12章-時系列分析

1296603c
埴岡　瞬

今回用いる例データ

- lh(小文字のエル) – ある女性の血液中の黄体ホルモンを10分間隔で測定した時系列データ

- UKgas – 1960年～1986年のイギリスのガス消費量を四半期ごとに観測した時系列データ

- ldeaths – 1974年～1979年のイギリスで喘息、気管支炎、肺気腫による死亡数を月ごとに記録した時系列データ。
  - mdeaths – ldeaths　を区別し、男性のみに絞ったもの
  - fdeaths – ldeaths　を区別し、女性のみに絞ったもの

時系列分析

- 基本概念とデータ操作
- 自己共分散と自己相関
- スペクトル分析
- ランダムウォークと単位根(単位根検定)
- ARモデル

基本概念

- 時系列データとは

  時間とともに変動する現象に対し、時間の順序で測定した結果を記録したデータ。
  通常、一定の時間間隔で測定される。
  Ex)医療データ/気象データ/金融・経済データ

- 目的

  変動を統計的に分析し、その特徴を捉えることによる現象の解明と将来の変動の予測・制御

- 時系列データの表記

  $y_1, y_2, \ldots, y_{t-k}, \ldots, y_{t-l}, y_t, y_{t+1}, \ldots y_{t+k}, \ldots y_{n-1}, y_n$

  観測・測定値=y / 標本サイズ=n / 測定した時点=t

データ操作(1)-属性・コンテンツの表示

例を用いて入力.

データ:lh

>class(lh)
[1] "ts"

“ts”=”time series”時系列データであることを示す。

>lh

```
Time Series:
Start = 1
End = 48
Frequency = 1
 [1] 2.4 2.4 2.4 2.2 2.1 1.5 2.3 2.3 2.5 2.0 1.9 1.7
[13] 2.2 1.8 3.2 3.2 2.7 2.2 2.2 1.9 1.9 1.8 2.7 3.0
[25] 2.3 2.0 2.0 2.9 2.9 2.7 2.7 2.3 2.6 2.4 1.8 1.7
[37] 1.5 1.4 2.1 3.3 3.5 3.5 3.1 2.6 2.1 3.4 3.0 2.9
```

他のデータを用いて入力.

データ:UKgas

>start(UKgas)
[1]1960　　1
>end(UKgas)
[1]1986　　4
>frequency(UKgas)
[1]4

start-データの開始時間/end-データの終了時間/frequency-測定回数

>UKgas

データ操作(2)-データの切り出し

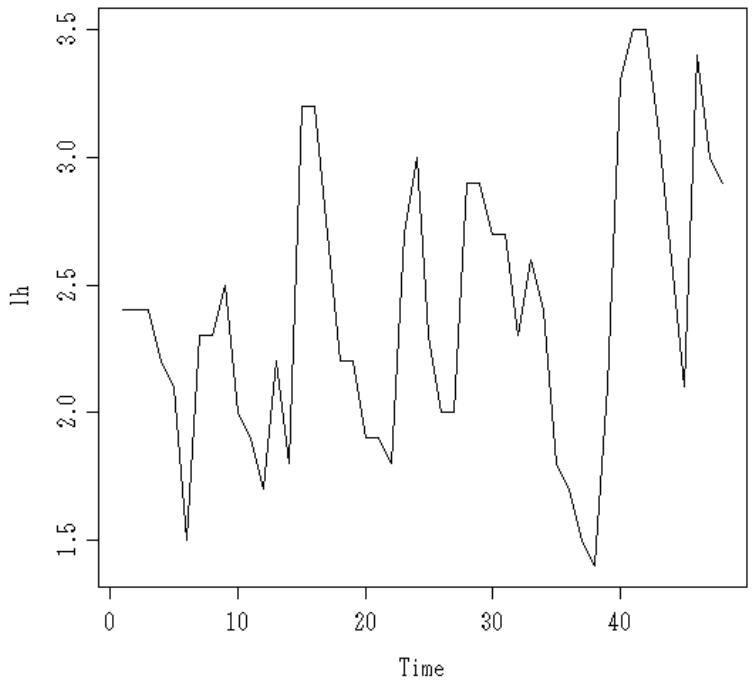
関数”window”
時系列データの一部を取り出す。

>window(UKgas,c(1975,2),c(1979,3))

