Lesson 8: ANOVA

Lesson 8.2

Data

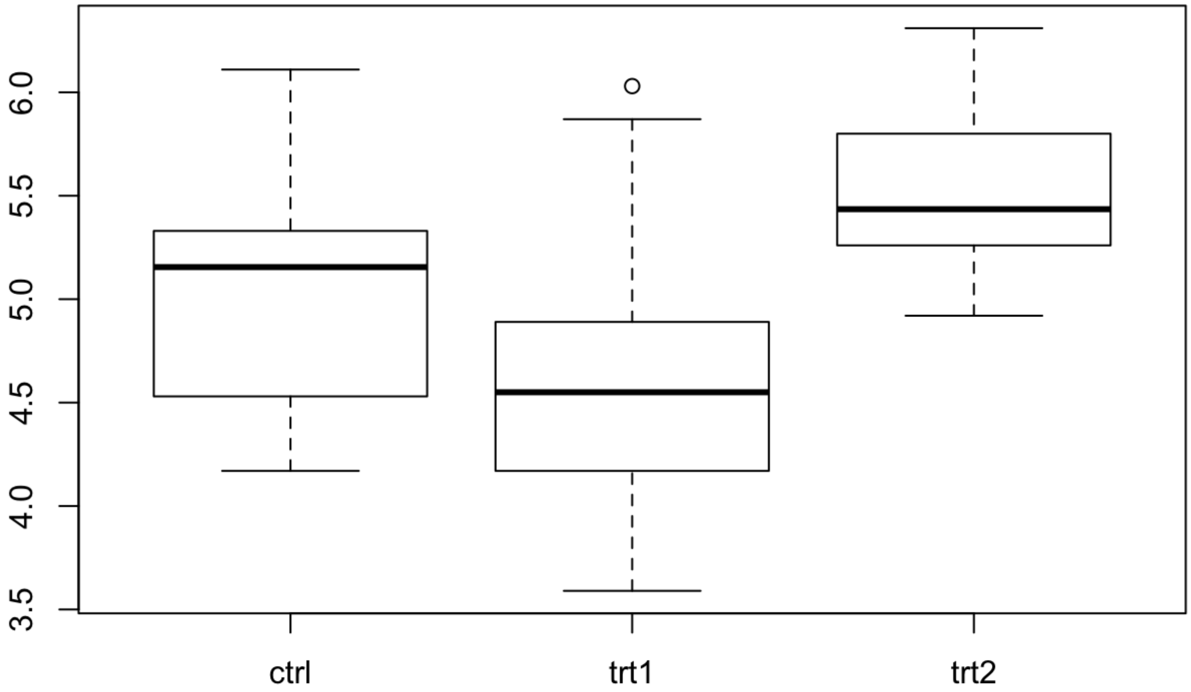
As an example of a one-way ANOVA, we'll look at the Plant Growth data in `R`.

```
data("PlantGrowth")
?PlantGrowth
head(PlantGrowth)
```

```
##   weight group
## 1   4.17  ctrl
## 2   5.58  ctrl
## 3   5.18  ctrl
## 4   6.11  ctrl
## 5   4.50  ctrl
## 6   4.61  ctrl
```

Because the explanatory variable `group` is a factor and not continuous, we choose to visualize the data with box plots rather than scatter plots.

```
boxplot(weight ~ group, data=PlantGrowth)
```



The box plots summarize the distribution of the data for each of the three groups. It appears that treatment 2 has the highest mean yield. It might be questionable whether each group has the same variance, but we'll assume that is the case.

Modeling

Again, we can start with the reference analysis (with a noninformative prior) with a linear model in `R`.

```
lmod = lm(weight ~ group, data=PlantGrowth)
summary(lmod)
```

```
##
## Call:
## lm(formula = weight ~ group, data = PlantGrowth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0710 -0.4180 -0.0060  0.2627  1.3690
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0320     0.1971  25.527   <2e-16 ***
## grouptrt1    -0.3710     0.2788  -1.331   0.1944
## grouptrt2     0.4940     0.2788   1.772   0.0877 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6234 on 27 degrees of freedom
## Multiple R-squared:  0.2641, Adjusted R-squared:  0.2096
## F-statistic: 4.846 on 2 and 27 DF,  p-value: 0.01591
```

```
anova(lmod)
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df  Sum Sq Mean Sq F value  Pr(>F)
## group      2  3.7663  1.8832  4.8461 0.01591 *
## Residuals 27 10.4921  0.3886
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# plot(lmod) # for graphical residual analysis
```

The default model structure in `R` is the linear model with dummy indicator variables. Hence, the “intercept” in this model is the mean yield for the control group. The two other parameters are the estimated effects of treatments 1 and 2. To recover the mean yield in treatment group 1, you would add the intercept term and the treatment 1 effect. To see how `R` sets the model up, use the `model.matrix(lmod)` function to extract the X matrix.

The `anova()` function in `R` compares variability of observations between the treatment groups to variability within the treatment groups to test whether all means are equal or whether at least one is different. The small p-value here suggests that the means are not all equal.

