Exercise 12 - Multiple regression

Zoltan Kekecs

14 november 2019

Contents

1 Abstract

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-2019

2 Data management and descriptive statistics

2.1 Loading packages

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

```
library(psych)  # for describe
library(lm.beta)  # for lm.beta
library(tidyverse)  # for tidy format
library(gridExtra)  # for grid.arrange
```

2.1.1 Load custom functions

This is a custom function that I wrote which helps in creating the final table for the regression coefficients.

```
coef_table = function(model) {
    require(lm.beta)
    mod_sum = summary(model)
    mod_sum_p_values = as.character(round(mod_sum$coefficients[,
        4], 3))
    mod_sum_p_values[mod_sum_p_values != "0" & mod_sum_p_values !=
        "1"] = substr(mod_sum_p_values[mod_sum_p_values != "0" &
        mod_sum_p_values != "1"], 2, nchar(mod_sum_p_values[mod_sum_p_values !=
        "0" & mod_sum_p_values != "1"]))
    mod_sum_p_values[mod_sum_p_values == "0"] = "<.001"


    mod_sum_table = cbind(as.data.frame(round(cbind(coef(model),
        confint(model), c(0, lm.beta(model)$standardized.coefficients[c(2:length(model$coefficients))]),
        2)), mod_sum_p_values)
    names(mod_sum_table) = c("b", "95%CI lb", "95%CI ub", "Std.Beta",
        "p-value")
    mod_sum_table["(Intercept)", "Std.Beta"] = "0"
    return(mod_sum_table)
}
```

2.2 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. This dataset contains house sale prices for King County, USA (Seattle and sorrounding area) which includes Seattle. It includes homes sold between May 2014 and May 2015. More info about the dataset here: https://www.kaggle.com/harlfoxem/housesalesprediction

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

```
# alternative:
# https://raw.githubusercontent.com/kekecsz/PSYP13_Data_analysis_class-2019
# /master/data_house_small_sub.csv
data_house = read_csv("https://bit.ly/2DpwKOr")
```

2.3 Check the dataset

You should always get familiar with the dataset you are usung, and check for any inconsistencies that need to be corrected.

In the code below we convert the area metrics that are in square feet in the original dataset to square meters. We also specify that the variable has_basement is a factor. Then we display simple descriptives with the summary() function.

