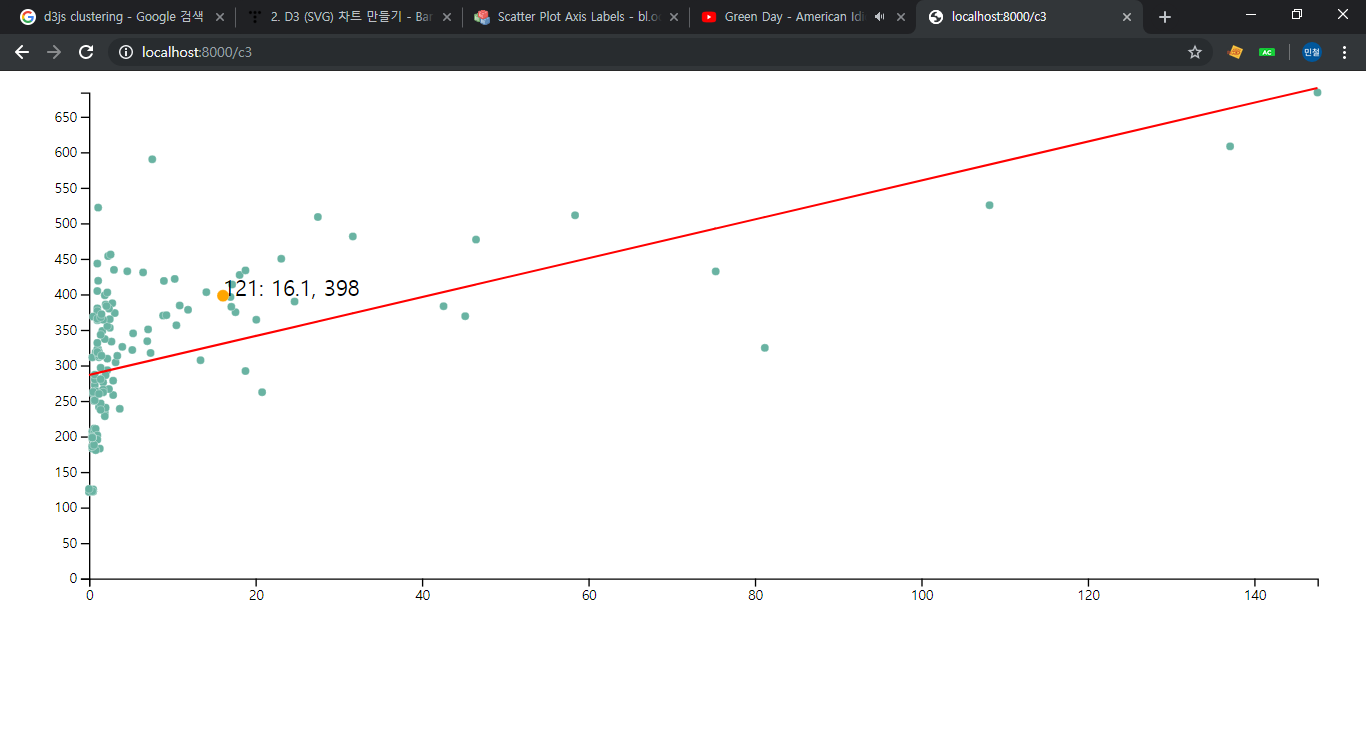
