

Bayesian Chap 8 and 9 Assignment

Meng Yuan

Question 1

Here's the dataset, scores are standardized, judge and wine are constructed as index variables:

```
data(Wines2012)
d <- Wines2012
dat_list <- list(
  S = standardize(d$score),
  jid = as.integer(d$judge),
  wid = as.integer(d$wine)
)
```

Use weakly informative priors

The outcome variable is standardized, $\text{Normal}(0,0.5)$ makes a weakly informative prior.

```
m1.1 <- ulam(
  alist(
    S ~ dnorm(mu, sigma),
    mu <- a[jid] + w[wid],
    a[jid] ~ dnorm(0,0.5),
    w[wid] ~ dnorm(0,0.5),
    sigma ~ dexp(1)
  ), data=dat_list, chains=4, cores=4, log_lik=TRUE)
```

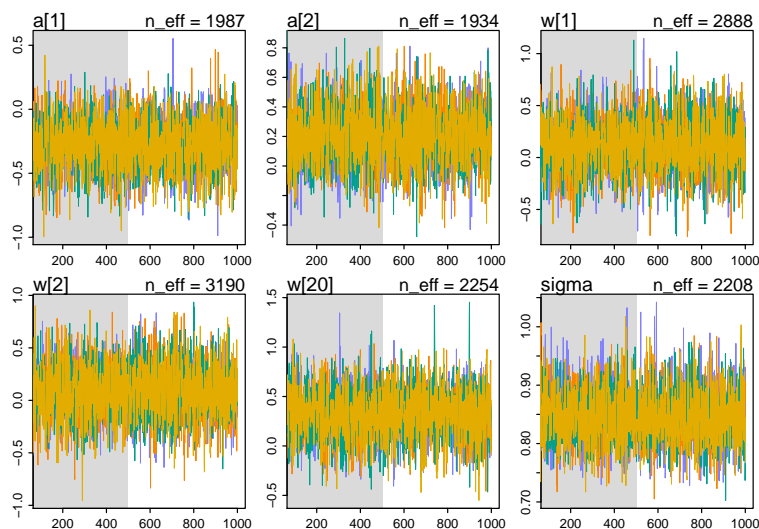
Four chains and the default sample size are used. The effective sample sizes show good efficiency of the chains, and Rhat values show good convergence of the chains (only some parameters are shown here).

```
precis(m1.1, 2, pars =c("a[1]", "a[2]", "w[1]", "w[2]", "w[20]", "sigma"))
```

##		mean	sd	5.5%	94.5%	n_eff	Rhat
##	a[1]	-0.28170711	0.20294642	-0.59874987	0.04857701	1987.101	1.0000470
##	a[2]	0.20898016	0.19475736	-0.10520990	0.51647929	1933.804	0.9993881
##	w[1]	0.11758405	0.27131757	-0.31420976	0.54245359	2887.922	0.9991201
##	w[2]	0.09204457	0.26346613	-0.32419560	0.51630949	3189.791	1.0008464
##	w[20]	0.33067030	0.24941093	-0.07977687	0.71176492	2253.771	1.0007444
##	sigma	0.84749386	0.04800843	0.77369716	0.92790991	2208.369	1.0000180

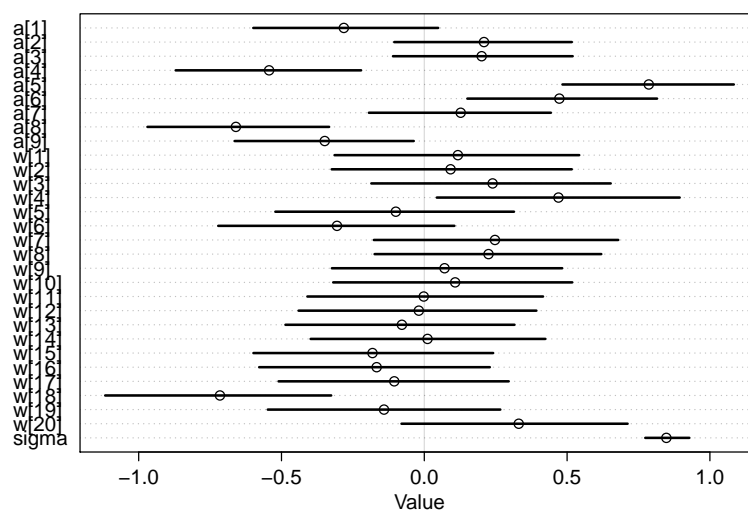
The shapes of traceplots show good mixing and stationarity of the chains (only some parameters are shown here).

```
traceplot( m1.1, pars =c("a[1]","a[2]","w[1]","w[2]","w[20]","sigma"))
```



