Lesson 11: Analysis of Variance

References

- Black, Chapter 11 Analysis of Variance and Design of Experiments (pp. 370-410)
- Kabakoff, Chapter 9 Analysis of Variance (pp. 212-221)
- Davies, Chapter 9.3 One-way ANOVA (pp. 218-223), Chapter 19 Analysis of Variance (pp. 435-449)
- Stowell, Chapter 6 Tabular Data (pp. 73-86), Chapter 10 Hypothesis Testing (pp. 144-146, 158)

Data set: tableware.csv

Variable Names:

1. TYPE: bowl, cass, dish, tray, plate
2. BOWL: Bowl (1) or not (0)
3. CASS: Casserole (1) or not (0)
4. DISH: Dish (1) or not (0)
5. TRAY: Tray (1) or not (0)
6. DIAM: Diameter of item, or equivalent (inches)
7. TIME: Grinding and polishing time (minutes)
8. PRICE: Retail price ($)
9. RATE: Retail price divided by Time ($ per minute)

Exercises:

1) Use the tableware.csv data to test the hypothesis that the mean RATE for the five levels of TYPE are equal. Test the hypothesis using a 0.05 significance level. Plot means and confidence intervals of RATE for each level of TYPE (Use the example given in Davies Chapter 9.3 One-way ANOVA (pp. 218-223)).

```r
# Read the comma-delimited text file creating a data frame object in R,
# then examine its structure:
tableware <- read.csv("tableware.csv")
str(tableware)
```

```
## 'data.frame':    59 obs. of  9 variables:
##  $ TYPE : Factor w/ 5 levels "bowl","cass",..: 2 2 2 1 3 2 5 5 3 3 ...
##  $ BOWL : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ CASS : int  1 1 1 0 0 1 0 0 0 0 ...
##  $ DISH : int  0 0 0 0 1 0 0 0 1 1 ...
##  $ TRAY : int  0 0 0 0 0 0 1 1 0 0 ...
##  $ DIAM : num  10.7 14 9 8 10 10.5 16 15 6.5 5 ...
##  $ TIME : num  47.6 63.1 58.8 34.9 55.5 ...
##  $ PRICE: num  144 169 105 69 134 129 155 99 38.5 36.5 ...
##  $ RATE : num  3.02 2.68 1.79 1.98 2.41 2.99 2.83 2.24 2.21 1.73 ...
```

```r
RATE_anova <- aov(RATE ~ TYPE - 1, data = tableware)
RATE_lm <- lm(RATE ~ TYPE - 1, data = tableware)

summary(RATE_anova)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## TYPE       5  317.4   63.48   219.9 <2e-16 ***
## Residuals 54   15.6    0.29
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

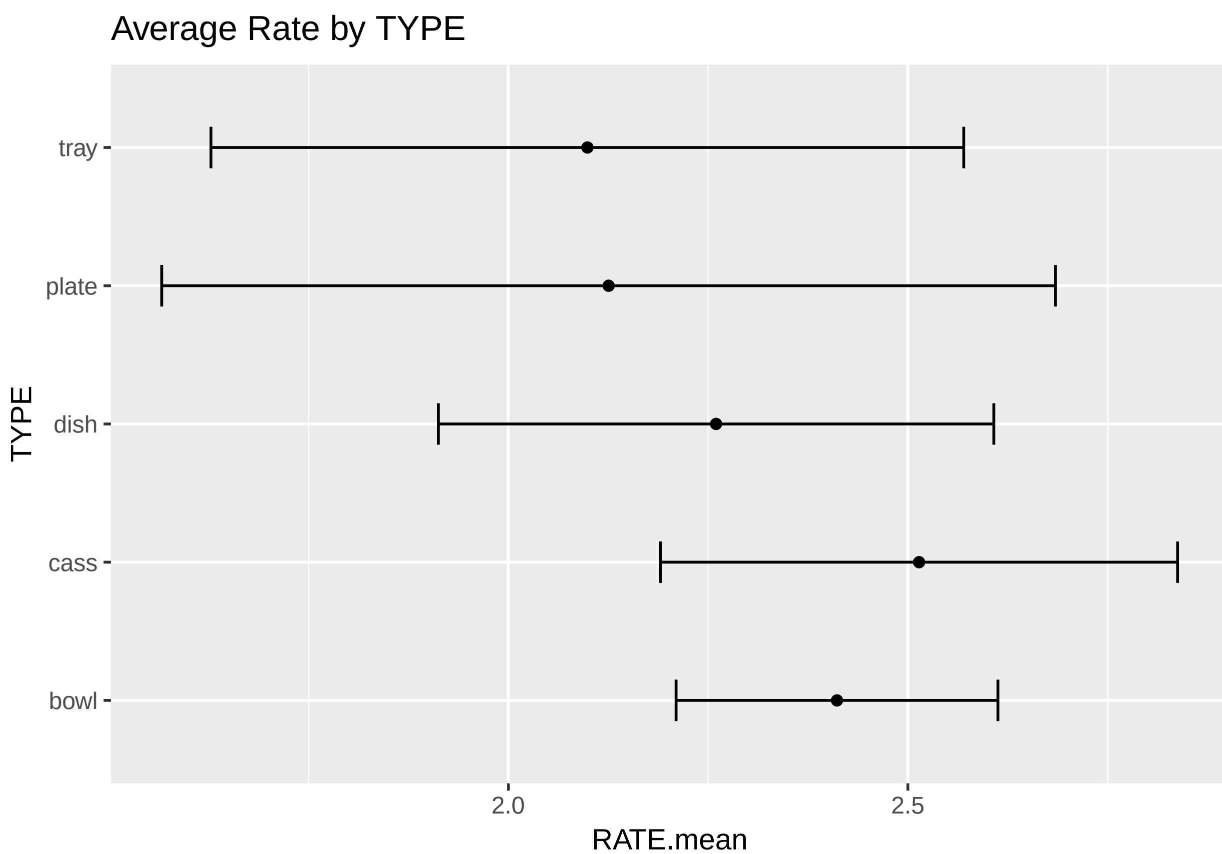
```r
summary(RATE_lm)
```

