Lineal model

Francisco J. Palmero Moya

2023-01-11

1. Introduction

Our main goal is to predict the length of a species of fish as a function of the age and water temperature. Therefore, the model has as inputs the age and water temperature and the output is the length. Such problems where the output is a number are regression problems. The task is to learn a mapping from an input X and an output Y (supervised learning). The approach is that we assume a model defined up to a set of parameters

$$y = g(x|\theta)$$

where $g(\cdot)$ is the model and θ are its parameters.

Here we assume, as inductive bias, that data can be modeled as a lineal model

$$g(x|\theta) = \theta_0 + \theta_1 x^1 + \theta_2 x^2$$

where

x^1; the age of the fish

x^2; the water temperature in degrees Celsius

That is, we would like to write the output (length), called the dependent variable, as a function of the input (age, temperature), called the independent variable. We should add some possible errors and assume that the numeric output is the sum of a deterministic function of the input and random noise.

$$r = f(x) + \epsilon$$

The model learn the optimal parameters that somehow minimizes that error. The most frequently used error function is called least squares estimates

$$E(\theta|x) = \frac{1}{N}\sum_{t=1}^{N}\left[r^t - g(x^t|\theta)\right]^2$$

where N is the number of training instances. During the document we will see different ways of error measuring, i.e., the different ways to choose a model.

The model implementation is in R language, where we will use three libraries: **dplyr**, **ggplot2** and **moderndive**.

The following code imports the desired libraries

```
library(dplyr)
library(ggplot2)
library(moderndive)
```

2. Exploratory data analysis

Exploratory data analysis (EDA) is an approach to analyzing and understanding data sets through summarizing their main characteristics, often through visual methods. Our task hear will consist in identifying patterns, outliers, missing data and any other interesting insights.

We start reading the file containing the data and assigning it to data data frame

```
data <- read.csv('data.txt', sep = ' ') %>% select(-c('Index'))
```

The column named as 'Index' is deleted since the data frame provides an index variable by default.

The **glimpse()** function allow as to view the structure and content of our data frame. It provides a concise summary of the data frame's structure and variables, including the number of rows and columns, variable types, and a sample of the data.

```
data %>% glimpse()
```

```
## Rows: 44
## Columns: 3
## $ Age         <int> 14, 28, 41, 55, 69, 83, 97, 111, 125, 139, 153, 14, 28, 41~
## $ Temperature <int> 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 27, 27, 27, 27~
## $ Length      <int> 620, 1315, 2120, 2600, 3110, 3535, 3935, 4465, 4530, 4570,~
```

We have a training data of 44 instances and 3 columns. It is important to note that the fish are kept in tanks at 25, 27, 29 and 31 degrees Celsius. That is, Temperature variable works as a kind of categorical variable in our model. We can easily check it

```
unique(data$Temperature)
```

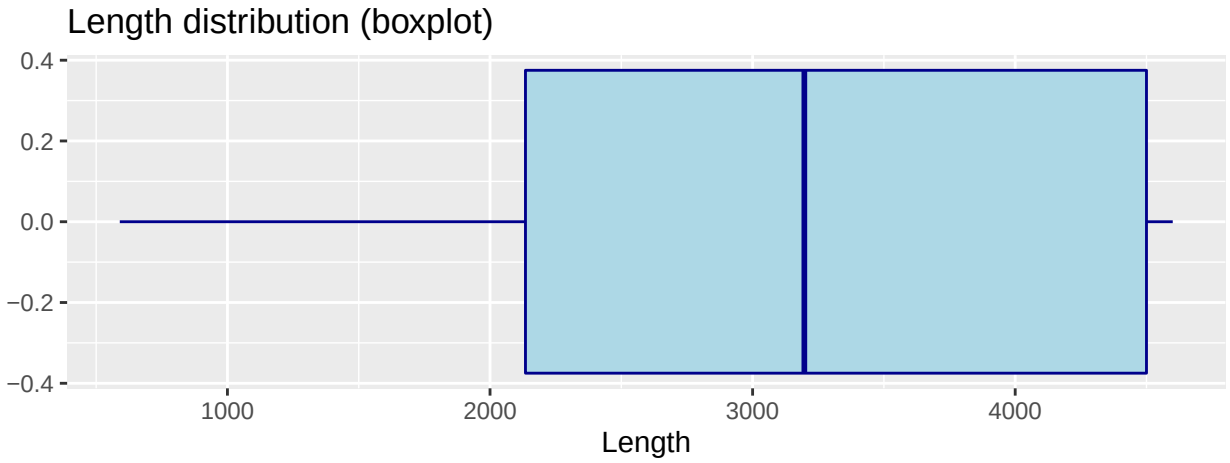
```
## [1] 25 27 29 31
```

Notice that Temperature can only be thought as a categorical variable in visualizations or interpretation since it is include in our model as a integer variable.

