Stanford GSB: OIT 276 - R Visualisation cheatsheet

Malo Marrec (malo@stanford.edu)

2/13/2017

Data Visualisation Basics

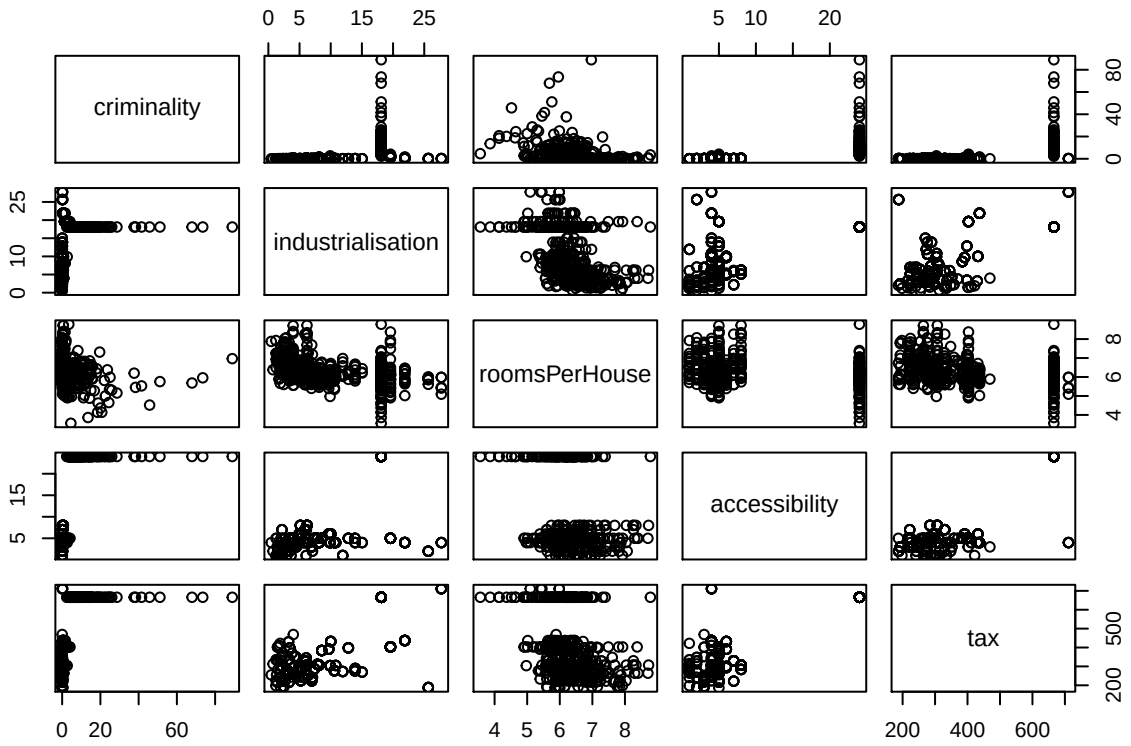
This cheatsheet present some methods to solve common data visualization methods. Don't hesitate to shoot me an email for improvements / clarification / questions malo@stanford.edu.

We'll look at how to visualize some Boston criminality data included in the package MASS.

What a mess! Plotting everything

One of the first things you might want to do is plotting everything (all the covariates and the output) . That will give you a feel of which covariate is important, which things are correlated, ...

```
pairs(criminality)
```

