Tidy tueday analysis

thank you to liza bolton on running the tidy tuesday tutorial through the u of t IssC, some of the code below belonged to her.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages ----------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.0     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
```

```
## Warning: package 'tibble' was built under R version 3.6.3
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'stringr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

```
## -- Conflicts -------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 3.6.3
```

```
library("tm")
```

```
## Warning: package 'tm' was built under R version 3.6.3
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library("SnowballC")
library("wordcloud")
```

```
## Warning: package 'wordcloud' was built under R version 3.6.3

## Loading required package: RColorBrewer
```

```
library("RColorBrewer")
library(ggwordcloud)
```

```
## Warning: package 'ggwordcloud' was built under R version 3.6.3
```

```
critic <- readr::read_tsv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/202
```

```
## Parsed with column specification:
## cols(
##   grade = col_double(),
##   publication = col_character(),
##   text = col_character(),
##   date = col_date(format = "")
## )
```

```
user_reviews <- readr::read_tsv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/da
```

```
## Parsed with column specification:
## cols(
##   grade = col_double(),
##   user_name = col_character(),
##   text = col_character(),
##   date = col_date(format = "")
## )
```

```
items <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020
```

```
## Parsed with column specification:
## cols(
##   num_id = col_double(),
##   id = col_character(),
##   name = col_character(),
##   category = col_character(),
##   orderable = col_logical(),
##   sell_value = col_double(),
##   sell_currency = col_character(),
##   buy_value = col_double(),
##   buy_currency = col_character(),
##   sources = col_character(),
##   customizable = col_logical(),
##   recipe = col_double(),
##   recipe_id = col_character(),
##   games_id = col_character(),
##   id_full = col_character(),
##   image_url = col_character()
## )
```

```
## Warning: 2 parsing failures.
##  row          col           expected actual
## 4472 customizable 1/0/T/F/TRUE/FALSE    Yes 'https://raw.githubusercontent.com/rfordatascience/tidytu
## 4473 customizable 1/0/T/F/TRUE/FALSE    Yes 'https://raw.githubusercontent.com/rfordatascience/tidytu
```

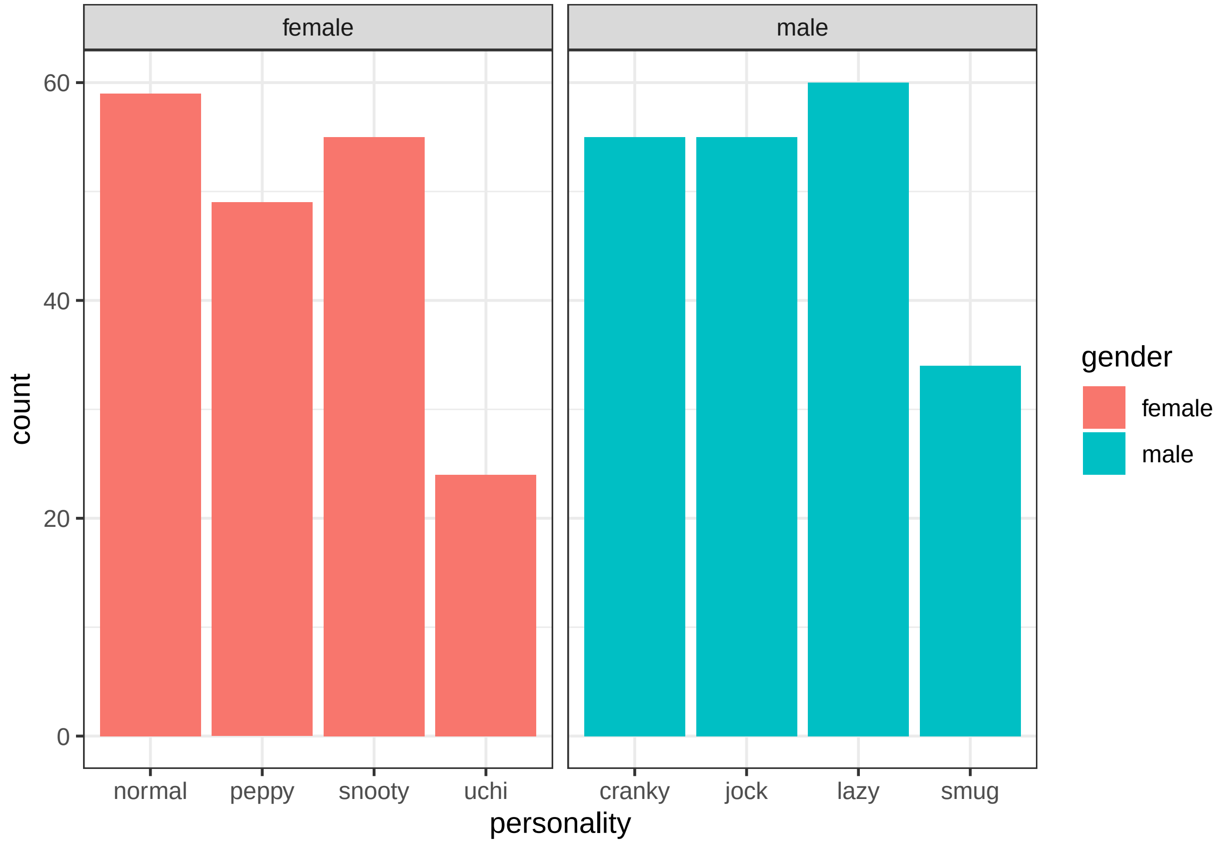
```
villagers <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/
```

```
## Parsed with column specification:
## cols(
##   row_n = col_double(),
##   id = col_character(),
##   name = col_character(),
##   gender = col_character(),
##   species = col_character(),
##   birthday = col_character(),
##   personality = col_character(),
##   song = col_character(),
##   phrase = col_character(),
##   full_id = col_character(),
##   url = col_character()
## )
```

```
villagers %>%
group_by(personality, gender) %>%
summarise(n = n()) %>%
arrange(desc(n))
```

