Installed Font Test

This file shows the fonts installed and accessible to Quarto/Matplotlib. It's used partly to audit the latest Docker/Podman image release and partly to allow students to explore what is available in a reasonably intuitive fashion.

Raw List

Taken directly from Matplotlib's font cache:

```
! grep '"name"' ~/.cache/matplotlib/fontlist-v390.json | sort | uniq
```

```
      "name": "Big Shoulders Display",
      "name": "Bitstream Charter",
      "name": "Computer Modern",
      "name": "Courier",
      "name": "Cousine",
      "name": "DejaVu Sans Display",
      "name": "DejaVu Sans Mono",
      "name": "DejaVu Sans",
      "name": "DejaVu Serif Display",
      "name": "DejaVu Serif",
      "name": "EB Garamond",
      "name": "Font Awesome 5 Brands",
      "name": "Font Awesome 5 Free",
      "name": "Font Awesome 6 Brands",
      "name": "Font Awesome 6 Free",
      "name": "Hanken Grotesk",
      "name": "Helvetica",
      "name": "ITC Avant Garde Gothic",
      "name": "ITC Bookman",
      "name": "ITC Zapf Chancery",
      "name": "ITC Zapf Dingbats",
      "name": "Inconsolata Condensed",
      "name": "Inconsolata Expanded",
      "name": "Inconsolata ExtraCondensed",
      "name": "Inconsolata ExtraExpanded",
      "name": "Inconsolata SemiCondensed",
      "name": "Inconsolata SemiExpanded",
      "name": "Inconsolata UltraCondensed",
      "name": "Inconsolata UltraExpanded",
      "name": "Inconsolata",
```

```
"name": "Inter Tight",
"name": "LMMono10",
"name": "LMMonoCaps10",
"name": "LMMonoLt10",
"name": "LMMonoLtCond10",
"name": "LMMonoProp10",
"name": "LMMonoPropLt10",
"name": "LMMonoSlant10",
"name": "LMRoman10",
"name": "LMRomanCaps10",
"name": "LMRomanDemi10",
"name": "LMRomanDunh10",
"name": "LMRomanSlant10",
"name": "LMRomanUnsl10",
"name": "LMSans10",
"name": "LMSansDemiCond10",
"name": "LMSansQuot8",
"name": "Liberation Mono",
"name": "Liberation Sans",
"name": "Liberation Serif",
"name": "Lobster",
"name": "Micro 5",
"name": "New Century Schoolbook",
"name": "Palatino",
"name": "Roboto Flex",
"name": "Roboto Mono",
"name": "Roboto Slab",
"name": "STIXGeneral",
"name": "STIXNonUnicode",
"name": "STIXSizeFiveSym",
"name": "STIXSizeFourSym",
"name": "STIXSizeOneSym",
"name": "STIXSizeThreeSym",
"name": "STIXSizeTwoSym",
"name": "Silkscreen",
"name": "Source Code Pro",
"name": "Source Sans 3",
"name": "Source Serif 4 18pt",
"name": "Source Serif 4 36pt",
"name": "Source Serif 4 48pt",
"name": "Source Serif 4",
"name": "Spectral SC",
"name": "Spectral",
"name": "Symbol",
"name": "Times",
"name": "Ubuntu Condensed",
"name": "Ubuntu Mono",
"name": "Ubuntu",
"name": "Utopia",
"name": "ZapfDingbats",
"name": "cmb10",
"name": "cmex10",
```

```
        "name": "cmmi10",
        "name": "cmr10",
        "name": "cmss10",
        "name": "cmsy10",
        "name": "cmtt10",
```

Formatted List

Taken from Matplotlib's font manager tool.

Please note that fonts may not render correctly in Safari, but will do so in Google Chrome and other browsers. The *list* is correct and these fonts *can* be used to render PDFs from Quarto.

```
from matplotlib import font_manager
from IPython.core.display import HTML

flist = font_manager.findSystemFonts()
names = []
for fname in flist:
    try:
        names.append(font_manager.FontProperties(fname=fname).get_name())
    except RuntimeError:
        print(f"- Problem detected with {fname}, skipping...") # Think the issue is

print(f"Found {len(set(names))} valid fonts.")

def make_html(fontname):
    return "<p><span style='font-family:{font}; font-size: 20px;'>{font}</span></p>"

code = "\n".join([make_html(font) for font in sorted(set(names))])

HTML("<div style='column-count: 2;'>{}</div>".format(code))
```

```
Found 42 valid fonts.
```

```
<IPython.core.display.HTML object>
```

Plotting Test

A quick test to ensure that a font is rendering correctly.

```
import numpy as np
import matplotlib.pyplot as plt
x = np.random.random(20)
y = np.random.random(20)

ff = 'SPectral SC'
tfont = {'fontname':ff}
bfont = {'fontname':ff, 'weight':'bold', 'horizontalalignment':'left'}
afont = {'fontname':ff}
```

```
10
11 f,ax = plt.subplots(1,1,figsize=(7,5))
12 plt.scatter(x,y)
13 f.suptitle("Testing Font Spec", x=0.025, ha='left', size=24, **tfont)
14 ax.set_ylabel('Median Price', size=20, **afont)
15 ax.set_xlabel('Count', size=20, **afont)
```

```
Text(0.5, 0, 'Count')
```