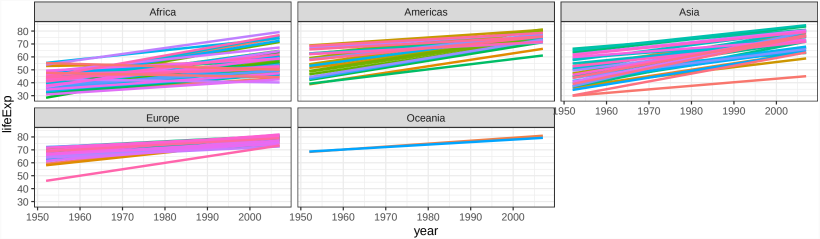
Multilevel models, part 2

Frank Edwards

4/24/2020

Interactions let us set a slope for each unit over time (a fixed effect growth model)

```r
library(gapminder)
ggplot(gapminder, aes(x = year, y = lifeExp, color = country)) +
  geom_smooth(method = "lm", se=F) +
  guides(color = F) +
  facet_wrap(~continent)
```



Partially pooling slopes

- An interaction model presumes that all countries come from different populations
- No information about the population of countries enters the model
- The unit-level interaction approach (fixed effects) constrains our modeling choices (time-invariant parameters can't be included)

Partially pooling slopes

- An interaction model presumes that all countries come from different populations
- No information about the population of countries enters the model
- The unit-level interaction approach (fixed effects) constrains our modeling choices (time-invariant parameters can't be included)

But, we can use the partial pooling technique to estimate a probability distribution for both intercepts *and* for slopes. We'll regularize the estimates, have parameters for the population we can use for inference, and have much more flexibility in our models.

Coffee shop wait times

- Coffee shops have different wait times in the morning and afternoon
- Coffee shops with longer wait times in the morning will have a greater difference between morning and afternoon wait times (a floor effect)
- We want to pool information about coffee shops, presumably they are all similar in important ways
- Now, we'll pool information on both their intercepts (morning wait time), and their slopes (difference between morning and afternoon wait time)

Simulate coffee shop data

```r
a<- 3.5
b<- -1
sigma_a<-1
sigma_b<-0.5
rho<- -0.7
```

Build a covariance matrix

These are symmetric matrices with variances on the diagonal, and covariances off-diagonal

So if our linear model is $y_i = \alpha_i + \beta_i x_i$ where α is a random slope and β is a random intercept, the covariance matrix can be written as

$$\left(\begin{array}{cc} \sigma_\alpha^2 & \sigma_\alpha \sigma_\beta \rho_{\alpha\beta} \\ \sigma_\alpha \sigma_\beta \rho_{\alpha\beta} & \sigma_\beta^2 \end{array} \right)$$

Compare to a correlation matrix

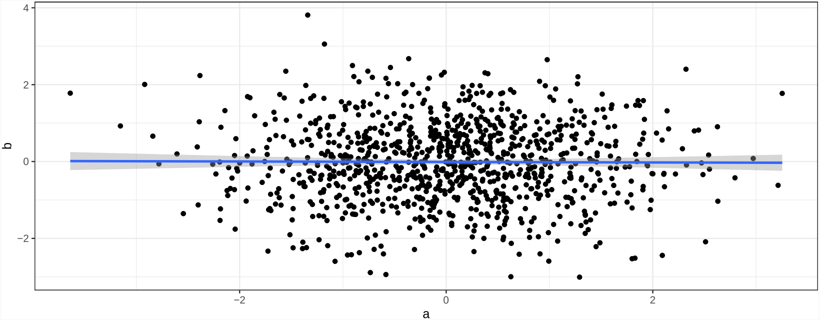
A correlation matrix represents correlations between random variables (covariances scaled to [-1, 1]).

$$\begin{pmatrix} 1 & \rho_{\alpha\beta} \\ \rho_{\alpha\beta} & 1 \end{pmatrix}$$

