Data_Project

Kenan Sooklall

2/17/2021

https://github.com/kateptrv/Candles/blob/main/Candles%20code%20examples.R

Abstract

One sign of Covid-19 is often the loss of taste and smell, also known as anosmia, and even those without other symptoms have experienced a loss in those senses. Reviews scraped from Amazon.com on protein powder were parsed for the proportion of tasteless or no taste comments before and during 2020. Through hypothesis testing with

Conducting a two-tailed z-test and A p value of 25% if outside the range to reject the null hypothesis. However there is a notable increase in the number of reviews and the proportion of comment stating a lack of taste does show an increase. We can say with 95% confidence that a sample of 1000 before 2020 would have between 12-15% (1.56% margin of error) of reviews that state there is no taste or equivalent in their protein powder. While in the year 2020 those reviews increased to 17-21% (1.96% margin of error) of reviews.

Introduction

Harvard professor Kate Petrova has analyzed sense of smell thorough reviews on scented candles. This report will expand on Professor Petrova research by look at how the sense of taste has changed since the first confirmed cased of Covid-19. The analysis will be done on the protein powders below

OPTIMUM NUTRITION
GOLD STANDARD®
100% WHEY
FOR MUSCLE SUPPORT & RECOVERY
24G PROTEIN
HELPS BUILD AND MAINTAIN MUSCLE†
5.5G BCAAs**
SUPPORTS ENDURANCE AND RECOVERY
WHEY PROTEIN ISOLATE PRIMARY SOURCE
BANNED SUBSTANCE TESTED
DOUBLE RICH CHOCOLATE
NATURALLY & ARTIFICIALLY FLAVORED
100% OF THE PROTEIN FROM WHEY
PROTEIN POWDER DRINK MIX
NET WT 5 LB (2.27 KG)
74 SERVINGS
NOTICE: USE THIS PRODUCT AS A FOOD SUPPLEMENT ONLY. DO NOT USE FOR WEIGHT REDUCTION.

1.

B000QSNYGI - Optimum Nutrition Gold Standard 100% Whey Protein Powder, Double Rich Chocolate, 5 Pound (Packaging May Vary)

NEW!
MUSCLETECH
RESEARCH & DEVELOPMENT
PERFORMANCESERIES
AMERICA'S #1 SELLING
PHASE8 PROTEIN
MULTI-PHASE 8-HOUR PROTEIN
PROVIDES HIGH BIOLOGICAL VALUE AND SUSTAINED RELEASE PROTEIN
26g Protein
7 Protein Sources
8 HOURS OF AMINO ACID DELIVERY¹
5g Glutamine & Precursor
5.6g BCAAs
INCREASES MUSCLE SIZE AND STRENGTH
MILK CHOCOLATE
NATURAL AND ARTIFICIAL FLAVORS
DIETARY SUPPLEMENT NET WT. 4.60 lbs. (2.09kg)

2.

B00BEOHFKO - Whey Protein Powder | MuscleTech Phase8 Protein Powder | Whey & Casein Protein Powder Blend | Slow Release 8-Hour Protein Shakes | Muscle Builder for Men & Women | Chocolate, 4.6 lbs (50 Servings)

3. B00NLR1PX0 - Vital Proteins Collagen Peptides Powder Supplement (Type I, III) for Skin Hair Nail Joint - Hydrolyzed Collagen - Non-GMO - Dairy and Gluten Free - 20g per Serving - Unflavored 10 oz Canister
4. B00QQA0H3S - Optimum Nutrition Gold Standard 100% Whey Protein Powder, Naturally Flavored Vanilla, 4.8 Pound (Packaging May Vary)
5. B01NAEHLFO - Garden of Life Sport Certified Grass Fed Clean Whey Protein Isolate, Vanilla, 22.57 Ounce
6. B06XX78LKR - Dymatize ISO 100 Whey Protein Powder with 25g of Hydrolyzed 100% Whey Isolate, Gluten Free, Fast Digesting, 1.6 Pound, Chocolate Peanut Butter, 25.6 Ounce (Pack of 1)
7. B00VZ0IoY8 - Dymatize ISO 100 Whey Protein Powder with 25g of Hydrolyzed 100% Whey Isolate, Gluten Free, Fast Digesting, Brown, Cinnamon Bun, 5 lbs
8. B07BL695D1 - Nutricost Whey Protein Isolate (Strawberry) 5LBS
9. B07L91HBFG - Nutrition Chocolate Milkshake Protein Powder, High Protein, Low Carb, Gluten Free, Soy Free, 1.6 lbs (Pack of 1)
10. B082TTFF87 - KOS Organic Plant Based Protein Powder, Chocolate Peanut Butter - Delicious Vegan Protein Powder - Keto Friendly, Gluten Free, Dairy Free & Soy Free - 1.3 Pounds, 15 Servings

This is an observational study as none of the items were actually purchased and no experiment was conducted.

Data collection

All of the data in this report were scraped from Amazon.com through product ID. The list above shows the mapping between product ID and protein powder. Ten protein powder were select through random searching and powders that author have sampled as well. To verify the reviews replace {ID} in the link below with one of the ids above.

