R Exploratory Analysis:
Cheese products variation in nutrition profiles

Jean Batista
Juniata College

Abstract

This study examines how product type and texture relate to caloric density in retail cheese items. Using a hand-curated dataset of 60 cheeses (six types; standardized to 28 g servings, with a second view for cottage cheese at 125 g), I evaluate macronutrient distributions and model calories from fat, protein, and carbohydrates, plus type and interaction effects. Fat alone explains ~93% of variation in calories; a two-predictor model with fat and protein explains ~99%. An interaction between fat and cheese type captures cottage cheese's unique moisture-driven nutrient density. Findings illustrate how formulation and texture shape caloric outcomes and demonstrate practical model building and diagnostics in R.

Introduction

Nutritional labels summarize product composition, yet products within a category can vary markedly. This project quantifies within-category variation for cheese and explains caloric differences using macronutrients and product attributes. Beyond a consumer lens, the analysis serves as a compact case study in exploratory data analysis (EDA), linear modeling, and model diagnostics.

Data & Standardization

I assembled a dataset of 60 cheese SKUs across six types (Cheddar, Swiss, Mozzarella, Parmesan, Brie, Cottage), recording calories, fat, protein, carbohydrates, type, and texture (Soft → Very Hard). Nutrition values were sourced from manufacturer sites or verified images; servings were standardized to 28 g (1 oz). Because cottage cheese is water-rich and typically consumed at larger portions, I also retained a 125 g view to illustrate texture-moisture effects on density. Analyses prioritize the 28 g standardization to make types comparable.

Methods

Using R with tidyverse and plotting via ggplot2, I created distributions and grouped bar charts by type/texture; scatterplots relating calories to macronutrients. Modeling using Ordinary least squares (OLS) regressions predicting calories from fat, protein, carbohydrates, and cheese type; I compare (i) single-predictor models, (ii) additive multi-predictor models, (iii) parallel-slopes models with type, and (iv) type×fat interaction models. Linearity/independence/normality/equal-variance (LINE) diagnostics were examined via residual plots.

Exploratory Findings

- **Texture trend.** Harder cheeses tend to be more calorie-dense; however, type moderates this trend (Parmesan is very hard yet not the highest in fat/calories per 28 g).
- **Macronutrients.** Calories rise with fat and (to a lesser extent) protein; carbohydrates show little systematic relationship in this category.
- **Portioning & density.** Cottage cheese appears as a low-calorie outlier at 28 g due to high moisture. Presenting it at 125 g clarifies that perceived "leanness" is partly an artifact of portion size and water content.

Models & Results

Single-predictor models (28 g dataset)

- **Calories ~ Fat.** Strong fit ($R^2 \approx 0.93$). The slope aligns with the intuition that fat (≈ 9 kcal/g) dominates variation; the estimated coefficient is a bit larger because fat co-moves with protein in richer cheeses. Residuals are centered and pattern-free, supporting LINE.
- **Calories ~ Protein.** Moderate fit ($R^2 \approx 0.69$). Residual structure suggests omitted-variable effects (fat).
- **Calories ~ Carbohydrates.** Very weak fit ($R^2 \approx 0.04$); not a meaningful driver here.

Two-predictor model

- **Calories ~ Fat + Protein.** Excellent fit ($R^2 \approx 0.99$) with residual SD ≈ 3 kcal. Coefficients are close to nutritional heuristics (~9 kcal/g fat, ~4 kcal/g protein), reflecting their additive energetic contributions. Diagnostics show well-behaved residuals.

Type effects (parallel slopes)

Adding **type** as a factor to **Calories ~ Fat** further improves fit ($R^2 \approx 0.994$). Not all type contrasts are significant, but several are, indicating brand-/style-level formulation differences beyond fat alone.

Interaction effects (type × fat)

Allowing the fat slope to vary by **type** yields the best statistical fit ($R^2 \approx 0.994$) and captures the lower fat→calorie slope for **cottage cheese**, consistent with moisture-driven density differences. Most other type-specific slopes do not deviate meaningfully from the common fat effect. This targeted interaction balances interpretability and fidelity.

Discussion

Across SKUs, caloric density is primarily a function of fat and, secondarily, protein. Texture correlates with density but is not determinative once type and macronutrients are considered. The cottage-cheese case underscores how moisture and serving conventions shape perceived “leanness.” From a modeling standpoint, the progression from univariate to additive and interaction models illustrates how to (i) diagnose omitted variables, (ii) reconcile domain heuristics with estimates, and (iii) judiciously add complexity only where the data supports it.

Conclusion

In retail cheeses, fat (with protein) explains nearly all variation in calories at a standardized portion. A compact interaction (type×fat) isolates cottage cheese’s moisture-driven behavior without over-fitting. Methodologically, the study demonstrates transparent modeling.

