

Multiple Linear Regression

Edps 590BAY

Carolyn J. Anderson

Department of Educational Psychology



©Board of Trustees, University of Illinois

Fall 2021

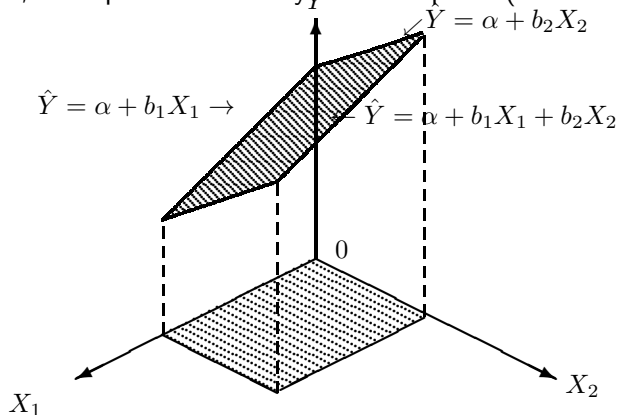
I Overview

- Multiple regression
- Model evaluation
- Model comparison
- Practice

Depending on the book that you select for this course, read either Gelman et al. pp xx or Kruschke Chapters chapters 13, 15 & 16 . Also I used the coda and jags, rjags, runjags and jagsUI manuals.

I Multiple Regression

If we have more than one predictor, we can add them to our model. For example, for 2 predictors we try to find a plane (rather than a line).



I Multiple Regression as a GLM

$$\begin{aligned}
 y_i &= b_0 + b_1x_{1i} + b_2x_{2i} + \dots + b_kx_{ki} + \epsilon_i \\
 &= \mu_i + \epsilon_i
 \end{aligned}$$

- **Random Component:** y is the response/outcome variable. We assume that $\epsilon_i \sim N(0, \sigma^2)$ so $y_i \sim N(\mu_i, \sigma^2)$.
- **Linear Predictor (Systematic component)** is

$$b_0 + b_1x_{1i} + b_2x_{2i} + \dots + b_kx_{ki}$$

- **Identity link:**

$$g(E(y_i)) = E(y_i) = \mu_i = b_0 + b_1x_{1i} + b_2x_{2i} + \dots + b_kx_{ki}$$

I NELLS: Exploratory Analysis

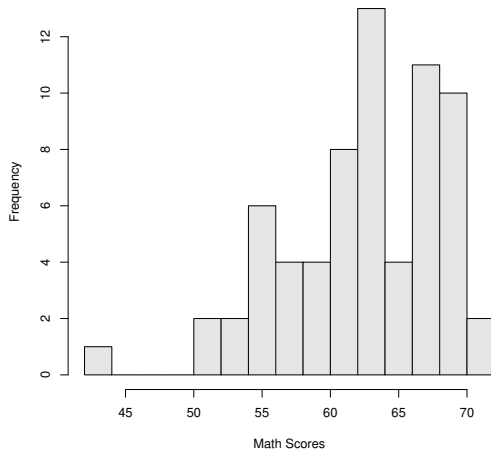
- We'll continue with the NELLS example.
- Before modeling the data, we should do a little exploratory analysis.
- Basic descriptive statistics of math scores:

N	$\bar{y}$	sd	var	min	median	max
67	62.8209	5.6754	32.3099	43.00	63.00	71.00

- Histogram (next slide)

I Distribution of Math Scores

NELS Math Scores



I Possible Predictor Variables

Information about variables:

- **sex**: 1 =male, 2 =female
- **race**: 1 =Asian/PI, 2 =Hispanic, 3 =Black, 4 =White. It would be best to dichotomize (white/not-white).
- **Time spent doing homework**: 0 =none, 1 = less then 1 hr, 2 =1 hour, 3 =2 hours, 4 = 3 hours, 5 =4 to 6 hours, 6 =7 to 9 hours, 7=more than 10 hours. This is ordinal, but we'll treat as numerical (i.e., "continuous").
- **ses**: I think this is composite of income, parent education, etc. We'll treat as numerical (i.e., "continuous").
- **Parents education**: 3 =HS (5), 4 = college grade (17), 5 =masters (24), 6 =doctorate (21). This may look odd, but this is a an urban private school in north central US. Ordinal but we may treat as numerical (i.e., "continuous").

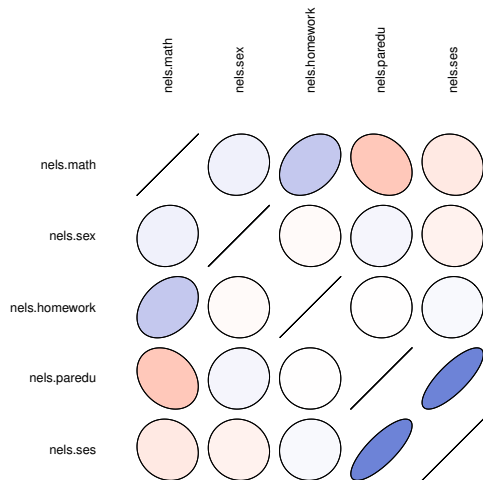
I Descriptive Statistics Predictor Variables

		N	$\bar{x}$	$sd(x)$	$\min(x)$	$\max(x)$
Sex	male	36				
	female	31				
Race	non-white	7				
	white	60				
Time homework		67	3.30	1.72	0	6
ses		67	1.04	0.46	-0.35	1.85
Parent education		67	4.91	0.93	3	6