Link: ”

Data Preprocessing

After pulling the html data dates, titles, reviews were all parsed.

```r
df = data.frame()

for (i in dfiles) {
  pdf <-  read.csv(paste0(path, i))

  titles = gather(pdf %>% select(starts_with('review_title')))$value
  reviews = gather(pdf %>% select(starts_with('review_text')))$value
  stars = gather(pdf %>% select(starts_with('review_star')))$value
  dates = gather(pdf %>% select(starts_with('review_date')))$value
  pages = gather(pdf %>% select(starts_with('page')))$value
  ndf <- data.frame(titles=titles, reviews=reviews, stars=stars, dates=dates, pages=pages) %>%
          mutate(stars=as.numeric(str_extract(stars, '\\d')),
                 date=str_remove_all(dates, 'Reviewed in the United States on ') %>% as.Date(format = "%
                 years=as.numeric(format(date, format='%Y')),
                 month=as.numeric(format(date, format='%m')),
                 days=as.numeric(format(date, format='%d')),
                 titles=gsub("[\r\n]", "", titles) %>% str_trim(side='both'),
                 protein=str_split(i, '.csv')[[1]][1],
                 reviews=str_to_lower(reviews) %>% gsub(pattern='[[:punct:] ]+', replace=' ') %>% gsub(p

  df <- bind_rows(df, ndf)
}
```

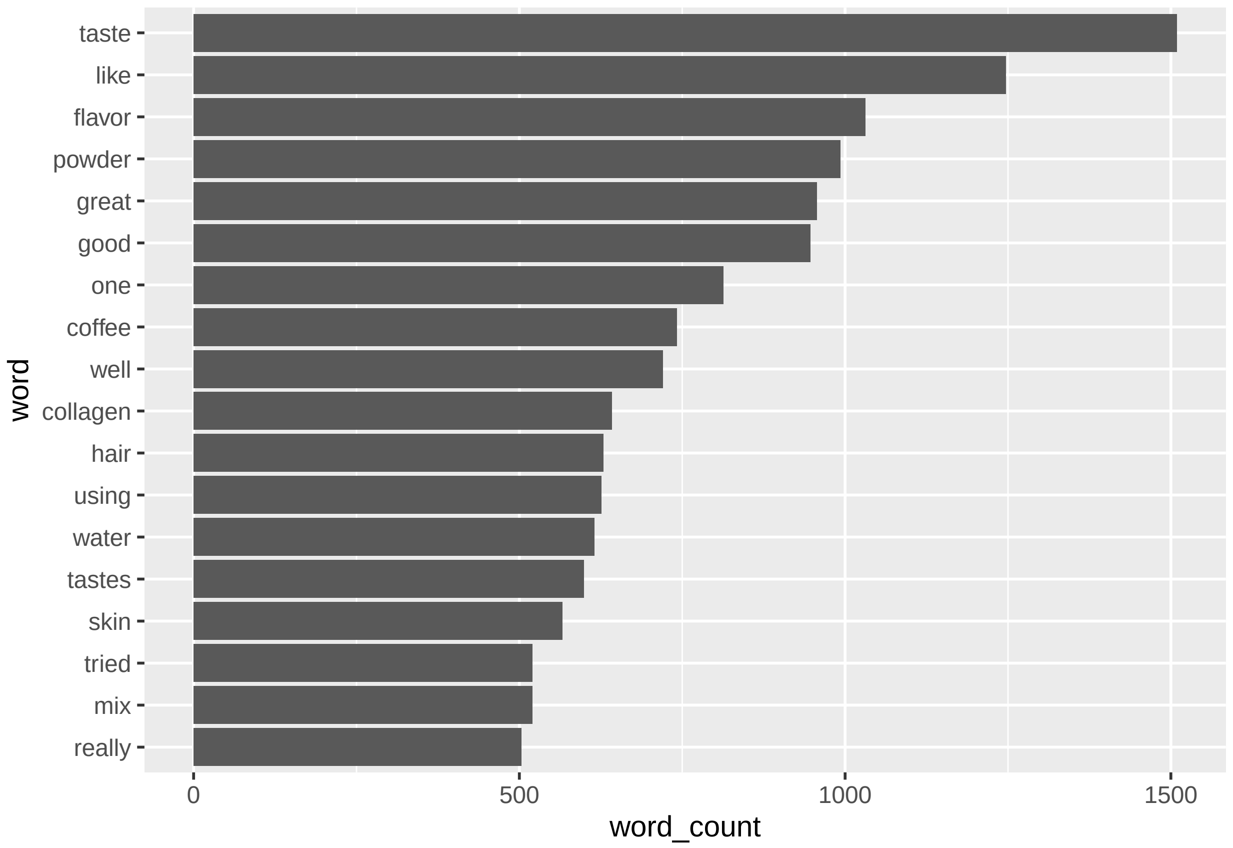
Reviews required some extra cleaning by removing punctuation and stop words. An iterative process of reading reviews, then updating the stopword list, then rerun and reread was conducted to clean up the reviews.

```
stop_word_list <- c(stopwords(), 'aarp', 'ab', 'azo', 'aug', 'asap', 'just', 'now', 'result', 'try', 'ba

frequent_words = c('protein', 'product')
df <- df %>% filter(years > 2016 & years <= 2020) %>%
            mutate(reviews=gsub(pattern='\\b[a-z]\\b{1}', replace=' ', reviews) %>%
                     gsub(pattern='[[:digit:]]+', replace='') %>%
                     removeWords(stop_word_list) %>%
                     str_trim())
```

Exploratory Data Analysis

Sentimental Analysis

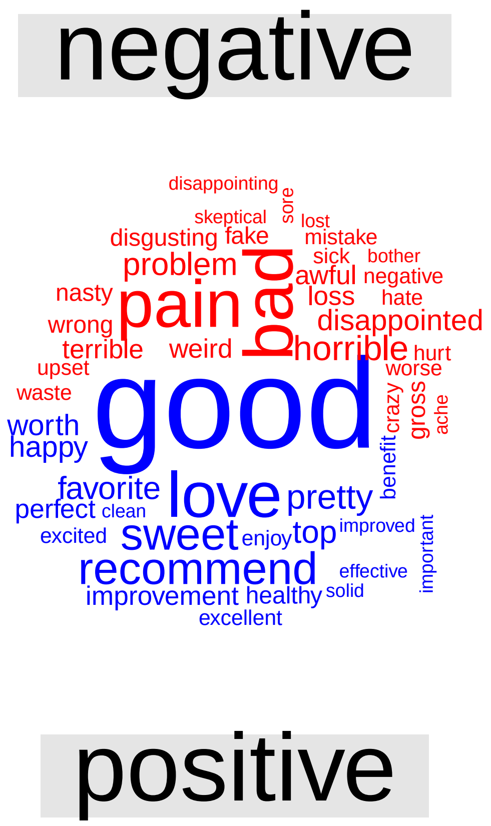
```
words = df %>% unnest_tokens(word, reviews) %>% count(word, name='word_count')

words %>% filter(word_count > 500 & !(word %in% frequent_words)) %>% mutate(word=reorder(word, word_cou
```



