423 project

Chris Chen

1/20/2022

Preliminaries

```r
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.6     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   2.1.0     ✓ forcats 0.5.1
```

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

```r
library(expm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
##
## Attaching package: 'expm'
```

```
## The following object is masked from 'package:Matrix':
##
##     expm
```

```
library(ggplot2)
library(lmvar)
library(leaps)
library(RColorBrewer)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Dataset

```
perfume = read.csv("noon_perfumes_dataset.csv")
sum(is.na(perfume))
```

```
## [1] 0
```

```
head(perfume)
```

	X	brand	name	old_price	new_price	ml	concentration	departme
	<int>	<chr>	<chr>	<dbl>	<dbl>	<int>	<chr>	<chr>
1	0	PACO RABANNE	1 Million Lucky	395	244.55	100	EDT	Men
2	1	Roberto Cavalli	Paradiso Assoluto	415	107.95	50	EDP	Women
3	2	S.T.Dupont	Royal Amber	265	186.90	100	EDP	Unisex
4	3	GUESS	Seductive Blue	290	103.20	100	EDT	Men
5	4	Roberto Cavalli	Uomo	260	94.95	50	EDP	Women
6	5	Roberto Cavalli	cavalli	260	94.95	50	EDP	Women

6 rows | 1-10 of 16 columns

no empty value. good.

```
perfume = perfume %>%
  mutate(scent = ifelse(scents == "Arabian", "Oriental", scents))
p1 = subset(perfume, scent != "Vanilla" & scent != "Aromatic" & scent != "Musk" & scent
 != "Jasmine" & scent != "Floral and Oriental" & scent != "Rose, Floral" & scent != "San
dalwood" & scent != "Woody, Sweet" & scent != "Aromatic,Citrus" & scent != "Clean" & sce
nt != "Oriental, Floral" & scent != "Sweet Aromatic" & scent != "Woody And Spicy" & scen
t != "Woody, Musky")
```

```
p2 = p1 %>%
  mutate(conc = ifelse(concentration == "PDT", "EDT", concentration))
p2 = subset(p2, select = -c(concentration))
```

```
p3 = p2 %>%
  mutate(brands1 = ifelse(brand == "ST Dupont", "S.T.Dupont", brand)) %>%
  mutate(brands2 = ifelse(brands1 == "armani", "GIORGIO ARMANI", brands1)) %>%
  mutate(brands3 = ifelse(brands2 == "Genie Collection", "Genie", brands2)) %>%
  mutate(brands4 = ifelse(brands3 == "LANVIN PARIS", "LANVIN", brands3)) %>%
  mutate(brands5 = ifelse(brands4 == "Mont Blanc", "MONTBLANC", brands4)) %>%
  mutate(brands6 = ifelse(brands5 == "marbert man", "Marbert", brands5)) %>%
  mutate(brands = ifelse(brands6 == "YSL" | brands6 == "YVES", "Yves Saint Laurent", bra
nds6))
p3 = subset(p3, select = -c(brand, brands1, brands2, brands3, brands4, brands5, brands
6))
```

```
p4 = subset(p3, seller_rating <= 5.0)
p5 = p4 %>%
  mutate(num_sel_ratings =
           ifelse(grepl("K", num_seller_ratings),
                  as.numeric(substring(num_seller_ratings, 1, nchar(num_seller_ratings)
 - 1)) * 1000,
                  as.numeric(num_seller_ratings)))
```

```
## Warning in ifelse(grepl("K", num_seller_ratings),
## as.numeric(substring(num_seller_ratings, : NAs introduced by coercion
```

```
p5 = subset(p5, select = -c(num_seller_ratings))
```

```r
# clean seller column
seller = as.vector(p5$seller)
seller = tolower(seller)
index_golden = which(grepl("golden", seller))
seller[index_golden] = "golden perfumes"
index_lolita = which(grepl("lolita", seller))
seller[index_lolita] = "lolita shop"
index_noon = which(grepl("noon", seller))
seller[index_noon] = "noon"
index_swiss = which(grepl("swiss", seller))
seller[index_swiss] = "swiss arabian perfumes"
index_pa = which(grepl("perfumes--addresses", seller))
seller[index_pa] = "perfumes"
index_ps = which(grepl("perfumes-shop", seller))
seller[index_ps] = "perfumes"

p6 = p5
p6$seller = seller
sb = c(48, 435, 651)
bf = c(109, 121, 470, 565, 576)
p6 = p6 %>%
  mutate(seller1 = ifelse(is.element(X, sb), "show biz", seller)) %>%
  mutate(sellers = ifelse(is.element(X, bf), "beauty fortune", seller))
p6 = subset(p6, select = -c(seller1, seller))
p6 = p6 %>%
  filter(conc != "EDC")
```

```
base_note = as.vector(p6$base_note)
base_note = tolower(base_note)
base_note = str_replace_all(base_note, " and ", ",")
base_note = str_replace_all(base_note, " ", "")
base_note = str_replace_all(base_note, "vanille", "vanilla")
base_note = str_replace_all(base_note, "woodsynotes", "wood")
base_note = str_replace_all(base_note, "orrisroot", "orris")
base_note = str_replace_all(base_note, "woodsynote", "wood")
base_note = str_replace_all(base_note, "woodynotes", "wood")
base_note = str_replace_all(base_note, "woody", "wood")
base_note = str_replace_all(base_note, "cedarwood", "cedar")
base_note = str_replace_all(base_note, "virginiacedar", "cedar")
base_note = str_replace_all(base_note, "whitemusk", "musk")
base_note = str_replace_all(base_note, "tonkabeans", "tonka")
base_note = str_replace_all(base_note, "tonkabean", "tonka")
base_note = str_replace_all(base_note, "amberwood", "amber")
base_note = str_replace_all(base_note, "sandalwood", "sandal")
base_note = str_replace_all(base_note, "cashmerewood", "cashmere")
base_note = str_replace_all(base_note, "guaiacwood", "guaiac")
base_note = str_replace_all(base_note, "ambergris", "AMBERGRIS")
base_note = str_replace_all(base_note, "mustyoud", "oud")
base_note = str_replace_all(base_note, "naturaloudoil", "oud")
base_note = str_replace_all(base_note, "agarwood\\(oud\\)", "oud")
base_note = str_replace_all(base_note, "agarwood", "oud")
base_note = str_replace_all(base_note, "oudh", "oud")
p6$base_note = base_note
```

```
mid_note = as.vector(p6$middle_note)
mid_note = tolower(mid_note)
mid_note = str_replace_all(mid_note, " and ", ",")
mid_note = str_replace_all(mid_note, " ", "")
mid_note = str_replace_all(mid_note, "lily-of-the-valley", "lily")
mid_note = str_replace_all(mid_note, "orrisroot", "orris")
mid_note = str_replace_all(mid_note, "lilyofthevalley", "lily")
mid_note = str_replace_all(mid_note, "bulgarianrose", "rose")
mid_note = str_replace_all(mid_note, "africanorangeflower", "orangeblossom")
mid_note = str_replace_all(mid_note, "neroli", "orangeblossom")
mid_note = str_replace_all(mid_note, "jasminesambac", "jasmine")
mid_note = str_replace_all(mid_note, "wildjasmine", "jasmine")
mid_note = str_replace_all(mid_note, "wildjasmine", "jasmine")
mid_note = str_replace_all(mid_note, "blackpepper", "pepper")
mid_note = str_replace_all(mid_note, "pinkpepper", "pepper")
mid_note = str_replace_all(mid_note, "vanille", "vanilla")
mid_note = str_replace_all(mid_note, "tuberose", "TUBEROSE")
mid_note = str_replace_all(mid_note, "orrisroot", "ORRISROOT")
mid_note = str_replace_all(mid_note, "honeysuckle", "HONEYSUCKLE")
mid_note = str_replace_all(mid_note, "rosemary", "ROSEMARY")
mid_note = str_replace_all(mid_note, "violetleaf", "VIOLETLEAF")
mid_note = str_replace_all(mid_note, "clarysage", "CLARYSAGE")
mid_note = str_replace_all(mid_note, "oudh", "oud")
mid_note = str_replace_all(mid_note, "burningoud", "oud")
mid_note = str_replace_all(mid_note, "agarwood\\(oud\\)", "oud")
mid_note = str_replace_all(mid_note, "agarwood", "oud")
mid_note = str_replace_all(mid_note, "oudwood", "oud")
p6$middle_note = mid_note
```

```
p7 = p6 %>%
  filter(ml > 5)
#
# # add ordinal version of ml
# vol = as.vector(p7$ml)
# unique_vol = as.data.frame(vol) %>%
#   group_by(vol) %>%
#   summarise(count = n()) %>%
#   subset(select = vol)
# unique_vol = as.vector(unique_vol$vol)
#
# order = vol
# rank = 0
# for (i in unique_vol) {
#   rank = rank + 1
#   index = which(vol == i)
#   order[index] = rank
# }
# p7$ml_order = order
# p7 = subset(p7, select = -c(ml))
p7 = p7 %>%
  mutate(gender = ifelse(department == "Kids Unisex", "Unisex", department)) %>%
  filter(middle_note != "shavingsoap")
```

```
perfume = subset(p7, select = -c(department, X, name, scents))
perfume = unique(perfume)

brand = as.vector(p7$brands)
brand = tolower(brand)
new_brands = as.data.frame(brand) %>%
  group_by(brand) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
big_brands = new_brands[which(new_brands$count > 10), ]$brand
perfume = perfume %>%
  mutate(big_brand = ifelse(is.element(tolower(brands), big_brands), 1, 0))
perfume = subset(perfume, select = -c(brands))

perfume = perfume %>%
  mutate(is_noon = ifelse(tolower(sellers) == 'noon', 1, 0))
perfume = subset(perfume, select = -c(sellers))

get_notes = function(base, middle) {
  bnote = as.vector(unlist(strsplit(base, split = ",")))
  mnote = as.vector(unlist(strsplit(middle, split = ",")))
  return(union(bnote, mnote))
}

complexity = function(notes) {
  return(length(notes))
}

luxury = function(notes) {
  score = 0
  for (i in 1:length(notes)) {
    if (notes[i] == "musk" | notes[i] == "orris") { # 100-200
      score = score + 1
    } else if (notes[i] == "neroli" | notes[i] == "jasmine" | notes[i] == "sandal") { #
 200-400
      score = score + 2
    } else if (notes[i] == "rose" | notes[i] == "tuberose") { # 400-800
      score = score + 3
    } else if (notes[i] == "AMBERGRIS") { # 800-1200
      score = score + 4
    } else if (notes[i] == "oud") { # 1200-1600
      score = score + 5
    } else {
      score = score + 0
    }
  }
  return(score)
}
```

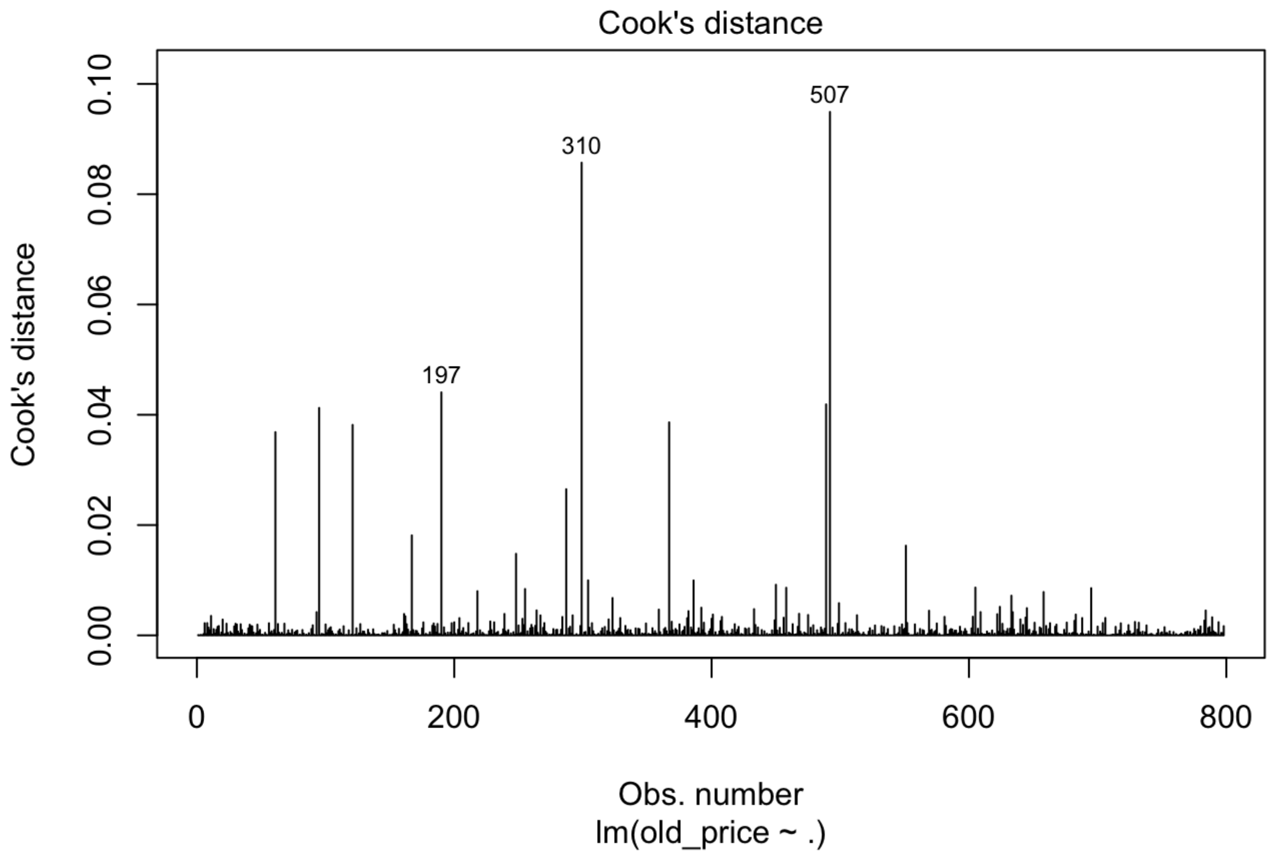
```r
N = nrow(perfume)
complex = lux = rep(0, N)
for (i in 1:N) {
  complex[i] = complexity(get_notes(perfume[i, ]$base_note, perfume[i, ]$middle_note))
  lux[i] = luxury(get_notes(perfume[i, ]$base_note, perfume[i, ]$middle_note))
}
comp_score = lux_score = rep(0, N)
for (i in 1:N) {
  x = complex[i]
  comp_score[i] = sum(complex <= x) / N * 100
  y = lux[i]
  lux_score[i] = sum(lux <= y) / N * 100
}
perfume = perfume %>%
  mutate(comp = complex)
# (comp_score * lux_score) / 100
```