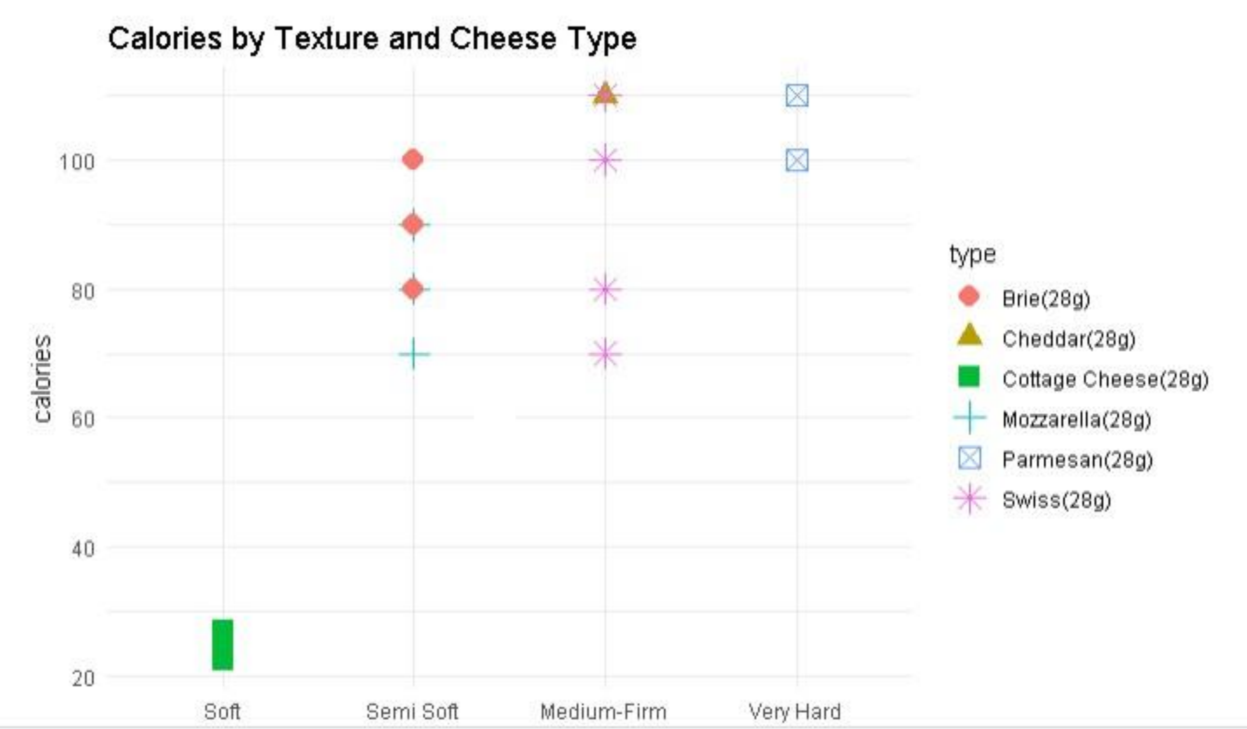
Reading the file

```
cheese_data <- read.csv("cheese_data.csv") cheese_data

##                        brand              type  calories   protein
## 1                      Kraft         Cheddar(28g) 110.00000
9.000000
## 36                Great Value        Parmesan(28g) 100.00000
```

Library load

```
library(tidyverse)

library(fivethirtyeight) library(moderndive)

library(readr)
```

```
cheese_data_cot_28 <- cheese_data[c(1:50, 61:70), ] cheese_data_cot_125 <-

cheese_data[1:60,]

ggplot(cheese_data_cot_28, aes(x = Texture, y = calories, shape = type, color
= type)) +
  geom_point(size = 4) +
  labs(title = “Calories by Texture and Cheese Type”)
```



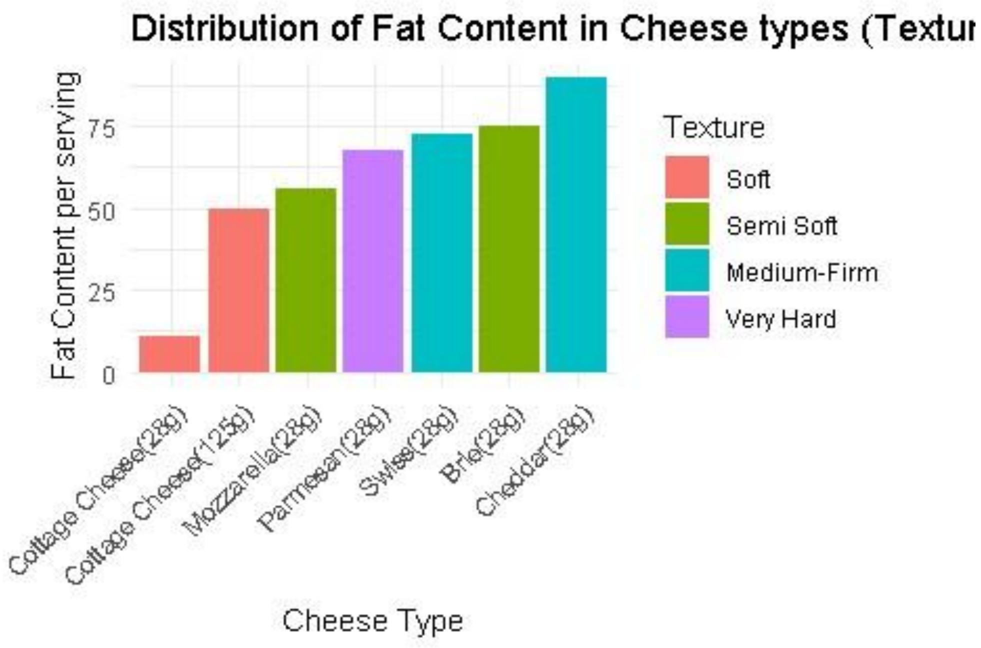
Cottage cheese as an outlier can be studied further to evaluate why exactly it has lower nutrient density.

(https://extension.psu.edu/fat-facts-the-right-amount-for-a-healthy-diet). A gram of fat is equal to 9 calories. Where Carbs and protein are both 4. Based on this information, cheese will often be a high calorie snack, considering its high fat content.
Consumers looking for a healthier alternative can filter for low fat content.

