Exploratory data analysis

Dominik Klepl

11/26/2019

```
pacman::p_load(ggplot2, ggthemes, tidyr, gridExtra, extrafont, patchwork)
```

Load the dataset

```
data = read.csv("data/x_y.csv", header = F)
```

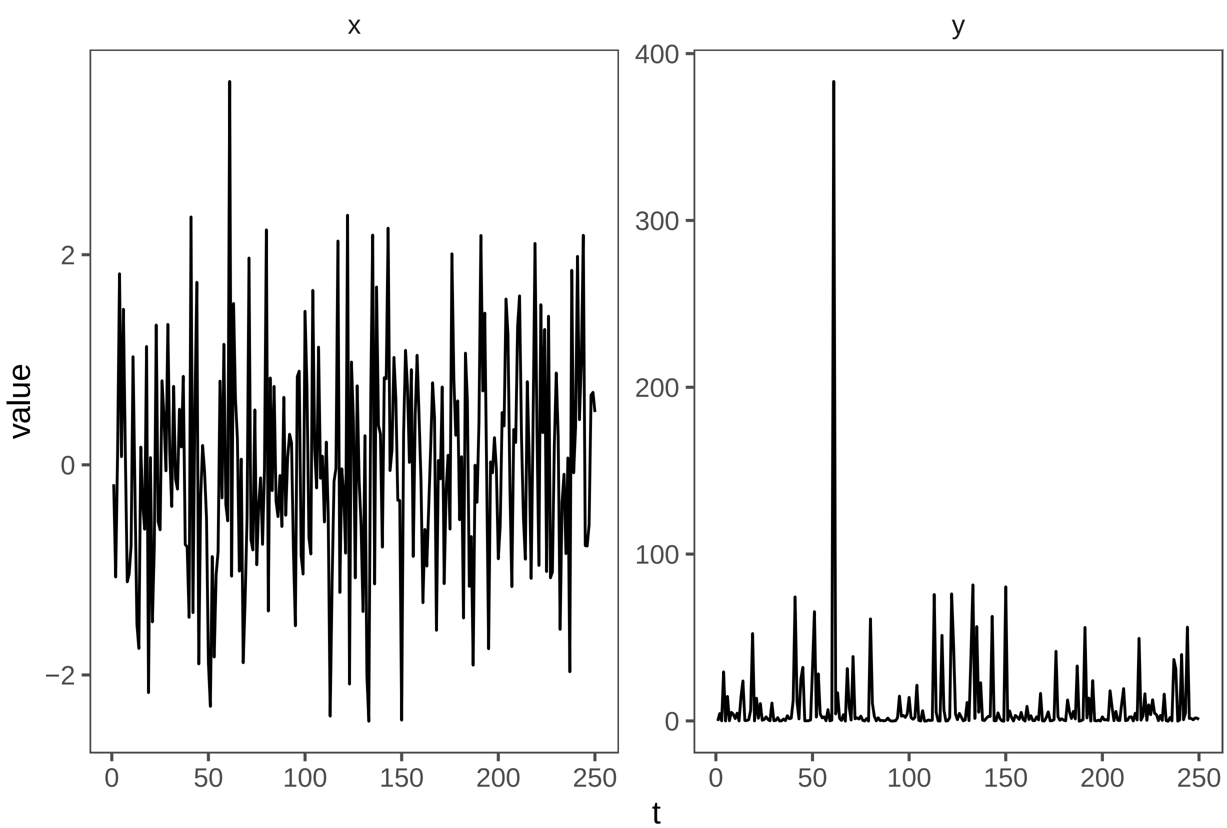
Rename the column names to x and y

```
colnames(data) = c("x", "y")
```

