CHAPTER

7

t-Test: Equal and Unequal Variances

OUTLINE

One of the most commonly used linear models is that underlying the simple t-test. Actually, almost as with the computation of the arithmetic mean, many people wouldn't even think of a t-test as being a form of linear model. The t-test comes in two flavors; one for the case with equal variances and another for unequal variances. We will look at both in this chapter.

Introduction to WinBUGS for Ecologists
DOI: 10.1016/B978-0-12-378605-0.00007-7

7.1 t-TEST WITH EQUAL VARIANCES

7.1.1 Introduction

The model underlying the t-test with equal variances states that:

$$y_i = \alpha + \beta * x_i + \varepsilon_i$$
$$\varepsilon_i \sim Normal\,(0, \sigma^2)$$

Here, a response y_i is a measurement on a continuous scale taken on individuals i, which belong to either of two groups, and x_i is an indicator or dummy variable for group 2. (See Chapter 6 for different parameterizations of this model.) This simple t-test model has three parameters, the mean α for group 1, the difference in the means between groups 1 and 2 (β) and the variance σ^2 of the normal distribution from which the residuals ε_i are assumed to have come from.

7.1.2 Data Generation

We first simulate data under this model and for a motivating example return to peregrine falcons. We imagine that we had measured male and female wingspan and are interested in a sex difference in size. For Western Europe, Monneret et al. (2006) gives the range of male and female wingspan as 70–85 cm and 95–115 cm, respectively. Assuming normal distributions for wingspan, this implies means and standard deviations of about 77.5 and 2.5 cm for males and 105 and 3 cm for females.

```
n1 <- 60                          # Number of females
n2 <- 40                          # Number of males
mu1 <- 105                        # Population mean of females
mu2 <- 77.5                       # Population mean of males
sigma <- 2.75                     # Average population SD of both

n <- n1+n2                        # Total sample size
y1 <- rnorm(n1, mu1, sigma)       # Data for females
y2 <- rnorm(n2, mu2, sigma)       # Date for males
y <- c(y1, y2)                    # Aggregate both data sets
x <- rep(c(0,1), c(n1, n2))       # Indicator for male
boxplot(y ~ x, col = "grey", xlab = "Male", ylab = "Wingspan (cm)", las = 1)
```

The manner in which we just generated this data set (Fig. 7.1) corresponds in a way to a means parameterization of the linear model of the t-test. Here is a different way to generate a set of the same kind of data. Perhaps it lets one see more clearly the principle of an effects parameterization of the linear model:

```
n <- n1+n2                       # Total sample size
alpha <- mu1                     # Mean for females serves as the intercept
```

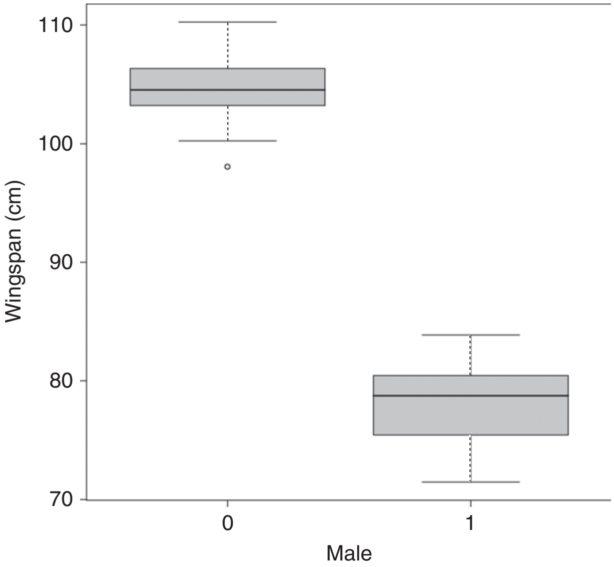

FIGURE 7.1 A boxplot of the generated data set on wingspan of female and male peregrines when the residual variance is constant (0 - females, 1 - males).

```
beta <- mu2-mu1                        # Beta is the difference male-female
E.y <- alpha + beta*x                  # Expectation
y.obs <- rnorm(n = n, mean = E.y, sd = sigma)     # Add random variation
boxplot(y.obs ~ x, col = "grey", xlab = "Male", ylab = "Wingspan (cm)", las = 1)
```

An important aside (again): To get a feel for the effect of chance, or technically, for sampling variance (= sampling error), you can repeatedly execute one of the previous sets of commands and observe how different repeated realizations of the same random process are.

