# Model selection

Peter Hoff

STAT 423

Applied Regression and Analysis of Variance

University of Washington

# Diabetes example:

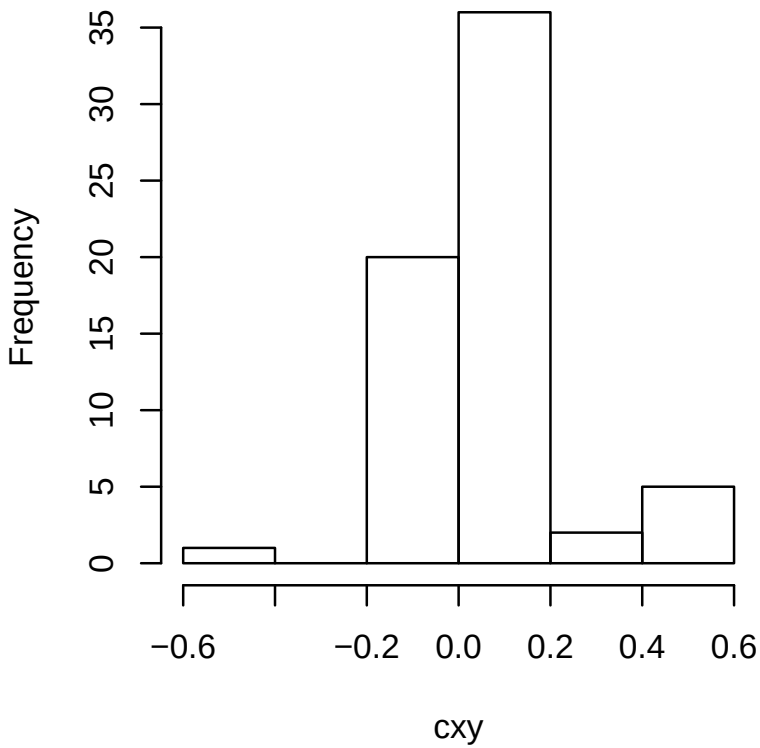
$$
\begin{aligned}
y &= \text{diabetes progression} \\
x_1 &= \text{age} \\
x_2 &= \text{sex} \\
&\vdots
\end{aligned}
$$

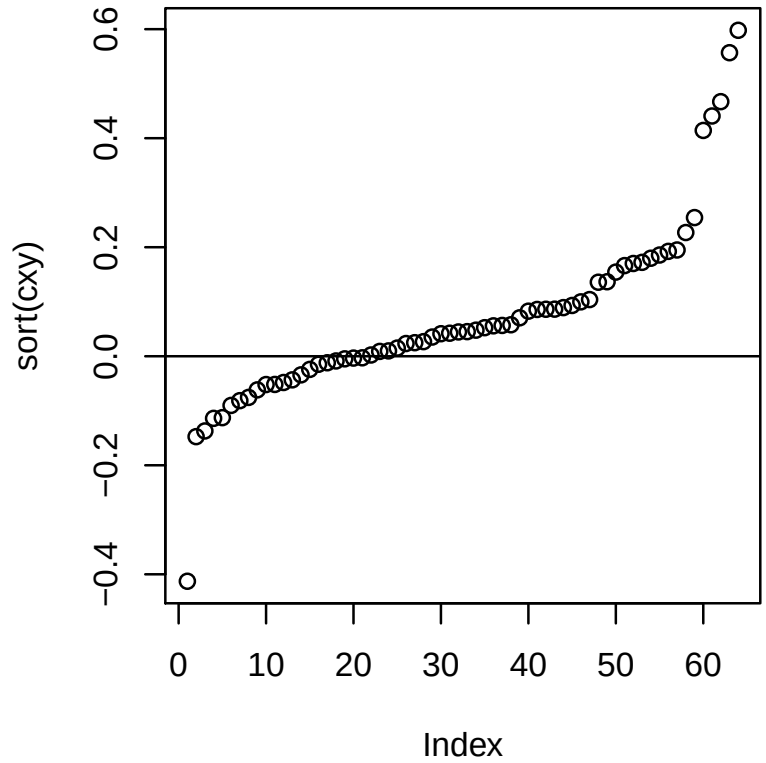
```
dim(X)

## [1] 442  64

colnames(X)

##  [1] "age"     "sex"     "bmi"     "map"     "tc"      "ldl"     "hdl"
##  [8] "tch"     "ltg"     "glu"     "age^2"   "bmi^2"   "map^2"   "tc^2"
## [15] "ldl^2"   "hdl^2"   "tch^2"   "ltg^2"   "glu^2"   "age:sex" "age:bmi"
## [22] "age:map" "age:tc"  "age:ldl" "age:hdl" "age:tch" "age:ltg" "age:glu"
## [29] "sex:bmi" "sex:map" "sex:tc"  "sex:ldl" "sex:hdl" "sex:tch" "sex:ltg"
## [36] "sex:glu" "bmi:map" "bmi:tc"  "bmi:ldl" "bmi:hdl" "bmi:tch" "bmi:ltg"
## [43] "bmi:glu" "map:tc"  "map:ldl" "map:hdl" "map:tch" "map:ltg" "map:glu"
## [50] "tc:ldl"  "tc:hdl"  "tc:tch"  "tc:ltg"  "tc:glu"  "ldl:hdl" "ldl:tch"
## [57] "ldl:ltg" "ldl:glu" "hdl:tch" "hdl:ltg" "hdl:glu" "tch:ltg" "tch:glu"
## [64] "ltg:glu"
```

# Training and test sets

```
nte<-100
ntr<-length(y)-nte

yte<-y[1:nte]
Xte<-X[1:nte,]


ytr<-y[ -(1:nte) ]
Xtr<-X[ -(1:nte),]
```

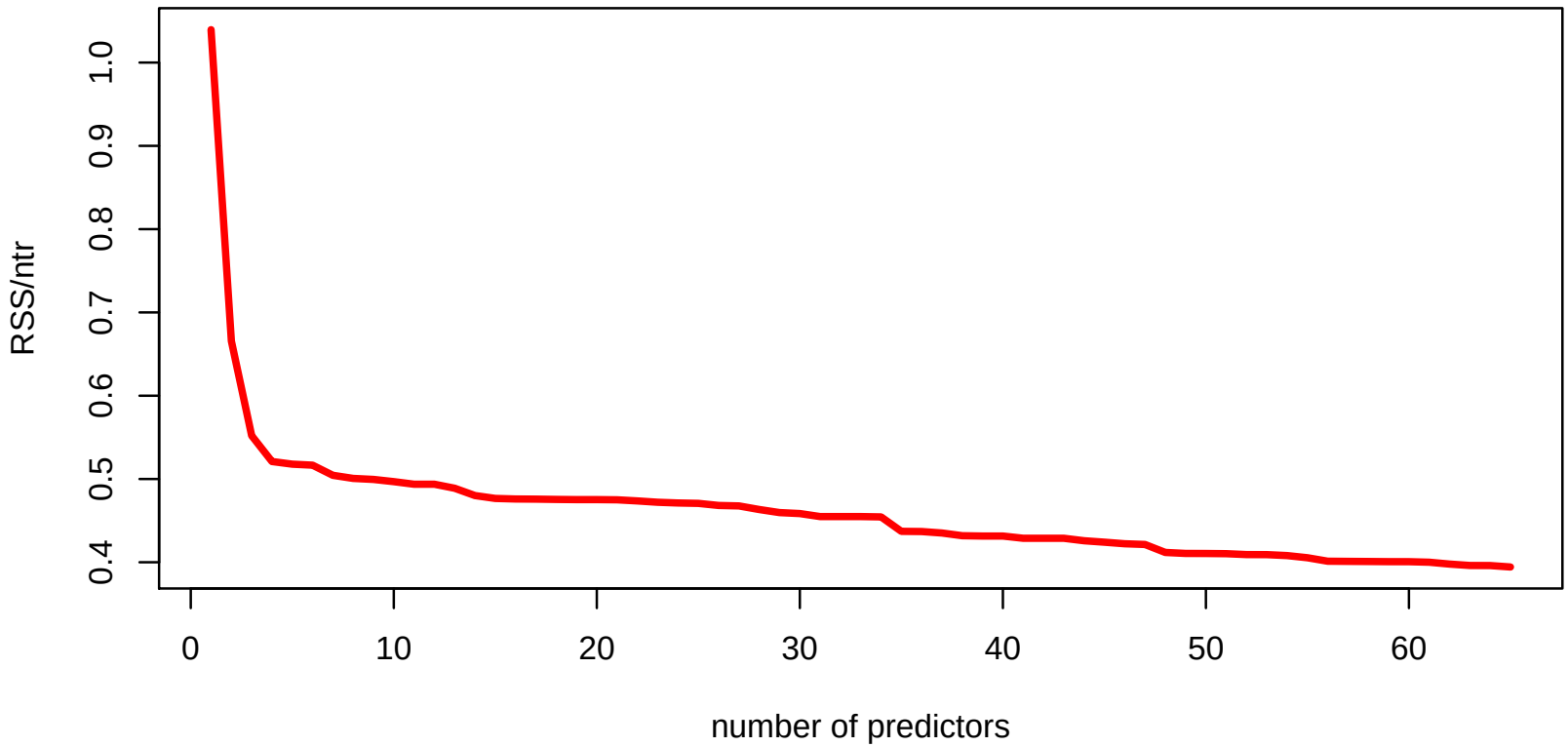
```r
cxy<-apply( Xtr, 2, function(x){ cor(x,ytr) } )
```



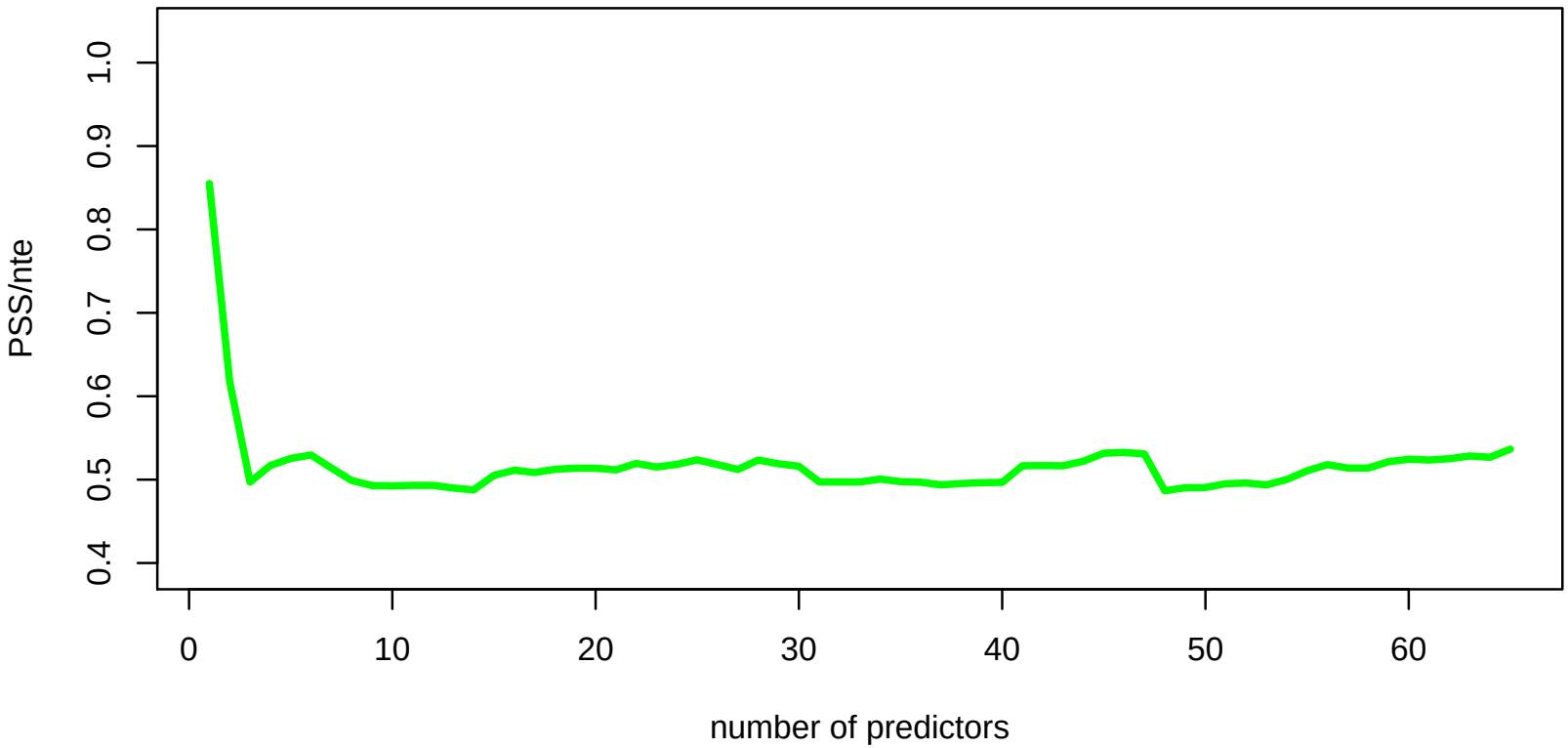



```
rcor<-order( abs(cxy), decreasing=TRUE)

RSS<-sum(ytr^2)
PSS<-sum(yte^2)

for(j in 1:ncol(Xtr))
{
  beta<-lm(ytr~ -1 + Xtr[,rcor[1:j]])$coef

  RSS<-c(RSS, sum( (ytr-Xtr[,rcor[1:j],drop=FALSE]%*%beta)^2 ) )

  PSS<-c(PSS, sum( (yte-Xte[,rcor[1:j],drop=FALSE]%*%beta)^2 ) )

}
```

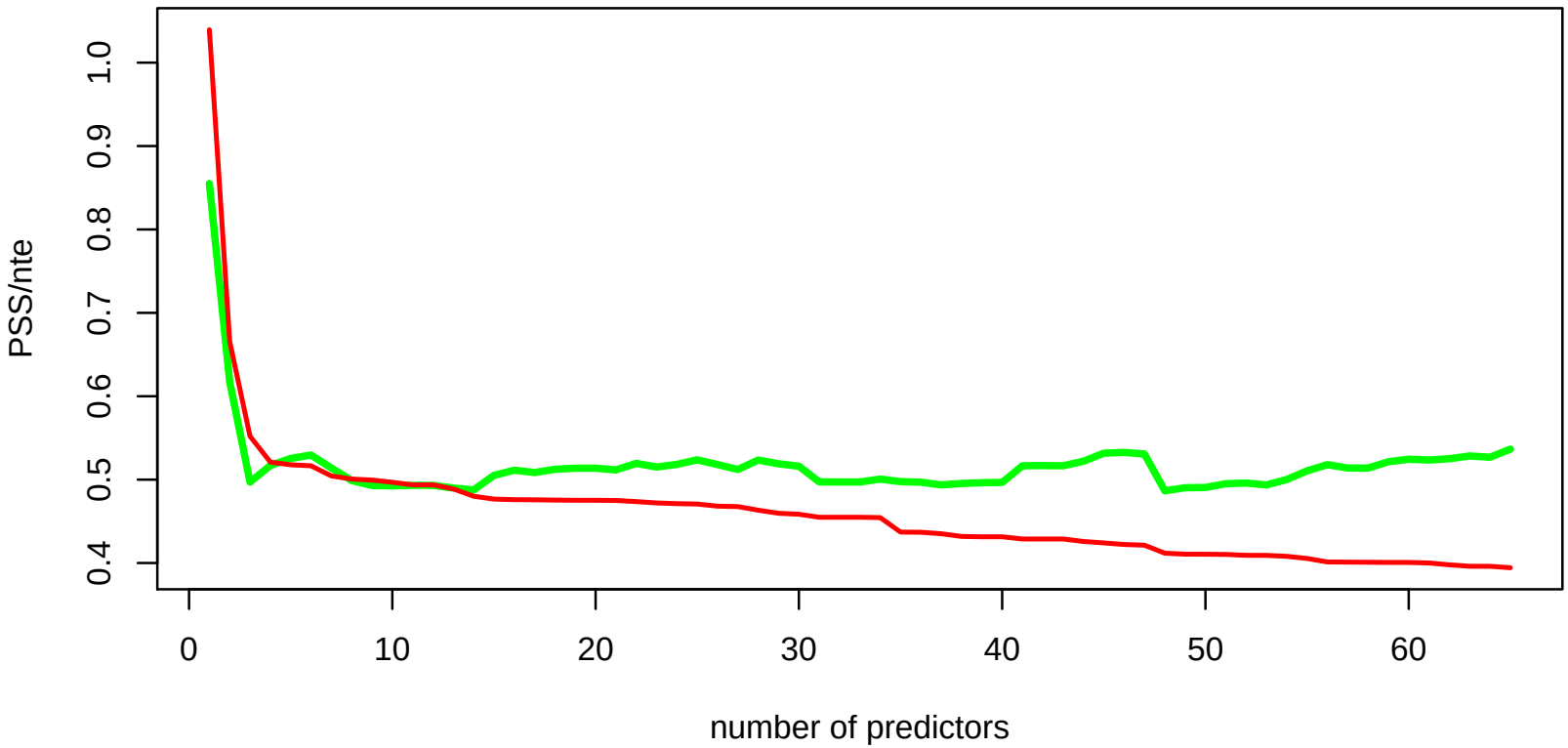
# RSS


RSS/ntr
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0
10
20
30
40
50
60
number of predictors

# PSS


PSS/nte
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0
10
20
30
40
50
60
number of predictors

# PSS


PSS/nte
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0
10
20
30
40
50
60
number of predictors

# Observations

RSS **decreases** with each additional predictor

PSS may **decrease** or **increase** with each additional predictor

RSS/ntr is a **good estimate** of PSS/nte when few predictors.

