Week 9 Assignment – Final Report

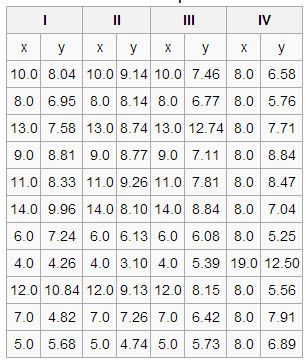
Final code and supporting analysis for this assignment can be found at [*https://github.com/DataProgrammingBridge2014/Final*](https://github.com/DataProgrammingBridge2014/Final)*.*

## I. Introduction

In this week's project, a team of CUNY MSDA Programming Bridge students worked together to explore and analyze four small data sets. The team used a variety of functions and libraries in R and Python, including the matplotlib library and ggplot2 package, to analyze the data. The team was able to illustrate the importance of looking at data from a variety of perspectives before drawing final conclusions.

## II. Data

These four datasets are together known as Anscombe's Quartet[[1]](#endnote-1), developed in 1973 by famed statistician Francis Anscombe[[2]](#endnote-2) to demonstrate the significance of graphically visualizing data and the potentially large influence on the data of one or two significant outliers. The data consist of four sets of *x, y* pairs each 11 rows deep.



## The data were clean and did not require significant preparation. Team members put all four sets in a .csv file for easy distribution.

## III. Analysis Methods

The project used two popular open-sourced programming languages for data analysis: R and Python.  For the data exploration, the main focus of the study was to use two graphing tools, ggplot2 with R and matplotlib with Python. In addition, exploratory datasets were generated to mimic a real-life process. The following procedures and methods were applied on all the datasets and used in both languages for the analysis: summary statistics, plots, and regression.

## IV. Exploration

## Descriptive statistics

## Summary statistics for each of the 4 datasets are similar. The *y* value in each dataset has a mean of approximately 7.5 and a variance of 4.123. The relatively similar range between the maximum and minimum values across each dataset further reinforces the idea that the identical means and variances are not a coincidence.

## x1 y1 x2 y2 x3

## *Min. : 4.0 Min. : 4.260 Min. : 4.0 Min. :3.100 Min. : 4.0*

## 1st Qu.: 6.5 1st Qu.: 6.315 1st Qu.: 6.5 1st Qu.:6.695 1st Qu.: 6.5

## *Median : 9.0 Median : 7.580 Median : 9.0 Median :8.140 Median : 9.0*

## *Mean : 9.0 Mean : 7.501 Mean : 9.0 Mean :7.501 Mean : 9.0*

## 3rd Qu.:11.5 3rd Qu.: 8.570 3rd Qu.:11.5 3rd Qu.:8.950 3rd Qu.:11.5

## *Max. :14.0 Max. :10.840 Max. :14.0 Max. :9.260 Max. :14.0*

## 

## y3 x4 y4

## *Min. : 5.39 Min. : 8 Min. : 5.250*

## 1st Qu.: 6.25 1st Qu.: 8 1st Qu.: 6.170

## *Median : 7.11 Median : 8 Median : 7.040*

## *Mean : 7.50 Mean : 9 Mean : 7.501*

## 3rd Qu.: 7.98 3rd Qu.: 8 3rd Qu.: 8.190

## *Max. :12.74 Max. :19 Max. :12.500*

**Variance:**

> apply(cuny9, 2, var)

x1 y1 x2 y2 x3 y3 x4 y4 11.000000 ***4.127269*** 11.000000 ***4.127629*** 11.000000 ***4.122620*** 11.000000 ***4.123249***

The skew measures the degree of asymmetry of a distribution about the mean. If the left tail is more pronounced than the right tail, the function has negative skewness. If the reverse is true, it has positive skewness. Symmetric data has a skew value of zero. The skew value of each of our datasets is close to zero, slightly to the left for series No. 1 and No. 2 and slightly to the right for series No. 3 and No. 4.

## Skewness ():

## [1] "x1"

## [1] 0

## [1] "y1"

## [1] -0.04837355

## [1] "x2"

## [1] 0

## [1] "y2"

## [1] -0.9786929

## [1] "x3"

## [1] 0

## [1] "y3"

## [1] 1.38012

## [1] "x4"

## [1] 2.466911

## [1] "y4"

## [1] 1.120774

## Kurtosis ():

## [1] "x1"

## [1] -1.528926

## [1] "y1"

## [1] -1.199123

## [1] "x2"

## [1] -1.528926

## [1] "y2"

## [1] -0.5143191

## [1] "x3"

## [1] -1.528926

## [1] "y3"

## [1] 1.240044

## [1] "x4"

## [1] 4.520661

## [1] "y4"

## [1] 0.6287

## Kurtosis measures the degree of peakness of a distribution. The above kurtosis numbers are derived from the excess kurtosis measure, which provides a comparison of the shape of the distribution to that of the normal distribution. Therefore, a kurtosis greater than 0 is said to be peaked and a kurtosis less than 0 is flat, relative to a standard normal distribution. The kurtosis for the distribution of our data is again close to 0. Series *x1, y1* and *x2, y2* show a slightly flat distribution, whereas series Nos. 3 and 4 show an increasingly peaked shape.

