Question 1.2.2 Choose two *different* words in the dataset with a magnitude (absolute value) of correlation higher than 0.2 and plot a scatter plot with a line of best fit for them. Please do not pick "outer" and "space" or "san" and "francisco". The code to plot the scatter plot and line of best fit is given for you, you just need to calculate the correct values to `r`, `slope` and `intercept`.

Hint 1: It's easier to think of words with a positive correlation, i.e. words that are often mentioned together. Try to think of common phrases or idioms.

Hint 2: Refer to Section 15.2 of the textbook for the formulas. For additional past examples of regression, see Homework 9.

In [ ]:
```
word_x = "hard"
word_y = "work"

# These arrays should make your code cleaner!
arr_x = movies.column(word_x)
arr_y = movies.column(word_y)

r = correlation(arr_x, arr_y)

def lr_slope(x, y):
    r = correlation(x, y)
    return np.std(y) * r / np.std(x)

slope = lr_slope(arr_x, arr_y)
intercept = slope * (0 - np.mean(arr_x)) + np.mean(arr_y)

# DON'T CHANGE THESE LINES OF CODE
movies.scatter(word_x, word_y)
max_x = max(movies.column(word_x))
plots.title(f"Correlation: {r}, magnitude greater than .2: {abs(r) >= 0.2}")
plots.plot([0, max_x * 1.3], [intercept, intercept + slope * (max_x*1.3)], color='gold');
```

Question 1.3.1 Draw a horizontal bar chart with two bars that show the proportion of Comedy movies in each dataset (`train_movies` and `test_movies`). The two bars should be labeled "Training" and "Test". Complete the function `comedy_proportion` first; it should help you create the bar chart.

Hint: Refer to Section 7.1 of the textbook if you need a refresher on bar charts.

In [ ]:
```
# movies
# test_movies
```

In [ ]:
```
def comedy_proportion(table):
    # Return the proportion of movies in a table that have the comedy genre.
    prop = table.where('Genre', 'comedy').num_rows / table.num_rows * 100
    return prop

# The staff solution took multiple lines.  Start by creating a table.
# If you get stuck, think about what sort of table you need for barh to work
test_comedy_prop = comedy_proportion(test_movies)
train_comedy_prop = comedy_proportion(train_movies)
comedy_dist = Table().with_columns(
    'dataset', make_array('Training', 'Test'),
    'proportion of Comedy', make_array(train_comedy_prop, test_comedy_prop)
)
comedy_dist.barh('dataset')
```

Question 3.1.7 In two sentences or less, describe how you selected your features.

I selected words mainly from the top left and bottom right parts of the graph because these words are common in one movie while it's uncommon in another movie.

Question 3.3.3

Do you see a pattern in the types of movies your classifier misclassifies? In two sentences or less, describe any patterns you see in the results or any other interesting findings from the table above. If you need some help, try looking up the movies that your classifier got wrong on Wikipedia.

We noticed that the classifier has better accuracy for thrillers. We also found out that some of the misclassified movies are a mix between comedy and thriller.

Question 4.2

Do you see a pattern in the mistakes your new classifier makes? How good an accuracy were you able to get with your limited classifier? Did you notice an improvement from your first classifier to the second one? Describe in two sentences or less.

Hint: You may not be able to see a pattern.

The new classifier has higher accuracy, probably due to using a different set of features.

Question 4.3

Given the constraint of five words, how did you select those five? Describe in two sentences or less.

We first choose words in the top right corner so that they appear frequently in both genres. We then find 5 words that deviates a lot from the y=x line so that they can help us differentiate between the two genres.