Linear Mixed-Effects Models

Imagine you are an Extension researcher who conducted an experiment last summer using 6 different insecticides to determine which spray was the most effect at decreasing insect damage to raspberries. The experiment was laid out as a randomized complete block design and sprays were administered according to label instructions; water was used as a control treatment. At the end of the season, you counted how many fruit per plot were damaged by insects and have recorded the data as a percent of the total number of raspberries harvested. Now it's time to analyze the data and draw conclusions about it. Use a linear mixed-effects model to analyze your data and create a publication-grade plot or table to present your results.
Note: This is simulated data, not from an actual experiment, so don't use it to make any decisions about raspberry maintenance.

Data-to-figure linear mixed-effects model procedure:
1. Read in data.
2. Look at the data and change necessary variables to factors.
3. Create the model.
4. Test the model assumptions.
5. Conduct the mean separation (**only proceed if treatment is significant**).
6. Summarize data.
7. Create a bar plot or table with compact letter display values.
8. Draw conclusions about the data.

1. Read in the data
```
library(readxl)
library(tidyverse)

library(lme4)

library(lmerTest)

library(emmeans)
library(multcomp)

library(car)

library(knitr)

berry <- read_xlsx("~/Documents/berry.xlsx")
```

2. Look at the data and change necessary variables to factors
```
summary(berry)

##    Treatment              trt           rep             damage
##   Length:28           Min.   :1   Min.   :1.00   Min.   : 2.00
```

```
##  Class :character    1st Qu.:2    1st Qu.:1.75    1st Qu.: 6.75
##  Mode  :character    Median :4    Median :2.50    Median :17.00
##                      Mean   :4    Mean   :2.50    Mean   :15.46
##                      3rd Qu.:6    3rd Qu.:3.25    3rd Qu.:21.50
##                      Max.   :7    Max.   :4.00    Max.   :31.00
```

```r
berry$rep <- as.factor(berry$rep)
berry$trt <- as.factor(berry$trt)
summary(berry)
```

```
##    Treatment          trt   rep         damage
##  Length:28           1:4   1:7   Min.   : 2.00
##  Class :character    2:4   2:7   1st Qu.: 6.75
##  Mode  :character    3:4   3:7   Median :17.00
##                      4:4   4:7   Mean   :15.46
##                      5:4         3rd Qu.:21.50
##                      6:4         Max.   :31.00
##                      7:4
```

```r
ggplot(berry, aes(trt, damage)) + geom_boxplot()
```
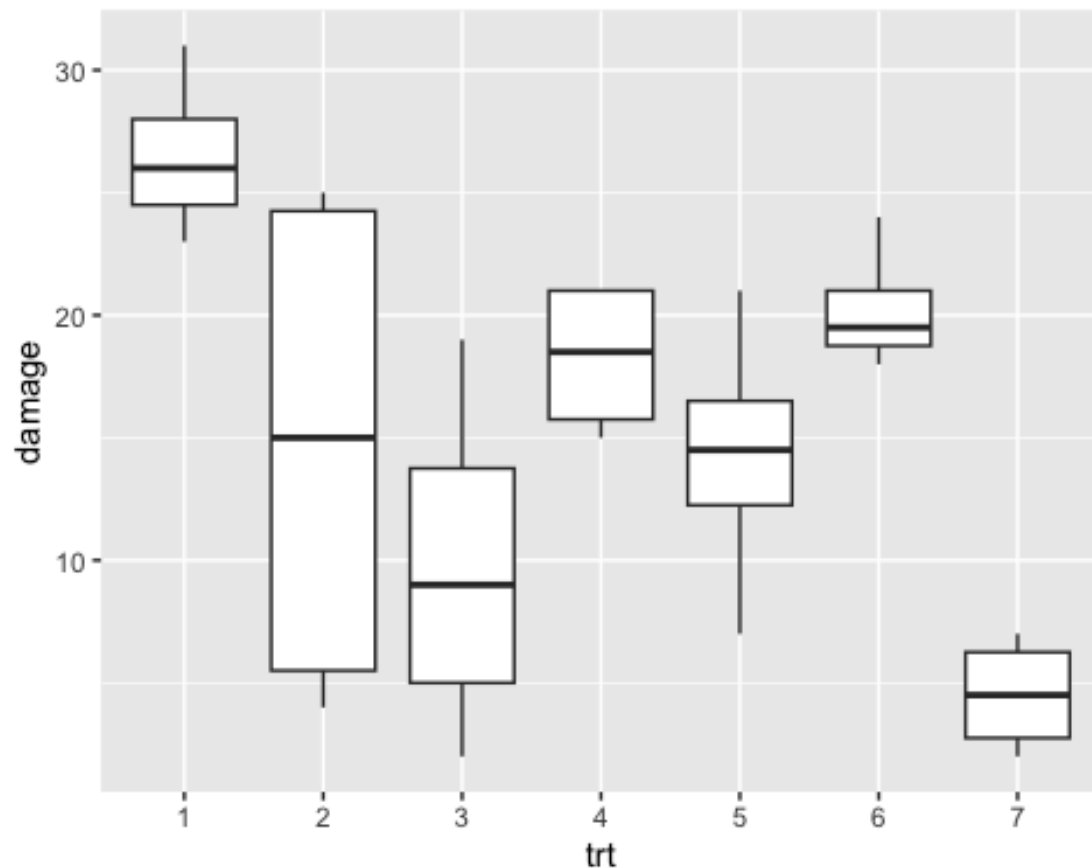