RSS/ntr is **not a good estimate** of PSS/nte when many predictors.

How should we choose which predictors to include?

# Observations

RSS **decreases** with each additional predictor

PSS may **decrease** or **increase** with each additional predictor

RSS/ntr is a **good estimate** of PSS/nte when few predictors.

RSS/ntr is **not a good estimate** of PSS/nte when many predictors.

How should we choose which predictors to include?

# Observations

RSS **decreases** with each additional predictor

PSS may **decrease** or **increase** with each additional predictor

RSS/ntr is a **good estimate** of PSS/nte when few predictors.

RSS/ntr is **not a good estimate** of PSS/nte when many predictors.

How should we choose which predictors to include?

# Observations

RSS **decreases** with each additional predictor

PSS may **decrease** or **increase** with each additional predictor

RSS/ntr is a **good estimate** of PSS/nte when few predictors.

RSS/ntr is **not a good estimate** of PSS/nte when many predictors.

How should we choose which predictors to include?

# Observations

RSS **decreases** with each additional predictor

PSS may **decrease** or **increase** with each additional predictor

RSS/ntr is a **good estimate** of PSS/nte when few predictors.

RSS/ntr is **not a good estimate** of PSS/nte when many predictors.

How should we choose which predictors to include?

# Regression models

The task of choosing predictors is called the **model selection problem**

In the context of linear regression, a **model** consists of

- an outcome variable $y$, or a outcome data vector $\mathbf{y}$;
- a set of explanatory variables $x_1, \ldots, x_p$, or a design matrix $\mathbf{X}$.

**Example:** "A regression model for $y$ with age, sex, bmi as predictors"

$\{y, age, sex, bmi\}$

$\mathbf{y}$, $\mathbf{X}_{[,1:3]}$

$y_i = \beta_0 + \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \beta_3 \times \text{bmi}_i + \epsilon_i$

**Example:** "A regression model for $y$ with all other vars as predictors"

$\{y, age, sex, \ldots, ltg : glu\}$

$\mathbf{y}$, $\mathbf{X}$

$y_i = \beta_0 + \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \cdots + \beta_{64} \times \text{ltg:glu}_i + \epsilon_i$

# Regression models

The task of choosing predictors is called the **model selection problem**

In the context of linear regression, a **model** consists of

- an outcome variable $y$, or a outcome data vector $\mathbf{y}$;
- a set of explanatory variables $x_1, \ldots, x_p$, or a design matrix $\mathbf{X}$.

**Example:** "A regression model for $y$ with age, sex, bmi as predictors"

$\{y, age, sex, bmi\}$

$\mathbf{y}$, $\mathbf{X}_{[,1:3]}$

$y_i = \beta_0 + \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \beta_3 \times \text{bmi}_i + \epsilon_i$

**Example:** "A regression model for $y$ with all other vars as predictors"

$\{y, age, sex, \ldots, ltg : glu\}$

$\mathbf{y}$, $\mathbf{X}$

$y_i = \beta_0 + \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \cdots + \beta_{64} \times \text{ltg:glu}_i + \epsilon_i$

# Regression models

The task of choosing predictors is called the **model selection problem**

In the context of linear regression, a **model** consists of

- an outcome variable $y$, or a outcome data vector $\mathbf{y}$;
- a set of explanatory variables $x_1, \ldots, x_p$, or a design matrix $\mathbf{X}$.

**Example:** "A regression model for $y$ with age, sex, bmi as predictors"

$\{y, age, sex, bmi\}$

$\mathbf{y}$, $\mathbf{X}_{[,1:3]}$

$y_i = \beta_0 + \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \beta_3 \times \text{bmi}_i + \epsilon_i$

**Example:** "A regression model for $y$ with all other vars as predictors"

$\{y, age, sex, \ldots, ltg : glu\}$

$\mathbf{y}$, $\mathbf{X}$

$y_i = \beta_0 + \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \cdots + \beta_{64} \times \text{ltg:glu}_i + \epsilon_i$

# Regression models

The task of choosing predictors is called the **model selection problem**

In the context of linear regression, a **model** consists of

- an outcome variable $y$, or a outcome data vector $\mathbf{y}$;
- a set of explanatory variables $x_1, \ldots, x_p$, or a design matrix $\mathbf{X}$.

**Example:** "A regression model for $y$ with age, sex, bmi as predictors"

$\{y, age, sex, bmi\}$

$\mathbf{y}$, $\mathbf{X}_{[,1:3]}$

$y_i = \beta_0 + \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \beta_3 \times \text{bmi}_i + \epsilon_i$

**Example:** "A regression model for $y$ with all other vars as predictors"

$\{y, age, sex, \ldots, ltg : glu\}$

$\mathbf{y}$, $\mathbf{X}$

$y_i = \beta_0 + \beta_1 \times \text{age}_i + \beta_2 \times \text{sex}_i + \cdots + \beta_{64} \times \text{ltg:glu}_i + \epsilon_i$

# Model comparison

Given a set of models, how should we compare them?

- *RSS*?
- *p*-values?
- *PSS* using test and training sets?
- cross validation?
- other criteria?

# Model comparison

Which models should we compare?

**Q:** Given $x_1, \ldots, x_p$, how many main-effects models are there?

**A:**

- Each variable may either be in or out of the model;
- There are $2^p$ models to consider.

For the `diabetes` data, $p = 64$ and so the number of models is

$$2^{64} \approx 1.8 \times 10^{19}.$$

We won't be able to fit and compare all possible models if $p$ is large.

# Model comparison

Which models should we compare?

**Q:** Given $x_1, \ldots, x_p$, how many main-effects models are there?

**A:**

- Each variable may either be in or out of the model;
- There are $2^p$ models to consider.

For the `diabetes` data, $p = 64$ and so the number of models is

$$2^{64} \approx 1.8 \times 10^{19}.$$

We won't be able to fit and compare all possible models if $p$ is large.

# Model comparison

Which models should we compare?

**Q:** Given $x_1, \ldots, x_p$, how many main-effects models are there?

**A:**

- Each variable may either be in or out of the model;
- There are $2^p$ models to consider.

For the `diabetes` data, $p = 64$ and so the number of models is

$$2^{64} \approx 1.8 \times 10^{19}.$$

We won't be able to fit and compare all possible models if $p$ is large.

# Model comparison

Which models should we compare?

**Q:** Given $x_1, \ldots, x_p$, how many main-effects models are there?

**A:**

- Each variable may either be in or out of the model;
- There are $2^p$ models to consider.

For the `diabetes` data, $p = 64$ and so the number of models is

$$2^{64} \approx 1.8 \times 10^{19}.$$

We won't be able to fit and compare all possible models if $p$ is large.

# Model comparison

Which models should we compare?

**Q:** Given $x_1, \ldots, x_p$, how many main-effects models are there?

**A:**

- Each variable may either be in or out of the model;
- There are $2^p$ models to consider.

For the `diabetes` data, $p = 64$ and so the number of models is

$$2^{64} \approx 1.8 \times 10^{19}.$$

We won't be able to fit and compare all possible models if $p$ is large.

# Model comparison

Which models should we compare?

**Q:** Given $x_1, \ldots, x_p$, how many main-effects models are there?

**A:**

- Each variable may either be in or out of the model;
- There are $2^p$ models to consider.

For the `diabetes` data, $p = 64$ and so the number of models is

$$2^{64} \approx 1.8 \times 10^{19}.$$

We won't be able to fit and compare all possible models if $p$ is large.

# Model selection

Selecting a model requires two things:

- A procedure for deciding which models to compare;
- A criteria with which to compare models.

If $p$ is small, we may be able to compare all possible models.

If $p$ is large, we may use a stepwise procedure:

- add or remove predictors from a model based on comparison criteria;
- this "searches" through the space of models, making moves that improve criteria.

# Model selection

Selecting a model requires two things:

- A procedure for deciding which models to compare;
- A criteria with which to compare models.

If $p$ is small, we may be able to compare all possible models.

If $p$ is large, we may use a stepwise procedure:

- add or remove predictors from a model based on comparison criteria;
- this "searches" through the space of models, making moves that improve criteria.

# Model selection

Selecting a model requires two things:

- A procedure for deciding which models to compare;
- A criteria with which to compare models.

If $p$ is small, we may be able to compare all possible models.

If $p$ is large, we may use a stepwise procedure:

- add or remove predictors from a model based on comparison criteria;
- this "searches" through the space of models, making moves that improve criteria.

# PSS

$$\mathrm{E}[\mathrm{PSS}] = n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\mu||^2$$

- $p$ represents estimation variability;
- $||(\mathbf{I} - \mathbf{P})\mu||^2$ represents bias.

Generally speaking, as $p$ goes up

- estimation variability goes up;
- bias goes down.

# PSS

$$\mathrm{E}[\mathrm{PSS}] = n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\mu||^2$$

- $p$ represents estimation variability;
- $||(\mathbf{I} - \mathbf{P})\mu||^2$ represents bias.

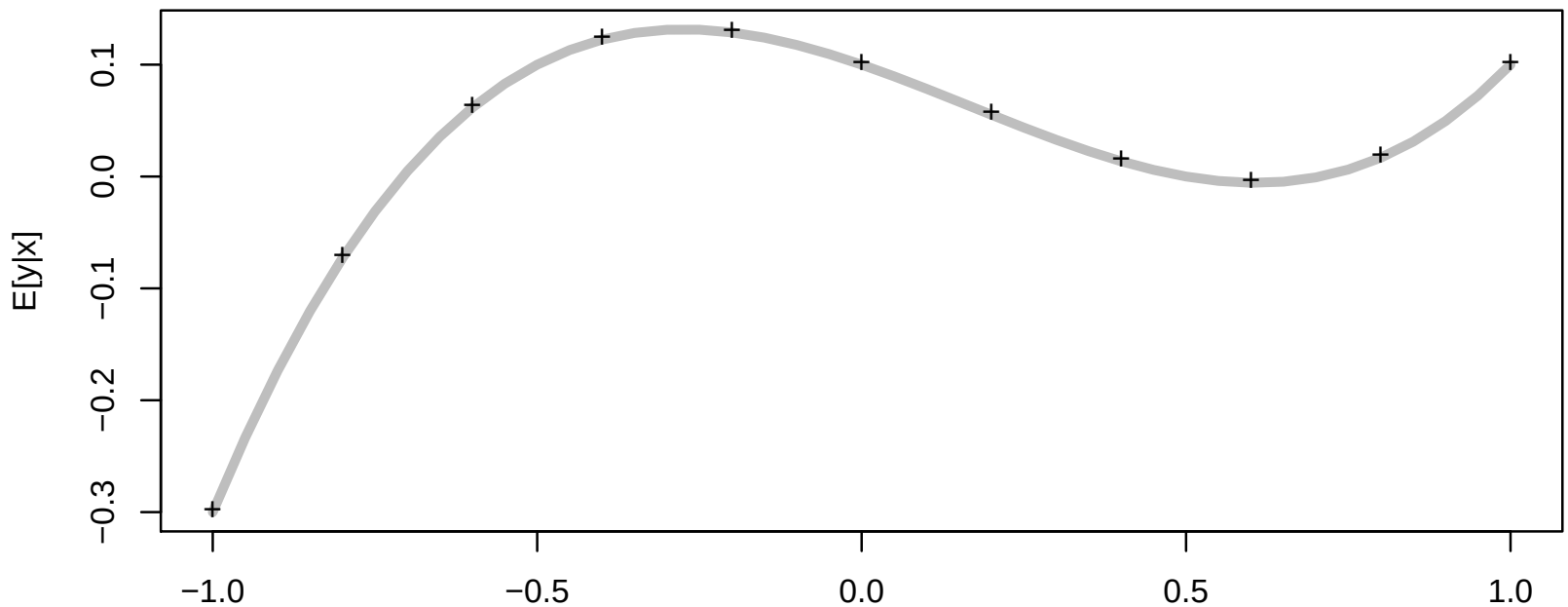
Generally speaking, as $p$ goes up

- estimation variability goes up;
- bias goes down.

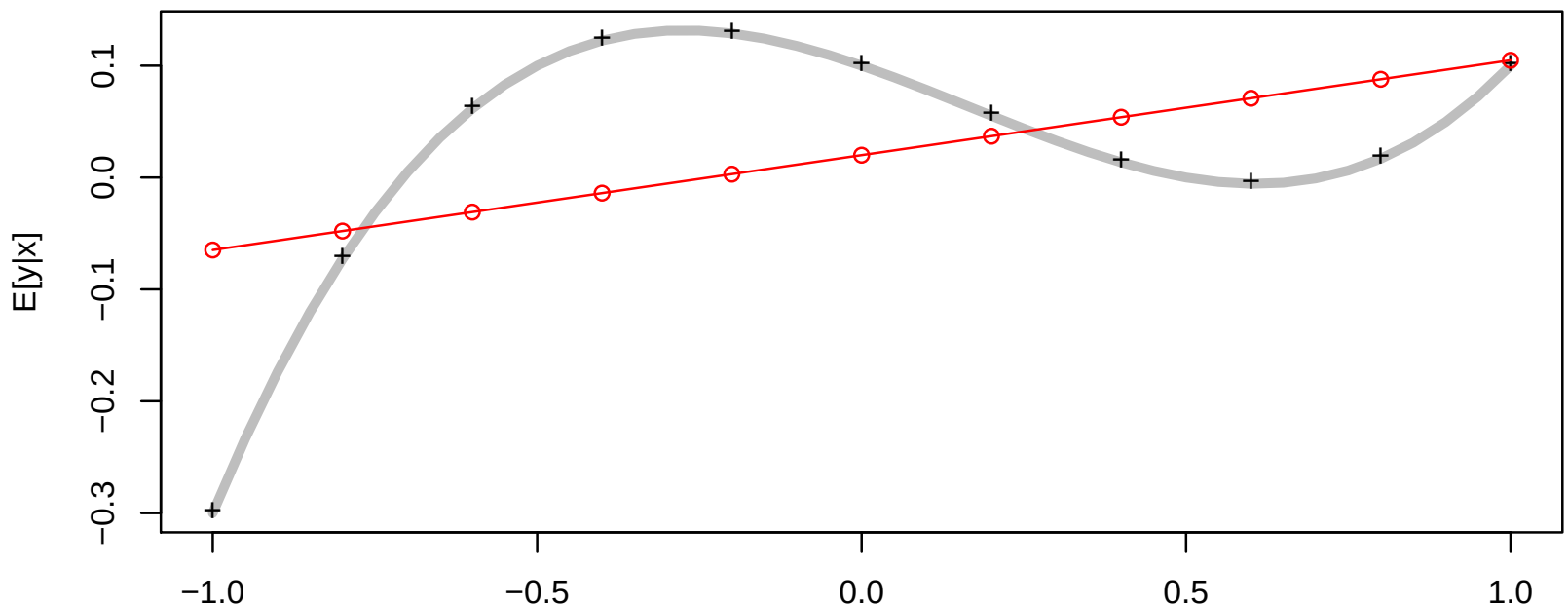
# Illustration: polynomial regression

```
x<-seq(-1,1,by=.2 )

X<-cbind(1 , x, x^2, x^3, x^4, x^5)

beta<-c(.1,-.2,-.2,.4,0,0)

mu<-X%*%beta
```

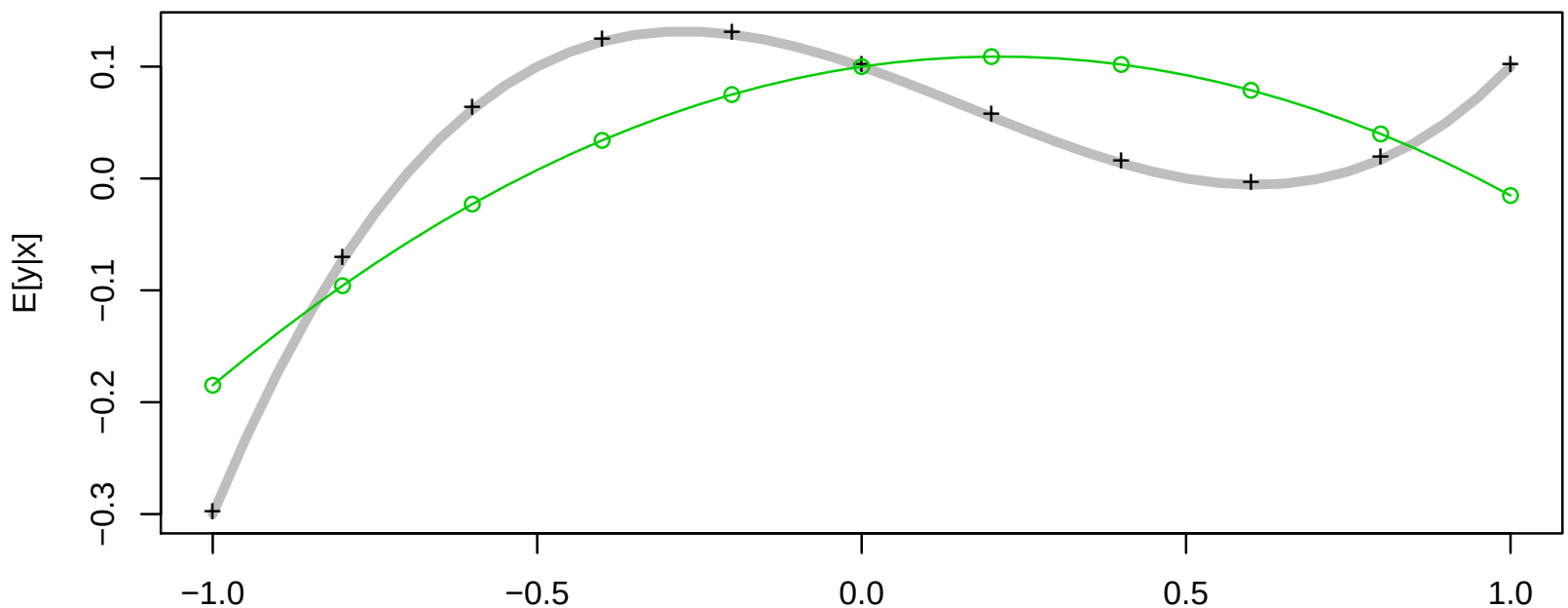
The true mean function is a 3rd degree polynomial in $x$.