Common words like taste and flavor are used the most as expected, unusual words like hair and collagen were also used. After reading some of the *hair* reviews, the users stated how the protein helped their hair.

```
words %>% inner_join(get_sentiments('afinn')) %>%
          inner_join(get_sentiments('nrc')) %>%
          inner_join(get_sentiments('bing')) %>%
          acast(word ~ sentiment, value.var='word_count', fill=0) %>%
  comparison.cloud(colors=c('red', 'blue'), max.words = 50)
```

```
## Joining, by = "word"
## Joining, by = "word"

## Joining, by = c("word", "sentiment")
```



A sentimental analysis was done on the reviews, we see nasty and bad in the negative set and good and perfect in the positive set.

Finally to prepare the data set for analysis we need specific words/phrases that would correspond to the powder having a poor taste or similar. Reviews that contain the *tasteless_words* will be labels as 1 in the *notaste* column else they will be labeled as 0.

```
tasteless_words = c('trash',
                    'disgusting',
                    'horrible',
                    'nasty',
                    'terrible',
                    'abomination',
                    'horrid',
                    'bad',
                    'tasteless',
                    'awful',
                    'flavor awful',
                    'gagging',
                    'flavor horrible',
                    'terrible tasting ',
                    'tasted too disgusting',
                    'tastes disgusting',
                    'taste horrible',
                    'tasted horrible',
                    'taste horrible',
                    'tastes bad',
                    'taste unbearable',
                    'taste like cheap whey protein',
                    'tastes nothing like',
                    'taste isnt great',
                    'worst after taste',
                    'hate taste',
                    'worst tasting protein',
                    'medicinal and artificial taste',
                    'no flavor',
                    'not very good taste')

    tasteless_words = paste(tasteless_words, sep='|', collapse ='|')[1]
```

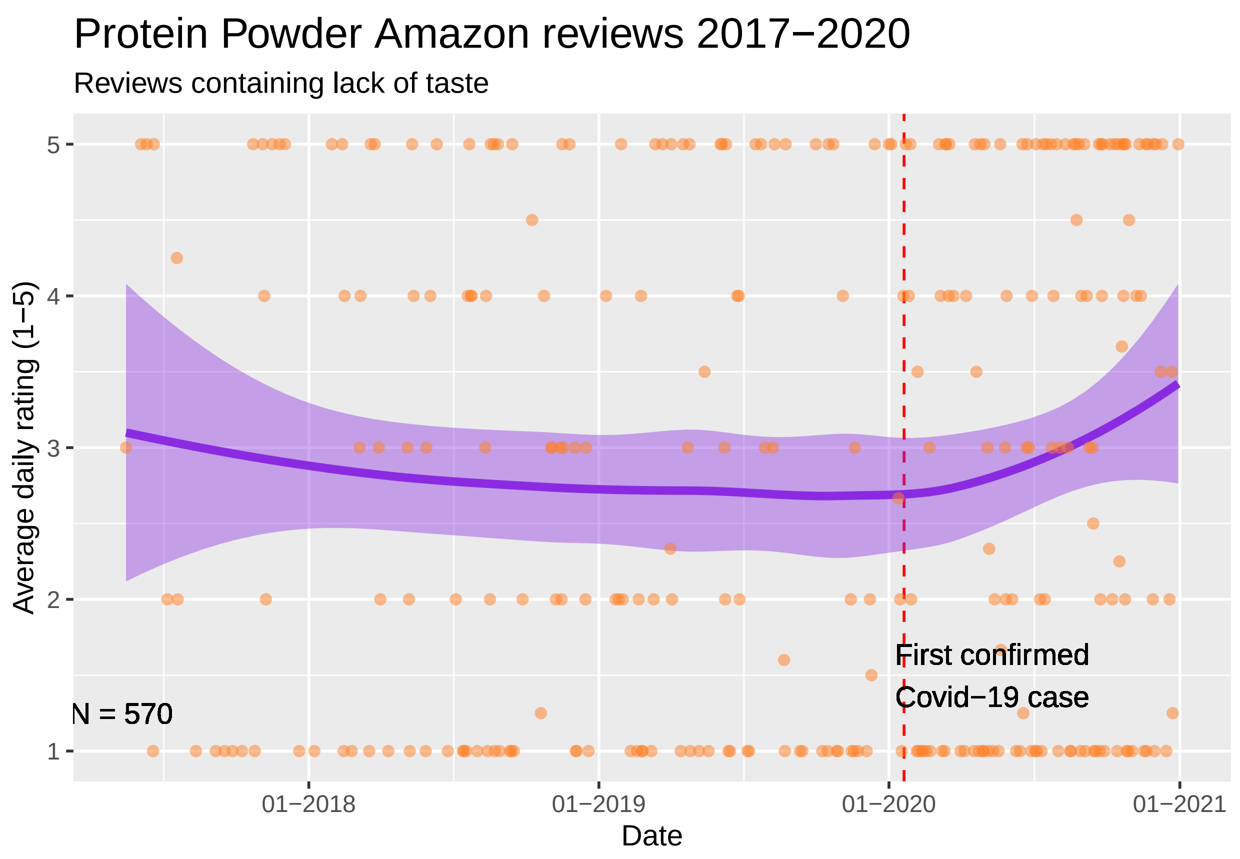
protein=recode(protein, ‘B000QSNYGI’=‘Optimum Double Rich Chocolate’, ‘B00BEOHFKO’= ‘MuscleTech Phase8’, ‘B00NLR1PX0’= ‘Vital Proteins Collagen’, ‘B01NAEHLFO’= ‘Garden of Life’, ‘B00VZ0IoY8’= ‘Dymatize ISO’, ‘B07BL695D1’=‘Nutricost Strawberry’, ‘B07L91HBFG’= ‘Nutrition Chocolate Milkshake’, ‘B082TTFF87’=‘KOS Chocolate Peanut Butter’),

```
df <- df %>% mutate(notaste = ifelse(str_detect(reviews, tasteless_words), 1, 0),
                    covidyear = ifelse(years <2020, 0, 1))
```