3. Create the model

```r
berry.lm <- lmer(damage ~ trt + (1|rep), data = berry)
anova(berry.lm)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##     Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## trt 1229.2  204.87     6    18  6.4253 0.0009405 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Treatment had an effect on berry damage
```

4. Test the model assumptions

```
standardRes <- residuals(berry.lm)/summary(berry.lm)$sigma
qqnorm(standardRes, main = "")
abline(a=0, b=1)
```
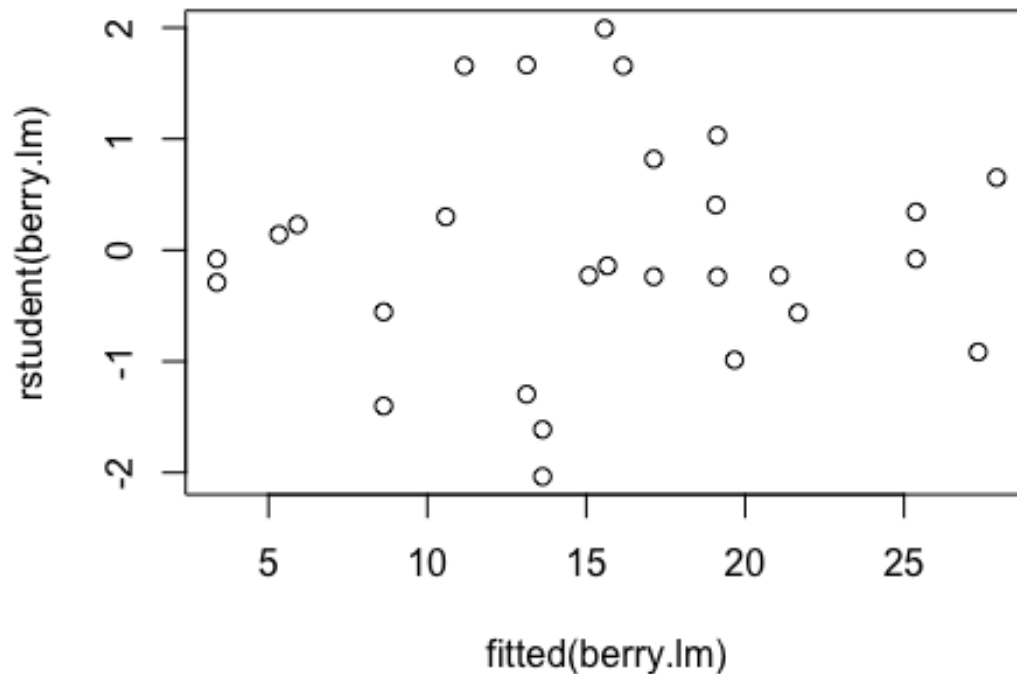


```
shapiro.test(standardRes)
```
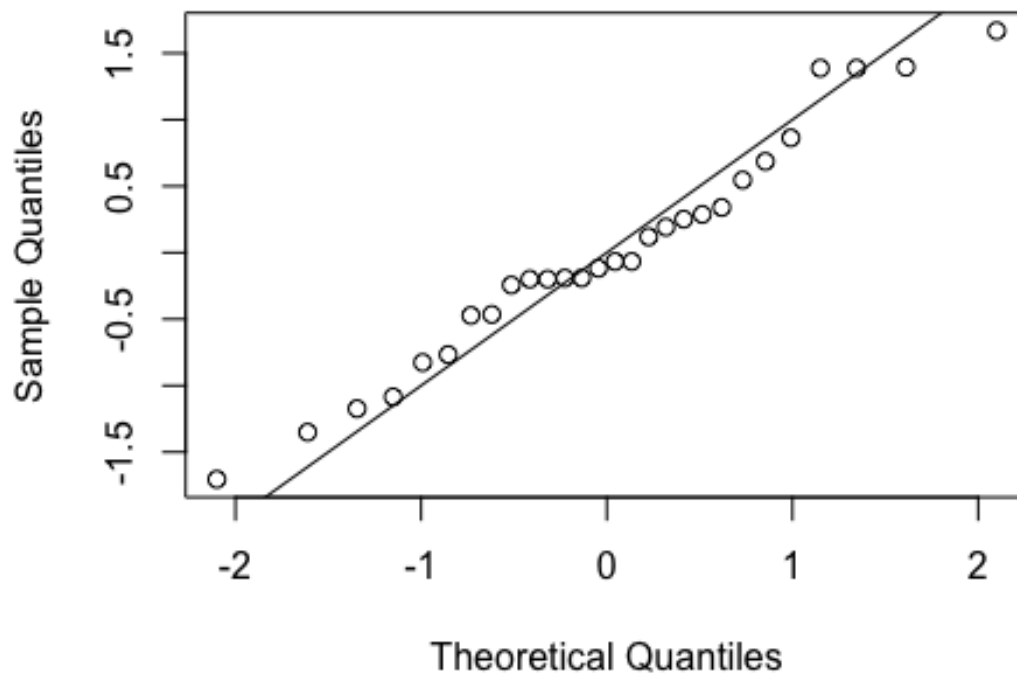
```
##
##  Shapiro-Wilk normality test
##
## data:  standardRes
## W = 0.97071, p-value = 0.5996
```

```
# Data is normally distributed
```

```
plot(fitted(berry.lm), rstudent(berry.lm))
```

```
leveneTest(residuals(berry.lm) ~ trt, data = berry)

## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value   Pr(>F)
## group  6   4.188 0.006357 **
##       21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# It looks like the variances are normal, but Levene's test says otherwise.
# I will use a natural log transformation to create a new damage variable to
remedy the violation of the variance test and reanalyze the data accordingly.

berry$damage.log <- log(berry$damage)
berry.lm2 <- lmer(damage.log ~ trt + (1|rep), data = berry)
anova(berry.lm)

## Type III Analysis of Variance Table with Satterthwaite's method
##      Sum Sq Mean Sq NumDF DenDF F value     Pr(>F)
## trt 1229.2  204.87     6    18  6.4253 0.0009405 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```
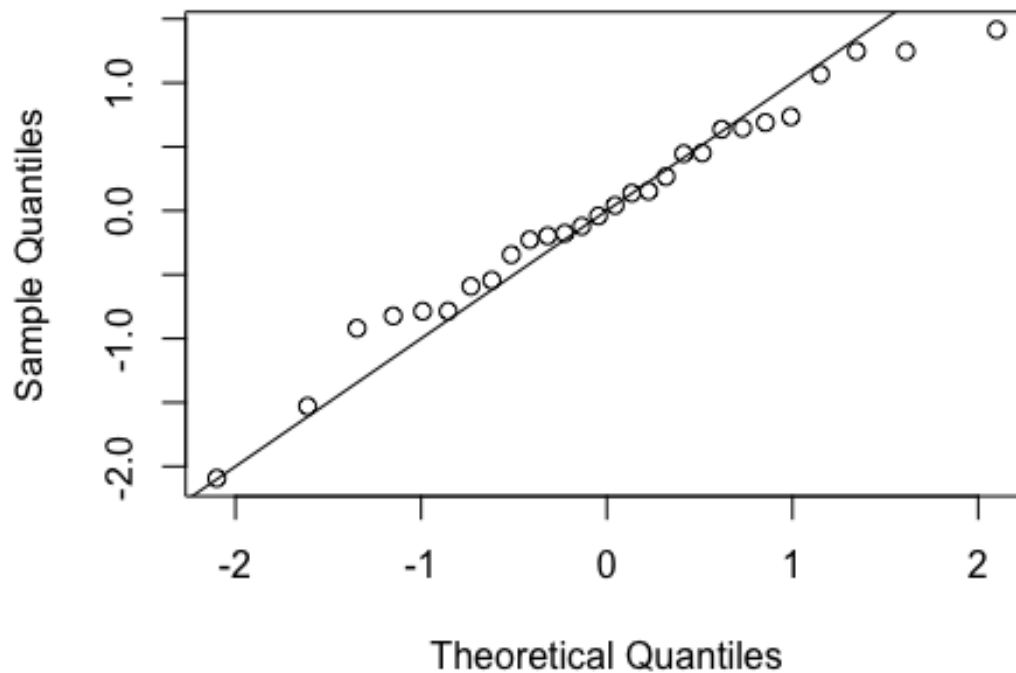
```
# Treatment is still significant

