Regression as a Special Case of Quadratic Programming

By Andrew Johnson

Install GAMS

- Go to https://www.gams.com/download/
- Download appropriate version of GAMS for your operating systems
- During installation you will be asked for your license, please use the license provided by email

Parts of a GAMS Program

```
$ontext
$offtext
set i /i1*i40/;
$include expdata.inc
alias (j,jj);
parameters;
X(i,'constant') = 1;
X(i,'coeff1')   = data(i,'income');
equations;
variables;

dummy_eq..  dummy_var =e= b('constant');
normal(j).. sum(jj, XX(j,jj)*b(jj)) =e= Xy(j);

model ols2 /dummy_eq,normal/;
solve ols2 using lp minimizing dummy_var;
display b.l;
```

Assign parameter values

Assign equation names specific equations

Linear Least Squares

$$y = \alpha + \boldsymbol{X}\beta + \epsilon$$

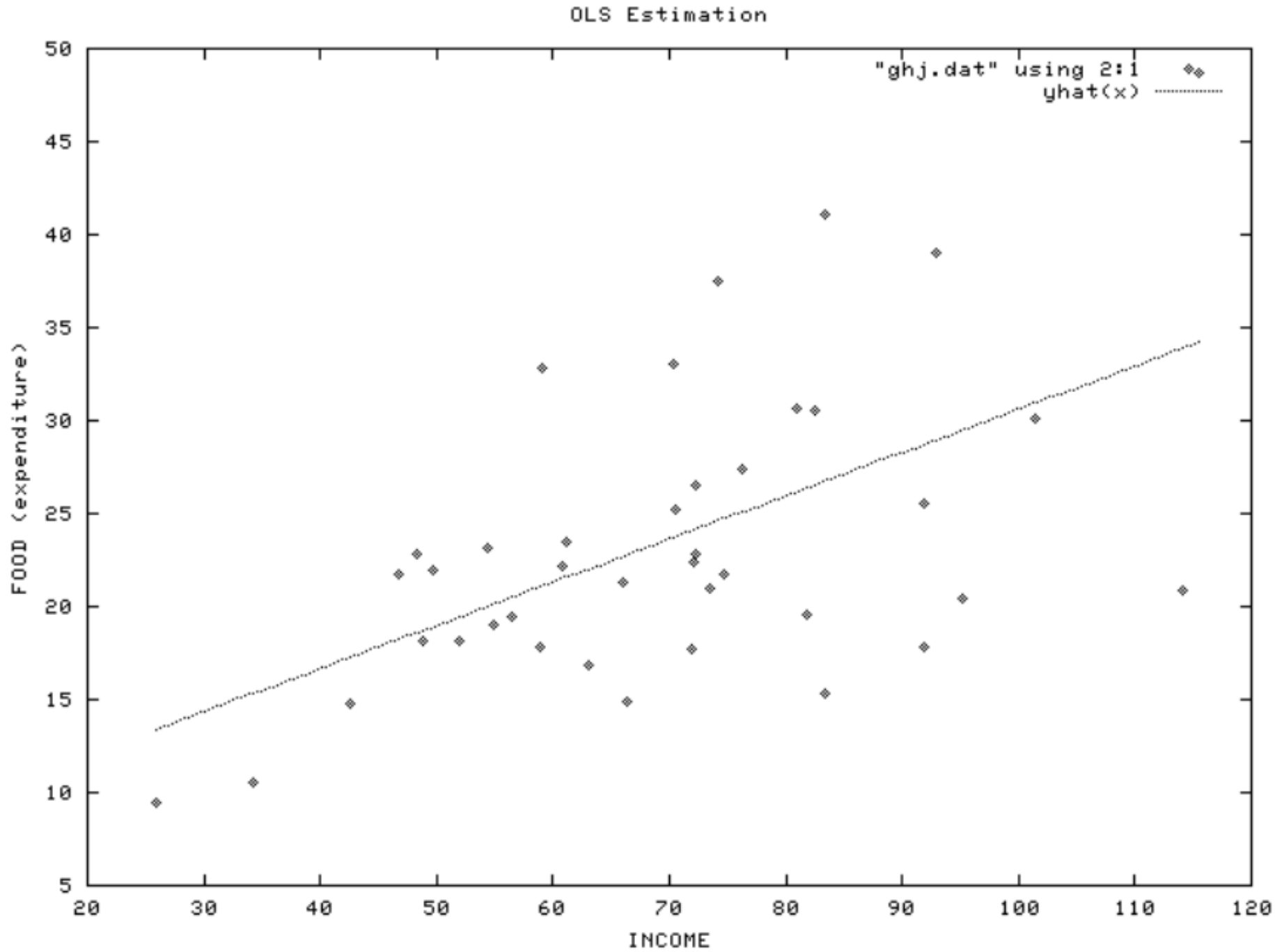
- y is the dependent (endogenous) variable that we want to predict and is stored as an $(n \times 1)$ vector
- $\boldsymbol{X}$ is an $(n \times k)$ matrix of k independent (exogeneous) variables
- ϵ is an error term
- We assume $E(\epsilon'\epsilon) = \sigma^2 I_n$ where I_n is an n dimensional identity matrix with 1's on the diagonal and 0 every where else. The zeros every where else imply ϵ_i's are independent.

