```
>> help subplot
 subplot Create axes in tiled positions.
    H = subplot(m,n,p), or subplot(mnp), breaks the Figure window
    into an m-by-n matrix of small axes, selects the p-th axes for
    the current plot, and returns the axes handle.  The axes are
    counted along the top row of the Figure window, then the second
    row, etc.  For example,
 
        subplot(2,1,1), PLOT(income)
        subplot(2,1,2), PLOT(outgo)
 
    plots income on the top half of the window and outgo on the
    bottom half. If the CurrentAxes is nested in a uipanel the
    panel is used as the parent for the subplot instead of the
    current figure.
 
    subplot(m,n,p), if the axes already exists, makes it current.
    subplot(m,n,p,'replace'), if the axes already exists, deletes it and
    creates a new axes.
    subplot(m,n,p,'align') places the axes so that the plot boxes
    are aligned, but does not prevent the labels and ticks from
    overlapping.
    subplot(m,n,P), where P is a vector, specifies an axes position
    that covers all the subplot positions listed in P.
    subplot(H), where H is an axes handle, is another way of making
    an axes current for subsequent plotting commands.
 
    subplot('position',[left bottom width height]) creates an
    axes at the specified position in normalized coordinates (in
    in the range from 0.0 to 1.0).
 
    subplot(..., PROP1, VALUE1, PROP2, VALUE2, ...) sets the
    specified property-value pairs on the subplot axes. To add the
    subplot to a specific figure pass the figure handle as the
    value for the 'Parent' property.
 
    If a subplot specification causes a new axes to overlap an
    existing axes, the existing axes is deleted - unless the position
    of the new and existing axes are identical.  For example,
    the statement subplot(1,2,1) deletes all existing axes overlapping
    the left side of the Figure window and creates a new axes on that
    side - unless there is an axes there with a position that exactly
    matches the position of the new axes (and 'replace' was not specified),
    in which case all other overlapping axes will be deleted and the
    matching axes will become the current axes.
 
    subplot(111) is an exception to the rules above, and is not
    identical in behavior to subplot(1,1,1).  For reasons of backwards
    compatibility, it is a special case of subplot which does not
    immediately create an axes, but instead sets up the figure so that
    the next graphics command executes CLF RESET in the figure
    (deleting all children of the figure), and creates a new axes in
    the default position.  This syntax does not return a handle, so it
    is an error to specify a return argument.  The delayed CLF RESET
    is accomplished by setting the figure's NextPlot to 'replace'.
    Be aware when creating subplots from scripts that the Position
    property of subplots is not finalized until either a drawnow
    command is issued, or MATLAB returns to await a user command.
    That is, the value obtained for subplot i by the command
    h(i).Position will not be correct until the script
    refreshes the plot or exits.

    See also  gca, gcf, axes, figure, uipanel

>> figure();
>> subplot(121);title('Color');imshow(I);
>> subplot(122);title('Gray');imshow(J);
>> help imwrite
 imwrite Write image to graphics file.
    imwrite(A,FILENAME,FMT) writes the image A to the file specified by
    FILENAME in the format specified by FMT.

    A can be an M-by-N (grayscale image) or M-by-N-by-3 (color image)
    array.  A cannot be an empty array.  If the format specified is TIFF,
    imwrite can also accept an M-by-N-by-4 array containing color data
    that uses the CMYK color space.

    FILENAME is a string that specifies the name of the file.

    FMT is a string specifying the format of the file.  See the reference
    page, or the output of the IMFORMATS function, for a list of
    supported formats.

    imwrite(X,MAP,FILENAME,FMT) writes the indexed image in X and its
    associated colormap MAP to FILENAME in the format specified by FMT.
    If X is of class uint8 or uint16, imwrite writes the actual values in
    the array to the file.  If X is of class double, imwrite offsets the
    values in the array before writing, using uint8(X-1).  MAP must be a
    valid MATLAB colormap.  Note that most image file formats do not
    support colormaps with more than 256 entries.

