Density Estimation in R

Henry Deng & Hadley Wickham
Density Estimation

- Focus on univariate, nonparametric
- Helps reveal underlying distributions
- Applicable in real-life scenarios
- Utility as intermediate step for other calculations
Motivation

- Over 25 packages in R that contain density estimation functions
  - Fifteen suitable for our specific needs
- Provide how and how well packages worked
- Packages rely on differing mathematical theoretical approaches
- Wanted to evaluate performance among the density estimation functions in the packages
- Benefits standard R users, developers
Procedures

- Identify which packages to study
- Theoretical overview of all packages
  - Reference manuals, articles, books
- Compare calculation speed and accuracy
  - Run tests to evaluate performance
- Summarize findings and investigate other ideas
  - Link theory and performance
## Packages Studied

<table>
<thead>
<tr>
<th>Package</th>
<th>Function</th>
<th>Dimensions</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASH</td>
<td>ash1</td>
<td>2</td>
<td>ASH</td>
</tr>
<tr>
<td>base</td>
<td>density</td>
<td>1</td>
<td>Kernel</td>
</tr>
<tr>
<td>ftnonpar</td>
<td>pmden</td>
<td>1</td>
<td>Taut String</td>
</tr>
<tr>
<td>GenKern</td>
<td>KernSec</td>
<td>2</td>
<td>Kernel</td>
</tr>
<tr>
<td>gss</td>
<td>dssden</td>
<td>≥1</td>
<td>Penalized</td>
</tr>
<tr>
<td>MASS</td>
<td>hist</td>
<td>1</td>
<td>Histogram</td>
</tr>
<tr>
<td>kerdiest</td>
<td>kde</td>
<td>1</td>
<td>Kernel</td>
</tr>
<tr>
<td>KernSmooth</td>
<td>bkde</td>
<td>2</td>
<td>Kernel</td>
</tr>
<tr>
<td>ks</td>
<td>kde</td>
<td>6</td>
<td>Kernel</td>
</tr>
<tr>
<td>locfit</td>
<td>density.lf</td>
<td>1</td>
<td>Local Likelihood</td>
</tr>
<tr>
<td>logspline</td>
<td>dlogspline</td>
<td>1</td>
<td>Penalized</td>
</tr>
<tr>
<td>np</td>
<td>npudens</td>
<td>1</td>
<td>Kernel</td>
</tr>
<tr>
<td>pendensity</td>
<td>pendensity</td>
<td>1</td>
<td>Penalized</td>
</tr>
<tr>
<td>plugdensity</td>
<td>plugin.density</td>
<td>1</td>
<td>Kernel</td>
</tr>
<tr>
<td>sm</td>
<td>sm.density</td>
<td>3</td>
<td>Kernel</td>
</tr>
</tbody>
</table>
Theoretical Approach

- Methods for Density Estimation
  - Histogram Approach
  - Kernel Density Estimation
  - Other techniques
    - Penalized Methods, Taut Strings, Splines

\[
\text{ASH: } \hat{f}(x; m) = \frac{1}{nh} \sum_{|i|<m} w_m(i) v_{k+i}
\]

\[
\text{KDE: } \hat{f}(x, H) = \frac{1}{n} \sum_{i=1}^{n} K_H(x - x_i)
\]
Calculation Speed

- **Procedure**
  - Random set of $n$ normally distributed points
  - Increasing number of points ($n$)
  - Multiple trials

- **Timing**
  - Microbenchmark package to record time
  - Measures nanoseconds
Estimation Accuracy

- Specifications
  - Distribution: uniform, normal, claw
  - Grid density evaluation points by 512
  - Used default parameters: automatic bandwidth selection, etc
  - Increasing number of data points
  - Multiple trials

- Measuring Error
  - Mean Absolute Error
  - Mean Squared Error
Additional Ideas

- Tradeoff between speed and accuracy
- Differences from uniform, normal and claw
- Impact of package update frequency
- Which theoretical approaches worked well?
  - Histograms, KDE’s, other approaches
Relative Calculation Time and Accuracy of Packages for Normally Distributed Points

- **ash**
- **ftnonpar_pm**
- **genkern**
- **gss**
- **hist**
- **kerdiest**
- **kernsmooth**
- **ks**
- **lgspline**
- **locfit**
- **np**
- **pendensity**
- **plugdensity**
- **sm**
- **stat**

**Mean Absolute Error** vs **Calculation Time (seconds)**
Package Updates

The diagram shows the last modified dates for various packages. The packages are color-coded and displayed over a timeline from 1999 to 2011.
Conclusion

- Best packages are fast, accurate, and regularly updated without a speed/accuracy tradeoff
- Recommended packages: KernSmooth or ASH
  - KernSmooth uses binned KDE for speed
  - ASH uses averaged shifted histograms
- Extensions
  - Multivariate scenarios
  - Other kinds of density estimation
- Paper to be submitted to JSS
Optional Slides
(not used in presentation)
<table>
<thead>
<tr>
<th>Package</th>
<th>Speed</th>
<th>Accuracy</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASH density</td>
<td>1</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>ftnonpar</td>
<td>4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>GenKern</td>
<td>10</td>
<td>15</td>
<td>11</td>
</tr>
<tr>
<td>gss</td>
<td>12</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>hist</td>
<td>3</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>kerdiest</td>
<td>13</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>KernSmooth</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>ks</td>
<td>9</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>locfit</td>
<td>6</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>logspline</td>
<td>7</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>np</td>
<td>14</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>pendensity</td>
<td>15</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>plugdensity</td>
<td>11</td>
<td>6</td>
<td>14</td>
</tr>
<tr>
<td>sm</td>
<td>5</td>
<td>12</td>
<td>5</td>
</tr>
</tbody>
</table>
Time Vs. Accuracy Plot

name_scale <- scale_colour_hue("Package", limits = unique(all$name), legend = FALSE)
mpe_scale <- scale_y_log10("Mean Absolute Error", limits = c(10^-4, 10^4), breaks = 10^c(-4, -2, 0, 2, 4))

ggplot(norm, aes(med_time, med, colour = name)) +
  geom_line(data = transform(norm, id = name, name = NULL), colour = "grey50", aes(group = id)) +
  geom_line(size = 1.5) +
  geom_point(size = 3) +
  scale_x_log10("Calculation Time (seconds)") +
  scale_y_log10() +
  facet_wrap(~ name) +
  name_scale +
  mpe_scale +
  opts(title = "Relative Calculation Time and Accuracy of Packages for Normally Distributed Points")