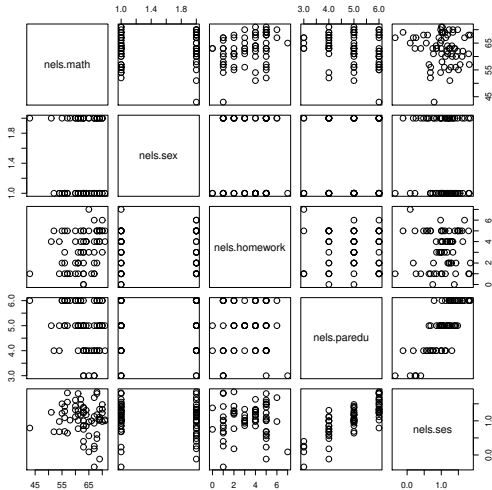
Correlations between Variables

	math	homework	paredu	ses
math	1.00	.33	-.33	-.10
homework	.33	1.00	.00	.04
paredu	-.26	.00	1.00	.79
ses	-.10	.04	.79	1.00

I Look at Correlations



Another look at Bi-variate Relationships



I An OLS of math

```
ols.lm ← lm(math gender+ses+paredu+homework+white,
data=nels)
```

Min	1Q	Median	3Q	Max
-13.8831	-2.4426	0.3711	3.4205	8.9577

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	70.3048	5.0930	13.804	< 2e-16 ***
gender2	1.5486	1.3083	1.184	0.24115
ses	3.6973	2.5384	1.457	0.15037
paredu	-3.0370	1.2020	-2.527	0.01413 *
homework	1.0629	0.3746	2.838	0.00616 **
white1	-0.7322	2.2943	-0.319	0.75072

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

Residual standard error: 5.227 on 61 degrees of freedom

Multiple R-squared: 0.2161, Adjusted R-squared: 0.1518

F-statistic: 3.363 on 5 and 61 DF, p-value: 0.009523

I JAGS: dataList

Just add more to data list

```
dataList ← list( y =nels$math,
                 pared =nels$pared,
                 hmwk =nels$homework,
                 ses =nels$ses,
                 gender=nels$gender,
                 white = nels$white,
                 N=length(nels$math),
                 sdY = sd(nels$math)
               )
```

I JAGS: model

```
mlr1 = "model { for (i in 1:N) {
  y[i] ~ dnorm(mu[i] , precision)
  mu[i] ← b0 + bpared*pared[i] + bhmwk*hmwk[i]
    + bses*ses[i]+ bsex*gender[i] + bwhite*white[i] }
  b0 ~ dnorm(0 , 1/(100*sdY^2) )
  bpared ~ dnorm(0 , 1/(100*sdY^2) )
  bhmwk ~ dnorm(0 , 1/(100*sdY^2) )
  bses ~ dnorm(0 , 1/(100*sdY^2) )
  bsex ~ dnorm(0 , 1/(100*sdY^2) )
  bwhite ~ dnorm(0 , 1/(100*sdY^2) )
  sigma ~ dunif( 1E-3, 1E+30 )
  precision ← 1/sigma^2
}"
writeLines(mlr1, con='mlr.txt')
```

I JAGS: starting values

```

initsList = list(
  list("b0"=mean(nels$math), "bpared"=0, "bhmwk"=0,
    "bses"=0, "bsex"=0, "bwhite"=0,
    "sigma"=sd(nels$math)),
  list("b0"=rnorm(1,50,5), "bpared"=rnorm(1,-2,1),
    "bhmwk"=rnorm(1,2,1), "bses"=rnorm(1,0,1),
    "bsex"=rnorm(1,1,1), "bwhite"=rnorm(1,0,1),
    "sigma"=sd(nels$math)),
  etc. )

```

I JAGS: runjags

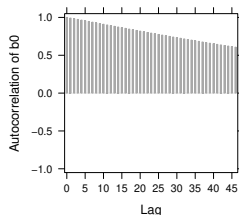
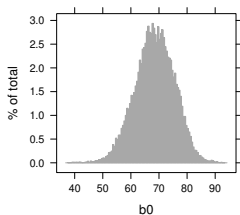
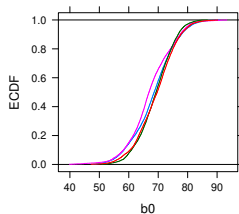
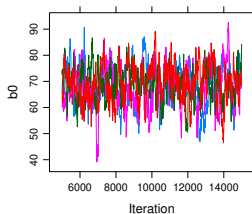
```
mlr1.runjags ← run.jags(model=mlr1,
  monitor=c("b0","bpared","bhmwk","bses",
    "bsex","bwhite","sigma","dic"),
  data=dataList,
  n.chains=4,
  inits=initsList)

plot(mlr1.runjags)
gelman.plot(mlr1.runjags)

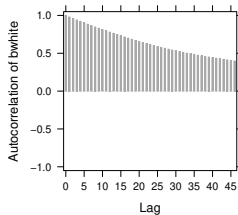
print(mlr1.runjags)
```

Look OK? No, Graphs showing not OK on following pages

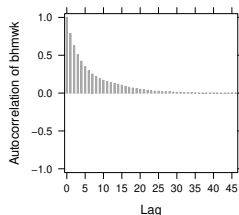
I Problems



Problem



Not All Bad, e.g.,



JAGS model summary statistics from 40000 samples (chains = 4; adapt+burnin = 5000):

	Lower95	Median	Upper95	Mean	SD	Mode
b0	54.566	68.759	81.282	68.524	6.7695	–
bpared	-5.2689	-2.8977	-0.36232	-2.8977	1.2387	–
bhmwk	0.32424	1.076	1.8235	1.0765	0.38455	–
bses	-1.9846	3.4508	8.336	3.4354	2.6071	–
bsex	-1.1208	1.5761	4.1695	1.5537	1.3494	–
bwhite	-4.9743	-0.52883	4.3163	-0.47132	2.3372	–
sigma	4.4271	5.2958	6.343	5.3354	0.49596	–

