Chapter 8

Enhanced plots

8.1 *Lights of New York* (1928) explored

In connection with the second plot for *Intolerance*, in Figure 7.2 of Section 7.2, discussion of its construction was deferred to here. Rather than plunging into the code, whose one page may look forbidding, the ideas involved will be introduced incrementally via a series of analyses for *Lights of New York*. This film is chosen simply because it was one of the first films I looked at when thinking about how shot-type information might be included in analyses of SLs (Baxter, 2012a).

It should be noted that if analysing data using the Cinemetrics software with data submitted in advanced mode it is possible to both colour-code the shots by type and overlay trend lines or moving averages where the degree of smoothing can be controlled. This is very much in the spirit of what is discussed in this chapter, though other methods of display, not readily available in Cinemetrics, are explored.

8.1.1 Colour-coding shots

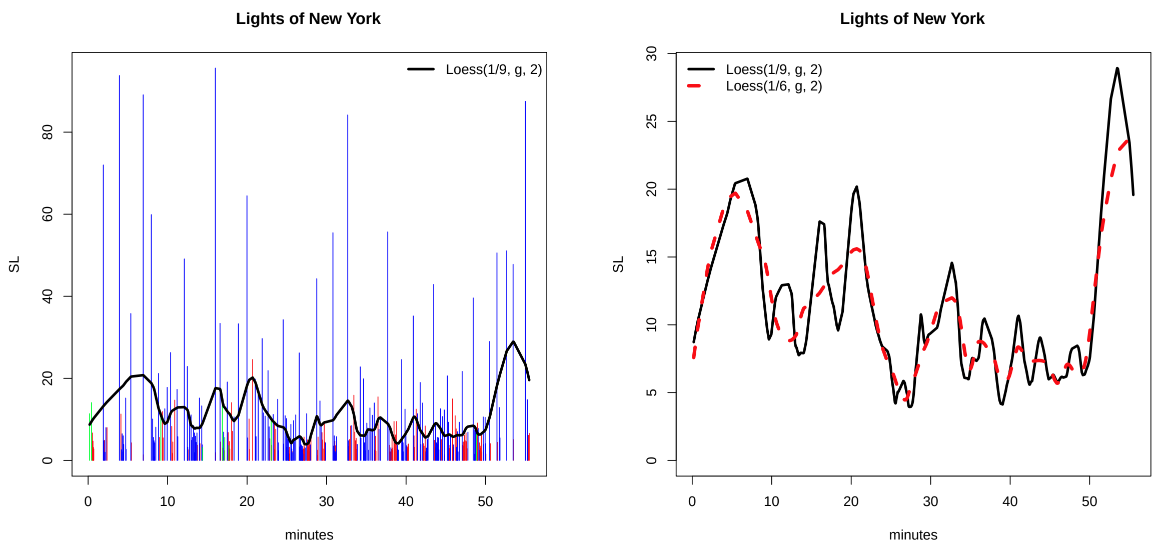


Figure 8.1: *Loess smooths for* Lights of New York *overlaid on colour-coded SL data to the left (red = 'action', blue = 'dialogue', green = 'title'), and compared, without enhancement, to the left.*

To the left of Figure 8.1 a plot of SLs against cut-point is shown; to the right two loess smooths are compared. I quite like the smoother (red, dashed) curve in the right panel which shows a nice reasonably rhythmic alternation of peaks and troughs over the first 35 minutes or so. Smoothing less shows more variation that the plot to the left (and others) suggests is 'real' so that this degree of smoothing is used for other plots unless otherwise indicated.

Thes plots are similar to ones already seen, except that shots are colour-coded according to whether they are action (red), dialogue (blue) or title (green) shots. The submission of the film to the Cinemetrics database, by Charles O'Brien, is in advanced mode with the 'Type' variable categorising shots as 'action', 'dialog' and 'exp.tit'. The data were imported to Excel, where column headers were edited as described in Section 3.2.2, then copied and imported into `R` with the name `Lights_of_New_York` (Section 3.2.1).

The `R` code for this and subsequent analyses can be built in a modular fashion. Initially the structures needed for plotting can be set up without actually doing any plotting. This is best done by creating a function that can be modified and added to as the need arises (Section 4.3). So begin with

```
LONY.plot <- function() {
z <- Lights_of_New_York
SL <- z$SL/10                   # SL in seconds
type <- z$Type                  # Shot type
minutes <- cumsum(SL)/60        # Cut-point in minutes
n <- length(SL)                 # Number of shots
order = c(1:n)                  # Shot sequence
SLrank <- rank(SL)              # SL rank
#
# Extract and name SLs for shots of different types
#
SLA <- SL[type == "action"]
SLD <- SL[type == "dialog"]
SLT <- SL[type == "exp.tit"]
#
# Do the same for cut-points
#
minA <- minutes[type == "action"]
minD <- minutes[type == "dialog"]
minT <- minutes[type == "exp.tit"]
#
# Loess smooths used for plots
#
LONYfit1 <- loess(SL ~ minutes, span = 1/9, family = "g", degree = 2)$fitted
LONYfit2 <- loess(SL ~ minutes, span = 1/6, family = "g", degree = 2)$fitted
}
```

This is intended to be reasonably self-explanatory. It won't at this stage produce anything visible, unless there are obvious errors, which may show if you run it using `LONY.plot()`. The next, slightly tricky, bit of code, if added at the end of the function (before the final }), should produce the left-hand panel of Figure 8.1.

```
plot(minutes, LONYfit1, type = "n", lwd = 2, col = "black", xlab = "minutes", ylab = "SL",
main = "Lights of New York", ylim = c(0, max(SL)))

for(i in 1:n) lines(c(minA[i],minA[i]), c(0, SLA[i]), col = "red")
for(i in 1:n) lines(c(minD[i],minD[i]), c(0, SLD[i]), col = "blue")
for(i in 1:n) lines(c(minT[i],minT[i]), c(0, SLT[i]), col = "green")

lines(minutes, LONYfit1, type = "l", lwd = 3, col = "black")
legend("topright", "Loess(1/9, g, 2)", lty = 1, lwd = 3, col = "black", bty = "n")
```

In the `plot` command the `type = "n"` argument produces an empty plot, with labelling, that is a framework for the additions to follow which are, in order, vertical lines of different colours corresponding to the shot types plotted at cut-points, and a loess smooth. The `lty` and `lwd` arguments control line type and width, and `col` selects the line colour. An initial blank plot is needed since any attempt to plot at this stage will be over-plotted by what follows. The `ylim` argument is needed to ensure proper limits for plotting the SLs.

The bit where care is needed is in the `for` commands, where what is happening is that the beginning and end points of each line corresponding to an SL, and the appropriate colour-coding, is being specified. Once added, get the plot using `LONY.plot()`.

