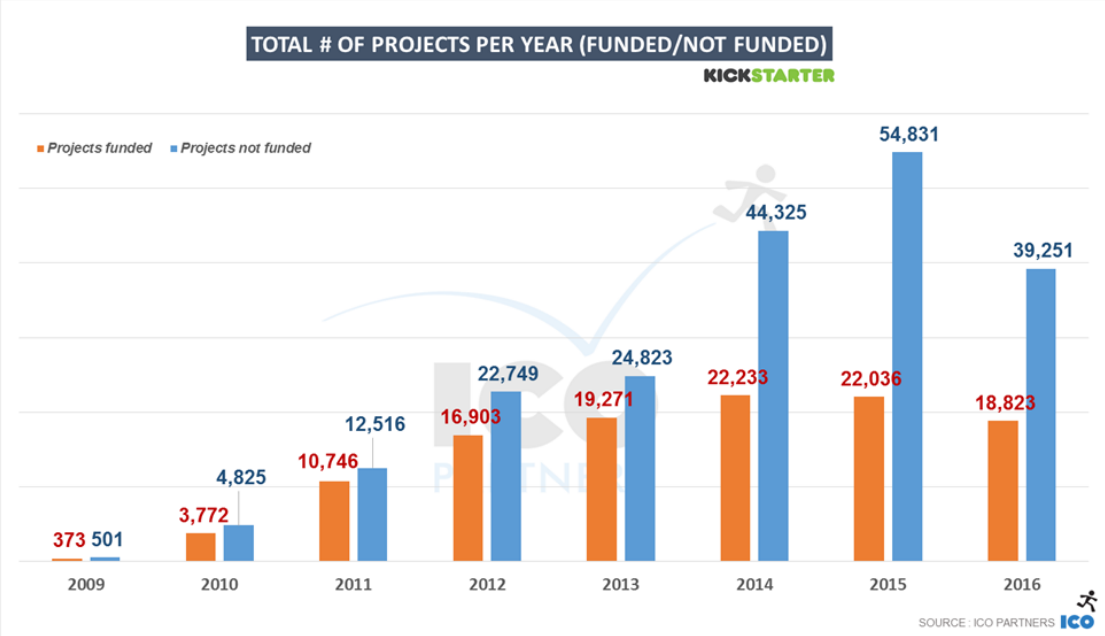
Assignment #4

This assignment is due on May 14th via email to christian.wallraven+EMS2019@gmail.com.

If you are done with the assignment, make one zip-file of the `assignment4` directory and call this zip-file `STUDENTID1_STUDENTID2_STUDENTID3_A4.zip` (e.g.: `2016010000_2017010001_A4.zip` for a team consisting of two students or `2016010000_2017010001_2017010002_A4.zip` for a three-student team). The order of the IDs does not matter, but the correctness of the IDs does! **Please double-check that the name of the file is correct!!**

Please make sure to comment the code, so that I can understand what it does. Uncommented code will reduce your points!
Downloading and copying and pasting other peoples' code is plagiarism and will NOT be tolerated. If you work as a team, the code needs to contain all team members' names!!

Part1 Function fitting (20 points)



Here is a graph showing the number of funded and non-funded projects on Kickstarter from 2009 to 2016.

In a script called `kickstarter.m`, enter the two datasets into two variables called funded and nonfunded. Then **fit this data with polynomials from degree 1 until 7**, using the **Vandermonde** method shown in class (you should use the backslash to solve for the polynomial coefficients).

Which of the models do you think fits the data **best** for each of the two datasets (**remember the compromise between fit quality and generalizability!!!**)? Do the degrees differ for the two datasets? Why would they? Why would they not?

Plot the measured data points, along with your **best-fit model** into the same plot. Using your model as a predictor model, how many projects will be funded and non-funded in 2020? Plot these points as well into the same plot.
Make sure to insert all observations and interpretations as comments into your script!

Part2 PCA and Linear Regression (40 points):

In this part you will analyze the data from `dataForTesting.mat` a bit further.

In the following create a script called `analyzeHousing.m`

Load and clean the data. For this set all data elements that are negative and that are above 1000000 to NaN. Then replace all NaN values in each column of the data by the column's median value.

Import the category names [they are written in LARGE CAPS] and descriptions from the file `housingDescription.txt` into two separate variables using a function or script [you can use the Import Data Wizard to create these!].

Split the data into 13 predictor variables `pred` [the first 13 columns of data] and 1 target variable `price` [the 14th column of the data].

a) Using `imagesc` and `corr`, create a matrix of the inter-correlations of `pred`. How many variables correlate well?

b) We want to now reduce the variables in `pred` to the "core directions" of our data. For this, we will do PCA. However, we have a problem – the data in `pred` is very differently scaled and PCA is sensitive to that. Therefore, we will weight each of the data dimensions using the inverse of their variance!

   You can do this in Matlab using the following commands:

```
w = 1./var(data); % create a vector of weights
[wcoeff,score,latent,~,explained] = pca(data,...
    'VariableWeights',w); % do weighted PCA
coefforth = inv(diag(std(data)))*wcoeff; % get orthogonal
eigenvectors
```

   The first variable `wcoeff` contains the eigenvectors of the matrix [the principal components] – these should be orthogonal, but the weighting creates non-orthogonal vectors as output, so we fix this here.

   The second variable `score` contains the projected coordinates for each of the points in the new coordinate system. You can use a so-called biplot to take a look at how each point is projected, and how the original 13 predictor axes have been rotated in a new, reduced coordinate system.

   Plot one with the first three principal components:

```
biplot(<vectors of components>,'Scores',…,'Varlabels',…);
```

   Can you identify "outliers" in this plot? Use the data cursor to write down a few indices of potential outliers and add them to the script as comments.

c) `Explained` contains the explained variance of each of the principal components. How many components do you need to explain 70% of the variance? How many to explain 90%? Insert as comments into the script.

d) Let's choose the number of components for reduction to be such that 70% is explained.

   Now do a linear regression of the 206 datapoints in this three-dimensional space to predict the price. Remember:

```
price(i) = a(1) + a(2)*pred(i,1)+a(3)*pred(i,2)+ a(4)*pred(i,3)
```

Use this logic to create a fitting matrix `P` and use the backslash to solve for the coefficients `a`.

Determine the residual as the `norm(P*a-price)` and print it out. This is the fit-quality of the three most "variance-containing" directions in our data.

e) So is this fit-quality good? Can PCA help us to select "good" dimensions for fitting our linear model?

In order to answer this question, do the full fit using all 13 dimensions of the original data pred using

```
price(i) = a(1) + a(2)*pred(i,1)+a(3)*pred(i,2)+…a(14)*pred(i,13)
```

What is the fit-quality of this full model?? Better or worse – and why?

In order to answer whether PCA can help us select the "best" 3 dimensions, how about we simply take all possible sets of 3 dimensions from the original 13 dimensions and perform a linear regression with them.

Implement code that lists all possible sets of 3 from 13, does the linear regression, and saves the fit-quality in a vector `r2orig`.

Which dimensions consistently have the lowest residual? Can you interpret them using the names and descriptions in the file? Insert all interpretations as comments into the script.