```
##
## Call:
## lm(formula = RATE ~ TYPE - 1, data = tableware)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2690 -0.3726  0.0800  0.4335  0.9287
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## TYPEbowl    2.4113     0.1120   21.52  < 2e-16 ***
## TYPEcass    2.5140     0.1699   14.80  < 2e-16 ***
## TYPEdish    2.2600     0.2031   11.13 1.36e-15 ***
## TYPEplate   2.1256     0.1791   11.87  < 2e-16 ***
## TYPEtray    2.0990     0.1699   12.35  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5373 on 54 degrees of freedom
## Multiple R-squared:  0.9532, Adjusted R-squared:  0.9488
## F-statistic: 219.9 on 5 and 54 DF,  p-value: < 2.2e-16
```

```r
# ddply(), package:  plyr, can hasten generation of our per TYPE confidence
# intervals and let us output a handy table
library(plyr)
RATEbyType <- ddply(tableware, "TYPE", summarize,
                    RATE.mean=mean(RATE), RATE.sd=sd(RATE),
                    Length=NROW(RATE),
                    tfrac=qt(p=.975, df=Length-1),
                    Lower=RATE.mean - tfrac*RATE.sd/sqrt(Length),
                    Upper=RATE.mean + tfrac*RATE.sd/sqrt(Length)
                    )
RATEbyType
```

```
##    TYPE RATE.mean   RATE.sd Length    tfrac    Lower    Upper
## 1  bowl  2.411304 0.4655720     23 2.073873 2.209976 2.612633
## 2  cass  2.514000 0.4521602     10 2.262157 2.190544 2.837456
## 3  dish  2.260000 0.3757659      7 2.446912 1.912475 2.607525
## 4 plate  2.125556 0.7274461      9 2.306004 1.566391 2.684720
## 5  tray  2.099000 0.6583219     10 2.262157 1.628065 2.569935
```