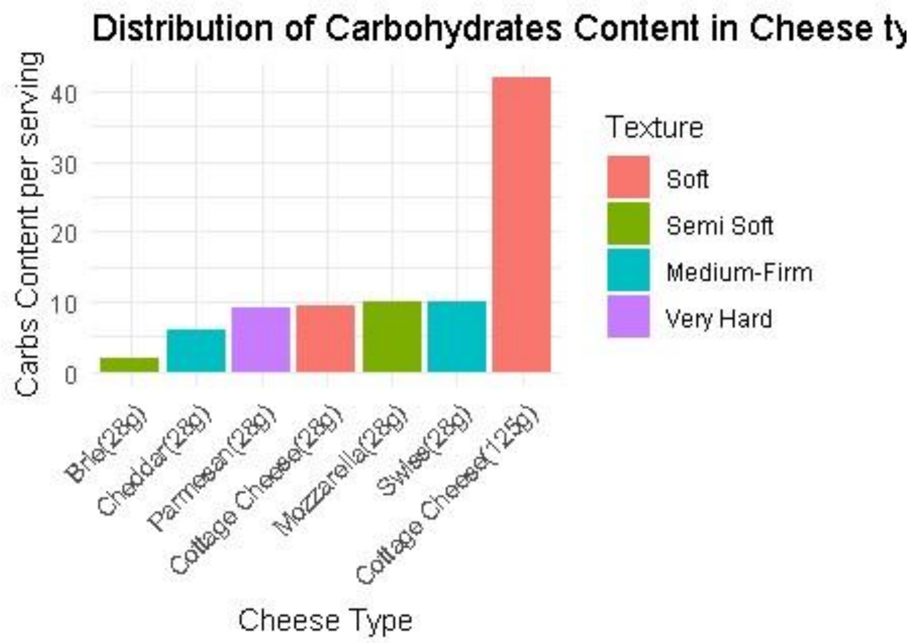
```
cheese_data %>%  filter(fat < 6)

##                  brand          type  calories  fat  ## 4           Organic Valley     Mozzarella(28g)
70.00000  5

4  more mozzarellas…
## 5                 Alpine Lace         Swiss(28g)  70.00000  4.5
## 6                Breakstone's  Cottage Cheese(125g) 110.00000  2.5

5  more cottage cheese…
```

I want to see the relationship between our categorical variable Texture and some of the numerical variables. Cottage cheese will be graphed with both 28 grams and 125 grams to showcase the effect of texture on nutrient density, and the need to increase serving size to accommodate for this difference.

```
ggplot(cheese_data, aes(x = reorder(type, fat), y = fat, fill = Texture)) +   geom_col() +
  scale_fill_discrete(limits = c("Soft", "Semi Soft", "Medium-Firm", "Very Hard")) +
  labs(title = "Distribution of Fat Content in Cheese types (Texture)",x =
"Cheese Type", y = "Fat Content per serving") +   theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1),         plot.margin = margin(1, 1, 1,
1, "cm"))
```

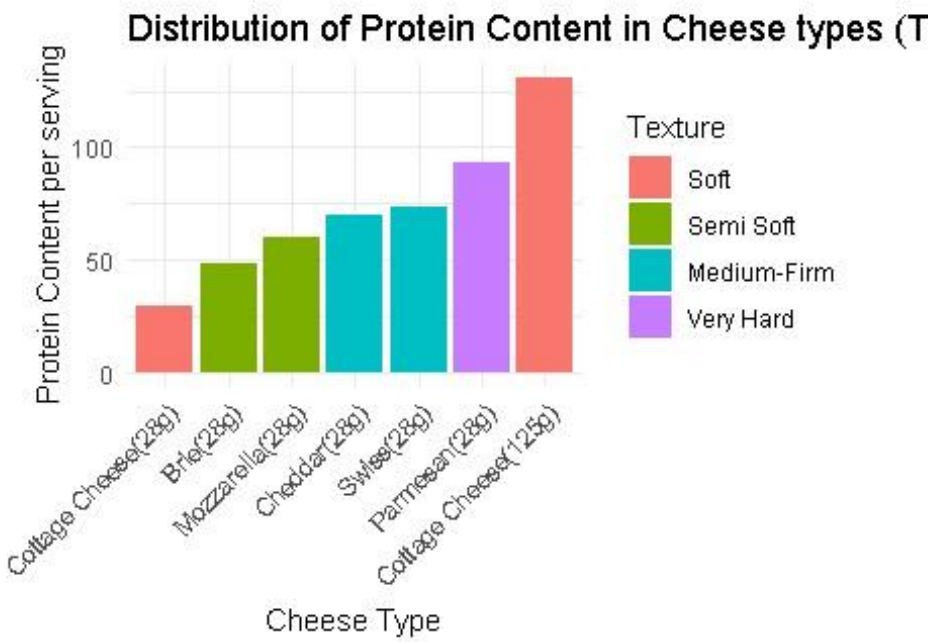


```
ggplot(cheese_data, aes(x = reorder(type, carbohydrates), y = carbohydrates, fill = Texture)) +
geom_col() +
  scale_fill_discrete(limits = c("Soft", "Semi Soft", "Medium-Firm", "Very
Hard")) +   labs(title = "Distribution of Carbohydrates Content in Cheese types (Texture)", x =
"Cheese Type", y = "Carbs Content per serving") +   theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1),


        plot.margin = margin(1, 1, 1, 1, "cm"))
```