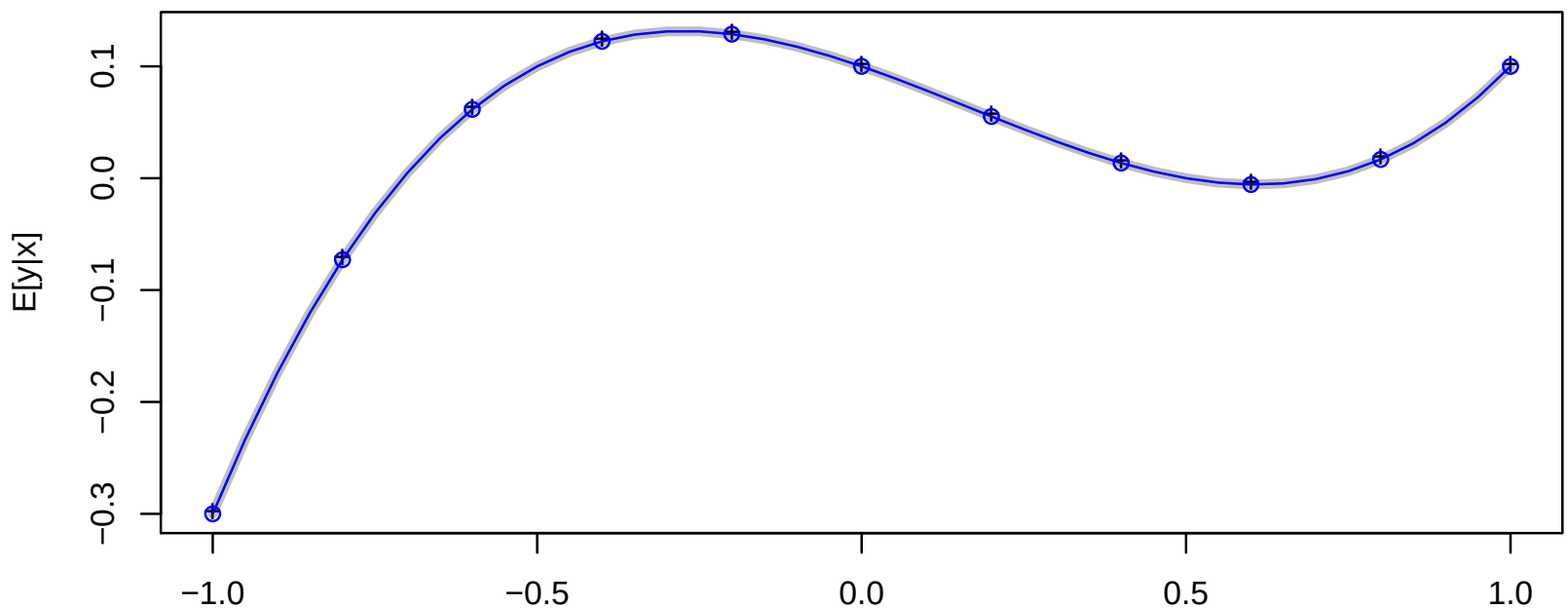
```r
X1<-X[,1:2]

P1<-X1%*%solve(t(X1)%*%X1)%*%t(X1)

c(P1%*% mu)

##  [1] -0.06480 -0.04784 -0.03088 -0.01392  0.00304  0.02000  0.03696
##  [8]  0.05392  0.07088  0.08784  0.10480

c( (I-P1)%*%mu )

##  [1] -0.23520 -0.02496  0.09248  0.13632  0.12576  0.08000  0.01824
##  [8] -0.04032 -0.07648 -0.07104 -0.00480
```



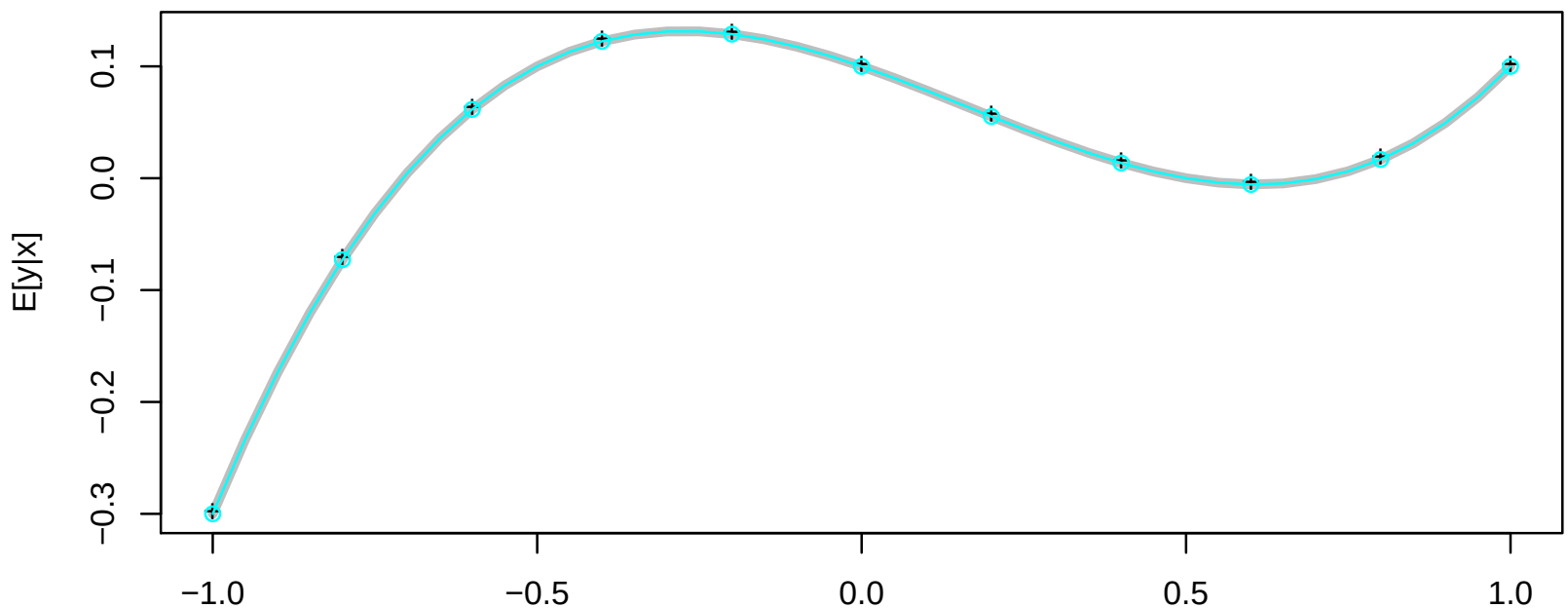
```r
X2<-X[,1:3]

P2<-X2%*%solve(t(X2)%*%X2)%*%t(X2)

c(P2%*% mu)

##  [1] -0.18480 -0.09584 -0.02288  0.03408  0.07504  0.10000  0.10896
##  [8]  0.10192  0.07888  0.03984 -0.01520

round( c( (I-P2)%*%mu ), 5)

##  [1] -0.11520  0.02304  0.08448  0.08832  0.05376  0.00000 -0.05376
##  [8] -0.08832 -0.08448 -0.02304  0.11520
```



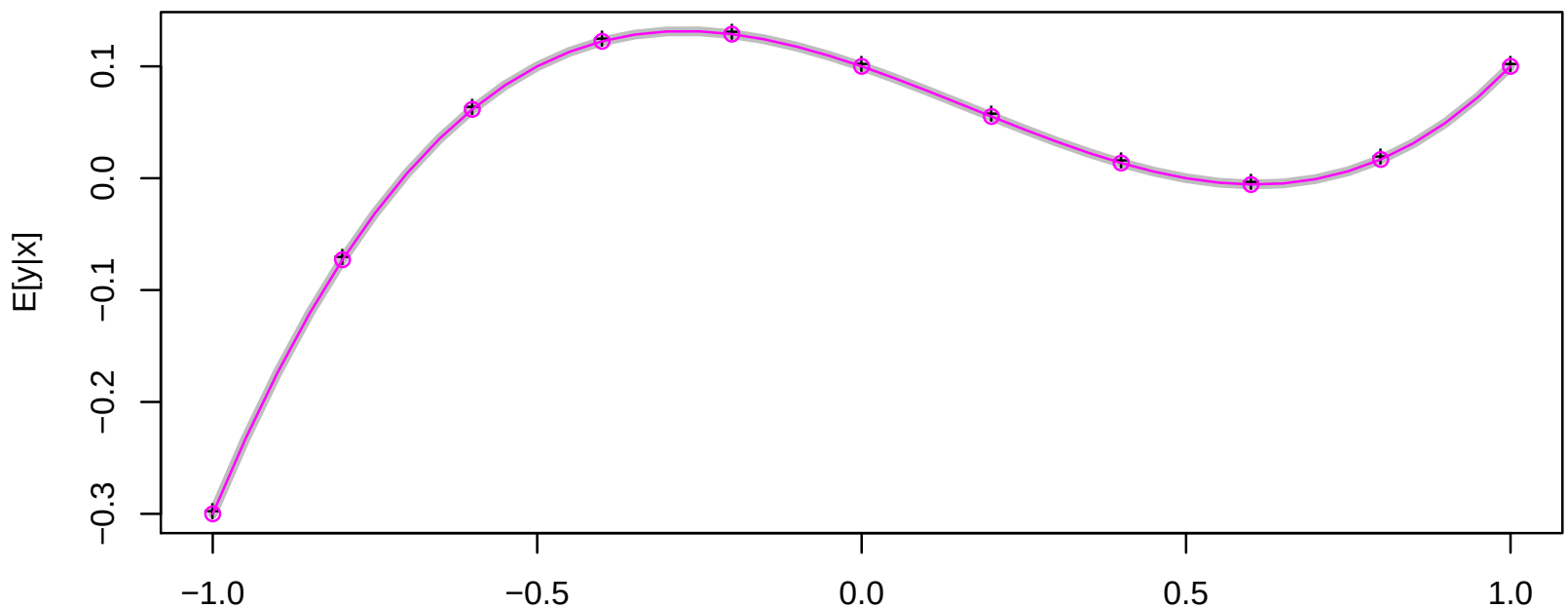
```r
X3<-X[,1:4]

P3<-X3%*%solve(t(X3)%*%X3)%*%t(X3)

c(P3%*% mu)

##  [1] -0.3000 -0.0728  0.0616  0.1224  0.1288  0.1000  0.0552  0.0136
##  [9] -0.0056  0.0168  0.1000

round(c( (I-P3)%*%mu ),5)

##  [1] 0 0 0 0 0 0 0 0 0 0 0
```

