Correlation between Health diet and COVID-19

Written by SEOYEONKIM

2020 6 8

(1) LINEAR REGRESSION

1단계: 데이터 수집

요즘 전세계는 코로나19로 떠들썩하다. 코로나에 대응하기 위해서 사람들이 어떤 노력을 할 수 있을지에 대해 생각을 해보다가, 식습관이 코로나 발병률에 영향을 주는지가 궁금해졌다. 이 분석은 Kaggle.com에서 제공하는 나라별 각 식품 군에서 섭취한 지방량, 비만률, 코로나 발병률 등이 담긴 데이터셋을 사용한다. 나라이름은 각각의 행이름에 불과하기 때문에 사용하지 않고, 몇가지 음식 지방량과 비만률이 코로나 발병률과 어떠한 상관관계가 있는지 알아보고자 한다.

다음은 사용하고자 하는 변수들의 설명이다.

1. **Animal fats**

- Butter, Ghee, Cream, Fats, Animals, Raw Fish, Body Oil, Fish, Liver Oil

2. **Animal Products**

- Aquatic Animals, Others; Aquatic Plants; Bovine Meat; Butter, Ghee; Cephalopods; Cream; Crustaceans; Demersal Fish; Eggs; Fats, Animals, Raw; Fish, Body Oil; Fish, Liver Oil; Freshwater Fish; Marine Fish, Other; Meat, Aquatic Mammals; Meat, Other; Milk - Excluding Butter; Molluscs, Other; Mutton & Goat Meat; Offals, Edible; Pelagic Fish; Pigmeat; Poultry Meat

3. **Vegetal Products**

- Alcohol, Non-Food; Apples and products; Bananas; Barley and products; Beans; Beer; Beverages, Alcoholic; Beverages, Fermented; Cassava and products; Cereals, Other; Citrus, Other; Cloves; Cocoa Beans and products; Coconut Oil; Coconuts - Incl Copra; Coffee and products; Cottonseed; Cottonseed Oil; Dates; Fruits, Other; Grapefruit and products; Grapes and products (excl wine); Groundnut Oil; Groundnuts (Shelled Eq); Honey; Infant food; Lemons, Limes and products; Maize and products; Maize Germ Oil; Millet and products; Miscellaneous; Nuts and products; Oats; Oilcrops Oil, Other; Oilcrops, Other; Olive Oil; Olives (including preserved); Onions; Oranges, Mandarines; Palm kernels; Palm Oil; Palmkernel Oil; Peas; Pepper; Pimento; Pineapples and products; Plantains; Potatoes and products; Pulses, Other and products; Rape and Mustard Oil; Rape and Mustardseed; Rice (Milled Equivalent); Ricebran Oil; Roots, Other; Rye and products; Sesame seed; Sesameseed Oil; Sorghum and products; Soyabean Oil; Soyabeans; Spices, Other; Sugar (Raw Equivalent); Sugar beet; Sugar cane; Sugar non-centrifugal; Sunflower seed; Sunflowerseed Oil; Sweet potatoes; Sweeteners, Other; Tea (including mate); Tomatoes and products; Vegetables, Other; Wheat and products; Wine; Yams

4. **Seafood**

- Cephalopods; Crustaceans; Demersal Fish; Freshwater Fish; Marine Fish, Other; Molluscs, Other; Pelagic Fish

5. **Spices**

- Cloves; Pepper; Pimento; Spices, Other

6. **Obesity**

- the percentage of obesity rate

7. **Confirmed**

- the percentage of COVID-19 Confirmed rate

1~5번 변수들은 각각 아래에 정렬되어 있는 음식들에 들어있는 **지방 섭취량(fat supply)**을 의미한다.모든 변수의 값은 전체 인구 대비 %값이다. 따라서, 나라와 인구는 생각하지 않고, 위에 있는 변수들 간의 관계만을 생각한다. 1~6의 독립변수들이 코로나 발병률과 어떻게 관련되어 있는지 생각해보는 것이 중요하다.

2단계: 데이터 탐색과 준비

위의 변수 이름들중 몇개는 띄어쓰기가 되어있어 R에서 이름인식이 잘 되지않을 것이다. 그래서, 띄어쓰기 대신 _만 붙여 준 후 R에 로드한다.
read.csv() 함수를 사용해 분석할 데이터를 로드한다. 데이터의 열 이름을 그대로 사용하기 위해,**header=TRUE**를 사용한다.
그 후, 데이터가 잘 구성되었는지 **str()**함수로 확인한다.

```
corona <- read.csv("C:/Users/KSY/Desktop/health diet &corona/Fat_Supply_Quantity_Data.csv", header=TRUE)
str(corona)
```

```
## 'data.frame':    170 obs. of  32 variables:
##  $ Country                      : Factor w/ 170 levels "Afghanistan",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Alcoholic.Beverages          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Animal_Products              : num  21.6 32 14.4 15.3 27.7 ...
##  $ Animal_fats                  : num  6.222 3.417 0.897 1.313 4.669 ...
##  $ Aquatic.Products..Other      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Cereals...Excluding.Beer     : num  8.04 2.67 4.2 6.55 3.22 ...
##  $ Eggs                         : num  0.686 1.645 1.217 0.154 0.387 ...
##  $ Seafood                      : num  0.0327 0.1445 0.2008 1.4155 1.5263 ...
##  $ Fruits                       : num  0.425 0.642 0.577 0.349 1.218 ...
##  $ Meat                         : num  6.12 8.74 3.9 11.03 14.32 ...
##  $ Miscellaneous                : num  0.0163 0.017 0.0439 0.0308 0.0898 0 0.0361 0.052 0 0.017 ...
##  $ Milk...Excluding.Butter      : num  8.28 17.76 8.09 1.23 6.66 ...
##  $ Offals                       : num  0.31 0.293 0.107 0.154 0.135 ...
##  $ Oilcrops                     : num  1.05 3.16 1.2 3.99 1.36 ...
##  $ Pulses                       : num  0.196 0.1148 0.2698 0.3282 0.0673 ...
##  $ Spices                       : num  0.2776 0 0.1568 0.0103 0.3591 ...
##  $ Starchy.Roots                : num  0.049 0.051 0.1129 0.7078 0.0449 ...
##  $ Stimulants                   : num  0.098 0.527 0.289 0.113 1.055 ...
##  $ Sugar.Crops                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Sugar.Sweeteners             : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Treenuts                     : num  0.7513 0.9181 0.8595 0.0308 0.202 ...
##  $ Vegetal_Products             : num  28.4 18 35.6 34.7 22.3 ...
##  $ Vegetable.Oils               : num  17.08 9.24 27.36 22.46 14.44 ...
##  $ Vegetables                   : num  0.359 0.65 0.514 0.123 0.247 ...
##  $ Obesity                      : num  4.5 22.3 26.6 6.8 19.1 28.5 20.9 30.4 21.9 19.9 ...
##  $ Undernourished               : Factor w/ 98 levels "<2.5","10","10.1",..: 48 78 54 40 NA 63 61 1 1 1 ..
.
##  $ Confirmed                    : num  0.039969 0.039783 0.021642 0.000274 0.026804 ...
##  $ Deaths                       : num  6.76e-04 1.15e-03 1.50e-03 1.27e-05 3.09e-03 ...
##  $ Recovered                    : num  3.49e-03 3.05e-02 1.32e-02 5.73e-05 1.96e-02 ...
##  $ Active                       : num  0.035803 0.008118 0.006895 0.000204 0.004124 ...
##  $ Population                   : num  38042000 2858000 43406000 31427000 97000 ...
##  $ Unit..all.except.Population.: Factor w/ 1 level "%": 1 1 1 1 1 1 1 1 1 1 ...
```

