Daisy Cutter Lego Land Lab SST-115

```
blocks <- read.csv('https://vinnys-classes.github.io/data/legos_data.csv')

blocks <- subset(blocks,
                 Theme != 'Duplo')
head(blocks)
```

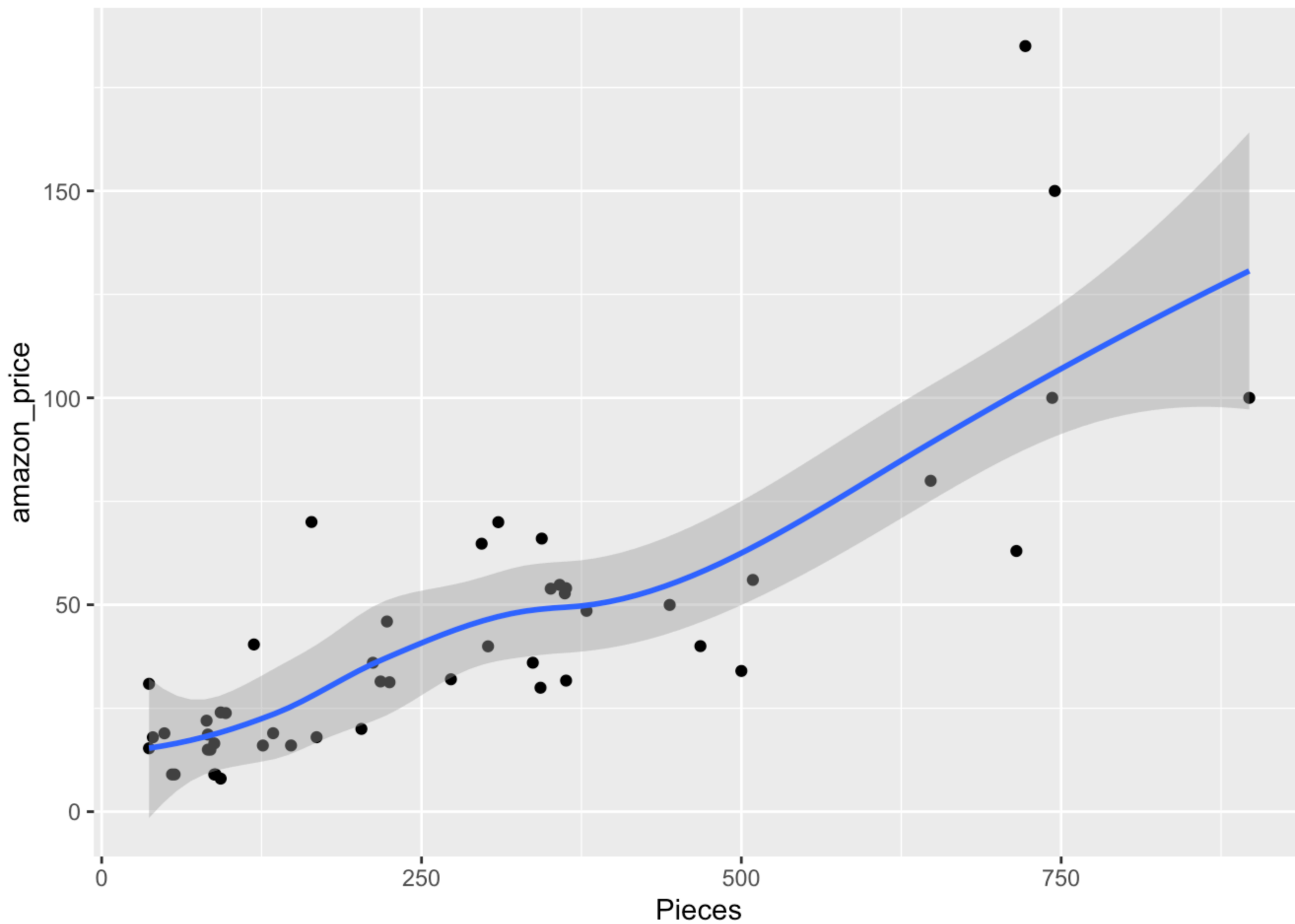
```
##    Item_Number                    Set_Name   Theme Pieces Year Pages
## 26       41330 Stephanie's Soccer Practice Friends    119 2018    48
## 27       41333    Olivia's Mission Vehicle Friends    223 2018    84
## 28       41335            Mia's Tree House Friends    351 2018   120
## 29       41340            Friendship House Friends    722 2018   164
## 30       41353     Friends Advent Calendar Friends    500 2018     4
## 31       41356       Stephanie's Heart Box Friends     85 2019    32
##    Minifigures Packaging Unique_Pieces  Size amazon_price age
## 26           1       Box            78 Small        40.40   6
## 27           1       Box           106 Small        45.95   6
## 28           2       Box           151 Small        53.88   6
## 29           3       Box           309 Small       184.99   6
## 30          NA       Box           202 Small        34.00   6
## 31           1      <NA>            36 Small        14.99   6
```