```r
# We can now use Lower and Upper CI values in a plot
library(ggplot2)
ggplot(RATEbyType, aes(x=RATE.mean, y=TYPE))+geom_point()+
  geom_errorbarh(aes(xmin=Lower, xmax=Upper), height=.3)+
  ggtitle("Average Rate by TYPE")
```




```r
# This shows the results without a -1.  The means are not different from each other.
RATE_anova <- aov(RATE ~ TYPE, data = tableware)
RATE_lm <- lm(RATE ~ TYPE, data = tableware)

summary(RATE_anova)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## TYPE         4   1.42  0.3550    1.23  0.309
## Residuals   54  15.59  0.2887
```

```r
summary(RATE_lm)
```

```
##
## Call:
## lm(formula = RATE ~ TYPE, data = tableware)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2690 -0.3726  0.0800  0.4335  0.9287
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.4113     0.1120  21.522   <2e-16 ***
## TYPEcass      0.1027     0.2035   0.505    0.616
## TYPEdish     -0.1513     0.2319  -0.652    0.517
## TYPEplate    -0.2857     0.2113  -1.353    0.182
## TYPEtray     -0.3123     0.2035  -1.534    0.131
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5373 on 54 degrees of freedom
## Multiple R-squared:  0.08348,    Adjusted R-squared:  0.01559
## F-statistic:  1.23 on 4 and 54 DF,  p-value: 0.3092
```

```r
# The reason the aov() summary output differs depending on whether -1 is used or
# not is due to different questions being asked.  If -1 is used, the F-test is
# evaluating if the group means are different from zero.  If -1 is not used, the
# F-test is considering if they differ among themselves.  As TukeyHSD shows,
# the rates do not differ.  We could simplify the model to using a single rate
# across the board.  The display of confidence intervals which overlap says this.
```

2) Use the tableware.csv data to test the hypothesis that the mean PRICE for the five levels of TYPE are equal. Test the hypothesis using a 0.05 significance level. Plot means and confidence intervals of PRICE for each level of TYPE (Use the example given in Davies Chapter 9.3 One-way ANOVA (pp. 218-223)).

```r
# What follows is an alternative way to code the problem:
my_price_model <- {PRICE ~ TYPE}
my_price_model_fit <- lm(my_price_model, data = tableware)
print(summary(my_price_model_fit))
```

```
##
## Call:
## lm(formula = my_price_model, data = tableware)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -76.950 -26.362  -2.333  26.109  90.050
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   67.391      7.575   8.897 3.62e-12 ***
## TYPEcass      59.529     13.760   4.326 6.59e-05 ***
## TYPEdish      12.752     15.681   0.813   0.4197
## TYPEplate    -15.558     14.283  -1.089   0.2809
## TYPEtray      31.559     13.760   2.294   0.0257 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.33 on 54 degrees of freedom
## Multiple R-squared:  0.3367, Adjusted R-squared:  0.2876
## F-statistic: 6.853 on 4 and 54 DF,  p-value: 0.0001548
```

