[R]R을 활용한 상관분석과 회귀분석 - 2

노트북: [TIL-MY]
만든 날짜: 2020-08-10 오후 5:36
URL: https://continuous-development.tistory.com/58

나무늘보의 개발 블로그

홈 태그 방명록

R

[R]R을 활용한 상관분석과 회귀분석 - 2

▪ by 꾸까꾸 ▪ 2020. 8. 10. ▪ 수정 ▪ 삭제

선형회귀분석

예측 모델에서 사용하는 알고리즘으로서 인과 관계를 분석하는 방법이다.

분류 전체보기

Python

Database

ASP.NET

Algorithm

Machine lerning | Deep lear..

선형 회귀 분석 세가지 조건

1. x(독립변수) 가 변하는 것에 따라서 y(종속변수)도 변한다.

2.시각적으로 선행 되어야 한다.

3.외생변수를 통제한다 (다른 요인을 통제하고 인과관계를 분석한다)

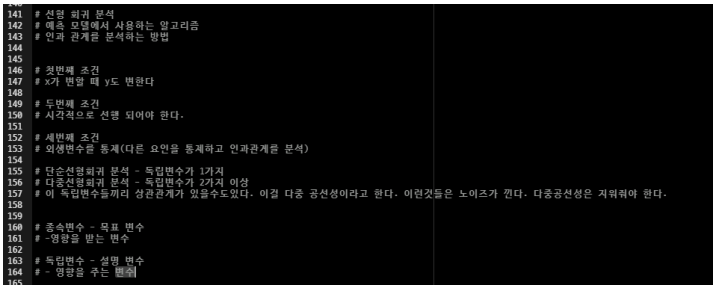
※독립변수 - 설명 변수로서 영향을 주는 변수이다.

※종속변수 - 목표변수로서 영향을 받는 변수이다.

선형회귀 분석 종류

단순선형 회귀 분석 - 독립변수가 1가지 인 경우

다중선형회귀 분석 - 독립변수가 2가지 이상인 경우



lm()

AWS

ETC..

R

공지사항

글 보실 때 주의사항

최근글 : 인기글

[Python] 파이썬 기초 3 ····

2020.08.10

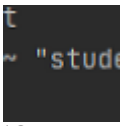
[Python] 파이썬 기초 2 ····

2020.08.10

[Python] 파이썬 기초 1 ····

2020.08.10

[Pyth
pyt...



2020.08.10

[Pyth
pyt...

2020.08.10

최근댓글

태그

python print, 파이썬 기본 출력문, python 튜플,

- lm함수는 linear model의 약자로 선형 모델을 맞추는 데 사용된다. 회귀 분석, 분산의 단일 계층 분석 및 공분산 분석을 수행하는 데 사용할 수 있다.

lm(종속변수 ~ 독립변수 ,data)

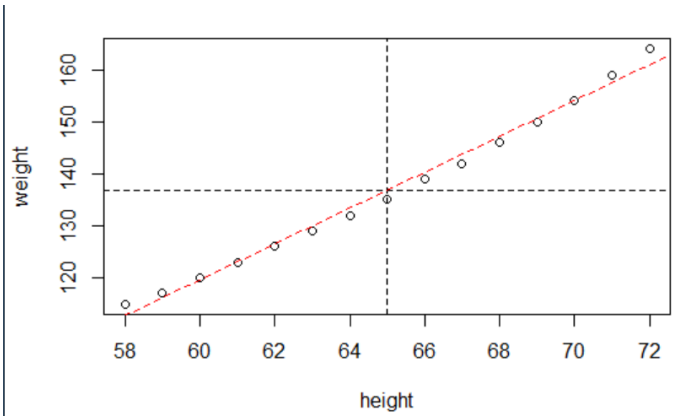
```
lm(종속변수 ~ 독립변수 ,data)
```

```
# model ← lm()
# plot(model)
# summary()
# abline()
# abline(intercept, slope)

# y = b0 + b1x + e
# b0 : 절편
# b1 : 기울기
# e(epslion) : 오차


women
str(women)

#    x = height
# weight=b0+b1x+e
cor(women)

fit_model ← lm(weight ~ height, data=women)
```

fit_model이라는 예측모델을 만든다. 키를 이용하여 몸무게를 예측하는 모델이다.

