Laptop Pricing

Attila Szuts

02/01/2021

Abstract

This analysis will try to answer how you can price a laptop based on its specifications and how you can try to find good deals among them. It uses 1300 data points to build a linear regression model on price using different properties like company that manufactured it, cpu model, RAM size, etc.

Contents

Introduction **1**

Data prep **1**

Data collection 1

Data cleaning 1

Descriptives 2

Model parameters **2**

Model interpretation 2

Residual analysis **3**

Robustness checks **3**

Summary **4**

Appendix **6**

Log Price distribution 6

Pattern of association between ln_price and predictors 7

Model Assumptions 24

Detailed model comparison 29

Standardized residuals 29

Y - Y hat plot 29

Compare Train and Test model 29

Introduction

Following my interest of tech gadgets I wanted to investigate a question, that was bothering me for a while. How can we accurately tell and compare prices of different laptops? It is hard by itself to price them, as usually they are priced at a premium than similar speced custom built PCs, not only because of the portability but also because the assembly costs. So in order to do this, I decided to build a linear regression model for pricing laptops based on their specifications (screen size, manufacturer, RAM, etc.)

Data prep

Data collection

Data was collected from kaggle, thus I do not know the original source of it. Since this is similar to administrative data, it is likely that the values are accurate and there is no inherent classical measurement error present. However, there can still be some abnormalities that can distort the model. For example it might be very important from a pricing perspective how long ago since the laptop has been released, since prices tend to decrease after the initial hype. Also, it can happen quite easily that someone mistyped something if data was entered manually. The question still remains, which is that is price dependent exclusively on specs? Likely that no, there are other factors at play as well, such as design, build quality, materials used, marketing, etc. Also, there are some important properties not present, like battery size, number of ports, keyboard type, etc but for our purpose it will be a good enough approximation of price.

To get as much information as possible, I will try to use all variables in the model, and I'll only exclude them if I have to.

Data cleaning

I had to clean almost every variable, to clean numeric variables (RAM, SSD size, HDD size, etc.). Alongside this, I also had to mine information from different variables such as IPS, touchscreen properties of screens, or CPU model, manufacturer, frequency, etc. I created functions that could extract these informations into new ones.

I recoded the baseline for my variables. I choose these partly based on my personal interest and also on sample size to consider the SE of the coefficients.

- Operating system -> windows 10
- Company -> Lenovo
- Type -> Notebook
- Screen size category -> screen_mid
- Screen resolution -> 1920x1080
- CPU model -> core i7
- Memory type -> ssd
- GPU type -> integrated

Finally, I created an 80-20 train-test split.

Descriptives

We can see in the appendix that the log transformed version of prices is the closest to distribution.

Detailed plots on the pattern of association between log-price and covariates can be found in the appendix.

Table 1: Summary statistics for price

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
174	598.9225	979	1117.583	1468.75	6099

Model parameters

For the baseline model I am going to use a simple regression of log price on the manufacturer company. I decided to use ln prices, as it was skewed to the left, having a long right tail and this transformation made it normal. This model gives the average price percentage change for each manufacturer compared to a baseline, which in this case is Lenovo.

reg1: ln_price ~ company

To find out which coefficients provide the best fit for my data I ran a simulation to find the best possible fit using AIC with the `stepAIC` function.

However this model contained multicollinearity, so I had to drop some variables. After this step I came up with the final form:

reg3: ln_price ~ company + type_name + inches + ram + screen_category + cpu_model + memory_type + ssd_size + hdd_size

Model interpretation

This model can be interpreted as giving the price percentage change, when we change a property. $exp(\alpha)$ gives us the average price for a Lenovo Notebook with Windows 10, Intel core i7 CPU, medium sized FullHD display, SSD and integrated GPU which is 765.7421174 Euros. We can see how the price changes, if we change some property of this imaginary laptop or we can also analyse residuals to find the best deal for a certain category. You can find the detailed comparison in the appendix.

We can see that among the companies, Lenovo is somewhere in the lower end having quite a few more expensive counterparts (e.g. Apple), but also having more budget options (e.g. Acer).

Ultrabooks, Gaming, Convertible and workstation products are all significantly more expensive compared to Notebooks. Netbooks are cheaper, but marginally and insignificantly.

Inches, that is screen size is not a significant predictor, however, the categories were significant. Both smaller and larger screens are around 20% more expensive compared to medium sized screens. This is probably because of other factors (like premium devices being smaller so they are more portable or gaming pcs having large screens).

1GB more RAM will cost you 2% more on average.

The Intel core i7 is a very expensive cpu, and virtually any other option will cost you less (sometimes even as much as 70%!). This is a very interesting finding, since the CPU is basically the heart of the machine, if you can find a suitable cpu for your needs, then you can potentially save a lot on a pc. But in this paper I am not going to go into more details on this for obvious reasons.

Having both SSD and HDD will be 23% higher on average compared to having only SSD. However, interestingly 1 GB more storage is associated with just a fraction of a percent higher prices on average. But this can still be an impactful coefficient if we investigate the standardized betas.

If you are interested which properties of a laptop are the most important, you can take a look at the appendix where I dive into more detail on the coefficients, and investigate the standardized version of them which can tell us how much they impact the slope of the regression compared to each other. As I suspected earlier, the most important of them are RAM size, CPU type, SSD and HDD size and finally screen size category.

Table 2: Underpriced laptops

Company	Product Name	Price - Actual	Price - Predicted
Asus	G701VO-IH74K (i7-6820HK/32GB/2x	1279.0	2365.0639
HP	250 G6	393.9	718.2363
HP	15-BS101nv (i7-8550U/8GB/256GB/FHD/W10)	659.0	1189.7375
Lenovo	IdeaPad 510s-14IKB	799.0	1417.9101
Lenovo	Yoga 500-14ISK	638.0	1115.8852
Lenovo	Yoga 500-14ISK	638.0	1115.8852

Our model has a very high $R^2 = 0.85$ compared to the baseline $R^2 = 0.16$. I think, this is a relatively accurate and robust result. We will check this with test sample.

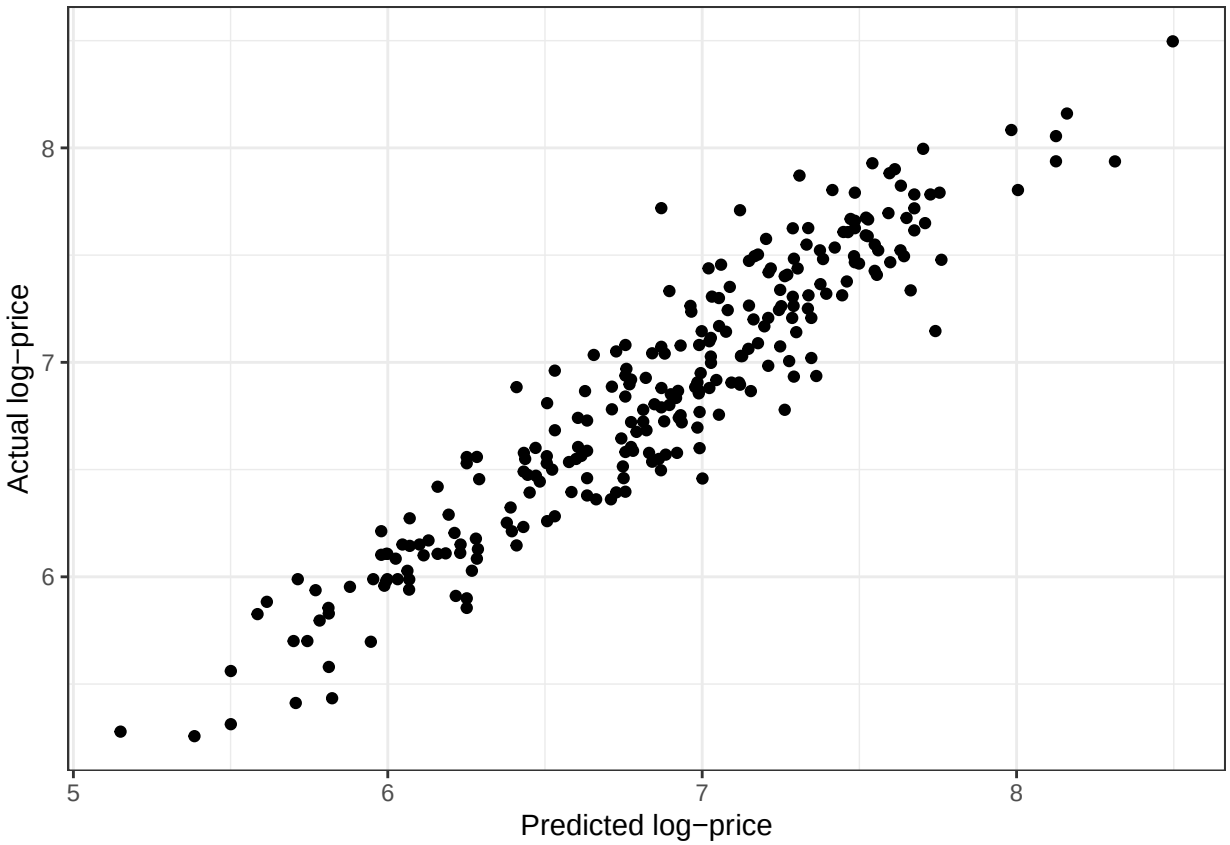
Residual analysis

Now that we are sure our model met all our assumptions (that can be checked in the appendix we can analyse the residuals to find the most valuable deals. I limited the results to be under €2000 because above that it might be that our model is less accurate and there are other factors in pricing premium products as well. Based on this, we get the following results.

Robustness checks

Let us see now the test sample results, how well does the model predict laptop prices.

You can see in the appendix the detailed comparison, but for the most part, all coefficients are the same (except for cases where the group size is small). The $R^2 = 0.86$ is also very similar to the one we got with the training sample. So we can say it with confidence that there is some reality to our model, if this sample is representative. The $Y - \hat{Y}$ plot is showing a good fit as well.



One more thing that I find worth mentioning is the case of overfitting. This might be the case, since we have a lot of different variables, however it is very hard to say anything about this. Hopefully thinking about external validity and finding different datasets that could be tested with this model could help shed some light on this question.

One of the most useful things to do with this model is to test it with prices from a different time. Maybe there we could uncover different patterns of associations for different laptops. For example, low-end pc-s price might drop significantly after release, but their performance could drop even faster so getting a deal on them might not be worth it. Similarly, high-end gaming laptops can hold their prices for a longer period of time and dropping in relative performance but not as much, so it would be worth it to buy it. It would also be interesting to see and try to fit this model on PCs or Smartphones as they share a lot of similiar properties.

