Chapter 14 Multiple Regression analysis

앞장에서 다룬 simple regression 은 하나의 independent 를 다루었고, 이 장에서는 여러개의 independent variable 로 하나의 dependent variable 의 변화를 예측하는 regression 에 대해 학습을 할 것입니다. 여기서 배우는 내용은 linear regression 을 위한 일반화된 방법으로 simple regression 에도 적용이 됩니다.

1. Multiple Regression Model

$$\hat{y} = b_0 + \sum_{i=1}^{k} b_i x_i \tag{14.1}$$

이 수식을 Matrix 로 표현을 하면 다음과 같습니다.

$$\begin{bmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{bmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_k \end{bmatrix}$$

여기서

x_{nk} 값은 n 번째 sample 의 k 번째 independent variable 의 값

b_k 는 regression line 에 k 번째 x 값만 변하고 나머지 x 값은 상수일 때 예측값 $\hat{y}$평균 변화값

b 값들은 다음과 같은 Matrix 연산을 통해서 얻을 수 있습니다.

$$\begin{gathered} Y = XB \\ X^T Y = X^T X B \\ B = (X^T X)^{-1} X^T Y \end{gathered} \tag{14.2}$$

여기서 X 에 transpose 를 사용하는 이유는 inverse matrix 를 계산하기 위해서는 해당 matrix 에 row 와 column 의 크기가 같은 square 이어야 되기 때문에 X transpose (Nx1)와 X(1xN)의 곱(NxN)으로 inverse 가 가능하도록 만들기 위해서 입니다.

예를 들어 13 장 3.1 절에 연료 소비량으로 예상 이동거리를 예측하기 위해서 b_0 와 b_1 을 계산해야 합니다. 수식 14.2 에 적용하여 이 값을 계산을 하는 것을 jMath 로 알아 보겠습니다.

$$X = \begin{bmatrix} 1 & 16 \\ 1 & 14 \\ 1 & 5 \\ 1 & 12 \\ \vdots & \vdots \\ 1 & 17 \end{bmatrix}, Y = \begin{bmatrix} 179.3353 \\ 168.2739 \\ 52.4114 \\ 127.9184 \\ \vdots \\ 200.0185 \end{bmatrix}$$

X, Y 값을 이용해서 계산한 결과를 알아 보겠습니다.

```
var X = jMath('16;14;5;12;8;21;11;16;8;17');
var Y = jMath('179.3353;168.2739; 52.4114; 127.9184; 117.682; 310.394; 141.6365;
164.7824; 89.3912; 200.0185');

X = jMath.ones(Y.rows, 1)[':='](X);
XT = X.transpose();

var XXTinv = XT['*'](X).inv();
var XXTinv_XT = XXTinv['*'](XT);
var B = XXTinv_XT['*'](Y);

console.log(B.toString());

-17.072882352941217
13.457597058823534
```

Matrix B 값을 보시게 되면 13 장에서 나타난 결과와 동일 함을 알 수 있습니다.

$$\hat{y} = 13.4576x - 17.0729$$

jMath 예제에서 계산 과정을 하나씩 보여드리기 위해서 중간 결과값들을 별도의 변수에 저장을 했는데 변수없이 한 줄로 표현을 하면 다음과 같습니다.

```
XT['*'](X).inv()['*'](XT)['*'](Y)
```

이렇게 얻어진 Matrix B 와 Matrix X 로 예상값을 계산할 수 있게 됩니다.

2. ANOVA 해석

Simple Regression 에서 사용된 방식과 동일한 방식을 적용하면 됩니다.

	df	SS	MS	F(k,n-k-1)
SSR	k	$\sum_{i=1}^{n}(\hat{y}_i - \overline{y})^2$	$\frac{SSR}{k}$	$\frac{MSR}{MSE}$
SSE	n-k-1	$\sum_{i=1}^{n}(y_i - \hat{y})^2$	$\frac{SSE}{n-k-1}$	
SST	n-1	$SSR + SSE = \sum_{i=1}^{n}(y_i - \overline{y})^2$		

(14.3)

SSR: Sum of Squares Regression,

SSE: Sum of Squares Error,

SST: Sum of Squares

MSR: Mean of Square Regression,

MSE: Mean of Square Error

k: Independent variable 들의 개수

n: Regression 계산에 사용된 총 sample 개수

F test 로 테스트 하는 항목은 dependent 와 independent variable 사이에 관계가 있는가 없는가를 나타내는 것입니다.

$$H_0: \beta_1 = \beta_2 = \cdots \beta_k = 0$$
$$H_1: \text{적어도 한개의 } \beta_i \neq 0$$

여기서 β_i는 population 으로 만들어진 regression 에 coefficient 값입니다.