```r
rse = function(model) {
  sqrt(sum(model$residuals ^ 2) / model$df.residual)
}

r2 = function(model) {
  summary(model)$adj.r.squared
}

mse = function(model) {
  mean(model$residuals ^ 2)
}

ge = function(model) {
  n = nobs(model)
  ge = 2 * (rse(model) ^ 2) * length(model$coefficients) / n
  return(ge)
}

Cp.lm = function(mdl.list) {
  n = nobs(mdl.list[[1]])
  DoFs = sapply(mdl.list, function(mdl) { sum(hatvalues(mdl)) })
  MSEs = sapply(mdl.list, function(mdl) { mean(residuals(mdl)^2) })
  biggest = which.max(DoFs)
  sigma2.hat = MSEs[[biggest]]*n/(n-DoFs[[biggest]])
  Cp = MSEs + 2*sigma2.hat*DoFs/n
  return(Cp)
}
```

```r
perfume = subset(perfume, select = -c(new_price, base_note, middle_note))
lm.1 = lm(old_price ~ ., data = perfume)
summary(lm.1)
```

```
##
## Call:
## lm(formula = old_price ~ ., data = perfume)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -377.71 -147.39  -16.41  115.27 1441.45
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.634e+01  1.671e+02   0.397 0.691464
## ml               1.279e+00  3.548e-01   3.606 0.000331 ***
## item_rating      3.702e+00  1.412e+01   0.262 0.793250
## seller_rating    5.571e+01  3.895e+01   1.430 0.153087
## scentFloral     -1.275e+01  3.017e+01  -0.423 0.672596
## scentFresh      -9.728e+01  4.215e+01  -2.308 0.021281 *
## scentFruity     -5.955e+01  3.753e+01  -1.587 0.113008

## scentOriental   -5.881e+01  3.575e+01  -1.645 0.100345
## scentSpicy      -4.004e+01  3.322e+01  -1.205 0.228429
## scentWoody       9.180e+00  3.024e+01   0.304 0.761549
## concEDT         -1.457e+02  1.966e+01  -7.410 3.29e-13 ***
## num_sel_ratings -1.254e-03  9.611e-04  -1.305 0.192323
## genderUnisex    -1.179e+02  3.615e+01  -3.261 0.001160 **
## genderWomen     -1.588e+01  2.213e+01  -0.718 0.473214
## big_brand        7.978e+01  1.597e+01   4.996 7.22e-07 ***
## is_noon          9.856e+01  9.317e+01   1.058 0.290441
## comp            -3.123e+00  2.380e+00  -1.312 0.189851
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 216.3 on 781 degrees of freedom
## Multiple R-squared:  0.1371, Adjusted R-squared:  0.1194
## F-statistic: 7.755 on 16 and 781 DF,  p-value: < 2.2e-16
```

```
# residual analysis
plot(lm.1, which = 4)
```




```
dwtest(lm.1, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  lm.1
## DW = 1.916, p-value = 0.2273
## alternative hypothesis: true autocorrelation is not 0
```

```
set1 = lm.1$residuals[which(lm.1$fitted.values >= 300)]
set2 = lm.1$residuals[which(lm.1$fitted.values < 300)]
var.test(set1, set2)
```

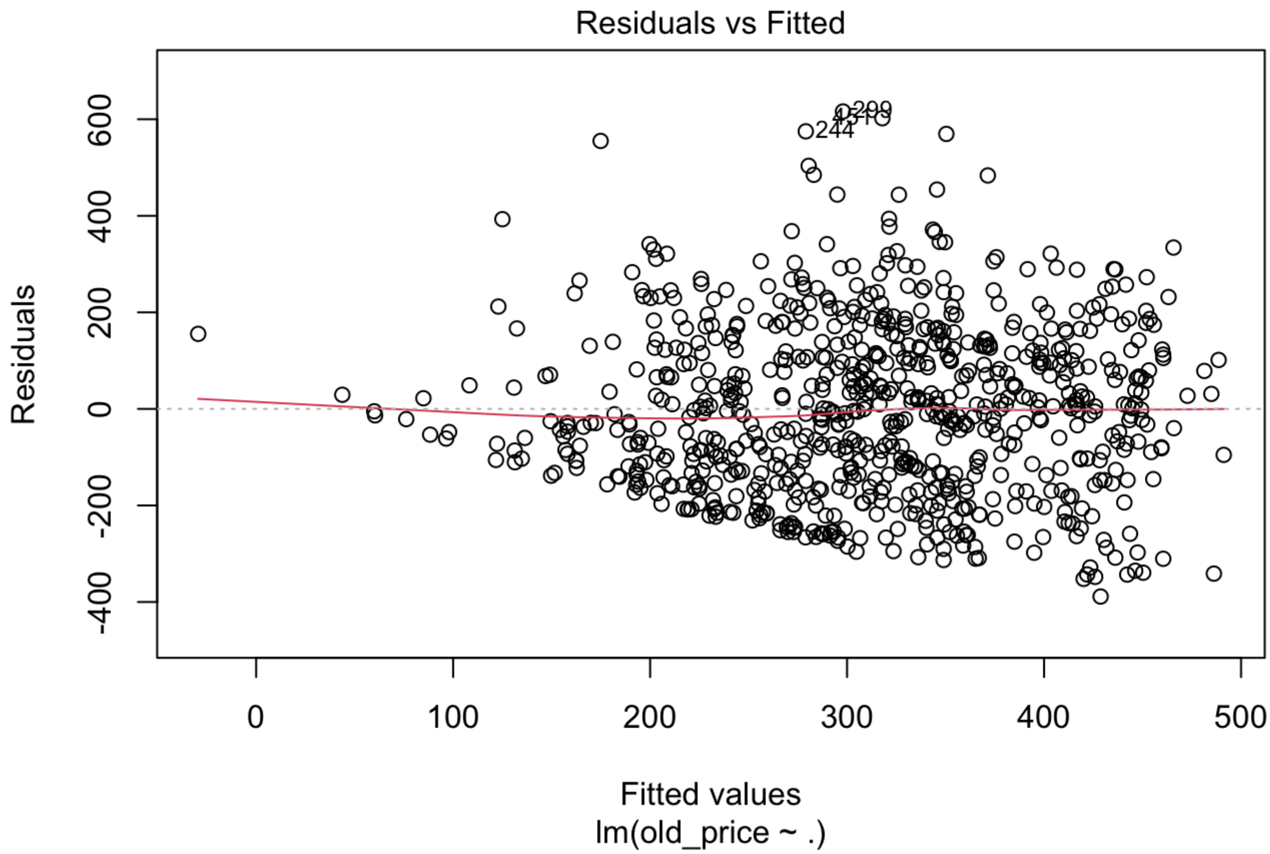
```
##
##  F test to compare two variances
##
## data:  set1 and set2
## F = 1.39, num df = 499, denom df = 297, p-value = 0.001831
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.131064 1.699080
## sample estimates:
## ratio of variances
##           1.389972
```

```
perfume2 = perfume %>%
  filter(old_price < 930)
```

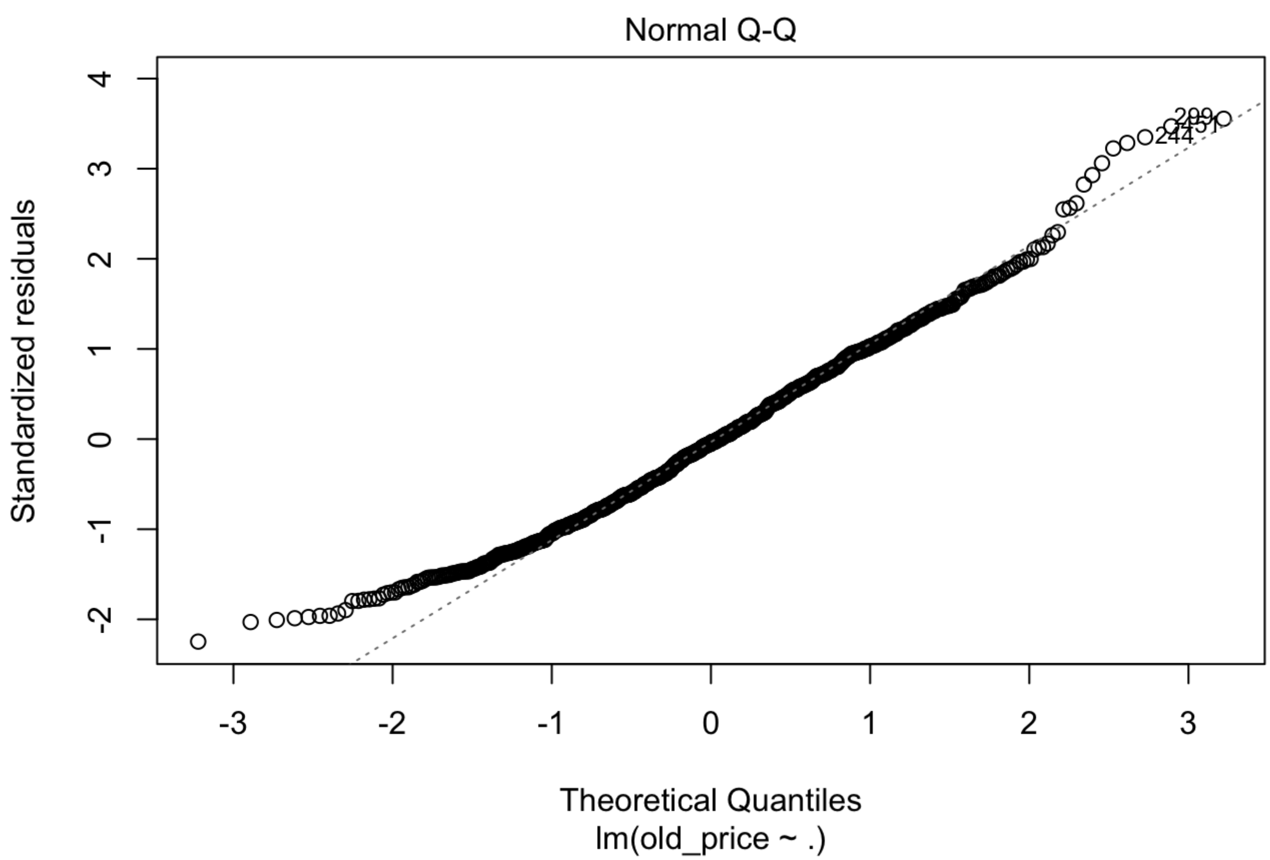
```
lm.2 = lm(old_price ~ ., data = perfume2)
summary(lm.2)
```

