

## 「データ解析」(下平英寿)

## 講義資料 7

## 検定と信頼区間

- 目標： 検定と信頼区間について理解する .
  1. 回帰係数の  $t$ -検定
  2. 回帰係数, 予測値の信頼区間
  3. 回帰モデルの  $F$  検定
  4. 予測式の信頼区間 (同時信頼区間)

1 回帰係数の  $t$ -検定

## 1.1 重回帰分析の復習

- 重回帰モデルは

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

ベクトル表現すると

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

- 回帰係数ベクトル  $\boldsymbol{\beta}$  を最小二乗法で推定すると

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

で与えられる .

- 正規モデル

$$\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$$

を仮定すると, 回帰係数ベクトルは多変量正規分布に従う

$$\hat{\boldsymbol{\beta}} \sim N_{p+1}(\boldsymbol{\beta}, \sigma^2 (\mathbf{X}'\mathbf{X})^{-1})$$

ここでサイズ  $(p+1) \times (p+1)$  の行列  $\mathbf{A}$  を

$$\mathbf{A} = (a_{ij}) = (\mathbf{X}'\mathbf{X})^{-1}, \quad i, j = 0, \dots, p$$

とおけば,

$$\hat{\beta}_i \sim N(\beta_i, \sigma^2 a_{ii})$$

- データから  $V(\hat{\beta}_i) = \sigma^2 a_{ii}$  を推定するには, 誤差分散  $\hat{\sigma}^2$  をその推定量で置き換える . 誤差  $\epsilon$  の分散  $\sigma^2$  の不偏推定は

$$S_\epsilon^2 = \frac{1}{n-p-1} \sum_{i=1}^n e_i^2 = \frac{\|\mathbf{e}\|^2}{n-p-1}$$

回帰係数  $\beta_0, \beta_1, \dots, \beta_p$  の個数は  $p+1$  個であることに注意 . したがって, 自由度は  $n-(p+1)$  .

- 不偏推定量  $S_\epsilon^2$  を用いると,

$$\hat{V}(\hat{\beta}_i) = S_\epsilon^2 a_{ii}$$

である．したがって標準誤差の推定は

$$\hat{se}(\hat{\beta}_i) = \sqrt{\hat{V}(\hat{\beta}_i)} = S_\epsilon \sqrt{a_{ii}}$$

Rの組み込み関数では, この式を用いている．

- $t$ -統計量,  $p$ -値は

$$t_i = \frac{\hat{\beta}_i}{\hat{se}(\hat{\beta}_i)}, \quad p_i = 2 \Pr\{T > |t_i|\}$$

で与えられる． $T$  は自由度  $n - p - 1$  の  $t$ -分布に従う確率変数．確率値は, 仮に  $\beta_i = 0$  であると仮定したとき, 実際に観測した  $|t_i|$  より絶対値の大きな  $t$ -統計量を観測する確率を表す．これが小さいほど, 仮定した  $\beta_i = 0$  と矛盾するので,  $\beta_i \neq 0$  と考えられる (背理法と同様のロジック.)

- 統計的仮説検定では, あらかじめ閾値  $\alpha$  をさだめておく (通常  $\alpha = 0.05$  または  $0.01$  を使うことが多い) . そして,  $p_i < \alpha$  ならば  $\beta_i \neq 0$ ,  $p_i \geq \alpha$  ならば  $\beta_i = 0$  と「判定」する .
- もし本当に  $\beta_i = 0$  だとすると,  $p_i$  は区間  $[0, 1]$  の一様分布に従う (確率変数をその分布関数で変換してるから) . 従って,  $p_i < \alpha$  となり誤って  $\beta_i \neq 0$  と判定してしまう確率は  $\alpha$  である .
- 100% 正確な判断はありえない .  $\alpha$  は判断の正確さを調整するパラメタと考えてよい .  $\alpha$  を小さくすればするほど,  $\beta_i = 0$  を誤って  $\beta_i \neq 0$  と判定する確率は小さくなる . しかし逆に, もし  $\beta_i \neq 0$  が真実のときに誤って  $\beta_i = 0$  と誤判定する確率は増える . この両者はトレードオフの関係にある .
- 正規モデルを仮定すると,  $t$ -統計量は, 自由度  $n - p - 1$  の  $t$ -分布に従う確率変数の実現値である . このことを次で示す .

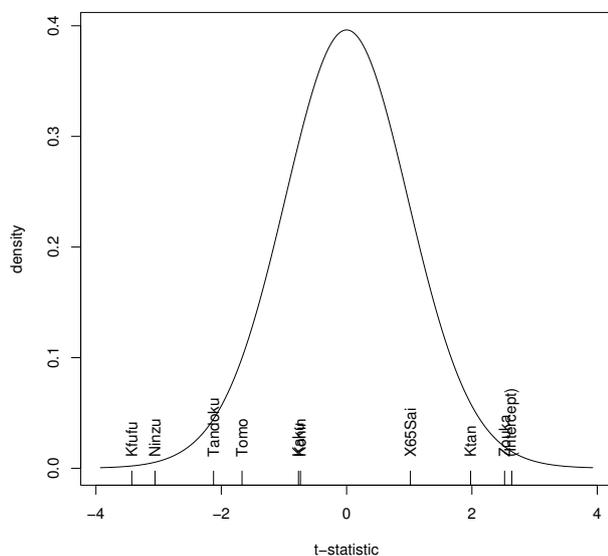
```
# run0073.R
# 回帰係数の t-検定
# あらかじめ x に説明変数の行列, y に目的変数のベクトルを設定しておく
fit <- lm(y~.,data.frame(x,y)) # 線形モデルの当てはめ
be <- coef(summary(fit)) # 結果の取り出し
cat("# 回帰係数, 標準誤差, t-統計量, p-値\n"); print(be)
tval <- be[,"t value"] # t-統計量
mx <- max(abs(tval))+0.5; x0 <- seq(-mx,mx,len=300) # プロットの範囲を決める
plot(x0,dt(x0,df.residual(fit)),type="l",
      xlab="t-statistic",ylab="density") # t-分布の密度関数
rug(tval); text(tval,0.01,names(tval),srt=90,adj=0)
dev.copy2eps(file="run0073-d.eps")
pv0 <- 2*pt(abs(x0),df.residual(fit),lower.tail=F) # p-値
plot(x0,pv0,type="l",log="y",xlab="t-statistic",ylab="p-value")
```

```
abline(h=c(0.01,0.05),lty=3)
rug(tval); text(tval,pv0[1]*1.1,names(tval),srt=90,adj=0)
dev.copy2eps(file="run0073-t.eps")
```

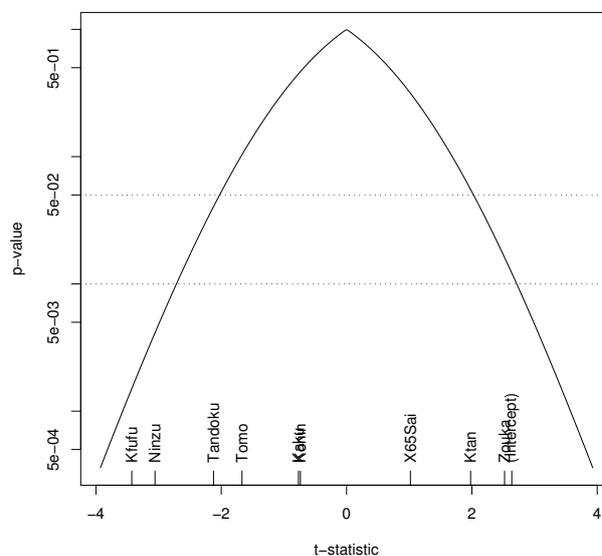
```
> dat <- read.table("dat0002.txt") # データの読み込み (47 x 10 行列)
> x <- dat[,-10]; y <- dat[,10]
> source("run0073.R")
```

# 回帰係数, 標準誤差, t-統計量, p-値

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	15.89979396	6.030963886	2.6363603	0.012175936
Zouka	1.35299378	0.536658497	2.5211448	0.016136162
Ninzu	-3.26224134	1.066976341	-3.0574636	0.004131840
Kaku	-0.02794050	0.036376197	-0.7680984	0.447303442
Tomo	-0.01586652	0.009506475	-1.6690220	0.103554493
Tandoku	-0.11120432	0.052370545	-2.1234134	0.040470167
X65Sai	0.03597994	0.035336821	1.0181996	0.315195078
Kfufu	-0.20127472	0.058708226	-3.4283904	0.001504385
Ktan	0.10625525	0.053686459	1.9791816	0.055273358
Konin	-0.08330936	0.113092312	-0.7366492	0.465981104



run0073-d



run0073-t

## 1.2 カイ二乗分布と $t$ -分布

- 正規モデルを仮定すると以下の結果が得られる .
- 残差二乗和を誤差分散  $\sigma^2$  で割ったものは, 自由度  $n - p - 1$  のカイ二乗分布に従う .

$$\sum_{i=1}^n \frac{e_i^2}{\sigma^2} \sim \chi_{n-p-1}^2$$

- 一般に  $n$  個の独立な正規分布  $Z_1, \dots, Z_n \sim N(0, 1)$  の二乗和は自由度  $n$  のカイ二乗分布に従う

$$\sum_{i=1}^n Z_i^2 \sim \chi_n^2$$

従って、誤差の二乗和

$$\sum_{i=1}^n \frac{\epsilon_i^2}{\sigma^2} \sim \chi_n^2$$

である。残差二乗和のほうでは、 $e_i \sim N(0, \sigma^2(1 - h_{ii}))$  だったので、大雑把にいて  $e_i/\sigma$  は近似的に  $N(0, 1)$  であるから、 $\sum_{i=1}^n \frac{e_i^2}{\sigma^2} \sim \chi_n^2$  でもよさそうな気もちよとするが、実際には自由度が  $n - p - 1$  になる。

- この事実をキチンと証明するには、まず

$$e = (I_n - H)\epsilon$$

$$\|e\|^2 = \epsilon'(I_n - H)^2\epsilon = \epsilon'(I_n - H)\epsilon = \|\epsilon\|^2 - \epsilon'H\epsilon$$

に注意する。 $H$  は  $\text{Span}(x_0, \dots, x_p)$  への射影行列であり、 $I_n - H$  はその直交補空間への射影行列である。適当に座標系をとりなおすことにより、 $n \times n$  の直交行列

$$U = (u_1, u_2, \dots, u_n)$$

を使って、

$$H = (u_1, \dots, u_{p+1})(u_1, \dots, u_{p+1})'$$

$$I_n - H = (u_{p+2}, \dots, u_n)(u_{p+2}, \dots, u_n)'$$

とかける。

$$z_i = u_i'\epsilon/\sigma, \quad i = 1, \dots, n$$

と置けば、これは互いに独立に  $N(0, 1)$  に従う確率変数である。従って、

$$\epsilon'H\epsilon/\sigma^2 = z_1^2 + \dots + z_{p+1}^2 \sim \chi_{p+1}^2$$

$$\|e\|^2/\sigma^2 = z_{p+2}^2 + \dots + z_n^2 \sim \chi_{n-p-1}^2$$

である。さらに、 $\hat{\beta}$  は  $z_1, \dots, z_{p+1}$  だけに関係していて  $z_{p+2}, \dots, z_n$  とは無関係であるから、 $\hat{\beta}$  と  $S_e^2 = \|e\|^2/(n - p - 1)$  は互いに独立である。

- ここまでの結果をまとめると、

1. 回帰係数の最小二乗推定  $\hat{\beta}$  の従う分布は

$$\hat{\beta} \sim N_{p+1}(\beta, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$$

ここでサイズ  $(p + 1) \times (p + 1)$  の行列  $A$  を

$$A = (a_{ij}) = (\mathbf{X}'\mathbf{X})^{-1}, \quad i, j = 0, \dots, p$$

とおけば、

$$\hat{\beta}_i \sim N(\beta_i, \sigma^2 a_{ii})$$

2. 誤差分散の不偏推定  $S_\epsilon^2$  の従う分布は

$$\frac{(n-p-1)S_\epsilon^2}{\sigma^2} \sim \chi_{n-p-1}^2$$

3.  $\hat{\beta}$  と  $S_\epsilon^2$  は互いに独立

- ところで一般に  $Z \sim N(0, 1)$  と  $S^2 \sim \chi_n^2$  が互いに独立なら, これらから作られる確率変数  $T = Z/\sqrt{S^2/n}$  の従う確率分布は自由度  $n$  の  $t$ -分布であることが知られている.

$$T = \frac{Z}{\sqrt{S^2/n}} \sim t\text{-分布}_n$$

確率密度関数は

$$f(t) = \frac{\Gamma((n+1)/2)}{\sqrt{n\pi}\Gamma(n/2)} \left(1 + \frac{t^2}{n}\right)^{-(n+1)/2}$$

である.