```r
print(anova(my_price_model_fit))
```

```
## Analysis of Variance Table
##
## Response: PRICE
##           Df Sum Sq Mean Sq F value    Pr(>F)
## TYPE       4  36174  9043.5  6.8532 0.0001548 ***
## Residuals 54  71258  1319.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Confidence intervals for the coefficients in the regression model.
print(confint(my_price_model_fit, level = 0.95))
```

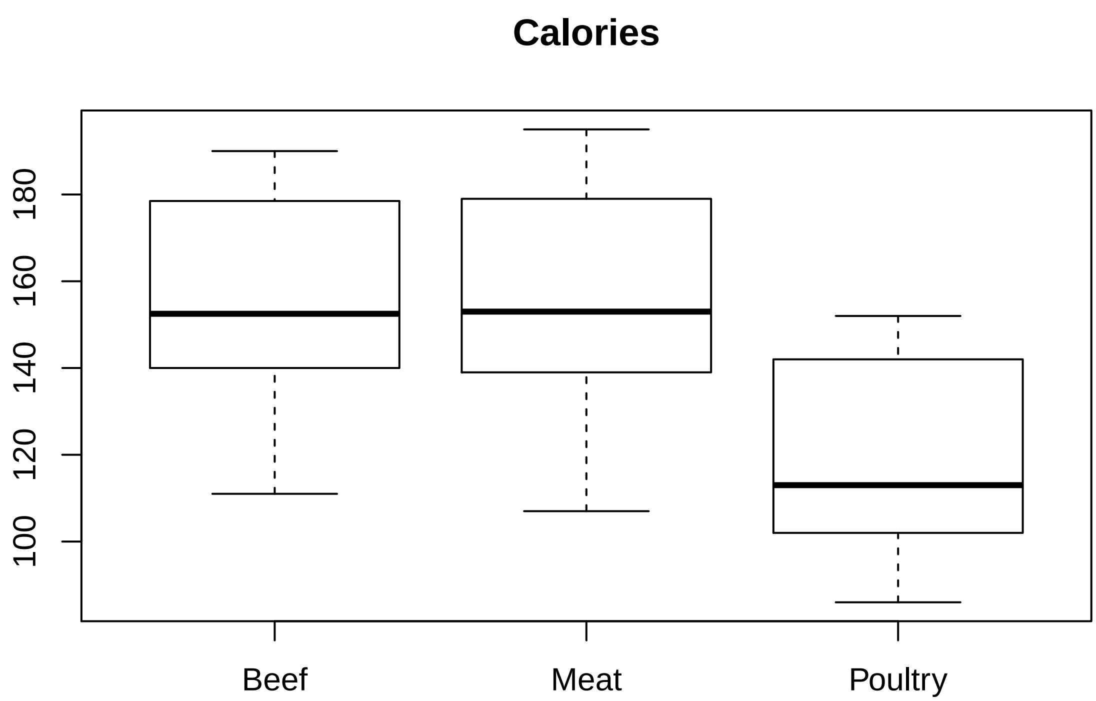
```
##                  2.5 %   97.5 %
## (Intercept)  52.205239 82.57737
## TYPEcass     31.941838 87.11555
## TYPEdish    -18.686590 44.18970
## TYPEplate   -44.193091 13.07715
## TYPEtray      3.971838 59.14555
```

3) Use the hot_dogs.csv data. Perform a one-way AOV by Type on Calories and also Sodium (Use the example given in Davies Chapter 9.3 One-way ANOVA (pp. 218-223)). Use Tukey's Honest Significant Difference Test if the F-test is significant. Generate boxplots.

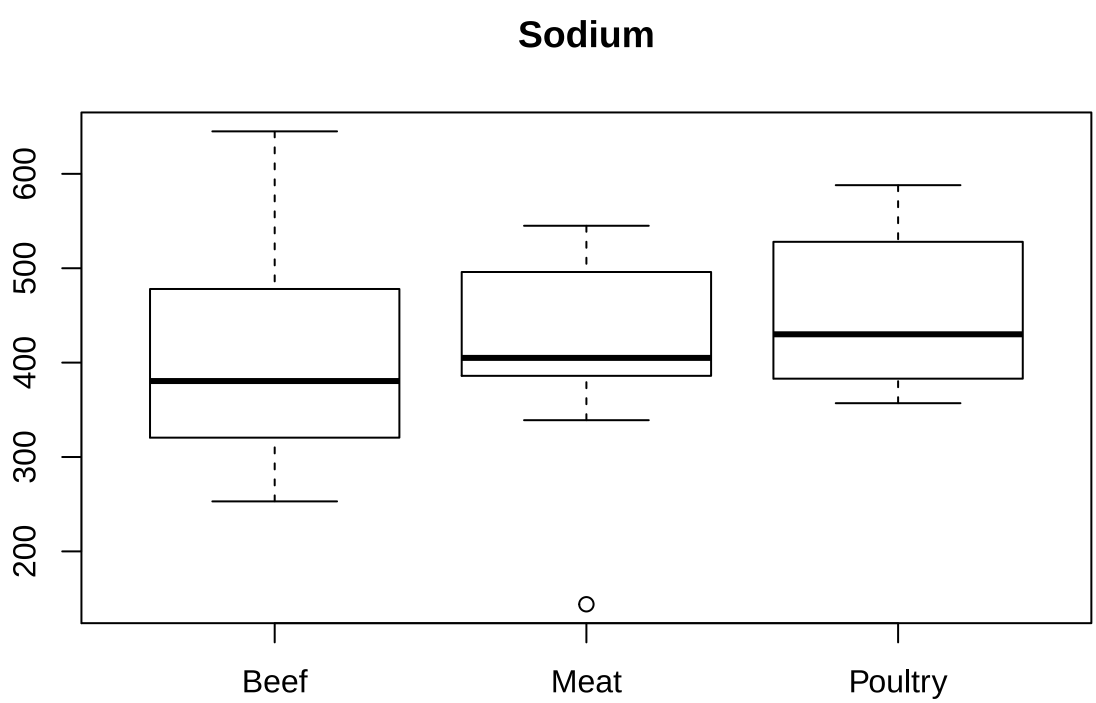
```r
# Read the comma-delimited text file creating a data frame object in R,
# then examine its structure:
hotdogs <- read.csv("hot_dogs.csv")
str(hotdogs)
```