예측모델을 통해 abline을 그린다.

```
plot(weight ~ height , data = women)
abline(h=mean(women$weight), lty=2)
abline(v=mean(women$height), lty=2)
abline(fit_model, lty=2, col='red')

fitted(fit_model)[1]
```



ggplot, Oracle, 파이썬 join, 파이썬 형변환, AWS, 사용법, 날짜함수, rbind, Python, Oracle SQL, 파이썬, python variable, substr, DDL, R로 하는 크롤링, 파이썬 함수, R을 통한 크롤링, 파이썬 tuple, SQL, 인스턴스, 파이썬 타입, anaconda 가상환경, cbind, 테이블 생성, 행렬, anaconda 가상환경 설정, 설정

```
> fitted(fit_model)[1]
       1
112.5833
```

fitted를 통해서 예측값을 볼 수 있다. 여기서 fit_model에서는 height값에 따른 weight를 구한다.

#모델 예측치 / 오차값

residuals(model) - 예측값과 실제 값 사이의 차이는 잔차를 나타낸다.

```
#모델 예측치
y_pred <- 87.52 + 3.45*58
y_pred

err <-115-112.58
err

residuals(fit_model)[1]
```

```
> women
   height weight
1      58    115
2      59    117
3      60    120
4      61    123
```

```
> fitted(fit_model)[1]
       1
112.5833
```

이런식으로 오류치를 찾는다.

```r
#모델 예측치
#예상 한 값
y_pred <- 87.52 + 3.45*58
y_pred

#오차값
err <-115-112.58
err

#오차를 확인한다.
residuals(fit_model)[1]


summary(fit_model)
```

아래는 모델을 summary 했을때 나오는 결과 값으로 해석하자면

R-squared 는 결정계수로서 99프로 신뢰할 수있다.

여기서 multiple 과 adJusted 차이가 크면 다시만들어야 된다. 잘못만든 것이다.

```
> summary(fit_model)

Call:
lm(formula = weight ~ height, data = women)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7333 -1.1333 -0.3833  0.7417  3.1167

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -87.51667    5.93694  -14.74 1.71e-09 ***
height        3.45000    0.09114   37.85 1.09e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.525 on 13 degrees of freedom
Multiple R-squared:  0.991,     Adjusted R-squared:  0.9903
F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

cor.test를 통해

상관분석으로 지금 귀무가설이 맞는지 확인하고

```
> cor.test(women$weight, women$height)

        Pearson's product-moment correlation

data:  women$weight and women$height
t = 37.855, df = 13, p-value = 1.091e-14
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.9860970 0.9985447
sample estimates:
      cor
0.9954948
```

이렇게 만들어 놓은 모델에 값을 넣어 예측함수를 통해서 height가 72일때 예측되는 파운드를 나타낸다.

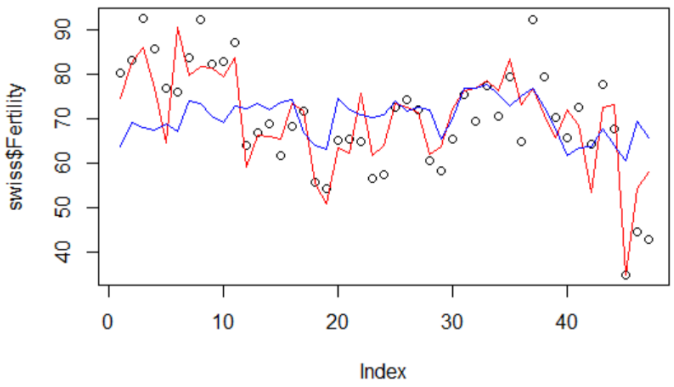
```
> #예측 함수
> #predict(모델 , 테스트 데이트)
> predict(fit_model, newdata = data.frame(height = 72))
       1
160.8833
```

#예측모델 평가 지표

시계열 분석을 위해 forecast를 install 한다.

여기서는 다항분석을 해보자.

