Exercise 12

Zoltan Kekecs

15 november 2018

Exercise 12 - Multiple regression

This exercise will show you how multiple predictors can be used in the same regression model to achieve better prediction efficiency.

The latest version of this document and the code the document refers to can be found in the GitHub repository of the class at: https://github.com/kekecsz/PSYP13_Data_analysis_class-2018

Loading packages

You will need to load the following packages for this exercise:

```
library(psych)  # for describe
library(lm.beta)  # for lm.beta
library(car)  # for scatter3d
library(ggplot2)  # for ggplot
library(rgl)  # for scatter3d
```

Data management and descriptive statistics

Load data about housing prices in King County, USA

In this exercise we will predict the price of apartments and houses.

We use a dataset from Kaggle containing data about housing prices and variables that may be used to predict housing prices. The data is about accomodations in King County, USA (Seattle and sorrounding area).

We only use a portion of the full dataset now containing information about N = 200 accomodations.

You can load the data with the following code

```
# data from
# github/kekecsz/PSYP13_Data_analysis_class-2018/master/data_house_small_sub.csv.
data_house = read.csv("https://bit.ly/2DpwKOr")
```

Check the dataset

You should always check the dataset for coding errors or data that does not make sense.

View data in the data viewer tool

```
View(data_house)
```

Display simple descriptive statistics and plots.

We are going to predict price of the apartment using the variables sqft_living (the square footage of the living area), and grade (overall grade given to the housing unit, based on King County grading system), so lets focus on these variables.

Later we are also going to use a categorical variable, has_basement (whether the apartment has a basement or not) as well.

```r
describe(data_house)
hist(data_house$price, breaks = 30)
hist(data_house$sqft_living, breaks = 30)
hist(data_house$grade, breaks = 30)

# scatterplot
plot(price ~ sqft_living, data = data_house)
plot(price ~ grade, data = data_house)

table(data_house$has_basement)
plot(data_house$price ~ data_house$has_basement)
```

Multiple regression

Fitting the regression model

We fit a regression model with multiple predictors: sqft_living and grade. In the formula, the predictors are separated by a + sign.

```r
mod_house1 = lm(price ~ sqft_living + grade, data = data_house)
```

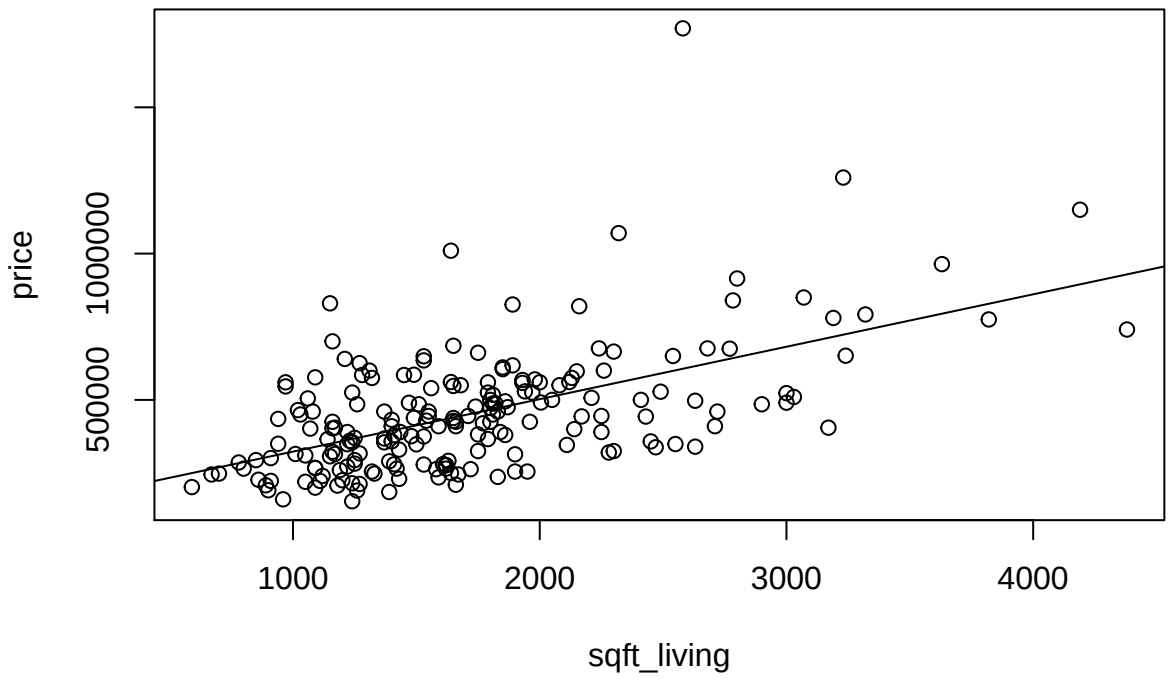
The regression equation is displayed just like in the case of simple regression

```r
mod_house1
```

