SDS291_FinalProject

Importing Packages

```
library(moderndive)
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2
--

v ggplot2 3.4.0     v purrr   1.0.1
v tibble  3.1.8     v stringr 1.5.0
v tidyr   1.3.0     v forcats 1.0.0
v readr   2.1.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(Stat2Data)
```

Data Importing

```
lego_clean <- read.csv("lego_clean.csv")
lego_clean$fem <- factor(lego_clean$fem, labels = c("No", "Yes"))
lego_clean$masc <- factor(lego_clean$masc, labels = c("No", "Yes"))
lego_clean$neutral <- factor(lego_clean$neutral, labels = c("No", "Yes"))
```

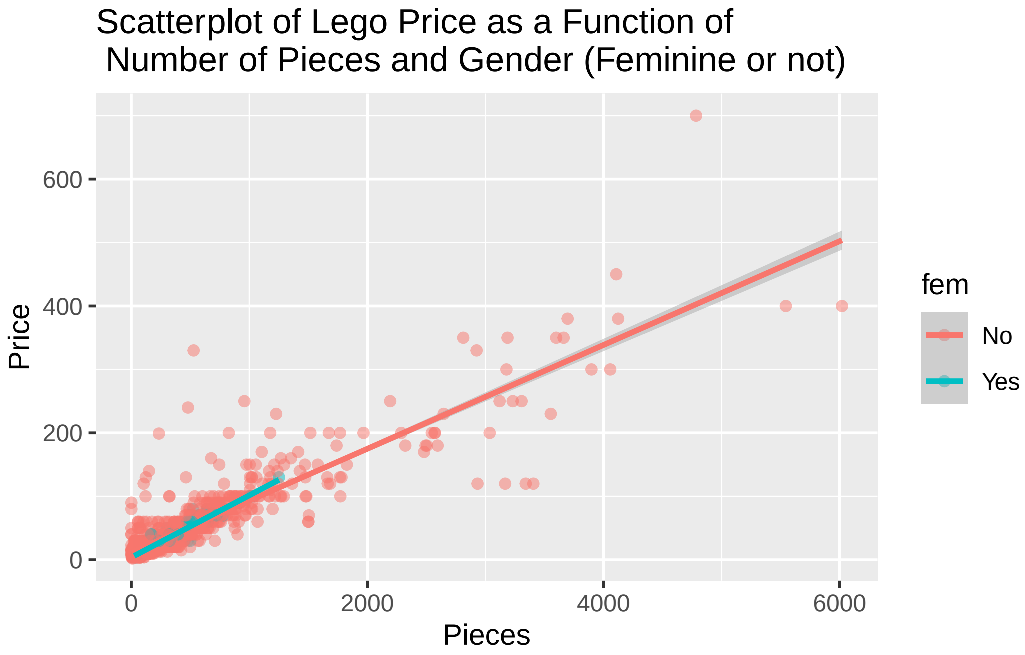
Exploratory Visualizations

```
price_piece_fem <- lm(Price ~ Pieces * fem, lego_clean)
ggplot(lego_clean, aes( x = Pieces, y = Price, color = fem)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") +
  labs(title = "Scatterplot of Lego Price as a Function of \n Number of Pieces and Gender
```

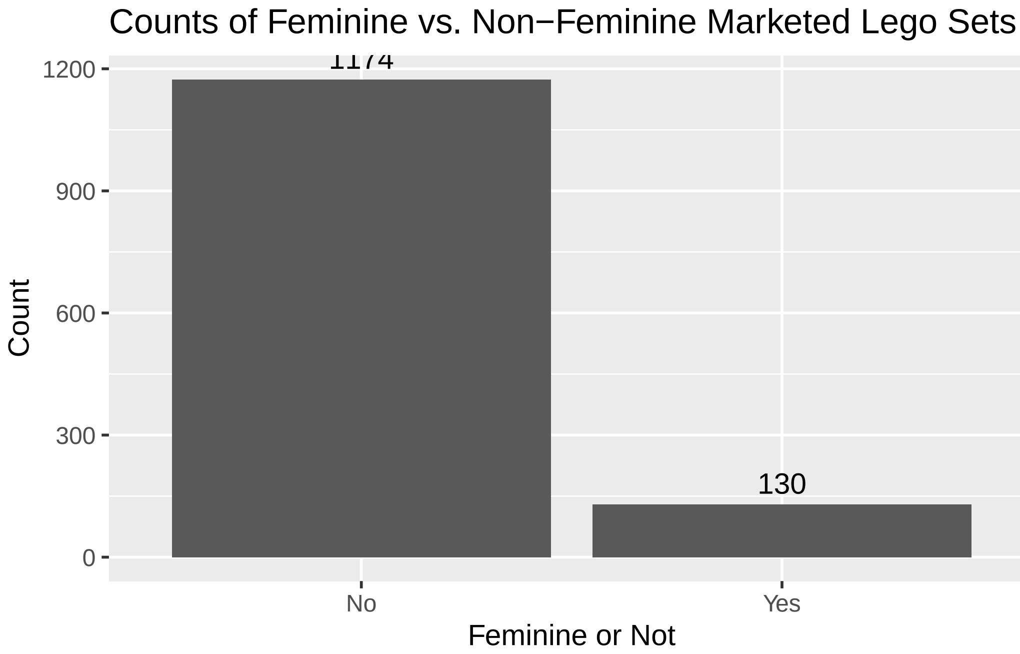
```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: Removed 239 rows containing non-finite values (`stat_smooth()`).
```