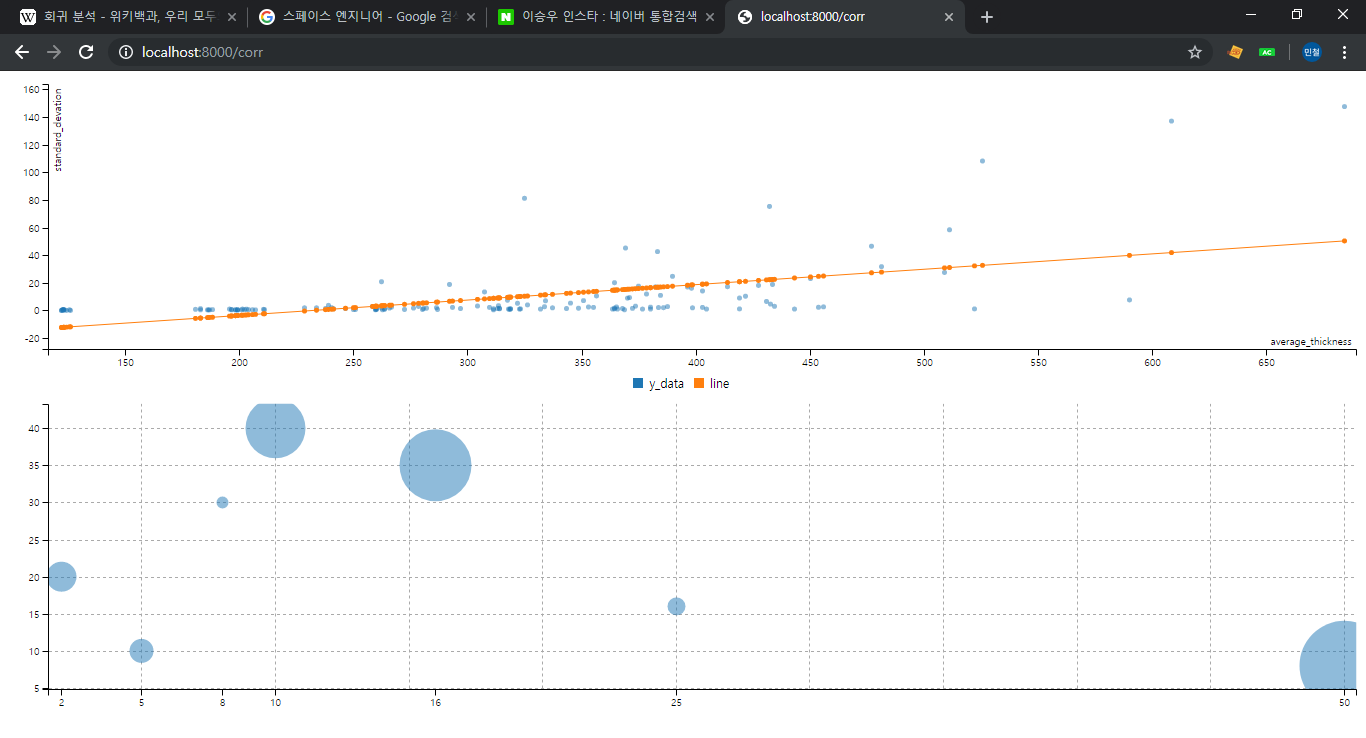
**몽고DB와 연결하여 구현한 결과**