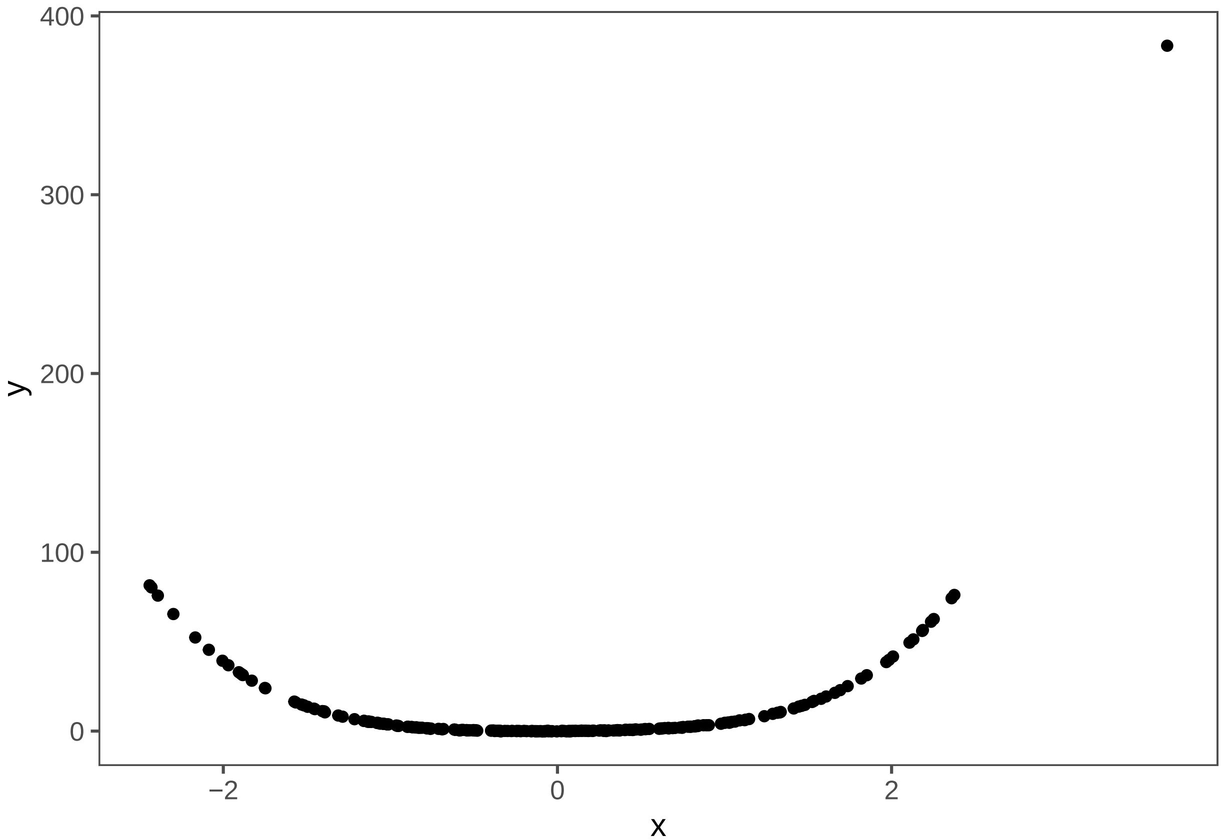
Add time variable to preserve the time-series structure

Relationship between x and y

We start with inspecting the input/output variables by plotting them. First on the same axis simply as two time-series signals.

