**3 – Introduction to Smoothing (Nonparametric Regression)**

**3.1 – Scatterplot Smoothing**

When estimating the mean of the response variable (Y) using a numeric predictor (X) we need to specify the mean and variance functions, E(Y|X) and Var(Y|X). Ideally we will assume Var(Y|X) = constant, i.e. does not depend on X but this need not be the case. If we are willing to make the assumption of constant variation then we need only concern ourselves with the mean function.

As the mean function is a function of X we express it in general as follows:

**Example 3.1** - As we saw in **Example 2.3**, we came up with several options for parametric forms of , e.g.

.

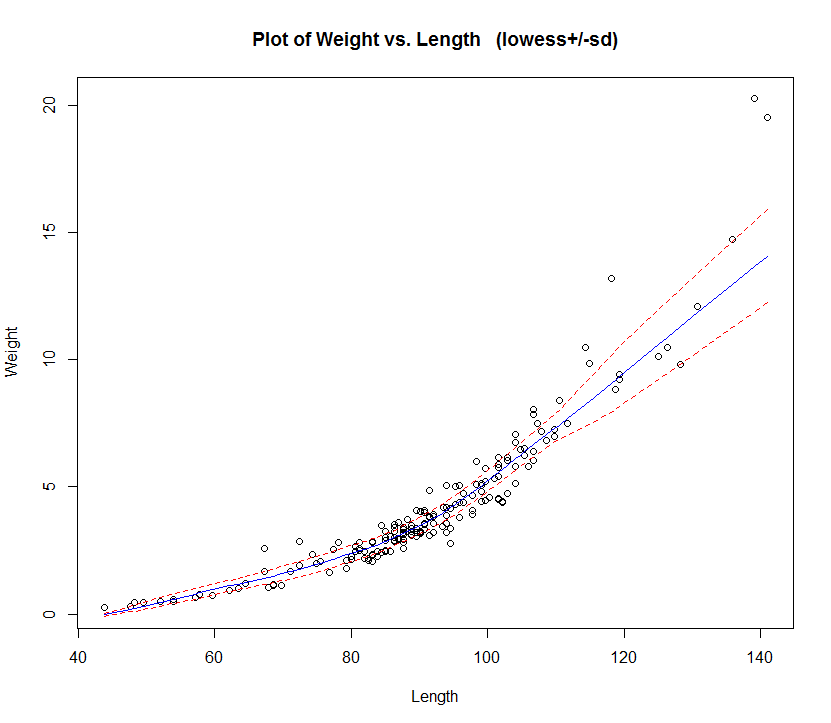
We also saw in that example that was not constant, in particular there was more variation in the weights of longer (i.e. larger) paddlefish.

In nonparametric regression we don’t specify the mean and variance functions explicitly, i.e. parametrically, but rather use *smoothing* to estimate these functions. There are numerous ways to “smooth a scatterplot” and we will examine a few of them in this section.

On the next page is a smoothed scatterplot of paddlefish weight (kg) vs. fork length (cm) with

or

added to the plot.



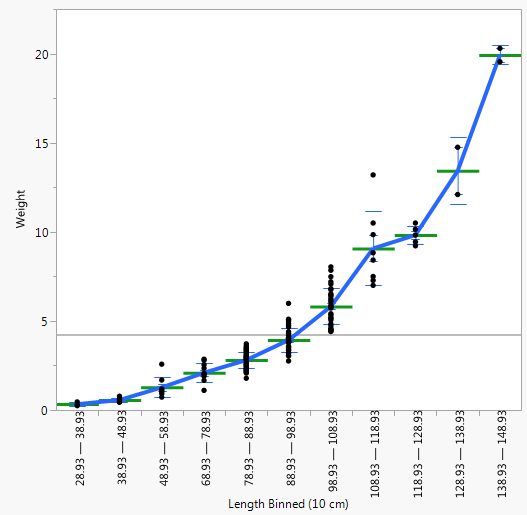
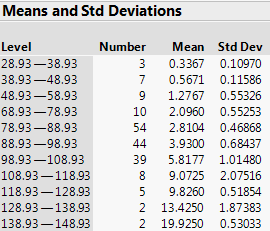
> trendscatter(Weight~Length,

data=Paddlefish)

You need to install the package s20x in order use the trendscatter function.

Here locally weighted regression (lowess) was used to conduct the smoothing. Before discussing the details of lowess smoothing we will consider simpler and more intuitive smoothing schemes that can be used.