```
229
230 # 예측모델 평가지표
231
232 #ME(Mean of Errors)  - 평균
233 #MSE(Mean Squared Error) - 제곱의 평균
234 #RMSE(Root Mean of Squared Error) -  제곱근 이게작으면 작을수록 신뢰도가 높은 모델이다.
235 #MAE(Mean of Absolute Error) - 오차의 개수를 절대값으로 나눈것
236 #MPE(Mean of Percentage Error)
237
238
239 install.packages("forecast")
240
241 swiss
242 str(swiss)
243
244 #다항 회귀분석
245 model01 <- lm(Fertility ~., data = swiss) #.을쓰면 모든걸 포함한다.
246 #단항 회귀분석
247 model02 <- lm(Fertility ~Agriculture, data = swiss) #Fertility을 분석하는데 Agriculture 통해 한다.
248
249 plot(swiss$Fertility)
250 lines(model01$fitted.values, col="red") #선을 그린다.
251 lines(model02$fitted.values, col="blue") # 선을 그린다.
252
253 forecast::accuracy(model01)     # RMSE가 중요하다 이 값이 작을수록 신뢰성이 높다.
254 forecast::accuracy(model02)
255 forecast::accuracy(fit_model)
```



forcast를 써서 정확도 평가를 할 수 있다.

```
> forecast::accuracy(model01)
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
                      ME     RMSE      MAE        MPE     MAPE      MASE
Training set -3.02617e-16 6.692395 5.32138 -0.9942129 7.857082 0.5555942
> forecast::accuracy(model02)
                     ME     RMSE      MAE       MPE     MAPE     MASE
Training set 3.01399e-16 11.56215 9.590092 -3.285437 14.88935 1.001282
> forecast::accuracy(fit_model)
            ME     RMSE      MAE          MPE     MAPE      MASE
Training set  0 1.419703 1.155556 0.0002300874 0.8409659 0.08947006
```

#ME(Mean of Errors) - 평균

#MSE(Mean Squared Error) - 제곱의 평균

#RMSE(Root Mean of Squared Error) - 제곱근 이게 작으면 작을수록 신뢰도가 높은 모델이다.

#MAE(Mean of Absolute Error) - 오차의 개수를 절대값으로 나눈것

#MPE(Mean of Percentage Error)

회귀분석을 위한 작업 순서

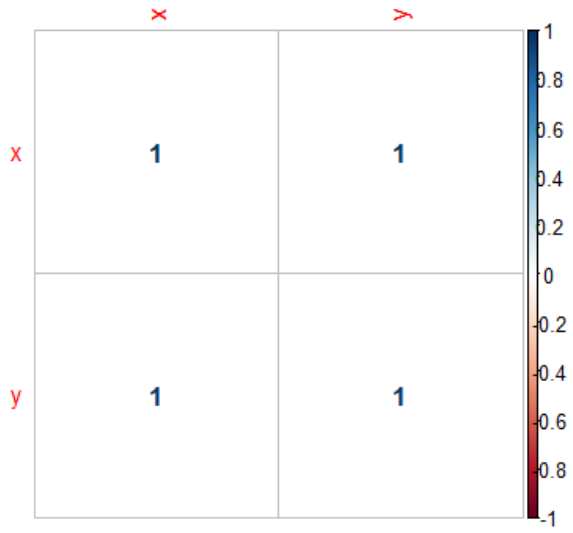
1.결측치 확인

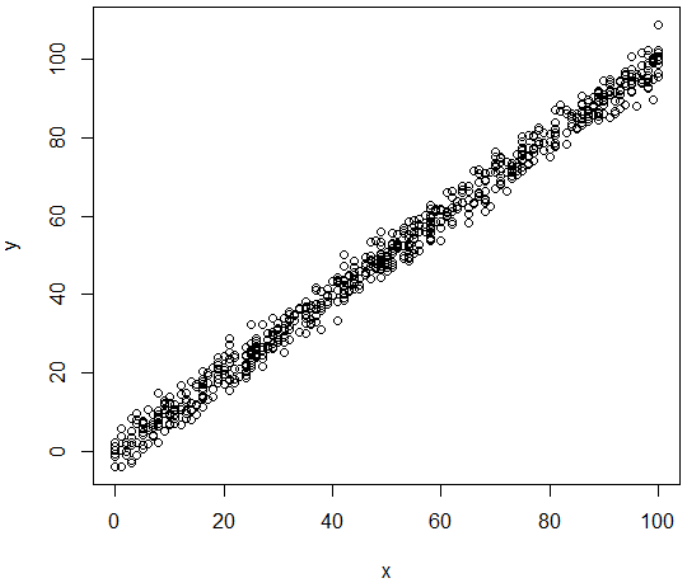
```
342 #1.결측치 확인
343 table(is.na(train))
344 colSums(is.na(train))
345
346 # complete.cases - 행에 누락된 데이터가 없는(NA가 존재하지 않는)지를 확인해주는 함수
347 train[!complete.cases(train), ]
348 train <-train[complete.cases(train), ]
```

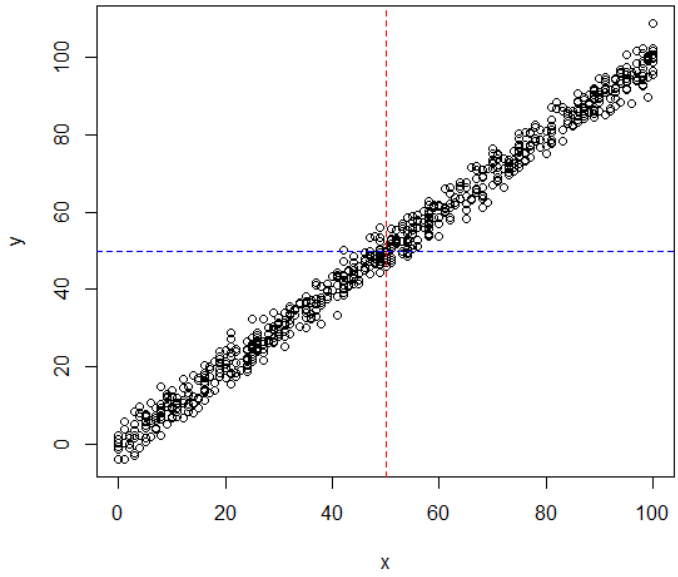
```
> #1.결측치 확인
> table(is.na(train))