2.1 Length distribution

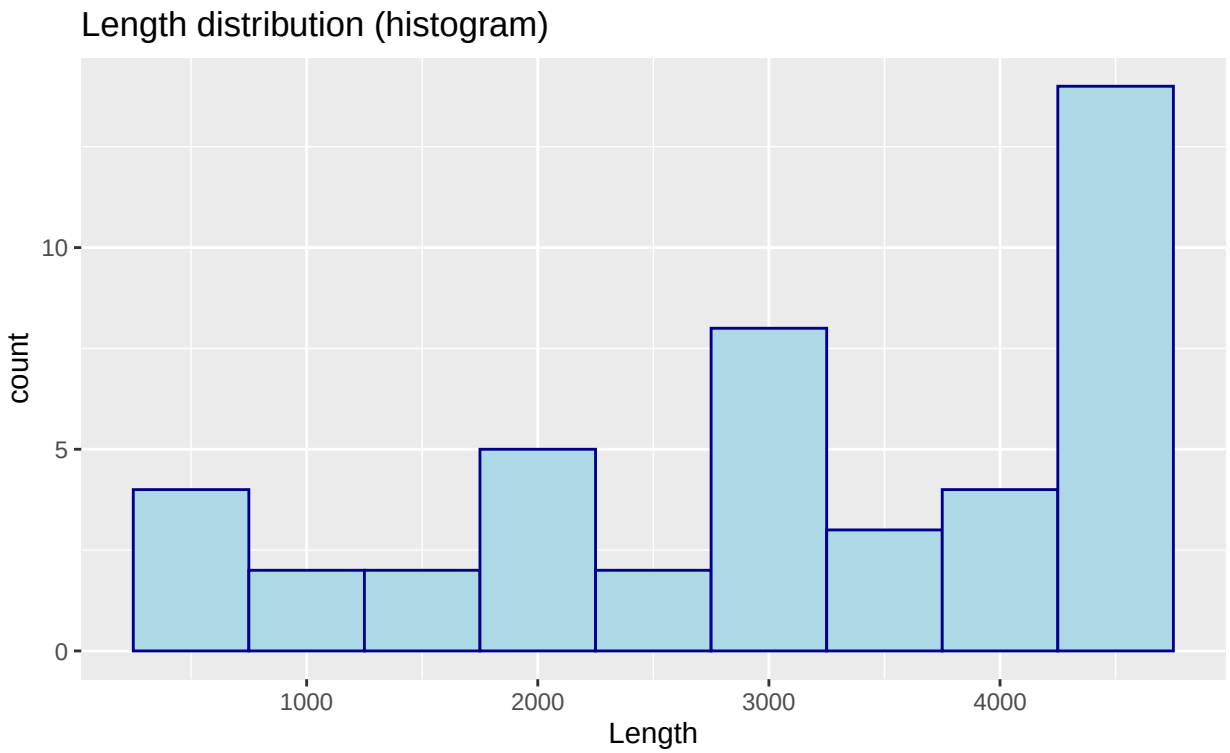
Let's check how is distributed our dependent variable using a boxplot

```
ggplot(data, aes(x = Length)) +
  geom_boxplot(color = 'darkblue', fill = 'lightblue') +
  labs(title = 'Length distribution (boxplot)')
```



or histogram

```
ggplot(data, aes(x = Length)) +
  geom_histogram(binwidth = 500, color = 'darkblue', fill = 'lightblue') +
  labs(title = 'Length distribution (histogram)')
```



On the other hand, the statistics can be shown as numerical values

```
data %>%
  summarize(meanLength = mean(Length),
            medianLength = median(Length),
            maxLength = max(Length),
            minLength = min(Length),
            Q1Length = quantile(Length, probs = 0.25),
            Q3Length = quantile(Length, probs = 0.75)
            )
```

```
##   meanLength medianLength maxLength minLength Q1Length Q3Length
## 1   3107.432         3197      4600       590     2135     4500
```