Question 1

Using ggplot, please create a scatterplot with Pieces on the x-axis and amazon_price on the y-axis. Add a best-fit-line using geom_smooth.

```
library(ggplot2)
ggplot(data = blocks) +
  geom_point(mapping = aes(Pieces, amazon_price)) +
  geom_smooth(mapping=aes(Pieces, amazon_price))
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

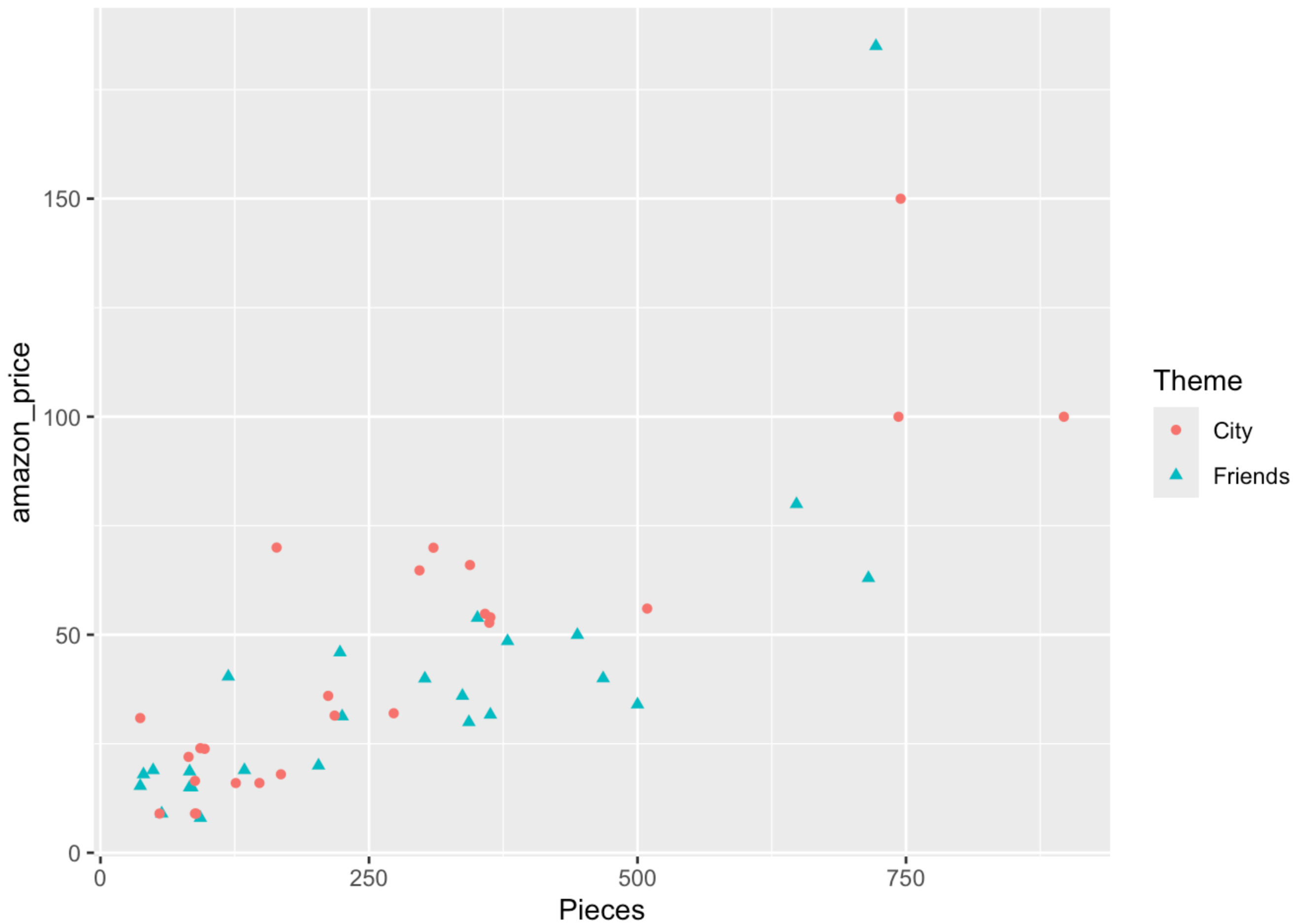


Question 2

Copy your graph from question 1 down here. Color and shape the points by Theme.