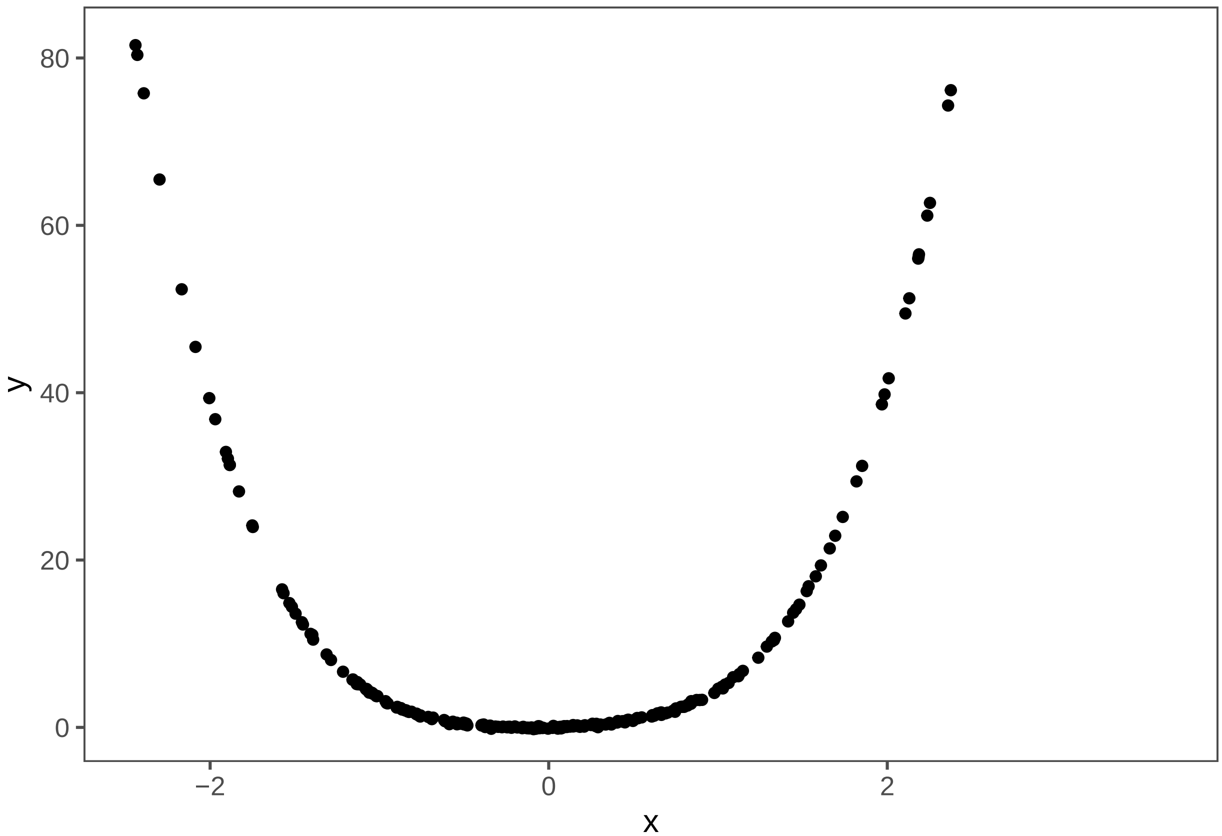


Now we also plot the signals each other.