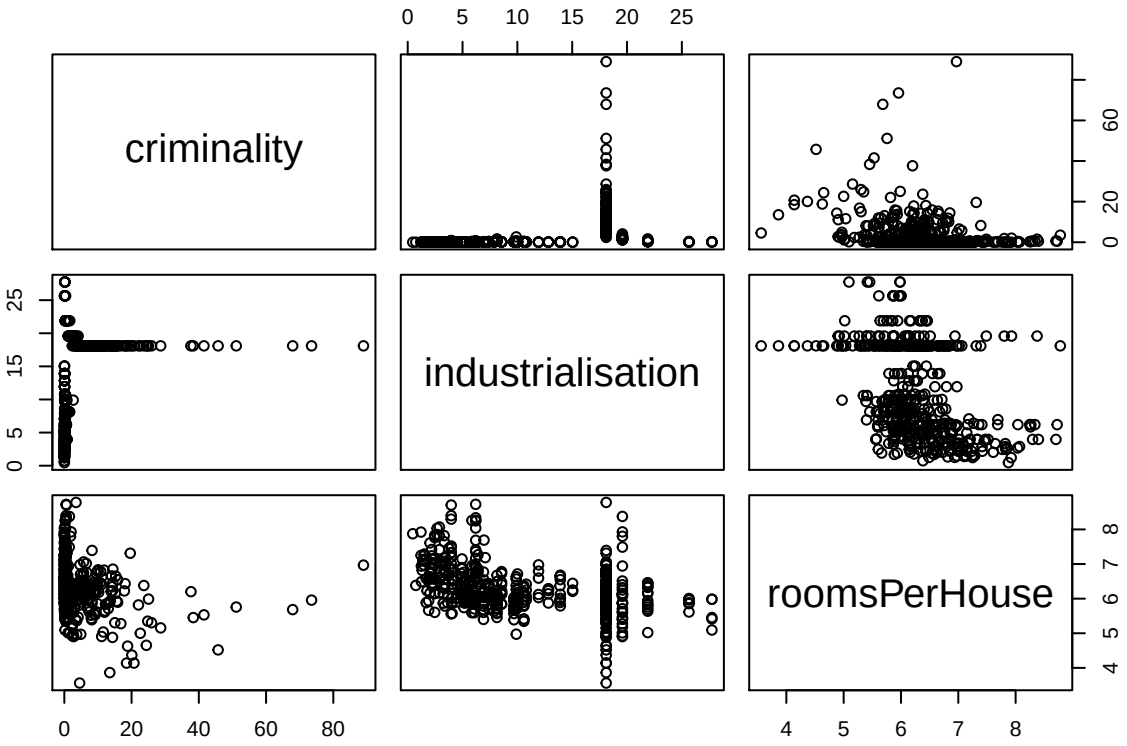


Note that the first row is the plots of criminality vs the other coavariates (eg first plot criminality vs industrialisation), whilst the first column is the plot of the other covariates vs criminality.

Partial pairs

Sometimes the number of covariates is so big you cannot do this (you will get an error if you try), and you need to select a subset of the covariates:

```
pairs(~criminality + industrialisation + roomsPerHouse ,data=criminality)
```



Installing a library

A lot of the wonderful things that R can do use libraries. Using a library takes two lines of code and gives you access to a wide range of tools. It is going to help you answer a lot of the problems encountered in class, as: * How can I visualize correlations? * There are too many points on this graph, I can't see anything!

And some problems you might come across during the projects: * Sounds like there are lots of missing values. How can I visualize that?

Here we install the library ggplot2., an incredibly useful visualisation package. Afterwards we give some code for basic visualizations.

```
# Run only once (per computer)
# Note the quotation marks around the name
install.packages("ggplot2")

# Run each time you want to use the library
# Note the absence of quotation marks
library(ggplot2)
```

