

# 統計学 2021 Lecture 11: 探索的データ解析-回帰分析入門-

北門 利英 (東京海洋大学海洋生物資源学科)

2022年6月29日

## 目次

1	線形回帰分析とは	1
1.1	モデルの定式化 . . . . .	2
2	気温上昇率の推定	3
2.1	データの読み込みと図示 . . . . .	3
2.2	線形回帰分析 . . . . .	4
2.3	予測能力の評価 . . . . .	5
2.4	演習 . . . . .	6
3	アロメトリー式の推定	7
3.1	体長-体重関係のアロメトリー式 . . . . .	7
3.2	ニジマスの観測値 . . . . .	7
3.3	ニジマスのアロメトリー式の推定 . . . . .	8
4	魚体中の PCB 蓄積量の経年変化	12
4.1	データの読み込みと図示 . . . . .	12
4.2	パラメータの推定 . . . . .	13
4.3	演習 . . . . .	16
5	魚の成長式の推定 (付録, 興味があればご一読下さい)	16

## 1 線形回帰分析とは

「ある値  $x$  を観測することで別の値  $y$  を予測できないか」、あるいは「 $y$  の値が  $x$  に依存して決まるかどうかを科学的に検討できないか」という問題を考えよう。例えば、

- 魚の体長  $x$  を観測すれば、(体重を測ることなく) 魚の体重  $y$  が予測できるようにできないか？
- ある年のサケの放流数  $x$  から、2年後の回帰数  $y$  を予測できるようにできないか？
- 気温は年々上昇傾向にあると言えるか？

• ...

2つの値  $x$  と  $y$  の関係があったとしても、必ずしも関係が1次式のように単純であるとは限らないが、ここでは最も簡単な1次元の線形回帰分析と呼ばれる方法について説明します。

## 1.1 モデルの定式化

いま、2つの値  $x$  と  $y$  に次のような関係が想定できるとします。

$$y = \alpha + \beta x$$

この式が正しいければ、 $x$  を観測することで  $y$  の値を予測できることを意味します。このような式を回帰式と呼びますが、回帰式を用いて予測するためにはパラメータ  $\alpha, \beta$  を定める必要があります。そこで、 $(x, y)$  のペアの観測値を用いて、それらを求めることになります。

ここで、以下のような観測モデルを考えてみましょう。

$$y_i = \alpha + \beta x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2) \quad (i = 1, 2, \dots, n)$$

これを1次元の正規線形回帰モデルとよびます。ここで、添え字  $i$  は、 $x$  と  $y$  の観測値が  $n$  組あるのでそれらを区別するために添えています。観測データは必ずしも単純な回帰式にピッタリ一致するわけではないので、このずれを誤差項  $\varepsilon_i$  で記述しています。ただ、このようなずれが最も小さくなるように  $\alpha, \beta$  を決めることができれば、最も予測力の高い回帰式を得ることができます。

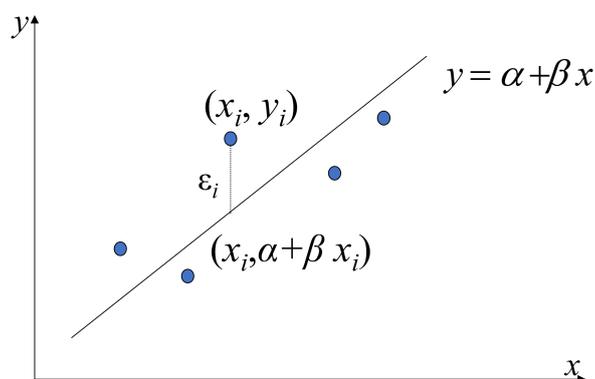


図1: 最小2乗法のイメージ図

そこで、未知パラメータ  $\alpha, \beta$  を観測値に基づいて推定する方式として、以下のような2乗誤差の和を考えます。

$$S(\alpha, \beta) = \sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n \{y_i - (\alpha + \beta x_i)\}^2$$

この2乗誤差の和を最小にする  $\alpha, \beta$  を求める方法を最小2乗法とよびます。この解は、 $S(\alpha, \beta)$  を  $\alpha, \beta$  で偏微分して0とおいた連立方程式の解(停留点)が極小値となる性質を利用し、

$$\hat{\alpha} = \bar{y} - \hat{\beta} \bar{x}, \quad \hat{\beta} = \frac{\sum_{i=1}^n x_i y_i - n \bar{x} \bar{y}}{\sum_{i=1}^n x_i^2 - n \bar{x}^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

のように求められます。また、誤差項の分散  $\sigma^2$  も次の式で求められます。

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum_{i=1}^n (y_i - \hat{\alpha} - \hat{\beta}x_i)^2$$

このようにして求めた回帰モデルのパラメータの推定値  $\hat{\alpha}, \hat{\beta}$  を用いれば、以下の予測式が作成できます。このような手法を 1 次元の線形回帰分析とよびます。

$$y = \hat{\alpha} + \hat{\beta}x$$

## 2 気温上昇率の推定

Lecture02 で気温のデータの図示をしました。ここではこれまでの気温の変化の情報を基に、今後の気温の予測を試みましょう。実際にはこのような単純な式で予測することは危険ではありますが、あくまで演習用の題材として理解ください。

### 2.1 データの読み込みと図示

```
Data_Tokyo <- read.csv("Data_Tokyo.csv", header=T) #CSVファイルの読み込み
names(Data_Tokyo)      #Data_Tokyo のオブジェクトの構成要素を確認
```

```
[1] "Year"      "Month"      "Temperature" "Rainfall"
```

```
dim(Data_Tokyo)      #Data_Tokyo の配列次元の確認 1392×4 の行列であることが分かる
```

```
[1] 1440    4
```

