Anàlisi de Dades Complexes - Homework 1

Ona Sánchez: 1601181, Alejandro Donaire: 1600697

February 28, 2022

Problem 1

The model in equation form

The multiple lineal regression model in equation form would look like the following:

$$SP_i = \beta_0 + \beta_1 Y_i + \beta_2 FE_i + \beta_3 FL_i + \beta_4 FP_i + \beta_5 ST_i + \beta_6 T_i + \beta_7 O_i + \beta_8 KM_i$$

where β_i are the coefficients we would like to estimate, and the rest of the variables mean, in order:

- $SP_i :=$ "Selling prince".
- $Y_i :=$ "Year".
- $FE_i :=$ "Fuel Electric".
- $FL_i :=$ "Fuel LPG".
- $FP_i :=$ "Fuel Petrol".
- $ST_i :=$ "Seller type".
- $T_i :=$ "Transmission".
- $O_i :=$ "Owner type".
- $KM_i :=$ "Kilometers driven".

The model in matrix form

The design matrix X for the model would look like the following:

$$X = \begin{pmatrix} 1 & Y_1 & FE_1 & FL_1 & FP_1 & ST_1 & T_1 & O_1 & KM_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & Y_{1775} & FE_{1775} & FL_{1775} & FP_{1775} & ST_{1775} & T_{1775} & O_{1775} & KM_{1775} \end{pmatrix}$$

Where the first column corresponds to the intercept.

We have had to separate the different types of fuel to take into account each of them, so in the matrix we put a 1 in the corresponding column, if the three columns (Electric, LPG and Petrol fuel) have the value 0, it means that the type of fuel is Diesel. For the seller type, we put a 1 if the type is “Individual” and a 0 if it is “Dealer”. For the transmission, a 1 indicates “Manual”, and a 0 “Automatic”. For the owner, a 1 indicates a “second above owner”, and a 0 a “first owner”. The first column, the intercept is filled by 1s and the year and km driven are the corresponding values, different for each row.

Interpretation of the coefficients

Once we have read the data from the Excel file, we can proceed to compute a multiple lineal regression model with the `lm()` command in `R`. Using the following code:

```
library("readxl")
path="C:/Users/onasa/Documents/2n2nsemestre/Dades Complexes/cars.xlsx"
data = read_excel(path)

cars_lm <- lm(data$selling_price ~ data$year + data$fuel + data$seller_type +
              + data$transmission + data$owner + data$km_driven,
              data = data)
```

we obtain a multiple linear regression model as specified. Using the `summary()` command in `R`, we obtain the following information regarding the model:

```
Residuals:
    Min      1Q  Median      3Q     Max
-1251.0  -186.1   -39.2   125.7  7520.4

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                    -8.962e+04  7.598e+03 -11.795  < 2e-16 ***
data$year                       4.528e+01  3.766e+00  12.024  < 2e-16 ***
data$fuelElectric              -8.259e+02  4.737e+02  -1.743  0.08143 .
data$fuelLPG                   -2.892e+02  1.581e+02  -1.829  0.06756 .
data$fuelPetrol                -2.851e+02  2.489e+01 -11.453  < 2e-16 ***
data$seller_typeIndividual     -3.790e+01  2.663e+01  -1.423  0.15479
data$transmissionManual        -8.721e+02  3.393e+01 -25.700  < 2e-16 ***
data$ownerSecond & Above Owner -3.624e+01  2.710e+01  -1.338  0.18120
data$km_driven                 -8.112e-01  3.119e-01  -2.601  0.00937 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 471.3 on 1766 degrees of freedom
Multiple R-squared:  0.4273,Adjusted R-squared:  0.4247
F-statistic: 164.7 on 8 and 1766 DF,  p-value: < 2.2e-16
```

Firstly, we can comment on the values of the coefficients $\beta_0, ..., \beta_8$:

- β_0 = `-8.962e+04`. An intercept with this value means that if every other parameter is 0, then the price of the car is -89,620$. This makes sense, for the car should be more than 2000 years old! The other coefficients do not affect the price as much, since they are mainly qualitative, except the kilometers driven. And if this last parameter is 0, this simply means that the car has essentially not moved, which is not as dramatic as having a 2000-years old car.