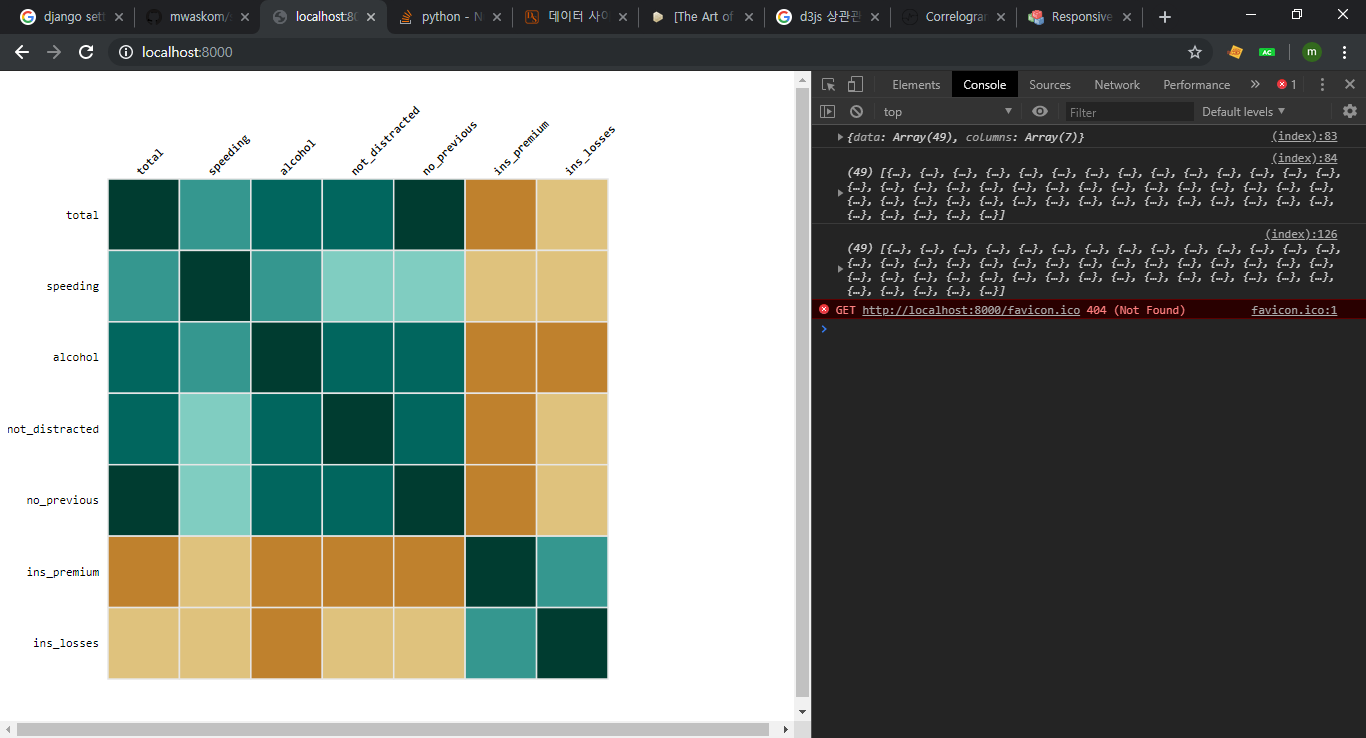
1. D3를 이용한 그래프



1. C3를 이용한 그래프



1. 히트맵 ( 실제 데이터 적용 X )



1. Mongo와 연결 코드
2. Mongo\_connect 클래스 정의

import pymongo as pm

class mongodb:

    def \_\_init\_\_(self):

        self.conn = pm.MongoClient('localhost', 27017)

    def set\_db(self, dbname):

        self.db = self.conn.get\_database(dbname)

        return self.db

    def set\_collection(self, collection\_name):

        self.collection = self.db.get\_collection(collection\_name)

        return self.collection

    def get\_collection(self):

        try:

            return self.collection

        except:

            print("you need to use set\_collection before use get\_collection")

            return None

1. C3 코드

# api/corr\_data

# c3 data type

def corr\_data(request):

    # connect to mongodb

    mc = mongo\_connect.mongodb()

    mc.set\_db('ceramicdb')

    mc.set\_collection('ceramic')

    col = mc.get\_collection()

    # db query

    result = col.find({})

    # average\_thickness, standard\_deviation

    thickness = []

    standard = []

    index = [] # ceramic number

    for r in result:

        try:

            c = float(r['average\_thickness'])

            s = float(r['standard\_deviation'])

            i = int(r['index'])

            thickness.append(c)

            standard.append(s)

            index.append(i)

        except:

            # if query value is unuseable, don't append value

            continue

    # first value must be label name

    x\_data = ['x\_data']

    y\_data = ['y\_data']

    # trend line

    graph = makegraph(thickness, standard)

    line = ['line']

    for g in graph:

        line.append(g['yhat'])