データは Data\_Tokyo という名前のオブジェクトに保存されています。いま、1月の気温だけを取り出して Data01 としましょう。

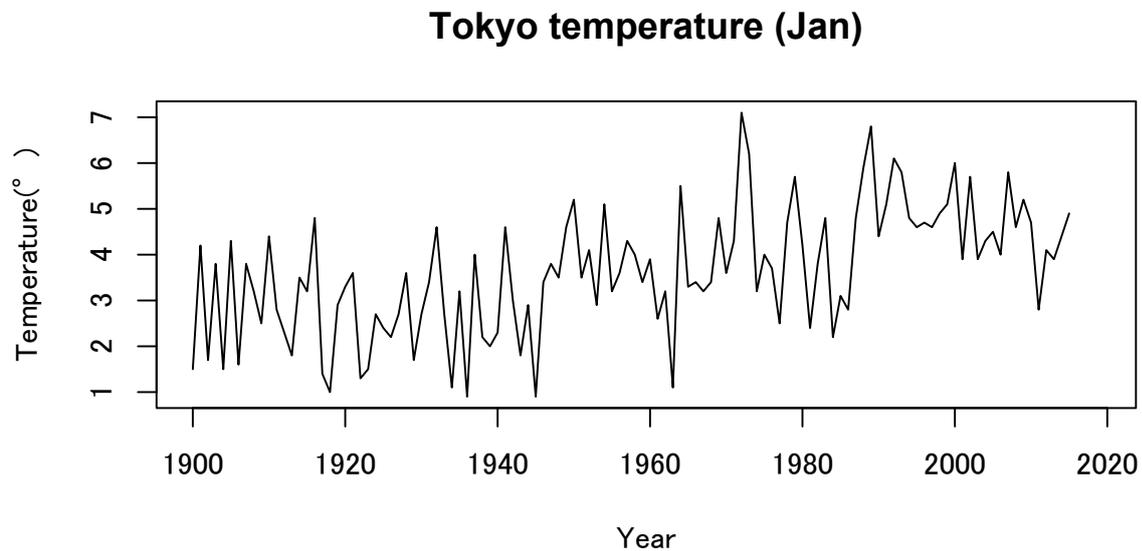
```
Data <- Data_Tokyo[Data_Tokyo$Month==1, ]
head(Data, 3)
```

```
   Year Month Temperature Rainfall
1  1900     1         1.5     69.2
13 1901     1         4.2     74.4
25 1902     1         1.7     33.4
```

```
tail(Data, 3)
```

```
   Year Month Temperature Rainfall
1405 2017     1          NA        NA
1417 2018     1          NA        NA
1429 2019     1          NA        NA
```

```
# 横軸に年, 縦軸に気温, それを "line" で結ぶ
plot(Data$Year, Data$Temperature, type="l",
      main="Tokyo temperature (Jan)", xlab="Year", ylab="Temperature(°)")
```



## 2.2 線形回帰分析

線形回帰分析を実行するには, “lm” という関数を用います. 実際に解析を実行してみると, 次のような結果を得ます.

```
Res <- lm(Temperature~Year, data=Data)
summary(Res)
```

Call:

```
lm(formula = Temperature ~ Year, data = Data)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.65414	-0.72097	-0.02539	0.64632	3.14301

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-40.49014	6.08909	-6.650	1.06e-09 ***
Year	0.02254	0.00311	7.247	5.44e-11 ***

---

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

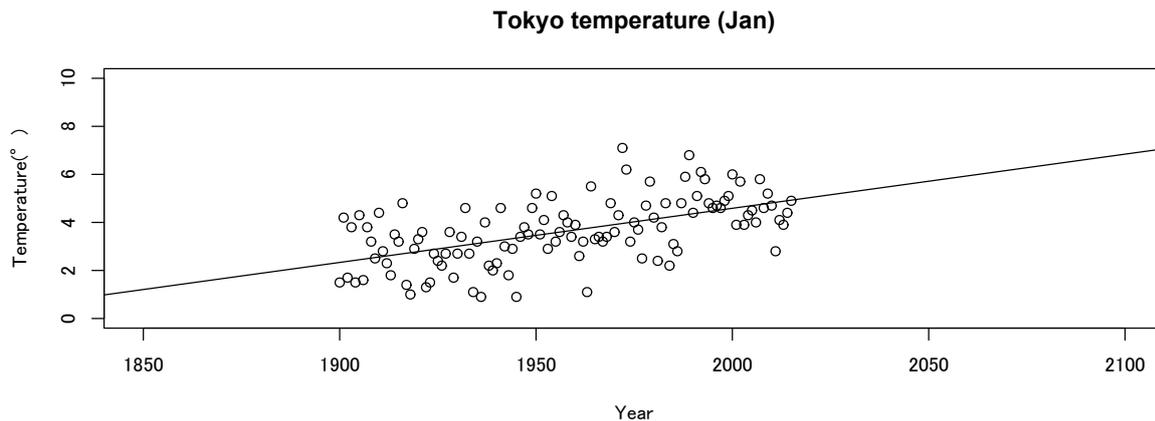
Residual standard error: 1.122 on 114 degrees of freedom

(4 observations deleted due to missingness)

Multiple R-squared: 0.3154, Adjusted R-squared: 0.3094

F-statistic: 52.52 on 1 and 114 DF, p-value: 5.442e-11

```
plot(Data$Year, Data$Temperature, xlim=c(1850,2100), ylim=c(0,10),
      main="Tokyo temperature (Jan)", xlab="Year", ylab="Temperature(°)")
abline(Res)
```



この結果,  $\hat{\alpha} = -40.49$ ,  $\hat{\beta} = 0.02254$  という値を得ますから, 回帰式として  $(1) = -40.49 + 0.02254( )$  という予測式ができたこととなります. これが本当だとすると, 1年あたり 0.02254度上昇していることとなります. 仮に西暦 2100 年を代入すると 6.844 度となります. ただし, 西暦 = 0 年を代入すると気温がマイナス 40 度になりますので, さすがにそれはおかしいですね. ですので, データの範囲外 (この場合は 1900 年以前, および 2020 年以降) にそのまま適用して予測することは危険ですが, これまでの傾向および直近の予測には使えると思います.

ちなみに Year に対する係数  $\beta$  の P 値が 0.05 よりも小さくなっています. これは帰無仮説  $\beta = 0$  に対する仮説検定の結果で, 有意に気温が上昇していることが分かります.

## 2.3 予測能力の評価

仮にデータを 1980 年までしか用いずに, 同じ解析をしてみても, 回帰式の係数  $\alpha, \beta$  が大きく変化してしまうか,あるいは 2019 年までのデータを用いた場合とあまり変わらないか検討してみると, それほど変わらないことが分かります. したがって, 1980 年の段階である程度現在の気温情報が予測できていたことにもなります.