    When writing multiframe GIF images, X should be an 4-dimensional
    M-by-N-by-1-by-P array, where P is the number of frames to write.

    imwrite(...,FILENAME) writes the image to FILENAME, inferring the
    format to use from the filename's extension. The extension must be
    one of the legal values for FMT.

    imwrite(...,PARAM1,VAL1,PARAM2,VAL2,...) specifies parameters that
    control various characteristics of the output file. Parameters are
    currently supported for GIF, HDF, JPEG, TIFF, PNG, PBM, PGM, and PPM
    files.

    Class Support
    -------------
    The input array A can be of class logical, uint8, uint16, single, or
    double.  Indexed images (X) can be of class uint8, uint16, single, or
    double; the associated colormap, MAP, must be double.  Input values
    must be full (non-sparse).
    The class of the image written to the file depends on the format
    specified.  For most formats, if the input array is of class uint8,
    imwrite outputs the data as 8-bit values.  If the input array is of
    class uint16 and the format supports 16-bit data (JPEG, PNG, and
    TIFF), imwrite outputs the data as 16-bit values.  If the format does
    not support 16-bit values, imwrite issues an error.  Several formats,
    such as JPEG and PNG, support a parameter that lets you specify the
    bit depth of the output data.

    If the input array is of class double, and the image is a grayscale
    or RGB color image, imwrite assumes the dynamic range is [0,1] and
    automatically scales the data by 255 before writing it to the file as
    8-bit values.

    If the input array is of class double, and the image is an indexed
    image, imwrite converts the indices to zero-based indices by
    subtracting 1 from each element, and then writes the data as uint8.

    If the input array is of class logical, imwrite assumes the data is a
    binary image and writes it to the file with a bit depth of 1, if the
    format allows it.  BMP, PNG, or TIFF formats accept binary images as
    input arrays.

    GIF-specific parameters
    -----------------------

    'WriteMode'         One of these strings: 'overwrite' (the default)
                        or 'append'.  In append mode, a single frame is
                        added to the existing file.

    'Comment'           A string or cell array of strings containing a
                        comment to be added to the image.  For a cell
                        array of strings, a carriage return is added
                        after each row.

    'DisposalMethod'    One of the following strings, which sets the
                        disposal method of an animated GIF:
                        'leaveInPlace', 'restoreBG', 'restorePrevious',
                        or 'doNotSpecify'.

    'DelayTime'         A scalar value between 0 and 655 inclusive, which
                        specifies the delay in seconds before displaying
                        the next image.

    'TransparentColor'  A scalar integer.  This value specifies which
                        index in the colormap should be treated as the
                        transparent color for the image.  If X is uint8
                        or logical, then indexing starts at 0.  If X is
                        double, then indexing starts at 1.

    'BackgroundColor'   A scalar integer.  This value specifies which
                        index in the colormap should be treated as the
                        background color for the image and is used for
                        certain disposal methods in animated GIFs.  If X
                        is uint8 or logical, then indexing starts at 0.
                        If X is double, then indexing starts at 1.

    'LoopCount'         A finite integer between 0 and 65535 or the value
                        Inf (the default) which specifies the number of
                        times to repeat the animation.  By default, the
                        animation will continuously loop.  For a value of
                        0, the animation will be played once.  For a
                        value of 1, the animation will be played twice,
                        etc.

    'ScreenSize'        A two element vector specifying the screen height
                        and width of the frame.  When used with
                        'Location', this provides a way to write frames
                        to the image which are smaller than the whole
                        frame.  The remaining values are filled in
                        according to the 'DisposalMethod'.

    'Location'          A two element vector specifying the offset of the
                        top left corner of the screen relative to the top
                        left corner of the image.  The first element is
                        the offset from the top, and the second element
                        is the offset from the left.