FALSE  TRUE
 1399     1
> colSums(is.na(train))
x y
0 1
```

2.상관분석

```
349
350 #2. 상관분석
351 cor(train)
352 corrplot(cor(train), method = "number")
353 plot(train)
354 abline(h=mean(train$y), lty=2,col="blue")
355 abline(v=mean(train$x), lty=2,col="red")
356
357 #3. 이상치 확인
```

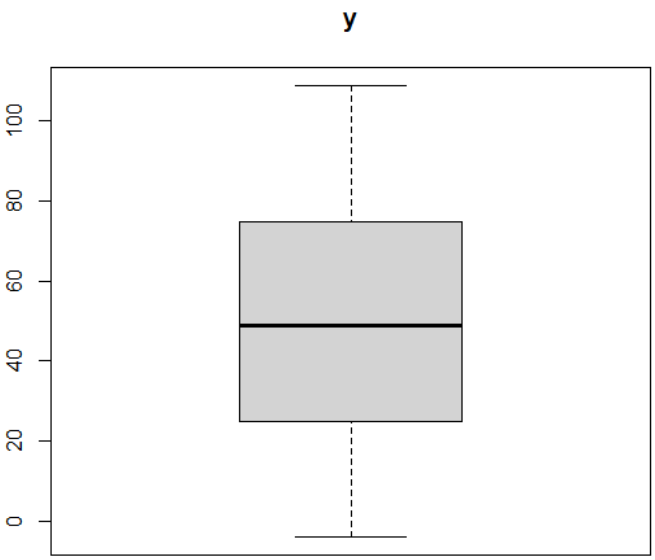




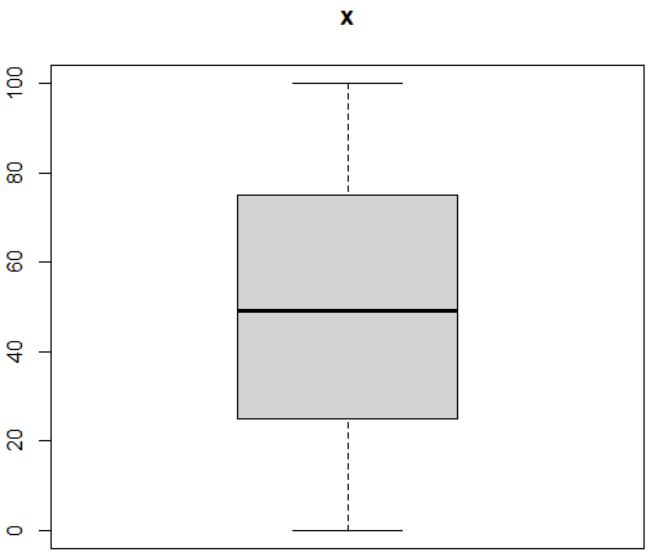


3.이상치 확인

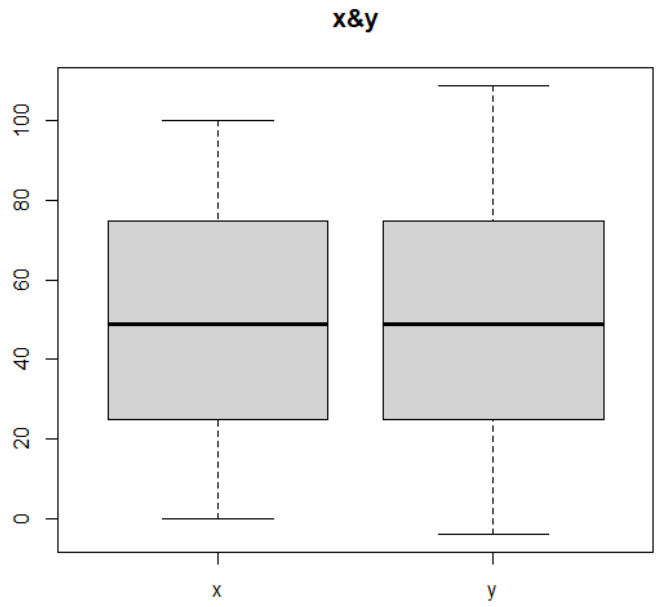
```
354
355  #3. 이상치 확인
356  boxplot(train$y,
357          main="y")
358
```




```
358
359 boxplot(train$x,
