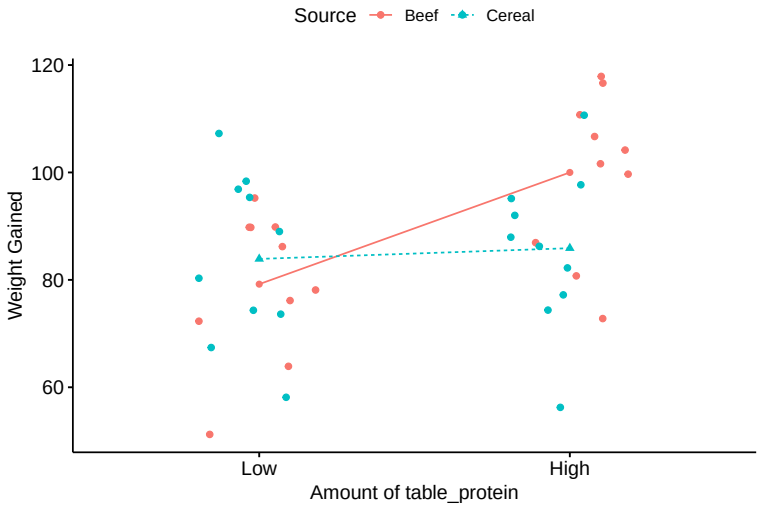
STAT565_Lab

Shen Qu

(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: 5
## $ 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...
## $ Trt1   <fct> Beef, Beef, Beef, Beef, Beef, Beef, Beef, Beef, Beef, B...
## $ Trt2   <fct> Low, Low, Low, Low, Low, Low, Low, Low, Low, Low, High,...
```



(b). Obtain the numerical summary for each treatment combination and factor levels separately. Report them here in a tabular form.

Source	min	Q1	median	Q3	max	mean	sd	n	missing
Beef	51	77.5	90	102.5	118	89.6	17.71	20	0
Cereal	56	74	87	95.5	111	84.9	14.99	20	0

Amount	min	Q1	median	Q3	max	mean	sd	n	missing
Beef.High	73	90.25	103	110	118	100	15.14	10	0
Cereal.High	56	78.25	87	94.25	111	85.9	15.02	10	0
Beef.Low	51	73	82	90	95	79.2	13.89	10	0
Cereal.Low	58	74	84.5	96.5	107	83.9	15.71	10	0
High	56	81.75	93.5	104.8	118	92.95	16.36	20	0
Low	51	73.5	83	91.25	107	81.55	14.63	20	0

(c). Fit the two-factor factorial model and report the complete ANOVA table here. Do not report code here. The complete ANOVA table should have a row for each of the following: main effects of each treatment, two-factor interaction effects, error and total.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Trt1	1	220.9	220.9	0.9879	0.3269
Trt2	1	1300	1300	5.812	0.02114
Trt1:Trt2	1	883.6	883.6	3.952	0.05447
Residuals	36	8049	223.6	NA	NA
Total	39	10453.5	268.0385	NA	NA

(d). Based on the ANOVA table write your conclusion appropriately. Perform all the necessary tests and report the conclusion along

with the p-value.

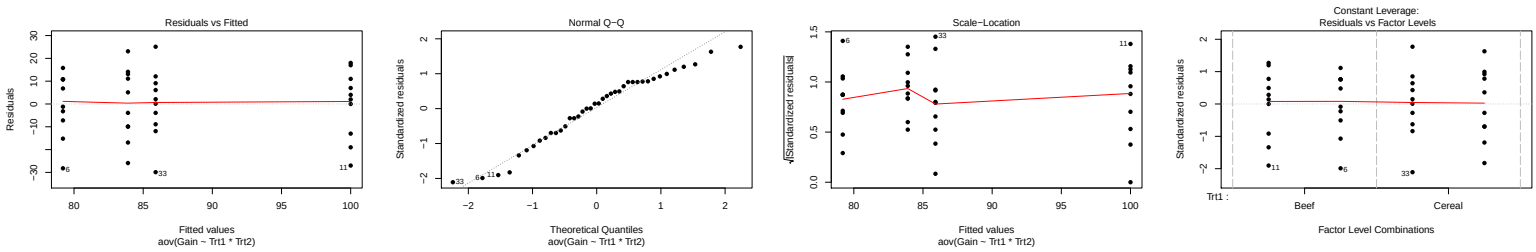
The line plot shows that not all lines are parallel. Difference in Gain between Trt1 is not same for different Trt2. There could be an interaction effect.

According to ANOVA table, there is a significant interaction effect from Trt1 and Trt2 on the Gain around 5% significance level (P-value=0.05447). That means, effect of method and effect of Trt1 and Trt2 on Gain is not independent. Therefore, examie the simple effects.

The table shows the simple comparisons of Trt1 Least Squares Means by Trt2 and Trt2 Least Squares Means by Trt1. Both Tukey and Scheffe methods indicate the difference in Gain between high-protein and low-protein is significant when Beef is applies (P-calue=0.01827,0.0338, respectively). The confidence intervals also support this results.

		Tukey			Scheffe		
	diff	lwr	upr	p adj	lwr.ci	upr.ci	pval
Cereal:High-Beef:High	-14.1	-32.11	3.91	0.1698	-33.71	5.509	0.2358