```
library(ggplot2)
```

```
ggplot(data = blocks) +
  geom_point(mapping = aes(x = Pieces, y = amazon_price, shape=Theme, color=Theme))
```




```
geom_smooth(mapping=aes(Pieces, amazon_price))
```

```
## mapping: x = ~Pieces, y = ~amazon_price
## geom_smooth: na.rm = FALSE, orientation = NA, se = TRUE
## stat_smooth: na.rm = FALSE, orientation = NA, se = TRUE
## position_identity
```

Question 3

Comment on the differences between the the city and friends group of legos.

The City group seems to have a larger spread, and more outliers towards the right side of the graph with some larger sets. The Friends group seems to be overall smaller.

Question 4

Fit a linear model using the lm() function with amazon_price as the response and Pieces as the explanatory variable.

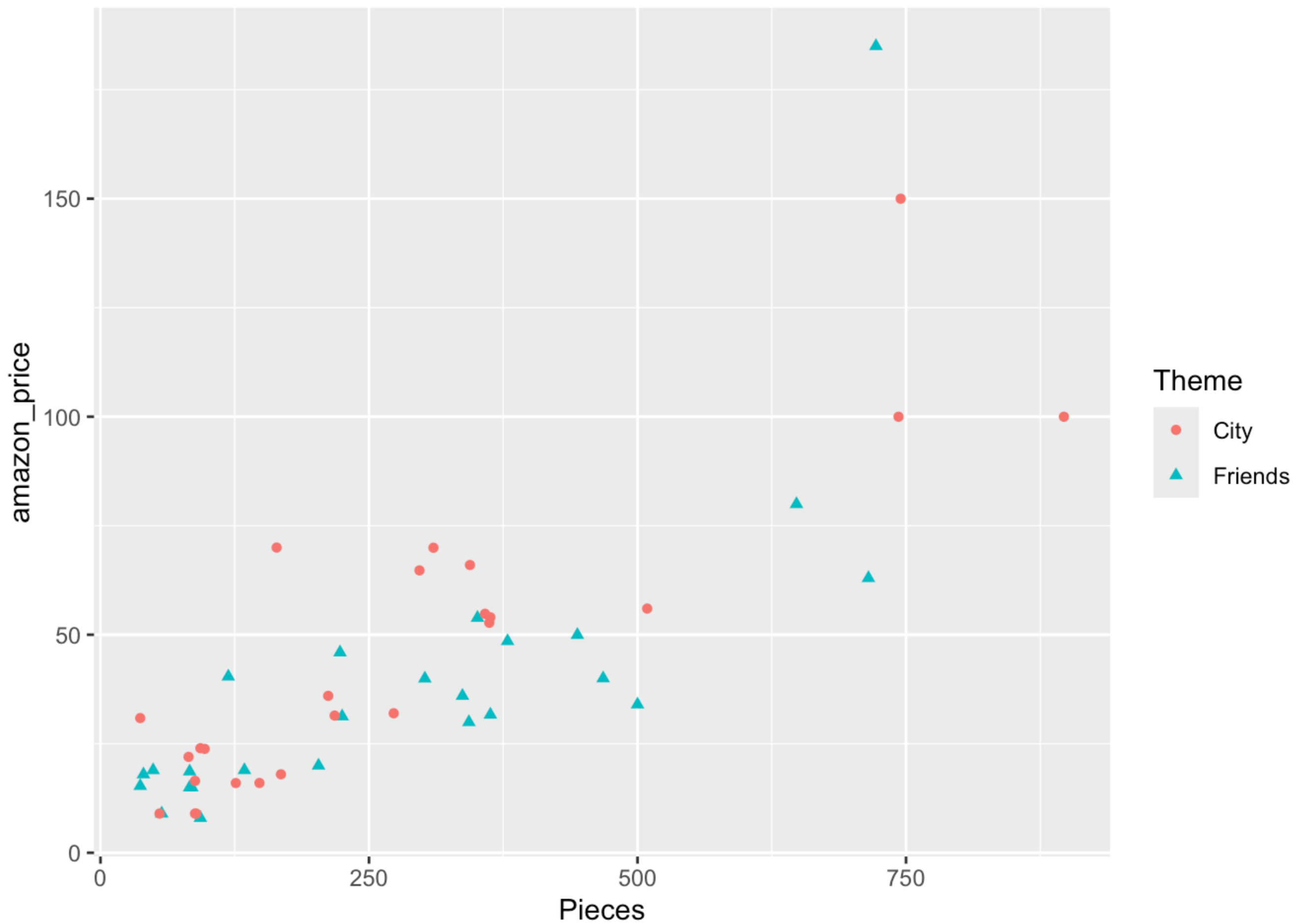
```
lm_legos = lm(amazon_price ~ Pieces,
              data = blocks)
summary(lm_legos)
```

```
##
## Call:
## lm(formula = amazon_price ~ Pieces, data = blocks)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.484  -9.476  -2.605   5.045  86.060
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.41779    4.66805   1.375    0.176
## Pieces       0.12813    0.01323   9.688 7.08e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.41 on 48 degrees of freedom
## Multiple R-squared:  0.6616, Adjusted R-squared:  0.6546
## F-statistic: 93.86 on 1 and 48 DF,  p-value: 7.077e-13
```

Question 5

Plot the residuals from question 3. Color and shape the points by Theme.