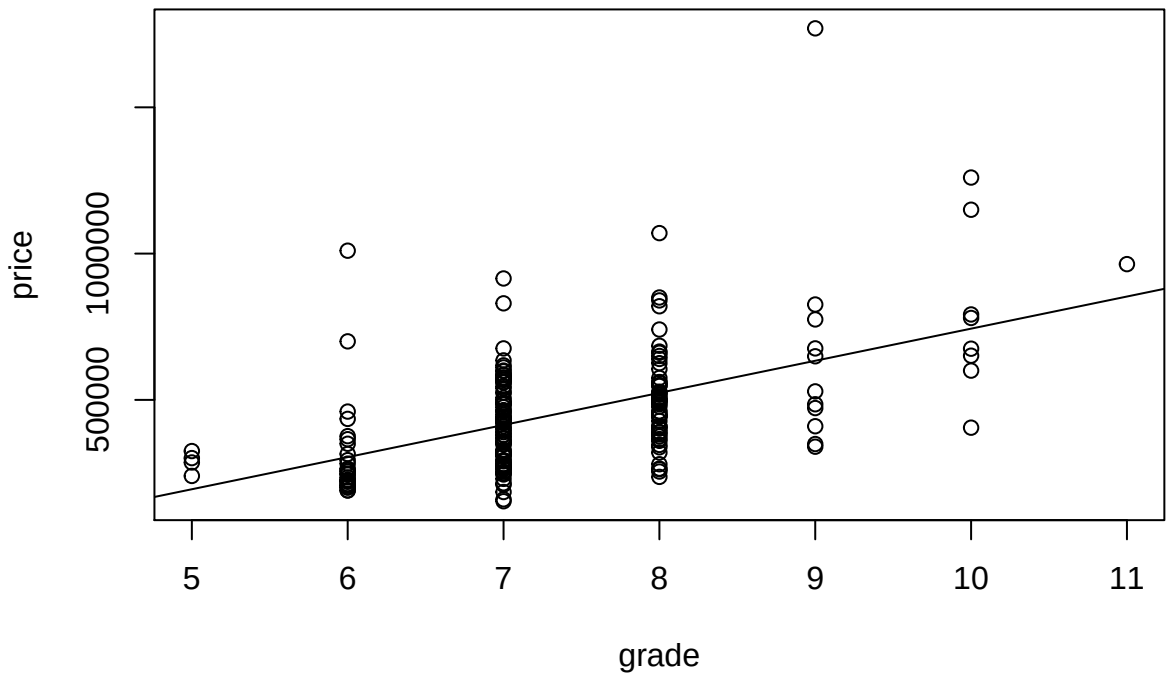
```
##
## Call:
## lm(formula = price ~ sqft_living + grade, data = data_house)
##
## Coefficients:
## (Intercept)  sqft_living        grade
##   -174389.9        119.2      57352.8
```

It is not trivial to visualize the regression equation in multiple regression. You can plot every simple regression separately, but that is not an accurate depiction of the prediction using the model.

```r
plot(price ~ sqft_living, data = data_house)
abline(lm(price ~ sqft_living, data = data_house))
```




```
plot(price ~ grade, data = data_house)
abline(lm(price ~ grade, data = data_house))
```



Alternatively, you can use a 3 dimensional plot to visualize the regression plane.

```
# plot the regression plane (3D scatterplot with regression
# plane) scatter3d(price ~ sqft_living + grade, data =
# data_house)
```

Prediction

Again, we can ask for predictions for specific values of predictors, but we need to specify all predictor values (in this case, both sqft_living and grade of the apartment) to get a prediction.