Beef:Low-Beef:High	-20.8	-38.81	-2.79	**0.01827**	-40.41	-1.191	**0.0338**
Cereal:Low-Beef:High	-16.1	-34.11	1.91	0.0937	-35.71	3.509	0.1418
Beef:Low-Cereal:High	-6.7	-24.71	11.31	0.7493	-26.31	12.91	0.8004
Cereal:Low-Cereal:High	-2	-20.01	16.01	0.9905	-21.61	17.61	0.9929
Cereal:Low-Beef:Low	4.7	-13.31	22.71	0.8953	-14.91	24.31	0.9195

(e). Provide the plots of residuals here. Do not report code here.



{(f). Based on the residual plots, clearly explain whether assumptions in the model are satisfied or violated.

The plot of studentized residual versus predicted (fitted) value shows that except few outliers, the residuals are evenly distributed about zero at each prededict value (zero mean) and vertical deviations of residuals from zero are about same for each predicted value (constant variance).

The plots of studentized residual versus factor levels didn't show obvious violation of zero mean and constant variance.

The QQ plot shows that some data points are not on the line and flattening at the extremes, which is a litte violation of normality.

(g). Report the code here without output.

```
table_protein <- read_excel("Protein.xlsx")
glimpse(table_protein)
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 = "Source of table_protein")
# 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")
# Install and load ggpubr package before using ggline function #
ggline(data = table_protein, x = "Amount", y = "Gain", add = c("mean", "jitter"),
    shape = "Source", color = "Source", linetype = "Source", ylab = "Weight Gained",
    xlab = "Amount of table_protein")
# Load mosaic package before using favstats function#
favstats(Gain ~ Source, data = table_protein)
favstats(Gain ~ Amount, data = table_protein)
favstats(Gain ~ Source | Amount, data = table_protein)
favstats(Gain ~ Source + Amount, data = table_protein)

table_protein$Trt1 = as.factor(table_protein$Source)
table_protein$Trt2 = as.factor(table_protein$Amount)
```

```r
model_protein <- aov(Gain ~ Trt1 * Trt2, data = table_protein)
summary(model_protein)
pander(summary(model_protein))
sum((table_protein$Gain - mean(table_protein$Gain))^2)
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 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")
# 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 rates#
plot(LSD.test(model_protein, trt = "Trt2", alpha = 0.05))
(LSD.test(model_protein, trt = "Trt2", alpha = 0.05))
pander(TukeyHSD(model_protein, conf.level = 0.95)[3])
pander(ScheffeTest(model_protein, conf.level = 0.95)[3])
```