Summary

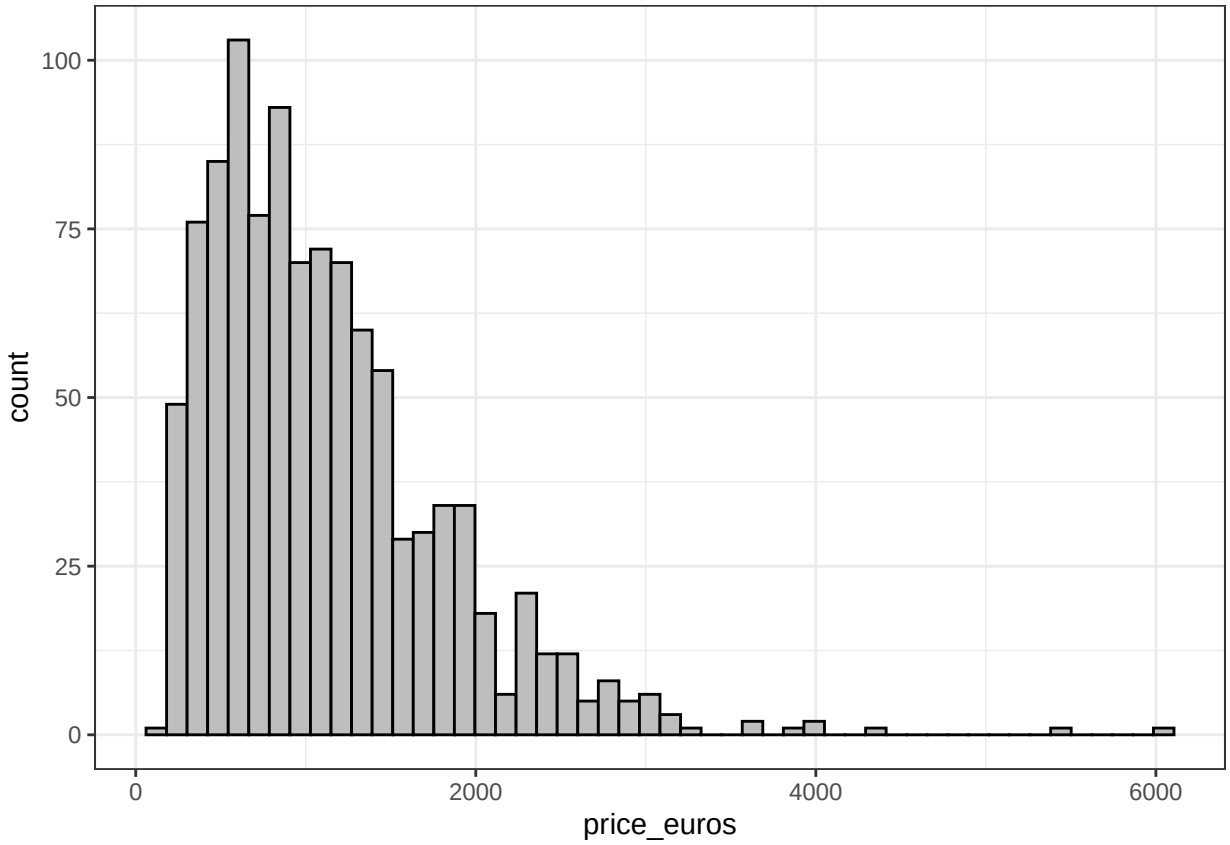
Finally, let me wrap up the findings here. All in all, the model I built fits the data very well and is suitable to find underpriced laptops. $Exp(\alpha)$ gives us the average price for a Lenovo Notebook with Windows 10, Intel core i7 CPU, medium sized FullHD display, SSD and integrated GPU which is 765.74 Euros. The most influential coefficients in the model were RAM size, CPU type, SSD and HDD size and finally, screen size category.

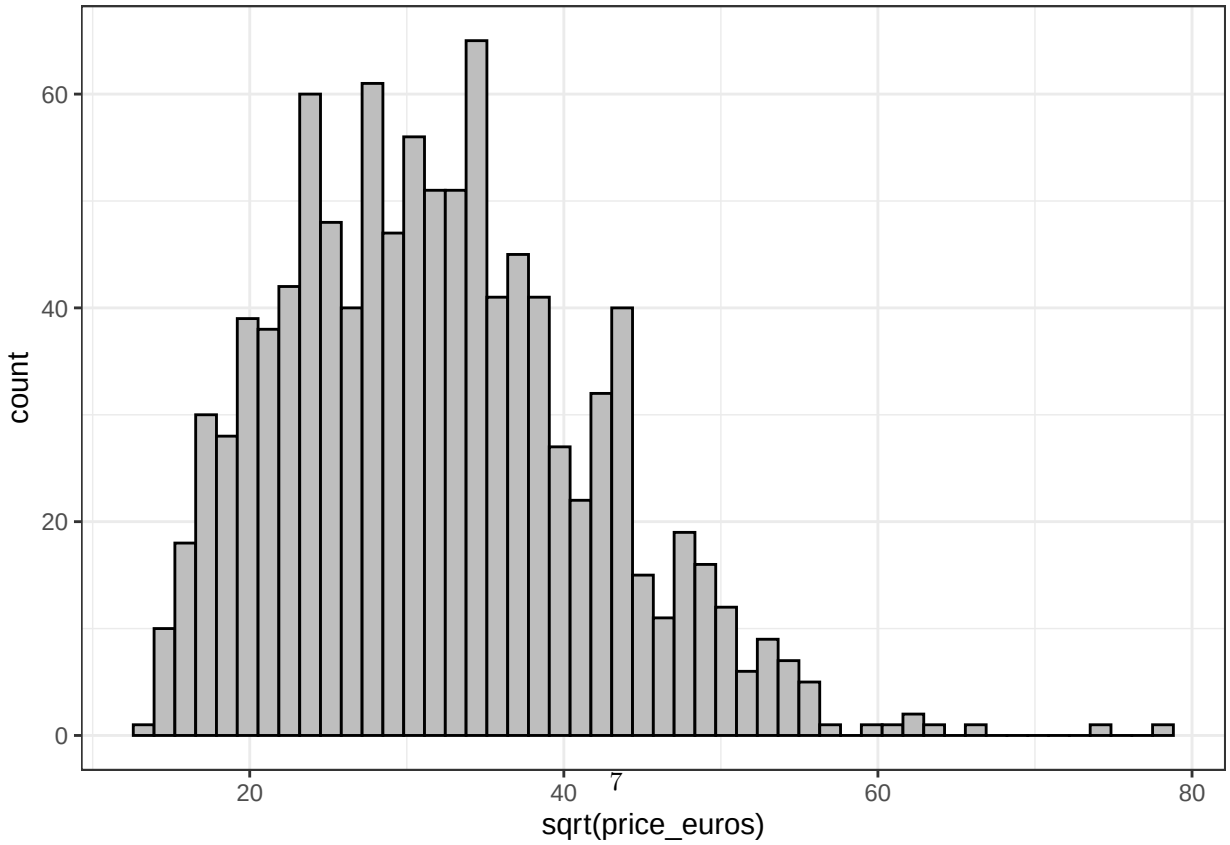
To make it more useful, and develop this project further, it would be a great next step, to investigate the question of external validity. It would be great if I could find data on different properties (for example benchmark scores on performance for each model) and try to model price with those or build a competing model with this and compare them. Or it would be another interesting research topic to try and model the

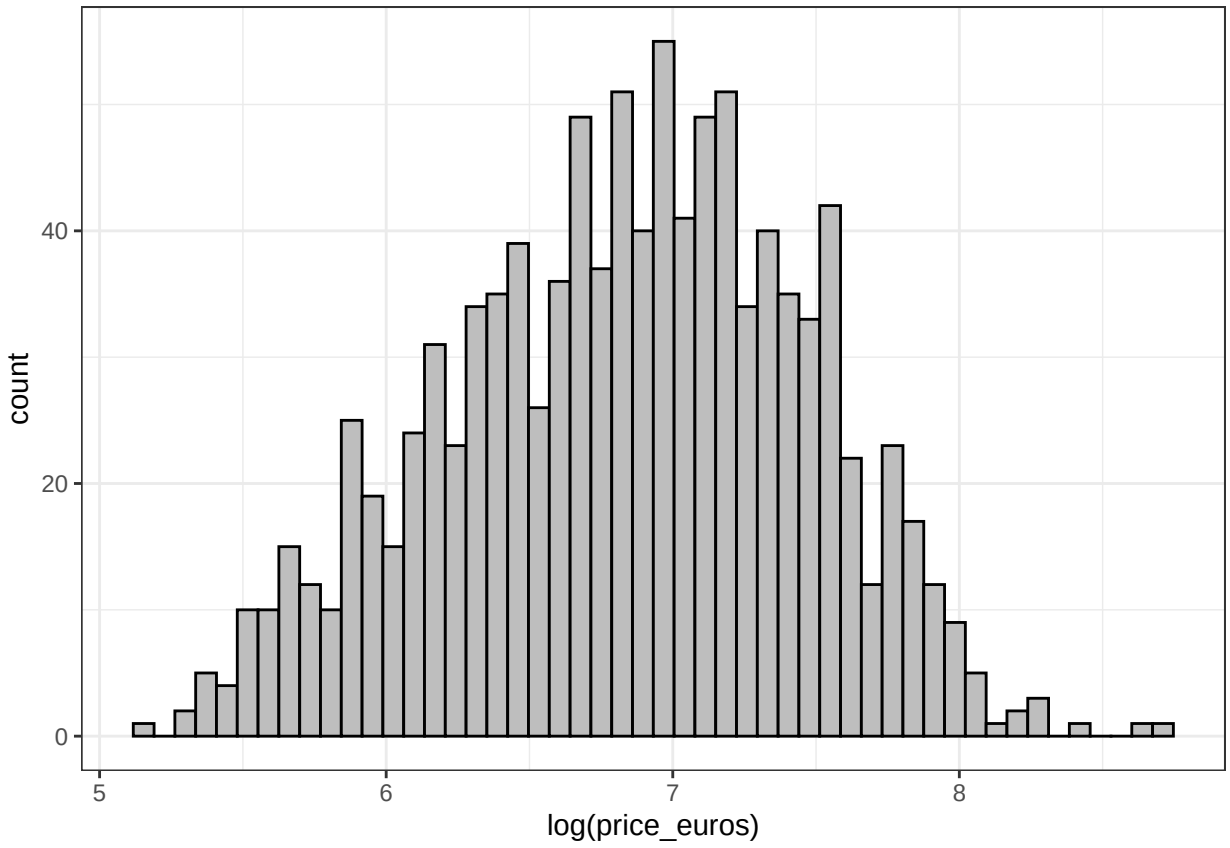
price conditioned on the number of days since release. To investigate external validity even further one could try to apply this model on a dataset with PCs and Smartphones.

