lab_stat565_2

shen

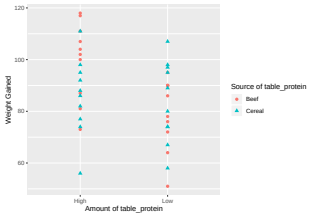
2019/3/8

Lab5 Two factor Factorial 1

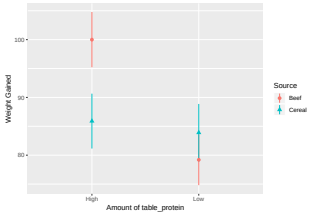
(a) Plot the data and report the plot here (A plot with data and means of treatment combinations). Do not report code here. Describe the observed relationship between two factors.

```
## Observations: 40
## Variables: 3
## $ Source <chr> "Beef", "Beef", "Beef", "Beef", "Beef", "Beef", "Beef",...
## $ Amount <chr> "Low", "Low", "Low", "Low", "Low", "Low", "Low", "Low",...
## $ Gain   <dbl> 90, 76, 90, 64, 86, 51, 72, 90, 95, 78, 73, 102, 118, 1...
```

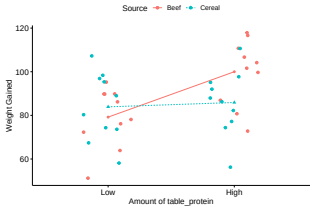
```
table_protein <- read_excel("Protein.xlsx")
glimpse(table_protein)
## Observations: 40
## Variables: 3
## $ Source <chr> "Beef", "Beef", "Beef", "Beef", "Beef", "Beef", "Beef",...
## $ Amount <chr> "Low", "Low", "Low", "Low", "Low", "Low", "Low", "Low",...
## $ Gain   <dbl> 90, 76, 90, 64, 86, 51, 72, 90, 95, 78, 73, 102, 118, 1...

# Install and load ggplot2 package before using ggplot function #
ggplot(data = table_protein, aes(x = Amount, y = Gain, colour=Source, group=Source))+
  geom_point(aes(shape = Source,color = Source),size = 2) +
  labs(y = "Weight Gained", x="Amount of table_protein", color ="Source of table_protein", shape ="Sour
```



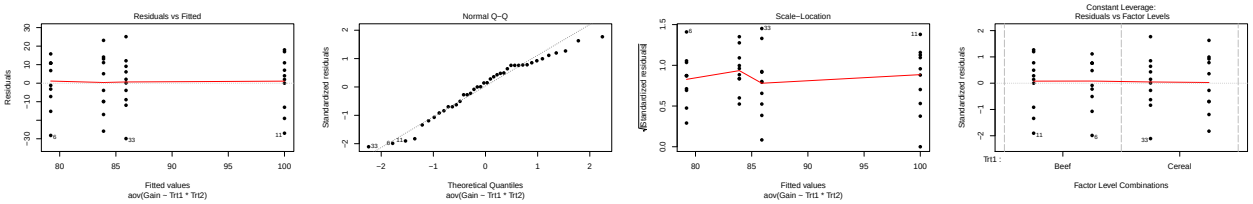

```
#Plots the Mean and 1SD error bars for each treatment group #
ggplot(data = table_protein, aes(x = Amount, y = Gain, colour=Source,shape = Source, group=Source)) +
  stat_summary() + labs(y= "Weight Gained", x="Amount of table_protein", color="Source", shape="Source"
## No summary function supplied, defaulting to `mean_se()
```




```
# Install and load ggpubr package before using ggline function #
ggline(data = table_protein, x = "Amount", y = "Gain", add = c("mean", "jitter"),shape= "Source",  colo
```

