

Statistical Rethinking Winter 2020 – Homework Week 5

Alessandro Gentilini .@gmail (just auditing)

January 9, 2021

1. Consider the `data(Wines2012)` data table. These data are expert ratings of 20 different French and American wines by 9 different French and American judges. Your goal is to model `score`, the subjective rating assigned by each judge to each wine. I recommend standardizing it.

In this first problem, consider only variation among judges and wines. Construct index variables of `judge` and `wine` and then use these index variables to construct a linear regression model. Justify your priors. You should end up with 9 judge parameters and 20 wine parameters. Use `ulam` instead of `quap` to build this model, and be sure to check the chains for convergence. If you'd rather build the model directly in Stan or PyMC3 or Julia (Turing is a good choice!), go ahead. I just want you to use MCMC instead of quadratic approximation.

How do you interpret the variation among individual judges and individual wines?

Do you notice any patterns, just by plotting the differences?

Which judges gave the highest/lowest ratings?

Which wines were rated worst/best on average?

```
library(rethinking)
data(Wines2012)
Wines2012$score_std <- (Wines2012$score - mean(Wines2012$score)) / sd(Wines2012$score)
Wines2012$w_id <- as.integer(Wines2012$wine)
Wines2012$j_id <- as.integer(Wines2012$judge)

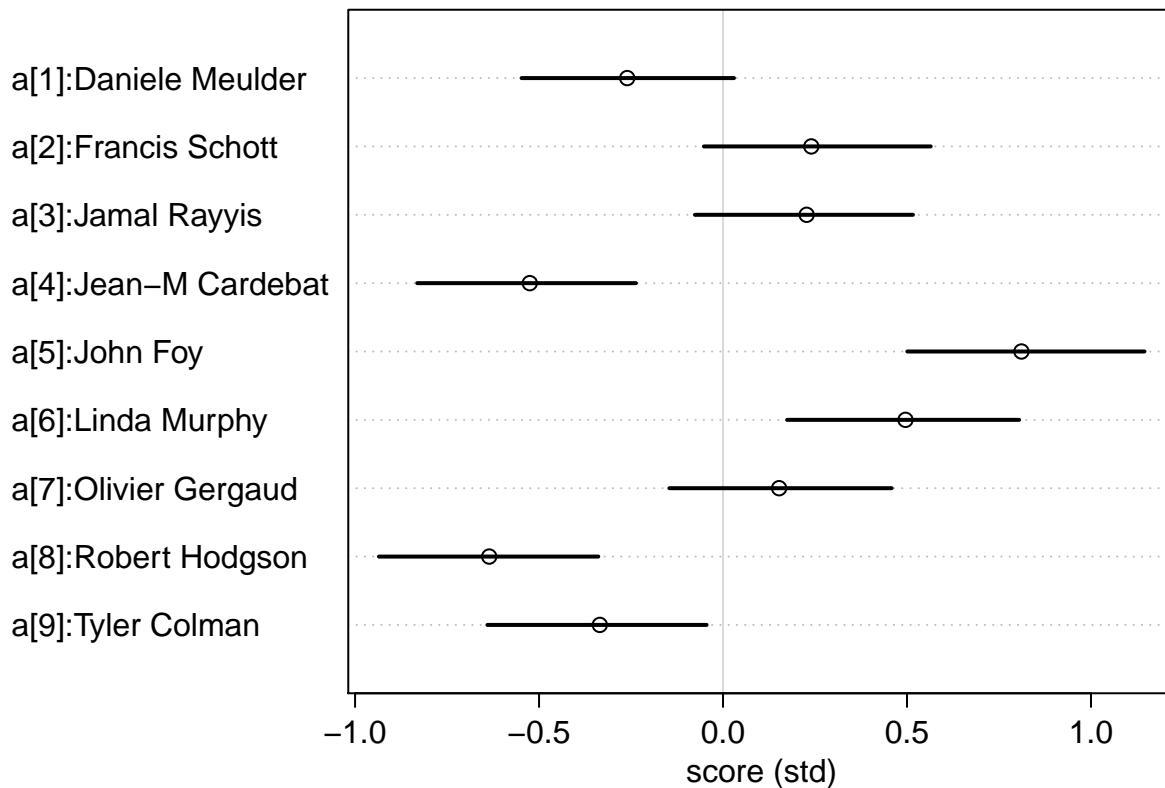
names(Wines2012)[names(Wines2012) == "wine.amer"] <- "wine_amer"
names(Wines2012)[names(Wines2012) == "judge.amer"] <- "judge_amer"

m1 <- ulam(
  alist(
    score_std ~ dnorm(mu, sigma),
    mu <- w[w_id] + j[j_id],
    w[w_id] ~ dnorm(0, 0.5),
    j[j_id] ~ dnorm(0, 0.5),
    sigma ~ dexp(1)
  ), data=Wines2012, chains=1)

##
## SAMPLING FOR MODEL '3b95caf74b1a96765826f44b98c606de' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
```

```
## Chain 1:
## Chain 1: Iteration: 1 / 1000 [ 0%] (Warmup)
## Chain 1: Iteration: 100 / 1000 [ 10%] (Warmup)
## Chain 1: Iteration: 200 / 1000 [ 20%] (Warmup)
## Chain 1: Iteration: 300 / 1000 [ 30%] (Warmup)
## Chain 1: Iteration: 400 / 1000 [ 40%] (Warmup)
## Chain 1: Iteration: 500 / 1000 [ 50%] (Warmup)
## Chain 1: Iteration: 501 / 1000 [ 50%] (Sampling)
## Chain 1: Iteration: 600 / 1000 [ 60%] (Sampling)
## Chain 1: Iteration: 700 / 1000 [ 70%] (Sampling)
## Chain 1: Iteration: 800 / 1000 [ 80%] (Sampling)
## Chain 1: Iteration: 900 / 1000 [ 90%] (Sampling)
## Chain 1: Iteration: 1000 / 1000 [100%] (Sampling)
## Chain 1:
## Chain 1: Elapsed Time: 0.066795 seconds (Warm-up)
## Chain 1: 0.055416 seconds (Sampling)
## Chain 1: 0.122211 seconds (Total)
## Chain 1:
```

```
labels <- paste( "a[" , 1:9 , "]" , levels(Wines2012$judge) , sep="" )
plot( precis( m1 , depth=2 , pars="j" ) , labels=labels ,
      xlab="score (std)" )
```



```
labels <- paste( "a[" , 1:20 , "]" , levels(Wines2012$wine) , sep="" )
plot( precis( m1 , depth=2 , pars="w" ) , labels=labels ,
      xlab="score (std)" )
```

a[1]:A1
a[2]:A2
a[3]:B1
a[4]:B2
a[5]:C1
a[6]:C2
a[7]:D1
a[8]:D2
a[9]:E1
a[10]:E2
a[11]:F1
a[12]:F2
a[13]:G1
a[14]:G2
a[15]:H1
a[16]:H2
a[17]:I1
a[18]:I2
a[19]:J1
a[20]:J2