데이터 전처리

분석에 적합하게 데이터를 가공하는 작업을 데이터 전처리라고 한다. 데이터 전처리 작업에 사용되는**dplyr**패키지를 로드한다. **dplyr**패키지의 **select()**함수를 이용해 필요한 변수만 추출한다.
str()을 통해 총 7개의 **numeric**변수가 잘 로드 된것을 볼 수 있다.
모델의 종속 변수는 **Confirmed**로 각 행에 있는 음식지방 섭취량과 비만율을 가진 사람들의 코로나 확진 비율을 나타낸 값이다.

```
library(dplyr)
corona <- corona %>% select(Animal_Products, Animal_fats, Vegetal_Products,Seafood, Spices,Obesity,Confirmed)

str(corona)
```

```
## 'data.frame':    170 obs. of  7 variables:
##  $ Animal_Products : num  21.6 32 14.4 15.3 27.7 ...
##  $ Animal_fats     : num  6.222 3.417 0.897 1.313 4.669 ...
##  $ Vegetal_Products: num  28.4 18 35.6 34.7 22.3 ...
##  $ Seafood         : num  0.0327 0.1445 0.2008 1.4155 1.5263 ...
##  $ Spices          : num  0.2776 0 0.1568 0.0103 0.3591 ...
##  $ Obesity         : num  4.5 22.3 26.6 6.8 19.1 28.5 20.9 30.4 21.9 19.9 ...
##  $ Confirmed       : num  0.039969 0.039783 0.021642 0.000274 0.026804 ...
```

is.na()을 사용하면 데이터에 결측치가 들어있는지 알 수 있다.**table(is.na())**을 적용하여 데이터에 결측치가 총 몇 개 있는지 출력한다. TRUE의 빈도를 보면 결측치가 12개 있다는 것을 알 수 있다.

```
table(is.na(corona))
```

```
##
## FALSE  TRUE
##  1178    12
```

구체적으로 어떤 변수에 결측치가 있는지 보기위해 **table(is.na())**에 변수명을 지정하였다.
아래 코드를 보면 **Obesity**와 **Confirmed**에 각각 3개, 9개가 있다는 것을 알 수 있다.

```
table(is.na(corona$Obesity))
```

```
##
## FALSE  TRUE
##   167     3
```

```
table(is.na(corona$Confirmed))
```

```
##
## FALSE  TRUE
##   161     9
```

따라서, **filter()**의 **&**기호를 이용해 조건을 나열하면, 이 두 변수에 모두 결측치가 없는 행을 새로운**corona_new**변수에 추출할 수 있다.
str()을 통해 170개 였던 데이터가 전처리 후 160개로 바뀐 것을 확인 할수있다.

```
corona_new <- corona %>% filter(!is.na(Obesity)  & !is.na(Confirmed))
table(is.na(corona_new))
```

```
##
## FALSE
##  1120
```

```
str(corona_new)
```

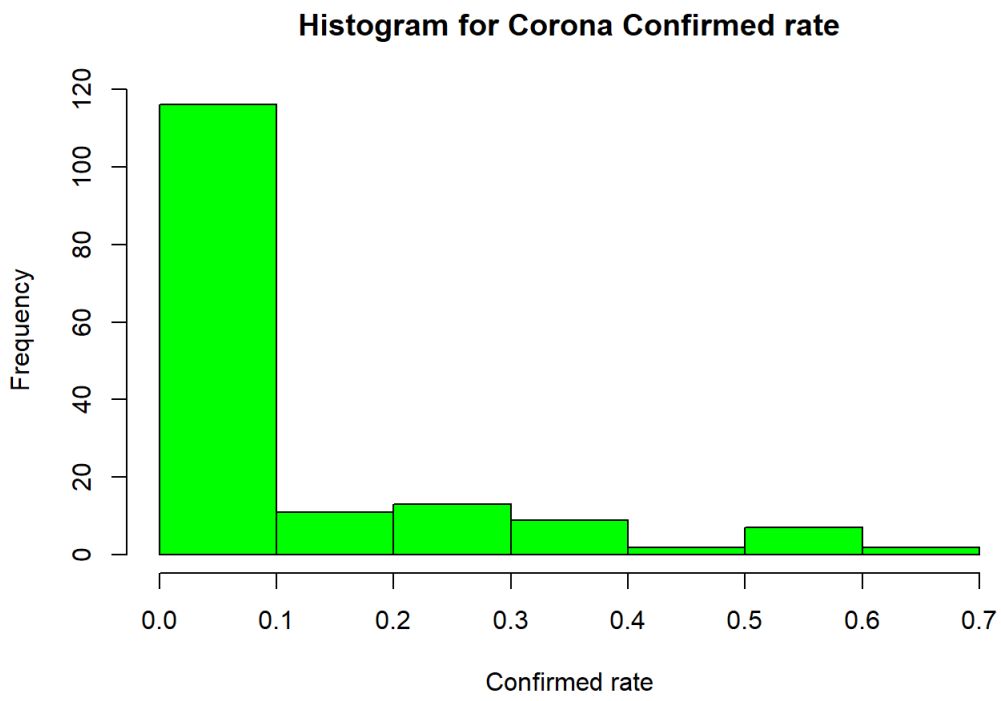
```
## 'data.frame':    160 obs. of  7 variables:
##  $ Animal_Products : num  21.6 32 14.4 15.3 27.7 ...
##  $ Animal_fats     : num  6.222 3.417 0.897 1.313 4.669 ...
##  $ Vegetal_Products: num  28.4 18 35.6 34.7 22.3 ...
##  $ Seafood         : num  0.0327 0.1445 0.2008 1.4155 1.5263 ...
##  $ Spices          : num  0.2776 0 0.1568 0.0103 0.3591 ...
##  $ Obesity         : num  4.5 22.3 26.6 6.8 19.1 28.5 20.9 30.4 21.9 19.9 ...
##  $ Confirmed       : num  0.039969 0.039783 0.021642 0.000274 0.026804 ...
```

다음은 데이터의 요약 통계를 보여준다. 평균값이 중앙값보다 크기 때문에, 코로나 확진률 분포는 오른쪽으로 꼬리가 긴 분포다.**hist()**을 통해 분포를 시각적으로 확인할 수 있다.
확률 분포를 보면 대부분이 0에서 0.1사이에 있고 오른쪽으로 꼬리가 긴 분포임을 알 수 있다.

```
summary(corona_new$Confirmed)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000941 0.0097051 0.0280948 0.1031408 0.1283605 0.6480645
```

```
hist(corona_new$Confirmed, main="Histogram for Corona Confirmed rate",xlab="Confirmed rate",border="black"
     ,col="green")
```



