Density Estimation Lab

Dennis Do

Stat 410, Module 6

In this lab, we will consider the `galaxies` data set from the package `MASS`, see Exercise 8.1:

The `galaxies` data set presents the velocities of 82 galaxies from six well-separated conic sections of space (Postman et al., 1986; Roeder, 1990). The data are intended to shed light on whether or not the observable universe contains superclusters of galaxies surrounded by large voids. The evidence for the existence of superclusters would be the multimodality of the distribution of velocities. We will use kernel density estimations to smooth histogram visualizations of the velocities distribution. The question of interest is: What may we conclude about the possible existence of superclusters of galaxies?

In Task 1 we will perform a brief exploratory data analysis. In Task 2, we will study the impact of kernels on the density estimate. In Task 3, we will study the impact of bandwidth on the density estimate. Throughout, keep in mind that our primary goal is to assess the existence of multiple modes in the data, thus suggesting potential superclusters of galaxies.

Task 1: Survey the data

The data set consists of a single variable (univariate), galaxy velocities. Let us look at some descriptive statistics.

Code set-up

- Read data: the data is in the MASS package as `galaxies`, list of velocities.

```
galaxies
```

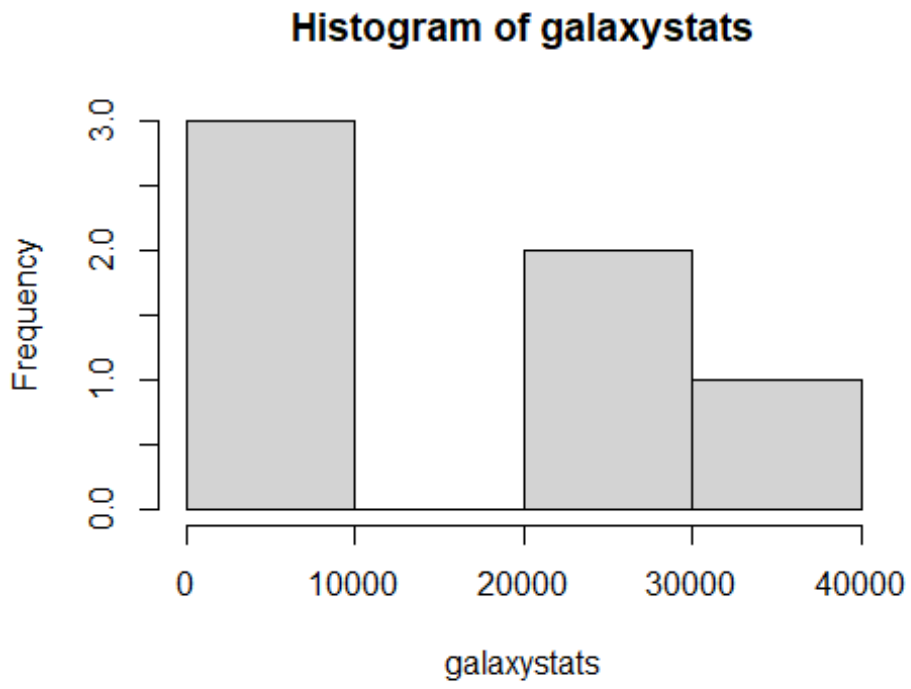
```
##  [1]  9172  9350  9483  9558  9775 10227 10406 16084 16170 18419 18552
18600
## [13] 18927 19052 19070 19330 19343 19349 19440 19473 19529 19541 19547
19663
## [25] 19846 19856 19863 19914 19918 19973 19989 20166 20175 20179 20196
20215
## [37] 20221 20415 20629 20795 20821 20846 20875 20986 21137 21492 21701
21814
## [49] 21921 21960 22185 22209 22242 22249 22314 22374 22495 22746 22747
22888
## [61] 22914 23206 23241 23263 23484 23538 23542 23666 23706 23711 24129
24285
## [73] 24289 24366 24717 24990 25633 26690 26995 32065 32789 34279
```

```
# sample size
n = length(galaxies)
# [Code descriptive statistics here]
```

```
galaxystats = c(min(galaxies), max(galaxies), mean(galaxies),
median(galaxies), IQR(galaxies), sd(galaxies))
pander(galaxystats)
```

