In [9]:

```
from ipywidgets import AppLayout, FloatSlider, interact, IntSlider, fixed, Dropdown
from knn_helper import plot_knn, plot_versus
%matplotlib inline
```

Why

We use these widgets below to:

- To get a better understanding of KNN,
- its parameter K,
- and how KNN compares to linear regression.

Please try to answer each question in the Jupyter notebook.

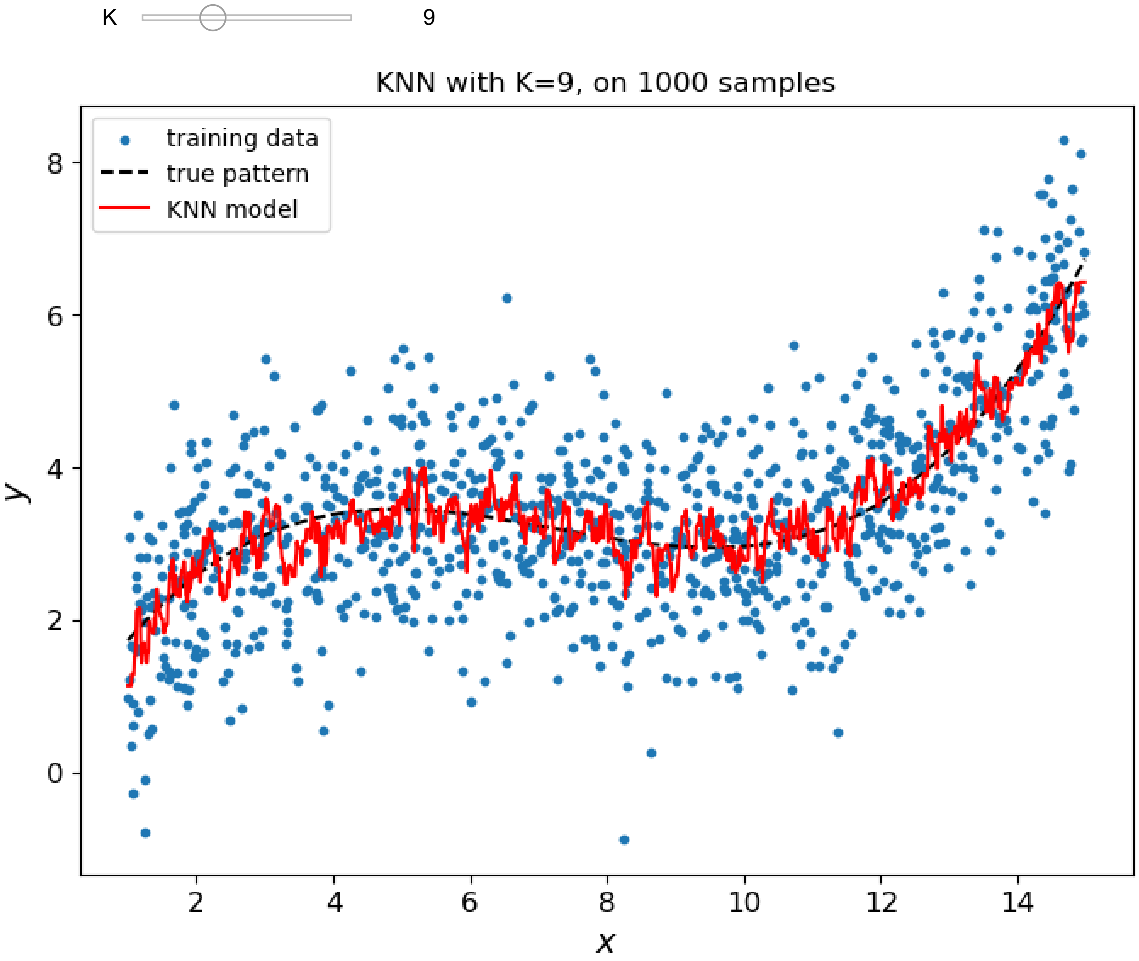
Widget 1

To start the widget, run the code cell above and the code cell below this text.

You can run a single cell by selecting it and pressing CTRL + ENTER.