The impact of covariance on two random variables

Let's simulate a couple of examples: $\rho = 0, \sigma_a = 1, \sigma_b = 1$

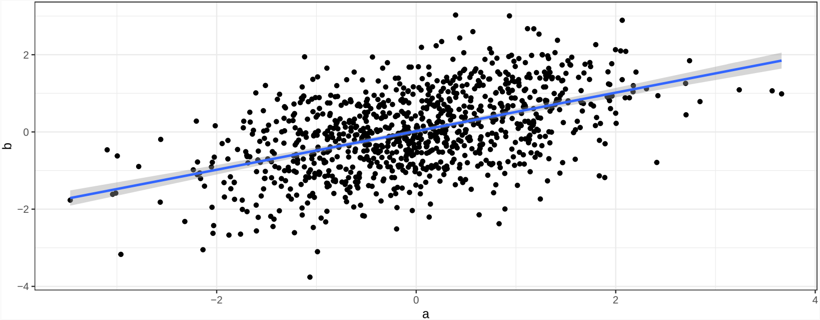
```
S<-matrix(c(1, 0, 0, 1), nrow=2)
mu<-c(0, 0)
sims<-mvrnorm(1000, mu, S)
plot_dat<-data.frame(a = sims[,1], b = sims[,2])
ggplot(plot_dat, aes(x = a, y = b)) +
  geom_point() +
  geom_smooth(method = "lm")
```



The impact of covariance on two random variables

Let’s simulate a couple of examples: $\rho = 0.5, \sigma_a = 1, \sigma_b = 1$

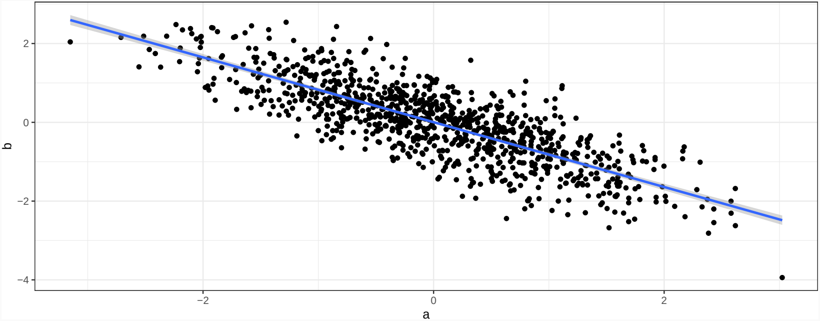
```
S<-matrix(c(1, 0.5, 0.5, 1), nrow=2)
mu<-c(0, 0)
sims<-mvrnorm(1000, mu, S)
plot_dat<-data.frame(a = sims[,1], b = sims[,2])
ggplot(plot_dat, aes(x = a, y = b)) +
  geom_point() +
  geom_smooth(method = ”lm”)
```



The impact of covariance on two random variables

Let's simulate a couple of examples: $\rho = -0.8, \sigma_a = 1, \sigma_b = 1$

```r
S<-matrix(c(1, -0.8, -0.8, 1), nrow=2)
mu<-c(0, 0)
sims<-mvrnorm(1000, mu, S)
plot_dat<-data.frame(a = sims[,1], b = sims[,2])
ggplot(plot_dat, aes(x = a, y = b)) +
  geom_point() +
  geom_smooth(method = "lm")
```



