Statistical Method
Advanced Statistical Models

I-Chen Lee, STAT, NCKU

Dec. 12, 2023

Statistical Models

The common structure of the statistical Model:

$$Y = f(x_1, x_2, \ldots, x_p) + \varepsilon(x_1, x_2, \ldots, x_p).$$

- $f(x)$
  1. linear form (linear function of unknown parameters)
     $$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2,$$
     $$E(y) = \beta_0 + \beta_1 x_1 + exp(\beta_2) x_2,$$
  2. nonlinear form: $E(y) = \exp\{\theta_1 x_1 \exp(-\theta_2 x_2)\}$
  3. categorical variables: using the techniques of dummy variable
- Assumptions of $\varepsilon(x_1, x_2, \ldots, x_p)$
  1. $\varepsilon \sim N(0, \sigma^2)$ (independent to covariates): regression
  2. ε follows non-normal distributions: generalized model, logistic regression, probit model, ...
  3. $\varepsilon(x_1, x_2, \ldots, x_p)$ (dependent to covariates): Variance Heterogeneity
  4. ε (dependent to time): Autoregressive (AR) errors

Overview

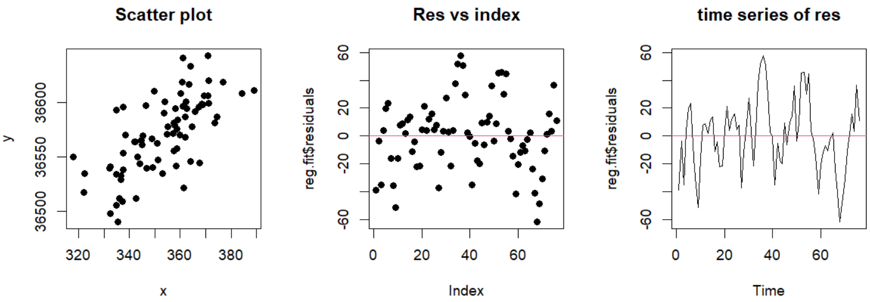
1. Linear regression models with autoregressive errors
2. Mixed effect model
3. Fitted a Non-linear function

Example: Economic Measure

`https://online.stat.psu.edu/stat510/lesson/8/8.1`

The economic indicator is the predictor and the measure of economy is the response.

The scatter plot and the residuals plot are shown as follows:



Linear regression models with autoregressive errors

What is called autoregressive errors?

$$\varepsilon_t = \theta_0 + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_p \varepsilon_{t-p} + e_t,$$

where e_t is the white noise. Then, we call it as the AR(p) model of the residuals.

The related hypothesis testing for AR(1) errors is called Durbin-Watson test.

$H_0: \ \varepsilon_t = e_t,$ where e_t is the white noise.

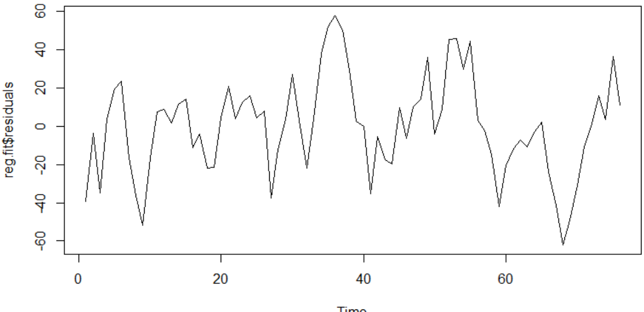
$H_1: \ \varepsilon_t = \theta_1 \varepsilon_{t-1} + e_t,$ where e_t is the white noise.

If the null hypothesis is rejected, then we can conclude that there is time dependent structure on the residuals.

Check time dependent structure

- In R, the package "car" has the Durbin-Watson test.
- The p-value is smaller than 0.05, then reject the time independent assumption.
- The "autocorrelation" is present in the residuals.

```
> durbinWatsonTest(reg.fit)
 lag Autocorrelation D-W Statistic p-value
   1       0.6356138     0.6952261       0
 Alternative hypothesis: rho != 0
```



What are the suitable order for the residuals?