In [2]:

```
interact(plot_knn, K=IntSlider(min=1,max=25), N=fixed(1000), degree=fixed(3), title=fixed(0))
```



Out[2]:

```
<function knn_helper.plot_knn(K, N, degree, title=0)>
```

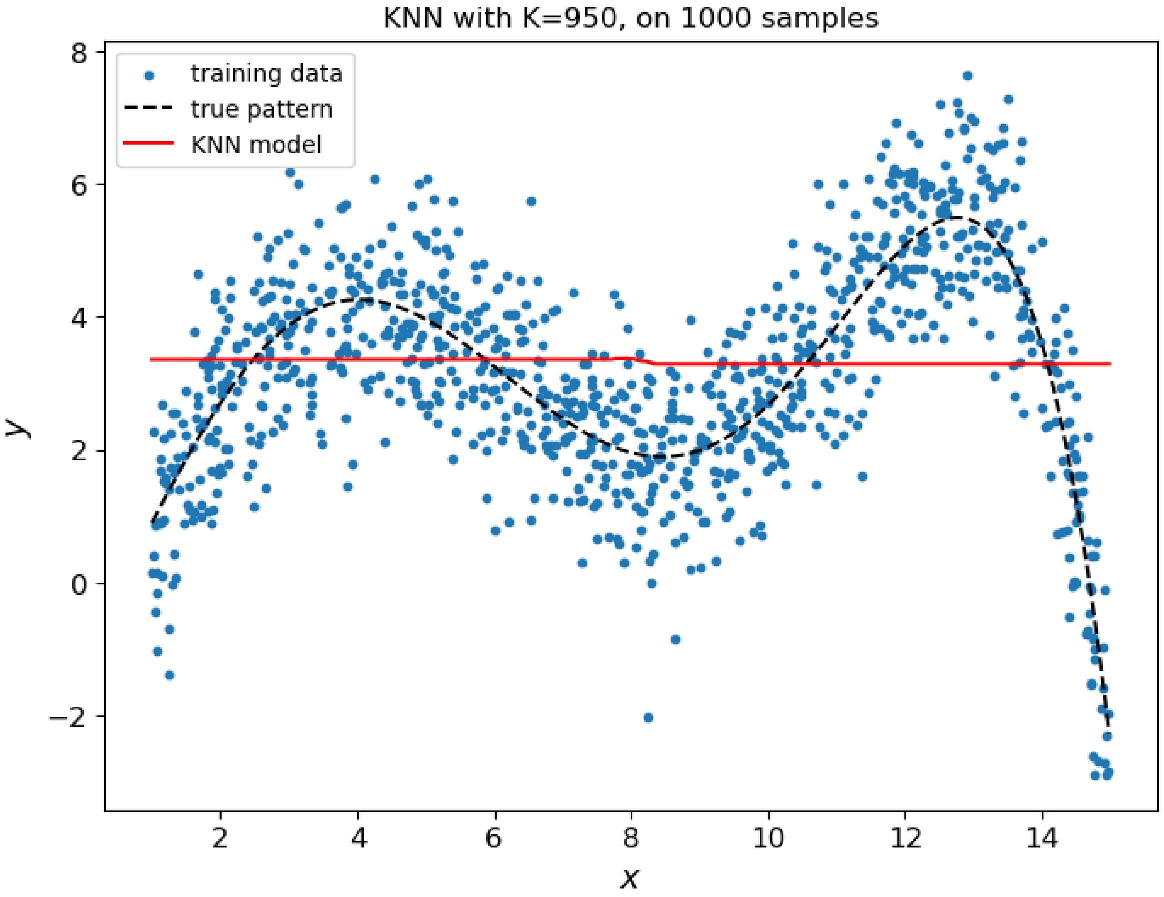
1a) What do you observe when you vary K?

```
In [ ]: # small K very noisy / wiggly fit, large K smoother
```

Widget 2

```
In [3]: interact(plot_knn, K=IntSlider(min=50,max=999,step=100), N=fixed(1000), degree=fixed(5), title
```

K 950



```
Out[3]: <function knn_helper.plot_knn(K, N, degree, title=0)>
```

2a) What do you observe for very large K? How can we explain this behavior?

