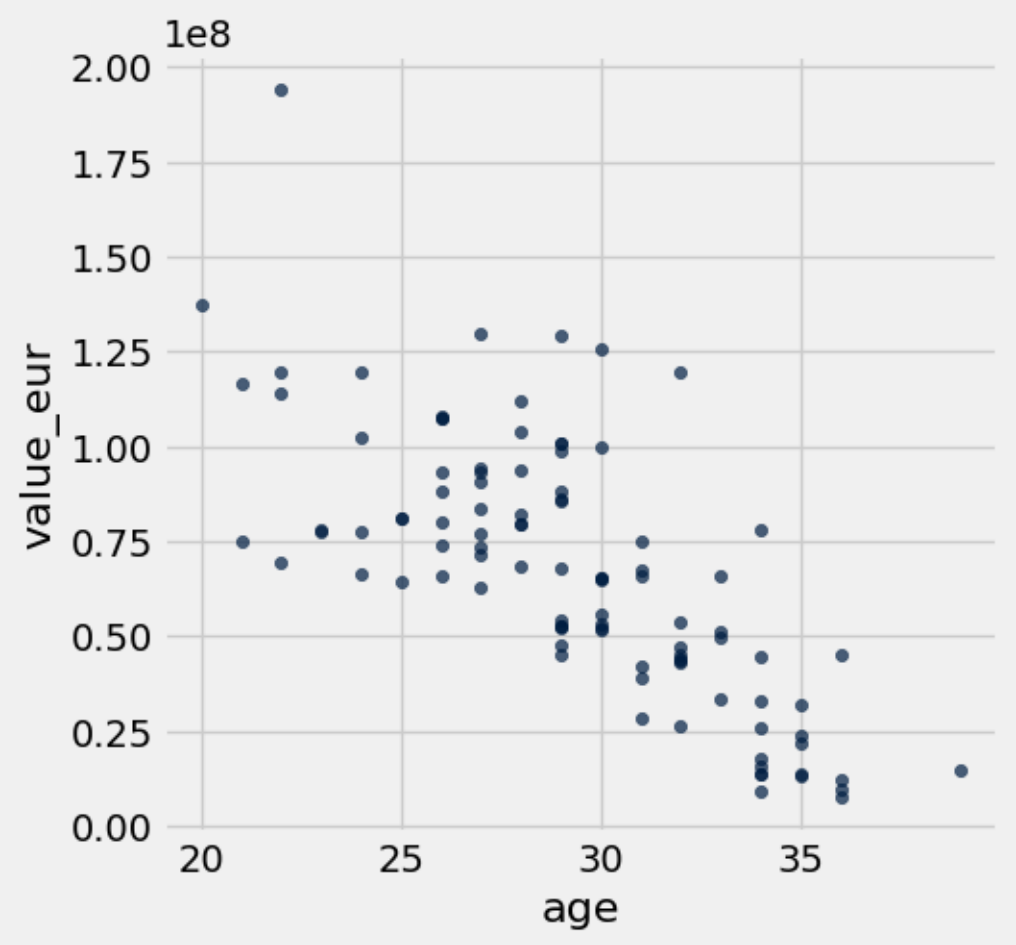

Question 2.1. Before jumping into any statistical techniques, it's important to see what the data looks like, because data visualizations allow us to uncover patterns in our data that would have otherwise been much more difficult to see. **(3 points)**

Create a scatter plot with age on the x-axis ("age"), and the player's value in Euros ("value_eur") on the y-axis.

```
In [22]: fifa.scatter("age", "value_eur")
```