```
##
## Call:
## lm(formula = old_price ~ ., data = perfume2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -388.70 -132.69   -8.52  121.88  616.08
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -4.924e+01  1.364e+02  -0.361 0.718195
## ml               1.170e+00  2.878e-01   4.065 5.30e-05 ***
## item_rating      6.780e+00  1.152e+01   0.589 0.556282
## seller_rating    8.123e+01  3.181e+01   2.554 0.010849 *
## scentFloral     -1.104e+01  2.451e+01  -0.451 0.652394
## scentFresh      -1.164e+02  3.440e+01  -3.383 0.000754 ***
## scentFruity     -6.434e+01  3.054e+01  -2.107 0.035429 *
## scentOriental   -4.197e+01  2.897e+01  -1.449 0.147821
## scentSpicy      -5.380e+01  2.697e+01  -1.995 0.046392 *
## scentWoody      -2.651e+01  2.466e+01  -1.075 0.282736
## concEDT         -1.201e+02  1.613e+01  -7.444 2.63e-13 ***
## num_sel_ratings -1.084e-03  7.783e-04  -1.393 0.164150
## genderUnisex    -1.570e+02  2.980e+01  -5.270 1.78e-07 ***
## genderWomen     -2.308e+01  1.820e+01  -1.268 0.205283
## big_brand        8.712e+01  1.306e+01   6.672 4.84e-11 ***
## is_noon          8.543e+01  7.543e+01   1.132 0.257784
## comp            -4.513e+00  1.934e+00  -2.333 0.019893 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 175.1 on 767 degrees of freedom
## Multiple R-squared:  0.1923, Adjusted R-squared:  0.1754
## F-statistic: 11.41 on 16 and 767 DF,  p-value: < 2.2e-16
```

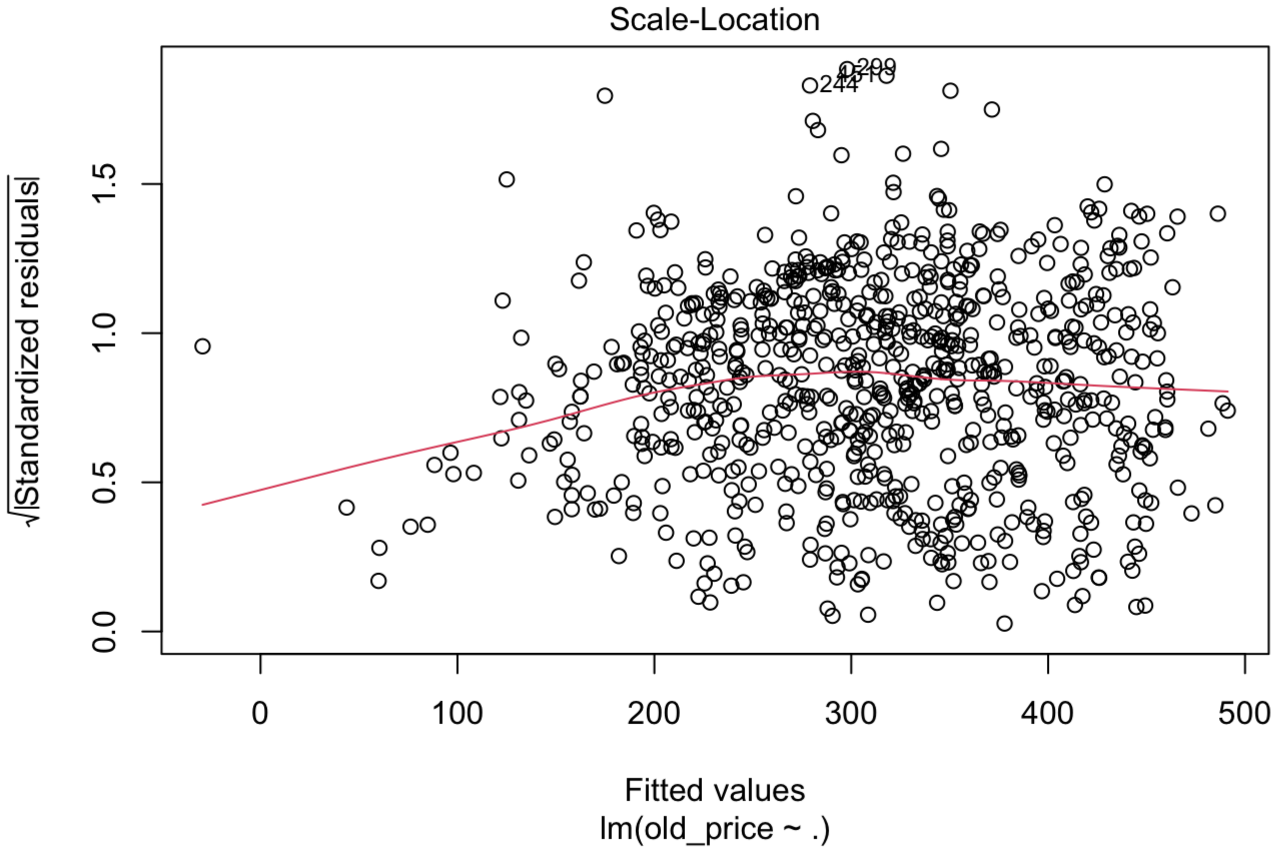
```
# residual analysis
plot(lm.2)
```



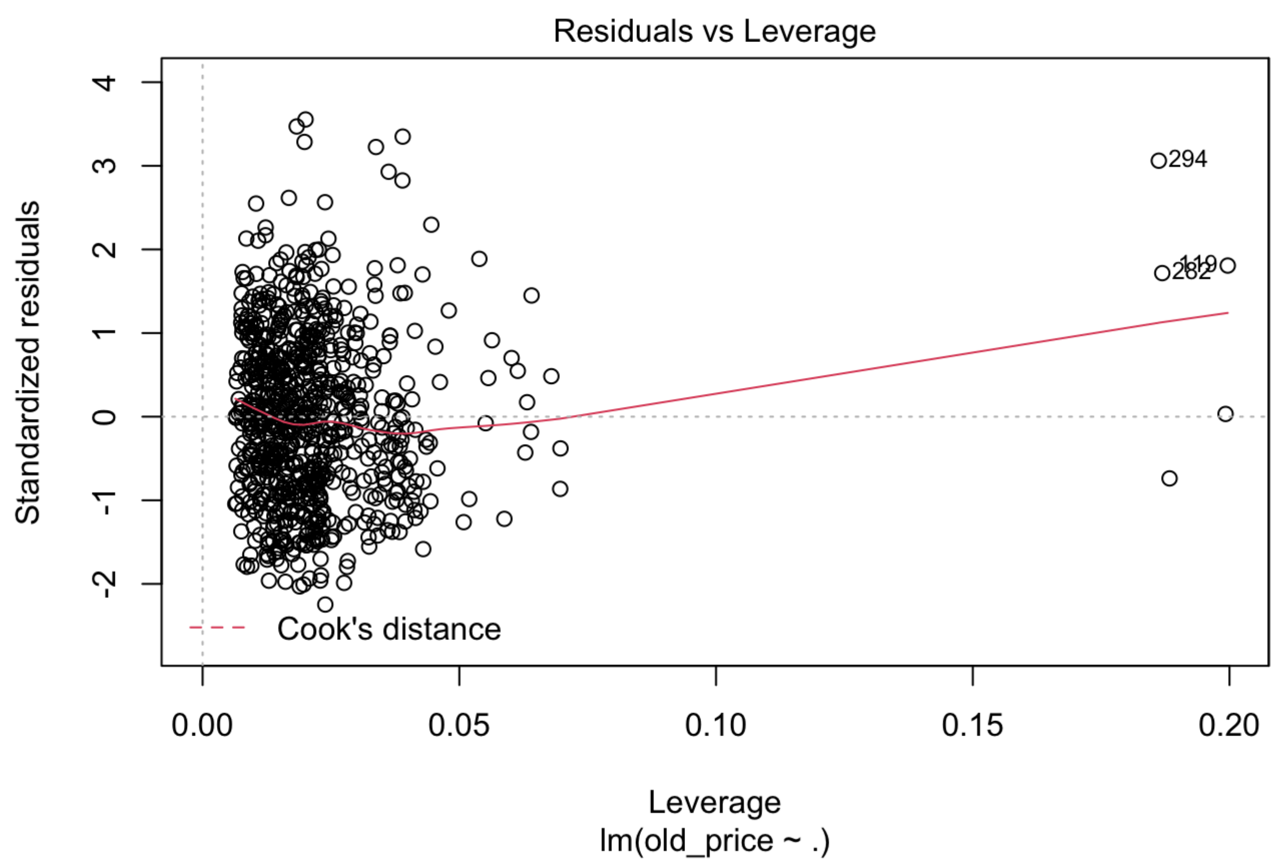
Residuals vs Fitted
Residuals
600
400
200
0
-200
-400
0
100
200
300
400
500
Fitted values
lm(old_price ~ .)
209
451
244



Normal Q-Q
Standardized residuals
4
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(old_price ~ .)
209
451
244

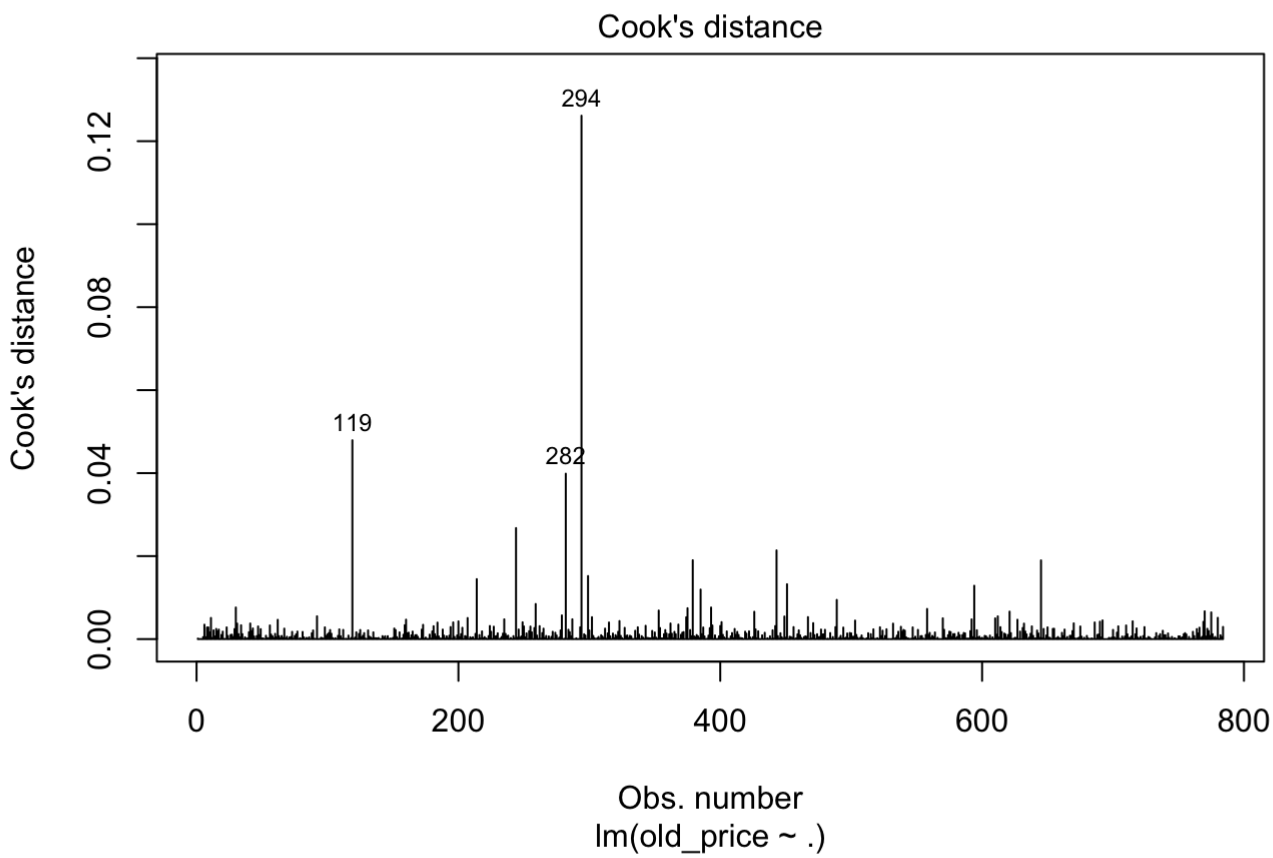


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
100
200
300
400
500
244
299
Fitted values
lm(old_price ~ .)



Residuals vs Leverage
Standardized residuals
-2
-1
0
1
2
3
4
0.00
0.05
0.10
0.15
0.20
294
282
Cook's distance
Leverage
lm(old_price ~ .)

```
plot(lm.2, which = 4)
```




```
dwtest(lm.2, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  lm.2
## DW = 1.8785, p-value = 0.08476
## alternative hypothesis: true autocorrelation is not 0
```

```
set1 = lm.2$residuals[which(lm.2$fitted.values >= 300)]
set2 = lm.2$residuals[which(lm.2$fitted.values < 300)]
var.test(set1, set2)
```

```
##
##  F test to compare two variances
##
## data:  set1 and set2
## F = 1.0427, num df = 451, denom df = 331, p-value = 0.6874
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8515493 1.2727029
## sample estimates:
## ratio of variances
##           1.042677
```

```
# remove is_noon?
lm.3 = lm(old_price ~ big_brand + comp + item_rating +
            conc + ml + num_sel_ratings +
            gender + seller_rating + scent, data = perfume2)
summary(lm.3)
```

```
##
## Call:
## lm(formula = old_price ~ big_brand + comp + item_rating + conc +
##     ml + num_sel_ratings + gender + seller_rating + scent, data = perfume2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -388.54 -132.61   -7.18  122.82  617.14
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -5.047e+01  1.364e+02  -0.370 0.711535
## big_brand        8.661e+01  1.305e+01   6.635 6.11e-11 ***
## comp            -4.504e+00  1.935e+00  -2.328 0.020175 *
## item_rating      6.865e+00  1.152e+01   0.596 0.551454
## concEDT         -1.212e+02  1.610e+01  -7.527 1.46e-13 ***
## ml               1.164e+00  2.878e-01   4.043 5.81e-05 ***
## num_sel_ratings -2.269e-04  1.822e-04  -1.245 0.213372
## genderUnisex    -1.579e+02  2.980e+01  -5.298 1.53e-07 ***
## genderWomen     -2.304e+01  1.821e+01  -1.265 0.206096
## seller_rating    8.164e+01  3.181e+01   2.566 0.010466 *
## scentFloral     -1.172e+01  2.451e+01  -0.478 0.632632
## scentFresh      -1.157e+02  3.440e+01  -3.365 0.000805 ***
## scentFruity     -6.415e+01  3.054e+01  -2.101 0.036006 *
## scentOriental   -4.053e+01  2.895e+01  -1.400 0.161917
## scentSpicy      -5.251e+01  2.695e+01  -1.949 0.051710 .
## scentWoody      -2.664e+01  2.467e+01  -1.080 0.280472
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 175.1 on 768 degrees of freedom
## Multiple R-squared:  0.1909, Adjusted R-squared:  0.1751
## F-statistic: 12.08 on 15 and 768 DF,  p-value: < 2.2e-16
```

```
anova(lm.2, lm.3)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	767	23519758	NA	NA	NA	NA
2	768	23559086	-1	-39328.1	1.282524	0.2577842

2 rows

```
# yes
```

```r
# remove item_rating?
lm.4 = lm(old_price ~ big_brand + comp +
            conc + ml + num_sel_ratings +
            gender + seller_rating + scent, data = perfume2)
summary(lm.4)
```

```
##
## Call:
## lm(formula = old_price ~ big_brand + comp + conc + ml + num_sel_ratings +
##     gender + seller_rating + scent, data = perfume2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -392.0 -131.7   -5.2  124.8  617.7
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2.279e+01  1.282e+02  -0.178 0.858964
## big_brand        8.682e+01  1.304e+01   6.656 5.33e-11 ***
## comp            -4.491e+00  1.934e+00  -2.323 0.020462 *
## concEDT         -1.217e+02  1.607e+01  -7.574 1.04e-13 ***
## ml               1.162e+00  2.877e-01   4.040 5.88e-05 ***
## num_sel_ratings -2.291e-04  1.821e-04  -1.258 0.208764
## genderUnisex    -1.581e+02  2.978e+01  -5.309 1.44e-07 ***
## genderWomen     -2.190e+01  1.810e+01  -1.210 0.226568
## seller_rating    8.252e+01  3.177e+01   2.598 0.009566 **
## scentFloral     -1.237e+01  2.447e+01  -0.506 0.613225
## scentFresh      -1.165e+02  3.436e+01  -3.390 0.000734 ***
## scentFruity     -6.452e+01  3.052e+01  -2.114 0.034848 *
## scentOriental   -4.094e+01  2.893e+01  -1.415 0.157398
## scentSpicy      -5.217e+01  2.693e+01  -1.937 0.053079 .
## scentWoody      -2.661e+01  2.466e+01  -1.079 0.280857
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 175.1 on 769 degrees of freedom
## Multiple R-squared:  0.1906, Adjusted R-squared:  0.1758
## F-statistic: 12.93 on 14 and 769 DF,  p-value: < 2.2e-16
```

```r
anova(lm.3, lm.4)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	768	23559086	NA	NA	NA	NA
2	769	23569977	-1	-10890.91	0.3550315	0.5514542

2 rows

```
# yes
```

```
# remove num_sel_ratings?
lm.5 = lm(old_price ~ big_brand + comp +
            conc + ml +
            gender + seller_rating + scent, data = perfume2)
summary(lm.5)
```

