<경데분 2 실습코드정리>

o 데이터파일이름 <— read_csv("file.csv") : file을 데이터파일 이름으로 저장하고 불러오기
☆우선 library(readr) 먼저하기
o str(데이터 or 데이터$변수) : 데이터 또는 데이터의 변수에 대해 다양한 정보 (척도, 개수, 예시 ...)
o glimpse(데이터) : 데이터의 다양한 정보를 보여줌 - str보다 조금 깔끔하게
☆우선 library(dplyr) 먼저하기
o summary(데이터) : 기술통계량들을 보여줌, 즉 문자형척도 제외하고 계량형척도 (int, num ...)의 수치적인 정보
결측치의 개수확인 (NA)
o freq(데이터$변수) : 해당변수의 빈도수와 막대그래프
☆우선 library(descr) 먼저하기 ← 변수여러개가능
o 데이터 %>% group_by(변수) %>% summarise(변수명= mean or var or sd ... (변수))
: 변수별로 그룹을 나누고 기술통계량을 조건에 맞게 제시
☆우선 library(dplyr) 먼저하기☆ %>% 함수들 필수
⊕ mean/var/sd(데이터$변수, na.rm=T) : 데이터 변수에서 결측치(NA)를 제외하고 계산

o boxplot(데이터$변수) : 해당변수의 상자그래프. 단 변수는 계량형척도 (이상치 check 가능)
o 데이터$변수 <—— as.factor(데이터$변수) : 해당변수 척도를 범주형으로 변경 ⊕numeric
o 데이터$변수 <—— as.POSIXct(데이터$변수) : 해당변수 척도를 시간형으로 변경
o table(데이터$변수) : 해당변수의 빈도수
o 데이터$변수 <— factor(데이터$변수, levels=c("a", "b", "c" ...)) (범주형)
: a, b, c 순으로 변수를 배열하고 척도를 범주형으로 바뀐뒤 해당데이터의 변수에 지정
o 데이터$변수 <—— ifelse(데이터$변수=="a", "b", 데이터$변수) :
만약 데이터의 변수값이 a면 b로 바꾸고 아니면 데이터의 원래 변수값 그대로 두어라
o is.na(데이터$변수) : 데이터의 변수가 결측치(na)이다.
o 데이터 <— 데이터 %>% drop_na() : 결측치(na)가 있는 행들을 다 제거
☆우선 library(tidyr) 먼저하기

o descriptive <— describe(데이터) : 해당데이터들을 계량적으로 계산후 descriptive에 할당
(mean, sd, min, max ...)
☆우선 library(psych) 먼저하기
o descriptive <— descriptive %>% mutate(LL= mean-3xsd, UL= mean+3xsd)
o 데이터 <—— 데이터 %>% filter(변수 < (k) UL값)
: 위에서 생성한 UL 값보다 작은 값들만 filter 이용해서 추출 (UL보다 큰 값은 이상치)
→ o 데이터 <—— 데이터 %>% mu...

회귀분석과정

→ 완벽하게 짝지어서 (결측치제외)

o corr.test(데이터[k,n], method="Pearson", alpha=0.05, use="pairwise.complete.obs")

(k가 공백이면 데이터의 전체 행을 의미하고 n열에서 c(7:14) → 7열~14열까지 (:) 으로 표현가능

우선 library(psych) 먼저하기

: 상관분석을 분석하는데 데이터에서 k열 n행끼리, 방법은 피어슨으로 실행

→ '+'로 여러변수나열가능

o 데이터저장이름 ← lm(DV~IV) data=데이터) : 다중회귀식 수립하기

o plot( ) : 도출한 4개의 그래프로 선형성, 정규성, 등분산성 확인

o ks.test(데이터$변수, pnorm, mean=mean(데이터$DV), sd=sd(데이터$DV)) : 데이터의 종속변수에서 정규분포를 그리는지 검정, 평균과 표준편차 이용

o shapiro.test(데이터$변수) : n이 작을 때의 데이터의 종속변수의 정규성 검토

o hist(데이터$변수, breaks = seq(N1, N2, k)) : 데이터의 변수를 범위를 N1~N2, 간격을 k 만큼하는 히스토그램 그리기 → (앞에 log를 추가하면 상대적으로 정규분포형태를 띈다.)