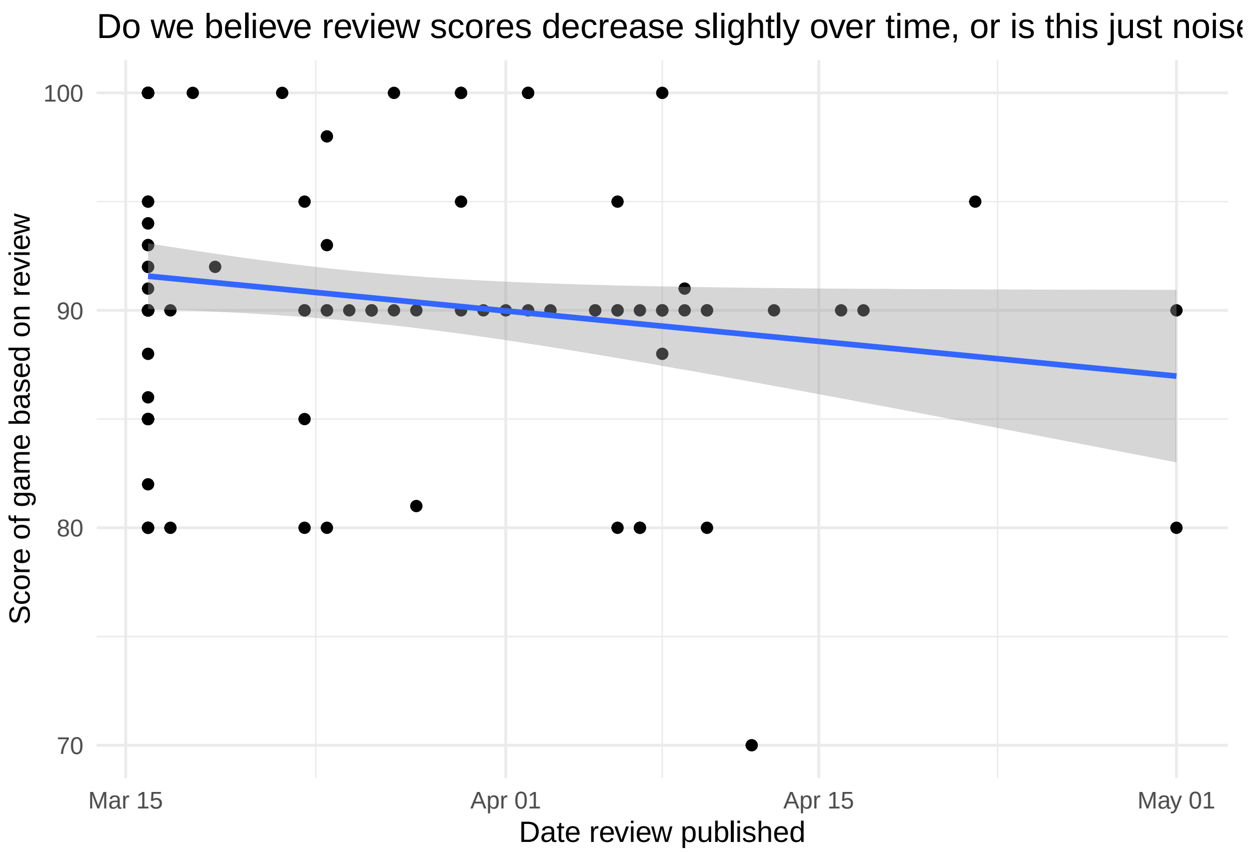
```
## # A tibble: 8 x 3
## # Groups:   personality [8]
##   personality gender     n
##   <chr>       <chr>  <int>
## 1 lazy        male      60
## 2 normal      female    59
## 3 cranky      male      55
## 4 jock        male      55
## 5 snooty      female    55
## 6 peppy       female    49
## 7 smug        male      34
## 8 uchi        female    24
```

```
villagers %>%
group_by(personality, gender) %>%
ggplot(aes(x = personality, fill = gender)) +
facet_wrap(~gender, scales = "free_x") + #great function for splitting out plots on a categorical varia
geom_bar() +
theme_bw() #change the look of a plot really quickly with different theme options
```



```
critic %>%
ggplot(aes(x = date, y = grade)) +
geom_point() +
geom_smooth(method = "lm") +
theme_minimal() +
ggtitle("Do we believe review scores decrease slightly over time, or is this just noise?") +
xlab("Date review published") +
ylab("Score of game based on review")
```