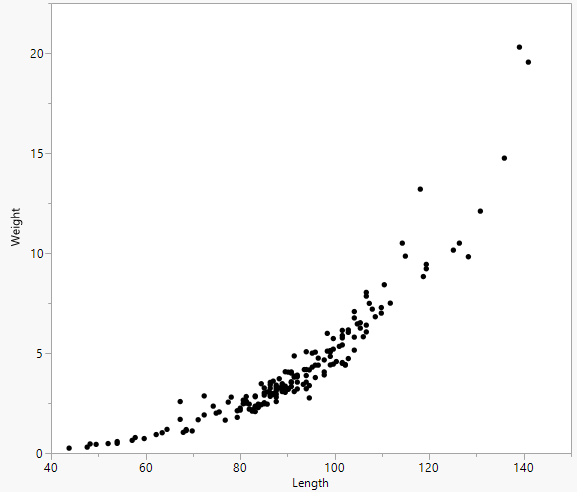
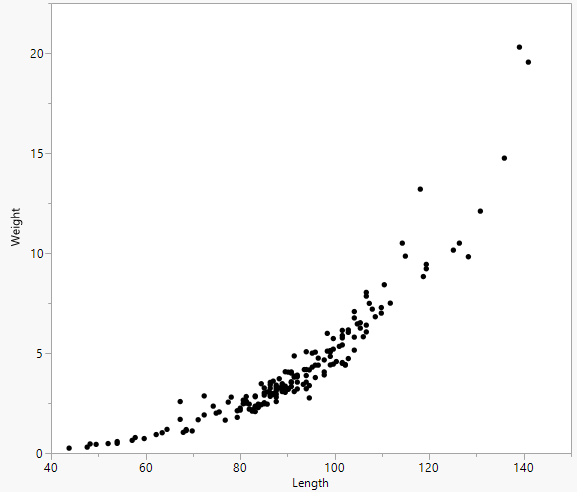
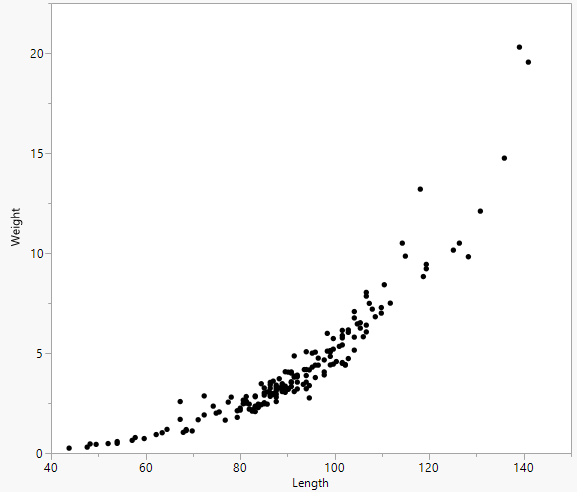
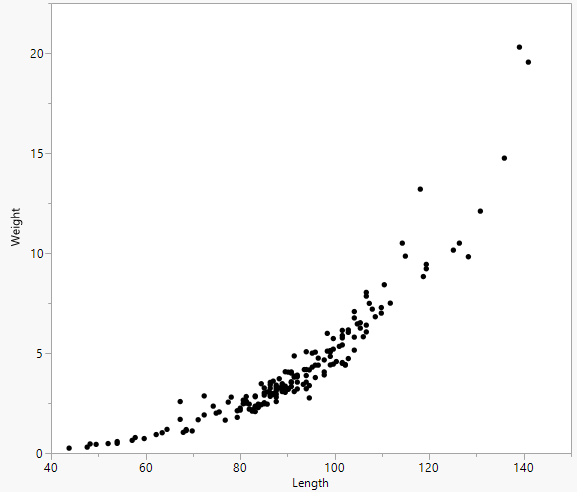
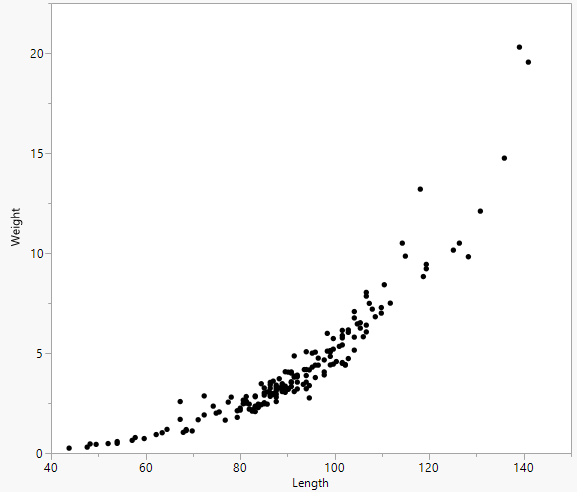
One simple scheme, that we used in Example 2.3, is form ranges/bins (or windows) of fish with similar lengths and find the sample mean and variance of the weights for the fish in the same bin or window. Below is the result of using bins of width 10 cm from Example 2.3.

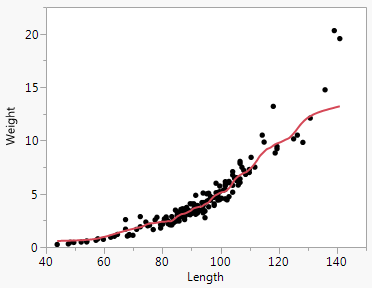
This is actually a form of smoothing. If we connect the means with a line and connect the +/- SD(Weight|Length) marks with a line as well we have mean and variation smooths looking looking pretty similar to those obtained using lowess smoothing in R.

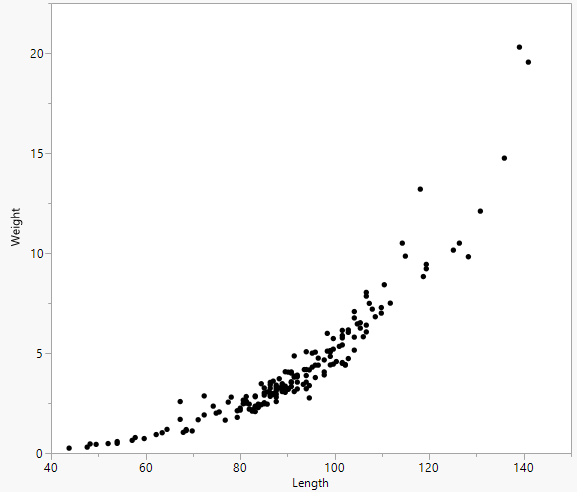
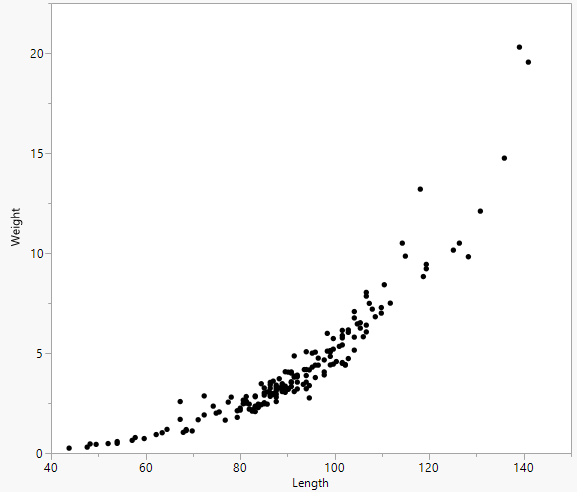
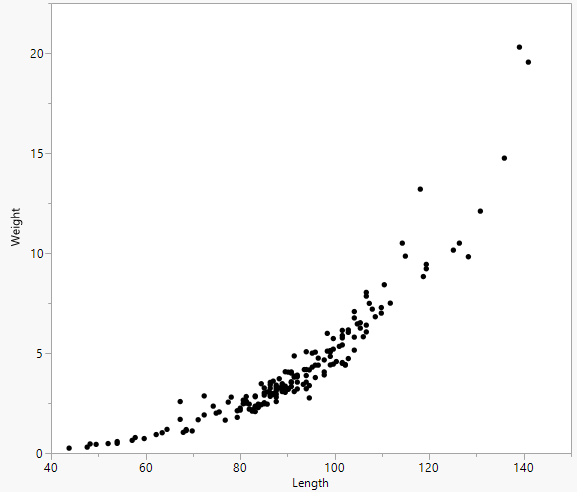
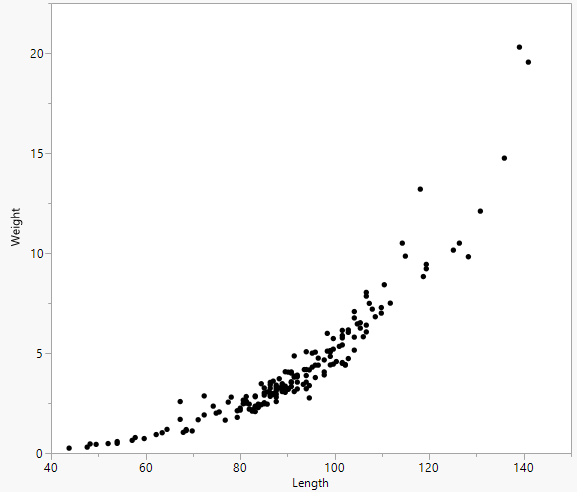
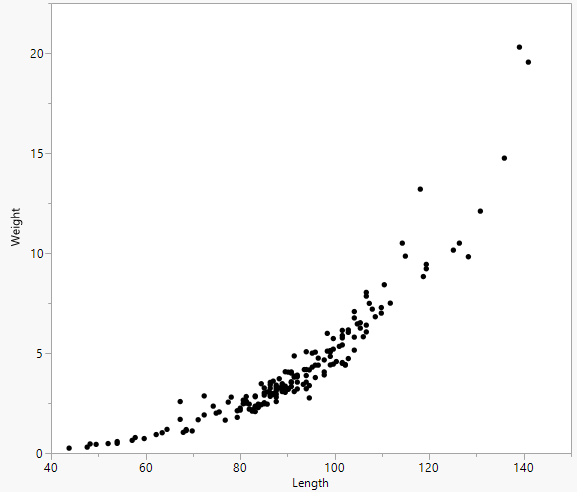
Rather than having static or fixed bins as in the example above, a better approach is set a window (or bin) width and move that window through the data from left to right computing means or fitting a line within the window as we go. The diagrams below demonstrate both of these approaches applied to these data.