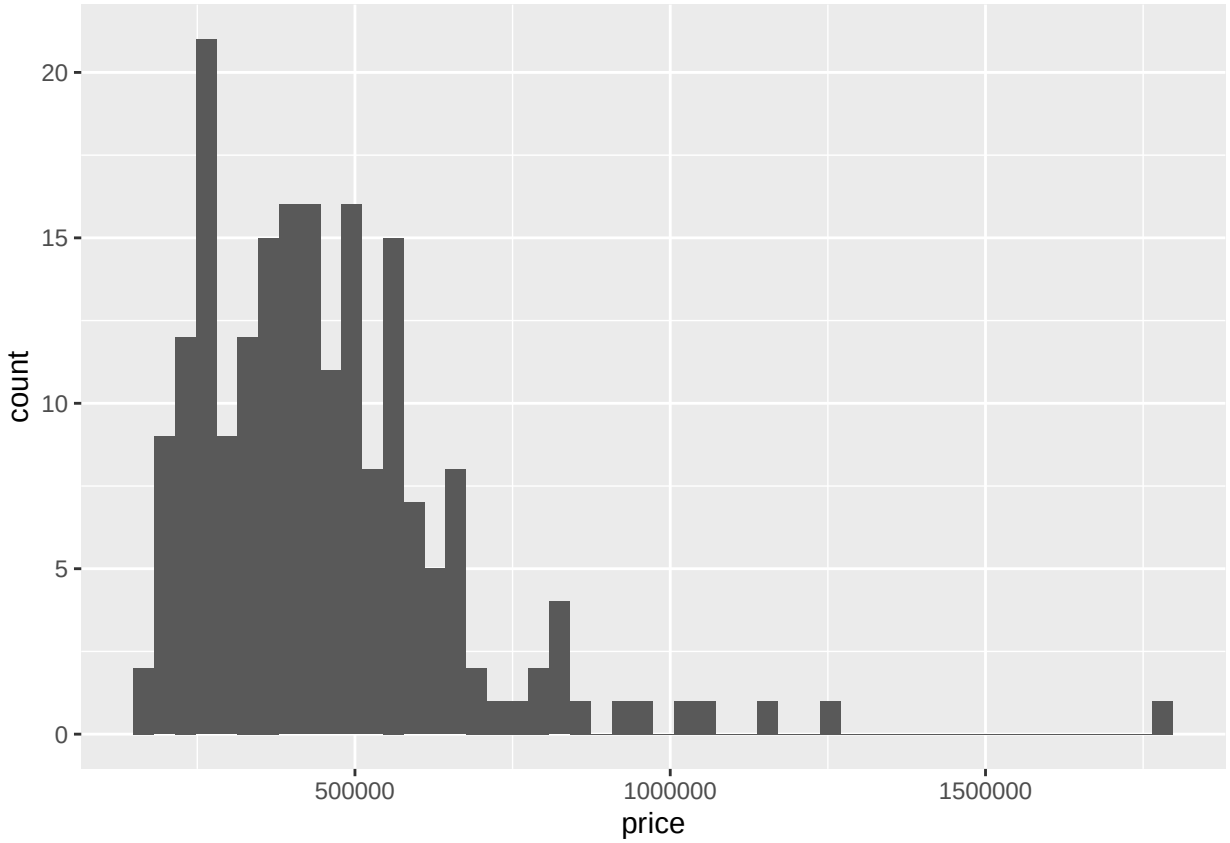
```
data_house = data_house %>% mutate(sqm_living = sqft_living *
    0.09290304, sqm_lot = sqft_lot * 0.09290304, sqm_above = sqft_above *
    0.09290304, sqm_basement = sqft_basement * 0.09290304, sqm_living15 = sqft_living15 *
    0.09290304, sqm_lot15 = sqft_lot15 * 0.09290304, has_basement = factor(has_basement))

data_house %>% summary()

# descriptive statistics
describe(data_house)
```

We should now do some more focused exploration on the variables of interest.

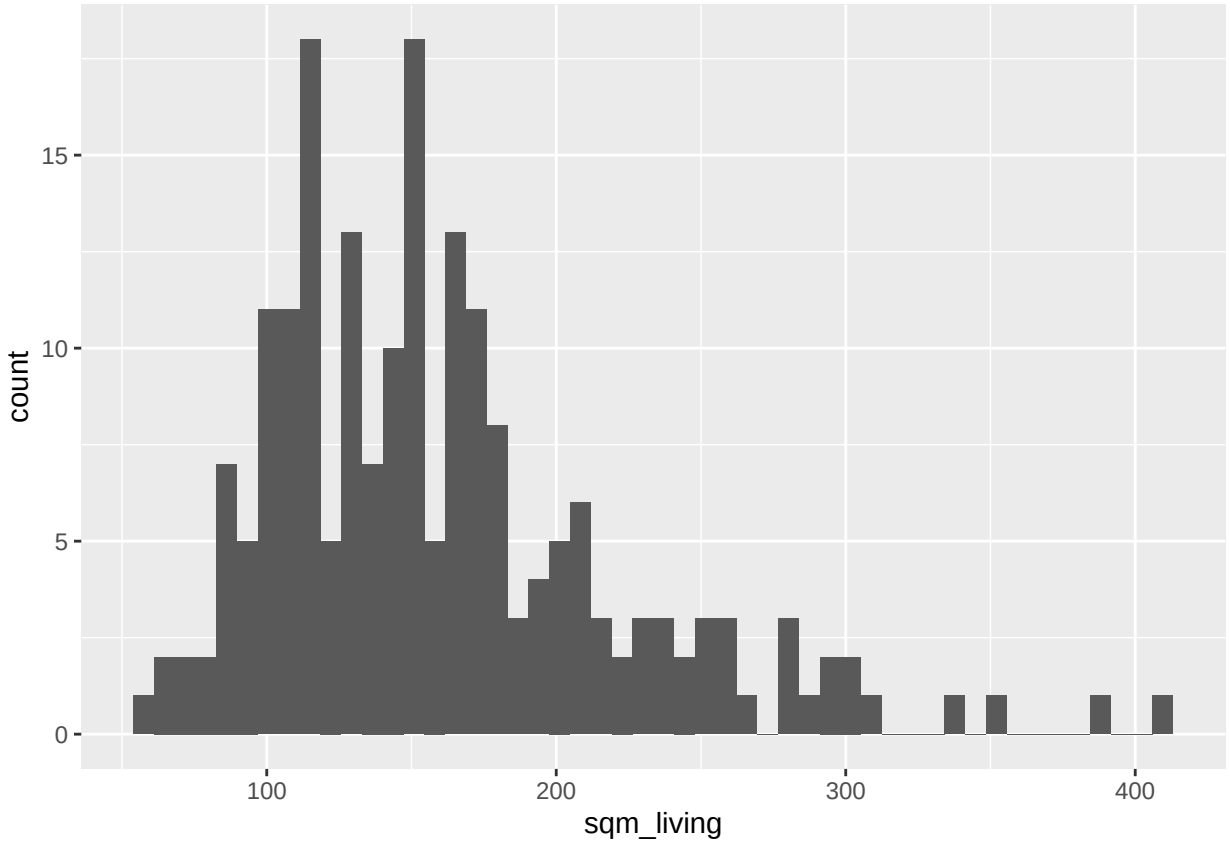
We are going to predict price of the apartment using the variables sqm_living (the area of the living area), and grade (overall grade given to the housing unit, based on King County grading system), so lets focus on these variables.