- これを回帰分析に利用する

$$\frac{\hat{\beta}_i - \beta_i}{\sigma\sqrt{a_{ii}}} \sim N(0, 1)$$

$$\frac{(n-p-1)S_\epsilon^2}{\sigma^2} \sim \chi_{n-p-1}^2$$

なので,

$$\frac{\hat{\beta}_i - \beta_i}{\sigma\sqrt{a_{ii}}} / \sqrt{\frac{S_\epsilon^2}{\sigma^2}} = \frac{\hat{\beta}_i - \beta_i}{\sqrt{a_{ii}}S_\epsilon} \sim t\text{-分布}_{n-p-1}$$

式を整理すると,

$$\frac{\hat{\beta}_i - \beta_i}{\hat{\text{se}}(\hat{\beta}_i)} \sim t\text{-分布}_{n-p-1}$$

- ところで R で計算している  $t$ -統計量は

$$t_i = \frac{\hat{\beta}_i}{\hat{\text{se}}(\hat{\beta}_i)}$$

であるから, もし  $\beta_i = 0$  ならば

$$t_i \sim t\text{-分布}_{n-p-1}$$

である. もし  $\beta_i > 0$  ならば,  $t$ -分布から予想されるよりも大きな  $t_i$  が得られる可能性が高く, 逆にもし  $\beta_i < 0$  ならば,  $t$ -分布から予想されるよりも小さな  $t_i$  が得られる可能性が高い.

### 1.3 シミュレーションで確認

```
# run0074.R
# 回帰係数の分布：シミュレーション
source("run0044.R") # drawhist のロード
```

```

#n <- 11 # データサイズ
#be <- c(0,1) # 真の回帰係数 (beta0,beta1)
#x <- seq(0,1,len=n) # x を決める
#filename <- "run0074-"
p <- length(be)-1 # p+1 が回帰係数の個数
X <- cbind(1,x) # データ行列
colnames(X) <- names(be) <- c("beta0","beta1")
A <- solve(t(X) %*% X) # A=(X'X)^-1
B <- A %*% t(X) # B = (X'X)^-1 X'
sqrA <- sqrt(diag(A)) # A の対角項の平方根
func0074 <- function(y) {
  be <- B %*% y # 回帰係数の推定
  pred <- X %*% be # 予測値
  resid <- y-pred # 残差
  se2 <- sum(resid^2)/(n-p-1) # sigma^2 の不偏推定
  se <- sqrt(se2) # sigma の推定
  tval <- be/(se*sqrA) # t-統計量
  pval <- 2*pt(abs(tval),n-p-1,lower.tail=F) # p-値
  list(be=be,se2=se2,tval=tval,pval=pval)
}
y0 <- X %*% be # 真の y (誤差 = 0)
sigma0 <- 0.3 # 誤差の標準偏差の真値
cat("# start simulation: ",date(),"\n")
b <- 10000 # シミュレーション繰り返し数
simy <- matrix(0,n,b) # y を格納するアレイ
simbe <- matrix(0,length(be),b) # 回帰係数を格納するアレイ
simse2 <- rep(0,b) # se2 を格納するアレイ
simtval <- matrix(0,length(be),b) # t 統計量を格納するアレイ
simpval <- matrix(0,length(be),b) # p-値を格納するアレイ
for(i in 1:b) {
  simy[,i] <- y0 + rnorm(n,mean=0,sd=sigma0)
  fit <- func0074(simy[,i])
  simbe[,i] <- fit$be; simse2[i] <- fit$se2
  simtval[,i] <- fit$tval; simpval[,i] <- fit$pval
}
cat("# end simulation: ",date(),"\n")
cat("# 1 回目のシミュレーション結果\n"); print(func0074(simy[,1]))
cat("# pval0 < 0.05 の回数 = ",sum(simpval[1,]<0.05)," \n")
cat("# pval1 < 0.05 の回数 = ",sum(simpval[2,]<0.05)," \n")
if(!is.null(filename)) {
  plot(x,y0); abline(be)
  dev.copy2eps(file=paste(filename,"s0.eps",sep=""))
}

```

```

plot(x,simy[,1]); abline(simbe[,1])
dev.copy2eps(file=paste(filename,"s1.eps",sep=""))
plot(simbe[1,],simbe[2,],pch=".",xlab="beta0",ylab="beta1")
dev.copy2eps(file=paste(filename,"th1.eps",sep=""))
plot(simse2,simbe[2,],pch=".",xlab="se2",ylab="beta1")
dev.copy2eps(file=paste(filename,"th2.eps",sep=""))
for(i in 1:2) {
  drawhist(simtval[i,],30,paste("tval",i-1,sep=""))
  t0 <- seq(min(simtval[i,]),max(simtval[i,]),len=300)
  lines(t0,dt(t0,n-p-1),col=4,lty=2)
  dev.copy2eps(file=paste(filename,"tval",i-1,".eps",sep=""))
  drawhist(simpval[i,],20,paste("pval",i-1,sep=""),filename)
}
}

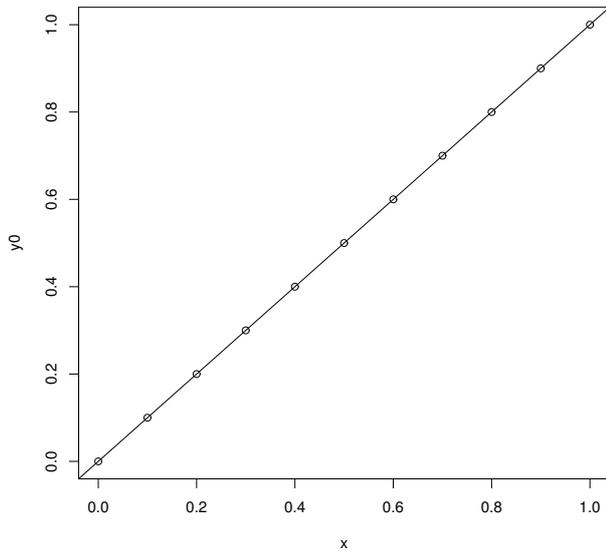
```

```

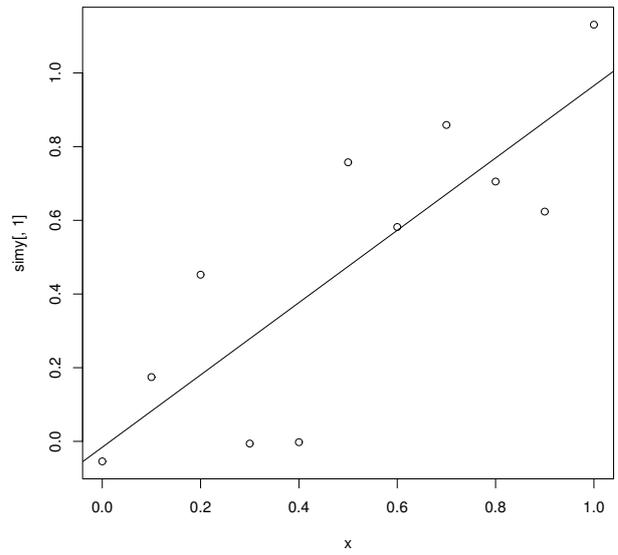
> n <- 11 # データサイズ
> be <- c(0,1) # 真の回帰係数 (beta0,beta1)
> x <- seq(0,1,len=n) # xを決める
> filename <- "run0074-"
> source("run0074.R")
# start simulation: Wed Oct 6 11:45:50 2004
# end simulation: Wed Oct 6 11:45:52 2004
# 1回目のシミュレーション結果
$be
      [,1]
beta0 -0.01592933
beta1  0.98142543
$se2
[1] 0.05716638
$tval
      [,1]
beta0 -0.1181108
beta1  4.3051007
$pval
      [,1]
beta0 0.908573939
beta1 0.001975822

# pval0 < 0.05 の回数 = 504
# pval1 < 0.05 の回数 = 8748

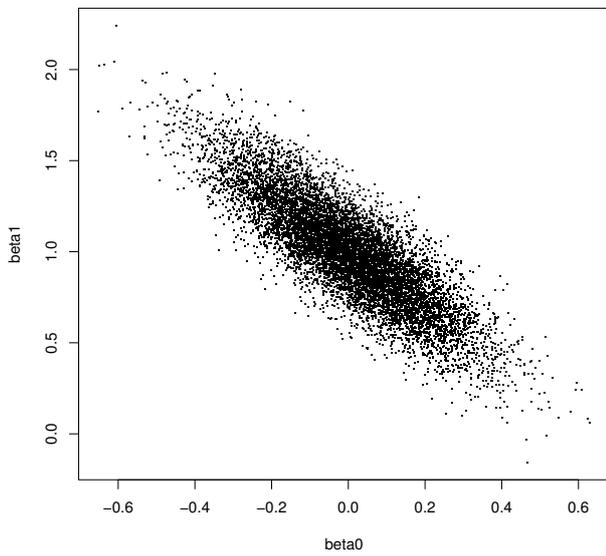
```