7.1.3 Analysis Using R

There is an R function called `t.test()`, but we will use the linear model function `lm()` instead to fit the t-test with equal variances for both groups.

```
fit1 <- lm(y ~ x)             # Analysis of first data set
fit2 <- lm(y.obs ~ x)         # Analysis of second data set
summary(fit1)
summary(fit2)
```

```
> fit1 <- lm(y ~ x)
> fit2 <- lm(y.obs ~ x)
> summary(fit1)
[...]
```

```
Coefficients:
             Estimate  Std. Error  t value   Pr(>|t|)
(Intercept) 104.6452      0.3592   291.37    <2e-16 ***
x           -26.5737      0.5679   -46.80    <2e-16 ***
[...]
Residual standard error: 2.782 on 98 degrees of freedom
Multiple R-squared: 0.9572,     Adjusted R-squared: 0.9567
F-statistic: 2190 on 1 and 98 DF,    p-value: < 2.2e-16

> summary(fit2)

[...]

Coefficients:
             Estimate  Std. Error  t value   Pr(>|t|)
(Intercept) 105.1785      0.3621   290.45    <2e-16 ***
x           -27.6985      0.5726   -48.38    <2e-16 ***
[...]
Residual standard error: 2.805 on 98 degrees of freedom
Multiple R-squared: 0.9598,    Adjusted R-squared: 0.9594
F-statistic:  2340 on 1 and 98 DF,  p-value: < 2.2e-16
```

The difference between the two analyses is because of sampling variance, i.e., the fact that two different samples from the same population were taken. You may use `anova()` for an analysis of variance (ANOVA) table of this model:

```
anova(fit1)
anova(fit2)
```

Just for fun check the design matrices for the two models (they are the same):

```
model.matrix(fit1)
model.matrix(fit2)
```

7.1.4 Analysis Using WinBUGS

Here's the Bayesian analysis of the first data set (and don't forget to load package R2WinBUGS). We need to specify the model first. As an extra, we also compute the residuals.

```
# Define BUGS model
sink("ttest.txt")
cat("
model {

# Priors
 mu1 ~ dnorm (0,0.001)                # Precision = 1/variance
 delta ~ dnorm (0,0.001)                # Large variance = Small precision
 tau <- 1/ (sigma * sigma)
 sigma ~ dunif(0, 10)

# Likelihood
 for (i in 1:n) {
    y[i] ~ dnorm(mu[i], tau)
    mu[i] <- mu1 + delta *x[i]
    residual[i] <- y[i] – mu[i]          # Define residuals
 }

# Derived quantities: one of the greatest things about a Bayesian analysis
 mu2 <- mu1 + delta              # Difference in wingspan
}
",fill=TRUE)
sink()
```