- β_1 = `4.528e+01`. This coefficient gives a value to the car depending on how old is the car. An older car would cost less than a brand new car, and the value of the car increases 45,28$ for year. So if the only parameter that is changed is the year, if we increase the parameter, the car is more expensive, and if we decrease it, it is cheaper.

- β_2 = `-8.259e+02`. If we had two cars with exactly the same parameters and the only thing that changed was the fact that the car runs on electric fuel, then we wold expect to observe a difference of about 830$. More precisely, if the car is in fact electric and not diesel, the car should be about 830$ cheaper.

- β_3 = `-2.892e+02`. Likewise, having a car that runs on LPG fuel, as opposed to diesel, would mean that the car should be about 280$ cheaper.

- β_4 = `-2.851e+02`. The explanation for this coefficient is just like before, and in this case we would have a decrease of exactly 285.1$.

- β_5 = `-3.790e+01`. This parameter indicates if the seller type is individual or not. If it is individual, the parameter will have the value 1, and the value of the car will decrease 37,90$ (given by the coefficient). If the seller type was dealer, the value would be 0.

- β_6 = `-8.721e+02`. In this case, (again, assuming all other parameters stay fixed), if we had a car with manual transmission as opposed to automatic, we would expect a decrease of 870$ in the selling price.

- β_7 = `-3.624e+01`. In this case, if the car was second above owner as opposed to first owner, the selling price would decrease by 36,24$.

- β_8 = `-8.112e-01`. If we had two cars with the same parameters and only difference was the kilometers driven, we would expect a decrease in the selling price of 81$ for each kilometer.

More comments on the output of `summary()`

Coefficients aside, we also have important values we should comment on, such as the multiple R-squared, the p-values associated with the `t` statistics and the standard errors:

R-squared: At first glance, we can see that its value is rather small, which means the observed variation is not perfectly explained by the model's input. Ideally, we would want a value as close as possible to 1, but we only have a value around 0.4.

P-value: For most coefficients the p-values are small, which means we can reject the null hypothesis in most cases. However, in the case of seller type and owner type, it is not clear that the parameter is affecting the final selling price.

Standard error: This number shows how much variation there is around the estimates of the regression coefficient, in our case, the residual standard error is 471,3 which means that if we make a prediction, we can expect a variation of ± 471.3$ in the selling price.

Problem 2

Fitting a fourth order polynomial

With `R` we can fit a fourth order polynomial using the selling price and years data. In the code below, first we saved our data in variables corresponding to the selling price and year and then proceeded to fit a polynomial using `lm()` and `poly()` commands. Right after that, we created a plot to visualize the resulting fitted polynomial using `predict()`, `lines()` and `plot()`.

```
# We fit a model
selling_price = data$seller_type
year = data$year
cars_poly <- lm(selling_price ~ poly(year,degree=4), data = data)

# We plot the polynomial using predictions
new_data <- data.frame(year=seq(from=1998,to=2020,by=0.1))
predict(cars_poly, new_data)
lines(predict(cars_poly, new_data))
plot(data$year,data$selling_price,col=rgb(0.4,0.4,0.8,0.6),
    pch=16 , cex=1.3, xlab=’Year’, ylab=’Selling price’)
lines(seq(from=1998,to=2020,by=0.1), predict(cars_poly, new_data),
    type="l", col="red", lwd=2)
```

Figure 0.1 shows the resulting plot.