Use the "partial autocorrelation function" (PACF) to examine the appropriate order for AR(p) model.
Given a tome series z_t, the PACF of lag k, denoted $\phi_{k,k}$, is the autocorrelation between z_t and z_{t+k} that is not accounted for by lags 1 through $k-1$, inclusive.

$$\phi_{1,1} = corr(z_{t+1}, z_t), \text{ for } k = 1,$$

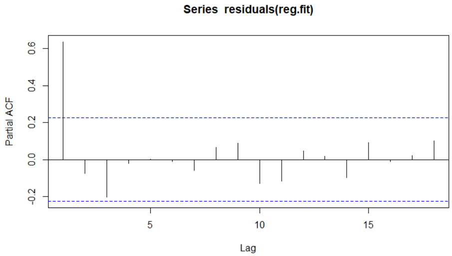
$$\phi_{k,k} = corr(z_{t+k} - \hat{z}_{t+k}, z_t - \hat{z}_t), \text{ for } k \geq 2,$$

where $\hat{z}_{t+k}$ and $\hat{z}_t$ are linear combination of $\{z_{t+1}, z_{t+2}, \ldots, z_{t+k-1}\}$, respectively.

Example: PACF of the residuals

It is found that the highest is 0.636 with the lag 1. Then, we can try to fit an AR(1) model for residuals.

```
> pacf(residuals(reg.fit))
```




```
> p <- pacf(residuals(reg.fit))
> p

Partial autocorrelations of series ‘residuals(reg.fit)’, by lag

     1      2      3      4      5      6      7      8      9     10
 0.636 -0.075 -0.203 -0.021  0.001 -0.011 -0.059  0.065  0.089 -0.130
    11     12     13     14     15     16     17     18
-0.118  0.047  0.018 -0.098  0.092 -0.011  0.021  0.101
```

AR(1) model for the residuals

```
> arima(residuals(reg.fit), order = c(1,0,0), include.mean = FALSE) #AR(1)

Call:
arima(x = residuals(reg.fit), order = c(1, 0, 0), include.mean = FALSE)

Coefficients:
         ar1
      0.6488
s.e.  0.0875

sigma^2 estimated as 378.1:  log likelihood = -333.64,  aic = 671.29
```

Then, the model for the residuals is

$$\varepsilon_t = 0.6488\varepsilon_{t-1} + e_t,$$

where $e_t \sim N(0, 378.1)$.

Combine with the response and the covariate

- Two-stage estimation:

$$y_t = 36001.84 + 1.61x_t + \varepsilon_t,$$

$$\varepsilon_t = 0.6488\varepsilon_{t-1} + e_t, \text{ where } e_t \sim N(0, 378.1).$$

It implies $y_t = 36001.84 + 1.61x_t + 0.6488\varepsilon_{t-1} + e_t$, where $e_t \sim N(0, 378.1)$.

- One-stage estimation in R by airma(): It implies $y_t = 35986 + 1.65x_t + 0.6496\varepsilon_{t-1} + e_t$, where $e_t \sim N(0, 392.8)$.

```
> arima(y, order = c(1, 0, 0), xreg = x)

Call:
arima(x = y, order = c(1, 0, 0), xreg = x)

Coefficients:
         ar1   intercept       x
      0.6496  35986.2860  1.6521
s.e.  0.0874     41.4672  0.1163

sigma^2 estimated as 377.3:  log likelihood = -333.57,  aic = 675.15
```

Check the white noise

Use the Ljung–Box test to test if the residuals are white noise.

$$H_0: \text{ The data are independently distributed}$$

$$H_1: \text{ The data are not independently distributed}$$

```
> fit.ar1reg <- arima(y, order = c(1, 0, 0), xreg = x)
> checkresiduals(fit.ar1reg)

        Ljung-Box test

data:  Residuals from ARIMA(1,0,0) with non-zero mean
Q* = 6.633, df = 9, p-value = 0.6753

Model df: 1.   Total lags used: 10
```

More on time series models

Keywords:

- Autocorrelation function (ACF)
- Partial autocorrelation function (PACF)
- AR model, moving average (MA) model, autoregressive moving average (ARMA) model, ...
  - AR(p)

$$X_t = c + \sum_{k=1}^{p} \phi_k x_{t-k} + e_t.$$

  - MA(q)

$$X_t = \mu + e_t + \sum_{k=1}^{q} \theta_k e_{t-k}.$$

  - ARMA(p,q)

$$X_t = c + e_t + \sum_{k=1}^{p} \phi_k x_{t-k} + \sum_{k=1}^{q} \theta_k e_{t-k}.$$

Use auto.arima() in R

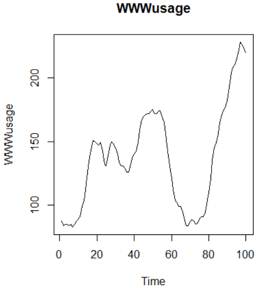
ARIMA(p, d, q) model: Autoregressive Integrated Moving Average model, where d is the degree of first differencing involved.