run0074-s0

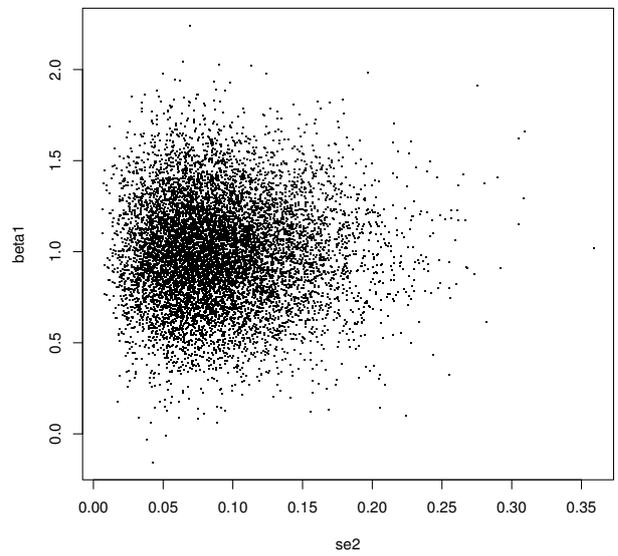


run0074-s1



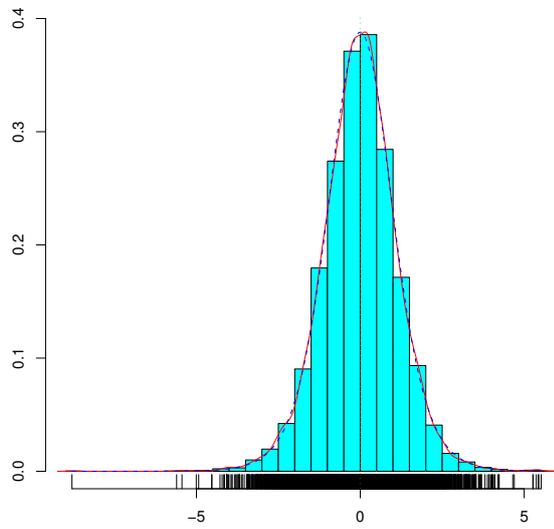
run0074-th1

**tval0**

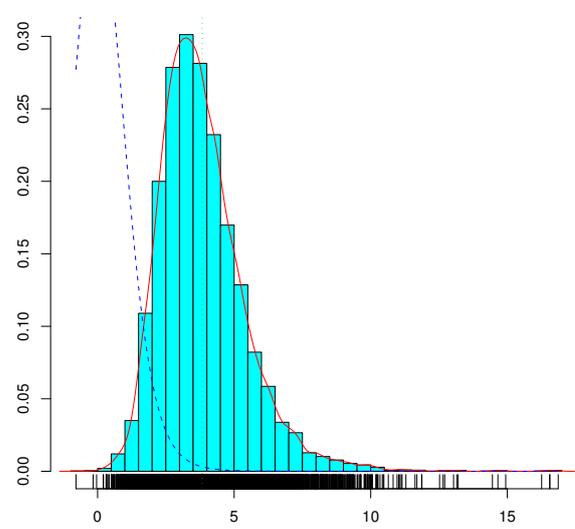


run0074-th2

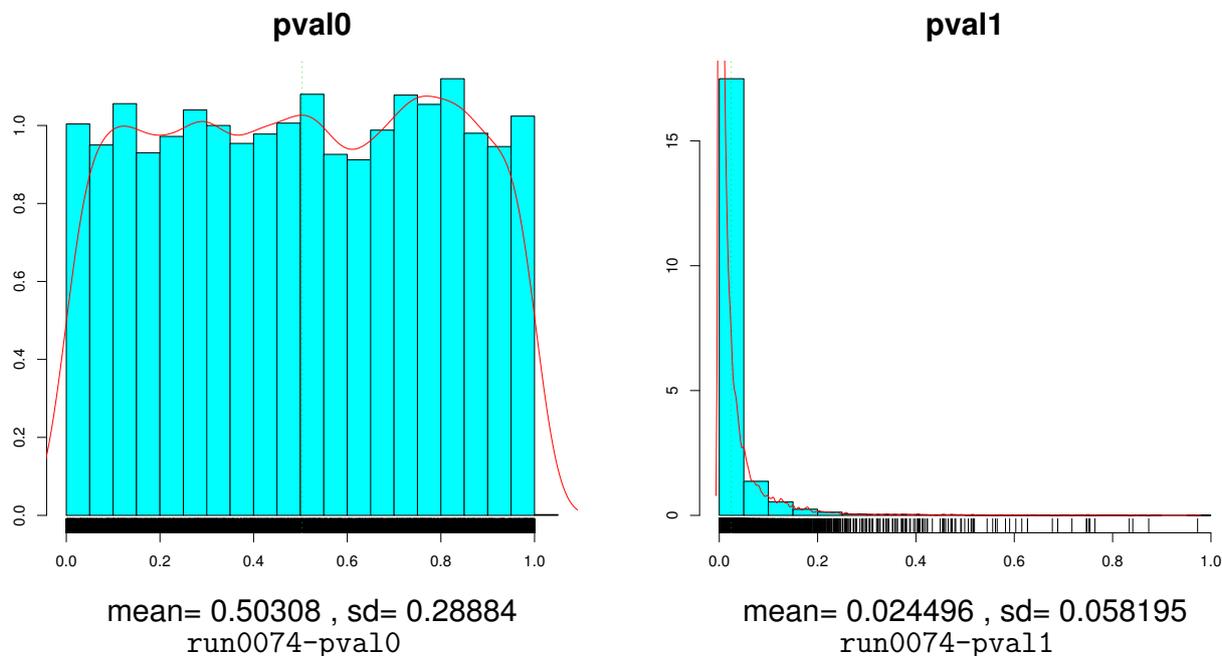
**tval1**



mean= -0.0082902 , sd= 1.1296  
run0074-tval0



mean= 3.83 , sd= 1.5606  
run0074-tval1



- $b = 10000$  回のシミュレーション . デスクトップ PC で計算して 2 秒程度 . Pentium 1G のノートパソコン ( vmware 上の linux 環境 ) で実行しても , 3 秒程度 .

## 2 回帰係数 , 予測値の信頼区間

### 2.1 回帰係数の線形結合

- 回帰係数の線形結合を考え , その性質を調べる

$$v = w_0\beta_0 + w_1\beta_1 + \cdots + w_p\beta_p$$

$$\hat{v} = w_0\hat{\beta}_0 + w_1\hat{\beta}_1 + \cdots + w_p\hat{\beta}_p$$

- $\mathbf{w} = (w_0, w_1, \dots, w_p)$  を設定することにより ,

$$w_j = 1, w_k = 0, k \neq j \Rightarrow \hat{v} = \hat{\beta}_j$$

$$w_0 = x_{i0}, \dots, w_p = x_{ip} \Rightarrow \hat{v} = \hat{y}_i$$

などが表現できるので便利 .

- ベクトル表現

$$v = \mathbf{w}'\boldsymbol{\beta}, \quad \hat{v} = \mathbf{w}'\hat{\boldsymbol{\beta}}$$

- 回帰係数は多変量正規分布に従う

$$\hat{\boldsymbol{\beta}} \sim N_{p+1}(\boldsymbol{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$$

従って ,  $\hat{v}$  は正規分布に従う

$$\hat{v} \sim N(\mathbf{w}'\boldsymbol{\beta}, \sigma^2\mathbf{w}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{w})$$

## 2.2 確率値

- 結局

$$\hat{v} \sim N(v, \sigma_v^2)$$

ただし

$$v = \mathbf{w}'\boldsymbol{\beta}, \quad \sigma_v^2 = \sigma^2 \mathbf{w}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{w}$$

- $\sigma^2$  を  $S_\epsilon^2$  で推定すると,

$$\hat{\sigma}_v^2 = S_\epsilon^2 \mathbf{w}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{w} = S_\epsilon^2 \frac{\sigma_v^2}{\sigma^2} = \frac{\|\mathbf{e}\|^2 / \sigma^2}{n-p-1} \sigma_v^2$$

つまり

$$(n-p-1) \frac{\hat{\sigma}_v^2}{\sigma_v^2} \sim \chi_{n-p-1}^2$$

- 以上より,

$$\frac{\hat{v} - v}{\sigma_v} / \sqrt{\frac{\hat{\sigma}_v^2}{\sigma_v^2}} \sim t\text{-分布}_{n-p-1}$$

式を整理すると,

$$\frac{\hat{v} - v}{\hat{\sigma}_v} \sim t\text{-分布}_{n-p-1}$$

- R では自由度  $n-p-1$  の  $t$ -分布に従う確率変数  $T$  が  $t$  以下の値をとる確率は

$$\Pr\{T \leq t\} = \text{pt}(t, n-p-1)$$

とかける．これを  $\text{pt}(t)$  とあらわしておく．

$$\Pr\{T > t\} = 1 - \text{pt}(t, n-p-1) = \text{pt}(t, n-p-1, \text{lower.tail}=\text{F})$$

というオプションも用意されていて，確率値の計算では良く用いる．

- 回帰係数の  $t$ -検定では， $\beta_i = 0$  という仮説を考えた．より一般的に，帰無仮説  $v = v_0$  を対立仮説  $v \neq v_0$  に対して検定することを考える． $p$ -値は

$$p\text{-値}(v_0) = \Pr\left\{|T| > \left|\frac{\hat{v} - v_0}{\hat{\sigma}_v}\right|\right\} = 2 \times \left(1 - \text{pt}\left(\left|\frac{\hat{v} - v_0}{\hat{\sigma}_v}\right|\right)\right)$$

ここで帰無仮説  $v = v_0$  の確率値を  $p$ -値  $(v_0)$  とあらわした．これがもし  $p$ -値  $(v_0) < \alpha$  ならば，この  $v = v_0$  という仮説が棄却され  $v \neq v_0$  と判断される．

- 特に真値  $v$  に関しては，

$$\frac{\hat{v} - v}{\sigma_v} / \sqrt{\frac{\hat{\sigma}_v^2}{\sigma_v^2}} \sim t\text{-分布}_{n-p-1}$$

より

$$p\text{-値}(v) \sim \text{一様分布}[0, 1]$$

となり，

$$\Pr\{p\text{-値}(v) < \alpha\} = \alpha$$

である．

## 2.3 信頼区間

- 先ほど，帰無仮説  $v = v_0$  の確率値を  $p$ -値 ( $v_0$ ) とあらわした．これがもし  $< \alpha$  ならば，この  $v = v_0$  という仮説が棄却され  $v \neq v_0$  と判断される．
- この方法で棄却されない  $v_0$  の集合を考える

$$\text{信頼区間 } (1 - \alpha) = \{v_0 \mid p\text{-値}(v_0) \geq \alpha\}$$

これを信頼係数（または信頼度） $1 - \alpha$  の信頼区間と呼ぶ．たとえば  $\alpha = 0.05$  ならば，信頼係数  $0.95$  の信頼区間である．

- 真値  $v$  が信頼区間に入る確率は  $1 - \alpha$ （以上）である．

$$\Pr\{v \in \text{信頼区間}(1 - \alpha)\} = 1 - \alpha$$

（証明）「 $v \in \text{信頼区間}(1 - \alpha)$ 」と「 $p\text{-値}(v) \geq \alpha$ 」は同値である．ところが  $p\text{-値}(v)$  は区間  $[0,1]$  の一様分布に従うから，この事象が起こる確率は  $1 - \alpha$  である．

- 信頼区間を計算するために，まず  $a = \text{pt}(b)$  の逆関数として， $b = \text{qt}(a)$  を用意する．つまり，自由度  $n - p - 1$  の  $t$ -分布に従う確率変数  $T$  が  $t$  以下の値をとる確率がちょうど  $a$  になるような  $t$  の値を  $\text{qt}(a)$  と書く．

$$\Pr\{T \leq \text{qt}(a)\} = a$$

R では

$$\text{qt}(a) = \text{qt}(a, n-p-1)$$

という関数が用意されている． $0 < a < 1$  である．

- 信頼区間は

$$2 \times \left(1 - \text{pt}\left(\left|\frac{\hat{v} - v_0}{\hat{\sigma}_v}\right|\right)\right) \geq \alpha$$

を整理して

$$\text{pt}\left(\left|\frac{\hat{v} - v_0}{\hat{\sigma}_v}\right|\right) \leq 1 - \alpha/2$$

従って，

$$\left|\frac{\hat{v} - v_0}{\hat{\sigma}_v}\right| \leq \text{qt}(1 - \alpha/2)$$

これを整理すると，

$$\hat{v} - \hat{\sigma}_v \text{qt}(1 - \alpha/2) \leq v_0 \leq \hat{v} + \hat{\sigma}_v \text{qt}(1 - \alpha/2)$$

つまり

$$\text{信頼区間}(1 - \alpha) = [\hat{v} - \hat{\sigma}_v \text{qt}(1 - \alpha/2), \hat{v} + \hat{\sigma}_v \text{qt}(1 - \alpha/2)]$$

## 2.4 回帰係数の信頼区間

- 回帰係数  $\beta_j$  は  $w$  を次のようにすればよい.

$$w_j = 1, w_k = 0, k \neq j \quad \Rightarrow \quad \hat{v} = \hat{\beta}_j$$

- $V(\hat{v})$  の推定は

$$\hat{\sigma}_v^2 = S_\epsilon^2 \mathbf{w}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{w} = S_\epsilon^2 a_{jj}$$

- $\beta_j$  の信頼区間は

$$\text{信頼区間}(1-\alpha) = [\hat{\beta}_j - S_\epsilon \sqrt{a_{jj}} \text{qt}(1-\alpha/2), \hat{\beta}_j + S_\epsilon \sqrt{a_{jj}} \text{qt}(1-\alpha/2)]$$

## 2.5 予測値の信頼区間

- 説明変数が  $\mathbf{x} = (1, x_1, \dots, x_p)'$  の時の予測値は  $\hat{y} = \mathbf{x}'\hat{\beta}$  である. この「真値」は  $y = \mathbf{x}'\beta$  である.  $v = y$  とするには  $w = x$  とおけばよい.