    HDF-specific parameters
    -----------------------
    'Compression'  One of these strings: 'none' (the default),
                   'rle' (only valid for grayscale and indexed
                   images), 'jpeg' (only valid for grayscale
                   and RGB images)

    'Quality'      A number between 0 and 100; parameter applies
                   only if 'Compression' is 'jpeg'; higher
                   numbers mean quality is better (less image
                   degradation due to compression), but the
                   resulting file size is larger

    'WriteMode'    One of these strings: 'overwrite' (the
                   default) or 'append'

    JPEG-specific parameters
    ------------------------
    'Quality'      A number between 0 and 100; higher numbers
                   mean quality is better (less image degradation
                   due to compression), but the resulting file
                   size is larger

    'Comment'      A column vector cell array of strings or a
                   char matrix.  Each row of input is written out
                   as a comment in the JPEG file.

    'Mode'         Either 'lossy' (the default) or 'lossless'

    'BitDepth'     A scalar value indicating desired bitdepth;
                   for grayscale images this can be 8, 12, or 16;
                   for truecolor images this can be 8 or 12.  Only
                   lossless mode is supported for 16-bit images.

    JPEG2000-specific parameters
    ----------------------------
    'Mode'             Either 'lossy' (the default) or 'lossless'.

    'CompressionRatio' A real value greater than 1 specifying the target
                       compression ratio which is defined as the ratio of
                       input image size to the output compressed size. For
                       example, a value of 2.0 implies that the output
                       image size will be half of the input image size or
                       less. A higher value implies a smaller file size and
                       reduced image quality. This is valid only with
                       'lossy' mode. Note that the compression ratio
                       doesn't take into account the header size, and hence
                       in some cases the output file size can be larger
                       than expected.

    'ProgressionOrder' A string that is one of 'LRCP', 'RLCP', 'RPCL',
                       'PCRL' or 'CPRL'. The four character identifiers are
                       interpreted as L=layer, R=resolution, C=component
                       and P=position. The first character refers to the
                       index which progresses most slowly, while the last
                       refers to the index which progresses most quickly.
                       The default value is 'LRCP'.

    'QualityLayers'    A positive integer (not exceeding 20) specifying the
                       number of quality layers. The default value is 1.

    'ReductionLevels'  A positive integer (not exceeding 8) specifying the
                       number of reduction levels or the wavelet
                       decomposition levels.

    'TileSize'         A 2-element vector specifying tile height and tile
                       width. The minimum tile size that can be specified
                       is [128 128]. The default tile size is same as the
                       image size.

    'Comment'          A cell array of strings or a char matrix.  Each row
                       of input is written out as a comment in the JPEG2000
                       file.

    TIFF-specific parameters
    ------------------------
    'Colorspace'   One of these strings: 'rgb', 'cielab', or
                   'icclab'.  The default value is 'rgb'.  This
                   parameter is used only when the input array,
                   A, is M-by-N-by-3.  See the reference page
                   for more details about creating L*a*b* TIFF
                   files.

                   In order to create a CMYK TIFF, the colorspace
                   parameter should not be used.  It is sufficient
                   to specify the input array A as M-by-N-by-4.
    'Compression'  One of these strings: 'none', 'packbits'
                   (default for nonbinary images), 'lzw', 'deflate',
                   'jpeg', 'ccitt' (default for binary images),
                   'fax3', 'fax4'; 'ccitt', 'fax3', and
                   'fax4' are valid for binary images only.

                   'jpeg' is a lossy compression scheme; other
                   compression modes are lossless.

                   When using JPEG compression, the 'RowsPerStrip'
                   parameter must be specified and must be a multiple
                   of 8.

    'Description'  Any string; fills in the ImageDescription
                   field returned by IMFINFO

    'Resolution'   A two-element vector containing the
                   XResolution and YResolution, or a scalar
                   indicating both resolutions; the default value
                   is 72

    'RowsPerStrip' A scalar value.  The default will be such that each
                   strip is about 8K bytes.

    'WriteMode'    One of these strings: 'overwrite' (the
                   default) or 'append'

    PNG-specific parameters
    -----------------------
    'Author'       A string

    'Description'  A string

    'Copyright'    A string

    'CreationTime' A string

    'ImageModTime' A MATLAB datenum or a string convertible to a
                   date vector via the DATEVEC function.  Values
                   should be in UTC time.

