

Hierarchical Models in Linear and Logistic Regression

Basic Ideas

The literature on random effects models (term generally used by frequentists) or hierarchical models (term generally used by Bayesians) is huge, deserving of an entire course (or two) by itself. We will focus on getting the basic idea, and some simple examples.

What do we mean by a hierarchical or random effects model?

These sorts of models refer to when the parameters of models themselves have distributions (hence there is a hierarchy of parameters), or the parameters come from random distributions (hence the term random effects).

For example, consider the basic simple linear regression model:

$$Y = \alpha + \beta X$$

This model assumes that for each individual contributing data, the effects (both intercept and effect of X as given by β) are identical. In other words, all subjects are subject to the identical intercept and slope.

It is easy to imagine that this is often not the case. Consider, for example, giving an individual a drug to increase their bone mineral density (BMD). Now, in a simple model we might assume that β is constant across individuals, but, more realistically, one might consider β as a mean or average effect, and the effects for each individual (owing to body chemistry, other medications they may be taking, etc.) may be different.

Now, if we knew all of these effects, we may begin creating interaction models, which would allow the slope (effect of the drug on BMD) to depend on other characteristics. But if there are one or more *unknown* effects, we can create a hierarchical or random effects model as follows:

$$Y = \alpha + \beta_{fixed}X + \beta_{random}X$$

$$\beta_{random} \sim N(0, \sigma^2)$$

or equivalently

$$Y = \alpha + \beta X$$

$$\beta \sim N(\mu_\beta, \sigma^2)$$

Even simpler, we can have random effects / hierarchical models on the parameters of a normal distribution. Consider the following example:

Suppose that *mean* diastolic blood pressure varies randomly in a population with a mean of 70 mmHg, and standard deviation 10 mmHg. Suppose that in addition, however, each individual's blood pressure also varies throughout the day about their own individual means, with a standard deviation of 5 mmHg.

How could we model this situation?

Suppose we have m measures on each of n subjects. The following model seems to mirror the two sources of variability:

$$x_{ij} \sim N(\mu_i, \sigma_w^2), \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m$$

$$\mu_i \sim N(\mu, \sigma_b^2), \quad i = 1, 2, \dots, n$$

Given some data and the above model, we could estimate the parameters from either a frequentist or Bayesian perspective. We will simulate some blood pressure data, and then show how estimation is done in WinBUGS.

```
# Simulate data for each individual in two stages:

# First stage: Randomly select means for each of 100 subjects:

> bp.means <- rnorm(100, mean=70, sd=10)

> bp.means
[1] 76.27620 73.28998 81.77357 71.82937 65.39472 ... etc ....

# Second stage, generate 10 individual data points
# for each of 100 subjects, one subject per row of the data set
```

```

# Create an empty vector to store individual data

> bp.data <- rep(NA, 1000)

# Fill in vector with individual values of correct means

> for(i in 1:100) {
  for(j in 1:10) {
bp.data[10*(i-1) + j] <- rnorm(1, mean=bp.means[i], sd=5)
  }
}

# Note how bp.means match each row

> bp.data
 [1] 78.491 81.989 77.345 78.327 75.952 86.086 81.257 75.299 69.783 71.300
[11] 72.918 70.602 65.253 70.061 75.387 69.744 68.688 68.035 67.458 82.689
[21] 82.443 78.124 71.675 77.090 90.796 86.059 81.560 83.870 82.229 76.975
[31] 70.801 79.688 81.731 77.678 73.670 69.109 75.066 68.126 80.726 68.882
[41] 66.379 60.884 69.669 69.959 59.245 61.227 71.388 76.070 64.878 66.273
[51] 63.037 58.527 59.353 54.285 59.518 61.577 55.452 65.067 62.005 65.482
..... etc .....

```

We will now create a program to analyze these data in WinBUGS:

```

# First convert data set from R for use in WinBUGS

> dput(bp.grouped$bp, file="g:\\bp.txt")

# We will cut and paste this file into a WinBUGS analysis window.