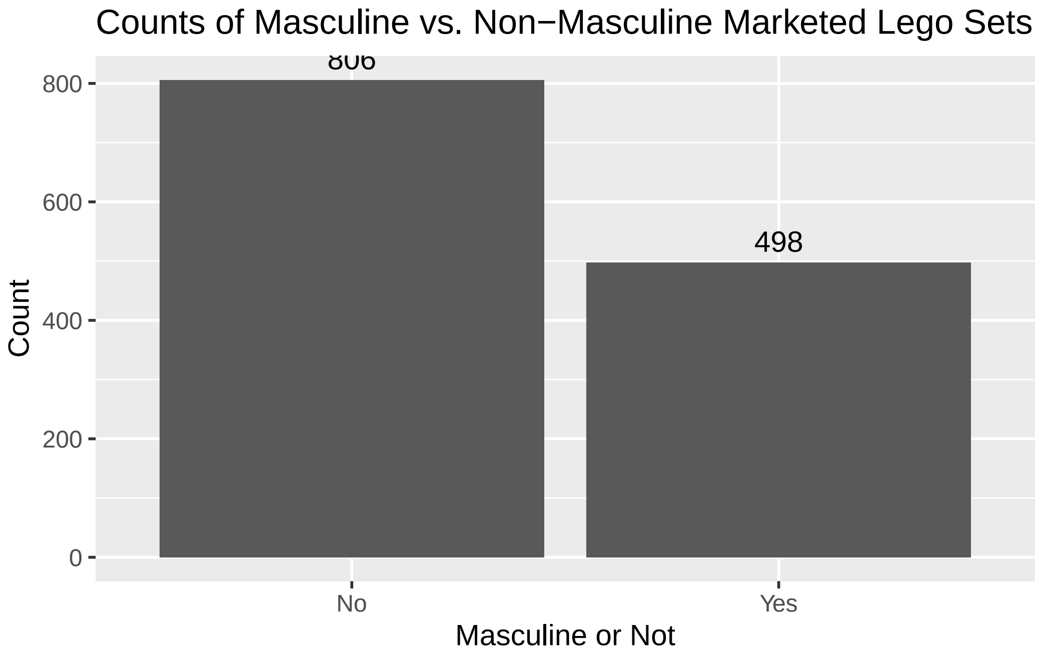
```
Warning: Removed 239 rows containing missing values (`geom_point()`).
```



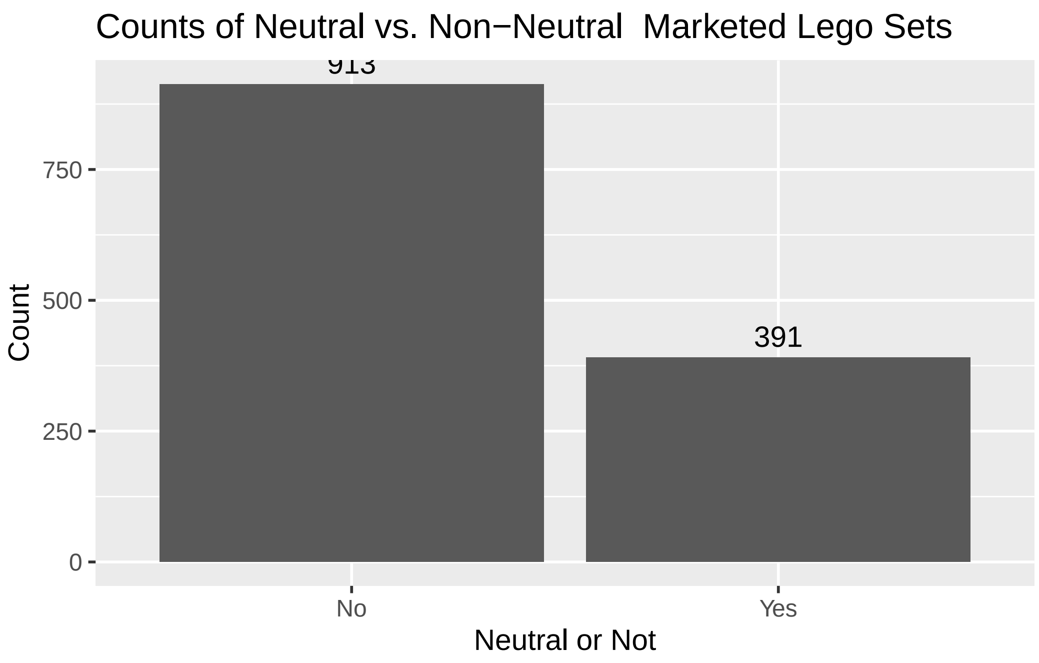
```
ggplot(lego_clean, aes(x = fem)) + geom_bar() + geom_text(stat = 'count', aes(label=after_
```



```
ggplot(lego_clean, aes(x = masc)) + geom_bar() + geom_text(stat = 'count', aes(label=after
```

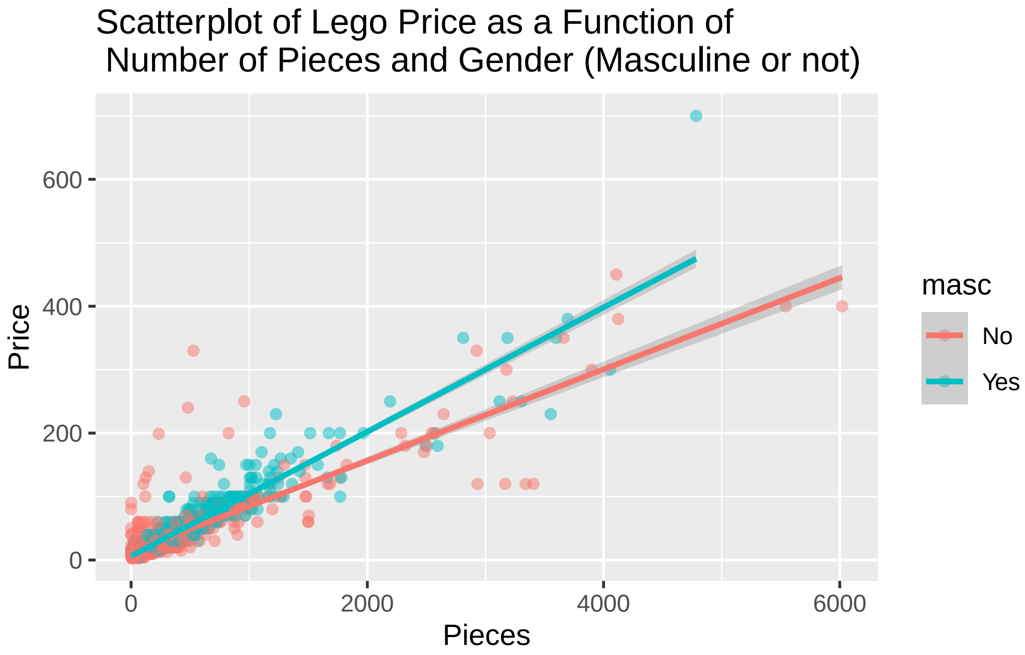


```
ggplot(lego_clean, aes(x = neutral)) + geom_bar() + geom_text(stat = 'count', aes(label=af
```