```
Data.pre <- Data[Data$Year<=1980,]
Data.post <- Data[Data$Year>1980,]

Res.pre <- lm(Temperature~Year, data=Data.pre)
summary(Res.pre)
```

Call:

```
lm(formula = Temperature ~ Year, data = Data.pre)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2.65662	-0.71084	-0.03093	0.68053	3.13765

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-41.115748	10.649413	-3.861	0.00023 ***
Year	0.022859	0.005489	4.165	7.9e-05 ***

---

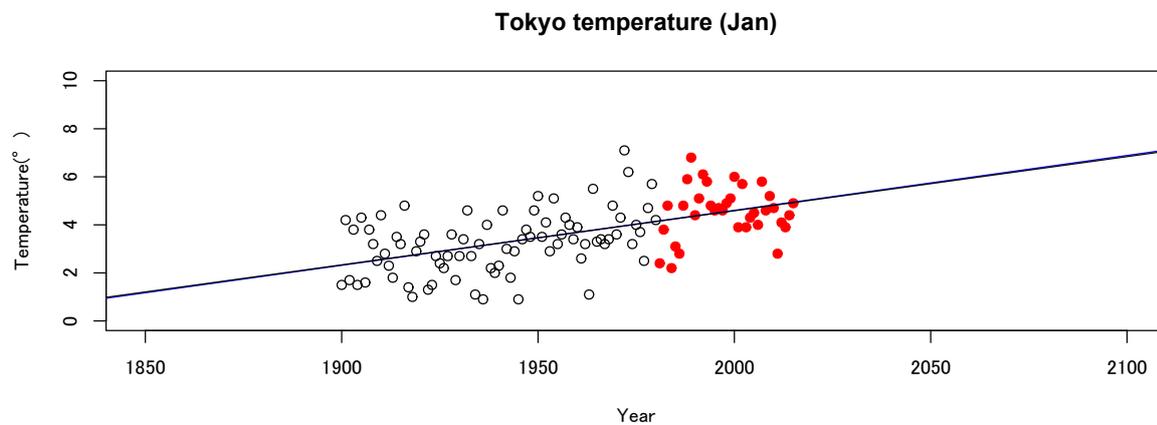
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.155 on 79 degrees of freedom

Multiple R-squared: 0.18, Adjusted R-squared: 0.1696

F-statistic: 17.34 on 1 and 79 DF, p-value: 7.901e-05

```
plot(Data.pre$Year, Data.pre$Temperature, xlim=c(1850,2100), ylim=c(0,10),
     main="Tokyo temperature (Jan)", xlab="Year", ylab="Temperature(°)")
points(Data.post$Year[], Data.post$Temperature, pch=19, col="red")
abline(Res.pre, col="blue")
abline(Res)
```



## 2.4 演習

では皆さん, 自由に月を選んで同様の解析をしてみてください (解析と解釈で所用 10 分).

### 3 アロメトリー式の推定

#### 3.1 体長-体重関係のアロメトリー式

体長を  $L$ , 体重を  $W$  とするとき, 体重の体長に対する相対成長率が体長-体重比に比例, すなわち

$$\frac{dW}{dL} = b \frac{W}{L} \quad (b > 0)$$

$$W = aL^b \quad (a > 0, b > 0)$$

で表せます. このような体長と体重の関係式をアロメトリー式とよびます. この式の両辺の自然対数をとると,

$$\log W = \log a + b \log L$$

という 1 次式となります.

いま,  $i$  ( $i = 1, 2, \dots, n$ ) 番目の個体の体長と体重の測定値をそれぞれ  $L_i, W_i$  とし, いずれも測定誤差なく観測されていると仮定します. このとき, ある統計モデル

$$\log W_i = \log a + b \log L_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2) \quad (i = 1, 2, \dots, n)$$

$x_i = \log L_i$ ,  $y_i = \log W_i$ , そして  $\alpha = \log a$ ,  $\beta = b$  とおけば, 先ほどの回帰モデルが利用できます.

#### 3.2 ニジマスの観測値

山梨県北杜市に位置する東京海洋大学大泉ステーションでは, サケ目サケ科に属する淡水魚のニジマス (rainbow trout) を実験実習用に飼育しています. 下図は, このニジマス 48 個体の尾叉長 (体長 mm) と体重 (g) の関係を示しています. このとき, ニジマスの体長と体重の関係を推定してみましょう.

```
# Reading and plotting the data
Data <- read.csv("Data_Rainbowtrout.csv", header=T)
names(Data)
```

```
[1] "Length" "Weight"
```

```
head(Data, 3)
```

```
  Length Weight
1    7.5     10
2    9.2      9
3    9.5     25
```

```
tail(Data, 3)
```

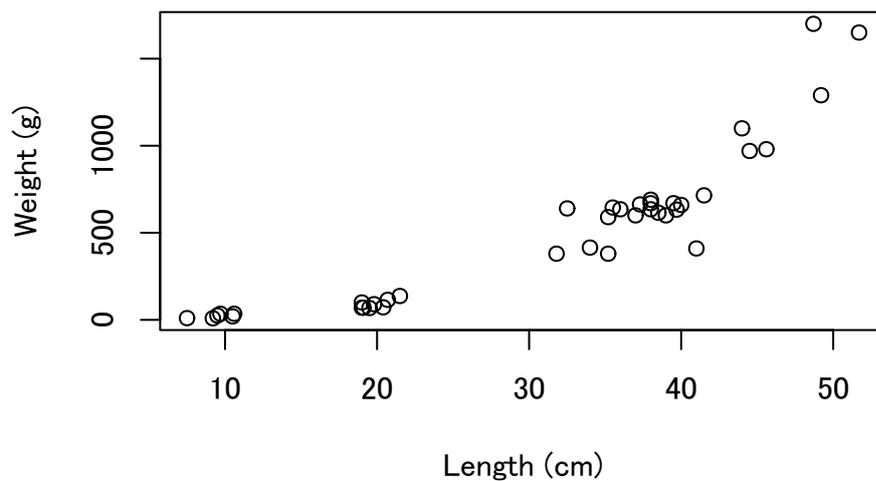
```
  Length Weight
37   48.7   1700
38   49.2   1290
39   51.7   1650
```

```

Length <- Data$Length
Weight <- Data$Weight
plot(Length, Weight, xlab="Length (cm)", ylab="Weight (g)",
     main="Allometric data for rainbow trouts")