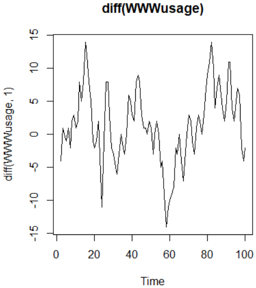
White noise	ARIMA(0,0,0)
Random walk	ARIMA(0, 1, 0)
AR(p)	ARIMA(p, 0, 0)
MA(q)	ARIMA(0, 0, q)

Example:

```
> auto.arima(y, xreg=x)
Series: y
Regression with ARIMA(1,0,0) errors

Coefficients:
         ar1   intercept    xreg
      0.6496  35986.2860  1.6521
s.e.  0.0874     41.4672  0.1163

sigma^2 = 392.8:  log likelihood = -333.57
AIC=675.15   AICc=675.71   BIC=684.47
```

Internet Usage per Minute by auto.arima() in R





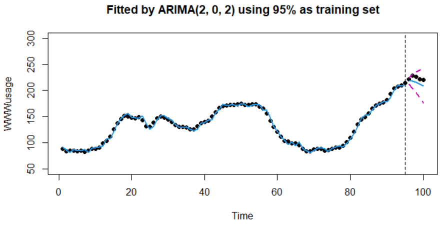

```
> auto.arima(WWWusage)
Series: WWWusage
ARIMA(1,1,1)

Coefficients:
         ar1     ma1
      0.6504  0.5256
s.e.  0.0842  0.0896

sigma^2 = 9.995:  log likelihood = -254.15
AIC=514.3   AICc=514.55   BIC=522.08
```

```
> auto.arima(diff(WWWusage))
Series: diff(WWWusage)
ARIMA(1,0,1) with zero mean

Coefficients:
         ar1     ma1
      0.6504  0.5256
s.e.  0.0842  0.0896

sigma^2 = 9.995:  log likelihood = -254.15
AIC=514.3   AICc=514.55   BIC=522.08
```

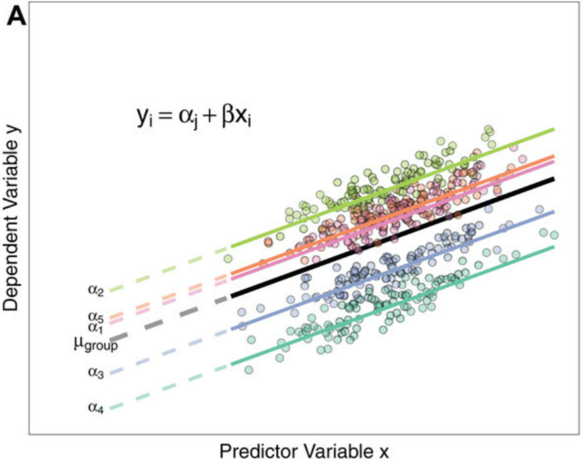
Forecasting by auto.arima() in R

95% observations are set to be the training set.

```
> train.U <- WWWusage[1:95]
> test.U <- WWWusage[96:100]
> train.arima <- auto.arima(train.U)
> train.arima
Series: train.U
ARIMA(2,0,2) with non-zero mean

Coefficients:
         ar1      ar2     ma1      ma2      mean
      1.9238  -0.9425  0.0273  -0.4392  136.5997
s.e.  0.0763   0.0757  0.2018   0.1908   10.0361

sigma^2 = 9.849:  log likelihood = -244.36
AIC=500.72   AICc=501.68   BIC=516.05
> predict <- forecast(train.arima, 5)
>
> par(mfrow = c(1,1))
> plot(WWWusage, type = "b", pch = 19, ylim = c(50, 300),
+      main = "Fitted by ARIMA(2, 0, 2) using 95% as training set")
> lines(train.arima$fitted, col = 4, lwd = 2)
> lines(96:100, predict$mean, col = 4, lwd = 2)
> lines(96:100, predict$lower[,2], col = 6, lwd = 2, lty = 2)
> lines(96:100, predict$upper[,2], col = 6, lwd = 2, lty = 2)
```