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

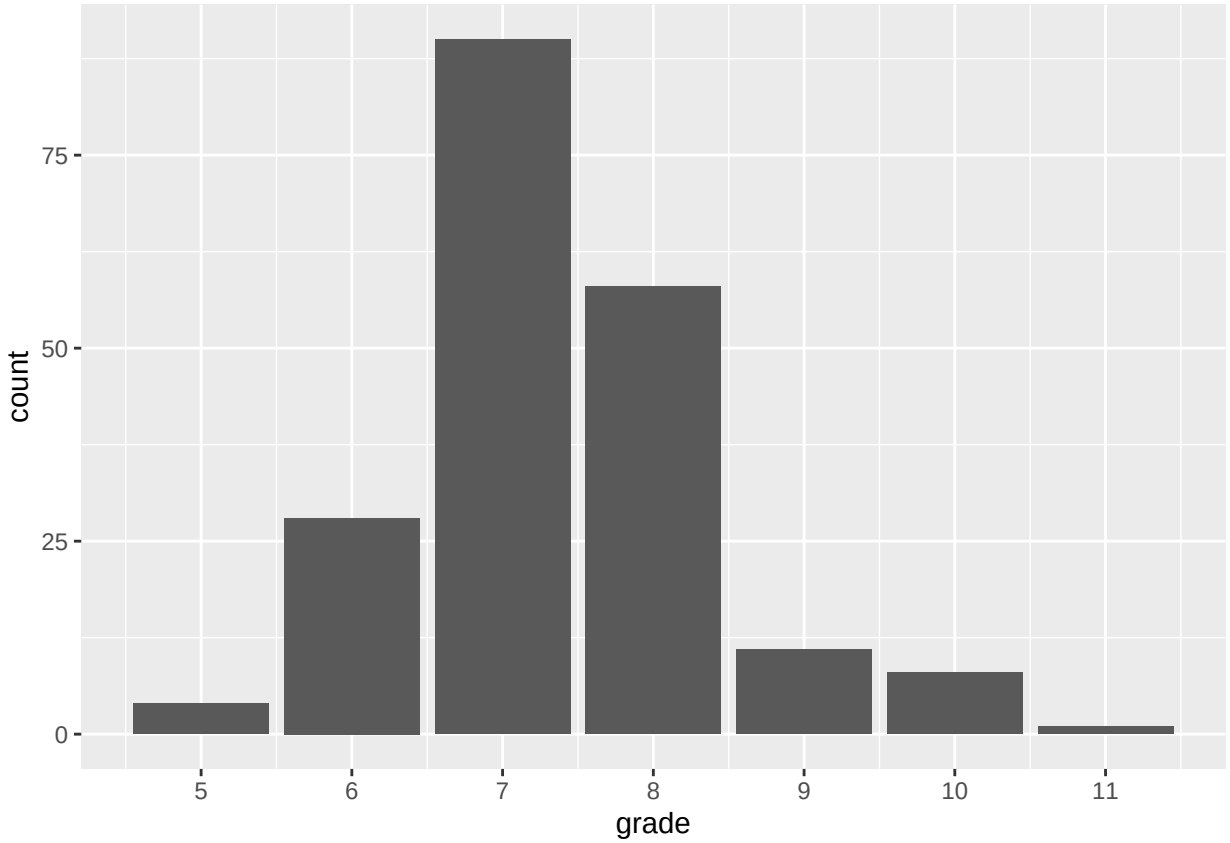
```
# histogram
data_house %>% ggplot() + aes(x = price) + geom_histogram(bins = 50)
```