The right-hand panel of the figure is obtained by adding the following to the function.

```
plot(minutes, LONYfit1, type = "l", lwd = 3, col = "black", xlab = "minutes", ylab = "SL",
main = "Lights of New York", ylim = c(0, max(LONYfit1)))

lines(minutes, LONYfit2, type = "l", lty = 2, lwd = 4, col = "red")

legend("topleft", c("Loess(1/9, g, 2)", "Loess(1/6, g, 2)"), lty = c(1,2), lwd = c(3,4),
col = c("black", "red"), bty = "n")
```

This kind of thing has already been illustrated in Section 7.5. As described so far, running `LONY.plot()` will only show the plot produced by the final block of code added. To show all plots insert `win.graph()` at the start of each block. Typing `graphs.off()` when the `R` prompt `>` is showing will clear the screen of graphical clutter; it can usefully be inserted at the start of the function, if it is to be run repeatedly.

8.1.2 Adding 'wallpaper'

Once the idea of adding colour-coding, and the means of doing it, is grasped, experimentation with a variety of different effects is possible. Adding 'wallpaper', as in Figure 8.2 is one possibility.

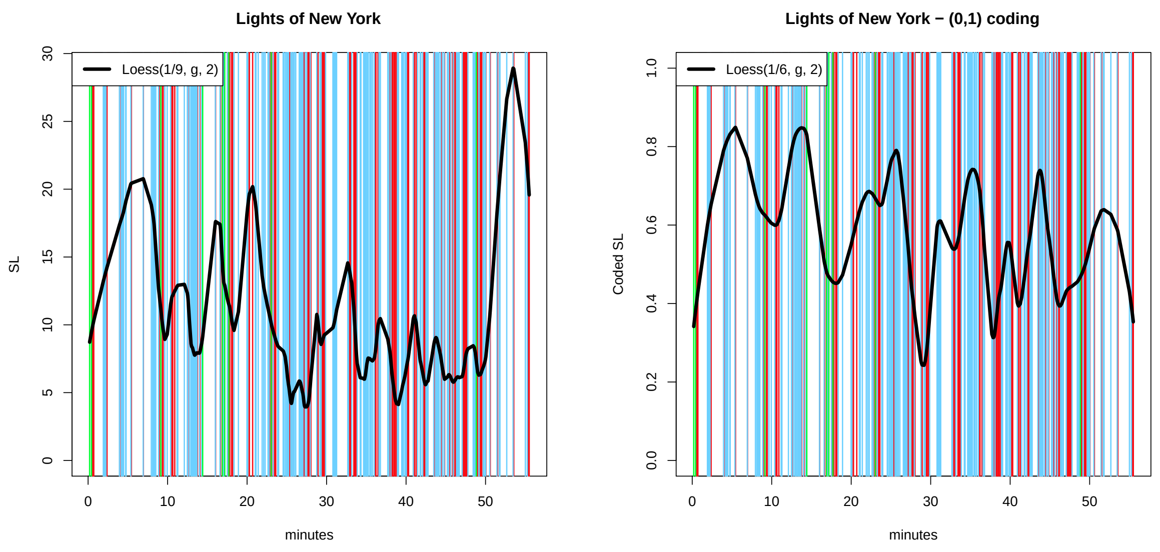


Figure 8.2: *Loess (1/9, g, 2) smooths for* Lights of New York *with colour-coded 'wallpaper' (red = 'action', blue = 'dialogue', green = 'titles'). To the left untransformed SL data is used; to the right action shots have been coded 0 and other shots 1.*

The code

```
plot(minutes, LONYfit1, type = "n", xlab = "minutes", ylab = "SL", main = "Lights of New York",
ylim = c(0, max(LONYfit1)))
for(i in 1:length(minA)) abline(v = minA[i], lwd = 1.5, lty = 1, col = "red")
for(i in 1:length(minD)) abline(v = minD[i], lwd = 1.5, lty = 1, col = "skyblue")
for(i in 1:length(minT)) abline(v = minT[i], lwd = 1.5, lty = 1, col = "green")
```

```
lines(minutes, LONYfit1, lwd = 4)
legend("topleft", "Loess(1/9, g, 2)", lty = 1, lwd = 4, col = "black", bty = "o", bg = "white")
```

produces the left-hand panel of the figure. This is actually a bit simpler than adding the colour-coding in Figure 8.1, since the vertical lines associated with the SLs don't need to be bounded in any way.

The right-hand plot introduces what I suggested in Baxter (2012b) might be a novel idea; it has not, in any case, previously been discussed in these notes. If the type categories admit dichotomization in any way then the dichotomy can be represented by a 0-1 coding that can then be treated like any other numerical variable. In the code that follows `SLtype` codes action shots as 0, and dialogue and titles as 1. There are relatively few titles so this will not affect the analysis much, even if one has doubts about amalgamating titles with dialogue. The loess fit `LONYfit3` is just the smoothing algorithm applied to these data with `span = 1/6` chosen after a little experimentation. Other than that, what's done is identical to the previous block of code and the wallpaper is the same[1].

```
SLtype <- ifelse(type == "action", 0, 1)
LONYfit3 <- loess(SLtype ~ minutes, span = 1/6, family = "g", degree = 2)$fitted
plot(minutes, LONYfit3, type = "n", xlab = "minutes", ylab = "Coded SL",
main = "Lights of New York - (0,1) coding", ylim = c(0,1))

for(i in 1:length(minA)) abline(v = minA[i], lwd = 1.5, lty = 1, col = "red")
for(i in 1:length(minD)) abline(v = minD[i], lwd = 1.5, lty = 1, col = "skyblue")
for(i in 1:length(minT)) abline(v = minT[i], lwd = 1.5, lty = 1, col = "green")

lines(minutes, LONYfit3, lwd = 4)
legend("topleft", "Loess(1/6, g, 2)", lty = 1, lwd = 4, col = "black", bty = "o", bg = "white")
```

Some care is needed in interpreting these plots. The wallpaper codes for shot type; action/dialogue, if the relatively few titles are ignored. The loess curve in the left-hand panel is based on SLs so there is not a necessary relationship between any pattern it shows and that of the background, though one is expected *if* SLs and shot-type are associated. For *Lights of New York* (and other films) there is an expectation that 'action' shots will tend to be shorter than 'dialogue' shots, confirmed by the left-hand panel of Figure 8.1, and this is reflected in a reasonable match between the patterns exhibited in the two displays embodied in the left-hand panel.

I'm not sure if mountain goats really do jump from peak-to-peak, but you can imagine one doing so in the right-hand panel, from the left, moving slowly but steadily down to lower altitudes. Here, because of the coding, the loess smooth directly models the variation between action and dialogue. The peaks correspond, as it were, to bursts of inactivity, and things get more exciting as the goat progresses down the range. This is to be hoped for, since the two representations of pattern are designed to show the same thing and should be concordant. This they reasonably well are. The level of smoothing has been chosen, for the convenience of the goat, to be not too rocky – goats are only interested in the heights.

8.1.3 Order structure matrices revisited

Order structure matrices were discussed in Section 7.6.4. In Section 7.8.3 the opinion was expressed that these were unecessarily complicated and might be improved upon. The gist of the argument was that the plots had redundant aspects that generated an appearance unpleasing to some eyes, that might be modified to more effectively diplay what was intended of them. As a reminder, an order structure matrix plot for *Lights of New York* is shown to the left of Figure 8.3.

