• When the nature of the response function (or mean structure) is unknown and one wishes to investigate it, a nonparametric regression approach may be useful.

• An example of a nonparametric regression fit for $Y$ vs. $X$:
• One method of nonparametric regression is **LOWESS**.

  **LO**cally
  **WE**ighted
  polynomial
  regre**SS**ion

• A lowess curve doesn’t assume any particular form to the mean structure.

• Instead, the data points themselves suggest the form of the relationship, allowing a mean structure that changes freely as $x$ changes.
• Though it looks like someone has simply drawn a line through the points, the fitting of the lowess curve is based on statistical modeling.

• Actually, a particular polynomial regression model will be fit many times over, but each time, a different *window* of the data will be used to fit the model.

• Thus, the term ‘local’ coincides with the fact that we’re only using a fraction of the data, a *window*, each time we do a fit.
• General idea for computing a $\hat{Y}_i$ (point on curve)

<table>
<thead>
<tr>
<th>Observed $x_i$</th>
<th>Observed $Y_i$</th>
<th>Fitted $\hat{Y}_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11,377</td>
<td>73.1</td>
<td>?</td>
</tr>
</tbody>
</table>

– Choose a window containing $x_i$. The points in the window are the only points that will contribute to the fit of the locally weighted polynomial at $x_i$.

*Wider windows give smoother lowess curves.*
– Before fitting the local model, we determine the weights of the points in the window.

Higher weights will be placed on the points near \( x_i \) (local weighting), and lower weights will be placed on the points far from \( x_i \), near the window edges.

One weight function used is the Tricube weight function:

\[
T(t) = \begin{cases} 
(1 - |t|^3)^3 & \text{for } |t| < 1 \\
0 & \text{for } |t| \geq 1
\end{cases}
\]

where \( t = \frac{x_{ij} - x_i}{h_i} \) for all observations \( x_{ij} \) in the window for \( x_i \), and \( h_i \) is the half-width of the window.
Use the weights (and the data) to fit a local polynomial regression. A first-order local polynomial (straight line) usually suffices (red line below shows best-fit-line).

Let $\hat{Y}_i$ from this locally fitted polynomial regression be the prediction of $Y_i$ at $x_i$. 

11,377
Then, we have no further use for this locally fitted polynomial.

All that work, for one fitted point $\hat{Y}_i$.

<table>
<thead>
<tr>
<th>Observed $x_i$</th>
<th>Observed $Y_i$</th>
<th>Fitted $\hat{Y}_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>11377</td>
<td>73.1</td>
<td>68.34</td>
</tr>
</tbody>
</table>

— Connect the $n$ fitted $\hat{Y}_i$ values (all computed in the same manner) to form the mean structure.
• You have some choice in how ‘smooth’ the fitted mean structure is.

A ‘smoother’ curve using the $f = 1$ setting:

A less smooth curve using $f = 1/3$:
In R using the `lowess` function:

```r
> attach(Prestige)
```

```r
## Lowess plot example:
> plot(income,prestige,pch=16)
> lowess.out=lowess(income,prestige,f=1/3)
> lines(lowess.out,lwd=2)
```

where \( f \) is the the smoothing parameter.

\( f \) gives the proportion of points in the plot which influence the smooth at each value.

Even if all points are included in the window \((f = 1)\), the points still have differing weights in the polynomial fit.

Larger values of \( f \) give more smoothness.
• You can also access all the fitted $\hat{Y}_i$ values for the given $x_i$. The results are returned in ascending order by the x-values:

```r
> lowess.out
$x
 [1] 611 918 1656 1890 2370 2448 2594 2847 2901 ...

$y
 [1] 19.28922 20.79308 24.32462 25.38495 27.52914 27.96910 ...

> points(lowess.out$x,lowess.out$y,col="purple",pch=1)
```
• There is another function that does local polynomial regression fitting called *loess*, which uses a slightly different weighting scheme (you can reset this if you want), but can easily be used for prediction at new x-values.
• The author of our book has also provided a lowess function inside the `scatterplot()` function in the car library.

```r
> scatterplot(income,prestige,smooth=T,span=1/3)
```

Dotted line \(\equiv\) Simple linear regression fit.
Solid line \(\equiv\) lowess fit.
...and in the `scatterplot.matrix()` function.

```r
> scatterplotMatrix(Ginzberg)
```
**Lab Example** (frog data)

In logistic regression, it can be useful to use the lowess fit to ‘suggest’ a model, or check on the relationship between $X$ and $Y$:

\[ f = \frac{2}{3} \text{ (default)} \]

\[ f = \frac{1}{4} \]

\[ f = \frac{1}{10} \]
(a) Outside dotted lines define window for observation \( i \), middle dotted line is on x-value of prediction (\( i^{th} \) obs).

(b) Weighting scheme. Higher weights to nearby points.

(c) Local polynomial fit (extends past window). \( \hat{Y}_i \) falls on the fitted line.

(d) Overall lowess curve (after repeating process for every observation and connecting the predicted points).