9172, *34279*, *20828*, *20834*, *3601* and *4564*

```
hist(galaxystats)
```



Report the following

- Present summary statistics of the galaxy velocities: min, max, mean, median, interquartile range, standard deviation.
- Present a histogram of the distribution of velocities. Make sure to label the graphs.
- What do you see? Comment on the skew, spread, middle, and modes of the distribution.
- Comment on the impact of the default number of bins in this histogram produced by the R function `hist`. Is it too few, too many, appropriate for interpretation?
- Change the number of bins to a value you think more appropriate for a visual interpretation. Make sure to label the graph, including stating the number of bins used in the title. Does this new histogram confirm your observations in bullet 3 above, particularly relative to the number of modes? Why or why not?

Task 2: Kernel density estimates with different kernels

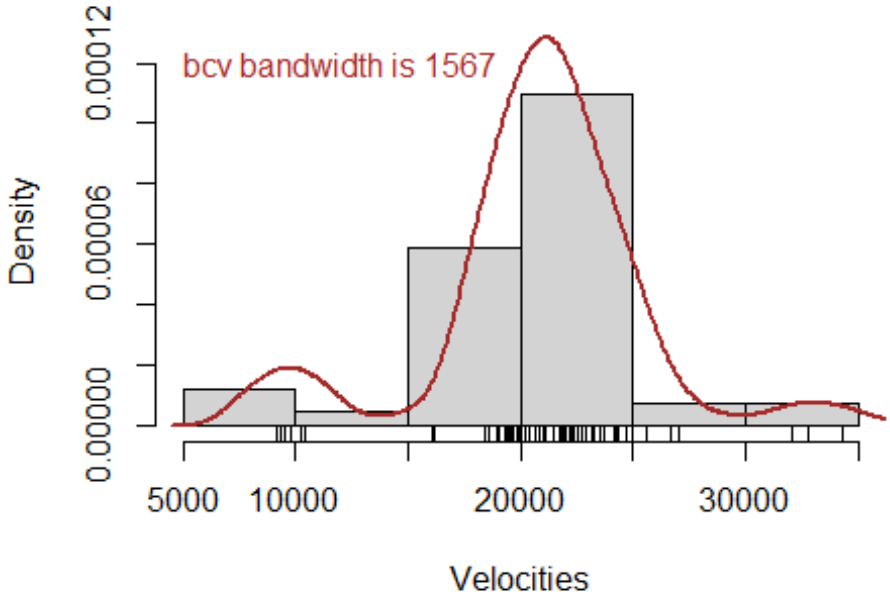
Recall that the kernel used to weight the data in a given bandwidth region (neighborhood) can take on a variety of shapes depending on how heavily you wish to weigh the extreme points in the neighborhood. Consider a kernel density estimate of the velocities with a fixed bandwidth but four different kernels: the classic Gaussian and Epanechnikov kernels as well as simple rectangular and triangular kernels. See Section 8.2.1 of the textbook, including Figure 8.1, for details on these kernels.

Code set-up

For clean code, we will store output from the kernel density estimate function `density` and then, using the `lines` function, overlay the kernel density estimate on the velocities histogram. For consistency across each kernel density smooth, use the biased cross-validation bandwidth `bcv`.

For example, the following code chunk uses the biased cross-validation bandwidth and biweight kernel to overlay a kernel density estimate on the histogram of velocities with a data rug underneath for reference. (We are not considering the biweight for this problem, the code is purely for illustration!) Notice that we need a histogram with density values on the y-axis to appropriately overlay the kernel density smooth. To do this, specify `freq=FALSE` in the `hist` function. Also notice that we set the range of the y-axis to ensure the density smooth does not get cut off at the top of the graphic.

