Resolución

1.1.

```
> library(rstan)
> rstan_options(auto_write = TRUE)
> options(mc.cores = parallel::detectCores())
> y <- c(15, 7, 5, 11, 10, 7, 25, 9, 21, 1, 2, 2, 0, 10, 4, 8, 0, 10, 17, 4)
> w <- c(2, 2, 3, 2, 3, 3, 1, 3, 1, 4, 5, 4, 5, 1, 5, 4, 5, 2, 1, 4)
> datos <- list(n = length(w), w = w, y = y)
> codigo <- "
+   data {
+     int<lower=0> n;
+     int<lower=0> w[n];
+     int<lower=0> y[n];
+   }
+
+   parameters {
+   real tau;
+   real omega;
+   }
+   model{
+   tau ~ normal(0,5);
+   omega ~ normal(0,5);
+         for(i in 1:n){
+                 y[i] ~ poisson(exp(tau+omega*w[i]));
+         }
+   }
+   generated quantities{
+   real D;
+   D =  0;
+   for (i in 1:n){
+   D += poisson_lpmf(y[i] | exp(tau + omega * w[i]));
+   }
+   D*= -2;
+   }
+   "
> fit <- stan(model_code = codigo, data = datos, iter = 2000)
> print(fit)
```

```
Inference for Stan model: anon_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

        mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
tau     3.47    0.00 0.15   3.16   3.37   3.47   3.57   3.76  1028    1
omega  -0.54    0.00 0.06  -0.67  -0.59  -0.54  -0.50  -0.42  1067    1
D     101.51    0.06 1.96  99.60 100.10 100.89 102.32 106.71  1089    1
lp__  229.59    0.03 0.98 226.99 229.18 229.89 230.30 230.54  1097    1

Samples were drawn using NUTS(diag_e) at Tue May  2 14:04:22 2023.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

```
> D1<- unlist(extract(fit, pars='D'))
> DIC1 <- mean(D1) + 0.5 *var(D1)
> print(DIC1)
```

```
[1] 103.443
```

1.2

```
> y <- c(15, 7, 5, 11, 10, 7, 25, 9, 21, 1, 2, 2, 0, 10, 4, 8, 0, 10, 17, 4)
> w <- c(2, 2, 3, 2, 3, 3, 1, 3, 1, 4, 5, 4, 5, 1, 5, 4, 5, 2, 1, 4)
> z <- c(4, 1, 1, 3, 4, 2, 4, 3, 3, 1, 3, 2, 1, 1, 4, 4, 2, 2, 2, 3)
> n <- length(y)
> m <- max(z)
```

```
> datos <- list(n=n, m=m, y=y, w=w, z=z)
> codigo <- "
+ data {
+ int n;
+ int m;
+ int y[n];
+ int w[n];
+ int z[n];
+ }
+
+ parameters {
+ real tau[m] ;
+ real omega ;
+ real muTau;
+ real<lower=0> sigmaTau;
+ }
+
+ model {
+ real peso;
+
+
+ muTau ~ normal(0, 5);
+ sigmaTau ~ chi_square(1);
+ tau ~ normal(muTau, sigmaTau);
+ omega ~ normal(0,5);
+
+ for(i in 1:n){
+ peso = tau[z[i]] + omega*w[i];
+ y[i] ~ poisson(exp(peso));
+ }
+
+ }
+
+ generated quantities{
+ int yPred[n];
+ real peso;
+ real D;
+
+ D = 0;
+ for(i in 1:n){
+ peso = tau[z[i]] + omega*w[i];
+ yPred[i] = poisson_rng(exp(peso));
+ D += poisson_lpmf(y[i] | exp(peso));
+ }
+ D *= -2;
+
+ }
+ "
> fit <- stan(model_code = codigo, data = datos)
> print(fit, pars=c("tau", "omega", "muTau", "sigmaTau"))
```

```
Inference for Stan model: anon_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

          mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
tau[1]    2.98    0.01 0.24  2.48  2.82  2.98  3.14  3.42  1727    1
tau[2]    3.33    0.01 0.20  2.94  3.19  3.33  3.46  3.71  1600    1
tau[3]    3.56    0.00 0.19  3.18  3.43  3.56  3.69  3.93  1746    1
tau[4]    3.81    0.00 0.19  3.45  3.68  3.81  3.94  4.16  1568    1
omega    -0.54    0.00 0.06 -0.67 -0.58 -0.54 -0.50 -0.42  1349    1
muTau     3.39    0.01 0.35  2.67  3.21  3.41  3.59  4.04  1381    1
sigmaTau  0.54    0.01 0.38  0.16  0.31  0.43  0.64  1.49  1340    1

Samples were drawn using NUTS(diag_e) at Tue May  2 14:05:43 2023.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

```
> D2 <- unlist(extract(fit, pars='D'))
> DIC2 <- mean(D2) + 0.5 * var(D2)
> print(DIC2)
```

```
[1] 90.15336
```

```
> yPred <- extract(fit, "yPred")
> for(i in 1:max(z)){
+ med <- mean(y[which(z==i)])
+ medPred <- rowMeans(yPred[[1]][,which(z==i)])
+ pPost <- mean(medPred >= med)
+ etiqueta <- sprintf("Universidad %1.0f. medObs = %.3f, medPred = %.3f(%.3f), pPost = %.3f\n",
+ i, med, mean(medPred), sd(medPred), pPost)
+ cat(etiqueta)
+ }
```

```
Universidad 1. medObs = 4.600, medPred = 5.213(1.459), pPost = 0.667
Universidad 2. medObs = 7.200, medPred = 7.334(1.659), pPost = 0.524
Universidad 3. medObs = 9.400, medPred = 9.185(1.883), pPost = 0.460
Universidad 4. medObs = 12.400, medPred = 11.854(2.155), pPost = 0.404
```

1.3.

```
> cat(sprintf("\nModelo 1. mean D = %8.2f. pD = %8.2f. DIC = %8.2f\n
+ Modelo 2. mean D = %8.2f. pD = %8.2f. DIC = %8.2f\n",
+ mean(D1), 0.5*var(D1), DIC1, mean(D2), 0.5*var(D2), DIC2))
```

```
Modelo 1. mean D =   101.51. pD =     1.93. DIC =   103.44

Modelo 2. mean D =    84.75. pD =     5.40. DIC =    90.15
```

Como podemos observar, el modelo 2 está más cercano a los datos, compensando su mayor complejidad. En esta misma línea de bondad de ajuste, observamos cómo por cada Universidad, las medias de errores observados y predichos no se alejan en exceso. Elegimos pues el modelo 2.