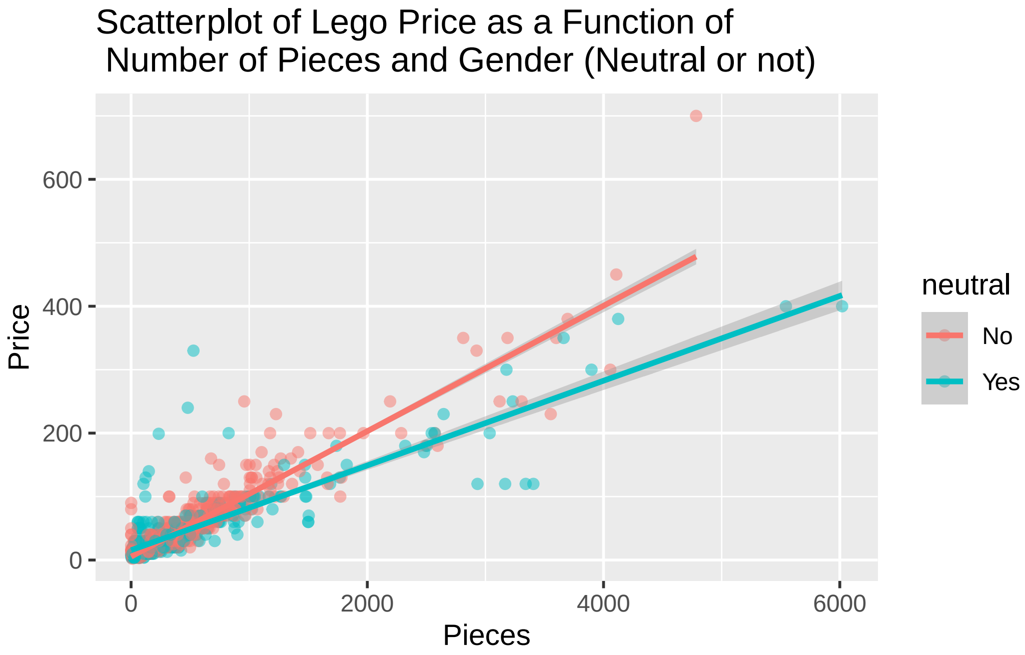


```r
price_piece_masc <- lm(Price ~ Pieces * masc, lego_clean)
ggplot(lego_clean, aes( x = Pieces, y = Price, color = masc)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") +
  labs(title = "Scatterplot of Lego Price as a Function of \n Number of Pieces and Gender
```

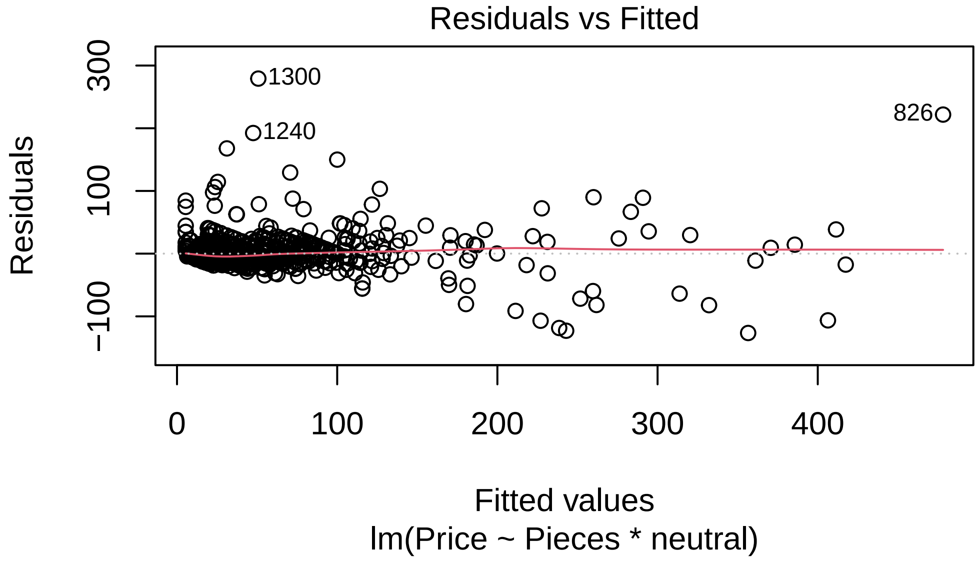
```
`geom_smooth()` using formula = 'y ~ x'

Warning: Removed 239 rows containing non-finite values (`stat_smooth()`).

Warning: Removed 239 rows containing missing values (`geom_point()`).
```