- $V(\hat{v})$  の推定は

$$\hat{\sigma}_v^2 = S_\epsilon^2 \mathbf{w}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{w} = S_\epsilon^2 \mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}$$

- $y$  の信頼区間は

$$\text{信頼区間}(1-\alpha) = [\mathbf{x}'\hat{\beta} - S_\epsilon \sqrt{\mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}} \text{qt}(1-\alpha/2), \mathbf{x}'\hat{\beta} + S_\epsilon \sqrt{\mathbf{x}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}} \text{qt}(1-\alpha/2)]$$

```
# run0075.R
# 回帰係数と予測値の信頼区間：多項式回帰
# x=説明変数のベクトル, y=目的変数のベクトル, p=多項式の次数
# a <- func0075a(x,p); b <- func0075b(y,0.05,a)
xpow <- function(a,p) a^(0:p) # c(a^0,a^1,...,a^p)
calcX <- function(x,p) { # デザイン行列 X を作る
  X <- matrix(0,length(x),p+1)
  for(i in 1:length(x)) X[i,] <- xpow(x[i],p)
  X
}
calcq0 <- function(n,p,alpha) qt(1-alpha/2,n-p-1) # 個別の信頼区間
calcq1 <- function(n,p,alpha) sqrt((p+1)*qf(1-alpha,p+1,n-p-1)) # 同時信頼区間
func0075a <- function(x,p) { # 回帰分析の準備
  X <- calcX(x,p) # データ行列
  colnames(X) <- paste("beta",0:p,sep="")
  A <- solve(t(X) %*% X) # A=(X'X)^-1
  B <- A %*% t(X) # B = (X'X)^-1 X'
  sqrA <- sqrt(diag(A)) # A の対角項の平方根
  x0 <- seq(min(x),max(x),len=300) # x の範囲を 300 等分しておく
```

```

X0 <- calcX(x0,p) # x0 に相当するデータ行列
sqrXAX <- apply(X0,1,function(x) sqrt(t(x) %*% A %*% x))
                                # sqrt(x'Ax) のベクトル
list(X=X,A=A,B=B,sqrA=sqrA,x0=x0,X0=X0,sqrXAX=sqrXAX)
}
func0075b <- function(y,alpha,a,calcq=calcq0) { # 信頼区間の計算
  n <- nrow(a$X); p <- ncol(a$X)-1
  q0 <- calcq(n,p,alpha)
  be <- a$B %*% y # 回帰係数の推定
  pred <- a$X %*% be # 予測値
  resid <- y-pred # 残差
  rss <- sum(resid^2) # 残差平方和
  se2 <- rss/(n-p-1) # sigma^2 の不偏推定
  se <- sqrt(se2) # sigma の推定
  bese <- se*a$sqrA # 回帰係数の標準誤差
  rsq <- 1-rss/sum((y-mean(y))^2) # 決定係数
  tval <- be/(se*a$sqrA) # t-統計量
  pval <- 2*pt(abs(tval),n-p-1,lower.tail=F) # p-値
  beconf <- cbind(be-q0*se*a$sqrA,be+q0*se*a$sqrA) # 信頼区間
  pred0 <- a$X0 %*% be # 予測値 (x0)
  pred0conf <- cbind(pred0-q0*se*a$sqrXAX,pred0+q0*se*a$sqrXAX)
  list(be=be,bese=bese,se=se,rss=rss,rsq=rsq,tval=tval,pval=pval,
       beconf=beconf,pred0=pred0,pred0conf=pred0conf)
}
func0075c <- function(x,y,a,b,col=2,lty=2,add=F) {
  if(!add) plot(x,y)
  lines(a$x0,b$pred0)
  lines(a$x0,b$pred0conf[,1],col=col,lty=lty)
  lines(a$x0,b$pred0conf[,2],col=col,lty=lty)
  coef <- cbind(b$be,b$bese,b$tval,b$pval,b$beconf)
  colnames(coef) <- c("Estimate","StdErr",
                    "t-value","p-value","Lower","Upper")
  invisible(list(coef=coef,se=b$se,rss=b$rss,rsq=b$rsq))
}

```

```

# run0076.R
# 回帰係数と予測値の信頼区間：多項式回帰 p 次まで
# x=説明変数ベクトル，y=目的変数ベクトル，p=多項式次数をセットしておく
source("run0075.R")
for(i in 0:p) {
  cat("# 次数=",i,"\n")

```

```

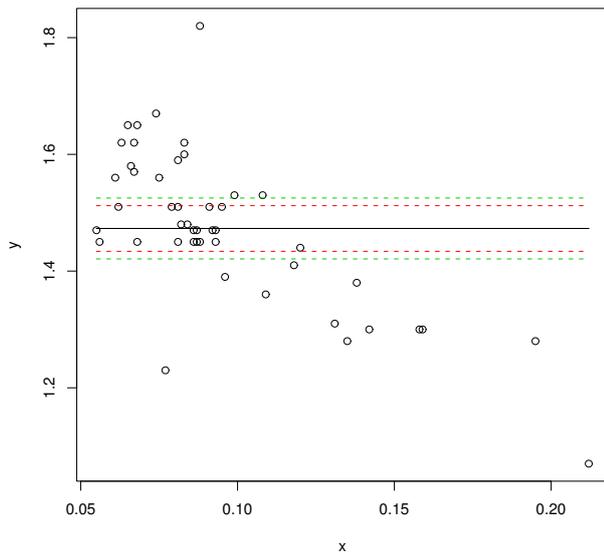
a <- func0075a(x,i)
c1 <- func0075c(x,y,a,func0075b(y,0.05,a))
c2 <- func0075c(x,y,a,func0075b(y,0.01,a),col=3,add=T)
colnames(c1$coef)[5:6] <- c("Lo05","Up05")
colnames(c2$coef)[5:6] <- c("Lo01","Up01")
coef <- cbind(c1$coef,c2$coef[,5:6,drop=F])
cat("RSS=",c1$rss," , RSQ=",c1$rsq,"\n")
print(round(coef,3))
dev.copy2eps(file=paste("run0076-s",i,".eps",sep=""))
}

```

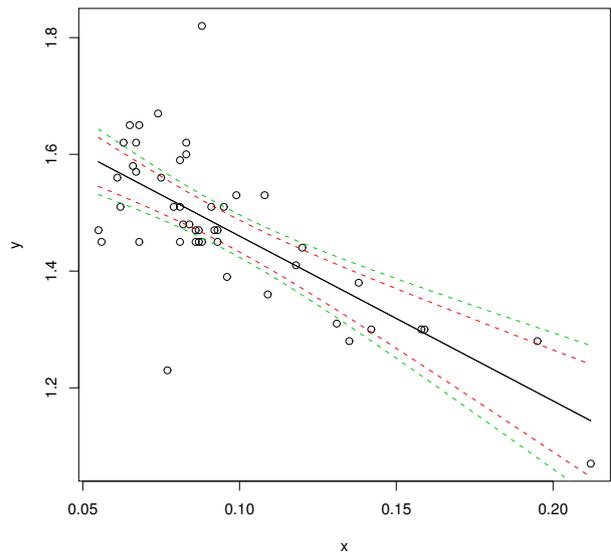
```

> dat <- read.table("dat0001.txt") # データの読み込み (47 x 2行列)
> x <- dat[,1]/100; y <- dat[,2]
> p <- 3
> source("run0076.R")
# 次数= 0
RSS= 0.815383 , RSQ= 0
      Estimate StdErr t-value p-value  Lo05  Up05  Lo01  Up01
beta0    1.473  0.019  75.848      0 1.434  1.512  1.421  1.525
# 次数= 1
RSS= 0.3812667 , RSQ= 0.5324078
      Estimate StdErr t-value p-value  Lo05  Up05  Lo01  Up01
beta0    1.742  0.040  43.592      0 1.662  1.823  1.635  1.850
beta1   -2.825  0.395  -7.158      0 -3.620 -2.030 -3.886 -1.763
# 次数= 2
RSS= 0.3786836 , RSQ= 0.5355758
      Estimate StdErr t-value p-value  Lo05  Up05  Lo01  Up01
beta0    1.681  0.120  14.043  0.000  1.440  1.922  1.359  2.003
beta1   -1.672  2.141  -0.781  0.439  -5.987  2.642  -7.436  4.091
beta2   -4.698  8.576  -0.548  0.587 -21.983 12.586 -27.788 18.391
# 次数= 3
RSS= 0.3747561 , RSQ= 0.5403926
      Estimate StdErr t-value p-value  Lo05  Up05  Lo01  Up01
beta0    1.462  0.347  4.212  0.000  0.762  2.162  0.527  2.398
beta1    4.415  9.320  0.474  0.638 -14.381 23.211 -20.704 29.534
beta2  -56.680 77.913  -0.727  0.471 -213.807 100.447 -266.664 153.304
beta3  135.499 201.844  0.671  0.506 -271.559 542.557 -408.492 679.491

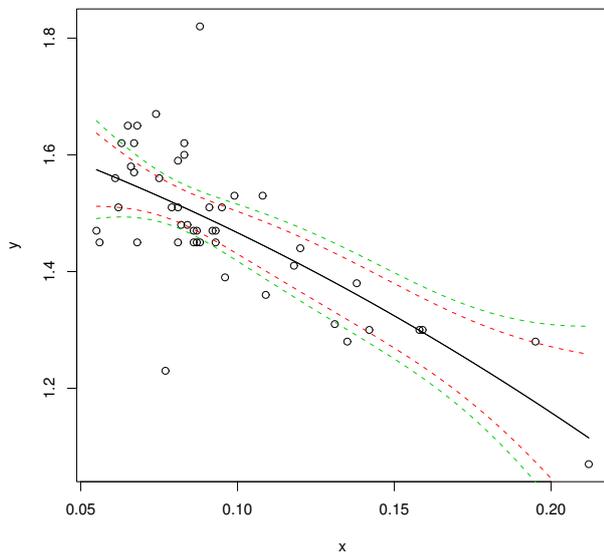
```



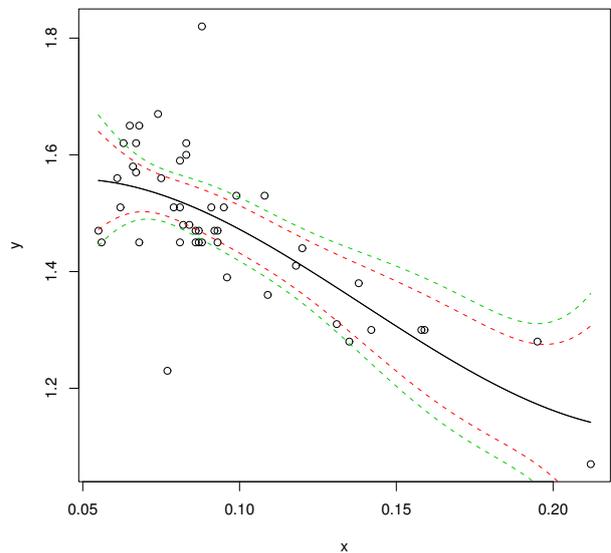
run0076-s0



run0076-s1



run0076-s2



run0076-s3

- 各次数  $p = 0, 1, 2, 3$  のグラフを描いた．予測値とその信頼区間も示した．赤は  $\alpha = 0.05$  ，緑は  $\alpha = 0.01$  である．
- 次数を上げると  $RSS = \|e\|^2$  (残差平方和) は小さくなる．決定係数  $R^2$  は大きくなる．いずれも，次数の増加とともに，回帰の当てはまりがよくなることを示唆する．
- ところがグラフを見ると，次数  $p = 1$  くらいで十分な感じ．いったい，次数はいくつにするのが適切なのか？
- 回帰係数の  $t$  検定の確率値を順番にみていく
  1. 次数  $p = 0$  のとき， $\beta_0 = 0$  を検定する確率値は，ほぼゼロ．つまり  $\beta_0 \neq 0$  と結論．つまり， $p \geq 0$  が示唆される．
  2. 次数  $p = 1$  のとき， $\beta_0 = 0$  を検定する確率値は，ほぼゼロ．つまり  $\beta_0 \neq 0$  と結論．同様に  $\beta_1 \neq 0$  と結論．つまり， $p \geq 1$  が示唆される．

3. 次数  $p = 2$  のとき,  $\beta_0 = 0$  を検定する確率値は, ほぼゼロ. つまり  $\beta_0 \neq 0$  と結論. ところが  $\beta_1 = 0$  を検定する確率値は 0.439 なので  $\beta_1 = 0$  と結論. 同様に  $\beta_2 = 0$  と結論. これは  $p = 0$  を示唆? 矛盾?
4. 次数  $p = 3$  のときは,  $\beta_0 \neq 0, \beta_1 = \beta_2 = \beta_3 = 0$  と結論. これは  $p = 0$  を示唆? 矛盾?