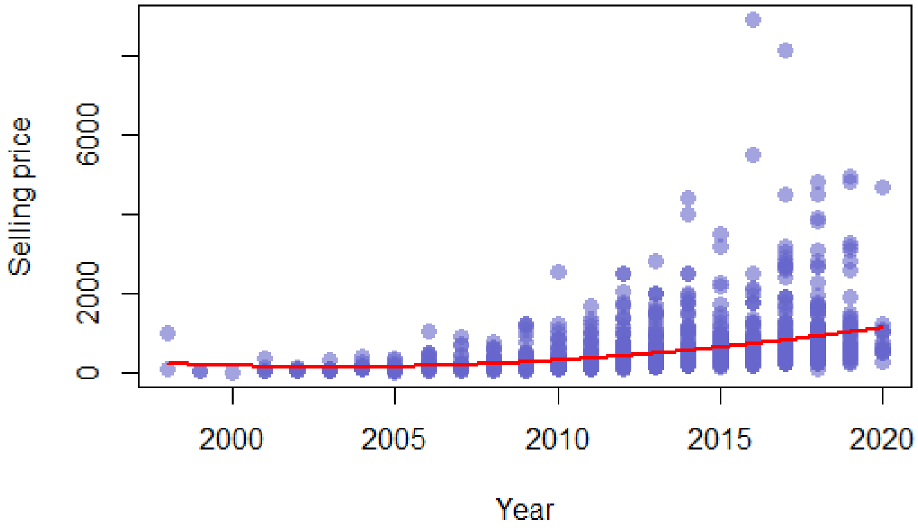

Figure 0.1: A fourth order polynomial fitting the data in red

Also, using `cars_poly$coefficients` we can see what our polynomial looks like:

$$p(x) = 558.39641 + 9904.31947x + 2899.89226x^2 - 56.32234x^3 - 45.15908x^4$$

In general, we can see that the polynomial does a good job approximating the data. As we expected, the tendency is upwards and the polynomial passes through the densest region of points.

Predictions

We can predict the selling price values for 2007 and 2017 using `R` with the function `predict()`. The code used is the following:

```
# Selling price predictions for years 2007 and 2017
```

```
my_preds <- data.frame(year=c(2007, 2017))
predict(cars_poly, my_preds)
```

And the resulting predictions for 2007 and 2017 are, in order:

```
    199.9586 826.8544
```

Confidence interval

We can create a confidence interval with using `confint()` in `R`. Right after that, we calculated the confidence "bands" and created a plot to visualize them.

```
    # 95% CI
confint(cars_poly, level=0.95)
    # Plot of the confidence interval
ci = predict(cars_poly, data.frame(year=seq(from=1998,to=2020,by=0.1)),
interval=c("confidence"))
lines(seq(from=1998,to=2020,by=0.1),ci[,2],col="orange",lty=1)
lines(seq(from=1998,to=2020,by=0.1),ci[,3],col="orange",lty=1)
```

The resulting confidence interval is:

```
                             2.5 %     97.5 %
(Intercept)               531.7867   585.0061
poly(year, degree = 4)1  8783.2334 11025.4055
poly(year, degree = 4)2  1778.8062  4020.9783
poly(year, degree = 4)3 -1177.4084  1064.7637
poly(year, degree = 4)4 -1166.2451  1075.9270
```

And the resulting plot is:

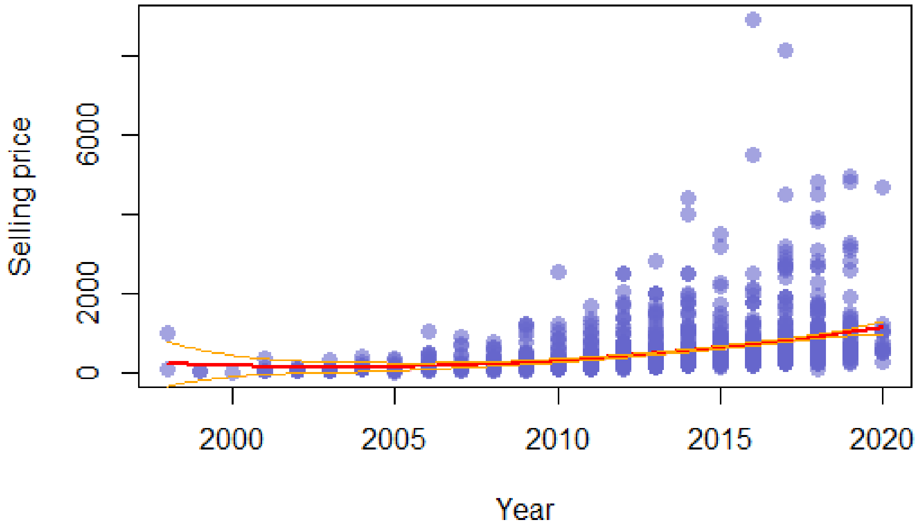

Figure 0.2: 95% CI for the fitted 4th order polynomial (in orange)

Problem 3

Boxplot

The brands we have chosen are Audi and Nissan. With the `R` command `subset()` we can filter our data so we can only keep those cars that are either Audi or Nissan. Once we have the data filtered, we can use `boxplot()` to generate the required plot.

