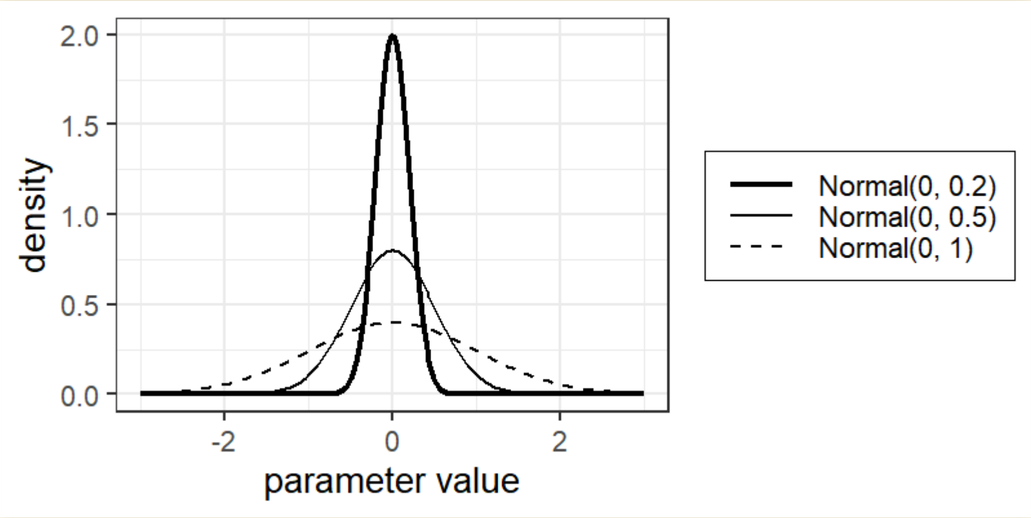
Interactions

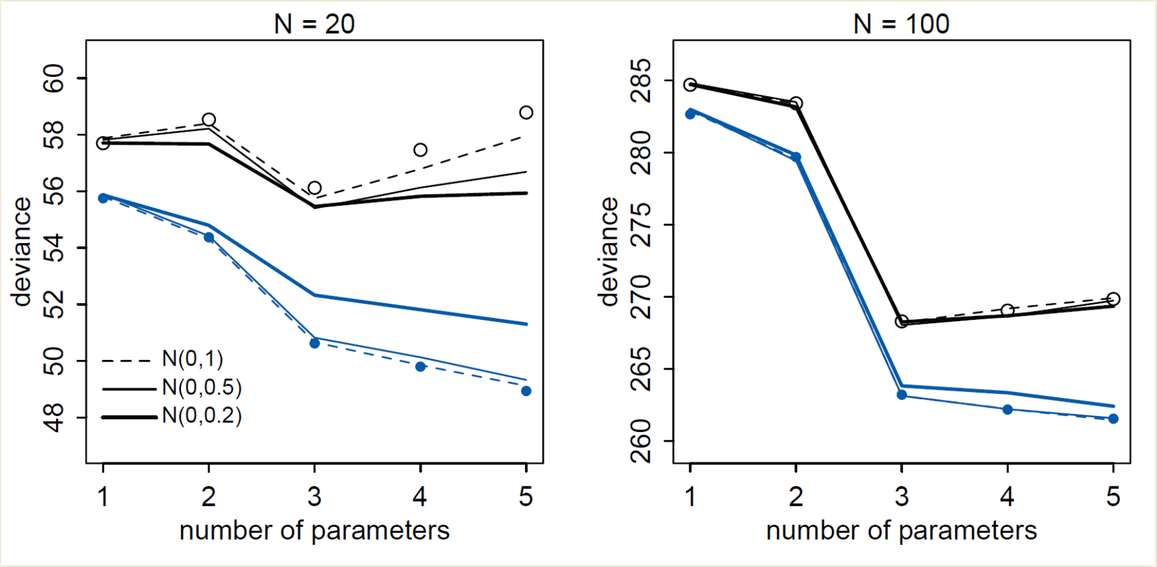
EES 4891-06/5891-01
Bayesian Statistical Methods
Jonathan Magnolia Gilligan
Class #10: Monday February 16, 2026

Regularization

Regularizing Priors

- Alternate approach: *regularizing priors*
  - Widely used in Machine Learning
  - Models that are worse at fitting *training data* can be better at predicting *test data*.
  - Regularizing priors tend to force unnecessary parameters to small (near-zero) values.
  - Regularizing normal prior for β params:





- Synthetic data generated from 3-parameter distribution.
  - *N* samples in training set, *N* in testing set.
  - blue = in-sample, black = out-of-sample
- With more training data (*N* = 100), regularizing priors keep out-of-sample deviance small, even with many parameters.
- You can use fancier regularizing priors than just normal distributions.