```

Write the program for WinBUGS:

```

model {

for( i in 1:100)      # Loop over 100 subjects {

for (j in 1:10)      # Loop over 10 data points per subset
  {
    # Note [j,i] not [i,j]
    bp[j,i] ~ dnorm(mu[i] , tau.w) # Data are normally distributed
  }
  # about individual mean mu[i]
  mu[i] ~ dnorm(mean.bp, tau.b) # Means are normally distributed
}
# with overall mean mean.bp

mean.bp ~ dnorm(70, .001)      # Prior for mean BP, var = 1/0.001 = 1000

tau.w <- 1/(sigma.w*sigma.w)   # Prior for within SD (precision)
sigma.w ~ dunif(0,100)

tau.b <- 1/(sigma.b*sigma.b)   # Prior for between SD (precision)
sigma.b ~ dunif(0,100)

}

# Initial Values

list(sigma.w = 7, sigma.b = 12, mean.bp = 70)

# Data

# Note some changes needed from simple dput file:
# Add .Data word, reverse .Dim

list(bp=structure(.Data=c(78.491534163798, 72.9179967852737,
82.4433694101257, 70.8014851147823, 66.3791467338957,
63.0373238043374, 67.4878552683662, 85.0656743211218,
79.0398796878923, 78.5438769720441, 78.7772194526441,
71.9931192939269, 53.0873697753945, 59.7622998188586,
..... etc .....
77.5633200078565, 58.1785984593208, 67.7575083592776,
78.578700312957, 60.9872446515102), .Dim = c(10, 100)))

# Results

node      mean      sd      MC error  2.5%      median  97.5%  start  sample
mean.bp  70.59    0.9571  0.009583  68.73     70.58   72.47  1001   10000
mu[1]    77.39    1.616   0.0165    74.22     77.4    80.57  1001   10000

```

mu[2]	71.07	1.615	0.01547	67.95	71.07	74.2	1001	10000
mu[3]	80.74	1.632	0.01429	77.55	80.73	83.95	1001	10000
mu[4]	74.44	1.614	0.01631	71.28	74.42	77.55	1001	10000
mu[5]	66.7	1.644	0.0169	63.49	66.73	69.89	1001	10000
..... etc								
mu[99]	72.54	1.614	0.01439	69.4	72.53	75.73	1001	10000
mu[100]	61.82	1.606	0.01577	58.68	61.81	64.94	1001	10000
sigma.b	9.592	0.7088	0.007846	8.305	9.553	11.1	1001	10000
sigma.w	5.204	0.1227	0.001459	4.968	5.203	5.453	1001	10000
tau.b	0.01104	0.001616	1.768E-5	0.008123	0.01096	0.0145	1001	10000
tau.w	0.03698	0.001744	2.065E-5	0.03363	0.03693	0.04052	1001	10000

Note the overall excellent results across all parameters.

Advantage of hierarchical models

Consider the BP example again. At one extreme, we could consider that each person has their own mean, and estimate the means 10 data points at a time. In other words, no pooling of data.

Advantage: Unbiased individual estimates. Disadvantage: Low accuracy

At the other extreme, we can consider all subjects similar enough, and just estimate a single pooled estimate.

Advantage: High accuracy. Disadvantage: No individual estimates.

Hierarchical estimates are a compromise between these two extremes. One lets the data decide on the degree of pooling. The more similar the mean estimates are, the smaller σ_b will be, and the more pooling takes place. The more different the individual means are, the larger σ_b will be, and less pooling takes place.

Advantages: High accuracy while retaining individual estimates.

Disadvantage: Some slight bias may be introduced to individual estimates, although overall Mean Square Error is optimized.

Example of “bias” in BP example:

Compare mean estimates from “straight average of 10 individual values” to means estimated by hierarchical (random effects) model:

Subject	Mean Ind.	Mean Hier Model
1	77.58343	77.39
2	71.08405	71.07
3	81.0827	80.74
4	74.548	74.44
5	66.59765	66.70

Recall that the overall mean was 70.59, and note that all estimates from the hierarchical model are slightly moved towards that value. This is the *slight* effect of pooling here, only slight because σ_b was rather large. If we repeat the exercise with a smaller values for σ_b , this “borrowing of strength” or “movement towards the middle would be larger.

The effect will also tend to be larger when there is a smaller number of observations per subject.