    # make context data

    x\_label = 'average\_thickness'

    y\_label = 'standard\_devation'

    x\_data.extend(thickness)

    y\_data.extend(standard)

    context = {

        "x\_data" : x\_data,

        "y\_data" : y\_data,

        "x\_label" : x\_label,

        "y\_label" : y\_label,

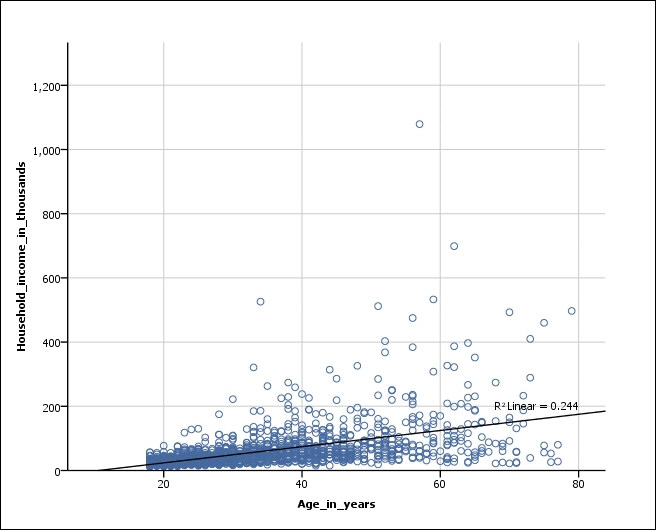
        "line" : line,

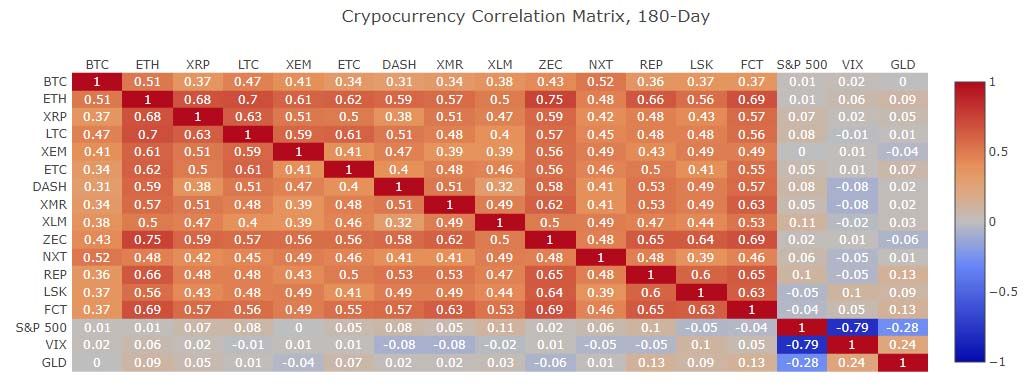
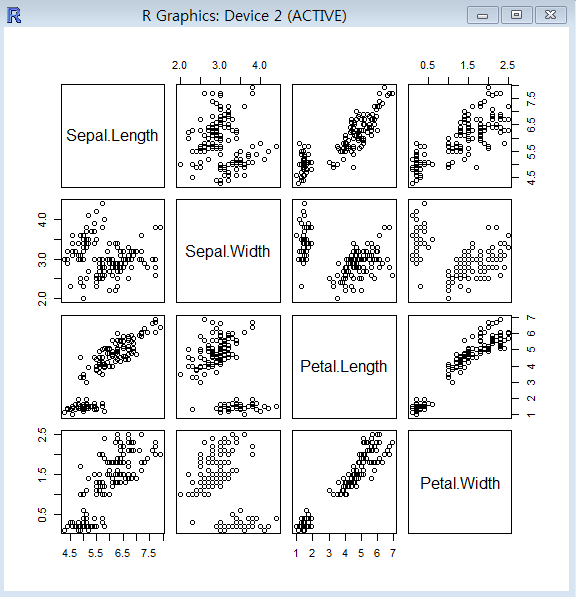
    }

    return JsonResponse(context)