Predicting Predictive Accuracy

Cross-Validation

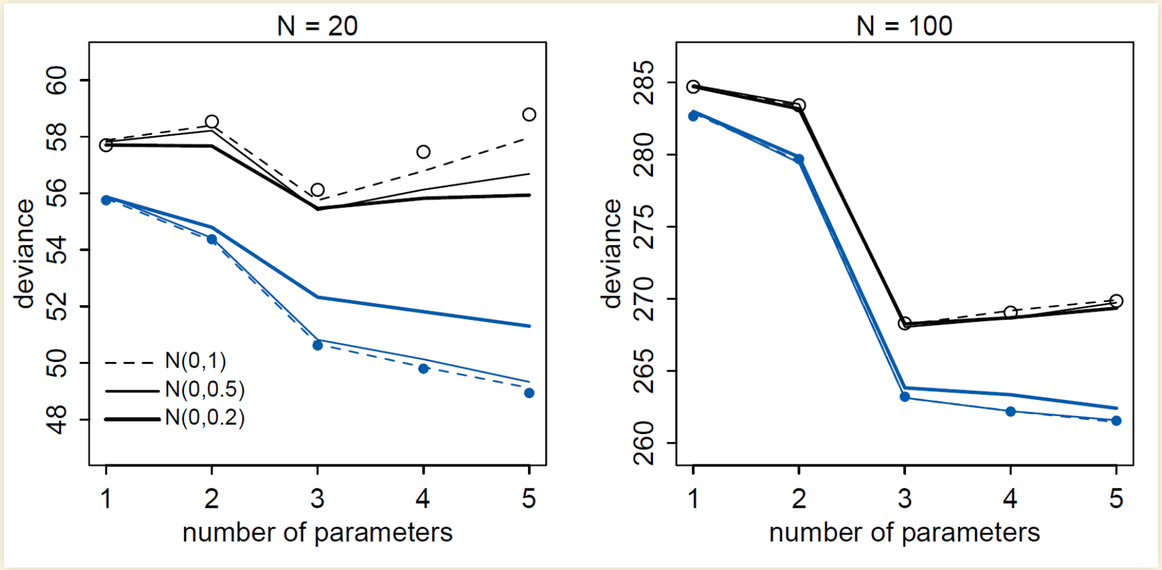
- How can we get a sense of how well our model will work with out-of-sample predictions?
- We started by splitting our data in half: *training* and *test* data.
- Sometimes it's not efficient to split our data in half.
- Can we do better?
- *k*-fold cross-validation:
  - Split data into k equal parts (example: $k = 5$)
    - For each part i (called a "fold"), fit the model to the other $k - 1$ parts and then predict part i.
    - Repeat this for all k parts.
    - Use all k folds to assess model performance
- Leave-one-out cross-validation (LOOCV):
  - An extreme form of *K*-fold cross-validation, where $k = N$, the size of the data.
  - For each data point, fit the model to all the others and then predict that one point.
- Problem: If you have *N* observations, then you have to fit your model *N* times. If *N* is large, this can be very slow.
- Pareto-Smoothed Importance Sampling (PSIS) is a fancy technique that lets us estimate LOOCV while we fit the model one time, without actually having to do real cross-validation.

Information Criteria

- As an alternative to cross-validation, use information theory to estimate the out-of-sample KL divergence.
- Examine the differences between in-sample and out-of-sample divergence in the figure
  - The difference is roughly twice the number of parameters.
    - In general, for relatively flat priors, the overfitting penalty is about twice the number of parameters.
  - Akaike Information Criterion (AIC)