A covariance matrix can be obtained from standard deviations and correlations

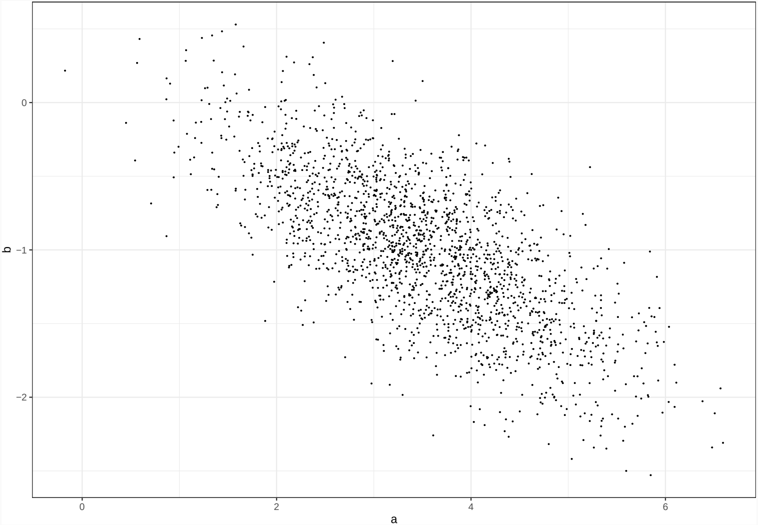
```
sigmas<-c(sigma_a, sigma_b)
Rho<-matrix(c(1, rho, rho, 1), nrow=2)
# matrix multiplication produces \Sigma
Sigma<-diag(sigmas) %*% Rho %*% diag(sigmas)
Sigma
```

```
##       [,1]  [,2]
## [1,]  1.00 -0.35
## [2,] -0.35  0.25
```

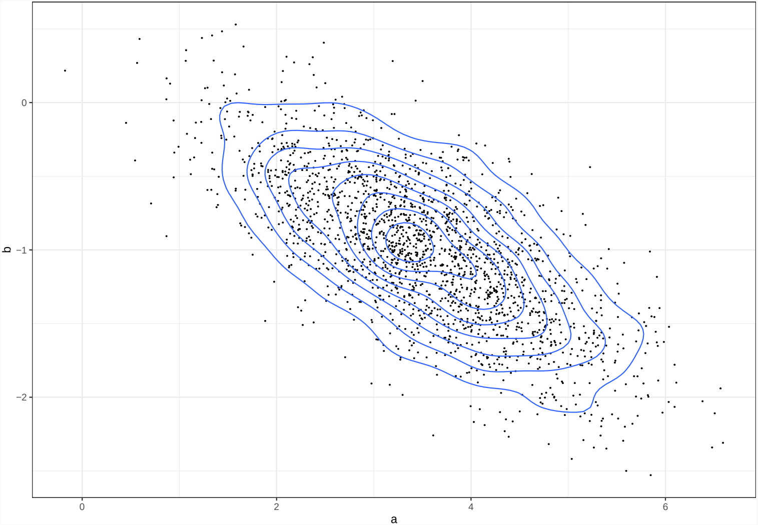
Simulate the data

```r
Mu<-c(a,b)
N_cafes<-2000
library(MASS)
vary_effects<-mvrnorm(N_cafes, Mu, Sigma)
sim_dat<-data.frame(a = vary_effects[,1],
                    b = vary_effects[,2])
```