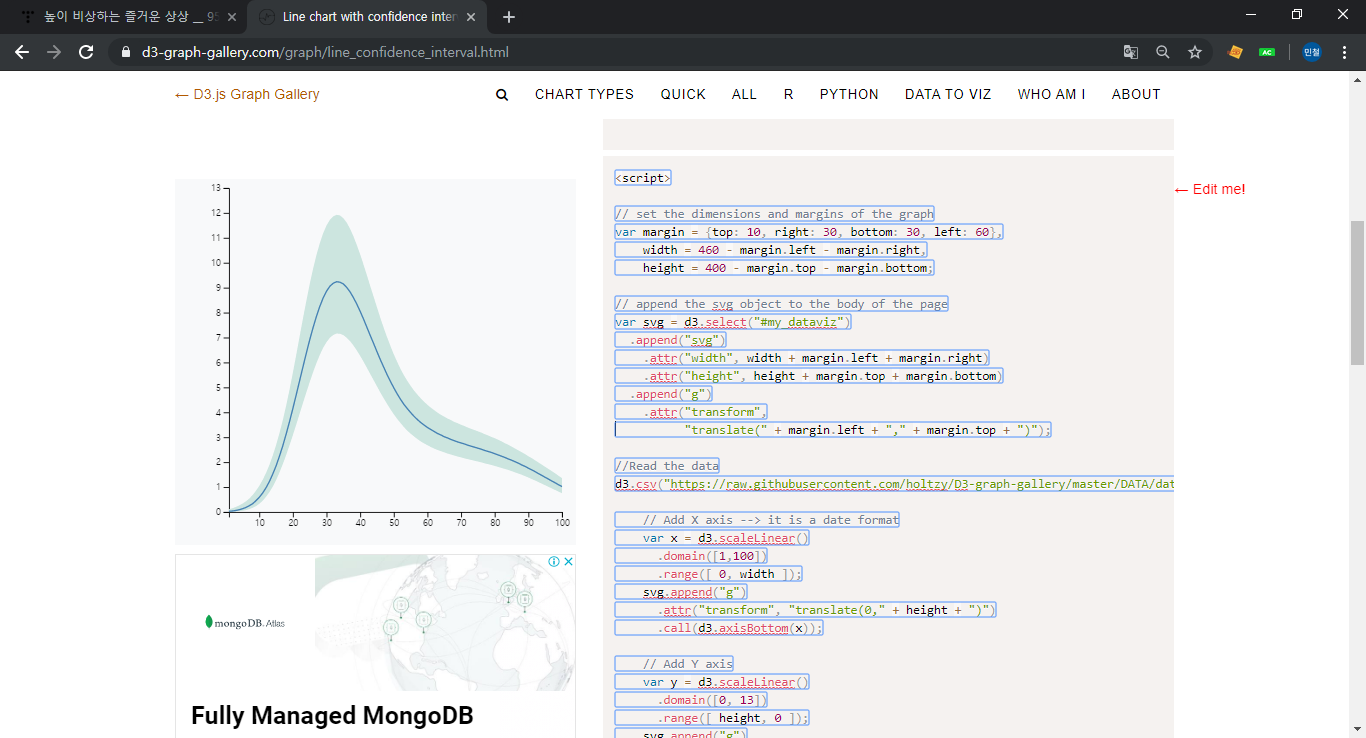
**시각화 자료조사**