$$\text{AIC} = D_{\text{train}} + 2p = -2\text{lppd} + 2p,$$

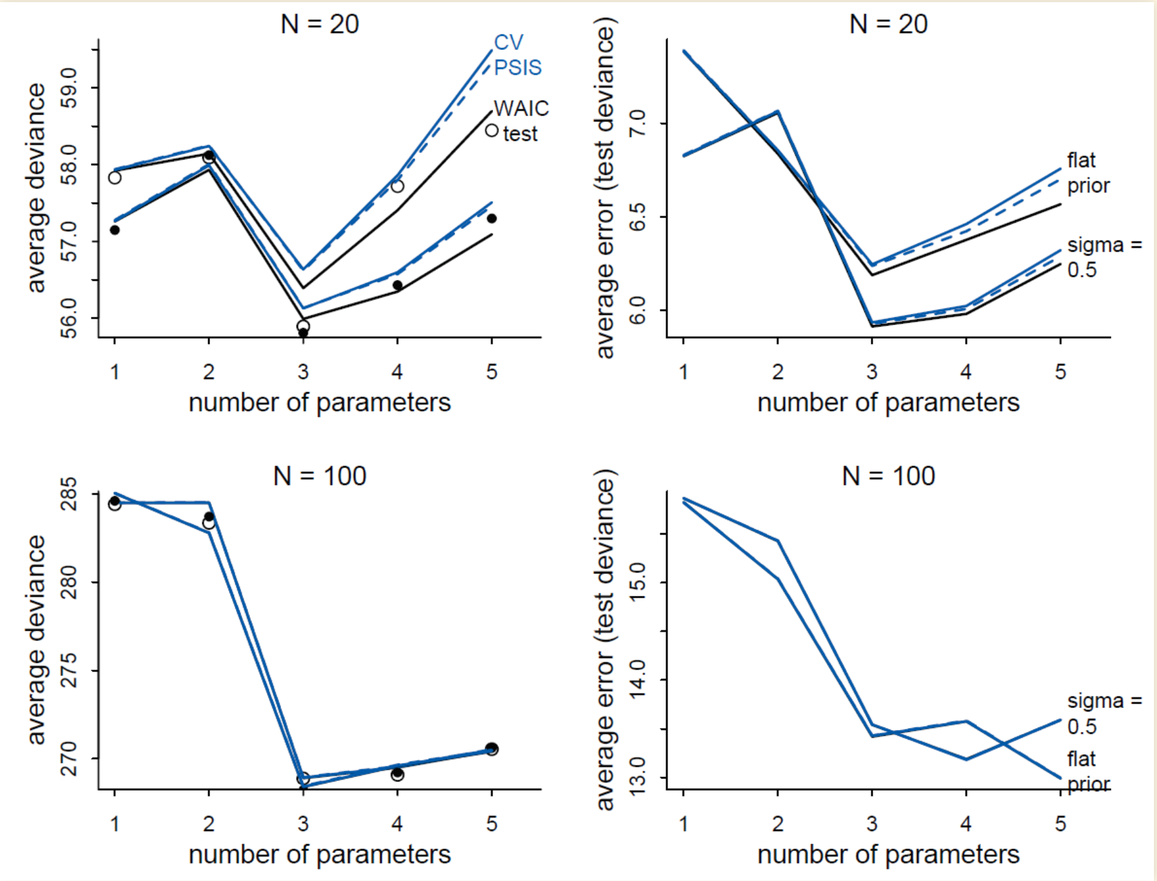
where `lppd` is the log-pointwise-predictive density (basically a sample of the posterior).



- Conditions for validity:
  - Priors are flat, or dominated by likelihood (data).
  - Posterior distribution is approximately Gaussian for each parameter.
  - The data sample size *N* is much greater than the number of parameters *k*.

Other Information Criteria

- AIC is only valid under these conditions:
  - Priors are flat, or dominated by likelihood (data).
  - Posterior distribution is approximately Gaussian for each parameter.
  - The data sample size *N* is much greater than the number of parameters *k*.
- Flat priors are usually not a good choice.
- DIC (Deviance Information Criteria) works with informative priors, but the other two criteria still apply.
- Watanabe-Akaike Information Criteria (WAIC, also called Widely Applicable Information Criterion) is more broadly applicable.
  - We won't go into details of calculating WAIC. The rethinking package will do it for us, and so will most other Bayesian analysis packages.
- General principle:
  - For all the information criteria we're examining, the smaller (more negative) they are, the better the model performs.



- Comparison of different measures

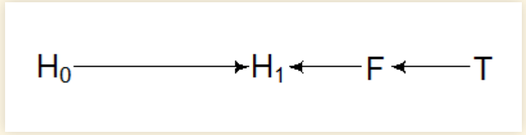
How to Compare Models

Comparison vs. Selection

- Many people use CV, PSIS, Deviance, or Information Criteria to select models
  - Use whatever model has the smallest score
- This is not wise. It only looks at what model is smallest, but doesn't consider how great the differences are between models.
  - This is like only looking at the mode (maximum) of the posterior and ignoring the rest of it.
  - The width the posterior matters too. It tells us about how uncertain the estimate is.
- When we compare models, look at how great the differences are between them.
- Remember that these criteria tell us about predictive power, but we have seen that predictive power doesn't tell us about causality.
  - Backdoor paths can have useful information, even though it's not causal.
    - But backdoor predictions only work if we don't interfere with the system.
      - In other words, if the future is just like the past.
    - In the plant-growth model, knowing about the fungus was a better predictor of plant growth than knowing about the anti-fungus treatment
      - but knowing about the fungus doesn't help us predict the effect of treating a field.

Example Using WAIC

- Plant growth experiment:
  - DAG



H_0 = height before, H_1 = height after,
T = anti-fungal treatment, F = fungus

  - Three models:

1. `mdl_0`: $\mu \sim$ log-Normal$(0, 0.25)$
2. `mdl_T`: $\mu = \alpha + \beta_T T$
3. `mdl_TF`: $\mu = \alpha + \beta_T T + \beta_F F$

```
set.seed(11)
round(WAIC(mdl_TF), 2)
```

```
##     WAIC    lppd penalty std_err
## 1 361.45 -177.17    3.55   14.17
```

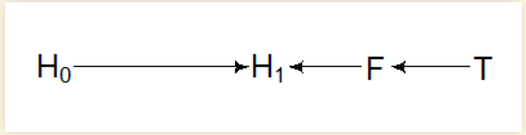
```
set.seed(77)
round(compare(mdl_0, mdl_T, mdl_TF, func = WAIC), 2)
```

```
##           WAIC    SE dWAIC   dSE pWAIC weight
## mdl_TF 361.81 14.26  0.00    NA  3.74      1
## mdl_T  402.65 11.20 40.84 10.44  2.58      0
## mdl_0  405.91 11.65 44.10 12.22  1.58      0
```

- Best predictions on top.
- “d” are differences from the best model.
- “SE” are standard errors.
- pWAIC is prediction penalty (estimate of *out-of-sample* vs. *in-sample*).
- weight gives the relative support for each model, given the data.
  - Useful for model-averaging.