Then, we provide the data, a function to generate inits, a list of parameters we want WinBUGS to keep track of, and specify the Markov chain Monte Carlo (MCMC) settings, after which, we use the function `bugs()` to run the analysis. Note the use of lognormal (instead of normal) random numbers as initial values for the positive-valued standard deviation `sigma`.

```
# Bundle data
win.data <- list("x", "y", "n")

# Inits function
inits <- function(){list(mu1=rnorm(1), delta=rnorm(1), sigma = rlnorm(1))}

# Parameters to estimate
params <- c("mu1","mu2", "delta", "sigma", "residual")

# MCMC settings
nc <- 3            # Number of chains
ni <- 1000         # Number of draws from posterior for each chain
nb <- 1            # Number of draws to discard as burn-in
nt <- 1            # Thinning rate

# Start Gibbs sampler
out <- bugs(data = win.data, inits = inits, parameters = params, model =
"ttest.txt", n.thin = nt, n.chains = nc, n.burnin = nb, n.iter = ni, debug = TRUE,
working.directory = getwd())

print(out, dig = 3)

> print(out, dig = 3)
Inference for Bugs model at "ttest.txt", fit using WinBUGS,
 3 chains, each with 1000 iterations (first 1 discarded)
 n.sims = 2997 iterations saved
```

```
                 mean    sd    2.5%      25%      50%      75%    97.5%  Rhat  n.eff
mu1           104.638 0.367 103.900 104.400 104.600 104.900 105.400 1.001  2100
mu2            78.070 0.448  77.200  77.770  78.080  78.370  78.950 1.001  3000
delta         -26.568 0.577 -27.680 -26.960 -26.560 -26.190 -25.410 1.001  3000
sigma           2.820 0.205   2.471   2.672   2.808   2.950   3.250 1.001  2400
residual[1]  -2.671 0.366  -3.411  -2.911  -2.669  -2.430  -1.937 1.001  2200
[ ...]
residual[100]  4.579 0.449   3.701   4.279   4.573   4.875   5.443 1.001  3000
deviance     489.489 2.579 486.600 487.700 488.800 490.600 496.300 1.001  3000

[ ...]

DIC info (using the rule, pD = Dbar-Dhat)
pD = 3.0 and DIC = 492.5
DIC is an estimate of expected predictive error (lower deviance is better).
>
```

Comparing the inference from WinBUGS with that using frequentist statistics, we see that the means estimates are almost identical, but that the residual standard deviation estimate is slightly larger in WinBUGS. This last point is general. In later chapters, we will often see that estimates of variances are greater in a Bayesian than in a frequentist analysis. Presumably, the difference will be greatest with smaller sample sizes. This is an indication of the approximate and asymptotic nature of frequentist inference that may differ from the exact inference under the Bayesian paradigm.

One of the nicest things about a Bayesian analysis is that parameters that are functions of primary parameters and their uncertainty (e.g., standard errors or credible intervals) can very easily be obtained using the MCMC posterior samples. Thus, in the above model code, the primary parameters are the female mean and the male–female difference, but we just added a line that computes the mean for males at every iteration, and we directly obtain samples from the posterior distributions of not only the female mean wingspan and the sex difference, but also directly of the mean male wingspan. In a frequentist mode of inference, this would require application of the delta method which is more complicated and also makes more assumptions. In the Bayesian analysis, estimation error is automatically propagated into functions of parameters.

Here are two further comments about model assessment. First, we see that the effective number of parameters estimated is 3.0, which is right because we are estimating one variance and two means. And second, before making an inference about the wingspan in this peregrine population, we should really check whether the model is adequate. Of course, the check of model adequacy is somewhat contrived because we use exclusively simulated and therefore, in a sense, perfect data sets. However, it is important to practice, so we will check the residuals here.

One of the first things to notice about the residuals, which is a little strange at first, is that each one has a distribution. This makes sense, because in a Bayesian analysis, every unknown has a posterior distribution representing our uncertainty about the magnitude of that unknown. Here, we plot the residuals against the order in which individuals were present in the data set and then produce a boxplot for male and female residuals to get a feel whether the distributions of residuals for the two groups are similar.

```
plot(1:100, out$mean$residual)
abline(h = 0)

boxplot(out$mean$residual ~ x, col = "grey", xlab = "Male", ylab = "Wingspan
residuals (cm)", las = 1)
abline(h = 0)
```

No violation of the model assumption of homoscedasticity is apparent from these residual checks.