Before and After Covid-19

```
df  %>% filter(notaste == 1) %>%
  arrange(date) %>%
  group_by(date) %>%
  summarise(Rating = mean(stars)) %>% ggplot(aes(x = (as.Date(date)), y = Rating)) +
  geom_vline(xintercept = as.numeric(as.Date("2020-01-20")), colour = "red", linetype = "dashed")+
  geom_smooth(method = "loess", size = 1.5, colour = "blueviolet", fill = "blueviolet") +
  geom_point(alpha = 0.5, colour = "chocolate1") +
  geom_text(x=as.Date("2020-05-10"), y=1.5, label='First confirmed\nCovid-19 case', color='black') +
  geom_text(x=as.Date("2017-05-10"), y=1.25, label='N = 570', color='black') +
  labs(x = "Date", y = "Average daily rating (1-5)", title = "Protein Powder Amazon reviews 2017-2020",
  theme(plot.title = element_text(size=16)) +
  scale_x_date(date_labels = "%m-%Y")
```

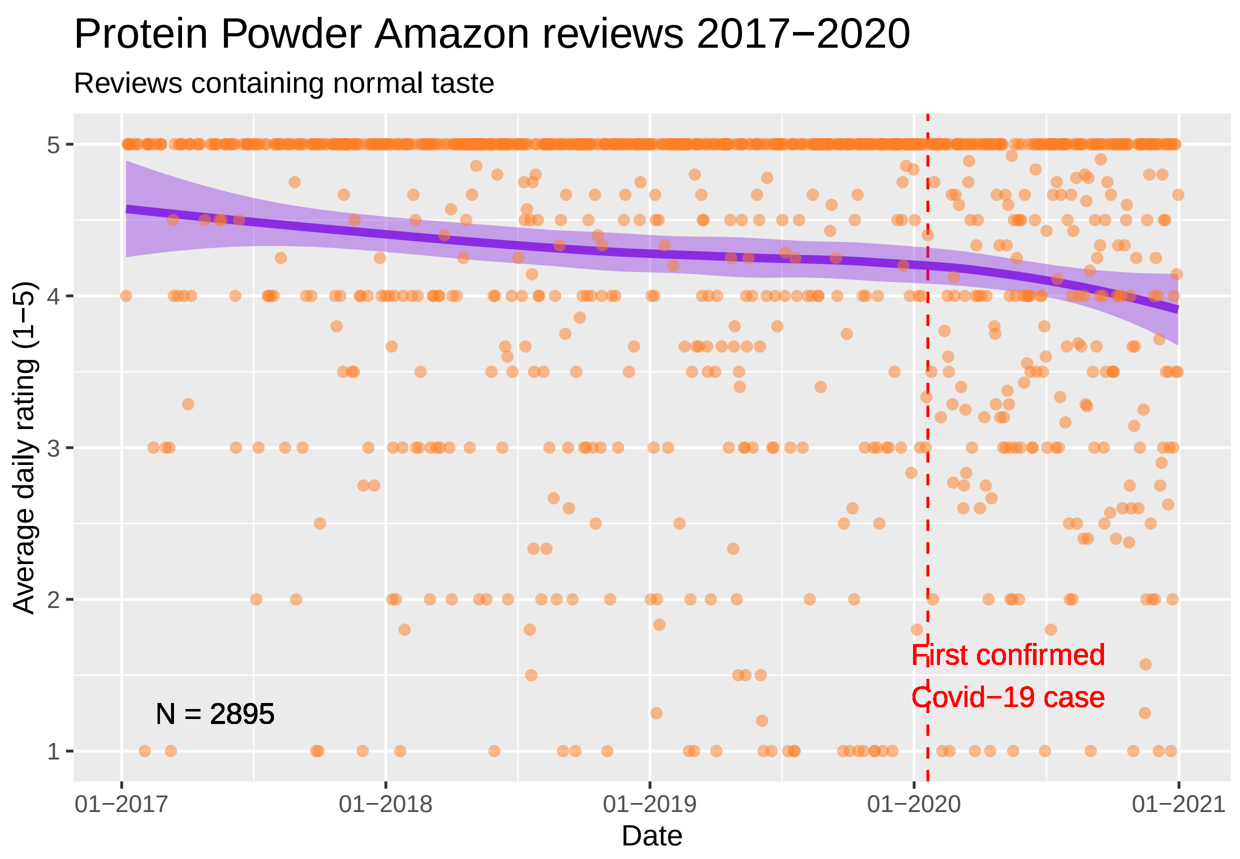
```
## `geom_smooth()` using formula 'y ~ x'
```



Before the first case of Covid-19 there is a the regression line is very normal; however after the first confirmed case the line strangely goes up. That motion states there are more positive reviews and those reviews contain a lot of comments on how the powder is lacking taste.