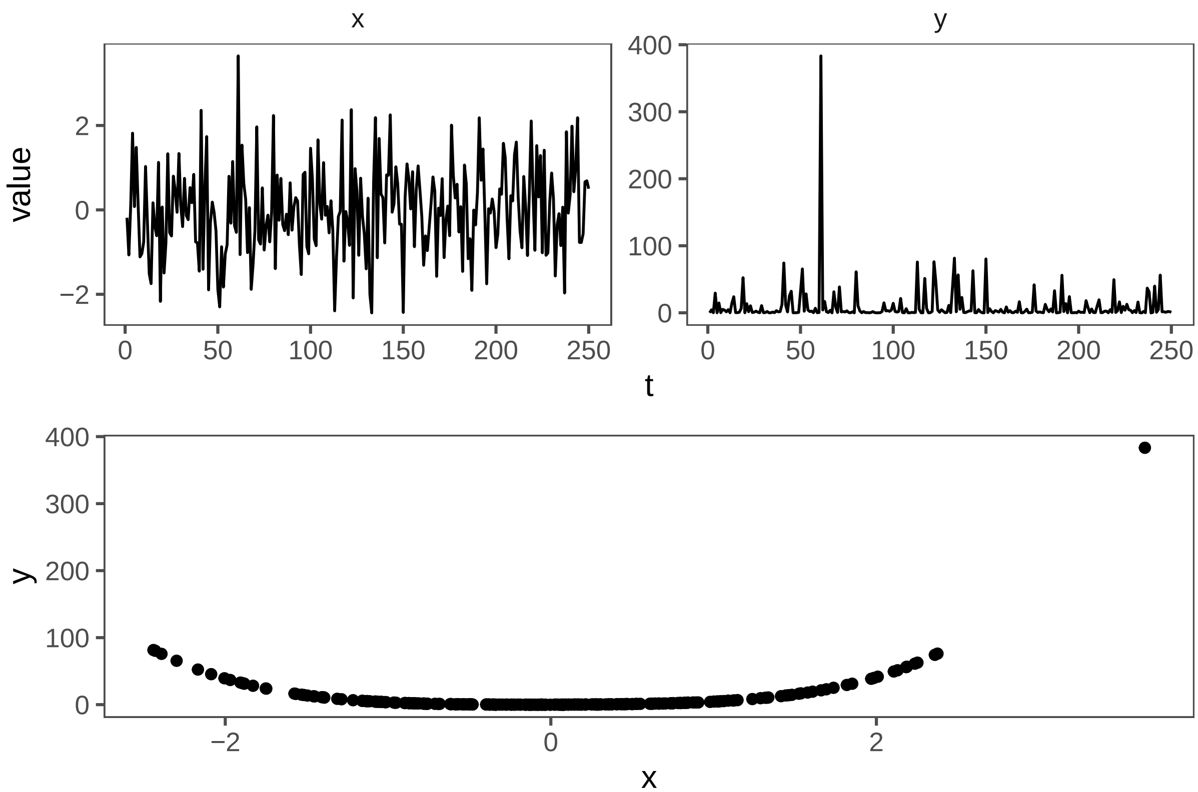
One point seems to be an **outlier**.

It might be a good idea to to remove the outlier now for plotting so that we have a more detailed (zoomed-in) look at the rest of the datapoints.



The x^2 component is ever clearer in the zoomed-in view.

Plot p1 and x_y_plot together in one *beautiful* plot.



From the scatterplot of the x and y variables we can assume that the a x^2 might be a good parameter for the model.

Correlation test

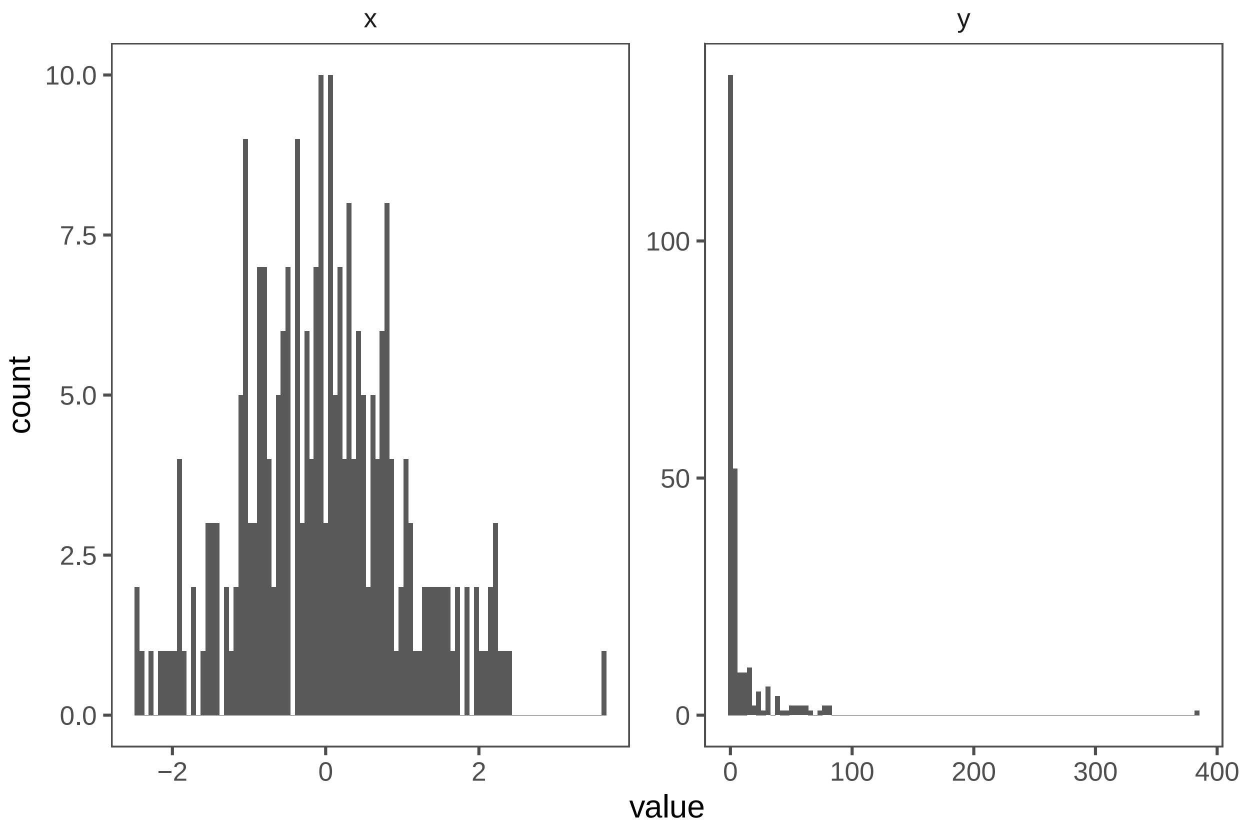
We can formally test whether there is correlation between x and y. Although we can already tell from the scatterplot that there must be some correlation. We can use **pearson's correlation coefficient**, testing hypothesis that true correlation differs from 0.

```
##
##  Pearson's product-moment correlation
##
## data:  data$x and data$y
## t = 3.5408, df = 248, p-value = 0.0004763
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.09796534 0.33433385
## sample estimates:
##       cor
## 0.2193661
```