Remember that you need to provide the predictors in a dataframe with the predictors having the same variable name as in the model formula.

```
sqft_living = c(600, 600, 1100, 1100)
grade = c(6, 9, 6, 9)
newdata_to_predict = as.data.frame(cbind(sqft_living, grade))
predicted_price = predict(mod_house1, newdata = newdata_to_predict)

cbind(newdata_to_predict, predicted_price)
```

```
##   sqft_living grade predicted_price
## 1         600     6        241230.8
## 2         600     9        413289.1
## 3        1100     6        300817.4
## 4        1100     9        472875.8
```

What to report in a publication

In a publication (and in the home assignment) you will need to report the following information:

First of all, you will have to specify the regression model you built. For example:

"In a linar regression model we predicted housing price (in USD) with square footage of living area (in ft) and King County housing grade as predictors."

Next you will have to indicate the effectiveness of the model. You can do this by after a text summary of the results, giving information about the F-test of the whole model, specifically, the F value, the degrees of freedom (note that there are two degrees of freedom for the F test), and the p-value. You can find all this information in the model summary. Also provide information about the model fit using the adjusted R squared from the model summary and the AIC values provided by the AIC() function.

Don't forget to use APA guidelines when determing how to report these statistics and how many decimal places to report (2 decimals for every number except for p values, which should be reported up to 3 decimals).

```
sm = summary(mod_house1)
sm
```

```
##
## Call:
## lm(formula = price ~ sqft_living + grade, data = data_house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -371917 -100605  -23119   66886 1120748
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -174389.86   95255.17  -1.831 0.068646 .
## sqft_living     119.17      24.76   4.813 2.96e-06 ***
## grade         57352.79   16052.79   3.573 0.000444 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 170100 on 197 degrees of freedom
## Multiple R-squared:  0.358,  Adjusted R-squared:  0.3515
## F-statistic: 54.94 on 2 and 197 DF,  p-value: < 2.2e-16
```

```
AIC(mod_house1)
```

```
## [1] 5390.142
```

"The multiple regression model was significantly better than the null model, explaining 35.15% of the variance in housing price (F (2, 197) = 54.94, p < .001, Adj. R^2 = 0.35, AIC = 5390.14)."

Furthermore, you will have to provide information about the regression equation and the predictors' added value to the model. You can do this by creating a table with the following information:

Regression coefficients with confidence intervals, and standardized beta values for each predictor, together with the p-values of the t-test.

The regression coefficients and p-values can be found in the model summary, and the confidence interavlas and std. betas can be computed by applying the confint() and lm.beta() functions on the model object. (the lm.beta package is needed for the lm.beta() function)

```
confint(mod_house1)
lm.beta(mod_house1)
```

The final table should look something like this:

```
##                        b   95%CI lb 95%CI ub Std.Beta p-value
## (Intercept) -174389.86 -362240.59 13460.86        0    .069
## sqft_living     119.17      70.34   168.01     0.37   <.001
## grade         57352.79   25695.42 89010.16     0.28   <.001
```

You should refer to your course book (Chapter 15 - Navarro D. (2015). Learning statistics with R: A tutorial for psychology students and other beginners (5th ed.). http://www.compcogscisydney.com/learning-statistics-with-r.html) for the interpretation of the data reported above.

Build better models

Experiment with different models based on your theories about what could influence housing prices.