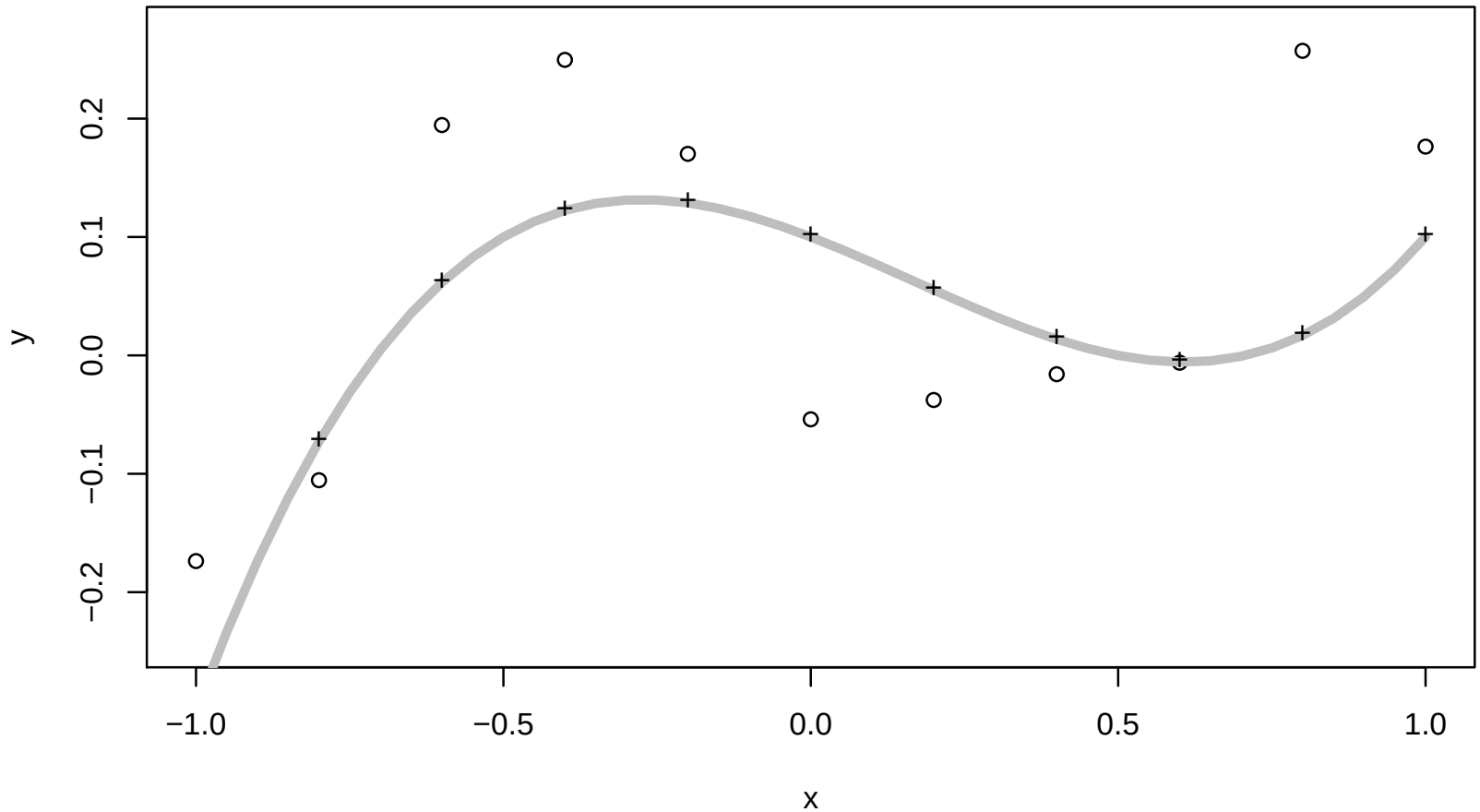


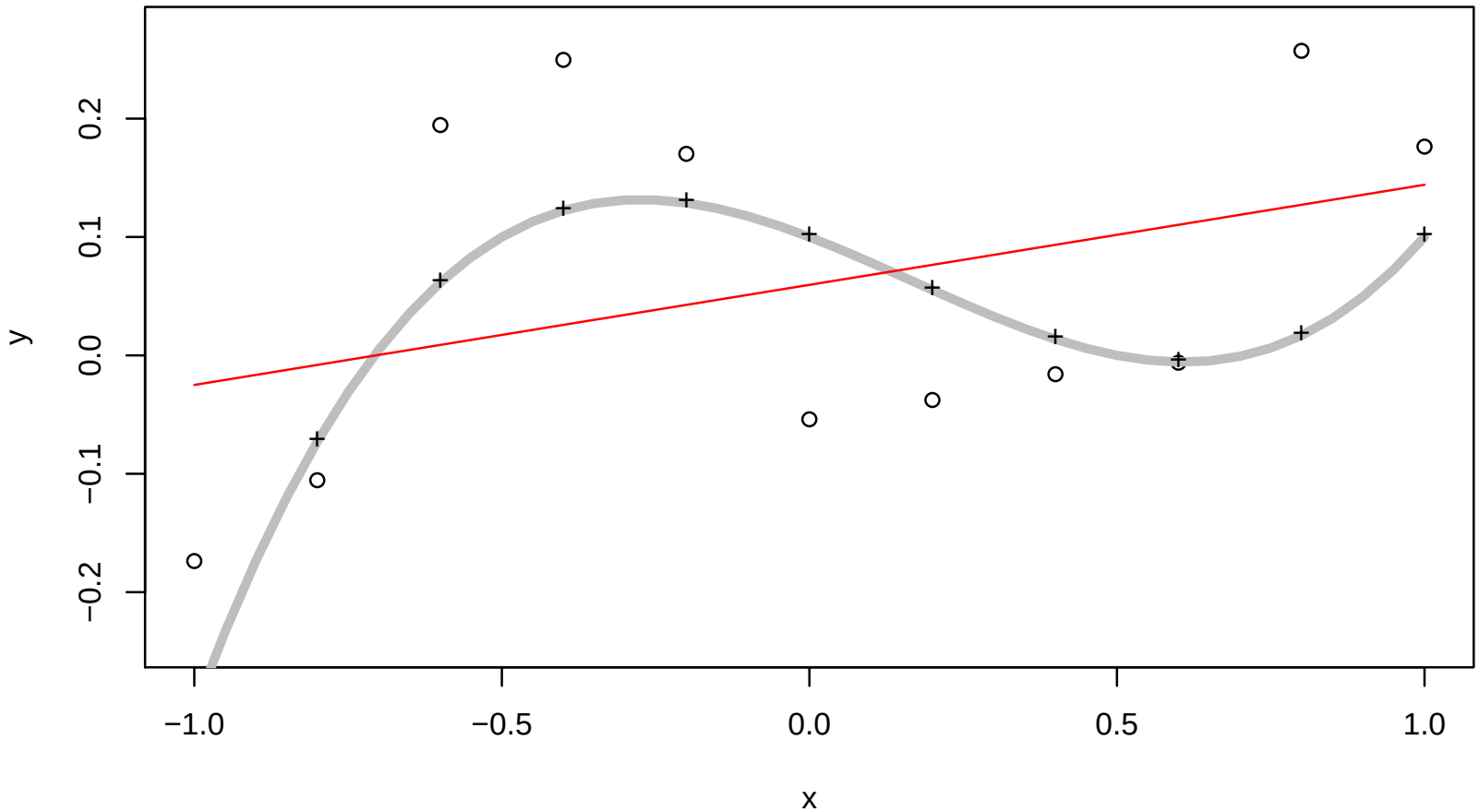
```r
X4<-X[,1:5]

P4<-X4%*%solve(t(X4)%*%X4)%*%t(X4)

c(P4%*% mu)

##  [1] -0.3000 -0.0728  0.0616  0.1224  0.1288  0.1000  0.0552  0.0136
##  [9] -0.0056  0.0168  0.1000

round( c( (I-P4)%*%mu ), 5)

##  [1] 0 0 0 0 0 0 0 0 0 0 0
```

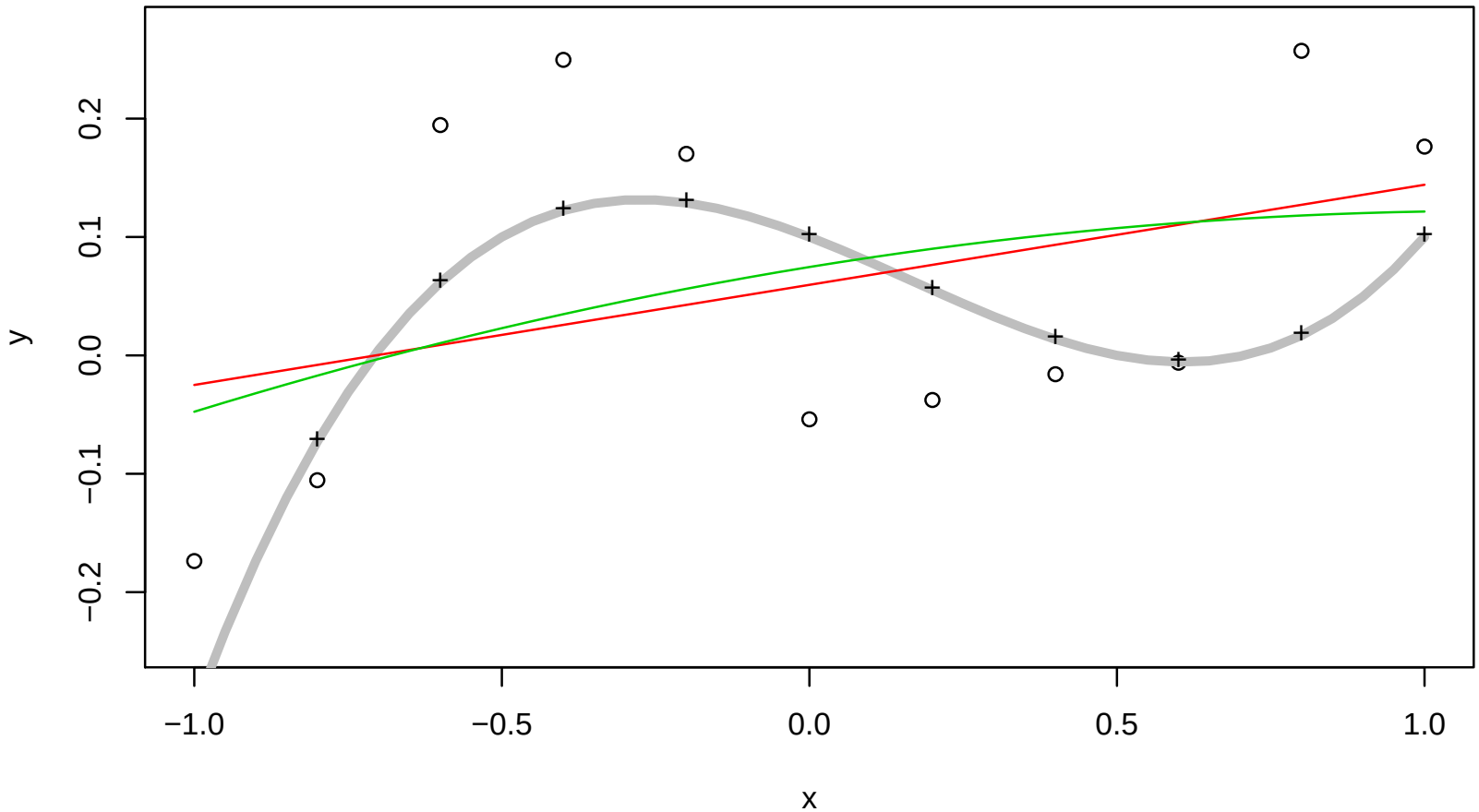


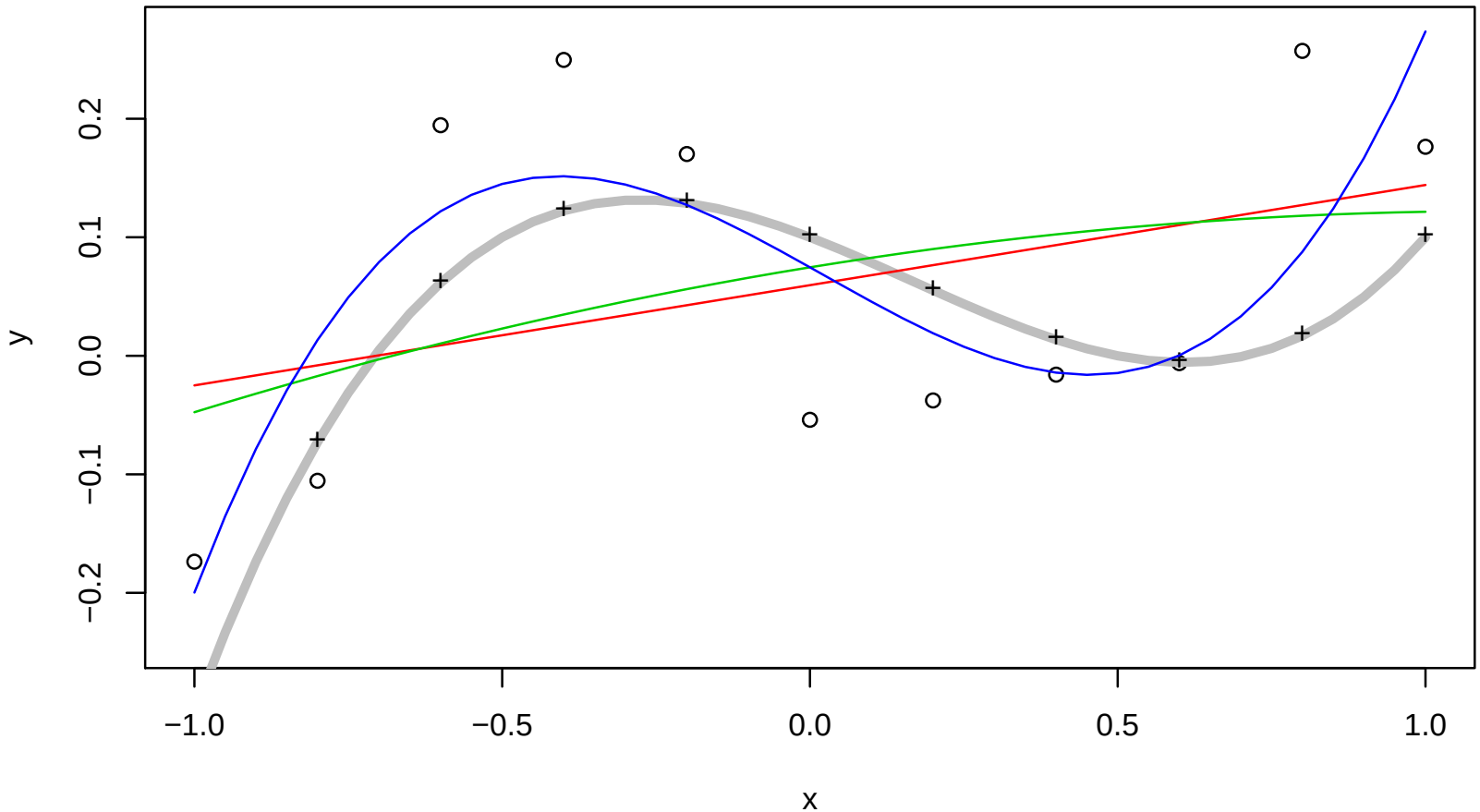
```r
X5<-X[,1:6]

P5<-X5%*%solve(t(X5)%*%X5)%*%t(X5)

c(P5%*% mu)

##  [1] -0.3000 -0.0728  0.0616  0.1224  0.1288  0.1000  0.0552  0.0136
##  [9] -0.0056  0.0168  0.1000

round( c( (I-P5)%*%mu ), 5)

##  [1] 0 0 0 0 0 0 0 0 0 0 0
```