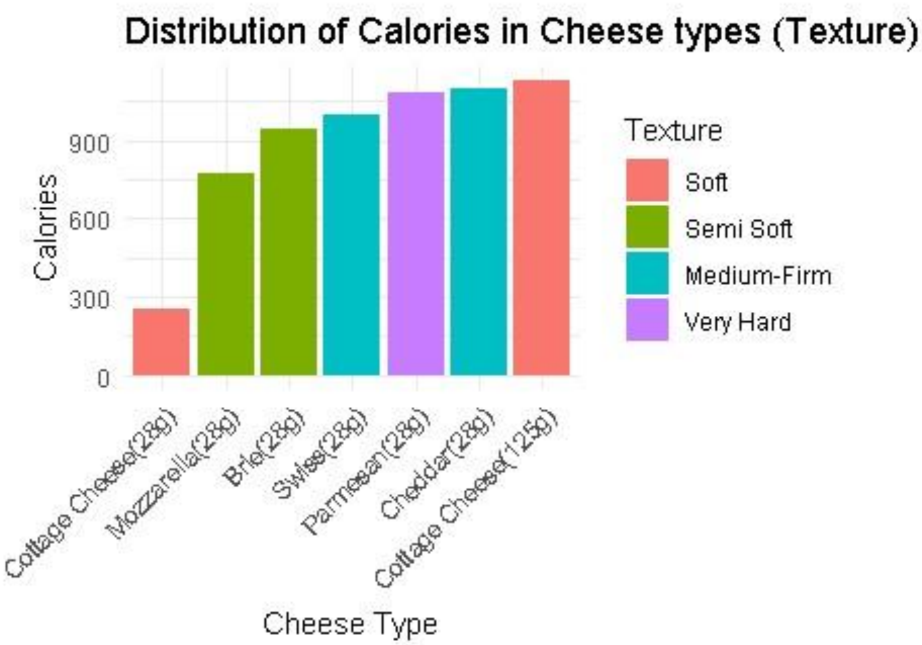
#Protein

```
ggplot(cheese_data, aes(x = reorder(type, protein), y = protein, fill = Texture)) +   geom_col() +
  scale_fill_discrete(limits = c("Soft", "Semi Soft", "Medium-Firm", "Very Hard")) +
  labs(title = "Distribution of Protein Content in Cheese types (Texture)", x
= "Cheese Type", y = "Protein Content per serving") +   theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1),         plot.margin = margin(1, 1, 1,
1, "cm"))
```



Cottage cheese is the softest (it has very high moisture). The water content in this type of cheese makes it less nutrient dense per ounce. People would not normally eat 28 grams of cottage cheese since it would only be about a spoonful, versus the denser cheeses that are compact with nutrients.

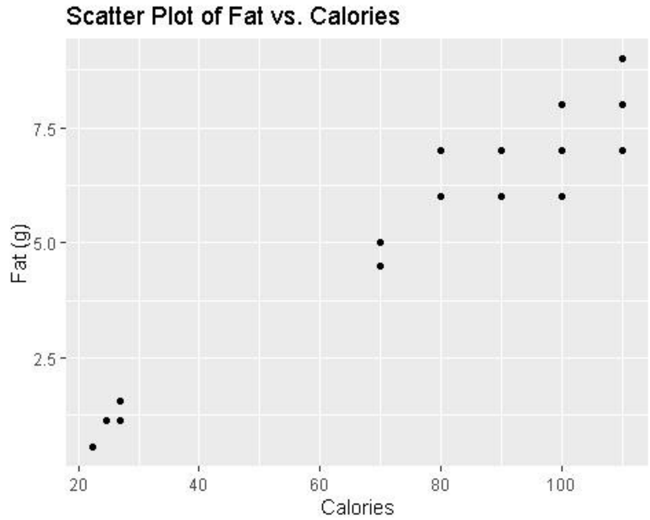
```
ggplot(cheese_data, aes(x = reorder(type, calories), y = calories, fill = Texture)) +   geom_col() +
  scale_fill_discrete(limits = c("Soft", "Semi Soft", "Medium-Firm", "Very Hard")) +
  labs(title = "Distribution of Calories in Cheese types (Texture)",x =
"Cheese Type", y = "Calories") +     theme_minimal()+
   theme(axis.text.x = element_text(angle = 45, hjust = 1),     plot.margin = margin(1, 1, 1, 1,
"cm"))
```



The trend shows as a cheese gets harder it will be more nutrient dense, containing more fat and calories. But this is a trend not absolute truth, as parmesan is the hardest cheese but contains less fat content and calories than several cheeses.

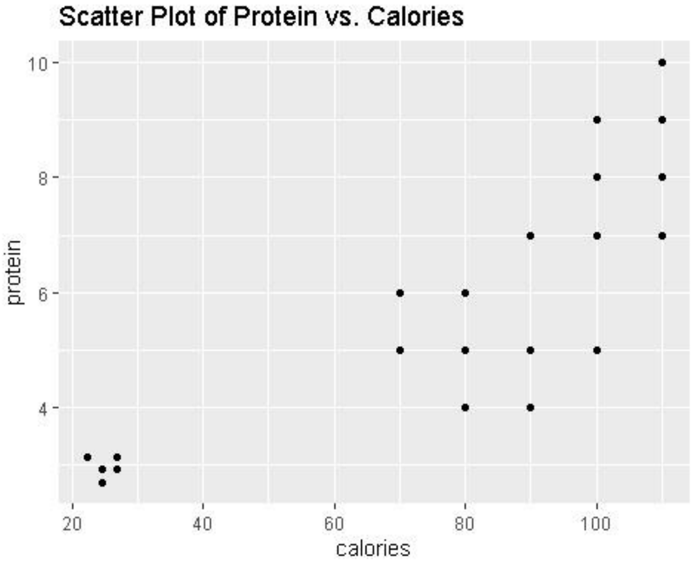
The following scatterplots will be used to visualize the correlation between calories and each numerical variable:

```
ggplot(cheese_data_cot_28, aes(x = calories, y = fat)) +   geom_point() +
labs(
   title = "Scatter Plot of Fat vs. Calories",     x = "Calories",
y = "Fat (g)"


  )
```