Data

- The data describes the expenditures on food for a set of 40 household in a week. The data also includes the household income.

File name: expdata.inc

- Research question:
  - Do households with a higher income spend more money on food?


OLS Estimation
"ghj.dat" using 2:1
yhat(x)
FOOD (expenditure)
INCOME
50
45
40
35
30
25
20
15
10
5
20
30
40
50
60
70
80
90
100
110
120

Solve as a Quadratic Programming Problem

- Using the file ols1.gms
- We solve a quadratic programming problem

$$\min_{\epsilon} \sum_{i=1}^{n} \epsilon_i^2$$

$$s.t. \quad y_i = \alpha + \sum_{j=1}^{k} \beta_j x_{ij} + \epsilon_i \; \forall i = 1, \ldots, n$$

k is the number of independent variables, in the expenditure data $k = 1$; n is the number of observations, in the expenditure data $n = 40$

Solving via the Normal Equations

- Define the i^{th} residual to be

$$\epsilon_i = \quad y_i - \sum_{l=1}^{k} \beta_j x_{ij}$$

Then $\boldsymbol{S}$ can be rewritten

$$\boldsymbol{S} = \sum_{i=1}^{n} \epsilon_i^2$$

Given that $\boldsymbol{S}$ is convex, it is minimized when its gradient vector is zero (This follows by definition: if the gradient vector is not zero, there is a direction in which we can move to minimize it further) The elements of the gradient vector are the partial derivatives of $\boldsymbol{S}$ with respect to the parameters:

Solving via the Normal Equations

$$\frac{\partial \boldsymbol{S}}{\partial \beta_j} = 2\sum_{i=1}^{n} \epsilon_i \frac{\partial \epsilon_i}{\partial \beta_j} (j = 1,2,\dots,k)$$

The derivatives are

$$\frac{\partial \epsilon_i}{\partial \beta_j} = -x_{ij}$$

Substitution of the expressions for the residuals and the derivatives into the gradient equations gives

$$\frac{\partial \boldsymbol{S}}{\partial \beta_j} = 2\sum_{i=1}^{n}\left(\ y_i - \sum_{l=1}^{k} \beta_j x_{ij}\right)\left(-x_{ij}\right)(j = 1,2,\dots,k)$$

Thus if $\hat{\beta}$ minimizes $\boldsymbol{S}$, we have

Solving via the Normal Equations

$$2\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{k}\beta_j x_{ij}\right)\left(-x_{ij}\right) = 0 \quad (j = 1,2,\ldots,k)$$

Upon rearrangement, we obtain the **normal equations**:

$$\sum_{i=1}^{n}\sum_{l=1}^{k} x_{ij}x_{il}\widehat{\beta_j} = \sum_{i=1}^{n} x_{ij}y_i \; (j = 1,2,\ldots,k)$$

The normal equations are written in matrix notation as

$\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)\hat{\beta} = \mathbf{X}^{\mathrm{T}}\mathbf{y}$ (where $\mathbf{X}^{\mathrm{T}}$ is the matrix transpose of $\mathbf{X}$).

The solution of the normal equations yields the vector $\hat{\beta}$ of the optimal parameter values.