Appendix

Log Price distribution

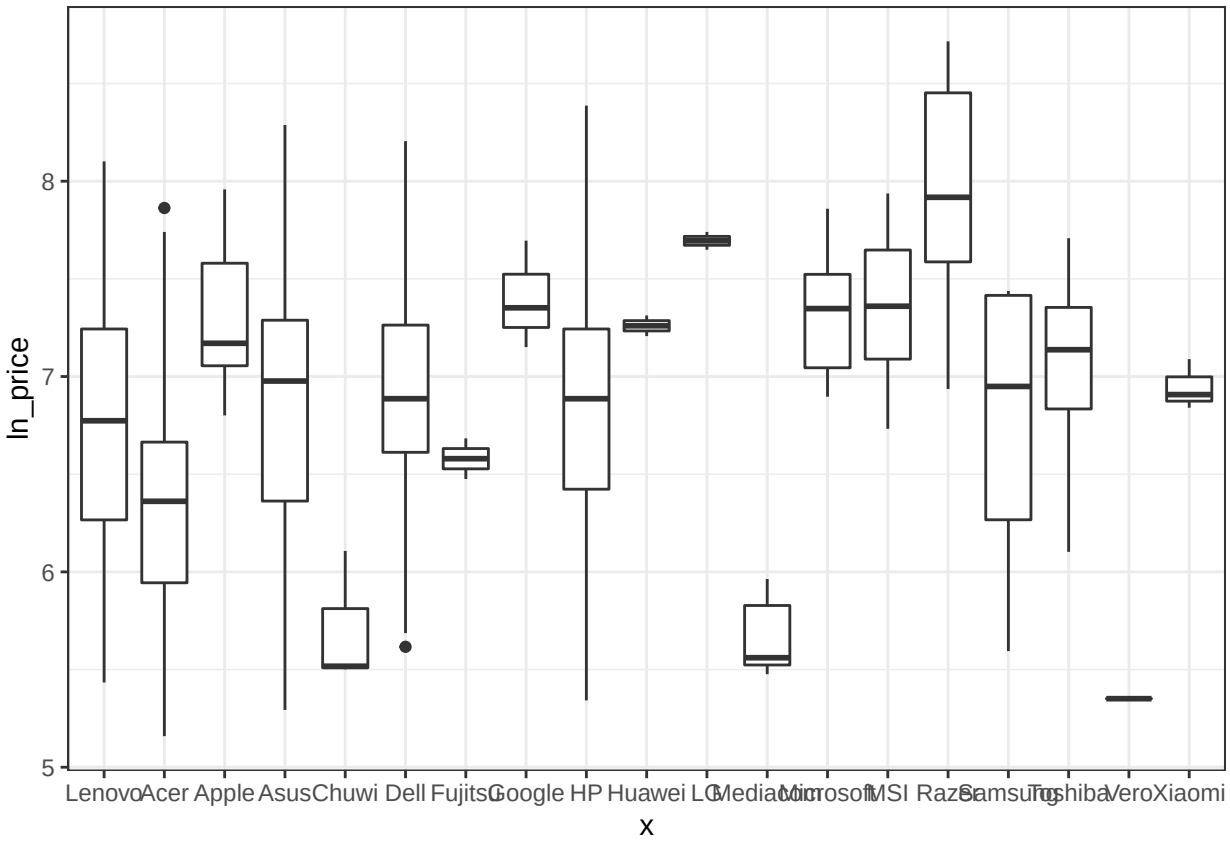




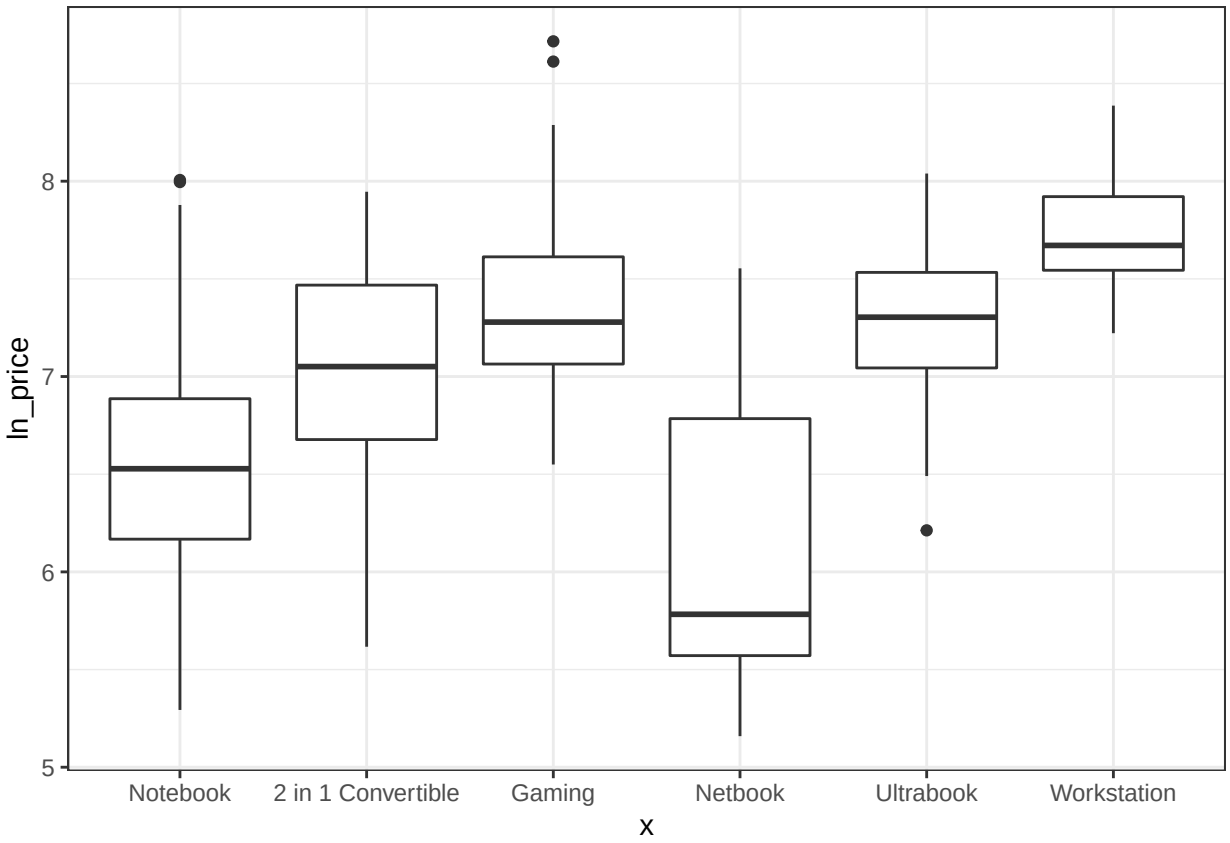


Pattern of association between ln_price and predictors

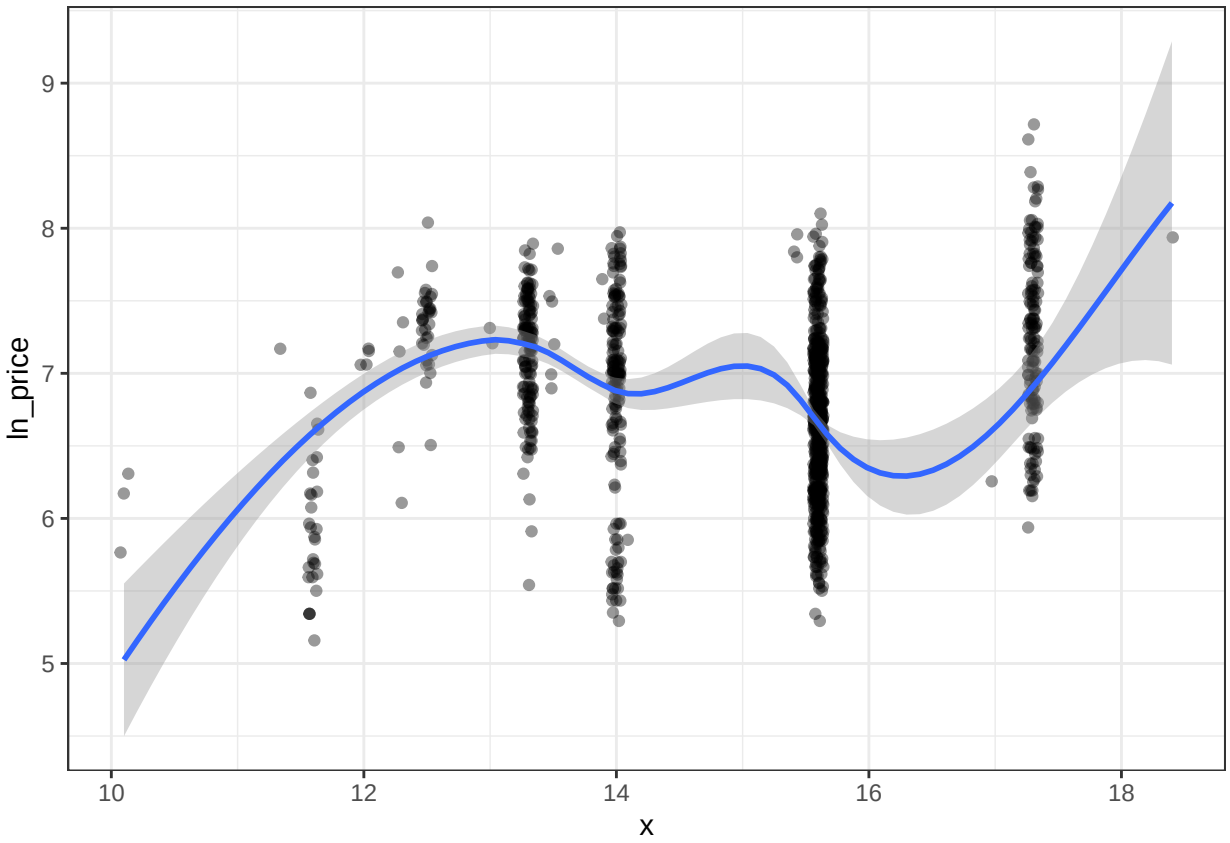
```
boxfun(laptop$company)
```



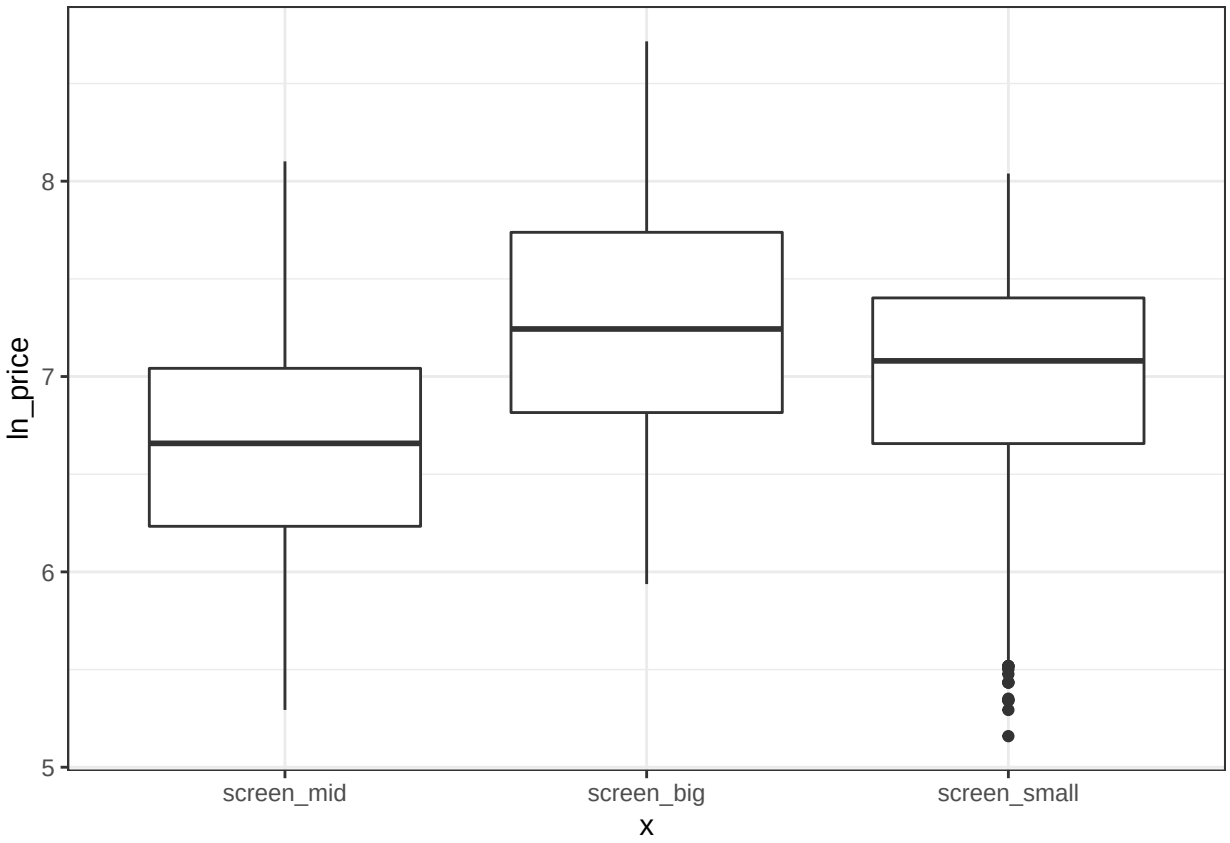

```
boxfun(laptop$type_name)
```



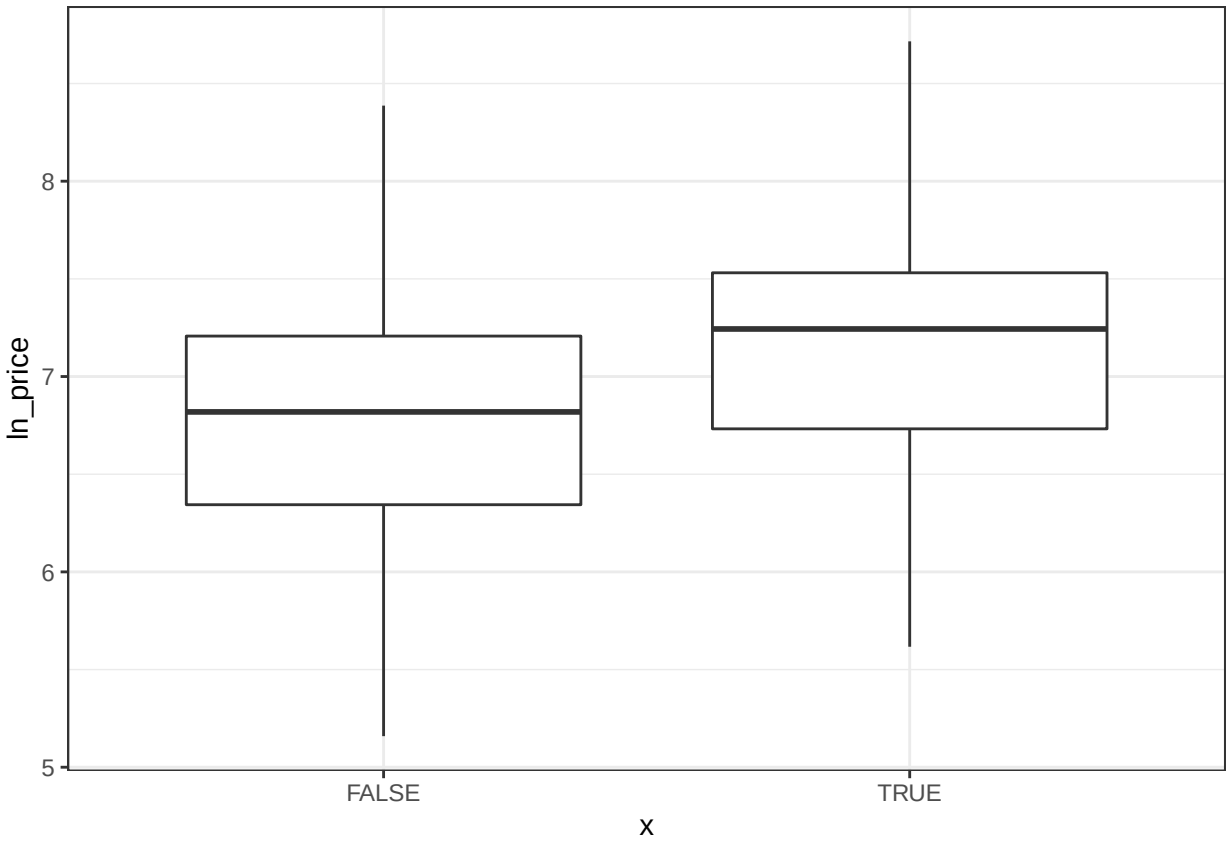

```
scatterfun(laptop$inches)
```