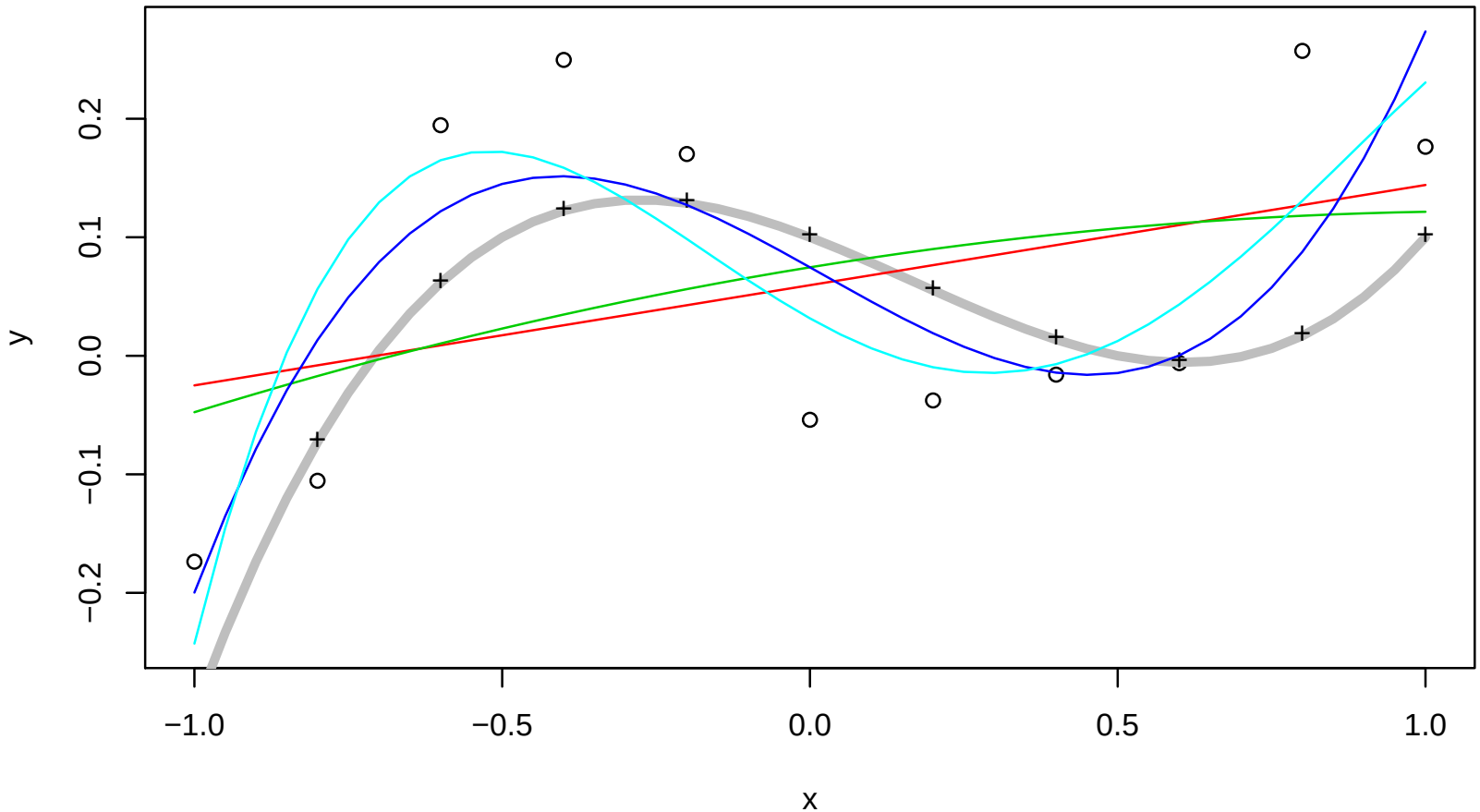


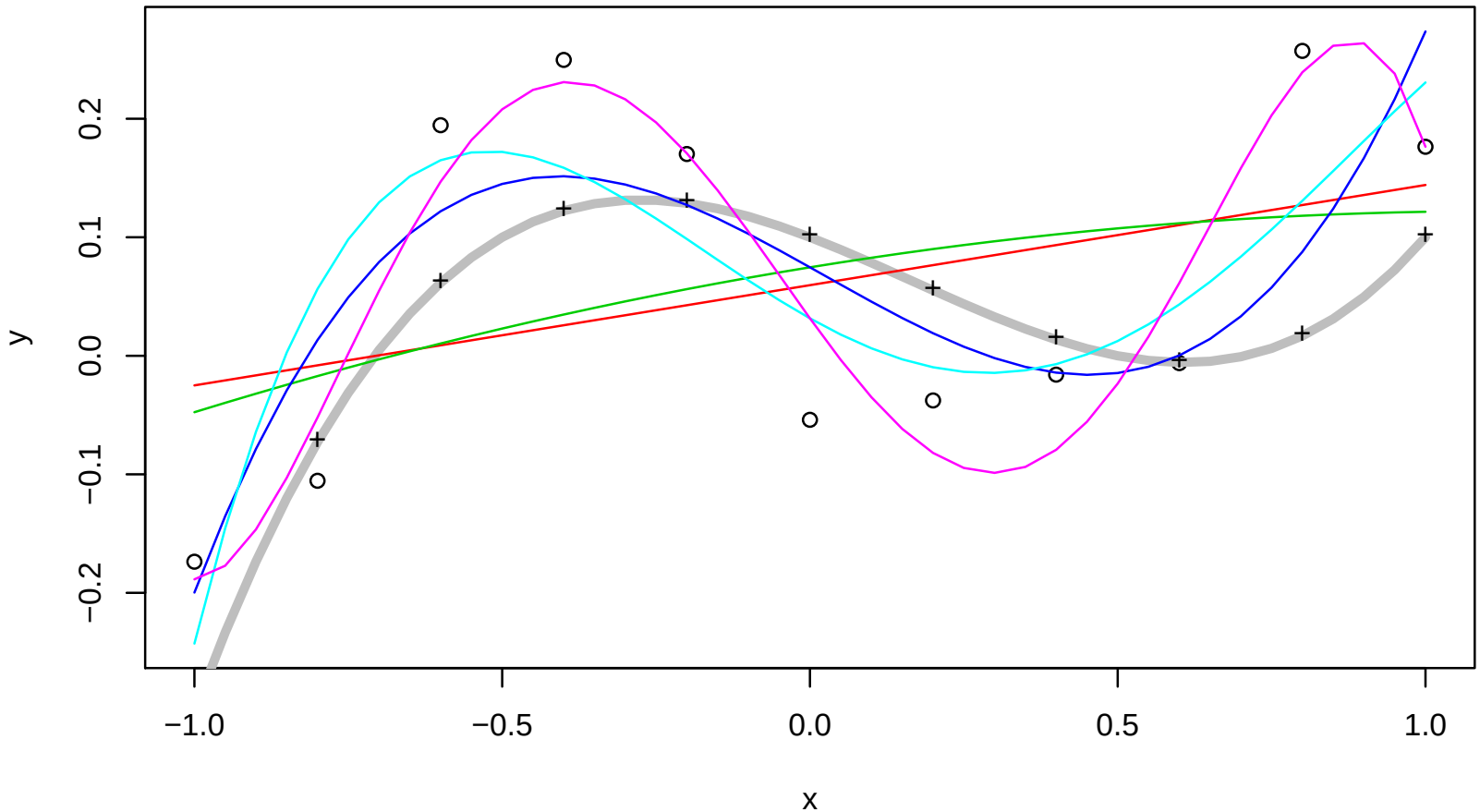

y
0.2
0.1
0.0
−0.1
−0.2
−1.0
−0.5
0.0
0.5
1.0
x


y
0.2
0.1
0.0
−0.1
−0.2
−1.0
−0.5
0.0
0.5
1.0
x


y
0.2
0.1
0.0
−0.1
−0.2
−1.0
−0.5
0.0
0.5
1.0
x

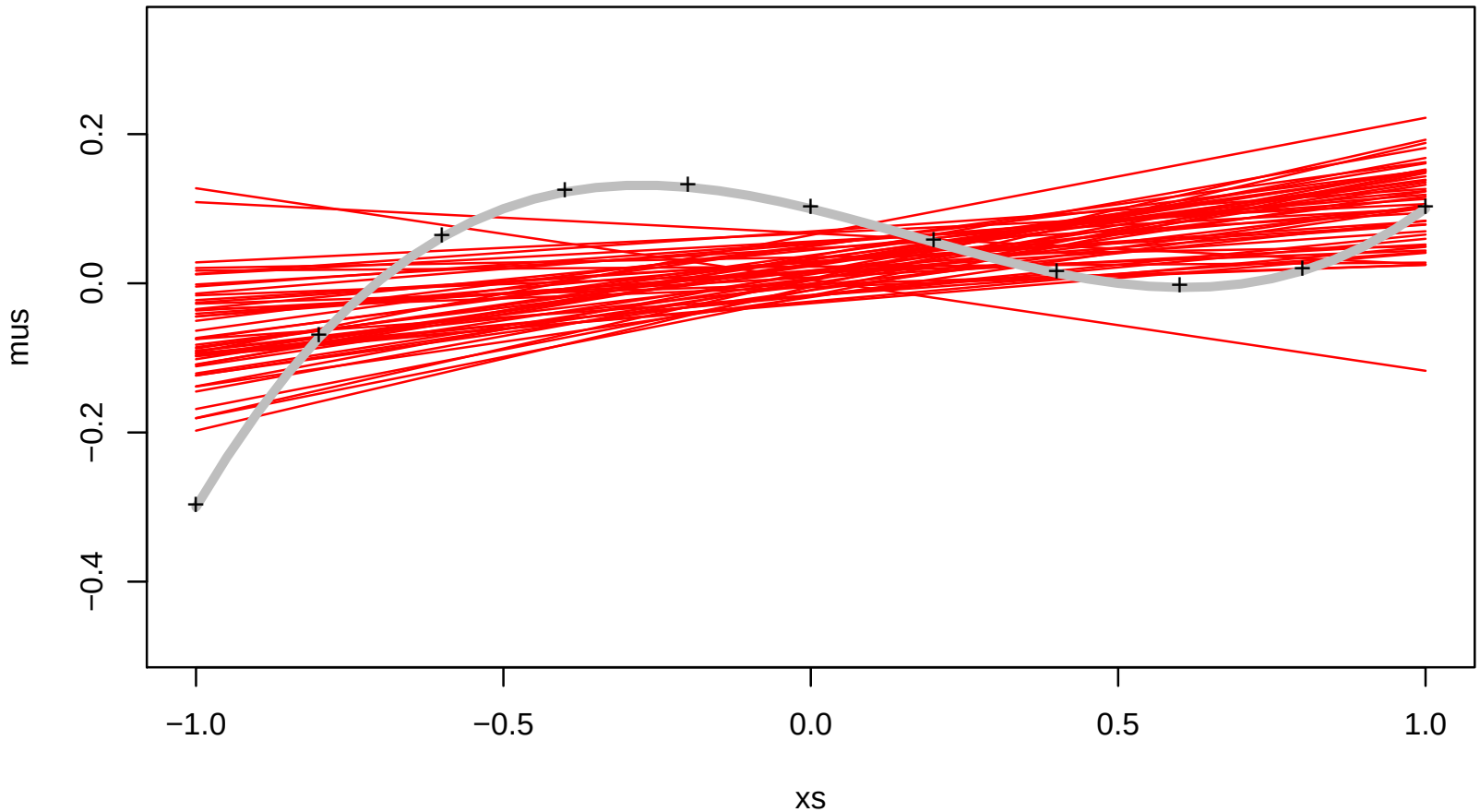

y
0.2
0.1
0.0
−0.1
−0.2
−1.0
−0.5
0.0
0.5
1.0
x

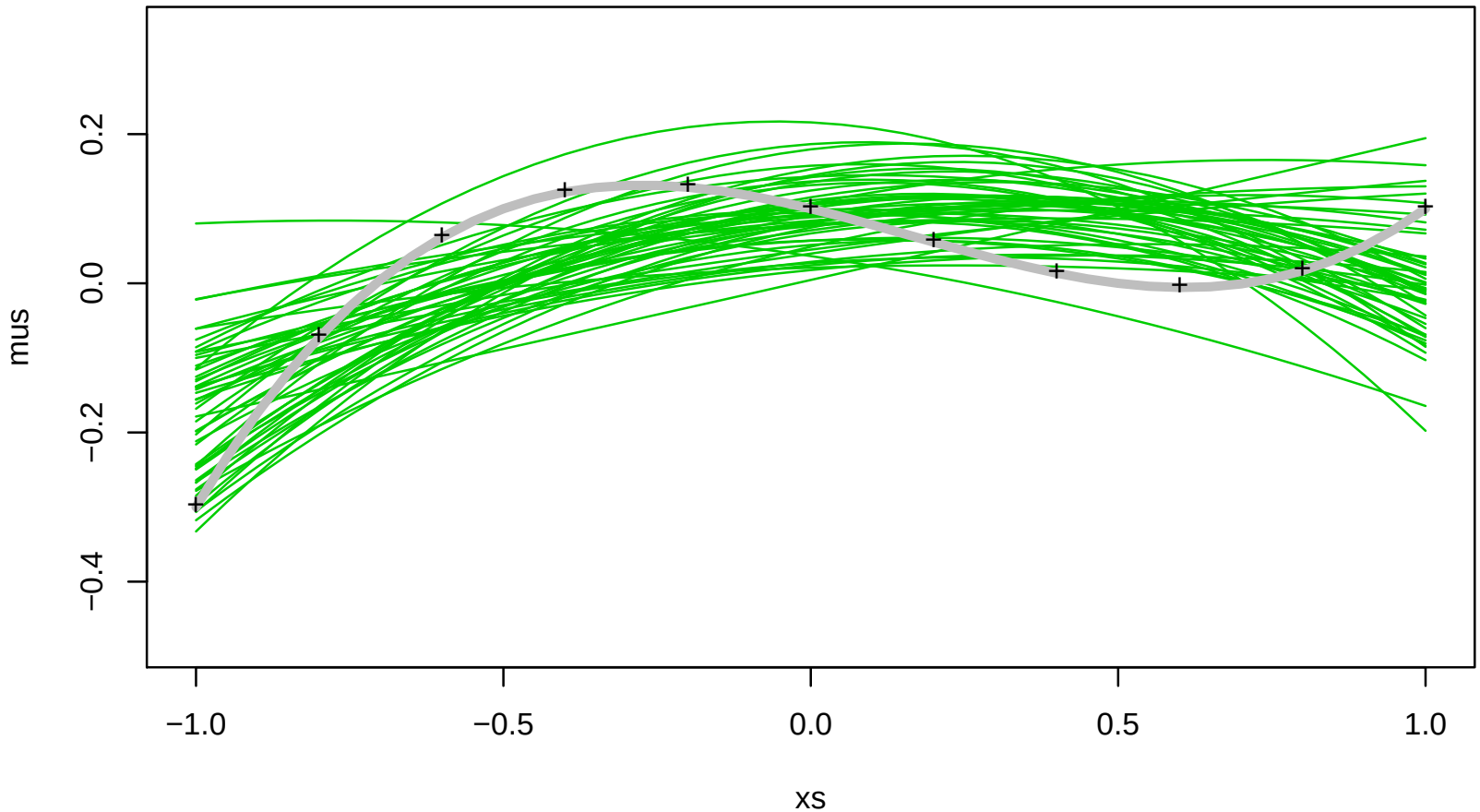

y
0.2
0.1
0.0
−0.1
−0.2
−1.0
−0.5
0.0
0.5
1.0
x

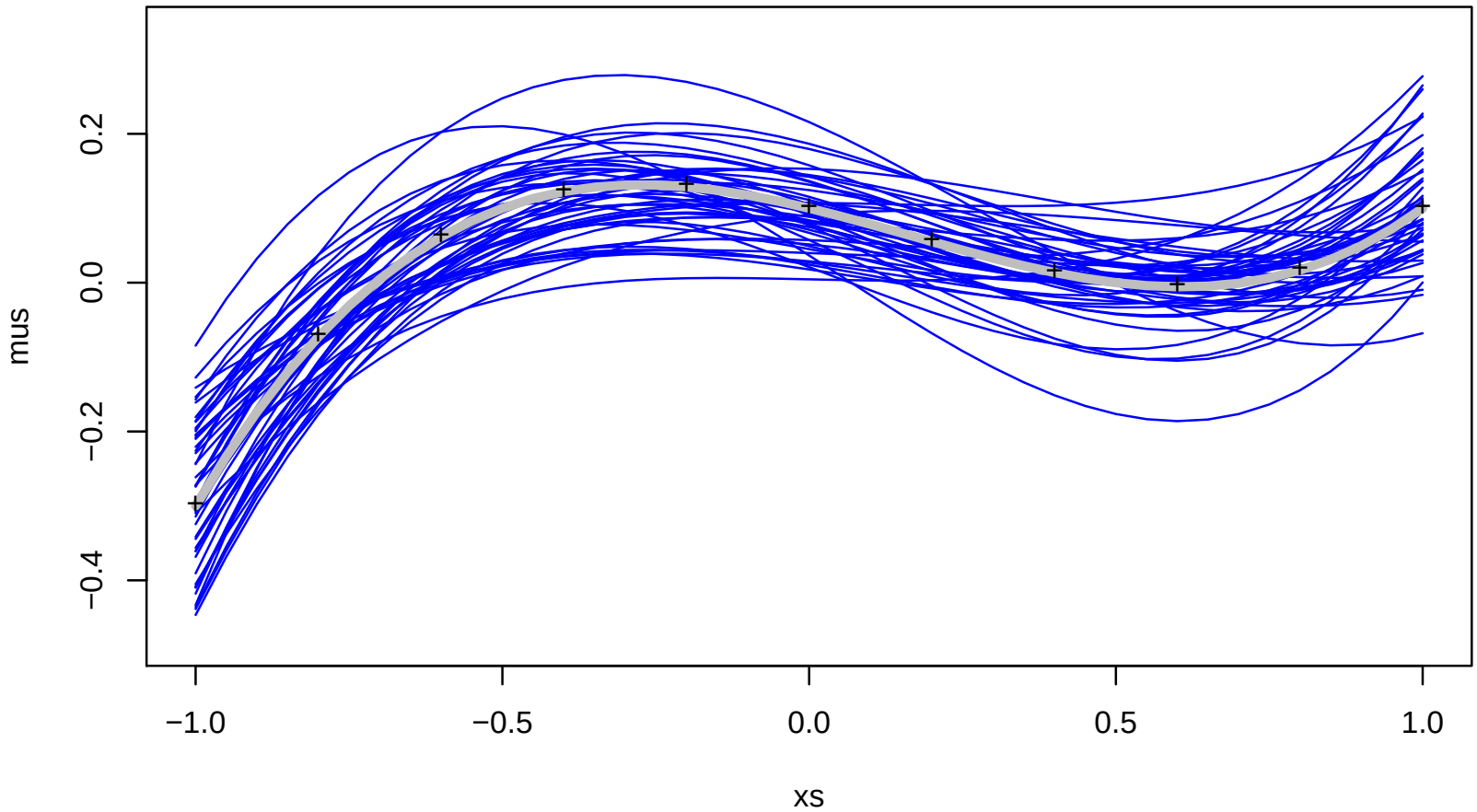

y
0.2
0.1
0.0
−0.1
−0.2
−1.0
−0.5
0.0
0.5
1.0
x

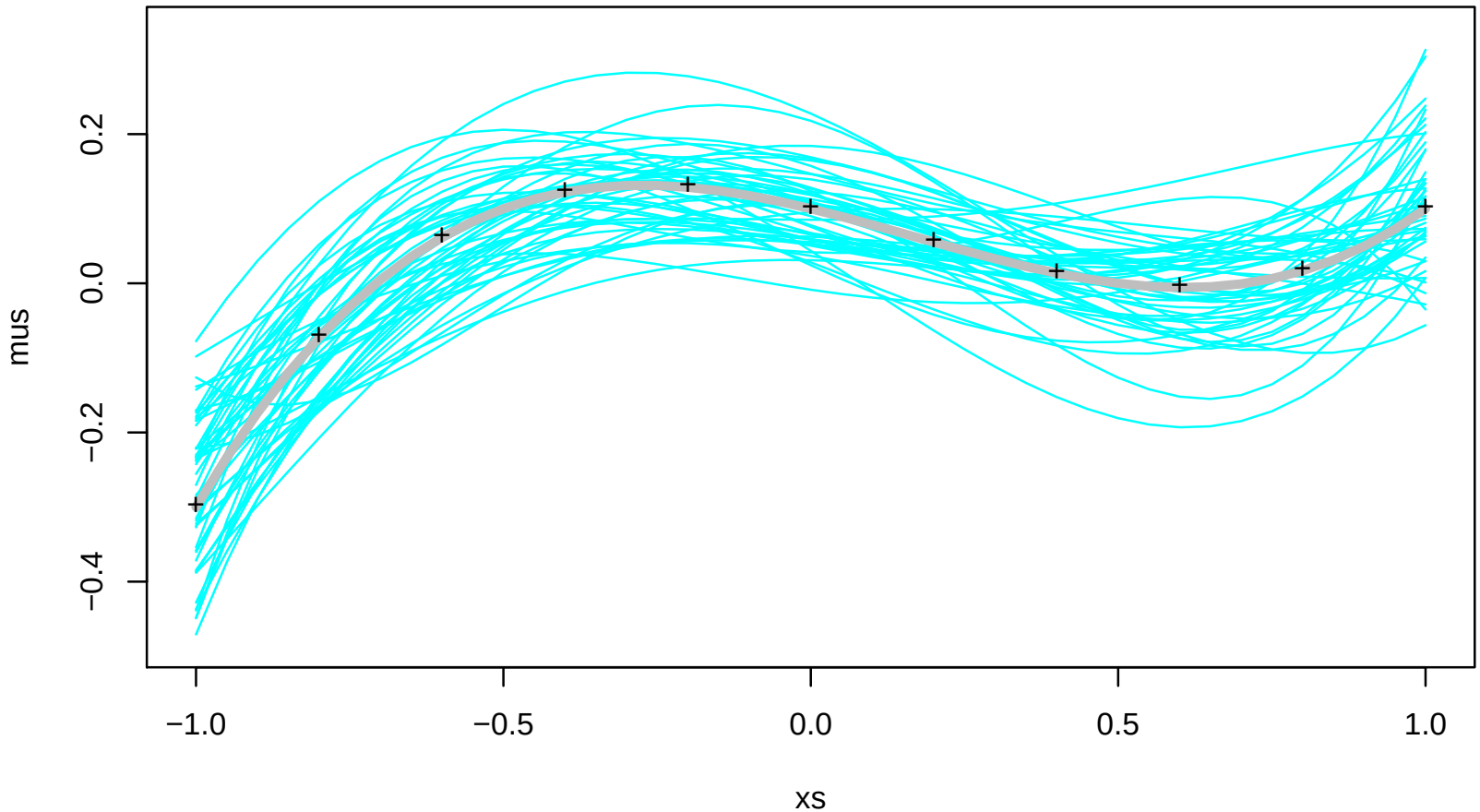
# Simulation study

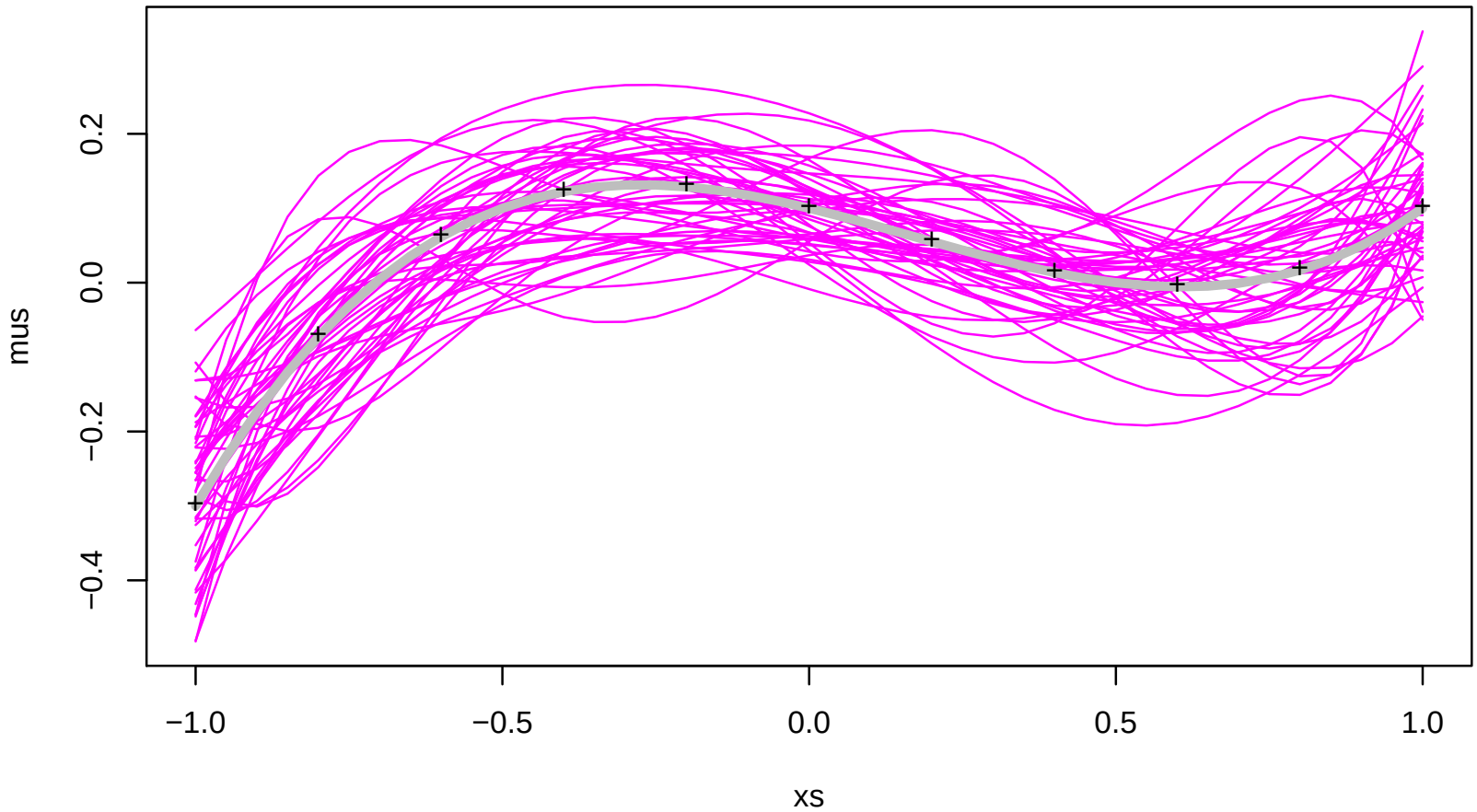
```
B1<-B2<-B3<-B4<-B5<-NULL
for(s in 1:50)
{
  y<-rnorm(length(x),mu,sigma)
  B1<-rbind(B1,lm(y~-1+X1)$coef )
  B2<-rbind(B2,lm(y~-1+X2)$coef )
  B3<-rbind(B3,lm(y~-1+X3)$coef )
  B4<-rbind(B4,lm(y~-1+X4)$coef )
  B5<-rbind(B5,lm(y~-1+X5)$coef )
}
```


mus
0.2
0.0
−0.2
−0.4
−1.0
−0.5
0.0
0.5
1.0
xs


mus
0.2
0.0
−0.2
−0.4
−1.0
−0.5
0.0
0.5
1.0
xs


mus
0.2
0.0
−0.2
−0.4
−1.0
−0.5
0.0
0.5
1.0
xs


mus
0.2
0.0
−0.2
−0.4
−1.0
−0.5
0.0
0.5
1.0
xs


mus
0.2
0.0
−0.2
−0.4
−1.0
−0.5
0.0
0.5
1.0
xs

# Observations

- If you include too few variables, bias will be high.
- If you include too many variables, estimation variability will be high.
- The best model for prediction balances bias and variance.
- The best model for prediction **might not be the correct model**.

The best model for prediction depends on knowledge of bias and variance.

In general, we don't know either of these (don't know $\mu$ or $\sigma^2$).

Various summary statistics attempt to *estimate* E[PSS].

- $C_p$;
- PRESS;
- AIC, BIC;
- Prediction error on test datasets.

# Observations

- If you include too few variables, bias will be high.
- If you include too many variables, estimation variability will be high.
- The best model for prediction balances bias and variance.
- The best model for prediction **might not be the correct model**.

The best model for prediction depends on knowledge of bias and variance.

In general, we don't know either of these (don't know $\mu$ or $\sigma^2$).

Various summary statistics attempt to *estimate* E[PSS].

- $C_p$;
- PRESS;
- AIC, BIC;
- Prediction error on test datasets.

# Observations

- If you include too few variables, bias will be high.
- If you include too many variables, estimation variability will be high.
- The best model for prediction balances bias and variance.
- The best model for prediction **might not be the correct model**.

The best model for prediction depends on knowledge of bias and variance.

In general, we don't know either of these (don't know $\mu$ or $\sigma^2$).

Various summary statistics attempt to *estimate* E[PSS].

- $C_p$;
- PRESS;
- AIC, BIC;
- Prediction error on test datasets.

# Observations

- If you include too few variables, bias will be high.
- If you include too many variables, estimation variability will be high.
- The best model for prediction balances bias and variance.
- The best model for prediction **might not be the correct model**.

The best model for prediction depends on knowledge of bias and variance.

In general, we don't know either of these (don't know $\mu$ or $\sigma^2$).

Various summary statistics attempt to *estimate* E[PSS].

- $C_p$;
- PRESS;
- AIC, BIC;
- Prediction error on test datasets.

# $C_p$

$$\begin{aligned} \text{E[PSS]} &= n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[RSS]} &= n\sigma^2 - p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\mu||^2 \\ \text{E[PSS]} &= \text{E[RSS]} + 2p\sigma^2 \end{aligned}$$

**Q:** What about $\sigma^2$ ?

**A:** Let $\tilde{\sigma}^2$ be the variance estimate from the largest model considered.

- Even if that model is too big, $\tilde{\sigma}^2$ is unbiased for $\sigma^2$.
- If the model is still too small, $\tilde{\sigma}^2$ is biased upward.

$C_p$:

- Evaluate each considered model with $C_p = \text{RSS} + 2p\tilde{\sigma}^2$
- $\text{E}[C_p] = \text{E[PSS]}$.

# $C_p$

$$\begin{aligned} \text{E[PSS]} &= n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[RSS]} &= n\sigma^2 - p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\mu||^2 \\ \text{E[PSS]} &= \text{E[RSS]} + 2p\sigma^2 \end{aligned}$$

**Q:** What about $\sigma^2$ ?

**A:** Let $\tilde{\sigma}^2$ be the variance estimate from the largest model considered.

- Even if that model is too big, $\tilde{\sigma}^2$ is unbiased for $\sigma^2$.
- If the model is still too small, $\tilde{\sigma}^2$ is biased upward.

$C_p$:

- Evaluate each considered model with $C_p = \text{RSS} + 2p\tilde{\sigma}^2$
- $\text{E}[C_p] = \text{E[PSS]}$.

# $C_p$

$$\begin{aligned} \text{E[PSS]} &= n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[RSS]} &= n\sigma^2 - p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[PSS]} &= \text{E[RSS]} + 2p\sigma^2 \end{aligned}$$

**Q:** What about $\sigma^2$ ?

**A:** Let $\tilde{\sigma}^2$ be the variance estimate from the largest model considered.

- Even if that model is too big, $\tilde{\sigma}^2$ is unbiased for $\sigma^2$.
- If the model is still too small, $\tilde{\sigma}^2$ is biased upward.

$C_p$:

- Evaluate each considered model with $C_p = \text{RSS} + 2p\tilde{\sigma}^2$
- $\text{E}[C_p] = \text{E[PSS]}$.

# $C_p$

$$\begin{aligned} \text{E[PSS]} &= n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[RSS]} &= n\sigma^2 - p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[PSS]} &= \text{E[RSS]} + 2p\sigma^2 \end{aligned}$$

**Q:** What about $\sigma^2$ ?

**A:** Let $\tilde{\sigma}^2$ be the variance estimate from the largest model considered.

- Even if that model is too big, $\tilde{\sigma}^2$ is unbiased for $\sigma^2$.
- If the model is still too small, $\tilde{\sigma}^2$ is biased upward.

$C_p$:

- Evaluate each considered model with $C_p = \text{RSS} + 2p\tilde{\sigma}^2$
- $\text{E}[C_p] = \text{E[PSS]}$.

# $C_p$

$$\begin{aligned} \text{E[PSS]} &= n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[RSS]} &= n\sigma^2 - p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[PSS]} &= \text{E[RSS]} + 2p\sigma^2 \end{aligned}$$