```
## 'data.frame':    54 obs. of  3 variables:
##  $ Type    : Factor w/ 3 levels "Beef","Meat",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Calories: int  186 181 176 149 184 190 158 139 175 148 ...
##  $ Sodium  : int  495 477 425 322 482 587 370 322 479 375 ...
```

```r
# Create boxplots using Calories and Sodium, by Type.
with(hotdogs, boxplot(Calories ~ Type, main = "Calories"))
```



```r
with(hotdogs, boxplot(Sodium ~ Type, main = "Sodium"))
```




```r
# Perform one-way AOV, Calories
calories.anova <- aov(Calories ~ Type, data = hotdogs)
summary(calories.anova)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)    
## Type         2  17692    8846   16.07 3.86e-06 ***
## Residuals   51  28067     550                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Perform one-way AOV, Sodium
sodium.anova <- aov(Sodium ~ Type, data = hotdogs)
summary(sodium.anova)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Type         2  31739   15869   1.778  0.179
## Residuals   51 455249    8926
```

```r
# Perform Tukey's Honest Significant Difference Test
TukeyHSD(calories.anova, conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Calories ~ Type, data = hotdogs)
##
```

```
## $Type
##                    diff       lwr       upr     p adj
## Meat-Beef      1.855882 -16.82550  20.53726 0.9688129
## Poultry-Beef -38.085294 -56.76667 -19.40391 0.0000277
## Poultry-Meat -39.941176 -59.36515 -20.51720 0.0000239
```

```
TukeyHSD(sodium.anova, conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Sodium ~ Type, data = hotdogs)
##
## $Type
##                  diff       lwr      upr     p adj
## Meat-Beef    17.37941 -57.85808  92.6169 0.8430421
## Poultry-Beef 57.85000 -17.38749 133.0875 0.1620137
## Poultry-Meat 40.47059 -37.75763 118.6988 0.4304240
```

Comparison of aov() and lm() on tableware data

The below does not include questions or code prompts. It is meant to further compare ANOVA and linear regression and the use or exclusion of an intercept term; i.e. "- 1."

```
tableware <- read.csv("tableware.csv", sep = ",")
str(tableware)
```

```
## 'data.frame':    59 obs. of  9 variables:
##  $ TYPE : Factor w/ 5 levels "bowl","cass",..: 2 2 2 1 3 2 5 5 3 3 ...
##  $ BOWL : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ CASS : int  1 1 1 0 0 1 0 0 0 0 ...
##  $ DISH : int  0 0 0 0 1 0 0 0 1 1 ...
##  $ TRAY : int  0 0 0 0 0 0 1 1 0 0 ...
##  $ DIAM : num  10.7 14 9 8 10 10.5 16 15 6.5 5 ...
##  $ TIME : num  47.6 63.1 58.8 34.9 55.5 ...
##  $ PRICE: num  144 169 105 69 134 129 155 99 38.5 36.5 ...
##  $ RATE : num  3.02 2.68 1.79 1.98 2.41 2.99 2.83 2.24 2.21 1.73 ...
```