o durbinWatsonTest(데이터) : 오차의 자기상관 확인

```
#### 다중회귀분석 실습####
### 1) 상관분석 실행###
corr.test(bicycle[,c(7:14)], method = "pearson",
          alpha = 0.05, use="pairwise.complete.obs")
#bicycle 데이터프레임에서 모든 행에서 7열~14열까지
#---->결과:2p-value가 알파인 0.05와 비교했을 때
대부분의 값이 0이므로 모든 변수들 간에 상관관계와
상관계수 추정치가 통계적으로 유의하다. (2p-value<0.05)
#가장 강한 양의 상관관계:temp & atemp(0.99)

### 2) 연구가설 수립###
# IV: temp, atemp, humidity, windspeed, difference & DV
: total

#H1: temp -> total(+) #temp는 total에 양의 인과관계를
미친다/두 변수간에는 양의 인과관계를 갖고 있다./temp가
높아질 수록 total도 높아진다.
#H2: atemp -> total(+)
#H3: humidity -> total(-)
#H4: windspeed -> total(+)
#H5: difference -> total(+)

### 3) 다중회귀식 수립###
lm1 <- lm(total~temp+atemp+humidity+windspeed
+difference, data=bicycle)
#Yi(hat)=a + b1X1i + b2X2i + .... + b5X5i
#X1i=temp.....X5i=difference
```

lm1 ← lm(종속변수 ~ 독립변수)

```
### 4) 다중회귀분석 전제조건 확인:plot
plot(lm1)

#residuals vs fitted:빨간색 선이 하얀색 점선을 따라서
일직선이어야 선형성 조건 만족(그렇지 않음)#
#normal Q-Q: 점들이 대각선 위에 존재하면 정규성 조건
만족(그렇지 않음)
#scale-location: 빨간색 선이 일직선이어야 등분산성과
선형성 조건 만족(그렇지 않음)
#residuals vs leverage: leverage는 사례가 다른
사례로부터 떨어진 정도로 0에 가까울 수록 바람직함. 일부
사례가 떨어져 존재하여 이상치로 볼 수도 있음. 빨간선은
수평의 일직선이 바람직함(등분산성). cook's distance는 0
.5나 혹은 1을 넘으면 해당 사례가 회귀계수 추정치에
지나치게 많은 영향을 미침(그런 사례가 없음)
#6720, 6721, 6722, 8915, 8917, 8918을 제거하면 선형성
/정규성/등분산성이 개선됨(각 case 번호는 6728,6729..)

bicycle <- bicycle %>% filter(case !=6728, case !=6729,
case !=6730, case !=9004, case !=9006, case !=9007)
lm1 <- lm(total~temp+atemp+humidity+windspeed
+difference, data=bicycle) #6개 사례 제외한 bicycle로
업데이트

### 5) 정규성 조건 확인 ###
ks.test(bicycle$total, pnorm,
        mean=mean(bicycle$total), sd=sd(bicycle$total))
#Kolmogorov-Smirnov test: n이 클 때 정규성 검토. 2p
-value=0 <0.05 즉 유의하기에 정규성 조건 만족 못함.
shapiro.test(bicycle$total)
#shapiro.test: n이 작을 때 정규성 검토(현재는 n이
10723이기에 실행하면 오류가 뜸)

hist(bicycle$total, breaks = seq(0,1000,10))
hist(log(bicycle$total), breaks = seq(0,10,0.1))
#log: 밑이 e(무리수)인 자연로그

### 6) 독립성(오차의 자기상관) 검토###
library(car)
durbinWatsonTest(lm1)
# 오차는 서로 양의(0.9153073) 자기상관 존재(p-value=0
혹은 rho !=0, 즉 상관관계가 있고 독립성이지 않는다.)
#DW가 0.1693747로 2보다 많이 멀리 떨어져 있기에 오차의
자기 상관이 존재한다(2에 가까우면 존재X)0에 가깝기에 자기 상관이
존재한다. 4에 가까우면 음의 자기 상관 존재
```

No. Date.