**Q:** What about $\sigma^2$ ?

**A:** Let $\tilde{\sigma}^2$ be the variance estimate from the largest model considered.

- Even if that model is too big, $\tilde{\sigma}^2$ is unbiased for $\sigma^2$.
- If the model is still too small, $\tilde{\sigma}^2$ is biased upward.

$C_p$:

- Evaluate each considered model with $C_p = \text{RSS} + 2p\tilde{\sigma}^2$
- $\text{E}[C_p] = \text{E[PSS]}$.

# $C_p$

$$\begin{aligned} \text{E[PSS]} &= n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[RSS]} &= n\sigma^2 - p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[PSS]} &= \text{E[RSS]} + 2p\sigma^2 \end{aligned}$$

**Q:** What about $\sigma^2$ ?

**A:** Let $\tilde{\sigma}^2$ be the variance estimate from the largest model considered.

- Even if that model is too big, $\tilde{\sigma}^2$ is unbiased for $\sigma^2$.
- If the model is still too small, $\tilde{\sigma}^2$ is biased upward.

$C_p$:

- Evaluate each considered model with $C_p = \text{RSS} + 2p\tilde{\sigma}^2$
- $\text{E}[C_p] = \text{E[PSS]}$.

# $C_p$

$$\begin{aligned} \text{E[PSS]} &= n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[RSS]} &= n\sigma^2 - p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[PSS]} &= \text{E[RSS]} + 2p\sigma^2 \end{aligned}$$

**Q:** What about $\sigma^2$ ?

**A:** Let $\tilde{\sigma}^2$ be the variance estimate from the largest model considered.

- Even if that model is too big, $\tilde{\sigma}^2$ is unbiased for $\sigma^2$.
- If the model is still too small, $\tilde{\sigma}^2$ is biased upward.

$C_p$:

- Evaluate each considered model with $C_p = \text{RSS} + 2p\tilde{\sigma}^2$
- $\text{E}[C_p] = \text{E[PSS]}$.

# $C_p$

$$\begin{aligned} \text{E[PSS]} &= n\sigma^2 + p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[RSS]} &= n\sigma^2 - p\sigma^2 + ||(\mathbf{I} - \mathbf{P})\boldsymbol{\mu}||^2 \\ \text{E[PSS]} &= \text{E[RSS]} + 2p\sigma^2 \end{aligned}$$

**Q:** What about $\sigma^2$ ?

**A:** Let $\tilde{\sigma}^2$ be the variance estimate from the largest model considered.

- Even if that model is too big, $\tilde{\sigma}^2$ is unbiased for $\sigma^2$.
- If the model is still too small, $\tilde{\sigma}^2$ is biased upward.

$C_p$:

- Evaluate each considered model with $C_p = \text{RSS} + 2p\tilde{\sigma}^2$
- $\text{E}[C_p] = \text{E[PSS]}$.

# Model comparison with $C_p$

$$C_p = RSS + 2p\tilde{\sigma}^2$$

$C_p(M_1) = RSS + 2p_1\tilde{\sigma}^2 = C_p$ statistic for model 1.

$C_p(M_2) = RSS + 2p_2\tilde{\sigma}^2 = C_p$ statistic for model 2.

Prefer $M_1$ to $M_2$ if $C_p(M_1) < C_p(M_2)$.

# Model comparison with $C_p$

$$C_p = RSS + 2p\tilde{\sigma}^2$$

$C_p(M_1) = RSS + 2p_1\tilde{\sigma}^2 = C_p$ statistic for model 1.

$C_p(M_2) = RSS + 2p_2\tilde{\sigma}^2 = C_p$ statistic for model 2.

Prefer $M_1$ to $M_2$ if $C_p(M_1) < C_p(M_2)$.

# Model comparison with $C_p$

**Note:** Often $C_p$ is defined as

$$C_p(M_j) = \text{RSS}_j/\tilde{\sigma}^2 + 2p_j - n$$

This doesn't change the ordering of model preferences.

---

**Caution:**
The minimizer of the unbiased estimators of E[PSS] *is not* an unbiased estimator of the minimizing E[PSS]. In general,

$$\text{E}[\min_M C_p(M)] \leq \min_M \text{E}[\text{PSS}]$$

So $\min_M C_p(M)$ is generally too optimistic in terms of prediction error.

# Model comparison with $C_p$

**Note:** Often $C_p$ is defined as

$$C_p(M_j) = \text{RSS}_j/\tilde{\sigma}^2 + 2p_j - n$$

This doesn't change the ordering of model preferences.

---

**Caution:**
The minimizer of the unbiased estimators of E[PSS] *is not* an unbiased estimator of the minimizing E[PSS]. In general,

$$\text{E}[\min_M C_p(M)] \leq \min_M \text{E}[\text{PSS}]$$

So $\min_M C_p(M)$ is generally too optimistic in terms of prediction error.

# Model comparison with $C_p$

**Note:** Often $C_p$ is defined as

$$C_p(M_j) = \text{RSS}_j/\tilde{\sigma}^2 + 2p_j - n$$

This doesn't change the ordering of model preferences.

---

**Caution:**
The minimizer of the unbiased estimators of E[PSS] *is not* an unbiased estimator of the minimizing E[PSS]. In general,

$$\text{E}[\min_M C_p(M)] \leq \min_M \text{E}[\text{PSS}]$$

So $\min_M C_p(M)$ is generally too optimistic in terms of prediction error.

# AIC and $C_p$

**Akaike information criterion:** A general estimate of prediction error.

For a general statistical model $p(\mathbf{y}|\theta_M)$, the AIC is

$$AIC(M) = -2 \log p(\mathbf{y}|\hat{\theta}_M) + 2p_M$$

where $\hat{\theta}_M$ is the MLE of $\theta_M$ and $p_M$ is the dimension of $\theta_M$.

For a linear regression model, this becomes

$$AIC(M) = n(1 + \log 2\pi/n) + n \log RSS_M + 2p_M$$

This is similar to $C_p$:

- Both balance fit and complexity.
- Choice based on $C_p$ and $AIC$ are asymptotically equivalent.
- $C_p$ and AIC generally select models that are
  - larger than the true model;
  - good at minimizing prediction error.

# AIC and $C_p$

**Akaike information criterion:** A general estimate of prediction error.

For a general statistical model $p(\mathbf{y}|\theta_M)$, the AIC is

$$AIC(M) = -2 \log p(\mathbf{y}|\hat{\theta}_M) + 2p_M$$

where $\hat{\theta}_M$ is the MLE of $\theta_M$ and $p_M$ is the dimension of $\theta_M$.

For a linear regression model, this becomes

$$AIC(M) = n(1 + \log 2\pi/n) + n \log RSS_M + 2p_M$$

This is similar to $C_p$:

- Both balance fit and complexity.
- Choice based on $C_p$ and $AIC$ are asymptotically equivalent.
- $C_p$ and AIC generally select models that are
  - larger than the true model;
  - good at minimizing prediction error.

# AIC and $C_p$

**Akaike information criterion:** A general estimate of prediction error.

For a general statistical model $p(\mathbf{y}|\theta_M)$, the AIC is

$$AIC(M) = -2 \log p(\mathbf{y}|\hat{\theta}_M) + 2p_M$$

where $\hat{\theta}_M$ is the MLE of $\theta_M$ and $p_M$ is the dimension of $\theta_M$.

For a linear regression model, this becomes

$$AIC(M) = n(1 + \log 2\pi/n) + n \log RSS_M + 2p_M$$

This is similar to $C_p$:

- Both balance fit and complexity.
- Choice based on $C_p$ and $AIC$ are asymptotically equivalent.
- $C_p$ and AIC generally select models that are
  - larger than the true model;
  - good at minimizing prediction error.

