Exploring functions further

Functional transformations

Shifting

Stretching

Reflecting functions

Even and odd functions

Increasing and decreasing functions

Functional Transformation - Vertical Shift

Consider the graph of the function $y = f(x)$

The graph of another function $y_t = f(x) + k$ shifts the function y by k units.

Adding k shifts upwards and subtracting k shifts downwards

Example - plot the functions

$$y = x^2$$

$$y_t = x^2 - 3$$

Vertical Shift

Plot the functions: $y = x^2$ and $y_t = x^2 - 3$

```python
import matplotlib.pyplot as plt
import numpy as np
########################################
x_lower_limit = -5 #Edit this variable
x_upper_limit = 5 #Edit this variable
x = np.linspace(x_lower_limit, x_upper_limit, num=100)
y =  x**2  #Code up the equation
y_ut = x**2 + 3
y_dt = x**2 - 3
y_lower_limit = -5 #Edit this variable
y_upper_limit = 10 #Edit this variable
########################################
fig, ax = plt.subplots()
ax.plot(x, y, label = 'Original function')
ax.plot(x, y_ut, label = 'Upward shift')
ax.plot(x, y_dt, label = 'Downward shift')
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```

Vertical Shift

Plot the functions: $y = x^2$ and $y_t = x^2 - 3$

```
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```

Horizontal Shift

Consider the graph of the function $y = f(x)$

To shift the graph horizontally by k units, we graph the function

$$y_t = f(x + k)$$

Example - plot the functions

$$y = f(x) = x^2$$

$$y_t = f(x + 3) = (x + 3)^2$$

Horizontal Shift

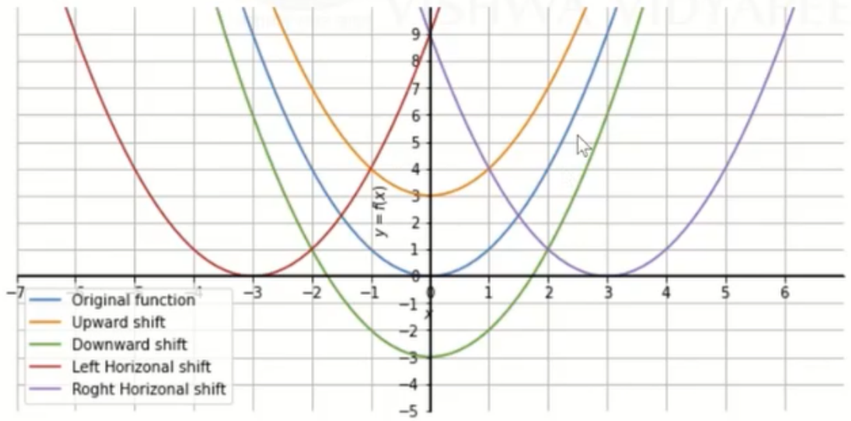
- Plot the functions: $y = x^2$ and $y_t = (x + 3)^2$

```
y_rt = (x-3)**2
y_lower_limit = -5 #Edit this variable
y_upper_limit = 10 #Edit this variable
########################################
fig, ax = plt.subplots()
ax.plot(x, y, label = 'Original function')
ax.plot(x, y_ut, label = 'Upward shift')
ax.plot(x, y_dt, label = 'Downward shift')
ax.plot(x, y_lt, label = 'Left Horizonal shift')
ax.plot(x, y_rt, label = 'Right Horizonal shift')
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```

Horizontal Shift

- Plot the functions: $y = x^2$ and $y_t = (x + 3)^2$

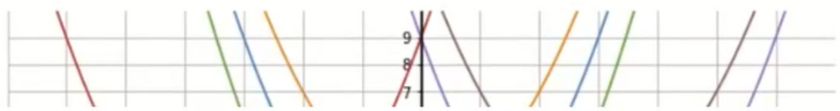
```
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```



Combination of horizontal and vertical shift

- $f(x) \rightarrow$ Vertical shift $f(x) + k \rightarrow$ Horizontal shift $f(x + h) + k$

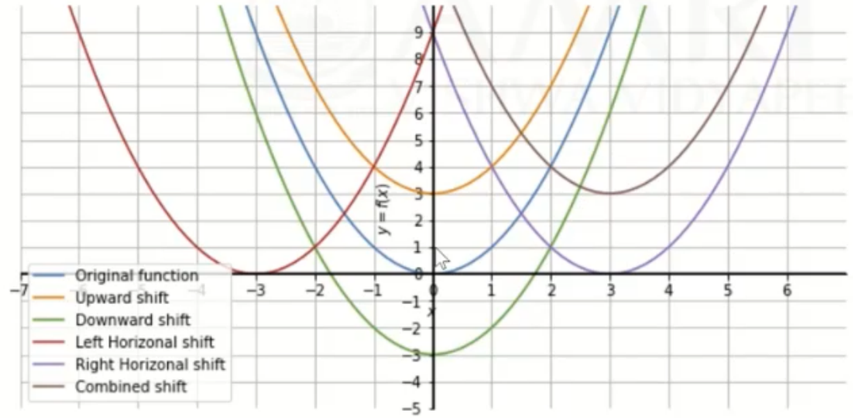
```
y_rt = (x-3)**2
y_ct = (x-3)**2 + 3
y_lower_limit = -5 #Edit this variable
y_upper_limit = 10 #Edit this variable
########################################
fig, ax = plt.subplots()
ax.plot(x, y, label = 'Original function')
ax.plot(x, y_ut, label = 'Upward shift')
ax.plot(x, y_dt, label = 'Downward shift')
ax.plot(x, y_lt, label = 'Left Horizonal shift')
ax.plot(x, y_rt, label = 'Right Horizonal shift')
ax.plot(x, y_ct, label = 'Combined shift')
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```



Combination of horizontal and vertical shift

- $f(x) \rightarrow$ Vertical shift $f(x) + k \rightarrow$ Horizontal shift $f(x + h) + k$