360         main="x")
```




```
361
362 boxplot(train,
363         main="x&y")
```



4.회귀 적합 모델 만들기

```
> #4.회귀 적합모델 만들기
> train_model <- lm(y~x,train) # 종속변수 다음 독립변수를 적는다.
> train_model

Call:
lm(formula = y ~ x, data = train)

Coefficients:
(Intercept)            x
    -0.1073       1.0007
```

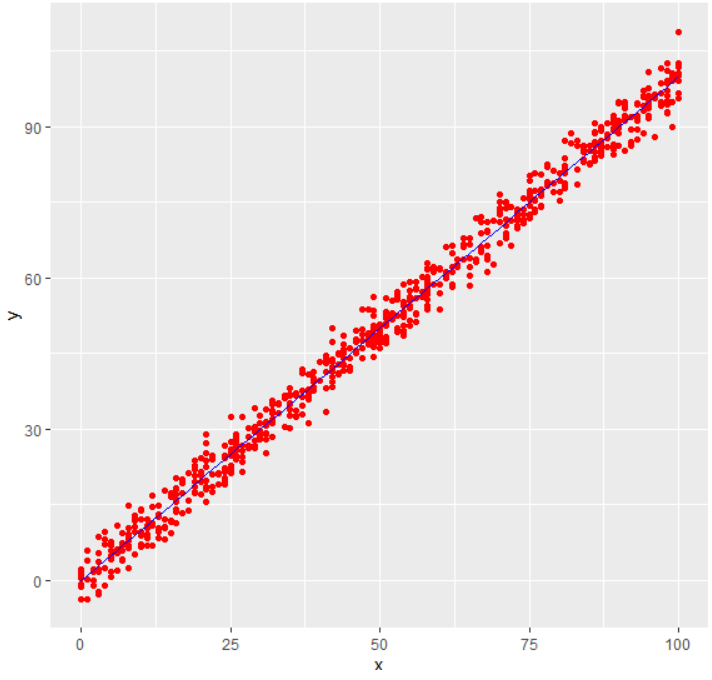
```
> y_pred <- 87.52 + 3.45*58
> y_pred
[1] 287.62
> y_pred <- -0.1073 + (24 *1.0007)  # y값 예측( 절편  (x * 기울기 ) )  #예측값
> y_pred
[1] 23.9095
>
>
> fitted(train_model)[1]# 적합된 값
       1
23.90849
```