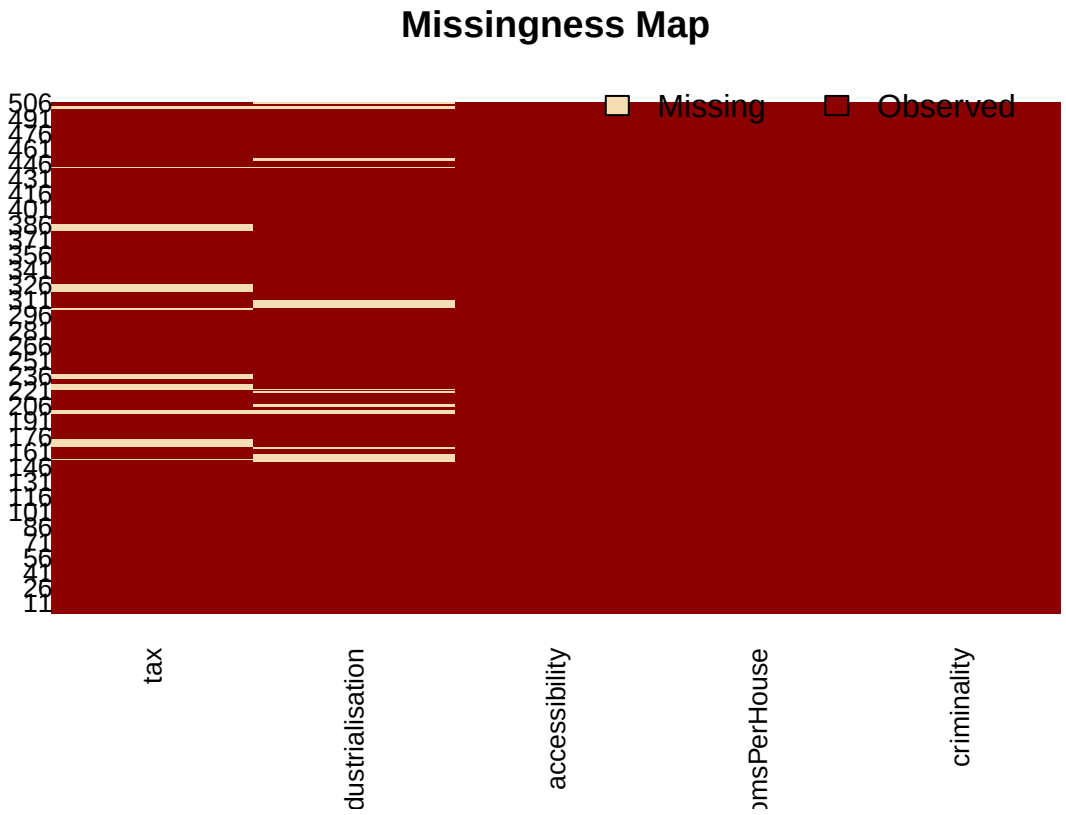
Useful Visualizations

Missing values

Here the dataset is perfectly clean, so I artifically created a "dirty" dataset.

```
#run once
#install.packages("Amelia")
library(Amelia)
```

```
missmap(criminality2)
```