```
##
## Call:
## lm(formula = old_price ~ big_brand + comp + conc + ml + gender +
##     seller_rating + scent, data = perfume2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -401.66 -131.55   -6.68  118.22  620.69
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     66.1517   107.0097   0.618 0.536637
## big_brand       85.9311    13.0284   6.596 7.87e-11 ***
## comp            -4.4495     1.9342  -2.300 0.021694 *
## concEDT       -121.6950    16.0785  -7.569 1.08e-13 ***
## ml               1.1437     0.2874   3.979 7.57e-05 ***
## genderUnisex  -159.7660    29.7651  -5.368 1.06e-07 ***
## genderWomen    -21.7679    18.1052  -1.202 0.229617
## seller_rating   58.8679    25.6163   2.298 0.021826 *
## scentFloral    -12.3393    24.4797  -0.504 0.614361
## scentFresh    -115.8323    34.3721  -3.370 0.000789 ***
## scentFruity    -64.5532    30.5335  -2.114 0.034821 *
## scentOriental  -40.8892    28.9379  -1.413 0.158060
## scentSpicy     -51.5216    26.9361  -1.913 0.056153 .
## scentWoody     -26.3411    24.6668  -1.068 0.285910
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 175.1 on 770 degrees of freedom
## Multiple R-squared:  0.1889, Adjusted R-squared:  0.1752
## F-statistic: 13.79 on 13 and 770 DF,  p-value: < 2.2e-16
```

```
anova(lm.4, lm.5)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	769	23569977	NA	NA	NA	NA
2	770	23618485	-1	-48508	1.582634	0.2087635

2 rows

```
# yes
```

```
# remove seller_rating?
lm.6 = lm(old_price ~ big_brand + comp +
            conc + ml +
            gender + scent, data = perfume2)
summary(lm.6)
```

```
##
## Call:
## lm(formula = old_price ~ big_brand + comp + conc + ml + gender +
##     scent, data = perfume2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -390.30 -132.39   -3.85  126.65  613.28
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    293.1350    41.2872   7.100 2.84e-12 ***
## big_brand       88.5349    13.0150   6.803 2.06e-11 ***
## comp            -4.6393     1.9378  -2.394 0.016900 *
## concEDT       -123.0394    16.1124  -7.636 6.62e-14 ***
## ml               1.1934     0.2874   4.153 3.65e-05 ***
## genderUnisex  -158.8406    29.8449  -5.322 1.35e-07 ***
## genderWomen    -19.5736    18.1301  -1.080 0.280649
## scentFloral    -11.9415    24.5470  -0.486 0.626767
## scentFresh    -118.3690    34.4496  -3.436 0.000622 ***
## scentFruity    -63.7574    30.6162  -2.082 0.037628 *
## scentOriental  -38.2330    28.9949  -1.319 0.187692
## scentSpicy     -52.7289    27.0056  -1.953 0.051239 .
## scentWoody     -27.4623    24.7303  -1.110 0.267142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 175.6 on 771 degrees of freedom
## Multiple R-squared:  0.1833, Adjusted R-squared:  0.1706
## F-statistic: 14.42 on 12 and 771 DF,  p-value: < 2.2e-16
```

```
anova(lm.5, lm.6)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	770	23618485	NA	NA	NA	NA
2	771	23780473	-1	-161988.3	5.281076	0.02182561

2 rows

```
# no
```

```
# remove comp?
lm.7 = lm(old_price ~ big_brand +
            conc + ml + seller_rating +
            gender + scent, data = perfume2)
summary(lm.7)
```

```
##
## Call:
## lm(formula = old_price ~ big_brand + conc + ml + seller_rating +
##     gender + scent, data = perfume2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -402.45 -134.17   -2.55  120.68  641.36
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     30.0020   106.1437   0.283  0.77752
## big_brand       90.4464    12.9155   7.003 5.46e-12 ***
## concEDT       -124.4426    16.0786  -7.740 3.13e-14 ***
## ml               1.0706     0.2864   3.738  0.00020 ***
## seller_rating   61.3847    25.6641   2.392  0.01700 *
## genderUnisex  -158.7315    29.8444  -5.319 1.37e-07 ***
## genderWomen    -22.2882    18.1541  -1.228  0.21993
## scentFloral    -10.8546    24.5392  -0.442  0.65837
## scentFresh    -113.6252    34.4542  -3.298  0.00102 **
## scentFruity    -62.4752    30.6050  -2.041  0.04156 *
## scentOriental  -42.8740    29.0054  -1.478  0.13978
## scentSpicy     -49.8770    27.0014  -1.847  0.06510 .
## scentWoody     -24.4305    24.7213  -0.988  0.32335
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 175.6 on 771 degrees of freedom
## Multiple R-squared:  0.1833, Adjusted R-squared:  0.1706
## F-statistic: 14.42 on 12 and 771 DF,  p-value: < 2.2e-16
```

```
anova(lm.5, lm.7)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	770	23618485	NA	NA	NA	NA
2	771	23780799	-1	-162314.2	5.2917	0.02169378

2 rows

```
# We cannot.
```

```
# remove gender?
lm.8 = lm(old_price ~ big_brand +
            conc + ml + comp + seller_rating +
            scent, data = perfume2)
summary(lm.8)
```

```
##
## Call:
## lm(formula = old_price ~ big_brand + conc + ml + comp + seller_rating +
##     scent, data = perfume2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -360.11 -132.84   -3.47  127.08  635.37
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     32.9219   108.2072   0.304  0.76102
## big_brand       86.3603    13.2545   6.516 1.31e-10 ***
## concEDT       -102.9945    14.6992  -7.007 5.32e-12 ***
## ml               1.1643     0.2885   4.036 5.99e-05 ***
## comp            -4.2661     1.9676  -2.168  0.03045 *
## seller_rating   57.9550    26.0292   2.227  0.02627 *
## scentFloral     -7.8200    24.3445  -0.321  0.74813
## scentFresh    -108.3212    34.8832  -3.105  0.00197 **
## scentFruity    -54.0073    30.7669  -1.755  0.07959 .
## scentOriental  -49.1566    29.4035  -1.672  0.09497 .
## scentSpicy     -44.6627    27.1583  -1.645  0.10047
## scentWoody     -21.9729    24.9056  -0.882  0.37792
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 178.2 on 772 degrees of freedom
## Multiple R-squared:  0.158,  Adjusted R-squared:  0.146
## F-statistic: 13.17 on 11 and 772 DF,  p-value: < 2.2e-16
```

```
anova(lm.5, lm.8)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	770	23618485	NA	NA	NA	NA
2	772	24518186	-2	-899700.9	14.66584	5.610808e-07

2 rows

```
# We cannot.
```

```
# remove concentration?
lm.9 = lm(old_price ~ big_brand + gender + comp + seller_rating +
            ml + scent, data = perfume2)
summary(lm.9)
```

```
##
## Call:
## lm(formula = old_price ~ big_brand + gender + comp + seller_rating +
##     ml + scent, data = perfume2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -381.77 -132.91  -10.13  127.95  675.50
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -45.8480   109.7821  -0.418  0.67634
## big_brand        84.8390    13.4948   6.287 5.42e-10 ***
## genderUnisex   -100.3190    29.7397  -3.373  0.00078 ***
## genderWomen      35.9366    17.0106   2.113  0.03496 *
## comp             -5.5370     1.9981  -2.771  0.00572 **
## seller_rating    65.9226    26.5174   2.486  0.01313 *
## ml                0.8317     0.2946   2.823  0.00488 **
## scentFloral      18.2723    25.0091   0.731  0.46523
## scentFresh      -85.8633    35.3676  -2.428  0.01542 *
## scentFruity     -30.8914    31.2911  -0.987  0.32384
## scentOriental     5.0601    29.3085   0.173  0.86297
## scentSpicy      -34.3525    27.8029  -1.236  0.21699
## scentWoody       -2.0542    25.3342  -0.081  0.93539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 181.4 on 771 degrees of freedom
## Multiple R-squared:  0.1285, Adjusted R-squared:  0.115
## F-statistic: 9.477 on 12 and 771 DF,  p-value: < 2.2e-16
```

```
anova(lm.5, lm.9)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	770	23618485	NA	NA	NA	NA
2	771	25375667	-1	-1757182	57.28691	1.077471e-13

2 rows

```
# No.
```

```
# remove scent?
lm.10 = lm(old_price ~ big_brand + gender + ml + conc + seller_rating + comp, data = per
fume2)
summary(lm.10)
```

```
##
## Call:
## lm(formula = old_price ~ big_brand + gender + ml + conc + seller_rating +
##     comp, data = perfume2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -401.68 -132.00   -7.78  117.95  657.06
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      13.3073   105.0236   0.127   0.8992
## big_brand        85.8179    13.0461   6.578 8.76e-11 ***
## genderUnisex   -151.7611    29.6931  -5.111 4.04e-07 ***
## genderWomen      -8.4502    16.2998  -0.518   0.6043
## ml                1.1538     0.2877   4.011 6.64e-05 ***
## concEDT        -116.2215    15.7987  -7.356 4.81e-13 ***
## seller_rating    61.2999    25.7333   2.382   0.0175 *
## comp             -4.3263     1.9427  -2.227   0.0262 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 176.5 on 776 degrees of freedom
## Multiple R-squared:  0.1697, Adjusted R-squared:  0.1622
## F-statistic: 22.66 on 7 and 776 DF,  p-value: < 2.2e-16
```

```
anova(lm.5, lm.10)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	770	23618485	NA	NA	NA	NA
2	776	24177525	-6	-559040.3	3.0376	0.006084229

2 rows

```
# No.
```

```r
# RSE
rses = c(rse(lm.1), rse(lm.2), rse(lm.3), rse(lm.4), rse(lm.5))
# R^2
r2s = c(r2(lm.1), r2(lm.2), r2(lm.3), r2(lm.4), r2(lm.5))
# MSE
mses = c(mse(lm.1), mse(lm.2), mse(lm.3), mse(lm.4), mse(lm.5))
# generalization error
ges = c(ge(lm.1), ge(lm.2), ge(lm.3), ge(lm.4), ge(lm.5))
# Marlow's Cp
Cps = Cp.lm(list(lm.1, lm.2, lm.3, lm.4, lm.5))
# AIC
aics = AIC(lm.1, lm.2, lm.3, lm.4, lm.5)[, 2]
```

```
## Warning in AIC.default(lm.1, lm.2, lm.3, lm.4, lm.5): models are not all fitted
## to the same number of observations
```

```r
# BIC
bics = BIC(lm.1, lm.2, lm.3, lm.4, lm.5)[, 2]
```

```
## Warning in BIC.default(lm.1, lm.2, lm.3, lm.4, lm.5): models are not all fitted
## to the same number of observations
```

```r
# cannot remove scent, conc
metrics = data.frame(rses, r2s, mses, ges, Cps, aics, bics); metrics
```

rses <dbl>	r2s <dbl>	mses <dbl>	ges <dbl>	Cps <dbl>	aics <dbl>	bics <dbl>
216.3359	0.1194184	45804.19	1994.036	47798.23	10864.87	10949.14
175.1131	0.1754292	29999.69	1329.843	31993.73	10343.11	10427.07
175.1453	0.1751258	30049.85	1252.077	31926.59	10342.42	10421.71
175.0719	0.1758177	30063.75	1172.838	31823.19	10340.78	10415.41
175.1381	0.1751940	30125.62	1095.477	31767.77	10340.39	10410.36

5 rows

```r
## 5-fold CV
pre.ols = rep(0, nrow(perfume2))
pre.best = rep(0, nrow(perfume2))
folds = 5
sb = round(seq(0, nrow(perfume2), length = (folds + 1)))
for (i in 1:folds) {
  test = (sb[((folds + 1) - i)] + 1):(sb[((folds + 2) - i)])
  train = (1:nrow(perfume2))[-test]
  ## fit models
  fit.ols = lm(old_price ~ ., data = perfume2[train, ])
  fit.best = lm(old_price ~ big_brand + comp + conc + ml +
                  gender + seller_rating + scent, data = perfume2[train, ])
  ## create predictions
  pre.ols[test] = predict(fit.ols, newdata = perfume2[test, ])
  pre.best[test] = predict(fit.best, newdata = perfume2[test, ])
}