As expounded by Redfern (Redfern, 2012c; and elsewhere[2]) changes in cutting style are read from left to right. This means that the horizontal axis, which repeats the same information with

[1]Or should be – they look identical on my laptop but copying them into the typesetting package used seems to introduce minor differences.

[2]http://nickredfern.wordpress.com/2011/06/16/time-series-analysis-of-top-hat-1935/

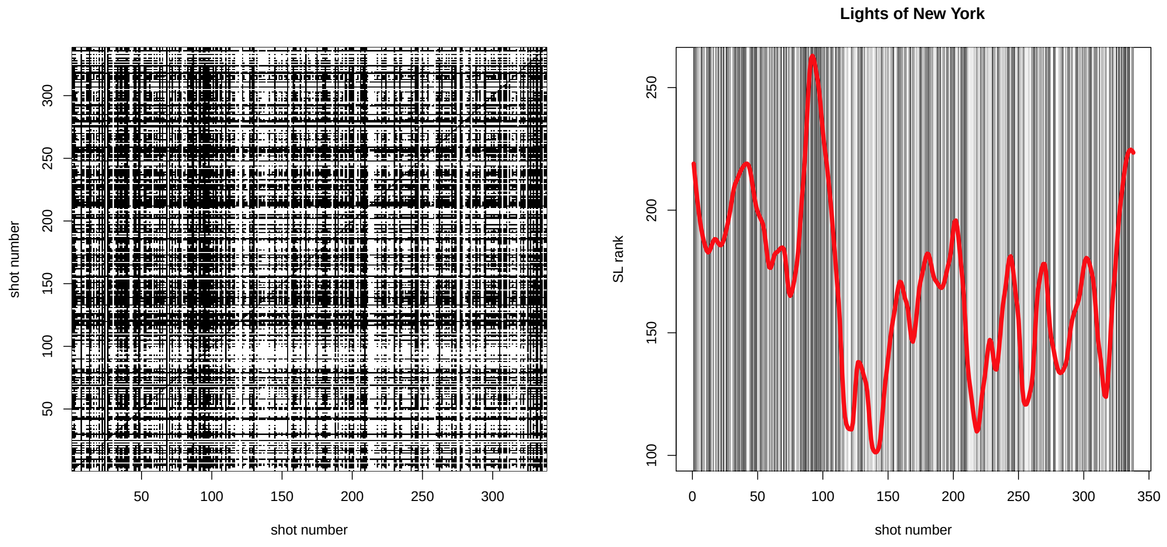


Figure 8.3: *An order structure matrix plot* Lights of New York *to the left, and an alternative representation to the right with a superimposed loess (1/9, g, 2) smooth based on ranked data.*

a colour reversal, is redundant and (in my view) complicates the picture. As the vertical lines that are being 'read' can be viewed as on a 'pseudo-gray' scale, the use of a proper gray-scale, 'discarding' the horizontal axis, seems simpler. This is illustrated in the right-hand panel of Figure 8.3.

The plot tells exactly the same story as the order structure matrix plot but, to my eye, is easier to read and nicer to look at. It is easier to explain (shots are colour-coded according to the rank of their SLs) and it is easier (I think) to overlay plots, such as the loess smooth shown, that can aid interpretation[3].

Code for getting the order structure matrix plot was given in Section 7.6.4, but is repeated below for the sake of completeness.

```
mat <- matrix(0, n, n)
for(i in 1:n) {
for(j in 1:n) {
mat[i,j] <- ifelse(SL[i] >= SL[j], 1, 0)
}
}
image(1:n, 1:n, mat, col = c("white", "black"), xlab = "shot number", ylab = "shot number")
```

The code for the right-hand panel of Figure 8.3 is as follows.

```
LONYfitrank <- loess(SLrank ~ order, span = 1/9, family = Family, degree = Degree)$fitted
test.colour <- gray((n - order)/n)

plot(order, LONYfitrank, type = "n", xlab = "shot number", ylab = "SL rank",
main = "Lights of New York", ylim = c(100,260))

for(i in 1:n) abline(v = i, col = test.colour[SLrank[i]], lwd = 1)
lines(order, LONYfitrank, type = "l", lwd = 5, col = "red")
```

Here, `test.colour` sets up the gray-scale colour palette to be used. It is defined in the way it is to mimic the same colouring as in the order structure matrix plot. The 'reverse' of the black/white

[3]To be clear about this, I have no problem with the *idea* behind using an order structure matrix plot, but they are cumbersome and the idea can be more easily implemented as shown. Adding loess smooths to the modified plot is a useful aid to selecting the degree of smoothing that captures the main patterns of variation in the data.

coding used is obtained with the simpler `test.colour <- gray((order)/n)`. The scale has to have limits of 0 and 1. Other palettes are available and discussed a little further in Section 8.3. Some of these are a little simpler to define but, in some cases, less useful.

Where loess smooths have been shown in the above figures the general message is the same, but the precise picture depends, as discussed in the previous chapter, on both the degree of smoothing and methods of data transformation adopted. Illustrated in the figures are smooths of the untransformed SLs, ranked SLs and 0-1 coded SLs. The last of these attempts something different from the other two, but should produce results with a similar interpretation if SLs are strongly related to shot-type. It is, I think, possibly a matter of taste and choice of emphasis as to which kind of plot is to be preferred when confining attention to the analysis of a single film.

8.2 *Intolerance* revisited

It would be remiss not to include the code for the right-hand panel of Figure 7.2 promised in Section 7.2. Here it is.

```
Intolerance.plot <- function() {
#
z <- Intolerance
SL <- z$SL/10
minutes <- cumsum(SL)/60 # cut-point in minutes
ASL <- mean(SL)
Type <- z$Type
n <- length(SL)
#
# extract SLs and cut-points for individual stories
#
B <- SL[Type == "Babylon"]
F <- SL[Type == "French"]
J <- SL[Type == "Judean"]
M <- SL[Type == "Modern"]
Babylon <- minutes[Type == "Babylon"]
French  <- minutes[Type == "French"]
Judean  <- minutes[Type == "Judean"]
Modern  <- minutes[Type == "Modern"]
#
fit1 <- lm(SL ~ poly(minutes, 1)) # straight-line fit
fit5 <- lm(SL ~ poly(minutes, 5)) # 5th order polynomial fit
Ylim <- c(9,0)                    # Limits for y-axis, reversed as in Tsivian (2005)
pred <- 1:max(minutes)            # plotting positions (x-axis) for fits
#
plot(pred, predict(fit5, data.frame(minutes = pred)), type = "n", ylim = Ylim,
xlab = "minutes", ylab = "SL", main = "Intolerance")

for(i in 1:length(Babylon)) abline(v = Babylon[i], lwd = 1.5, lty = 1, col = "green")
for(i in 1:length(French)) abline(v = French[i], lwd = 1.5, lty = 1, col = "skyblue")
for(i in 1:length(Judean)) abline(v = Judean[i], lwd = 1.5, lty = 1, col = "red")
for(i in 1:length(Modern)) abline(v = Modern[i], lwd = 1.5, lty = 1, col = "orange")
lines(pred, predict(fit5, data.frame(minutes = pred)), lwd = 3)
lines(pred, predict(fit1, data.frame(minutes = pred)), lty = 2, lwd = 3)
abline(h = ASL, lty = 3, lwd = 4) # adds line at SL
#
legend("topleft", legend = c("Babylonian", "French", "Judean", "Modern"), lty = c(1,1,1,1),
lwd = c(2,2,2,2), col = c("green", "skyblue", "red", "orange"), title = "Story", bg = "white")

legend("topright", legend = c("ASL", "Linear trend", "5th order polynomial"), lty = c(3,2,1),
lwd = c(4,3,3), bg = "white")
}
```