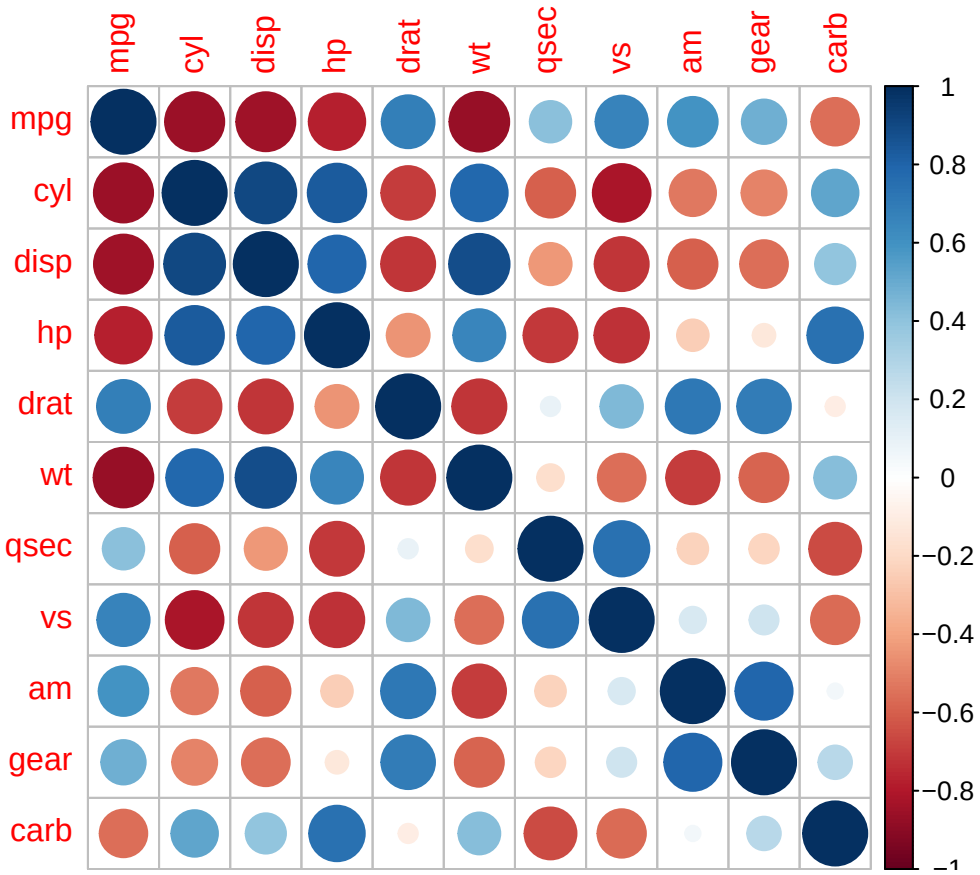


Correlations

```
typeof(mtcars$mpg)
```

```
## [1] "double"
```

```
#install.packages("corrplot")
library(corrplot)
correlations <- cor(mtcars)
corrplot(correlations,method="circle")
```



Once More Unto The Breach: Nice plots using ggplot2

(advanced but copy pasting works!)

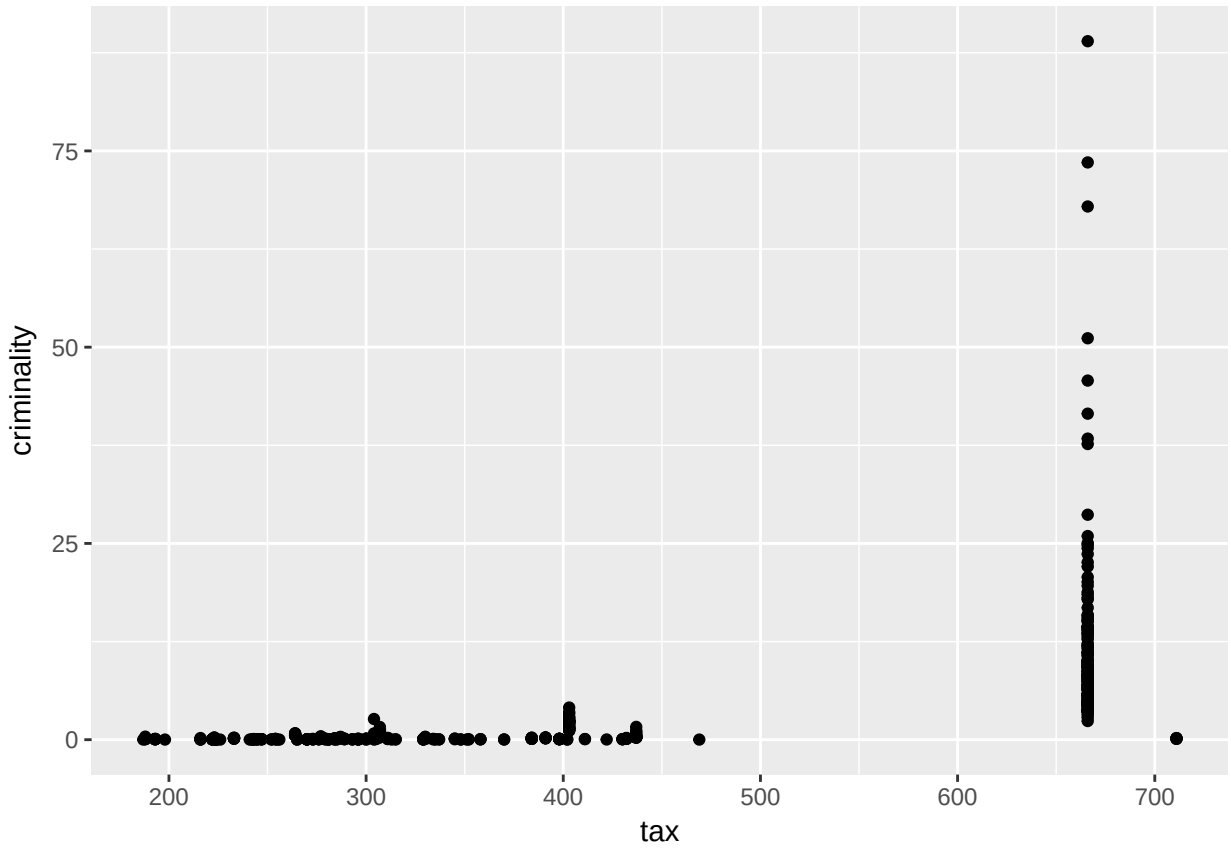
ggplot has a somehow counterintuitive syntax at first, but produces outstanding clean graphs. Note that each instruction is separated from the others y a "+", indicating "add this element to the graph". The "+" *needs* to be at the end of a line, not at the beginning of a new line. If you don't want to bother with the technicalities, just copy paste the code samples below.