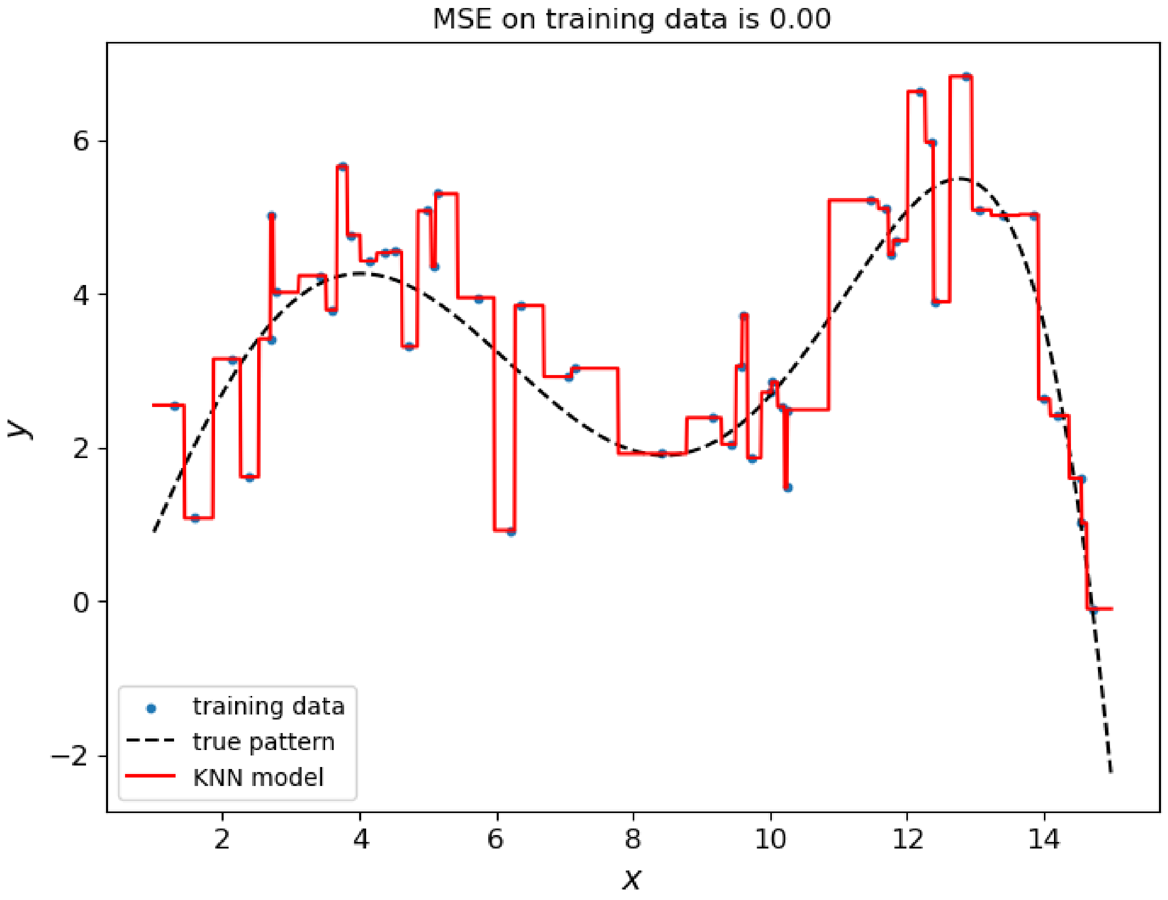
```
In [ ]: # the fit becomes a straight line
        # all samples are neighbours all the time. so we average all the data and get a straight line.
```

Widget 3

In [4]:

```
interact(plot_knn, K=IntSlider(min=1,max=10, value=10), N=fixed(50), degree=fixed(5), title=fi
```

K ○ 1



Out[4]:

```
<function knn_helper.plot_knn(K, N, degree, title=0)>
```

In the widget above we can see the mean squared error (MSE) on the training data (it is just above the plot).

3a) What happens to the MSE on the training data for very small K? How can we understand this?