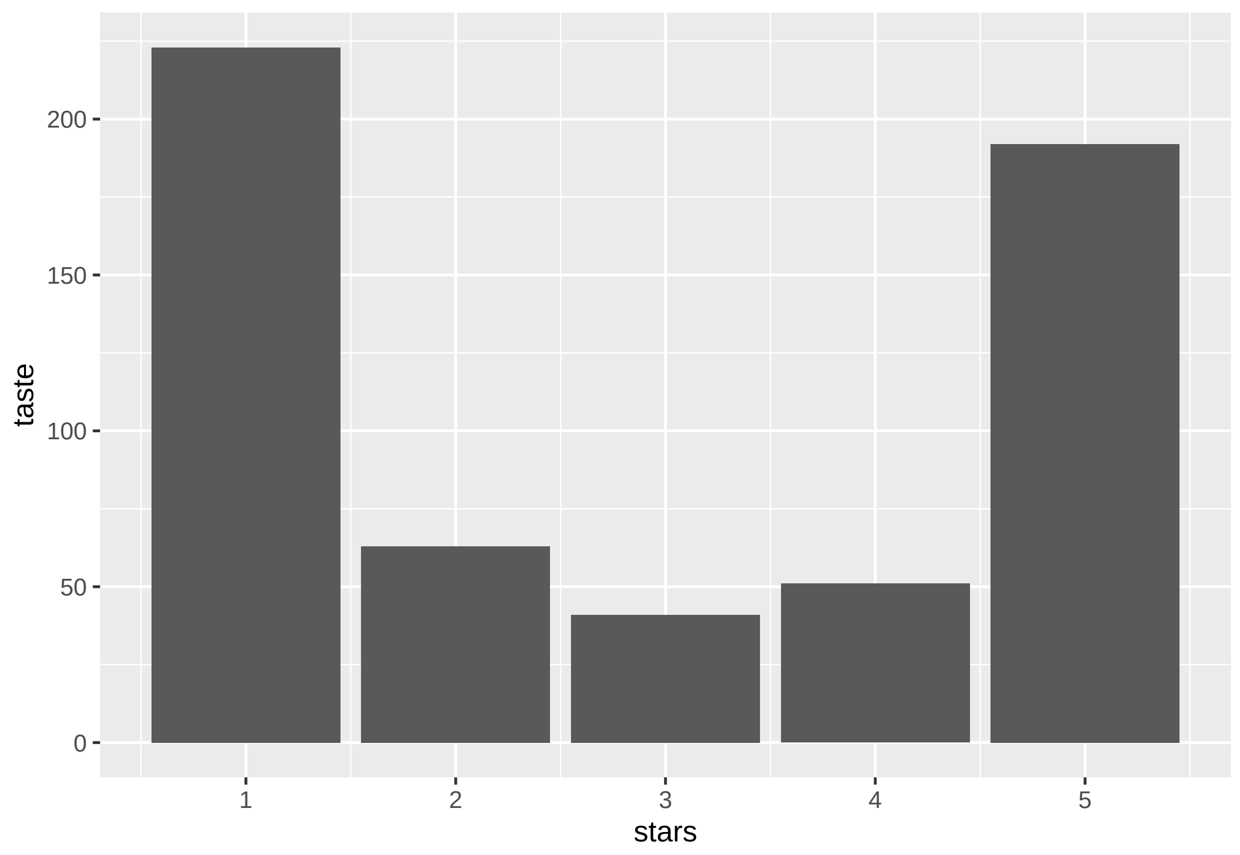
```
df  %>% filter(notaste == 0) %>%
  arrange(date) %>%
  group_by(date) %>%
  summarise(Rating = mean(stars)) %>% ggplot(aes(x = (as.Date(date)), y = Rating)) +
  geom_vline(xintercept = as.numeric(as.Date("2020-01-20")), colour = "red", linetype = "dashed")+
  geom_smooth(method = "loess", size = 1.5, colour = "blueviolet", fill = "blueviolet") +
  geom_point(alpha = 0.5, colour = "chocolate1") +
  geom_text(x=as.Date("2020-05-10"), y=1.5, label='First confirmed\nCovid-19 case', color='red') +
  geom_text(x=as.Date("2017-05-10"), y=1.25, label='N = 2895', color='black') +
  labs(x = "Date", y = "Average daily rating (1-5)", title = "Protein Powder Amazon reviews 2017-2020",
  theme(plot.title = element_text(size=16)) +
  scale_x_date(date_labels = "%m-%Y")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

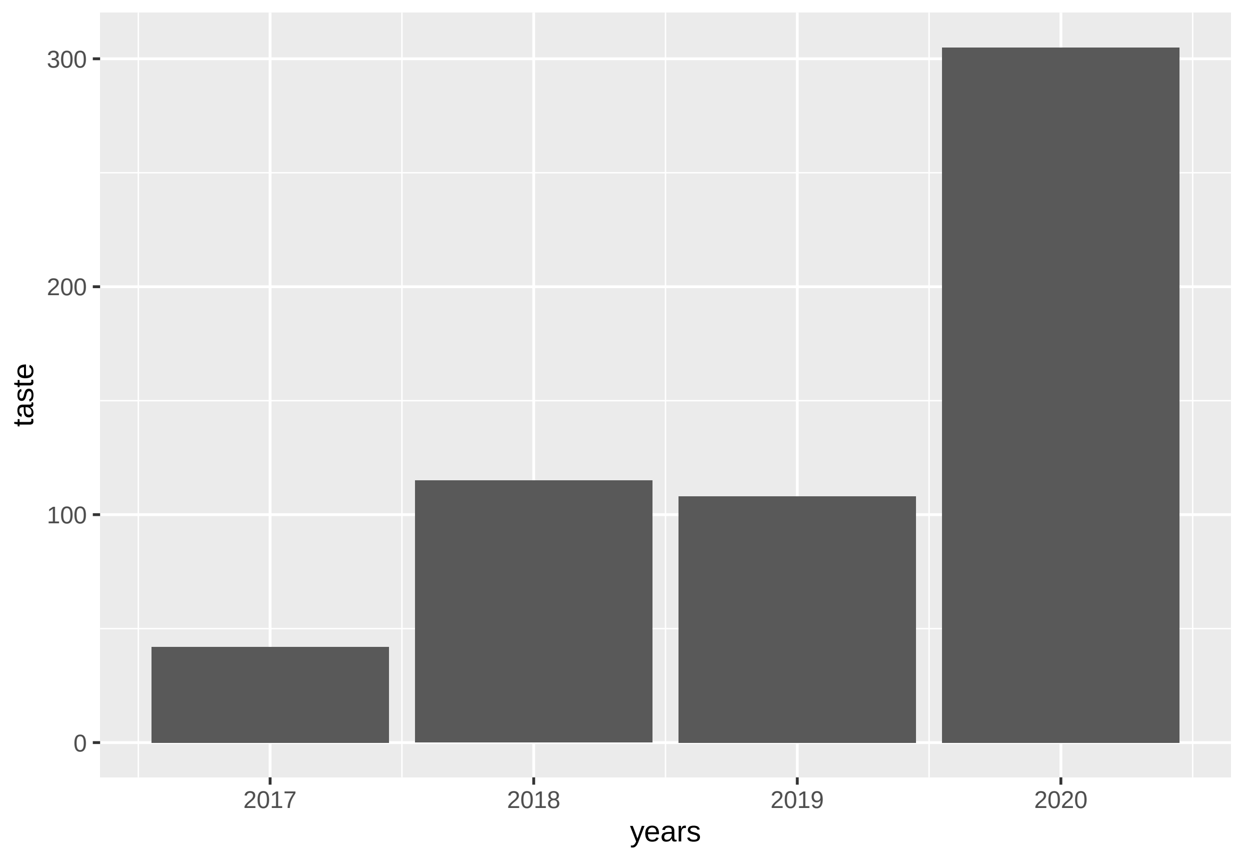


The rating distribution for sample that have *notaste* is inline with what we would expect from normal reviews. Those who really like the powder and really hate it will give a review, while those in the middle (2,3,4) are less likely to review.

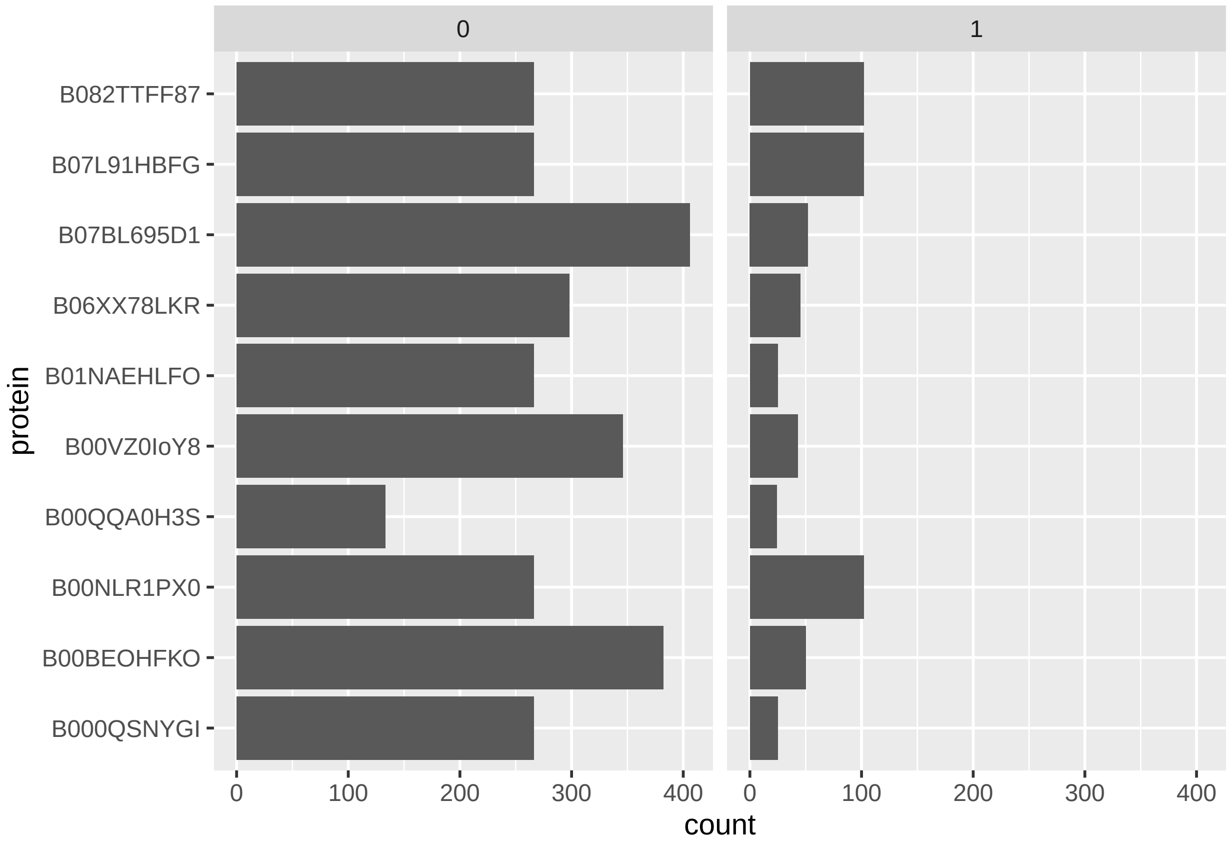
```
df %>% group_by(stars) %>% summarise(taste=sum(notaste)) %>% ggplot(aes(x=stars, y=taste)) + geom_col()
```