I Results

	MCerr	MC%ofSD	SSeff	AC.10	psrf
b0	0.46992	6.9	208	0.90431	1.0043
bpared	0.083361	6.7	221	0.89339	1.006
bhmwk	0.0062373	1.6	3801	0.17402	1.0005
bses	0.11288	4.3	533	0.7076	1.0076
bsex	0.030858	2.3	1912	0.39883	1.0009
bwhite	0.11038	4.7	448	0.80176	1.0058
sigma	0.0043969	0.9	12723	0.030837	1.0004

Model fit assessment:

DIC = 420.2391

PED not available from the stored object

Estimated effective number of parameters: $pD = 7.25924$

Total time taken: 6.0 seconds

I What Could We Try

- Try different priors.
- Add more iterations using `extend.jags`.
- Use thinning as option with `runjags`, maybe `thin=10`?
- See what `autorun.jags` yields.
- Drop variables that include 0 in their high density intervals.
- Use t-distribution.

I More iterations

```
mlr1.extend <- extend.jags(mlr1.runjags, burnin=0,
sample=500000)
```

JAGS model summary statistics from 2040000 samples (chains = 4; adapt+burnin = 5000):

	Lower95	Median	Upper95	Mean	SD	Mode
b0	55.396	68.525	82.17	68.484	6.8368	—
bpared	-5.369	-2.9114	-0.52627	-2.9094	1.2326	—
bhmwk	0.32381	1.0766	1.8294	1.0774	0.38401	—
bses	-1.8041	3.4867	8.4596	3.4878	2.6044	—
bsex	-1.0097	1.5677	4.2522	1.5725	1.3356	—
bwhite	-5.0723	-0.47575	4.1749	-0.4633	2.3434	—
sigma	4.4021	5.3	6.3216	5.3376	0.49789	—

I More iterations

	MCerr	MC%ofSD	SSeff	AC.10	psrf
b0	0.066779	1	10481	0.90838	1.0002
bpared	0.011467	0.9	11554	0.88977	1.0001
bhmwk	0.0013947	0.4	75807	0.16714	1
bses	0.020807	0.8	15668	0.69977	1.0001
bsex	0.0047707	0.4	78378	0.38238	1.0001
bwhite	0.018316	0.8	16369	0.80555	1.0001
sigma	0.001826	0.4	74349	0.038153	1.0001

Model fit assessment:

DIC = 420.2096

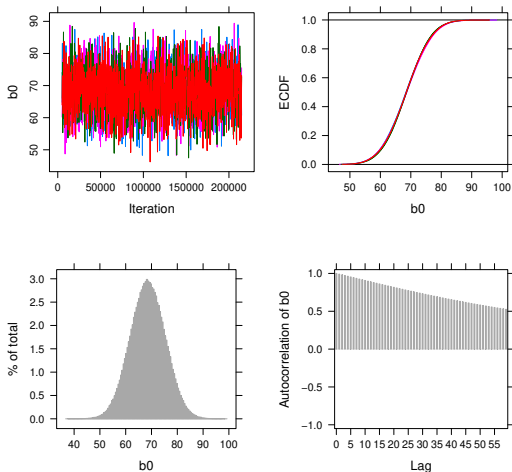
PED not available from the stored object

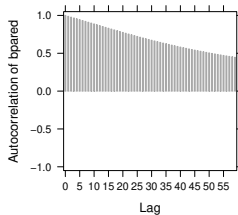
Estimated effective number of parameters: $pD = 7.22784$

Total time taken: 2.6 minutes

Better mixing but still some large auto-correlations—see figures.

I Still Problems, e.g.,





I Thinning

```
mod1.thin10 ← run.jags(model=model1,
  monitor=c("b0", "bpared", "bhmwk", "bses", "bsex",
    "bwhite", "sigma", "dic"),
  data=dataList,
  n.chains=4,
  inits=initsList,
  sample=100000,
  thin=10)
```

See Rmarkdown for graphs and convergence diagnostics

I Thinning

	Lower95	Median	Upper95	Mean	SD	Mode
b0	54.666	68.547	82.022	68.523	6.9802	—
bpared	-5.3965	-2.9301	-0.45199	-2.921	1.2568	—
bhmwk	0.31793	1.0774	1.8239	1.0758	0.385	—
bses	-1.7317	3.5355	8.6108	3.5192	2.641	—
bsex	-1.0702	1.5872	4.2049	1.5916	1.34	—
bwhite	-5.1564	-0.48398	4.1666	-0.48483	2.3717	—
sigma	4.3981	5.303	6.3228	5.3414	0.49807	—

I Thinning

	MCerr	MC%ofSD	SSeff	AC.100	psrf
b0	0.1536	2.2	2065	0.3578	1.0005
bpared	0.02495	2	2537	0.29614	1.0002
bhmwk	0.0023659	0.6	26480	0.001516	1.0002
bses	0.046777	1.8	3188	0.2184	1.0002
bsex	0.0093853	0.7	20386	-0.013683	1.0001
bwhite	0.042069	1.8	3178	0.18788	1.0003
sigma	0.0027615	0.6	32529	0.010286	1.0001

Model fit assessment:

DIC = 420.286

PED not available from the stored object

Estimated effective number of parameters: $pD = 7.26381$

Total time taken: 22.3 seconds

Better?