Local Mean Smoothing – estimation using the mean of the response within the window. The window moves continuously through the data from left to right. The smooth simply connects the window means.

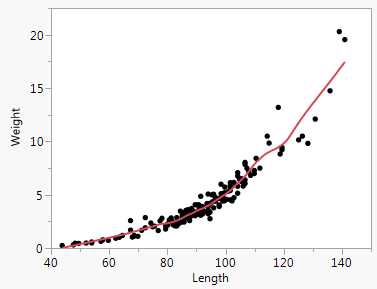


The results of local mean smoother for these data is shown below:

  
  
LOESS Smoothing – fitting a line within each window (no observation weighting). Again the window moves continuously from left to right. The final smooth connects estimated target values.

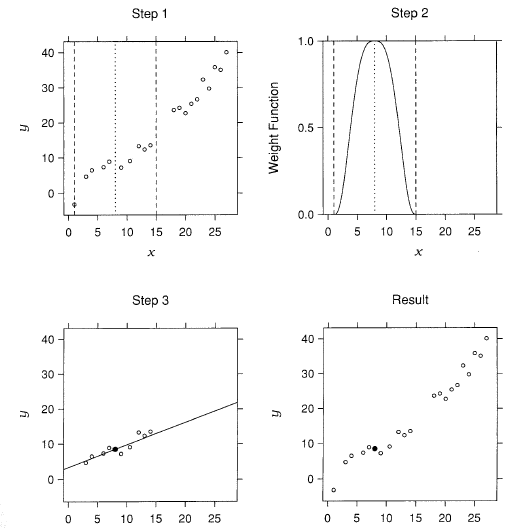


The results of local regression smoother for these data is shown below:

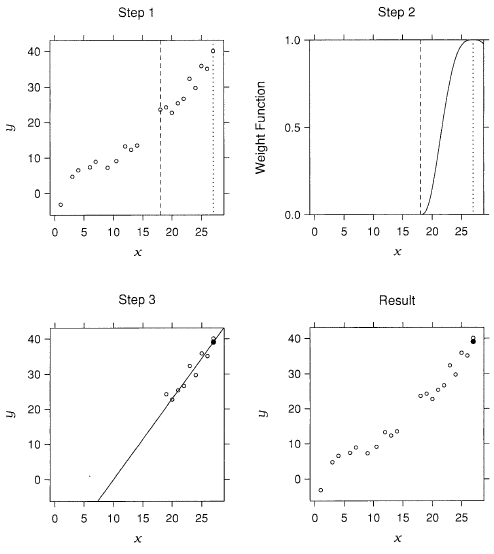


Most smoothers do not weight each observation equally. Points near the *target point* receive more weight than the observations near the edges of the window. Different weight (kernel) functions are used for this purpose such as the gaussian (normal) distribution or *tri-cube* function used in the local regression examples below (taken from *Visualizing Data*, Cleveland, 1993).

Locally weighted regression (lowess) at x = 8

  
 *x* *x*

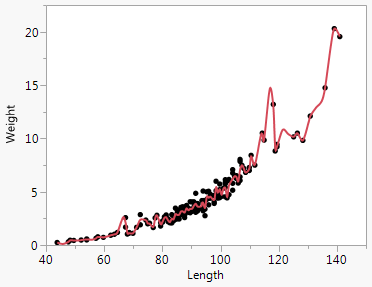
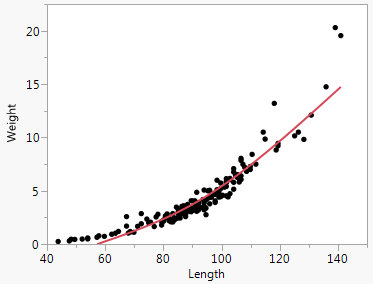
Locally weighted regression (lowess) at x = 27



Again the final smooth simply connects the fitted values within each window.