- この例からも分かるように, 回帰係数の  $t$ -検定は解釈に注意する.
- この場合は,  $p = 1$  と判断するのが適切. 次数  $p = 2$  のチェックのときは,  $\beta_2 = 0$  or  $\beta_2 \neq 0$  だけをチェックすべき.

### 3 回帰モデルの $F$ 検定

#### 3.1 部分モデル

- これまで  $p$  個の説明変数の重回帰モデルを考えた

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

ベクトル表現すると

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

- 最初の  $k$  個の説明変数だけを用いる重回帰モデルを考える (例: 多項式回帰で次数を  $p$  から  $k$  に変更する)

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k + \epsilon$$

ベクトル表現すると

$$\mathbf{y} = \mathbf{X}^{(k)}\boldsymbol{\beta}^{(k)} + \boldsymbol{\epsilon}$$

ただし

$$\mathbf{X}^{(k)} = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k), \quad \boldsymbol{\beta}^{(k)} = (\beta_0, \beta_1, \dots, \beta_k)'$$

- なお,

$$\mathbf{X}^{(-k)} = (\mathbf{x}_{k+1}, \mathbf{x}_{k+2}, \dots, \mathbf{x}_p), \quad \boldsymbol{\beta}^{(-k)} = (\beta_{k+1}, \beta_{k+2}, \dots, \beta_p)'$$

とおけば

$$\mathbf{X} = (\mathbf{X}^{(k)}, \mathbf{X}^{(-k)}), \quad \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}^{(k)} \\ \boldsymbol{\beta}^{(-k)} \end{bmatrix}$$

と分割してかける.

- 回帰係数  $\boldsymbol{\beta}^{(k)}$  の最小二乗推定は,

$$\hat{\boldsymbol{\beta}}^{(k)} = (\mathbf{X}^{(k)'} \mathbf{X}^{(k)})^{-1} \mathbf{X}^{(k)'} \mathbf{y}$$

$i = 0, \dots, k$  について, 一般に  $\hat{\boldsymbol{\beta}}^{(k)}$  の第  $i$  成分と  $\hat{\boldsymbol{\beta}}$  の第  $i$  成分は一致しない.

- $p$  個の説明変数をもつモデルにおいて,

$$\beta_{k+1} = \beta_{k+2} = \cdots = \beta_p = 0$$

と設定すれば,  $k$  個の説明変数をもつモデルになる. したがって, 後者は前者の「部分モデル」または「部分回帰」と呼ばれる.

- ここでは最初の  $k$  個の説明変数としたが, 添え字を付け替えることにより, 実際にはどの  $k$  個を選んでも, 同様な議論ができる.
- さらに以降の議論で本質的なのは,

$$\text{Span}(\mathbf{x}_0, \dots, \mathbf{x}_k) \subset \text{Span}(\mathbf{x}_0, \dots, \mathbf{x}_p)$$

ということなので, 説明変数の線形結合をとっても良い.

### 3.2 残差平方和と $F$ 検定

- 回帰モデル ( $k$ ) のハット行列

$$\mathbf{H}^{(k)} = \mathbf{X}^{(k)} (\mathbf{X}^{(k)'} \mathbf{X}^{(k)})^{-1} \mathbf{X}^{(k)'}$$

予測値のベクトル

$$\hat{\mathbf{y}}^{(k)} = \mathbf{H}^{(k)} \mathbf{y}$$

残差ベクトル

$$\mathbf{e}^{(k)} = \mathbf{y} - \hat{\mathbf{y}}^{(k)} = (\mathbf{I}_n - \mathbf{H}^{(k)}) \mathbf{y}$$

- 回帰モデル ( $k$ ) の残差平方和は

$$\text{RSS}^{(k)} = \|\mathbf{e}^{(k)}\|^2$$

である. もしモデル ( $k$ ) が正しければ,

$$\frac{\text{RSS}^{(k)}}{\sigma^2} \sim \chi_{n-k-1}^2$$

である. さらに残差平方和の差

$$\frac{\text{RSS}^{(k)} - \text{RSS}^{(p)}}{\sigma^2} = \frac{\|\mathbf{e}^{(k)}\|^2}{\sigma^2} - \frac{\|\mathbf{e}^{(p)}\|^2}{\sigma^2} \sim \chi_{p-k}^2$$

である. これと

$$\frac{\text{RSS}^{(p)}}{\sigma^2} \sim \chi_{n-p-1}^2$$

は互いに独立である.

- もしモデル ( $k$ ) が正しくない場合,  $\frac{\text{RSS}^{(k)}}{\sigma^2}$  は「非心カイ二乗分布」という分布に従う.
- モデル ( $k$ ) が正しければ,

$$F = \frac{(\text{RSS}^{(k)} - \text{RSS}^{(p)}) / (p - k)}{\text{RSS}^{(p)} / (n - p - 1)}$$

は互いに独立な自由度  $p - k$  のカイ二乗と自由度  $n - p - 1$  のカイ二乗の比であり, したがって, 自由度  $p - k, n - p - 1$  の  $F$  分布に従う.

$$F \sim F_{p-k, n-p-1}$$

- F分布の分布関数はRの関数 pf で計算できる．一般に  $X$  が自由度  $m, n$  の  $F$  分布に従うとき，この分布関数は

$$\Pr\{X \leq x\} = \text{pf}(x, m, n)$$

$$\Pr\{X > x\} = 1 - \text{pf}(x, m, n) = \text{pf}(x, m, n, \text{lower.tail} = \text{F})$$

pf の逆関数は qf である．

$$\Pr\{X \leq \text{qf}(a, m, n)\} = a$$

である．

```
# run0077.R
# F検定
fkentei <- function(fitk,fitp) {
  rssk <- sum(resid(fitk)^2) # RSS^(k)
  rssp <- sum(resid(fitp)^2) # RSS^(p)
  dfk <- df.residual(fitk) # n-k-1
  dfp <- df.residual(fitp) # n-p-1
  bunshi <- (rssk - rssp)/(dfk-dfp) # (RSS^(k)-RSS^(p))/(p-k)
  bunbo <- rssp/dfp # RSS^(p) / (n-p-1)
  fvalue <- bunshi/bunbo # F統計量
  pvalue <- pf(fvalue,dfk-dfp,dfp,lower.tail=F)
  list(fvalue=fvalue,df1=dfk-dfp,df2=dfp,pvalue=pvalue)
}
```

```
> source("run0077.R")
> # 多項式回帰
> dat <- read.table("dat0001.txt") # データの読み込み (47 x 2行列)
> x <- dat[,1]/100; y <- dat[,2]
> fit0 <- lm(y ~ 1, data.frame(x,y)) # k=0
> fit1 <- lm(y ~ x, data.frame(x,y)) # k=1
> fit2 <- lm(y ~ x + I(x^2), data.frame(x,y)) # k=2
> summary(fit2) # k=2のサマリ
```

Call:

```
lm(formula = y ~ x + I(x^2), data = data.frame(x, y))
```

Residuals:

Min	1Q	Median	3Q	Max
-0.29411	-0.04875	-0.00797	0.04600	0.32282

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.6807	0.1197	14.043	<2e-16 ***

```
x          -1.6725      2.1408  -0.781    0.439
I(x^2)     -4.6985      8.5762  -0.548    0.587
---
```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.09277 on 44 degrees of freedom

Multiple R-Squared: 0.5356, Adjusted R-squared: 0.5145

F-statistic: 25.37 on 2 and 44 DF, p-value: 4.7e-08

```
> unlist(fkentei(fit0,fit2)) # 上のF検定はコレ!
```

```
      fvalue      df1      df2      pvalue
2.537048e+01 2.000000e+00 4.400000e+01 4.700319e-08
```

```
> unlist(fkentei(fit1,fit2)) # 上のt検定(x^2の項)はコレ!
```

```
      fvalue      df1      df2      pvalue
0.3001376  1.0000000 44.0000000 0.5865649
```

```
> sqrt(0.3001376) # これがt統計量(x^2の項)の絶対値
```

```
[1] 0.5478482
```

- $F$  検定は、モデル ( $k$ ) からモデル ( $p$ ) に変更したときに、当てはまり (RSS 値) がどれだけ改善したかをチェックしている。
- 特によく利用されるのは次の 2 パターン
  1.  $k = 0$  とする。これは  $y = \text{定数}$  というモデルと比較して、モデル ( $p$ ) の回帰が意味あるのかどうかを調べる。これで  $F$  検定の  $p$ -value が  $< \alpha$  となり有意だとしても、 $p$  個の説明変数すべてが必要であるという意味ではない (その一部で十分かもしれない)。
  2.  $k = p - 1$  とする。これは説明変数  $x_p$  が必要かどうかをチェックし、回帰係数  $\beta_p = 0$  or  $\beta_p \neq 0$  を判定する。数学的に  $\beta_p$  の  $t$ -検定と等価。(おなじ確率値になる。) また、統計量も  $|t|^2 = F$  の関係がある。したがって、通常  $F$  検定ではなくて、 $t$ -検定として実装されている。

### 3.3 尤度比検定 \*

- 一般の確率モデルの比較では、 $F$  検定の変わりに尤度比検定が適用できる。
- 尤度比検定を重回帰モデルに適用すると、 $F$  検定とほぼ同じ結果になる。
- モデル 1 とモデル 2 は互いに包含関係 (ネスト) であり、モデル 2 はモデル 1 を一般化したものとする。モデル 1 はモデル 2 の特殊な場合になる。  
(重回帰モデルの例) モデル 1 = モデル ( $k$ ) , モデル 2 = モデル ( $p$ ) 。
- まず二つの確率モデルの対数尤度を  $l_1(\theta_1)$  ,  $l_2(\theta_2)$  とする。ここで  $\theta_1$  と  $\theta_2$  はパラメタベクトルで、次元はそれぞれ  $p_1 = \dim \theta_1, p_2 = \dim \theta_2$  とおく。  
(重回帰モデルの例)  $\theta_1 = (\beta_0, \beta_1, \dots, \beta_k, \sigma)$  ,  $\theta_2 = (\beta_0, \beta_1, \dots, \beta_p, \sigma)$  ,  $p_1 = k + 2$  ,  $p_2 = p + 2$  。

- パラメタの最尤推定を  $\hat{\theta}_1, \hat{\theta}_2$  , 最大対数尤度を  $\ell_1(\hat{\theta}_1)$  ,  $\ell_2(\hat{\theta}_2)$  と書く .  
(重回帰モデルの例)

$$\ell_1(\hat{\theta}_1) = \ell(\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_k, \hat{\sigma}) = -\frac{n}{2} \left\{ 1 + \log\left(2\pi \frac{\text{RSS}^{(k)}}{n}\right) \right\}$$

$$\ell_2(\hat{\theta}_2) = \ell(\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p, \hat{\sigma}) = -\frac{n}{2} \left\{ 1 + \log\left(2\pi \frac{\text{RSS}^{(p)}}{n}\right) \right\}$$

- モデル 1 が正しいとき , 対数尤度の差の 2 倍は , 近似的に カイ二乗分布に従う . 自由度は  $p_2 - p_1$  である (  $n$  が十分大きいとき , 近似がよくなる ) . つまり

$$2 \times (\ell_2(\hat{\theta}_1) - \ell_1(\hat{\theta}_2)) \sim \chi_{p_2-p_1}^2$$