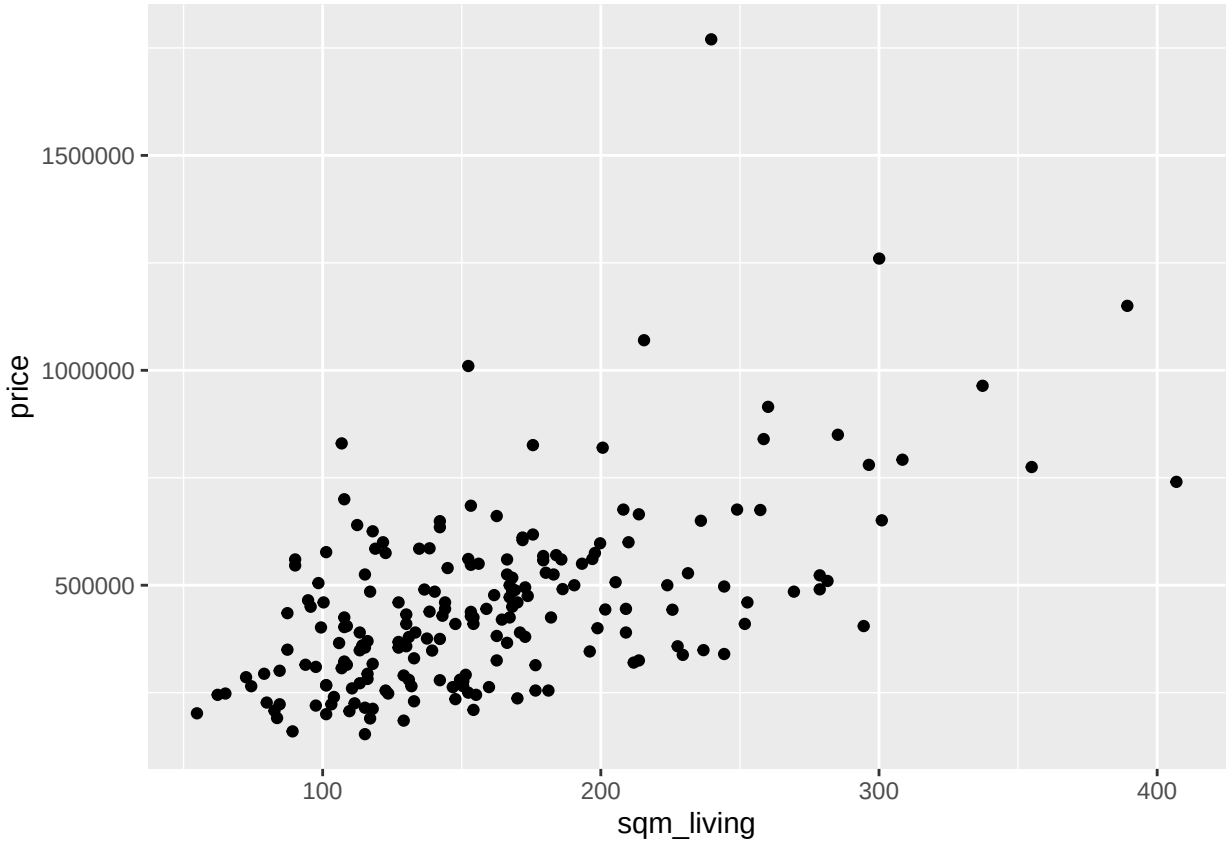

```
data_house %>% ggplot() + aes(x = sqm_living) + geom_histogram(bins = 50)
```