Based on the posterior distribution of the parameters, $a[4]$, $a[5]$, $a[8]$, $a[9]$ show reliable deviations from zero, these judges give higher or lower ratings than average. There's less variation among the wines than among the judges, still $w[18]$ is smaller than zero, meaning the 18th wine rates the lowest. $w[4]$ is larger than the others, the 4th wine is the best.

```
precis_plot(precis(m1.1,2))
```



Use weaker priors

Normal(0,1) makes a weaker prior for the index variables compared to the previous Normal(0,0.5).

```
m1.2 <- ulam(
  alist(
    S ~ dnorm( mu , sigma ),
    mu <- a[jid] + w[wid],
```

```

a[jid] ~ dnorm(0,1),
w[wid] ~ dnorm(0,1),
sigma ~ dexp(1)
), data=dat_list, chains=4, cores=4, log_lik=TRUE)

```

The effective sample sizes (Rhat) are smaller, chains have less good convergence in this model (only some parameters are shown here).

```

precis( m1.2 , 2, pars =c("a[1]","a[2]","w[1]","w[2]","w[20]","sigma"))

```

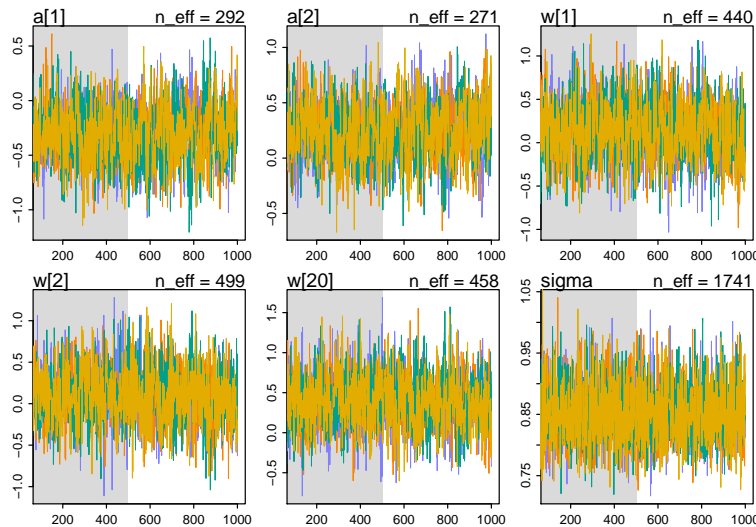
##		mean	sd	5.5%	94.5%	n_eff	Rhat
##	a[1]	-0.29250932	0.2657202	-0.7264541	0.1303364	291.7496	1.0095344
##	a[2]	0.25081309	0.2625365	-0.1699898	0.6498174	270.5595	1.0111108
##	w[1]	0.13260630	0.3303929	-0.3925266	0.6456976	439.7866	1.0050458
##	w[2]	0.09506653	0.3339924	-0.4442955	0.6224011	498.8694	1.0066893
##	w[20]	0.36960376	0.3353986	-0.1614452	0.8901225	458.4313	1.0066645
##	sigma	0.85122071	0.0462919	0.7811200	0.9292371	1741.4576	0.9998006

The traceplots show good mixing (only some parameters are shown).

```

traceplot(m1.2, pars =c("a[1]","a[2]","w[1]","w[2]","w[20]","sigma"))