(重回帰モデルの例)

$$2 \times (\ell_2(\hat{\theta}_1) - \ell_1(\hat{\theta}_2)) = n \log \text{RSS}^{(k)} - n \log \text{RSS}^{(p)} \sim \chi_{p-k}^2$$

```
# run0081.R
# 尤度比検定
yudohikentei <- function(fitk,fitp) {
  rssk <- sum(resid(fitk)^2) # RSS^(k)
  rssp <- sum(resid(fitp)^2) # RSS^(p)
  dfk <- df.residual(fitk) # n-k-1
  dfp <- df.residual(fitp) # n-p-1
  n <- length(resid(fitk)) # n
  chisqvalue <- n*(log(rssk)-log(rssp)) # 尤度比統計量
  pvalue <- pchisq(chisqvalue,dfk-dfp,lower.tail=F)
  list(chisqvalue=chisqvalue,df=dfk-dfp,n=n,pvalue=pvalue)
}
```

```
> dat <- read.table("dat0001.txt") # データの読み込み (47 x 2 行列)
> x <- dat[,1]/100; y <- dat[,2]
> fit0 <- lm(y ~ 1, data.frame(x,y)) # k=0
> fit1 <- lm(y ~ x, data.frame(x,y)) # k=1
> fit2 <- lm(y ~ x + I(x^2), data.frame(x,y)) # k=2
> source("run0081.R")
> unlist(fkentei(fit0,fit2)) # F 検定
      fvalue      df1      df2      pvalue
2.537048e+01 2.000000e+00 4.400000e+01 4.700319e-08
> unlist(yudohikentei(fit0,fit2)) # 尤度比検定
  chisqvalue      df      n      pvalue
3.604697e+01 2.000000e+00 4.700000e+01 1.487646e-08
> unlist(fkentei(fit1,fit2)) # F 検定
```

```

      fvalue      df1      df2      pvalue
0.3001376  1.0000000  44.0000000  0.5865649
> unlist(yudohikentei(fit1,fit2)) # 尤度比検定
chisqvalue      df      n      pvalue
0.3195130  1.0000000  47.0000000  0.5719005

```

### 3.4 QR 分解

- 最小二乗法の計算では、通常、QR 分解と呼ばれる行列の分解をしている。

$$X = QR$$

ここで、 $X$  はサイズ  $n \times (p+1)$ 、 $Q$  はサイズ  $n \times (p+1)$ 、 $R$  はサイズ  $(p+1) \times (p+1)$ 。  
 $Q$  の列は互いに直交していて、 $Q'Q = I_{p+1}$  である。 $R$  は上三角行列である。

$$X'X = R'Q'QR = R'R$$

$$(X'X)^{-1} = (R'R)^{-1} = R^{-1}R^{-1'}$$

- QR 分解は、 $\hat{\beta}$  の計算時に利用される。

$$\hat{\beta} = (X'X)^{-1}X'y = R^{-1}R^{-1'}R'Q'y = R^{-1}Q'y$$

まず  $Q'y$  を計算した後、 $\hat{\beta} = R^{-1}(Q'y)$  の計算時には  $R^{-1}$  を計算する必要はない。 $R$  が上三角であることを利用して、 $Q'y$  と  $R$  から直接  $\hat{\beta}$  を計算するアルゴリズムがあり、このほうが  $R^{-1}$  を経由するより演算数が少ないし、数値的にも安定する。

```

> X <- matrix(c(1^(1:5),2^(1:5),3^(1:5)),5,3)
> X
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    4    9
[3,]    1    8   27
[4,]    1   16   81
[5,]    1   32  243
> QR <- qr(X)
> R <- qr.R(QR)
> Q <- qr.Q(QR)
> R
      [,1]      [,2]      [,3]
[1,] -2.236068 -27.72724 -162.33854
[2,]  0.000000  24.39672  197.92824
[3,]  0.000000   0.00000   29.99355
> Q
      [,1]      [,2]      [,3]

```

```

[1,] -0.4472136 -0.4262868  0.4925791
[2,] -0.4472136 -0.3443086  0.1516455
[3,] -0.4472136 -0.1803521 -0.3301785
[4,] -0.4472136  0.1475608 -0.6936976
[5,] -0.4472136  0.8033866  0.3796515
> Q %*% R
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    1    4    9
[3,]    1    8   27
[4,]    1   16   81
[5,]    1   32  243
> t(Q) %*% Q
      [,1]      [,2]      [,3]
[1,] 1.000000e-00 -8.180305e-17  1.924459e-17
[2,] -8.180305e-17  1.000000e-00 -1.428436e-17
[3,]  1.924459e-17 -1.428436e-17  1.000000e-00

```

### 3.5 F 検定のつづき \*

- まえに議論した直交基底

$$U = (\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n)$$

をうまく定義すると,

$$H^{(k)} = (\mathbf{u}_1, \dots, \mathbf{u}_{k+1})(\mathbf{u}_1, \dots, \mathbf{u}_{k+1})', \quad k = 0, \dots, p$$

とできる.  $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_p$  を順番にグラム・シュミットの直交化すればよい. じつは QR 分解  $X = QR$  は, これと同等の計算をしている (ここでは述べないが, グラム・シュミットの直交化よりも効率の良いアルゴリズム).  $\mathbf{u}_i = \mathbf{q}_i, i = 1, \dots, p+1$ , のこりの  $\mathbf{u}_{p+2}, \dots, \mathbf{u}_n$  はこれらと直交する適当な基底を選べばよい.

- 回帰モデル ( $p$ ) で正規モデルを仮定する.

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

$$\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$$

これは

$$\mathbf{y} = X^{(k)}\boldsymbol{\beta}^{(k)} + X^{(-k)}\boldsymbol{\beta}^{(-k)} + \boldsymbol{\epsilon}$$

ともかけるので, 回帰モデル ( $k$ ) の残差ベクトルは

$$\mathbf{e}^{(k)} = (\mathbf{I}_n - H^{(k)})\mathbf{y} = (\mathbf{I}_n - H^{(k)})(X^{(-k)}\boldsymbol{\beta}^{(-k)} + \boldsymbol{\epsilon})$$

もし  $\boldsymbol{\beta}^{(-k)} = \mathbf{0}$  ならば

$$\mathbf{e}^{(k)} = (\mathbf{I}_n - H^{(k)})\mathbf{y} = (\mathbf{I}_n - H^{(k)})\boldsymbol{\epsilon} = (\mathbf{u}_{k+2}, \dots, \mathbf{u}_n)(\mathbf{u}_{k+2}, \dots, \mathbf{u}_n)'\boldsymbol{\epsilon}$$

したがって,  $z_i = \mathbf{u}'\epsilon/\sigma, i = 1, \dots, n$  とおけば,

$$\frac{\|\mathbf{e}^{(k)}\|^2}{\sigma^2} = z_{k+2}^2 + \dots + z_n^2$$

- 結局, モデル  $(k)$  が正しいとき,

$$\frac{\text{RSS}^{(k)}}{\sigma^2} = \frac{\|\mathbf{e}^{(k)}\|^2}{\sigma^2} = z_{k+2}^2 + \dots + z_n^2 \sim \chi_{n-k-1}^2$$

$$\frac{\text{RSS}^{(p)}}{\sigma^2} = \frac{\|\mathbf{e}^{(p)}\|^2}{\sigma^2} = z_{p+2}^2 + \dots + z_n^2 \sim \chi_{n-p-1}^2$$

$$\frac{\text{RSS}^{(k)} - \text{RSS}^{(p)}}{\sigma^2} = \frac{\|\mathbf{e}^{(k)}\|^2}{\sigma^2} - \frac{\|\mathbf{e}^{(p)}\|^2}{\sigma^2} = z_{k+2}^2 + \dots + z_{p+1}^2 \sim \chi_{p-k}^2$$

そして,  $\text{RSS}^{(p)}$  と  $\text{RSS}^{(k)} - \text{RSS}^{(p)}$  は, 互いに共通の  $z_i$  を含まないので, 独立.

## 4 予測式の信頼区間 (同時信頼区間)

### 4.1 回帰係数ベクトルの同時信頼域

- 復習

1. 正規回帰モデルを仮定する. 回帰係数の最小二乗推定  $\hat{\beta}$  は多変量正規分布に従う.

$$\hat{\beta} \sim N_{p+1}(\beta, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$$

2. 誤差分散  $\sigma^2$  の不偏推定  $S_\epsilon^2$  は

$$\frac{(n-p-1)S_\epsilon^2}{\sigma^2} \sim \chi_{n-p-1}^2$$

であり, これは  $\hat{\beta}$  とは独立である.

- ここで, 適当な  $(p+1) \times (p+1)$  行列  $\mathbf{R}$  を選ぶと,

$$\mathbf{X}'\mathbf{X} = \mathbf{R}'\mathbf{R}$$

とできることに注意する. 具体的には, 前に説明した QR 分解  $\mathbf{X} = \mathbf{Q}\mathbf{R}$  から得られる  $\mathbf{R}$  でよい.

$$(\mathbf{X}'\mathbf{X})^{-1} = (\mathbf{R}'\mathbf{R})^{-1} = \mathbf{R}^{-1}\mathbf{R}'^{-1}$$

- $p+1$  ベクトル  $\mathbf{z} = (z_1, z_2, \dots, z_{p+1})'$  を

$$\mathbf{z} = \frac{1}{\sigma}\mathbf{R}(\hat{\beta} - \beta)$$

で定義する.  $\mathbf{z}$  は多変量正規分布に従い, 平均は  $E(\mathbf{z}) = \mathbf{0}$ , 分散は

$$V(\mathbf{z}) = \frac{1}{\sigma}\mathbf{R}\sigma^2(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}'\frac{1}{\sigma} = \mathbf{R}\mathbf{R}^{-1}\mathbf{R}^{-1'}\mathbf{R}' = \mathbf{I}_{p+1}$$

従って,

$$\mathbf{z} \sim N_{p+1}(\mathbf{0}, \mathbf{I}_{p+1})$$

成分で書けば,

$$z_1, z_2, \dots, z_{p+1} \sim N(0, 1)$$

でこれらは互いに独立.

- 以上より,

$$F = \frac{\|z\|^2/(p+1)}{S_\epsilon^2/\sigma^2} \sim F_{p+1, n-p-1}$$

は自由度  $p+1, n-p-1$  の F 分布に従う.  $\|z\|^2 = \|\mathbf{X}(\hat{\beta} - \beta)\|^2/\sigma^2$  を代入すると,

$$F = \frac{\|\mathbf{X}(\hat{\beta} - \beta)\|^2}{(p+1)S_\epsilon^2}$$

である.

- $\hat{\beta}$  の信頼域 (信頼係数 or 信頼度 =  $1 - \alpha$ ) を作るには次のように考える.

$$\text{qf}(a) = \text{qf}(a, p+1, n-p-1)$$

と置けば,

$$\Pr\{F \leq \text{qf}(1-\alpha)\} = 1 - \alpha$$

である. そこで

$$\text{信頼域}(1-\alpha) = \left\{ \beta : (\beta - \hat{\beta})'(\mathbf{X}'\mathbf{X})(\beta - \hat{\beta}) \leq (p+1)S_\epsilon^2 \text{qf}(1-\alpha) \right\}$$

と定めれば, これは  $\hat{\beta}$  を中心とする楕円体である. そして

$$\Pr\{\beta \in \text{信頼域}(1-\alpha)\} = 1 - \alpha$$

である.

