

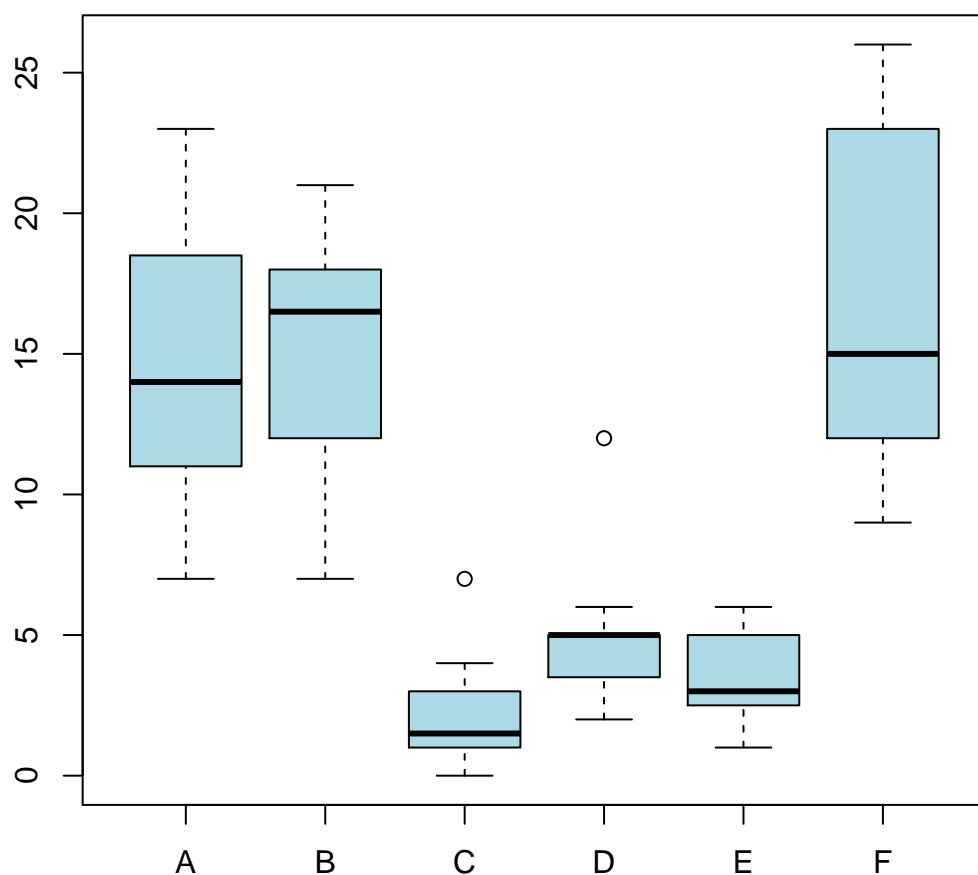
MAST20005/MAST90058: Week 9 Lab Solutions