```
> err <- 21.54945 - 23.9095 #오차값
> err
[1] -2.36005
>
> residuals(train_model)[1] #오차값
         1
-2.359036
```

```
> summary(train_model)

Call:
lm(formula = y ~ x, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-9.1523 -2.0179  0.0325  1.8573  8.9132

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.107265   0.212170  -0.506    0.613
x            1.000656   0.003672 272.510   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.809 on 697 degrees of freedom
Multiple R-squared:  0.9907,    Adjusted R-squared:  0.9907
F-statistic: 7.426e+04 on 1 and 697 DF,  p-value: < 2.2e-16
```

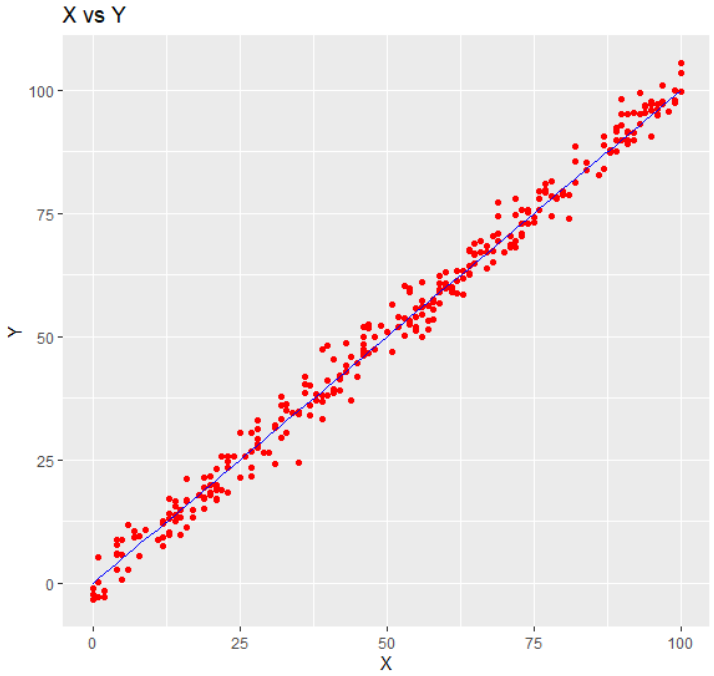
5.분석결과 시각화

```
388
389 #5.분석결과 시각화
390 #predict( ) 데이터에 대한 예측값
391 ggplot(train, aes(x,y))+
392   geom_point(col='red')+
393   geom_line(aes(x=train$x,
394                 y=predict(train_model,newdata = train )),
395             col='blue')
396
397 abline(v=mean(train$x), lty=2,col="red")
398
```



6. 정확도 계산

```
#6. 정확도 계산

y_predict <-predict(train_model , newdata = test)
head(test,1)
head(y_predict,1)


ggplot(test, aes(x,y))+ #데이터는 이제 test로
  geom_point(col='red')+
  geom_line(aes(x=test$x,   # 여기서는 x축에 test로 하고 y 값은 데이터를 예측한 값을 통한다.train 데이터로
                y=predict(train_model,newdata = test )), #
            col='blue')+
  ggtitle('X vs Y')+
  xlab('X')+
  ylab('Y')
```



```
#------선형 회귀 분석 PART02
# service_dataSets_product_regression.csv
# 회귀분석은 정규분포를 따르는 것으로 해야된다. 이런건 의미가 없다.
regressionData <-read.csv(file.choose())

#상관분석 - 변수간의 관계를 보고 그 인과성을 회귀로 가져간다.
regressionData

regressionData.cor <- cor(regressionData)

#이걸로 상관관계
corrplot(regressionData.cor, method = "number") #regressionData.cor를 표안에 숫자로 표현해준다.
```