Example Using WAIC

- Plant growth experiment:
  - DAG

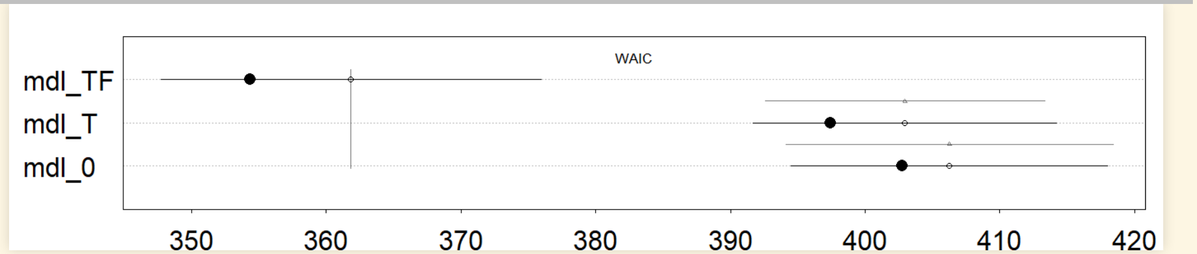


H_0 = height before, H_1 = height after,
T = anti-fungal treatment, F = fungus

  - Three models:

1. `mdl_0`: $\mu \sim \text{log-Normal}(0, 0.25)$
2. `mdl_T`: $\mu = \alpha + \beta_T T$
3. `mdl_TF`: $\mu = \alpha + \beta_T T + \beta_F F$

```
plot(compare(mdl_0, mdl_T, mdl_TF, func = WAIC), cex=2, lwd=2)
```



- Plot:
  - Line is range of estimated out-of-sample deviance
  - Gray point is best estimate of out-of-sample deviance
  - Black point is in-sample deviance
  - Light lines over models are differences from best model
- TF model is clearly the best for predictions
  - We can't tell which of the others is better
- TF model has post-treatment confounder
  - WAIC can't tell us about causation

Interactions

Interactions

- Multiple regression models may or may not include interactions:
  - Non-interacting models are *separable*: The effect of each variable is independent of all the other variables.

$$\mu = \alpha + \beta_x x + \beta_z z$$

    - Doubling *x* has the same effect on μ regardless what *z* is.
  - Interacting models are *non-separable*: The effect of one variable depends on another.

$$\mu = \alpha + \beta_x x + \beta_z z + \beta_{xz} xz$$

    - The effect of doubling *x* depends on *z*.
    - Example: Compared to non-smokers who aren't exposed to asbestos,
      - Smokers have 10.3 times greater risk of lung cancer.
      - Non-smokers exposed to asbestos have 7.4 times greater risk.
      - Smokers exposed to asbestos have 36.8 times greater risk.
        - If there was no interaction, the increase would be 17.7 times (10.3 + 7.4).

Interactions and DAGs

- A DAG can't tell you about an interaction.

- This figure could represent a model with or without interactions:

$$\mu = \alpha + \beta_x x + \beta_z z$$

or

$$\mu = \alpha + \beta_x x + \beta_z z + \beta_{xz} xz$$

- The DAG says that *x* and *z* influence *y*, but it doesn't say **how** they influence *y*.