In [ ]:

```
# the MSE becomes 0. Because we only look at 1 neighbour, and that is the training point itsel
```

3b) Does a small MSE on training data mean it work well on new unseen test data? Why (not)?

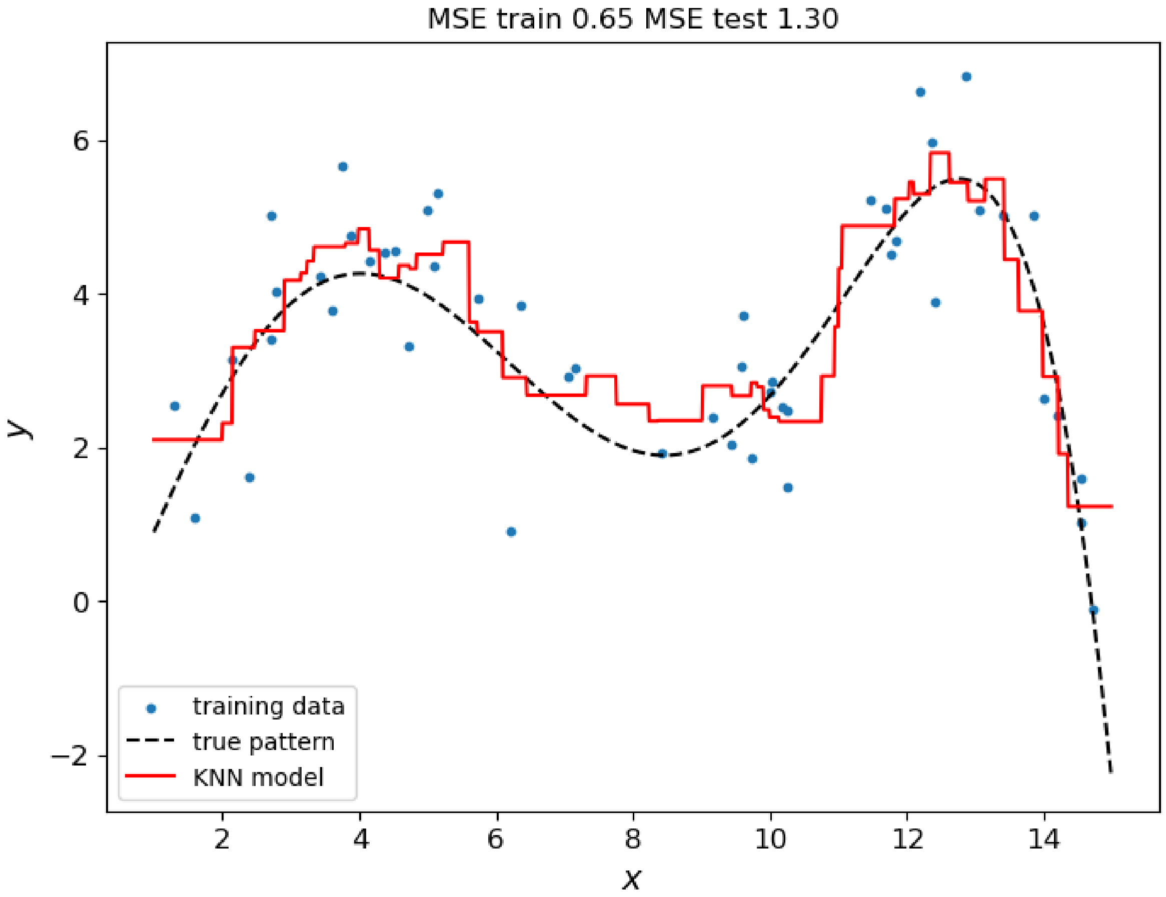
In [5]:

```
# no it could be overfitting. this means that the fit on the training data is very good, but o
# the performance is not good
```

Widget 4

```
In [6]: interact(plot_knn, K=IntSlider(min=1,max=10), N=fixed(50), degree=fixed(5), title=fixed(1))
```

K ○ 4



```
Out[6]: <function knn_helper.plot_knn(K, N, degree, title=0)>
```

In the widget above we can see the mean squared error (MSE) on the training data and new unseen data. New unseen data means the model did not see this data during training. Such new data is also called "test data"

4a) For which value of K do we get the lowest test MSE?