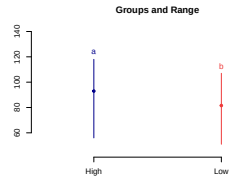



```
#Load mosaic package before using favstats function#
favstats(Gain ~ Source, data=table_protein)
##   Source min   Q1 median    Q3 max mean       sd  n missing
## 1   Beef  51 77.5     90 102.5 118 89.6 17.71232 20       0
## 2 Cereal  56 74.0     87  95.5 111 84.9 14.99438 20       0
favstats(Gain ~ Amount , data=table_protein)
##   Amount min    Q1 median     Q3 max  mean       sd  n missing
## 1   High  56 81.75   93.5 104.75 118 92.95 16.36259 20       0
## 2    Low  51 73.50   83.0  91.25 107 81.55 14.63045 20       0
favstats(Gain ~ Source|Amount, data=table_protein)
##        Amount min    Q1 median     Q3 max   mean       sd  n missing
## 1   Beef.High  73 90.25  103.0 110.00 118 100.00 15.13642 10       0
## 2 Cereal.High  56 78.25   87.0  94.25 111  85.90 15.02184 10       0
## 3    Beef.Low  51 73.00   82.0  90.00  95  79.20 13.88684 10       0
## 4  Cereal.Low  58 74.00   84.5  96.50 107  83.90 15.70881 10       0
## 5        High  56 81.75   93.5 104.75 118  92.95 16.36259 20       0
## 6         Low  51 73.50   83.0  91.25 107  81.55 14.63045 20       0
#favstats(Gain ~ Source+Amount, data=table_protein)

#Create Categorical variables so that plot of residuals versus each treatment combination can be obtain
table_protein$Trt1 = as.factor(table_protein$Source)
table_protein$Trt2 = as.factor(table_protein$Amount)

model_protein <- aov(Gain ~ Trt1*Trt2, data=table_protein)
summary(model_protein)
##             Df Sum Sq Mean Sq F value Pr(>F)
## Trt1         1    221   220.9   0.988 0.3269
## Trt2         1   1300  1299.6   5.812 0.0211 *
## Trt1:Trt2    1    884   883.6   3.952 0.0545 .
## Residuals   36   8049   223.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
plot(model_protein, pch=16)
```




```
# Pairwise comparisons using t tests with pooled Standard Deviation #
# The output gives a matrix of p values for each pair of treatments #
pairwise.t.test(table_protein$Gain, table_protein$Trt2, p.adj = "none")
##
```

```
##  Pairwise comparisons using t tests with pooled SD
##
## data:  table_protein$Gain and table_protein$Trt2
##
##     High
## Low 0.026
##
## P value adjustment method: none

# Pairwise comparisons using t tests with pooled Standard Deviation and Bonferroni adjustment #
# The output gives a matrix of p values for each pair of treatments #
pairwise.t.test(table_protein$Gain, table_protein$Trt2, p.adj = "bonf")
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  table_protein$Gain and table_protein$Trt2
##
##     High
## Low 0.026
##
## P value adjustment method: bonferroni

#Install and load the agricolae package before running the LSD.test function below #
#p.adj option in the LSD.test function can be used to apply different adjustments to control error rate
plot(LSD.test(model_protein, trt = "Trt2", alpha = 0.05))
```




```
#The treatments sharing the same letter on the plot are not different#

(LSD.test(model_protein, trt = "Trt2", alpha = 0.05)) # Using outer parentheses prints the output#
## $statistics
##    MSerror Df  Mean       CV  t.value  LSD
##   223.5944 36 87.25 17.13819 2.028094 9.59
##
## $parameters
##         test p.ajusted name.t ntr alpha
##   Fisher-LSD      none   Trt2   2  0.05
##
## $means
##       Gain      std  r      LCL      UCL Min Max   Q25  Q50    Q75
## High 92.95 16.36259 20 86.16885 99.73115  56 118 81.75 93.5 104.75
## Low  81.55 14.63045 20 74.76885 88.33115  51 107 73.50 83.0  91.25
##
## $comparison
## NULL
##
## $groups
```

```
##       Gain groups
## High 92.95      a
## Low  81.55      b
##
## attr(,"class")
## [1] "group"

#Tukey's test to get observed difference in means, CI and p value#
TukeyHSD(model_protein, conf.level = 0.95)
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Gain ~ Trt1 * Trt2, data = table_protein)
##
## $Trt1
##             diff    lwr  upr     p adj
## Cereal-Beef -4.7 -14.29 4.89 0.3268783
##
## $Trt2
##           diff    lwr   upr     p adj
## Low-High -11.4 -20.99 -1.81 0.0211449
##
## $`Trt1:Trt2`
##                          diff      lwr       upr     p adj
## Cereal:High-Beef:High  -14.1 -32.1102  3.910198 0.1697711
## Beef:Low-Beef:High     -20.8 -38.8102 -2.789802 0.0182745
## Cereal:Low-Beef:High   -16.1 -34.1102  1.910198 0.0936982
## Beef:Low-Cereal:High    -6.7 -24.7102 11.310198 0.7492577
## Cereal:Low-Cereal:High  -2.0 -20.0102 16.010198 0.9905411
## Cereal:Low-Beef:Low      4.7 -13.3102 22.710198 0.8952934

# Scheffe's test to get observed difference in means, CI and p value #
# Install and load the DescTools package before using the ScheffeTest function #
ScheffeTest(model_protein, conf.level = 0.95)
##
##   Posthoc multiple comparisons of means : Scheffe Test
##     95% family-wise confidence level
##
## $Trt1
##             diff    lwr.ci   upr.ci   pval
## Cereal-Beef -4.7 -18.56594 9.165941 0.8042
##
## $Trt2
##           diff    lwr.ci   upr.ci   pval
## Low-High -11.4 -25.26594 2.465941 0.1410
##
## $`Trt1:Trt2`
##                          diff   lwr.ci    upr.ci   pval
## Cereal:High-Beef:High  -14.1 -33.7094  5.509402 0.2358
## Beef:Low-Beef:High     -20.8 -40.4094 -1.190598 0.0338 *
## Cereal:Low-Beef:High   -16.1 -35.7094  3.509402 0.1418
## Beef:Low-Cereal:High    -6.7 -26.3094 12.909402 0.8004
## Cereal:Low-Cereal:High  -2.0 -21.6094 17.609402 0.9929
```

```
## Cereal:Low-Beef:Low      4.7 -14.9094 24.309402 0.9195
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```