Worked Example: Africa

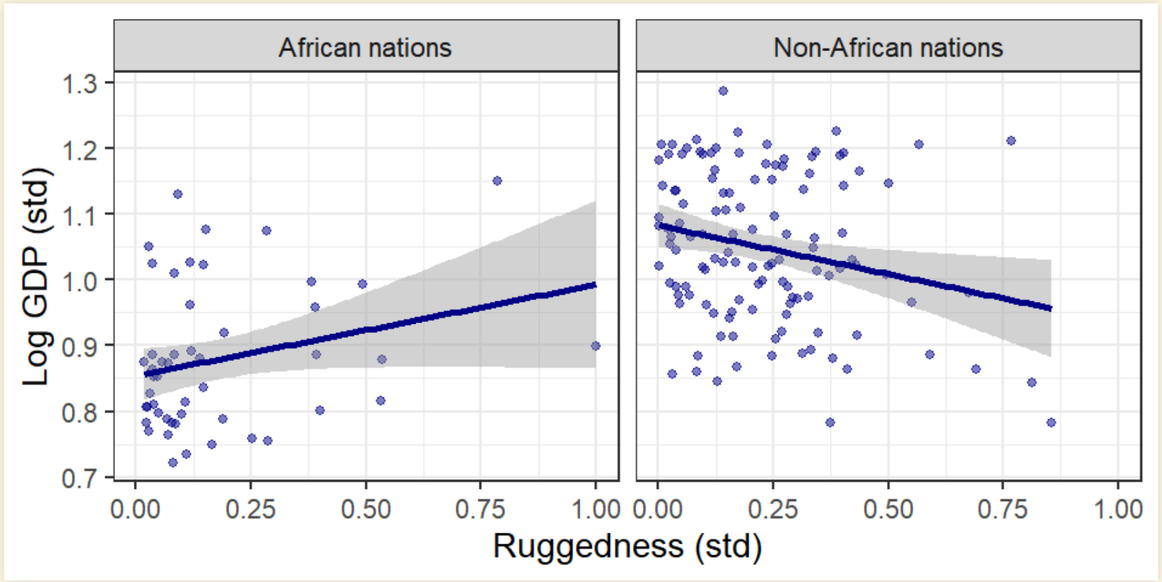
Geography and Economy in Africa

- Is there a relationship between topographic roughness and per-capita GDP?

```
data(rugged)
d <- rugged
```

  - Columns include:
    - `rgdppc_2000`: GDP per-capita in 2000
    - `rugged`: Terrain ruggedness.
    - `cont_africa`: Indicator variable if continent is Africa
  - Transform data

```
dd <- d |> filter(complete.cases(rgdppc_2000)) |>
  mutate(
    log_gdp = log(rgdppc_2000),
    log_gdp_std = log_gdp / mean(log_gdp),
    rugged_std = rugged / max(rugged)
  )
```




```
dd |> mutate(Africa = ifelse(cont_africa,"African nations",
                             "Non-African nations")) |>
  ggplot(aes(x = rugged_std, y = log_gdp_std)) +
  geom_point(size = 2, alpha = 0.5, color = "darkblue") +
  geom_smooth(method = "lm", color = "darkblue") +
  labs(x = "Ruggedness (std)", y = "Log GDP (std)") +
  facet_wrap(~Africa)
```

Model

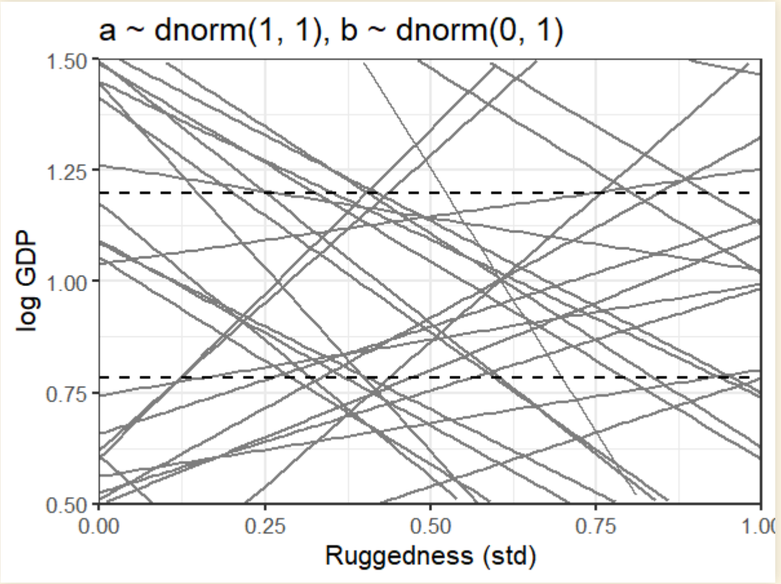
```
mdl_simple <- quap(
  alist(
    log_gdp_std ~ dnorm(mu , sigma),
    mu <- a + b * (rugged_std - 0.215),
    a ~ dnorm(1, 1),
    b ~ dnorm(0, 1),
    sigma ~ dexp(1)
  ), data = dd)
```

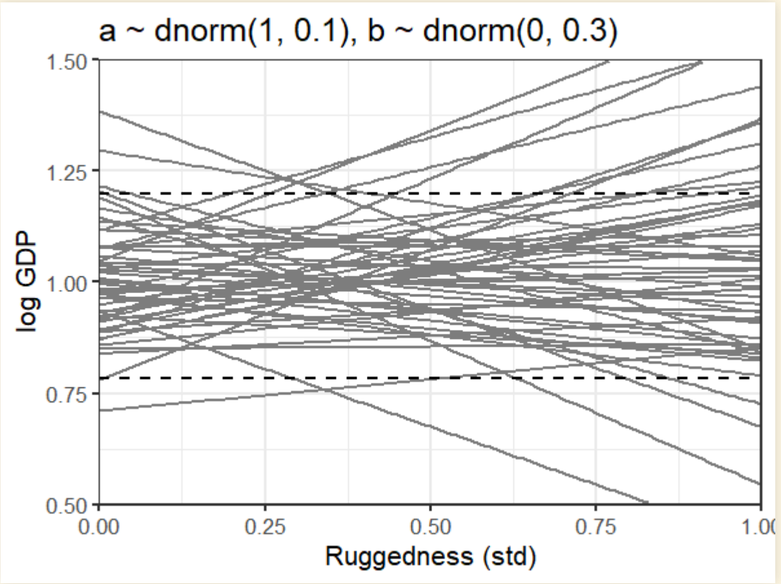
- The `0.215` is the average roughness for all countries.
- Now check the priors

```
prior <- extract.prior(mdl_simple, n = 50)
new_data <- tibble(rugged_std = seq(0, 1, 0.01))
prior_samples <- linpred_draws(
  mdl_simple, new_data, post = prior, value = "mu")
```