<다중회귀분석 real 실행>

① 다중회귀식 추정 결과를 통한 가설검정

◦ summary(lm1) : summary 함수에 앞서 만든 회귀식을 넣어서 회귀계수 추정치(Estimate)와 유의성(P-value) 검토하기 (양측검정이니까 P-value vs α 형태, R에서 나온 P-value값은 2P-value이니까 2α(=0.05×2)=0.1과 비교)

② 추정된 다중회귀식 구하기

Y_i(hat) = 44.630 (Intercept의 estimate) + 1.125X_{1i} + 4.129X_{2i} ……

(①에서 나온 2P-value 값과 2α=0.1을 비교해서 유의한 변수들의 estimate를 이용해서 추정 다중회귀식 작성)

(Multiple R : 0.7612, adjusted R : 0.7611 큰 수치이기에 적합도, 설명력이 높은 회귀식이다)

(P-value < 2.2e-16으로 0에 가깝고 유의 즉, 모든 회귀계수가 0이다인 귀무가설 기각하고 적어도 하나의 회귀계수는 0이 아니다인 대립가설 채택)

③ 모형적합도 제고를 위한 다중회귀식 추정방법

◦ lm1_f <- step(lm1, direction="foward") : lm1 데이터로 'foward' 다중회귀식 추정

◦ lm1_b <- step(lm1, direction="backward") : lm1 데이터로 'backward' 다중회귀식 추정

◦ lm1_s <- step(lm1, direction="both") : lm1 데이터로 'foward', 'backward' 혼합

◦ summary(lm1_f) : summary 함수를 이용해서 위에 데이터 형성할 때 나온 Start : AIC=K 의 값을 각각 비교해서 작은 값이 나온 방법을 택함 (결과가 3가지 방법 다 아! 고로 밑의 과정은 변수를 제거하는 수가 다함)

④ 다중공선성 확인

◦ vif(lm1) : vif 함수를 이용해서 다중공선성을 확인

* 우선 library(car) 먼저하기

이후 vif의 값이 5.3을 넘고 그중 제일 값이 큰 변수를 제외하고 앞선 과정 실시하기 (atemp 변수 제외)

lm2 <- lm(total ~ temp + humidity + windspeed + difference, data=bicycle)

vif(lm2) : atemp 변수 제외하고 만든 회귀식의 다중공선성 확인

summary(lm2) : 새로운 회귀식의 요약치 검토 (P-value, estimate ……)

* (동일한 방법으로 temp 변수도 제외한 데이터 lm3을 만들고 vif, summary로 검사)

⑤ IV의 중요도 (표준화 회귀계수 추정치의 절대값 크기)

◦ lm.beta(lm1) : 표준화 회귀계수 추정치들을 절대값 기준으로 크기 비교해서 중요도 확인

⑥ 새로운 변수(IV) 추가 타당성 검토 — * 우선 library(lm.beta) 먼저하기

◦ lm4 <- lm(total ~ humidity + windspeed + difference + working, data=bicycle) : working 추가

◦ summary(lm4) : R-squared 와 adjusted R-squared의 값을 확인해서 추가 타당성 검토

◦ anova(lm3, lm4) : R2의 변화량인 0.0485가 통계적으로 유의한지 확인

⊕ 이후 summary를 이용해서 lm4에 대한 추정결과 자세히 확인

◦ lm5 <- lm(total ~ humidity + windspeed + difference + working + season, data=bicycle)

: 두번째 추가 IV로 season 추가하기

No. Date.

o summary (lm5)

o anova (lm4, lm5) : R^2의 변화량인 0.0348이 통계적으로 유의한지 확인

⊕ 이후 summary를 이용해서 lm5에 대한 추정법과 자세히 확인

o vif(lm5) : 다중공선성 확인

☆ 우선 library (car) 인식하기

GVIF^(1/(2xDf))의 값이 2보다 작으면 다중공선성은 문제없음

⑦ 새로운 독립변수 atemp 추가 문제 확인

o lm6 <— lm(total ~ humidity + windspeed + difference + working + season + atemp, data...