Basic scatterplot

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

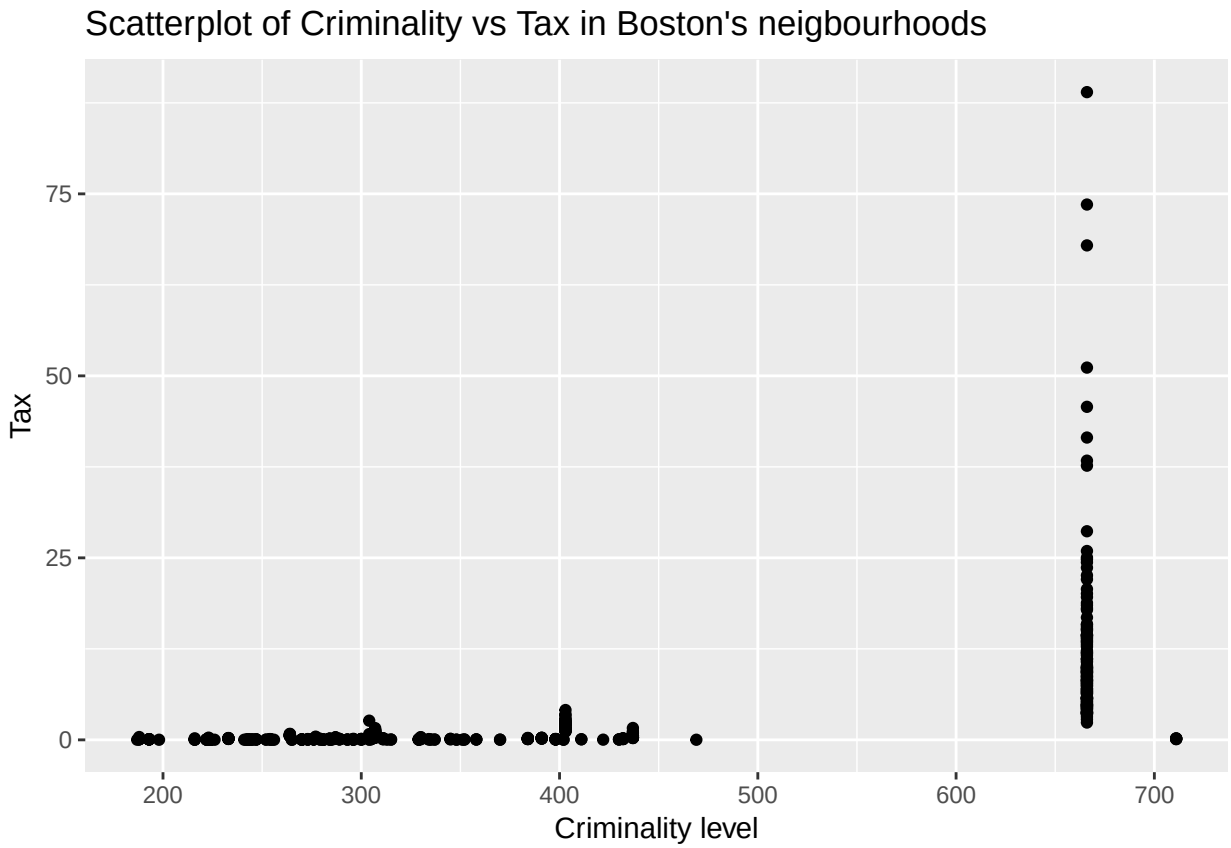
```
#replace "criminality" with your dataset, and y and x by the name of the covariates you want to plot.
ggplot(data = criminality) +
  geom_point(aes(x=tax, y=criminality))
```



Adding title and labels

To personalize the axis labels and add a title, use the following code. You can use the xlab, ylab an ggtitle for any kind of graphs, including the histogram in the next section.