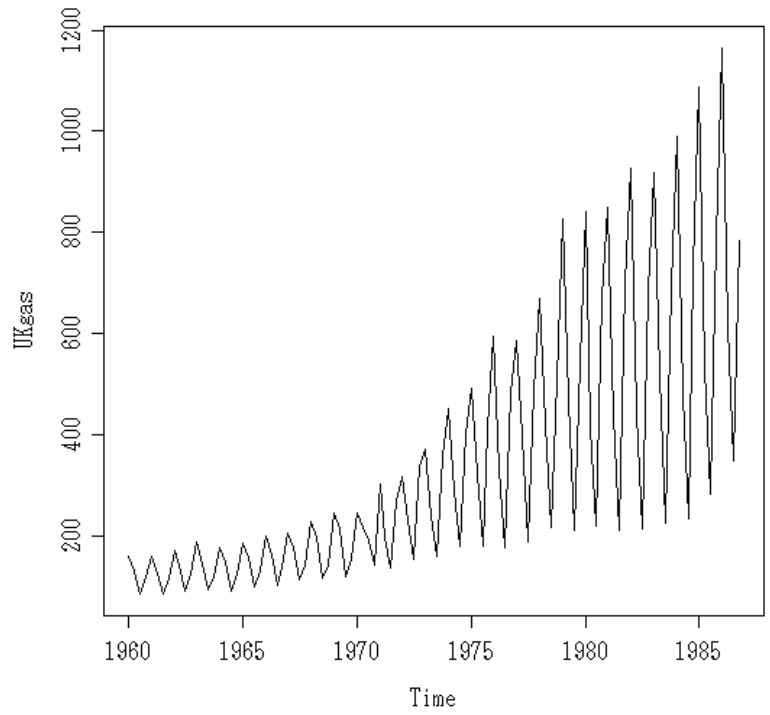
	Qtr1	Qtr2	Qtr3	Qtr4
1975		321.8	177.7	409.8
1976	593.9	329.8	176.1	483.5
1977	584.3	395.4	187.3	485.1
1978	669.2	421.0	216.1	509.1
1979	827.7	467.5	209.7	

データの操作(3)-時系列データの図示

>ts.plot(lh)

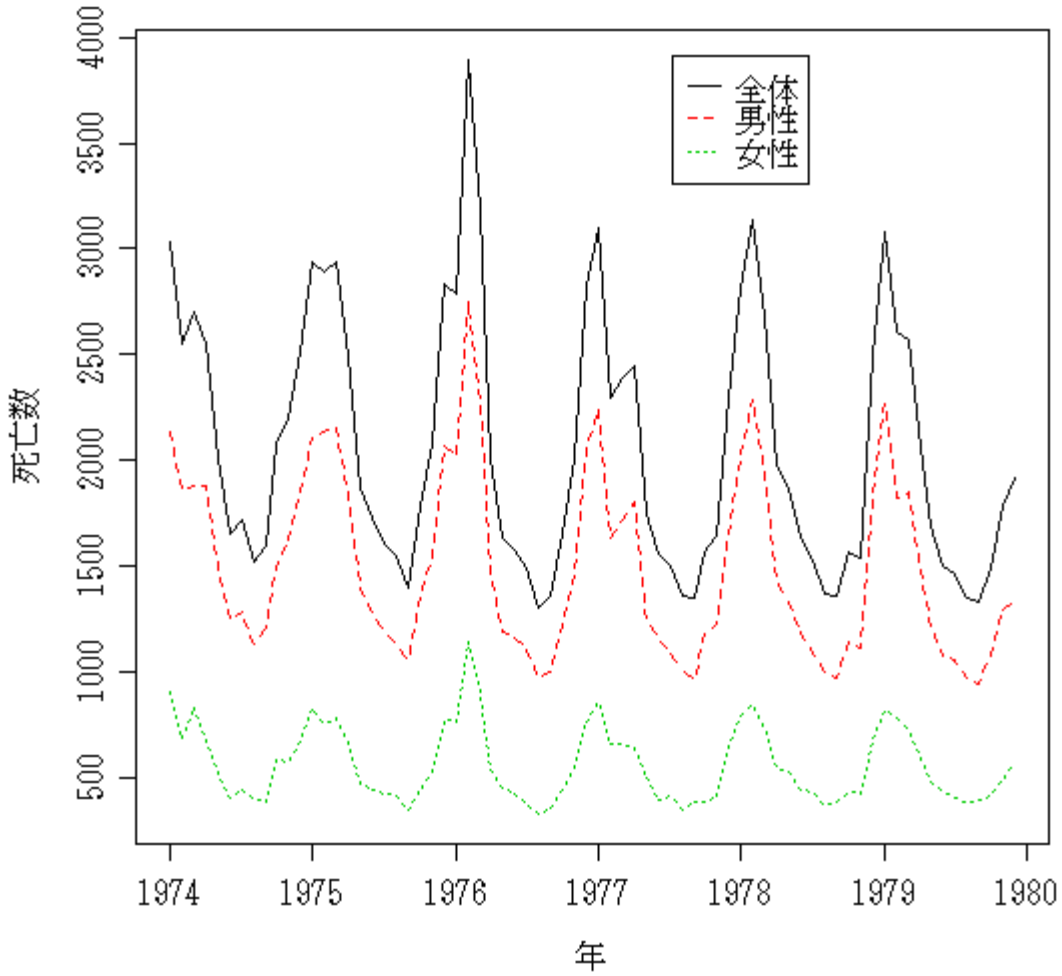


>ts.plot(UKgas)