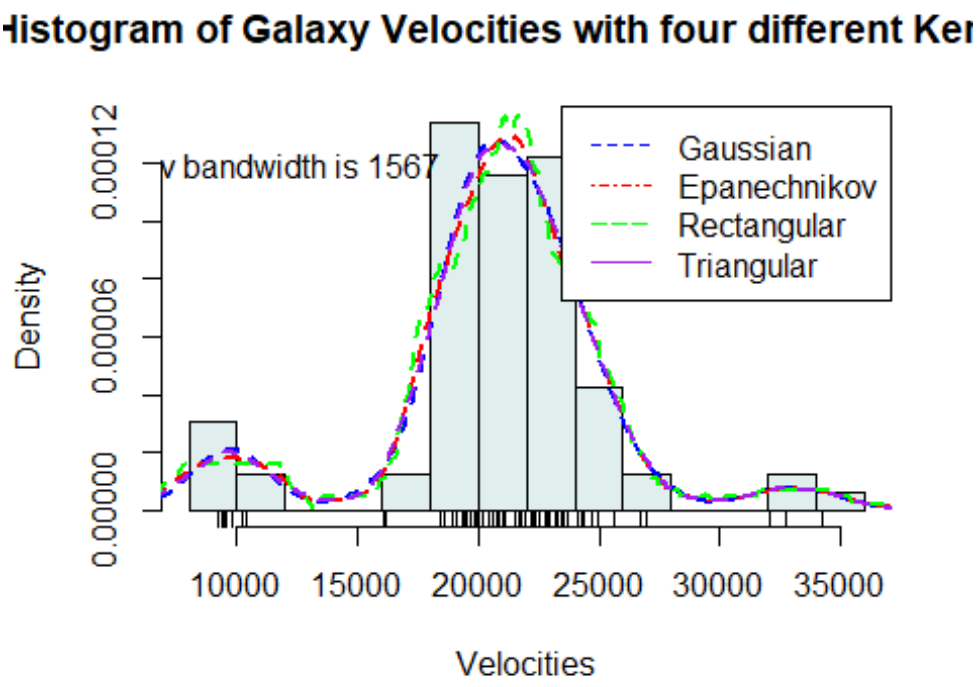
```
kde_biweight = density(galaxies, bw="bcv", kernel="biweight")
kde_maxy = max(kde_biweight$y)
# freq=FALSE forces the y-axis to be density values
# ylim options allows specification of the y-axis range to plot the kde
hist(galaxies, freq=FALSE, main="", xlab="Velocities",
     ylim=c(0, kde_maxy))
lines(kde_biweight, col="brown", lwd=2)
text(12000, 0.00012, paste("bcv bandwidth is",
                           signif(kde_biweight$bw,4)), col="brown")
rug(jitter(galaxies)) # data rug below the histogram
```




```
#paste("The bcv bandwidth is ", signif(kde_biweight$bw,6))

kde_gaussian = density(galaxies, bw="bcv", kernel = "gaussian")
kde_epan = density(galaxies, bw="bcv", kernel = "epanechnikov")
kde_rect = density(galaxies, bw="bcv", kernel = "rectangular")
kde_tri = density(galaxies, bw="bcv", kernel = "triangular")
hist(galaxies, freq=FALSE, main = "Histogram of Galaxy Velocities with four
different Kernels", xlab= "Velocities", breaks = 9, col = "azure2")
lines(kde_gaussian, col="blue", lwd=2, lty=2)
lines(kde_epan, col="red", lwd=2, lty=2)
lines(kde_rect, col="green", lwd=2, lty=2)
lines(kde_tri, col="purple", lwd=2, lty=2)
text(12000, 0.00012, paste('bcv bandwidth is', signif(kde_gaussian$bw,4)))

rug(jitter(galaxies))
legend("topright", legend = c("Gaussian", "Epanechnikov", "Rectangular",
"Triangular"), col= c("blue", "red", "green", "purple"), lty= c(2, 4, 5 ,1))
```



The problem

- Present, on the same set of axes, the histogram of velocities and four kernel density estimates (so four kernels: Gaussian, Epanechnikov, rectangular, and triangular). Label the axes, identify the bandwidth used, use a different line type and color for each density estimate, and include a legend on the plot.
- Use the biased cross-validation (`bcv`) bandwidth for all kernel density smooths in the task.
- Include a data rug on your histogram as in the code-set up illustration.