```

### Allometric data for rainbow trouts



### 3.3 ニジマスのアロメトリー式の推定

```

Res <- lm(log(Weight)~log(Length))
summary(Res)

```

Call:

```
lm(formula = log(Weight) ~ log(Length))
```

Residuals:

Min	1Q	Median	3Q	Max
-0.59889	-0.08977	-0.00832	0.14493	0.69146

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-3.04898	0.28442	-10.72	6.67e-13 ***
log(Length)	2.60235	0.08474	30.71	< 2e-16 ***

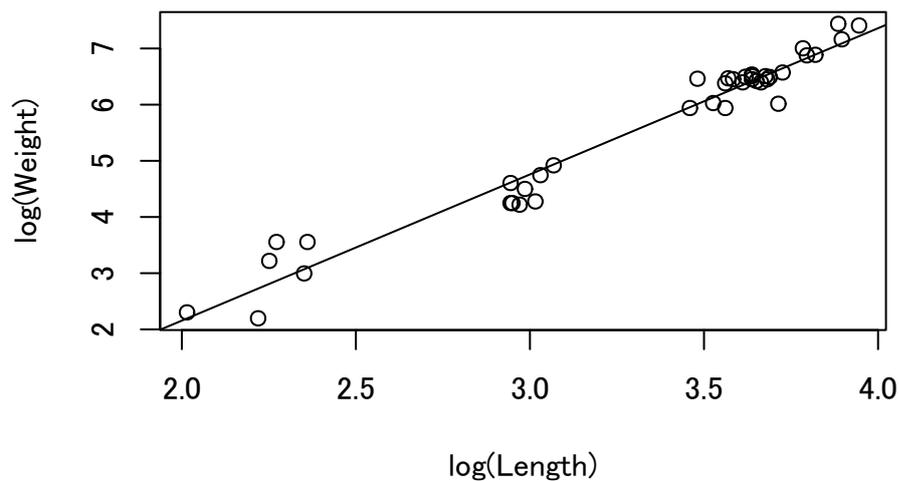
---

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

Residual standard error: 0.286 on 37 degrees of freedom

Multiple R-squared: 0.9622, Adjusted R-squared: 0.9612  
 F-statistic: 943.1 on 1 and 37 DF, p-value: < 2.2e-16

```
plot(log(Length), log(Weight))
abline(Res)
```



推定した切片と傾きを取り出しましょう。

```
summary(Res)$coef
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-3.048982	0.28442057	-10.71998	6.667534e-13
log(Length)	2.602353	0.08474089	30.70953	6.255463e-28

```
coef(Res)
```

```
(Intercept) log(Length)
-3.048982    2.602353
```

```
alpha.est <- as.numeric(coef(Res)[1])
beta.est <- as.numeric(coef(Res)[2])
data.frame(alpha.est, beta.est)
```

```
alpha.est beta.est
1 -3.048982 2.602353
```

次に、求めた  $\alpha, \beta$  の値をもとのパラメータ  $a, b$  に変換します。

```
a.est <- exp(alpha.est)
b.est <- beta.est
Est <- data.frame(a.est, b.est)
```

```
Est
```

```
      a.est    b.est
1 0.04740718 2.602353
```

今回は信頼区間も取り出してみましょう。

```
confint(Res)
```

```
      2.5 %    97.5 %
(Intercept) -3.625272 -2.472691
log(Length)  2.430651  2.774054
```

```
alpha.ci <- confint(Res)[1,]
beta.ci <- confint(Res)[2,]
```

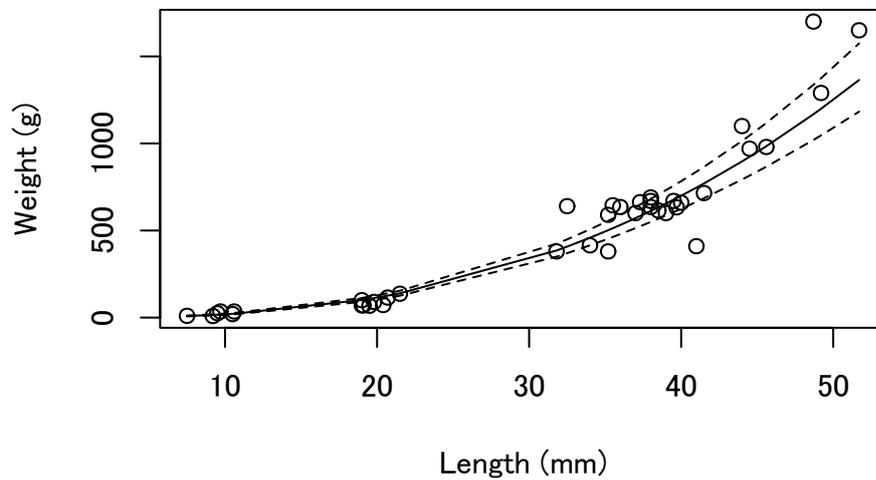
```
a.ci <- exp(alpha.ci)
b.ci <- beta.ci
CI <- rbind(a.ci, b.ci)
CI
```

```
      2.5 %    97.5 %
a.ci 0.02664184 0.08435757
b.ci 2.43065137 2.77405408
```

推定したアロメトリー式とその信頼区間 (CI=confidence interval) を表示します。

```
plot(Length, Weight, xlab="Length (mm)", ylab="Weight (g)", main="With CI for curve")
pred <- exp(predict(Res, int="c", level = 0.95))
matlines(Length, pred, lty=c(1,2,2), col=1)
```

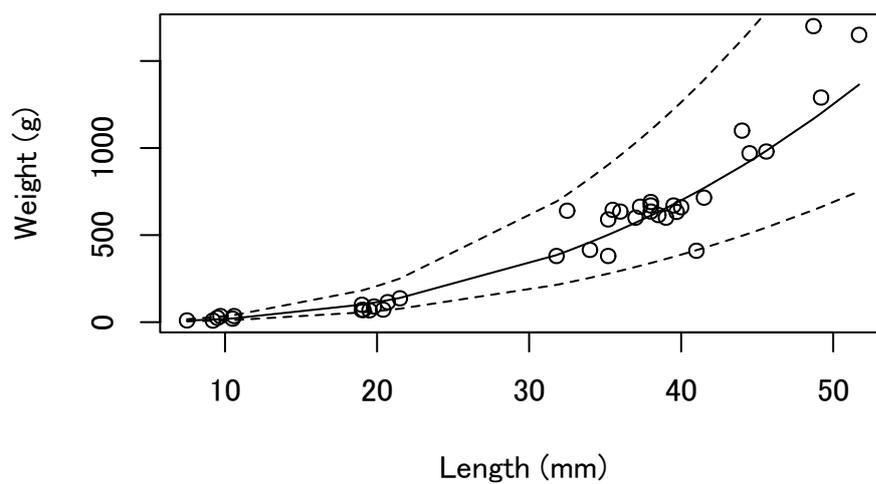
### With CI for curve



上記の信頼区間は推定した回帰式の信頼区間でした。回帰式に基づいて、別の個体の体長から体重を予測する場合の信頼区間を求めることもできます。