```
ggplot(data = blocks) +
  geom_point(mapping = aes(x = Pieces, y = amazon_price, shape=Theme, color=Theme))
```



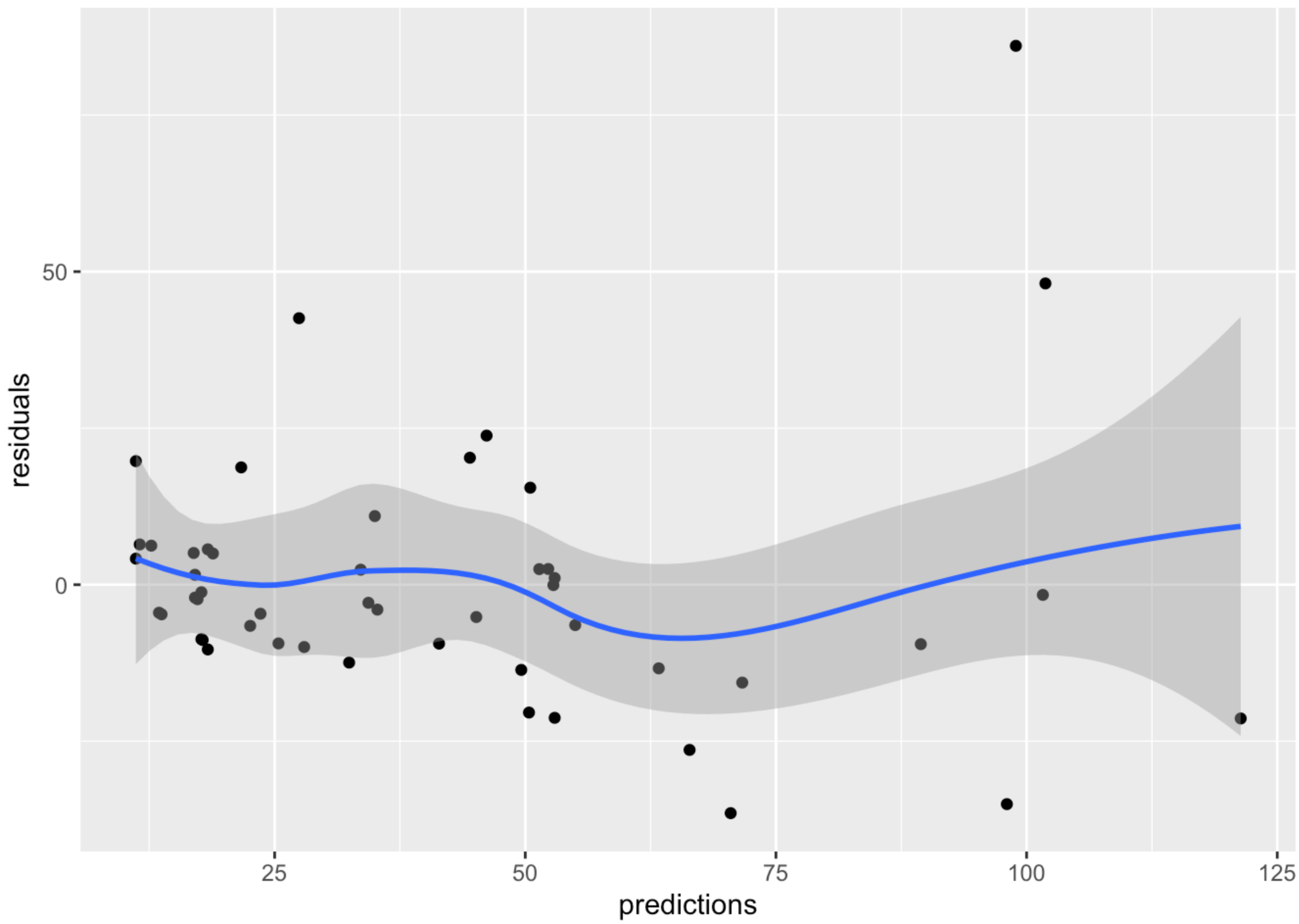

```
  geom_smooth(mapping=aes(Pieces, amazon_price))
```

```
## mapping: x = ~Pieces, y = ~amazon_price
## geom_smooth: na.rm = FALSE, orientation = NA, se = TRUE
## stat_smooth: na.rm = FALSE, orientation = NA, se = TRUE
## position_identity
```

```
predictions <- predict(lm_legos)
residuals <- resid(lm_legos)

ggplot(data.frame(predictions, residuals), aes(x = predictions, y = residuals)) +
  geom_point() +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Question 6

Do you believe the normality and homoskedasticity assumptions are met? Do you think all points are "identically distributed" (eg no pattern regardless of theme or location on the graph)

There is a fair amount of normality, and a fair amount of homoskedasticity, but it does start to get a megaphone shape towards the end with some of the outliers.