There is small positive correlation between the two variables. Null hypothesis was rejected.

Distributions

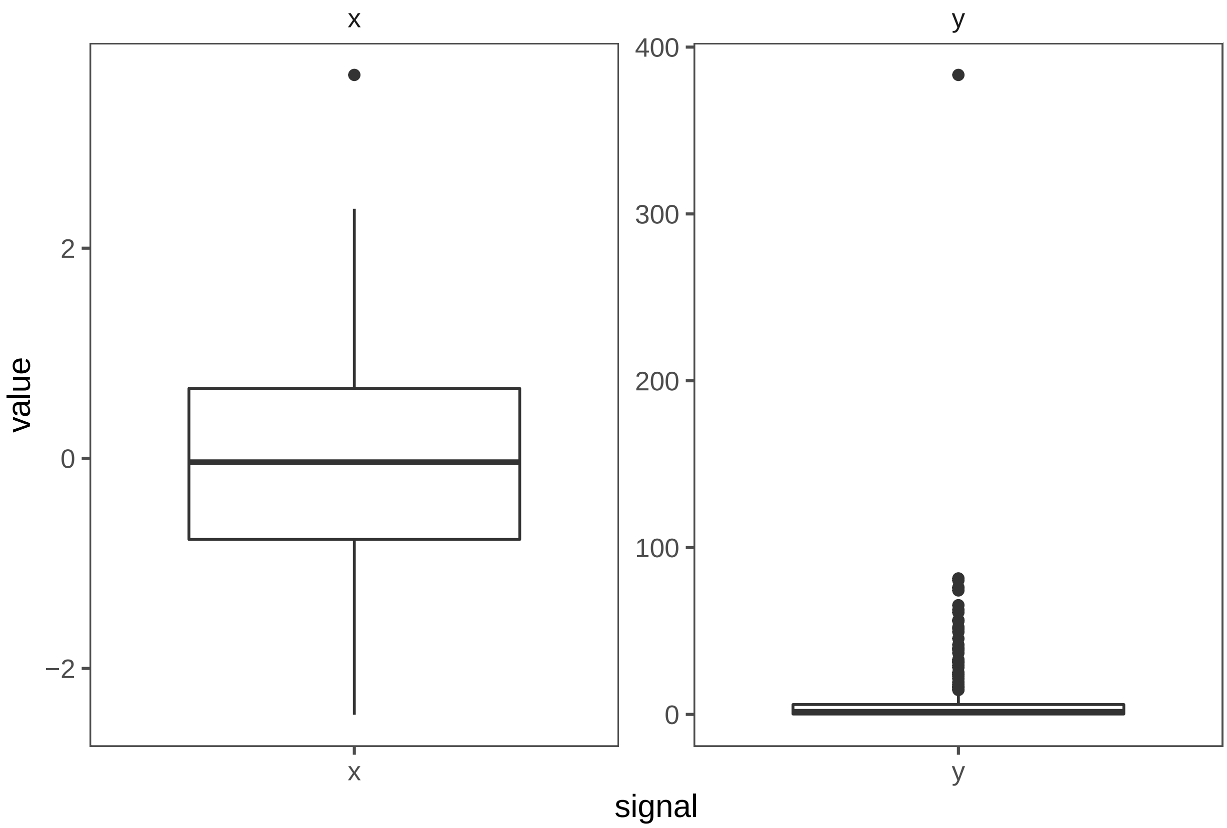
Now we inspect the distribution of both x and y