1. (a) `anova(lm(count ~ spray, InsectSprays))`

```
## Analysis of Variance Table
##
## Response: count
##          Df Sum Sq Mean Sq F value    Pr(>F)
## spray      5 2668.8   533.77  34.702 < 2.2e-16 ***
## Residuals 66 1015.2    15.38
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

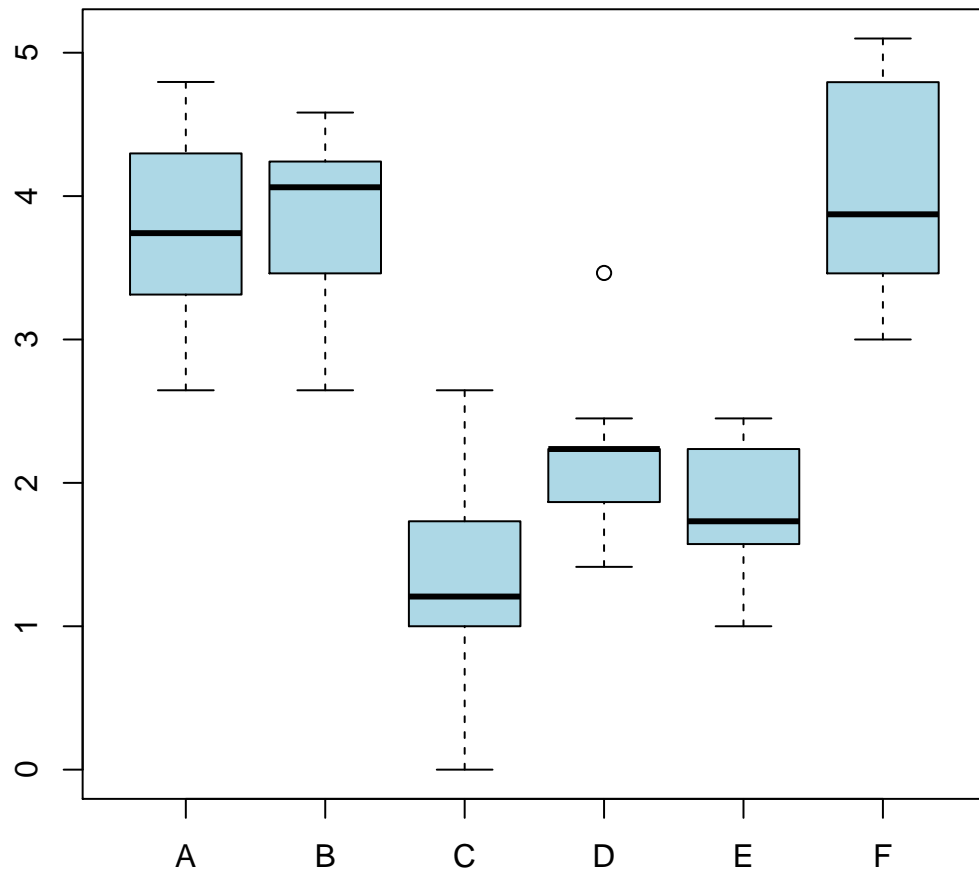
Very small p-value, can clearly reject the hypothesis that all sprays lead to the same average insect count.

(b) `boxplot(count ~ spray, InsectSprays, col = "lightblue")`



Sprays C–E seem to have smaller variance than the others, but in the ANOVA we assume all groups have the same variance.

(c) `boxplot(sqrt(count) ~ spray, InsectSprays, col = "lightblue")`



```
anova(lm(sqrt(count) ~ spray, InsectSprays))

## Analysis of Variance Table
##
## Response: sqrt(count)
##          Df Sum Sq Mean Sq F value    Pr(>F)
## spray      5  88.438  17.6876   44.799 < 2.2e-16 ***
## Residuals 66  26.058   0.3948
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The conclusions haven't changed but the data look more consistent with the assumption of equal variances.