Question 7

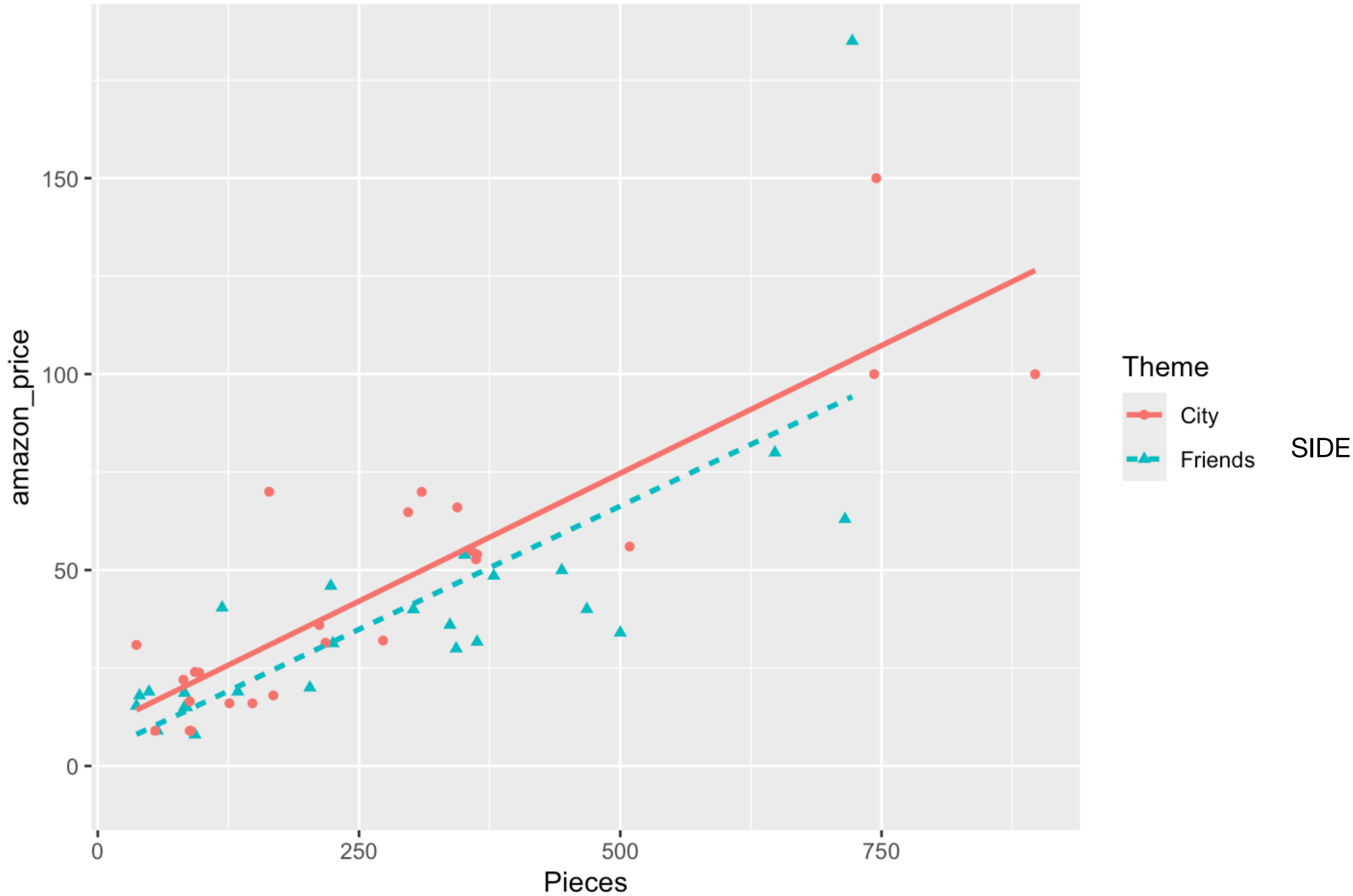
Copy down your answer to question 2. Add a best fit line via geom_smooth. Also, add an aes() within geom_smooth() like you do for geom_point() or ggplot(). Set the parameter called "group" equal to Theme. That is…

```
#geom_smooth(method = "lm",
#             aes(group = Theme))
```

```
ggplot(data = blocks) +
  geom_point(aes(x = Pieces, y = amazon_price, shape=Theme, color=Theme)) +
  geom_smooth(method = "lm", aes(x = Pieces, y = amazon_price, group = Theme, color = Th
eme, linetype=Theme, linetype=4), fill=NA)
```

```
## Warning: Duplicated aesthetics after name standardisation: linetype
## Duplicated aesthetics after name standardisation: linetype
```

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



NOTE: You can actually color and shape your lines as well in the aes() function here using color = Theme and linetype = Theme. Try it to see what happens.