## These descriptive statistics (calculated using R functions summary, skewness and kurtosis) collectively imply that our four datasets are quite comparable.

## Regression models

Regression allows us to explore data by seeing what kind of mathematical models are a best match with the data. An initial round of linear regression yields models consisting of slope and intercept parameters that are the same for all four data sets. In addition, the measures that tell how well the model explains the data are all nearly the same, i.e., R2 is about 0.67 and p is about 0.026.

At first blush one might conclude that the data sets are from the same underlying process. (See discussion of an alternate view, this section below.) However, creating and viewing scatter plots tells a different story.

First, here are the nearly identical results of a strait linear fit with *y* as the dependent variable and *x* as the predictor:

**> summary(xy1)**

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3.0001 1.1247 2.667 0.02573 \*

x1 0.5001 0.1179 4.241 0.00217 \*\*

---

Residual standard error: 1.237 on 9 degrees of freedom

Multiple R-squared: 0.6665, Adjusted R-squared: 0.6295

F-statistic: 17.99 on 1 and 9 DF, p-value: 0.00217

**> summary(xy2)**

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3.001 1.125 2.667 0.02576 \*

x2 0.500 0.118 4.239 0.00218 \*\*

---

Residual standard error: 1.237 on 9 degrees of freedom

Multiple R-squared: 0.6662, Adjusted R-squared: 0.6292

F-statistic: 17.97 on 1 and 9 DF, p-value: 0.002179

**> summary(xy3)**

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3.0025 1.1245 2.670 0.02562 \*

x3 0.4997 0.1179 4.239 0.00218 \*\*

---

Residual standard error: 1.236 on 9 degrees of freedom

Multiple R-squared: 0.6663, Adjusted R-squared: 0.6292

F-statistic: 17.97 on 1 and 9 DF, p-value: 0.002176

**> summary(xy4)**

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 3.0017 1.1239 2.671 0.02559 \*

x4 0.4999 0.1178 4.243 0.00216 \*\*

---

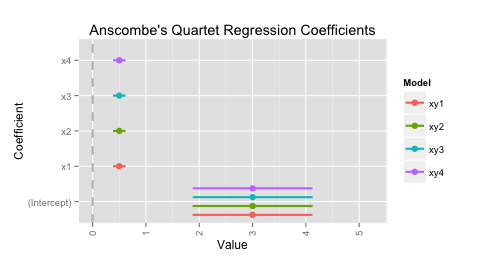
Residual standard error: 1.236 on 9 degrees of freedom

Multiple R-squared: 0.6667, Adjusted R-squared: 0.6297

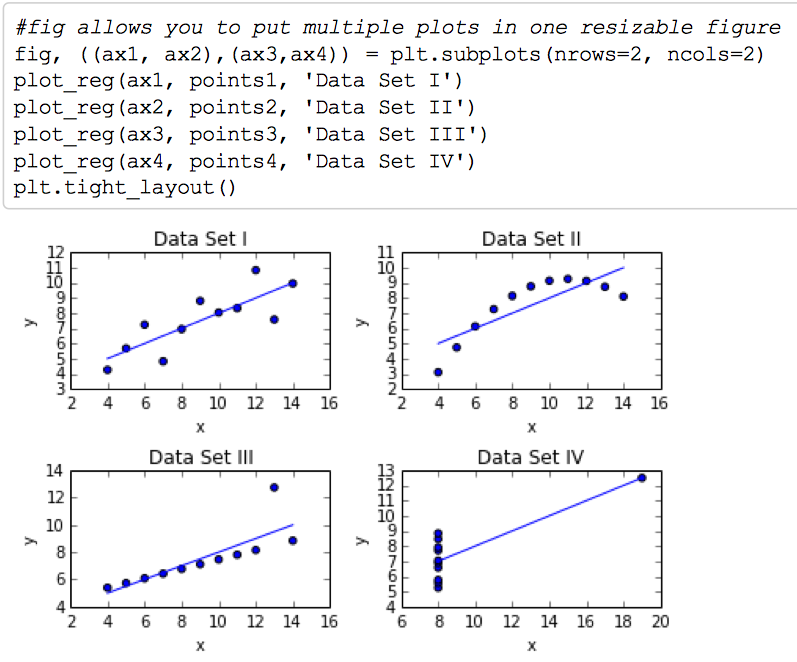
F-statistic: 18 on 1 and 9 DF, p-value: 0.002165

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

The identical coefficients are shown in this chart (R package coefplot, *multiplot()*.):



Scatterplots with a linear fit show that the data sets have a very different character from each other.



This observation suggests that we try an incremental approach with each data set separately to determine what model best fits each data set.