```
> cor(regressionData$제품_만족도, regressionData$제품_적절성) #값으로로 확인
[1] 0.7668527
> cor(regressionData$제품_만족도, regressionData$제품_친밀도) #값으로로 확인
[1] 0.467145
> cor.test(regressionData$제품_만족도, regressionData$제품_적절성) # test로 어떤지 확인 0.76

        Pearson's product-moment correlation

data:  regressionData$제품_만족도 and regressionData$제품_적절성
t = 19.34, df = 262, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7120469 0.8123706
sample estimates:
      cor
0.7668527
```

```
> cor.test(regressionData$제품_만족도, regressionData$제품_친밀도) # test로 어떤지 확인 0.46

        Pearson's product-moment correlation

data:  regressionData$제품_만족도 and regressionData$제품_친밀도
t = 8.5519, df = 262, p-value = 1.026e-15
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3671226 0.5564877
sample estimates:
      cor
0.467145
```

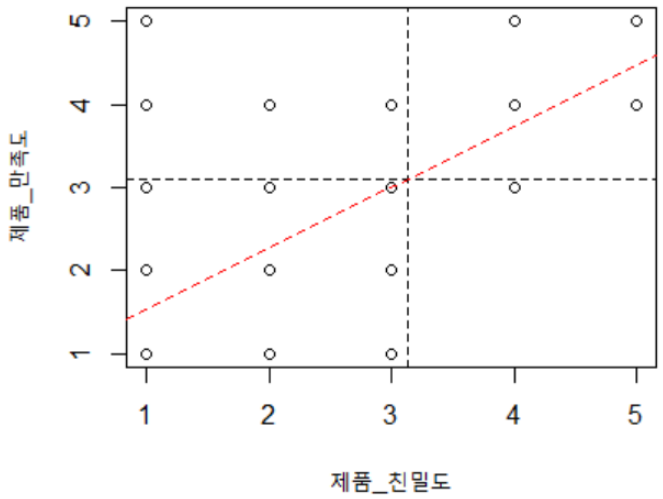
성능 테스트

회귀분석 모델을 만들고

```
279
280  #회귀분석
281  regression_model1 <- lm(제품_만족도 ~ 제품_적절성, data=regressionData) # 첫번째 모델은 적절성에 따른 만족도
282
283  regression_model2 <- lm(제품_만족도 ~ 제품_친밀도, data=regressionData) # 두번째 모델은 친밀도에 따른 만족도
284
```

그래프를 그려본다.

```
293 abline(h=mean(regressionData$제품_만족도), lty=2) #plot 차트에 만족도에 대한 평균을 그린다.
294 abline(v=mean(regressionData$제품_적절성), lty=2)
295 abline(regression_model1, lty=2, col='red')
```




```
326 # ----- [실습]Part03
327 # Linear Regression
328 # https://www.kaggle.com/andonians/random-linear-regression
329
330 # service_datasets_train_ml.csv
331 service_train ← read.csv(file.choose())
332
333 # service_datasets_train_test_ml.csv
334 service_test ← read.csv(file.choose())
335
336 train ← service_train
337 test ← service_test
338 str(train)
339 str(test)
```

```
> colSums(is.na(train))
x y
0 0
> cor(train)
          x         y
x 1.0000000 0.9953399
y 0.9953399 1.0000000
```

```r
## 다항분석
# 종속변수 : charges
# 독립변수 : age,bmi, children

# 상관계수 확인
library(dplyr)
insu_train_test<-insu_train %>%
  select(age,bmi,children,charges)
cor(insu_train_test)

# method - circle, square, ellipse, shade, color, pie
corrplot(cor(insu_train_test), method = "number") #regressionData.cor를 표안에 숫자로 표현해준다.

corrplot(cor(insu_train_test), method = "shade",order="FPC",addCoef.col="black") # 상관계수 숫자 색) #regressionData.cor를 표안에 숫자로 표현해준다.

```