Question 8

Calculate the mean amazon_price for both Themes by using the aggregate() function. See the below hashtaged code for the format of the function

```
aggregate(Pieces  ~ amazon_price,
         data = blocks,
         FUN = mean)
```

```
##    amazon_price Pieces
## 1          7.99  93.00
## 2          8.99  72.25
## 3         14.99  84.00
## 4         15.32  37.00
## 5         15.99 126.00
## 6         16.00 148.00
## 7         16.49  88.00
## 8         17.97  40.00
## 9         17.99 168.00
## 10        18.64  83.00
## 11        18.93  49.00
## 12        18.95 134.00
## 13        19.99 203.00
## 14        21.99  82.00
## 15        23.83  97.00
## 16        23.98  93.00
## 17        29.95 343.00
## 18        30.89  37.00
## 19        31.27 225.00
## 20        31.46 218.00
## 21        31.67 363.00
## 22        31.99 273.00
## 23        34.00 500.00
## 24        35.98 212.00
## 25        35.99 337.00
## 26        39.95 302.00
## 27        40.00 468.00
## 28        40.40 119.00
## 29        45.95 223.00
## 30        48.53 379.00
## 31        49.95 444.00
## 32        52.74 362.00
## 33        53.88 351.00
## 34        53.99 363.00
## 35        54.80 358.00
## 36        56.00 509.00
## 37        62.99 715.00
## 38        64.75 297.00
## 39        65.99 344.00
## 40        69.95 310.00
## 41        69.99 164.00
## 42        79.95 648.00
## 43        99.99 820.00
## 44       149.99 745.00
## 45       184.99 722.00
```

A Step Up

Consider a situation where the estimated slopes for Lego City and Lego Friends were identical, 0.13, the y-intercept for Lego City was 9.44, and the y-intercept for Lego Friends was 3.40. In this case we have the following two estimated models.

$\hat{y_{city}}$ = 9.44 + .13 * Number of Pieces $\hat{y_{friends}}$ = 3.40 + .13 * Number of Pieces

Question 9

Find the difference between the two y-intercepts. What does this number represent?

```
9.44 - 3.4
```

```
## [1] 6.04
```

This number (6.04) represents the (theoretical) starting price for each type of Lego set.

Question 10

Think about how you could use the Theme variable to put the two linear regressions together into a single model (formula). Try to write it out using the estimates given right before question 9.

HINT: You'll need to use an indicator variable to do this

$\hat{y_{city}}$ = 9.44 + .13 * Number of Pieces $\hat{y_{friends}}$ = 3.40 + .13 * Number of Pieces

$(I_C ity)$ = 0 if Friends, 1 if City

$\hat{y_{combined}}$ = 3.40 + 6.04* $\boxed{\text{\{I\_\{City\}}}$ + .13 * Number of Pieces

Question 11

In R we can represent the equation that you should hopefully have found in question 10 by putting into the lm() function 2 different explanatory variables.

Do this by replacing Expalantory_var_1 and 2 in the below code with Pieces and Theme. Also replace the response with amazon_price and the data with whatever you named your data set (blocks most likely). Remove the hashtags as well.

```
combo_model_lm <- lm(amazon_price ~ Pieces + Theme,
                     data = blocks)
```

Question 12

Using the summary() function, write down the estimated regression equation

```
summary(combo_model_lm)
```

```
##
## Call:
## lm(formula = amazon_price ~ Pieces + Theme, data = blocks)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.869 -10.142  -2.064   5.923  89.628
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.02122    5.42889   1.846   0.0712 .
## Pieces        0.12834    0.01314   9.767 6.83e-13 ***
## ThemeFriends -7.32372    5.73584  -1.277   0.2079
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.28 on 47 degrees of freedom
## Multiple R-squared:  0.673,  Adjusted R-squared:  0.6591
## F-statistic: 48.36 on 2 and 47 DF,  p-value: 3.915e-12
```

Question 13

What does the -7.32 next to ThemeFriends mean in the Coefficients table?

That is the coefficient of the linear term for the estimated linear regression model.