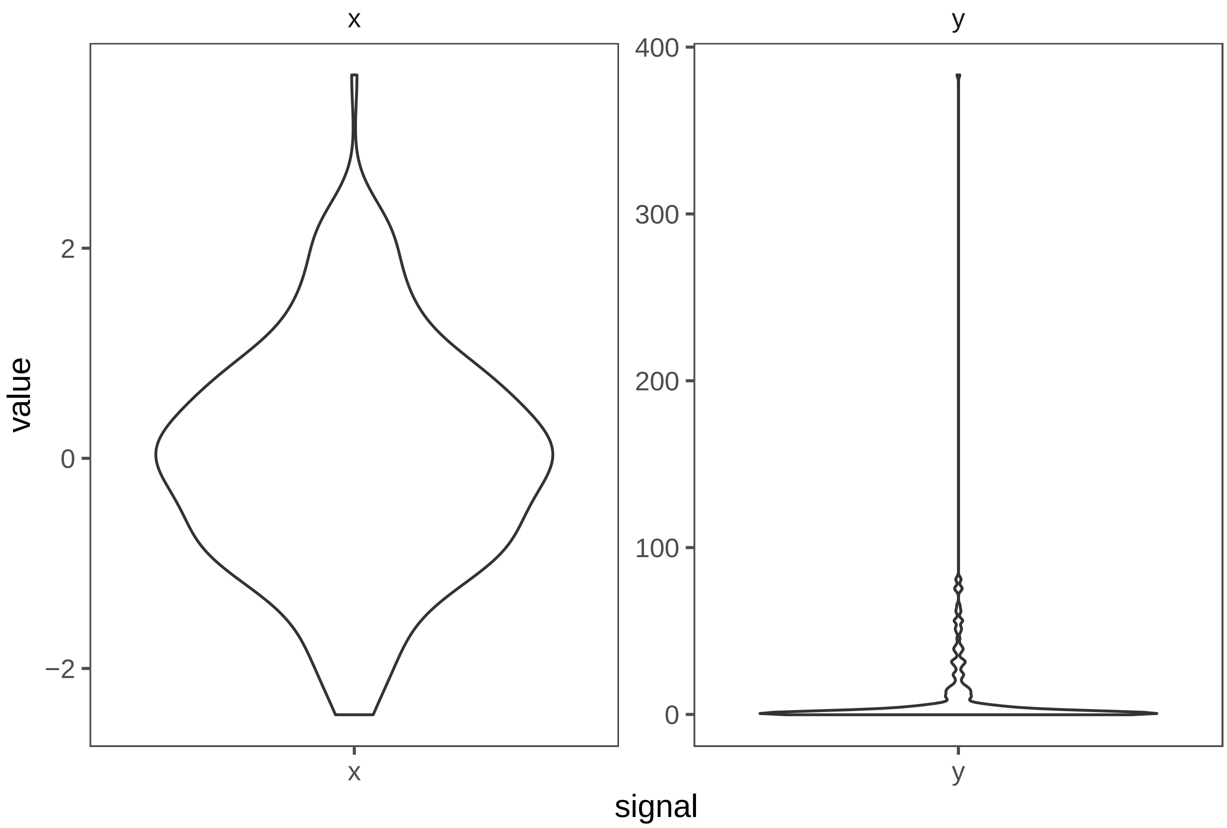
X seems to be approximately **normal** slightly skewed with heavy left tail. Y seems to be **exponentially distributed**. A hypothesis that y is **log-normal** might be worth testing.