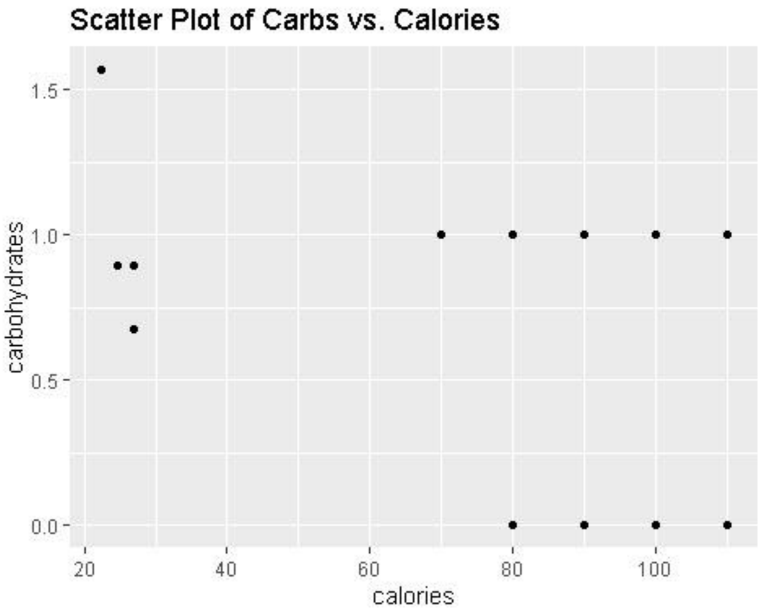
```
ggplot(cheese_data_cot_28, aes(x = calories, y = protein)) +   geom_point() +
labs(
   title = "Scatter Plot of Protein vs. Calories",   )
```

There is an upwards trend for fat.



```
ggplot(cheese_data_cot_28, aes(x = calories, y = carbohydrates)) +   geom_point() +
labs(
   title = "Scatter Plot of Carbs vs. Calories",   )
```

There is an upwards trend for protein.



There is an upwards trend for both fat and protein, but there is not any specific trend for carbohydrates. As cheese becomes richer, it will increase in fat and protein, causing the calories to increase.

We can isolate different variables to see their effect on calories. Based on the observations above we see some correlation between fat, protein, and texture. But our models below will be used to predict the expected amount.

```
# Model 1: Predict Calories from Fat model_fat <- lm(calories ~ fat, data =
cheese_data_cot_28)

# Model 2: Predict Calories from Protein model_protein <- lm(calories ~ protein, data =
cheese_data_cot_28)

# Model 3: Predict Calories from Carbs
model_carbs <- lm(calories ~ carbohydrates, data = cheese_data_cot_28) summary(model_fat)

##
## Call:
## lm(formula = calories ~ fat, data = cheese_data_cot_28) ##
## Residuals:
##     Min      1Q  Median      3Q     Max  ## -14.622  -5.886  -
3.357   4.114  16.642
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  15.7730     2.7691   5.696  4.3e-07 *** ## fat         11.2641     0.4123  27.318
< 2e-16 *** ## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.166 on 58 degrees of freedom
## Multiple R-squared:  0.9279, Adjusted R-squared:  0.9266
## F-statistic: 746.3 on 1 and 58 DF,  p-value: < 2.2e-16
```

For our Fat Model:

B0 = 15.773 : When fat is 0g we can expect 15.773 calories

B1 = 11.2641: For every gram of fat, there is an expected increase of 11.26 calories

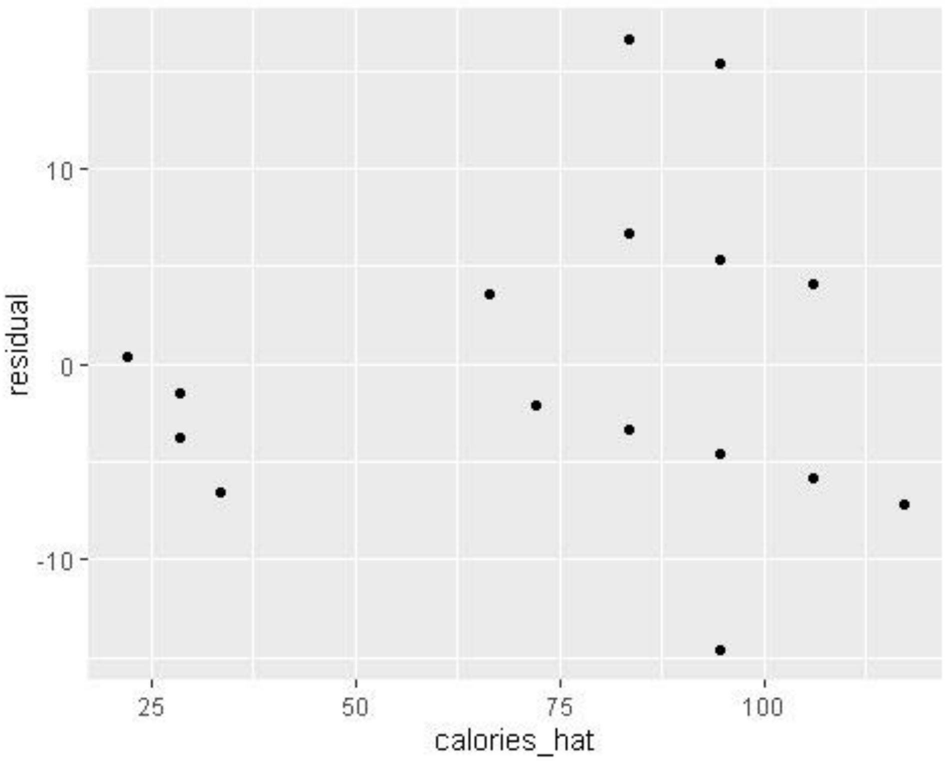
R-squared = .9279: About 92.8% of the variation in calories is explained by Fat

RSE = 8.166: Expected vs Actual calories may differ at about 8 calories.

Fat is a strong predictor, the high R-squared suggests as such. I found it interesting that the slope is 11.26, because as we mentioned earlier, each gram of fat is equal to 9 calories. What this tells us is that as fat increases, an increase in carbs and/or protein aswell.

