In [43]:

```
# If you don't yet have these packages, you can uncomment the follow line & and run it
#!pip install numpy pandas matplotlib seaborn sklearn
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
sns.set_theme()
```

Directions

- This examination consists of *2* problems, each with multiple subproblems. You have between 2 and 3:50pm to work on it.
- It is an open-everything exam. However, don't communicate with another person or AI. The Internet can only be used in read-only mode. **Be reasonable.**
- Problem 1 from the Stats module must be submitted as a PDF file. You can use Maple or Python. Show your work and answer the related questions in the PDF file you're submitting.
- Problem 2 from the Machine Learning module must be submitted as an IPython notebook. You will work directly on this notebook. Show your work and answer all the questions in here.
- Before the time is up, you'll make two separate submissions on Canvas: one for Problem 1 and one for Problem 2.

Problem 1: Regression (Stats Module)

Labor and material costs are two basic components in the cost of construction. Changes in the component costs of course lead to changes in total construction costs. The accompanying table tracks changes in construction cost and cost of all construction materials for 8 consecutive months.

In [44]:

```
labor_material = pd.read_csv("cccm.csv")
labor_material
```

Out[44]:

	Month	Construction Cost (y)	Index of All Construction Materials (x)
0	January	193.2	180.0
1	February	193.1	181.7
2	March	193.6	184.1
3	April	195.1	185.3
4	May	195.6	185.7
5	June	198.1	185.9
6	July	200.9	187.7
7	August	202.7	189.6

Your Task

1. Prepare a scatter diagram.
2. Find the least squares straight line.
3. Compute the sample correlation coefficient.
4. Do the data provide sufficient evidence to indicate a nonzero correlation between the monthly construction costs and indexes of all construction materials? Test at $\alpha = 0.05$.
5. Do the residual plot. In your opinion, are the assumptions of the model satisfied?

In [45]:

```
X = labor_material.iloc[:, 2]
X
```

Out[45]:

```
0    180.0
1    181.7
2    184.1
3    185.3
4    185.7
5    185.9
6    187.7
7    189.6
Name:  Index of All Construction Materials (x), dtype: float64
```

In [46]:

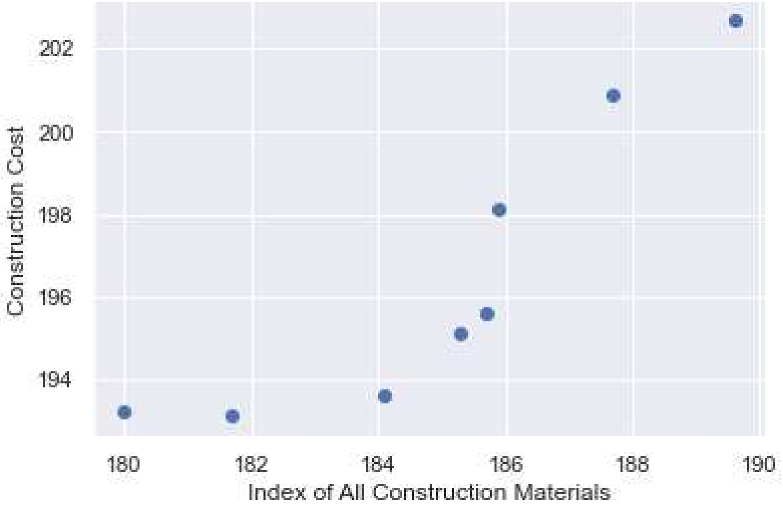
```
Y = labor_material.iloc[:, 1]
Y
```

Out[46]:

```
0    193.2
1    193.1
2    193.6
3    195.1
4    195.6
5    198.1
6    200.9
7    202.7
Name: Construction Cost (y), dtype: float64
```

In [58]:

```
plt.scatter(X, Y)
plt.xlabel("Index of All Construction Materials")
plt.ylabel ("Construction Cost")
plt.show()
```