- Plot the priors

```
ggplot(prior_samples, aes(x = rugged_std, y = mu, group = .draw)) +
  geom_line(color = "gray50") +
  geom_hline(yintercept = quantile(dd$log_gdp_std, c(0.055, 0.945)),
             linetype = "dashed", size = 1) +
  labs(x = "Ruggedness (std)", y = "log GDP", )
```





Examine the model

```
mdl_simple <- quap(
  alist(
    log_gdp_std ~ dnorm(mu , sigma),
    mu <- a + b * (rugged_std - 0.215),
    a ~ dnorm(1, 0.1),
    b ~ dnorm(0, 0.3),
    sigma ~ dexp(1)
  ), data = dd)
```

```
precis_show(precis(mdl_simple, digits = 2))
```

```
##       mean   sd  5.5% 94.5%
## a     1.00 0.01  0.98  1.02
## b     0.00 0.05 -0.09  0.09
## sigma 0.14 0.01  0.12  0.15
```

- Create an interaction:
  - Intercept depends on the continent
  - Create an indicator variable:

```
# Make an indicator variable
dd <- dd |> mutate(cid = ifelse( dd$cont_africa==1 , 1 , 2 ))
```

```
mdl_inter <- quap(
    alist(
        log_gdp_std ~ dnorm(mu, sigma) ,
        mu <- a[cid] + b * (rugged_std - 0.215) ,
        a[cid] ~ dnorm(1, 0.1) ,
        b ~ dnorm(0, 0.3) ,
        sigma ~ dexp(1)
    ), data=dd)
```

```
compare(mdl_simple, mdl_inter) |> round(2)
```

```
##                WAIC    SE dWAIC   dSE pWAIC weight
## mdl_inter  -252.27 15.28  0.00    NA  4.24      1
## mdl_simple -188.70 13.31 63.57 15.16  2.71      0
```

```
precis_show(precis(mdl_inter, depth = 2, digits = 2))
```

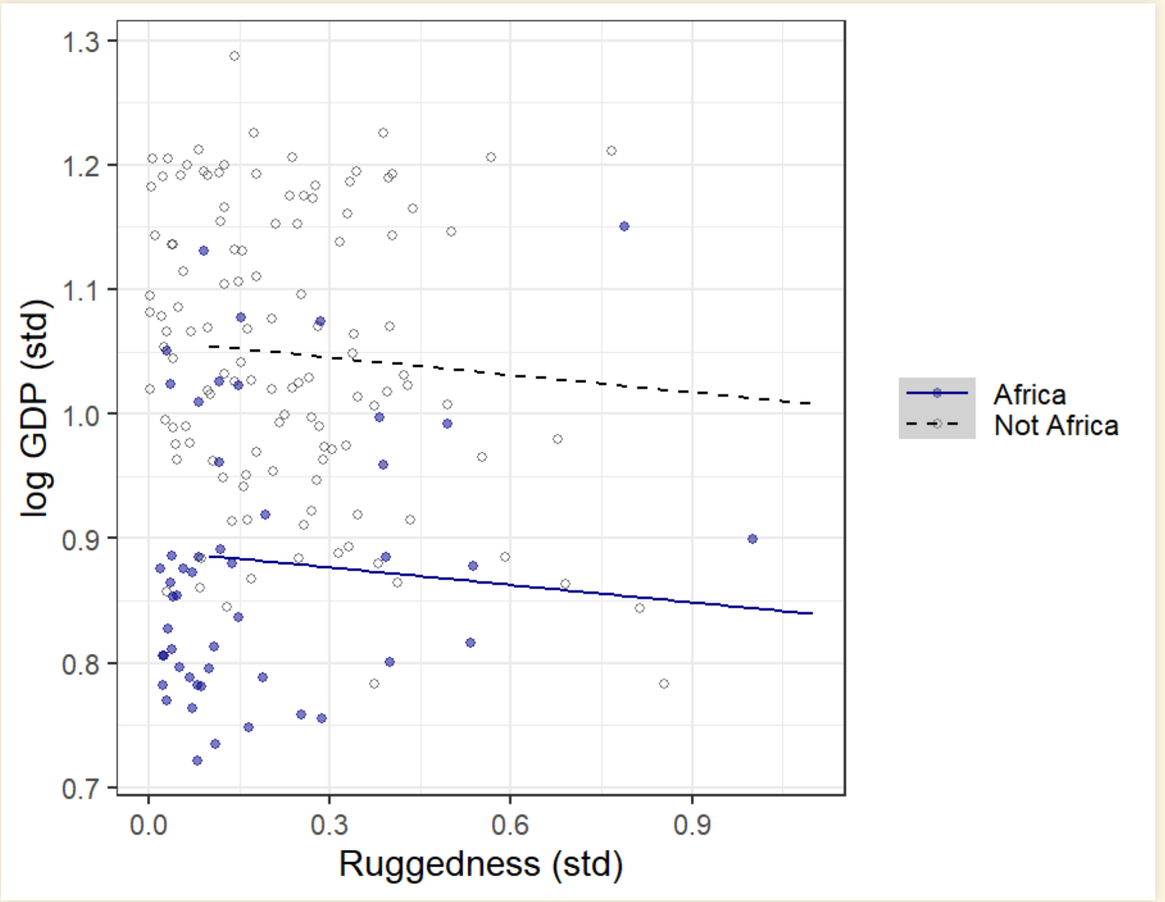
```
##         mean   sd  5.5% 94.5%
## a[1]    0.88 0.02  0.85  0.91
## a[2]    1.05 0.01  1.03  1.07
## b      -0.05 0.05 -0.12  0.03
## sigma   0.11 0.01  0.10  0.12
```