Multiple Coefficient of determination

$$R^2 = \frac{SSR}{SST} \tag{14.4}$$

이 값의 백분률의 의미는 dependent variable 의 변화는 사용된 independent variable 들로 설명될 수 있는 확률로 이 값이 1 에 가까울 수록 independent variable 들은 예측을 거의 정확하게 한다는 의미입니다.

Adjusted Multiple Coefficient of Determination

Regression 수식에서 independent variable 의 추가는 R^2을 항상 증가 시키기 때문에 실제로 추가된 변수에 의미를 R^2로 해석하기에는 문제가 있습니다. 그래서 이러한 문제를 해결하기 위해서 R^2값을 수정한 방식을 적용해야 합니다.

$$R_A^2 = 1 - \frac{MSE}{MST} = 1 - \frac{SSE}{SST}\frac{n-1}{n-k-1} = 1 - (1-R^2)\frac{n-1}{n-k-1} \tag{14.5}$$

이 수식에서 보듯이 multiple regression model 에서 independent variable 의 개수 및 sample 의 크기의 의해 R^2 값을 조정을 하는 것입니다.
이 값의 용도는 multiple regression 을 만들기 위해 새로운 independent variable 을 추가한 결과 R^2는 증가하겠지만 R_A^2가 감소를 한다면 새로 추가된 변수를 포함 하지 않는 것이 좋습니다.

회사에서 신입사원들이 업무 평가 점수를 신입 사원의 학업성적과 영어성적으로 예측을 하려고 합니다. 성적은 100 점 만점으로 환산된 값입니다.

업무평가점수	학업 성적	영어 성적
97	88	98
77	79	82
89	95	81
91	89	91
92	86	96

82	81	90
75	83	83
85	72	82
93	82	88
88	98	89
92	70	99
82	71	92
80	83	87
91	90	88
96	73	96
79	83	91
79	79	81
100	71	96
82	86	89

우선 영어 성적과 업무 평가 점수만으로 regression 분석을 하겠습니다.
jMath 에서 jMath.regress 을 이용하여 게산을 하면 다음과 같음 결과를 얻게 됩니다.

```
jMath.regress(X, Y, alpha, extra )
```

X 값은 regression 에서 사용될 independent variable 들에 값으로 크기는

"sample 개수 x 변수 개수"

입니다. Extra 는 regression 분석시 예측값의 CI 를 알아 볼 때 측정을 원하는 입력값 혹은 분석시 사용된 X 를 이용할 것인가를 나타내는 값입니다. 이 값이 없으면 이러한 결과를 얻지 못합니다.

이 함수를 실행을 하면 다양한 결과를 보실 수 있지만 이번 예제에서는 지금까지 학습한 내용만을 설명하겠습니다.

```
var eng = jMath('98;82;81;91;96;90;83;82;88;89;99;92;87;88;96;91;81;96;89');
var gpa = jMath('88;79;95;89;86;81;83;72;82;98;70;71;83;90;73;83;79;71;86');
var score = jMath('97;77;89;91;92;82;75;85;93;88;92;82;80;91;96;79;79;100;82');
var enggpa = eng.clone()[':='](gpa);

result1 = jMath.regress( eng, score );
console.log(result1);

result2 = jMath.regress( enggpa, score );
console.log(result2);
```

Result1 은 영어 성적과 업무평가 점수만을 갖고 판단했을 경우이고, result2 는 영어와 학업 성적을 동시에 사용해서 업무평가 점수를 얻은 것입니다.