특징 간 관계 탐색: 상관행렬

회귀모델을 데이터에 적합시키기 전에 독립 변수가 종속 변수와 어떤 관계를 가지는지 알아보는 것이 중요하다.**상관 행렬**은 이러한 관계에 대한 요약을 제공한다. **cor()** 명령을 이용하여 **corona_new** 데이터 프레임에 있는 6개의 수치 변수에 관한 상관행렬을 생성한다.

```
cor(corona_new[c("Animal_Products","Animal_fats","Vegetal_Products","Seafood","Spices","Obesity","Confirmed")]
```

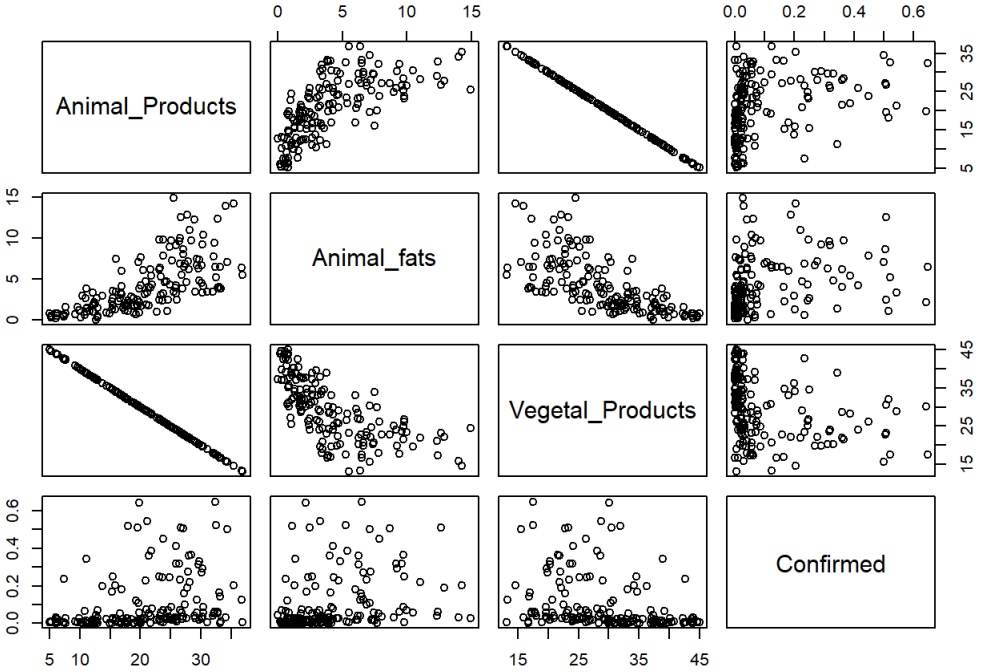
```
##                 Animal_Products Animal_fats Vegetal_Products      Seafood
## Animal_Products       1.000000000   0.6837163     -0.999999855  0.004194152
## Animal_fats           0.683716299   1.0000000     -0.683703920 -0.102092750
## Vegetal_Products     -0.999999855  -0.6837039      1.000000000 -0.004245142
## Seafood               0.004194152  -0.1020928     -0.004245142  1.000000000
## Spices               -0.187724220  -0.2069006      0.187638449  0.238484775
## Obesity               0.475672511   0.4307724     -0.475660492 -0.240079954
## Confirmed             0.316090708   0.2970380     -0.316124801  0.056206350
##                          Spices    Obesity   Confirmed
## Animal_Products  -0.1877242  0.4756725  0.31609071
## Animal_fats      -0.2069006  0.4307724  0.29703798
## Vegetal_Products  0.1876384 -0.4756605 -0.31612480
## Seafood           0.2384848 -0.2400800  0.05620635
## Spices            1.0000000 -0.2312019 -0.13708207
## Obesity          -0.2312019  1.0000000  0.41250423
## Confirmed        -0.1370821  0.4125042  1.00000000
```

각 행과 열이 만나는 지점에서 해당 행과 열이 가리키는 변수의 상관관계가 나열된다. 따라서, 대각선은 자신의 변수끼리의 상관관계를 나타내는 것이므로 항상 1을 나타내고, 대각선을 기준으로 대칭이기 때문에 위 아래 값은 동일하다. **cor()**값은 -1부터 1까지이며, 그것의 절댓값이 0.5가 넘어가면 강한 상관관계를 나타낸다.
결과를 보면, 어떠한 독립변수들도 종속변수와의 상관계수가 0.5을 넘지 않는다. 그래도, 눈에 띄는 것은 **Animal_Products**와 **Obesity**는 **Confirmed**와 **약한 양의 상관관계**를 가지고, **Vegetal_Products** 은 **Confirmed**와 **약한 음의 상관관계**를 가지는 것이다. 이러한 연관성은,**Animal Porducts**와 **Obesity**가 높을수록 코로나 발병률이 높아진다는 것, 그리고**Vegetal Products**가 높을수록 코로나 발병률이 낮아진다는 것을 의미한다.
최종 회귀 모델 구축을 통해, 이러한 관계를 좀 더 정확하게 알아볼 것이다.

특징 간 관계 시각화: 산포도 행렬

산포도 행렬을 통해 수치 특징 간의 관계를 시각화 할 수 있다. 한번에 두개의 특징만을 관찰하기 때문에, 데이터가 어떻게 상호 연관되어 있는지 일반적인 이해를 제공한다. 6개의 독립변수가 있기 때문에, 효과적인 시각화를 위해 3개씩 두번에 나누어서 산포도를 생성 할 것이다.

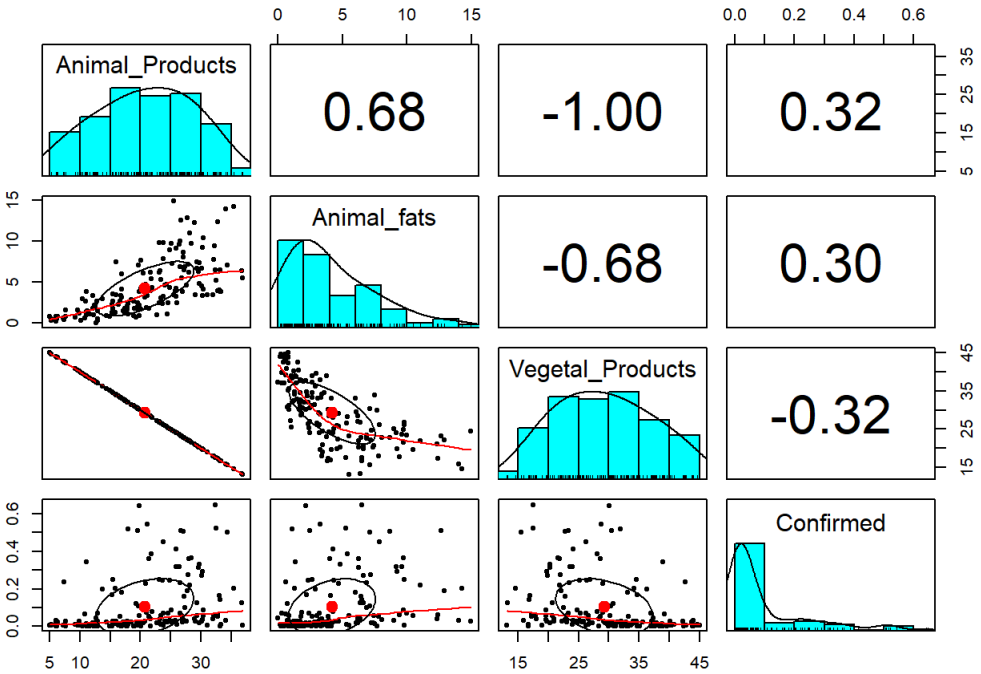
```
pairs(corona_new[c("Animal_Products","Animal_fats","Vegetal_Products","Confirmed")])
```