```r

insu_model <- lm(charges~age+bmi+children,data = insu_train)
insu_model

```

```r

insu_model <- lm(charges~age+bmi+children,data = insu_train)
age <- insu_model$coefficients[2]
bmi <- insu_model$coefficients[3]
child <-insu_model$coefficients[4]

head(insu_train,1)
# y = (a1*x1)+(a2*x2)+(a3*x3)
# y = (19*age)+(27.9*bmi)+(0*children)

lm(charges~age+bmi+children,data = insu_train)
```

```
477:47  (Top Level) ÷
Console  Terminal ×  Jobs ×
~/ ➦
oefficients:
Intercept)          age          bmi     children
   -6916.2        240.0        332.1        542.9

 insu_model$coefficients[2]
    age
39.9945
 insu_model$coefficients[3]
    bmi
32.0834
 insu_model$coefficients[4]
hildren
42.8647
 insu_model$coefficients[5]
NA>
 NA
 age <- insu_model$coefficients[2]
 bmi <- insu_model$coefficients[3]
 child <-insu_model$coefficients[4]
 head(insu_train,1)
 age    sex  bmi children smoker    region  charges
  19 female 27.9        0    yes southwest 16884.92
```

```r
install.packages("car")
library(car)

Prestige
str(Prestige)

# 종속변수 : income
# 독립변수 : education , women, prestige
lm(charges~age+bmi+children+smoker+region ,data = insu_train)

# 상관분석
Prestige_data<-Prestige %>%
  select(income,education,women, prestige)

cor_presige<-cor(Prestige_data)

#일반 독립변수들 사이에 연관성이 높은것들은 삭제해줘야 한다. 이런걸 다중공선성이라고 한다.
corrplot(cor(Prestige_data), method = "number") #regressionData.cor를 표안에 숫자로 표현해준다.
prestige_model<-lm(income ~.,data=Prestige_data)

#y=(a1*x1)+(a2*x2)+(a3*x3)

head(Prestige_data,1)
predict_y = (177.2*13.11) + (-50.9*11.16) + (141.4*68.8) - 253.8 # 마지막 intercept값
error = 12351 - predict_y

summary(prestige_model)
# 별의 의미는 거긴 모델에서는 영향력이 없다는 것을 말한다. 별이 있어야 유의미한 애들이라는 말이다. 그래서 예측률을 올리기 위해서는 이런 속성값들을 빼줘야한다.
```

```
Call:
lm(formula = income ~ ., data = Prestige_data)

Residuals:
    Min      1Q  Median      3Q     Max
-7715.3  -929.7  -231.2   689.7 14391.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -253.850   1086.157  -0.234    0.816
education    177.199    187.632   0.944    0.347
women        -50.896      8.556  -5.948 4.19e-08 ***
prestige     141.435     29.910   4.729 7.58e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2575 on 98 degrees of freedom
Multiple R-squared:  0.6432,    Adjusted R-squared:  0.6323
F-statistic: 58.89 on 3 and 98 DF,  p-value: < 2.2e-16
```

```r
#영향력이 없는 education을 빼고 했을 경우
prestige_model2<-lm(income ~.-education,data=Prestige_data)
summary(prestige_model2)
```

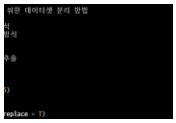
```
510:1  (Top Level)

Console  Terminal  Jobs

~/

Residuals:
    Min      1Q  Median      3Q     Max
-7620.9 -1008.7  -240.4   873.1 14180.0

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  431.574    807.630   0.534    0.594
women        -48.385      8.128  -5.953 4.02e-08 ***
prestige     165.875     14.988  11.067  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2573 on 99 degrees of freedom
Multiple R-squared:   0.64,     Adjusted R-squared:  0.6327
F-statistic: 87.98 on 2 and 99 DF,  p-value: < 2.2e-16
```

'**R**' 카테고리의 다른 글

[R]R을 활용한 상관분석과 회귀분석 - 2 (0)	02:42:26
[R] R을 활용한 크롤링 - 로또 1등 당첨 배출점 크롤링 하기 (0)	2020.08.07
[R] R에서 교차검증을 위한 데이터 셋 분리방법 3가지 (0)	2020.08.07
[R] R을 활용한 상관분석과 회귀분석 - 1 (0)	2020.08.06
[R] R을 통한 텍스트마이닝에서 워드클라우드 까지 (0)	2020.08.05
[R] R로 하는 비정형 데이터 처리 (facebook 데이터를 통한 긍정/부정 나누기) (0)	2020.08.03

관련글


[R] R을 활...


[R] R에서 ...


[R] R을 활...


[R] R을 통...

댓글 0