```
# First, we establish group means for comparison.

with(tableware, aggregate(RATE, by = list(TYPE), mean))
```

```
##   Group.1        x
## 1    bowl 2.411304
## 2    cass 2.514000
## 3    dish 2.260000
## 4   plate 2.125556
## 5    tray 2.099000
```

```
# Now, compare the group means to aov() coefficients.  They match.

RATE_anova <- aov(RATE ~ TYPE - 1, data = tableware)
summary(RATE_anova)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## TYPE       5  317.4   63.48   219.9 <2e-16 ***
## Residuals 54   15.6    0.29
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
RATE_anova$coefficients
```

```
##  TYPEbowl  TYPEcass  TYPEdish TYPEplate  TYPEtray
##  2.411304  2.514000  2.260000  2.125556  2.099000
```

```r
# Using aov() in this way estimates group means directly and with a
# pooled estimate of the within-group variance.

# Now compare to aov() without -1.

RATE_anova2 <- aov(RATE ~ TYPE, data = tableware)
summary(RATE_anova2)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## TYPE         4   1.42  0.3550    1.23  0.309
## Residuals   54  15.59  0.2887
```

```r
RATE_anova2$coefficients
```

```
## (Intercept)    TYPEcass    TYPEdish   TYPEplate    TYPEtray
##   2.4113043   0.1026957  -0.1513043  -0.2857488  -0.3123043
```

```r
# The intercept is the coefficient for bowl.  The coefficients for the other levels
# are determined by adding coefficients.  The results match the prior means.

RATE_anova2$coefficients[1]  #bowl
```

```
## (Intercept)
##    2.411304
```

```r
RATE_anova2$coefficients[1]+RATE_anova2$coefficients[2]  #cass
```

```
## (Intercept)
##       2.514
```

```r
RATE_anova2$coefficients[1]+RATE_anova2$coefficients[3]  #dish
```

```
## (Intercept)
##        2.26
```

```r
RATE_anova2$coefficients[1]+RATE_anova2$coefficients[4]  #plate
```

```
## (Intercept)
##    2.125556
```

```r
RATE_anova2$coefficients[1]+RATE_anova2$coefficients[5]  #tray
```

```
## (Intercept)
##       2.099
```

```r
# Now, consider lm().  aov() is based on a linear model.  The model submitted to
# lm() is the same linear model.  Both use least square for coefficient estimation.
# The coefficient estimates should match.
```

```r
RATE_lm <- lm(RATE ~ TYPE - 1, data = tableware)
summary(RATE_lm)
```

```
##
## Call:
## lm(formula = RATE ~ TYPE - 1, data = tableware)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2690 -0.3726  0.0800  0.4335  0.9287
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## TYPEbowl    2.4113     0.1120   21.52  < 2e-16 ***
## TYPEcass    2.5140     0.1699   14.80  < 2e-16 ***
## TYPEdish    2.2600     0.2031   11.13 1.36e-15 ***
## TYPEplate   2.1256     0.1791   11.87  < 2e-16 ***
## TYPEtray    2.0990     0.1699   12.35  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5373 on 54 degrees of freedom
## Multiple R-squared:  0.9532, Adjusted R-squared:  0.9488
## F-statistic: 219.9 on 5 and 54 DF,  p-value: < 2.2e-16
```

```r
RATE_lm$coefficients  # These coefficients match RATE_anova$coefficients with -1
```

```
##  TYPEbowl  TYPEcass  TYPEdish TYPEplate  TYPEtray
##  2.411304  2.514000  2.260000  2.125556  2.099000
```