```
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```



Functional transformations - Stretching

Consider the graph of the function $y = f(x)$

Stretching

$y = bf(x)$ $\quad b > 0$

Explore $b > 1$ and $b < 1$

Stretching

Plot the functions: $y = x^2, y_s = 2x^2, y_s = 0.5x^2$

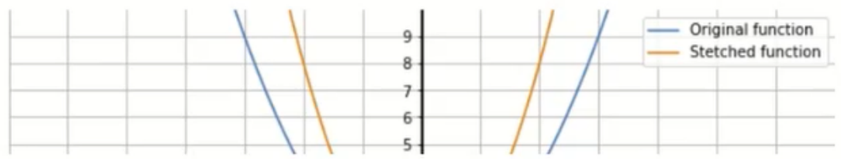
▾ Strectching of the function

```
import matplotlib.pyplot as plt
import numpy as np
########################################
x_lower_limit = -7 #Edit this variable
x_upper_limit = 7 #Edit this variable
x = np.linspace(x_lower_limit, x_upper_limit, num=100)
y =  x**2  #Code up the equation
y_lower_limit = -5 #Edit this variable
y_upper_limit = 10 #Edit this variable
########################################
fig, ax = plt.subplots()
ax.plot(x, y, label = 'Original function')
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
```


Stretching

Plot the functions: $y = x^2, y_s = 2x^2, y_s = 0.5x^2$

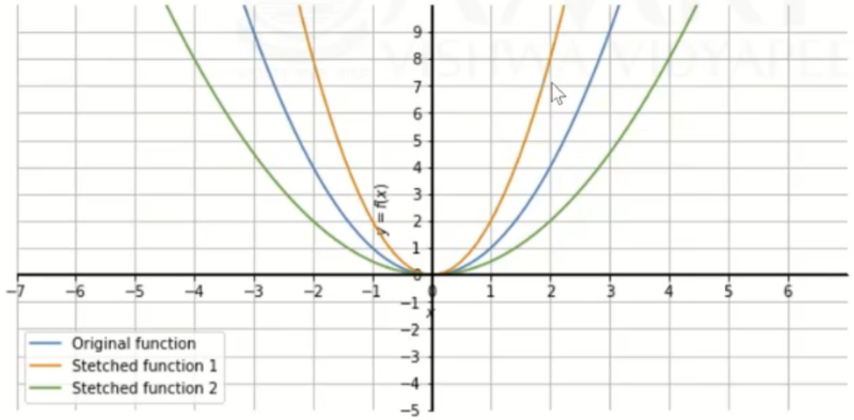
```
x = np.linspace(x_lower_limit, x_upper_limit, num=100)
y =  x**2  #Code up the equation
y_s1 = 2*x**2
y_s2 = 0.5*x**2
y_lower_limit = -5 #Edit this variable
y_upper_limit = 10 #Edit this variable
########################################
fig, ax = plt.subplots()
ax.plot(x, y, label = 'Original function')
ax.plot(x, y_s1, label = 'Stetched function 1')
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```



Stretching

Plot the functions: $y = x^2, y_s = 2x^2, y_s = 0.5x^2$