Solving via the Normal Equations

- Using the file ols2.gms
- We solve the normal equations

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X})\hat{\beta} = \mathbf{X}^{\mathrm{T}}\mathbf{y}$$

Since solving the normal equations does not require solving an optimization problem we need to create a “dummy” optimization problem and objective function to get GAMS to solve the normal equations.

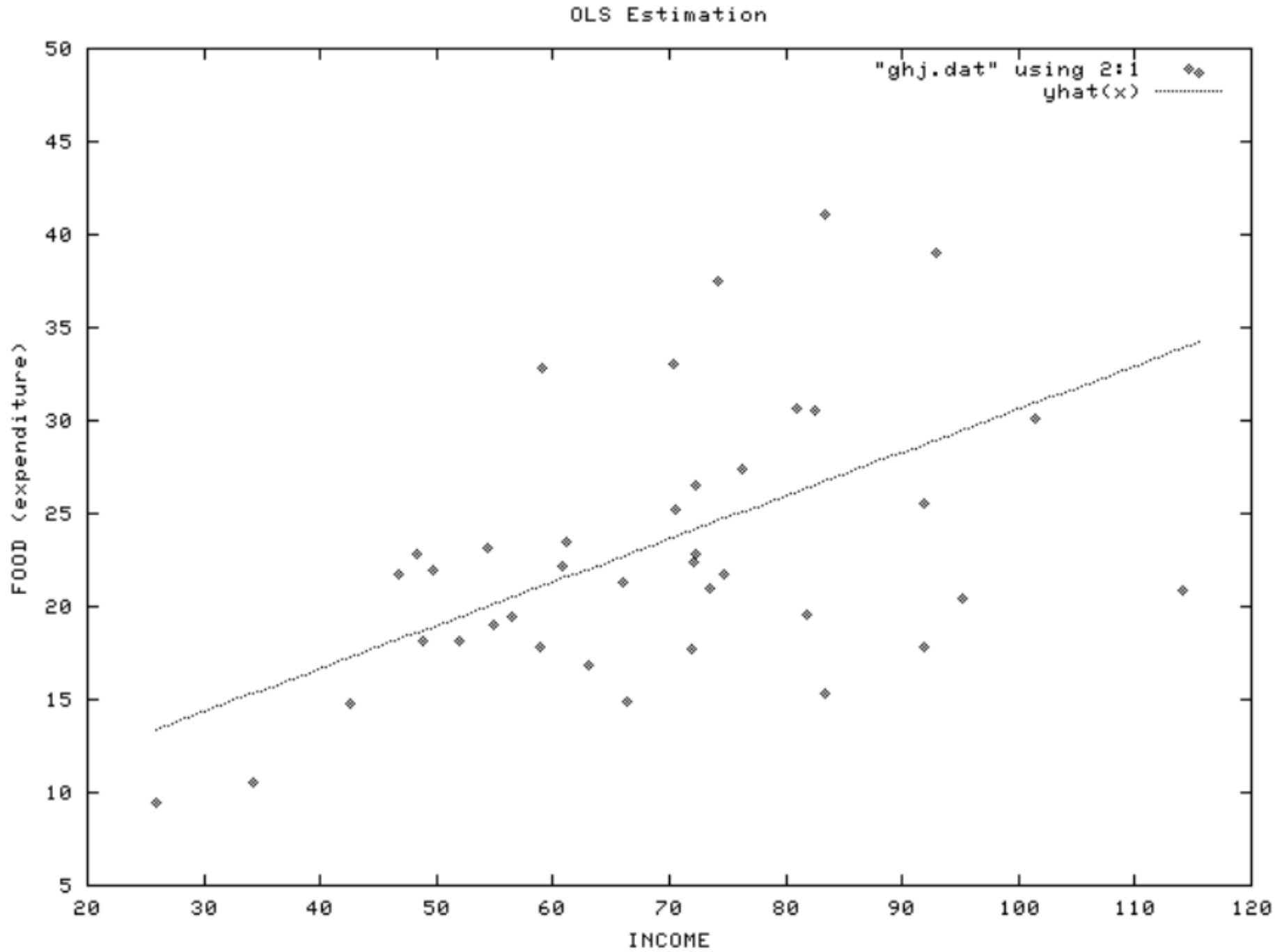
Solving via the Normal Equations

```
equations
    dummy_eq   'dummy objective equation'
    normal(j)  "normal equations (X'X)b = X'y"  ;

variables
    b(j)       'parameters to estimate'
    dummy_var  'dummy objective variable'   ;

dummy_eq..  dummy_var =e= b('constant');
normal(j).. sum(jj, XX(j,jj)*b(jj)) =e= Xy(j);

model ols2 /dummy_eq,normal/;
solve ols2 using lp minimizing dummy_var;
```