It is built up in the same way as the code for *Lights of New York*. There are seven categories in the ‘Type’ variable, the majority of which identify which of the four stories that comprise *Intolerance* a shot is located in. Of the 1989 shots 56 are in three different ‘other’ categories, and were not used explicitly in constructing the figure. It is assumed that columns have headers labelled as discussed in Section 3.2.2 and that the file is imported into `R` with the name `Intolerance`.

8.3 Jeu d'esprit

In constructing plots like those in Figure 8.2 it is possible to be distracted by the fact that, stripped of labelling and the loess smooths, the picture looks something like a production of 'modern' art. The temptation to seek out matching Bridget Riley's (it can be done reasonably well) or design your own wallpaper or Bridget Riley needs to be resisted. The purpose of constructing figures like those illustrated is, after all, to convey a message, and I'm not sure how much modern art aspires to this.

Similar temptations, and pitfalls, exist in constructing graphics like that in the right-hand panel of Figure 8.3. The free choice of colour available in Figure 7.2 and 8.2 (where a discrete and unorderd set of categories is represented) is constrained by the need to have a gradation of colours that reflects the coding (if SLs are ranked) from shortest to longest shot. A gray-scale seems sensible, but you get bored.

What follows are disjointed ideas that occurred at times when my concentration on the business to hand wavered a little. `R` code for everything that follows is discussed in Section 8.3.3.

8.3.1 *Halloween* (1928) – matching colour to content

In looking at Redfern's (2012c) analyses of slasher films, using order structure matrices, and seeing how the alternative representation explored in Figure 8.3 worked, I wondered (idly) whether a colour scheme more suited to the films' content might be devised. It turns out it can, and with surprising ease, as the left panel of Figure 8.4, for *Halloween* (1978), shows[4]. Compared to the earlier analysis of *Lights of New York* the only change is to use the `heat.colors` rather than `gray` palette, which produces, without any tweaking, the satisfactory splatter of bright red towards the end.

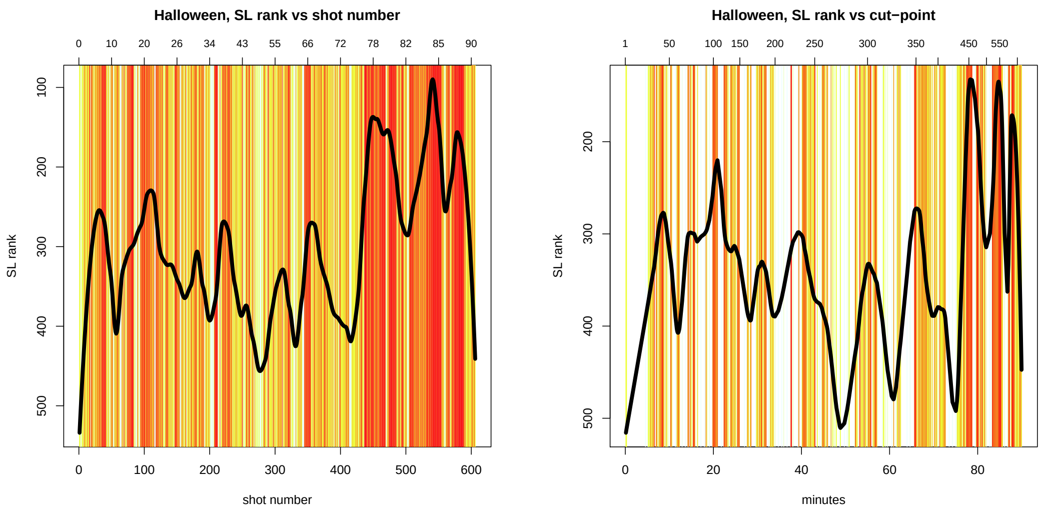


Figure 8.4: *Heat colour representations of SL data for* Halloween*. The same number of shots, with the same colouring are used, but plotted against shot number to the left and cut-point to the right. The scales on the upper axes are minutes and shot number respectively. Loess (1/9. g, 2) smooths of the ranked SLs are shown.*

[4]See Redfern's research blog, http://nickredfern.wordpress.com/tag/horror-films/, for links to the data.

A loess (1/9, g, 2) smooth to the *ranked* SLs is shown. This can obviously be modified if wished, but corresponds well enough with the wallpaper. The y-axis has been reversed so that peaks correspond to concentrations of shots with short SLs.

In Redfern (2010c) the visual representation is 'validated' by a reading of the film based on a close knowledge of it, overlaid with an interpretation of the way different passages in the film function. I suspect most interpretations of these and other forms of graphical representation are undertaken in a similar way; more interestingly, for present purposes, are attempts to abstract from individual patterning and recognise common structure across a range of films – of similar genre, or whatever comparison seems interesting. Redfern, for example, concludes on the basis of analysis of four slasher films that 'the genre is characterised by a common editing pattern to such an extent that the films that followed *Halloween* may be described as stylistically formulaic'.

My usual approach when meeting a new methodology, or one new to me, is to prod it a little to see how it reacts, and how claims for it stand up[5]. In trying to compare *Halloween* with other films some problems, or at least variants of the way it might be displayed, became apparent.

One issue is that the visual impact of a graphic can depend on the number of shots a film has. Films with a relatively small number of shots can look 'washed-out' by comparison with films with a significantly larger number when viewed on a large screen[6]. Ways of getting round this are discussed shortly.

A second issue is revealed by looking at the scale at the top of the graph; this is in minutes and not linearly related to the lower scale of shot order. Thus, the dominant impression, the splodge of red at the right, is produced by over a quarter of the shots in the film, but only accounts for about a sixth of the running time. There is one shot in *Halloween* of over four minutes, occuring early, and five others of over one minute between shots 155 and 342. It is possible to pick out disjoint sets of 20 shots or more in the final sixth of the film (in terms of time) that occupy less than a minute. In the visual representation of *Halloween* in Figure 8.4 these blocks of shots receive over 20 times the weight of individual long shots that occupy a similar or longer period of time.

That is to say, the left-hand plot in the figure, and the related order structure matrix plot, privileges short shots. This is doubtless the intent; the fact that there is a concentrated flurry of short shots towards the end is emphasised.