```
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```



Reflections

Reflection about x axis

$y = f(x) \qquad y_r = -f(x)$

Reflection about y axis

$y = f(x) \qquad y_r = f(-x)$

Examples

$y = f(x) = x^2$

$y = f(x) = 3^x$

Stretching

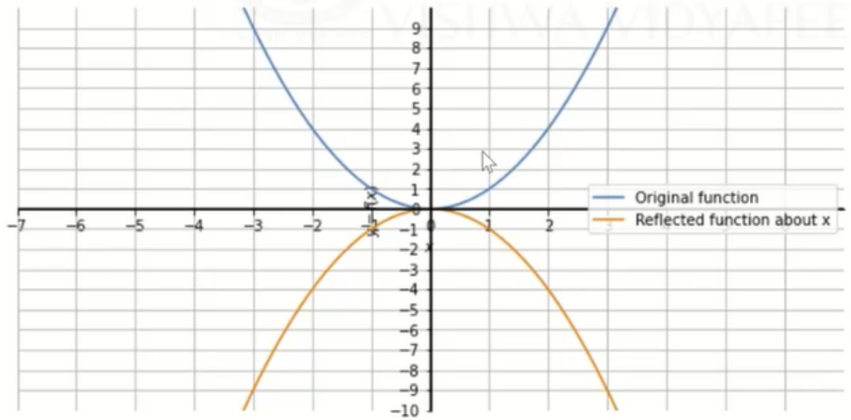
Plot the functions: $y = x^2$ and $y = 3^x$ and reflect about x and y axis

```python
import matplotlib.pyplot as plt
import numpy as np
########################################
x_lower_limit = -7 #Edit this variable
x_upper_limit = 7 #Edit this variable
x = np.linspace(x_lower_limit, x_upper_limit, num=100)
y =  x**2  #Code up the equation
y_rx = -x**2
y_lower_limit = -5 #Edit this variable
y_upper_limit = 10 #Edit this variable
########################################
fig, ax = plt.subplots()
ax.plot(x, y, label = 'Original function')
ax.plot(x, y_rx, label = 'Reflected function about x')
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```

Stretching

Plot the functions: $y = x^2$ and $y = 3^x$ and reflect about x and y axis

```
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```



Stretching

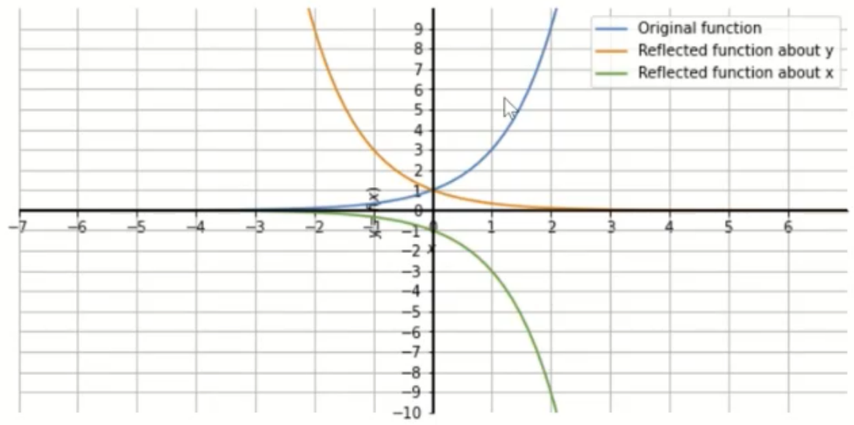
- Plot the functions: $y = x^2$ and $y = 3^x$ and reflect about x and y axis

```python
import numpy as np
########################################
x_lower_limit = -7 #Edit this variable
x_upper_limit = 7 #Edit this variable
x = np.linspace(x_lower_limit, x_upper_limit, num=100)
y =  3**x  #Code up the equation
y_ry = 3**(-x)
y_rx = -3**x
y_lower_limit = -10 #Edit this variable
y_upper_limit = 10 #Edit this variable
########################################
fig, ax = plt.subplots()
ax.plot(x, y, label = 'Original function')
ax.plot(x, y_ry, label = 'Reflected function about y')
ax.plot(x, y_rx, label = 'Reflected function about x')
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```

Stretching

- Plot the functions: $y = x^2$ and $y = 3^x$ and reflect about x and y axis

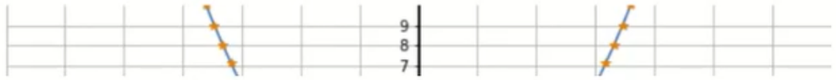
```
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```



Other effects

- Plot the functions: $y = x^2 - 3$ and $y_a = |x^2 - 3|$ (absolute value)

```
import numpy as np
########################################
x_lower_limit = -7 #Edit this variable
x_upper_limit = 7 #Edit this variable
x = np.linspace(x_lower_limit, x_upper_limit, num=100)
y =  x**2 - 3  #Code up the equation
y_a = np.abs(x**2-3)
y_lower_limit = -10 #Edit this variable
y_upper_limit = 10 #Edit this variable
########################################
fig, ax = plt.subplots()
ax.plot(x, y, label = 'Original function')
ax.plot(x, y_a, '*', label = 'Absolute of the original function')
ax.legend()
plt.rcParams['figure.figsize'] = [10, 5]
plt.axhline(color="black")
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```



Other effects

- Plot the functions: $y = x^2 - 3$ and $y_a = |x^2 - 3|$ (absolute value)

```
plt.axvline(color="black")
plt.xlabel('$x$')
plt.ylabel('$y=f(x)$')
ax.spines['top'].set_color('none')
ax.spines['bottom'].set_position('zero')
ax.spines['left'].set_position('zero')
ax.spines['right'].set_color('none')
ax.set_xlim([x_lower_limit,x_upper_limit])
ax.set_ylim([y_lower_limit,y_upper_limit])
ax.set_xticks(np.arange(x_lower_limit, x_upper_limit, 1))
ax.set_yticks(np.arange(y_lower_limit, y_upper_limit, 1))
plt.grid()
plt.show()
```