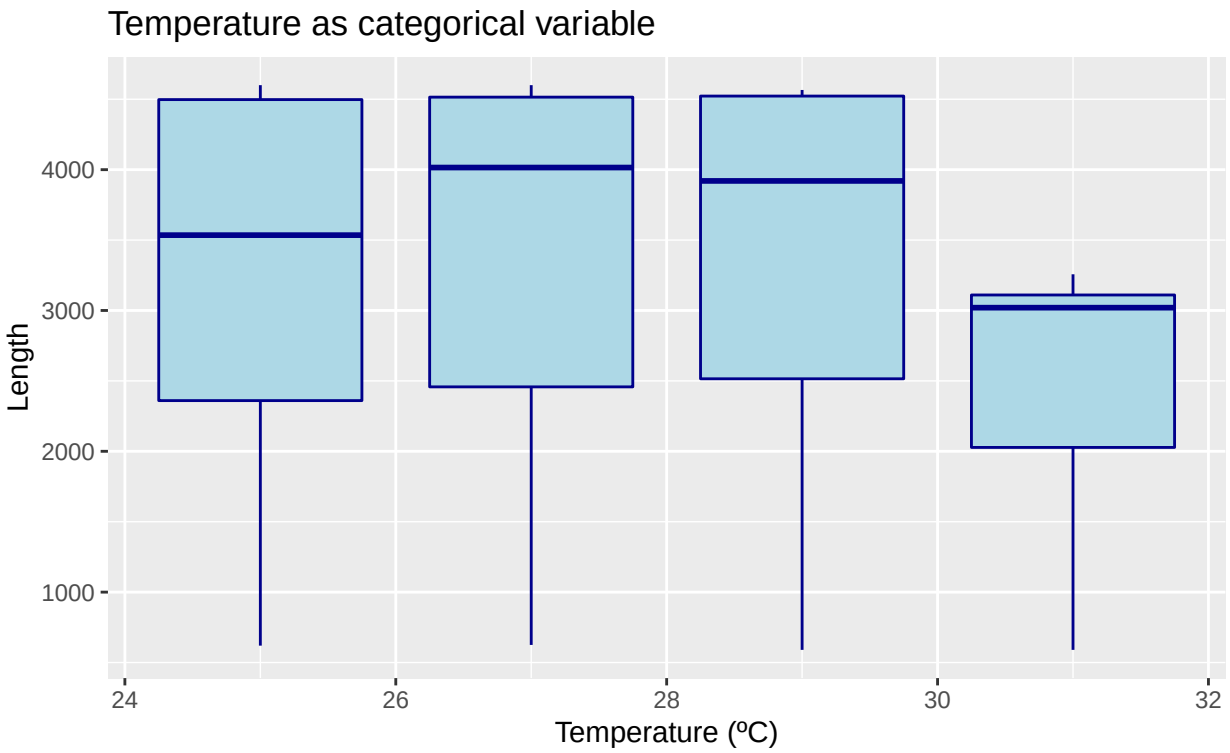
Most fishes have length between 2135 and 4500 (we do not provide units since the length variable in data is adimensional). The expected value for a random fish is 3107.432 and the length always lies between 590 and 4600 in our data. The distribution is weakly skew-left, that is, the tail on the right side of the distribution is longer than the tail on the left side. Therefore, the expected length value for a fish choose at random is close to the most often fish length but a little less.

We are interested in the relation between the variables and how the explanatory variables, temperature and age,are correlated to our dependent variable length. Thus, we will dive into that relation giving some insights.

2.2 Temperature variable

The water temperature can be used to group by its values and obtain the following plot

```
ggplot(data, aes(x = Temperature, y = Length, group = Temperature)) +
  geom_boxplot(color = 'darkblue', fill = 'lightblue') +
  labs(title = 'Temperature as categorical variable', x = 'Temperature (ºC)')
```



The plot shows that tanks having midst temperatures could be correlated with having greater length. Nevertheless, that assumption is outside our model since we did not take into account the age in our guess.