```

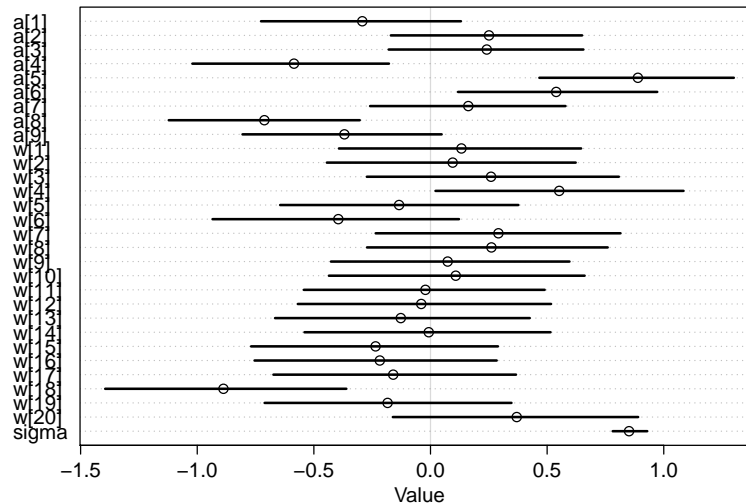


There are not many noticeable changes in the posterior distributions, in fact, model m1.1 and m1.2 have similar posterior distributions.

```

precis_plot(precis(m1.2,2))

```



But based on WAIC, model m1.1 is supported, meaning the weakly informative prior works better than the weaker prior. The regularization provided by the weakly informative prior leads to better prediction.

```
compare( m1.1 , m1.2 , func=WAIC )
```

	WAIC	SE	dWAIC	dSE	pWAIC	weight
m1.1	475.1142	18.39149	0.000000	NA	21.94665	0.94831105
m1.2	480.9331	18.29061	5.818877	2.376127	25.30361	0.05168895

Question 2

Indicator variable vs. index variable

Here's the model using **indicator variables**. W=1 indicates American wine, J=1 indicates American judge, and R=1 indicates red wine.

```
dat_list2 <- list(
  S = standardize(d$score),
  W = d$wine.amer,
  J = d$judge.amer,
  R = ifelse(d$flight=="red",1L,0L)
)
```

From the results of question 1 we know there's not much variation among the wines. Weakly informative prior can be used as: Normal(0,0.2) for the intercept and Normal(0,0.5) for the slopes.

```
m2a <- ulam(
  alist(
    S ~ dnorm( mu , sigma ),
    mu <- a + bW*W + bJ*J + bR*R,
```

```

a ~ dnorm( 0 , 0.2 ),
c(bW,bJ,bR) ~ dnorm( 0 , 0.5 ),
sigma ~ dexp(1)
), data=dat_list2 , chains=4 , cores=4, log_lik=TRUE)

```

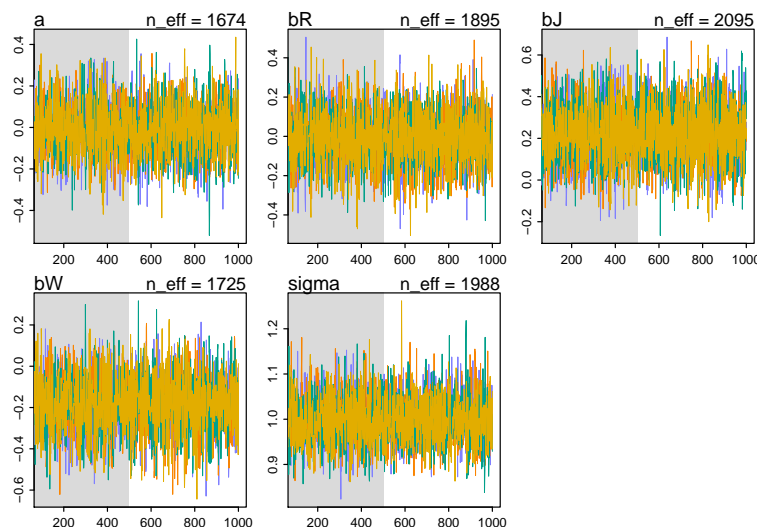
The posterior distributions show there are not much variation for the flight ($bR=0$), meaning the red and white wines are on average the same. It's also shown that American judges give higher ratings on average ($bJ>0$). American wines have slightly lower ratings on average than French wines (bW is mostly below zero). However, the absolute sizes of the differences due to wines or judges are not very large.

```
precis(m2a, 2)
```