2. (a) `anova(lm(len ~ factor(supp) + factor(dose), ToothGrowth))`

```
## Analysis of Variance Table
##
```

```
## Response: len
##           Df Sum Sq Mean Sq F value    Pr(>F)
## factor(supp)  1  205.35   205.35   14.017 0.0004293 ***
## factor(dose)  2 2426.43  1213.22   82.811 < 2.2e-16 ***
## Residuals    56  820.43    14.65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Very small p-values for both factors, which means we have strong evidence that both of them have an impact on average tooth growth.

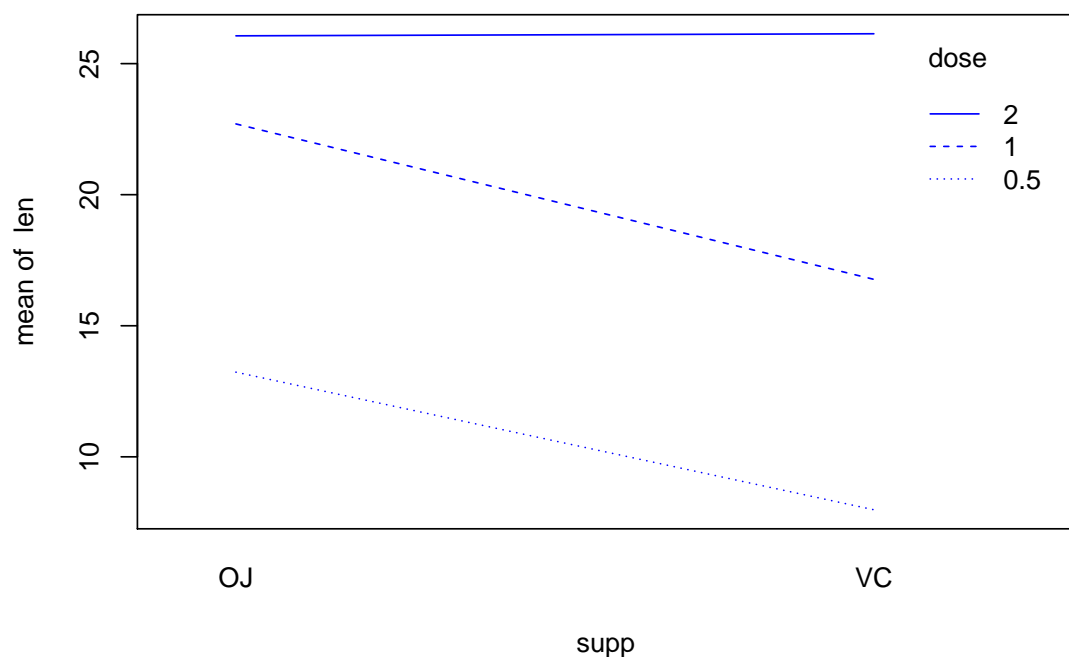
(Note: `supp` is already a factor variable so you will get the same result if you used `supp` instead of `factor(supp)` in the model formula.)

(b) `anova(lm(len ~ factor(supp) * factor(dose), ToothGrowth))`

```
## Analysis of Variance Table
##
## Response: len
##           Df Sum Sq Mean Sq F value    Pr(>F)
## factor(supp)      1  205.35   205.35   15.572 0.0002312 ***
## factor(dose)      2 2426.43  1213.22   92.000 < 2.2e-16 ***
## factor(supp):factor(dose)  2  108.32    54.16    4.107 0.0218603 *
## Residuals        54  712.11    13.19
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The interaction p-value of 0.02 indicates there is evidence of an interaction.