I autorun.jags

Give this a try (it's in Rmarkdown):

```
mod1b.runjags ← autorun.jags(model=model1,
  monitor=c("b0", "bpared", "bhmwk", "bses",
    "bsex", "bwhite", "sigma", "dic"),
  data=dataList,
  n.chains=4,
  inits=initsList,
  thin=10)
plot(mod1b.runjags)
print(mod1b.runjags)
```

I Drop gender, ses and white

Remove the corresponding b's from code.

JAGS model summary statistics from 40000 samples (chains = 4; adapt+burnin = 5000):

	Lower95	Median	Upper95	Mean	SD	Mode
b0	59.249	66.417	73.688	66.466	3.679	—
bpared	-2.855	-1.476	-0.14254	-1.482	0.6905	—
bhmwk	0.35491	1.1001	1.8656	1.0985	0.38373	—
sigma	4.4145	5.2973	6.2767	5.3305	0.4823	—

I Drop gender, ses and white

	MCerr	MC%ofSD	SSeff	AC.10	psrf	
b0		0.14282	3.9	664	0.71231	1.0022
bpared		0.025421	3.7	738	0.69378	1.0018
bhmwk		0.0055812	1.5	4727	0.080932	1.0001
sigma		0.0035156	0.7	18820	0.01696	1.0001

Model fit assessment:

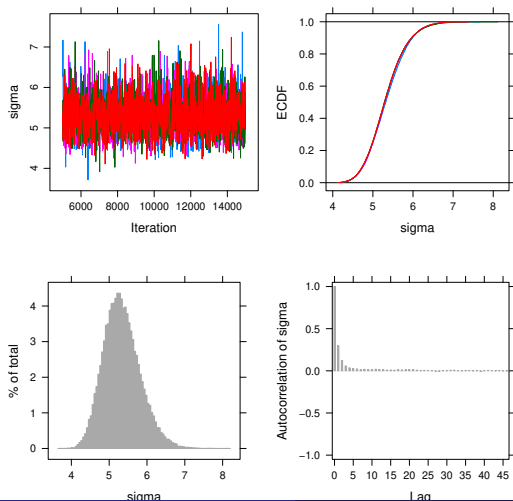
DIC = 417.0302

PED not available from the stored object

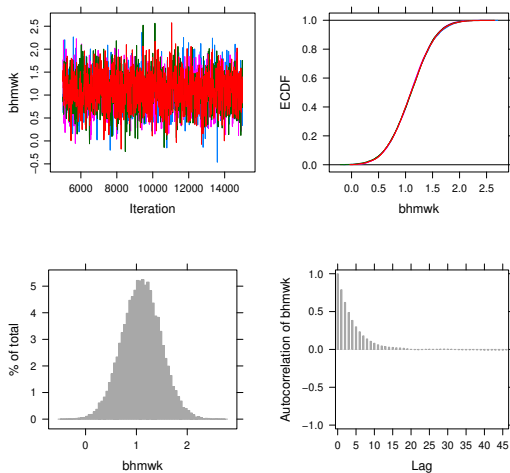
Estimated effective number of parameters: $pD = 4.11395$

Total time taken: 4.2 seconds

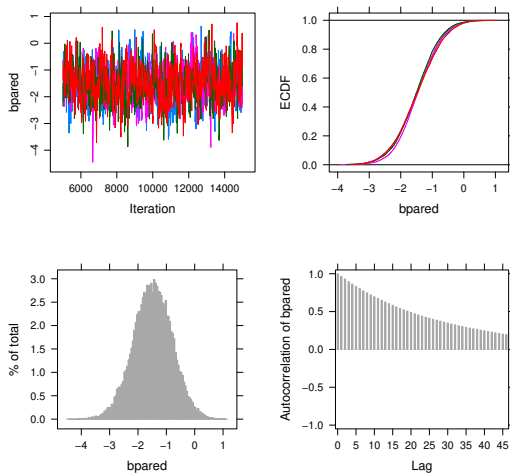
I Figure: sigma



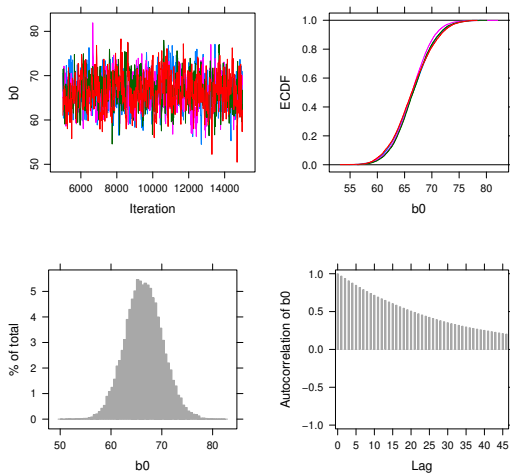
I Figure: parent education



I Figure: homework



I Figure: intercept



I Thin Again: thin=5

JAGS model summary statistics from 40000 samples (thin = 5; chains = 4; adapt+burnin = 5000):

Graphs look good (in a few pages).