Many Steps Up

We are now reading in our data again but this time NOT removing the duplo sets. Note the data is now being saved as "legos". Also, we are no long focusing on the difference between the three themes and are instead focusing on the size of the lego bricks.

```
legos <- read.csv('https://vinnys-classes.github.io/data/legos_data.csv')

head(legos)
```

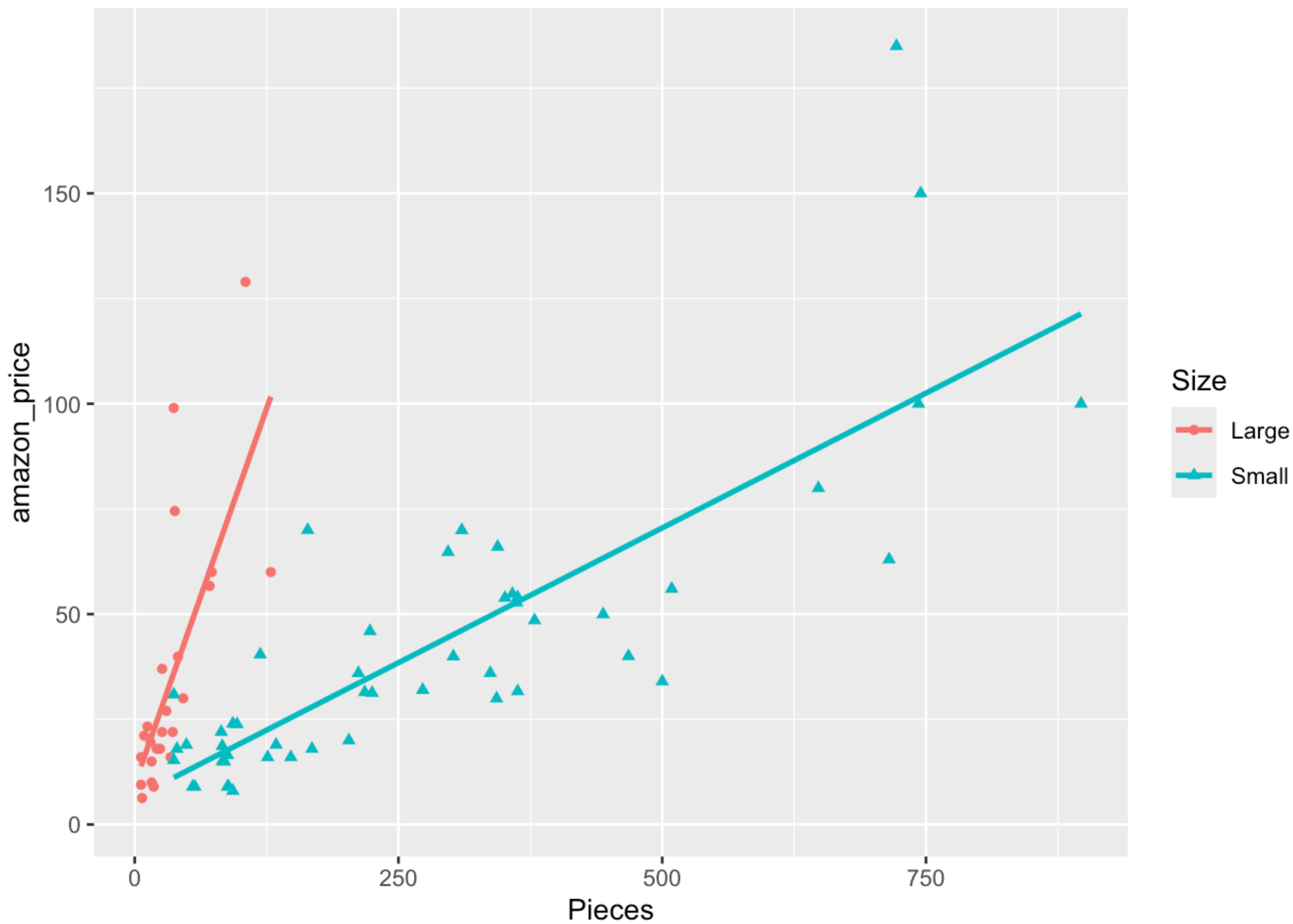
```
##   Item_Number                     Set_Name Theme Pieces Year Pages Minifigures
## 1       10859          My First Ladybird Duplo      6 2018     9          NA
## 2       10860          My First Race Car Duplo      6 2018     9          NA
## 3       10862       My First Celebration Duplo     41 2018     9          NA
## 4       10864 Large Playground Brick Box Duplo     71 2018    32           2
## 5       10867              Farmers' Market Duplo     26 2018     9           3
## 6       10870                 Farm Animals Duplo     16 2018     8          NA
##     Packaging Unique_Pieces  Size amazon_price age
## 1         Box             5 Large        16.00   1
## 2         Box             6 Large         9.45   1
## 3         Box            18 Large        39.89   1
## 4 Plastic box            49 Large        56.69   2
## 5         Box            18 Large        36.99   2
## 6         Box            13 Large         9.99   2
```

Question 14

Plot the amazon_price by Pieces and color by the Size. Include two best-fit-lines like in question 7.

```
ggplot(data = legos) +
  geom_point(aes(x = Pieces, y = amazon_price, shape=Size, color=Size)) +
  geom_smooth(method = "lm", aes(x = Pieces, y = amazon_price, group = Size, color = Siz
e), fill=NA)
```

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



Question 15

Do the slopes of the two lines look similar? If not, how so?

