2.

```
nsamp, =u.shape
ns_train=nsamp/2
ns_test=nsamp/2
X_tr=u[:ns_train]
y_tr=y[:ns_test]
X_test=u[ns_train:]
y_test=y[ns_train:]
dtest=np.arrange(1,11)
RSS_test=[]
for index in range(len(dtest)):
    regr=linear_model.LinearRegression()
    X_tr=np.exp((-X_tr/dtest[index])*np.arrange(dtest[index]+1))
    regr.fit(X_tr,y_tr)
    X_test=np.exp((-X_test/dtest[index])*np.arrange(dtest[index]+1))
    y_test_pred=regr.predict(X_test)
    RSS_test.append(np.mean((y_test_pred-y_test)**2)/(np.std(y_test)**2))
print min(RSS_test)
```

3.

(a)$E\left(\frac{\sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} x_i^2}\ x^2\right) - \beta_0 x^2 \ = \ E\left(\beta_0 x^2\right) - \beta_0 x^2 \ = 0$

(b) $\frac{\sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} x_i^2}\ x^2 \ - \beta_0 x^2 \ = \beta_0 x_i^2 + E\left(\frac{N\epsilon_i}{\sum_{i=1}^{N} x_i^2}\right) - \beta_0 x_i^2 \quad = 0$

(c)$E\left(\frac{\sum_{i=1}^{N} y_i}{\sum_{i=1}^{N} x_i^2}\ x^2\right) - \beta_0 (x\ \ )^2 = \beta_0 \frac{\sigma^2}{\sum_{i=1}^{N} x_i^2} x^2$

4.

(a)

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

$$\frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \beta_0} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)$$

$$\frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \beta_1} = -2 \sum_{i=1}^{n} x_i (y_i - \beta_0 - \beta_1 x_i)$$

Let $\frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \beta_0}$ and $\frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \beta_1}$ both equal to zero to calculate β_0 and β_1

(b) $\frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \beta_0} = -2 \sum_{i=1}^{n} (\beta_{00} + \beta_{01} x_i + \beta_{02} x_i^2 - \beta_0 - \beta_1 x_i)$

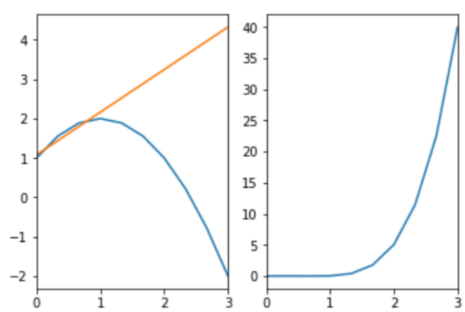
$$\frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \beta_1} = -2\sum_{i=1}^{n} x_i(\beta_{00} + \beta_{01}x_i + \beta_{02}x_i^2 - \beta_0 - \beta_1 x_i)$$

In the same way to calculate $\widehat{\boldsymbol{\beta}}$, let $\frac{\partial \widehat{\beta}}{\partial \beta_0}$ and $\frac{\partial \widehat{\beta}}{\partial \beta_1}$ both equal to zero.

(c)(d)

In [26]:
```
beta=np.array([1,2,-1])
nsamp=10
xdat=np.random.uniform(0,1,nsamp)
ydat=poly.polyval(xdat,beta)
d=1
beta_hat=poly.polyfit(xdat,ydat,d)
xp=np.linspace(0,3,nsamp)
yp=poly.polyval(xp,beta)
yp_hat=poly.polyval(xp,beta_hat)
plt.subplot(121)
plt.xlim(0,3)
plt.plot(xp,yp)
plt.plot(xp,yp_hat)
plt.subplot(122)
y = (yp_hat-yp)**2
#print (y)
plt.xlim(0,3)
plt.plot(xp,y)
```

Out[26]: [<matplotlib.lines.Line2D at 0x10f5cca90>]



so when x=3, $Bias^2(x)$ will be largest.

5.(a)x_0 : cancer volume

x_1: age

x_1 : cancer type

Model 1: $y = \beta_0 + \beta_1 x_0$

Mode12: $y = \beta_0 + \beta_1 x_0 + \beta_1 x_1$

Model3: $y = \beta_0 + \beta_1\phi_0 x_0 + \beta_2\phi_1 x_0 + \beta_3 x_1$, ϕ_0 and ϕ_1 are binary features

Type I: $y = \beta_0 + \beta_1 x_0 + \beta_3 x_1$

Type II: $y = \beta_0 + \beta_2 x_0 + \beta_3 x_1$

(b) Model 1 has two parameters

Model 2 has three parameters

Model 3 has three parameters

Model 3 is the most complex.

(c)Model 1: $A = \begin{pmatrix} 1 & 0.7 \\ 1 & 1.3 \\ 1 & 1.6 \end{pmatrix}$

Model 2: $A = \begin{pmatrix} 1 & 0.7 & 55 \\ 1 & 1.3 & 65 \\ 1 & 1.6 & 70 \end{pmatrix}$

Model 3: $A = \begin{pmatrix} 1 & 0.7 & 0 & 55 \\ 1 & 0 & 1.3 & 65 \\ 1 & 0 & 1.6 & 70 \end{pmatrix}$

(d)Rsq_tgt = 0.7 +0.05/$\sqrt{10-1}$ =0.7171

Mode13's Rss_mean is less than Rsq_tgt , so we can choose model 3.