Try to increase the adjusted R^2 above 52%. If you want to get access to the whole dataset or get ideas on which model works best, go to Kaggle, check out the top kernels, and download the data. https://www.kaggle.com/harlfoxem/housesalesprediction/activity

Categorical predictor

Categorical variables can be included in models just like continuous variables. Here, we include the variable has_basement as a predictor, which is a categorical variable that has two levels: 'has basement' and 'no basement'. In this case, the intercept can be interpreted as the predicted value for all continuous predictor values as 0, and the has_basement variable at its default level: 'has basement'. The regression coefficient for has_basement indicates how much price is predicted to change if the apartment has no basement compared to if it has basement.

```
mod_cat = lm(price ~ sqft_living + grade + has_basement, data = data_house)

summary(mod_cat)
```

```
##
## Call:
## lm(formula = price ~ sqft_living + grade + has_basement, data = data_house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -419366  -90722  -24318   64652 1147241
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -138667.75   94177.76  -1.472  0.14252
## sqft_living               107.86      24.58   4.388 1.86e-05 ***
## grade                   62112.50   15822.84   3.925  0.00012 ***
## has_basementno basement  -75859.09   25485.56  -2.977  0.00328 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 166800 on 196 degrees of freedom
## Multiple R-squared:  0.3858, Adjusted R-squared:  0.3764
## F-statistic: 41.04 on 3 and 196 DF,  p-value: < 2.2e-16
```

The default level (reference level) of categorical variables is the level earliest in the alphabet. For this reason, the reference level of the variable has_basement is "has basement". For more intuitive interpretation, it would make sense to change the reference level to "no basement", so that the model coefficient for this variable

would be positive, and it would indicate how much price increase would a basement mean for the apartment sales.

This can be done with the relevel() function. We have to re-run the model for this change to take effect in the model object.

```
data_house$has_basement = relevel(data_house$has_basement, ref = "no basement")

mod_cat = lm(price ~ sqft_living + grade + has_basement, data = data_house)
summary(mod_cat)
```

```
##
## Call:
## lm(formula = price ~ sqft_living + grade + has_basement, data = data_house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -419366  -90722  -24318   64652 1147241
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -214526.84   94378.23  -2.273  0.02411 *
## sqft_living                 107.86      24.58   4.388 1.86e-05 ***
## grade                     62112.50   15822.84   3.925  0.00012 ***
## has_basementhas basement  75859.09   25485.56   2.977  0.00328 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 166800 on 196 degrees of freedom
## Multiple R-squared:  0.3858, Adjusted R-squared:  0.3764
## F-statistic: 41.04 on 3 and 196 DF,  p-value: < 2.2e-16
```

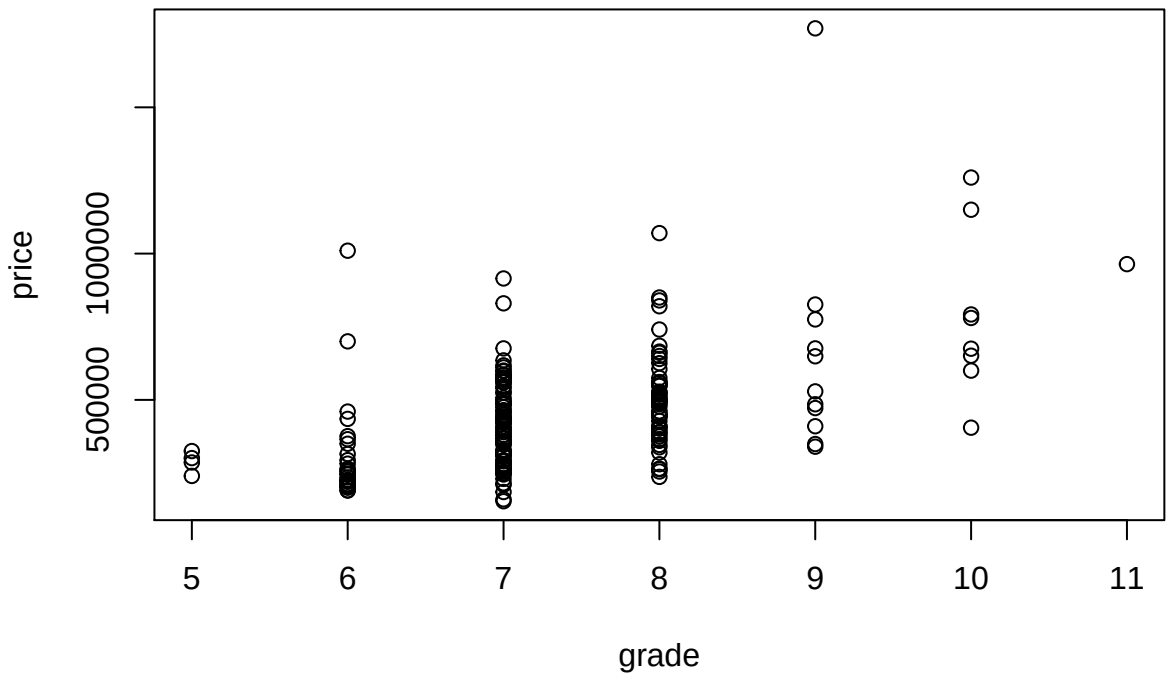
Now it is apparent from the model summary that if an apartment has a basement, this means a 75859 USD increase in price compared to apartments which do not have a basement (the reference level).