	Lower95	Median	Upper95	Mean	SD	Mode
b0	59.236	66.514	73.633	66.516	3.6742	—
bpared	-2.8677	-1.4972	-0.14535	-1.4989	0.69509	—
bhmwk	0.36269	1.1062	1.8661	1.1075	0.38321	—
sigma	4.4489	5.2984	6.3126	5.3351	0.4832	—

b0	0.062372	1.7	3470	0.16743	1.0017
bpared	0.011504	1.7	3651	0.16326	1.0016
bhmwk	0.0026408	0.7	21057	-0.0081007	1.0002
sigma	0.0024917	0.5	37605	0.0030993	1

Model fit assessment:

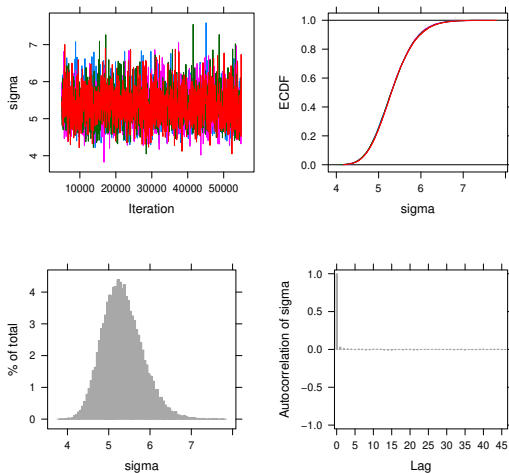
DIC = 417.0318

PED not available from the stored object

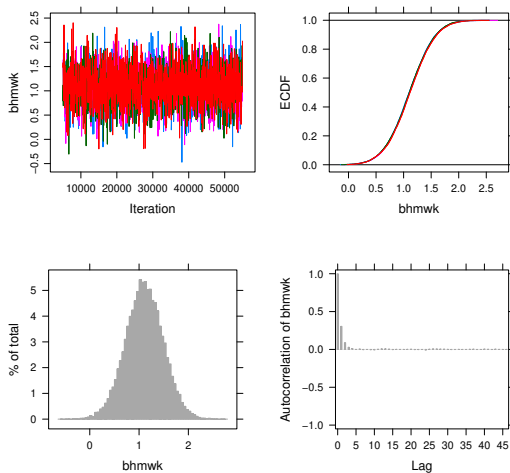
Estimated effective number of parameters: $pD = 4.11197$

Total time taken: 9.9 seconds

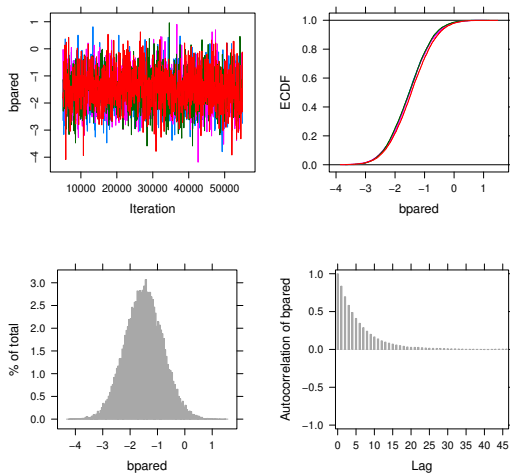
I Figure: sigma



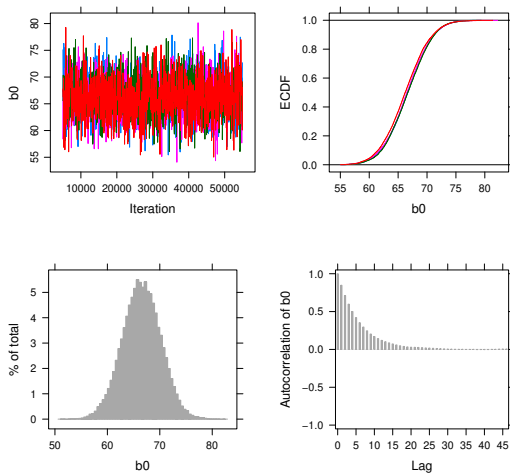
I Figure: parent education



I Figure: homework



I Figure: intercept



I Add Interaction

Adding an interaction is just like adding another variable. I centered the variables to deal with multicollinearity so our model is now

Use these center values in dataList, model, list of initial values, and in monitor.

```
nels$homework ← nels
```

\$

```
homework - mean(nels$homework)
```

```
nels$cparedu ← nels$paredu - mean(nels$paredu )
```


The model appears to converge fine.

JAGS model summary statistics from 40000 samples (thin = 5; chains = 4; adapt+burnin = 5000):

	Lower95	Median	Upper95	Mean	SD	Mode
b0	61.547	62.825	64.088	62.825	0.64877	–
bpared	-2.906	-1.4911	-0.13913	-1.4919	0.70276	–
bhmwk	0.44105	1.194	1.9534	1.1955	0.38454	–
bint	-0.12368	0.5566	1.2505	0.55805	0.34913	–
sigma	4.3446	5.2292	6.209	5.2656	0.48384	–

I Results

	MCerr	MC%ofSD	SSeff	AC.50	psrf
b0	0.0032438	0.5	40000	0.003358	1.0001
bpared	0.0035273	0.5	39696	0.004679	1
bhmwk	0.0019555	0.5	38668	-0.013487	1
bint	0.0017537	0.5	39636	0.0031527	0.99997
sigma	0.0024083	0.5	40362	0.0030145	0.99998

Model fit assessment:

DIC = 416.4373

PED not available from the stored object

Estimated effective number of parameters: $pD = 5.21494$

Total time taken: 11.1 seconds

I Model Evaluation

There are many things that you can do here using the data and posterior distribution.

I will present 2 methods of getting samples from the posterior.

- Add code to your model statement so that you sample from the posterior; that is, within the loop for the likelihood add, for example

`emp.new[i] ~ dnorm(mu[i],precision)`

and add `emp.new` to list of parameters to monitor (output).

- Use posterior parameters and draw from posterior.

See Rmarkdown for first method and next pages for the other.

I Monte Carlo of Posterior

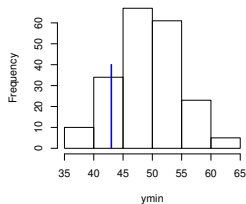
Use Monte Carlo to get posterior predictive distribution: $S = 200$ replications of “data” using draws from the posterior distribution of parameters.