산포도 행렬에서 각 행과 열의 교차점에는 행, 열 쌍이 가리키는 변수의 산포도가 있다. 다이어그램은 x축과 y축을 바꾼 것이기 때문에 전치다. 무작위 구름처럼 보이는 것도 있지만, **Animal_Products**와 **Vegetal_Products**는 상대적으로 직선으로 어떤 추세를 미세하게나마 어떤 추세를 나타내는 듯 하다.

psych패키지의 **pairs.panels()**함수를 사용하면, **Correlation**과 **Scattered plot**을 동시에 보여주는 좀 더 유익한 산포도 행렬을 생성할 수 있다.

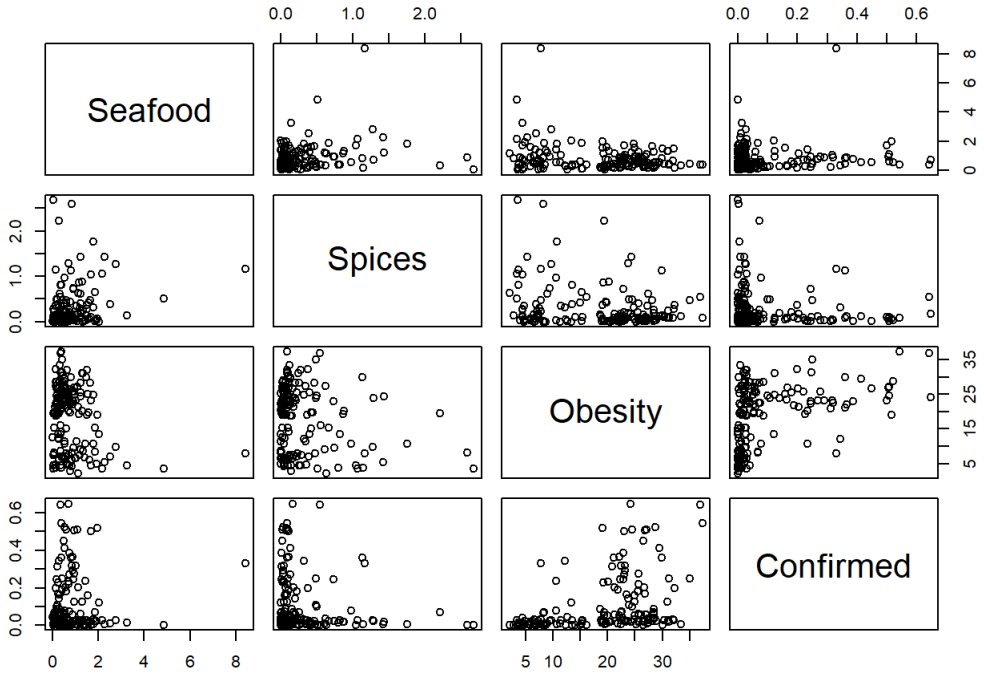
```
library(psych)
pairs.panels(corona_new[c("Animal_Products","Animal_fats","Vegetal_Products","Confirmed")])
```



산포도에 있는 타원모양의 객체는 상관관계 타원형으로, 타원이 늘어질 수록 강한 상관관계를 의미하고, 원에 가까울수록 아주 약한 상관관계를 의미한다. **Anianml_Products**와 **Animal_fats**에 대한 타원은 위로 늘어졌으며, **0.3**에 가까운 **양의 상관관계**를 보여준다. **Vegetal_Products**에 대한 타원은 아래로 늘어졌으며, **-0,3**에 가까운 **음의 상관관계**를 보여준다. 이는 전에 산포도 행렬에서 어느정도 예측했던 특징이다. 산포도에 그려진 곡선을 **뢰스 곡선** 이라고 하는데, x축과 y축 변수사이에 일반적인 관계를 나타낸다. 예를 들어,**3x1행렬**을 보면 **Animal_Products**를 많이 섭취할수록, **Vegetal_Products**를 적게 섭취하는 것을 알수있다.

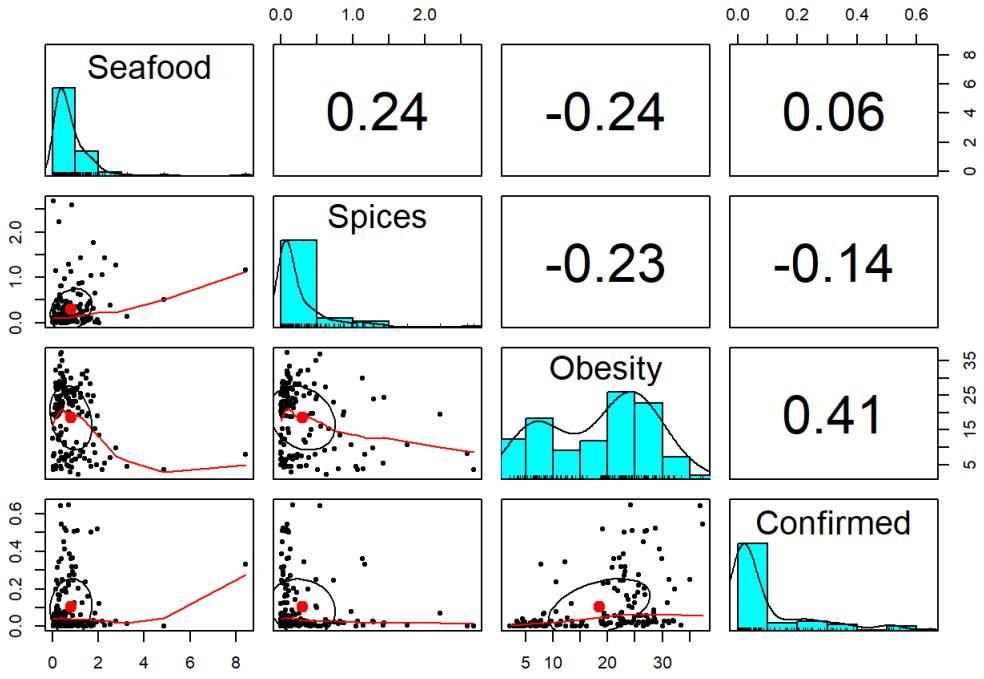
다음으로, 나머지 3개의 독립변수에 대한 산포도 행렬을 생성한다.

```
pairs(corona_new[c("Seafood","Spices","Obesity","Confirmed")])
```



이번에는 대부분의 행렬이 무작위 구름처럼 보인다. 전에 생성했던 산포도에 비해, 종속변수와 낮은 상관관계를 가질 것이라고 예측 할 수 있다. 이중에서, **Obesity**와 **Confirmed**의 관계가 유일하게 어떤 추세를 보이는 것 같다. 다음으로는, 더 자세한 산포도 행렬을 생성한다.

```
pairs.panels(corona_new[c("Seafood","Spices","Obesity","Confirmed")])
```



산포도를 보면, 많은 타원모양의 객체가 원에 가까운 모양을 가지고, 상관관계가 0에 가까운것을 보인다. 앞서 관찰한 산포도행렬로 예측했던 결

과이다. **Obesity**와 **Confirmed**에 대한 타원은 위쪽으로 늘어졌으며, **0.41**에 가까운 양의 상관관계를 보여준다. 즉, **Obesity**이 높을수록, **Confirmed**가 높아지는 것으로 볼 수 있다.