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

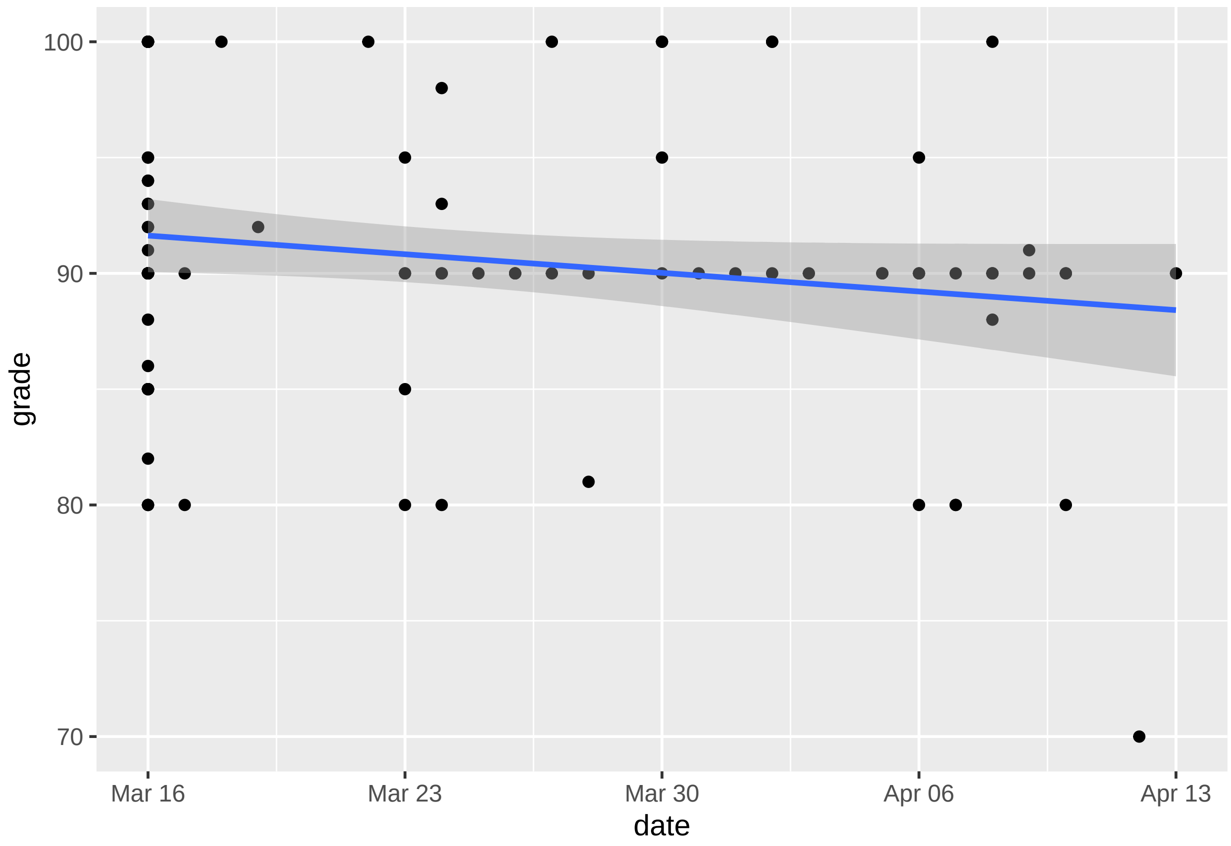


```
summary(lm(grade ~ date, data = critic))
```

```
##
## Call:
## lm(formula = grade ~ date, data = critic)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.8750  -1.5732  -0.0742   2.7253  10.7253
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1924.07418  958.94999   2.006   0.0474 *
## date          -0.09993    0.05227  -1.912   0.0586 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.039 on 105 degrees of freedom
## Multiple R-squared:  0.03364,    Adjusted R-squared:  0.02444
## F-statistic: 3.655 on 1 and 105 DF,  p-value: 0.05861
```

```
critic_restricted <- critic %>%
  filter(date < "2020-04-16")
critic_restricted %>%
  ggplot(aes(x = date, y = grade)) +
  geom_point() +
  geom_smooth(method = "lm")
```