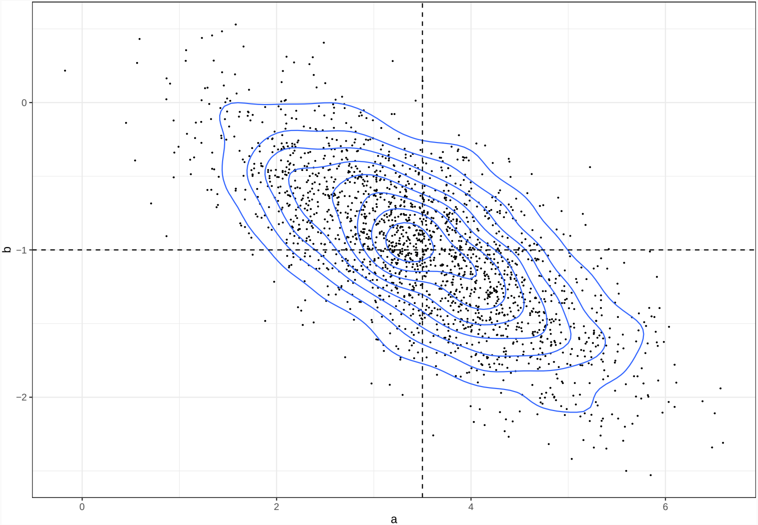
Plot the distribution



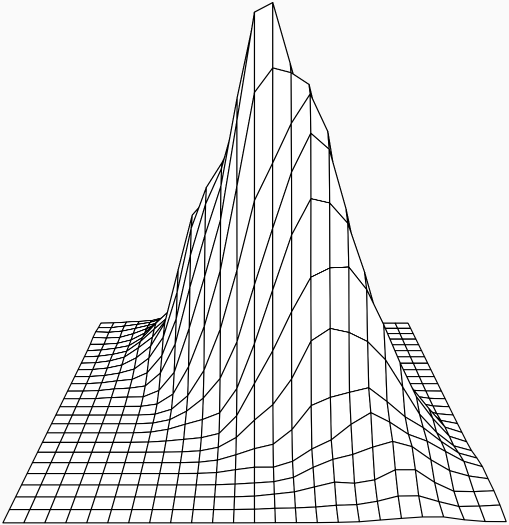
Plot the distribution



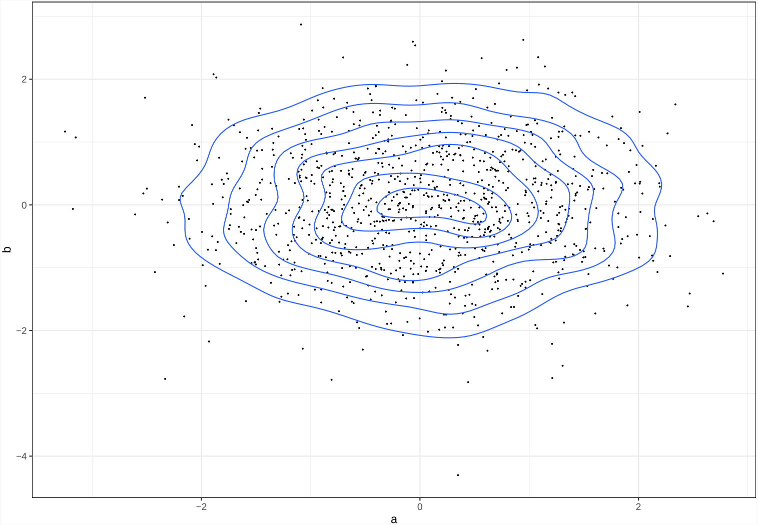
Plot the distribution



Seeing the contours in 3d



For comparison: no correlation between a and b



The setup

Our simulated data is the truth for 20 cafes. Now let's say we visit each of them 10 times to try to infer the truth.

```r
N_visits<-20
afternoon<-rep(0:1, N_visits * N_cafes/2)
cafe_id<-rep(1:N_cafes, each = N_visits)
mu<-a_cafe[cafe_id] + b_cafe[cafe_id] * afternoon
sigma<-0.5
wait<-rnorm(N_visits * N_cafes, mu, sigma)
d<-data.frame(cafe = cafe_id,
              afternoon = afternoon,
              wait = wait)
glimpse(d)
```

```
## Observations: 400
## Variables: 3
## $ cafe      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ afternoon <int> 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, ...
## $ wait      <dbl> 5.2069230, 2.5616650, 4.7259680, 2.2468053, 4.9995825, 3....
```

Defining the varying slopes model

Wait times (W) follow a normal likelihood, but each cafe can have it's own average wait time (intercept) and it's own average difference between morning and afternoon (A) waits (slope).

$$W_i \sim \mathrm{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha_{\mathrm{cafe}_i} + \beta_{\mathrm{cafe}_i} A_i$$

$$\begin{bmatrix} \alpha_{\mathrm{cafe}} \\ \beta_{\mathrm{cafe}} \end{bmatrix} \sim \mathrm{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathrm{S} \right)$$

The covariance matrix for an MVNormal variable

We can write a covariance matrix here as

$$\mathrm{S} = \begin{pmatrix} \sigma_\alpha^2 & \sigma_{\alpha\beta} \\ \sigma_{\alpha\beta} & \sigma_\beta^2 \end{pmatrix}$$