To control the smoothness of the final estimate can vary the window width. Narrow windows will produce noisy or “wiggly” estimates and wide windows will produce overly smooth estimates do not capture important structure.

is too noisy is a bit too smooth

Window width can be controlled through the width of the weight function or by choosing a certain fraction of the total observations ( *f* ) to be contained in the windows, e.g. *f = .25* means 25% of the observations will be contained in the windows. Note that with the fraction approach not all windows will be the same width.

To estimate the variance function , and hence , we can use the following procedure.

1. Form the smooth estimate of the mean function, and find the residuals

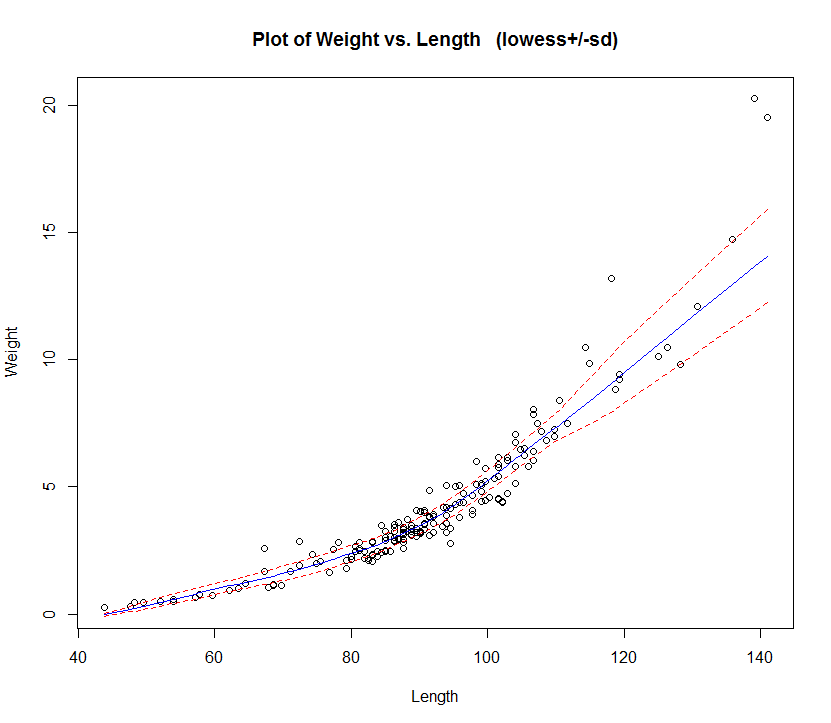
.

1. Form the squared residuals and smooth the scatterplot of . This smooth is an estimate of the conditional variance, i.e. . The square root of this estimate gives .
2. We can then add the lines given and to obtain a visualization of how variability changes a function of .

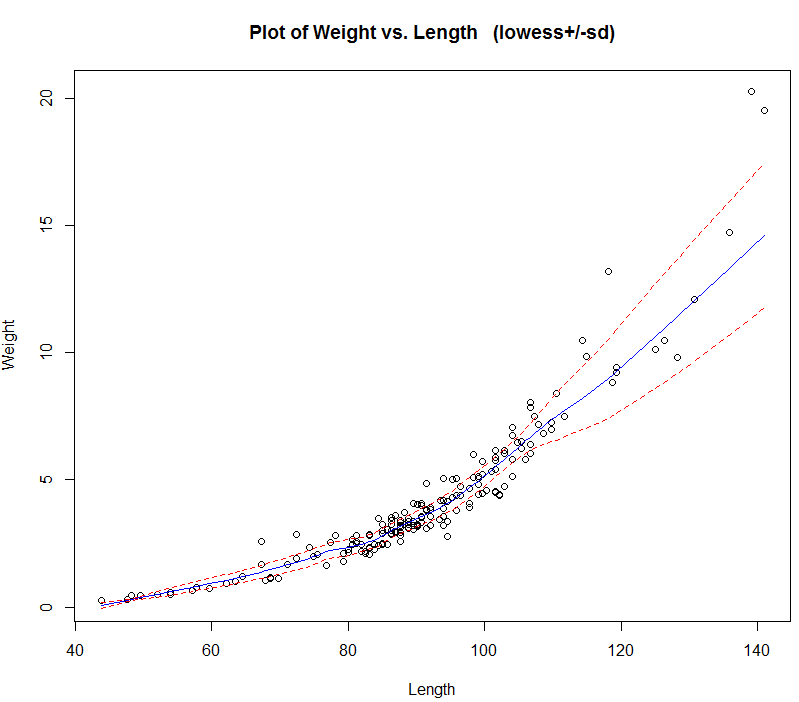
The function trendscatter(Y~X,data) in the s20x package uses lowess smoothing to estimate the mean and variance/standard deviation functions as outlined above.

> library(s20x)

> trendscatter(Weight~Length,f=0.5,data=Paddlefish)

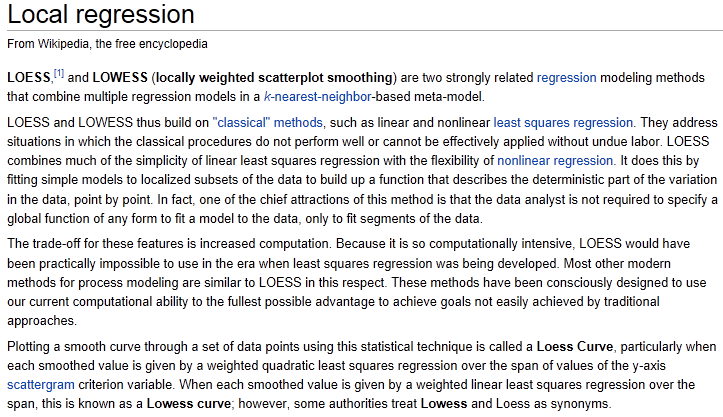


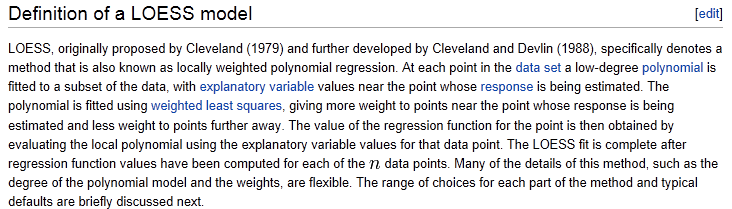
> trendscatter(Weight~Length,f=0.25,data=Paddlefish)