3단계: 데이터로 모델 훈련

R에서 선형 회귀 모델을 데이터에 적합시키려면 r에 내장되어 있는 **stats**패키지의 **lm()**함수를 사용할 수 있다. 다음 명령은, 여섯개의 독립 변수를 코로나 발병률에 연관시키는 선형 회귀 모델을 적합시킨다. 모델을 설명하고자 **틸드문자(~)**을 사용한다. 종속변수 **Confirmed**는 틸드의 왼쪽으로 가고, 나머지 독립변수들은 **+**기호로 분리돼 오른쪽으로 간다. 모델 구축 후, 모델 객체 이름을 입력한다.

```
corona_model <- lm(Confirmed ~ Animal_Products + Animal_fats + Vegetal_Products +
                     Seafood + Spices + Obesity, data=corona_new)
corona_model
```

```
##
## Call:
## lm(formula = Confirmed ~ Animal_Products + Animal_fats + Vegetal_Products +
##     Seafood + Spices + Obesity, data = corona_new)
##
## Coefficients:
##      (Intercept)   Animal_Products       Animal_fats  Vegetal_Products
##       120.634935         -2.412844          0.004699         -2.414004
##          Seafood            Spices           Obesity
##         0.027316         -0.023159          0.005932
```

베타 계수(Coefficient)는 각 특징에 대해 하나가 증가하고 다른 값은 모두 고정됐다고 가정할 때 코로나 발병률 추정치의 증가를 나타낸다. 예를 들어, 다른 모든 것은 같다고 가정하고 **Obestiy**과 **Seafood**값이 높을수록 코로나 발병률이 각각 **0.006%**와 **0.027%**정도로 높아질 것으로 예상한다. 또한, 똑같은 조건에서 **Animal Product**가 증가할 때, 코로나 발병률이 **2.4%**정도 낮아 질것이라는 결과가 나왔다.
이는 전에 산포도 행렬에서 양의 상관관계를 보였던 것과 좀 다른 것을 관찰할 수 있다.

4단계: 모델 성능 평가

corona_model을 이용하여 얻은 **계수 추정치**는 독립변수가 종속 변수와 얼마나 연관되어 있는지 알려주지만, 모델이 데이터에 얼마나 잘 맞는지는 알려주지 않는다. 모델 성능을 평가하기 위해 **summary()**명령을 사용한다.

```
summary(corona_model)
```

```
##
## Call:
## lm(formula = Confirmed ~ Animal_Products + Animal_fats + Vegetal_Products +
##     Seafood + Spices + Obesity, data = corona_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21533 -0.08824 -0.02389  0.03411  0.49508
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      120.634935 125.394844   0.962   0.3375
## Animal_Products   -2.412844   2.507811  -0.962   0.3375
## Animal_fats        0.004699   0.004482   1.049   0.2961
## Vegetal_Products  -2.414004   2.507876  -0.963   0.3373
## Seafood            0.027316   0.012655   2.158   0.0325 *
## Spices            -0.023159   0.024954  -0.928   0.3548
## Obesity            0.005932   0.001388   4.274 3.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1352 on 153 degrees of freedom
## Multiple R-squared:  0.223,  Adjusted R-squared:  0.1926
## F-statistic: 7.319 on 6 and 153 DF,  p-value: 6.84e-07
```

- **잔차(Residuals)**
  - 잔차는 **(실제값 - 예측값)**을 나타낸다. 최대 오차 **0.49508**은 모델이 최소 하나의 관측에 대해 거의 0.5% 정도 코로나 확진률을 높게 예측했다는 것은 의미하고, 최소 오차 **-0.21533**은 모델이 최소 하나의 관측에 대해 거의 0.2%정도 코로나 확진률을 낮게 예측했다는 것을 의미한다.

코로나에 걸리지 않았는데 걸렸다고 할 확률보다, 코로나에 걸렸는데 걸리지 않았다고 할 확률이 더 중요하기 때문에 **0.2%**의 확률을 더 주의깊게 관찰할 필요가 있다. 한편, 대부분의 오차는 **1Q(1사분위)**와 **3Q(3사분위)**에 있으므로, 대부분의 예측은 실제 값보다 **-0.88%**작은 값과 **0.034%**큰 값 사이에 있다.

- **추정계수(Coefficient)**
  - 추정된 회귀 계수별로 **Pr(>|t|)**로 표시된 **p-값**은 추정된 계수가 실제 관계가 없는 0일 확률 추정치다. 즉,**p-값**이 낮을수록, 결과가 우연히 발생할 확률이 낮음을 의미한다. 일부 **p-값**에는 별이 보이는데, 개수가 많아질수록(최대 3개) 믿을만한 변수라는 것이다. 이 모델에서는 여섯개중 두개의 통계적으로 유의한 변수를 가지고있고, 이 변수들이 논리적인 방식으로 결과와 연관된 것으로 관찰된다. 특히, **Obesity**는 별세개로, 거의 0에 가까운 **p-값**을 나타낸다. 하지만, 다른 변수들의 유의확률을 확인 해볼 때, 사용되는 특징이 결과를 잘 예측하지는 못할 것이라고 예상된다.

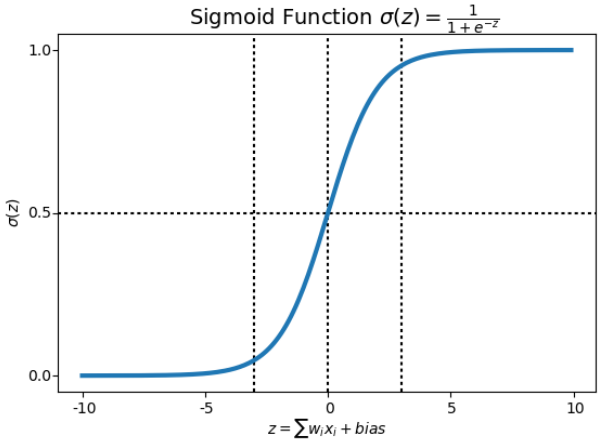
---

(2) ANN으로 코로나 발병률 모델링

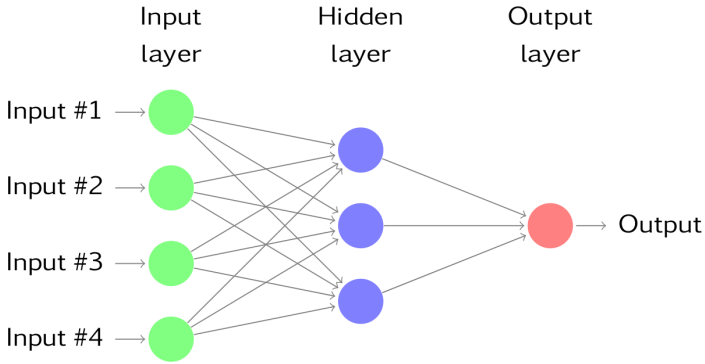
인공신경망(Artificial Neural Network)은 생물학적 뇌가 감각 입력의 자극에 어떻게 반응하는지를 이해해서 유도한 모델을 이용한다. 신경세포의 기본 단위 **neuron**을 따라하여 **artificial nueron**또는 **node**를 사용한다. ANN은 계산량이 많고 복잡한 대신에 좋은 결과를 낸다는 장점이 있다.