Hierarchical Models in Logistic Regression

Motivation by Example

Suppose that we have a data set where nine different MD’s made decisions on 133 patients in total. On average, about 15 patients per MD, although the data were not evenly distributed across MDs. The decisions involved whether to give a patient thrombolysis or not in the emergency room.

There are Canadian guidelines as to whether to thrombolyse such patients or not, so each physician was evaluated as to whether they followed the guidelines for each subject they treated or not. For example, the first physician followed the guidelines in 19 out of 20 patients she/he treated. While each physician has their own rate of “success” (following the guidelines), it may be that overall, these rates may themselves follow a pattern, i.e., have a statistical distribution.

The question is:

How can we best estimate each physician’s rate of correct decision making?

If we use each physician’s rate separately, then the sample sizes will be too small.

For example, the largest number of patients contributed by any MD was 24, and so

we would expect a very wide interval estimate from even the best case scenario.

On the other hand, it is probably quite unreasonable to assume that the true rate of correct decision making is constant across physicians, as each has their own training, background, and so on. Thus, we expect rates to differ across physicians.

So, pooling of all data would provide an accurate estimate, but ignores individual variability. Individual estimates take each MD's ability into account, but will lead to very poor accuracy owing to small sample sizes.

Is there a way to address this problem?

One answer is to use a hierarchical/random effects model, which allows us to reasonably accurately estimate both individual and overall rates of guideline adherence, in exchange for an assumption: That the nine MDs are like a sample from a group of MDs that collectively have rates that follow a statistical distribution. This "higher level" distribution is why these models are called "hierarchical models".

The data are from the MSc thesis of Dr. Michael Schull.

Model We will now program this problem using a hierarchical model in WinBUGS.

```

model                                # Usual model statement in WinBUGS
{
  for (i in 1:nmd) {                 # Loop over number of MDs = 9
    x[i] ~ dbin(p[i],n[i])           # For each MD, number of
                                     # successes is binomial
                                     # with size = n[i], rate = p[i]

    logit(p[i]) <- z[i]              # Create a hierarchical
    z[i] ~ dnorm(mu,tau)              # distribution, by letting
    }                                  # logit of rates be normally
                                     # distributed
                                     # z is a "transformed" variable
                                     # representing the rates p[i]
                                     # on the logit scale

    mu ~ dnorm(0,0.001)              # Prior distribution for mu
    tau ~ gamma(0.001, 0.001)        # Prior distribution for tau
    y ~ dnorm(mu, tau)               # Predictive distribution for rate on logit scale
    sigma <- 1/sqrt(tau)             # Prior SD on the logit scale, FROM TAU
    w <- exp(y)/(1+exp(y))           # Predictive dist transformed back to prob scale
  }

```

Data

```
list(n=c( 20, 6, 24, 13, 12, 4, 24, 12, 18),
     x=c( 19, 5, 22, 12, 11, 4, 23, 12, 16),
     nmd=9)
```

Initial Values

```
list(mu=0, sigma=1)
```

Results

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	2.642	0.3716	0.0294	1.973	2.63	3.357	1001	5000
tau	215.9	403.8	21.61	1.189	51.67	1447.0	1001	5000
sigma	0.2245	0.2467	0.01485	0.02631	0.1393	0.9187	1001	5000
p[1]	0.9301	0.02673	0.0018	0.8693	0.9332	0.9718	1001	5000
p[2]	0.9242	0.03447	0.002022	0.8463	0.9294	0.9689	1001	5000
p[3]	0.9277	0.0266	0.001763	0.8665	0.9316	0.9684	1001	5000
p[4]	0.9281	0.02856	0.001852	0.8598	0.9321	0.9712	1001	5000
p[5]	0.9279	0.0286	0.001851	0.8635	0.9319	0.9721	1001	5000
p[6]	0.9293	0.02987	0.001896	0.8636	0.9333	0.9758	1001	5000
p[7]	0.9313	0.02574	0.001803	0.8745	0.9343	0.9739	1001	5000
p[8]	0.932	0.02688	0.001838	0.8748	0.9347	0.9779	1001	5000
p[9]	0.9257	0.0288	0.001872	0.8595	0.93	0.9677	1001	5000
w	0.927	0.03535	0.001954	0.8524	0.9324	0.9725	1001	5000
y	2.638	0.4917	0.02993	1.753	2.624	3.565	1001	5000

