Homework Instructions

How to submit assignments

1. **Place your answer in the indicated place in the document**. Each question is followed by one or more sections, each with an empty cell called "ANSWER". Insert your answer for that question below the word "ANSWER". You often want to use multiple cells for your answer, including both markdown and code.
2. **Ensure that all cells have been run**. Cells that have not been run will be treated as unanswered questions and assigned zero points.
3. **Create a PDF document of your notebook**. There are a few ways to do this. For example, using Google Chrome, go to File->Print Preview, then after verifying the document looks correct, go to print and for your printer choose "Save as PDF."
4. **Check that your content is easily legible prior to submission**. Look over your PDF before you submit it. If I cannot read it, or parts are missing, I cannot grade it, and no credit will be given.
5. **Submit your pdf on Sakai before Sunday Midnight in your local time zone**. For some homeworks you might also need to upload your Jupyter notebook. In that case run *Restart & Clear Output* first.

Guidelines for assignments

- Code should be valid (able to run and producing the correct answer), understandable, and well commented.
- All math should be typeset using Latex equations. See e.g. this page (https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Typesetting%20Equations.html) for some examples.
- All text that is not code should be formatted using markdown. Two references to help include: ref1 (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) and ref2 (http://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html).
- For full credit, submitted assignments should be easily navigable with answers for subquestions clearly indicated (e.g. using "(a)" to indicate a subquestion).
- Text should be descriptive and provide adequate detail to answer questions. When questions as "why" or for you to describe findings, the expectation is that the response will be much more than one or two words.

Example Question and Response

Below is an example question to demostrate how to answer a question using a Jupyter notebook.

1

Calculate the (a) first and (b) second derivative of $f(x) = x^3$. lastly, (c) numerically evaluate the second derivative for $x = 3$.

ANSWER The first and second derivatives are calculated as follows:

(a) The first derivative:

$$\frac{df}{dx} = 3x^2$$

(b) The second derivative:

$$\frac{d^2f}{dx^2} = 6x$$

In [1]:
```python
# (c) Numerically evaluate the second derivative for f(x)

# Initialize variables for the analysis
x = 3

# Compute the derivative
df2 = 6 * x

print('(c) The second derivative at x = 3 is {}'.format(df2))
```

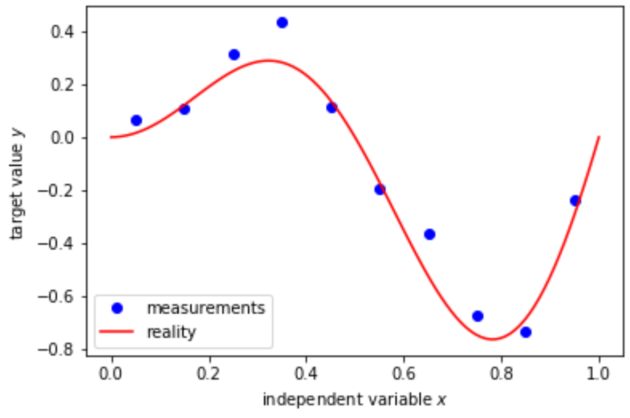
```
(c) The second derivative at x = 3 is 18
```

Figure guidelines

- **All** plots should have axes labels and legible fonts. Axis lable have to be explicit, using just the letter used in the equations is not sufficient.
- Legends should be used when there are multiple series plotted on a single plot.
- All figures should be of adequate size to read and should be clear.
- Markers on plots should properly sized (e.g. not be too big nor too small) to be able to clearly read the data.

Figure Example:

In [2]:
```
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline

# make results reproducible
np.random.seed(42)

# the mathematical law behind our subject of study
def world(x):
    return x * np.sin(2*np.pi*x)

# for these x values we will take measurements
N = 10  # the number of measurements we make
x=np.linspace(0.05,0.95,N)

# the real values of the system, unknown to us
reality = world(x)

# all measurements are inherently noisy
noise = 0.1 *np.random.randn(N)
y = reality + noise

plt.plot(x,y, "ob", label = "measurements")

x_plot = np.linspace(0, 1, 100) # generate points for making a line plot
plt.plot(x_plot, world(x_plot), "r", label="reality")

plt.xlabel('independent variable $x$')
plt.ylabel('target value $y$')
plt.legend(loc='lower left')

plt.show()
```



MAC USERS TAKE NOTE:

For clearer plots in Jupyter notebooks on macs, run the following line of code:

%config InlineBackend.figure_format = 'retina'

Sources

Some questions on the assignments are adapted from sources including:

1. Kyle Bradburry, IDS 705: Principles of Machine Learning, Duke University
2. Christopher Bishop, *Pattern Recognition and Machine Learning*