Grouping by Temperature we can obtain the same information as numerical values

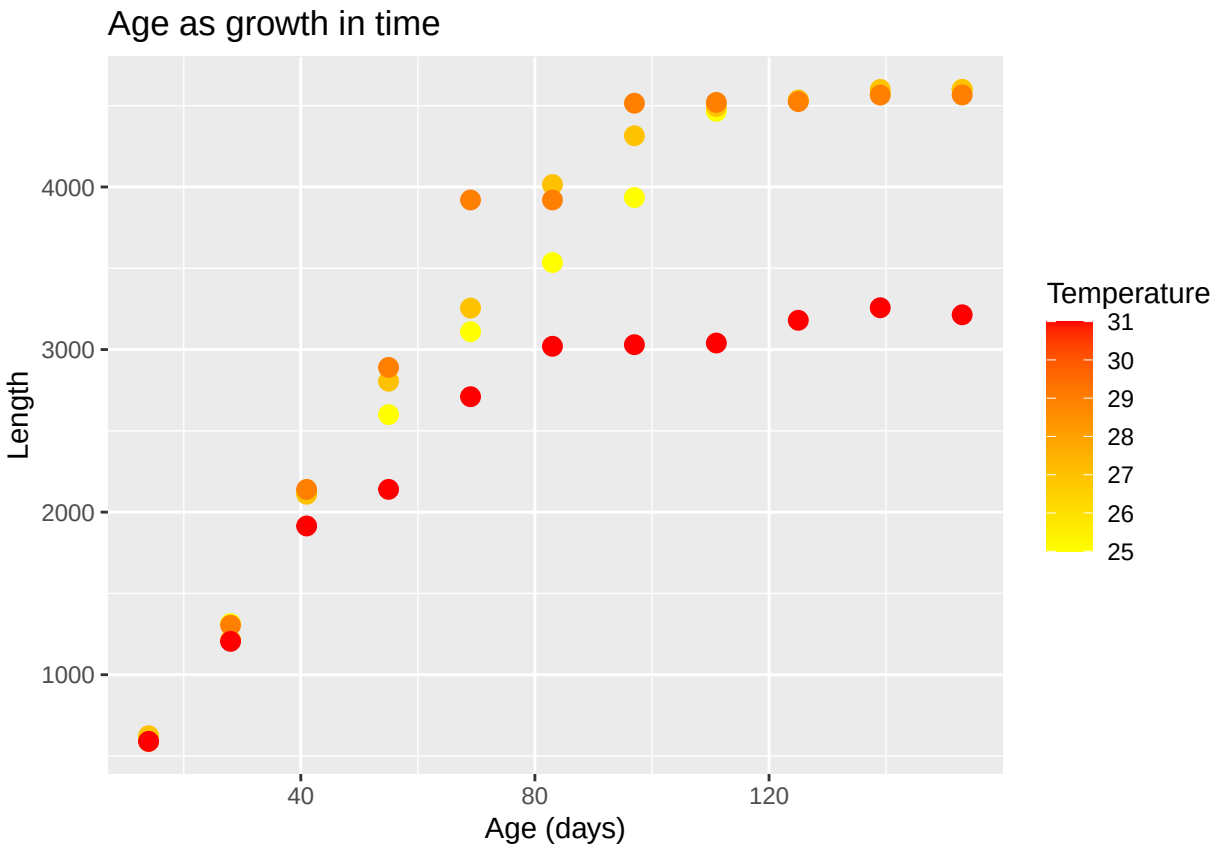
```
data %>%
  group_by(Temperature) %>%
  summarize(meanLength = mean(Length),
            medianLength = median(Length),
            maxLength = max(Length),
            minLength = min(Length),
            Q1Length = quantile(Length, probs = 0.25),
            Q3Length = quantile(Length, probs = 0.75)
            )
```

```
## # A tibble: 4 x 7
##   Temperature meanLength medianLength maxLength minLength Q1Length Q3Length
##         <int>      <dbl>        <int>     <int>     <int>    <dbl>    <dbl>
## 1          25      3218.         3535      4600       620    2360     4498.
## 2          27      3325.         4015      4600       625    2458.    4515
## 3          29      3405.         3920      4566       590    2515     4522.
## 4          31      2482.         3020      3257       590    2028.    3110
```

2.3 Age variable

One way to visualize the fishes growth in time is by means of a scatter plot.

```
ggplot(data, aes(x = Age, y = Length, color = Temperature)) +
  geom_point(size = 3) +
  scale_color_gradient(low="yellow", high="red") +
  labs(title = 'Age as growth in time', x = 'Age (days)')
```



The temperature is given just as extra information, but we are not interested in that variable right now. A quick view is enough to check how age and length are positive correlated and how temperature affects in the pattern.