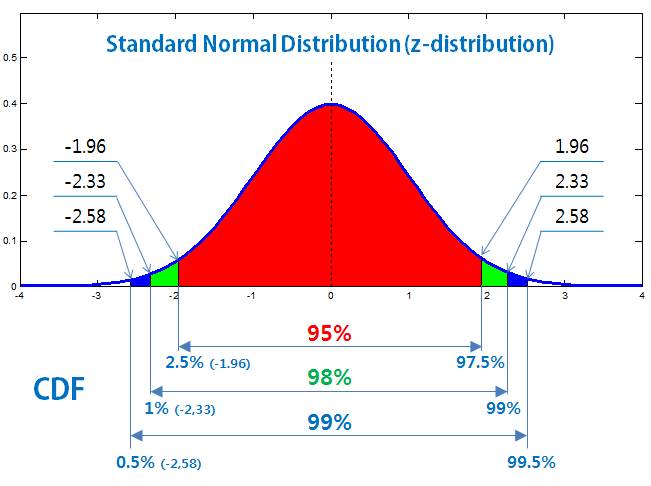
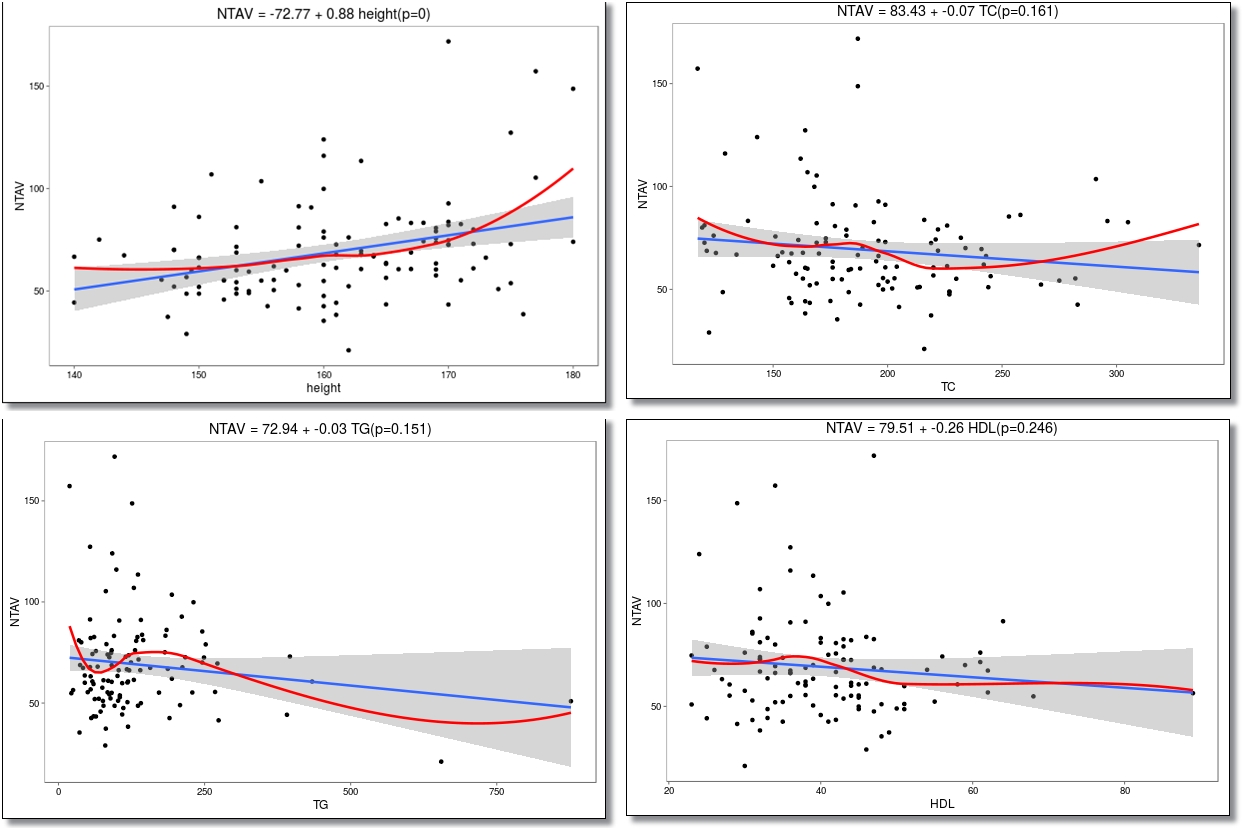
**상관 관계**