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

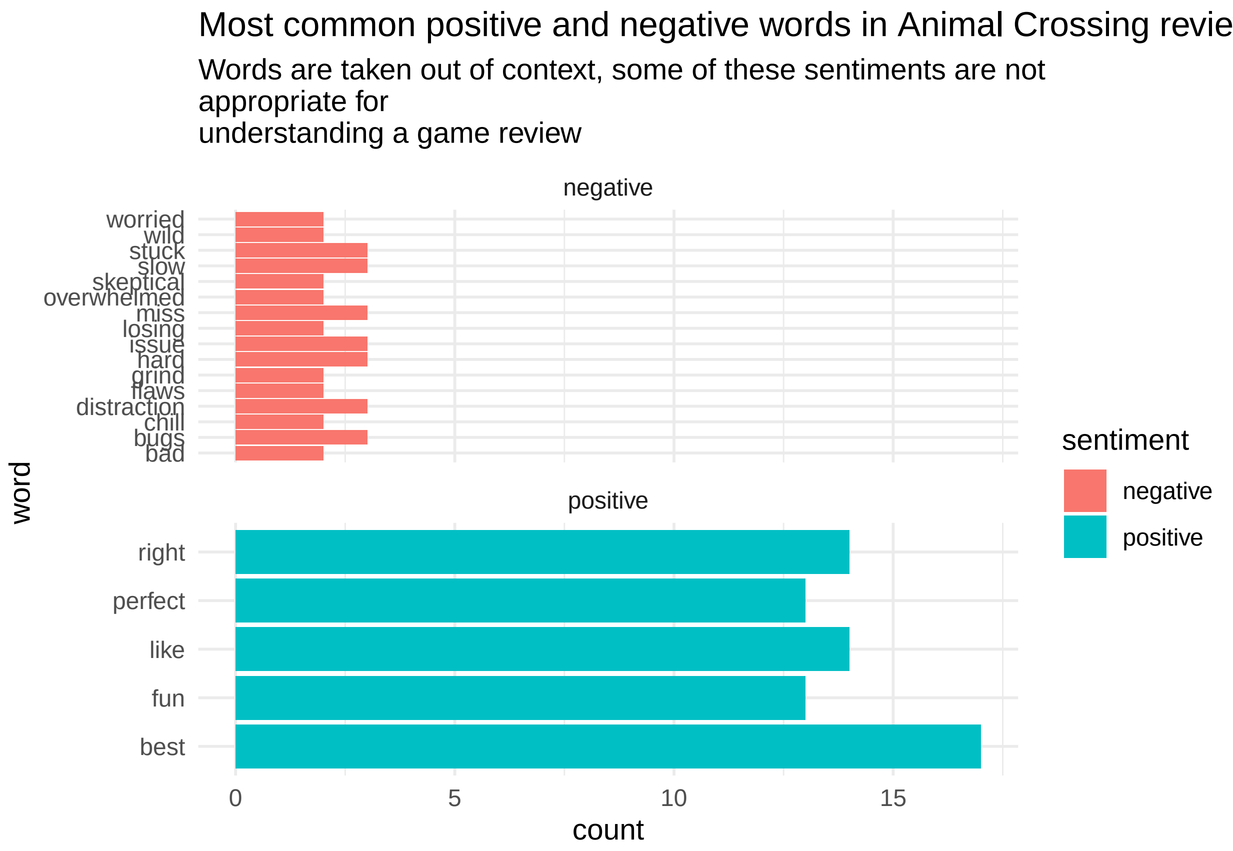


```
summary(lm(grade ~ date, data = critic_restricted))
```

```
##
## Call:
## lm(formula = grade ~ date, data = critic_restricted)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.5276  -1.6306  -0.1365   2.3694  11.0127
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2199.02983 1197.17899   1.837   0.0692 .
## date          -0.11493    0.06526  -1.761   0.0813 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.094 on 100 degrees of freedom
## Multiple R-squared:  0.03008,    Adjusted R-squared:  0.02038
## F-statistic: 3.101 on 1 and 100 DF,  p-value: 0.08128
```