Report the following:

- View the help screen for `density`. Present the options available for the kernel and for the bandwidth selection. For bandwidth options, click on the `bw.nrd` link under "See Also" before the Examples. ?density
- Your output should include one graphic: comparison of kernel density estimates across four kernels, including a histogram, data rug, and legend.
- Evaluate the density smooths, commenting on the affect of each kernel on the density estimate.
- Which kernel would you choose?

Each kernel are very similar on the histogram, with rectangular having the highest peak of the four. The rectangular kernel is very uneven at several points while the other kernels are

very similar if not the same besides the peaks. If I had to choose I would go with the Epanechnikov kernel.

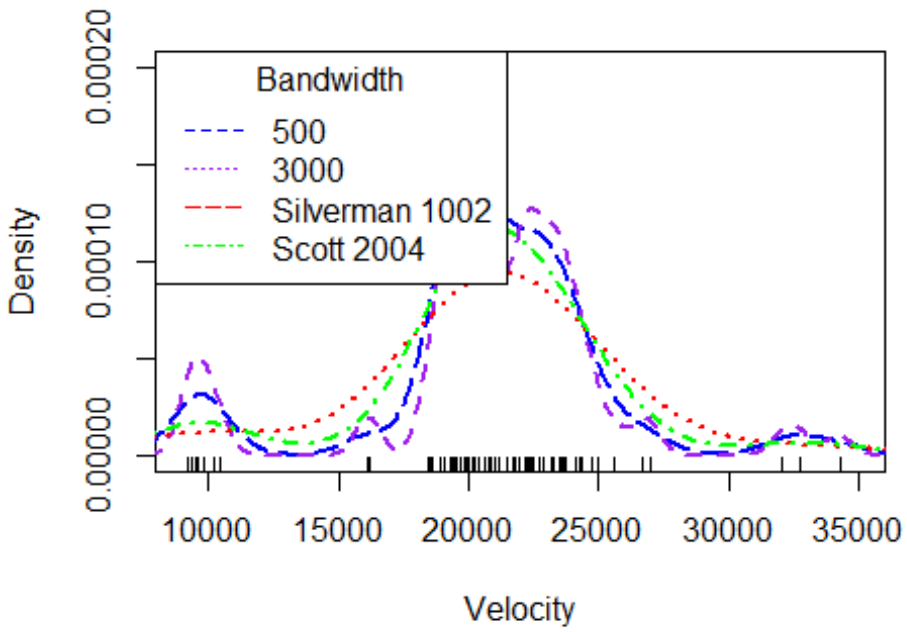
Task 3: Kernel density estimates with different bandwidths

Fit a kernel density estimate to velocities data with bandwidths 500 and 3000 as well as the plug-in bandwidths of Scott (1992) and Silverman (1986) being respectively $1.06sn^{-1/5}$ and $0.9\min(s,\ IQR/1.34)n^{-1/5}$ with sample standard deviation s.