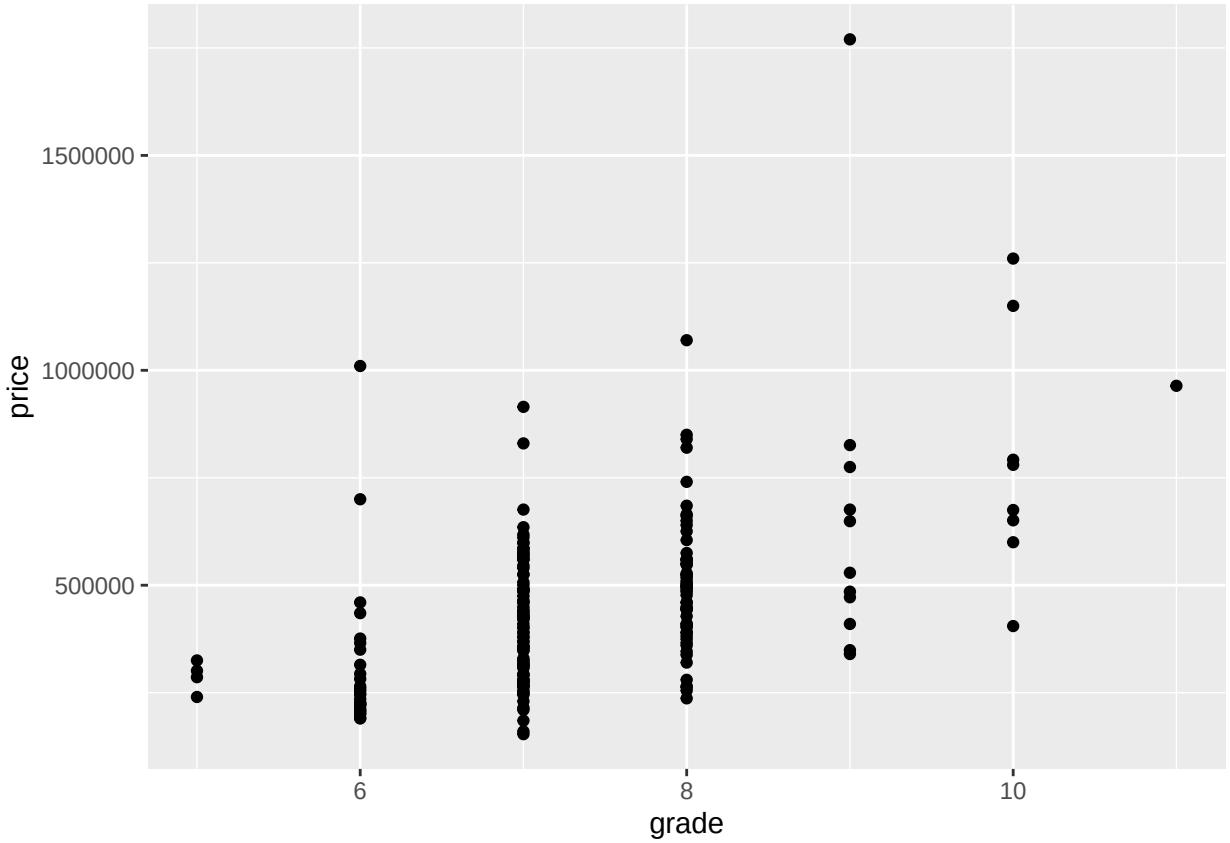

```
data_house %>% ggplot() + aes(x = grade) + geom_bar() + scale_x_continuous(breaks = 4:12)
```




```
# scatterplot
data_house %>% ggplot() + aes(x = sqm_living, y = price) + geom_point()
```




```
data_house %>% ggplot() + aes(x = grade, y = price) + geom_point()
```