standardRes <- residuals(berry.lm2)/summary(berry.lm2)$sigma
qqnorm(standardRes, main = "")
abline(a=0, b=1)
```
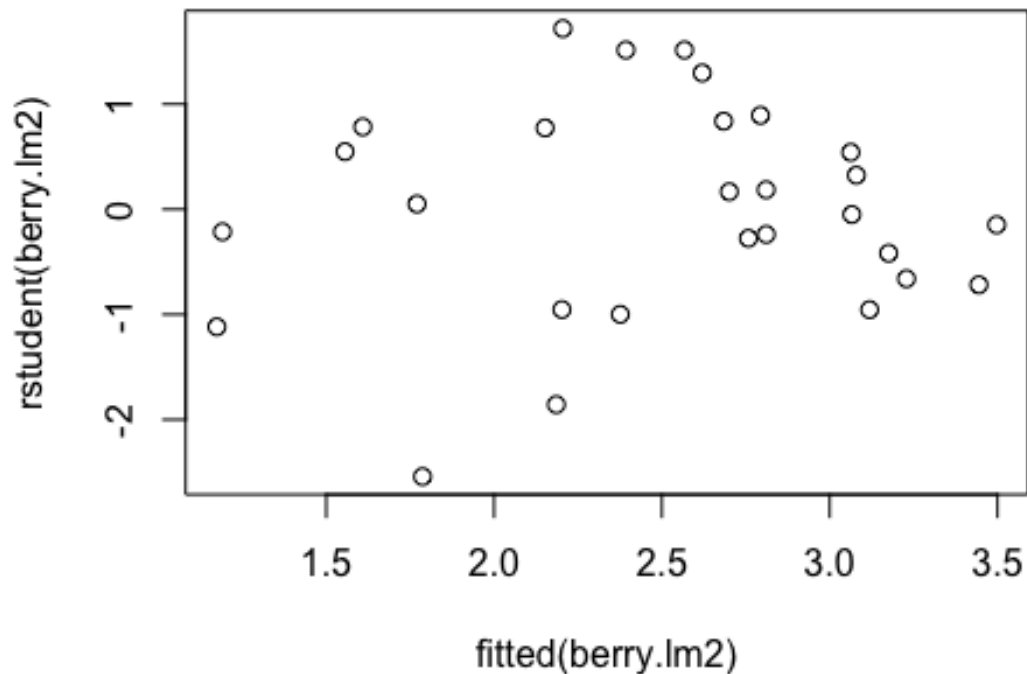