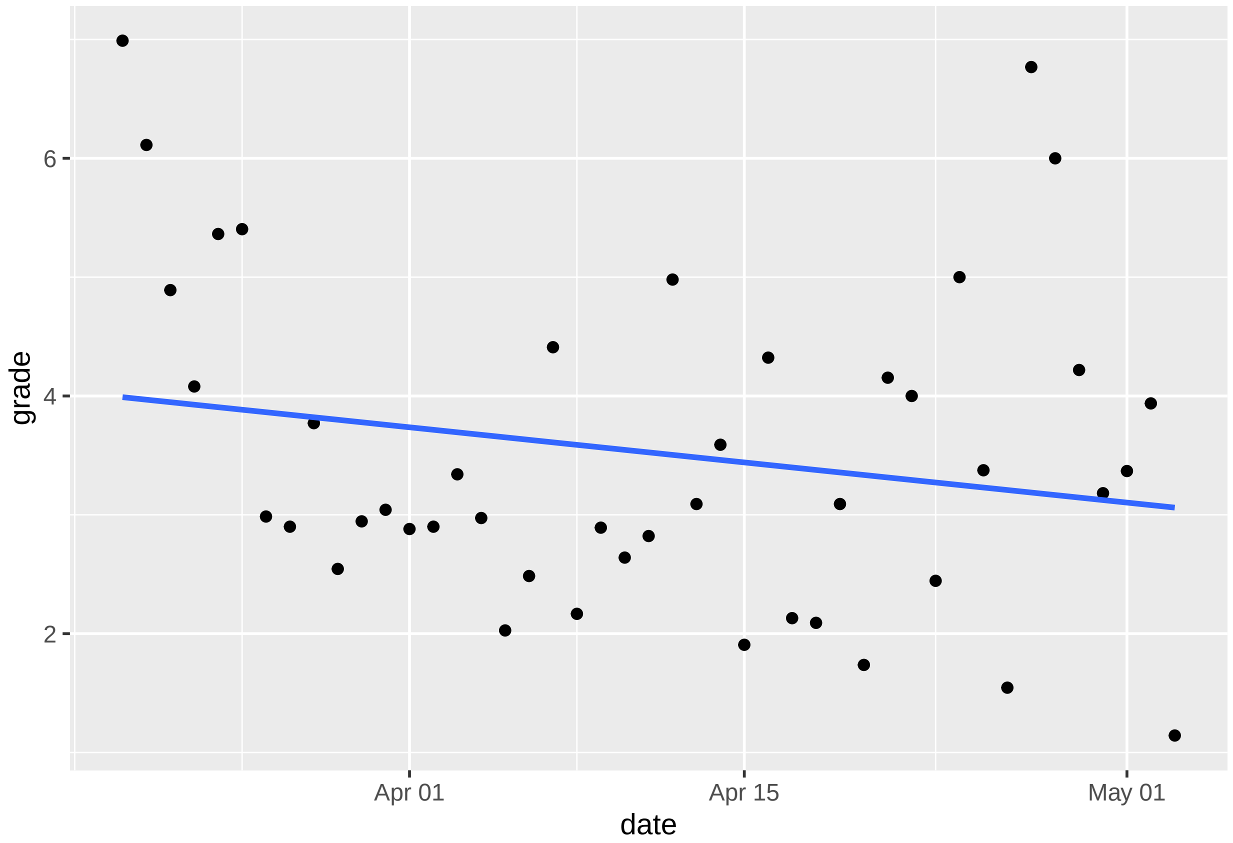
```
bing <- get_sentiments("bing")
```

```
critic %>%
select(text) %>%
unnest_tokens(word, text) %>%
#group_by(word) %>%
#summarise(count = n()) %>%
left_join(bing, by="word") %>%
filter(!is.na(sentiment)) %>%
group_by(word, sentiment) %>%
summarise(count = n()) %>%
filter(count>1) %>% # filter to words appearing more than once (and a sentiment score)
arrange(desc(count)) %>%
group_by(sentiment) %>%
filter(count > max(count) - 5) %>% # get the top couple words of each sentiment
ggplot(aes(x = word, y = count, fill = sentiment)) +
geom_bar(stat = "identity") + #to just use the count var for the height of the bars
coord_flip() +
facet_wrap(~sentiment, nrow = 2, scales = "free_y") + # this drops the unused levels
theme_minimal() +
ggtitle("Most common positive and negative words in Animal Crossing reviews",
subtitle = "Words are taken out of context, some of these sentiments are not
appropriate for\nunderstanding a game review")
```



```r
df <- user_reviews %>% group_by(date) %>% summarise(grade = mean(grade))
df %>%
  ggplot(aes(x = date, y = grade)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

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