Does it match LINE?

```
get_regression_points(model_fat) -> fat_residual_info ggplot(fat_residual_info, aes(x = calories_hat,
y=residual )) +   geom_point()
```



The plot looks randomly centered around zero, it matches with LINE.

```
summary(model_protein)

##
## Call:
## lm(formula = calories ~ protein, data = cheese_data_cot_28) ##
## Residuals:
##     Min      1Q  Median      3Q     Max  ## -26.368 -14.479
-3.054  14.970  30.899
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.196      6.958   1.609    0.113     ## protein       11.976
1.061  11.287 2.89e-16 *** ## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.01 on 58 degrees of freedom
## Multiple R-squared:  0.6872, Adjusted R-squared:  0.6818
## F-statistic: 127.4 on 1 and 58 DF,  p-value: 2.888e-16
```

For our Protein Model:

B0 = 11.196 : When protein is 0g we can expect 11.196 calories (very close to our fat slope)

B1 = 11.976: For every gram of protein, there is an expected increase of 11.976 calories

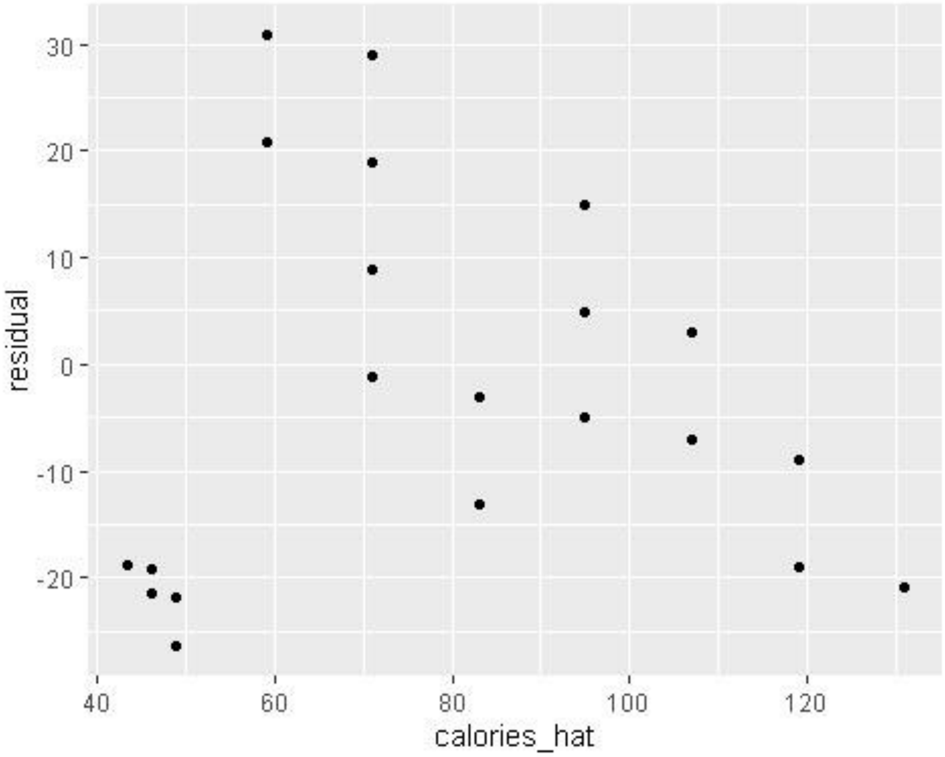
R-squared = .68772: About 68.7% of the variation in calories is explained by protein

RSE = 17.01: Expected vs Actual calories may differ at about 17 calories.

Protein is a fair predictor. Each gram of protein is equal to 4 calories. The other calories occur because an increase of protein leads to an increase of fat, which as seen before was the primary diver of calorie variation.

Does it match LINE?

```
get_regression_points(model_protein) -> protein_residual_info ggplot(protein_residual_info, aes(x =
calories_hat, y=residual )) +
  geom_point()
```



The points show a downwards trend. The model does not match LINE.

```
summary(model_carbs)
```

```
##
## Call:
## lm(formula = calories ~ carbohydrates, data = cheese_data_cot_28) ##
## Residuals:
##    Min     1Q Median     3Q    Max  ## -60.20 -12.63
10.54  27.37  27.37
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    96.296      8.120  11.860   <2e-16 *** ## carbohydrates  -13.668
9.238  -1.479    0.144     ## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.85 on 58 degrees of freedom
## Multiple R-squared:  0.03637,    Adjusted R-squared:  0.01975
## F-statistic: 2.189 on 1 and 58 DF,  p-value: 0.1444 \
```

For our Carbs Model:

B0 = 96.296 : When carbs is 0g we can expect 96 calories.

B1 = -13.668: For every gram of carbs, there is an expected decrease of 13.6 calories

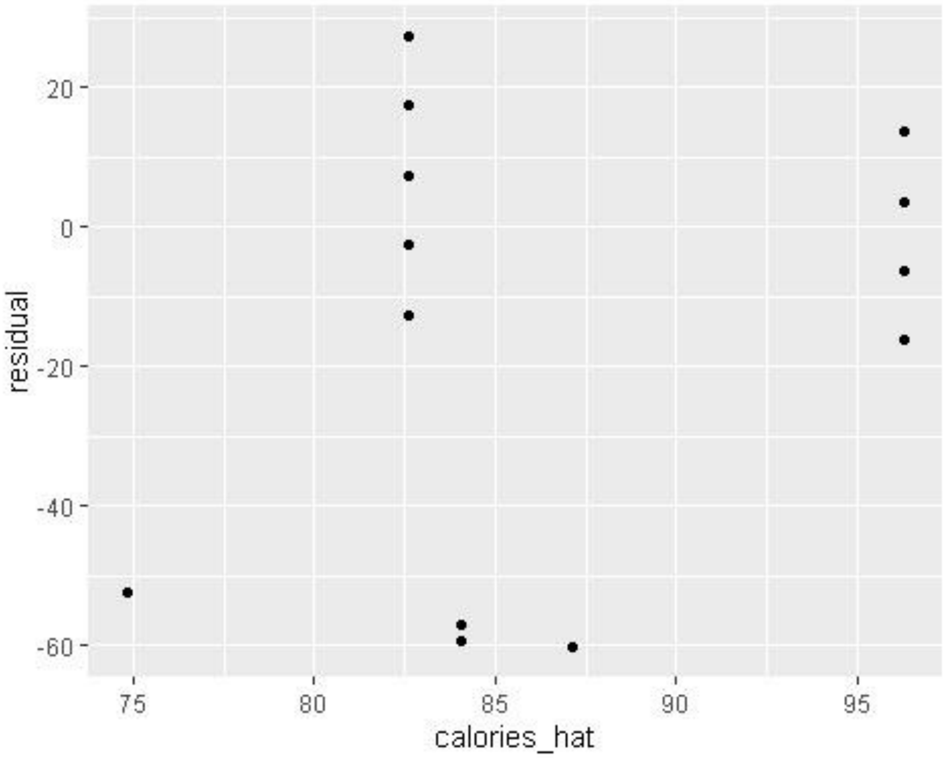
R-squared = .03637: About 3.6% of the variation in calories is explained by carbs