```
df %>% group_by(years) %>% summarise(taste=sum(notaste)) %>% ggplot(aes(x=years, y=taste)) + geom_col()
```

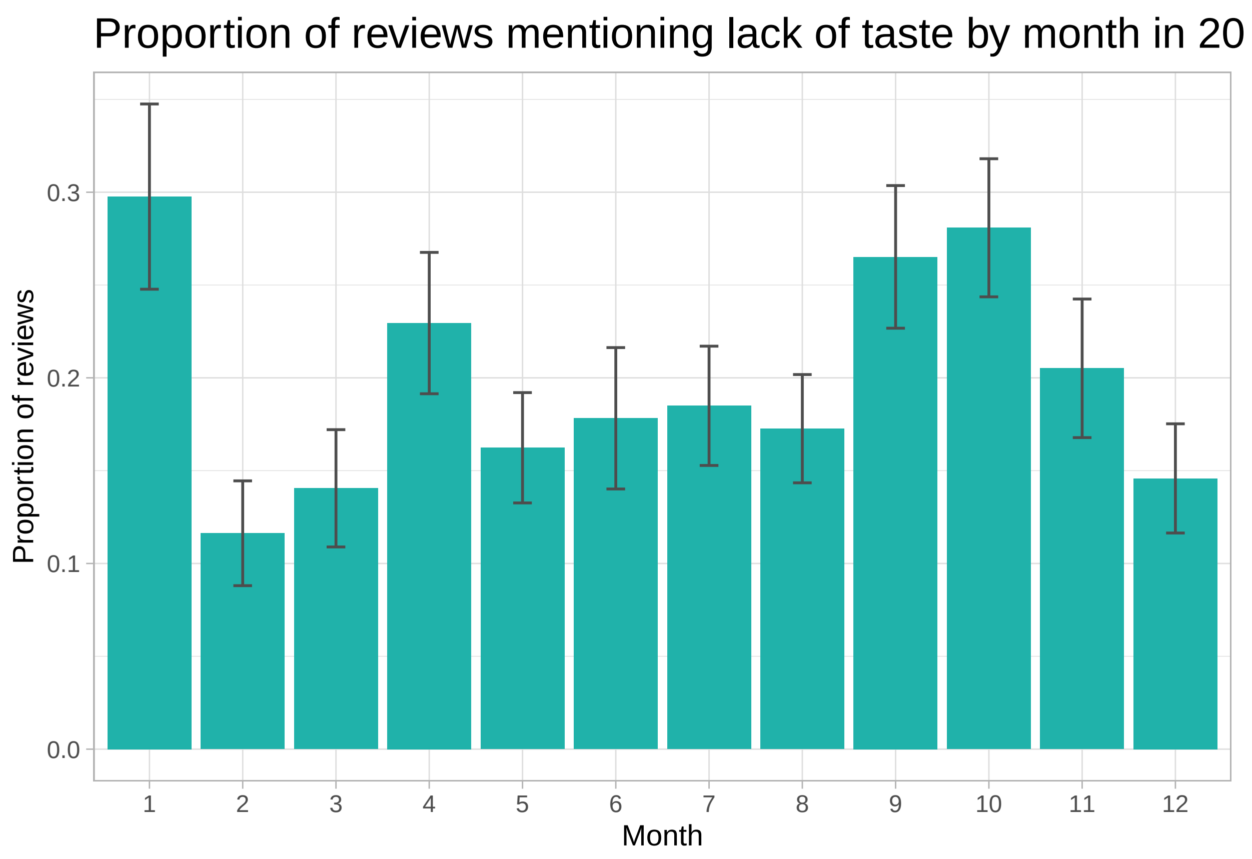