Result 1

	df(result1.df)	SS(result1.anova.SS*)	MS(result1.anova.MS*)	F(1,17) = 4.45132
SSR	1	426.392115	426.3921148	13.272682682
SSE	17	546.134201	32.1255412	
SST	18	972.526316	54.0292398	

R^2(result1.r2.value) = 0.4384376112676308, R^2_A(result.r2.adjust) = 0.4054045295774915

$$\hat{y} = 12.119 + 0.8356x_e$$

Result 2

	df(result2.df)	SS(result2.anova.SS*)	MS(result2.anova.MS*)	F(2,16) = 3.63372
SSR	2	447.263555	223.6317776	6.812035252
SSE	16	525.262761	32.8289225	
SST	18	972.526316	54.0292398	

R^2(result1.r2.value) = 0.45989866590798734, R^2_A(result.r2.adjust) = 0.39238599914648575

$$\hat{y} = -3.051 + 0.8805x_e + 0.136x_g$$

결과를 보시면 두 결과 다 F test statistic 값들이 F critical value 들보다 크기 때문에 모든 β값이 0 이 되지 않습니다. 하지만 multiple coefficient R^2 값들을 보시면 학업성적을 regression 분석에 반영하므로써 값이 0.43 에서 0.45 로 증가하는 것을 볼 수 있지만 adjusted multiple coefficient R^2_A 값들은 0.405 에서 0.392 로 감소하는 것을 알 수 있습니다.

즉 학업 성적은 업무 성적을 예측하는데 사용되지 않는 것이 좋은 결과를 나타남을 알 수 있습니다.

3. Confidence Interval

각 coefficient 값에 대한 Hypothesis test 와 confidence interval 을 계산하는 과정은 다음과 같습니다.

1) Null hypothesis 설정

$$H_0: \beta_i = 0$$
$$H_1: \beta_i \neq 0$$

2) Significance level: 0.05

3) Test statistic

$$t_i = \frac{b_i - \beta_i}{s_{b_i}} = \frac{b_i}{s_{b_i}} \quad (14.6)$$

여기서 i 는 independent variable 이 k 개 있을 때 0 부터 k 까지 입니다.
β_i값은 null hypothesis 에서 0 이기 때문에 수식 14.6 에서 제거가 됩니다. 그럼 여기서 coefficient 의 standard error 값을 계산하는 방식을 설명하겠습니다.

$$S_b = \sqrt{diag(\text{MSE} \times (X^T X)^{-1})} = \sqrt{diag(C)} \quad (14.7)$$

Diag 는 Matrix 에 diagonal 값들을 추출하는 함수이고 square root 는 matrix 에 element 에 개별적으로 적용되는 것입니다.

4) Critical Value

Student t-distribution 을 이용하므로 degree of freedom 을 알아야 합니다.

$$Degree\ of\ Freedom = n - k - 1$$

n 은 sample 크기
k 는 independent variable 의 개수

이 값을 적용하여 test statistic 값이 null hypothesis 를 reject 하지 못하는 영역을 계산하면 됩니다.

5) Null hypothesis reject

만일 coefficient 들 중에 test statistic 값이 critical value 의 영역에 있다면 즉 p-value 가 significance level 보다 크다면 해당 coefficient 는 해당 independent variable 은 dependent variable 과 관계가 없다는 의미로 regression model 에서 제거를 하는 것이 좋습니다.

6) Confidence interval

Sample 로 부터 계산된 Coefficient 값 b_i 가 significance level 에 있을 수 있는 범위는

$$CI_i = b_i \pm t_{\alpha/2} s_{b_i} = b_i \pm t_{\alpha/2}\sqrt{C_{ii}} \tag{14.8}$$

다음은 regression model 의 예측값에 평균값의 confidence interval 을 계산하는 것과 independent variable 의 입력값에 나올 수있는 예측값의 prediction interval 을 계산하는 공식입니다.

$$CI_i = \hat{y}_i \pm t_{\alpha/2}\sqrt{\text{MSE} \times D_i'(X^TX)^{-1}D_i} = \hat{y}_i \pm t_{\alpha/2}\sqrt{D_i' C D_i} \tag{14.9}$$

$$PI_i = \hat{y}_i \pm t_{\alpha/2}\sqrt{\text{MSE} \times (1 + D_i'(X^TX)^{-1}D_i)} = \hat{y}_i \pm t_{\alpha/2}\sqrt{\text{MSE} + D_i' C D_i} \tag{14.10}$$

여기서 D_i 값은 i 번째 입력값이고 matrix 로 형태는 다음과 같습니다.

$$\begin{bmatrix} 1 \\ d_0 \\ \vdots \\ d_k \end{bmatrix}$$

$\hat{y}_i$ 는 D_i 값을 적용하여 계산된 예측값입니다.