```
plot(Length, Weight, xlab="Length (mm)", ylab="Weight (g)", main="With CI for prediction")
pred <- exp(predict(Res, int="p", level = 0.95))
matlines(Length, pred, lty=c(1,2,2), col=1)
```

### With CI for prediction



## 4 魚体中の PCB 蓄積量の経年変化

北アメリカ北部に生息するレイクトラウトは、近年の生息環境の改善によって魚体中の PCB 蓄積量に変化がみられるであろうか。線形モデルを用いて解析してみましょう。

### 4.1 データの読み込みと図示

```
Data_LT <- read.csv("Data_Laketrout.csv", header=T)
Data_LT$Lenclass <- cut(Data_LT$Length, breaks=c(0,20,25,30,50))
names(Data_LT)
```

```
[1] "Length" "PCB" "Year" "Lenclass"
```

```
head(Data_LT, 3)
```

```
  Length PCB Year Lenclass
1  29.9 31.3 1974 (25,30]
2  29.5  7.9 1974 (25,30]
3  27.0 26.7 1974 (25,30]
```

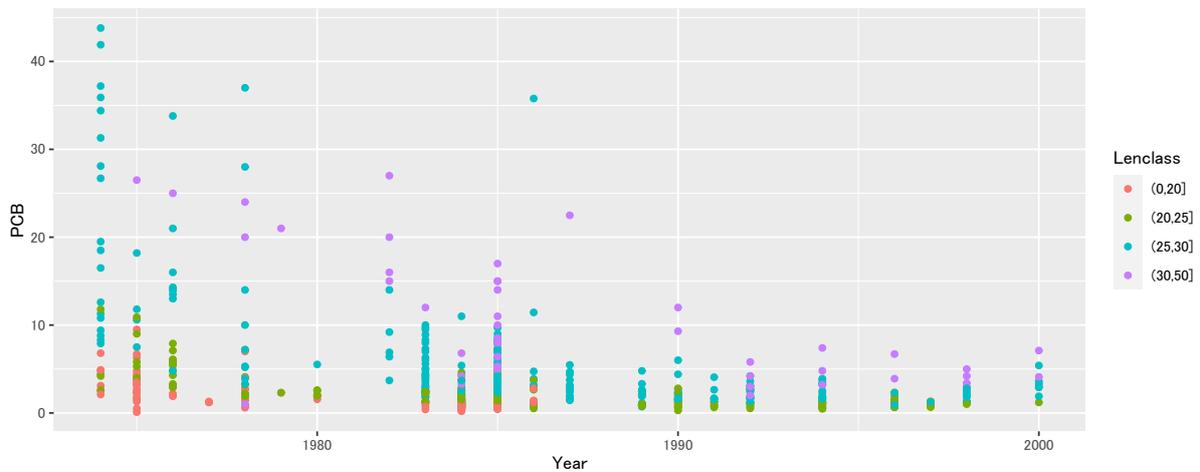
```
tail(Data_LT, 3)
```

```
  Length PCB Year Lenclass
630  31.8  7.1 2000 (30,50]
631  26.0  3.0 2000 (25,30]
632  30.8  4.2 1998 (30,50]
```

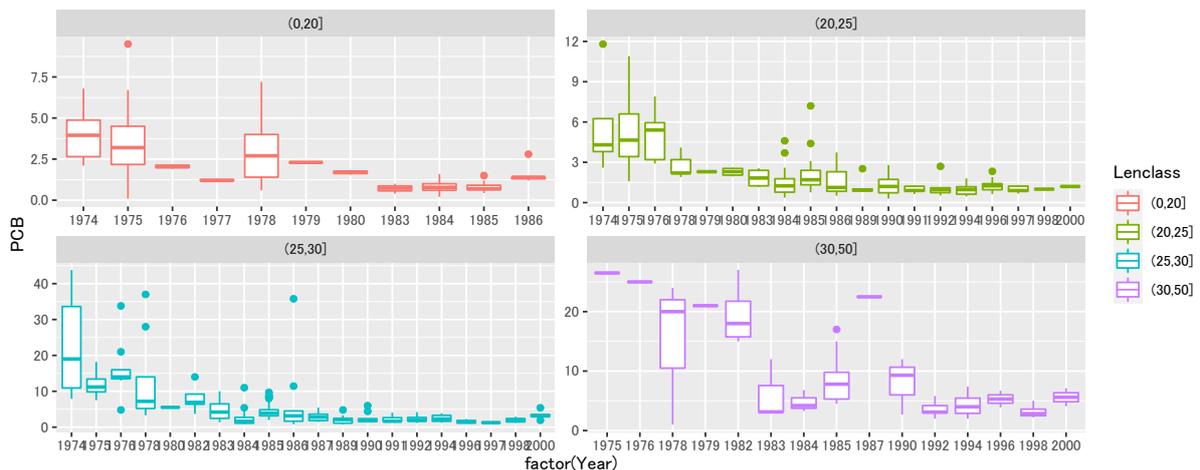
今度は練習のために、ggplot を用います。

```
library(ggplot2)
```

```
ggplot(Data_LT, aes(x=Year, y=PCB, color=Lenclass)) + geom_point()
```



```
ggplot(Data_LT, aes(x=factor(Year), y=PCB, color=Lenclass)) +
  geom_boxplot() + facet_wrap(Lenclass~., ncol=2, scales="free")
```



体長階級によって PCB の蓄積量が違うようです。また経年的に減っていることも示唆されます。ここでは体長階級 25-30cm の魚にフォーカスをあて、

$$PCB_i = e^{\alpha + \beta t_i}$$

というモデルを仮定してパラメータを推定してみよう。  $PCB_i$  は個体  $i$  の PCB 蓄積量,  $t_i$  は個体  $i$  のサンプリング年を表しています。

## 4.2 パラメータの推定

$$\log PCB_i = \alpha + \beta t_i + \varepsilon_i$$