RSE = 29.85: Expected vs Actual calories may differ at about 29 calories.

Carbs have a weak relationship with cheese in this model, as suggested by the very low r squared. Scientific research states each gram of carbohydrates is equal to 4 calories. But here there is a negative slope for carbs. Due to this variable being insignificant we can't conclude there's a real negative association in this model.

Does it match LINE?

```
get_regression_points(model_carbs) -> carbs_residual_info ggplot(carbs_residual_info, aes(x =
calories_hat, y=residual )) +
  geom_point()
```



There is a vertical line pattern, and the residuals are not centered around zero, so this does not match LINE.

Since fat is our primary driver for calories, I'm going to use it as the variable to make predictions…

Calories = 15.77 +11.26 (fat)

Suppose we have cheese with 9 grams of fat, we plug in at 9 * 11.26 and get our result of 117.11

Predicted calories of 117.11, if we use cheddar with 9 grams of fat, actual calories are around 110. Meaning this model is somewhat accurate but is not exactly correct yet.

Now I'm going to look at the model for Fat + Protein.

```
model_fat_and_protein <- lm(calories ~ fat + protein, data = cheese_data_cot_28)
summary(model_fat_and_protein)

##
## Call:
## lm(formula = calories ~ fat + protein, data = cheese_data_cot_28) ##
## Residuals:
##    Min     1Q Median     3Q    Max  ## -4.952 -2.834
0.012  2.889  8.124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0580     1.3465   1.528    0.132
## fat           8.6558     0.2206  39.240   <2e-16 *** ## protein       4.8069
0.2725  17.638   <2e-16 *** ## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.242 on 57 degrees of freedom
## Multiple R-squared:  0.9888, Adjusted R-squared:  0.9884
## F-statistic:  2524 on 2 and 57 DF,  p-value: < 2.2e-16
```

For our Fat and Protein Model:

B0 = 2.05 : When carbs and protein is 0g we can expect 2.05 calories (carbs).

B1 = 8.6558: For every gram of fat, there is an expected decrease of 8.6 calories

B2 = 4.8069: For every gram of protein, there is an expected decrease of 4.8 calories

R-squared = .9888: About 98.9% of the variation in calories is explained by carbs

RSE = 3.242: Expected vs Actual calories may differ at about 29 calories.

This model is our strongest one yet, as suggested by the very high r squared. The slopes in this model are more in line with the scientific research of 9 cals per gram of fat and 4 cals per gram of protein. But here there is a negative slope for carbs.

The equation for this model is:

Calories = 2.06 + 8.66(fat) + 4.81(protein)

Therefore 8 grams of fat and 5 grams of protein (were taking Brie cheese as an example):

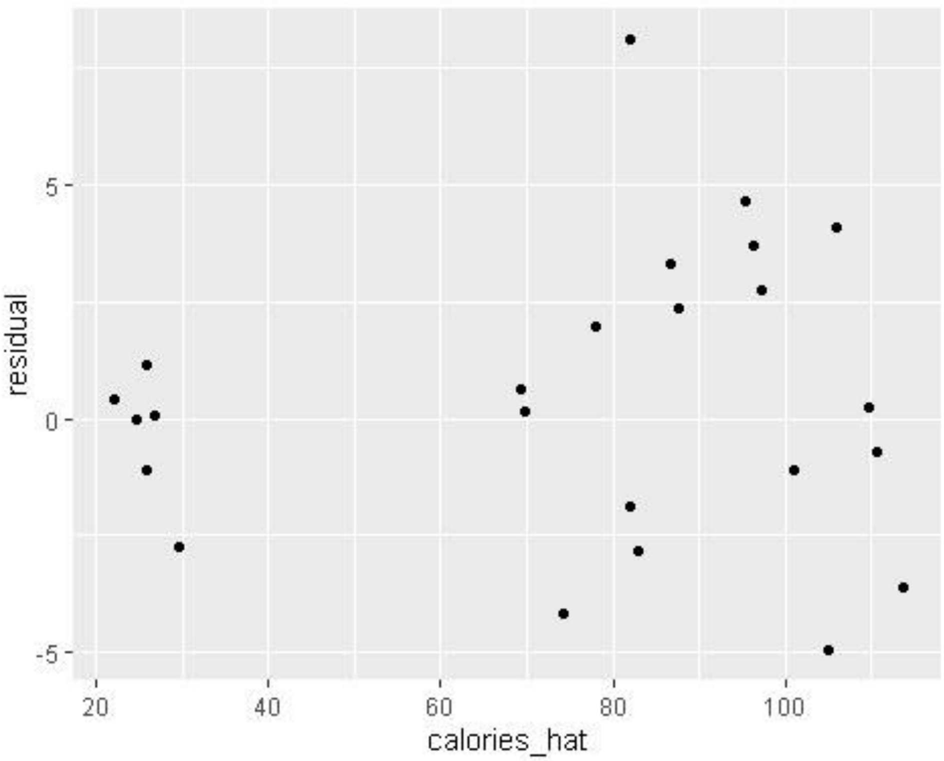
(8* 8.66 = 69.28) (5* 4.81 = 24.05), The sum of these 2 is 93.33. Plus the interception (2.06) equals 95.39.