How does the model do?

```
rugged_seq <- seq(0.1, 1.1, length.out = 30)
mu_afr <- data.frame(rugged_std = rugged_seq, cid = 1)
mu_not_afr <- data.frame(rugged_std = rugged_seq, cid = 2)
new_data <- bind_rows(mu_afr, mu_not_afr)

post <- linpred_draws(mdl_inter, new_data) |>
  mutate(cid = ordered(cid, levels = c(1,2),
                       labels = c("Africa", "Not Africa")))
post_sum <- post |> group_by(.row, rugged_std, cid) |>
  summarize(mu = mean(.value),
            pi = list(PI(.value) |>
                        set_names(c("lower", "upper")))) |>
  ungroup() |> unnest_wider(pi)
```

```
dd |> mutate(cid = ordered(cid, levels = c(1,2),
                       labels = c("Africa", "Not Africa"))) |>
ggplot(aes(x = rugged_std, y = log_gdp_std,
           color = cid, shape = cid, linetype = cid)) +
  geom_point(size = 2, alpha = 0.5) +
  geom_smooth(aes(y = mu, ymin = lower, ymax = upper),
              data = post_sum, size = 1) +
  scale_color_manual(values = c("Africa" = "darkblue",
                                "Not Africa" = "black"),
                     name = NULL) +
  scale_linetype_manual(values = c("Africa" = "solid",
                                   "Not Africa" = "dashed"),
                        name = NULL) +
  scale_shape_manual(values = c("Africa" = 19,
                                "Not Africa" = 1),
                     name = NULL) +
  labs(x = "Ruggedness (std)", y = "log GDP (std)") +
  theme(legend.key.width = unit(3, "lines"))
```



Better Interactions

```
mdl_int_2 <- quap(
    alist(
        log_gdp_std ~ dnorm(mu, sigma) ,
        mu <- a[cid] + b[cid] * (rugged_std - 0.215) ,
        a[cid] ~ dnorm(1, 0.1) ,
        b[cid] ~ dnorm(0, 0.3) ,
        sigma ~ dexp(1)
    ), data=dd)
```