```
shapiro.test(standardRes)

##
##  Shapiro-Wilk normality test
##
## data:  standardRes
## W = 0.97628, p-value = 0.7539

# Data is normally distributed

plot(fitted(berry.lm2), rstudent(berry.lm2))
```

```
leveneTest(residuals(berry.lm2) ~ trt, data = berry)

## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  6  1.4199 0.2537
##       21
```

```
# The residual variances are now equal
```

5. Conduct the mean separation

```
berry.emmeans <- emmeans(berry.lm2, ~ trt)
cld(berry.emmeans)

##   trt emmean    SE df lower.CL upper.CL .group
##   7     1.38 0.297 16    0.753     2.01  1
##   3     1.98 0.297 16    1.349     2.61  12
##   2     2.39 0.297 16    1.764     3.02  123
##   5     2.58 0.297 16    1.955     3.21  123
##   4     2.89 0.297 16    2.263     3.52   23
##   6     3.00 0.297 16    2.372     3.63   23
##   1     3.27 0.297 16    2.641     3.90    3
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

```
## P value adjustment: tukey method for comparing a family of 7 estimates
## significance level used: alpha = 0.05
## NOTE: If two or more means share the same grouping symbol,
##        then we cannot show them to be different.
##        But we also did not show them to be the same.
```

```
# Now, I will create a vector of the compact letter display in treatment orde
r (1, 2, 3, etc.), so that the letters can be easily added to a bar graph cor
responding with each treatment
```

```
berry.cld <- c("a", "abc", "bc", "ab", "abc", "ab", "c")
```