음식쓰레기 배출량(kg)을 도시 가스사용량(m^3)과 수돗물 사용량(ℓ)로 예측을 하려고 합니다.

쓰레기(kg)	가스(m3)	수돗물(ℓ)	쓰레기(kg)	가스(m3)	수돗물(ℓ)
2044	193	333	2054	187	291
1215	92	223	1862	163	297
2236	220	338	1447	115	263
1959	180	307	1660	148	221
1131	105	199	2138	195	366
2029	200	305	2084	204	335
1718	172	273	1703	148	287
1914	170	342	1810	169	322
1390	116	214	1447	146	214
1491	107	241	2131	195	375

jMath 를 이용하는 다른 방법으로 jMath object 로 부터 regression 함수를 호출 하는 방법을 소개 하겠습니다.

```
jMath.prototype.regress(alpha, extra)
```

jMath.regress()와 다른 점은 data 가 jMath object 로 존재하는 것입니다. Regression analysis 를 위해서 data 의 형태는 n 개의 independent variable 들이 있고 k 개의 관찰된 자료가 있을 경우

$$\begin{bmatrix} x_{11} & x_{21} & x_{31} & \dots & x_{n1} & y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1k} & x_{2k} & x_{3k} & \dots & x_{nk} & y_k \end{bmatrix}$$

```
var data = jMath([
[193, 333,2044],
[92, 223,1215],
[220, 338,2236],
[180, 307,1959],
[105, 199,1131],
[200, 305,2029],
[172, 273,1718],
[170, 342,1914],
[116, 214,1390],
```

```
[107, 241,1491],
[187, 291,2054],
[163, 297,1862],
[115, 263,1447],
[148, 221,1660],
[195, 366,2138],
[204, 335,2084],
[148, 287,1703],
[169, 322,1810],
[146, 214,1447],
[195, 375,2131]
]);

result = data.regress(0.05,true);
console.log(result);
```

$$\hat{y} = 237.15 + 5.92x_g + 2.024x_w$$

b 값에 대한 가설 검사 결과는 다음과 같습니다.

	변수	b_0	b_g	b_w
값	result.B.value	237.152168413514	5.91890275547791	2.0242769309629893
p-value	result.B.pvalue	0.01743444548512696	9.3100109799237e-7	0.0019025220311406
CI	result.B.ci	47.1453 ~ 427.15903	4.24515 ~ 7.59265	0.86017 ~ 3.18838
S_b	result.B.se	90.05851765848873	0.7933150875360516	0.5517572761826409

결과에서 보듯이 Confidence interval 에 0 이 포함된 경우는 없고 p-value 역시 0.05 보다 모두 작습니다. 즉 $\beta_i = 0$ 인 null hypothesis 를 reject 할 근거가 됩니다.

다음 입력값에 대한 평균 예상값의 범위와 예측값의 범위는 result.extra 에 값들을 보면 알 수 있습니다.

```
extra: Array[20]
     0: Object
         ci: Array[2]
              0: 2007.2968170235815
              1: 2099.872419439271
               length: 2
              __proto__: Array[0]
         pi: Array[2]
               0: 1893.1848374721606
               1: 2213.9843989906917
               length: 2
              __proto__: Array[0]
         yhat: 2053.5846182314262
         __proto__: Object
```

```
      1: Object
            ci: Array[2]
            pi: Array[2]
            yhat: 1233.1049775222282
           __proto__: Object
      2: Object
      3: Object
        ...
```

result.extra 는 배열로 각 element 값은 입력값에 대한 예측값(yhat), 예측값 평균의 범위(ci), 예측값의 범위(pi)로 구성되어 있습니다.

4. Qualitative independent variable 적용하기

지금까지 사용된 independent variable 은 온도, 무게등과 같은 quantitative 데이타들이지만 남성, 여성과 같은 숫자로 표현이 되지 않는 데이타를 regression model 에 적용을 하기 위해서는 dummy variable 들을 추가하여 남성은 0, 여성은 1 과 같이 숫자로 표현될 수 있도록 변화해야 합니다.