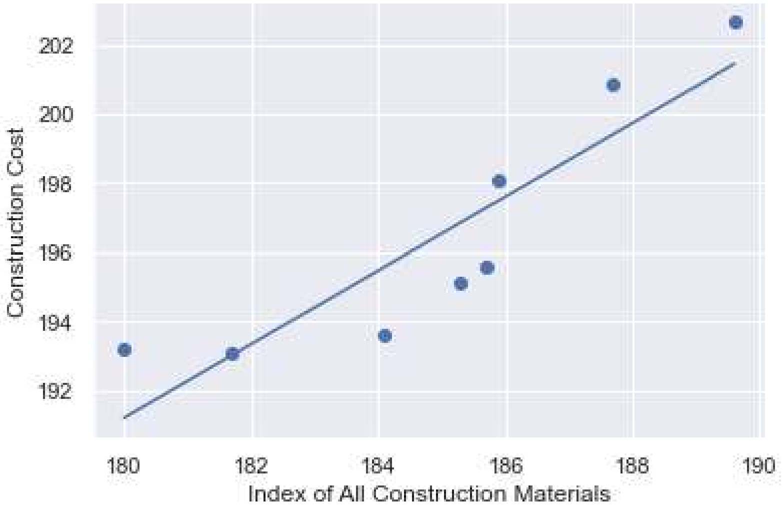


In [59]:

```
from scipy import stats
slope, intercept, r, p, std_err = stats.linregress(X, Y)

def myfunc(X):
 return slope * X + intercept

mymodel = list(map(myfunc, X))
plt.scatter(X, Y)
plt.plot(X, slope * X + intercept)
plt.xlabel("Index of All Construction Materials")
plt.ylabel ("Construction Cost")
plt.show()
```



In [55]:

```
print('Correlation:', r)
```

```
Correlation: 0.8993640374131749
```

In [60]:

```
p
```

Out[60]:

```
0.002359556529633942
```

In [ ]:

```
```

Problem 2: Hodgepodges (Machine Learning Module)

Consider the following dataset. Each row has 7 sensor readings (x0, x1, ..., x6) and a label column (y).

In [50]:

```
sensors_meh = pd.read_csv("muzoo.csv")
sensors_meh.head()
```

Out[50]:

	x0	x1	x2	x3	x4	x5	x6	y
0	-4.597071	-2.999242	-5.735565	-3.984228	-5.149052	-7.775142	-3.785790	1.0
1	8.130288	5.906141	6.174752	2.137881	8.168127	5.449366	4.690545	0.0
2	-5.506438	-2.172412	-5.438821	-2.389202	-3.391176	-6.585607	-2.252300	1.0
3	2.722533	-1.120358	-1.180225	-2.296887	-1.722477	-0.065035	-0.809943	1.0
4	-6.638931	-3.538184	-4.431006	-5.122881	-5.451171	-7.744939	-3.800036	1.0

Problem 2.1: Train, Test, Classify

1. Prepare training and validation data from the given data set. The train/validate split ratio shoule be 80:20.
2. Train *TWO* classification models and report the test accuracy of each trained model. The two models can be of the same learning algorithm with different hyperparameters, or they can be two different learning algorithms.
3. Choose the better model, perform 5-fold cross validation, and report the average accuracy, along with the standard deviation.
4. Suggest a way to improve the classification accuracy

In [51]:

```
# Begin your work for this problem here. Add additional cells as you see fit. Switch t
# Math mode using LaTeX-like syntax (e.g., $x = 2 + y$) works in Markdown mode as well
```