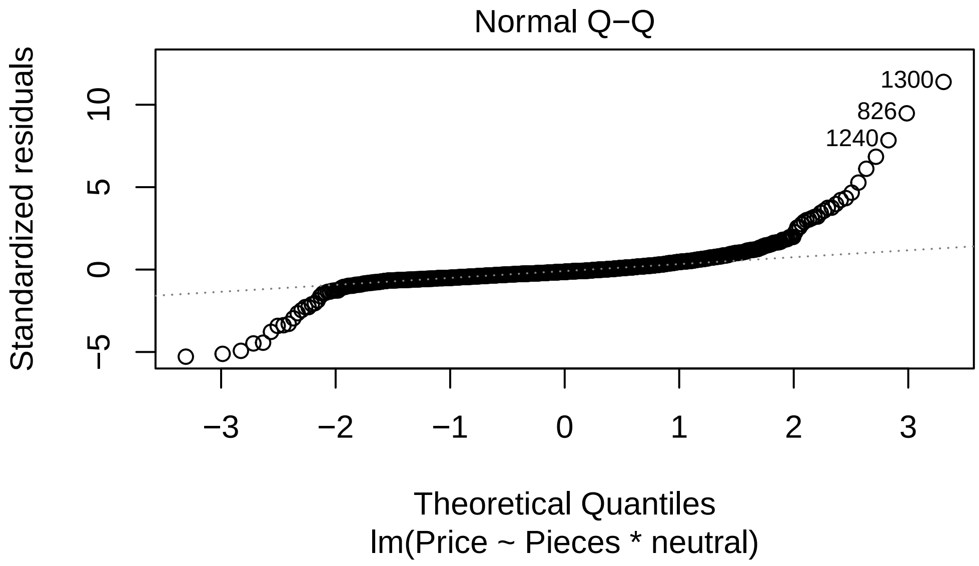


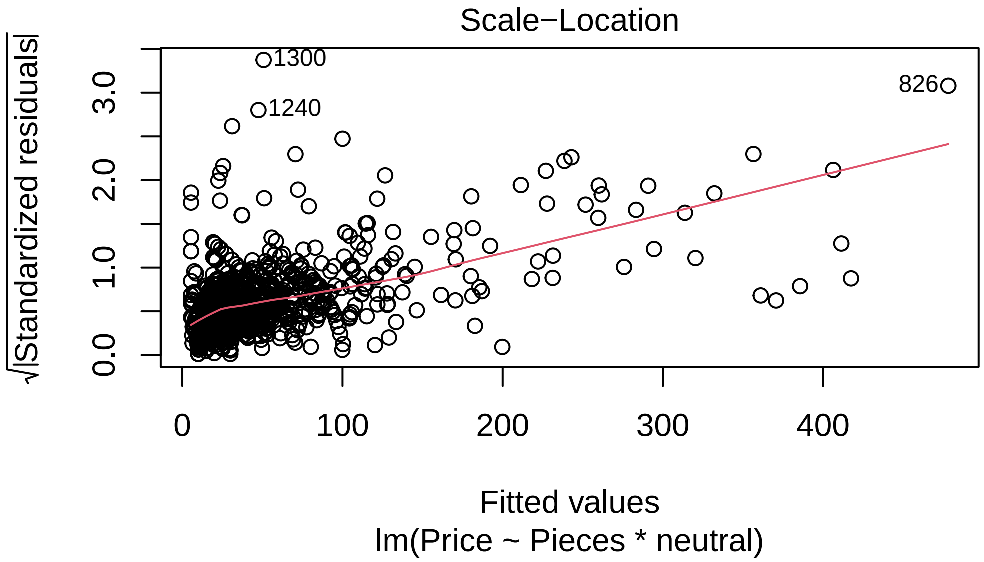
```
price_piece_neutral <- lm(Price ~ Pieces * neutral, lego_clean)
ggplot(lego_clean, aes( x = Pieces, y = Price, color = neutral)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") +
  labs(title = "Scatterplot of Lego Price as a Function of \n Number of Pieces and Gender
```