그런데 만약에 regression model 에 사용될 data 값의 종류가 여러가지가 있을 경우 새로 추가될 dummy variable 의 개수는 "data 의 종류 – 1"개 입니다. 에를 들어 도자기를 판매하는 업체가 판매되는 도자기의 가격과 상태에 따른 판매량을 예측하려고 할 때 상태를 매우 좋음, 좋음, 보통, 나쁨, 매우 나쁨으로 5 가지로 구분을 할 경우 dummy variable 은 총 4 개가 필요합니다.

D1	D2	D3	D4	상태
0	0	0	0	매우 나쁨
0	0	0	1	나쁨
0	0	1	0	보통
0	1	0	0	좋음
1	0	0	0	매우 좋음

학력과 소득으로 교육비 지출비용에 대한 예측하는 예제를 보겠습니다. 우선 학력은 Categorical data 로 고졸/대졸/대학원졸로 구분을 했을 경우 총 필요한 dumpy variable 은 "3-1"개로 총 2 개 입니다.

D1	D2	학력
0	0	고졸
0	1	대졸
1	0	대학원졸

소득	D1	D2	교육비	소득	D1	D2	교육비

4110	1	0	1412	3593	1	0	1145
4261	0	1	1365	4113	0	1	1424
3196	0	1	1238	4938	0	1	1639
3491	1	0	1144	4378	0	0	1229
4231	0	0	1187	4436	0	0	1181
3610	0	0	1076	4118	1	0	1292
4534	0	1	1265	4067	0	0	1239
3534	0	1	1384	4751	0	0	1358
3079	0	1	1225	3786	0	0	1007

소득과 교육비 단위는 천만원

Regression model

$$\hat{y} = 390.661 + 0.1894x_i + 132.4718D_1 + 229.51713D_2$$

jMath

```
function converter(v,r)
{
  switch( v[0] )
  {
      case 1:
         return [ 0, 1 ];
         break;
      case 2:
         return [ 1, 0 ];
         break;
      case 0:
      default:
         return [ 0, 0 ];
         break;
  }
  return [ 0, 0 ];
}

var grade = jMath('2 1 1 2 0 0 1 1 1 2 1 1 0 0 2 0 0 0 1 1')
              .transpose().mapInRow(converter);
var income = jMath('4110 4261 3196 3491 4231 3610 4534 3534 3079 3593 4113 4938
4378 4436 4118 4067 4751 3786 3916 3416').transpose();
var Y = jMath('1412 1365 1238 1144 1187 1076 1265 1384 1225 1145 1424 1639 1229
1181 1292 1239 1358 1007 1430 1239').transpose();
data = income[':='](grade)[':='](Y);
result = data.regress();
console.log(result);
```

변수	b_0	X_i	D_1	D_2
result.B.value	390.661	0.1894	132.47	229.51713
result.B.pvalue	0.03491962	0.00021508	0.02721566	0.00008329
result.B.ci	31.3275 ~ 749.9944	0.104979 ~ 0.27387	16.9265 ~ 248.01712	136.4055 ~ 322.629
result.B.se	169.5045	0.03983	54.50494	43.92255

https://github.com/handuck/jMath

5. Model building

Multiple Regression model 을 만들기 위해서 중요한것은 필요한 independent variable 들을 찾아 반영을 시키는 것으로 이러한 과정을 model building 이라고 합니다.
이 과정에서 중요한 내용중 하나는 independent variable 끼리에 높은 관계성을 갖고 있을 경우 multiple regression model 에 문제를 야기 시킬 수 있습니다. independent variable 들에 이러한 현상이 존재를 하는 것을 multicollinearity 라고 합니다.
예를 들어 10 대 손님의 키와 나이로 음식 지출 비용을 예측을 하기 위해서 자료를 수집을 했습니다.