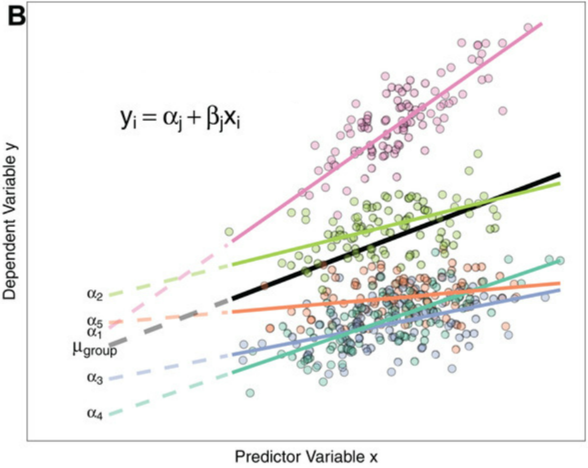


Interpretation of the coefficients in regression



https://peerj.com/articles/4794/

How about it?



https://peerj.com/articles/4794/

Motivation

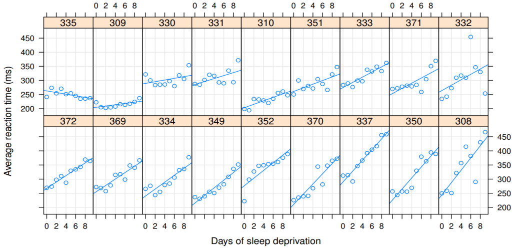
Research questions:

- Different intercepts or different slopes?
- Too many parameters lead the unstable estimation and explanation.
- Assume that we don't care about the exact values of slopes of different groups, we can construct a population of the slopes and make the inference via the distribution of the slopes.
- For example, the group index is the ID of the patients. Usually, the number of patients is larger than 5.

Example: sleepstudy

https://cran.r-project.org/web/packages/lme4/vignettes/lmer.pdf

- The average reaction time per day for subjects in a sleep deprivation study (Belenky et al. 2003)
- On day 0 the subjects had their normal amount of sleep.
- Starting that night they were restricted to 3 hours of sleep per night.
- The response variable, Reaction, represents average reaction times in milliseconds (ms) on a series of tests given each Day to each subject.



Fixed effect models

Fixed effect models for each subject:

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \varepsilon_{ij}, \;\; i = 1, \ldots, 10, \;\; j = 1, \ldots, 18.$$

i is the index of days, and j is the index of subjects.

```
> round(ind.coef,2)
       Int Slope        Int Slope
335 263.03 -2.88 372 267.04 11.30
309 205.05  2.26 369 254.97 11.35
330 289.69  3.01 334 240.16 12.25
331 285.74  5.27 349 215.11 13.49
310 203.48  6.11 352 276.37 13.57
351 261.15  6.43 370 210.45 18.06
333 275.02  9.14 337 290.10 19.03
371 253.64  9.19 350 225.83 19.50
332 264.25  9.57 308 244.19 21.76
```

There are at least 36 parameters in the model.

Random effect models

Random effect models for each subject:

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \varepsilon_{ij}, \;\; i = 1, \ldots, 10, \;\; j = 1, \ldots, 18.$$

Note that $\beta_{0j} \sim N(\beta_0, \sigma_0^2)$, $\beta_{1j} \sim N(\beta_1, \sigma_1^2)$, and $\varepsilon_{ij} \sim N(0, \sigma^2)$.
Or,

$$\begin{pmatrix} \beta_{0j} \\ \beta_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \begin{bmatrix} \sigma_0^2 & \rho\sigma_0\sigma_1 \\ \rho\sigma_0\sigma_1 & \sigma_1^2 \end{bmatrix} \right).$$

There are 5 or 6 parameters in total.
Note that: The fitted values of $\hat{\beta}_{0j}$ and $\hat{\beta}_{1j}$ can be obtained be the conditional expectations $E(\beta_{0j} | x_{ij}, y_{ij})$ and $E(\beta_{1j} | x_{ij}, y_{ij})$ for Subject j.

Mixed effect models

Mixed effect models for each subject j:

$$y_{ij} = \beta_{0j} + \beta_{1j} x_{ij} + \varepsilon_{ij}, \;\; i = 1, \ldots, 10, \;\; j = 1, \ldots, 18.$$

Either $\beta_{0j} \sim N(\beta_0, \sigma_0^2)$ or $\beta_{1j} \sim N(\beta_1, \sigma_1^2)$. It means parts are fixed effects and parts are random effects.