##		mean	sd	5.5%	94.5%	n_eff	Rhat
##	a	-0.010514237	0.12615742	-0.213388783	0.19118745	1673.965	1.0012982
##	bR	-0.006364967	0.13400240	-0.225062357	0.20785074	1894.720	0.9988933
##	bJ	0.226558835	0.14168115	0.001351866	0.45040351	2094.905	1.0000318
##	bW	-0.182218608	0.13943649	-0.413888120	0.03470489	1724.591	0.9995300
##	sigma	0.999000586	0.05217143	0.919835712	1.08222234	1987.693	0.9983090

The effective sample sizes show good efficiency of the chains, and Rhat shows good convergence of the chains. The shapes of traceplots show good mixing and stationarity of the chains.

```
traceplot(m2a)
```



Here's the model using **index variables**: wid (1 for French wines and 2 for American/NJ wines), jid (1 for French judges and 2 for American judges), and fid (1 for red wines and 2 for white).

```
dat_list2b <- list(
  S = standardize(d$score),
  wid = d$wine.amer + 1L,
  jid = d$judge.amer + 1L,
  fid = ifelse(d$flight=="red",1L,2L)
)
```

Weakly informative priors $\text{Normal}(0,0.5)$ are used for the index variables.

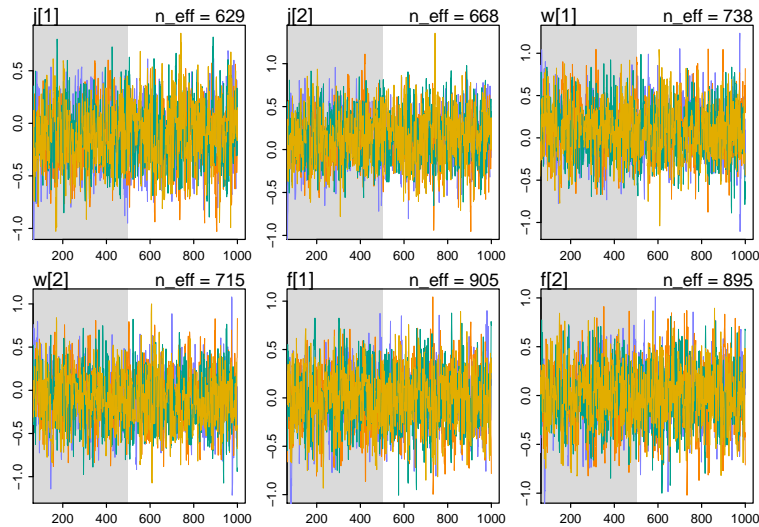
```
m2b <- ulam(
  alist(
    S ~ dnorm( mu , sigma ),
    mu <- w[wid] + j[jid] + f[fid],
    w[wid] ~ dnorm( 0 , 0.5 ),
    j[wid] ~ dnorm( 0 , 0.5 ),
    f[wid] ~ dnorm( 0 , 0.5 ),
    sigma ~ dexp(1)
  ), data=dat_list2b , chains=4 , cores=4, log_lik=TRUE)
```

```
precis(m2b, 2)
```

##		mean	sd	5.5%	94.5%	n_eff	Rhat
##	w[1]	0.082122385	0.31237318	-0.3908987	0.5998943	738.3898	1.0059307
##	w[2]	-0.099299857	0.31542658	-0.5907464	0.3928557	714.5424	1.0068358
##	j[1]	-0.104282850	0.29316844	-0.5897185	0.3688942	629.3233	1.0086428
##	j[2]	0.133792127	0.29562316	-0.3399557	0.6127106	667.9759	1.0086689
##	f[1]	-0.001798014	0.30010635	-0.4854507	0.4591932	905.3572	0.9993189
##	f[2]	0.001979142	0.30003176	-0.4671775	0.4730489	894.8212	0.9985813
##	sigma	0.998916623	0.05434317	0.9188838	1.0860208	1004.8749	0.9994578

The effective sample sizes show good efficiency of the chains, and Rhat values show good convergence of the chains. The shapes of traceplots show good mixing and stationarity of the chains (only some parameters are shown).

```
traceplot(m2b, pars =c("j[1]","j[2]","w[1]","w[2]","f[1]","f[2]"))
```



The posterior distributions from models using indicator and index variables are the same, as is shown by the wine parameter. The posterior distribution of `diff_w` from model `m2b` is very close to that of `bW` from model `m2b`.