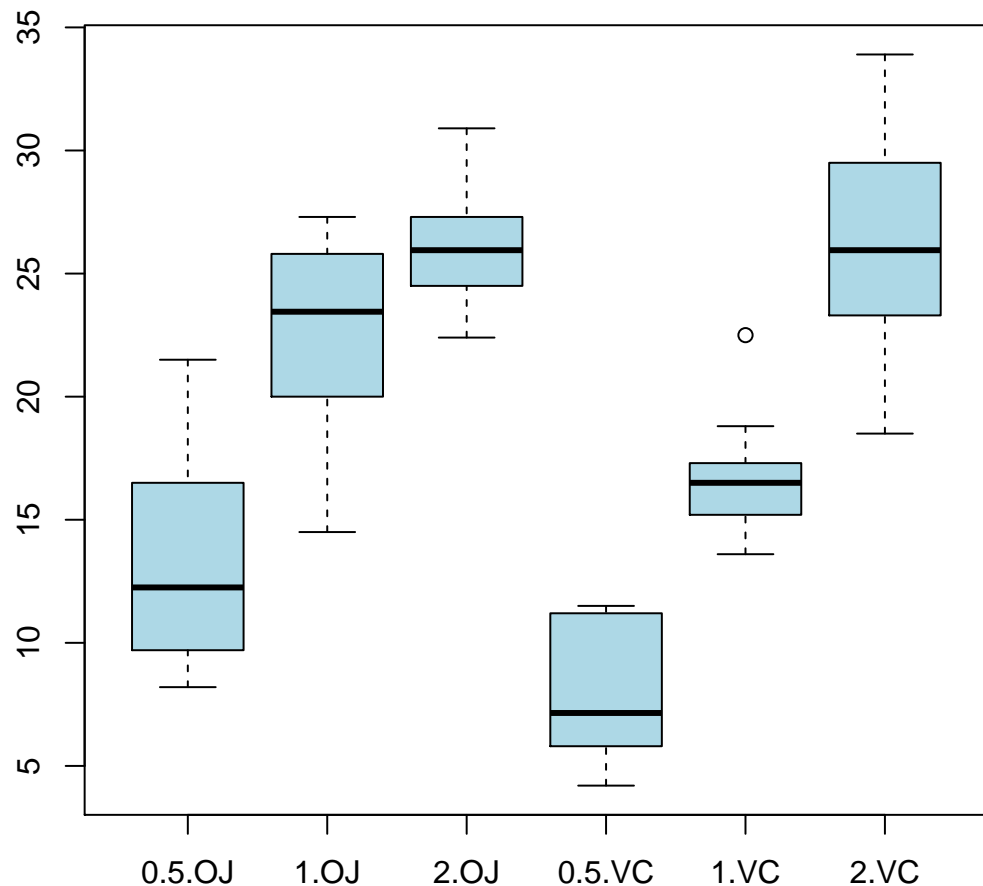
(c) `with(ToothGrowth, interaction.plot(supp, dose, len, col = "blue"))`



It looks like the difference between delivery methods (orange juice vs ascorbic acid) is only at the lower doses. At a higher dose the difference disappears.

Tip: boxplots would allow us to see more detail than in the interaction plot. You can use the interaction operator, `:` (or the factor crossing operator, `*`), as a convenient way to generate boxplots for each of the combinations of factors:

```
boxplot(len ~ dose:supp, ToothGrowth, col = "lightblue")
```

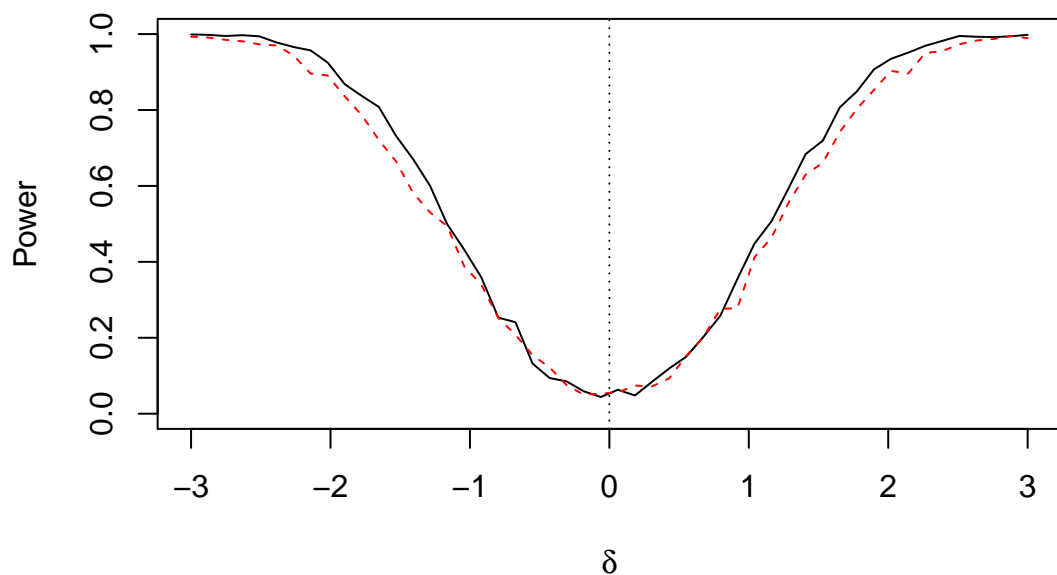


3. First, set up some variables common to both sets of simulations.

```
B <- 1000 # number of simulation runs
R <- 50    # number of power values
n1 <- 5    # sample sizes
n2 <- 10
delta.seq <- seq(-3, 3, length = R) # sequence of true differences
power1 <- numeric(R)                # initialize power vectors
power2 <- numeric(R)
```