```
df %>% ggplot(aes(x=protein)) + geom_bar() + coord_flip() + facet_wrap(~notaste)
```



```
df %>%
  filter(years == 2020) %>%
  group_by(month) %>%
  add_tally() %>%
  summarise(n =n, notaste = sum(notaste)) %>%
  mutate(nsprop = notaste/n) %>%
  mutate(se = sqrt((nsprop*(1-nsprop))/n)) %>%
  summarise(n=mean(n), se=mean(se), nsprop=mean(nsprop)) %>%
  ggplot(aes(x=as.factor(month), y = nsprop, group = month))+
  geom_bar(stat = "identity", fill = "lightseagreen")+
  geom_errorbar(aes(ymin = (nsprop-se), ymax = (nsprop+se)), width=0.2, colour = "gray30")+
  labs(x = "Month", y = "Proportion of reviews", title = "Proportion of reviews mentioning lack of taste
  theme_light()+
  theme(plot.title = element_text(size=16))
```

```
## `summarise()` has grouped output by 'month'. You can override using the `.groups` argument.
```



Use reviews to predict if there is taste or no taste

Hypothesis testing

$$H_o : \mu_{tasteless-before-2020} = \mu_{tasteless-after-2020} Ha : \mu_{tasteless-before-2020} \neq \mu_{tasteless-after-2020}$$

Type 1 Error : Rejecting the null hypothesis when H_o is actually true

Type 2 Error : Failing to reject the null hypothesis when H_a is actually true

Before running any analysis we must agree on an alpha 0.01.

Power analysis

To avoid making a type 1 error we conduct a power analysis

We will use power=0.8, meaning there is an 80% probability that we correctly reject H_o and a threshold for significance $\alpha = 0.05$

```
pre_2020 <- df %>%
  filter(years < 2020) %>%
  count(notaste) %>%
  mutate(prop=n/sum(n))

p = pre_2020$prop[2]
n = sum(pre_2020$n)
```

```
post_2020 <- df %>%
  filter(years == 2020) %>%
  count(notaste) %>%
  mutate(prop=n/sum(n))

p = post_2020$prop[2]
n = sum(post_2020$n)
```

Given pre_stdev = 13.9% and post_stdev = 19.5 a sample size of 1390 will be used to achieve the desired parameters.

T-test

A t-test is a type of inferential statistic used to determine if there is a significant difference between the means of two groups, which may be related in certain features.