    'Software'     A string

    'Disclaimer'   A string

    'Warning'      A string

    'Source'       A string

    'Comment'      A string

    'InterlaceType' Either 'none' or 'adam7'

    'BitDepth'     A scalar value indicating desired bitdepth;
```

```
                   for grayscale images this can be 1, 2, 4,
                   8, or 16; for grayscale images with an
                   alpha channel this can be 8 or 16; for
                   indexed images this can be 1, 2, 4, or 8;
                   for truecolor images with or without an
                   alpha channel this can be 8 or 16

    'Transparency' This value is used to indicate transparency
                   information when no alpha channel is used.

                   For indexed images: a Q-element vector in
                     the range [0,1]; Q is no larger than the
                     colormap length; each value indicates the
                     transparency associated with the
                     corresponding colormap entry
                   For grayscale images: a scalar in the range
                     [0,1]; the value indicates the grayscale
                     color to be considered transparent
                   For truecolor images: a 3-element vector in
                     the range [0,1]; the value indicates the
                     truecolor color to be considered
                     transparent

                   You cannot specify 'Transparency' and
                   'Alpha' at the same time.

    'Background'   The value specifies background color to be
                   used when compositing transparent pixels.

                   For indexed images: an integer in the range
                     [1,P], where P is the colormap length
                   For grayscale images: a scalar in the range
                     [0,1]
                   For truecolor images: a 3-element vector in
                     the range [0,1]

    'Gamma'        A nonnegative scalar indicating the file
                   gamma

    'Chromaticities' An 8-element vector [wx wy rx ry gx gy bx
                   by] that specifies the reference white
                   point and the primary chromaticities

    'XResolution'  A scalar indicating the number of
                   pixels/unit in the horizontal direction

    'YResolution'  A scalar indicating the number of
                   pixels/unit in the vertical direction

    'ResolutionUnit' Either 'unknown' or 'meter'

    'Alpha'        A matrix specifying the transparency of
                   each pixel individually; the row and column
                   dimensions must be the same as the data
                   array; may be uint8, uint16, or double, in
                   which case the values should be in the
                   range [0,1]

    'SignificantBits' A scalar or vector indicating how many
                   bits in the data array should be regarded
                   as significant; values must be in the range
                   [1,bitdepth]

                   For indexed images: a 3-element vector
                   For grayscale images: a scalar
                   For grayscale images with an alpha channel:
                     a 2-element vector
                   For truecolor images: a 3-element vector
                   For truecolor images with an alpha channel:
                     a 4-element vector

    In addition to these PNG parameters, you can use any
    parameter name that satisfies the PNG specification for
    keywords: only printable characters, 80 characters or
    fewer, and no leading or trailing spaces.  The value
    corresponding to these user-specified parameters must be a
    string that contains no control characters except for
    linefeed.

    RAS-specific parameters
    -----------------------
    'Type'         One of these strings: 'standard'
                   (uncompressed, b-g-r color order with
                   truecolor images), 'rgb' (like 'standard',
                   but uses r-g-b color order for truecolor
                   images), 'rle' (run-length encoding of 1-bit
                   and 8-bit images)

    'Alpha'        A matrix specifying the transparency of each
                   pixel individually; the row and column
                   dimensions must be the same as the data
                   array; may be uint8, uint16, or double. May
                   only be used with truecolor images.

    PBM, PGM, and PPM-specific parameters
    ------------------------
    'Encoding'     One of these strings: 'ASCII' for plain encoding
                   or 'rawbits' for binary encoding.  Default is 'rawbits'.
    'MaxValue'     A scalar indicating the maximum gray or color
                   value.  Available only for PGM and PPM files.
                   For PBM files, this value is always 1.  Default
                   is 65535 if image array is 'uint16' and 255 otherwise.