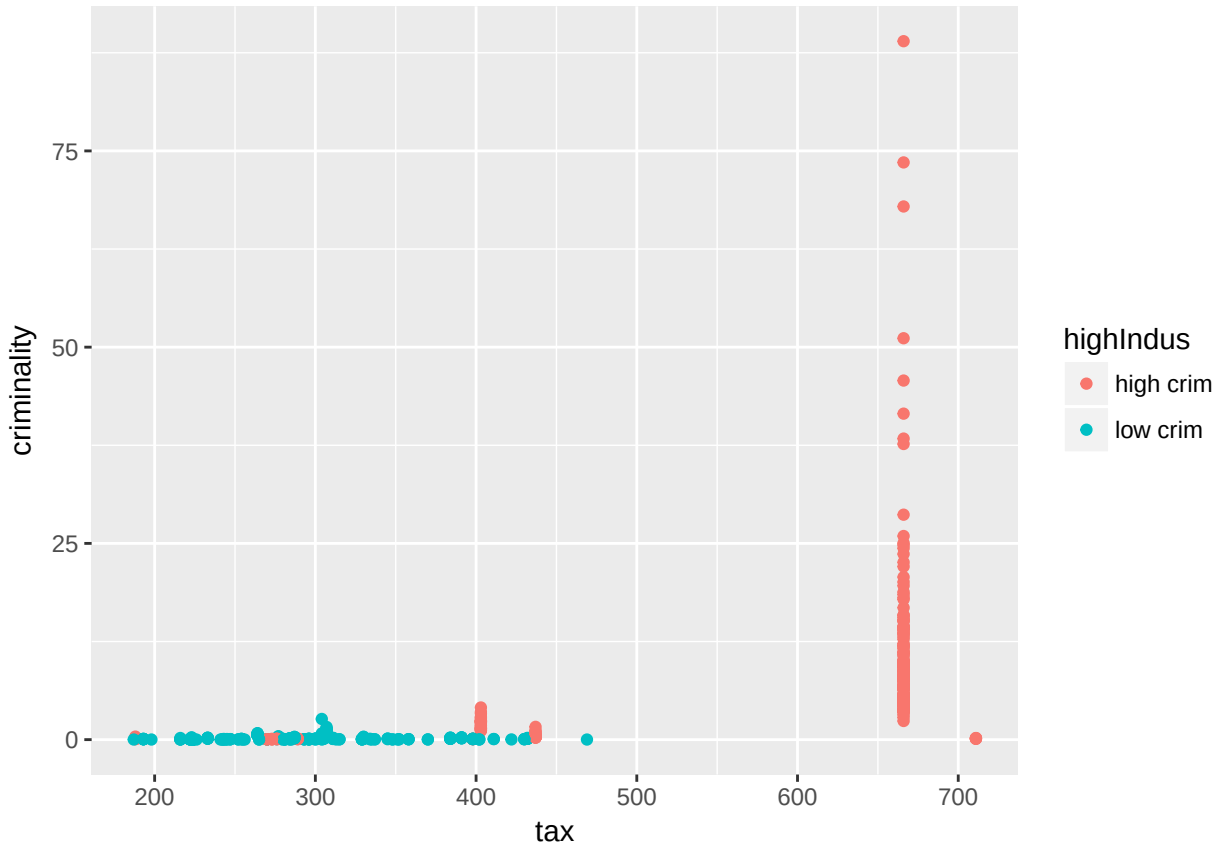
```
ggplot(data = criminality) +
  geom_point(aes(x=tax, y=criminality)) +
  ggtitle("Scatterplot of Criminality vs Tax in Boston's neigbourhoods") +
  xlab("Criminality level") +
  ylab("Tax")
```