## Data Set I

Data set I (*x1, y1*) showed evidence of a good regression. The points are randomly scattered with no particular pattern in the Residuals vs. Fitted plots. The Normal Q-Q plots indicate a normal distribution since points are basically on the line. Points are grouped together fairly well on both the Scale-Location and Residual vs. Leverage plots.



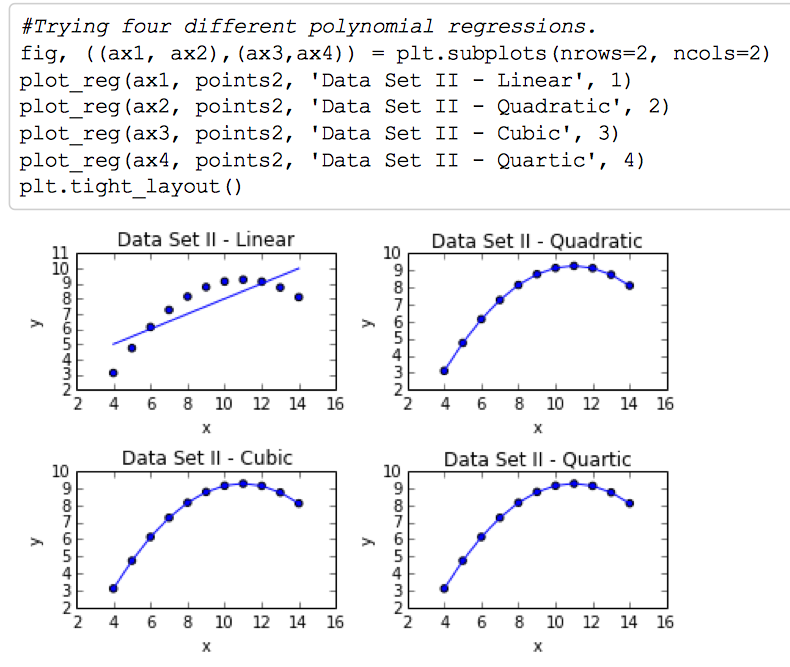
## Data Set II

This data set (*x2, y2*) suggested the opposite of data set I. The Normal Q–Q plot has more points off the line than it does for the good regression above, and both the Scale–Location and Residuals vs. Leverage plots show points scattered away from the center. The Residuals vs. Fitted plot shows a parabolic shape, which suggests that the model is incomplete. Some quadratic factor is missing that could explain more variation in *y*. Evaluating the plots of residuals and their patterns is an important part of analyzing linear regression models.



Visually the linear model does not represent the data well. One could hypothesize that a polynomial model would be a better fit. Indeed, the R2 is improved to 0.9999 from about 0.67. Our hypothesis was correct. The correlation coefficient cannot be improved from a value so close to 1 so there is nothing further we can to with this data set. The best fit model is y = -0.1267x2 + 2.7808x - 5.9957.

Code and Example:



summary(p)

lm(formula=y2~x2+ I(x2^2)+I(x2^3))

Coefficients:

Estimate Std.Error tvalue Pr(>|t|)

Intercept)-6.00e+00 1.37e-02 -436 <2e-16 \*\*\*

x2 2.78e+00 5.24e-03 530 <2e-16 \*\*\*

I(x2^2) -1.27e-01 6.17e-04 -205 1.7e-14 \*\*\*

I(x2^3) -1.65e-17 2.27e-05 0 1

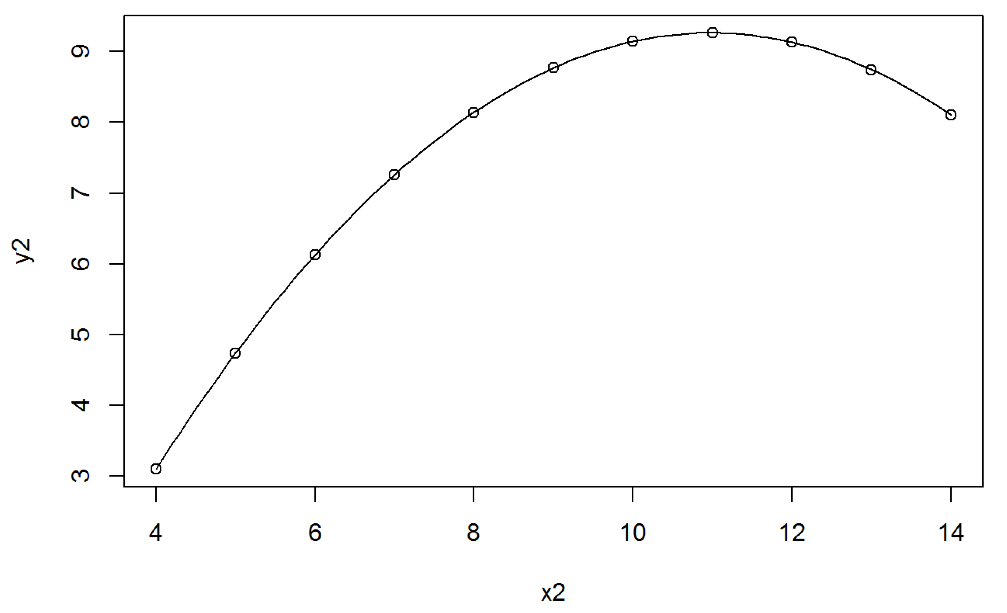
Signif.codes: 0'\*\*\*'0.001 '\*\*'0.01'\*'0.05'.'0.1''1