## Finally, compute the mean squared prediction error:
mean((perfume2$old_price - pre.ols) ^ 2)
```

```
## [1] 32890.85
```

```r
mean((perfume2$old_price - pre.best) ^ 2)
```

```
## [1] 31796.32
```

```r
price_by_scent = perfume2 %>%
  group_by(scent) %>%
  summarise(avg_price = mean(old_price))
count_by_scent = perfume2 %>%
  group_by(scent) %>%
  summarise(count = n())
scent_df = data.frame(price_by_scent, count_by_scent[, 2]); scent_df
```

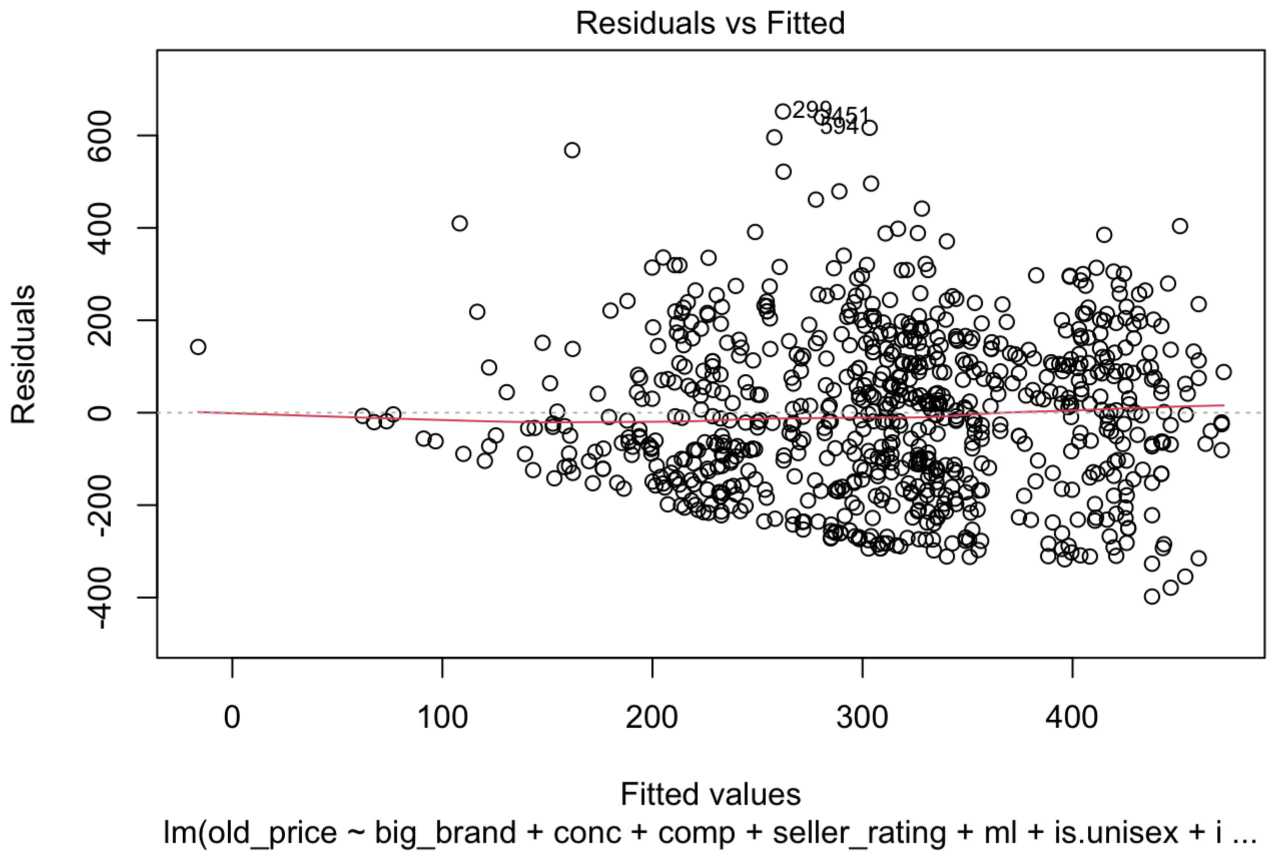
scent <chr>	avg_price <dbl>	count <int>
Citrus	317.9231	78
Floral	344.5989	266
Fresh	223.6600	40
Fruity	293.3083	66
Oriental	307.5211	83
Spicy	285.9680	97
Woody	305.5562	154

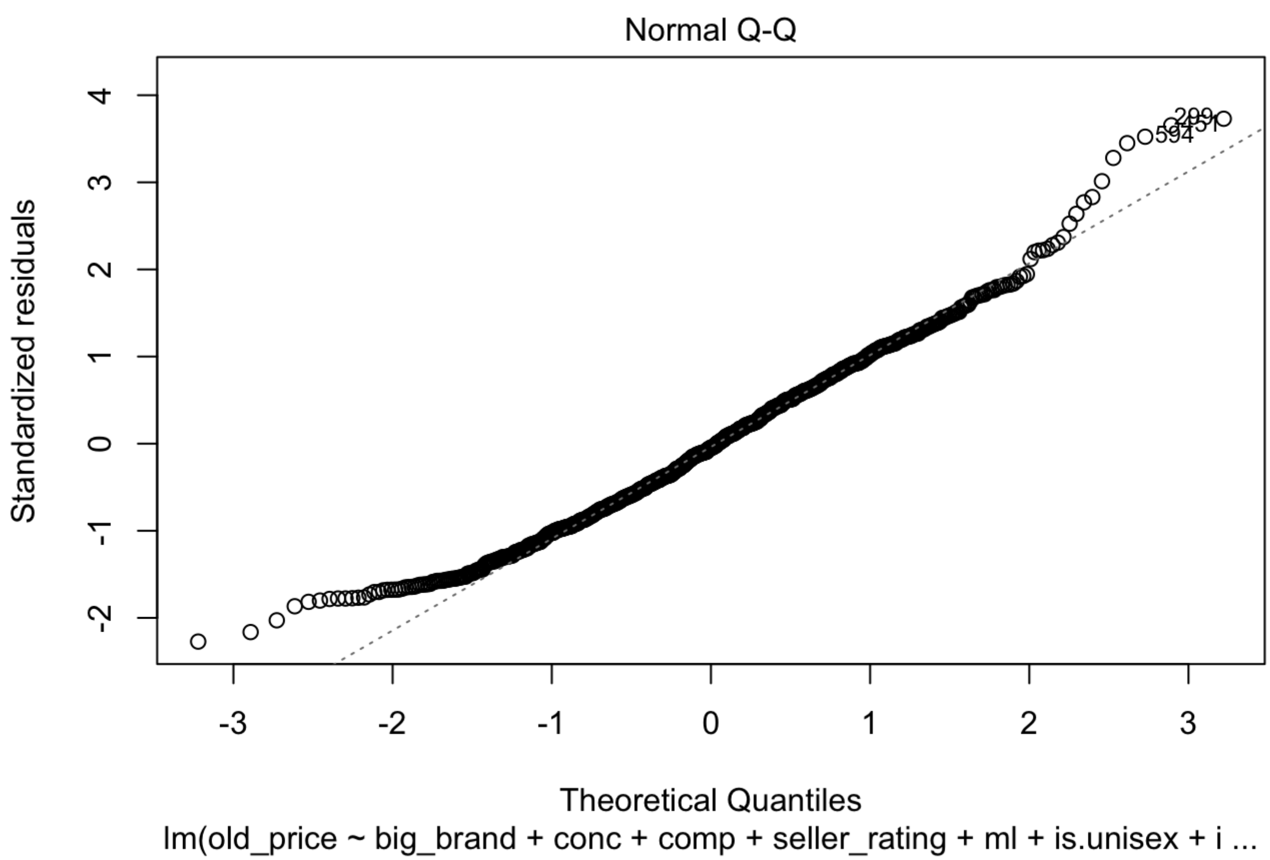
7 rows

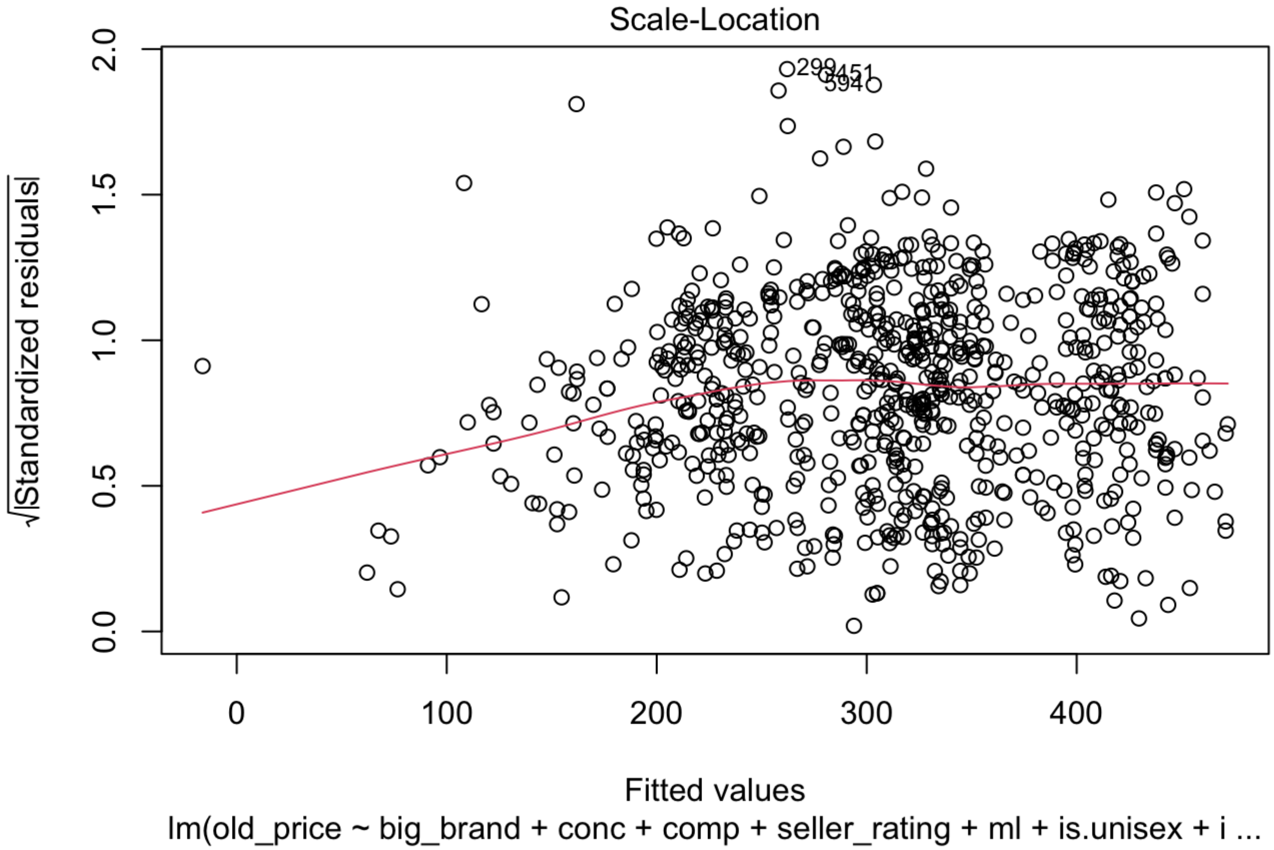
```
perfume3 = perfume2 %>%
  mutate(is.fresh = ifelse(scent == "Fresh", 1, 0)) %>%
  mutate(is.unisex = ifelse(gender == "Unisex", 1, 0))
lm.11 = lm(old_price ~ big_brand + conc + comp + seller_rating + ml + is.unisex + is.fre
sh, data = perfume3)
summary(lm.11)
```

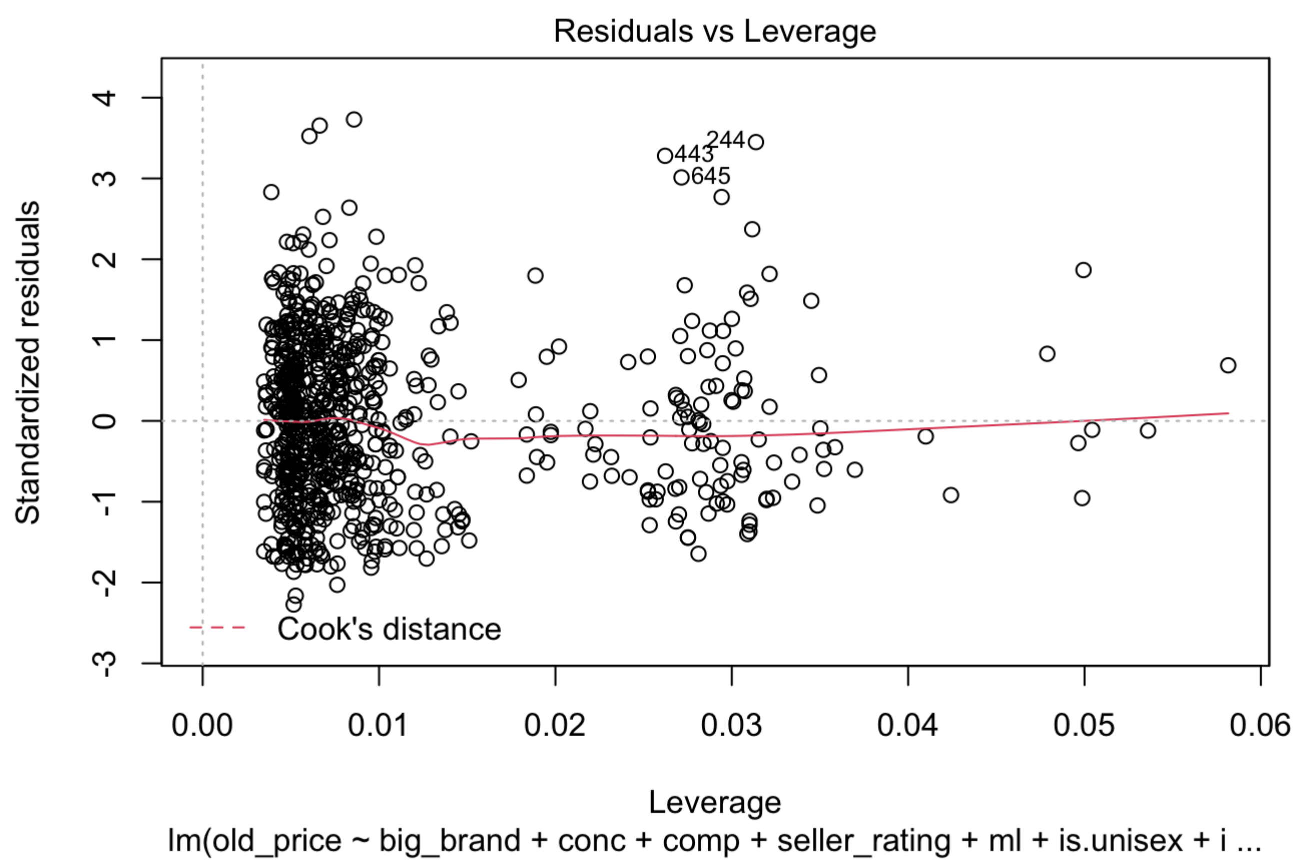
```
##
## Call:
## lm(formula = old_price ~ big_brand + conc + comp + seller_rating +
##     ml + is.unisex + is.fresh, data = perfume3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -397.8 -130.0   -8.3  117.3  651.9
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     23.3471   104.0130   0.224  0.82246
## big_brand       85.5629    12.9732   6.595 7.84e-11 ***
## concEDT       -111.0210    13.5369  -8.201 9.83e-16 ***
## comp            -4.4408     1.9320  -2.298  0.02180 *
## seller_rating   56.8586    25.5389   2.226  0.02628 *
## ml               1.2119     0.2797   4.332 1.67e-05 ***
## is.unisex     -146.3030    27.4428  -5.331 1.28e-07 ***
## is.fresh       -85.3706    28.5594  -2.989  0.00289 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 175.5 on 776 degrees of freedom
## Multiple R-squared:  0.1789, Adjusted R-squared:  0.1714
## F-statistic: 24.15 on 7 and 776 DF,  p-value: < 2.2e-16
```