```
Data <- Data_LT[Data_LT$Lenclass=="(25,30)",]
Res <- lm(log(PCB)~Year, data=Data)
summary(Res)
```

Call:

```
lm(formula = log(PCB) ~ Year, data = Data)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.65719	-0.37952	-0.04429	0.45508	2.29719

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	178.570071	12.970747	13.77	<2e-16 ***
Year	-0.089270	0.006532	-13.67	<2e-16 ***

---

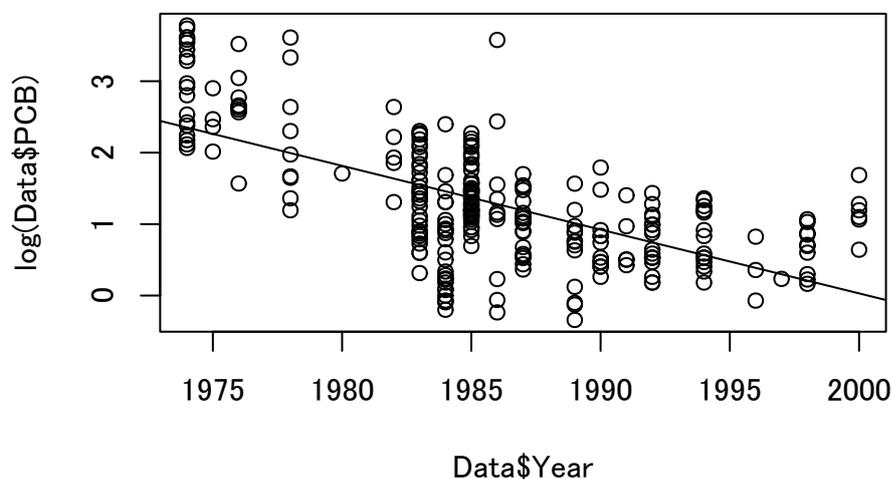
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6583 on 259 degrees of freedom

Multiple R-squared: 0.419, Adjusted R-squared: 0.4168

F-statistic: 186.8 on 1 and 259 DF, p-value: < 2.2e-16

```
plot(Data$Year, log(Data$PCB))
abline(Res)
```



```
alpha.est <- as.numeric(coef(Res)[1])
beta.est <- as.numeric(coef(Res)[2])
data.frame(alpha.est, beta.est)
```

```
alpha.est  beta.est
1 178.5701 -0.08926983
```

```
confint(Res)
```

```

                2.5 %      97.5 %
(Intercept) 153.0285227 204.11161849
Year         -0.1021317  -0.07640797

```

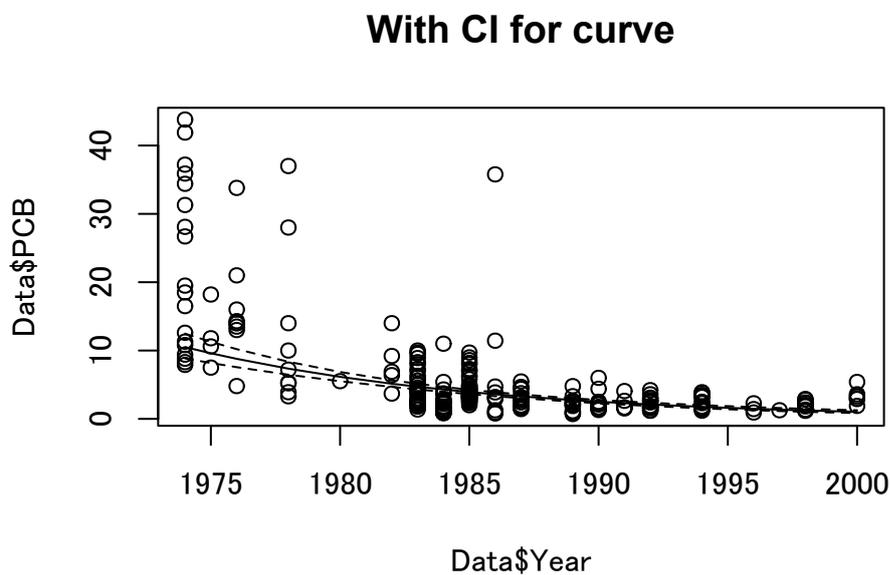
この場合も気温上昇と同じで、Year に対する係数  $\beta$  の P 値が 0.05 よりも小さくなっていますね。これも帰無仮説  $\beta = 0$  に対する仮説検定の結果で、ここから有意に PCB 蓄積量が減少していることが分かります。

信頼区間と共に年変化の結果を図示します。

```

plot(Data$Year, Data$PCB, main="With CI for curve")
pred <- exp(predict(Res, int="c", level = 0.95))
matlines(Data$Year, pred, lty=c(1,2,2), col=1)

```



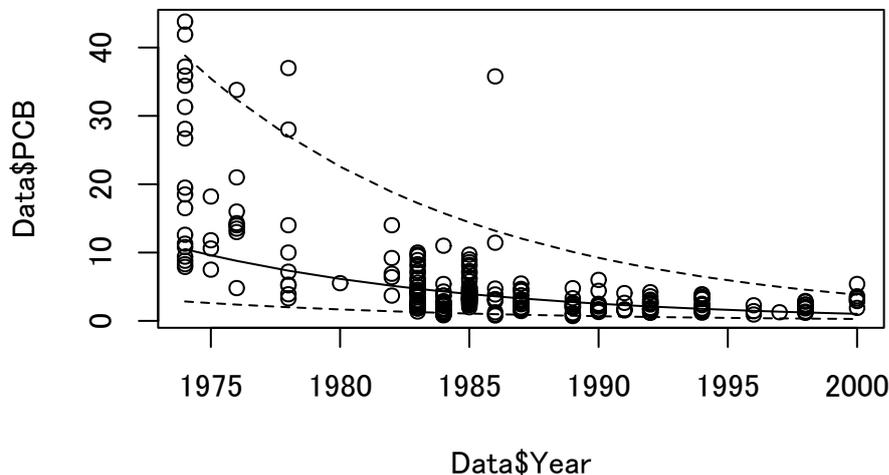
予測信頼区間と合わせて表示すると、データを大方説明していることが分かります。

```

plot(Data$Year, Data$PCB, main="With CI for curve")
pred <- exp(predict(Res, int="p", level = 0.95))
matlines(Data$Year, pred, lty=c(1,2,2), col=1)