It is possible to expand the limits of our plot axes and see how is the length when the fish is newborn, nevertheless it does not make sense in our EDA since the younger fish in the data is 14

```
data %>% summarize(minAge = min(Age))
```

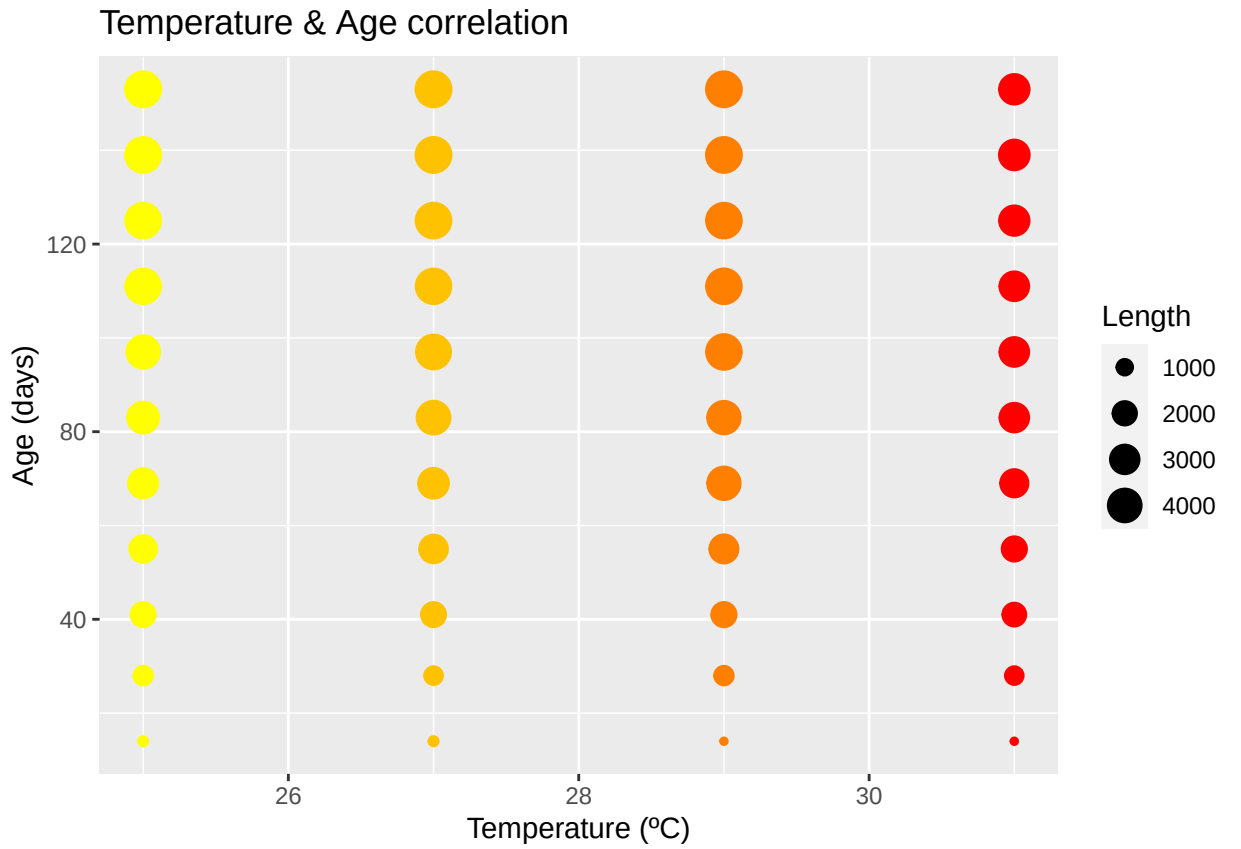
```
##   minAge
## 1     14
```

The plot will become important once we run the model and obtain results from it. So far it is enough to visualize it and extract simple insights to prepare our model.

2.4 Temperature and age correlation

Finally, given a concrete temperature we can see how the length increase in time (days)

```
ggplot(data, aes(x = Temperature, y = Age, size = Length, color = Temperature)) +
  geom_point() +
  scale_color_gradient(low="yellow", high="red") +
  labs(title='Temperature & Age correlation', x='Temperature (ºC)', y='Age (days)') +
  guides(color = FALSE)
```



The result is really important for our model, as we will see in the results and conclusions sections.