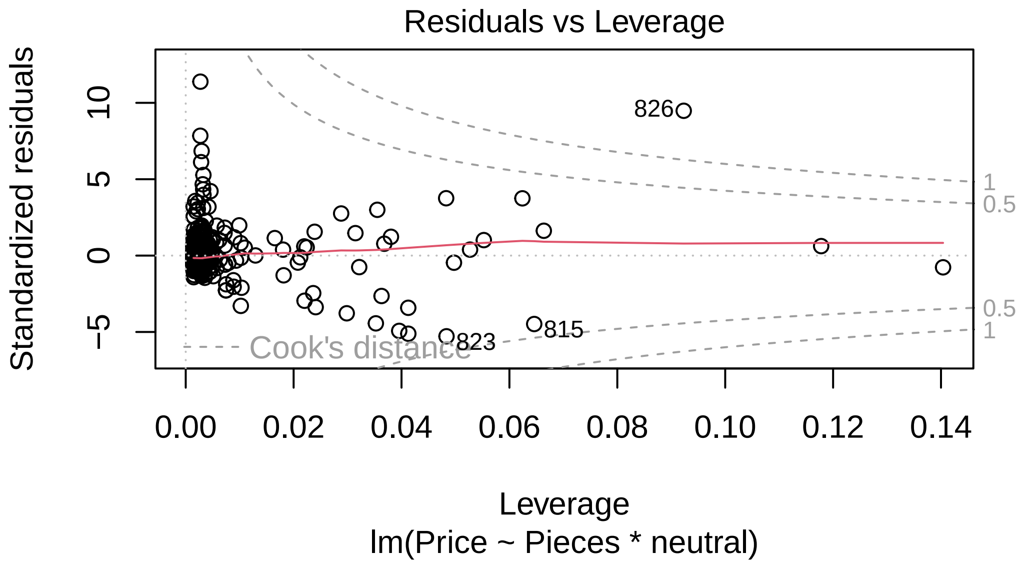


```
plot(price_piece_neutral)
```

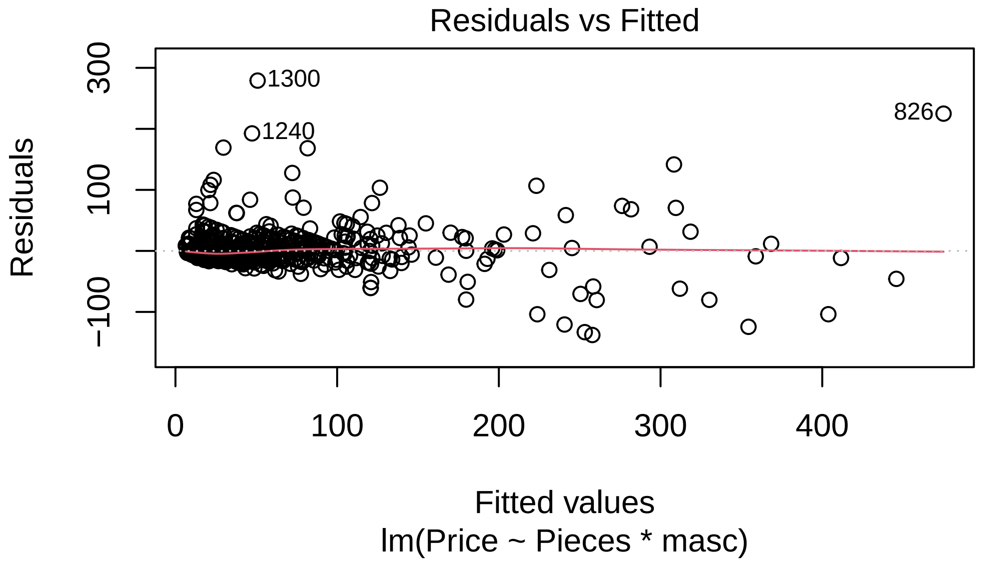


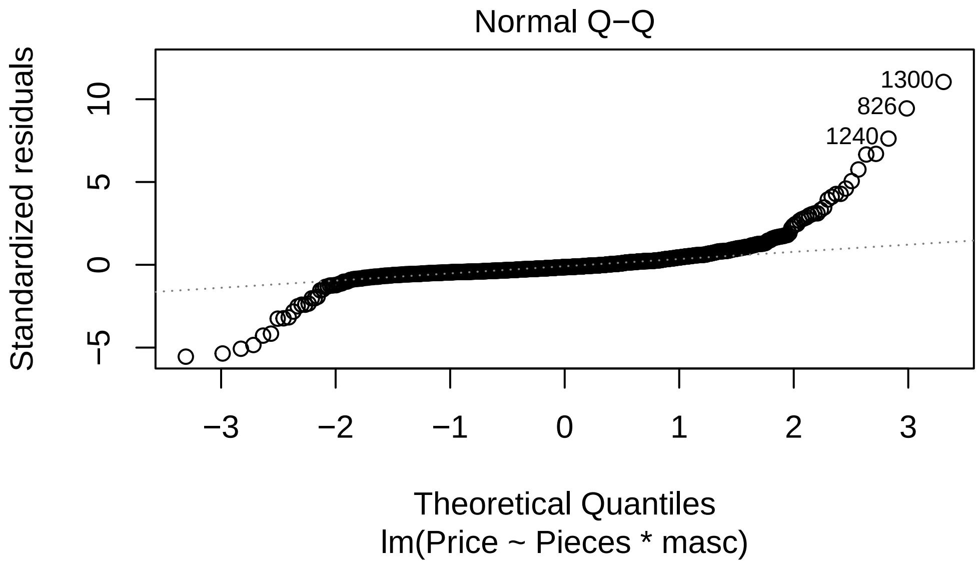

Normal Q−Q
Standardized residuals
−5
0
5
10
1300
826
1240
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Price ~ Pieces * neutral)

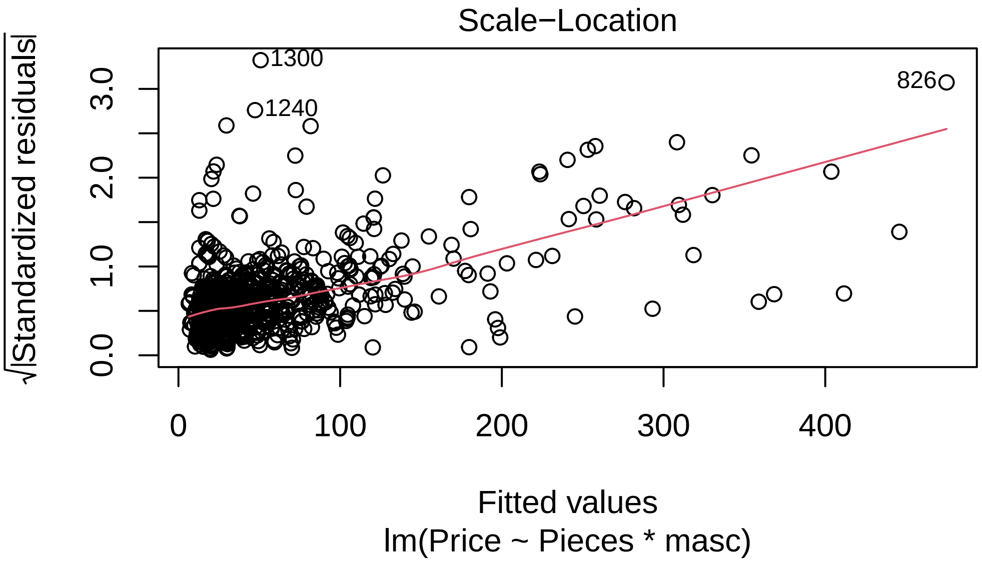

Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
1300
1240
826
0
100
200
300
400
Fitted values
lm(Price ~ Pieces * neutral)

