Lecture 11:

• Histograms (briefly)
• Linear Regression
• Normalizing values

Announcements:

• HW-3 out, due next Tues (extended)
• HW-4 out soon
• Quiz 5 Tues (types of visualizations, binning)
Histograms

Similar to a bar chart, but for frequencies (counts)

The main function:

\[
\text{pyplot.hist}(xs, \text{bins}=n, \ldots \text{other args})
\]

- default is \( n=10 \) bins \ldots \text{equal-width!}

More information:

https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html

Lots of settings, variations (including 2D versions), etc.
Example:

```python
# reset figure
plt.figure()

# create some data
xs = [random.randint(0, 1+round(i/2)) for i in range(1,200)]

# add y-axis grid
plt.grid(axis='y', color='0.85', zorder=0)

# create the bar chart (rwidth is relative bar width)
plt.hist(xs, bins=20, alpha=0.5, color='b', rwidth=0.8, zorder=3)

# add labels
plt.xlabel('Feature')
plt.ylabel('Counts')
plt.title('Example histogram (20 bins)')

# show the plot
plt.show()
```
Linear Regression

In scatter plots, can be nice to “fit a line”

- the “best fit” line is sometimes called a linear regression
- can be helpful to determine if two features are (linearly) correlated

The basic idea: Find a line that “best” fits a collection of \((x, y)\) data points

- where a line is defined as: \(y = mx + b\) \hspace{1cm} \(m = \text{slope}, b = \text{intercept}\)
- goal is to find good \(m\) and \(b\) values given the data points

We will use ordinary least squares: \hspace{1cm} \(\text{... simple linear regression}\)

- error: the vertical distance from an \((x, y)\) point to the line
- goal: find a slope \(m\) and intercept \(b\) that minimizes the errors (distances)
- least squares: find an \(m\) that minimizes the sum of the squared distances
- note that once we find a slope \(m\), finding \(b\) isn’t difficult
Simple “algorithm”: Assuming \( n \) data points

1. Calculate the mean \( \bar{x} \) of the \( x \) values and the mean \( \bar{y} \) of the \( y \) values
   - note the line must go through the point \((\bar{x}, \bar{y})\)
   - this point is right in the “middle” of the scatter plot
   - the line needs to “cross through” \( \bar{x} \), and at \( \bar{x} \) the minimum error is at \( \bar{y} \)

2. Calculate the slope using the means

\[
m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}
\]

- what is \( m \) if at every point, the \( x - \bar{x} \) and \( y - \bar{y} \) distances are the same?
- at each point, \( m \) “shifted” if \( y \)’s distance to \( \bar{y} \) differs from \( x \)’s distance to \( \bar{x} \)

3. Calculate the \( y \) intercept as \( b = \bar{y} - m\bar{x} \)
   - since we know the line passes through \((\bar{x}, \bar{y})\)
   - just plug in \( \bar{x} \) and \( \bar{y} \) into line formula \( (y = mx + b) \) to get \( b \)
Note that the “best fit” line may not be a “good fit” line ...

- the two variables might not form a linear relationship
- there are various metrics to test a linear “correlation”

1. **Covariance** is defined as:

\[
cov(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n}
\]

- measures how \(x\) and \(y\) vary together
- note that \(cov(x, x)\) is the variance of \(x\)
- similar trends to \(m\) in terms of positive, negative, and zero values

2. **Correlation coefficient** \(r\) to check strength of linear relationship:

\[
r = \frac{cov(x, y)}{\sigma_x \sigma_y}
\]

- where \(\sigma\) is the standard deviation
- called the “Pearson correlation coefficient”

Note that the above is equivalent to:

\[
r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}
\]

- if correlation is perfectly linear, then \(r = 1\)
- if correlation is perfectly inverse linear, then \(r = -1\)
- if no relationship, the \(r = 0\)
2. **Standard error** also used to help check fit

$$ stderr = \sqrt{\frac{\sum(y_i - y'_i)^2}{n}} $$

- where $y'$ is the “predicted” value and $y$ is the actual value
- $(y - y')$ is called a **residual**
- lower standard error values imply “better” fit (correlation)

Plus more along these lines (looking at the distribution of the “residuals”)

Q: What does it mean if a strong (linear) correlation exists between features?

- the features are potentially redundant because one is implied by the other
- i.e., one is a good predictor for the other
- *note*: regression is one way to make predictions (more later)
Example with correlation coefficient

```python
# allow latex math symbols, reset the figure
plt.rcParams['text.usetex'] = True
plt.figure()

# compute the distributions
n, s = 150, round(100 * .3)
xs = [i + random.randint(-s, s) for i in range(s,n+s)]
ys = [i + random.randint(-s, s) for i in range(s,n+s)]

# add the grid and scatter plot
plt.grid(color='0.85', zorder=0)
plt.plot(xs, ys, color='b', marker='.', alpha=0.2, markersize=16,
        linestyle='', zorder=3)

# add the mean lines
plt.axvline(avg(xs), color='r', linestyle='--', label=r'$\bar x$', alpha=0.4)
plt.axhline(avg(ys), color='r', linestyle='--', label=r'$\bar y$', alpha=0.4)

# add the regression line
m, b = least_squares(xs, ys)
line_xs = [i for i in range(min(xs),max(xs))]
line_ys = [m*i + b for i in range(min(xs),max(xs))]
plt.plot(line_xs, line_ys, color='g', label='best fit')

# add the correlation coefficient
r_xy = round(correlation(xs, ys), 2)
txt = r'$r = {}$'.format(r_xy)
plt.text(max(xs), min(ys), txt, color='red')

# add labels, title, legend, and show
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Example scatter plot')
plt.legend()
plt.show()
```
Normalizing Values

Convert continuous-valued features (interval or ratio) to values between 0 and 1

- can make it easier to compare features
- can make it easier to use features in prediction algorithms
- avoids feature scales from influencing results (e.g., 0-105 vs 50,000-200,000)

Assume a feature with values \( x \in X \) s.t. \( \min(X) \leq x \leq \max(X) \)

Q: Write a formula that normalizes a given value \( x \in X \)

- note that the normalized \( \min(X) \) should be 0
- and the normalized \( \max(X) \) should be 1

\[
\text{normalized}(x) = \frac{x - \min(X)}{\max(X) - \min(X)}
\]