Residual standard error:0.00179 on 7 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: 1

F-statistic: 4.3e+06 on 3 and 7DF, p-value:<2e-16

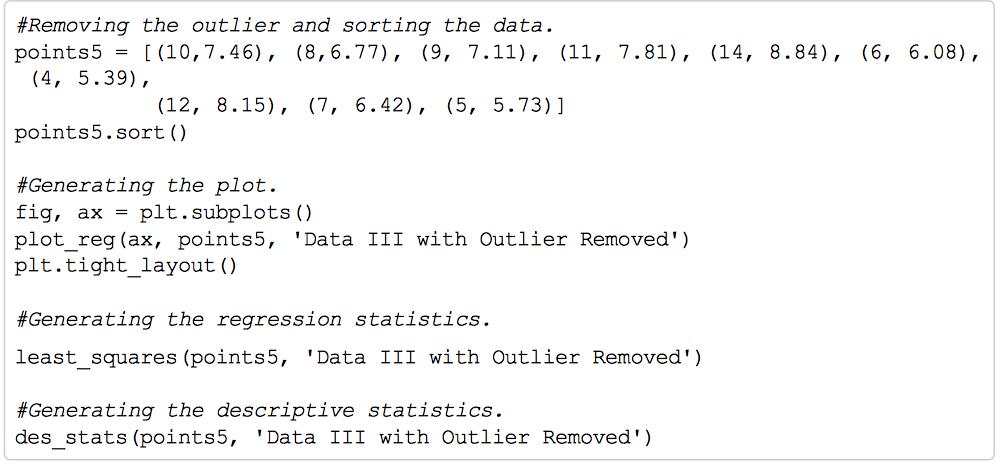
Note that the coefficient for the x3 term is insignificant and is dropped.