```
cdf <- df %>% group_by(covidyear) %>% count(notaste) %>% mutate(notaste=ifelse(notaste == 0, 'Tastefull

df_table = cdf %>% add_row(covidyear='Total', Tastefull=sum(cdf$Tastefull), Tasteless=sum(cdf$Tasteless
```

```
p_pool = df_table %>% filter(covidyear != 'Total') %>% mutate(p_pool=sum(Tasteless)/sum(Total)) %>% dis
total = df_table[3,4]

n1 = df_table %>% filter(covidyear == 'Before Covid') %>% last()
n2 = df_table %>% filter(covidyear == 'During Covid') %>% last()

p1 = df_table %>% mutate(prop=Tasteless/ Total) %>% filter(covidyear == 'Before Covid') %>% last()
p2 = df_table %>% mutate(prop=Tasteless/ Total) %>% filter(covidyear == 'During Covid') %>% last()

pdiff = p1 - p2
```

Success-failure condition

```
n *(p_pool * (1 - p_pool))
```

```
##    p_pool
## 1 214.958
```

There should be at least 10 expected successes and 10 expected failures in a sample in order to use the normal distribution as an approximation. $n*(1-p) \geq 10$. There are 3465 samples and 0.164502164502165 expected successes. Substituting gives 214 which is well above the required amount to use a normal distribution.

Independence test

Another test is the assumption that the data came from a simple random sample and collected from a representative, randomly selected portion of the total population. I see no reason to believe the reviews have any meaningful influence with one another; however there is no way to prove that.

Confidence interval

```r
z = 1.96
se = sqrt(p1*(1-p1)/n1 + p1*(1-p2)/n2)
lower_conf <- pdiff - z * se
upper_conf <- pdiff + z * se

me1 = sqrt(p1*(1-p1)/n1)
me2 = sqrt(p2*(1-p2)/n2)
```

We can say with 95% confidence that a sample of 3465 before 2020 would have between -0.0783735–0.0328514 with 0.007944% error for before covid and 0.0100186 during covid.

```r
se = sqrt(p_pool * (1- p_pool) / n1 + p_pool * (1- p_pool) / n2)
pe = df_table %>% filter(covidyear != 'Total') %>% mutate(prop=Tasteless/Total) %>% pull(prop) %>% diff

z = (pe/se)$p_pool
pval <- (1 - pnorm(z)) * 2
```

The pval for the z-test is 1.1122429×10^{-5} which is very small

Chi-squared tests

The Chi-square goodness of fit test checks whether your sample data is likely to be from a specific theoretical distribution. We have a set of data values, and an idea about how the data values are distributed. The test gives us a way to decide if the data values have a "good enough" fit to our H_a, or if the H_o cannot be rejected.

$$X^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

```r
df_table %>%  filter(covidyear != 'Total') %>% select(Tastefull, Tasteless) %>% chisq.test(correct=F)
```

```
##
##  Pearson's Chi-squared test
##
## data:  .
## X-squared = 19.308, df = 1, p-value = 1.112e-05
```

With an X^2 of 19.3 that corresponds to a p-value of 0.0000112 which is well beyond on alpha of 0.01 The X^2 test further in forces our approach to accept the alternative of reject the null hypothesis

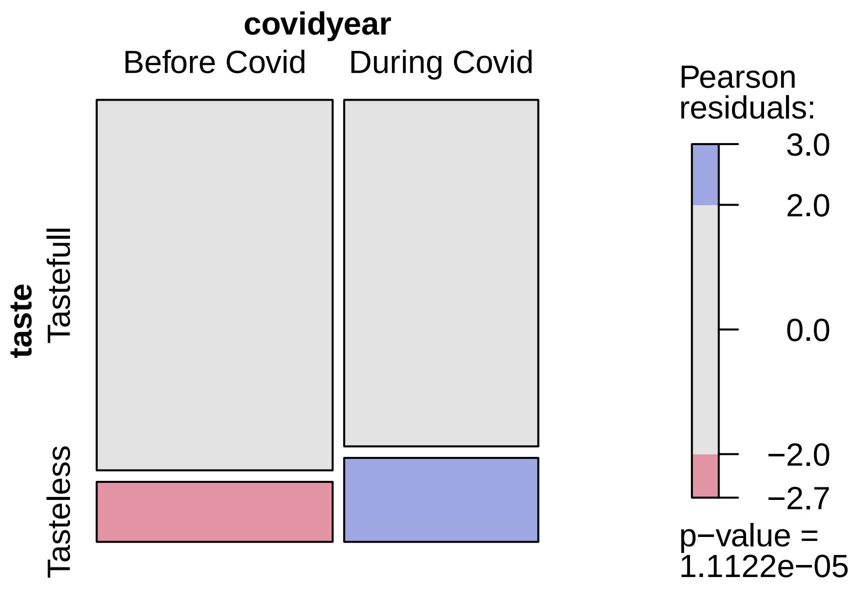
```r
library(vcd)
```

```
## Loading required package: grid
```

```r
dat <- df %>% mutate(covidyear=ifelse(covidyear == 0, 'Before Covid', 'During Covid'), taste=ifelse(nota
```

```
## Rows: 3,465
## Columns: 2
## $ covidyear <chr> "During Covid", "Before Covid", "Before Covid", "Before Covi~
## $ taste     <chr> "Tasteless", "Tasteless", "Tastefull", "Tastefull", "Tastefu~
```

```r
mosaic(~ covidyear + taste,
  direction = c("v", "h"),
  data = dat,
  shade = TRUE
)
```


covidyear
Before Covid
During Covid
taste
Tastefull
Tasteless
Pearson
residuals:
p−value =
1.1122e−05