```r
summary(lm(grade ~ date, data = df))$coefficients
```

```
##                 Estimate   Std. Error   t value  Pr(>|t|)
## (Intercept) 391.34429713 286.44589313  1.366207 0.1789768
## date         -0.02111959   0.01559908 -1.353900 0.1828418
```

```r
items %>% filter(sell_value > buy_value) %>% group_by(name) %>% summarise()
```

```
## # A tibble: 15 x 1
##    name
##    <chr>
##  1 Nook Inc. Aloha Shirt
##  2 Nook Inc. Bandanna
##  3 Nook Inc. Blouson
##  4 Nook Inc. Botanical Rug
##  5 Nook Inc. Cap
##  6 Nook Inc. Eye Mask
##  7 Nook Inc. Flooring
##  8 Nook Inc. Knapsack
##  9 Nook Inc. Rug
## 10 Nook Inc. Slippers
## 11 Nook Inc. Socks
## 12 Nook Inc. Tee
## 13 Nook Inc. Uchiwa Fan
## 14 Nook Inc. Umbrella
## 15 Nook Inc. Wall
```

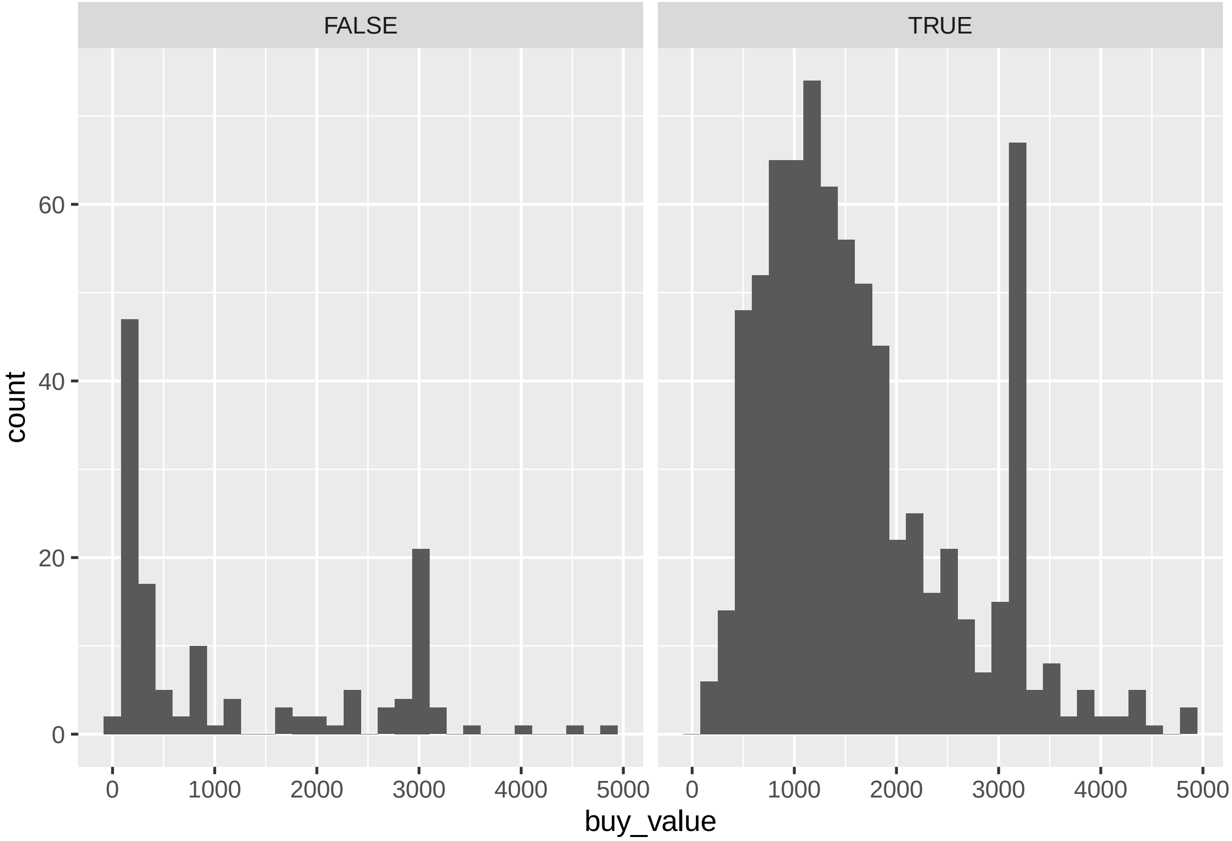
these are the itemse that have a higher sell value than a buy value, that you can make money off of by selling