The order structure matrix approach is promoted as a robust method; it achieves robustness, in the context of the analysis of SLs, by ignoring all the information contained in the measurements except that one shot is longer or shorter than another and occurs earlier or later. In graphical terms, and in effect, this 'robustness' manifests itself in two ways. The first is by using graded colour-coding of shots that reflects their rank ordering only; the second is by plotting the lines that represent a shot against their equi-spaced order in the shot sequence, ignoring their length and hence the time of cut-points between the shots.

With the simpler form of plotting advocated here the constraint of plotting against shot order is removed. It is straightforward to retain the robustness inherent in the colour-coding, while reintroducing other SL information by plotting against cut-points. The right panel of Figure 8.4 is the result. If the left-hand panel privileges the representation of the shorter shots, associated with what Redfern terms 'the frenzied violence of body horror', then the right-hand panel privileges longer shots 'creating a pervading sense of foreboding'[7].

I will leave it to the reader to decide if these ways of displaying the data have any merit and, if they think they do, which of the the two ways of displaying the data is to be preferred, if any and if a choice has to be made. What I'm doing here is evading any discussion of any 'meaning' the graphs might have beyond the statistical display of pattern in the data in two different ways. This is because I suspect it might be a difficult question I can't answer. The next section concentrates

[5]My original, unworthy, thought was to see if I could find films like *Bambi*, or others with seemingly innocuous titles, that, judged 'cold', had editing stuctures indistinguishable from that of *Halloween*. The idea was abandoned, temporarily anyway, at an early stage.

[6]This is obvious with screen output from `R` if the screen is large enough. The effect is hidden a bit on reducing graphs to the size needed for a normal-sized page - because of the compression involved - so is not attempted here.

[7]In the context of other genres these might be what I think I've seen referred to as contemplative passages, romantic interludes or, as the attention-deficit might have it, the boring bits.

on an interesting technical problem.

8.3.2 Controlling for shot numbers - an idea

This, touched on briefly above, is how you might go about comparing patterns in films with different numbers of shots, beyond the qualitative comparison of graphs of the kind illustrated in this and previous chapters. The question is of more than passing interest. Arguably this difficulty was one of the rocks on which the attempt by Cutting *et al.* (2011, 2012) to demonstrate the generality of four-act structure in films foundered. One of their basic ideas, converting actual SLs into an equal number of what they call 'adjusted' SLs for each film is sound. The manner in which this was achieved, however, introduced artefacts into the adjusted data that meant that the subsequent detection of four-act structure was a self-fulfilling prophecy.

What follows presents an idea – and it is only that – of how 'adjusted' SLs might be calculated, to then be put to whatever purpose one devises them for. The idea has a simple core, which is to fit a KDE to the data and then design the wallpaper using the KDE (which can be thought of as estimating SLs) rather than the SLs. It is the detail concealed by this idea that is important, so it will be taken in stages.

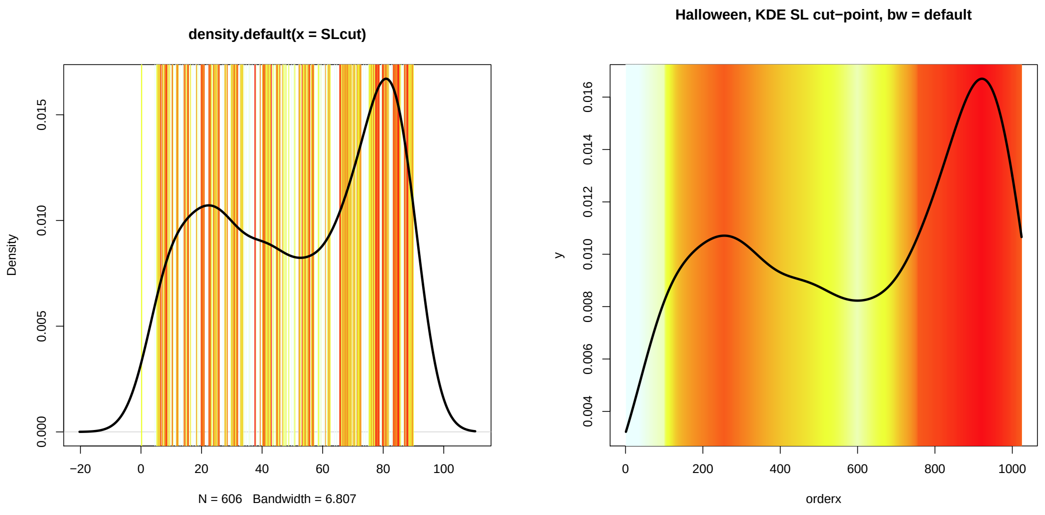


Figure 8.5: *Heat colour representations of SL data for* Halloween. *The KDEs are the same, with different scaling, in both plots. Wallpaper to the left is based on the SL ranks; wallpaper to the right is based on the KDE – see the text for details of the latter.*

The left-hand panel of Figure 8.5 shows the default KDE for the cut-points of *Halloween* overlaying wallpaper defined by the ranks of the SLs (see Section 7.7 for a discussion of this usage of KDEs). There are several problems with the plot. The KDE is over-smoothing and, as is its wont, strays well into areas beyond the limits of the film (negative time, and after you've stooped watching). This also has the effect of compressing the wallpaper a bit, since this is confined to the real duration of the film.

Leaving aside, momentarily, the over-smoothing issue, it is possible to do something about other aspects of its appearance. Left to its own devices the `density` command in `R` computes density estimates at 512 equally spaced points with limits defined by an internal algorithm. For the purposes of plotting, both the limits and the number of plotting points can be controlled, so in the right panel of Figure 8.5 the KDE is 'forced' to lie between the beginning and end of the film and 1024 plotting positions are used[8].

[8]For algorithmic reasons the number of points needs to be a power of 2, so $512 = 2^9$; the choice of $1024 = 2^{10}$ was a bit arbitrary, I don't think it makes much difference.

The KDE is just a magnified portion of that appearing to the left; the important difference is that the wallpaper is designed using the heights of the KDE at each of the 1024 plotting positions. Since the KDE smooths the data the wallpaper pattern, in contrast to that to the left where ranked SL colouring is used, is also smoothed and more 'solid' (a function, also, of the greater number of plotting positions). The index on the x-axis is the number of the plotting position; it could be converted to time but is left as it is. For later reference it means that the 512th plotting position corresponds to the middle of the film; the 256th position to the end of the first quarter, and so on.

All this, coupled with the over-smoothing, simplifies the picture of the film, too much so. Interpretation is sensible enough; a slow beginning; a touch of 'frenzied violence' to let you know you're in a slasher movie; a long lull in which to anticipate what you've come to see; which is the prolonged ouburst of frenzied violence at the end. This, of course, misses out on the subtleties of construction, where the calm is punctuated by bits of action to keep you on your toes. The obvious thing to do is reduce the smoothing by reducing the bandwidth from its default of 6.8. This is done in the left-hand panel of Figure 8.6.