지출액(원)	키(cm) x_1	나이 x_2
20000	170	17
15000	165	16
17000	162	16
8000	150	12
10000	155	13
13000	166	16
8500	145	12
9000	152	13
9000	155	14
14000	161	15

이 데이타에서 보면 키와 나이의 관계성을 correlation coefficient 로 계산을 해 보면 0.97 로 매우 높게 나타납니다. 즉 한 개의 값으로 다른 값을 예측할 수 있기 때문에 하나만 사용하면 충분합니다.
이러한 문제가 regression model 에 어떻게 나타나는가를 보기 위해서 우선 키만을 갖고 regression 을 모델을 구하게 되었을 경우 coefficient 에 p-value 들을 보시면

$$\hat{y} = 459.04x_h - 60224.82$$

	Coefficient	SE	T statistic	p-value
Intercept	-60224.82	13302.22	-4.53	0.00193
키(x_h)	459.04	84.04	5.46	0.0006

여기서 보면 모든 p-value 는 0.05 보다 작기 때문에 이 값이 0 이 확률은 매우 작습니다.

이제 나이를 합쳐서 regression model 을 만들어 보겠습니다.

$$\hat{y} = -12597.16 + 2261.45x_a - 48.183x_h$$

	Coefficient	SE	T statistic	p-value
Intercept	-12597.16	31065.06	-0.4055	0.69721
키(x_a)	-48.183	314.32	-0.1533	0.88249
나이(x_h)	2261.45	1359.75	1.6631	0.14023

이 수식의 특징은 키가 크면 소비량이 작아지게 됩니다. 이 결과는 키만을 갖고 측정을 했을 때와 다른 의미입니다. 또한 나이를 추가하므로써 발생되는것은 coefficient 값들에 p-value 들이 모두 0.05 보다 크다는 것입니다. 즉 null hypothesis 인 coefficient 가 0 이라는 것을 reject 하지 못하게 됩니다.

하지만 단순히 예측값만을 얻기 위한 목적으로 regression model 을 사용할 경우를 전체 regression model 의 significance 를 보기 위한 R^2 의 p-value 가 0.001356 로 0.05 보다는 작기 때문에 예측을 위해서는 이용이 가능합니다.

이러한 multicollinearity 가 존재하는가를 측정하는 방법으로 variance inflation factor(VIF)를 이용할 수 있습니다.

$$VIF_i = \frac{1}{1 - R_i^2} \qquad (14.11)$$

여기서 R_i^2 는 i 번째 independent variable 을 dependent variable 로 취급하고 나머지 independent variable 들을 갖고 multiple coefficient of determination 값 계산하여 얻은 값입니다. 만약 i 째 변수가 다른 변수와 관계가 높다면 R_i^2 값은 1 에 가까워지게 되어 VIF_i 값은 무한대가 되고 반대로 관계가 없다면 0 이되어 VIF 는 1 이 됩니다.
만일 VIF_i 값이 5 보다 크다면 independent variable 들 사이에 관계가 있다고 보고 regression model 에 multicolinearity 가 존재한다고 판단하면 됩니다.

한국 방문한 외국인이 여행 기간 동안 일인당 쇼핑으로 지출한 금액을 동반인원, 방한횟수, 방문기간, 성별(남자:0, 여자:1)로 구분하였습니다.

동반 인원수	방한횟수	방문기간(날)	성별	지출비용(USD)
2	3	9	0	604
2	2	8	1	644
3	2	8	1	888
3	2	8	1	924
2	1	3	0	336
2	1	4	0	304
2	3	10	0	948
3	3	10	1	932
3	2	6	0	520
3	1	4	1	520
2	2	6	1	416
3	2	7	0	664
3	3	10	0	532
1	1	5	0	648
1	1	4	0	236
2	1	5	0	220
4	3	11	1	1208
2	3	10	0	636
3	3	10	0	856
2	2	6	1	589

VIF 값들을 조사한 결과는 다음과 같습니다.

```
jMath.prototype.vif()
```

jMath object 의 형태는 regress()에서 처럼 마지막 column 이 y 값(dependent variable)이고 나머지 값(independent variable)들이 x 값이 됩니다.

```
data = jMath([
        [2,     3,     9,     0,   604],
        [2,     2,     8,     1,   644],
        [3,     2,     8,     1,   888],
        [3,     2,     8,     1,   924],
        [2,     1,     3,     0,   336],
        [2,     1,     4,     0,   304],
        [2,     3,    10,     0,   948],
        [3,     3,    10,     1,   932],
        [3,     2,     6,     0,   520],
        [3,     1,     4,     1,   520],
        [2,     2,     6,     1,   416],
        [3,     2,     7,     0,   664],
        [3,     3,    10,     0,   532],
        [1,     1,     5,     0,   648],
        [1,     1,     4,     0,   236],
```

```
        [2,     1,     5,     0,   220],
        [4,     3,    11,     1,  1208],
        [2,     3,    10,     0,   636],
        [3,     3,    10,     0,   856],
        [2,     2,     6,     1,   589]
]);
var result = data.vif();
console.log(result);
```

결과를 보시면 배열값으로

```
[1.5671539672041421, 11.616516523365322, 12.001970488288821, 1.2306794079412506]
```

이 값의 의미는 다음과 같습니다.

