt.figure()
        ax = fig.gca(projection='3d')

        xs = np.arange(xlower, xupper, xstep)
        ys = np.arange(ylower, yupper, ystep)
        xs, ys = np.meshgrid(xs, ys)
        zs = func(xs, ys, cov)

        ax.plot_surface(xs, ys, zs, alpha=0.7, cmap=cm.viridis)
        if proj:
            cset = ax.contourf(xs, ys, zs, zdir='z', offset=zs.min(), cmap=cm.coolwarm)

        ax.set_xlim(xlower, xupper); ax.set_ylim(ylower, yupper); ax.set_zlim(zs.min(), zs.max())
        ax.set_title(title)
        ax.set_xlabel(xlabel); ax.set_ylabel(ylabel); ax.set_zlabel(zlabel)
        plt.tight_layout()
        plt.show()

    return graph3d
```

executed in 13ms, finished 17:12:29 2019-08-15

1 Definition of multi-variate normal distribution

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}) \right\}$$

2 Definition of Mahalanobis distance, $\triangle$

$\triangle = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma} (\mathbf{x} - \boldsymbol{\mu})$ which reduces to Euclidean distance when $\boldsymbol{\Sigma}$ is the identity matrix.

3 Understand bi-variate Gaussian through algebraic expansion of $\triangle$

Concepts needed:

- Recall the definition of **covariance**: $\text{cov}(\mathbf{x}, \mathbf{y}) = \mathbb{E}[(\mathbf{x} - \mu_\mathbf{x})(\mathbf{y} - \mu_\mathbf{y})]$ Intuitively, covariance measures the extent to which two random variables simultaneously (not necessarily w.r.t time) move above and below their means.
- Recall the definition of **matrix inverse**:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \underbrace{\frac{1}{ad-bc}}_{\text{determinant}} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Understanding the expression in the exponent:

- $\mathbf{x}$ is $[\mathbf{x_1}, \mathbf{x_2}]^\mathbf{T}$ (column vector) and $\boldsymbol{\mu}$ is $[\mu_1, \mu_2]^T$ (column vector). $\mathbf{x_1}$ and $\mathbf{x_2}$ can be considered as two random variables. Whether they are independent or dependent is determined by the covariance matrix.

- $\boldsymbol{\Sigma}$ is a 2-by-2 covariance matrix, $\begin{bmatrix} \text{cov}(\mathbf{x}_1, \mathbf{x}_1) & \text{cov}(\mathbf{x}_1, \mathbf{x}_2) \\ \text{cov}(\mathbf{x}_1, \mathbf{x}_2) & \text{cov}(\mathbf{x}_2, \mathbf{x}_2) \end{bmatrix}$.

- $(\mathbf{x} - \boldsymbol{\mu})^T = ([x_1, x_2]^T - [\mu_1, \mu_2]^T)^T = [x_1 - \mu_1, x_2 - \mu_2]$ is a row vector.

-

$$\begin{aligned}
\boldsymbol{\Sigma}^{-1} &= \begin{bmatrix} \text{cov}(\mathbf{x}_1, \mathbf{x}_1) & \text{cov}(\mathbf{x}_1, \mathbf{x}_2) \\ \text{cov}(\mathbf{x}_1, \mathbf{x}_2) & \text{cov}(\mathbf{x}_2, \mathbf{x}_2) \end{bmatrix}^{-1} && (1) \\
&= \frac{1}{\det(\boldsymbol{\Sigma})} \begin{bmatrix} \text{cov}(\mathbf{x}_2, \mathbf{x}_2) & -\text{cov}(\mathbf{x}_1, \mathbf{x}_2) \\ -\text{cov}(\mathbf{x}_1, \mathbf{x}_2) & \text{cov}(\mathbf{x}_1, \mathbf{x}_1) \end{bmatrix} && (2) \\
&= \frac{1}{\text{cov}(\mathbf{x}_1, \mathbf{x}_1)\text{cov}(\mathbf{x}_2, \mathbf{x}_2) - \text{cov}(\mathbf{x}_1, \mathbf{x}_2)^2} \begin{bmatrix} \text{cov}(\mathbf{x}_2, \mathbf{x}_2) & -\text{cov}(\mathbf{x}_1, \mathbf{x}_2) \\ -\text{cov}(\mathbf{x}_1, \mathbf{x}_2) & \text{cov}(\mathbf{x}_1, \mathbf{x}_1) \end{bmatrix} && (3) \\
&= \begin{bmatrix} \frac{\text{cov}(\mathbf{x}_2,\mathbf{x}_2)}{\text{cov}(\mathbf{x}_1,\mathbf{x}_1)\text{cov}(\mathbf{x}_2,\mathbf{x}_2) - \text{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} & -\frac{\text{cov}(\mathbf{x}_1,\mathbf{x}_2)}{\text{cov}(\mathbf{x}_1,\mathbf{x}_1)\text{cov}(\mathbf{x}_2,\mathbf{x}_2) - \text{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} \\ -\frac{\text{cov}(\mathbf{x}_1,\mathbf{x}_2)}{\text{cov}(\mathbf{x}_1,\mathbf{x}_1)\text{cov}(\mathbf{x}_2,\mathbf{x}_2) - \text{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} & \frac{\text{cov}(\mathbf{x}_1,\mathbf{x}_1)}{\text{cov}(\mathbf{x}_1,\mathbf{x}_1)\text{cov}(\mathbf{x}_2,\mathbf{x}_2) - \text{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} \end{bmatrix} && (4)