# AIC and $C_p$

**Akaike information criterion:** A general estimate of prediction error.

For a general statistical model $p(\mathbf{y}|\theta_M)$, the AIC is

$$AIC(M) = -2 \log p(\mathbf{y}|\hat{\theta}_M) + 2p_M$$

where $\hat{\theta}_M$ is the MLE of $\theta_M$ and $p_M$ is the dimension of $\theta_M$.

For a linear regression model, this becomes

$$AIC(M) = n(1 + \log 2\pi/n) + n \log RSS_M + 2p_M$$

This is similar to $C_p$:

- Both balance fit and complexity.
- Choice based on $C_p$ and $AIC$ are asymptotically equivalent.
- $C_p$ and AIC generally select models that are
  - larger than the true model;
  - good at minimizing prediction error.

# AIC and BIC

$$AIC(M) = n \log RSS_M + 2 \times p_M$$
$$BIC(M) = n \log RSS_M + log(n) \times p_M$$

BIC more heavily penalizes complexity as the sample size grows.

- BIC will (asymptotically) select the correct model (if in model space)
- AIC will (asymptotically) select the best model for prediction.

# AIC and BIC

$$AIC(M) = n \log RSS_M + 2 \times p_M$$
$$BIC(M) = n \log RSS_M + log(n) \times p_M$$

BIC more heavily penalizes complexity as the sample size grows.

- BIC will (asymptotically) select the correct model (if in model space)
- AIC will (asymptotically) select the best model for prediction.

# AIC and BIC

$$AIC(M) = n \log RSS_M + 2 \times p_M$$
$$BIC(M) = n \log RSS_M + log(n) \times p_M$$

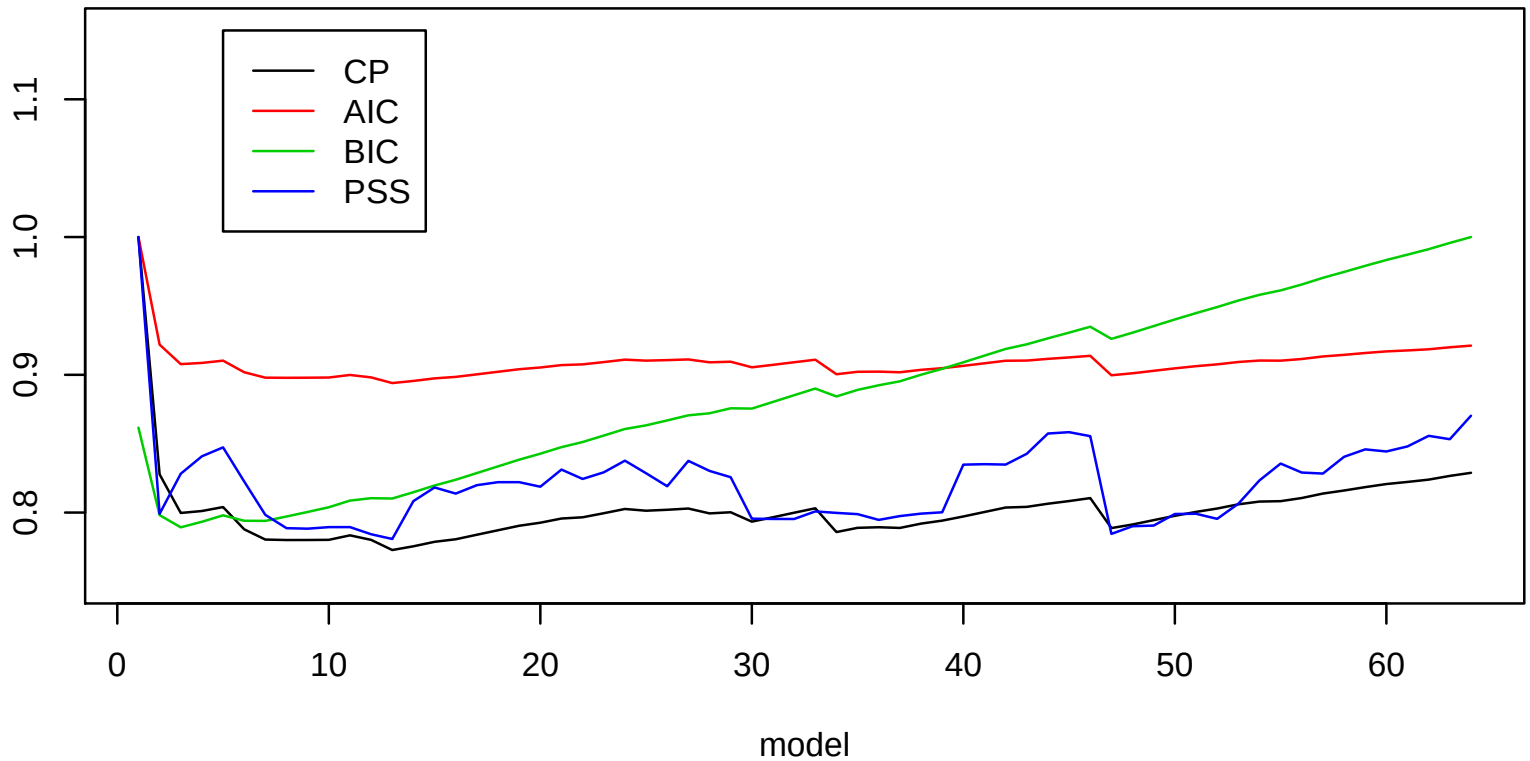
BIC more heavily penalizes complexity as the sample size grows.

- BIC will (asymptotically) select the correct model (if in model space)
- AIC will (asymptotically) select the best model for prediction.

# Example: Diabetes progression

```
AIC<-BIC<-RSS<-PSS<-NULL

for(j in 1:ncol(Xtr))
{
  fit<- lm( y ~ -1 + X[,rcor[1:j]] )

  RSS<-c(RSS, sum(fit$res^2) )

  AIC<-c(AIC, AIC(fit) )

  BIC<-c(BIC, BIC(fit) )


  fit_tr<-lm( ytr ~ Xtr[,rcor[1:j]] )
  PSS<-c(PSS, sum( (yte-cbind(1,Xte[,rcor[1:j],drop=FALSE])%*%fit_tr$coef)^2 )

}

s2hat<-summary(fit)$sigma^2

CP<-RSS + 2*s2hat*(1:64)
```

# Comparison



```r
which.min(CP)
```

```
## [1] 13
```

```r
which.min(AIC)
```

```
## [1] 13
```

```r
which.min(BIC)
```

```
## [1] 3
```

```r
which.min(PSS)
```

```
## [1] 13
```

```
round(summary( lm( y ~ -1+  X[,rcor[1:13]] ))$coef,3)
```

```
##                          Estimate Std. Error t value Pr(>|t|)
## X[, rcor[1:13]]bmi          0.313      0.047   6.732    0.000
## X[, rcor[1:13]]ltg          0.510      0.107   4.764    0.000
## X[, rcor[1:13]]map          0.162      0.041   3.933    0.000
## X[, rcor[1:13]]tch          0.044      0.100   0.440    0.660
## X[, rcor[1:13]]glu          0.016      0.041   0.387    0.699
## X[, rcor[1:13]]hdl          0.098      0.132   0.742    0.459
## X[, rcor[1:13]]bmi^2        0.047      0.042   1.111    0.267
## X[, rcor[1:13]]tc          -0.529      0.259  -2.043    0.042
## X[, rcor[1:13]]map^2        0.012      0.039   0.298    0.766
## X[, rcor[1:13]]ldl          0.377      0.212   1.781    0.076
## X[, rcor[1:13]]age          0.001      0.037   0.020    0.984
## X[, rcor[1:13]]bmi:map      0.069      0.042   1.655    0.099
## X[, rcor[1:13]]glu^2        0.087      0.035   2.499    0.013
```

```
round(summary( lm( y ~ -1 + X[,rcor[1:3]] ))$coef,3)
```

```
##                       Estimate Std. Error t value Pr(>|t|)
## X[, rcor[1:3]]bmi        0.373      0.040   9.335        0
## X[, rcor[1:3]]ltg        0.336      0.040   8.426        0
## X[, rcor[1:3]]map        0.162      0.039   4.170        0
```

# Comparison to backwards selection

```
fit_bs<-step(lm(y~X[,1]+X[,2]+X[,3]+X[,4]+X[,5]+X[,6] +X[,7] + X[,8] + X[,9] +
 X[,10]+X[,11]+X[,12]+X[,13]+X[,14]+X[,15]+X[,16]+X[,17]+X[,18]+X[,19]+
 X[,20]+X[,21]+X[,22]+X[,23]+X[,24]+X[,25]+X[,26]+X[,27]+X[,28]+X[,29]+
 X[,30]+X[,31]+X[,32]+X[,33]+X[,34]+X[,35]+X[,36]+X[,37]+X[,38]+X[,39]+
 X[,40]+X[,41]+X[,42]+X[,43]+X[,44]+X[,45]+X[,46]+X[,47]+X[,48]+X[,49]+
 X[,50]+X[,51]+X[,52]+X[,53]+X[,54]+X[,55]+X[,56]+X[,57]+X[,58]+X[,59]+
 X[,60]+X[,61]+X[,62]+X[,63]+X[,64] ) , direction="backward", trace=0)
```

```
AIC(fit_bs)

## [1] 929.7878

AIC(lm( y ~ -1+  X[,rcor[1:13]] )  )

## [1] 957.5259
```

```r
summary(fit_bs)$coef
```

```
##                  Estimate Std. Error       t value     Pr(>|t|)
## (Intercept) -1.829230e-17 0.03195039 -5.725221e-16 1.000000e+00
## X[, 2]      -1.645210e-01 0.03666673 -4.486928e+00 9.363808e-06
## X[, 3]       3.064222e-01 0.04054688  7.557233e+00 2.648130e-13
## X[, 4]       2.118805e-01 0.03885998  5.452408e+00 8.537035e-08
## X[, 5]      -5.297840e-01 0.12121620 -4.370571e+00 1.567216e-05
## X[, 6]       4.222447e-01 0.11423267  3.696357e+00 2.479796e-04
## X[, 9]       6.008533e-01 0.05733979  1.047882e+01 5.863357e-23
## X[, 11]      5.158435e-02 0.03647666  1.414174e+00 1.580587e-01
## X[, 14]      3.821722e+00 1.81889793  2.101120e+00 3.623155e-02
## X[, 15]      2.429713e+00 1.36095686  1.785297e+00 7.494154e-02
## X[, 16]      6.506550e-01 0.33770999  1.926668e+00 5.470188e-02
## X[, 18]      9.191220e-01 0.32170474  2.857036e+00 4.490711e-03
## X[, 19]      1.207572e-01 0.03999570  3.019255e+00 2.690002e-03
## X[, 20]      1.119235e-01 0.03807215  2.939774e+00 3.467639e-03
## X[, 23]     -6.293280e-02 0.04445406 -1.415682e+00 1.576170e-01
## X[, 25]      6.827735e-02 0.04217416  1.618938e+00 1.062185e-01
## X[, 27]      1.129612e-01 0.05013625  2.253084e+00 2.477409e-02
## X[, 30]      5.197597e-02 0.03575293  1.453754e+00 1.467685e-01
## X[, 37]      1.058287e-01 0.03675174  2.879557e+00 4.188009e-03
## X[, 49]     -7.404143e-02 0.04415067 -1.677017e+00 9.429031e-02
## X[, 50]     -5.959216e+00 2.99470385 -1.989918e+00 4.725411e-02
## X[, 51]     -1.820864e+00 0.89611593 -2.031951e+00 4.279353e-02
## X[, 53]     -2.771258e+00 1.14225981 -2.426119e+00 1.568555e-02
## X[, 55]      1.439655e+00 0.75084949  1.917369e+00 5.587565e-02
## X[, 57]      2.136519e+00 0.90533586  2.359918e+00 1.873990e-02
## X[, 60]      9.691256e-01 0.40717554  2.380118e+00 1.775682e-02
```

# Training and test sets

One strategy for estimating PSS is with *training* and *test* sets:

- Randomly divide the data into $\{\mathbf{y}_{tr}, \mathbf{X}_{tr}\}$ $\{\mathbf{y}_{te}, \mathbf{X}_{te}\}$;
- Obtain $\hat{\beta}$ from $\{\mathbf{y}_{tr}, \mathbf{X}_{tr}\}$;
- Compare $\mathbf{y}_{te}$ to $\mathbf{X}_{te}\hat{\beta}$:

$$\mathrm{E}[PSS] \approx ||\mathbf{y}_{te} - \mathbf{X}_{te}\hat{\beta}||^2$$

**Q:** Why not use $||\mathbf{y}_{tr} - \mathbf{X}_{tr}\hat{\beta}||^2$ ?

**A:** This is just RSS, which will underestimate prediction error

- $\hat{\beta}$ matches the signal *and* noise in $\mathbf{y}_{tr}$;
- A complex model will do great at matching the noise;
- We want to see how well it does at matching the signal.

**Note:** The book calls these *construction* and *validation* sets.

# Training and test sets

One strategy for estimating PSS is with *training* and *test* sets:

- Randomly divide the data into $\{\mathbf{y}_{tr}, \mathbf{X}_{tr}\}$ $\{\mathbf{y}_{te}, \mathbf{X}_{te}\}$;
- Obtain $\hat{\beta}$ from $\{\mathbf{y}_{tr}, \mathbf{X}_{tr}\}$;
- Compare $\mathbf{y}_{te}$ to $\mathbf{X}_{te}\hat{\beta}$:

$$\mathrm{E}[PSS] \approx ||\mathbf{y}_{te} - \mathbf{X}_{te}\hat{\beta}||^2$$

**Q:** Why not use $||\mathbf{y}_{tr} - \mathbf{X}_{tr}\hat{\beta}||^2$ ?

**A:** This is just RSS, which will underestimate prediction error

- $\hat{\beta}$ matches the signal *and* noise in $\mathbf{y}_{tr}$;
- A complex model will do great at matching the noise;
- We want to see how well it does at matching the signal.

**Note:** The book calls these *construction* and *validation* sets.

# Training and test sets

One strategy for estimating PSS is with *training* and *test* sets:

- Randomly divide the data into $\{\mathbf{y}_{tr}, \mathbf{X}_{tr}\}$ $\{\mathbf{y}_{te}, \mathbf{X}_{te}\}$;
- Obtain $\hat{\beta}$ from $\{\mathbf{y}_{tr}, \mathbf{X}_{tr}\}$;
- Compare $\mathbf{y}_{te}$ to $\mathbf{X}_{te}\hat{\beta}$:

$$\mathrm{E}[PSS] \approx ||\mathbf{y}_{te} - \mathbf{X}_{te}\hat{\beta}||^2$$

**Q:** Why not use $||\mathbf{y}_{tr} - \mathbf{X}_{tr}\hat{\beta}||^2$ ?

**A:** This is just RSS, which will underestimate prediction error

- $\hat{\beta}$ matches the signal *and* noise in $\mathbf{y}_{tr}$;
- A complex model will do great at matching the noise;
- We want to see how well it does at matching the signal.

**Note:** The book calls these *construction* and *validation* sets.

# Training and test sets

One strategy for estimating PSS is with *training* and *test* sets:

- Randomly divide the data into $\{\mathbf{y}_{tr}, \mathbf{X}_{tr}\}$ $\{\mathbf{y}_{te}, \mathbf{X}_{te}\}$;
- Obtain $\hat{\beta}$ from $\{\mathbf{y}_{tr}, \mathbf{X}_{tr}\}$;
- Compare $\mathbf{y}_{te}$ to $\mathbf{X}_{te}\hat{\beta}$:

$$\mathrm{E}[PSS] \approx ||\mathbf{y}_{te} - \mathbf{X}_{te}\hat{\beta}||^2$$

**Q:** Why not use $||\mathbf{y}_{tr} - \mathbf{X}_{tr}\hat{\beta}||^2$ ?

**A:** This is just RSS, which will underestimate prediction error

- $\hat{\beta}$ matches the signal *and* noise in $\mathbf{y}_{tr}$;
- A complex model will do great at matching the noise;
- We want to see how well it does at matching the signal.

**Note:** The book calls these *construction* and *validation* sets.

# Training and test sets

Use of a single training and test set can be problematic:

- If you select a model $M$, the estimates from the full data might substantially different than those from the training data.
- The best model using the full data might be different than the best model based on training/test sets of half the size.
- An estimate of E[$PSS$] from a half-size dataset might not be very good.

# Cross validation

Consider instead the following:

- Divide the dataset into $\{\mathbf{y}_1, \mathbf{X}_1\}$ $\{\mathbf{y}_2, \mathbf{X}_2\}$;
- Obtain $\hat{\boldsymbol{\beta}}_1$ from $\{\mathbf{y}_1, \mathbf{X}_1\}$, $\hat{\boldsymbol{\beta}}_2$ from $\{\mathbf{y}_2, \mathbf{X}_2\}$
- Compare $\mathbf{y}_1$ to $\mathbf{X}_1\hat{\boldsymbol{\beta}}_2$ and $\mathbf{y}_2$ to $\mathbf{X}_2\hat{\boldsymbol{\beta}}_1$.

$$\mathrm{E}[PSS] \approx ||\mathbf{y}_1 - \mathbf{X}_1\hat{\boldsymbol{\beta}}_2||^2 + ||\mathbf{y}_2 - \mathbf{X}_2\hat{\boldsymbol{\beta}}_1||^2$$

This is called *two-fold cross validation*.