Note: The posterior parameters are a bit different, because I used a previous run when I worked up this example. The results should be about the same.

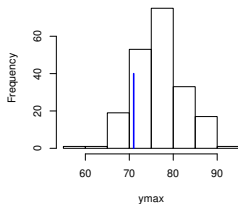
```
n ← length(nels2$math)
replications ← 200
yrep ← matrix(99,nrow=n,ncol=replications)
for (s in 1:replications){
  b0 ← rnorm(1,66.586,sd=3.7576)
  bpared ← rnorm(1,-1.517,sd=0.70876)
  bhmwk ← rnorm(1,1.1041,sd=0.38207)
  for (i in 1:n){
    yrep[i,s] = b0 + bpared*nels$paredu[i]
      + bhmwk*nels$homework[i] + rnorm(1,0,5.3372)
  }
}
```


I Statistics on Distribution

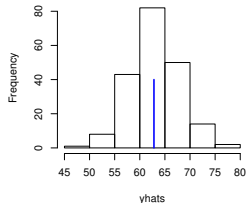
Simulated N=200 Minimums
Bayesian P-value = 0.86



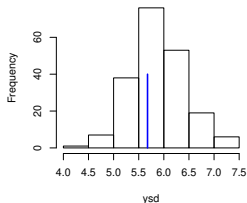
Simulated N=200 Maximums
Bayesian P-value = 0.85



Simulated N=200 Means
Bayesian P-value = 0.5

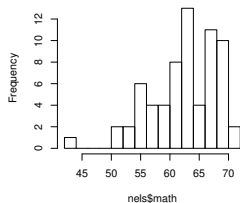


Simulated N=200 SDs
Bayesian P-value = 0.66

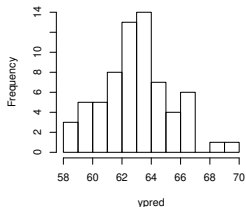


I Data and Posterior Pred Distribution

Data Distribution



Predicted Posterior Distribution



I Robust Multiple Linear Regression

Often we don't fit the tails of distribution very well when we use the normal distribution. An alternative is to use [Students-t distribution](#) for the data model (i.e., the likelihood).

Maybe this will further improve our model

We will need to get posterior distribution for ν , the degrees of freedom. This leads to the following model:

I JAGS: model t-distribution

```

tmr = ‘‘model { for (i in 1:N){
    y[i] ~ dt(mu[i] , precision, nu)
    mu[i] ← b0 + bpared*pared[i] + bhmwk*hmwk[i]
}
b0 ~ dnorm(0 , 1/(100*sdY^2) )
bpared ~ dnorm(0 , 1/(100*sdY^2) )
bhmwk ~ dnorm(0 , 1/(100*sdY^2) )
sigma ~ dunif( 1E-3, 1E+30 )
precision ← 1/sigma^2
nuMinusOne ~ dexp(1/29)
nu ← nuMinusOne+1
}
} ’’

```

I Results with t-distribution

JAGS model summary statistics from 40000 samples (chains = 4; adapt+burnin = 5000):

	Lower95	Median	Upper95	Mean	SD	Mode
b0	59.735	67.108	74.143	67.101	3.6502	—
bpared	-2.9685	-1.5622	-0.1805	-1.5658	0.70403	—
bhmwk	0.38532	1.1156	1.8194	1.1114	0.36607	—
sigma	3.5702	4.8388	6.0077	4.8334	0.61171	—
nu	1.564	16.637	75.416	25.083	24.539	—

I Results with t-distribution

	MCerr	MC%ofSD	SSeff	AC.10	psrf
b0	0.1767	4.8	427	0.80761	1.0052
bpared	0.033095	4.7	453	0.80163	1.0045
bhmwk	0.0061505	1.7	3542	0.16689	1.0022
sigma	0.0065167	1.1	8811	0.040988	1.0004
nu	0.34048	1.4	5194	0.082768	1.0009

Model fit assessment:

DIC = 416.5414

PED not available from the stored object

Estimated effective number of parameters: $pD = 4.88061$

Total time taken: 2.3 minutes

From plots, we see that b_1 and b_0 are not mixing well and have large auto-correlations—Lets fix this.

I Results with t-distribution with thin=5

JAGS model summary statistics from 40000 samples (thin = 5; chains = 4; adapt+burnin = 5000):

	Lower95	Median	Upper95	Mean	SD	Mode
b0	59.665	66.922	73.888	66.888	3.6247	—
bpared	-2.8874	-1.5278	-0.16677	-1.5283	0.69211	—
bhmwk	0.3915	1.1176	1.8391	1.1177	0.37009	—
sigma	3.6036	4.8466	6.0131	4.843	0.60732	—
nu	1.5402	16.895	77.435	25.816	26.091	—

I Results with t-distribution with thin=5

JAGS model summary statistics from 40000 samples (thin = 5; chains = 4; adapt+burnin = 5000):

	MCerr	MC%ofSD	SSeff	AC.50	psrf
b0	0.076658	2.1	2236	0.3222	1.0008
bpared	0.014479	2.1	2285	0.31784	1.001
bhmwk	0.0028562	0.8	16790	0.008508	1
sigma	0.0036165	0.6	28201	-0.0035392	1.0001
nu	0.18429	0.7	20044	-0.013753	1.0002

Model fit assessment:

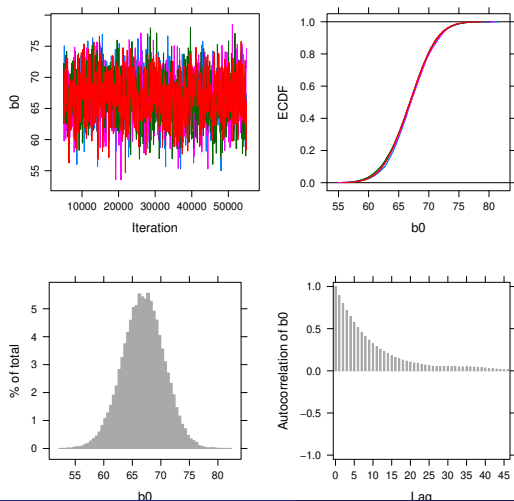
DIC = 416.543

PED not available from the stored object

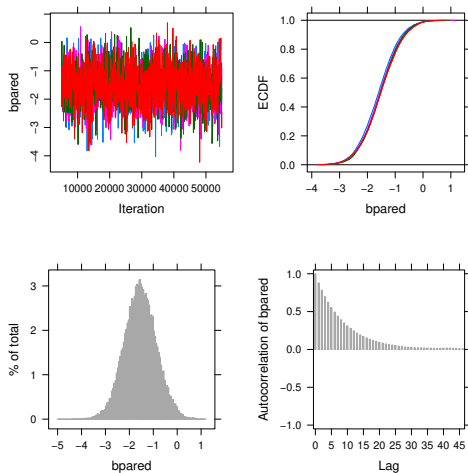
Estimated effective number of parameters: pD = 4.87464

Total time taken: 5.1 minutes

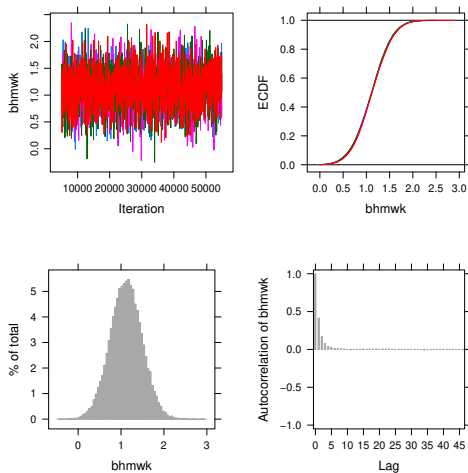
I Figure: b0



I Figure: bpared



I Figure: bhmwk



I Figure: sigma

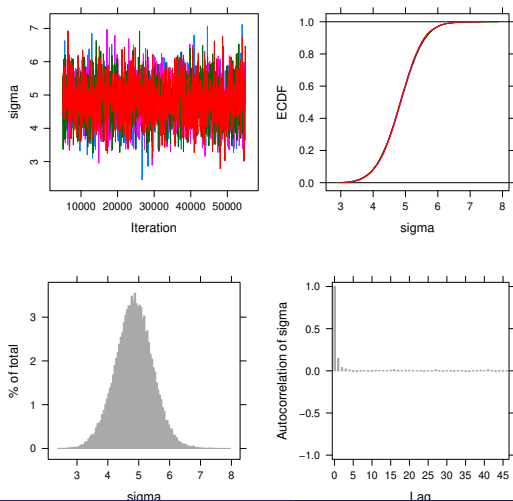
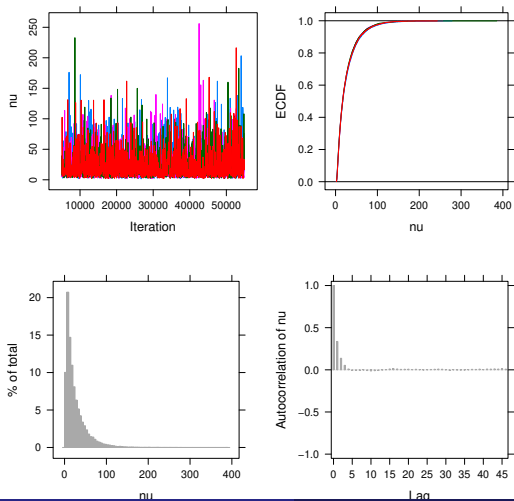
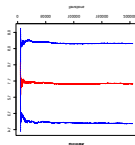


Figure: nu



I Gelman Plots



I Figure: Examine posterior statistics

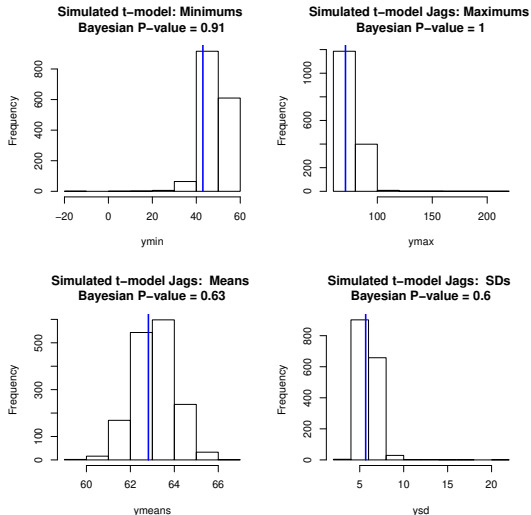
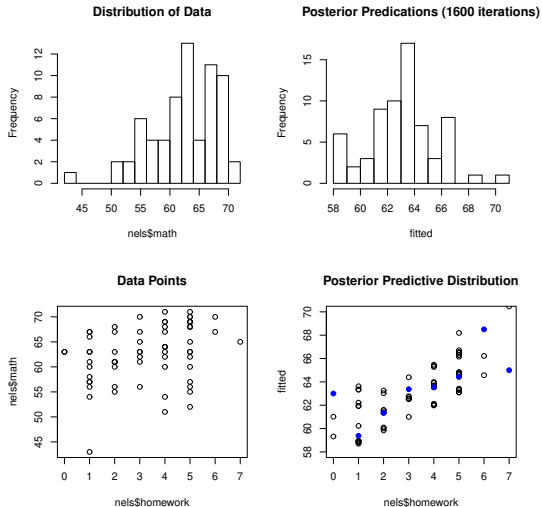