Problem 2.2: Smashing Dimensions

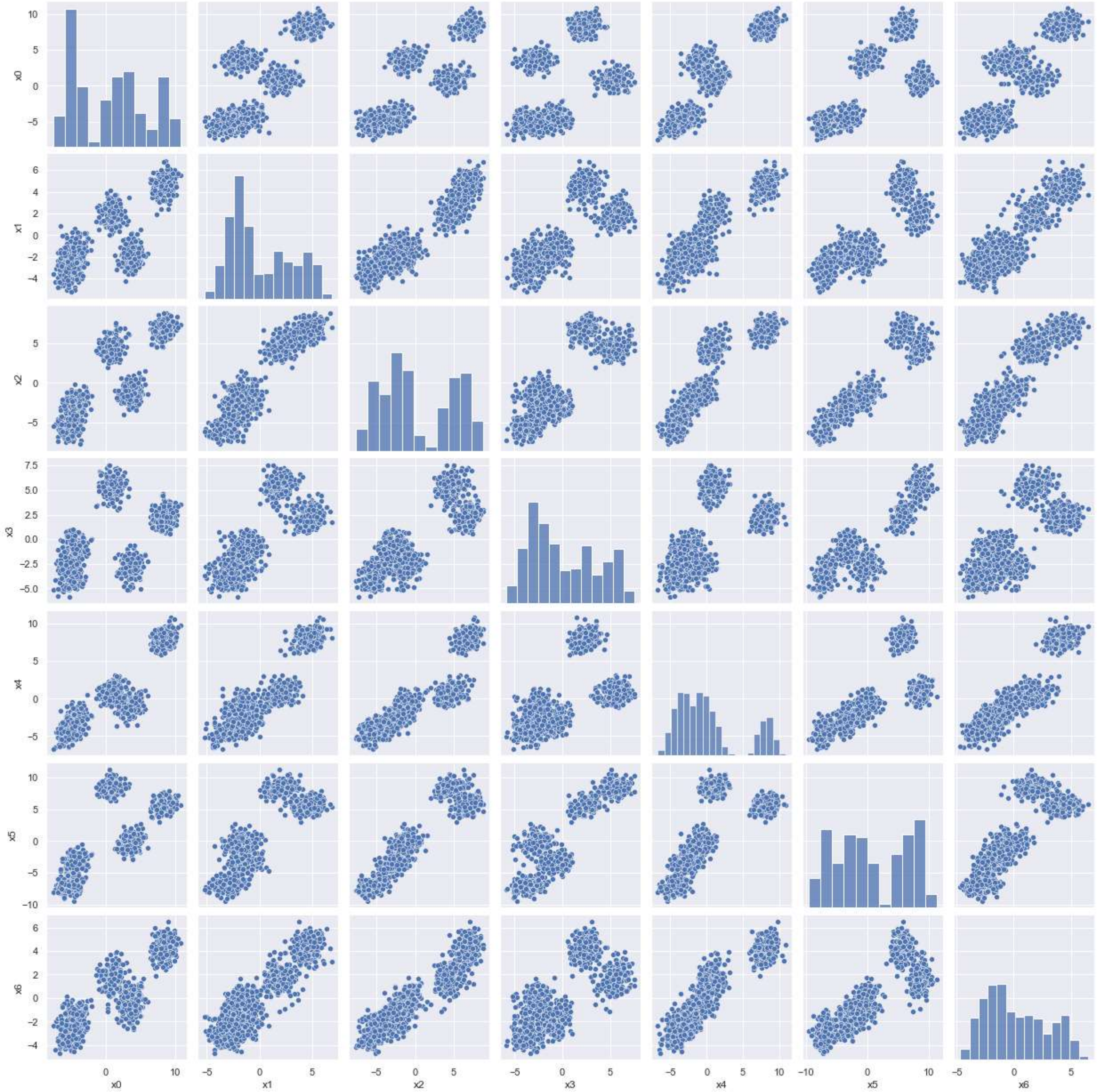
When we have a dataset in 7 dimensions, visually understanding it in the current form is nearly impossible. To get us started, below are their pair plots, showing the scatter plot for each pair of dimensions (xi vs xj), excluding the label column.

In [52]:

```
sns.pairplot(sensors_meh.iloc[:,:-1])
```

Out[52]:

```
<seaborn.axisgrid.PairGrid at 0x2b4d0caba30>
```



We will drill down further. Towards this goal, we'd like to reduce the dimensions down to the bare minimum. For this part, you'll **only use x1, ..., x6; ignore the labels**.

Your Task: Apply PCA (properly) to this dataset. You must show your work. Explain your reasoning as you go along.

1. How many dimensions should be kept? Our goal is to use the fewest dimensions that will have at least 90% variance explained. (*Hint:* ≤ 3 dimensions, so their scatter plot can be visually displayed.)
2. How did you figure this out and confirm it?
3. Optionally (no credit but it might help you understand the nature of the data a bit more), what does the reduced-dimension dataset look like? Show a scatter plot.

In [53]:

```
# Begin your work for this problem here. Add additional cells as you see fit. Switch t
# Math mode using LaTeX-like syntax (e.g., $x = 2 + y$) works in Markdown mode as well
```

Problem 2.3: Cluster 'Em

This (sub)problem will be a continuation of the previous one. You should **use the reduced-dimension dataset**; however, if you are unable to, you can use the initial dataset. Hence, for this part, too, you'll **ignore the labels**.

Your Task: Apply a clustering algorithm from class (K-Means, Agglomerative clustering) to the (reduced-dimension) dataset. You must show your work. Explain your reasoning as you go along.

1. What is the best number of clusters? How did you determine that?
2. Generate a clustering using the above number of clusters. Your variable `labels` will be a list or a numpy array where `labels[i]` indicates which cluster the data point in row `i` is. More specifically, if there are `k` clusters, `labels[i]` is a number between 0 and $k-1$ (inclusive).
3. Optionally (no credit), plot the dataset color-coding each point so we know which cluster it belongs to.

In [54]:

```
# Begin your work for this problem here. Add additional cells as you see fit. Switch t
# Math mode using LaTeX-like syntax (e.g., $x = 2 + y$) works in Markdown mode as well
```