OLS Estimation
"ghj.dat" using 2:1
yhat(x)
FOOD (expenditure)
INCOME
50
45
40
35
30
25
20
15
10
5
20
30
40
50
60
70
80
90
100
110
120

Solving via the Normal Equations

$$(\mathbf{X}^{\mathrm{T}}\mathbf{X})\hat{\beta} = \mathbf{X}^{\mathrm{T}}\mathbf{y}$$

Notice there is no intercept term in the normal equations. To include an intercept term in the model, the matrix $\mathbf{X}$ has to be augmented by a vector of $\mathbf{1}$'s.

Notice

```
X(i,'constant') = 1;
X(i,'coeff1')   = data(i,'income');
```

Least Absolute Deviation

- For our linear model $y = \alpha + \boldsymbol{X}\beta + \epsilon$ we may want an estimator that is more robust to outliers. In this case Least Absolute Deviation (LAD) might be preferred

$$\min_{\epsilon} \sum_{i=1}^{n} |\epsilon_i|$$

$$s.t. \;\; y_i = \alpha + \sum_{j=1}^{k} \beta_j x_{ij} + \epsilon_i \; \forall i = 1, \ldots, n$$

Least Absolute Deviation

- Splitting technique: replace ϵ_i by $\epsilon_i^+ - \epsilon_i^-$ and $|\epsilon_i|$ by $\epsilon_i^+ + \epsilon_i^-$ where $\epsilon_i^+, \epsilon_i^- \geq 0$ are non-negative variables. We don't need to add the nonlinear constraint $\epsilon_i^+ \epsilon_i^- = 0$. Why?

$$\min_{\epsilon} \sum_{i=1}^{n} \epsilon_i^+ + \epsilon_i^-$$

$$s.t. \quad y_i = \alpha + \sum_{j=1}^{k} \beta_j x_{ij} + \epsilon_i^+ - \epsilon_i^- \quad \forall i = 1, \dots, n$$