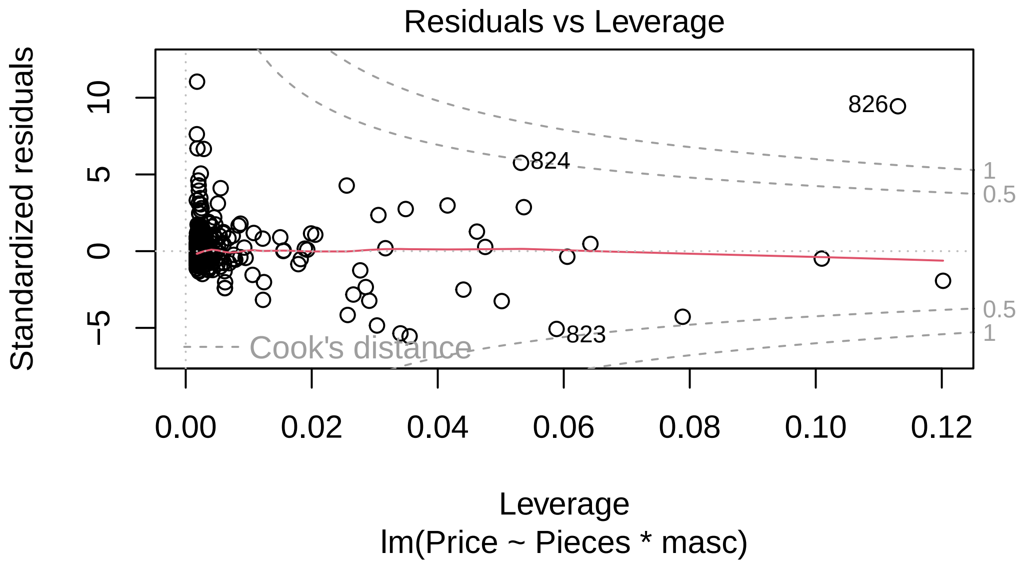


```
plot(price_piece_masc)
```

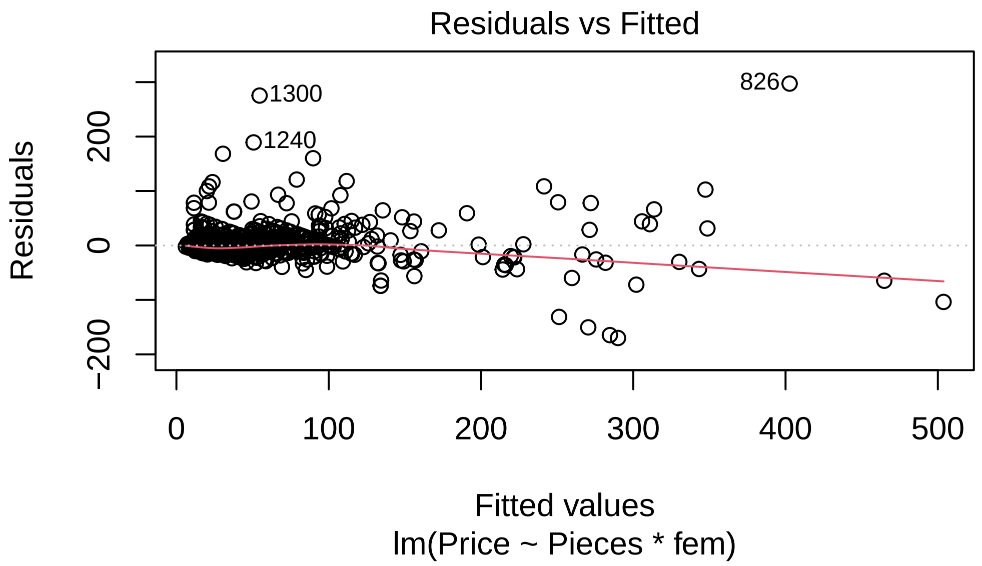


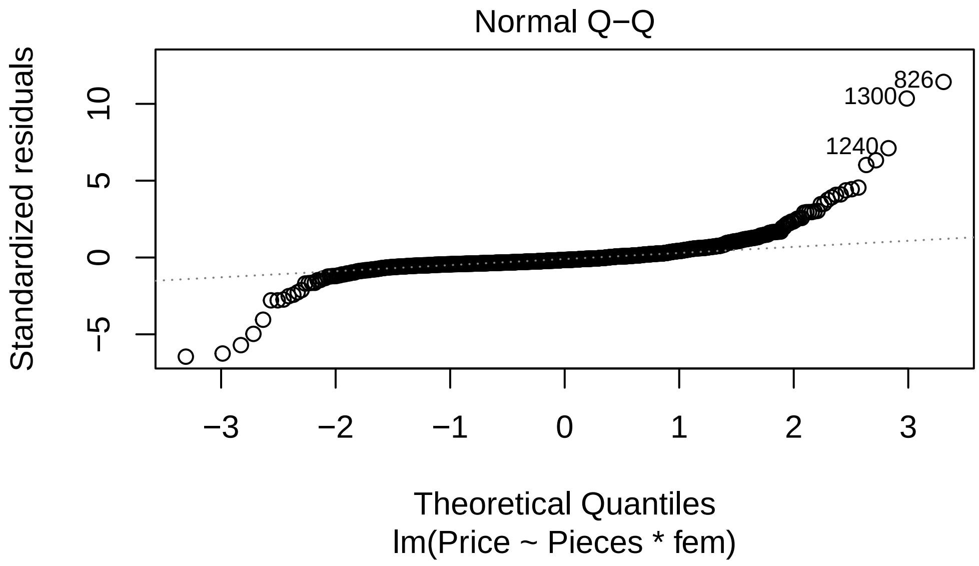

Normal Q−Q
Standardized residuals
−5
0
5
10
−3
−2
−1
0
1
2
3
1300
826
1240
Theoretical Quantiles
lm(Price ~ Pieces * masc)

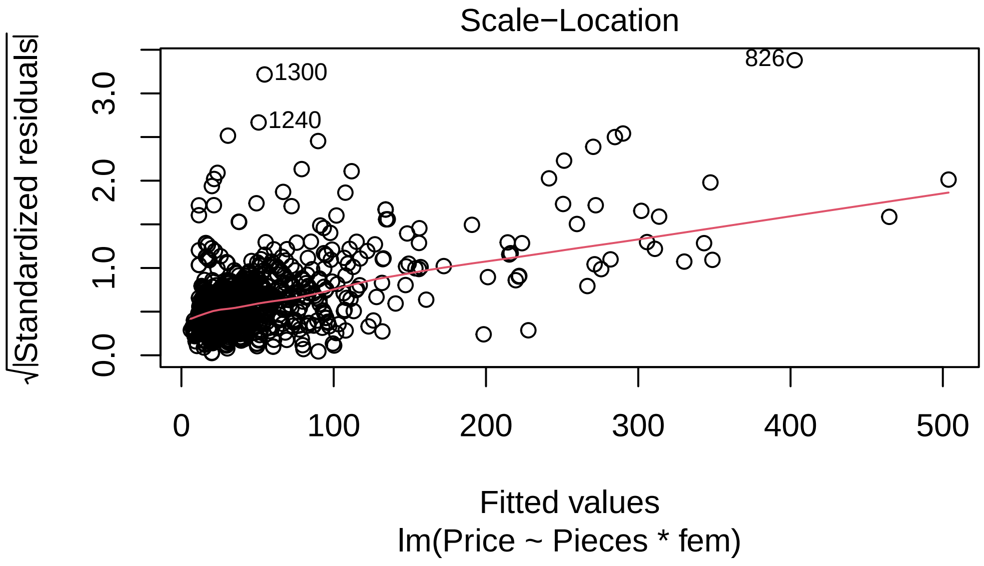

Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
0
100
200
300
400
1300
1240
826
Fitted values
lm(Price ~ Pieces * masc)