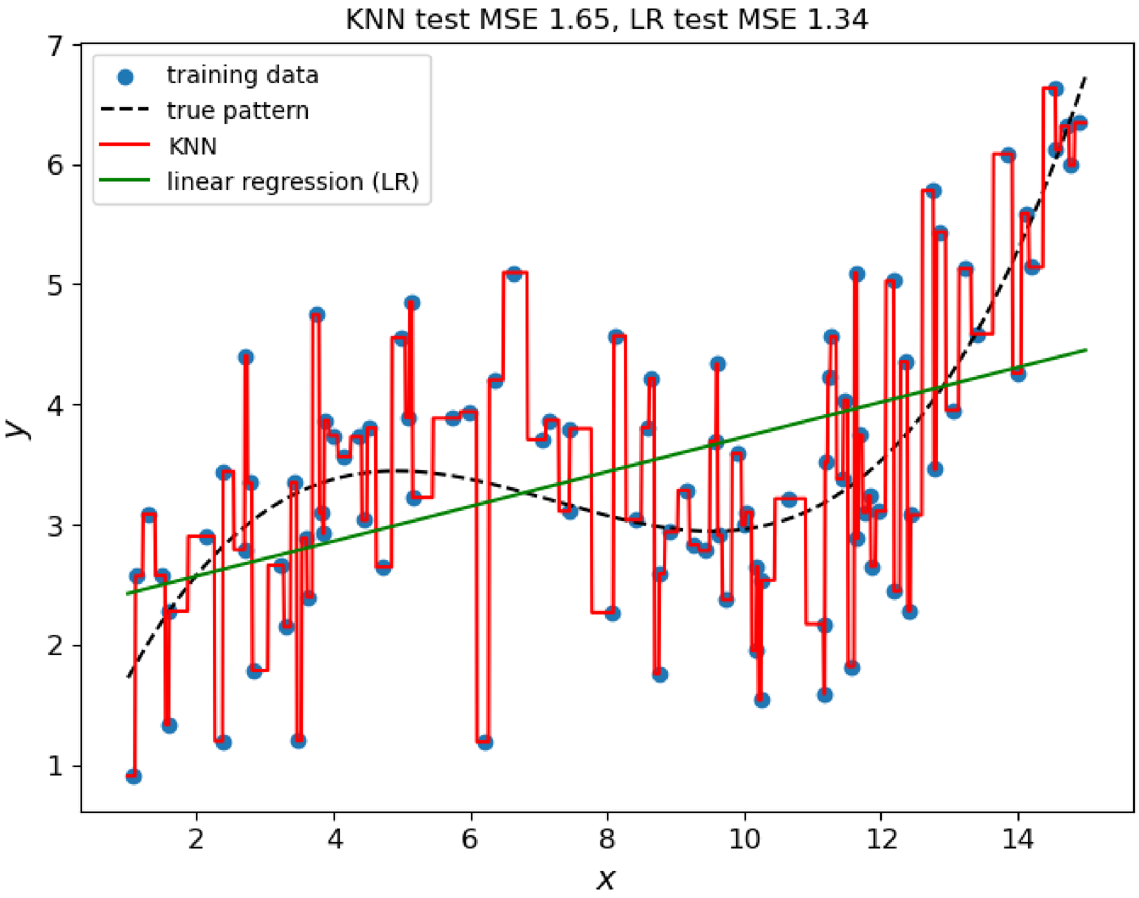
```
In [ ]: # k = 4
```

Widget 5

In [7]:
```
interact(plot_versus, K=IntSlider(min=1,max=10,value=5),N=IntSlider(min=1,max=100,value=100),
```

K 1

N 100



Out[7]:
```
<function knn_helper.plot_versus(K, N, degree, train=True)>
```

This widget has several controls. K controls the number of neighbours for KNN, this model is plotted in red. The linear regression line is plotted in green. N controls the number of training samples. The test MSE of KNN and linear regression (LR) are put in the title.

a) Try several values for K and N. When is KNN better (in terms of test MSE) than linear regression? Can you explain this behavior?