분석에 앞서, **ANN**은 중요한 몇가지 특성을 가진다.

- **활성 함수(activation function)**
  - 뉴런의 순 입력 신호를 하나의 출력 신호로 변환해서 네트워크에 더 멀리 퍼지게 한다. 일반적으로,**시그모이드 활성 함수(sigmoid activation function)**을 많이 사용한다. 이 함수는 전체 입력 범위에서 미분 가능하여,**ANN 최적화 알고리즘**을 만드는데 큰 장점을 가진다.



- **네트워크 토폴로지(network topology)**
  - 인공신경망의 구조를 나타낸다.모델의 계층수, 계층 별 노드 개수, 정보가 역방향으로 이동할 수 있는 루프에 따라 구별된다. 다음 사진은 한개의 **hidden layer**을 추가한 다층 네트워크이다.



- **훈련 알고리즘(training algorithm)**
  - 입력 신호에 비례해서 뉴런을 억제하거나 흥분시키고자 연결 가중치를 설정하는 방식을 명시한다.가중치는 알고리즘의 오차를 최소화 시키기위해 **learning rate**만큼 변경한다.

이러한 특성을 이해하고 실제 데이터 분석을 해보고자 한다.

1단계: 데이터 탐색 및 준비

앞서 (1)Linear regression에서 사용한 **corona_new**데이터를 이용할 것이다.

```
str(corona_new)
```

```
## 'data.frame':    160 obs. of  7 variables:
##  $ Animal_Products : num  21.6 32 14.4 15.3 27.7 ...
##  $ Animal_fats     : num  6.222 3.417 0.897 1.313 4.669 ...
##  $ Vegetal_Products: num  28.4 18 35.6 34.7 22.3 ...
##  $ Seafood         : num  0.0327 0.1445 0.2008 1.4155 1.5263 ...
##  $ Spices          : num  0.2776 0 0.1568 0.0103 0.3591 ...
##  $ Obesity         : num  4.5 22.3 26.6 6.8 19.1 28.5 20.9 30.4 21.9 19.9 ...
##  $ Confirmed       : num  0.039969 0.039783 0.021642 0.000274 0.026804 ...
```

신경망은 입력 데이터가 0 주변의 좁은 범위로 조정될 때 효과적으로 작동하기 때문에, 정규화를 통해 데이터를 재조정 해야한다. 데이터가 비정규직이기 때문에 0-1범위로 정규화를 하는 것이 적절하다. **normalize**함수는 다음과 같이 정의된다.

```
normalize <- function(x) {
  return((x-min(x))/(max(x)-min(x)))
}
```

위 함수를 실행 후, **lapply()** 함수를 통해 **corona_new**데이터 프레임의 모든 열에 **normalize**함수를 적용한다.

```
corona_norm <- as.data.frame(lapply(corona_new,normalize))
```

원래의 최소,최대 발병률이 0과 1로 바뀐것을 볼 수있다.

```
summary(corona_new$Confirmed)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## 0.0000941 0.0097051 0.0280948 0.1031408 0.1283605 0.6480645
```

```
summary(corona_norm$Confirmed)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.01483 0.04321 0.15903 0.19795 1.00000
```

2단계: 데이터로 모델 훈련

임의의 순서로 정렬되어 있던 데이터를 **75%**는 **훈련 집합**, **25%**는 **테스트 집합**으로 분리한다. 훈련 데이터 셋은 신경망을 구축하고자 사용할 것이고, 테스트 데이터셋은 모델평가를 위해 사용할 것이다.

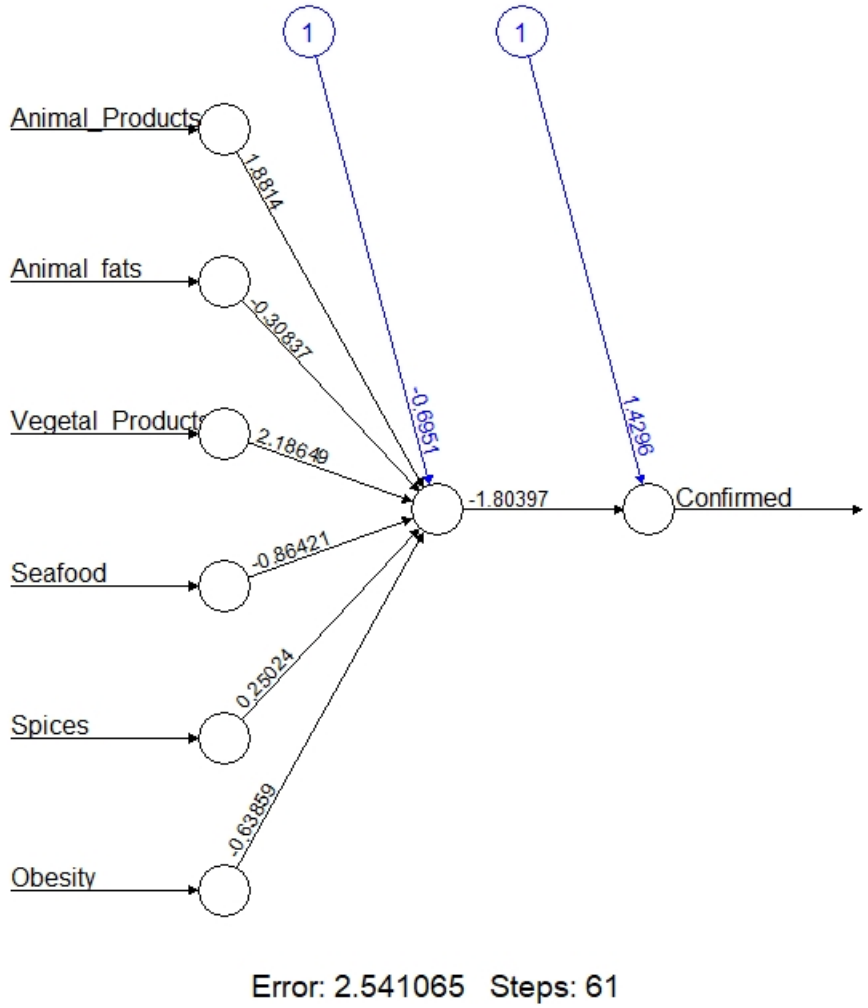
```
corona_train <- corona_norm[1:120,]
corona_test <- corona_norm[121:160,]
```