- 信頼域を実際にグラフ描いてみるには,

$$\gamma = \mathbf{R}\beta, \quad \hat{\gamma} = \mathbf{R}\hat{\beta}$$

と回帰係数を変数変換して  $\gamma$  のほうで考えたほうが分かりやすい. それを  $\beta = \mathbf{R}^{-1}\gamma$  で戻せばよい.  $\gamma$  の信頼域は

$$\{\gamma : \|\gamma - \hat{\gamma}\|^2 \leq (p+1)S_\epsilon^2 \text{qf}(1-\alpha)\}$$

であるから  $\hat{\gamma}$  を中心とする半径  $S_\epsilon \sqrt{(p+1)\text{qf}(1-\alpha)}$  の球体.

```
# run0078.R
# 回帰係数の信頼域: 単回帰
# x=説明変数のベクトル, y=目的変数のベクトル
source("run0075.R")
p <- 1 # 単回帰なので次数=1
n <- length(y) # サンプルサイズ
alpha <- 0.05 # 信頼係数=1-alpha
a <- func0075a(x,p) # 回帰分析の準備
a$QR <- qr(a$X); a$R <- qr.R(a$QR) # QR分解
a$IR <- solve(a$R) # R^-1
b0 <- func0075b(y,alpha,a) # 回帰係数など
```

```

th <- seq(0,2*pi,len=300) # 描画の範囲 (0..2*pi を 300 等分)
ga <- b0$se*calcq1(n,p,alpha)*rbind(cos(th),sin(th)) # gamma でみた円
be <- as.vector(b0$be) + a$IR %*% ga # beta の信頼域 (楕円)
plot(t(be),type="l") # 楕円
points(t(b0$be)) # 中心
abline(v=b0$beconf[1,],lty=2,col=2) # beta0 の信頼区間 (赤)
abline(h=b0$beconf[2,],lty=2,col=2) # beta1 の信頼区間 (赤)
b1 <- func0075b(y,alpha,a,calcq1) # calcq1 を使ってもう一度
abline(v=b1$beconf[1,],lty=3,col=4) # beta0 の同時信頼区間 (青)
abline(h=b1$beconf[2,],lty=3,col=4) # beta1 の同時信頼区間 (青)
dev.copy2eps(file="run0078-b.eps")
coef <- cbind(b0$be,b0$beconf,b1$beconf)
colnames(coef) <- c("Estimate","Lo","Up","JointLo","JointUp")
print(coef)

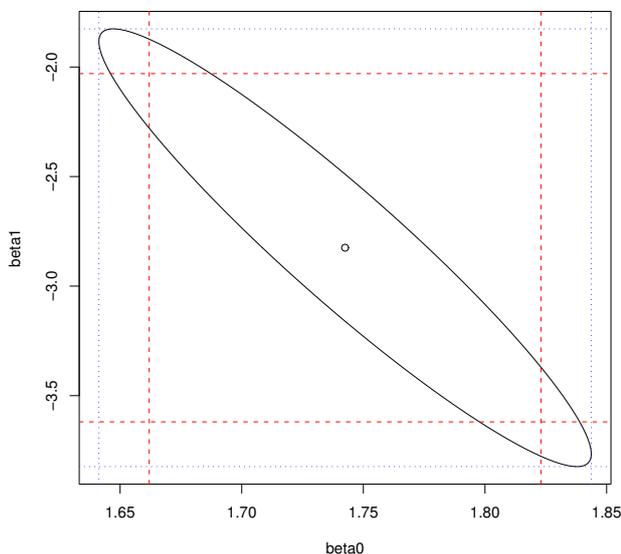
```

```

> dat <- read.table("dat0001.txt") # データの読み込み (47 x 2 行列)
> x <- dat[,1]/100; y <- dat[,2]
> source("run0078.R")

```

	Estimate	Lo	Up	JointLo	JointUp
beta0	1.742483	1.661974	1.822993	1.641291	1.843676
beta1	-2.824869	-3.619719	-2.030019	-3.823917	-1.825821



run0078-b

- 単回帰の例 :  $\beta = (\beta_0, \beta_1)'$  の同時信頼域 (楕円) を描いた . 以下すべて  $\alpha = 0.05$  とした .

$$\Pr\{\beta \in \text{同時信頼域}\} = 1 - \alpha$$

- 各回帰係数  $\beta_i, i = 0, 1$  の信頼区間を直線で示した (赤) . これは各係数を別々に見て信頼区間を  $t$ -検定から得ている . つまり ,

$$\Pr\{\beta_0 \in \text{信頼区間}_0\} = 1 - \alpha$$

$$\Pr\{\beta_1 \in \text{信頼区間}_1\} = 1 - \alpha$$

であるが、一般に

$$\Pr\{\beta_0 \in \text{信頼区間}_0, \beta_1 \in \text{信頼区間}_1\} < 1 - \alpha$$

となる可能性がある。

- 各係数について、同時信頼域の最小値、最大値を直線（青）で示した。これを「同時信頼区間」と呼ぶと

$$\Pr\{\beta_0 \in \text{同時信頼区間}_0, \beta_1 \in \text{同時信頼区間}_1\} \geq 1 - \alpha$$

である。 $\beta$ の集合としてみると、同時信頼域より大きくなってしまっている。

## 4.2 回帰式の信頼区間

- 「回帰式の信頼区間」、言い換えると、「予測値の同時信頼区間」を求める。一般には、回帰係数の線形結合  $v = \mathbf{w}'\beta$  の同時信頼区間を考える。
- 同時ではない、個別の信頼区間は以前求めた。

$$\text{信頼区間}(1 - \alpha) = [\hat{v} - \hat{\sigma}_v \text{qt}(1 - \alpha/2), \hat{v} + \hat{\sigma}_v \text{qt}(1 - \alpha/2)]$$

ただし

$$\hat{\sigma}_v = S_\epsilon \sqrt{\mathbf{w}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{w}}$$

$$\text{qt}(a) = \text{qt}(a, n-p-1)$$

この信頼区間は

$$\Pr\{v \in \text{信頼区間}(1 - \alpha)\} = 1 - \alpha$$

を満たす。

- ここで求めたい「同時信頼区間」は、 $\mathbf{w}$ に関する任意の集合  $W$  に対して、

$$\Pr\{\mathbf{w}'\beta \in \text{同時信頼区間}_w(1 - \alpha), \mathbf{w} \in W\} \geq 1 - \alpha$$

となるようなものである。

- (例) 前節で求めた、回帰係数  $\beta_0$  と  $\beta_1$  は集合  $W$  の要素が二つの場合に相当する。そこで求めた同時信頼区間は

$$\Pr\{\beta_0 \in \text{同時信頼区間}_0, \beta_1 \in \text{同時信頼区間}_1\} \geq 1 - \alpha$$

を満たしていた。

- 導出は後ほど述べるが、結論としては、次のように計算すればよい。

$$\text{同時信頼区間}_{w(1-\alpha)} = [\hat{v} - \hat{\sigma}_v \sqrt{(p+1)qf(1-\alpha)}, \hat{v} + \hat{\sigma}_v \sqrt{(p+1)qf(1-\alpha)}]$$

ただし

$$qf(a) = qf(a, p+1, n-p-1)$$

である。

```
# run0079.R
# 回帰係数と予測値の同時信頼区間：多項式回帰
# x=説明変数のベクトル, y=目的変数のベクトル, p=多項式の次数, alpha も .
source("run0075.R")
func0079 <- function(x,y,a,alpha,output=T) {
  b0 <- func0075b(y,alpha,a)
  b1 <- func0075b(y,alpha,a,calcq1)
  coef <- cbind(b0$be,b0$bese,b0$tval,b0$pval,b0$beconf,b1$beconf)
  colnames(coef) <- c("Estimate","StdErr","t-value","p-value",
                    "Lo","Up","JointLo","JointUp")
  if(output) {
    func0075c(x,y,a,b0); func0075c(x,y,a,b1,col=4,lty=3,add=T)
    cat("RSS=",b0$rss," RSQ=",b0$rsq,"\n")
    print(round(coef,3))
  }
  list(b0=b0,b1=b1,coef=coef)
}
func0079b <- function(x,y,p,alpha,filename="run0079-") {
  for(i in 0:p) {
    cat("# 次数=",i,"\n")
    a <- func0075a(x,i)
    func0079(x,y,a,alpha)
    if(!is.null(filename))
      dev.copy2eps(file=paste(filename,"s",i,".eps",sep=""))
  }
}
```

```
> source("run0079.R")
> dat <- read.table("dat0001.txt") # データの読み込み (47 x 2行列)
> x <- dat[,1]/100; y <- dat[,2]
> p <- 3; alpha <- 0.05
> func0079b(x,y,p,alpha)
# 次数= 0
RSS= 0.815383 , RSQ= 0
```

	Estimate	StdErr	t-value	p-value	Lo	Up	JointLo	JointUp
beta0	1.473	0.019	75.848	0	1.434	1.512	1.434	1.512

# 次数= 1

RSS= 0.3812667 , RSQ= 0.5324078

	Estimate	StdErr	t-value	p-value	Lo	Up	JointLo	JointUp
beta0	1.742	0.040	43.592	0	1.662	1.823	1.641	1.844
beta1	-2.825	0.395	-7.158	0	-3.620	-2.030	-3.824	-1.826

# 次数= 2

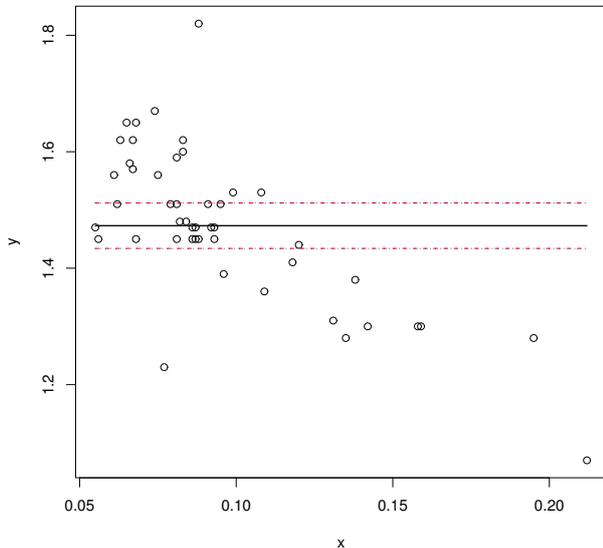
RSS= 0.3786836 , RSQ= 0.5355758

	Estimate	StdErr	t-value	p-value	Lo	Up	JointLo	JointUp
beta0	1.681	0.120	14.043	0.000	1.440	1.922	1.333	2.029
beta1	-1.672	2.141	-0.781	0.439	-5.987	2.642	-7.895	4.550
beta2	-4.698	8.576	-0.548	0.587	-21.983	12.586	-29.628	20.231

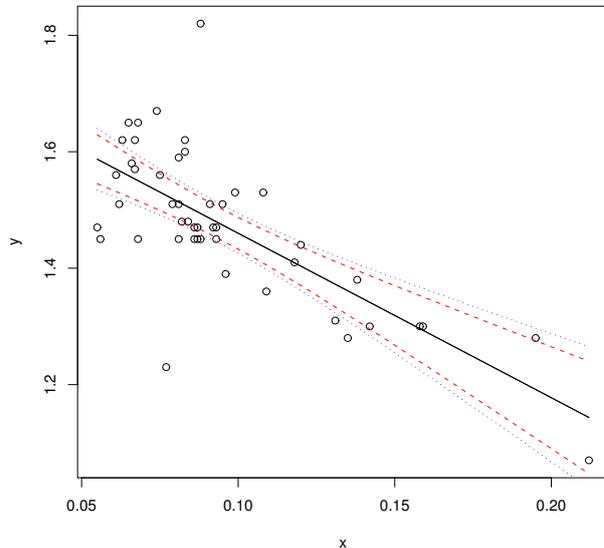
# 次数= 3

RSS= 0.3747561 , RSQ= 0.5403926

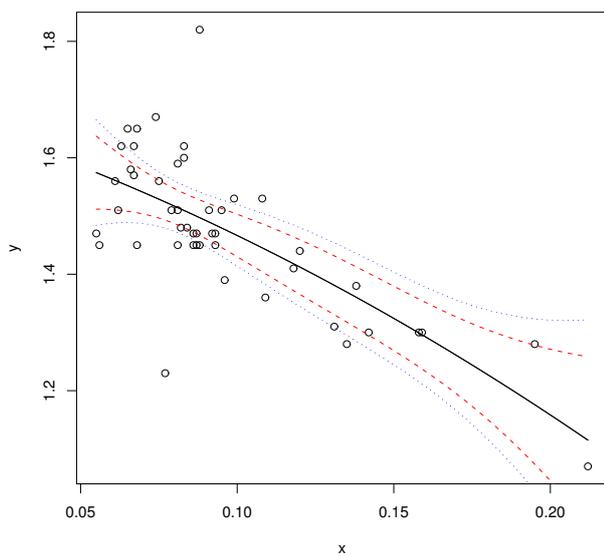
	Estimate	StdErr	t-value	p-value	Lo	Up	JointLo	JointUp
beta0	1.462	0.347	4.212	0.000	0.762	2.162	0.345	2.579
beta1	4.415	9.320	0.474	0.638	-14.381	23.211	-25.578	34.407
beta2	-56.680	77.913	-0.727	0.471	-213.807	100.447	-307.403	194.042
beta3	135.499	201.844	0.671	0.506	-271.559	542.557	-514.031	785.029