7.2 t-TEST WITH UNEQUAL VARIANCES

7.2.1 Introduction

The previous analysis assumed that interindividual variation in wingspan is identical for male and female peregrines. This may well not be the case and it may be better to use a model that can accommodate possibly different variances. Our model then becomes as follows:

$$y_i = \alpha + \beta * x_i + \varepsilon_i$$
$$\varepsilon_i \sim Normal(0, \sigma_1^2) \text{ for } x_i = 0 \text{ (females)}$$
$$\varepsilon_i \sim Normal(0, \sigma_2^2) \text{ for } x_i = 1 \text{ (males)}$$

7.2.2 Data Generation

We first simulate data under the heterogeneous groups model (Fig. 7.2).

```
n1 <- 60                           # Number of females
n2 <- 40                           # Number of males
mu1 <- 105                         # Population mean for females
mu2 <- 77.5                        # Population mean for males
sigma1 <- 3                        # Population SD for females
sigma2 <- 2.5                      # Population SD for males

n <- n1+n2                         # Total sample size
y1 <- rnorm(n1, mu1, sigma1)       # Data for females
```

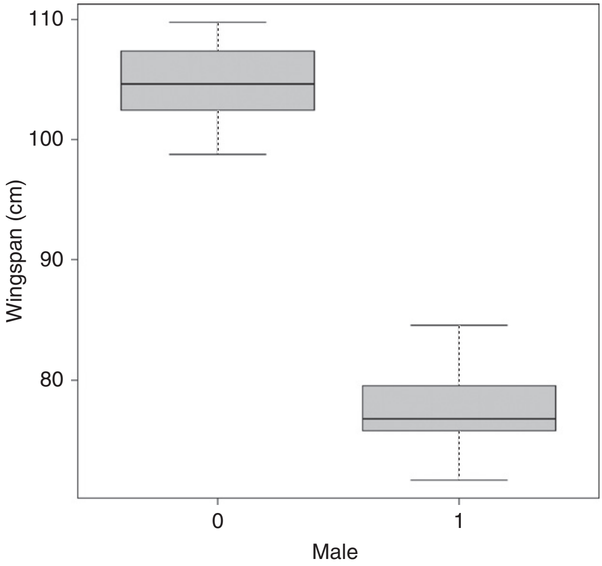

FIGURE 7.2 A boxplot of the generated data set on wingspan of female and male peregrines when the residuals depend on sex (0 - females, 1 - males).

```
y2 <- rnorm(n2, mu2, sigma2)        # Data for males
y <- c(y1, y2)                      # Aggregate both data sets
x <- rep(c(0,1), c(n1, n2))         # Indicator for male
boxplot(y ~ x, col = "grey", xlab = "Male", ylab = "Wingspan (cm)", las = 1)
```

7.2.3 Analysis Using R

A frequentist analysis, using the Welch test to allow for unequal variances, is easy. R defaults to heterogeneous variances when calling the t-test function:

```
t.test(y ~ x)
> t.test(y ~ x)

        Welch Two Sample t-test

data: y by x
t = 46.7144, df = 88.497, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 25.93319  28.23750
sample estimates:
mean in group 0  mean in group 1
      104.67641         77.59106
```