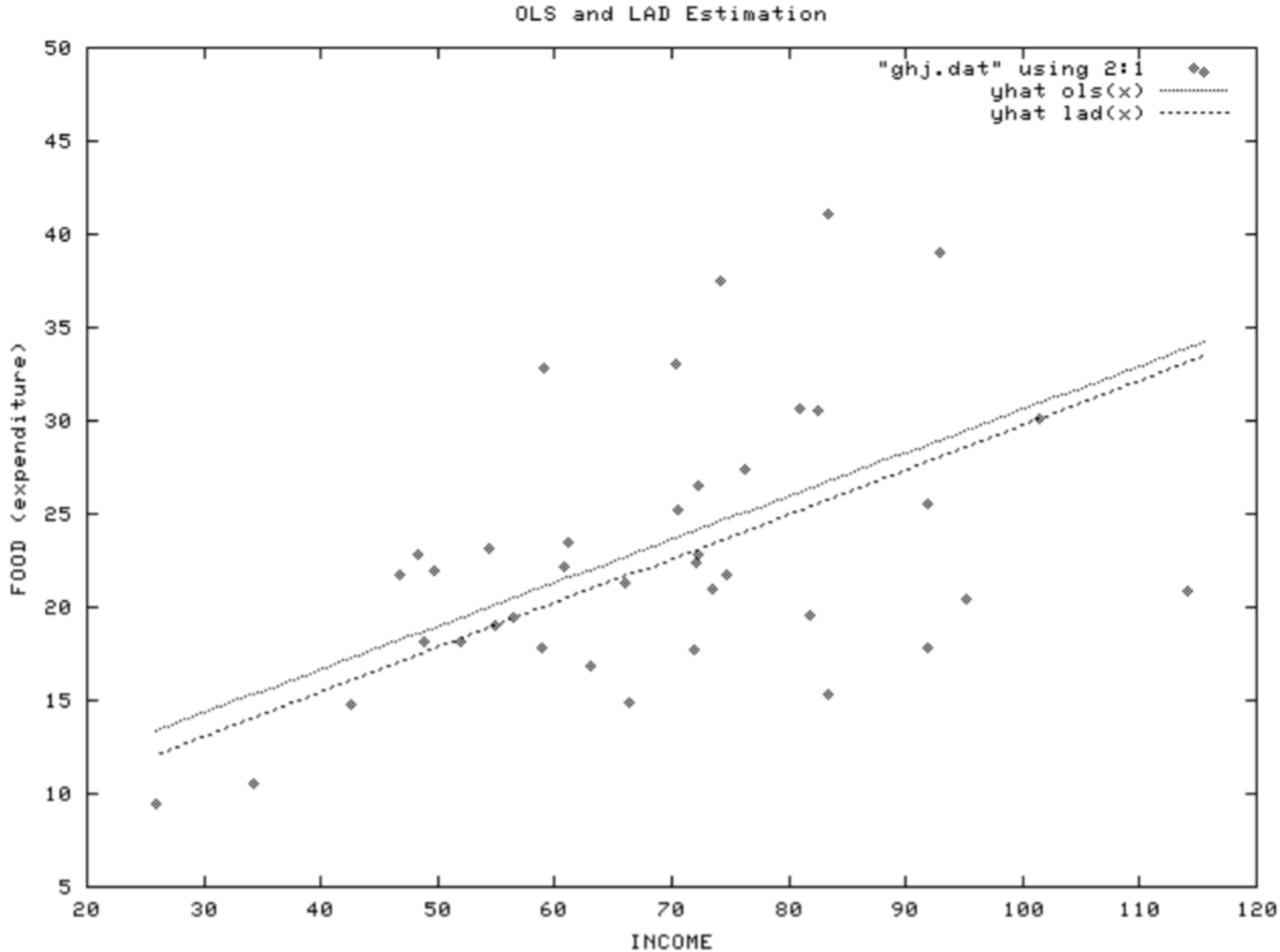
Least Absolute Deviation

$$\min_{\epsilon} \sum_{i=1}^{n} e_i$$

$$s.t. \quad y_i = \alpha + \sum_{j=1}^{k} \beta_j x_{ij} + \epsilon_i \quad \forall i = 1, \dots, n$$

$$e_i \geq \epsilon_i \; \forall i = 1, \dots, n$$

$$e_i \geq -\epsilon_i \; \forall i = 1, \dots, n$$


OLS and LAD Estimation
"ghj.dat" using 2:1
yhat ols(x)
yhat lad(x)
FOOD (expenditure)
INCOME
50
45
40
35
30
25
20
15
10
5
20
30
40
50
60
70
80
90
100
110
120

Duality

$$\min_{\boldsymbol{e},\boldsymbol{\beta}} \sum_{i=1}^{n} 1e_i + \sum_{j=1}^{k} 0\beta_j$$

$$s.t.$$

$$e_i + \sum_{j=1}^{k} \beta_j x_{ij} \geq \quad y_i \quad \forall i = 1, \dots, n$$

$$e_i - \sum_{j=1}^{k} \beta_j x_{ij} \geq -y_i \quad \forall i = 1, \dots, n$$

$\boldsymbol{e}, \boldsymbol{\beta}$ free variables

$$\max_{\boldsymbol{v},\boldsymbol{w}} \sum_{i=1}^{n} y_i v_i + \sum_{i=1}^{n} y_i w_i$$

$$s.t.$$

$$\sum_{i=1}^{n} x_i v_i + \sum_{i=1}^{n} x_i w_i = 0$$

$$\forall j = 1, \dots, k$$

$$v_i + w_i = 1 \quad \forall i = 1, \dots, n$$

$$\boldsymbol{v}, \boldsymbol{w} \geq \boldsymbol{0}$$

Least Absolute Deviation

$$\max_{\boldsymbol{v}} \sum_{i=1}^{n} y_i v_i$$

$$s.t. \sum_{i=1}^{n} x_i v_i = \frac{1}{2} \sum_{i=1}^{n} x_i \quad \forall j = 1, \dots, k$$

$$0 \leq v_i \leq 1 \quad \forall i = 1, \dots, n$$