```python
def find_extents(data_list):
    """  Finds min and max value for each dimension in a DimN data list
         Returns a list of (min, max) per dimension"""

    min_dim = list(data_list[0])
    max_dim = list(data_list[0])

    for data in data_list:
        for i in range(len(data)):
            if min_dim[i] > data[i]:
                min_dim[i] = data[i]
            elif max_dim[i] < data[i]:
                max_dim[i] = data[i]

    for data in data_list:
        for i in range(len(data)):
            if min_dim[i] > data[i]:
                min_dim[i] = data[i]
            elif max_dim[i] < data[i]:
                max_dim[i] = data[i]

    # transpose the result so it is more usable
    ext = []
    for i in range(len(min_dim)):
        ext.append( (min_dim[i], max_dim[i]) )

    return (ext)

def get_canvas_scaling(data_list, dim1, dim2, canvas_height, canvas_width):
    '''  Calculates a scaling factor and x y offset to allow all data points to be drawn on a given canvas '''

    # calls the find_extents function and returns a list of (min, max) per dimension
    extents = find_extents(data_list)

    # get the X max and min from the extents list and work out a scale facter
    xmin, xmax= extents[dim1]
    scaleFactorX = canvas_width/(xmax-xmin)

    # same for for Y
    ymin, ymax= extents[dim2]
    scaleFactorY = canvas_height/(ymax-ymin)

    # choose the smaller of these two scale facter to keep the same aspect ratio
    # multiply by 0.9 to leave soem room around the edges being displayed
    scaleFactor = min(scaleFactorX, scaleFactorY)*0.9

    #calc the translation offsets  ( we want to move the minX and minY to 0,0 in canvas)
    offsetX = (xmin * scaleFactor) - (canvas_width-(xmax-xmin)*scaleFactor)/2
    offsetY = (ymin * scaleFactor) - (canvas_height-(ymax-ymin)*scaleFactor)/2

    return (scaleFactor, offsetX, offsetY)
```