Higher order terms

If you suspect that there is non-linear relationship between the outcome and some predictor, you can try to include a second or third order term.

For example, here we can see that the relationship of price and grade is not entirely linear.

```
plot(price ~ grade, data = data_house)
```



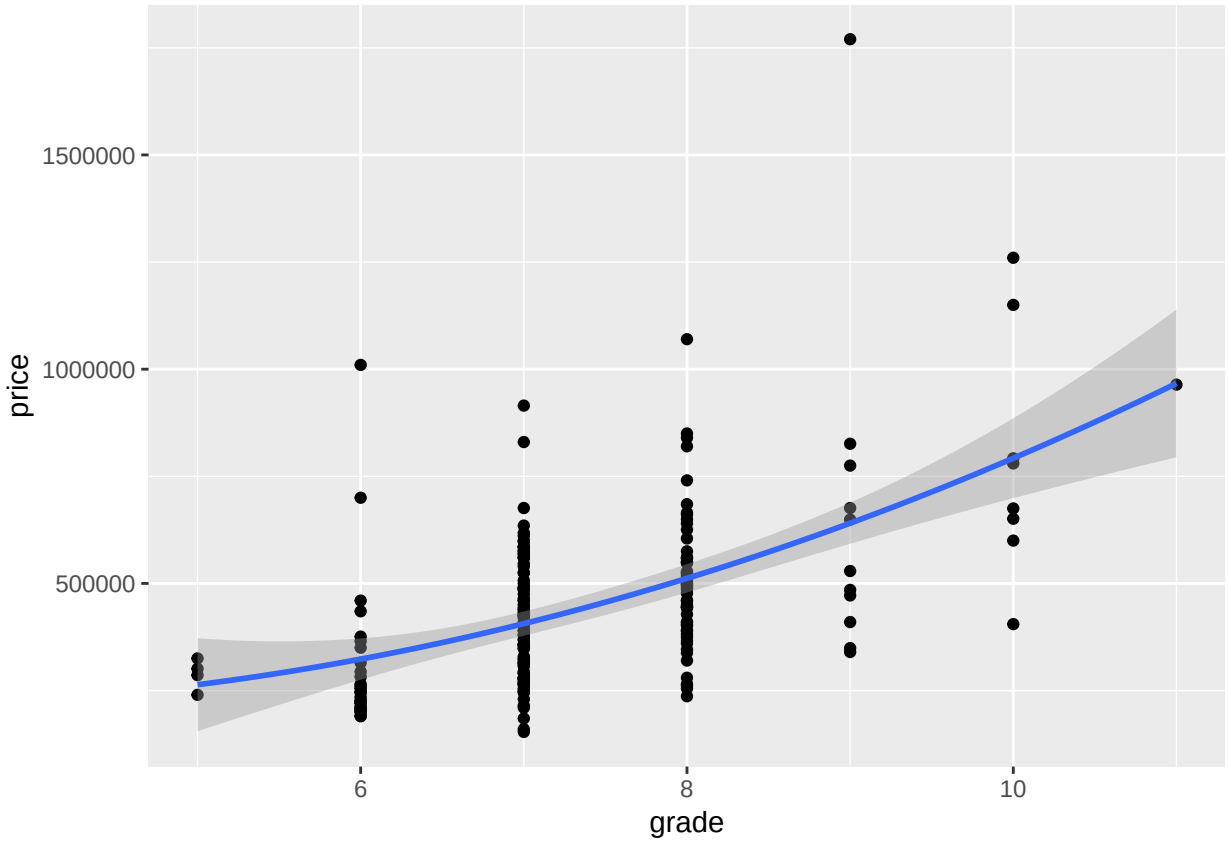
So we build a model including the second order term of grade, to account for a quadratic relationsip.

