Project #3: Advanced Displays

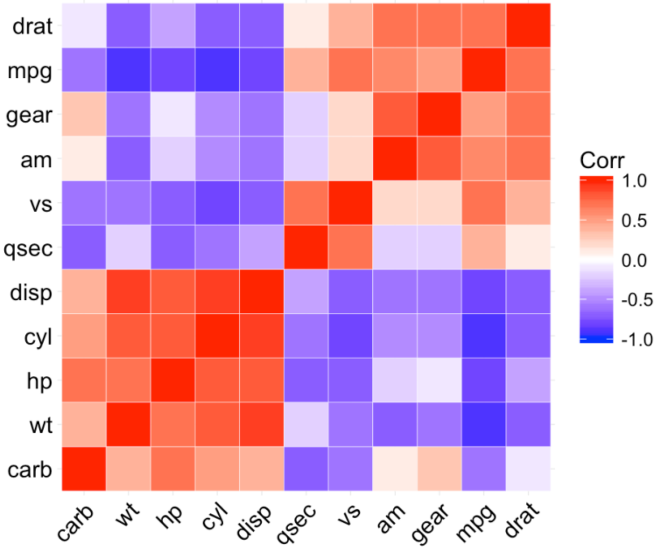
Theme: compare several visualization techniques for high-D data

- use D3 for visualization and python for analysis when needed
- use the data you selected with your 8 favorite attributes
- numerical data only, some parts mention categorical variables

Make separate web pages for the following (10 points for each):

1. 8×8 correlation matrix (map positive/negative correlations to red/blue with intensity indicating correlation strength)
2. 5×5 scatter plot matrix (choose attributes with greatest aggregated correlation strength, see next slide), add 2 categorical variables of your choice for a 7×7 scatterplot matrix
3. parallel coordinates display with 8 axes (choose pairs by correlation strength, see next slide), add 2 categorical variables for 2 more axes
4. PCA plot (top 2 eigenvectors) with associated scree plot (8 bars)
5. biplot with 8 projected axes (project all into top 2 PCA vectors)
6. MDS display of the data (use Euclidian distance)
7. MDS display of the attributes (use 1-|correlation| distance)

Some Notes



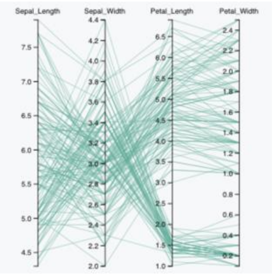
Correlation matrix

- the colors should look like this

Scatterplot matrix plot selection

- add |correlation| along each correlation matrix column
- pick the 5 attributes with the highest sums and display

Parallel coordinates display axes ordering scheme



- pick pair with greatest |correlation| → axes A1, A2
- axis A1 is the attribute with highest correlation sum
- axis A3 is the attribute that has the highest |correlation| with A2
- axis A4 is the attribute that has the highest |correlation| with A3
- and so on….

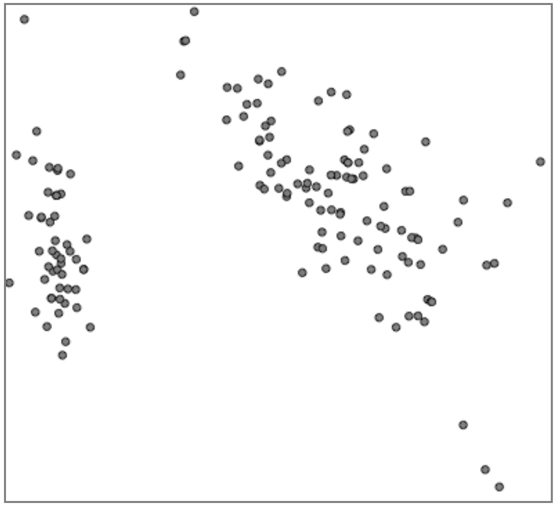
More Notes

Scree plot

- use the bar charts you already have

MDS plots

- should look like this
- we will add cluster information in lab 5



Libraries

- you can precompute correlations, PCA, and MDS in python and read them as data files (not images!!) into your webpage
- The next slides will have information on the use of the libraries

Correlations

Install pandas

- pip install pandas numpy

Import pandas

- import pandas as pd

Load the data from a CSV file into a pandas dataframe

- file_path = 'your_data_file.csv’
- df = pd.read_csv(file_path)

Compute the correlation matrix

- correlation_matrix = df.corr()

PCA

Install scikit-learn and numpy

- pip install scikit-learn numpy

Import numpy

- import numpy as np

Load the data with np.load

Be sure you standardize the data first

- use theStandardScaler from sklearn.preprocessing

Use the PCA class from the sklearn.decomposition module

- principal_components contains the PCA vectors (each row corresponds to a principal component) – you can use them to project your data
- explained_variance contains the lambda values for each component (eigenvalues) – you can use them to plot in a scree plot

MDS

Use the sklearn.manifold library

- from sklearn.manifold import MDS

Preparations

- read the csv datafile using a pandas dataframe
- compute the distance matrix using the pairwise_distances routine

Compute metric MDS

- mds = MDS(n_components=2, dissimilarity='precomputed', random_state=<some integer seed number you can choose>)
- mds_result = mds.fit_transform(distance_matrix).
- the 1st line sets up the MDS object, the 2nd produces the result
- the result holds 2-D coordinates of scatterplot points

Deliverables

Submit by Thursday, October 24, 11:59 pm

- **report** discussing pros and cons for each of the seven displays (20 pts)
- relate these observation to your data
- are there any interesting findings you can make?
- what information of your data do these displays show well
- what information can't they show

- **video** that shows all capabilities of your interface

- **archive file** (zip, rar, tar) of your code and data

Point decomposition (the two w's of lab 3 execution)

- 8 points – works (does the job)
- 2 points – wow (does the job nicely)