Wiki Entry for Lowess Smoothing

<http://en.wikipedia.org/wiki/Local_regression>

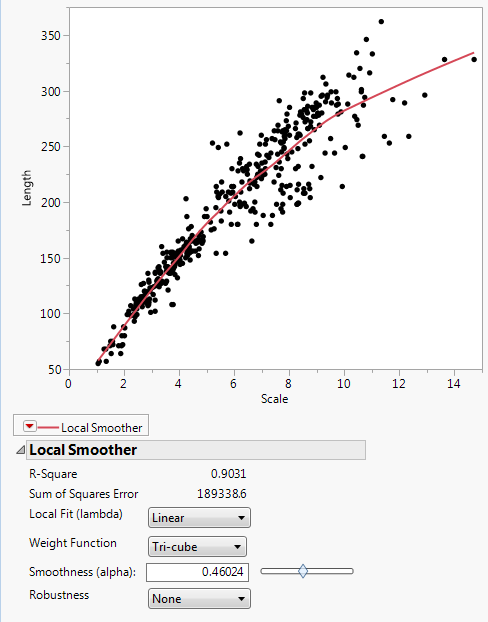
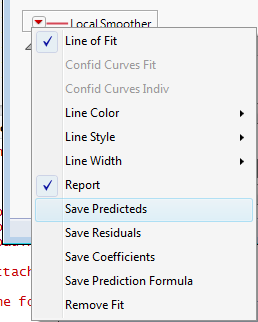




**3.2 –More Examples of Mean & Variance Function Estimation via Smoothing**

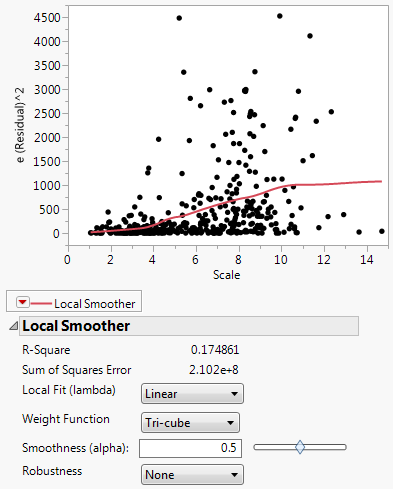
**Example 3.2** – Length of Smallmouth Bass (Y) and Scale Radius (X)

We can go through the process of estimating ***E(Y|X)***, ***Var(Y|X)*** and ***SD(Y|X***) as outlined above. We first construct a scatterplot of length (mm) vs. scale radius (mm).

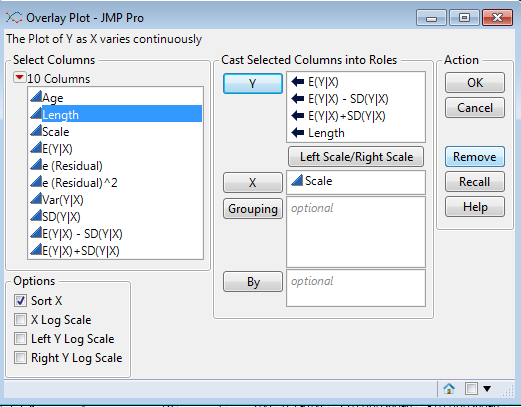
– nonparametric estimate of the mean function.

We can **Save Predicteds** to obtain (or ) and **Save Residuals** () which can be used to estimate the and as outlined above. After saving the residuals we can square them and then smooth them versus scale radius as shown below to obtain a nonparametric estimated of .

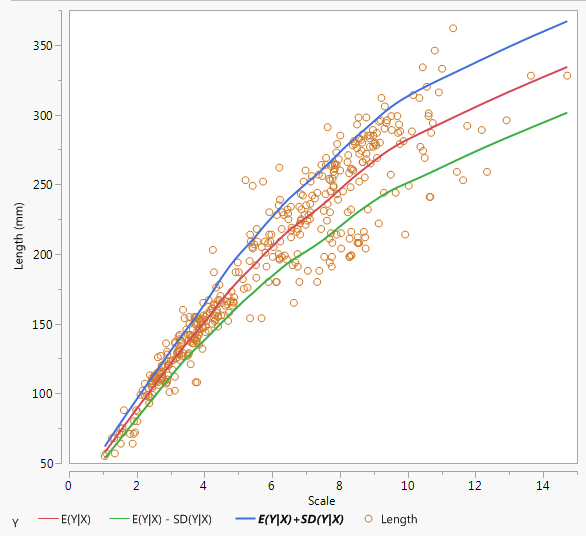


– lowess smooth of . We then Save Predicteds from this smooth and take the square root of the predicted values to obtain .

We can then add and subtract from and display them all in a **Graph > Overlay Plot**.



The resulting plot is shown below.

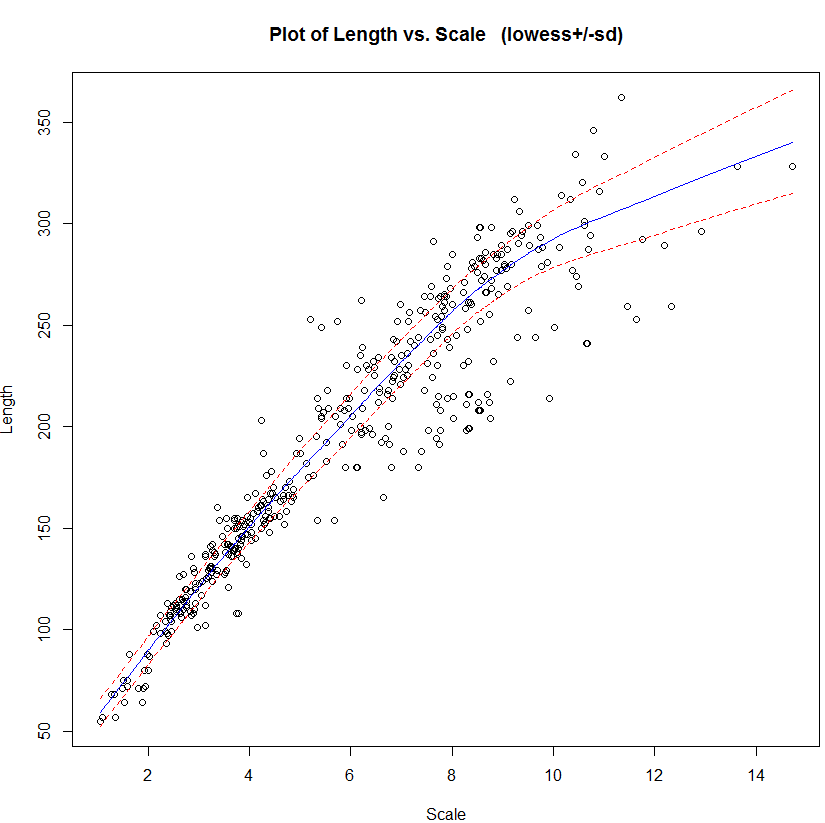


This can be achieved by using the trendscatter command in the s20x package in R.