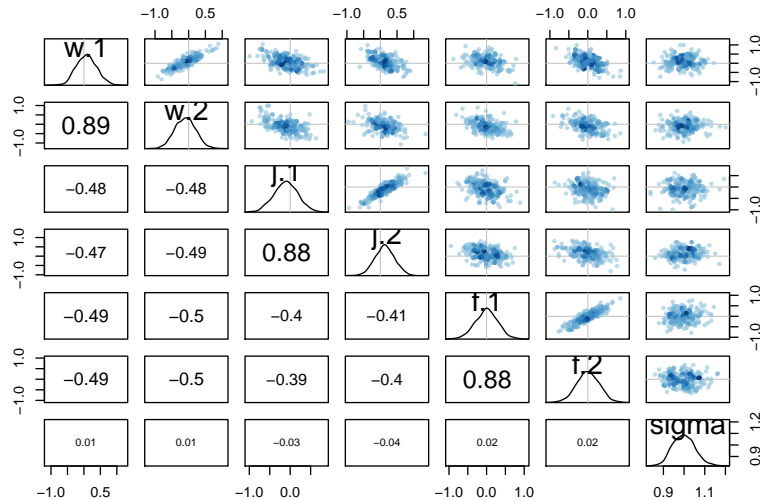
```
post <- extract.samples(m2b)
diff_w <- post$w[,2] - post$w[,1]
precis(diff_w)
```

```
##           mean          sd      5.5%      94.5%    histogram
## diff_w -0.1814222 0.1474229 -0.4184514 0.05706835
```

But using index variables means using more parameters. Based on the diagnostics, in the model using indicator variables (model `m2a`), the chains explore more efficiently (larger `n_eff`).

There are tight correlations for each pair of index parameters of the same type in model `m2b`. The redundant parameterization by using index variables causes higher correlations in the posterior, and less efficiency for the chains.

```
pairs(m2b)
```



Prior choice

Next I'll use tighter priors for the above models using indicator and index variables.

```
m2.1a <- ulam(
  alist(
    S ~ dnorm( mu , sigma ),
    mu <- a + bW*W + bJ*J + bR*R,
    a ~ dnorm( 0 , 0.2 ),
    c(bW,bJ,bR) ~ dnorm( 0 , 0.2 ),
    sigma ~ dexp(1)
  ), data=dat_list2 , chains=4 , cores=4, log_lik=TRUE)
```

```
m2.1b <- ulam(
  alist(
    S ~ dnorm( mu , sigma ),
    mu <- w[wid] + j[jid] + f[fid],
    w[wid] ~ dnorm( 0 , 0.2 ),
    j[wid] ~ dnorm( 0 , 0.2 ),
    f[wid] ~ dnorm( 0 , 0.2 ),
    sigma ~ dexp(1)
  ), data=dat_list2b , chains=4 , cores=4, log_lik=TRUE)
```

The difference in effective sample sizes becomes smaller. In the model using index variables (m2b, m2.1b), regularization of the prior (narrower priors) increase the efficiency of the chains.

```
precis(m2.1a, 2)
```

##		mean	sd	5.5%	94.5%	n_eff	Rhat
##	a	-0.0146502263	0.10943415	-0.19256934	0.16356708	1566.118	0.9996832
##	bR	-0.0009582105	0.11156203	-0.17494471	0.17801683	1855.451	1.0010383
##	bJ	0.1601155837	0.11821673	-0.02773514	0.34181624	1535.907	0.9988936


```
## bW      -0.1207990954 0.11268074 -0.29564686 0.07078237 1552.818 1.0018981
## sigma   0.9970085373 0.05271369  0.91587482 1.08522624 1931.666 0.9985435
```

```
precis(m2.1b, 2)
```

```
##           mean      sd      5.5%      94.5%    n_eff      Rhat
## w[1]    0.076629987 0.13709603 -0.1445064 0.2943151 1027.7003 1.0019241
## w[2]   -0.073953171 0.13582016 -0.3017318 0.1349857 1203.1859 1.0008761
## j[1]   -0.096458657 0.13804650 -0.3099899 0.1238903 1174.8546 1.0001248
## j[2]    0.098361847 0.13540716 -0.1183226 0.3104992 1209.1097 0.9996405
## f[1]    0.001881040 0.13596847 -0.2171607 0.2099885  988.6732 0.9994210
## f[2]   -0.000440123 0.13733871 -0.2188936 0.2185969 1117.1164 0.9997294
## sigma   0.999005958 0.05542621  0.9139139 1.0915056 1572.1493 1.0005897
```