Unless you know what you are doing, always add the first order term in the model as well, like here:

```
mod_house_quad <- lm(price ~ grade + I(grade^2), data = data_house)
summary(mod_house_quad)
```

```
##
## Call:
## lm(formula = price ~ grade + I(grade^2), data = data_house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -387267 -114876  -22817   71243 1129351
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   308733     440605   0.701    0.484
## grade         -66385     115136  -0.577    0.565
## I(grade^2)     11474       7455   1.539    0.125
##
## Residual standard error: 178700 on 197 degrees of freedom
## Multiple R-squared:  0.2911, Adjusted R-squared:  0.2839
## F-statistic: 40.44 on 2 and 197 DF,  p-value: 1.923e-15
```

```
ggplot(data_house, aes(x = grade, y = price)) + geom_point() +
    geom_smooth(method = "lm", formula = y ~ x + I(x^2))
```



Interactions

A relationship of different predictors can also be modelled, if you suspect that the association of a predictor and the outcome might depend on the value of another predictor.

For example here we first build a model where we include the effect of geographic location (longitude and latitude) in the model (mod_house_geolocation), and next, we include the interaction of longitude and latitude in the model, because we suspect that these parameters might influence each others association with price.

```
mod_house_geolocation = lm(price ~ sqft_living + grade + long +
    lat, data = data_house)
summary(mod_house_geolocation)
```

```
##
## Call:
## lm(formula = price ~ sqft_living + grade + long + lat, data = data_house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -258792  -86955  -19769   51876 1088588
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.657e+07  8.291e+06  -4.411 1.70e-05 ***
## sqft_living  1.303e+02  2.209e+01   5.899 1.59e-08 ***
## grade        5.282e+04  1.423e+04   3.710  0.00027 ***
```

```
## long        -8.043e+04  6.541e+04  -1.230  0.22035
## lat          5.588e+05  7.772e+04   7.190 1.35e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 150300 on 195 degrees of freedom
## Multiple R-squared:  0.5034, Adjusted R-squared:  0.4932
## F-statistic: 49.42 on 4 and 195 DF,  p-value: < 2.2e-16
```

```
mod_house_geolocation_inter2 = lm(price ~ sqft_living + grade +
    long * lat, data = data_house)
summary(mod_house_geolocation_inter2)
```

```
##
## Call:
## lm(formula = price ~ sqft_living + grade + long * lat, data = data_house)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -241443  -83349  -19197   50096 1088613
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.631e+09  3.314e+09   1.397 0.163874
## sqft_living  1.310e+02  2.204e+01   5.942 1.28e-08 ***
## grade        5.433e+04  1.424e+04   3.816 0.000183 ***
## long         3.811e+07  2.712e+07   1.406 0.161466
## lat         -9.760e+07  6.969e+07  -1.400 0.162969
## long:lat    -8.031e+05  5.702e+05  -1.408 0.160586
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 150000 on 194 degrees of freedom
## Multiple R-squared:  0.5084, Adjusted R-squared:  0.4958
## F-statistic: 40.13 on 5 and 194 DF,  p-value: < 2.2e-16
```

Note that the adjusted R squared did not increase substancially due to the inclusion of the interaction term, so it might not be so useful to take into account the interaction, it might be enoug to take into account the main effects of longitude and latitude. This needs to be further evaluated with model comparison. See the exercise related to that.