> library(s20x)

> wblake = read.table(file.choose(),header=T,sep=”,”) 🡨 download ***wblake.txt*** from my website.

> trendscatter(Length~Scale,data=wblake)



**3.3 – Tasmanian Abalones (Paua)**

Predicting the age of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

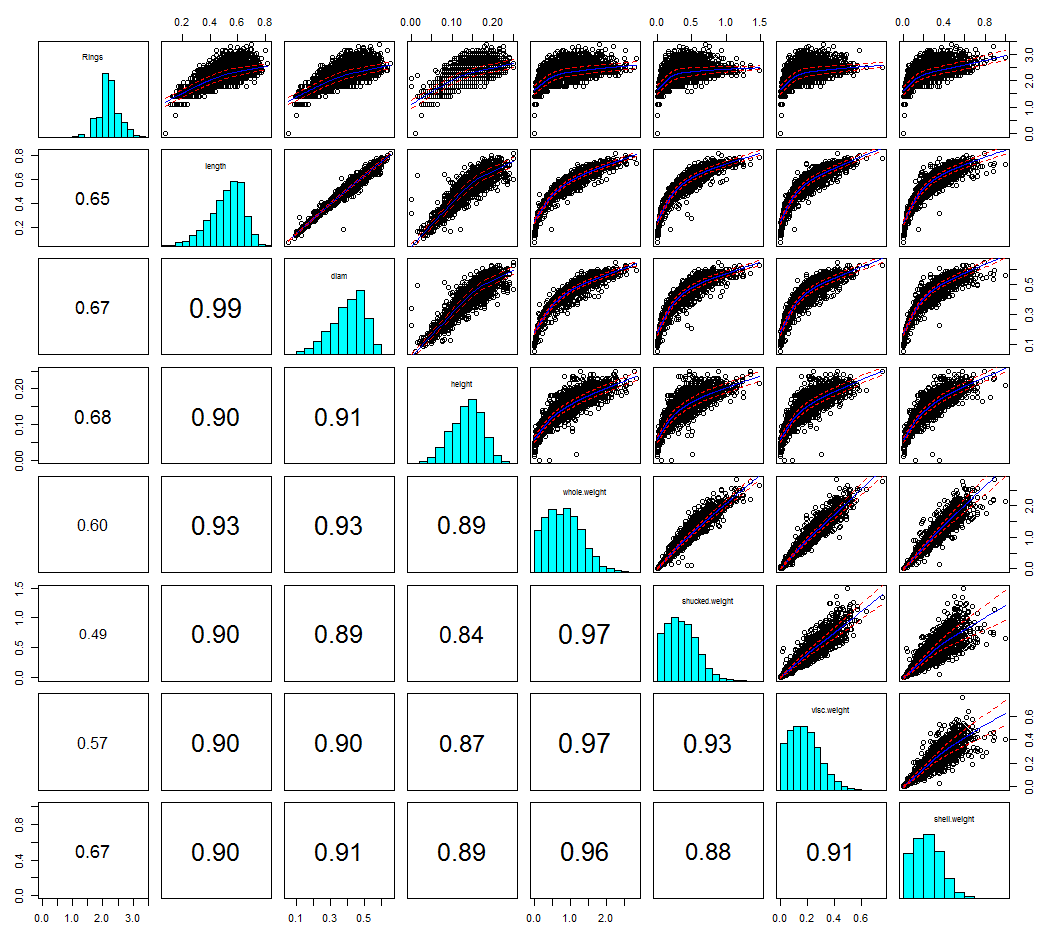
Variables:

* Rings / integer / -- / +1.5 gives the age in years
* Sex / nominal / -- / M, F, and I (infant)
* Length / continuous / mm / Longest shell measurement
* Diameter / continuous / mm / perpendicular to length
* Height / continuous / mm / with meat in shell
* Whole weight / continuous / grams / whole abalone
* Shucked weight / continuous / grams / weight of meat
* Viscera weight / continuous / grams / gut weight (after bleeding)
* Shell weight / continuous / grams / after being dried

These data are contained in the data frame Abalone in mult.Rdata folder.

> pairs.plus(Abalone)

**Notes:**

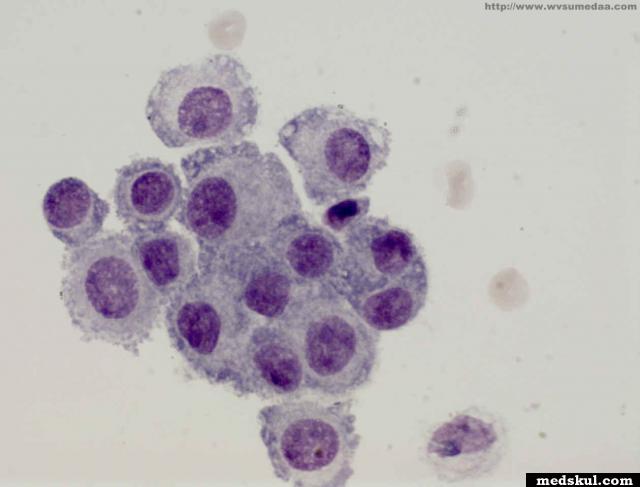
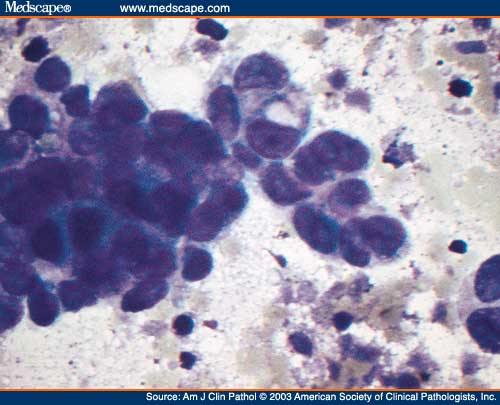


**Example 3.4 – Cell Radius, Perimeter, and Area of Breast Tumor Cells**

Consider the following subset of the variables from the Wisconsin Diagnostic Breast Cancer dataset that is commonly used by those studying machine learning. The goal from a machine learning perspective is to use the cell measurements of the cells sampled using a fine needle aspiration (FNA) to classify a breast tumor as benign or malignant. These data were previously used in **Example 2.1**.