WAIC supports models using indicator variables (m2a, m2.1a) and models with regularized priors (m2.1a, m2.1b), as discussed above.

```
compare(m2a, m2b, m2.1a, m2.1b)
```

```
##           WAIC      SE    dWAIC      dSE    pWAIC    weight
## m2.1a  513.1607 17.03607 0.000000      NA  3.423792 0.3941971
## m2.1b  514.1668 17.07538 1.006113 0.6725872 4.079696 0.2383629
## m2a    514.3074 17.12692 1.146758 1.2599288 4.231857 0.2221765
## m2b    515.1573 17.23320 1.996603 1.4131143 4.663209 0.1452636
```

Question 3

Here's the model using indicator variables, with interaction terms added to the model. The priors for the interaction terms are narrower as the interactions represent only part of the data.

```
m3a <- ulam(
  alist(
    S ~ dnorm( mu , sigma ),
    mu <- a + bW*W + bJ*J + bR*R +
          bWJ*W*J + bWR*W*R + bJR*J*R,
    a ~ dnorm(0,0.2),
    c(bW,bJ,bR) ~ dnorm(0,0.5),
    c(bWJ,bWR,bJR) ~ dnorm(0,0.25),
    sigma ~ dexp(1)
  ), data=dat_list2 , chains=4 , cores=4, log_lik=TRUE)
```

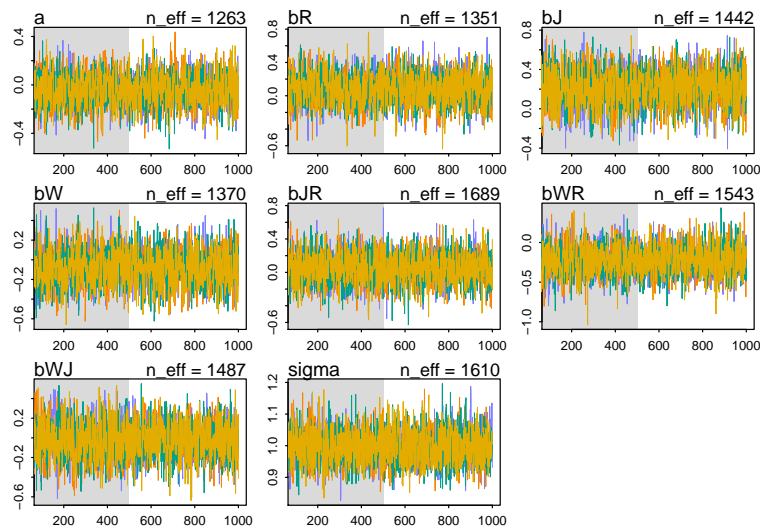
The effective sample sizes, Rhat values, and shapes of traceplots all look good. The effective sample sizes show good efficiency of the chains, and Rhat values show good convergence of the chains.

```
precis(m3a)
```

```
##           mean      sd      5.5%      94.5%    n_eff      Rhat
## a      -0.04605965 0.13296433 -0.26220271 0.16272358 1263.154 1.0026624
## bR       0.07957143 0.18787013 -0.21748748 0.37391877 1350.554 1.0012816
## bJ       0.21415771 0.17679223 -0.06890078 0.49715035 1441.883 0.9992292
## bW      -0.07595840 0.17738050 -0.36792160 0.20291056 1369.783 1.0012619
## bJR      0.04560301 0.17998210 -0.24352632 0.32763920 1688.696 0.9990402
## bWR     -0.21889912 0.18513889 -0.51873385 0.07922053 1543.468 1.0029046
## bWJ     -0.02885145 0.17833407 -0.30666519 0.25440370 1486.682 0.9988268
## sigma   0.99624717 0.05453164  0.91377656 1.08601533 1609.711 1.0014221
```

The shapes of traceplots show good mixing and stationarity of the chains.