    Table: summary of supported image types
    ---------------------------------------
    BMP       1-bit, 8-bit and 24-bit uncompressed images

    GIF       8-bit images

    HDF       8-bit raster image datasets, with or without associated
```

```
              colormap; 24-bit raster image datasets; uncompressed or
              with RLE or JPEG compression

    JPEG      8-bit, 12-bit, and 16-bit Baseline JPEG images

    JPEG2000  1-bit, 8-bit, and 16-bit JPEG2000 images

    PBM       Any 1-bit PBM image, ASCII (plain) or raw (binary) encoding.

    PCX       8-bit images

    PGM       Any standard PGM image. ASCII (plain) encoded with
              arbitrary color depth. Raw (binary) encoded with up
              to 16 bits per gray value.

    PNG       1-bit, 2-bit, 4-bit, 8-bit, and 16-bit grayscale
              images; 8-bit and 16-bit grayscale images with alpha
              channels; 1-bit, 2-bit, 4-bit, and 8-bit indexed
              images; 24-bit and 48-bit truecolor images; 24-bit
              and 48-bit truecolor images with alpha channels

    PNM       Any of PPM/PGM/PBM (see above) chosen automatically.

    PPM       Any standard PPM image. ASCII (plain) encoded with
              arbitrary color depth. Raw (binary) encoded with up
              to 16 bits per color component.

    RAS       Any RAS image, including 1-bit bitmap, 8-bit indexed,
              24-bit truecolor and 32-bit truecolor with alpha.

    TIFF      Baseline TIFF images, including 1-bit, 8-bit, 16-bit,
              and 24-bit uncompressed images, images with packbits
              compression, images with LZW compression, and images
              with Deflate compression; 8-bit and 24-bit images with
              JPEG compression; 1-bit images with CCITT 1D, Group 3,
              and Group 4 compression; CIELAB, ICCLAB, and CMYK images.

    XWD       8-bit ZPixmaps

    Please read the file libtiffcopyright.txt for more information.

    See also imfinfo, imread, imformats, fwrite, getframe.

>> imwrite(J, 'Lenna_gray.jpg', 'JPEG')

>> imwrite(J, './images/Lenna_gray.jpg', 'JPEG');
>> help hist;
 hist  Histogram.
    hist is not recommended. Use HISTOGRAM instead.

    N = hist(Y) bins the elements of Y into 10 equally spaced containers
    and returns the number of elements in each container.  If Y is a
    matrix, hist works down the columns.

    N = hist(Y,M), where M is a scalar, uses M bins.
    N = hist(Y,X), where X is a vector, returns the distribution of Y
    among bins with centers specified by X. The first bin includes
    data between -inf and the first center and the last bin
    includes data between the last bin and inf. Note: Use HISTC if
    it is more natural to specify bin edges instead.

    [N,X] = hist(...) also returns the position of the bin centers in X.

    hist(...) without output arguments produces a histogram bar plot of
    the results. The bar edges on the first and last bins may extend to
    cover the min and max of the data unless a matrix of data is supplied.

    hist(AX,...) plots into AX instead of GCA.

    Class support for inputs Y, X:
       float: double, single

    See also histogram, histcounts, mode.

    Other functions named hist

>> hist(J, 256);
Error using  .*
Integers can only be combined with integers of the same class, or scalar doubles.

Error in linspace (line 33)
    y = d1 + (0:n1).*(d2 - d1)./n1;

Error in hist (line 96)
        edges = linspace(miny,maxy,x+1);

>> hist(J);
Error using  .*
Integers can only be combined with integers of the same class, or scalar doubles.

Error in linspace (line 33)
    y = d1 + (0:n1).*(d2 - d1)./n1;

Error in hist (line 96)
        edges = linspace(miny,maxy,x+1);

>> K = rgb2gray(J);
Error using rgb2gray>parse_inputs (line 80)
MAP must be a m x 3 array.

Error in rgb2gray (line 52)
isRGB = parse_inputs(X);

>>
```