음식 섭취량, 비만률과 코로나 발병률 사이에 관계를 모델링 하고자**다층피드포워드** 신경망을 사용한다.표준 신경망의 구현과 네트워크 토폴로지를 그리는함수를 제공하는 **library(neuralnet)**로 패키지를 로드한다.
아래의 함수에서 **confirmed**는 종속변수, **+**로 연결된 나머지 6개의 변수는 독립변수를 의미한다. 은닉노드의 기본값은 1이므로,**훈련 집합**을 사용하여 은닉 노드가 하나인 신경망 객체를 반환한다.

```
library(neuralnet)
corona_model2 <- neuralnet(Confirmed ~ Animal_Products + Animal_fats + Vegetal_Products +
                           Seafood + Spices + Obesity, data=corona_train)
```

생성한 모델을 **plot()**함수로 시각화한다.

```
plot(corona_model2)
```



이 모델의 왼쪽에는 각 독립변수의 **입력노드**가 하나씩 있고, 그 옆에 하나의 **은닉노드**와 코로나 발병률을 예측하는 하나의 **출력 노드**가 있다. 각 연결에 써져있는 숫자는 가중치이고, 숫자 1로 써있는 노드는 바이어스 항이다.
Error는 **오차제곱합(SSE)**으로, 예측 값 빼기 실제 값의 제곱을 합한 것이다.**SSE**가 낮아질수록, 예측성능이 좋은 것을 의미한다. 이 값은 훈련 데이터에 대한 모델의 성능평가 하는데는 도움되지만, 처음 보는 데이터에 대해서는 예측하기 어렵다.

3단계: 모델 성능 평가

테스트 데이터셋에 대해 예측을 생성하기 위해 **compute()**를 사용한다. 이 함수는 두개의 구성요소로 된 리스트를 반환하는데, 이를 **model_results**변수에 넣는다.

```
model_results <- compute(corona_model2, corona_test[1:6])
```

net.result는 모델의 예측 값을 가지기 때문에, 코로나 발병률의 예측값을**predicted_corona**변수에 넣는다.

```
predicted_corona <- model_results$net.result
```

예측된 코로나 발병률과 실제 값의 상관관계를 측정하기 위해 **cor()**함수를 사용한다.

앞으로 관찰할 cor()값들은 실행할때마다 다시 모델을 생성하면서 계산에 차이가 나기 때문에 조금씩 값이 변할수도 있다는 점을 유의해야한다.

```
cor(predicted_corona, corona_test$Confirmed)
```

```
##           [,1]
## [1,] 0.5488165
```

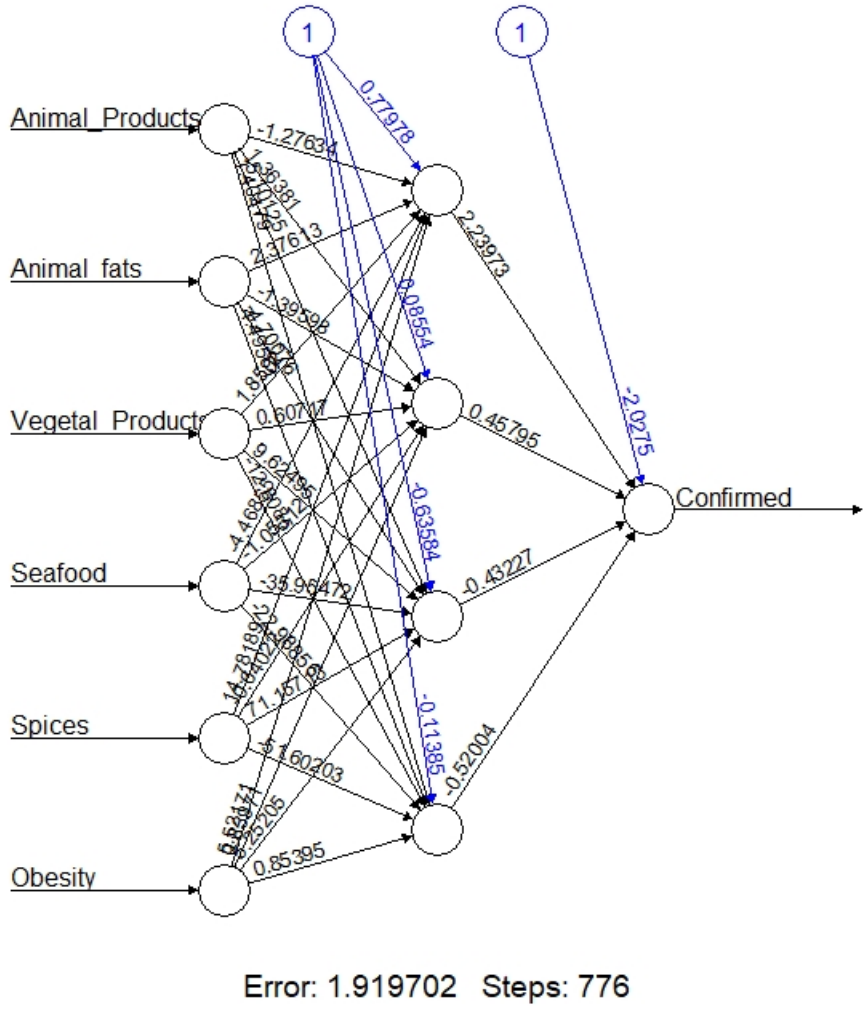
상관관계가 1에 가까울수록 강한 선형 관계를 가진다. 이 모델은 약 0.5의 약하지 않은 상관 관계를 나타낸다. 현재,모델이 하나의 은닉노드만을 가졌기 때문에, 모델의 성능에 개선 될 여지를 기대할 수있다.

4단계: 모델 성능 개선

모델의 성능을 개선하기 위해 은닉 노드의 개수를 증가시켜 볼 것이다.**nueralnet()**함수를 사용하는데 **hidden=4**파라미터를 추가한다. 히든노드를 1개에서 4개로 늘린다는 말이다.

```
corona_model3 <- neuralnet(Confirmed ~ Animal_Products + Animal_fats + Vegetal_Products +
                             Seafood + Spices + Obesity,
                           data=corona_train, hidden=4)
```

```
plot(corona_model3)
```



다음 모델을 보면 **오차제곱합**이 이전 모델의 약2.54에서 약1.5로 감소한 것을 볼 수있다. 은닉노드를 여러개 추가하면서, 훈련단계의 횟수인 **Step**은 엄청나게 증가하였다.

앞서 한것처럼, **compute()**함수와 **net.result**를 이용하여 새로운 코로나 발병률의 모델과의 관계를 알아보겠다.

```
model_results2 <- compute(corona_model3, corona_test[1:6])
predicted_corona2 <- model_results2$net.result
cor(predicted_corona2, corona_test$Confirmed)
```

```
##           [,1]
## [1,] 0.5072104
```

아까와 비교했을때 비슷하지만 조금 낮아진 상관관계를 관찰할 수 있다. 따라서, 은닉노드를 추가한다고 해서 꼭 모델 성능이 개선된다는 것은 아니라는 걸 관찰 할 수있다. 너무 많은 은닉노드는 **과적합(overfitting)**으로 인해, 모델의 성능을 떨어뜨릴수 있다.