Let’s fit the cell means model in `JAGS`.

```
library("rjags")
```

```
## Loading required package: coda
```

```
## Linked to JAGS 4.2.0
## Loaded modules: basemod,bugs
```

```
mod_string = " model {
    for (i in 1:length(y)) {
        y[i] ~ dnorm(mu[grp[i]], prec)
    }

    for (j in 1:3) {
        mu[j] ~ dnorm(0.0, 1.0/1.0e6)
    }

    prec ~ dgamma(5/2.0, 5*1.0/2.0)
    sig = sqrt( 1.0 / prec )
} "

set.seed(82)
str(PlantGrowth)
data_jags = list(y=PlantGrowth$weight,
              grp=as.numeric(PlantGrowth$group))

params = c("mu", "sig")

inits = function() {
    inits = list("mu"=rnorm(3,0.0,100.0), "prec"=rgamma(1,1.0,1.0))
}

mod = jags.model(textConnection(mod_string), data=data_jags, inits=inits, n.chains=3)
update(mod, 1e3)

mod_sim = coda.samples(model=mod,
                        variable.names=params,
                        n.iter=5e3)
mod_csim = as.mcmc(do.call(rbind, mod_sim)) # combined chains
```

Model checking

As usual, we check for convergence of our MCMC.

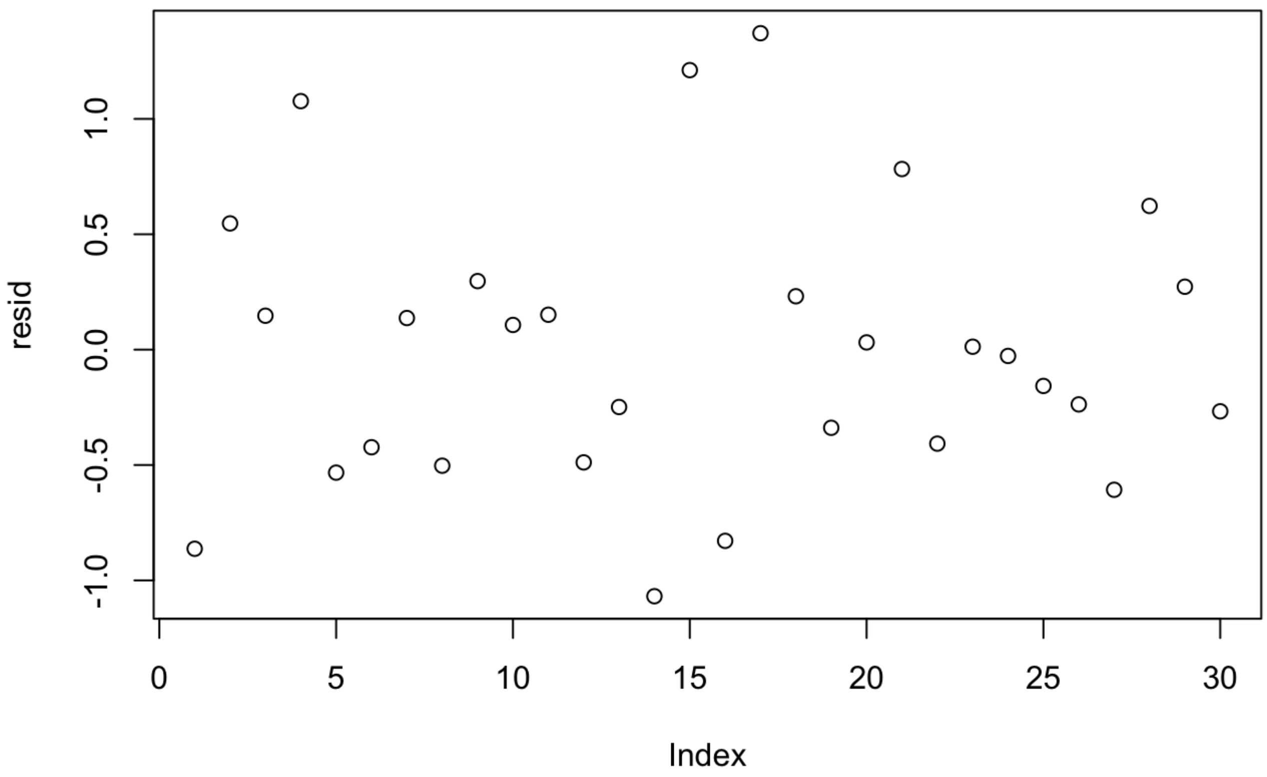
```
plot(mod_sim)

gelman.diag(mod_sim)
autocorr.diag(mod_sim)
effectiveSize(mod_sim)
```

We can also look at the residuals to see if there are any obvious problems with our model choice.