Or write it as the product of the standard deviations (twice to square) and the correlation matrix R

$$\mathrm{S} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathrm{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix}$$

The priors and hyper-priors for the varying intercepts and slopes

$$\alpha \sim \text{Normal}(5, 2)$$

$$\beta \sim \text{Normal}(-1, 0.5)$$

$$\sigma \sim \text{Exponential}(1)$$

$$\sigma_\alpha \sim \text{Exponential}(1)$$

$$\sigma_\beta \sim \text{Exponential}(1)$$

$$R \sim \text{LKJcorr}(2)$$

A prior for a correlation matrix

A correlation matrix for two variables takes the form:

$$\mathrm{R} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

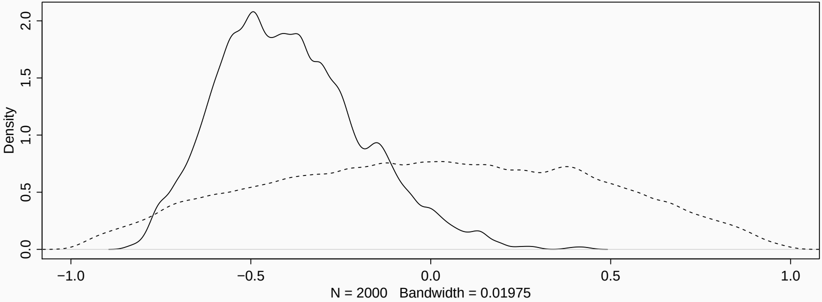
The LKJ correlation distribution samples correlation matrices where ρ can take on values on [-1, 1]. Higher values for the shape parameter makes the prior skeptical of extreme values (near -1 or 1).

Fitting the model with ulam()

```r
m_cafe<-ulam(alist(
  wait ~ dnorm(mu, sigma),
  mu<-a_cafe[cafe] + b_cafe[cafe] * afternoon,
  c(a_cafe, b_cafe)[cafe] ~ multi_normal(c(a,b), Rho, sigma_cafe),
  a ~ dnorm(5, 2),
  b ~ dnorm(-1, 0.5),
  sigma_cafe ~ exponential(1),
  sigma ~ exponential(1),
  Rho ~ lkj_corr(2)
), data = d, chains = 4, cores = 4)
```

What the model tells us about ρ, prior and posterior

```
post <- extract.samples(m_cafe)
dens( post$Rho[,1,2] , xlim=c(-1,1) ) # posterior
R <- rlkjcorr( 1e4 , K=2 , eta=2 ) # prior
dens( R[,1,2] , add=TRUE , lty=2 )
```



Our adaptive prior

Shrinks each posterior estimate toward group averages through pooling information

Our adaptive prior

Shrinks each posterior estimate toward group averages through pooling information

- Pools information on the correlation of slopes and intercepts (ρ) by learning from the data

Our adaptive prior

Shrinks each posterior estimate toward group averages through pooling information

- Pools information on the correlation of slopes and intercepts (ρ) by learning from the data
- Pools information on the distribution of average wait times to estimate individual wait times (α)

Our adaptive prior

Shrinks each posterior estimate toward group averages through pooling information

- Pools information on the correlation of slopes and intercepts (ρ) by learning from the data
- Pools information on the distribution of average wait times to estimate individual wait times (α)
- Pools information on the difference between morning and afternoon wait times to estimate individual differences (β)

Our adaptive prior

Shrinks each posterior estimate toward group averages through pooling information

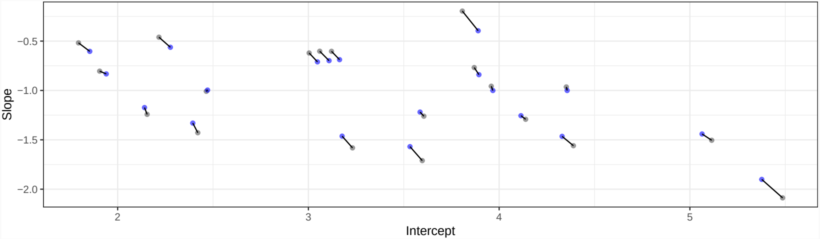
- Pools information on the correlation of slopes and intercepts (ρ) by learning from the data
- Pools information on the distribution of average wait times to estimate individual wait times (α)
- Pools information on the difference between morning and afternoon wait times to estimate individual differences (β)