Playing with more than two variables

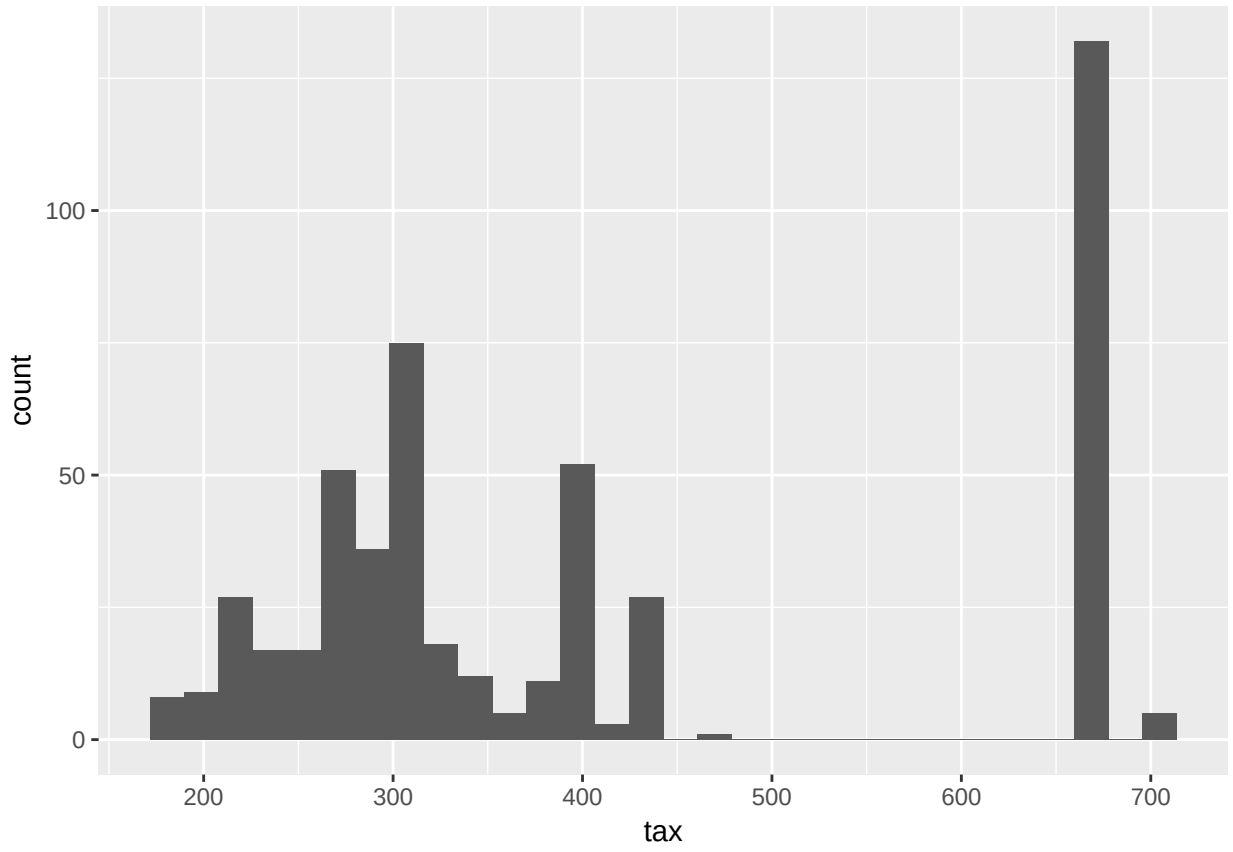
Now assume I want to visualize how the number of rooms impact the tax and criminality. I can represent each of the point in a different color depending on the number of room. That works for any variable. You might want to transform them to factor first to get a nicer result.

```
criminality$highIndus <- as.factor(ifelse(criminality$industrialisation > mean(criminality$industrialis
ggplot(data = criminality) +
  geom_point(aes(x=tax, y=criminality,color = highIndus))
```



Histograms

```
ggplot(data = criminality) +
  geom_histogram(aes(x=tax))
```


count
100
50
0
200
300
400
500
600
700
tax