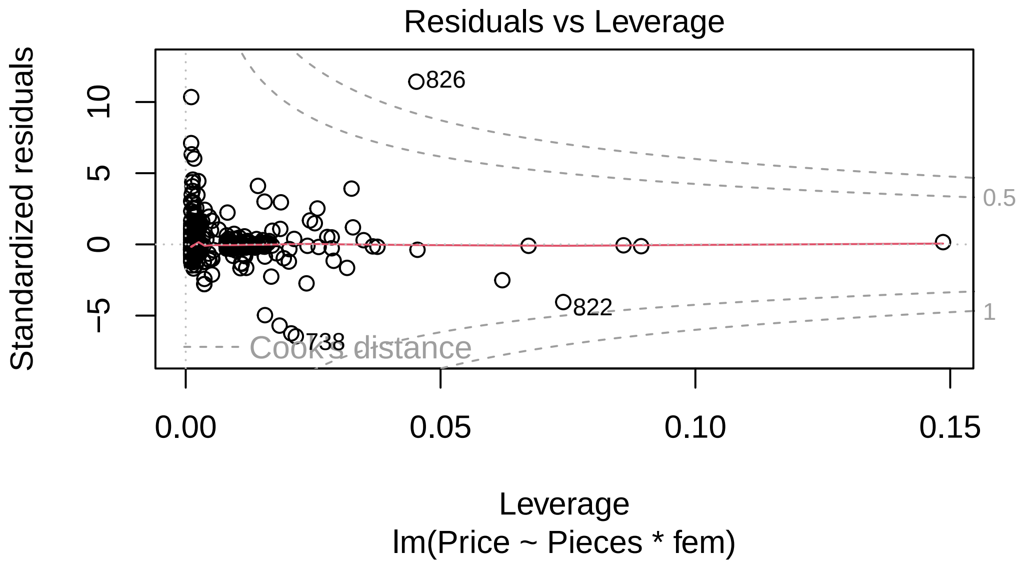


```
plot(price_piece_fem)
```

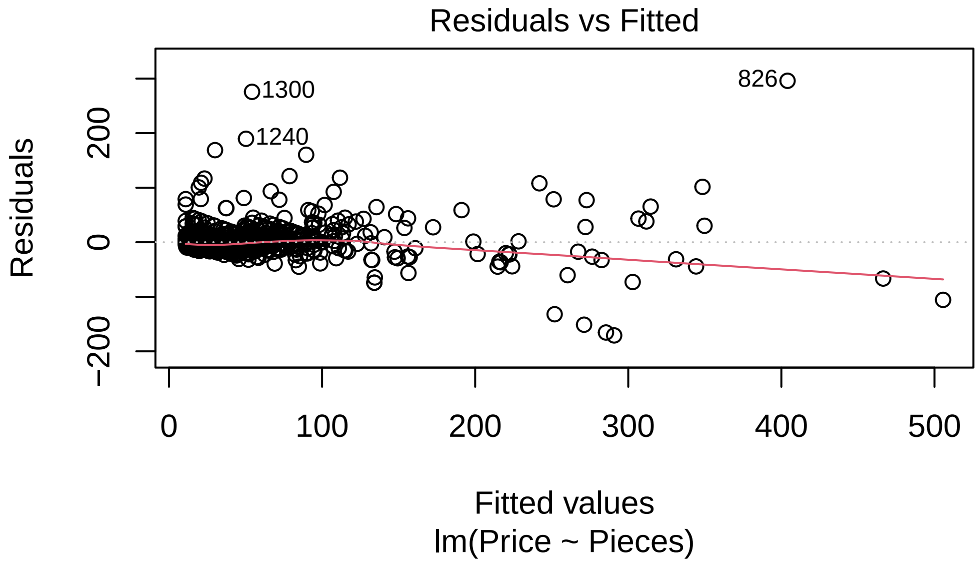


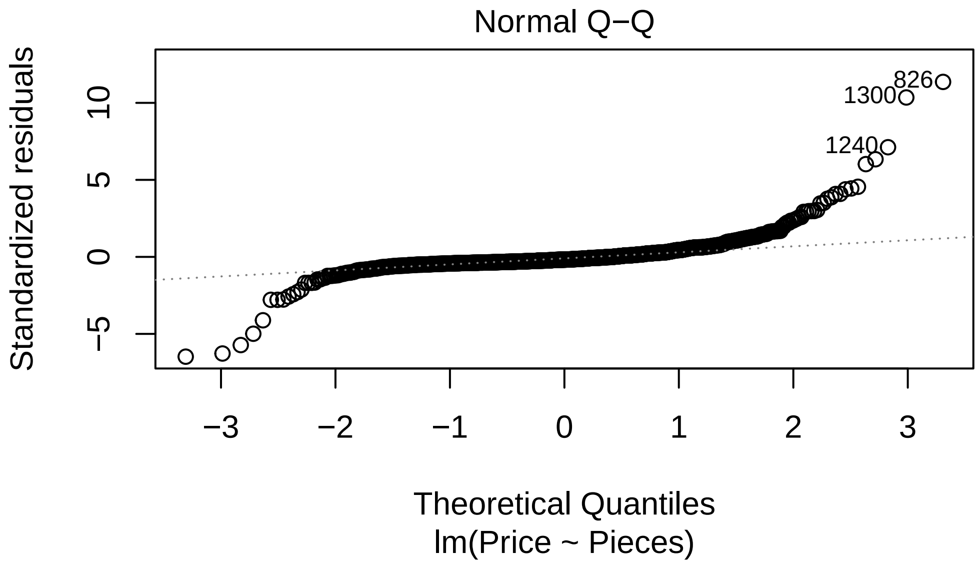

Normal Q−Q
Standardized residuals
−5
0
5
10
−3
−2
−1
0
1
2
3
826
1300
1240
Theoretical Quantiles
lm(Price ~ Pieces * fem)

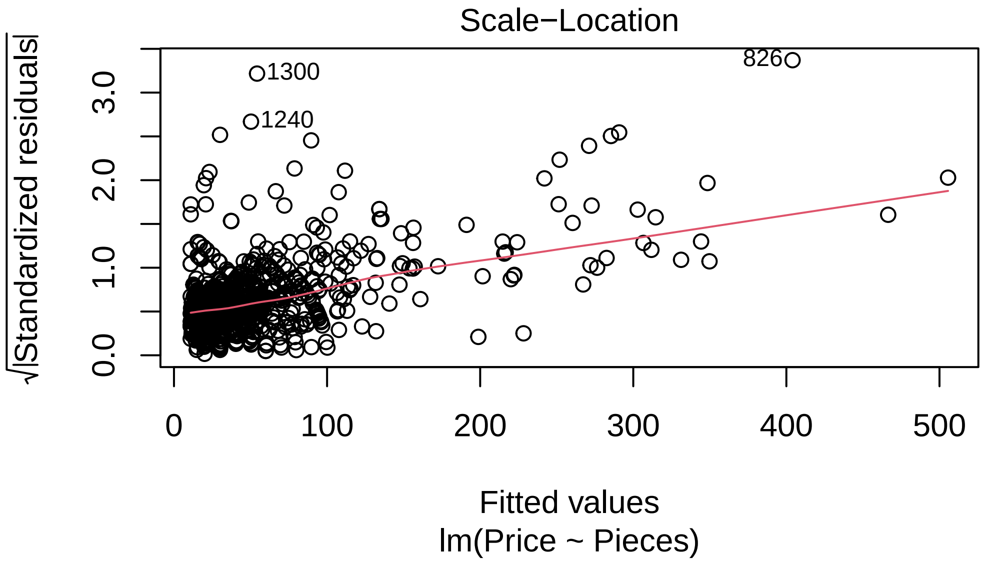

Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
0
100
200
300
400
500
1300
1240
826
Fitted values
lm(Price ~ Pieces * fem)