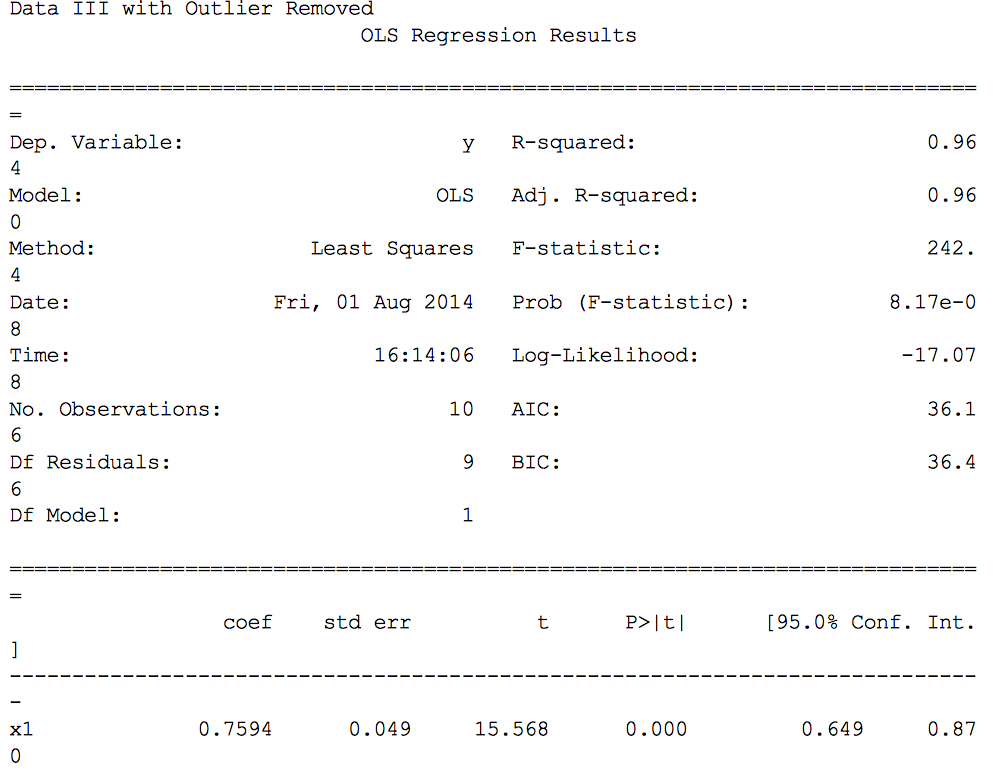


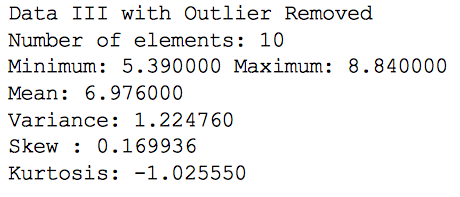
## Data Set III

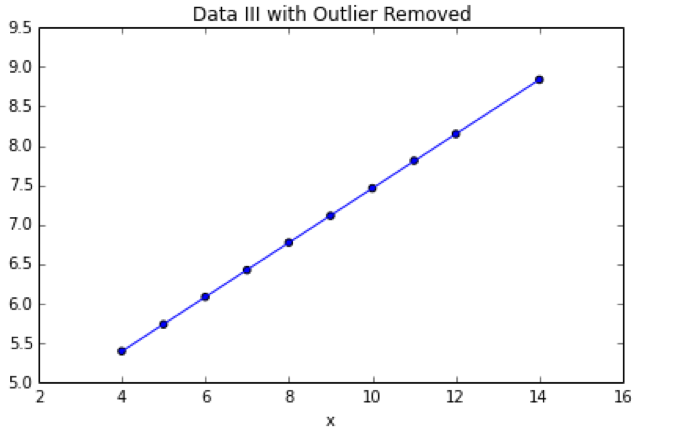
Visually the data in *x, y* No. 3 are tightly in line, except for the point 13, 12.74. The data is like a car traveling in one direction at a constant speed. Our first instinct regarding the outlier is that it's some kind of misreading, like someone wrote down the wrong value. In a real situation, one would want to investigate the data source or discuss with someone who is familiar with how the measurements were taken to try to discern if it is just a mistake, or if further study around that *x* value is warranted. For the sake of the exercise, we assume the outlier was an error and that we can’t redo the measurement. Fitting a linear model to the remaining data after removing the outlier improves R2 to 0.9999. The correlation coefficient cannot be improved from a value so close to 1 so there is no further benefit to trying other models with this data set. The best fit model of the altered data is *y* = 0.3454*x* + 4.0056.

Code and Example:



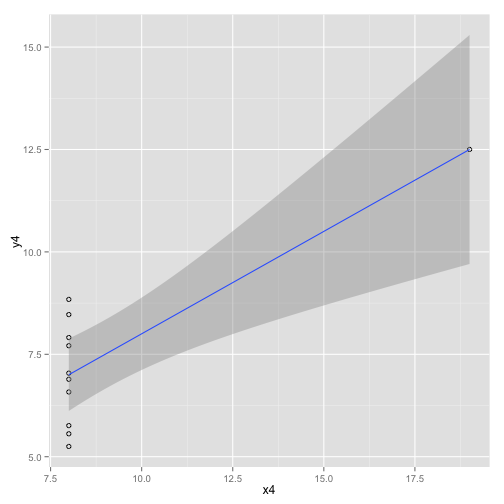






## Data Set IV

Visually there does not seem to be much one can do to understand the relationship of the data. The "outlier" point and one other fall right on the best fit line for data set I.  Does this mean that the person or machine recording the *x* axis values got stuck on the value 8, except for the value of 19?  One would want to investigate back to the source of the data if that were possible.



With what we have to go on one can only speculate, but it is a curious coincidence to have the 8.0, 7.04 and 19, 12.5 data points in such good correlation.  Aside from further investigation of the data source there is nothing more one can do with this data set.

After a linear regression has been modeled, diagnostic checks should be run to verify the model. Finding the observations that are most influencing the model can be useful for diagnosing problems with the model. Influence Measures give several statistics for each observation (Cook's Distance, etc.) and flag observations that are most influential. Those are important to identity because they could be outliers that distort the model.

Deletion of outlier data is a practice that is frowned on by many in the scientific community. Rejection of outliers is more acceptable practice where the underlying model of the process being measured and the usual distribution of measurement error is confidently known. An outlier resulting from an instrument reading error may be excluded, but it is desirable that the reading is verified. In regression problems, an alternative approach may be to only exclude points that exhibit a large degree of influence on the parameters, using a measure such as Cook's distance.

The result of checking influence measures for the data sets in this project follow. There were one or two observations in each data set that could be outliers.

Influence Measures I:

dfb.1\_ dfb.x dffit cov.r cook.d hat inf  
 1 0.00033 3.15e-03 0.0104 1.406 6.14e-05 0.1000   
 2 -0.00818 4.11e-03 -0.0136 1.406 1.04e-04 0.1000   
 3 0.61888 -9.08e-01 -1.1578 0.698 4.89e-01 0.2364   
 4 0.11812 1.40e-17 0.3563 1.037 6.16e-02 0.0909   
 5 0.01197 -2.85e-02 -0.0534 1.443 1.60e-03 0.1273   
 6 0.01618 -2.21e-02 -0.0261 1.856 3.83e-04 0.3182 \*  
 7 0.45415 -3.51e-01 0.5104 1.145 1.27e-01 0.1727   
 8 -0.46907 4.07e-01 -0.4813 1.646 1.23e-01 0.3182   
 9 -0.34342 5.78e-01 0.8400 0.756 2.79e-01 0.1727   
 10 -0.46987 3.20e-01 -0.5990 0.848 1.54e-01 0.1273   
 11 0.08250 -6.84e-02 0.0872 1.647 4.27e-03 0.2364