```

### With CI for curve



#### 4.3 演習

30-50cm の魚ではどうでしょうか？解析してみてください。

---

ここからは、もし興味があればご覧ください。

### 5 魚の成長式の推定 (付録, 興味があればご一読下さい)

ヒトも含めて生物は年齢を追うごとに体の大きさが変化します。いま、年齢を  $t$  ( $t > 0$ ), 体長を  $L$  とおくと、この変化は次のような微分方程式で表現できます。

$$\frac{dL}{dt} = K(L_{\infty} - L)$$

この微分方程式の導関数  $dL/dt$ , すなわち体長の瞬間成長率は、体長  $L$  に関する一次関数でかつ負の傾きを持ちます。成長率が  $L = 0$  のところで最大となり、体長が大きくなるにしたがって次第に成長率が単調に減少し、体長が  $L_{\infty}$  に到達すると増加率が 0 となることを示しています。このような性質から、 $L_{\infty}$  は極限体長といい、また  $K$  は成長係数とよびます。ところで、この微分方程式も変数分離形であるから、

$$L = L_{\infty} - Ce^{Kt} \quad (C \text{ は任意定数})$$

という一般解が得られます。ここで、体長が 0 となるときの仮想上の年齢を  $t = t_0$  とおくと、

$$L(t) = L_{\infty} \{1 - e^{-K(t-t_0)}\}$$

という式が導けます。この式を von Bertalanffy の成長式とよびます。

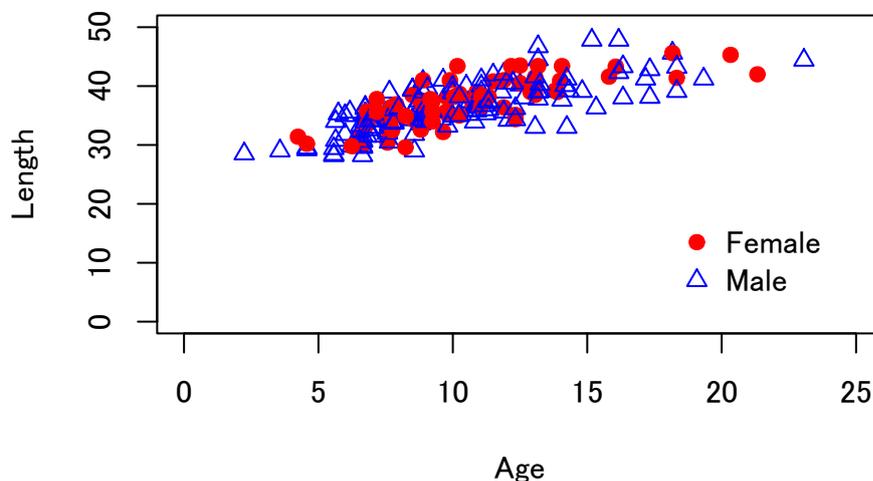
いま,  $n$  個体のサンプリングについて,  $i (= 1, 2, \dots, n)$  番目の個体の年齢と体長をそれぞれ  $t_i, L_i$  とおくことにします. まず, 観測値  $L_i$  に対して以下のような加法誤差モデルを考えましょう.

$$L_i = L_\infty \{1 - e^{-K(t_i - t_0)}\} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2) \quad (i = 1, 2, \dots, n)$$

先ほどまでの式とちがって, 左辺と右辺の関係が非線形となります. このようなモデルを正規非線形回帰モデルとよびます.

図は 1996 年に伊豆諸島周辺で漁獲され下田市場でサンプリングされたキンメダイの年齢と体長 (cm) の関係を示しています. キンメダイの年齢は, スライスされた耳石の輪紋 (年輪) 数を電子顕微鏡で計測し, 漁獲日と産卵時期の情報から年齢に変換しています. ここで年齢査定に誤差はないものとし, このデータを利用して年齢と体長の関係を推定してみよう.

```
Data <- read.csv("Data_Alfonsino.csv", header=T)
Age <- Data$Age
Length <- Data$Length
Sex <- as.numeric(Data$Sex) # "1" for female, "2" for male
mark <- c(19,2)
color <- c("red","blue")
plot(Length~Age,pch=mark[Sex],col=color[Sex],xlim=c(0,25),ylim=c(0,50))
legend(18,20,pch=mark,col=color,legend=c("Female","Male"),bty="n")
```



詳細は省略しますが, このデータを基に成長式を推定してみます. 最初は雌雄を区別しない場合.

```
# Definition of von Bertalanffy
growth.VB <- function(t,Linf,K,t0) Linf*(1.0-exp(-K*(t-t0)))
```

```
# 推定
start <- list(Linf=max(Length),K=0.1,t0=0)
```

```
res.000 <- nls(Length~growth.VB(t=Age,Linf, K, t0), start=start)
summary(res.000)
```

Formula: Length ~ growth.VB(t = Age, Linf, K, t0)

Parameters:

	Estimate	Std. Error	t value	Pr(> t )
Linf	47.95842	2.91479	16.453	< 2e-16 ***
K	0.08936	0.02413	3.704	0.00027 ***
t0	-6.38637	2.14489	-2.977	0.00324 **