Rerunning above, but with data changed so that first row is 10/20. Note how this “outlier” is not pulled back as strongly towards the overall mean.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	2.347	0.6308	0.02193	1.315	2.286	3.806	1001	4000
p[1]	0.5969	0.1161	0.003224	0.366	0.6	0.8094	1001	4000
p[2]	0.8614	0.0981	0.001751	0.6063	0.8834	0.9825	1001	4000
p[3]	0.9076	0.05137	0.001297	0.7848	0.9152	0.9808	1001	4000
p[4]	0.9075	0.06156	0.001603	0.7596	0.9193	0.9889	1001	4000
p[5]	0.9027	0.06282	0.001518	0.7504	0.9153	0.9875	1001	4000
p[6]	0.9104	0.07954	0.001975	0.6975	0.9322	0.9972	1001	4000
p[7]	0.933	0.04373	0.001305	0.8296	0.9413	0.9918	1001	4000
p[8]	0.9404	0.05143	0.001714	0.8086	0.9537	0.9982	1001	4000
p[9]	0.8888	0.06163	0.001315	0.7372	0.8989	0.9775	1001	4000
sigma	1.205	0.5925	0.02313	0.366	1.106	2.64	1001	4000
tau	1.525	2.806	0.1283	0.1441	0.8176	7.476	1001	4000
w	0.8579	0.1527	0.002682	0.385	0.9027	0.9957	1001	4000
y	2.339	1.452	0.03215	-0.4683	2.228	5.446	1001	4000

Rerunning above, but with data changed so that first row is 100/200. Now very strong outlier, results stays near 50% observed rate.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
mu	2.399	0.6767	0.01764	1.236	2.345	3.935	1001	4000
p[1]	0.5088	0.03571	5.547E-4	0.4406	0.5092	0.5817	1001	4000
p[2]	0.8565	0.1035	0.002056	0.5942	0.8801	0.9838	1001	4000
p[3]	0.9104	0.05155	9.935E-4	0.7889	0.9198	0.9822	1001	4000
p[4]	0.91	0.06435	0.00122	0.746	0.9248	0.99	1001	4000
p[5]	0.9066	0.06611	0.001272	0.7396	0.9202	0.9895	1001	4000
p[6]	0.9191	0.08212	0.002	0.6984	0.9439	0.9985	1001	4000
p[7]	0.9397	0.04166	8.286E-4	0.8367	0.9485	0.9934	1001	4000
p[8]	0.9499	0.04702	0.001143	0.8217	0.9635	0.9989	1001	4000
p[9]	0.8894	0.06441	0.001169	0.7325	0.902	0.9784	1001	4000
sigma	1.433	0.6118	0.01963	0.6696	1.311	3.024	1001	4000
tau	0.7414	0.5728	0.01634	0.1097	0.5827	2.242	1001	4000
w	0.8457	0.1816	0.003351	0.2951	0.9117	0.9977	1001	4000
y	2.396	1.714	0.03377	-0.8707	2.335	6.091	1001	4000

So, to summarize:

The hierarchical model allowed one to estimate the rate of each physician reasonably accurately, assuming the model is reasonable.

When all rates were similar, sigma is small, and rates are bunched together (a lot of “borrowing of strength”). When one (or in general, any number) rate sticks out from the rest, the degree of “pull” back towards the overall mean depends on how strong the result is (which in turn, depends on how different it is, and on the sample

sizes. Note that sigma grows in size as the individual rates are spread further apart, meaning less “pull” back towards the overall mean.

Using this model, can also predict what the overall rate is, as well as the rate for the “next similar physician”.

Example 2: Hierarchical Logistic Regression

The above example used a logit function, but there were no covariates. as we saw for linear regression, on can also place hierarchical components on intercepts and beta coefficients.

Consider the following example:

We would like to estimate the rates of osteoporosis in Canada, but believe that they differ not only by age and sex (higher rates in older females compared to younger males), but also by province.