6. Summarize the data

```
# Need to backtransform the data before presenting it
berry$damage.back <- exp(berry$damage.log)
head(berry)
```

```
## # A tibble: 6 × 6
##   Treatment trt   rep   damage damage.log damage.back
##   <chr>     <fct> <fct>  <dbl>      <dbl>       <dbl>
## 1 Water     1     1         23       3.14          23
## 2 Water     1     2         27       3.30          27
## 3 Water     1     3         31       3.43          31
## 4 Water     1     4         25       3.22          25
## 5 Spinosad  2     1         25       3.22          25
## 6 Spinosad  2     2          4       1.39           4
```

```
# The data is exactly the same as before transformation, but it is good pract
ice to do the calculations because the values can change a bit, especially if
you have decimals
```

```
berry.summary <- berry %>%
  group_by(trt, Treatment) %>%
  summarize(mean.damage = mean(damage.back))
```

```
## `summarise()` has grouped output by 'trt'. You can override using the `.gr
oups`
## argument.
```

```
berry.summary
```

```
## # A tibble: 7 × 3
## # Groups:   trt [7]
##   trt   Treatment    mean.damage
##   <fct> <chr>              <dbl>
## 1 1     Water               26.5
## 2 2     Spinosad            14.8
## 3 3     Carbaryl             9.75
## 4 4     Esfenvalerate       18.2
## 5 5     Malathion           14.2
```
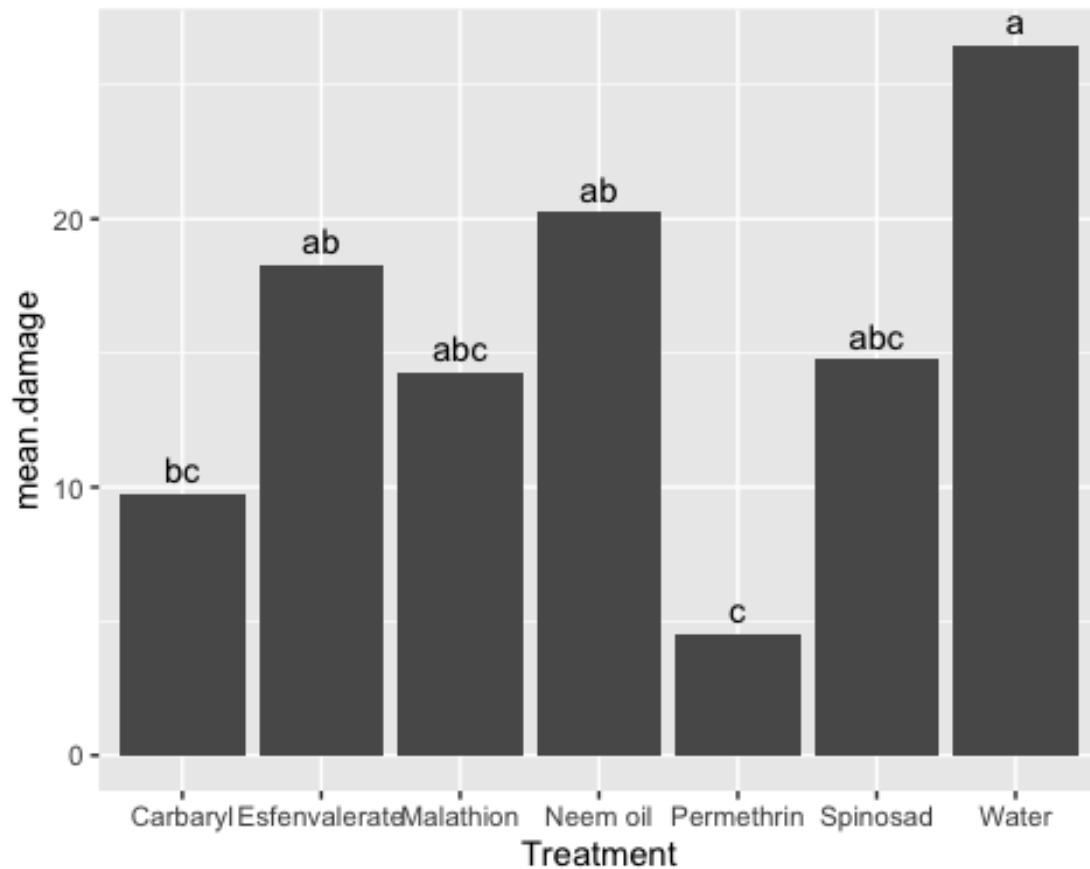
```
## 6 6      Neem oil           20.2
## 7 7      Permethrin          4.5
```

```
# Now I have the final dataset to create my figure
```