関数“ts.plot”-線種、色を自由に設定可能。
データ：ldeaths / mdeaths / fdeaths

> ts.plot(ldeaths, mdeaths, fdeaths,gpars=list(xlab="年", ylab="死亡数",lty=c(1:3),col=c(1:3)))
> legend(locator(1),c("全体","男性","女"),lty=c(1:3),col=c(1:3))



データ操作(4)-データオブジェクトの作成

非時系列データオブジェクト⇒時系列データオブジェクト

関数”ts”を使う（※開始時間(start)/観測数(frequency)の指定が必要）

Ex) 1～120の整数を1995年から2004年まで1年に12回観測したデータとして時系列データオブジェクトを作成した場合

>temp<-ts(1:120,start=c(1995,6),frequency=12)
> class(temp)
[1]"ts"
>temp

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1995						1	2	3	4	5	6	7
1996	8	9	10	11	12	13	14	15	16	17	18	19
1997	20	21	22	23	24	25	26	27	28	29	30	31
1998	32	33	34	35	36	37	38	39	40	41	42	43
1999	44	45	46	47	48	49	50	51	52	53	54	55
2000	56	57	58	59	60	61	62	63	64	65	66	67
2001	68	69	70	71	72	73	74	75	76	77	78	79
2002	80	81	82	83	84	85	86	87	88	89	90	91
2003	92	93	94	95	96	97	98	99	100	101	102	103
2004	104	105	106	107	108	109	110	111	112	113	114	115
2005	116	117	118	119	120							

データ操作(5)-ラグ処理

- ラグ(lag)-時間の遅れ
- データの遅れを修正して分析する際に使用する。
- $y_1, y_2, \ldots, y_{t-k}, \ldots, y_{t-l}, y_t, y_{t+1}, \ldots y_{t+k}, \ldots y_{n-1}, y_n$　y_{t-1}=1次ラグ.y_{t-2}=２次ラグ

>ldeaths

>lag(ldeaths,k=5)

この２つによって呼び出されたデータを比較

データ操作(6)-トレンド除去

- ▲$y_t=y_t-y_{t-1}$　⇒　「差分(階差)」
- トレンド-傾向変動(季節や時期などによるデータの上昇,下落傾向)
- 差分(階差)操作により、時系列データからトレンドを取り除く
  ⇒時系列データを見やすくする
  関数”diff”を使用

> plot(diff(UKgas))

自己共分散と自己相関

・自己共分散を標準化　⇒　自己相関

・異なる時点間でその過程が何かしらの相関(関係性)を持っているかどうかをします指標

・関数”acf”を使う。

“correlation(自己相関)”/”covarianc(自己共分散)”/”partial”(偏相関)

acf(x, type =”□□□”, plot = TRUE,...)

>acf(UKgas)

※信頼区間を指定することも可能(引数”ci”)

>acf(Ukgas,ci=0.9)

データの周期性・トレンドを除去しトレンドや周期ごとに図示する

>par(mai=rep(0.2,4),mfrow=c(4,1))

> for(i in 1:4)plot(diff(log(UKgas),lag=i))

>par(op)

・相互分散．相互相関(異なる2データ間の関係性)

関数”ccf”

>ccf(mdeaths,fdeaths)

・偏自己共分散．相関(異なる時点のデータ間の純粋な関係性)

関数”pacf”

スペクトル分析

- 時系列データ=トレンド+周期的に変動する成分+ノイズ(統計学的ばらつき)
- スペクトル分析ー時系列データに隠されている周期性を解析する方法
- スペクトル⇒時系列における自己共分散C_kのフーリエ変換※が可能である時周波数2/1≦f≦-2/1の間で定義される関数P(f)
- ピリオドグラム⇒スペクトルを標本データにおける自己共分散C'_kを代用して定義した関数

※フーリエ変換 - 周波数を見やすく変換すること。(参照:http://www.geocities.co.jp/AnimalPark-Shiro/1620/ft/1.html)