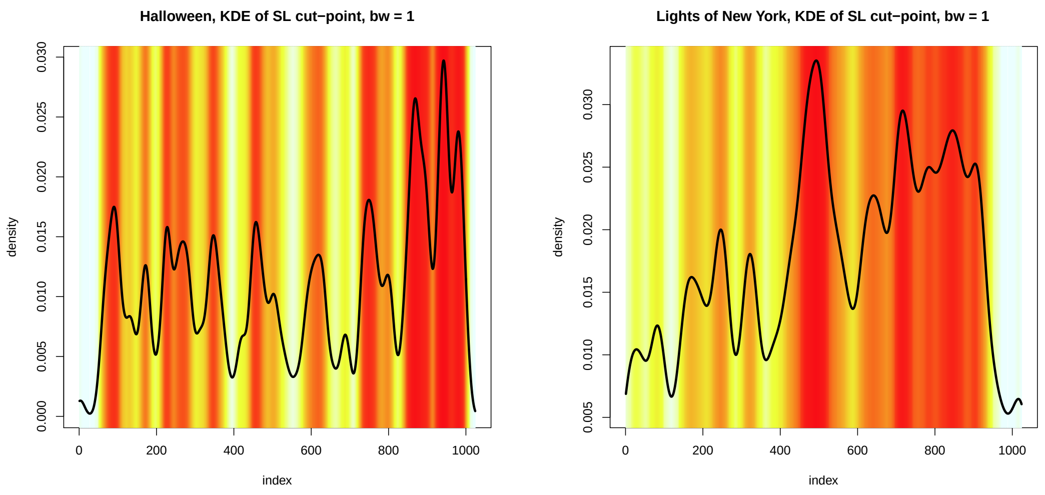


Figure 8.6: *Heat colour representations of SL data for* Halloween *and* Lights of New York *using a bandwidth of 1 minute in both cases.*

The selected bandwidth is 1 minute. You can get close to reproducing something like wallpaper based on the actual SLs by reducing this to 1 second (bearing in mind an estimate with, in this case, rather more points than SLs is being used) but this loses the advantages of simplification[9]. The picture is quite pleasing; before the leap into red violence at the end, and as these things go, there is a reasonably regular bounce both in terms of spacing and the heights/depths achieved ('as these things go' should be emphasised).

Rather than filling in Figure 8.6 with yet another graph for *Halloween* with a different bandwidth, the right-hand panel of Figure 8.6 is for *Lights of New York* with the same bandwidth of 1 minute. It was, in fact, initial problems encountered in effecting a comparison between the two films that motivated some of the above development. *Lights of New York* was made 50 years before the official advent of the slasher genre, so presumably can't be admitted to the slasher canon. Readers can supply their own commentary on what aspects of the graphics alert you to this.

[9]In conducting experiments on this what was at the forefront of my mind was what might be a suitable choice of bandwidth that could be used for comparing a body of films, so you need to lose detail if you want to compare broader underlying structure while maintaining a similar level of simplification. Or at least that's the current idea.

It is worth standing back a little to reflect on what has been done above. The story began with the stringently robust order structure matrix representation, replaced by an analogous but simpler plot of colour-coded ranked SLs plotted against shot order. This led on to the idea of plotting against cut-point, thereby introducing the lengths of shots back into the fold, and thence to the earlier idea of using KDEs to smooth the data. Loess smooths could be used, but KDEs have the advantage (in the `R` implementation) of allowing greater control over the number and location of plotting positions. Finally, the idea of colour-coding by SL ranks was abandoned, to be replaced by colour-coding of the *estimated* values of the KDEs at the plotting positions.

Two things have happened here. One is that the original idea of a robust representation of the data has been almost abandoned. The one vestige that remains is that KDEs might be regarded as robust in that they smooth out the effects of the more extreme SLs. That they are, in effect, estimates of SLs that downplay extremes incidentally avoids representational issues that would arise if actual SLs were colour-coded. The second thing that has happened is that the end result is two equivalent representations of the data, the KDE of the cut-points, and the KDE at each plotting point coded by colour. Either or both could be used, depending on the taste of the analyst and what they think best conveys the message they have to impart. These or any other graphical simplifications can be checked against the raw data (e.g., the left-hand panel of Figure 8.1).

It is then also legitimate to ask why bother, if one goes down the above route and opts for the KDE, with all the manipulation needed to get the colour. An answer to this is attempted in Section 8.3.4.

8.3.3 Code

In the code to follow labelling and style arguments, including the code needed for the upper axes in Figure 8.4, have been omitted.

The left-hand plot of Figure 8.4 is obtained from

```
SL <- film1
minutes <- cumsum(SL)/60
SLrank <- rank(SL)
n <- length(SL)
order <- c(1:n)
fitrank <- loess(SLrank ~ order, span = 1/9)$fitted
Ylim = c(max(fitrank), min(fitrank))
test.colour <- heat.colors(n)
plot(order, fitrank, type = "n", ylim = Ylim)
for(i in 1:n) abline(v = i, col = test.colour[SLrank[i]])
lines(order, fitrank)
```

The `test.colour <- heat.colors(n)` command selects the heat colour palette[10]. I used this because it was convenient for what I wanted to do. A little bit of Googling will bring up a lot of possibilities I haven't explored. For the right-hand plot, replace `order` with `minutes` in the `fitrank` definition and `plot` command; replace `v = i` in `abline` with `v = minutes[i]`.

For Figure 8.5 the left-hand plot is produced by

```
D <- density(minutes)
plot(D, type = "n")
for(i in 1:n) abline(v = minutes[i], col = test.colourx[SLrank[i]])
lines(D)
```

The right-hand plot needs a little more effort.

```
D <- density(minutes, from = 0, to = max(SLcut), bw = "nrd0", n = 1024)
x <- D$x
y <- D$y
nx <- length(x)
orderx <- c(1:nx)
test.colourx <- heat.colors(nx)
```

[10]See `?heat.colors` for this and other options

```
SLrank <- rank(-y)
plot(orderx, y, type = "n")
for(i in 1:nx) abline(v = i, col = test.colourx[SLrank[i]])
lines(orderx, y)
```

The density estimate, `D` specifies the bandwidth, via `bw` where `"nrd0"` is the default, and the number of plotting points via `n = 1024` (where 512 would be the default). The variables defined as `x` and `y` extract the plotting positions and the heights. The next four commands re-define variables needed for the plots, `SLrank` being defined as it is to ensure reds at the short end of the SL scale. For analyses with any other bandwidth simply substitute for `"nrd0"`, so `bw = 1` for the right-hand plot of Figure 8.6.

8.3.4 Coda

Before discussing how some of the development in Section 8.3.2 *might* be useful, some further aspects of the kind of plots introduced there are considered. Figure 8.7 compares plots of different kinds for *The House on Sorority Row* (1983) and *Pinocchio* (1940).