```
items %>% filter(!is.na(orderable), !is.na(buy_value)) %>% group_by(orderable)%>%summarise(mean = mean(
                                                                                           standard_dev
```

```
## # A tibble: 2 x 4
##   orderable  mean median standard_deviation
##   <lgl>     <dbl>  <dbl>              <dbl>
## 1 FALSE     8602.   2620             16352.
## 2 TRUE      4081.   1500             36097.
```

the average buy value of things that are orderable are more

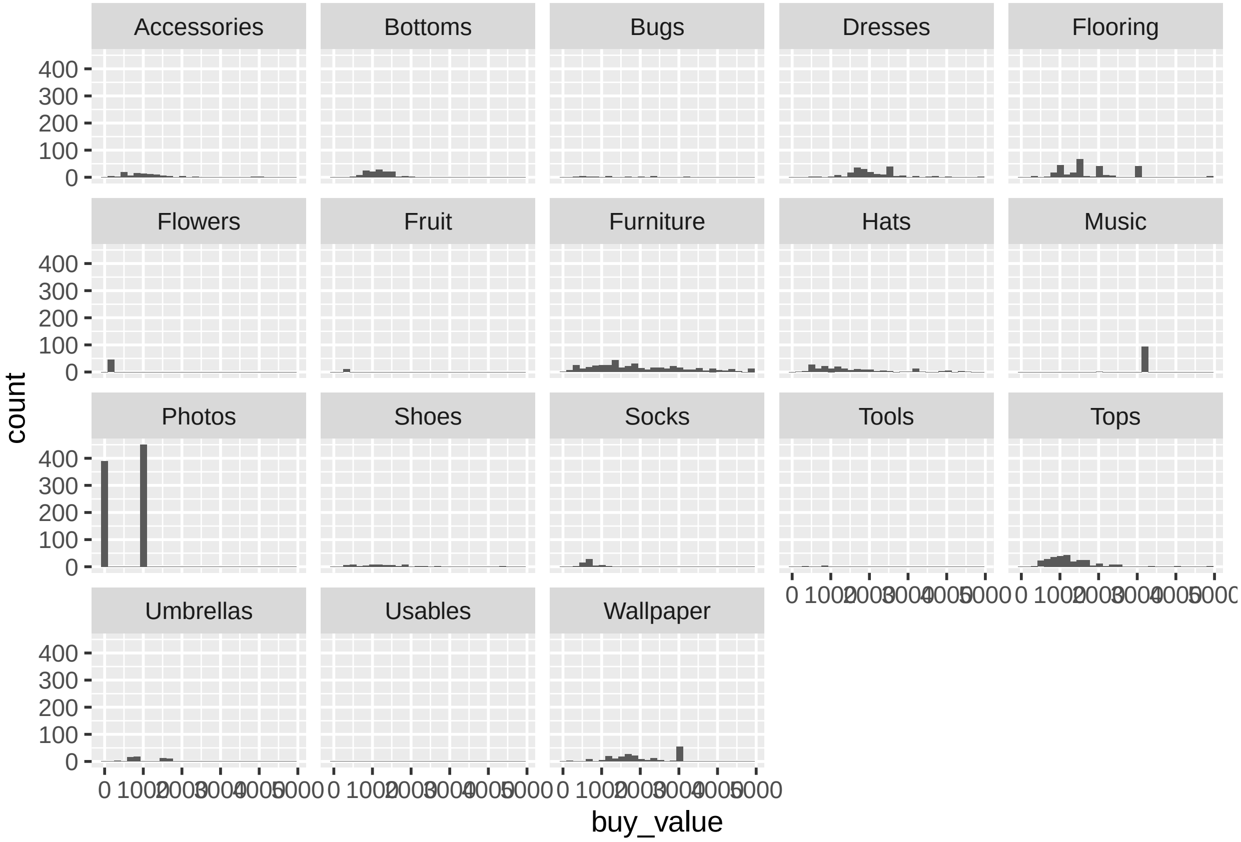
```
items %>% filter(!is.na(orderable), !is.na(buy_value), buy_value < 5000) %>%  ggplot(aes(x = buy_value)
```



the distribution of the orderable is focused at around 1000 currencys for prices below 5000, while the prices for non-orderable are much more unevenly distributed