3 Multiple regression

3.1 Fitting the regression model

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

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

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

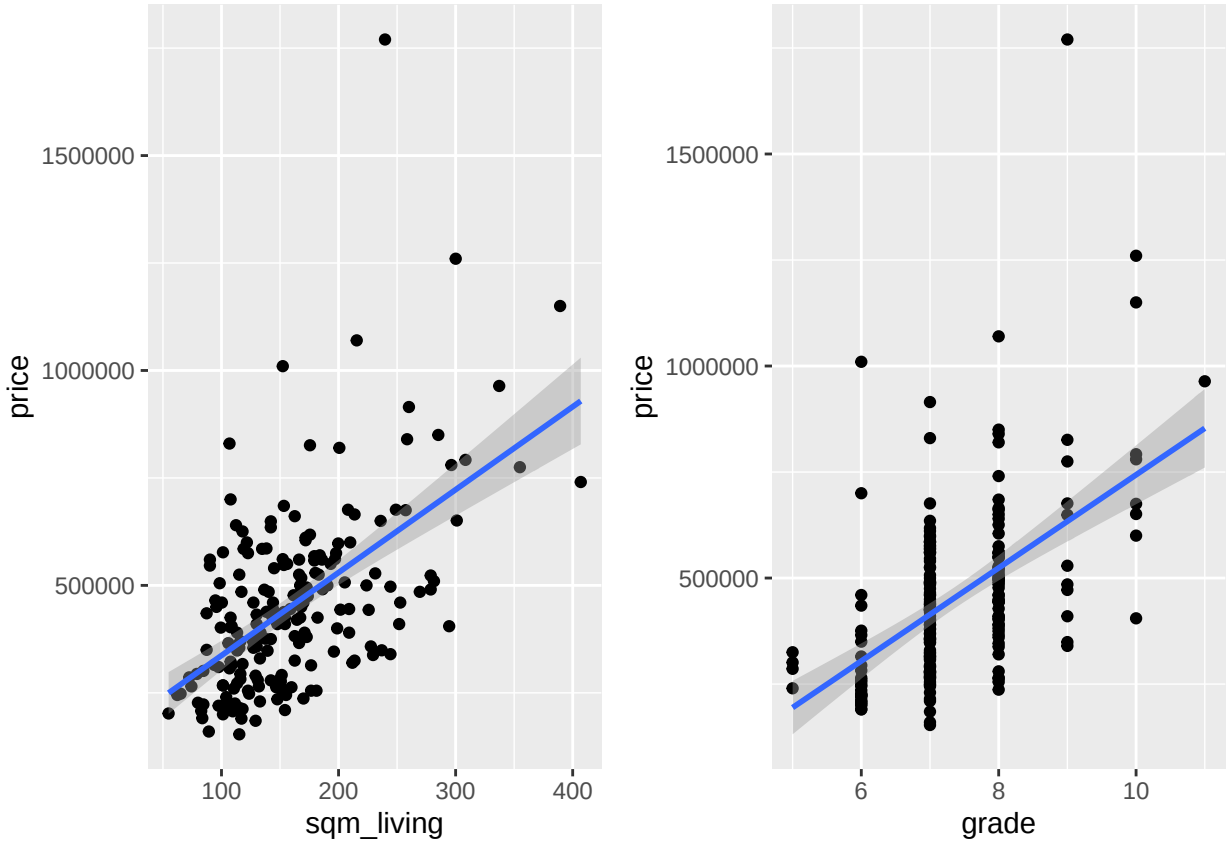
```
mod_house1
```

```
##
## Call:
## lm(formula = price ~ sqm_living + grade, data = data_house)
##
## Coefficients:
## (Intercept)   sqm_living        grade
##     -174390         1283        57353
```