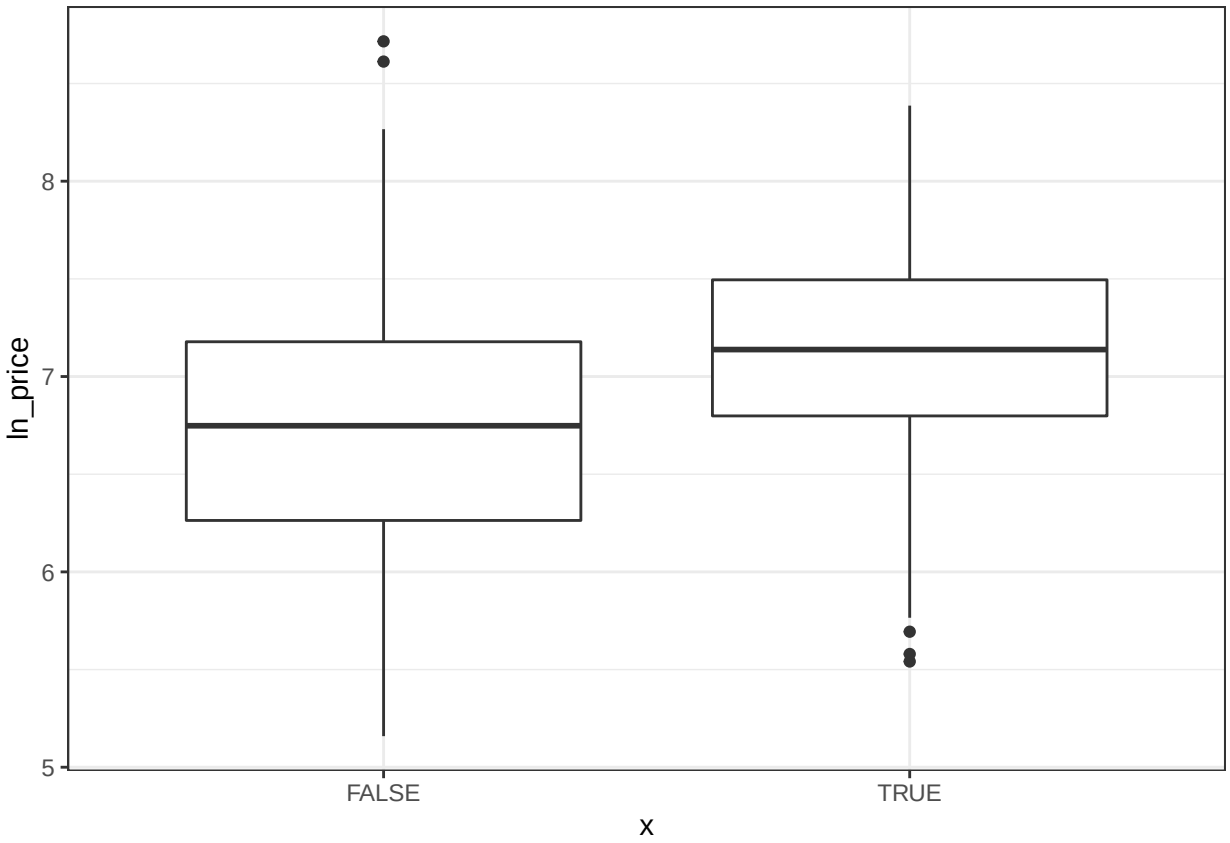

```
boxfun(laptop$screen_category)
```



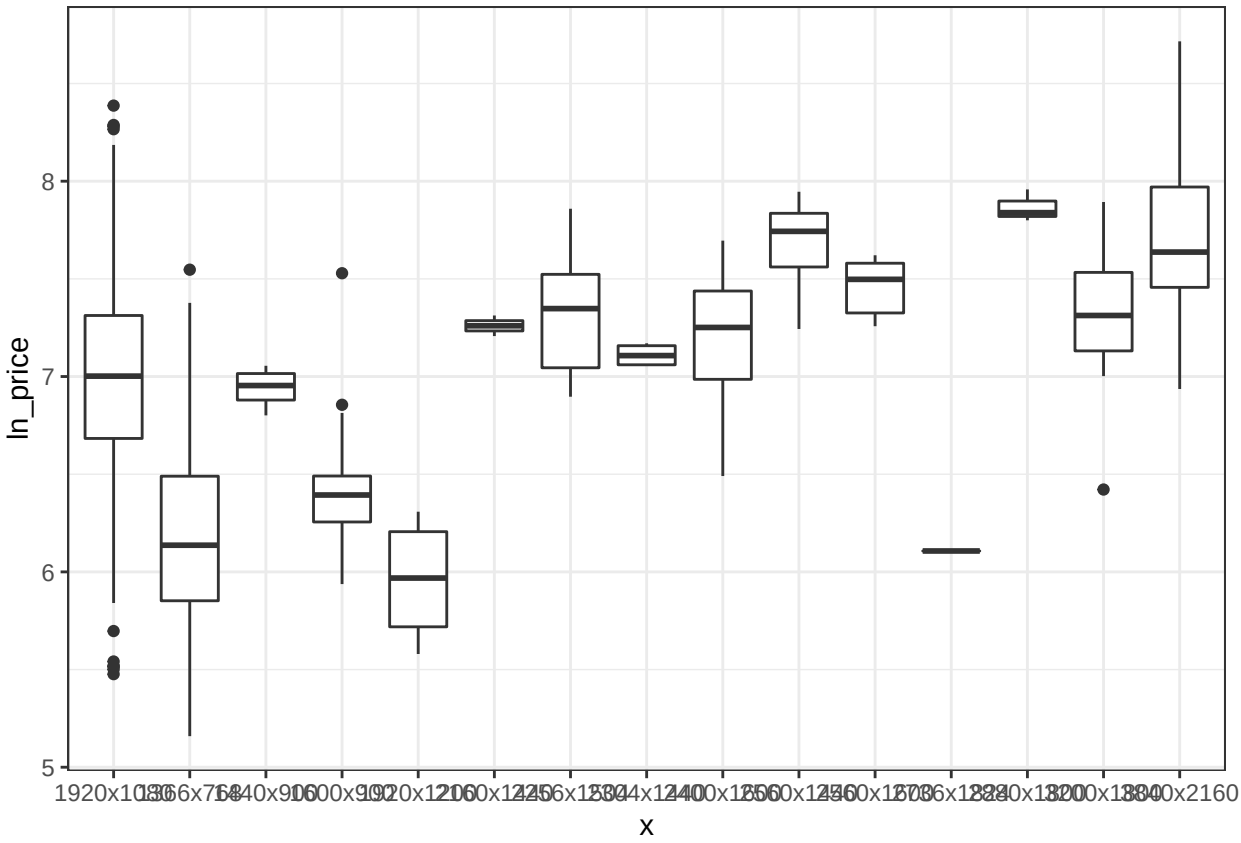

```
boxfun(laptop$touchscreen)
```



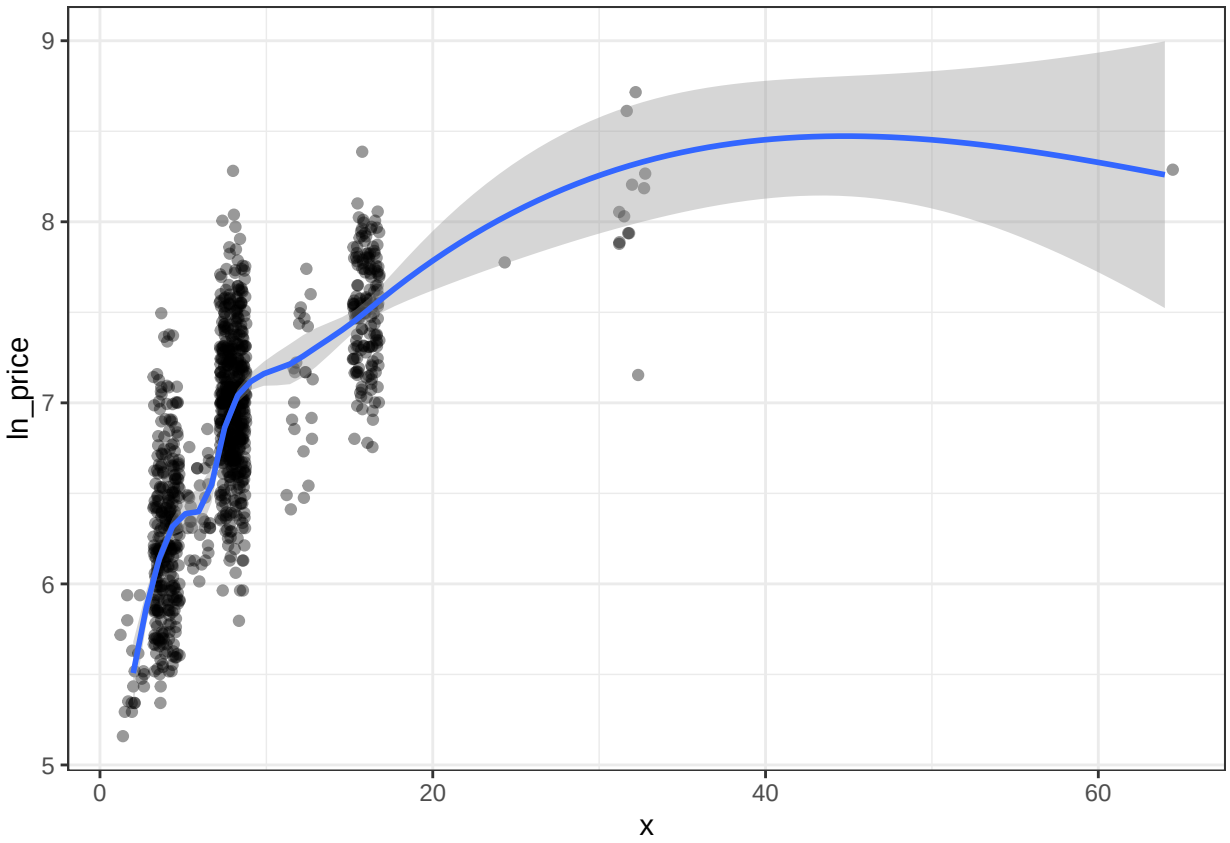

```
boxfun(laptop$ips)
```



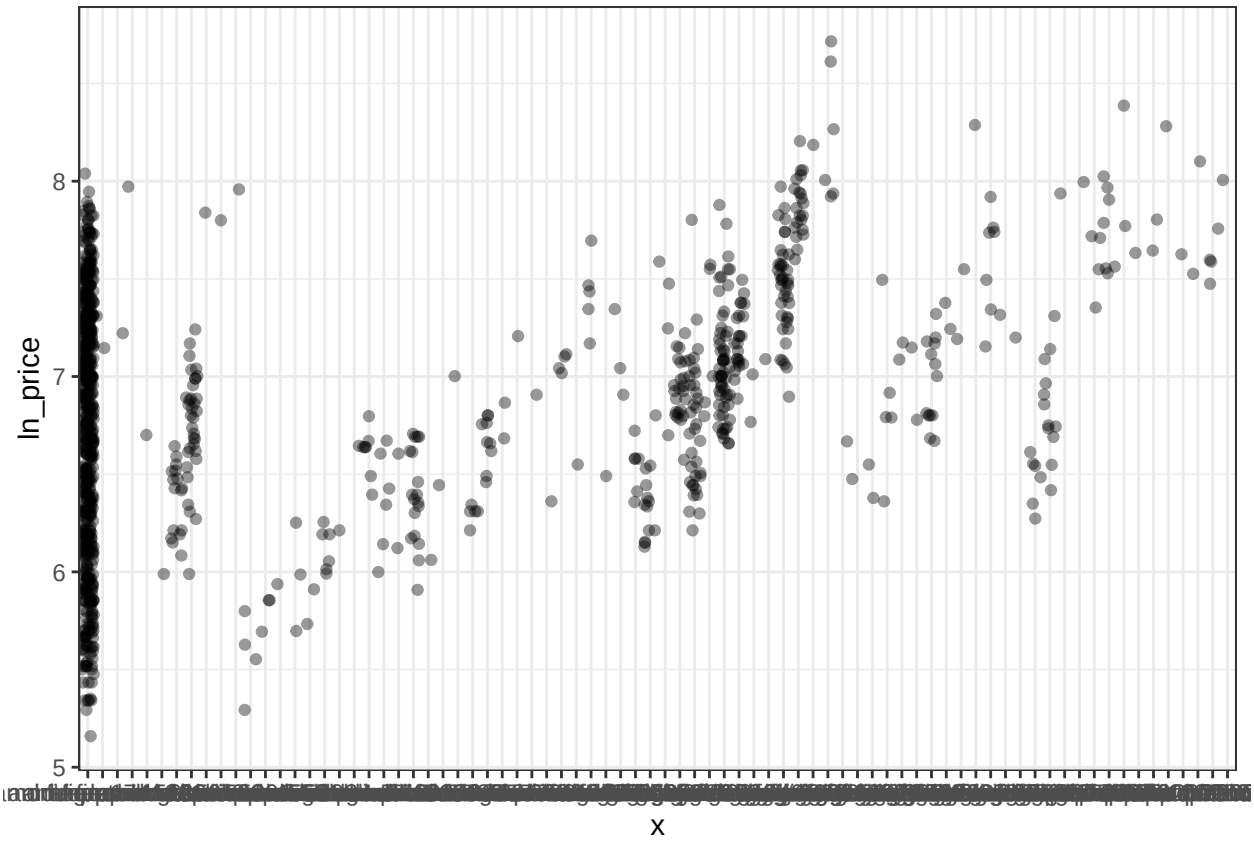

```
boxfun(laptop$resolution)
```



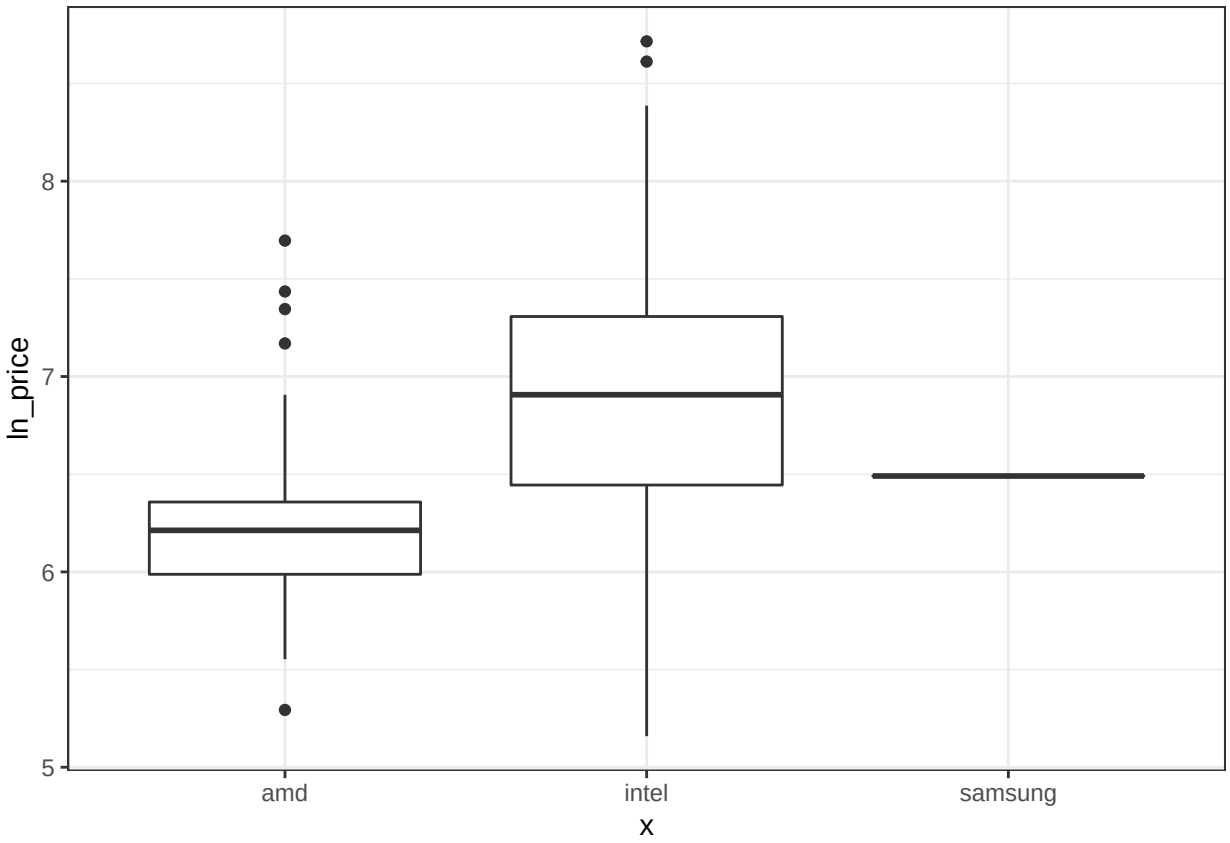

```
scatterfun(laptop$ram)
```



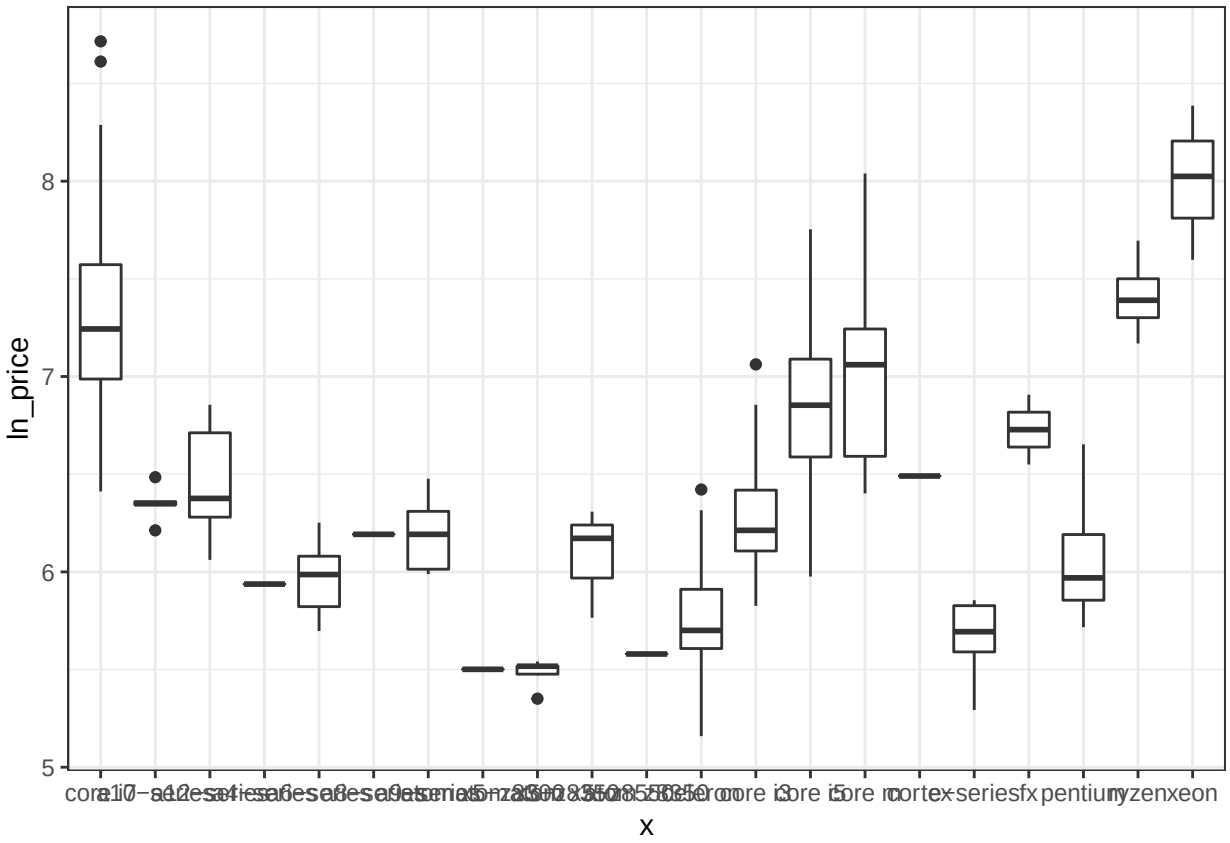

```