7.2.4 Analysis Using WinBUGS

Now here is the Bayesian analysis.

```
# Define BUGS model
sink("h.ttest.txt")
cat("
model {

# Priors
 mu1 ~ dnorm(0,0.001)
 mu2 ~ dnorm(0,0.001)
 tau1 <- 1 / ( sigma1 * sigma1)
 sigma1 ~ dunif(0, 1000)          # Note: Large var. = Small precision
 tau2 <- 1 / ( sigma2 * sigma2)
 sigma2 ~ dunif(0, 1000)

# Likelihood
 for (i in 1:n1) {
    y1[i] ~ dnorm(mu1, tau1)
 }

 for (i in 1:n2) {
    y2[i] ~ dnorm(mu2, tau2)
 }

# Derived quantities
 delta <- mu2 - mu1
}
",fill=TRUE)
sink()

# Bundle data
win.data <- list("y1", "y2", "n1", "n2")

# Inits function
inits <- function(){ list(mu1=rnorm(1), mu2=rnorm(1), sigma1 = rlnorm(1), sigma2 =
rlnorm(1))}

# Parameters to estimate
params <- c("mu1","mu2", "delta", "sigma1", "sigma2")

# MCMC settings
nc <- 3                # Number of chains
ni <- 2000             # Number of draws from posterior for each chain
nb <- 500              # Number of draws to discard as burn-in
nt <- 1                # Thinning rate
```

```
# Unleash Gibbs sampler
out <- bugs(data = win.data, inits = inits, parameters = params, model =
"h.ttest.txt", n.thin = nt, n.chains = nc, n.burnin = nb, n.iter = ni, debug =
TRUE)

print(out, dig = 3)
> print(out, dig = 3)
Inference for Bugs model at "h.ttest.txt", fit using WinBUGS,
 3 chains, each with 2000 iterations (first 500 discarded)
 n.sims = 4500 iterations saved
            mean    sd    2.5%     25%     50%     75%   97.5%  Rhat n.eff
mu1      104.667 0.400 103.900 104.400 104.700 104.900 105.400 1.001  4500
mu2       77.573 0.444  76.705  77.270  77.570  77.870  78.460 1.001  2700
delta    -27.093 0.592 -28.250 -27.490 -27.090 -26.700 -25.890 1.001  4500
sigma1     3.053 0.292   2.545   2.848   3.029   3.236   3.688 1.001  4500
sigma2     2.825 0.328   2.269   2.592   2.795   3.025   3.540 1.001  4500
deviance 497.784 2.882 494.200 495.600 497.200 499.300 504.952 1.001  4500

[ ... ]

DIC info (using the rule, pD = Dbar-Dhat)
pD = 3.9 and DIC = 501.7
DIC is an estimate of expected predictive error (lower deviance is better).
```

The complication of sex-dependent variances is trivial to deal with in the Bayesian framework. To formally test whether the two variances really differ, one could reparameterize the model such that the variance for one group is expressed as the variance of the other plus some constant to be estimated. Actually, this could also be done "outside" of WinBUGS in R by forming the difference, for each draw in the Markov chain, between `sigma1` and `sigma2`. If the credible interval for that parameter covers zero, then that would be taken as lack of evidence for different variances. This is an important idea; that derived variables with their full posterior uncertainty can also be computed outside of WinBUGS in R if posterior samples of all of their components are available. This is often easier than putting the added code into the WinBUGS model description.

7.3 SUMMARY AND A COMMENT ON THE MODELING OF VARIANCES

We have used WinBUGS to conduct the most widely used statistical test, the t-test. The version of that test with unequal variances is the only place in this book where we explicitly model the variance (except for the modeling of variances by variance components; see Chapters 9, 12, 16, 19, 20, and 21). This chapter shows that not only the mean but also the variance may be modeled. In classical statistics, variance modeling may be rather hard and fairly obscure in its application to an ecologist. In contrast in WinBUGS, the modeling of variances, e.g., as a function of some covariate, could be simply undertaken by use of a log link function; see Lee and Nelder (2006) and Lee et al. (2006) for (frequentist) examples of such models and Exercise 4 (below) for a Bayesian example. Variance modeling, either for the residuals or for random effects, may be required to adequately characterize the stochastic system components when inference is focusing on the mean structure. Alternatively, one may focus on a relation between an explanatory variable and a variance, for instance, to test a hypothesis that some conditions increase the variance in some trait.

EXERCISES

1. *Comparing variances*: See whether you can adapt the WinBUGS code directly to test for equal variances of wingspan. Try a solution using the quantities monitored in the previous analysis.
2. *Assumption violations*: Use simulation to study the effects of heterogeneous variances on the inference by a t-test that assumes homogeneous variances. Assemble data with different SD for males and females, but a common mean, and analyze them in WinBUGS assuming a common dispersion. See what kind of bias is introduced.
3. *Swiss hare data 1*: Use WinBUGS to fit a t-test to the *mean.density* in arable and grassland sites. Repeat that assuming unequal variances. Test for a difference of the variance.
4. *Swiss hare data 2*: Use WinBUGS to fit a t-test to the *mean.density* in arable and grassland sites and introduce a log-linear regression of the variance on *elevation*.