Independent variable	VIF_i	Multicollinearity
동반인원	1.5671539672041421	NO
방한횟수	11.616516523365322	YES
방문기간(날)	12.001970488288821	YES
성별	1.2306794079412506	NO

만일 VIF 값이 5 이상 나타나게 되면 높은 VIF 값 순서대로 제거하고 VIF 검사를 다시 해 모든 VIF 값이 5 보다 작게되는 variable 들만 있을 때까지 합니다.
이 예제에서 방문기간(날)이 가장 높은 VIF 값을 갖고 있기 때문에 이를 제거하고 다시 VIF 를 계산해보면

```
newData = data.removeColumns(2);
var result2 = newData.vif();
console.log(result2);
```

jMath.prototype.removeColumns 함수를 이용해서 방문기간(날) column 을 삭제한 후에 jMath.prototype.vif 를 다시 실행을 한 결과

Independent variable	VIF_i	Multicollinearity
동반인원	1.5371718102257024	NO
방한횟수	1.3122984799631507	NO

https://github.com/handuck/jMath

성별	1.1994472593274992	NO

남은 모든 Independent variable 들의 VIF 모두 5 보다 작아 더 이상 multicollinearity 가 없음을 확인 할 수 있습니다.

이제 이렇게 찾은 independent variable 들을 갖고 regression model 에 포함 여부를 결정하는 방법들을 알아 보겠습니다.

5.1. General Stepwise Regression

변수를 하나씩 추가하면서 regression 에 사용될 independent variable 을 선별하는 방식입니다.

1) Independent variable 별로 regression 을 측정하여 R^2 의 p-value 가 낮은 순서대로 정렬합니다.
2) R^2 의 p-value 가 가장 낮은 independent variable 를 추가하여 얻은 R^2 의 p-value 의 통계적 중요성이 있으면 붙입니다.
3) 만일 2)에서 새로 추가된 independent variable 가 이미 추가된 변수의 coefficient test 결과 p-value 가 alpha 값보다 크게 만들면 제거 시킵니다.
4) 2-3 번을 모든 independent variable 들이 다 적용할 때 까지 반복을 합니다.

이 방식의 단점은 전체를 보지 않고 하나씩 붙이면서 변수들의 영향을 보기 때문에 independent variable 의 조합에 의한 결과를 보지는 못합니다. 이를 보완하기 위해서 다음에 소개할 방법을 적용할 수 있습니다.

5.2. Best Subset Regression

모든 가능한 독립 변수의 조합을 만들어 Mallow C_p-Statistic 값과 R_A^2 값을 갖고 가장 좋은 independent variable 의 조합을 찾는 방식입니다.

1) 모든 가능한 independent variable 조합을 만듭니다. K 개의 independent variable 가 있을 경우 총 가능한 조합은 2^K-1 개가 됩니다. 예를들어 3 개의 independent variable 가 있을 경우 가능한 조합은 X_1, X_2, X_3, X_1X_2, X_1X_3, X_2X_3, $X_1X_2X_3$ 로 총 7 개 입니다.
2) independent variable 조합에 해당하는 R_A^2값을 계산을 합니다.
3) independent variable 조합에 해당하는 C_p 값을 계산을 합니다.

$$C_p = \frac{SSE_p}{MSE} - (n - 2p) \tag{14.12}$$

SSE_p: 조합된 independent variable 를 갖고 측정된 SSE
MSE: 모든 independent variable 를 갖고 측정된 MSE 값
n: 총 sample 의 개수,
p: 조합된 "independent variable 개수 + 1". 여기서 1 은 intercept 입니다. 예를 들어 2 개의 independent variable 가 있다면 p 는 3 이 됩니다.

4) independent variable 조합 선택시 Cp 값은 수식 14.12 에 p 값보다 작거나 같은 independent variable 조합들만 선택을 합니다.