Intuitively, this should provide a better approximation to $\mathrm{E}[PSS]$.

# Cross validation

Consider instead the following:

- Divide the dataset into $\{\mathbf{y}_1, \mathbf{X}_1\}$ $\{\mathbf{y}_2, \mathbf{X}_2\}$;
- Obtain $\hat{\boldsymbol{\beta}}_1$ from $\{\mathbf{y}_1, \mathbf{X}_1\}$, $\hat{\boldsymbol{\beta}}_2$ from $\{\mathbf{y}_2, \mathbf{X}_2\}$
- Compare $\mathbf{y}_1$ to $\mathbf{X}_1\hat{\boldsymbol{\beta}}_2$ and $\mathbf{y}_2$ to $\mathbf{X}_2\hat{\boldsymbol{\beta}}_1$.

$$\mathrm{E}[PSS] \approx ||\mathbf{y}_1 - \mathbf{X}_1\hat{\boldsymbol{\beta}}_2||^2 + ||\mathbf{y}_2 - \mathbf{X}_2\hat{\boldsymbol{\beta}}_1||^2$$

This is called *two-fold cross validation*.

Intuitively, this should provide a better approximation to $\mathrm{E}[PSS]$.

# Cross validation

Consider instead the following:

- Divide the dataset into $\{\mathbf{y}_1, \mathbf{X}_1\}$ $\{\mathbf{y}_2, \mathbf{X}_2\}$;
- Obtain $\hat{\boldsymbol{\beta}}_1$ from $\{\mathbf{y}_1, \mathbf{X}_1\}$, $\hat{\boldsymbol{\beta}}_2$ from $\{\mathbf{y}_2, \mathbf{X}_2\}$
- Compare $\mathbf{y}_1$ to $\mathbf{X}_1\hat{\boldsymbol{\beta}}_2$ and $\mathbf{y}_2$ to $\mathbf{X}_2\hat{\boldsymbol{\beta}}_1$.

$$\mathrm{E}[PSS] \approx ||\mathbf{y}_1 - \mathbf{X}_1\hat{\boldsymbol{\beta}}_2||^2 + ||\mathbf{y}_2 - \mathbf{X}_2\hat{\boldsymbol{\beta}}_1||^2$$

This is called *two-fold cross validation*.

Intuitively, this should provide a better approximation to $\mathrm{E}[PSS]$.

# $K$-fold cross validation

Consider instead the following:

- Divide the dataset into $\{\mathbf{y}_1, \mathbf{X}_1\}, \ldots, \{\mathbf{y}_K, \mathbf{X}_K\}$;
- Obtain $\hat{\boldsymbol{\beta}}_{-k}$ from $\{\mathbf{y}_j, \mathbf{X}_j : j \neq k\}$,
- Compare $\mathbf{y}_k$ to $\mathbf{X}_k \hat{\boldsymbol{\beta}}_{-k}$

$$\mathrm{E}[PSS] \approx \sum_k ||\mathbf{y}_k - \mathbf{X}_k \hat{\boldsymbol{\beta}}_{-k}||^2$$

This is called *$K$-fold cross validation*.

# $K$-fold cross validation

Consider instead the following:

- Divide the dataset into $\{\mathbf{y}_1, \mathbf{X}_1\}, \ldots, \{\mathbf{y}_K, \mathbf{X}_K\}$;
- Obtain $\hat{\boldsymbol{\beta}}_{-k}$ from $\{\mathbf{y}_j, \mathbf{X}_j : j \neq k\}$,
- Compare $\mathbf{y}_k$ to $\mathbf{X}_k \hat{\boldsymbol{\beta}}_{-k}$

$$\mathrm{E}[PSS] \approx \sum_k ||\mathbf{y}_k - \mathbf{X}_k \hat{\boldsymbol{\beta}}_{-k}||^2$$

This is called *K-fold cross validation*.

# $K$-fold cross validation

Consider instead the following:

- Divide the dataset into $\{\mathbf{y}_1, \mathbf{X}_1\}, \ldots, \{\mathbf{y}_K, \mathbf{X}_K\}$;
- Obtain $\hat{\boldsymbol{\beta}}_{-k}$ from $\{\mathbf{y}_j, \mathbf{X}_j : j \neq k\}$,
- Compare $\mathbf{y}_k$ to $\mathbf{X}_k \hat{\boldsymbol{\beta}}_{-k}$

$$\mathrm{E}[PSS] \approx \sum_k ||\mathbf{y}_k - \mathbf{X}_k \hat{\boldsymbol{\beta}}_{-k}||^2$$

This is called *K-fold cross validation*.

# "Leave out one" cross validation

Consider instead the following:

- Let $\mathbf{y}_{-i}, \mathbf{X}_{-i}$ denote the dataset with the $i$th case removed;
- Obtain $\hat{\boldsymbol{\beta}}_{-i}$ from $\mathbf{y}_{-i}, \mathbf{X}_{-i}$
- Compare $y_i$ to $\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{-i}$

$$\mathrm{E}[PSS] \approx \sum_i (y_i - x_i^T \hat{\boldsymbol{\beta}}_{-i})^2 = \mathrm{PRESS}$$

This procedure is called *n-fold cross validation*.

This seems ideal:

- The model is selected based on fits using $n-1$ observations;
- The parameter estimates from $\mathbf{y}_{-i}, \mathbf{X}_{-i}$ should be pretty close.

**Problem?** It seems you would need to fit the model $n$ times!

# "Leave out one" cross validation

Consider instead the following:

- Let $\mathbf{y}_{-i}, \mathbf{X}_{-i}$ denote the dataset with the $i$th case removed;
- Obtain $\hat{\boldsymbol{\beta}}_{-i}$ from $\mathbf{y}_{-i}, \mathbf{X}_{-i}$
- Compare $y_i$ to $\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{-i}$

$$\mathrm{E}[PSS] \approx \sum_i (y_i - x_i^T \hat{\boldsymbol{\beta}}_{-i})^2 = \mathrm{PRESS}$$

This procedure is called *n-fold cross validation*.

This seems ideal:

- The model is selected based on fits using $n-1$ observations;
- The parameter estimates from $\mathbf{y}_{-i}, \mathbf{X}_{-i}$ should be pretty close.

**Problem?** It seems you would need to fit the model $n$ times!

# "Leave out one" cross validation

Consider instead the following:

- Let $\mathbf{y}_{-i}, \mathbf{X}_{-i}$ denote the dataset with the $i$th case removed;
- Obtain $\hat{\boldsymbol{\beta}}_{-i}$ from $\mathbf{y}_{-i}, \mathbf{X}_{-i}$
- Compare $y_i$ to $\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{-i}$

$$\mathrm{E}[PSS] \approx \sum_i (y_i - x_i^T \hat{\boldsymbol{\beta}}_{-i})^2 = \mathrm{PRESS}$$

This procedure is called *n-fold cross validation*.

This seems ideal:

- The model is selected based on fits using $n-1$ observations;
- The parameter estimates from $\mathbf{y}_{-i}, \mathbf{X}_{-i}$ should be pretty close.

**Problem?** It seems you would need to fit the model $n$ times!

# "Leave out one" cross validation

Consider instead the following:

- Let $\mathbf{y}_{-i}, \mathbf{X}_{-i}$ denote the dataset with the $i$th case removed;
- Obtain $\hat{\boldsymbol{\beta}}_{-i}$ from $\mathbf{y}_{-i}, \mathbf{X}_{-i}$
- Compare $y_i$ to $\mathbf{x}_i^T \hat{\boldsymbol{\beta}}_{-i}$

$$\mathrm{E}[PSS] \approx \sum_i (y_i - x_i^T \hat{\boldsymbol{\beta}}_{-i})^2 = \mathrm{PRESS}$$

This procedure is called *n-fold cross validation*.

This seems ideal:

- The model is selected based on fits using $n-1$ observations;
- The parameter estimates from $\mathbf{y}_{-i}, \mathbf{X}_{-i}$ should be pretty close.

**Problem?** It seems you would need to fit the model $n$ times!

# Matrix magic

Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ be the "hat" matrix.

Previously we called this matrix "**P**" for "projection."

$$\begin{aligned}\mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= \mathbf{X}\hat{\beta} = \hat{\mathbf{y}}\end{aligned}$$

Let $h_{ii} = \mathbf{H}_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$. Then

$$\text{PRESS} = \sum(\tfrac{\hat{\epsilon}_i}{1-h_{ii}})^2,$$

where $\hat{\epsilon}_i$ is the $i$th residual using the all the data.

# Matrix magic

Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ be the "hat" matrix.

Previously we called this matrix "**P**" for "projection."

$$\begin{aligned}\mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= \mathbf{X}\hat{\beta} = \hat{\mathbf{y}}\end{aligned}$$

Let $h_{ii} = \mathbf{H}_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$. Then

$$\text{PRESS} = \sum(\tfrac{\hat{\epsilon}_i}{1-h_{ii}})^2,$$

where $\hat{\epsilon}_i$ is the $i$th residual using the all the data.

# Matrix magic

Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ be the "hat" matrix.

Previously we called this matrix "**P**" for "projection."

$$\begin{aligned}\mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= \mathbf{X}\hat{\beta} = \hat{\mathbf{y}}\end{aligned}$$

Let $h_{ii} = \mathbf{H}_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$. Then

$$\text{PRESS} = \sum(\tfrac{\hat{\epsilon}_i}{1-h_{ii}})^2,$$

where $\hat{\epsilon}_i$ is the $i$th residual using the all the data.

# Matrix magic

Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ be the "hat" matrix.

Previously we called this matrix "**P**" for "projection."

$$\begin{aligned}\mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= \mathbf{X}\hat{\beta} = \hat{\mathbf{y}}\end{aligned}$$

Let $h_{ii} = \mathbf{H}_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$. Then

$$\text{PRESS} = \sum(\tfrac{\hat{\epsilon}_i}{1-h_{ii}})^2,$$

where $\hat{\epsilon}_i$ is the $i$th residual using the all the data.

# Matrix magic

Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ be the "hat" matrix.

Previously we called this matrix "**P**" for "projection."

$$\begin{aligned}\mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= \mathbf{X}\hat{\beta} = \hat{\mathbf{y}}\end{aligned}$$

Let $h_{ii} = \mathbf{H}_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$. Then

$$\text{PRESS} = \sum(\tfrac{\hat{\epsilon}_i}{1-h_{ii}})^2,$$

where $\hat{\epsilon}_i$ is the $i$th residual using the all the data.

# Matrix magic

Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ be the "hat" matrix.

Previously we called this matrix "**P**" for "projection."

$$\begin{aligned}\mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= \mathbf{X}\hat{\boldsymbol{\beta}} = \hat{\mathbf{y}}\end{aligned}$$

Let $h_{ii} = \mathbf{H}_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$. Then

$$\text{PRESS} = \sum(\tfrac{\hat{\epsilon}_i}{1-h_{ii}})^2,$$

where $\hat{\epsilon}_i$ is the $i$th residual using the all the data.

# Matrix magic

Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ be the "hat" matrix.

Previously we called this matrix "**P**" for "projection."

$$\begin{aligned}\mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= \mathbf{X}\hat{\boldsymbol{\beta}} = \hat{\mathbf{y}}\end{aligned}$$

Let $h_{ii} = \mathbf{H}_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$. Then

$$\text{PRESS} = \sum(\tfrac{\hat{\epsilon}_i}{1-h_{ii}})^2,$$

where $\hat{\epsilon}_i$ is the $i$th residual using the all the data.

# Matrix magic

Let $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ be the "hat" matrix.

Previously we called this matrix "**P**" for "projection."

$$\begin{aligned}\mathbf{H}\mathbf{y} &= \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\ &= \mathbf{X}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}] \\ &= \mathbf{X}\hat{\boldsymbol{\beta}} = \hat{\mathbf{y}}\end{aligned}$$

Let $h_{ii} = \mathbf{H}_{ii} = \mathbf{x}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i$. Then

$$\text{PRESS} = \sum(\tfrac{\hat{\epsilon}_i}{1-h_{ii}})^2,$$

where $\hat{\epsilon}_i$ is the $i$th residual using the all the data.

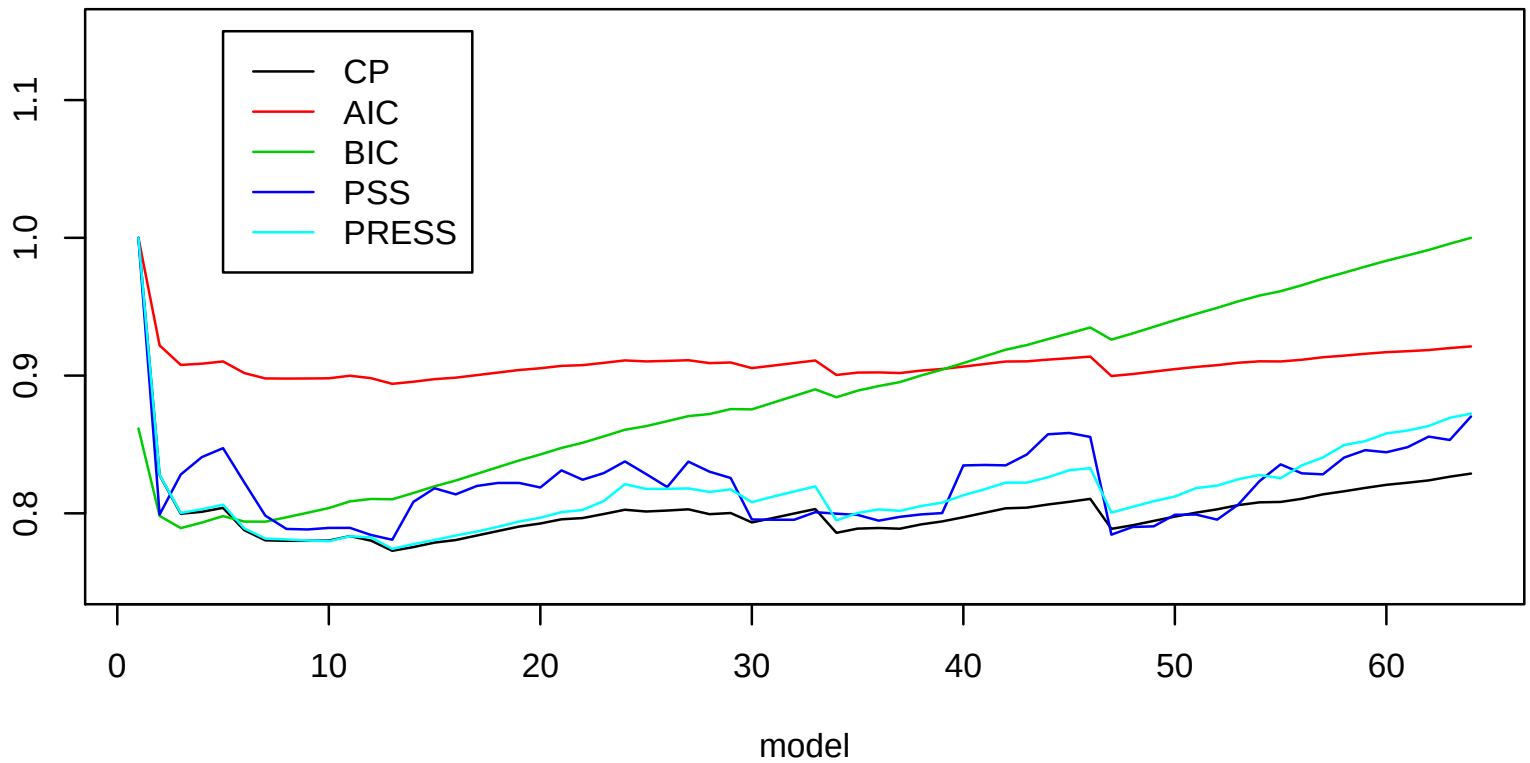
# Diabetes example

```
H<-X%*% solve( t(X)%*%X ) %*%t(X)

diag(H)[1:5]

##           1           2           3           4           5
## 0.06397911 0.11228880 0.12754413 0.09562653 0.04944736

influence(fit)$hat[1:5]

##           1           2           3           4           5
## 0.06397911 0.11228880 0.12754413 0.09562653 0.04944736
```

# PRESS for diabetes example

```
PRESS<-NULL

for(j in 1:ncol(Xtr))
{
  fit<- lm( y ~ -1 + X[,rcor[1:j]] )

  h<-influence(fit)$hat

  PRESS<-c(PRESS, sum( (fit$res/(1-h))^2 ) )

}
```

```
PRESS[1:15]

##  [1] 290.4278 240.4526 232.4235 233.2080 234.1532 229.1721 227.0113
##  [8] 226.8352 226.6060 226.4541 227.5069 227.2611 224.8457 225.8457
## [15] 226.7070
```




```
which.min(PRESS)
```

```
## [1] 13
```