No, the slopes are very different and the slope of the Large line is much steeper than the Small line.

Question 16

Based on your previous answers, would it be enough to add an indicator variable like we did in Question 11? Why or why not?

Yes, because we have two sets of data that we are looking at; large and small. We could use an indicator variable and run a linear regression model to analyze them like in question 11.

Question 17

Use the subset() function to split our data set into two. One will be large bricks and the other will be small bricks. To do this you will need to create two new data sets. I did the first for you, now make the second for Small bricks.

```
data_large <- subset(legos,
                     Size == "Large")
data_small <- subset(legos,
                     Size == "Small")
```

Question 18

Run a linear model using the lm() function for both data sets with amazon_price as the response and Pieces as the explanatory variable. Write out the two estimated linear regression equations (one for large bricks, and one for small bricks).

```
lm(amazon_price ~ Pieces,
                     data = data_large)
```

```
##
## Call:
## lm(formula = amazon_price ~ Pieces, data = data_large)
##
## Coefficients:
## (Intercept)       Pieces
##      9.5546       0.7141
```

```
lm(amazon_price ~ Pieces,
                     data = data_small)
```

```
##
## Call:
## lm(formula = amazon_price ~ Pieces, data = data_small)
##
## Coefficients:
## (Intercept)       Pieces
##      6.4178       0.1281
```

$\hat{y_{large}}$ = 9.55 + .7141 * Number of Pieces $\hat{y_{small}}$ = 6.42 + .1281 * Number of Pieces

Question 19

Using indicator variables, how could you write out a linear model that would allow you to account for two different slopes? Use $\hat{\beta}$'s and not numbers. HINT: You will again need an indicator variable for this.

$(I_L arge)$ = 0 if Small, 1 if Large $\hat{y_{combined}} = \hat{\beta 1} + \hat{\beta 2} * I_{Large}$ + $ * Number of Pieces + $\hat{\beta 4} * I_L arge$ * Number of Pieces

Question 20

Run one more linear model similar to question 11. This time run the code similar to before where between the two explanatory variables there is a * and not a +

Write out the estimated linear equation

```
combo_model_lm2 <- lm(amazon_price  ~ Pieces *
                          Theme,
              data = blocks)
summary(combo_model_lm2)
```

```
##
## Call:
## lm(formula = amazon_price ~ Pieces * Theme, data = blocks)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.316  -9.869  -2.073   5.933  90.741
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          9.437468   6.428054   1.468    0.149
## Pieces               0.130469   0.018032   7.235 4.05e-09 ***
## ThemeFriends        -6.034602   9.398995  -0.642    0.524
## Pieces:ThemeFriends -0.004644   0.026653  -0.174    0.862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.49 on 46 degrees of freedom
## Multiple R-squared:  0.6732, Adjusted R-squared:  0.6519
## F-statistic: 31.58 on 3 and 46 DF,  p-value: 3.073e-11
```

Question 21

Predict the price for a lego set with 55 pieces using your above equation that has small bricks

$(I_L\, arge)$ = 0 if Small, 1 if Large $\hat{y_{combined}}$ = $\hat{\beta}1$ + $\hat{\beta}2$ * Number of Pieces + $ * $I_L\, arge$ + $\hat{\beta}4$ * $I_L\, arge$

$\hat{y_{combined}}$ = 9.44 + (0.1304 * 55) - 6.034 * 0 - (55*0)

```
9.44 + (0.1304 * 55) - 6.034 * 0 - (55*0)
```

```
## [1] 16.612
```

= 16.61

Question 22

Locate the Monster Truck lego set in the data (row 71, you can use the function tail(legos) to see it actually). Using your prediction in question 21 and the price of this lego set calculate a residual.

```
tail(legos)
```

```
##    Item_Number              Set_Name Theme Pieces Year Pages Minifigures
## 70       60249         Street Sweeper  City     89 2020    48           1
## 71       60251          Monster Truck  City     55 2020    32           1
## 72       60252 Construction Bulldozer  City    126 2020    84           2
## 73       60258        Tuning Workshop  City    897 2020   389           7
## 74       60266 Ocean Exploration Ship  City    745 2020   229           8
## 75       60267      Safari Off-Roader  City    168 2020    76           2
##    Packaging Unique_Pieces  Size amazon_price age
## 70       Box            55 Small         8.99   5
## 71       Box            34 Small         8.99   5
## 72       Box            81 Small        15.99   4
## 73       Box           411 Small        99.99   6
## 74       Box           314 Small       149.99   7
## 75       Box            86 Small        17.99   5
```

Residual:

```
8.99 - 16.62
```

```
## [1] -7.63
```

Overpredicted by 7.63 dollars