Visualizing adaptive regularization: set up plot

```r
## compute observed a and b for each cafe
observed<-d %>%
  group_by(cafe, afternoon) %>%
  summarise(a = mean(wait)) %>%
  pivot_wider(id_cols = cafe,
              values_from = a,
              names_from = afternoon,
              names_prefix = ”a”)
observed<-observed%>%
  mutate(b = a1 - a0) %>%
  ungroup()
## attach posterior means for a and b for each cafe
post<-extract.samples(m_cafe)
post_a<-apply(post$a_cafe, 2, mean)
post_b<-apply(post$b_cafe, 2, mean)
plot_dat<-observed %>%
  mutate(post_a = post_a, post_b = post_b)
```

The observed and posterior mean for α and β for each cafe

```
ggplot(plot_dat,
       aes(x = a0, y = b)) +
  geom_point(alpha = 0.4) +
  geom_point(aes(x = post_a, y = post_b),
             alpha = 0.6, color = ”blue”) +
  geom_segment(aes(xend = post_a, yend=post_b)) +
  labs(x = ”Intercept”, y = ”Slope”)
```



Let’s return to the life expectancy data

```
library(gapminder)
dat<-gapminder %>%
  mutate(L_c = (lifeExp - mean(lifeExp)) / sd(lifeExp),
         country = factor(country),
         year_c = (year - min(year))/5) %>%
  select(lifeExp, L_c, country, year, year_c, continent)
```

Multilevel life expectancy?

- Countries differ pretty dramatically in average life expectancy

Multilevel life expectancy?

- Countries differ pretty dramatically in average life expectancy
- All countries saw life expectancy go up over time

Multilevel life expectancy?

- Countries differ pretty dramatically in average life expectancy
- All countries saw life expectancy go up over time
- But rates of increase differ across places

Multilevel life expectancy?

- Countries differ pretty dramatically in average life expectancy
- All countries saw life expectancy go up over time
- But rates of increase differ across places
- Countries that started with high life expectancy probably saw low growth, as there’s a natural limit on how high it can go

Multilevel life expectancy?

- Countries differ pretty dramatically in average life expectancy
- All countries saw life expectancy go up over time
- But rates of increase differ across places
- Countries that started with high life expectancy probably saw low growth, as there’s a natural limit on how high it can go
- Countries with low life expectancy in the 1950s had much higher potential gains

Proposing a model

$$L_i \sim \mathrm{Normal}(\mu, \sigma)$$

$$\mu_i = \alpha_{[\mathrm{country}]_i} + \beta_{[\mathrm{country}]_i} \times year_i$$

$$\begin{bmatrix} \alpha_{\mathrm{country}} \\ \beta_{\mathrm{country}} \end{bmatrix} \sim \mathrm{MVNormal}\left(\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \mathrm{S}\right)$$

$$\mathrm{S} = \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathrm{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix}$$