Influence Measures II:

dfb.1\_ dfb.x dffit cov.r cook.d hat inf  
 1 0.0102 9.72e-02 0.3223 1.127 0.05233 0.1000   
 2 0.1936 -9.72e-02 0.3223 1.127 0.05233 0.1000   
 3 0.2030 -2.98e-01 -0.3798 1.480 0.07666 0.2364   
 4 0.1139 9.91e-18 0.3436 1.057 0.05787 0.0909   
 5 -0.0543 1.30e-01 0.2423 1.315 0.03145 0.1273   
 6 0.9475 -1.29e+00 -1.5278 0.703 0.80787 0.3182 \*  
 7 0.0440 -3.40e-02 0.0495 1.525 0.00137 0.1727   
 8 -1.4889 1.29e+00 -1.5278 0.703 0.80787 0.3182 \*  
 9 -0.0202 3.40e-02 0.0495 1.525 0.00137 0.1727   
 10 0.1901 -1.30e-01 0.2423 1.315 0.03145 0.1273   
 11 -0.3591 2.98e-01 -0.3798 1.480 0.07666 0.2364

Influence Measures III:

dfb.1\_ dfb.x dffit cov.r cook.d hat inf  
 1 -4.63e-03 -4.41e-02 -0.1464 1.34e+00 0.011765 0.1000   
 2 -3.71e-02 1.86e-02 -0.0618 1.39e+00 0.002141 0.1000   
 3 -3.58e+02 5.25e+02 669.5875 5.06e-11 1.392849 0.2364 \*  
 4 -3.29e-02 -3.37e-18 -0.0992 1.36e+00 0.005473 0.0909   
 5 4.92e-02 -1.17e-01 -0.2193 1.34e+00 0.025984 0.1273   
 6 4.90e-01 -6.67e-01 -0.7897 1.36e+00 0.300571 0.3182   
 7 2.70e-02 -2.09e-02 0.0303 1.53e+00 0.000518 0.1727   
 8 2.41e-01 -2.09e-01 0.2472 1.80e+00 0.033817 0.3182 \*  
 9 1.37e-01 -2.31e-01 -0.3362 1.34e+00 0.059536 0.1727   
 10 -1.97e-02 1.34e-02 -0.0251 1.45e+00 0.000355 0.1273   
 11 1.05e-01 -8.74e-02 0.1114 1.64e+00 0.006948 0.2364

Influence Measures IV:

dfb.1\_ dfb.x dffit cov.r cook.d hat inf  
 1 -0.06827 0.03428 -0.1137 1.366 7.17e-03 0.1   
 2 -0.21353 0.10721 -0.3556 1.078 6.23e-02 0.1   
 3 0.11654 -0.05851 0.1941 1.294 2.03e-02 0.1   
 4 0.34734 -0.17439 0.5784 0.742 1.37e-01 0.1   
 5 0.26029 -0.13069 0.4334 0.958 8.72e-02 0.1   
 6 0.00628 -0.00315 0.0105 1.406 6.15e-05 0.1   
 7 -0.32505 0.16320 -0.5413 0.795 1.24e-01 0.1   
 8 0.00000 0.00000 NaN NaN NaN 1.0 \*  
 9 -0.25432 0.12769 -0.4235 0.974 8.39e-02 0.1   
 10 0.15149 -0.07606 0.2523 1.225 3.34e-02 0.1   
 11 -0.01788 0.00898 -0.0298 1.403 4.98e-04 0.1

Another useful test is the Durbin-Watson test which checks the residuals for autocorrelation. The output of this test in R includes a p-value. Typically, only a very small p-value (less than 0.05) would indicate significant correlation in the residuals.

The results of running these data sets through the Durbin-Watson test are shown here. The first data set had a very high p-value indicating no correlation in the residuals. That p-value was smaller in the other data sets, but not small enough to suggest significant autocorrelation in any of the data sets.

Durbin-Watson Test I:  
   
 DW = 3.212, p-value = 0.988  
 alternative hypothesis: true autocorrelation is greater than 0

Durbin-Watson Test II:

DW = 2.188, p-value = 0.6299  
 alternative hypothesis: true autocorrelation is greater than 0

Durbin-Watson Test III:

DW = 2.144, p-value = 0.6016  
 alternative hypothesis: true autocorrelation is greater than 0

Durbin-Watson Test IV:

DW = 1.662, p-value = 0.2892  
 alternative hypothesis: true autocorrelation is greater than 0

## Basically, a simple linear regression model is one way to start. That model should be tested though, and alternative models should be considered. Analysis must verify that the model is a good fit, and that the assumptions behind the linear regression model are satisfied.

## Alternate theories

Another path of inquiry assumed the separate data sets were related. Data points were combined and analyzed with a multivariate model using *x* as the dependent variable.