3. Results

A linear model is a mathematical model that represents a relationship between a dependent variable and one or more independent variables using a linear equation. Our lineal model is easy to understand and implement

```
model <- lm(Length ~ Age + Temperature, data = data)
```

Running the following code we can check the results

```
get_regression_table(model)
```

```
## # A tibble: 3 x 7
##   term        estimate std_error statistic p_value lower_ci upper_ci
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept     3904.    1149.        3.40   0.002   1584.    6225.
## 2 Age             26.2      2.06     12.8    0         22.1     30.4
## 3 Temperature   -106.      40.5      -2.63   0.012   -188.     -24.7
```

The interpretation of the table is as follows:

- $\theta_0 = 3904.266$ <- intercept: Predicted length value for Age = 0 and Temperature = 0. It does not have a physical meaning at all, since Age = 0 is not defined.
- $\theta_1 = 26.241$ <- slope in Age: given a Temperature value, increment in Length due to increment in Age.
- $\theta_2 = -106.414$ <- slope in Temperature: given an Age value, decrease of Length due to increment in Temperature.

Therefore, it seems like Age & Length are positive correlated and Temperature & Length are negatively correlated. Let's check it

```
data %>% summarize(corAgeLength = cor(Age, Length), corTempLength = cor(Temperature, Length))
```

```
##   corAgeLength corTempLength
## 1    0.8791157     -0.181118
```

Notice than correlation value is a statistic and it is outside our model, i.e., slope concept is different from correlation value. Nevertheless, since we are accounting Age and Temperature as explanatory variables, they should be correlated with Length.

The error can be measured by the folowing parameters

```
get_regression_summaries(model)
```

```
## # 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     0.806         0.796 335452.  579.  600.      85.0       0     2    44
```

where

- $R^2 = 0.806$ <- r squared: shows how correlated the variables are. It is a good value since it is close to $R^2 = 1$.
- mse <- mean squared error: shows how close are the predicted values to the actual values. The mean error is shown in the output.
- rmse <- root mean squared error: a different measured of the error given the actual values. The main difference comes from the fact rmse is normalized.

Therefore, the final result is

$$\hat{y} = 3904.266 + 26.241x^1 - 106.414x^2$$

dimensions are based on variable dimensions. Notice that each parameter comes with an associated error given by the model as shown before. A good visualization of our plane need a 3D plot, we will see in our conclusions section another interesting visuals.

4. Conclusions

The Age variable has a stronger correlation than Temperatures has. In fact, Temperature and Length are close to do not be correlated. That suggests we could drop it out of our model and try to find better results with only Age. Instead, we treat temperature as categorical variable

```
datacat <- read.csv('datacat.csv', sep = ',') %>% select(-c('Index'))

new_model <- lm(Length ~ Age + Temperature, data = datacat)

get_regression_table(new_model)
```

```
## # A tibble: 5 x 7
##   term               estimate std_error statistic p_value lower_ci upper_ci
##   <chr>                 <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
## 1 intercept           1035.     223.        4.63    0        583.    1487.
## 2 Age                   26.2      1.84     14.2     0         22.5     30.0
## 3 Temperature: 27ºC    106.     230.        0.463   0.646   -358.     571.
## 4 Temperature: 29ºC    187.     230.        0.814   0.421   -278.     652.
## 5 Temperature: 31ºC   -736.     230.       -3.20    0.003  -1201.    -272.
```

The interpretation is different now. The temperature estimates are now offsets and the result is a line in a 2D plot where the different regression lines are parallel and the height depends on the offset, i.e., the temperature. Notice that it is analogous to substitute θ_2 for its value in the results.

```
ggplot(datacat, aes(x = Age, y = Length, color = Temperature)) +
  geom_point(size = 2) +
  geom_function(
    fun = function(x) 3904.266 + 26.241*x - 106.414*25,
    color = "#F8766D",
    size = 1
    ) +
  geom_function(fun = function(x) 3904.266 + 26.241*x - 106.414*27,
    color = "#00BA38",
    size = 1
    ) +
  geom_function(fun = function(x) 3904.266 + 26.241*x - 106.414*29,
    color = "#619CFF",
    size = 1
    ) +
  geom_function(fun = function(x) 3904.266 + 26.241*x - 106.414*31,
    color = "#C77CFF",
    size = 1
    )
```