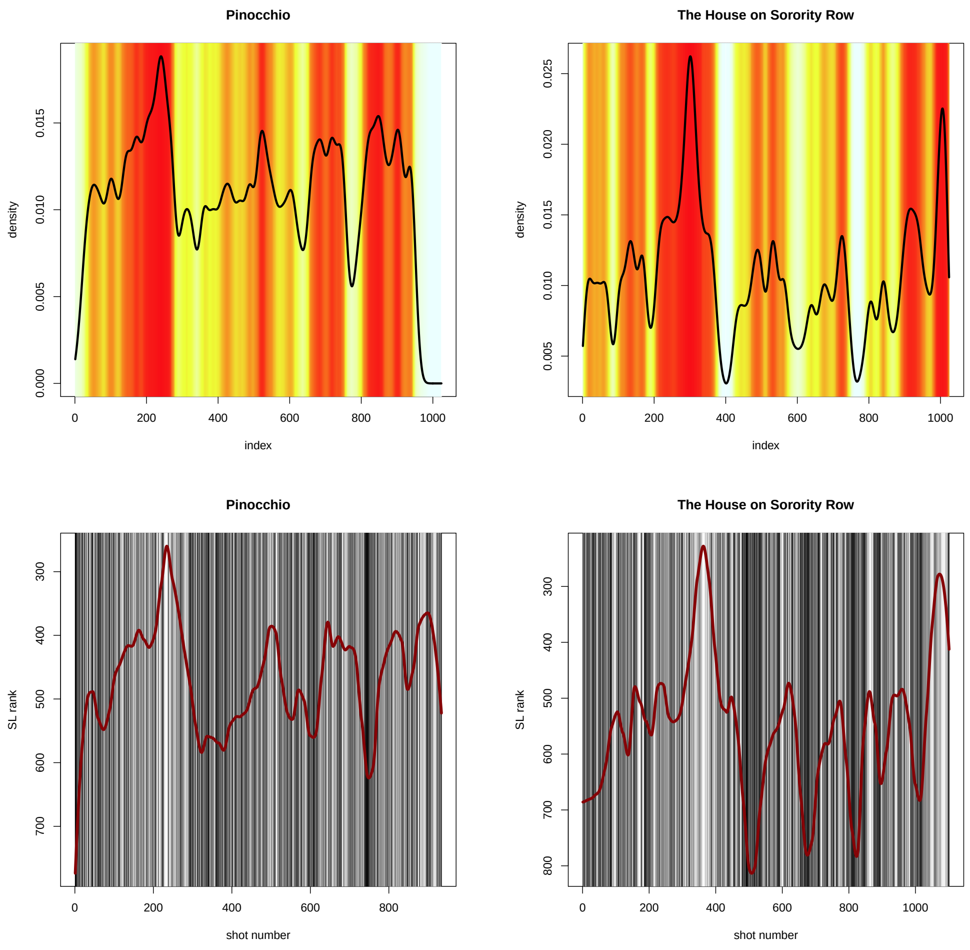
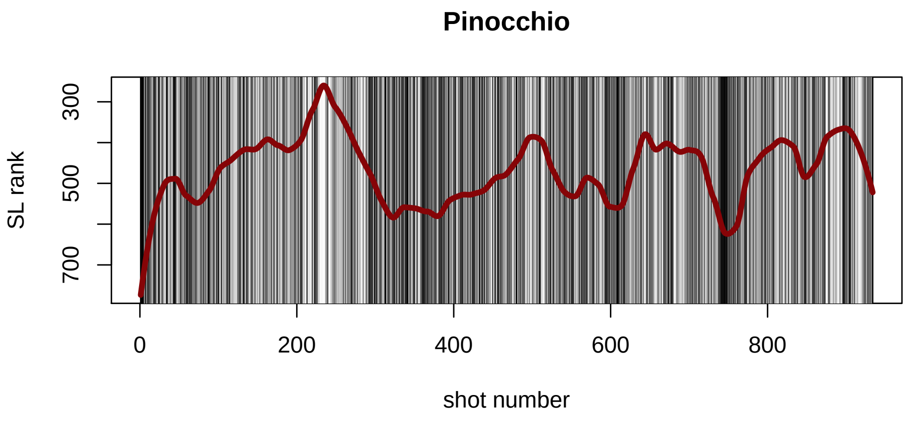


Figure 8.7: *Colour-coded plots based on the KDEs for* The House on Sorority Row *and* Pinocchio *(1940), based on a bandwidth of 1, and gray-scale ranked data with superimposed loess (1/9, g, 2) smooths.*

The House on Sorority Row is an accredited slasher film; *Pinocchio* (1940) is not usually admitted to the canon. Google brings up an interesting summary of a topic noting 'Disney's 1940 Pinocchio, and … House on Sorority Row, the similarly themed and plotted 1983 flick' but, alas, the ellipsis conceals the fact that mention of the two films is separated by an intervening distance in the full text. The films are rarely mentioned in the same breath.

Looking at the upper plots in Figure 8.7, and ignoring the KDEs for a moment, that the two film have distinctly different structures, one characteristic of a slasher film, is not something that screams out at me. If anything *Pinocchio* looks bloodier in its later stages, but as this perception is affected by the choice of colour this observation can be dismissed. The more 'rhythmic' build-up in the *The House on Sorority Row* to the ending, between (about) plotting positions 400 and 900 may indicate an intentionally structured difference, and something of the sort was also evident for Halloween. There are, however and emotive colouring apart, other difficulties with this kind of plot. One is that the smoothing based on the KDE may either induce or conceal perceptions of structure in the data; another is that the nature of the gradation in colour can affect perceptions of emphasis given to different portions of the graph.

The gray-scale coding of ranked SL data in the lower plots is also not without problems. Unaided by the plotted smooth, a loess fit in this case, the 'rhythmic' temporal pattern in *The House on Sorority Row* just alluded to is no longer so evident.



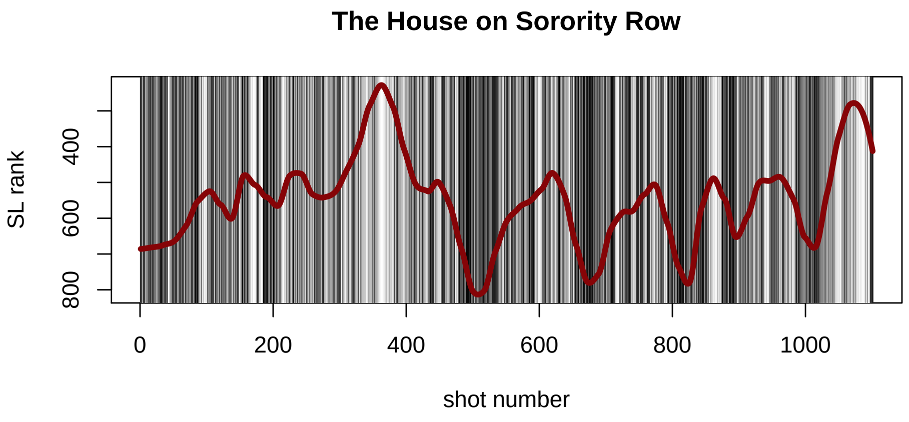


Figure 8.8: *Gray-scale plots for the ranked data of* Pinocchio *and* The House on Sorority Row, *with a different aspect ratio from that in Figure 8.7.*

The generally 'darker' nature of the second half of the plot for *The House on Sorority Row*, with the, perhaps, more prominent darker banding is at least partly a perceptual artefact of the aspect ratio chosen for the figure. Stretch out the plots, as should maybe have been done in the first place, now remedied in Figure 8.8, and I, at least, find it hard to see major differences in the structures.