```
items %>%
  filter(!is.na(buy_value), buy_value < 5000) %>%
  ggplot(aes(x = buy_value)) +
  geom_histogram() +
  facet_wrap(~category)
```

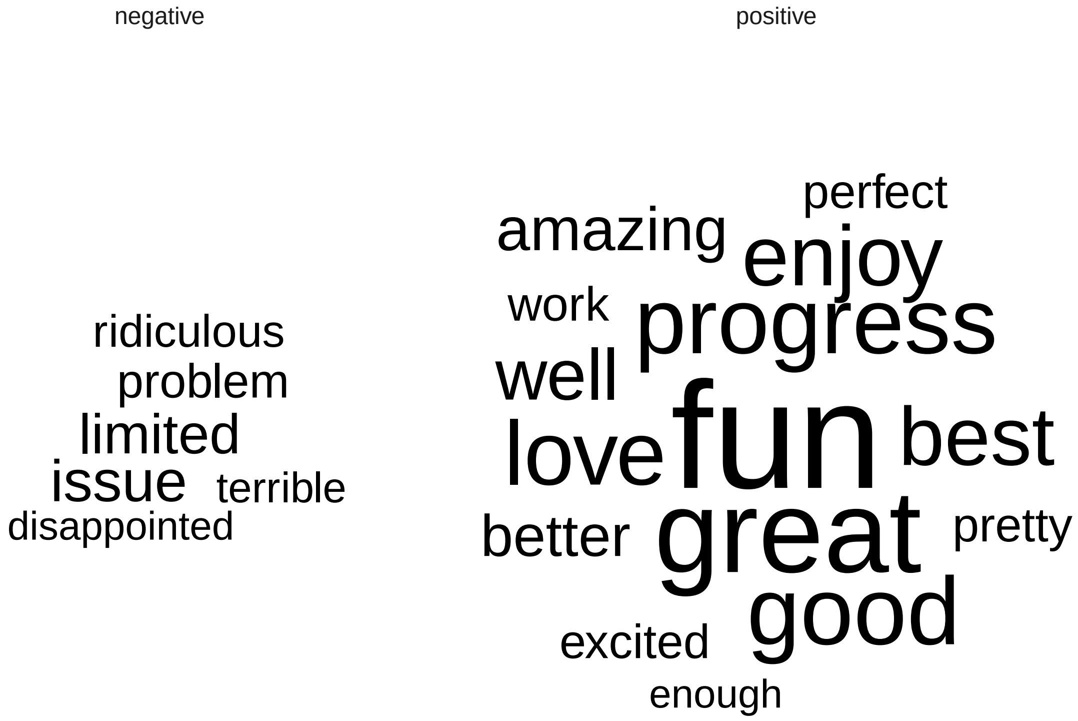
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
bing2 <- get_sentiments("bing")
```

```
 user_reviews %>%
select(text) %>%
unnest_tokens(word, text) %>%
#group_by(word) %>%
#summarise(count = n()) %>%
left_join(bing2, by="word") %>%
filter(!is.na(sentiment)) %>%
group_by(word, sentiment) %>%
summarise(count = n()) %>%
filter(count>1) %>% # filter to words appearing more than once (and a sentiment score)
arrange(desc(count)) %>%
```

```r
filter(word != "like", word != "bad", count > 150) %>%   # get the top couple words of each sentiment
  ggplot(aes(label = word, size = count)) +
  geom_text_wordcloud_area() +
  scale_size_area(max_size = 20) +
  theme_minimal() +
  facet_wrap(~sentiment)
```



```r
#
# ggplot(aes(x = word, y = count, fill = sentiment)) +
# geom_bar(stat = "identity") + #to just use the count var for the height of the bars
# coord_flip() +
# facet_wrap(~sentiment, nrow = 2, scales = "free_y") + # this drops the unused levels
# theme_minimal() +
# ggtitle("Most common positive and negative words in Animal Crossing user reviews",
# subtitle = "Words are taken out of context, some of these sentiments are not
# appropriate for\nunderstanding a game review")
```