Looking at all of the data as a single *x, y* linear model gives a fair p value (.05 on two of three predictors per below). This is better than the individual models, even with dataset No. 4 and its outlier. If we use this as a model, the outliers aren't so impactful, which can be useful.  Given the small sample size it is really difficult to know if the outliers are really outliers, so combining the data sets provides a more inclusive model.

Since the *x* variable in the first three data sets is identical (*x1* = *x2* = *x3*) the data could be interpreted as 11 observations where the *y1, y2, y3* and *y4* are supposed to be attributes of *x*.  Since the *x4* is the odd one out, we could make the assumption that it represents bad data, and see if we can model *x* based on the remaining *y1, y2* and *y3*.  A basic linear model of (*x ~ y1+y2+y3*) results in an interestingly nice fit, with a good R-squared (0.89) and p-value, indicating at least the sum of those attributes could be a good predictor of x.

## > fitcheryl <- with(cuny9, lm(x1~y1+y2+y3))

## > summary(fitcheryl)

## Call:

## lm(formula = x1 ~ y1 + y2 + y3)

## Residuals:

## Min 1Q Median 3Q Max

## -1.2211 -0.5633 -0.2243 0.5134 1.9733

## Coefficients:

## Estimate Std. Error t value Pr(>|t|)

## (Intercept) -4.8317 1.5579 -3.101 0.0173 \*

## y1 0.7108 0.2611 2.723 0.0297 \*

## y2 0.3237 0.2851 1.136 0.2935

## y3 0.8096 0.2136 3.790 0.0068 \*\*

## ---

## Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

## Residual standard error: 1.108 on 7 degrees of freedom

## Multiple R-squared: 0.9219, Adjusted R-squared: 0.8884

## F-statistic: 27.54 on 3 and 7 DF, p-value: 0.0003005

## V. Comments on Applications of R vs. Python

These comments are limited to applications of R and Python to our specific assignment and are not intended to cover an overall functionality available in R and Python for the purposes of exploratory data analysis.

## Data Load

Both R and Python provide functionality for reading a comma-separated file. R provides a native method that loads data from a *.csv* file into a data frame. Python does not provide methods for loading data into a data frame out of the box (instead a *.csv* file can be opened for a sequential reading). However with the use of an open source libraries *pandas* and *numpy* we can load a *.csv* file directly into a data frame in Python.

|  |  |  |
| --- | --- | --- |
|  | **R Example** | **Python Example** |
| Load data | cuny9 <- read.table  ('C:/CUNY/cuny9.csv',header=TRUE, sep=",") | import pandas as pd  import numpy as np  cuny9 = pd.read\_csv('C:/CUNY/cuny9.csv') |
| Access column x1 | cuny9[,'x1'] | cuny9.x1 |

## Basic Summary Statistics

R, as the language designed for statistical analysis, provides an extensive native set of basic summary statistics capabilities. Basic summary statistics for all columns of a data set can be generated with one command *summary()*. Python has very limited number of stat functions out of the box. In order to access more extensive functionality we need to load package *scipy* and use sub-package *stats*. Still in Python every element of summary statistics should be accessed/printed separately via procedural code.

|  |  |  |
| --- | --- | --- |
|  | **R Example** | **Python Example** |
| Descriptive Statistics | summary(cuny9) | import scipy as sp  from scipy import stats  n, min\_max, mean, var, skew, kurt = stats.describe(cuny9.y1)  print("Minimum: {0:8.6f} Maximum: {1:8.6f}".format(min\_max[0], min\_max[1]))  print("Mean: {0:8.6f}".format(mean)) |

## Basic Plots

Basic plots can be created in R using *plot()* function. Python does not provide similar functionality out of the box.

|  |  |  |
| --- | --- | --- |
|  | **R Example** | **Python Example** |
| Basic Plots | x1 <- cuny9[,'x1']  y1 <- cuny9[,'y1']  plot(y1~x1) | Not available |

## Advanced Plots

Advanced plotting functionality is available in both R and Python through external packages. In R – package *ggplot2*. In Python – package *matplotlib*.

Package *ggplot2* provides two main functions:

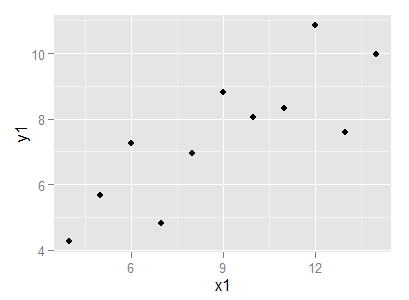
* *qplot* – for quick plots without extensive customization
* *ggplot* – allows to customize plot appearance. Function *ggplot* can be used with a number of parameters in the following categories:
  + type of plot
  + aesthetics (mapping of variables)
  + scales (mapping between data and aesthetics)
  + coordinate systems
  + faceting
  + position adjustments
  + annotation
  + themes