scatterfun(laptop$gpu_type)
```



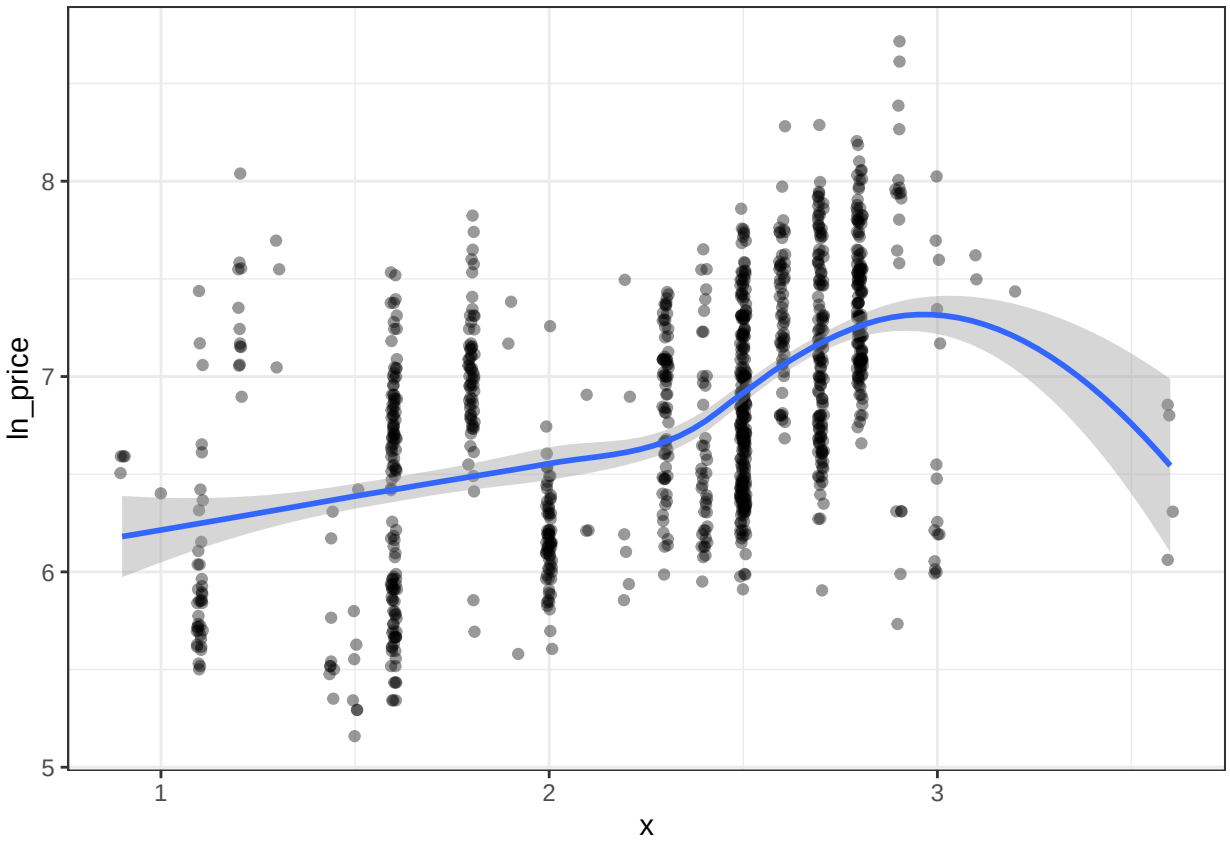

```
boxfun(laptop$cpu_manufac)
```



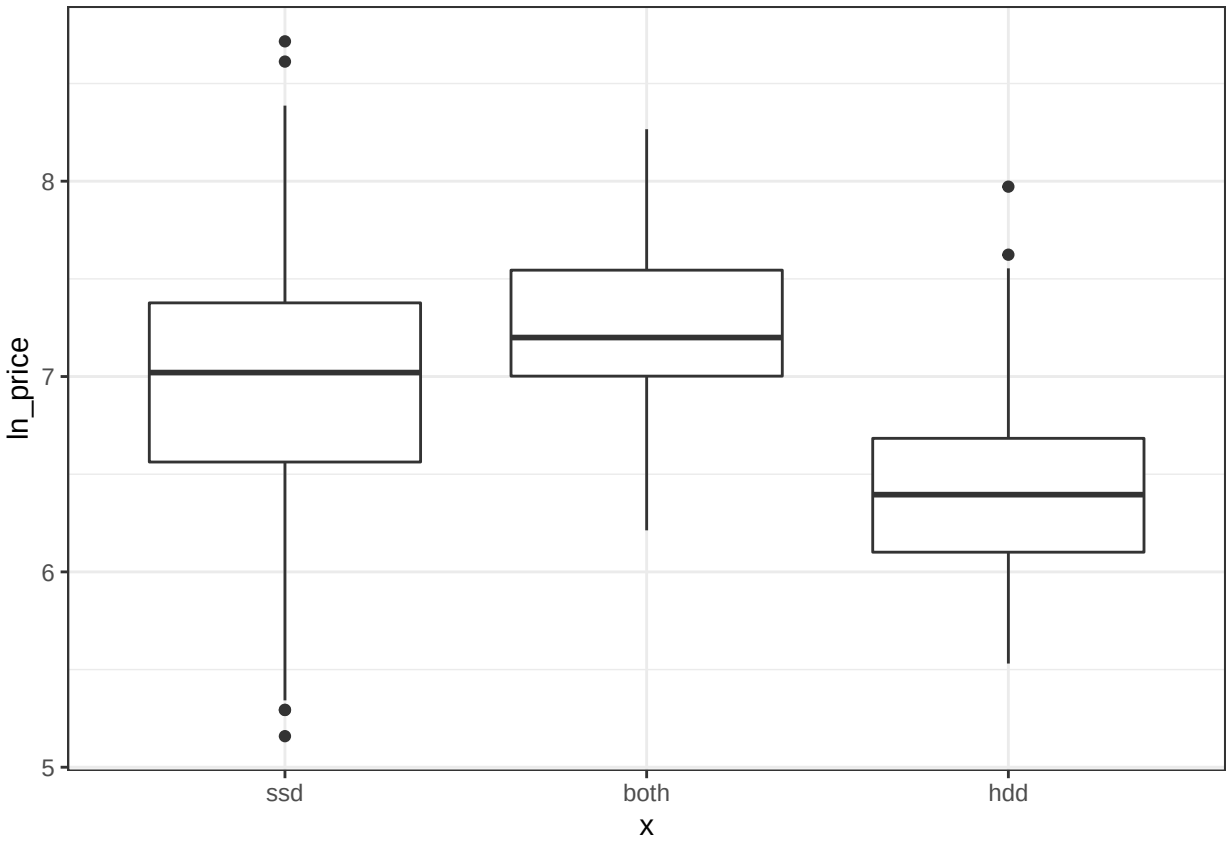

```
boxfun(laptop$cpu_model)
```



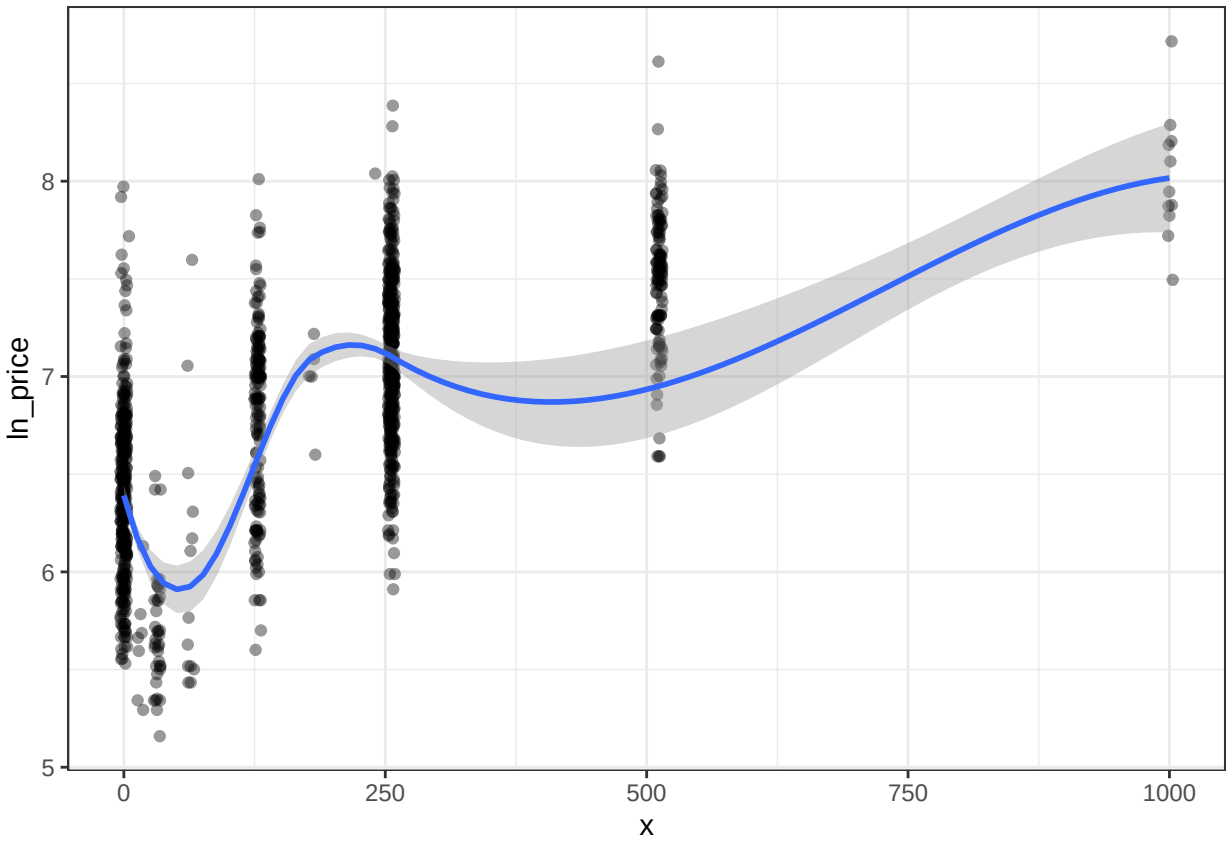

```
scatterfun(laptop$cpu_freq)
```



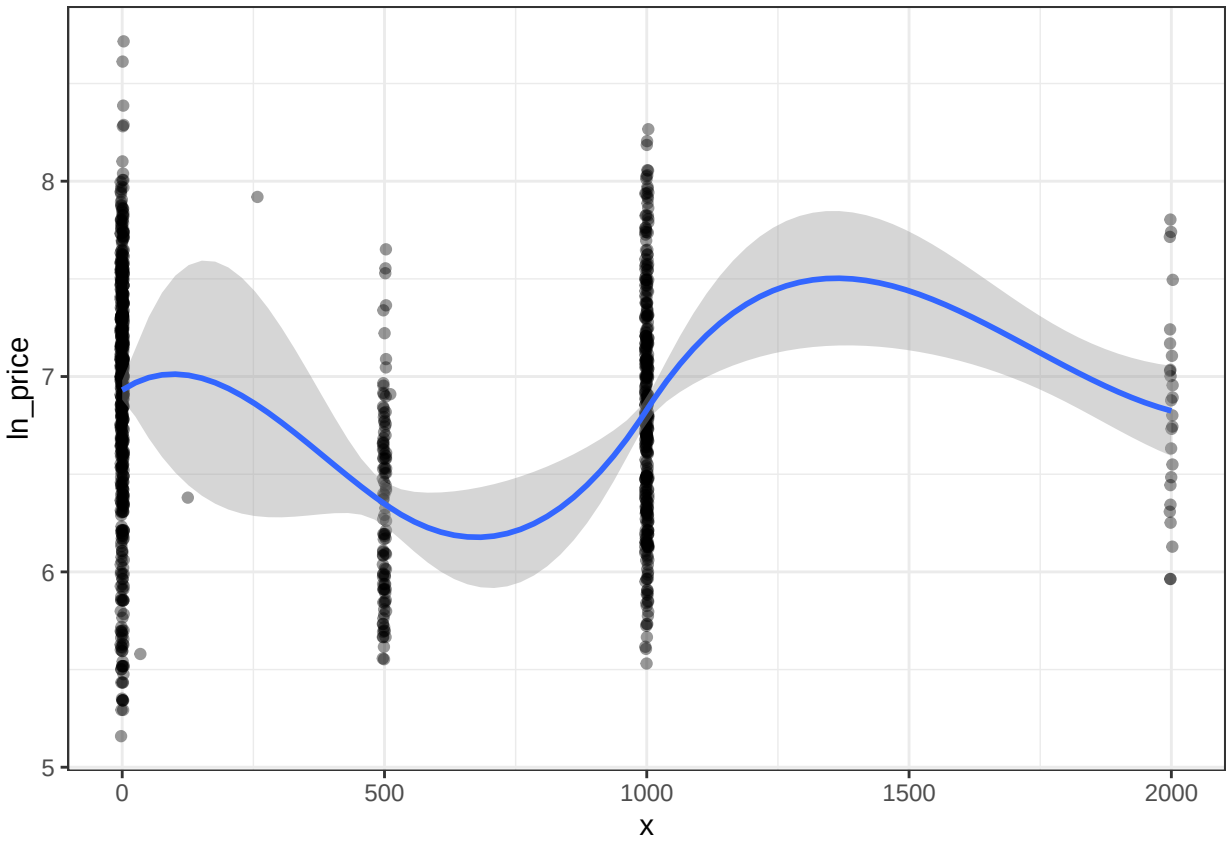

```
boxfun(laptop$memory_type)
```



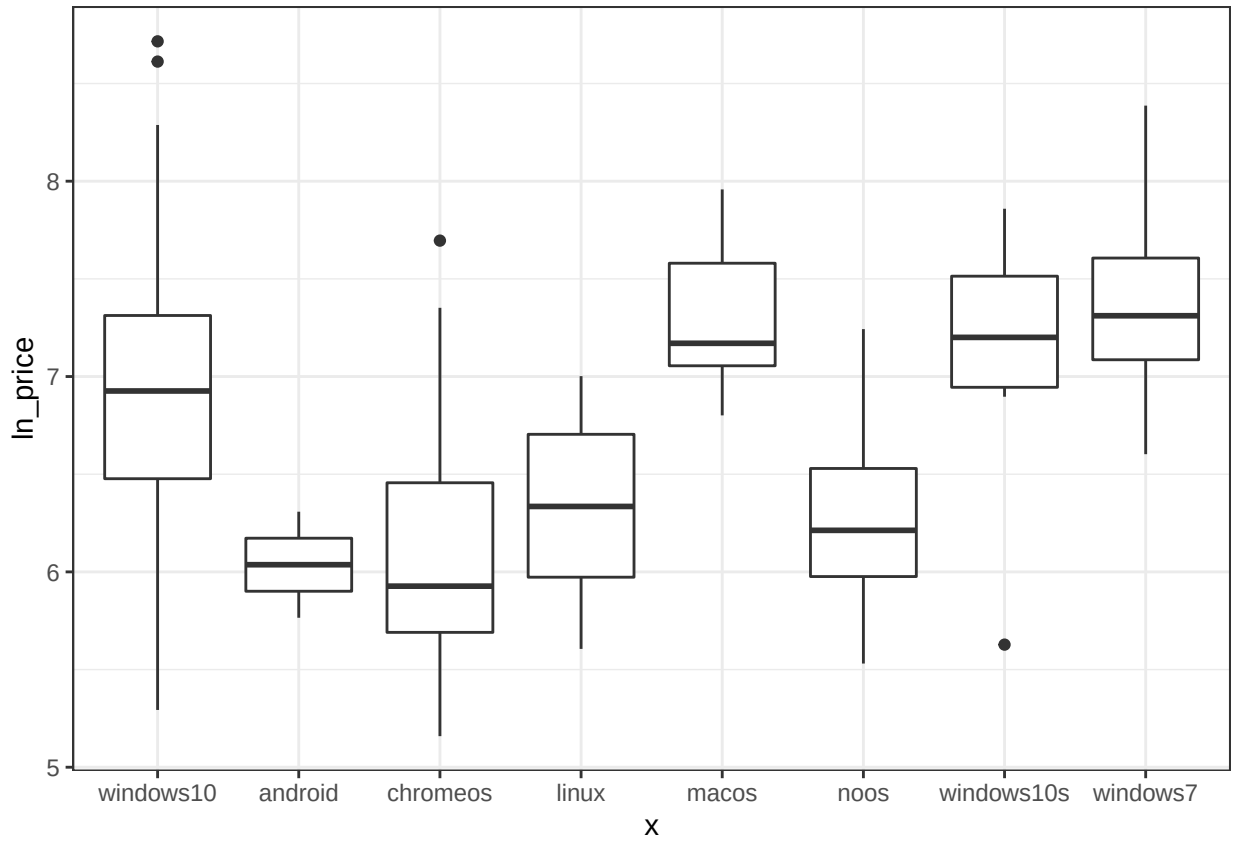

```
scatterfun(laptop$ssd_size)