***Original Source***: W.N. Street, W.H. Wolberg and O.L. Mangasarian *Nuclear feature extraction for breast tumor diagnosis*. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, Volume 1905, pages 861-870, San Jose, CA, 1993.

***Benign Cells Malignant Cells***

In this example we consider the radius, perimeter, and area of the cells in this study. Again keep in mind there are both malignant and benign tumor cells in our sample, (347 benign, 212 malignant).

One question we might consider is to what degree breast tumor cells are circular (2D)/spherical (3D). From basic geometry we know that for circles the area is given by and the perimeter is given by (see below).

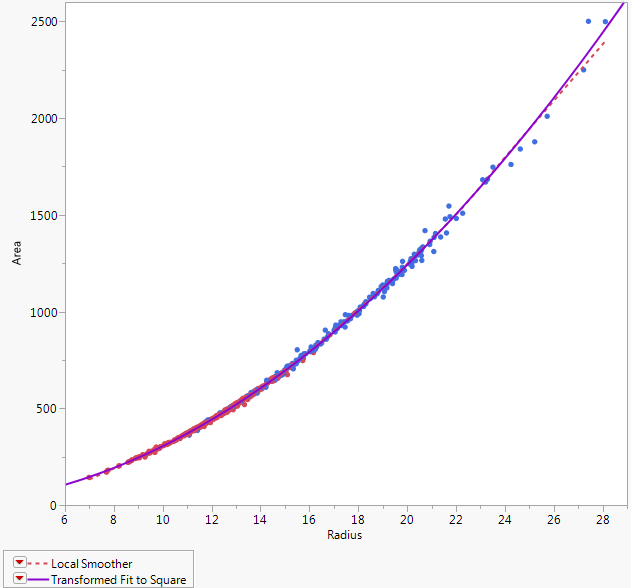
|  |  |
| --- | --- |
| **Are breast tumor cells circular?** | |
| Area | Perimeter |

Does the answer to the circularity question depend on what cell type we are considering, i.e. benign or malignant? To check that theory we can smoothers to estimate the *E(Area|Radius)* and *E(Perimeter|Radius)* and compare to the theoretical/geometric conditions for circularity.

From above we know if breast tumor cells are perfectly circular in shape then

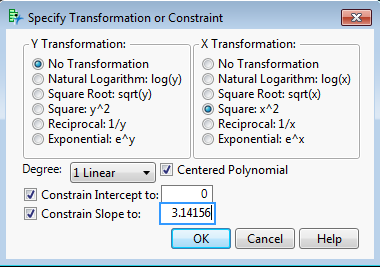
and

Below is a plot of cell Area vs. cell Radius for the cells sampled in this study with a lowess smooth and the theoretical function added.

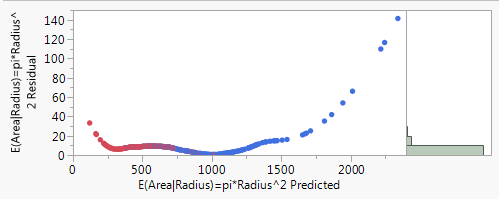
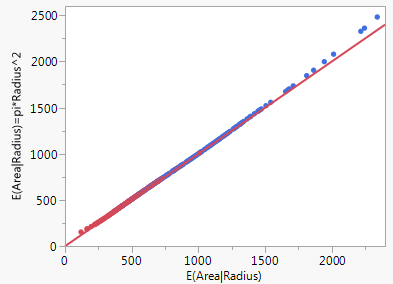
   
Select **Bivariate Fit > Fit Special** and set up the dialog box as shown below to fit the circular theoretical model, .

The solid “blue” line is the theoretical model that says the cells are circular and the dashed line is the lowess smooth estimate of .

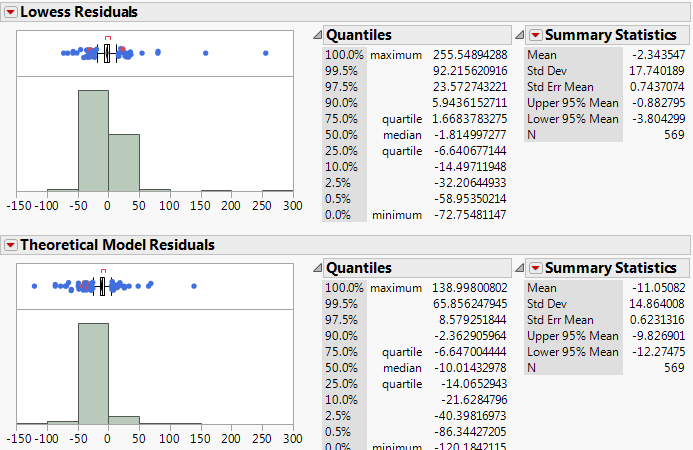
Do the two estimates agree? They certainly appear to, except possibly for larger malignant cells.



We can save the mean functions from each of these estimates and compare them directly by selecting **Save Predicteds** from the **Local Smooth** and **Transformed Fit to Square** drop-down menus. Below is a plot of with the line Y = X added to the plot.