\end{aligned}$$

- $$(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = [x_1-\mu_1, x_2-\mu_2]\begin{bmatrix} \frac{\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)}{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} & -\frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)}{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} \\ -\frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)}{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} & \frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)}{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} \end{bmatrix}[\mathbf{x}_1-\mu_1, \mathbf{x}_2-\mu_2]^T \tag{5}$$
$$= [(\mathbf{x}_1-\mu_1)^2\frac{\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)}{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} - (\mathbf{x}_1-\mu_1)(\mathbf{x}_2-\mu_2)\frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)}{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} \tag{6}$$
$$- (\mathbf{x}_2-\mu_2)(\mathbf{x}_1-\mu_1)\frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)}{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} + (\mathbf{x}_2-\mu_2)^2\frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)}{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1)\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2)-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2}]$$

- Assume that we are modelling normalized data, that is, data with zero mean ($\mu_1 = 0$, $\mu_2 = 0$) and unit variance, then $\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_1) = 1$ and $\mathrm{cov}(\mathbf{x}_2,\mathbf{x}_2) = 1$. As a result, $-1 \le \mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2) \le 1$ and $0 \le \mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2 \le 1$.

- $$(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) = \mathbf{x}_1^2\frac{1}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} - 2\mathbf{x}_1\mathbf{x}_2\frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} + \mathbf{x}_2^2\frac{1}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} \tag{7}$$
$$= -\frac{1}{2}\mathbf{x}_1^2\frac{1}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} + \mathbf{x}_1\mathbf{x}_2\frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} - \frac{1}{2}\mathbf{x}_2^2\frac{1}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} \tag{8}$$
$$= -a\mathbf{x}_1^2 + b\mathbf{x}_1\mathbf{x}_2 - a\mathbf{x}_2^2 \text{ where } a \text{ and } b \text{ are constants.} \tag{9}$$

- I name the two terms in red **"quadratic terms"** simply because they form concave quadratic surfaces with perfect rotational symmetry about the z-axis.
- I name the term in green **"covariance term"**. It deserves a name of its own because it modifies the shape of quadratic surfaces when added to the quadratic terms. The extent to which it does this depends on the magnitude of covariance.

3.1 Visualize quadratic terms, $-\frac{1}{2}\mathbf{x}_1^2\frac{1}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2} - \frac{1}{2}\mathbf{x}_2^2\frac{1}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2}$

The higher the magnitude of the covariance (regardless or sign, due to the fact that the covariance is squared):

- the higher the magnitude of the quadratic terms (negatively) and
- the sharper the exponentiated surface.

In [5]:

```
def quadratic_term(xs, ys, cov):
    zs = np.zeros(xs.shape)
    for i in range(xs.shape[0]):
        for j in range(xs.shape[1]):
            z = - 0.5 * (1 / (1 - cov**2)) * (xs[i, j]**2 + ys[i, j]**2)  # quadratic term formula
            zs[i][j] = z
    return zs
```

executed in 5ms, finished 17:12:31 2019-08-15

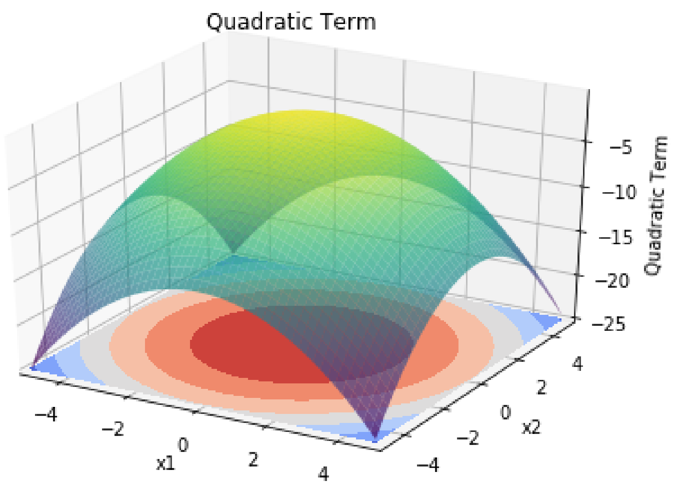
In [6]:

```
quadratic_term_plotter = graph3d_wrapper(quadratic_term,
                                         'x1', 'x2', 'Quadratic Term', 'Quadratic Term',
                                         -5, 5, 0.05, -5, 5, 0.05,
                                         proj=True)
interact_manual(quadratic_term_plotter, cov=FloatSlider(min=-0.9, max=0.9, step=1e-2), continuous_update=False);
```

executed in 40ms, finished 17:12:31 2019-08-15

cov -0.76

Run Interact



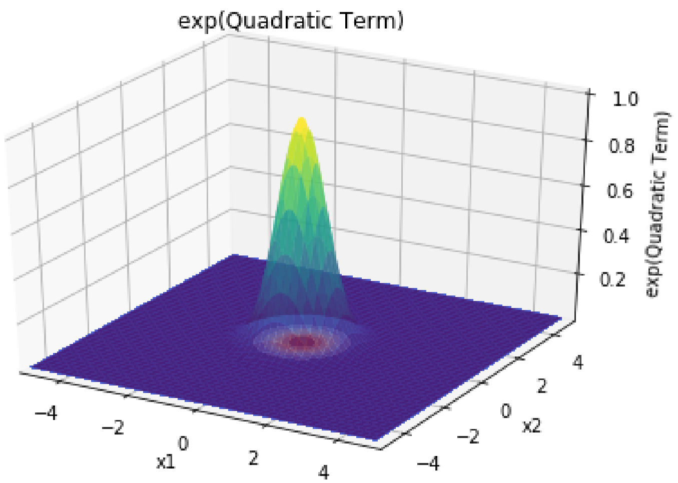
In [7]:

```python
exp_quadratic_term_plotter = graph3d_wrapper(lambda xs, ys, cov : np.e ** quadratic_term(xs, ys, cov),
                                             'x1', 'x2', 'exp(Quadratic Term)', 'exp(Quadratic Term)',
                                             -5, 5, 0.05, -5, 5, 0.05,
                                             proj=True)
_ = interact_manual(exp_quadratic_term_plotter, cov=FloatSlider(min=-0.9, max=0.9, step=1e-2), continuous_update=False)
```

executed in 37ms, finished 17:12:31 2019-08-15

cov -0.78

Run Interact



3.2 Visualize negative covariance term, $\mathbf{x}_1\mathbf{x}_2\frac{\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)}{1-\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2)^2}$ when $\mathrm{cov}(\mathbf{x}_1,\mathbf{x}_2) < 0$

$\mathbf{x}_1\mathbf{x}_2$ is multiplied by a negative scalar.

- The covariance term is positive if $\mathbf{x}_1\mathbf{x}_2$ is negative, which happens in second and fourth quadrants of the outcome space.
- The covariance term is negative if $\mathbf{x}_1\mathbf{x}_2$ is positive, which happens in first and third quadrants of the outcome space.

Quadrants / Term	covariance term	exp(covariance term)
first and third quadrant	negative, grows linearly	converge to zero
second and fourth quadrant	positive, grows linearly	positive, grows exponentially

A similar analysis can be done easily for positive covariance.

In [8]:

```python
def covariance_term(xs, ys, cov):
    zs = np.zeros(xs.shape)
    for i in range(xs.shape[0]):
        for j in range(xs.shape[1]):
            z = xs[i, j] * ys[i, j] * cov / (1 - cov**2)  # covariance term formula
            zs[i][j] = z
    return zs
```

executed in 5ms, finished 17:12:31 2019-08-15

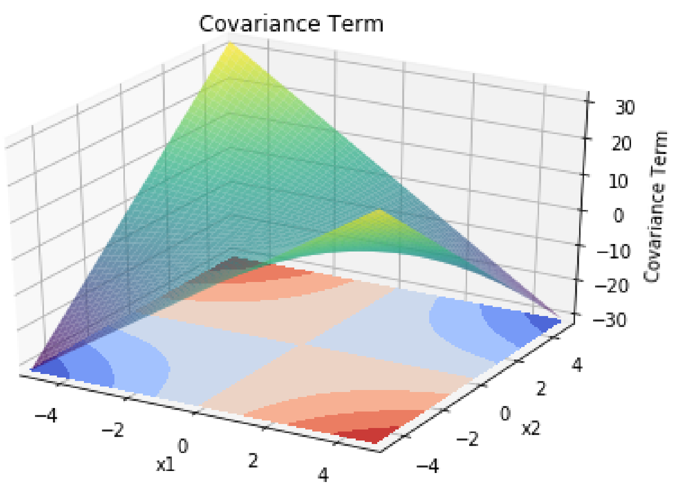
In [9]:

```python
covariance_term_plotter = graph3d_wrapper(covariance_term,
                                          'x1', 'x2', 'Covariance Term', 'Covariance Term',
                                          -5, 5, 0.05, -5, 5, 0.05,
                                          proj=True)
interact_manual(covariance_term_plotter, cov=FloatSlider(min=-0.9, max=0.9, step=1e-2), continuous_update=False);
```

executed in 34ms, finished 17:12:31 2019-08-15

cov -0.68

Run Interact

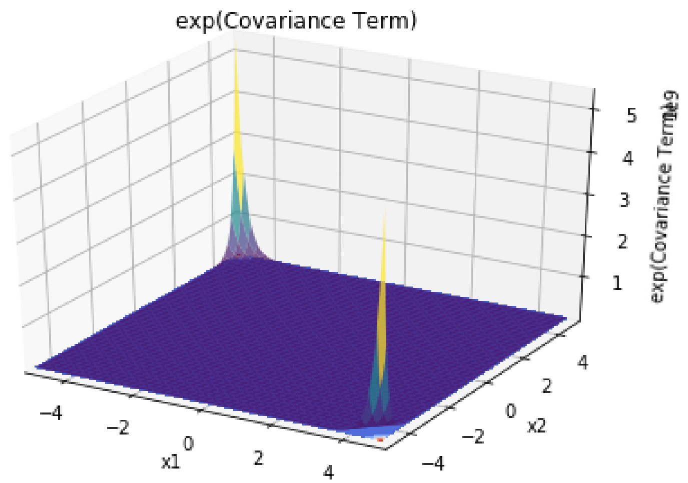


In [10]:
```python
exp_covariance_term_plotter = graph3d_wrapper(lambda xs, ys, cov : np.e ** covariance_term(xs, ys, cov),
                                              'x1', 'x2', 'exp(Covariance Term)', 'exp(Covariance Term)',
                                              -5, 5, 0.05, -5, 5, 0.05,
                                              proj=True)
interact_manual(exp_covariance_term_plotter, cov=FloatSlider(min=-0.9, max=0.9, step=1e-2), continuous_update=False);
```

executed in 49ms, finished 17:12:32 2019-08-15

cov -0.59

Run Interact



3.3 Visualize the quadratic terms and the covariance term together

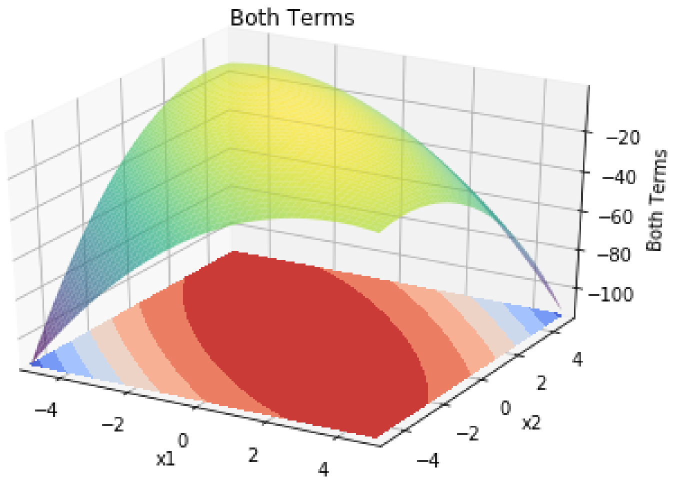
In [11]:

```python
all_terms_plotter = graph3d_wrapper(lambda xs, ys, cov : quadratic_term(xs, ys, cov) + covariance_term(xs, ys, cov),
                                    'x1', 'x2', 'Both Terms', 'Both Terms',
                                    -5, 5, 0.05, -5, 5, 0.05,
                                    proj=True)
interact_manual(all_terms_plotter, cov=FloatSlider(min=-0.9, max=0.9, step=1e-2), continuous_update=False);
```

executed in 37ms, finished 17:12:32 2019-08-15

cov -0.78

Run Interact

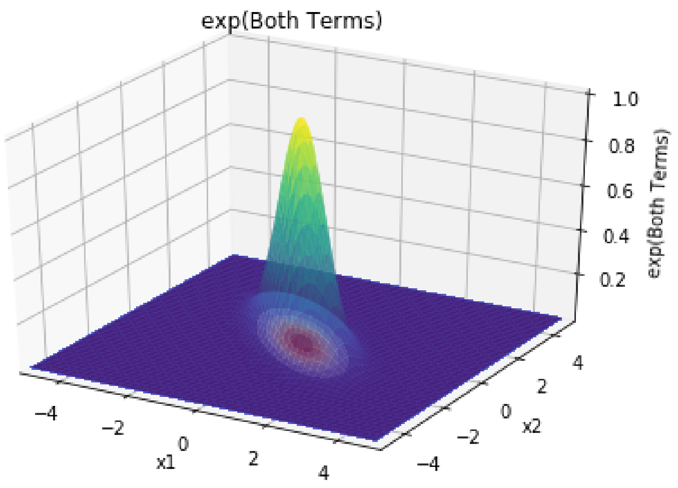


In [12]:
```
exp_all_terms_plotter = graph3d_wrapper(lambda xs, ys, cov : np.e ** (quadratic_term(xs, ys, cov) + covariance_term(xs, ys, cov))
                                        'x1', 'x2', 'exp(Both Terms)', 'exp(Both Terms)',
                                        -5, 5, 0.05, -5, 5, 0.05,
                                        proj=True)
interact_manual(exp_all_terms_plotter, cov=FloatSlider(min=-0.9, max=0.9, step=1e-2), continuous_update=False);
```

executed in 48ms, finished 17:12:32 2019-08-15

cov -0.75

Run Interact



4 References

- Matrix inverse: [https://www.mathsisfun.com/algebra/matrix-inverse.html (https://www.mathsisfun.com/algebra/matrix-inverse.html)](https://www.mathsisfun.com/algebra/matrix-inverse.html)