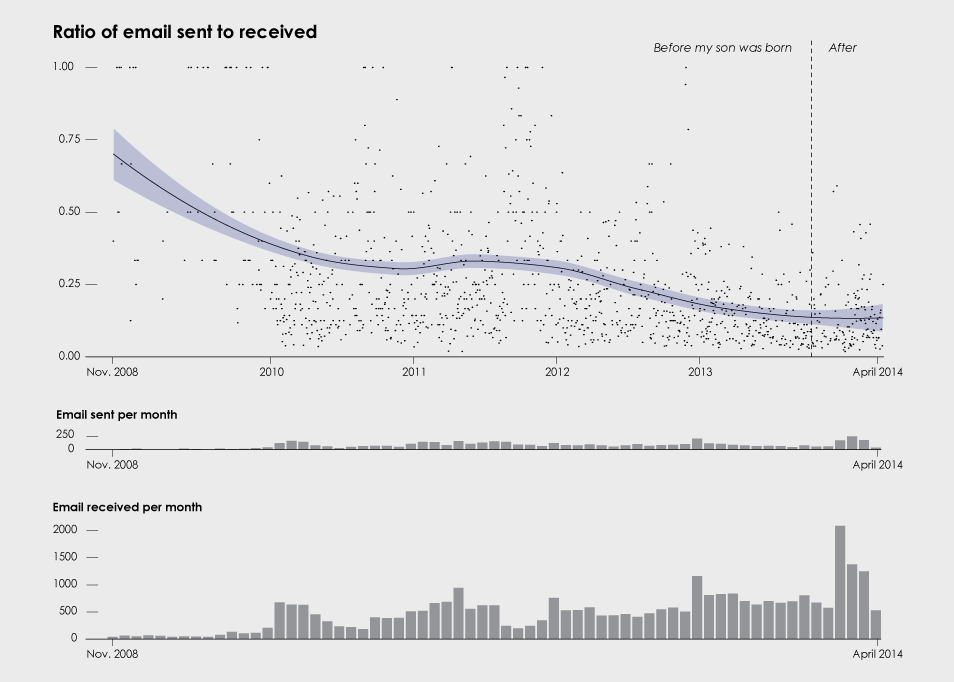
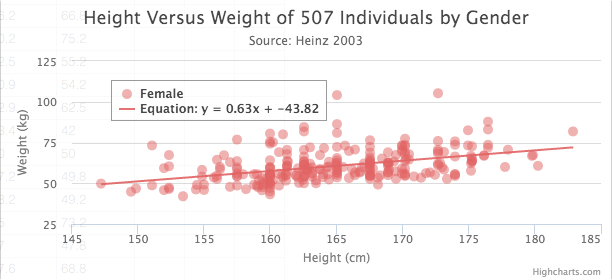




**회귀 분석, 신뢰구간 시각화**







**회귀 분석**

관찰된 연속형 변수들에 대해 두 변수 사이의 모형을 구한 뒤 적합도를 측정해내는 방법

두 변수가 선형의 관계를 이루는가?

한 종속 변수와 한 독립 변수 사이의 관계를 분석할 경우 단순회귀분석  
한 종속 변수와 다수의 독립 변수 사이의 관계를 분석할 경우 다중회귀분석

독립 변수 : 입력 값  
종속 변수 : 출력 값

회귀 분석 표준 가정

1. 오차항은 모든 독립변수에 관하여 동일한 분산을 갖는다.
2. 오차항의 평균은 0이다.
3. 수집된 데이터의 확률 분포는 정규분포를 이루고 있다.
4. 독립변수 상호간에는 상관관계가 없어야 한다.
5. 시간에 따라 수집한 데이터들은 잡음의 영향을 받지 않아야 한다.

**신뢰구간**

모평균을 포함할 확률이 95%가 되는 구간

추정치들의 모평균에 대한 신뢰구간을 구함으로써 신뢰성을 어느정도 측정할 수 있게 하는 것