(a) Do the first simulation:

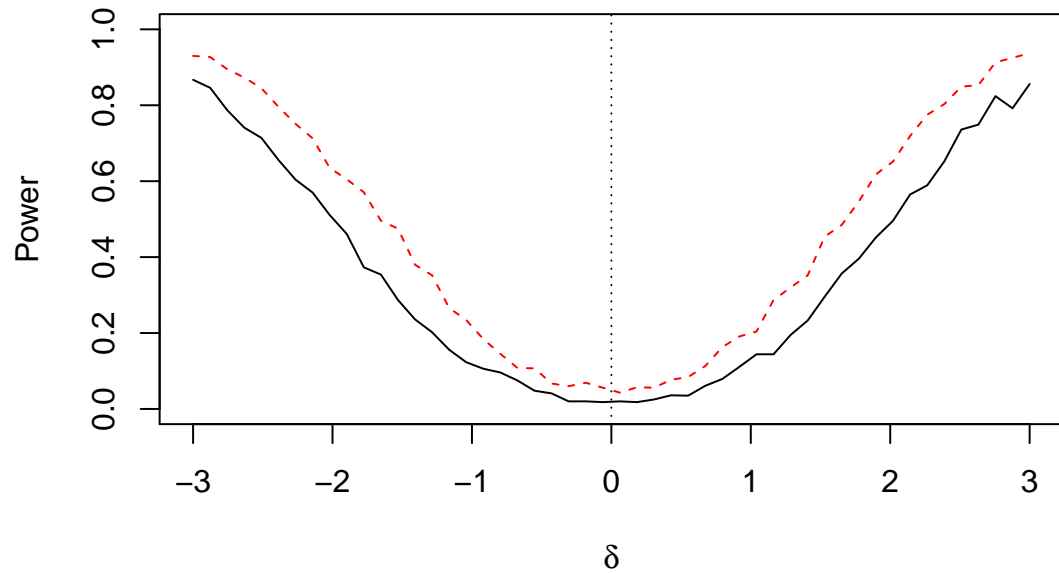
```
for (i in 1:R) {  
  delta <- delta.seq[i]  
  pvals1 <- replicate(B, t.test(rnorm(n1),  
                                rnorm(n2, delta),  
                                var.equal = TRUE)$p.value)  
  pvals2 <- replicate(B, t.test(rnorm(n1),  
                                rnorm(n2, delta))$p.value)  
  power1[i] <- mean(pvals1 < 0.05)  
  power2[i] <- mean(pvals2 < 0.05)  
}  
plot(delta.seq, power1, type = "l", ylim = c(0, 1),  
      ylab = "Power", xlab = expression(delta))  
lines(delta.seq, power2, lty = 2, col = 2)  
abline(v = 0, lty = 3)
```



(b) Do the second simulation:

```
for (i in 1:R) {  
  delta <- delta.seq[i]  
  pvals1 <- replicate(B, t.test(rnorm(n1),  
                                rnorm(n2, delta, 2),  
                                var.equal = TRUE)$p.value)  
  pvals2 <- replicate(B, t.test(rnorm(n1),  
                                rnorm(n2, delta, 2))$p.value)  
  power1[i] <- mean(pvals1 < 0.05)  
  power2[i] <- mean(pvals2 < 0.05)  
}  
plot(delta.seq, power1, type = "l", ylim = c(0, 1),  
      ylab = "Power", xlab = expression(delta))
```

```
lines(delta.seq, power2, lty = 2, col = 2)
abline(v = 0, lty = 3)
```



The two tests have similar power when the variances are the same, but Welch's t-test has noticeably extra power when the variances differ.