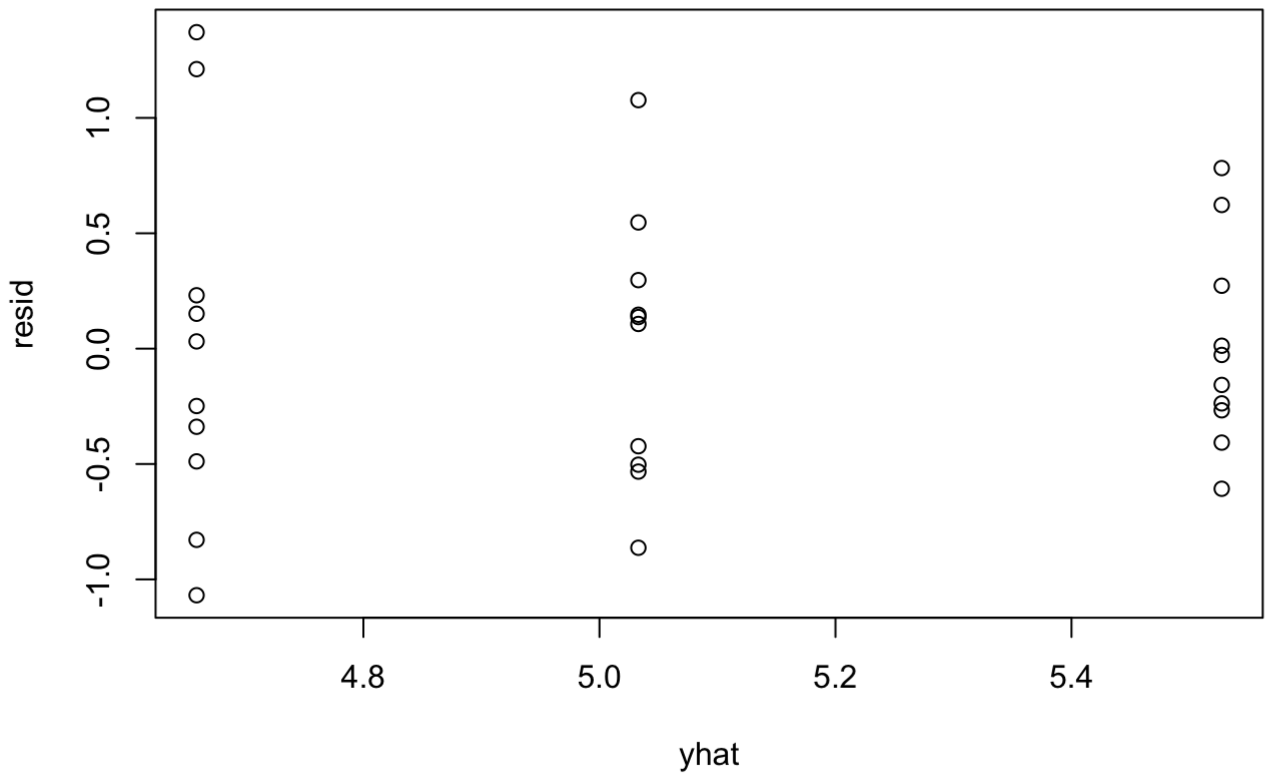
```
(pm_params = colMeans(mod_csim))
```

```
##    mu[1]    mu[2]    mu[3]      sig
## 5.032975 4.658663 5.527312 0.711631
```

```
yhat = pm_params[1:3][data_jags$grp]
resid = data_jags$y - yhat
plot(resid)
```



```
plot(yhat, resid)
```



Again, it might be appropriate to have a separate variance for each group. We will have you do that as an exercise.

Results

Let’s look at the posterior summary of the parameters.

```
summary(mod_sim)
```

```
##
## Iterations = 1001:6000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 5000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##         Mean     SD  Naive SE Time-series SE
## mu[1] 5.0330 0.2261 0.0018460      0.0018458
## mu[2] 4.6587 0.2252 0.0018384      0.0018384
## mu[3] 5.5273 0.2277 0.0018590      0.0018573
## sig   0.7116 0.0927 0.0007569      0.0008271
##
## 2. Quantiles for each variable:
##
##         2.5%    25%    50%   75%  97.5%
## mu[1] 4.5856 4.8847 5.0343 5.185 5.4792
## mu[2] 4.2173 4.5108 4.6577 4.808 5.1065
## mu[3] 5.0775 5.3772 5.5279 5.678 5.9722
## sig   0.5582 0.6457 0.7015 0.767 0.9213
```

```
HPDinterval(mod_csim)
```

```
##           lower     upper
## mu[1] 4.5848092 5.4782671
## mu[2] 4.2193156 5.1085423
## mu[3] 5.0669822 5.9605551
## sig   0.5404645 0.8934269
## attr(,"Probability")
## [1] 0.95
```

The `HPDinterval()` function in the `coda` package calculates intervals of highest posterior density for each parameter.

We are interested to know if one of the treatments increases mean yield. It is clear that treatment 1 does not. What about treatment 2?

```
mean(mod_csim[,3] > mod_csim[,1])
```

```
## [1] 0.9397333
```

There is a high posterior probability that the mean yield for treatment 2 is greater than the mean yield for the control group.

It may be the case that treatment 2 would be costly to put into production. Suppose that to be worthwhile, this treatment must increase mean yield by 10%. What is the posterior probability that the increase is at least that?

```
mean(mod_csim[,3] > 1.1*mod_csim[,1])
```

```
## [1] 0.4893333
```

We have about 50/50 odds that adopting treatment 2 would increase mean yield by at least 10%.