In [ ]:
```
# for large k and n, KNN is better
# need a large n so we can find the pattern
# need a large k to average out the noise
```

b) When is LR better than KNN? Can you explain this behavior?

In [ ]:
```
# for a low k, KNN doesn't perform well (because noise not averaged out enough)
# for a low sample size, KNN doesn't work well (because not enough neighbours around each poin
```

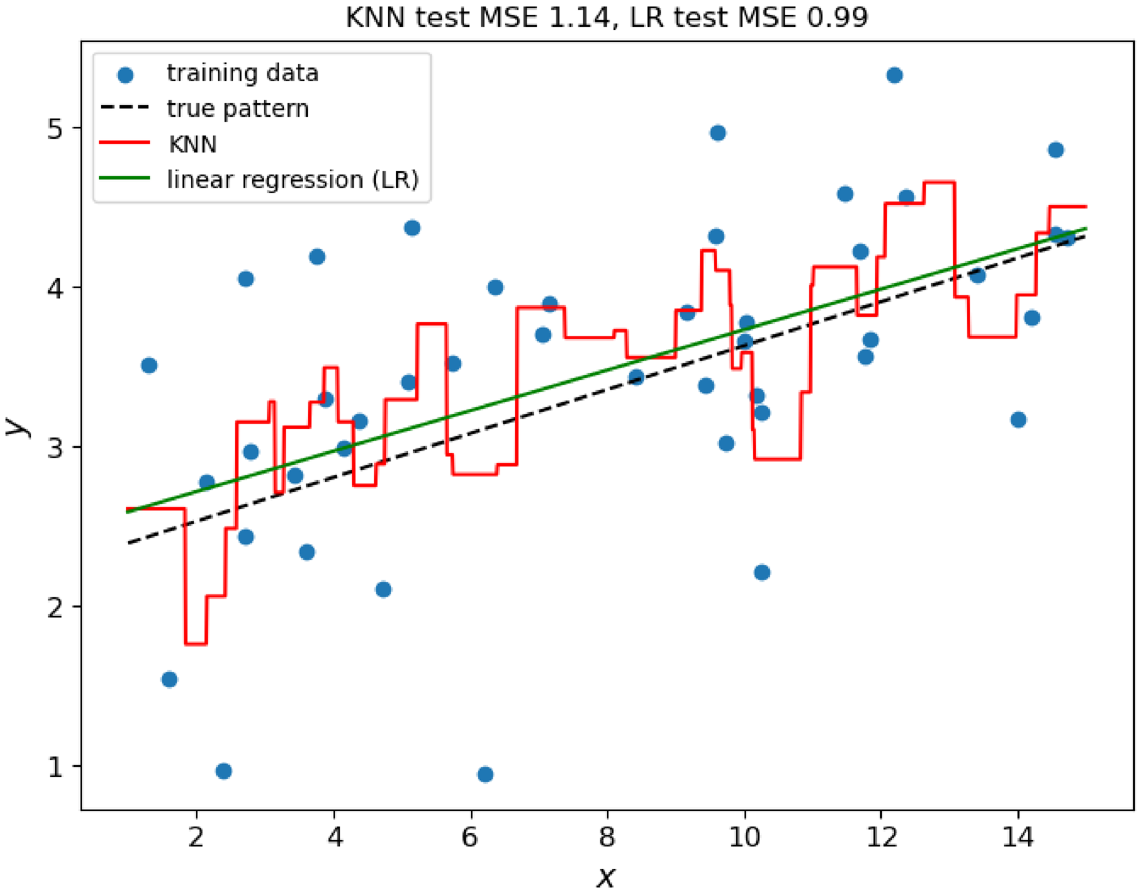
Widget 6

In [8]:

```
interact(plot_versus, K=IntSlider(min=1,max=10,value=10),N=IntSlider(min=1,max=100,value=100),
```

K 3

N 44



Out[8]:

```
<function knn_helper.plot_versus(K, N, degree, train=True)>
```

a) Try several values for K and N. Which model is better in this case most of the time? Can you explain why?

In [ ]:

```
# here linear regression often is better, because the true pattern is a straight line
```