Boxplots and violin plots

Let's continue with other tests about properties of the signals. First use boxplot and violin plots.



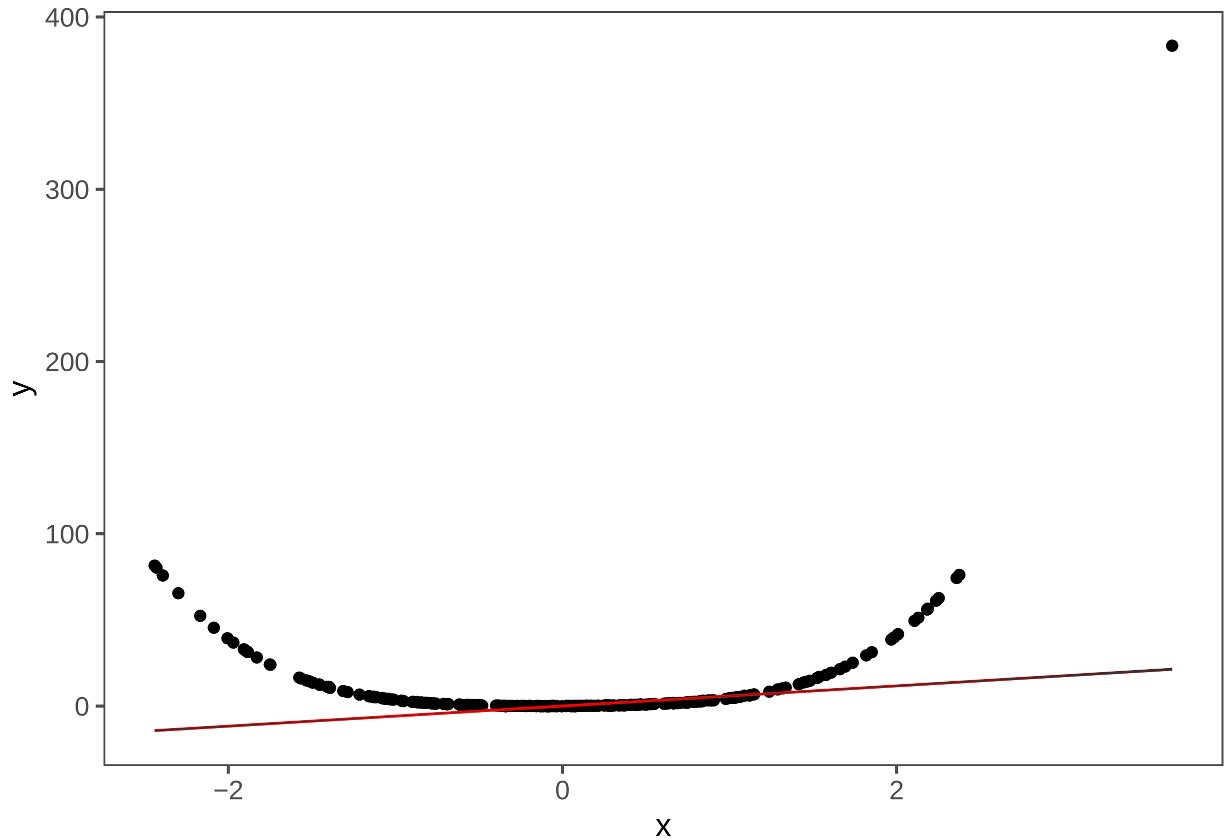

```
violins
```



Fit linear model

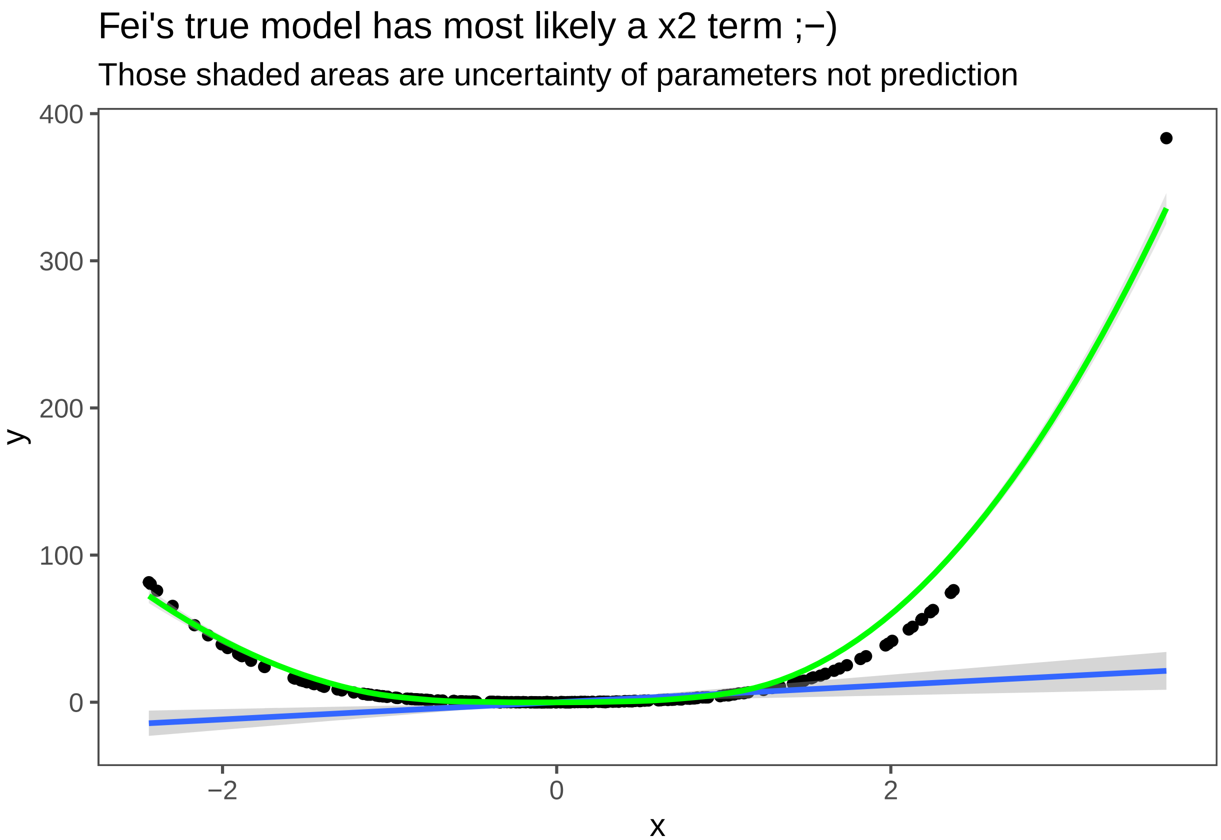
Try to fit a linear model with just one parameter: y ~ ß1*x

```
## [1] "SSE of the fitted model is: 470.702"
```



A bit of cheating

Just for fun, ggplot has function for fitting a simple linear model. There's also function for fitting a local polynomial surface/line which basically tries to find the best polynomial model (yes exactly what is our task in the


Fei's true model has most likely a x2 term ;−)
Those shaded areas are uncertainty of parameters not prediction
400
300
200
100
0
y
−2
0
2
x

coursework).