|  |  |  |
| --- | --- | --- |
|  | **R Example** | **Python Example** |
| Basic scatter plot | library("ggplot2")  x1 <- cuny9[,'x1']  y1 <- cuny9[,'y1']  qplot(x1,y1) | import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  %matplotlib inline  x = cuny9.x1  y = cuny9.y1  cuny9.plot(x ='x1',y ='y1',kind = 'scatter') |
| Plot with regression line | ggplot(cuny9,aes(y=y1,x=x1))+  geom\_smooth(method='lm') | m,b = np.polyfit(x,y,1)  plt.plot(x,y,'yo',x,m\*x+b, '--k')  plt.xlabel("x1")  plt.ylabel("y1") |
| Box Plot | ggplot(cuny9, aes(y=y1,x=x1))+geom\_boxplot() | fig = plt.figure()  ax = fig.add\_subplot(111)  ax.boxplot([y]) |

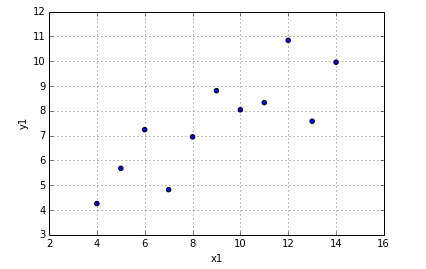
## Appearance of Graphs

If we compare similar graphs generated by ggplot2 and matplotlib, the difference is not very significant, but ggplot2 graphs have a more professional and ready-to-be-published look.

Graph generated by *ggplot2*:



Graph generated by *matplotlib*:



## Bottom line: R vs. Python

As a language created specifically for statistical analysis, R outperforms Python in areas of data manipulation and exploratory data analysis. However with an introduction of *matplotlib* library Python developers received a powerful graphing tool that is comparable to R library *ggplot2* and allows Python developers to produce highly customizable graphs without leaving Python environment.

## VI. Overall Conclusions

With nearly identical means, sample variances, correlation coefficients and linear regression lines, the datasets of Anscombe’s Quartet demonstrate the significance of graphically visualizing data, as well as the potentially large influence on the data of one or two significant outliers. Using R and Python code, our team was able to analyze both the similarities and differences of the datasets in this quartet. The primary takeaway of this exercise is the idea that representing data graphically is at least as significant as running layers of strictly quantitative analysis. A second conclusion is that linear regression is not always good for summarizing data.

Although it’s true that we “discovered” the origin of this dataset early in the week, it’s one thing to read over the analysis that’s been done by others before us and another to “get our hands dirty” in the data and look at these numbers for ourselves from several different perspectives. The brilliance of these four datasets is that they get across a fundamental reality about data analysis in a simple and elegant manner. With a few lines of code, a computer can tell us that these datasets have many attributes in common (mean of both x and y, sample variance, linear regression lines etc.). However, when graphed, we humans immediately recognize that these four datasets are far from the same.

What we take away by graphing this data turns out to be far more telling with regard to the relationship between *x* and *y* than any summary statistics. This underscores something that we’ve all known since day one, namely that when it comes to analysis, the computer is still just a tool. When we as data scientists go through lots of layers of analysis through R, Python or any means of modeling this data, that’s when we best get a feel for what a computer can and can’t tell us.

For example, it’s immediately obvious when visualized that *x, y* pair 1 provides more evidence of a relationship (linear) between two variables than *x, y* pair 4, where all *x* variables have the same value except one. As professor Anscombe suggested, if the outlier in *x, y* pair No. 4 is a legitimate sample and not an error, we can still include the outlier in an analysis but must emphasize its heavy contribution. Of course, Anscombe’s point in designing these datasets was to demonstrate that representing data graphically is as important, perhaps more in some cases, as running layer upon layer of strictly quantitative analysis. (Though ironically, it’s only by running these layers of quantitative analysis that we can see this to be true.)

## Alternate views

One alternate way to view the data that could invite further exploration is to consider *x, y* pairs No. 1, 2 and 3 separate measurements from the same experiment and the No. 4 pair from a different experiment.

Still another view postulates that this is real data from some unknown process. As such, it may be useful to investigate a possible relationship among the different data sets. This exercise showed that a modeling the *x* data using *y* values from datasets No. 1, 2, and 3 as predictor variables yields a result with acceptable significance (F=27.54) and a high adjusted R-squared of 0.88.

1. “Graphs in Statistical Analysis”, Anscombe, F.J., The American Statistician, February 1973, Vol. 27, No. 1. at <https://www.coursesites.com/courses/1/IS-02/db/_3187926_1/anscombe1973.pdf> [↑](#endnote-ref-1)
2. “Noted Statistician Francis Anscombe Dies,” Yale Bulletin & Calendar, http://www.yale.edu/opa/arc-ybc/v30.n9/whitespace.gifNov. 2, 2001http://www.yale.edu/opa/arc-ybc/v30.n9/whitespace.gifVolume 30, Number 9, at <http://www.yale.edu/opa/arc-ybc/v30.n9/story11.html> [↑](#endnote-ref-2)