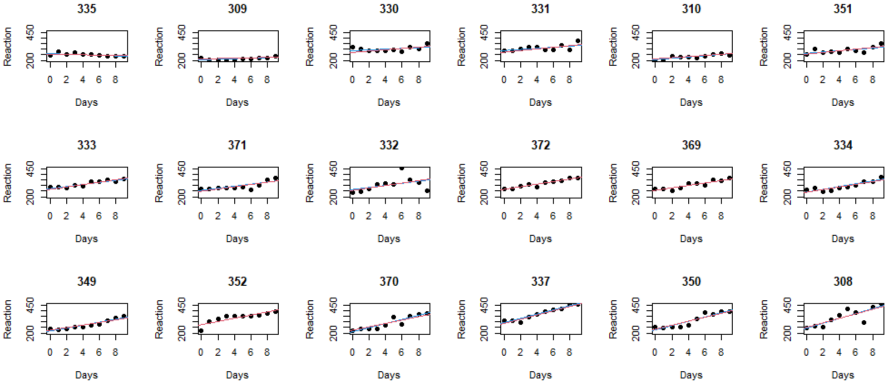
Note that: The fitted values of $\hat{\beta}_{0j}$ and $\hat{\beta}_{1j}$ can be obtained be the conditional expectations $E(\beta_{0j} | x_{ij}, y_{ij})$ or $E(\beta_{1j} | x_{ij}, y_{ij})$ for Subject j.

Fitting of random effect model by lme4

```
> fm1 <- lmer(Reaction ~ Days + (Days | Subject), data = sleepstudy)
> summary(fm1)
Linear mixed model fit by REML ['lmerMod']
Formula: Reaction ~ Days + (Days | Subject)
   Data: sleepstudy

REML criterion at convergence: 1743.6

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.9536 -0.4634  0.0231  0.4634  5.1793

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 Subject  (Intercept) 612.10   24.741
          Days         35.07    5.922   0.07
 Residual             654.94   25.592
Number of obs: 180, groups:  Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  251.405      6.825  36.838
Days          10.467      1.546   6.771

Correlation of Fixed Effects:
     (Intr)
Days -0.138
```

Fitting results

Blue: fixed effect models, Red: random effect models



compare some possible models

```
> anova(fm1,fm2,fm3)
refitting model(s) with ML (instead of REML)
Data: sleepstudy
Models:
fm2: Reaction ~ Days + (1 | Subject)
fm3: Reaction ~ Days + ((1 | Subject) + (0 + Days | Subject))
fm1: Reaction ~ Days + (Days | Subject)
    npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)    
fm2    4 1802.1 1814.8 -897.04   1794.1                          
fm3    5 1762.0 1778.0 -876.00   1752.0 42.0754  1  8.782e-11 ***
fm1    6 1763.9 1783.1 -875.97   1751.9  0.0639  1     0.8004    
```

Non-linear functions

The common structure of the statistical Model:

$$Y = f(x_1, x_2, \ldots, x_p) + \varepsilon(x_1, x_2, \ldots, x_p).$$

1. linear form (linear function of unknown parameters)

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2,$$

$$E(y) = \beta_0 + \beta_1 x_1 + exp(\beta_2) x_2,$$

2. nonlinear form: $E(y) = \exp\{\theta_1 x_1 \exp(-\theta_2 x_2)\}$

Estimation methods:

- Least squares methods (package: nls)
- Maximum likelihood methods (package: nlme)

Important: Select a suitable objective function!

Example: Growth curves for bacteria

The logistic growth curves:

$$y(t) = \frac{ky_0}{y_0 + (k - y_0)e^{-rt}} + \varepsilon,$$

where $[y_0, r, k]$ are the model parameters. Estimation methods:

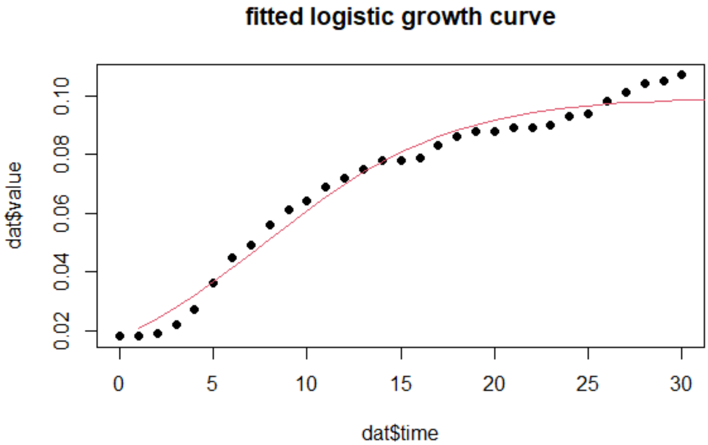
- Least squares methods (package: nls)
  objective function is

$$\sum_{k=1}^{n} \left(y(t) - \frac{ky_0}{y_0 + (k - y_0)e^{-rt}} \right)^2$$

- Maximum likelihood methods (package: nlme)

Important: Select a suitable objective function!