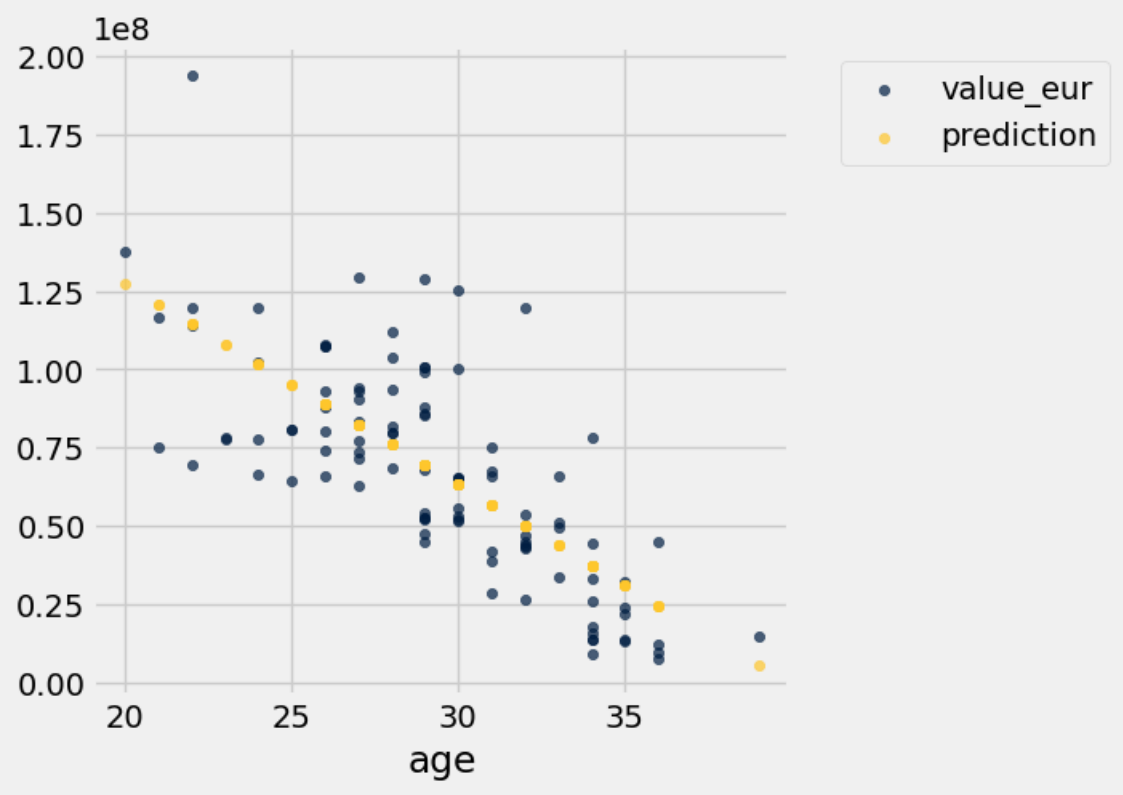


Question 2.3. Create a scatter plot with player age ("age") along the x-axis and both real player value ("value_eur") and predicted player value along the y-axis. The predictions should be created using a fitted **regression line**. The color of the dots for the real player values should be different from the color for the predicted player values. **(8 points)**

Hint 1: Feel free to use functions you have defined previously.

Hint 2: 15.2 and 7.3 has examples of creating such scatter plots.

```
In [25]: predictions = predict(fifa, "age", "value_eur")
         fifa_with_predictions = fifa.with_columns("prediction", predictions)
         fifa_with_predictions.scatter("age", ["value_eur", "prediction"])
```



Question 2.4. Looking at the scatter plot you produced above, is linear regression a good model to use? If so, what features or characteristics make this model reasonable? If not, what features or characteristics make it unreasonable? **(5 points)**