Observed values of Brie are at around 100 calories. The prediction is accurate although it still could improve.

Does it match LINE?

#Look at the residual plot for this model. How well does it match LINE?

```
get_regression_points(model_fat_and_protein) -> fat_protein_residual_info
ggplot(fat_protein_residual_info, aes(x = calories_hat, y=residual )) +   geom_point()
```



This model matches with LINE as there is no pattern and spread is centered around zero.

Now I'm going to evaluate the parallel model and interaction model, and determine which model fits best.

```
model_parallel_fat_and_type <- lm(calories ~ fat + type, data = cheese_data_cot_28)
summary(model_parallel_fat_and_type)

##
## Call:
## lm(formula = calories ~ fat + type, data = cheese_data_cot_28) ##
## Residuals:
##     Min      1Q  Median      3Q     Max  ## -8.4527 -0.2541
0.0000  1.4432  8.4919
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               9.4728     4.3354   2.185  0.03333 *
## fat                      11.2703     0.5682  19.834  < 2e-16 ***
## typeCheddar(28g)         -0.9054     1.4123  -0.641  0.52422
## typeCottage Cheese(28g)   3.3550     3.8016   0.883  0.38148
## typeMozzarella(28g)       4.4136     1.5601   2.829  0.00658 **
## typeParmesan(28g)       21.8892     1.1943  18.328  < 2e-16 *** ## typeSwiss(28g)           8.8176
1.1351   7.768 2.63e-10 *** ## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.518 on 53 degrees of freedom
## Multiple R-squared:  0.9937, Adjusted R-squared:  0.993
## F-statistic:  1401 on 6 and 53 DF,  p-value: < 2.2e-16
```

This model fits the data exceptionally well, as seen in the r-squared of .9937. Not all cheese types are significant (cheddar and cottage cheese). But the other 4 have low p values and are significant.

The equation is:

Calories = 11.27(FAT) + 9.47(intercept) -0.90(cheddar) + 3.35(cottage cheese) + 4.41(mozzarella) + 21.88(parmesean) + 8.81(swiss)

I want to look at cottage cheese alone, because of the observations made earlier:

Cottage Cheese Calories = 11.27(1.1) + 9.47(b0) + 3.35(1) = 25.217

Our observed value for cottage cheese is at around 25.62, so this prediction came extremely close.

If I look at the prediction for cheddar calories:

Cheddar Calories = 11.27(9) + 9.47(b0) - 8.1(1) = 102.8

Our observed value for cheddar is at around 110, so this prediction is off by around 7.2 calories.

Now for the interaction model:

```
model_parallel_fat_and_type <- lm(calories ~ fat * type, data = cheese_data_cot_28)
summary(model_parallel_fat_and_type)

##
## Call:
## lm(formula = calories ~ fat * type, data = cheese_data_cot_28) ##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6139 -0.1042  0.0000  1.3861  8.3333
```

```
##
## Coefficients: (1 not defined because of singularities)
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4.0000    11.9181   0.336   0.7386
## fat                         12.0000     1.5856   7.568 8.78e-10 ***
## typeCheddar(28g)            -2.0000     2.6294  -0.761   0.4505
## typeCottage Cheese(28g)     16.2890    12.5591   1.297   0.2007
## typeMozzarella(28g)          7.6667    14.9932   0.511   0.6114     ## typeParmesan(28g)
36.0000    18.0086   1.999   0.0512 .
## typeSwiss(28g)              12.7327    12.9938   0.980   0.3319
## fat:typeCheddar(28g)             NA         NA      NA       NA     ## fat:typeCottage Cheese(28g)  -
7.4523     3.8394  -1.941   0.0580 .
## fat:typeMozzarella(28g)     -0.3333     2.2656  -0.147   0.8836
## fat:typeParmesan(28g)       -2.0000     2.5381  -0.788   0.4345     ## fat:typeSwiss(28g)
-0.5149     1.7355  -0.297   0.7680     ## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.507 on 49 degrees of freedom
## Multiple R-squared:  0.9943, Adjusted R-squared:  0.9931  ## F-statistic: 848.5 on
10 and 49 DF,  p-value: < 2.2e-16
```

This model fits the data exceptionally well, as seen in the r-squared of .9943, slightly higer than the parallel model. But most of the interaction terms are insignificant, except for the cottage cheese (.0580). This coincides with our observations earlier, that cottage cheese is very soft and low in fat(or has more water content), making its calorie density lower than that of the other cheeses.

Let's look in this model in action:

cottage cheese calories = (4 + 16.28) + (12-7.4523) fat = cottage

cheese calories = 20.29 + 4.55(fat)

If we plug in 1.1 for fat, the calories are 25.295

Our observed value for cottage cheese is at around 25.62, so this prediction came extremely close.

If we look at a type of cheese that is insignificant in this model for example Parmesan:

calories = (4 + 36) + (12-2)fat = calories

= 40 + 10(fat) =

If fat is 7, calories are 110. Our observed values are 110, so it perfectly matched.

Let's see for Mozzarella: **calories = (4 + 7.667) + (12-.0333)fat =**

calories = 11.667 + 11.667(fat) =

if fat is 6, calories are 81.668. Our observed value is 80, making this model very accurate.

Based on my testing, the interaction model is the best fit. Overall, while the interaction model is slightly more complex than the parallel model, it highlights the unique behavior of Cottage Cheese. For the other types, the effect of fat on calories is similar, but for Cottage Cheese, the lower slope in the interaction model(-7.45) captures its distinctive nutritional profile.