Here is the code we used:

```
brand = data$brand
brands_data <- subset(data, brand=='Audi' | brand=='Nissan')
boxplot(selling_price~brand, data=brands_data, cex.axis=0.8,
        xlab='Brand', ylab='Selling price')
```

And here is the resulting plot:

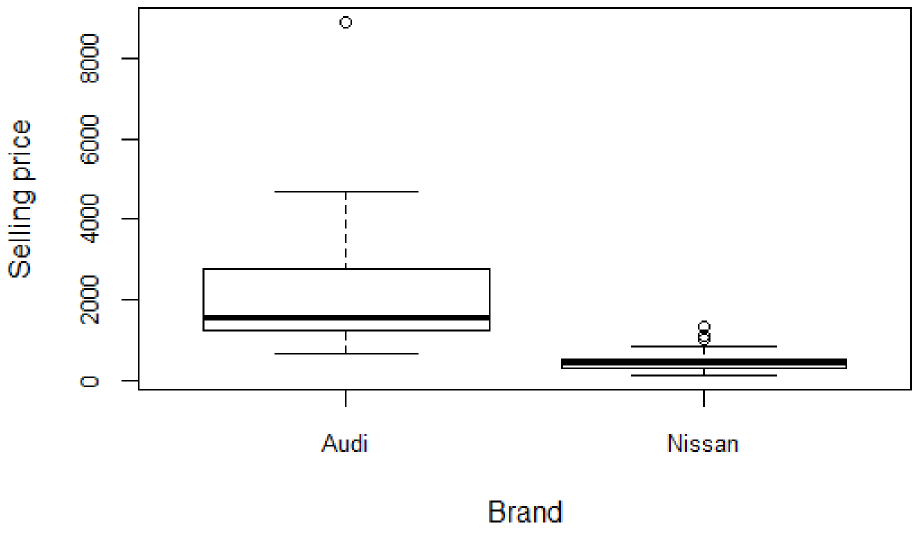

Figure 0.3: Boxplot for brands Audi and Nissan and their prices

Generalized T-test

We now perform a generalized T-test using the following code in `R`:

```
subset_nissan <- subset(data,brand=='Nissan')
subset_audi <- subset(data, brand=='Audi')
subset <-subset(data, brand=="Audi" | brand=="Nissan")

var(subset_nissan$selling_price)
var(subset_audi$selling_price)
sqrt(var(subset_nissan$selling_price))
sqrt(var(subset_audi$selling_price))

n1<-length(which(data$brand=="Nissan"))
```

```
n2<-length(which(data$brand=="Audi"))
var(subset_nissan$selling_price)*n1+var(subset_audi$selling_price)*n2
df=n1+n2-2

sigma=sqrt((var(subset_nissan$selling_price)*(n1-1)+
            var(subset_audi$selling_price)*(n2-1))/(n1+n2-2))
mean(subset_audi$selling_price)-mean(subset_nissan$selling_price)
num=(sum(subset_nissan$selling_price)/n1-sum(subset_audi$selling_price)/n2)

denom=sigma*sqrt(1/n1+1/n2)
Tstat<-num/denom
2*pt(abs(Tstat),df=81, lower.tail=FALSE)
```

And we obtain a notably small p-value ($2.7 \cdot 10^{-11}$), signifying that the means for the Audi and Nissan selling prices are different. In particular, Audis are more expensive.

We can also use the function `pairwise.t.test()` to check that our calculations are correct:

```
xx<-pairwise.t.test(subset$selling_price, subset$brand, "none")
pairwise.t.test(subset$selling_price, subset$brand, "none")
```

And we do indeed see that the results are the same.