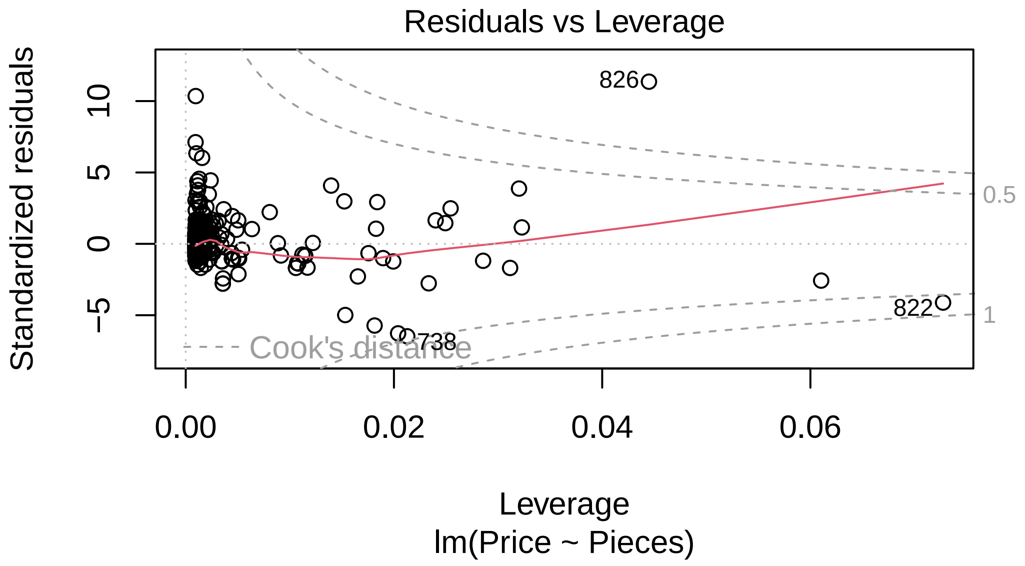


```
price_piece <- lm(Price ~ Pieces, data = lego_clean)
plot(price_piece)
```




Normal Q−Q
Standardized residuals
−5
0
5
10
826
1300
1240
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(Price ~ Pieces)


Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
1300
1240
826
0
100
200
300
400
500
Fitted values
lm(Price ~ Pieces)



```
get_regression_table(price_piece)
```

```
# A tibble: 2 x 7
  term      estimate std_error statistic p_value lower_ci upper_ci
  <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept   10.9       0.986      11.0       0     8.93   12.8
2 Pieces       0.082     0.001      64.3       0     0.08    0.085
```

```
get_regression_table(price_piece_fem)
```

```
# A tibble: 4 x 7
  term          estimate std_error statistic p_value lower_ci upper_ci
  <chr>            <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept       11.4       1.05      10.9    0         9.37    13.5
2 Pieces           0.082     0.001     63.3    0         0.079    0.084
3 fem: Yes        -7.69      3.80      -2.02   0.043   -15.2     -0.228
4 Pieces:femYes    0.016     0.01       1.56   0.12     -0.004    0.036
```

```
get_regression_table(price_piece_masc)
```

```
# A tibble: 4 x 7
  term           estimate std_error statistic p_value lower_ci upper_ci
  <chr>             <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept        12.8       1.23      10.4    0        10.4     15.3
2 Pieces            0.072     0.002     46.4    0         0.069    0.075
3 masc: Yes        -6.62      1.90      -3.48   0.001  -10.4     -2.89
4 Pieces:mascYes    0.026     0.002     10.5    0         0.021    0.031
```

```
get_regression_table(price_piece_neutral)
```

```
# A tibble: 4 x 7
  term              estimate std_error statistic p_value lower_ci upper_ci
  <chr>                <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept             5.35      1.16       4.59       0     3.06     7.63
2 Pieces                0.099     0.002     58.4        0     0.096    0.102
3 neutral: Yes         10.2       1.89       5.37       0     6.45    13.9
4 Pieces:neutralYes    -0.032     0.002    -13.6        0    -0.037   -0.027
```

Model Comparison

compare price_piece model (consider this are nested model) to the models that look at both price and gender (this would be the full model) null hypothesis: the nested model that only looks at price as a function of pieces is enough (coefficent of sex has no effect = 0) *alternative hypothesis: need the full model (coefficent of sex has an effect)

```
anova(price_piece, price_piece_neutral)
```

```
Analysis of Variance Table

Model 1: Price ~ Pieces
Model 2: Price ~ Pieces * neutral
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)    
1   1063 755013                                  
2   1061 639726  2    115287 95.603 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

*wheter a set is marked as gender neutral or not has an effect on price (the full model is better the fit) - Gender neutrality is a necessary component of the model

```
anova(price_piece, price_piece_masc)
```

```
Analysis of Variance Table

Model 1: Price ~ Pieces
Model 2: Price ~ Pieces * masc
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1   1063 755013
2   1061 678766  2     76248 59.593 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

*whether a set is marked as masculine or not has an effect on price (the full model is a better fit) - masculinity is a necessary component of the model.

```
anova(price_piece, price_piece_fem)
```

```
Analysis of Variance Table

Model 1: Price ~ Pieces
Model 2: Price ~ Pieces * fem
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   1063 755013
2   1061 752110  2    2903.1 2.0477 0.1295
```

*fail to reject the null hypothesis and conclude the reduced model (price_piece) explains more variability in the dataset than a model accounting for femine marketing.

LINE Violations

*all models violate the normality condition at the extremity points. Since we are primarily doing a model comparison, this violation carries through and a model comparison should still hold relatively well. This will impact generalizability to all lego sets (make it less generalizable).

Comparing The Interaction Models to Each other

```
get_regression_summaries(price_piece_neutral)
```

```
# A tibble: 1 x 9
  r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
      <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
1     0.827         0.826  601.  24.5  24.6     1688.       0     3  1065
```

```
get_regression_summaries(price_piece_masc)
```

```
# A tibble: 1 x 9
  r_squared adj_r_squared   mse  rmse sigma statistic p_value    df  nobs
      <dbl>         <dbl> <dbl> <dbl> <dbl>     <dbl>   <dbl> <dbl> <dbl>
1     0.816         0.816  637.  25.2  25.3     1570.       0     3  1065
```

*the adjusted r squared values indicate that the neutral model is a better predictor of price when controlling for pieces. However, the difference in the adjusted r squared values is very low (0.01). Since the number lego sets marketed to a gender neutral audience is much higher than ones marked to a masculine audience, it is likely that both of these models to a similarly good job of predicting price when controlling for number of pieces.

Interpreting Coefficents for Masculine Model:

- Intercept: For a non-masculine model with 0 pieces the predicted price is 12.84. *Pieces: For each additional piece in a non-masuline set, the predicted price increase is 0.07.
- masc: yes: For a masculine model with 0 pieces, the predicted price is -6.62.
- Pieces: mascYes: For each additional piece in a masculine set, the predicted price increase is 0.03.