Figure: Examine posterior distribution



I Summary Comments on the NELS

- In notes here I report results from raw scores.
- In the code online after doing interactions I switched back to un-centered.
- Thinning seemed to be needed to get good mixing and low auto-correlations.
- Model Evaluations:
 - Model parameter estimates seemed reasonable.
 - The normal distribution is about the same as the t-distribution; however, the t -produced more outlying statistics in the posterior predictive distribution.
- Improvements would not allow predicted value to be higher than the maximum on the test (i.e., deal with ceiling). Possibilities include using a different likelihood:
 - Truncated or censored distribution.
 - Beta distribution.

I Model Comparison

From: Richare E. Turner “Why Gelman “hates” Bayesian model comparison” at <http://www.gatsby.ucl.ac.uk/~turner/TeaTalks/bayes-model-comp/bayes-model-comp.pdf>

Conclusions

- Discrete Bayesian model comparison:
 - beware the prior
 - Uninformative priors dangerous (improper priors apocalyptic)
 - Perform a sensitivity analysis
 - Common tactic: convert model comparison into parameter estimation problem
- Philosophical inconsistency - model comparison is just (discrete) inference
- Posterior predictive tests: can tell you in what way your model is wrong without needing another to compare to another model
- Original references: Kass & Greenhouse 1989, Statistical Science; Kass 1993, Journal of the Royal Statistical Society; Kass & Raftery 1995, Journal of the American Statistical Society.
- Suggestion read both Gelman's book and MacKay's book (*Information theory, inference and learning algorithms*)

I If you are compelled to compare Models

We have 2 models M_1 and M_2 and data $\mathbf{y}$.

$$p(\theta|\mathbf{y}, M_k) = \frac{p(\mathbf{y}|\theta, M_k)p(\theta|M_k)}{p(\mathbf{y}|M_k)} \leftarrow \text{Bayesian evidence (model likelihood)}$$

From Bayes Theorem:

$$p(M_k|\mathbf{y}) = \frac{p(\mathbf{y}|M_k)p(M_k)}{p(\mathbf{y})}$$

Compute posterior odds:

$$\begin{aligned} \frac{p(M_1|\mathbf{y})}{p(M_2|\mathbf{y})} &= \frac{p(\mathbf{y}|M_1)}{p(\mathbf{y}|M_2)} \times \frac{p(M_1)}{p(M_2)} \\ &= \text{Bayes factor} \times \text{Prior Odds} \end{aligned}$$

$$\text{Bayes factor} = \frac{p(\mathbf{y}|M_1)}{p(\mathbf{y}|M_2)}$$

I Bayes Factor

$$\text{Bayes factor} = BF = \frac{p(\mathbf{y}|M_1)}{p(\mathbf{y}|M_2)}$$

- Marginalized (collapsed) over parameters.
- Shows how much the prior odds change given data.
- Making a decision:
 - If $BF > 3.0$, then substantial evidence for model 1 (M_1).
 - If $BF < 1/3$, then substantial evidence for model 2 (M_2).
- BF takes into account quality of model fit to data and model complexity.
- BF favors highly predictive model and penalizes for too many unnecessary or unimportant parameters.
- Sometimes $\ln(BF)$ is reported.
- Use DIC and model parameter estimation.

I Simple Method

Use the BayesFactor package in R compares all possible with model with only an intercept.

<http://bayesfactorpcl.r-forge.r-project.org/>

```
nels$xwhite ← as.numeric(nels$white)
bf ← regressionBF(math ~ cparedu + chomework + ses
                    + xwhite + sex, data=nels)
bf
```

Also, the best, say 5,

```
head(bf,n=5)
```

Note: Online code I used un-centered...you get the same results.

I Best 5

Bayes factor analysis

cparedu + chomework	:	17.83269	$\pm 0\%$
cparedu + chomework + ses	:	11.84489	$\pm 0\%$
cparedu + chomework + sex	:	8.491123	$\pm 0\%$
cparedu + chomework + ses + sex	:	7.718168	$\pm 0\%$
chomework	:	6.983524	$\pm 0\%$

Against denominator:

Intercept only

Bayes factor type: BFlinearModel, JZS

I Alternative Comparisons

`top_compare` $\leftarrow$ `head(bf)/max(bf)`

Bayes factor analysis

[1]	cparedu + chomework	:	1	$\pm 0\%$
	cparedu + chomework + ses	:	0.6642233	$\pm 0\%$
	cparedu + chomework + sex	:	0.4761549	$\pm 0\%$
	cparedu + chomework + ses + sex	:	0.4328101	$\pm 0\%$
	chomework	:	0.3916136	$\pm 0\%$
	cparedu + chomework + xwhite	:	0.3474389	

Against denominator:

$\text{math} \sim \text{cparedu} + \text{chomework}$

I Comparisons Using DIC

The smaller the better:

Model	DIC
paredu + homework + ses + sex	420.1328
paredu + homework	417.0576
paredu + homework + paredu*homework	416.4373
Using t-distribution	416.4214

I Next Steps

An alternative to model “comparison” (model selection and assessing important variables) is Bayesian Model Averaging (see BMA package).

Let’s try to figure out why more expenditures lead to lower State average total SAT scores.