```
# residual analysis
plot(lm.11)
```

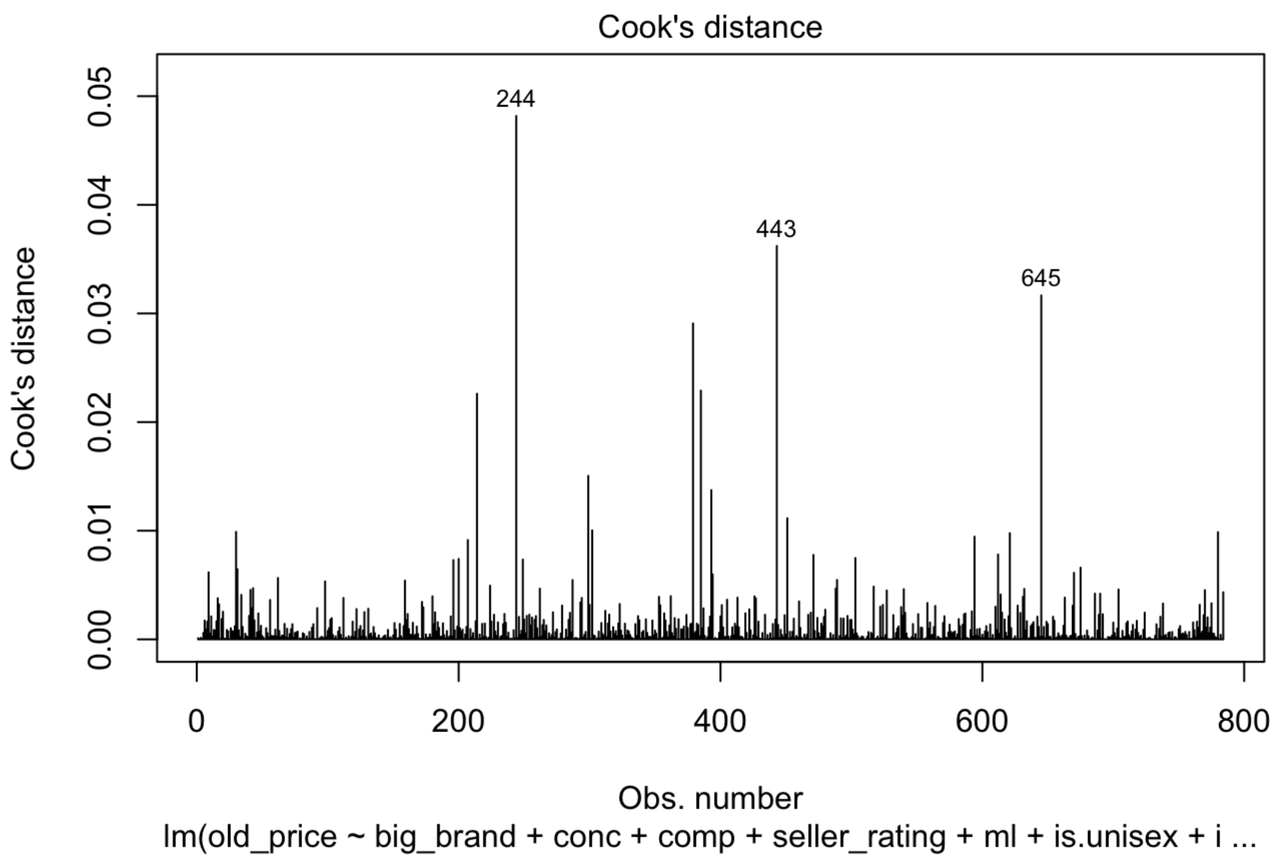

Residuals vs Fitted
299
451
594
Residuals
Fitted values
lm(old_price ~ big_brand + conc + comp + seller_rating + ml + is.unisex + i ...


Normal Q-Q
299
451
594
Standardized residuals
Theoretical Quantiles
lm(old_price ~ big_brand + conc + comp + seller_rating + ml + is.unisex + i ...


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
0
100
200
300
400
Fitted values
299
451
594
lm(old_price ~ big_brand + conc + comp + seller_rating + ml + is.unisex + i ...


Residuals vs Leverage
Standardized residuals
4
3
2
1
0
-1
-2
-3
0.00
0.01
0.02
0.03
0.04
0.05
0.06
Leverage
244
443
645
Cook's distance
lm(old_price ~ big_brand + conc + comp + seller_rating + ml + is.unisex + i ...

```
plot(lm.11, which = 4)
```




```
dwtest(lm.11, alternative = "two.sided")
```

```
##
##  Durbin-Watson test
##
## data:  lm.11
## DW = 1.8654, p-value = 0.0573
## alternative hypothesis: true autocorrelation is not 0
```

```
set1 = lm.11$residuals[which(lm.11$fitted.values >= 300)]
set2 = lm.11$residuals[which(lm.11$fitted.values < 300)]
var.test(set1, set2)
```

```
##
##  F test to compare two variances
##
## data:  set1 and set2
## F = 0.94906, num df = 473, denom df = 309, p-value = 0.6078
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7729524 1.1602054
## sample estimates:
## ratio of variances
##          0.9490633
```

```
lm.12 = lm(old_price ~ ., data = perfume3)
lm.13 = lm(old_price ~ big_brand + comp + item_rating +
            conc + ml + num_sel_ratings +
            is.unisex + seller_rating + is.fresh, data = perfume3)
lm.14 = lm(old_price ~ big_brand + comp +
            conc + ml + num_sel_ratings +
            is.unisex + seller_rating + is.fresh, data = perfume3)
lm.15 = lm(old_price ~ big_brand + comp +
            conc + ml + is.unisex + seller_rating + is.fresh, data = perfume3)

# RSE
rses2 = c(rse(lm.12), rse(lm.13), rse(lm.14), rse(lm.15))
# R^2
r2s2 = c(r2(lm.12), r2(lm.13), r2(lm.14), r2(lm.15))
# MSE
mses2 = c(mse(lm.12), mse(lm.13), mse(lm.14), mse(lm.15))
# generalization error
ges2 = c(ge(lm.12), ge(lm.13), ge(lm.14), ge(lm.15))
# Marlow's Cp
Cps2 = Cp.lm(list(lm.12, lm.13, lm.14, lm.15))
# AIC
aics2 = AIC(lm.12, lm.13, lm.14, lm.15)[, 2]
# BIC
bics2 = BIC(lm.12, lm.13, lm.14, lm.15)[, 2]
# cannot remove scent, conc
metrics = data.frame(rses2, r2s2, mses2, ges2, Cps2, aics2, bics2); metrics
```

rses2 <dbl>	r2s2 <dbl>	mses2 <dbl>	ges2 <dbl>	Cps2 <dbl>	aics2 <dbl>	bics2 <dbl>
175.1131	0.1754292	29999.69	1486.2950	31329.53	10343.11	10427.07
175.5700	0.1711208	30431.66	786.3479	31213.92	10340.31	10391.62
175.4785	0.1719850	30439.21	706.9752	31143.25	10338.51	10385.15
175.5352	0.1714500	30498.18	628.8285	31123.99	10338.03	10380.01

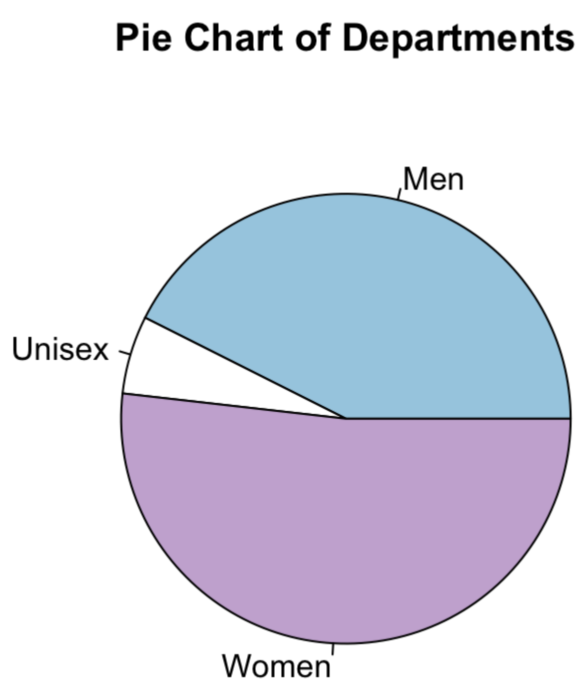
4 rows

```
palette(brewer.pal(n = 12, name = "Paired"))

# Department
p7 %>%
  group_by(department) %>%
  summarise(count = n())
```

department <chr>	count <int>
Kids Unisex	1
Men	378
Unisex	49
Women	461

4 rows

```
dept_slice <- c(379, 50, 461)
lbls <- c("Men", "Unisex", "Women")
pie(dept_slice, labels = lbls, main="Pie Chart of Departments", col = c("#A6CEE3", "#fff
fff", "#CAB2D6"))
```

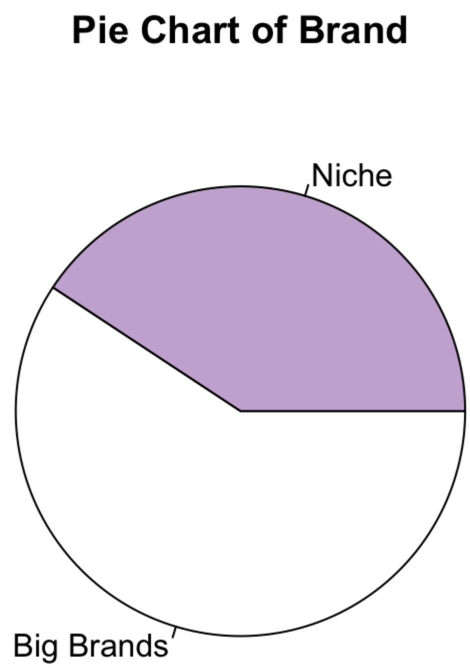


```r
# Brands
p3 %>%
  group_by(brands) %>%
  summarise(count = n())
```

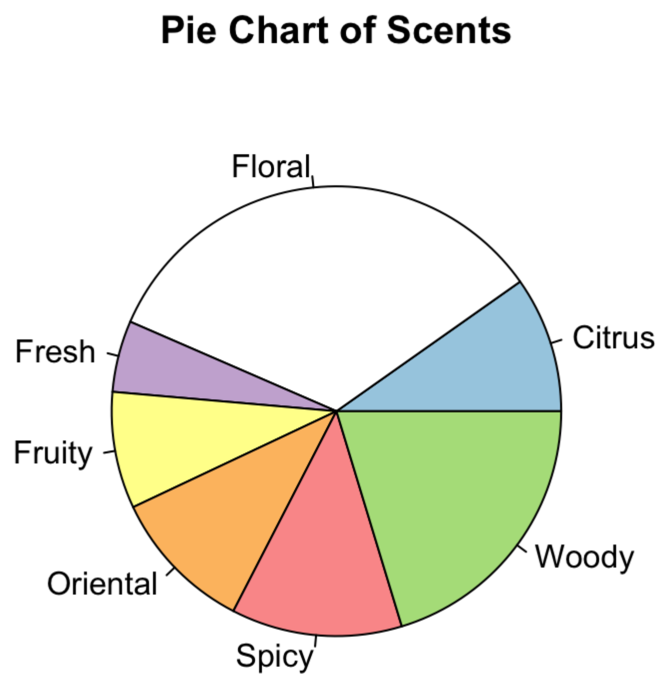
brands <chr>	count <int>
ADOLFO DOMINGUEZ	1
AIGNER	3
Ajmal	14
Al Fakhr	1
al raheeb	1
Al Rasasi	1
Alina Corel	5
Alrehab	2
AMOUAGE	3
ANGEL SCHLESSER	2

1-10 of 148 rows Previous **1** 2 3 4 5 6 ... 15 Next

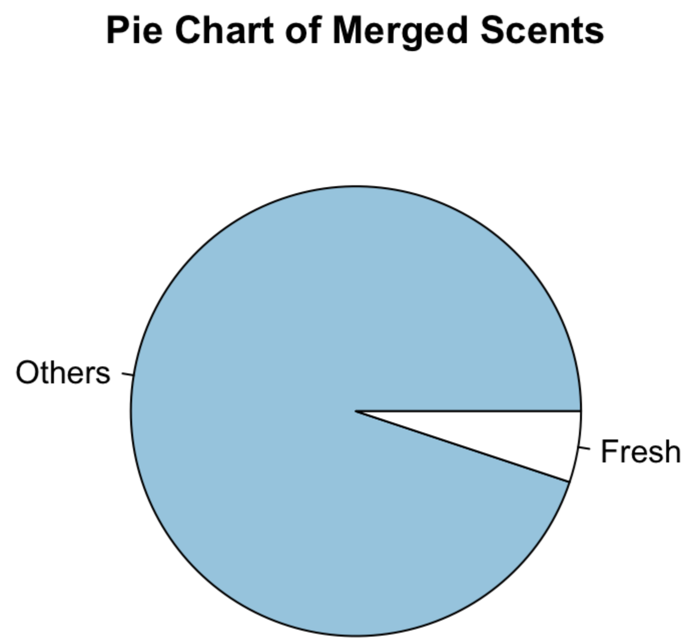
```r
brand_slice <- perfume %>%
  group_by(big_brand) %>%
  summarise(count = n())
pie(brand_slice$count, labels = c("Niche", "Big Brands"), main="Pie Chart of Brand", col
= c("#CAB2D6", "#ffffff"))
```



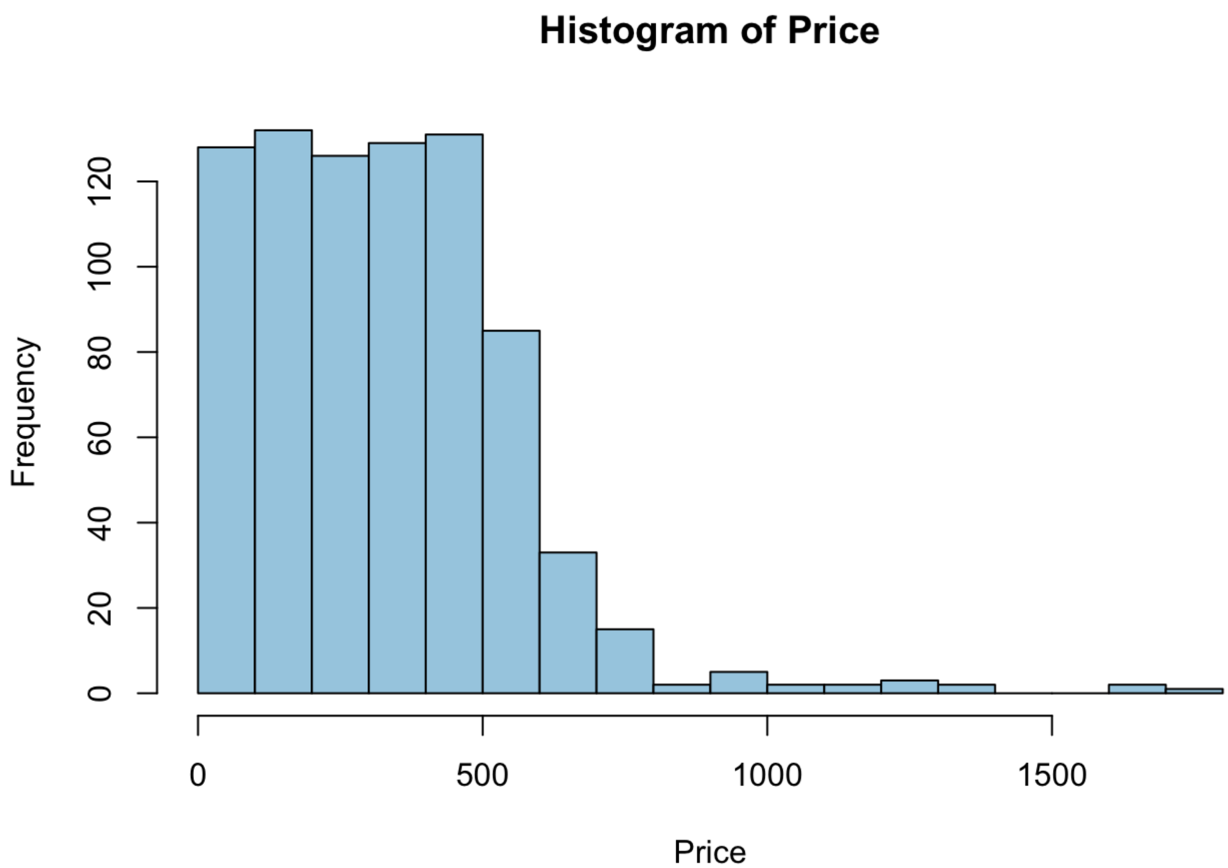