### 7. Create a bar plot or table with compact letter display values

```
# This is the bare minimum plot
ggplot(berry.summary, aes(Treatment, mean.damage)) +
  geom_bar(stat = "identity") +
  geom_text(label = berry.cld, vjust = c(-0.5))
```



```
# This would be my publication grade version of the plot
ggplot(berry.summary, aes(factor(Treatment, level = c("Water", "Neem oil", "E
sfenvalerate", "Spinosad", "Malathion", "Carbaryl", "Permethrin")), mean.dama
ge)) +
  geom_bar(stat = "identity") +
  geom_text(label = berry.cld, vjust = c(-0.5)) +
  theme_classic() +
  labs(x = "Insecticide treatment", y = "Amount of damaged fruit (%)")
```
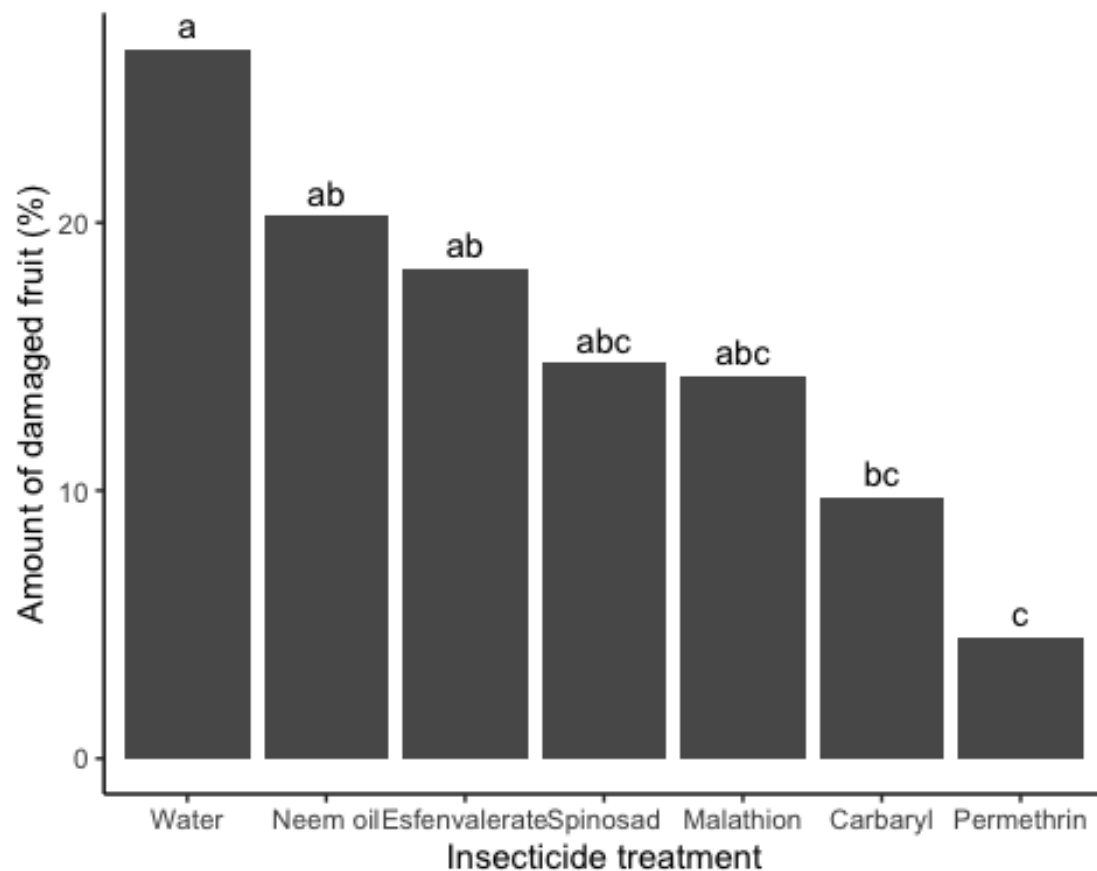
I would just create the table in Word, but this is one way to do it in R.

```
berry.table <- data.frame(berry.treatment = c("Water", "Neem oil", "Esfenvale
rate", "Spinosad", "Malathion", "Carbaryl", "Permethrin"), round(berry.summar
y$mean.damage,0), berry.cld = c("a", "ab", "ab", "abc", "abc", "bc", "c") )
kable(berry.table,col.names=c("Insecticde treatment", "Amount of damaged frui
t (%)", "Tukey's HSD"))
```

Insecticde treatment	Amount of damaged fruit (%)	Tukey's HSD
Water	26	a
Neem oil	15	ab
Esfenvalerate	10	ab
Spinosad	18	abc
Malathion	14	abc
Carbaryl	20	bc
Permethrin	4	c

8. Draw conclusions about the data

Based on this experiment, the most effective treatments for keeping insects from damaging raspberries were spinosad, malathion, carbaryl, and permethrin and I would recommend

growers to choose between those options when considering an insecticide treatment. Outside of effectiveness, factors to consider when choosing an insecticde may be cost, timing of application, number of application required, and human toxicity.