```



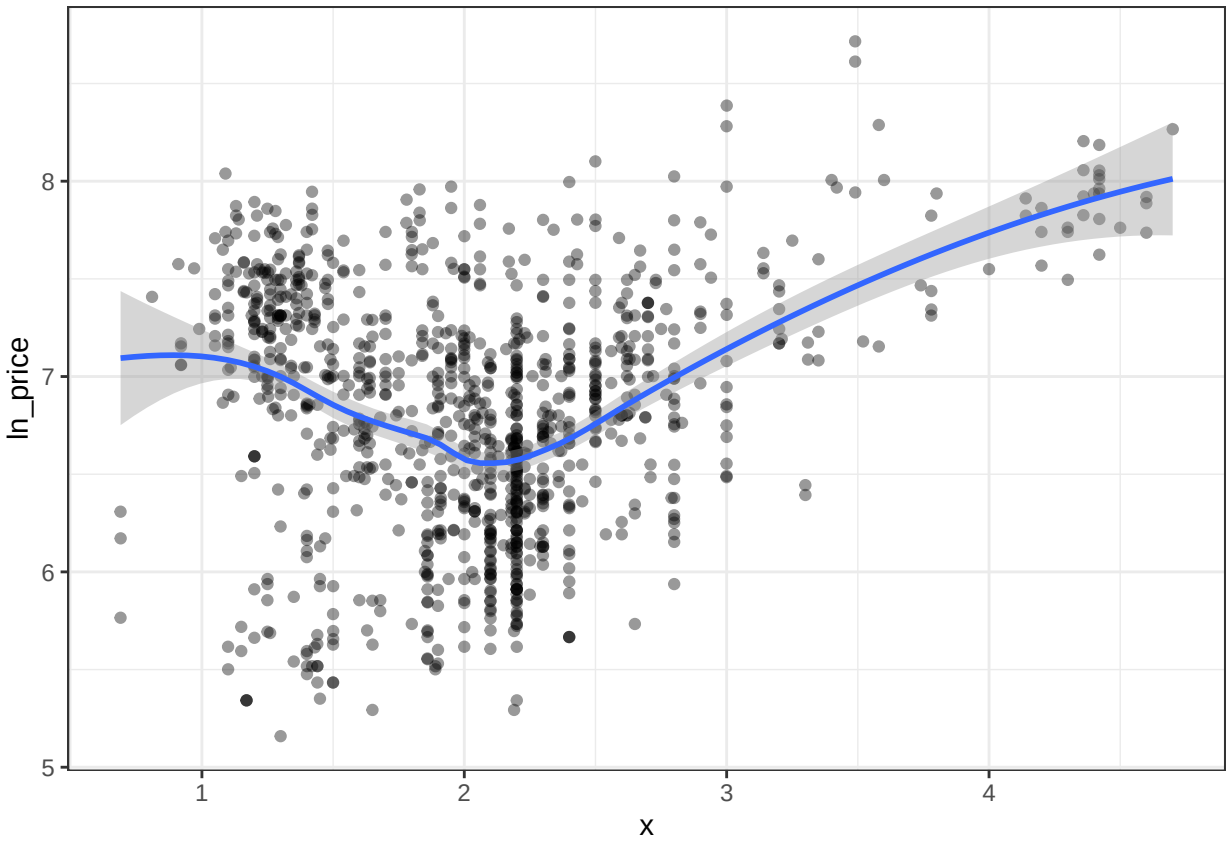

```
scatterfun(laptop$hdd_size)
```




```
boxfun(laptop$op_sys)
```




```
scatterfun(laptop$weight)
```



Model Assumptions

To see if my model meets all the assumptions of multiple linear regression I will investigate outliers and influential cases, multicollinearity, residuals and the independence of errors.

Outliers and influential cases