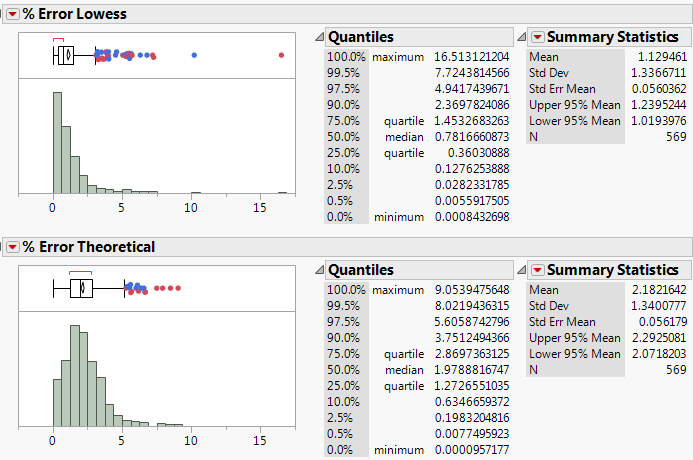


We can see that the two estimates do not agree, but which is more correct in terms of the actual cell areas? We can compute the residuals where and from the model (lowess or theoretical).



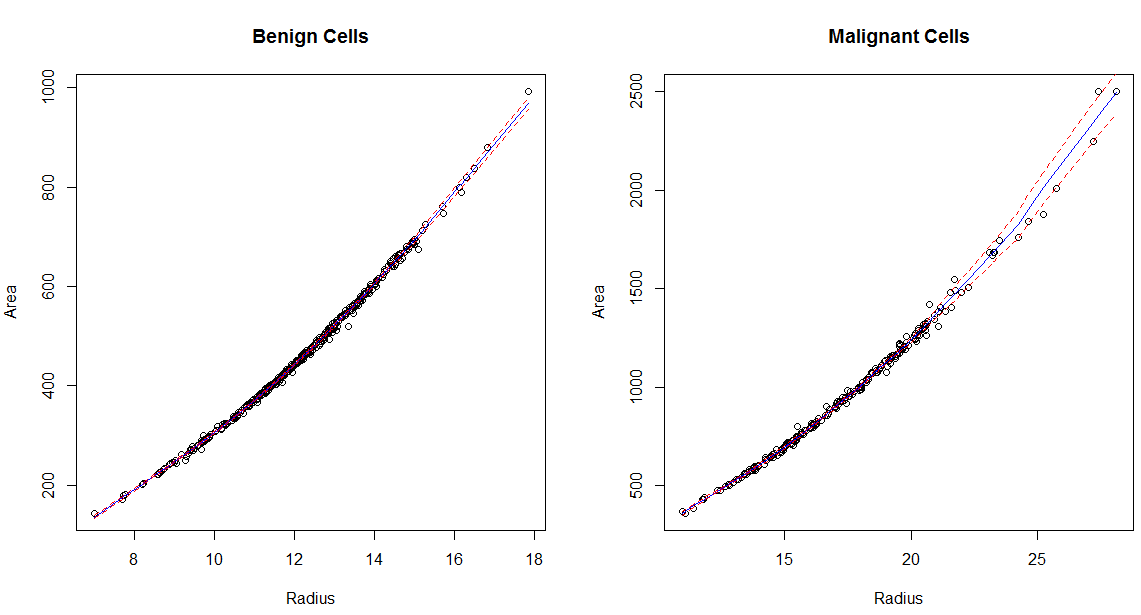
Or we could compute the absolute percent error

The percent errors distributions for both fits are shown below.



The absolute percent errors from the theoretical model are definitely larger than those from the *data-driven* nonparametric model. However in the theoretical model, no parameters are being estimated from the data!

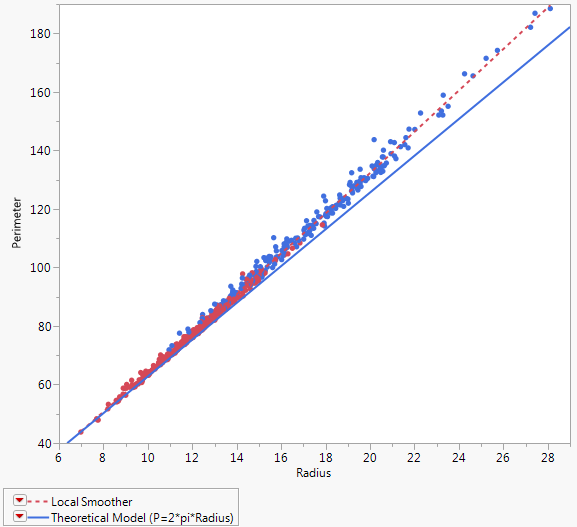
Below is a comparison of the results from trendscatter(Area~Radius) conditional on tumor type. Are there any differences in the estimated mean and variance functions that are apparent?



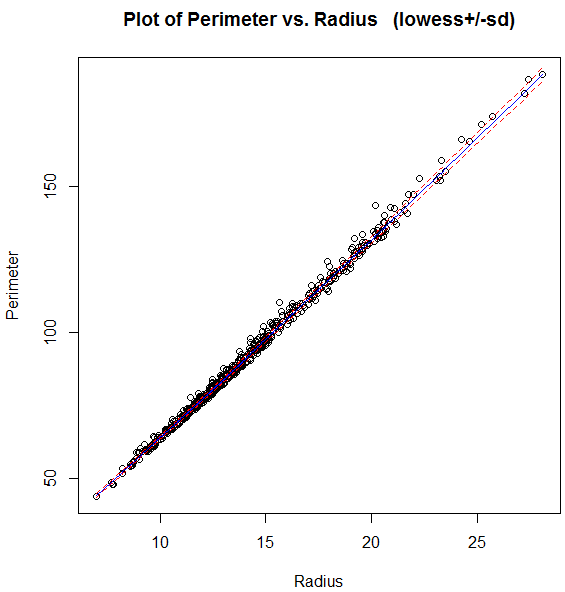
> trendscatter(Area~Radius,data=BreastDiag[BreastDiag$Diagnosis=="B",],f=.2,  
main="Benign Cells")

> trendscatter(Area~Radius,data=BreastDiag[BreastDiag$Diagnosis=="M",],f=.2,  
main="Malignant Cells")

For perimeter the results of the theoretical model are even less promising. Below is a plot of cell perimeter vs. radius. The dashed line is the lowess smooth fit to these data and the solid “blue” line is the line , i.e. a line with y-intercept = 0 and slope = .



Clearly from a perimeter perspective, tumor cells are not perfectly circular!



***What can we say about***?

=

=

> trendscatter(Perimeter~Radius,data=BreastDiag)