다음으로는, 초반에 언급했던 **활성 함수(activation function)**을 이용하여, 모델 성능을 개선해보고자 한다.

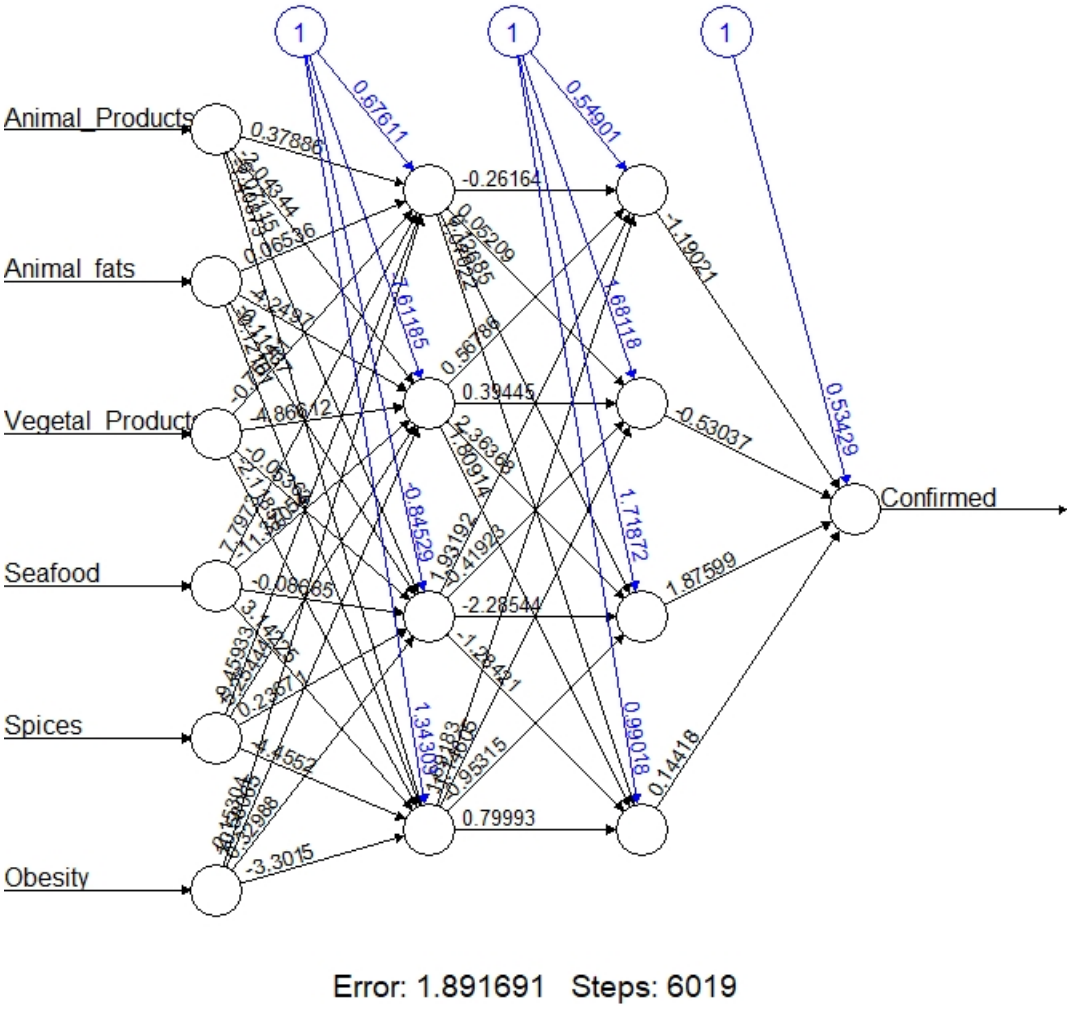
5단계:모델 성능개선

활성함수는 **log(1+exp(x))**로 정의된다. **set.seed()**를 통해 난수를 발생시킨후, **neuralnet()**함수를 통해 모델을 생성한다. 이 때, **hidden=c(4,4)**로 각각 4개 노드를 가진 2 계층 네트워크를 형성하고, 소프트플러스 활성함수를 사용한다.

```
softplus <- function(x) {log(1+exp(x))}
set.seed(12345)
corona_model4 <- neuralnet(Confirmed ~ Animal_Products + Animal_fats + Vegetal_Products +
                             Seafood + Spices + Obesity,
                           data=corona_train, hidden=c(4,4),
                           act.fct=softplus)
```

만들 모델을 **plot()**을 사용하여 시각화한다.

```
plot(corona_model4)
```



오차제곱합이 전 모델들에 비해 많이 감소한 것을 볼 수있다. 히든노드를 여러개 추가하고, 계층을 높이면서 훈련단계 또한 많이 증가하였다.

그리고, 예측 값과 실제 코로나 발병률의 상관관계를 다음과 같이 계산한다.

```
model_results3 <- compute(corona_model4, corona_test[1:6])
predicted_corona3 <- model_results3$net.result
cor(predicted_corona3, corona_test$Confirmed)
```

```
##           [,1]
## [1,] 0.5804032
```

예측과 실제 강도의 상관관계는 약 0.58로 지금까지 모델중에서 제일 좋은 성능을 갖고있음을 보여준다. 하지만, 0.58은 여전히 강한 상관관계를 의미하진 않는다.

데이터 분석에 대한 고찰

- 선형회귀 분석을 통해, 여러 독립변수들 중에 **비만률(Obesity)**이 유일하게 코로나 발병률과 유의미한 관계가 있다고 관찰할 수 있었다. 비만률과 발병률은 양의 상관관계를 가져, 비만률이 높을 수록 코로나에 걸릴 확률이 높아진다는 결론을 얻을 수 있다.

- 하지만, 코로나에 걸렸는데 걸리지 않았다고 할 확률이**0.2%**정도 나왔다. 이는, 코로나가 급성하고있는 현 시점에서 치명적인 결과이다. 평균 코로나 확진률이 **0.1%**것과 비교를 하면, 선택한 독립변수들이 전체적으로 코로나 발병률과 관계가 없다고 봐도 무방한 결과이다.
- 인공신경망을 통해, 코로나 발병률을 모델을 만들어 예측해보았다.하지만,**0.5**근방으로의 상관관계를 계속해서 보이면서, 강한 선형관계를 보여주지 못하였다. 따라서, 선택한 변수들로 만든 모델이 코로나 발병률을 정확하게 예측하지 못하는 것을 보여준다.
- 종합적으로, 코로나는 선택한 종속변수들과 큰 연관성을 갖지 않는 것으로 관찰된다. 물론, 아예 관계가 없다고는 볼 수 없지만, 현시점 코로나 발병률을 낮추기 위해 도움이 될만한 확실한 요소는 찾지 못한것이다. 이번 분석에서 사용한 **회귀**는 사용자가 특징을 선택하고 모델을 명시하기 때문에, 이러한 경우가 드물지 않다고 생각된다. 만약, 어느정도의 전문지식이나 배경지식이 있는 분야에 대해 분석을 한다면, 훨씬 효과적인 모델을 만들 수 있을 것이라 본다.