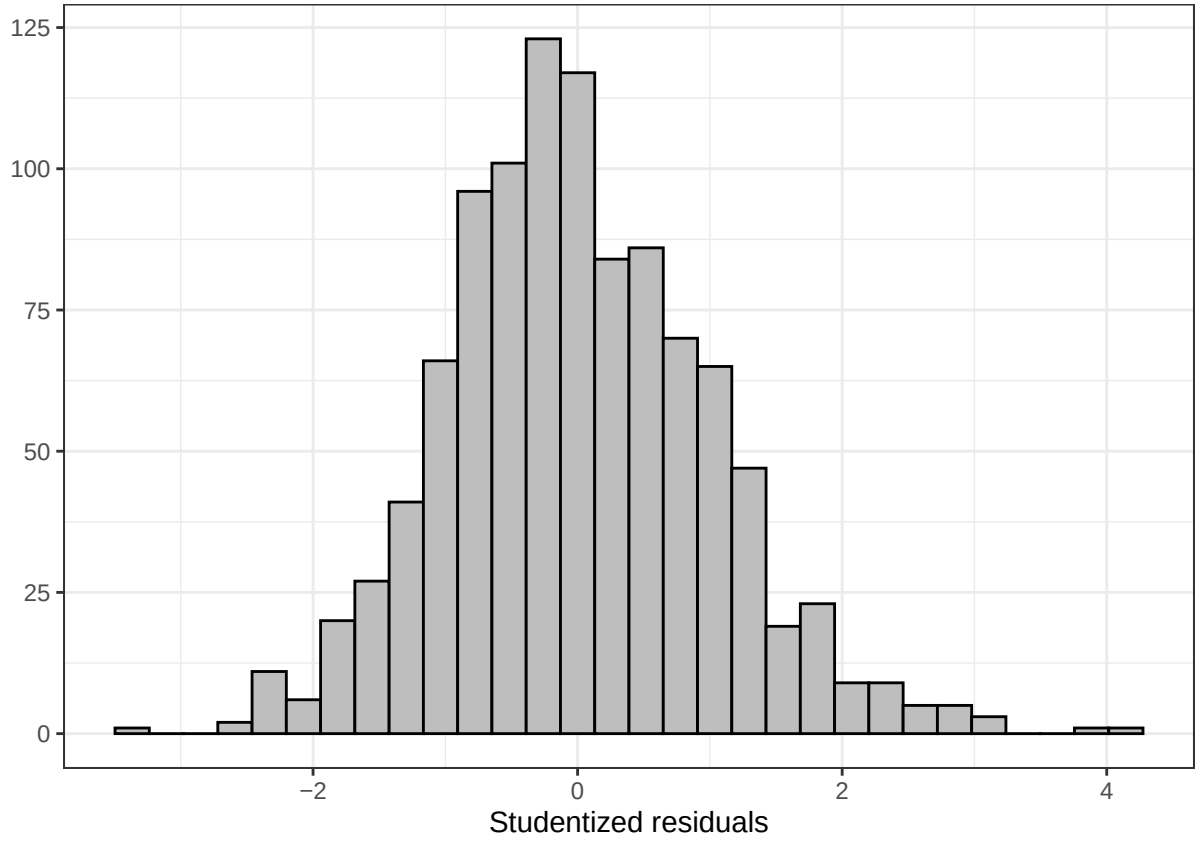
There are in total 52 outliers. This in itself is not necessarily a bad thing, it might be that they are very good deals. However it is worth taking a look at them to see if they exert undue influence on the model which could in turn distort the results. Investigating the cook's distance of the outliers we can find that there ar 0 observation above 1 which means they are not influential cases.