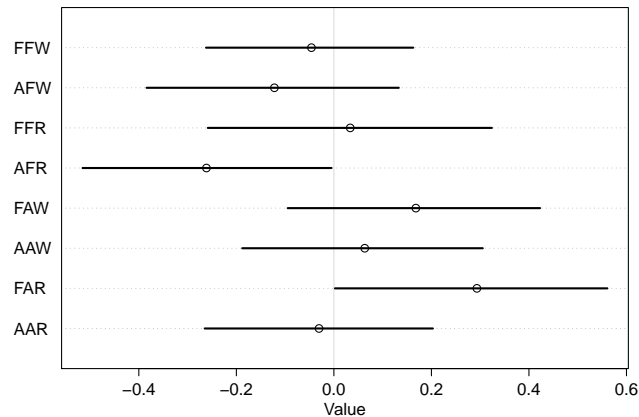
```
traceplot(m3a)
```



Next we can look at the posterior predictions of parameters for the interactions. The three letters represent country, judge, and flight, respectively. The rows labeled with AFR (NJ red wines as judged by French judges) and FAR (French red wines as judged by American judges) show the most deviation from zero. This shows that French judges don't like NJ red wines a lot, and American judges like French red wines more. The parameters for other interactions do not show reliable deviation from zero.

```
pred_dat <- data.frame(
  W = rep( 0:1 , times=4 ),
  J = rep( 0:1 , each=4 ),
  R = rep( c(0,0,1,1) , times=2 )
)
mu <- link( m3a , data=pred_dat )
row_labels <- paste( ifelse(pred_dat$W==1,"A","F") ,
  ifelse(pred_dat$J==1,"A","F") ,
  ifelse(pred_dat$R==1,"R","W") , sep=" " )
```

```
precis_plot( precis( list(mu=mu) , 2 ) , labels=row_labels)
```



Prior choice

Previously in model m3a, I used regularized priors $\text{Normal}(0, 0.25)$ for the interaction terms. Next I'll use tighter, $\text{Normal}(0, 0.1)$, and flatter, $\text{Normal}(0, 10)$ priors and see how they influence the results.

```
# flat prior
m3b <- ulam(
  alist(
    S ~ dnorm( mu , sigma ),
    mu <- a + bW*W + bJ*J + bR*R +
      bWJ*W*J + bWR*W*R + bJR*J*R,
    a ~ dnorm(0,0.2),
    c(bW,bJ,bR) ~ dnorm(0,0.5),
    c(bWJ,bWR,bJR) ~ dnorm(0,10),
    sigma ~ dexp(1)
  ), data=dat_list2 , chains=4 , cores=4, log_lik=TRUE)
```

```
# tighter prior
m3c <- ulam(
  alist(
    S ~ dnorm( mu , sigma ),
    mu <- a + bW*W + bJ*J + bR*R +
      bWJ*W*J + bWR*W*R + bJR*J*R,
    a ~ dnorm(0,0.2),
    c(bW,bJ,bR) ~ dnorm(0,0.5),
    c(bWJ,bWR,bJR) ~ dnorm(0,0.1),
    sigma ~ dexp(1)
  ), data=dat_list2 , chains=4 , cores=4, log_lik=TRUE)
```

Here's what their WAIC scores look like:

```
compare(m3a, m3b, m3c)
```

```
##           WAIC           SE      dWAIC      dSE    pWAIC    weight
## m3c 513.9799 17.15488 0.0000000      NA 4.462567 0.4685071
## m3a 514.4283 17.13076 0.4483974 1.199660 5.460224 0.3744104
## m3b 516.1655 17.28482 2.1855611 3.225452 6.871794 0.1570824
```

Model m3a and m3c with regularized priors perform similarly in terms of WAIC, and they are better supported than model m3b with flatter priors.

Model m3b uses a very flat prior for the interactions. The posterior distributions have some noticeable changes (AFW, FAW). A flat prior is less informative as a regularized one, which influences the predictions.

```
pred_dat <- data.frame(
  W = rep( 0:1 , times=4 ),
  J = rep( 0:1 , each=4 ),
  R = rep( c(0,0,1,1) , times=2 )
)
mu <- link( m3b , data=pred_dat )
row_labels <- paste( ifelse(pred_dat$W==1,"A","F") ,
  ifelse(pred_dat$J==1,"A","F") ,
  ifelse(pred_dat$R==1,"R","W") , sep="" )
precis_plot( precis( list(mu=mu) , 2 ) , labels=row_labels )
```