```
# Scent
scent_count <- perfume %>%
  group_by(scent) %>%
  summarise(count = n())
pie(scent_count$count, labels = scent_count$scent, main="Pie Chart of Scents", col = c(
"#A6CEE3", "#ffffff", "#CAB2D6", "#FFFF99", "#FDBF6F", "#FB9A99", "#B2DF8A"))
```



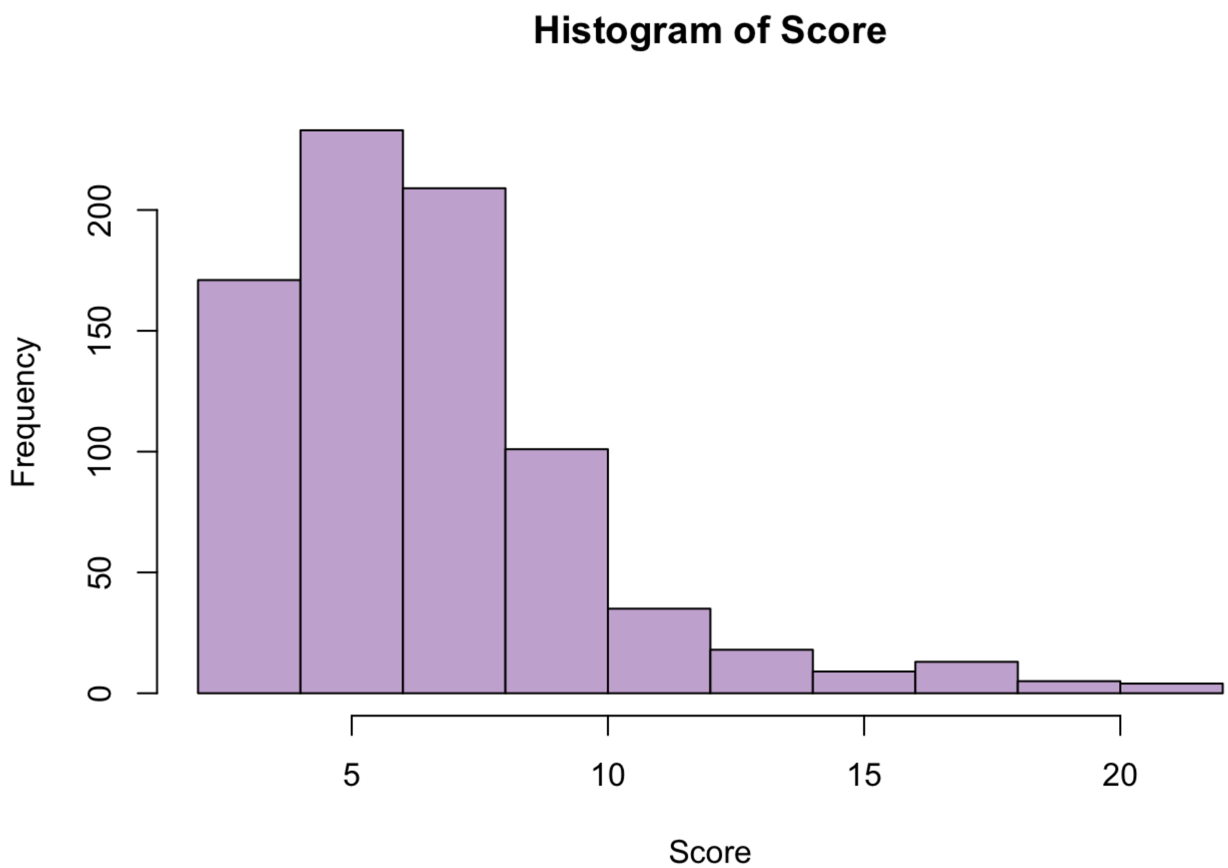

```
scent_slice <- perfume3 %>%
  group_by(is.fresh) %>%
  summarise(count = n())
pie(scent_slice$count, labels = c("Others", "Fresh"), main="Pie Chart of Merged Scents",
col = c("#A6CEE3", "#ffffff"))
```




```
# Price
hist(perfume$old_price, main = "Histogram of Price", xlab = "Price", col = "#A6CEE3", br
eaks = 20)
```




```
# score
hist(perfume$comp, main = "Histogram of Score", xlab = "Score", col = "#CAB2D6")
```

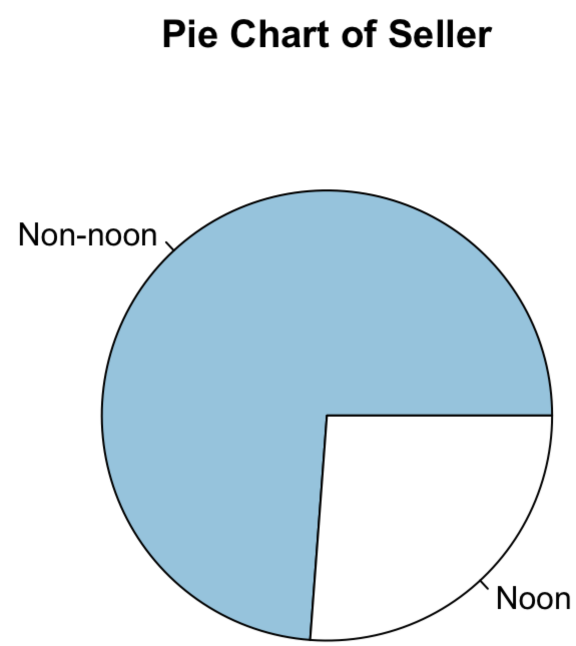



```
# Seller
p1 %>%
  group_by(seller) %>%
  summarise(count = n())
```

seller <chr>	count <int>
Abu Al Tayyeb Perfumes	11
Acacia Rose	5
Ahmed Mohamed Abbas Abbas Trading Est	1
Al-Najm	18
Alabeer	2
alkhalijiah perfume	4
alryhanaksa	1
AMLAQ	87
aRt Ti Ci	1
Asrar Aljamal	4

1-10 of 115 rows	Previous **1** 2 3 4 5 6 ... 12 Next

```
seller_slice <- perfume %>%
  group_by(is_noon) %>%
  summarise(count = n())
pie(seller_slice$count, labels = c("Non-noon", "Noon"), main="Pie Chart of Seller", col
 = c("#A6CEE3", "#ffffff"))
```



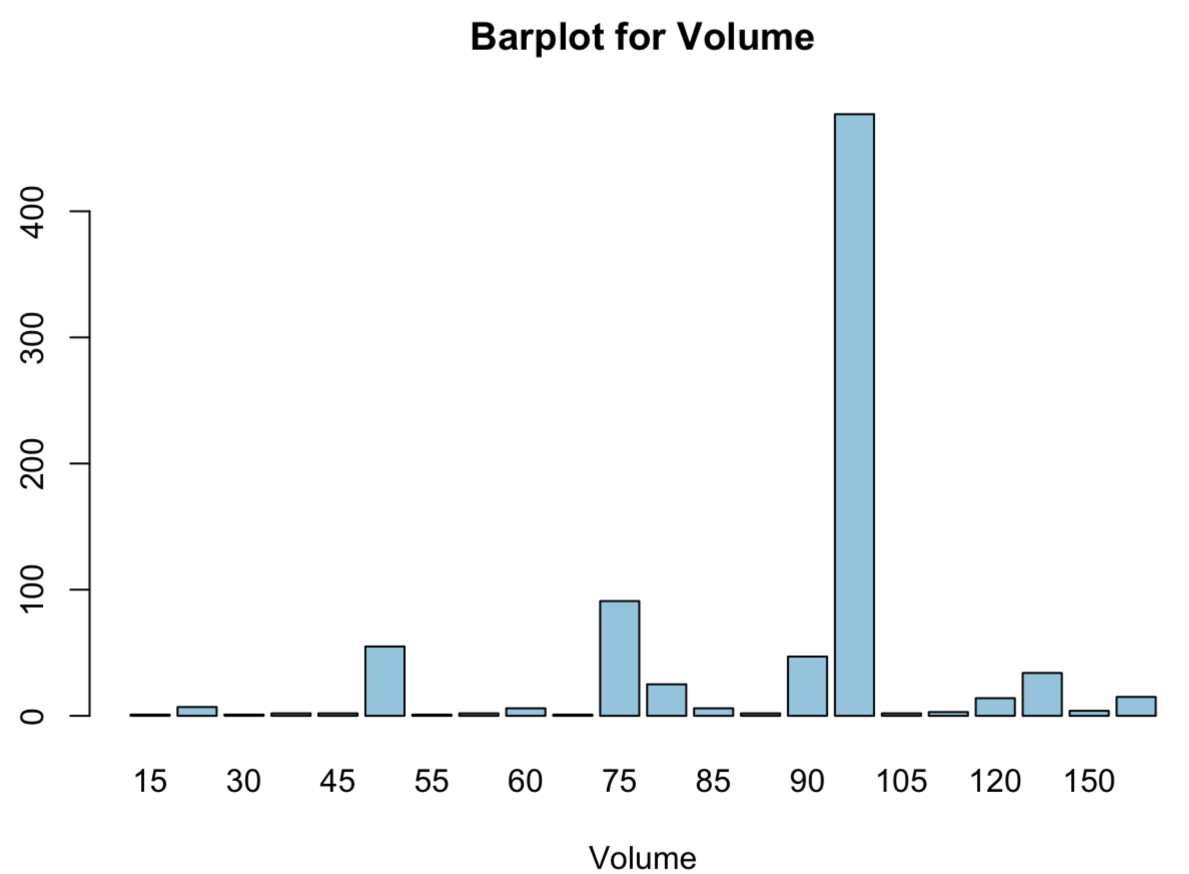