Goals: Growth curves for bacteria



Codes by yourself via optim()

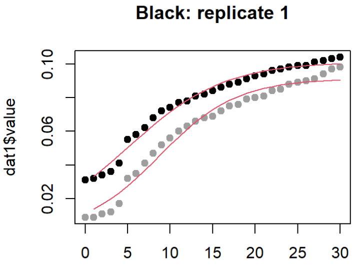
```
> obj.grow.logistic <- function(pars, time, value){
+   y0 <- pars[1]
+   r <- pars[2]
+   k <- pars[3]
+
+   y <- k*y0/(y0 + (k-y0)*exp(-r*time))
+   return(sum((y-value)^2))
+ }
>
> opt <- optim(c(0.01, 0.2, 0.1), obj.grow.logistic,
+              time = dat$time, value = dat$value )
> opt$par
[1] 0.01748257 0.20007333 0.09962540
```

Codes via nls() and nlme()

```
> ## fit by nls and nlme
>
> nls.opt <- nls(value~grow.logistic(y0, r, k, time), data = dat,
+                start = list(y0 = 0.01, r = 0.2, k = 0.1))
> coef(summary(nls.opt))
     Estimate  Std. Error  t value     Pr(>|t|)
y0 0.01748254 0.001580910 11.05853 9.980316e-12
r  0.20007090 0.013978851 14.31240 2.097144e-14
k  0.09962591 0.001849493 53.86659 7.967196e-30
>
> library(nlme)
> nlme.opt <- nlme(value~grow.logistic(y0, r, k, time), data = dat,
+                  fixed = y0 + r + k ~ 1, groups = ~ strain,
+                  start = c(y0 = 0.01, r = 0.2, k = 0.1))
>
> coef(nlme.opt)
          y0         r          k
D 0.01748374 0.2000552 0.09962776
```

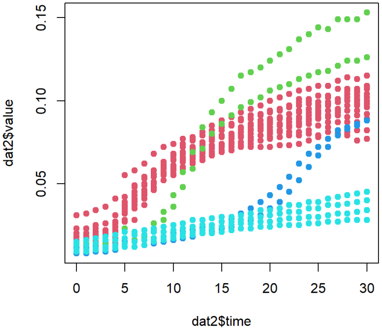
More on nlme() (1)

```
nlme.opt <- nlme(value~grow.logistic(y0, r, k, time), data = dat1,
                 fixed = y0 + r + k ~ 1,
                 random = y0+ r+ k~ 1,
                 groups = ~ replicate,
                 start = c(y0 = 0.01, r = 0.2, k = 0.1))
coef(nlme.opt)
coe.n <- coef(nlme.opt)

lines(1:30, grow.logistic(
  coe.n[1,1], coe.n[1,2], coe.n[1,3], 1:30), col = 2)
lines(1:30, grow.logistic(
  coe.n[2,1], coe.n[2,2], coe.n[2,3], 1:30), col = 2)
```



More on nlme() (2)

```
> nlme.opt.all <- nlme(value~grow.logistic(y0, r, k, time), data = dat2,
+                     fixed = y0 + r + k ~ 1,
+                     random = y0+ r+ k~ 1,
+                     groups = ~ groups,
+                     start = c(y0 = 0.01, r = 0.2, k = 0.1))
Warning message:
In nlme.formula(value ~ grow.logistic(y0, r, k, time), data = dat2,  :
  Iteration 2, LME step: nlminb() did not converge (code = 1). Do increas
e 'msMaxIter'!
> nlme.opt.all
Nonlinear mixed-effects model fit by maximum likelihood
  Model: value ~ grow.logistic(y0, r, k, time)
  Data: dat2
  Log-likelihood: 2608.506
  Fixed: y0 + r + k ~ 1
         y0           r           k
0.009698763 0.158771903 0.155711395

Random effects:
 Formula: list(y0 ~ 1, r ~ 1, k ~ 1)
 Level: groups
 Structure: General positive-definite, Log-Cholesky parametrization
         StdDev      Corr
y0       0.004303459 y0     r
r        0.088014407 -0.315
k        0.043506207 -0.311 -0.804
Residual 0.007041115

Number of Observations: 744
Number of Groups: 4
```



More on nlme() (2)