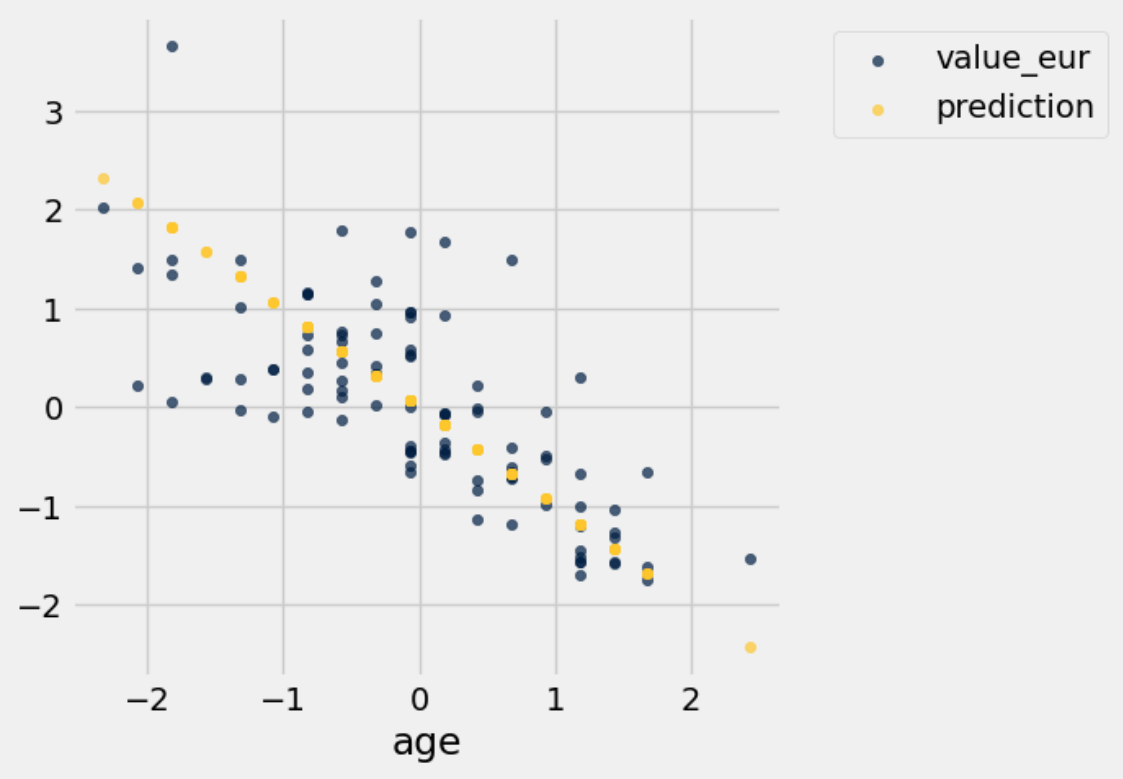
Linear is regression is good to use. Because the value_eur seems to be inversely proportional to age, and the predictions using linear regression is quite close to value_eur

---

Question 2.5. In 2.3, we created a scatter plot **in original units**. Now, create a scatter plot with player age **in standard units** along the x-axis and both real and predicted player value **in standard units** along the y-axis. The color of the dots of the real and predicted values should be different. **(8 points)**

Hint: Feel free to use functions you have defined previously.

```
In [26]: predictions_su = standard_units(predictions)
         fifa_su = Table().with_columns(
             "age", standard_units(fifa.column("age")),
             "value_eur", standard_units(fifa.column("value_eur")),
             "prediction", predictions_su
         )
         fifa_su.scatter("age")
```



Question 2.6. Compare your plots in 2.3 and 2.5. What similarities do they share? What differences do they have? **(5 points)**

Overall, the value_eur and age are still correlated, with inversely proportional correlation. However, compared to the plots in 2.3, in this plot, the value_eur is closer to the prediction in term of distance.

Question 2.8. Use the `rmse` function you defined along with `minimize` to find the least-squares regression parameters predicting player value from player age. Here's an example of using the `minimize` function from the textbook. **(10 points)**

Then set `lsq_slope` and `lsq_intercept` to be the least-squares regression line slope and intercept, respectively.

Finally, create a scatter plot like you did in 2.3 with player age ("age") along the x-axis and both real player value ("value_eur") and predicted player value along the y-axis. **Be sure to use your least-squares regression line to compute the predicted values.** The color of the dots for the real player values should be different from the color for the predicted player values.

Note: Your solution should not make any calls to the slope or intercept functions defined earlier.