---

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

Residual standard error: 2.564 on 215 degrees of freedom

Number of iterations to convergence: 4

Achieved convergence tolerance: 5.63e-06

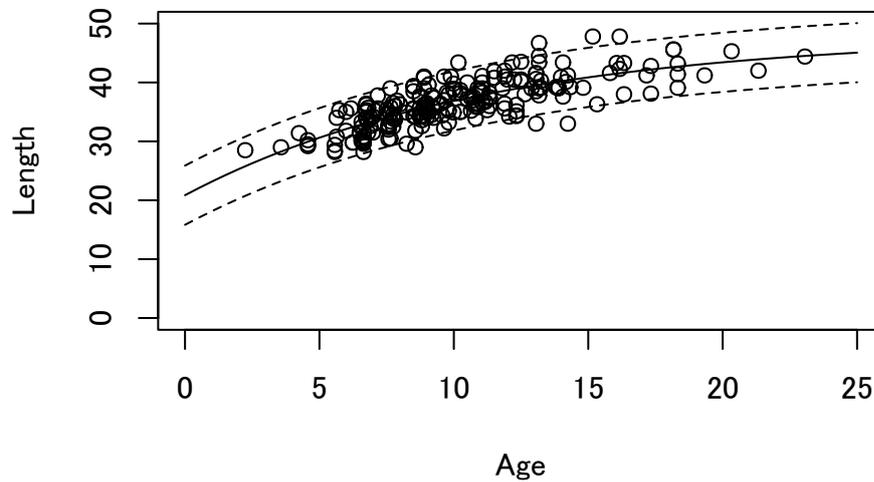
```
confint(res.000)
```

	2.5%	97.5%
Linf	44.10249477	58.3766050
K	0.04465578	0.1360594
t0	-12.34296242	-3.3029532

# 図示

```
taxis <- seq(0,25,0.5); tlen <- length(taxis);
pred.000 <- predict(res.000, list(Age=taxis))
plot(Length~Age,xlim=c(0,25),ylim=c(0,50), main="Results under res.000 with prediction CI")
points(taxis, pred.000, type="l")
points(taxis, pred.000-1.96*sigma(res.000), type="l", lty=2)
points(taxis, pred.000+1.96*sigma(res.000), type="l", lty=2)
```

## Results under res.000 with prediction CI



次に雌雄別に推定してみましょう。

# 推定

```
start <- list(Linf=rep(max(Length),2),K=c(0.1,0.1),t0=c(0,0))
res.111 <- nls(Length~growth.VB(t=Age,Linf[Sex], K[Sex], t0[Sex]), start=start)
summary(res.111)
```

Formula: Length ~ growth.VB(t = Age, Linf[Sex], K[Sex], t0[Sex])

Parameters:

	Estimate	Std. Error	t value	Pr(> t )
Linf1	48.15050	4.52391	10.644	< 2e-16 ***
Linf2	47.70053	3.75965	12.687	< 2e-16 ***
K1	0.09797	0.04315	2.270	0.02418 *
K2	0.08646	0.02995	2.887	0.00429 **
t01	-5.12851	3.38334	-1.516	0.13106
t02	-6.94091	2.80413	-2.475	0.01410 *

---

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

Residual standard error: 2.567 on 212 degrees of freedom

Number of iterations to convergence: 4

Achieved convergence tolerance: 3.178e-06

```
confint(res.111)
```

2.5%      97.5%

```

Linf1 43.09967561 81.7105984
Linf2 43.18112305 66.9658334
K1    0.02164258 0.1833475
K2    0.03031102 0.1453427
t01   -18.02360053 -0.9526554
t02   -16.13978541 -3.1310768

```

# 図示

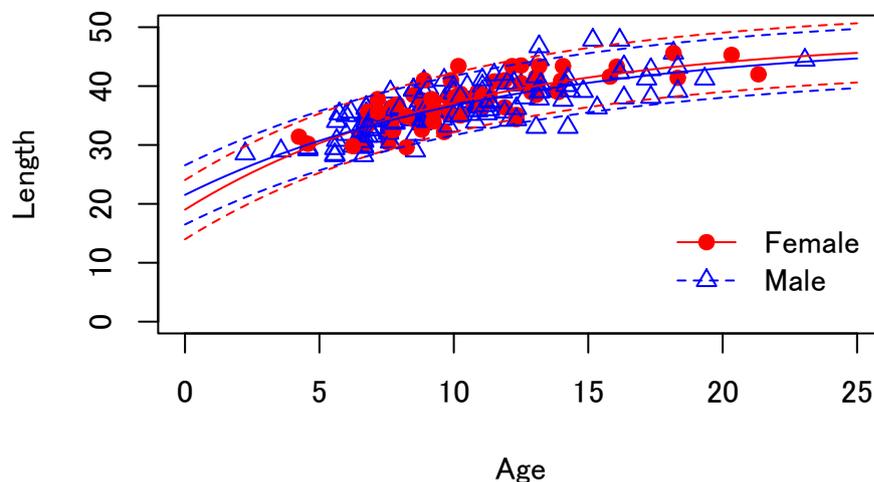
```

taxis <- seq(0,25,0.5); tlen <- length(taxis);
pred.female.111 <- predict(res.111, list(Age=taxis, Sex=rep(1,tlen)))
pred.male.111 <- predict(res.111, list(Age=taxis, Sex=rep(2,tlen)))

plot(Length~Age,pch=mark[Sex],col=color[Sex],xlim=c(0,25),ylim=c(0,50),
     main="Results under res.111 with prediction CI")
points(taxis, pred.female.111, type="l", col="red", lty=1)
points(taxis, pred.female.111-1.96*sigma(res.111), type="l", col="red", lty=2)
points(taxis, pred.female.111+1.96*sigma(res.111), type="l", col="red", lty=2)
points(taxis, pred.male.111, type="l", col="blue", lty=1)
points(taxis, pred.male.111-1.96*sigma(res.111), type="l", col="blue", lty=2)
points(taxis, pred.male.111+1.96*sigma(res.111), type="l", col="blue", lty=2)
legend(18,20,pch=mark,col=color,lty=c(1,2),legend=c("Female","Male"),bty="n")

```

### Results under res.111 with prediction CI



統計的な手法を用いれば、どちらのモデルによる結果が妥当か比較できます。詳しくは3年次の授業で学びますが、AICという規準値が小さい方がよいモデルと言えます。この場合、雌雄区別しないモデルの方がデータからは支持されます（雌雄別に解析をしてもそれほど違いが無いことが分かり、節約的なモデルである雌雄区別なしモデルが選択されます）。

```
AIC(res.000) # 雌雄区別しない場合の AIC
```

```
[1] 1034.122
```

```
AIC(res.111) # 雌雄区別する場合の AIC
```

```
[1] 1037.531
```

```
plot(Length~Age,pch=mark[Sex],col=color[Sex],xlim=c(0,25),ylim=c(0,50))  
points(taxis, pred.000, type="l", lwd=2)  
points(taxis, pred.female.111, type="l", col="red", lty=1)  
points(taxis, pred.male.111, type="l", col="blue", lty=2)
```