We have a data set of 10,000 female subjects, 1000 from each province. For each subject, we have their age, province, and whether they have been diagnosed with osteoporosis.

The data are in a file called `osteo.txt`, and we will create a hierarchical model that accounts for possible differences across provinces, and provides probabilities that one province has a higher rate compared to any other province.

The data structure is as follows:

The first 1000 data points are from Newfoundland, the next 1000 from Nova Scotia, the next 1000 from New Brunswick, and then PEI, Quebec, Ontario, Manitoba, Saskatchewan, Alberta, and BC, in that order.

```

model                                     # Usual model statement in WinBUGS
{
  for (j in 1:10)                         # Loop over 10 provinces
  {
                                          # Common index trick
    for (i in index[j]:index2[j]) # Index for jth province
    {
      logit(p[i]) <- alpha[j] + beta*age[i] # Logit for individual probability
      osteo[i] ~ dbern(p[i])                # Likelihood function for ith individual
    }
    alpha[j] ~ dnorm(mu, tau)               # Hierarchical component: provincial rates
  }
  mu ~ dnorm(0,0.001)                      # "tied together" through normal distribution
  tau <- 1/(sigma*sigma)                   # Prior on hierarchical mean
  sigma ~ dunif(0,20)                      # Needed for WinBUGS
  beta ~ dnorm(0, 0.001)                   # Prior for hierarchical Sd
  beta ~ dnorm(0, 0.001)                   # Prior for beta
  pred.NFLD.50 <- exp(alpha[1] + beta*50)/(1+exp(alpha[1] + beta*50))
  pred.QUEBEC.50 <- exp(alpha[5] + beta*50)/(1+exp(alpha[1] + beta*50))
  pred.BC.50 <- exp(alpha[10] + beta*50)/(1+exp(alpha[10] + beta*50))
  diff.quebec.bc <- pred.QUEBEC.50 - pred.BC.50 # Quebec - BC diff
  p.quebec.bc <- step(pred.QUEBEC.50 - pred.BC.50) # p(Quebec > BC)
}

# Inits

list(alpha=c(0,0,0,0,0,0,0,0,0,0,0), beta=0.5, mu=0, sigma = 1)

# Data

list(index = c(1, 1001, 2001, 3001, 4001, 5001, 6001, 7001,
8001, 9001),
index2 =c(1000, 2000, 3000, 4000, 5000, 6000, 7000,
8000, 9000, 10000),
age=c(39, 25, 36, 68, 67, 50, 39, 68, 31, 30, 54, 30, 30,
71, 30, 28, 49, 54, 29, 61, 35, 52, 29, 74, 70, 50, 41, 74, 74,
38, 29, 41, 27, 28, 57, 32, 32, 64, 45, 60, 56, 66, 65, 40, 64,
58, 51, 30, 48, 63, 50, 52, 67, 47, 44, 37, 41, 32, 35, 27, 58,
.....etc.....
55, 40, 27, 33, 37, 39, 28, 75, 37, 52, 57, 67, 25, 33, 34, 55,
39, 46, 28), osteo=c(0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,
0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0,
0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0,
0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1,

```

```

.....etc.....
0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0,
1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1,
0, 0, 1, 1, 1, 1, 1, 1, 1))