```
# volume
p1 %>%
  group_by(ml) %>%
  summarise(count = n())
```

ml <int>	count <int>
1	1
2	10
5	7
15	1
25	7

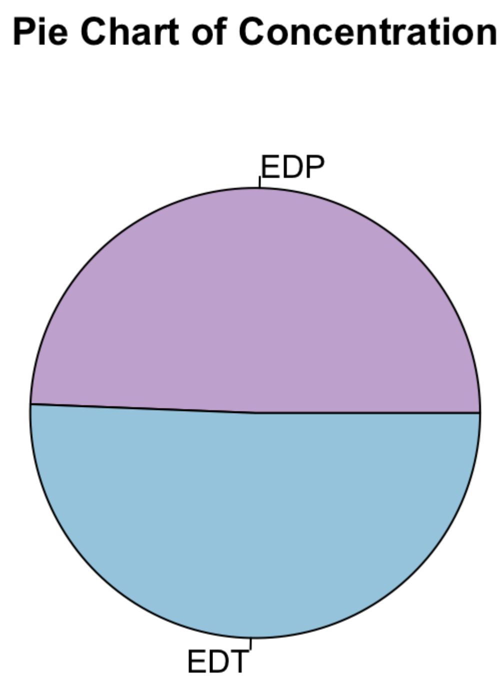
ml <int>	count <int>
30	1
35	1
40	2
45	2
50	62

1-10 of 26 rows   Previous **1** 2 3 Next

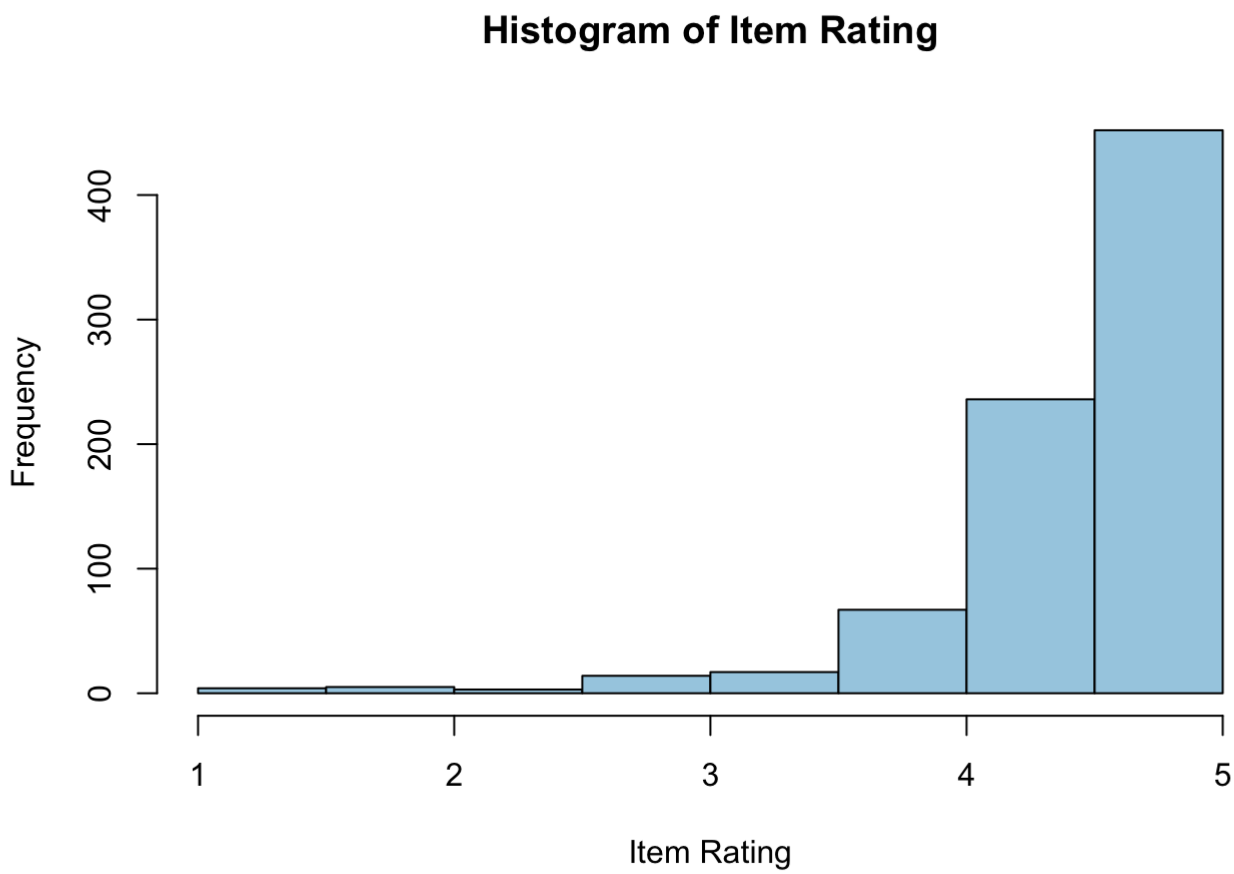
```
ml_count <- perfume %>%
  group_by(ml) %>%
  summarise(count = n())
barplot(ml_count$count, names.arg = ml_count$ml, main = "Barplot for Volume", xlab = "Vo
lume", col = "#A6CEE3")
```



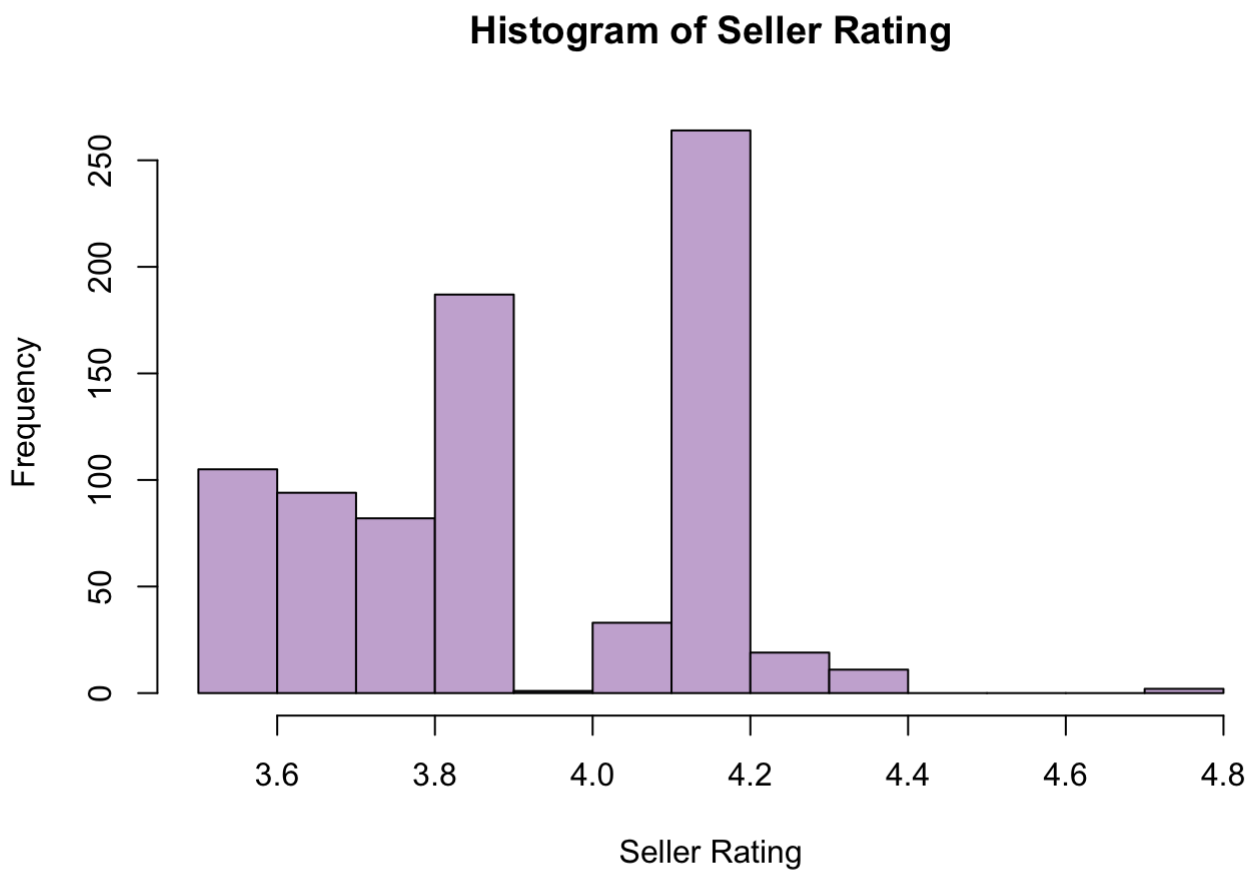
```
# conc
conc_slice <- perfume %>%
  group_by(conc) %>%
  summarise(count = n())
pie(conc_slice$count, labels = conc_slice$conc, main="Pie Chart of Concentration", col =
c("#CAB2D6", "#A6CEE3"))
```



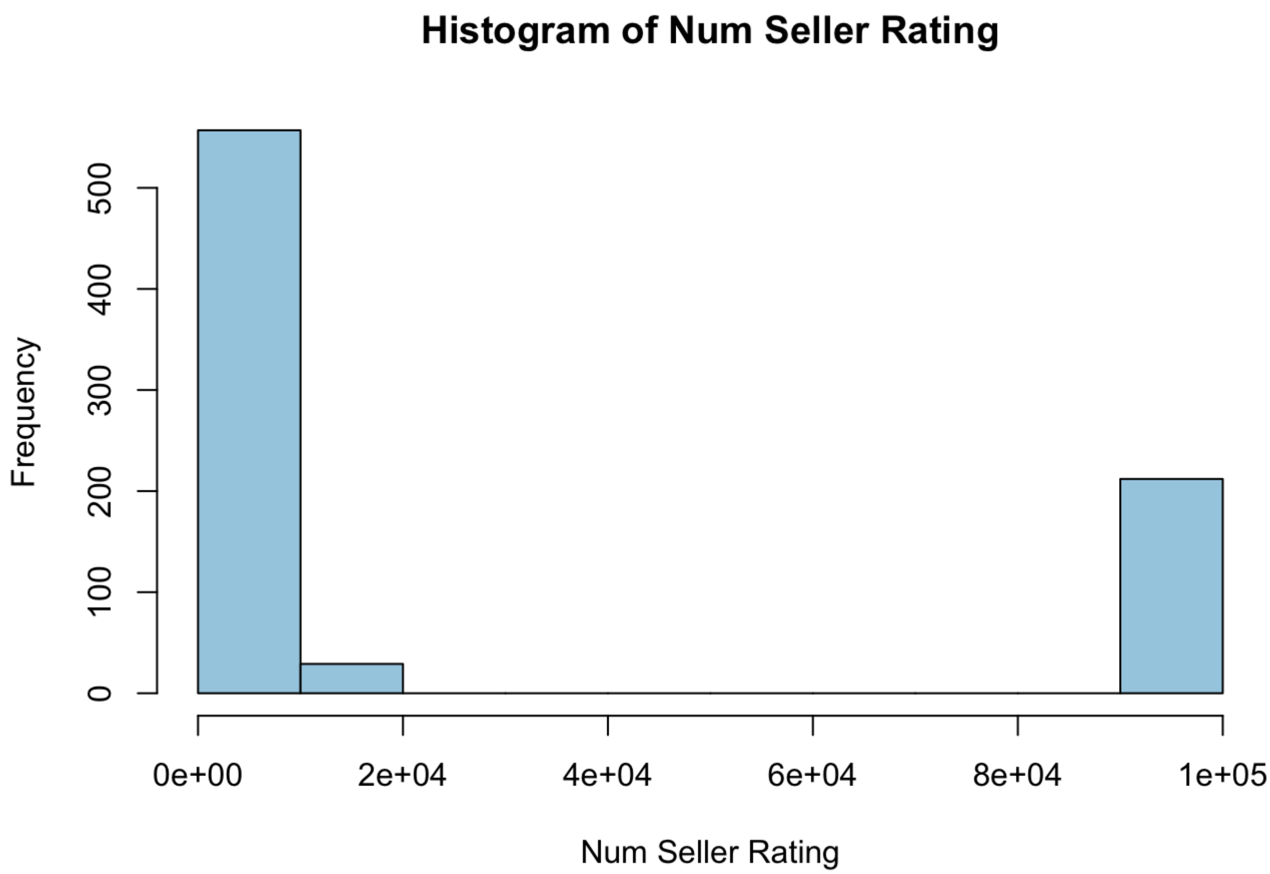

```
# Item rating
hist(perfume$item_rating, col = "#A6CEE3", main = "Histogram of Item Rating", xlab = "It
em Rating")
```



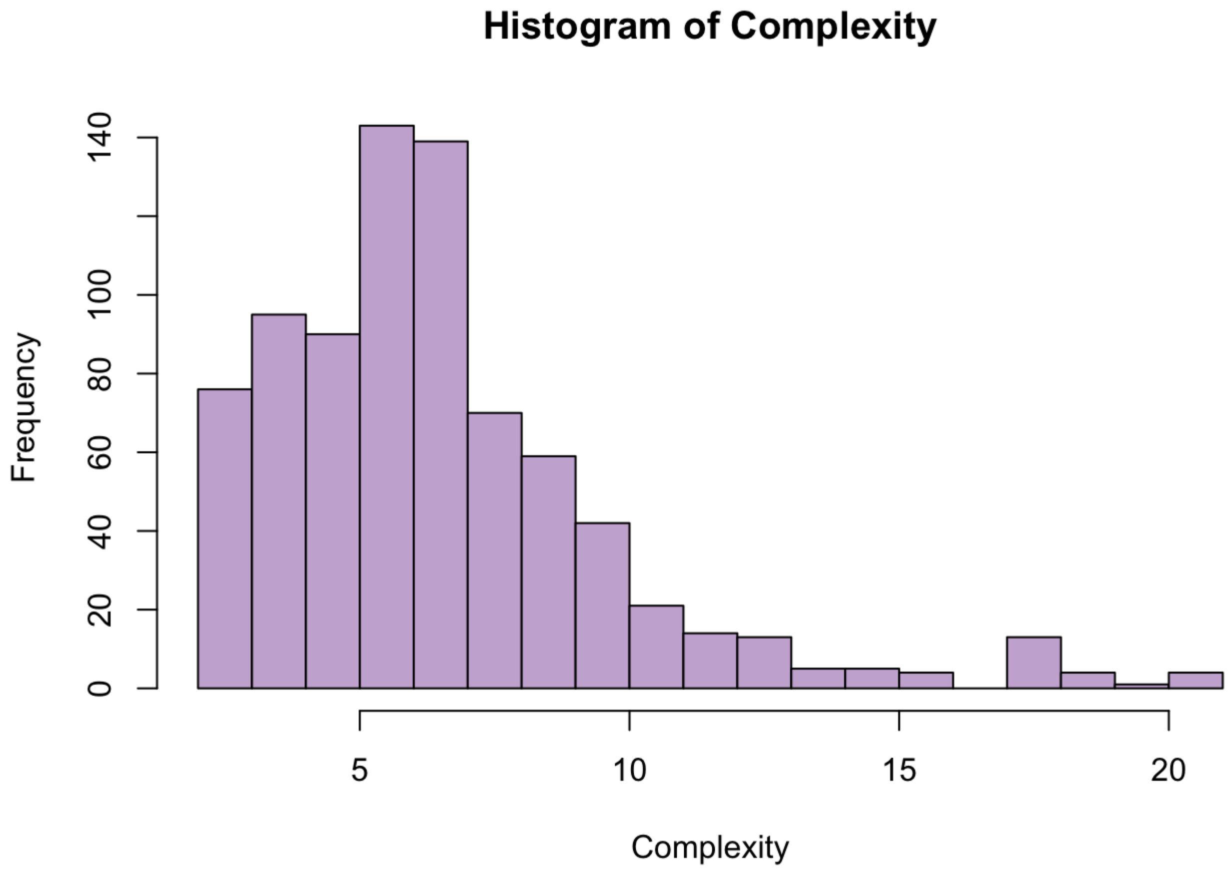

```
# seller rating
hist(perfume$seller_rating, col = "#CAB2D6", main = "Histogram of Seller Rating", xlab =
"Seller Rating")
```




```
# num seller rating
hist(perfume$num_sel_ratings, col = "#A6CEE3", main = "Histogram of Num Seller Rating",
 xlab = "Num Seller Rating")
```




```
# Comp
hist(perfume$comp, breaks = 20, main = "Histogram of Complexity", xlab = "Complexity", c
ol = "#CAB2D6")
```




```
as.data.frame(base_note) %>%
  group_by(base_note) %>%
  summarise(count = n())
```

base_note <chr>
absolutevanillaorchidtahitensis,praline,patchouli,papyrus
acaciawood
akigalawood
aldehyde,jasmine,whiteflowers
almonds,cedar,oakmoss,tonka,vanilla
amber
amber,benzoin,cedar,tonka
amber,cactus,cotton
amber,cactus,cottonflower
amber,cactus,cottonplantblossom

1-10 of 601 rows   Previous 1 2 3 4 5 6 ... 61 Next