만일 전체 independent variable 들을 사용한 경우 최상의 조합이라면 이 말은 이 경우 SSE_p 값이 최저가 된다는 말이 됩니다. 이 경우 MSE 는 모든 independent variable 들에 대한 값이므로

$$C_p = \frac{(n-V)MSE}{MSE} - (n-2V) = (n-V) - (n-2V) = V$$

이는 수식 14.3 에서 SSE=MSE(n-k-1)=MSE(n-(k+1))에서 k+1 가 V 입니다. 하지만 만일 p 개의 independent variable 들로 부터 최소의 MSE 값을 나타낸다면

$$C_p = \frac{(n-p)MSE_p}{MSE} - (n-2p) = r \times (n-p) - (n-2p) \leq p$$

여기서 r 은

$$0 < r \leq 1$$

5) 선택된 independent variable 조합의 R_A^2 값이 큰 순서대로 선택을 하는데 만약에 다음으로 큰 independent variable 조합의 R_A^2 값의 차이가 크지 않으면 multicollinearity 가 발생되는 경우를 최소화하기 위해서 independent variable 개수가 작은 것을 independent variable 조합으로 선택 합니다.

Stepwise 와 best subset 의 선택은 일반적으로 independent variable 가 개수가 7 개 이상일 경우 stepwise 를 선택하고 그보다 작으면 best subset 방식을 사용합니다.

jMath

```
jMath.prototype.modelbuilding(method)
```

method 는 "stepwise", "forward", "bestsubset" 하나이어야 합니다. 결과는 제시된 independent varaible 들의 정보와 이를 이용한 regression 수행 결과 입니다.
여기서 stepwise 와 forward 는 general stepwise regression 방식이지만 차이점은 forward 에서 coefficient test 를 하지 않습니다.

jMath object 의 형태는 regress()에서 처럼 마지막 column 이 y 값(dependent variable)이고 나머지 값(independent variable)들이 x 값이 됩니다. 이를 방한 기간 동안 외국인의 일인당 쇼핑 지출 금액에 적용을 하겠습니다.

```
data = jMath([
        [2,     3,     9,     0,   604],
        [2,     2,     8,     1,   644],
        [3,     2,     8,     1,   888],
        [3,     2,     8,     1,   924],
        [2,     1,     3,     0,   336],
        [2,     1,     4,     0,   304],
        [2,     3,    10,     0,   948],
        [3,     3,    10,     1,   932],
        [3,     2,     6,     0,   520],
        [3,     1,     4,     1,   520],
        [2,     2,     6,     1,   416],
        [3,     2,     7,     0,   664],
        [3,     3,    10,     0,   532],
        [1,     1,     5,     0,   648],
        [1,     1,     4,     0,   236],
        [2,     1,     5,     0,   220],
        [4,     3,    11,     1,  1208],
        [2,     3,    10,     0,   636],
        [3,     3,    10,     0,   856],
        [2,     2,     6,     1,   589]
]);
data = data.removeColumns(2);
console.log( data.modelbuilding('stepwise') );
console.log( data.modelbuilding('bestsubset') );
```

앞서 VIF 검사 결과 방한기간(날)이 필요없기 때문에 removeColumns 로 제거를 한 후에 실행을 했습니다. 실행 결과 "stepwise"는 "방한횟수", "성별"을 사용할 independent

variable 들로 제시 했지만 "bestsubset"은 남은 모든 independent variable 들을 제시 했습니다.
두 가지 경우에 MSE 값을 비교하면 "stepwise"는 31204.986, "bestsubset"는 29978.094 로 bestsubset 이 작은 것을 알 수 있는데 이유는 "bestsubset"은 모든 independent variable 들의 조합을 비교 분석하지만 "stepwise"는 하나씩 붙이면서 조사하기 때문에 "bestsubset"이 더 좋은 예측 결과를 얻게 됩니다 .

5.3. Residue 분석

선택된 independent variable 조합으로 예측된 값과 실제 값의 차이인 residue 값은 simple regression model 에서처럼 residue 분포가 특정 pattern 이 없이 분산이 잘 되어있나 확인하고 residue 값이 변수 값에 따라 같은 편차가 있는지를 확인하면 됩니다.