Multicollinearity

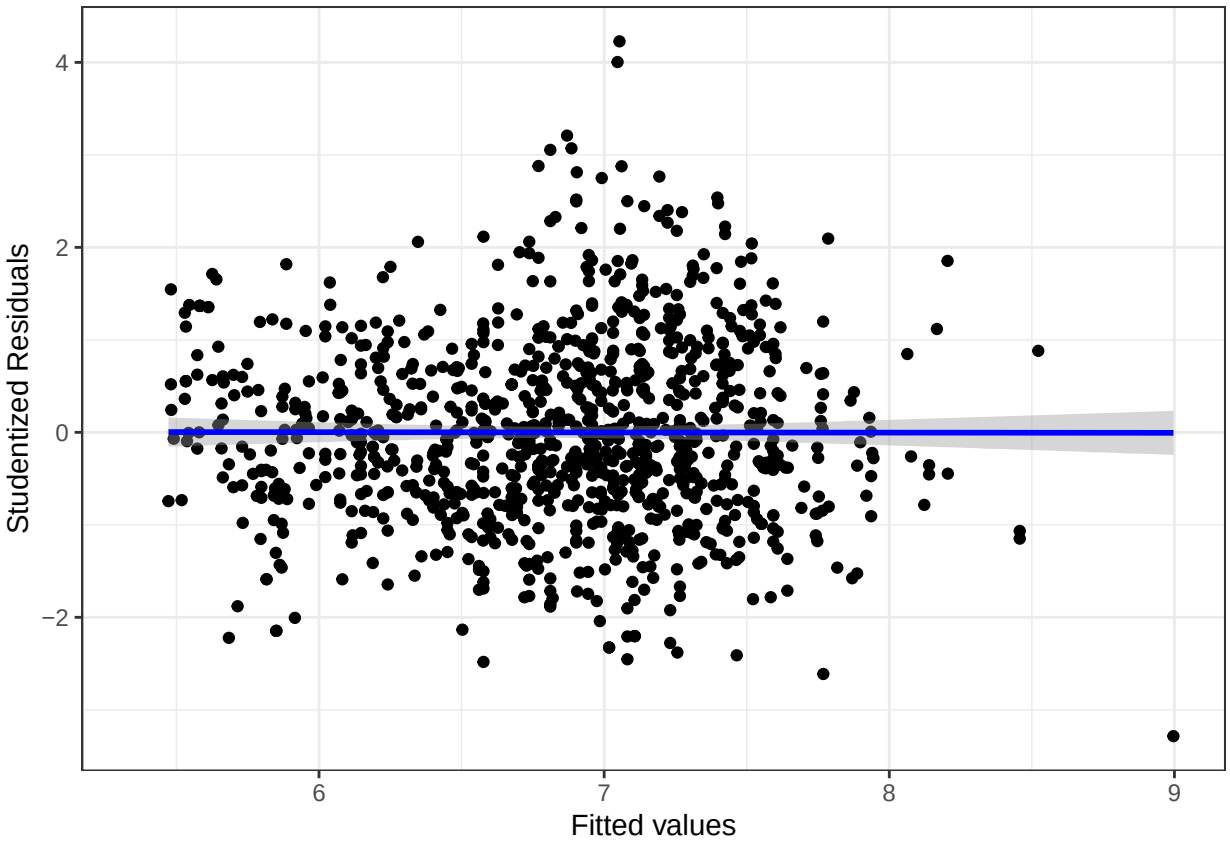
Upon inspecting the first model (reg3), I found that there is multicollinearity among the variables. So I had to investigate which variables cause this, and I found that there was multicollinearity among the operating system, resolution, gpu type and cpu model confounders. I experimented to see if there are some combination of these variables that can be included in the model, without multicollinearity but only the cpu model could be used. So due to this, I had to go back and exclude the aforementioned variables. But even with this modification, there still may be some multicollinearity among my variables, as the average VIF is above 1.

Multicollinearity is an issue, since it limits the R^2 of the model and increases the SE of the β coefficients. This makes it more difficult to interpret the results and the model parameters will vary a lot based on the sample provided.

Residuals - homoskedasticity and normality



We can see on this plot, that the residuals are normally distributed, because they follow a bell-shaped curve.



And on this plot we can see, that there is no heteroskedasticity or non-linearity in the data: points are scattered all over at random.

Independent errors

The Durbin-Watson test is 2.0345952272064, so the errors are independent in the sample.

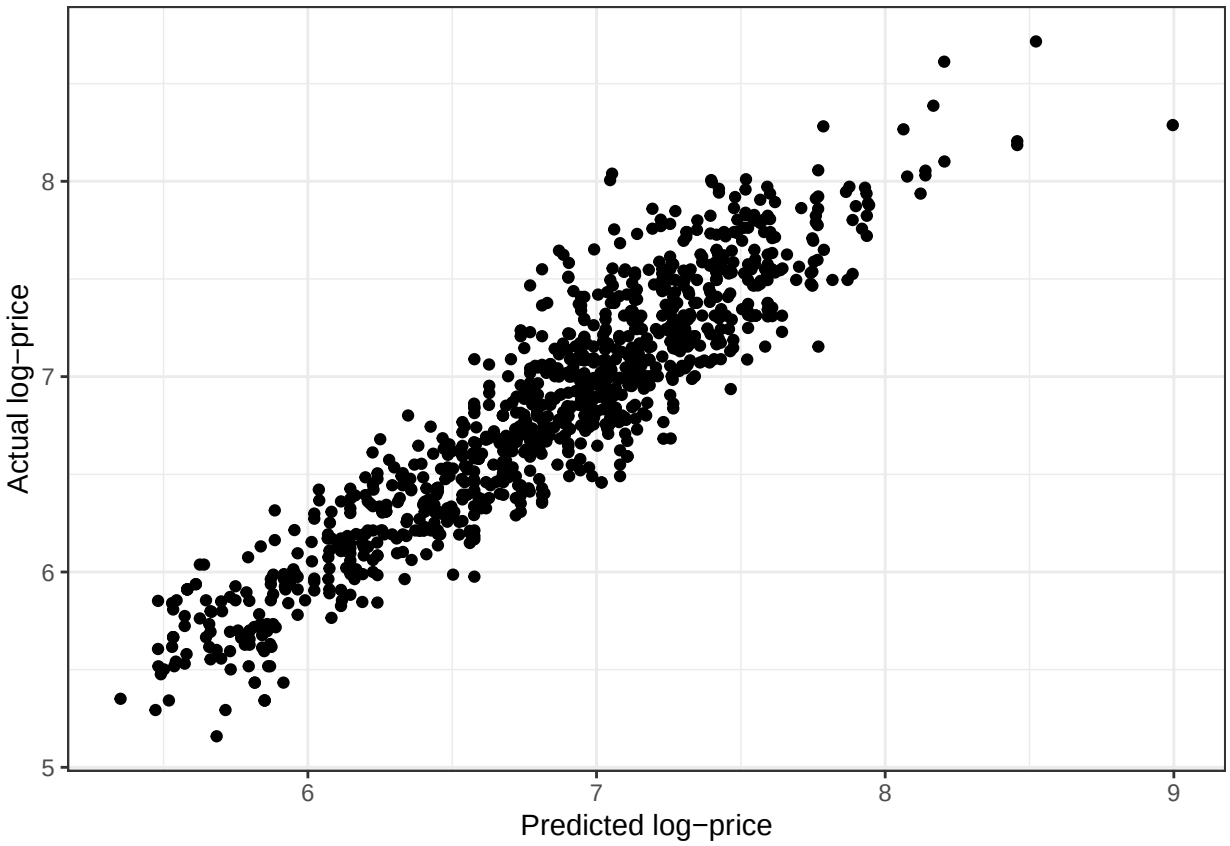
	Baseline - simple linear	Extended - Multiple regression
(Intercept)	6.76	6.64***
		(0.39)
companyAcer	−0.44	−0.09**
		(0.03)
companyApple	0.53	0.15*
		(0.07)
companyAsus	0.05	−0.04
		(0.03)
companyChuwi	−1.05	0.22
		(0.20)
companyDell	0.17	0.03
		(0.02)
companyFujitsu	−0.18	−0.11
		(0.17)
companyGoogle	0.64	0.16
		(0.15)
companyHP	0.06	0.07**
		(0.02)
companyHuawei	0.50	0.01
		(0.18)
companyLG	0.93	0.43*
		(0.18)
companyMediacom	−1.10	0.08
		(0.16)
companyMicrosoft	0.57	0.18
		(0.10)
companyMSI	0.62	0.02
		(0.05)
companyRazer	1.17	0.23*
		(0.10)
companySamsung	−0.03	0.04
		(0.14)
companyToshiba	0.30	0.16***
		(0.04)
companyVero	−1.41	−0.06
		(0.31)
companyXiaomi	0.18	−0.03
		(0.14)
type_name2 in 1 Convertible		0.10**
		(0.03)
type_nameGaming		0.21***
		(0.03)
type_nameNetbook		−0.04
		(0.08)
type_nameUltrabook		0.14***
		(0.03)
type_nameWorkstation		0.66***
		(0.06)
inches		−0.01
		(0.02)
ram		0.02***
		(0.00)
screen_categoryscreen_big		0.19***
		(0.05)
screen_categoryscreen_small		0.21***
		(0.05)
cpu_modela10-series		−0.38***
		(0.10)
cpu_modela12-series		−0.32**

	Extended - Multiple regression
(Intercept)	0.00***
	(0.39)
companyAcer	−0.04**
	(0.03)
companyApple	0.03*
	(0.07)
companyAsus	−0.02
	(0.03)
companyChuwi	0.02
	(0.20)
companyDell	0.02
	(0.02)
companyFujitsu	−0.01
	(0.17)
companyGoogle	0.01
	(0.15)
companyHP	0.05**
	(0.02)
companyHuawei	0.00
	(0.18)
companyLG	0.03*
	(0.18)
companyMediacom	0.01
	(0.16)
companyMicrosoft	0.02
	(0.10)
companyMSI	0.01
	(0.05)
companyRazer	0.03*
	(0.10)
companySamsung	0.00
	(0.14)
companyToshiba	0.05***
	(0.04)
companyVero	−0.00
	(0.31)
companyXiaomi	−0.00
	(0.14)
type_name2 in 1 Convertible	0.05**
	(0.03)
type_nameGaming	0.12***
	(0.03)
type_nameNetbook	−0.01
	(0.08)
type_nameUltrabook	0.08***
	(0.03)
type_nameWorkstation	0.16***
	(0.06)
inches	−0.02
	(0.02)
ram	0.19***
	(0.00)
screen_categoryscreen_big	0.10***
	(0.05)
screen_categoryscreen_small	0.16***
	(0.05)
cpu_modela10-series	−0.05***
	(0.10)
cpu_modela12-series	−0.04**

Detailed model comparison

Standardized residuals

Y - Y hat plot



We can see on this plot, that the predicted values fit the actual values fairly well.

Compare Train and Test model

	Training model	Testing model
(Intercept)	6.64***	6.29***
	(0.39)	(0.87)
companyAcer	−0.09**	−0.08
	(0.03)	(0.07)
companyApple	0.15*	0.09
	(0.07)	(0.14)
companyAsus	−0.04	−0.03
	(0.03)	(0.06)
companyChuwi	0.22	
	(0.20)	
companyDell	0.03	0.10*
	(0.02)	(0.05)
companyFujitsu	−0.11	0.07
	(0.17)	(0.24)
companyGoogle	0.16	
	(0.15)	
companyHP	0.07**	0.07
	(0.02)	(0.05)
companyHuawei	0.01	
	(0.18)	
companyLG	0.43*	0.10
	(0.18)	(0.24)
companyMediacom	0.08	−0.30
	(0.16)	(0.27)
companyMicrosoft	0.18	
	(0.10)	
companyMSI	0.02	0.16
	(0.05)	(0.10)
companyRazer	0.23*	0.21
	(0.10)	(0.27)
companySamsung	0.04	0.18
	(0.14)	(0.12)
companyToshiba	0.16***	0.10
	(0.04)	(0.09)
companyVero	−0.06	−0.37*
	(0.31)	(0.18)
companyXiaomi	−0.03	0.09
	(0.14)	(0.24)
type_name2 in 1 Convertible	0.10**	0.16**
	(0.03)	(0.06)
type_nameGaming	0.21***	0.16
	(0.03)	(0.08)
type_nameNetbook	−0.04	−0.02
	(0.08)	(0.16)
type_nameUltrabook	0.14***	0.23***
	(0.03)	(0.06)
type_nameWorkstation	0.66***	0.54***
	(0.06)	(0.11)
inches	−0.01	0.01
	(0.02)	(0.05)
ram	0.02***	0.03***
	(0.00)	(0.00)
screen_categoryscreen_big	0.19***	0.05
	(0.05)	(0.11)
screen_categoryscreen_small	0.21***	0.18
	(0.05)	(0.11)
cpu_modela10-series	−0.38***	
	(0.10)	
cpu_modela12-series	−0.32**	−0.38*