10.Problem: Implement the non-parametric Locally Weighted Regression (LOWESS) algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

In [ ]:

```
# Author : Dr.Thyagaraju G S , Context Innovations Lab , DEpt of CSE , SDMIT - Ujire
# Date : July 11 2018
# Reference: https://medium.com/100-days-of-algorithms/day-97-locally-weighted-regression-
c9cfaff087fb
```

In [28]:

```
import numpy as np
from bokeh.plotting import figure, show, output_notebook
from bokeh.layouts import gridplot
from bokeh.io import push_notebook

output_notebook()
```

Loading BokehJS ...

In [29]:

```
import numpy as np

def local_regression(x0, X, Y, tau):
    # add bias term
    x0 = np.r_[1, x0] # Add one to avoid the loss in information
    X = np.c_[np.ones(len(X)), X]

    # fit model: normal equations with kernel
    xw = X.T * radial_kernel(x0, X, tau)   # XTranspose * W

    beta = np.linalg.pinv(xw @ X) @ xw @ Y   # @ Matrix Multiplication or Dot Product

    # predict value
    return x0 @ beta    # @ Matrix Multiplication or Dot Product for prediction

def radial_kernel(x0, X, tau):
    return np.exp(np.sum((X - x0) ** 2, axis=1) / (-2 * tau * tau))   # Weight or Radial Kernal Bias
Function
```

In [30]:

```
n = 1000
# generate dataset
X = np.linspace(-3, 3, num=n)
print("The Data Set ( 10 Samples) X :\n",X[1:10])
Y = np.log(np.abs(X ** 2 - 1) + .5)
print("The Fitting Curve Data Set (10 Samples) Y  :\n",Y[1:10])
# jitter X
X += np.random.normal(scale=.1, size=n)
print("Normalised (10 Samples) X :\n",X[1:10])
```

```
The Data Set ( 10 Samples) X :
 [-2.99399399 -2.98798799 -2.98198198 -2.97597598 -2.96996997 -2.96396396
 -2.95795796 -2.95195195 -2.94594595]
The Fitting Curve Data Set (10 Samples) Y  :
 [ 2.13582188  2.13156806  2.12730467  2.12303166  2.11874898  2.11445659
  2.11015444  2.10584249  2.10152068]
Normalised (10 Samples) X :
 [-3.17013248 -2.87908581 -3.37488159 -2.90743352 -2.93640374 -2.97978828
 -3.0549104  -3.0735006  -2.88552749]
```

In [31]:

```
domain = np.linspace(-3, 3, num=300)
print(" Xo Domain Space(10 Samples)  :\n",domain[1:10])

def plot_lwr(tau):
    # prediction through regression
    prediction = [local_regression(x0, X, Y, tau) for x0 in domain]
    plot = figure(plot_width=400, plot_height=400)
    plot.title.text='tau=%g' % tau
    plot.scatter(X, Y, alpha=.3)
    plot.line(domain, prediction, line_width=2, color='red')
    return plot
```

```
 Xo Domain Space(10 Samples)  :
 [-2.97993311 -2.95986622 -2.93979933 -2.91973244 -2.89966555 -2.87959866
 -2.85953177 -2.83946488 -2.81939799]
```

In [32]:

```
# Plotting the curves with different tau
show(gridplot([
    [plot_lwr(10.), plot_lwr(1.)],
    [plot_lwr(0.1), plot_lwr(0.01)]
]))
```