This is by way of concluding that, detached from other methods of representation, graphics based on graded colour-coding of shots have their limitations for the purposes of comparison. This applies to order structure matrix plots, colour-coded ranked data (essentially the same thing) or colour-coded KDEs. 'Time' is represented by its rank in the first kind of plot; by cut-points in KDE plots; and either can be used for the second kind of plot. SLs are coded by their rank in the first two kind of plots, and by estimates derived from KDEs in the final kind. These last might be thought of as 'adjusted' SLs, though defined differently from the procedure used in Cutting *et al.* (2011).

I find smoothed estimates, which can be selected with the aid of colour-coded plots, more satisfactory for visual comparison. Looking at those in Figure 8.7 suggests that for any particular film much the same pattern is suggested.

The reason for embarking on the development leading to the idea of colour-coded KDEs was, though, motivated more by an interest in numerical than graphical comparison. The patterns exhibited by the overlaid smooths in the upper and lower plots of Figure 8.7 are similar enough to suggest that if this mode of summary is preferred the choice of analysis using either cut-points or ranks, and hence KDEs or loess smooths, is not critical (see, also, Section 7.7). KDEs of cut-points are used in what follows.

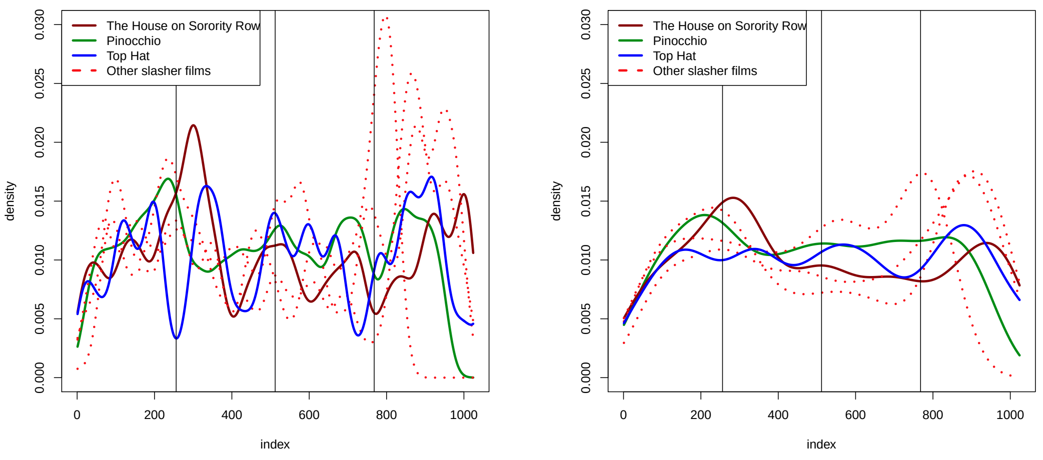


Figure 8.9: *KDEs for the cut-points of several films, standardised to eliminate differences in length. Bandwidths of 2 and 4 are used for the left and right plots; vertical lines divide the films into (temporal) quarters.*

The KDEs of *The House on Sorority Row* and *Pinocchio* are compared in Figure 8.9 using bandwidths of 2 and 4. *Top Hat* (1935) and three other slasher films, *Halloween* (1978), *Friday the 13th* and *The Slumber Party Massacre* (1982), are thrown in for good measure[11].

The method of construction, based on 1024 plotting positions running from the beginning to end of the film, ensures that a direct visual comparison unaffected by the differing lengths of the films films is legitimate, but is unavoidably affected by the level of smoothing chosen. To my eye *Top Hat* looks to have distinctive qualities (more 'rhythmic' and 'peakier'), while three of the

[11]The slasher films are those analysed in Redfern (2012c) and the data were obtained via his research blog. A common editing pattern is claimed for all four films though some of the singularities of *The House on Sorority Row*, whose KDE looks different from the other three, are noted.

four slasher films, *The House on Sorority Row* being the exception, are distinguised by the more intense 'action' – equating this with shorter shots – in the final quarter or so.

The quarters were inserted with the ideas of Cutting *et al.* (2011) about 'act structure' in films in mind. The method of deriving 'adjusted' SLs is possibly simpler than that of Cutting *et al.* without the flaws subsequently admitted in their adjustment procedure (Cutting *et al.* (2012)), but depends on the level of smoothing, and there is no obvious 'right' way of doing this. With the eye of faith, and ignoring *Top Hat*, one might discern in the smoother plot of Figure 8.9, a pattern where over the first quarter or so 'action' rises to a peak, eases off and flattens out for a bit, then begins to rise again about two-thirds to three-quarters of the way through, before declining at the end. The less smooth plot shows considerable variation within these broad divisions.

The main point to make is that the KDE for a film is a *function*, quantified by measurements at (in this instance) 1024 equi-spaced points. There is a body of statistical methodology, functional data analysis (http://www.psych.mcgill.ca/misc/fda/ - accessed 01/11/2012), directed at the analysis of such data that might be useful if more 'targeted hypotheses' than anything attempted here can be formulated, that would benefit from a quantitative comparison that goes beyond the graphical.

I can see considerable problems in implementing this kind of idea. One general thought is that given two functions quantified as described, it is possible to measure the 'distance' between them and – in principle – given data for a body of films to identify clusters of films with similar profiles. Another idea would be to identify the number and location of important bumps and troughs in a KDE and compare these across films. This is not especially easy to begin with, and complicated both by the dependence on the bandwidth chosen and the fact that films with otherwise similar profiles would be measured as distant from each other if the profiles were displaced, unless corrective action was taken. It's possible that analagous problems have been dealt with in the literature, but I don't know it well enough to offer comment.

The merits, if any, of this kind of methodology applied to cinemetric data might be better studied applied to the simpler problem of comparing SL distribution. This was the area in which I first had the thought. There is already an 'hypothesis' out there, that a large body of films have an approximate lognormal distribution. This implies that the distributions are approximately normal after after log-transformation, and data so transformed can be subsequenly standardised. The distributions after this treatment can be estimated using KDEs which might then be compared using the ideas outlined above.

That this is a simpler problem is because any pattern in the data will be much less complex that for SL patterns within films; the pattern would be very simple indeed if all SL distributions were lognormal. In principle it would be possible to pick-up different recurring patterns distinct from (log)normality so the idea goes beyond just seeing whether or not lognormality applies generally.

My earlier attempts at investigating this, before thinking of the methodology decsribed above, might charitably be described as a miserable failure. A major problem was finding a way to appropriately weight tail behaviour, where differences may be important but less 'visible'. I suspect this problem will remain, but it might be worth re-visting the idea.