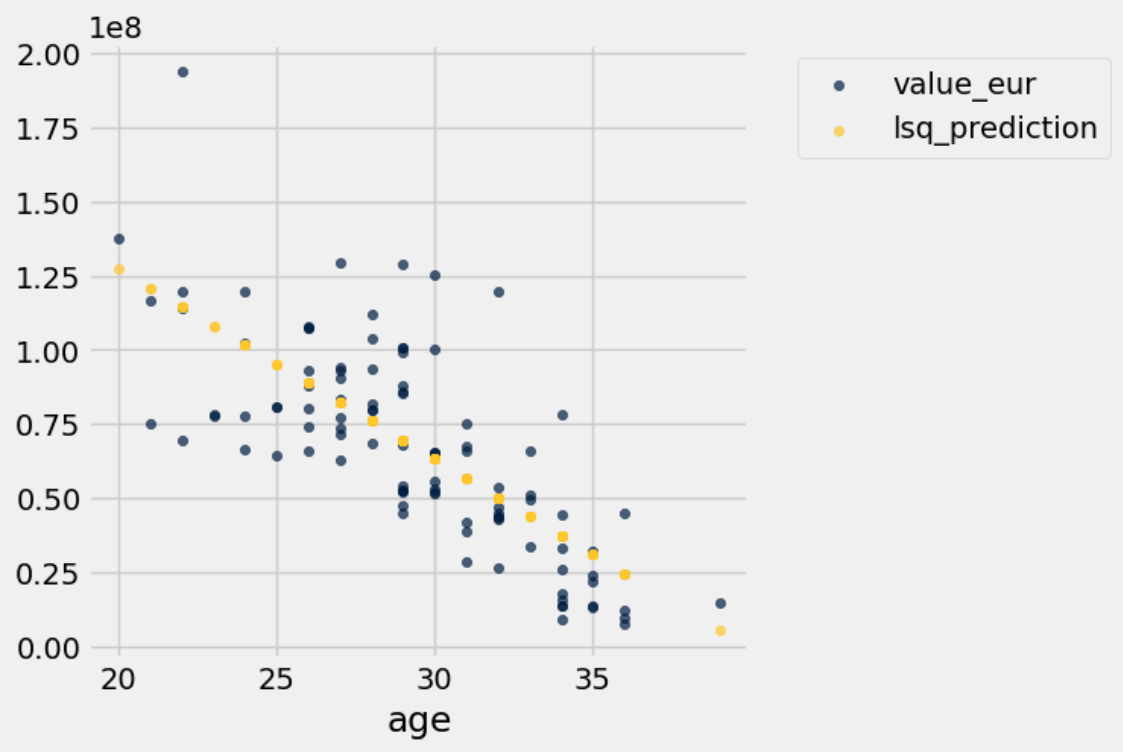
Hint: Your call to `minimize` will return an array of argument values that minimize the return value of the function passed to `minimize`.

```
In [29]: minimized_parameters = minimize(rmse)
         lsq_slope = minimized_parameters.item(0)
         lsq_intercept = minimized_parameters.item(1)

         # This just prints your slope and intercept
         print("Slope: {:g} | Intercept: {:g}".format(lsq_slope, lsq_intercept))

         fifa_with_lsq_predictions = fifa.with_columns("lsq_prediction", lsq_slope * fifa.column("age")
         fifa_with_lsq_predictions.scatter("age", ["value_eur", "lsq_prediction"])
```

```
Slope: -6.41462e+06 | Intercept: 2.55525e+08
```


1e8
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
20
25
30
35
age
value_eur
lsq_prediction

Question 2.9. The resulting line you found in 2.8 should appear very similar to the line you found in 2.3. Why were we able to minimize RMSE to find nearly the same slope and intercept from the previous formulas? **(5 points)**

Hint: Re-reading 15.3 might be helpful here.

By minimizing RMSE, we are able to approximate the intercept and slope of the line that is closer to the ground truth via trial-and-error. And as the regression line we've found using formulas is the closet line to the overall dataset, minimizing RMSE will eventually leads us to the slope and intercept of this line.