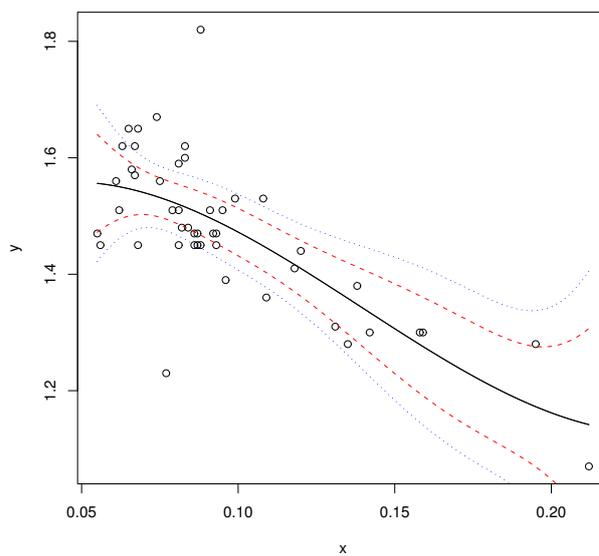
run0079-s0



run0079-s1



run0079-s2



run0079-s3

- すべて  $\alpha = 0.05$  , つまり信頼係数 95%である .
- 回帰係数の信頼区間 (Up,Lo) と同時信頼区間 (JointLo,JointUp)
- 予測値の信頼区間 (赤) と同時信頼区間 (青)

#### 4.3 同時信頼区間の導出 \*

- $v = \mathbf{w}'\boldsymbol{\beta}$ ,  $\mathbf{w} \in W$  の同時信頼区間を求める
- $\boldsymbol{\gamma} = \mathbf{R}\boldsymbol{\beta}$ ,  $\mathbf{a} = \mathbf{R}'^{-1}\mathbf{w}$  と変換すると ,  $v = \mathbf{w}'\boldsymbol{\beta} = \mathbf{a}'\boldsymbol{\gamma}$  とかける . このほうが , 扱いやすい .
- シュバルツ (Schwartz) の不等式より

$$|\hat{v} - v| = |\mathbf{a}'(\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma})| \leq \|\mathbf{a}\| \cdot \|\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}\|$$

で等号は  $\mathbf{a}$  と  $\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}$  が平行のときのみ .

- $\boldsymbol{\gamma}$  の同時信頼域 (球体) は

$$\left\{ \boldsymbol{\gamma} : \|\boldsymbol{\gamma} - \hat{\boldsymbol{\gamma}}\| \leq S_\epsilon \sqrt{(p+1)\text{qf}(1-\alpha)} \right\}$$

であったから ,  $\boldsymbol{\gamma}$  がこの同時信頼域に入っていれば ,

$$|\hat{v} - v| \leq \|\mathbf{a}\| S_\epsilon \sqrt{(p+1)\text{qf}(1-\alpha)}$$

である .  $\boldsymbol{\gamma}$  が同時信頼域に入る確率は  $1 - \alpha$  なので ,

$$\Pr \left\{ |\hat{v} - v| \leq \|\mathbf{a}\| S_\epsilon \sqrt{(p+1)\text{qf}(1-\alpha)} \right\} \geq 1 - \alpha$$

である .

- ここで  $\|\mathbf{a}\| = \sqrt{\mathbf{w}'\mathbf{R}^{-1}\mathbf{R}'^{-1}\mathbf{w}} = \sqrt{\mathbf{w}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{w}}$  に注意すれば ,

$$\text{同時信頼区間}_{\mathbf{w}} = \left\{ v : |\hat{v} - v| \leq S_\epsilon \sqrt{\mathbf{w}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{w}} \sqrt{(p+1)\text{qf}(1-\alpha)} \right\}$$

## 4.4 シミュレーションで確認

```
# run0080.R
# 信頼区間, 同時信頼区間: シミュレーション
# run0074.R のシミュレーション結果を利用する.
# run0075.R, run0079.R の関数をつかって信頼区間を計算する.
source("run0079.R")
func0080a <- function(v) { # check if v1 in [v2,v3]
  (v[1] >= v[2]) && (v[1] <= v[3])
}
func0080b <- function(be,y00,d) {
  coef <- cbind(be,d$coef)
  be0 <- apply(coef[,c("be","Lo","Up")],1,func0080a)
  be1 <- apply(coef[,c("be","JointLo","JointUp")],1,func0080a)
  pred0 <- all(apply(cbind(y00,d$b0$pred0conf),1,func0080a))
  pred1 <- all(apply(cbind(y00,d$b1$pred0conf),1,func0080a))
  list(be0=be0,be1=be1,pred0=pred0,pred1=pred1)
}
a <- func0075a(x,p) # 準備
y00 <- a$X0 %*% be # 300 分割した点に対応する真の y
cat("# 一回目のシミュレーション結果のチェック\n")
d <- func0079(x,simy[,1],a,alpha) # 信頼区間等の計算
abline(be,col=3,lty=4) # 真の回帰直線
cat("# 信頼区間に入っていたか?\n")
yesno1 <- unlist(func0080b(be,y00,d)); print(yesno1)
dev.copy2eps(file="run0080-s1.eps")
cat("# シミュレーション結果のチェック\n")
yesno <- matrix(0,length(yesno1),b) # 結果をしまうアレイ
rownames(yesno) <- names(yesno1)
for(i in 1:b) {
  d <- func0079(x,simy[,i],a,alpha,output=F)
  yesno[,i] <- unlist(func0080b(be,y00,d))
}
print(yesno[,1:5])
cat("# シミュレーションで信頼区間に入っていた回数\n")
print(apply(yesno,1,sum))
```

```
> n <- 11 # データサイズ
> be <- c(0,1) # 真の回帰係数 (beta0,beta1)
> x <- seq(0,1,len=n) # xを決める
> filename <- NULL
> source("run0074.R")
```

```

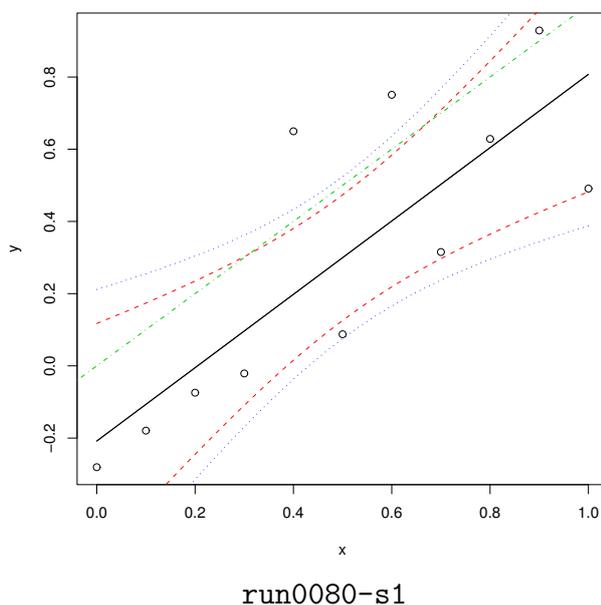
# start simulation: Wed Oct 6 11:46:43 2004
# end simulation: Wed Oct 6 11:46:45 2004
# 1回目のシミュレーション結果
$be
      [,1]
beta0 -0.1491011
beta1  1.3627077
$se2
[1] 0.04218202
$tval
      [,1]
beta0 -1.287003
beta1  6.958817
$pval
      [,1]
beta0 2.302066e-01
beta1 6.619553e-05

# pval0 < 0.05 の回数 = 508
# pval1 < 0.05 の回数 = 8751

> alpha <- 0.05
> source("run0080.R")
# 一回目のシミュレーション結果のチェック
RSS= 0.5859556 , RSQ= 0.6594943
      Estimate StdErr t-value p-value      Lo      Up JointLo JointUp
beta0  -0.208  0.144  -1.447  0.182 -0.534  0.117  -0.628  0.212
beta1   1.016  0.243   4.175  0.002  0.465  1.566   0.306  1.726
# 信頼区間に入っていたか?
be0.beta0 be0.beta1 be1.beta0 be1.beta1      pred0      pred1
      TRUE      TRUE      TRUE      TRUE      FALSE      TRUE
# シミュレーション結果のチェック
      [,1] [,2] [,3] [,4] [,5]
be0.beta0  1  1  1  1  1
be0.beta1  1  1  1  0  1
be1.beta0  1  1  1  1  1
be1.beta1  1  1  1  1  1
pred0      0  1  1  0  1
pred1      1  1  1  1  1
# シミュレーションで信頼区間に入っていた回数
be0.beta0 be0.beta1 be1.beta0 be1.beta1      pred0      pred1
      9498      9453      9835      9825      8736      9519

```

```
> sum(yesno[1,] & yesno[2,]) # joint0
[1] 9235
> sum(yesno[3,] & yesno[4,]) # joint1
[1] 9750
```



- シミュレーションなので回帰係数の真値  $\beta_0 = 0, \beta_1 = 1$  が分かっている．これが信頼区間に何回入ったかを数える．シミュレーションは 10000 回の繰り返しなので，信頼係数 = 95% ならば，9500 回程度が理想的．

- 個別の（同時でない）信頼区間

$$\#\{\beta_0 \in \text{信頼区間}_0\} = 9498$$

$$\#\{\beta_1 \in \text{信頼区間}_1\} = 9453$$

であるから，個別に  $\beta_0$  と  $\beta_1$  を見るかぎりほぼ 95%．しかし，

$$\#\{\beta_0 \in \text{信頼区間}_0, \beta_1 \in \text{信頼区間}_1\} = 9235$$

となり，同時に入るのは約 92% に低下する．

- 同時信頼区間

$$\#\{\beta_0 \in \text{同時信頼区間}_0, \beta_1 \in \text{同時信頼区間}_1\} = 9750$$

となり，同時に入るのは約 98% になる．

- 予測値の信頼区間各  $x$  で個別に見れば，

$$\#\{\beta_0 + \beta_1 x \in \text{信頼区間}(x),\} = \text{約 } 9500, \quad \text{すべての } x$$

のはずである．ところが

$$\#\{\beta_0 + \beta_1 x \in \text{信頼区間}(x), \text{ すべての } x\} = 8736$$

なので，予測式としてみると信頼区間に入る確率は約 87% となり，95% よりかなり低下する．

- 予測値の同時信頼区間

$$\#\{\beta_0 + \beta_1 x \in \text{同時信頼区間}(x), \text{すべての } x\} = 9519$$

なので、予測式としてみると信頼区間に入る確率は約 95% となり理想的。

## 5 課題

### 5.1 課題 7-1

X2000 データセットから自由に変数を選び多項式回帰を次数  $p = 1, 2, 3$  について行え。R に組み込み関数や講義で説明した関数など自由に用いてよい。自分で書いたプログラムはレポートに添付すること。

- それぞれの次数について、決定係数、および各回帰係数の推定値、標準誤差、t 統計量、 $p$ -値を示せ。
- 結局、どの次数  $p$  を用いるのが最も適切かを判断せよ。
- 上で選んだ次数について  $x, y$  の散布図上に推定した回帰直線（曲線）とその 95% 信頼区間（赤線）、および 95% 同時信頼区間（青線）を重ねて描け。

### 5.2 課題 7-2

run0075.R では次の二つの関数を定義している。

```
calcq0 <- function(n,p,alpha) qt(1-alpha/2,n-p-1) # 個別の信頼区間
calcq1 <- function(n,p,alpha) sqrt((p+1)*qf(1-alpha,p+1,n-p-1)) # 同時信頼区間
```

$n = 30, \alpha = 0.05$  とおき、 $\text{calcq0}(n,p,\alpha)$  と  $\text{calcq1}(n,p,\alpha)$  を  $p = 0, 1, \dots, 10$  の範囲で計算して重ねてプロットせよ（横軸= $p$ 、縦軸=関数値とする）。これらの関数値を比較せよ。この違いは回帰直線（曲面）の信頼区間、同時信頼区間についてどのような結果をもたらすかを説明せよ。