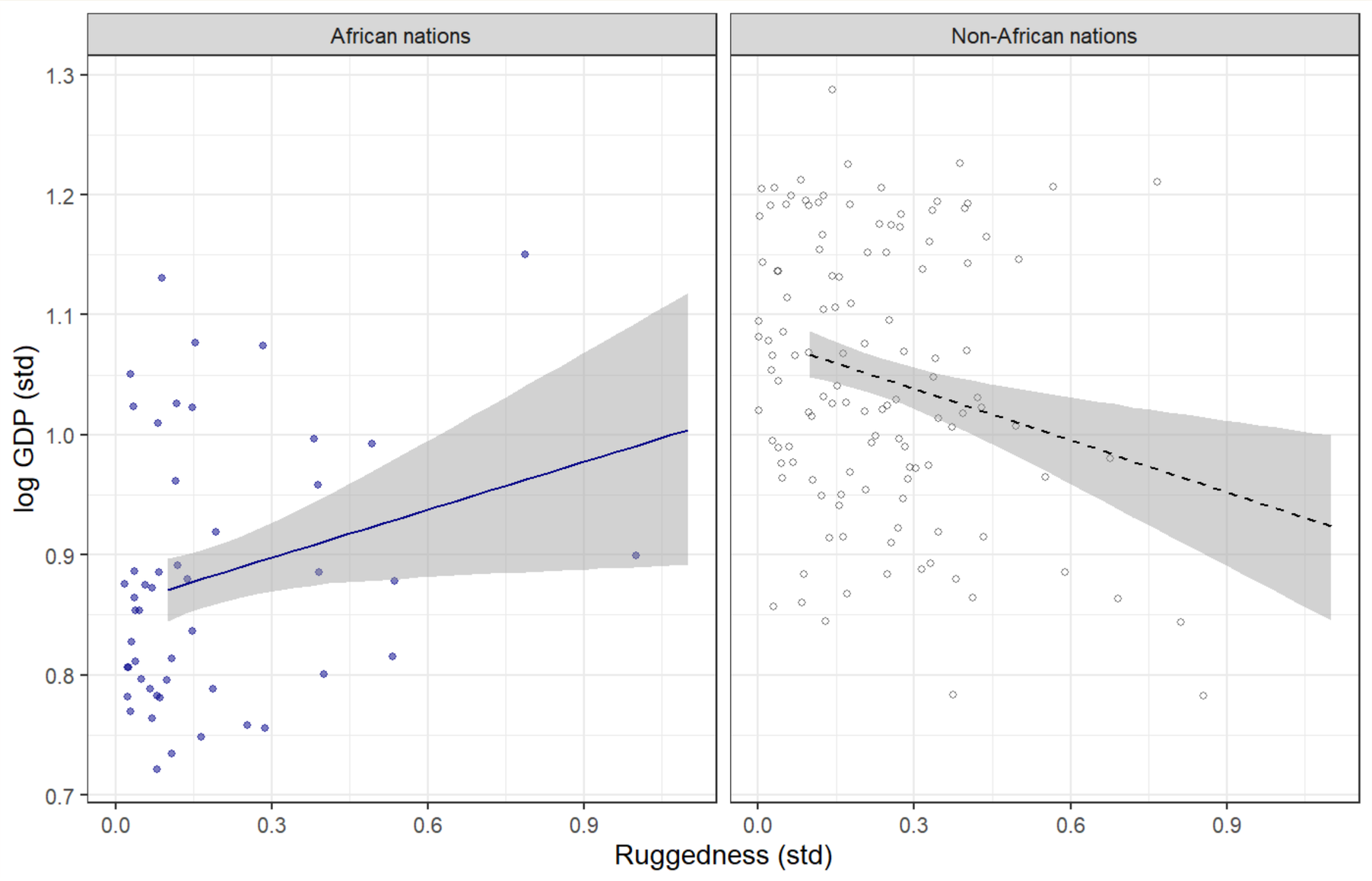
```
precis_show(precis(mdl_int_2, depth = 2, digits = 2))
```

```
##         mean   sd  5.5% 94.5%
## a[1]    0.89 0.02  0.86  0.91
## a[2]    1.05 0.01  1.03  1.07
## b[1]    0.13 0.07  0.01  0.25
## b[2]   -0.14 0.05 -0.23 -0.06
## sigma   0.11 0.01  0.10  0.12
```

```
compare(mdl_simple, mdl_inter, mdl_int_2) |> round(2)
```

```
##                WAIC    SE dWAIC   dSE pWAIC weight
## mdl_int_2  -258.68 15.21  0.00    NA  5.38   0.97
## mdl_inter  -251.99 15.25  6.69  6.69  4.36   0.03
## mdl_simple -189.00 13.29 69.68 15.47  2.58   0.00
```

How does the model do?



Continuous Interactions

A Winter Flower

- Tulips grown in greenhouses
  - Soil
  - Light
  - Water

```
data(tulips)
d <- tulips
head(tulips)
```

```
##   bed water shade blooms
## 1   a     1     1   0.00
## 2   a     1     2   0.00
## 3   a     1     3 111.04
## 4   a     2     1 183.47
## 5   a     2     2  59.16
## 6   a     2     3  76.75
```

```
levels(tulips$bed)
```

```
## [1] "a" "b" "c"
```

```
d <- d |> mutate(
  blooms_std = blooms / max(blooms),
  water_cent = water - mean(water),
  shade_cent = shade - mean(shade)
  )
```

- Predict `blooms` from `water`, and `shade`.
- Two models:
  - **Non-interacting:**

```
mdl_tulip_non <- quap(
  alist(
    blooms_std ~ dnorm(mu, sigma),
    mu <- a + bw * water_cent + bs * shade_cent,
    a ~ dnorm(0.5, 0.25),
    bw ~ dnorm(0, 0.25),
    bs ~ dnorm(0, 0.25),
    sigma ~ dexp(1)
  ), data=d)
```

  - **Interacting:**

```
mdl_tulip_inter <- quap(
  alist(
    blooms_std ~ dnorm(mu, sigma) ,
    mu <- a + bw * water_cent + bs * shade_cent +
          bws * water_cent * shade_cent,
    a ~ dnorm(0.5, 0.25),
    bw ~ dnorm(0, 0.25),
    bs ~ dnorm(0, 0.25),
    bws ~ dnorm(0, 0.25),
    sigma ~ dexp(1)
  ), data=d)
```

Interpreting the models