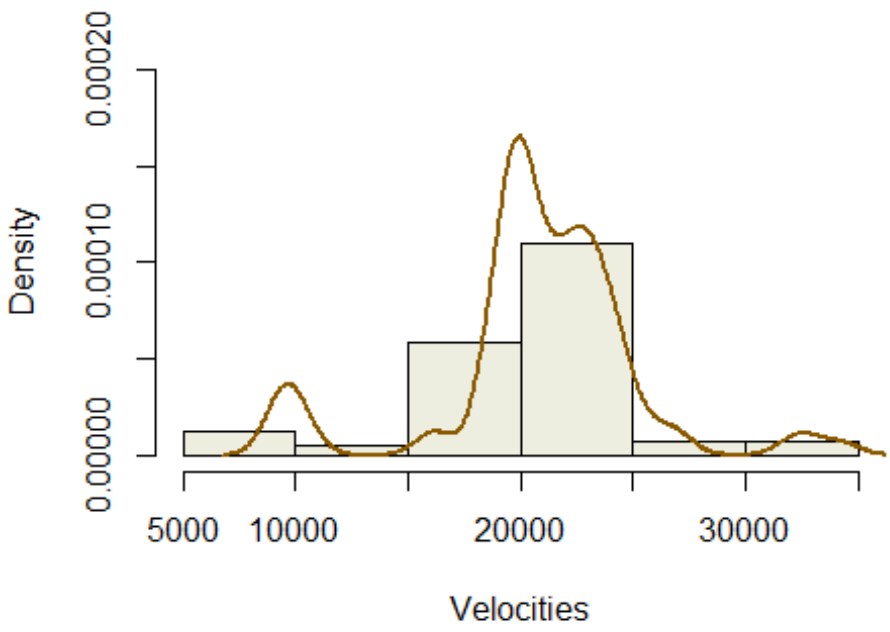
Code set-up

Like in Tasks 1 and 2, store the results from the kernel density estimate of the R `density` function and then use plot and lines to present graphics of the estimates. The bandwidth option is `bw`. For example, for the Silverman (1986) density estimate, the following code chunk will present the histogram, overlay the density estimate, and then present a data rug underneath for reference. The legend is used to identify the bandwidth.

```
# Silverman (1986) bandwidth
sman_h = 0.90*min(c(IQR(galaxies)/1.34, sd(galaxies)))*n^(-1/5)
scott = 1.06*sd(galaxies)*n^(-1/5)
# density estimate using Silverman bandwidth, store for plotting
kdesman = density(galaxies, bw = sman_h) # default Gaussian kernel
kdefive = density(galaxies, bw = 500)
kdethree = density(galaxies, bw = 3000)
kdescott = density(galaxies, bw = scott)
kde_maxy = max(kdefive$y)
# histogram overlay by density plot
plot(NULL, xlim=c(9000,35000), ylim=c(0,.00020), ylab="Density",
xlab="Velocity")
lines(kdesman, col="blue", lwd = 2, lty = 5)
lines(kdefive, col="purple", lwd = 2, lty = 2)
lines(kdethree, col ="red", lwd = 2, lty = 3)
lines(kdescott, col="green", lwd=2, lty=4)
rug(jitter(galaxies))
legend("topleft", legend=c(paste("500"), paste("3000"),
paste("Silverman",signif(kdesman$bw,4)), paste("Scott",
signif(kdescott$bw,4))), lty=c(2,3,5,4), col = c("blue", "purple", "red",
"green"), title= "Bandwidth")
```




```
optbw =dpik(galaxies)
kdeopt = density(galaxies, bw=optbw)
hist(galaxies, freq=FALSE, main="", xlab= "Velocities", ylim=c(0,kde_maxy),
col="ivory2")
lines(kdeopt, col="orange4", lwd=2)
```



The problem

- Present, on the same set of axes, the four kernel density estimates (bandwidths 500, 3000, Silverman, and Scott). Present only the kernel density estimates, not the histogram for clarity. Label the axes, use a different line type and color for each estimate, and include a legend on the plot.

- On a separate plot, use the function `dpik` from the KernSmooth package to choose an "optimal" plug-in bandwidth for kernel density estimation of the data. Present the plot of this kernel density estimate along with a histogram, a data rug, and specification of the bandwidth value.

Report the following:

- Your output should include two graphics: comparison of kernel density estimates across four bandwidths; kernel density estimate with the optimal bandwidth and including a histogram and data rug.

- Evaluate the density smooths, commenting on the affect of each bandwidth on the density estimate.

- Which bandwidth would you choose?

- Describe the density smooth using your chosen bandwidth: skew/symmetry, number of modes, modal velocities, high probability region(s), extrema.

From the kernel density estimates, the 500 and 3000 estimates were not smooth at all compared to Silverman and Scott. The bandwidth that I would choose is Scott since the symmetry and smoothness seems the most accurate and better than Silverman.