$$\alpha \sim \mathrm{Normal}(0, 2)$$

$$\beta \sim \mathrm{Normal}(0, 2)$$

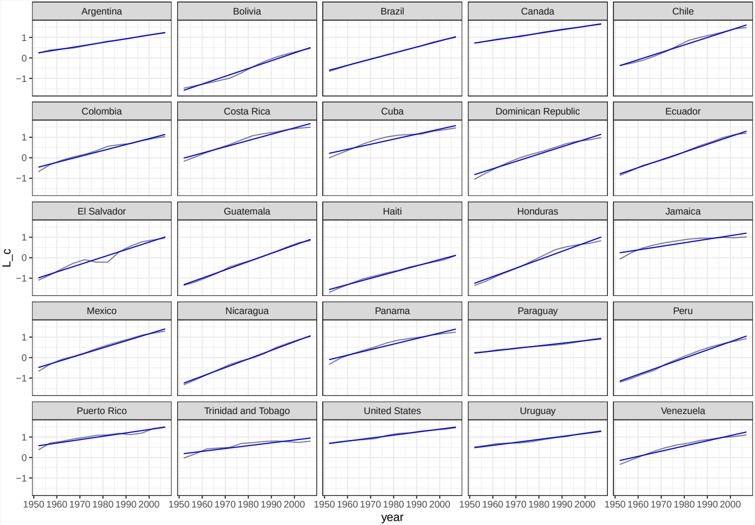
$$\sigma, \sigma_\alpha, \sigma_\beta \sim \mathrm{Exponential}(1)$$

$$\mathrm{R} \sim \mathrm{LKJcorr}(2)$$

Estimating the model

```
m_gm1<-ulam(alist(
  L_c ~ dnorm(mu, sigma),
  mu <- alpha[country] + beta[country] * year_c,
  c(alpha, beta)[country] ~ multi_normal(c(0, b), Rho, sigma_countr
  b ~ dnorm(0, 2),
  sigma_country ~ dexp(1),
  sigma ~ dexp(1),
  Rho ~ lkj_corr(1)
), data = dat, chains = 4, cores = 4)
```

Visualizing the inferences

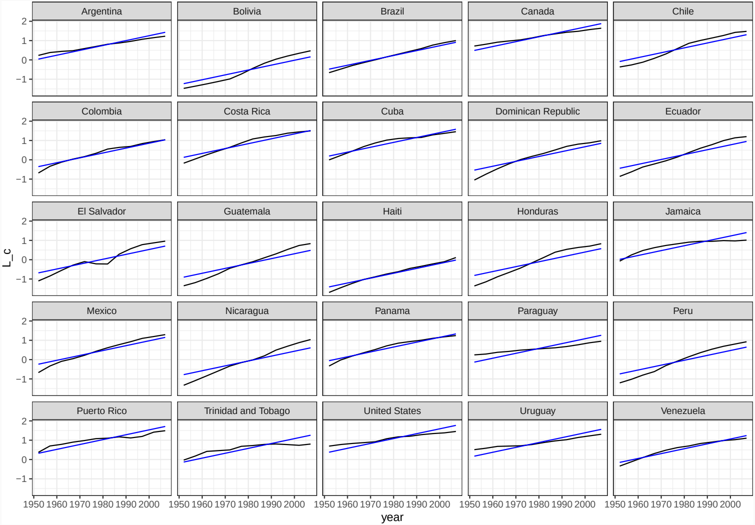


Compare to alternative models

Let’s compare this to a model with a single slope and variable intercepts

```r
m_gm2<-ulam(alist(
  L_c ~ dnorm(mu, sigma),
  mu <- alpha[country] + beta * year_c,
  alpha[country] ~ dnorm(a, sigma_country),
  a ~ dnorm(0,2),
  beta ~ dnorm(0, 2),
  sigma_country ~ dexp(1),
  sigma ~ dexp(1)
), data = dat, chains = 4, cores = 4)
```

Visualize the results



What did we learn about country parameters

```
precis(m_gm1, pars = ”sigma_country”, depth = 3)
```

```
##                        mean          sd      5.5%      94.5%    n_eff     Rhat4
## sigma_country[1] 1.18160832 0.070861593 1.0708565 1.29840568 4547.179 0.9985532
## sigma_country[2] 0.06481852 0.004645245 0.0578517 0.07257805 2791.211 0.9997337
```

Summary

- We can use partial pooling for slopes just as we did for intercepts
- This treats both the intercepts and slopes as coming from the same population
- This makes a lot of sense when our units are exchangeable, or come from the same population
- Make sure to read the chapter for some advanced examples, including applications for causal inference
- Homework: Ch14 easy and medium questions, H1 if you want a challenge