```

The results are as follows:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha[1]	-0.6307	0.09387	0.003661	-0.8123	-0.6306	-0.4466	1001	10000
alpha[2]	-0.5916	0.09053	0.00362	-0.7634	-0.5923	-0.414	1001	10000
alpha[3]	-0.437	0.08762	0.003624	-0.6053	-0.4376	-0.2607	1001	10000
alpha[4]	-0.4934	0.08708	0.003542	-0.6622	-0.4931	-0.3201	1001	10000
alpha[5]	-0.4047	0.0882	0.003568	-0.5745	-0.4054	-0.2329	1001	10000
alpha[6]	-0.3533	0.09058	0.003677	-0.5269	-0.3532	-0.1722	1001	10000
alpha[7]	-0.3739	0.09001	0.003639	-0.5478	-0.3743	-0.1951	1001	10000
alpha[8]	-0.4318	0.08846	0.003605	-0.6019	-0.433	-0.2543	1001	10000
alpha[9]	-0.4584	0.0879	0.003628	-0.627	-0.4588	-0.2862	1001	10000
alpha[10]	-0.5	0.087912	0.003544	-0.6734	-0.5013	-0.3295	1001	10000
beta	0.002226	0.001352	6.719E-5	-4.721E-4	0.002239	0.004797	1001	10000
diff.quebec.bc	0.06421	0.03328	4.504E-4	0.003522	0.06245	0.1322	1001	10000
mu	-0.4677	0.08147	0.003604	-0.6229	-0.4686	-0.3052	1001	10000
p[1]	0.3674	0.01559	3.223E-4	0.3369	0.3675	0.3984	1001	10000
p[2]	0.3602	0.01694	4.869E-4	0.3275	0.3602	0.394	1001	10000
p[3]	0.3659	0.01579	3.533E-4	0.3348	0.3659	0.3973	1001	10000
p[4]	0.3825	0.01654	3.561E-4	0.3499	0.3826	0.415	1001	10000
p[9998]	0.3982	0.014	2.502E-4	0.3705	0.3983	0.4261	1001	10000
p[9999]	0.4019	0.01366	1.677E-4	0.3749	0.402	0.4289	1001	10000
p[10000]	0.3924	0.0152	4.08E-4	0.3623	0.3923	0.4223	1001	10000
p.quebec.bc	0.9823	0.1319	0.001347	1.0	1.0	1.0	1001	10000
pred.BC.50	0.4041	0.01363	1.404E-4	0.377	0.4041	0.4311	1001	10000
pred.NFLD.50	0.3731	0.01532	2.523E-4	0.3428	0.3731	0.4033	1001	10000
pred.QUEBEC.50	0.4683	0.03007	3.949E-4	0.4144	0.4666	0.5307	1001	10000
sigma	0.1212	0.04609	8.6E-4	0.05246	0.114	0.2299	1001	10000
tau	172.6	1786.0	69.06	18.95	76.94	364.7	1001	10000

One could have predicted rate for next person from each province of a given age, and so on. Note that sigma was relatively large (on the logit scale) indicating substantial differences in rates between provinces. Note that Quebec is highly likely to have a larger rate than BC, with probability = 0.98. Also, note the very direct statement arising from this variable, in comparison with p -values.

Situations where random effects or hierarchical models can be used

- Repeated measures models: Random effects/hierarchical models are especially useful when subjects are followed over time, where each subject will contribute their own regression over time, so each subject will have their own individual intercept and slope. Putting a hierarchical model on the intercepts and/or slopes will improve individual estimates (again, at the expense of a small amount of individual bias, and *decreased* overall mean square error).
- Nested data collection: Suppose patients are nested within doctors treating them, the doctors are nested within hospitals where they work, hospitals are nested within health regions, which are nested within cities, which are nested within provinces, which are nested within the same country. A multiple layers hierarchical model can then be used to take into account the complex data structure.
- Unknown interactions: A medication, for example, may act differently on each and every subject (indeed, this is almost always the case, although individual variation may be large or small). If factors which may effect how the drug works for a given individual are unknown or unrecorded, then a hierarchical model can be used (as in our introductory example).
- One can have multiple level hierarchical models, accounting for hierarchical data structure. For a detailed example see the paper:

Brophy J, Joseph L, Theroux P, on behalf of the Quebec Acute Coronary Care Working Group. Medical decision making about the choice of thrombolytic agent for Acute Myocardial Infarction. *Medical Decision Making* 1999;19(4):411-418.

This paper used a four level model. It was programmed in WinBUGS, and formed part of the PhD thesis of Dr. James Brophy.

- As discussed for linear regression models, one can almost always argue for a hierarchial component to any regression model in epidemiology: Subjects are treated by different physicians, in different hospitals, at different times, in different provinces, and so on.

Overall, hierarchical modelling is a very important topic, we have just scratched the surface here, but be on the lookout for opportunities to better your models through addition of hierarchical levels.

In short, in epidemiology, one can argue that just about *every* model should include a hierarchical component, and, indeed, these models are becoming extremely popular.