- ピリオドグラムを用いてスペクトルを推定

関数”spec.pgram”

```
>par(mfrow=c(2,2))
>spec.pgram(UKgas)
>spec.pgram(UKgas,spans=c(3,3))
>spec.pgram(ldeaths)
>spec.pgram(ldeaths,spans=c(3,3))
```

- 自己回帰によりスペクトルを推定

関数”spectrum”

```
> par(mfrow=c(1,2))
> spectrum(UKgas,method="ar")
> spectrum(ldeaths,method="ar")
```

※自己回帰モデル-時系列自体の過去の過程が現在の値にどのような影響を及ぼしたのかをモデリングしたもの。

単位根検定

- 「見せかけの回帰」を避けるために

- ある時系列データが

  $y_t = ay_{t-1} + e_t$ で表現でき、|a| = 1(単位根)であるとき、
  このデータはランダムウォークであるという。

- 検定方法
  - 関数”PP.test”
    > PP.test(lh)

  - 関数”adf.test”
    >install.packages(“tseries”); library(tseries)
    >adf.test(UKgas)

- P値で単位根があるかどうかを判断
- p.value = □□
- P値が有意水準(0.1,0.05)を超えているかどうかで判定

<u>ランダムウォークであるデータは同時に非定常である</u>

非定常データ ⇒ 定常化(対数や差分を取る)

>adf.test(diff(UKgas))$p.value

ARモデル(1)-自己回帰モデル

- 時系列のある時点t-pからtまでの各データの関係式

$y_t = \sum_{i=1}^{p} a_i y_{t-i} + e_t$ を自己回帰モデル(ARモデル)と呼ぶ

p – 次数(order)

e_t – 残差(通常平均0,分散σ^2 の正規分布に従う確率変数であると仮定)

自己回帰分析では、

1,適切に次数pを決定すること

2,適切に自己回帰係数a_iを推定すること　　が主な作業になる

この作業を、モデルの当てはめ/モデルの推定　　と呼ぶ。

ARモデル(2)-モデルの推定

- 推定法
  - ユールウォーカ(yule-walker)
  - 最小2乗法(ols)
  - 最尤法(mle)
  - バーグ法(Burg)　　の4つが提案されている。
- 関数”ar”(ar(x, aic = TRUE, method=“”,order.max = NULL,…)

  > (lh.ar <-ar(lh))
- 関数”ar”で作成した自己回帰モデルに関連する項目のリストを呼び出す

  > summary(lh.ar)
- 自己回帰モデルの次数($order)を呼び出す[AICの値($aic)、残差($resid)]

  > lh.ar$order
- 次数に基づいて小数点以下3桁までに丸めたデータを呼び出す

  >round(lh.ar$ar,3)

この結果から、AR(3)のモデルは $y'_t = 0.653y_{t-1} - 0.064y_{t-2} - 0.227y_{t-3}$ であると推定できる

ARモデル(3)-モデルの診断

- 作成したモデルの適切さを判断 – 残差分析が必要

- ARモデルにおける残差 – 平均0の正規分布に従い、残差同士が独立であることが理想的

- 時系列データの各残差間の独立性を検定(関数”Box.test”)

```
> Box.test(lh.ar$res, type="Ljung")
```

この結果から残差は独立であると言える。

- 残差分布の正規性を検定(関数”jarque.bera.test”)

```
> temp<- window(lh.ar$res,start=4)
> jarque.bera.test(temp)
```

※あくまでもこの関数から得られる結果は参考程度にしかならない

ARモデル(4)-予測

- モデル構築 → 予測
- 関数”predict”

> (lh.pr<-predict(lh.ar,n.ahead=10))

n.ahead = 予測の期間の指定

```
$pred
Time Series:
Start = 49
End = 58
Frequency = 1
 [1] 2.461588 2.272267 2.199151 2.262914 2.352194 2.423066
 [7] 2.449223 2.441544 2.418779 2.398456

$se
Time Series:
Start = 49
End = 58
Frequency = 1
 [1] 0.4425687 0.5286675 0.5525786 0.5527502 0.5592254
 [6] 0.5665903 0.5688786 0.5694299 0.5698047 0.5698047
```

$pred = 予測値/ $se = 標準誤差

- “predict”により得た自己回帰予測値、2倍の標準誤差を図示(plot)

```
> SE1<-lh.pr$pred+2*lh.pr$se
> SE2<-lh.pr$pred-2*lh.pr$se
> ts.plot(lh,lh.pr$pred,SE1,SE2,gpars=list(lt=c(1,2,3,3),col=c(1,2,4,4)))
> legend(locator(1),c("実測値","予測値","2*SE"), lty=c(1,2,3),col=c(1,2,4))
```