```r
RATE_lm <- lm(RATE ~ TYPE , data = tableware)
summary(RATE_lm)
```

```
##
## Call:
## lm(formula = RATE ~ TYPE, data = tableware)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2690 -0.3726  0.0800  0.4335  0.9287
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.4113     0.1120  21.522   <2e-16 ***
## TYPEcass      0.1027     0.2035   0.505    0.616
## TYPEdish     -0.1513     0.2319  -0.652    0.517
## TYPEplate    -0.2857     0.2113  -1.353    0.182
## TYPEtray     -0.3123     0.2035  -1.534    0.131
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5373 on 54 degrees of freedom
## Multiple R-squared:  0.08348,    Adjusted R-squared:  0.01559
## F-statistic:  1.23 on 4 and 54 DF,  p-value: 0.3092
```

```r
RATE_lm$coefficients  # These coefficients match RATE_anova$coefficients without -1
```

```
## (Intercept)    TYPEcass    TYPEdish   TYPEplate    TYPEtray
##   2.4113043   0.1026957  -0.1513043  -0.2857488  -0.3123043
```

```r
# The output from the two approaches does differ, but the results are the same as
# far as the coefficients are concerned.  Now consider the aov() tables and TukeyHSD.
# We find the same conclusion.  The RATES between the different levels of TYPE are
# not statistically significant when pairwise comparisons are considered.  TukeyHSD
# can not be used with lm().

RATE_anova <- aov(RATE ~ TYPE - 1, data = tableware)
summary(RATE_anova)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## TYPE       5  317.4   63.48   219.9 <2e-16 ***
## Residuals 54   15.6    0.29
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(RATE_anova)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = RATE ~ TYPE - 1, data = tableware)
##
## $TYPE
##                   diff        lwr       upr     p adj
## cass-bowl   0.10269565 -0.4716667 0.6770580 0.9865727
## dish-bowl  -0.15130435 -0.8058509 0.5032422 0.9654709
## plate-bowl -0.28574879 -0.8819361 0.3104385 0.6599774
## tray-bowl  -0.31230435 -0.8866667 0.2620580 0.5451195
## dish-cass  -0.25400000 -1.0012541 0.4932541 0.8719822
## plate-cass -0.38844444 -1.0851486 0.3082598 0.5205458
## tray-cass  -0.41500000 -1.0931221 0.2631221 0.4264365
## plate-dish -0.13444444 -0.8986014 0.6297126 0.9873619
## tray-dish  -0.16100000 -0.9082541 0.5862541 0.9732292
## tray-plate -0.02655556 -0.7232598 0.6701486 0.9999687
```

```r
RATE_anova2 <- aov(RATE ~ TYPE, data = tableware)
summary(RATE_anova2)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## TYPE         4   1.42  0.3550    1.23  0.309
## Residuals   54  15.59  0.2887
```

```r
TukeyHSD(RATE_anova2)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = RATE ~ TYPE, data = tableware)
##
## $TYPE
##                   diff        lwr       upr     p adj
## cass-bowl   0.10269565 -0.4716667 0.6770580 0.9865727
```

```
## dish-bowl  -0.15130435 -0.8058509 0.5032422 0.9654709
## plate-bowl -0.28574879 -0.8819361 0.3104385 0.6599774
## tray-bowl  -0.31230435 -0.8866667 0.2620580 0.5451195
## dish-cass  -0.25400000 -1.0012541 0.4932541 0.8719822
## plate-cass -0.38844444 -1.0851486 0.3082598 0.5205458
## tray-cass  -0.41500000 -1.0931221 0.2631221 0.4264365
## plate-dish -0.13444444 -0.8986014 0.6297126 0.9873619
## tray-dish  -0.16100000 -0.9082541 0.5862541 0.9732292
## tray-plate -0.02655556 -0.7232598 0.6701486 0.9999687
```

```
# The reason the aov() summary output differs depending on whether -1 is used or
# not is due to different questions being asked.  If -1 is used, the F-test is
# evaluating if the group means are different from zero.  If -1 is not used, the
# F-test is considering if they differ among themselves.  As TukeyHSD shows,
# the rates do not differ.  We could simplify the model to using a single rate
# across the board.  The display of confidence intervals which overlap says this.
```