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.

```
# scatterplot
plot1 = data_house %>% ggplot() + aes(x = sqm_living, y = price) +
    geom_point() + geom_smooth(method = "lm")
```

```
plot2 = data_house %>% ggplot() + aes(x = grade, y = price) +
    geom_point() + geom_smooth(method = "lm")

grid.arrange(plot1, plot2, nrow = 1)
```



(Alternatively, you can use a 3 dimensional plot to visualize the regression plane, but there are no good tools for that right now that I know of.)

3.2 Prediction

Again, we can ask for predictions for specific values of predictors, but we need to specify all predictor values (in this case, both sqm_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.

```
sqm_living = c(60, 60, 110, 110)
grade = c(6, 9, 6, 9)
newdata_to_predict = as.data.frame(cbind(sqm_living, grade))
predicted_price = predict(mod_house1, newdata = newdata_to_predict)

cbind(newdata_to_predict, predicted_price)
```

```
##   sqm_living grade predicted_price
## 1         60     6        246693.0
## 2         60     9        418751.4
## 3        110     6        310831.5
```

```
## 4        110     9        482889.9
```

4 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 size of living area (in m^2) 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 ~ sqm_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) -174389.9    95255.2  -1.831 0.068646 .
## sqm_living     1282.8      266.5   4.813 2.96e-06 ***
## grade         57352.8    16052.8   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)
```

```
##                     2.5 %    97.5 %
## (Intercept) -362240.5876 13460.864
## sqm_living      757.1467  1808.393
## grade         25695.4160 89010.156
```

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

```
##
## Call:
## lm(formula = price ~ sqm_living + grade, data = data_house)
##
## Standardized Coefficients::
## (Intercept)  sqm_living       grade
##   0.0000000   0.3740724   0.2776905
```

The final table should look something like this:

4.1 Final coefficient table

This is how the final coefficient table should look like. This was generated using the coef_table() function. This is one of my own functions that is included in the top of this code, but it is not necessary to use this function, you can also manually write out all relevant numbers from the outputs above.

```
##                        b   95%CI lb 95%CI ub Std.Beta p-value
## (Intercept) -174389.86 -362240.59 13460.86        0    .069
## sqm_living     1282.77     757.15  1808.39     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.

4.1.1 Interpretation of the regression coefficients

The interpretation of the regression coefficients of the predictors is: this is the amount by which the outcome variable's estimate would change if the predictor's value is increased by 1.
In our example the regression coefficient linked to sqm_living is 1282.77. This means that an increas in the area of the apartment by 1 m^2 results in an increased estimate in the price of the apartment by 1282.77.

4.1.2 Interpretation of the estimate of the intercept

The coefficient of the intercept is a contant (different for each regression model) that is not dependent on the values of the predictors. It can be interpreted as if all the predictors in the model would have the value of zero (0), this would be the estimated value for the outcome. (Be careful that this often does not represent a true physical reality, if a 0 predictor value is meaningless, nevertheless, the matematical interpretation stays the same.)

4.1.3 Interpretation of the standard beta

The benefit of the regression coefficient is that it is on the same metric as the predicted variable, making the influence of each predictor easy to interpret. However, we need to realize that this value is also dependent on the scale of the predictor. This makes it hard to directly compare the influence of predictors that use different scales just using the regression coefficient.

In order to be able to directly compare the predictive value contained by each predictor in the model, we can use the standardized Beta coefficient. This value is computed by refitting the model with the standardized

predictors. Using the standardized Beta coefficient we can directly compare the predictive value of predictors within the context of the whole model. It is important to note that the predictive value of any given predictor in a model might be very different from its individual correlation with the outcome. This is because multiple predictors can explain the same portion of the variance, this way, the predictive value of any single predictor can be "maked" in the model by the predicitve value of other predictors explaining the same portion of the variance.