: atemp 변수 추가

o summary(lm6) :

o anova (lm5, lm6) : R^2의 변화량인 0.032가 통계적으로 유의한지 확인

o vif(lm6) : 다중공선성 확인

o lm.beta(lm6) : 각 독립변수의 상대적 중요도를 절댓값 기준으로 확인

⑧ 가장 설명력이 좋은 lm5에 기반하여 IV 측정값이 존재할때 종속변수(예측치) 예측

: lm5 회귀식에다가 부여한 측정값 넣어서 예측치 구함

→ 새로운 독립변수를 추가했을때 모형설명력과 적합도가 좋아졌다고 무조건 이 변수를 추가하는것이 아님. 상대적으로 중요도가 떨어지는 변수들에 의해 왜곡이 발생한것(season fall은 ... 음으로 잘못 나옴)

⑨ 조절효과 확인하기

o bicycle $working <— as.numeric (bicycle $ working) : working 변수를 수치형으로 변경

o bicycle <— bicycle %>% mutate(inter = humidity * working) : 조절효과에 쓰일 새변...

o lm7 <— lm(total ~ humidity + windspeed + difference + working + inter, data = bicycle)

: inter를 추가한 회귀식 만듦 (lm4에서 inter 추가)

o summary (lm7)

o anova (lm4, lm7) : R^2의 변화량인 0.0122가 통계적으로 유의한지 확인

☆☆ 이후 조절효과를 상세하게 확인

No. Date.

<경데분2 기말정리> <로짓회귀분석>

- 데이터파일이름 <— read_csv("파일.csv") : 데이터 파일불러오기
  - └ 우선 library(readr) 먼저하기
- summary(데이터) : 척도 및 NA 존재 확인하기
- str(데이터) : 해당 데이터의 척도상세히 (ex) num, chr . . .)
- desc <— describe(데이터[A:B]) : 데이터에서 A ~ B 까지의 순서의 변수의 기술통계량 확인
- desc <— desc %>% mutate (UL=mean+3*sd, LL=mean-3*sd) : 이상치 기준추가
  - └ 우선 library(dplyr) / library(psych) 먼저하기
- logit1 <— glm(종속변수 ~ 독립변수, data=데이터이름, family=binomial()) : 로짓회귀식수립
- outlierTest(logit1) : 이상치 확인하기 (위에 로짓회귀식을 바탕으로) → 우선 library(car) 먼저하기
- 데이터 <— 데이터 % filter(변수~조건) : 변수가 조건에 맞는 행들만 추출해서 데이터에 할당
  - └ 우선 library(dplyr) 먼저하기
- summary(logit1) : 로짓회귀식에서 중요한 결과들을 설명
- hoslem.test(logit1$y, logit1$fitted.values) : logit1의 모형적합도 확인
  - └ 우선 library(ResourceSelection) 먼저하기
- PseudoR2(logit1, which= c("CoxSnell", "Nagelkerke")) : logit1의 모형적합도 확인
  - └ 우선 library(DescTools) 먼저하기

→ 이후 logit1에 intern변수 추가한 logit2 만들기 → 이상치확인후 제거하고 로짓회귀식 다시수립

→ summary로 각각의 -2LL을 비교해서 logit2가 더 적절함을 확인후 증명

- difference <— logit1$deviance - logit2$deviance
- dof <— logit1$df.residual - logit2$df.residual
- 1 - pchisq(difference, dof)
  - └ 나오는 값으로 적합도확인

→ 이후 logit2 회귀식을 대상으로 hoslem.test 함수와 PseudoR2 함수 이용해서 모형적합도 확인

- prediction <— predict(logit2, newdata= employ) : logit value 도출
- prediction <— ifelse(prediction<0, 0, 1) : 위에 값을 0을 기준으로 0, 1로 변경
- employ$result <— as.factor(employ$result) : result의 척도를 범주형으로 변경 (prediction도 동일하게 변경)
- confusionMatrix(employ$result, prediction) : 결과를 통해 hit ratio 구하기 (=accuracy
- C1001 <— data.frame(id=1001, GPA=3.5, . . . ~) : 문제에 맞게 만들기
- predict(logit2, C1001) : logit2를 이용해서 합격여부예측
  - └ 생성한 모델에 예측을 원하는 사람 적용

<군집분석> 1-계층군집분석 데이터 이름

o distance_hr <- dist(CHRA_hr, method="power")
; 사례간의 거리측정하여 객체생성
L 위한 library(cluster) 먼저하기
o HRA_CA_hr <- hclust(distance_hr, method="ward.D2") ; 위에서 만든 객체에 hclust 실행
o plot(HRA_CA_hr, cex="색깔이름", main="HRA") ; 덴드로그램 만들기
o HRA_hr$변수 <- as.numeric(HRA_hr$변수) ; 계량형 척도로 변경
o set.seed(K) ; 실행할때마다 결과가 달라지는것을 막기위해 실행 (K는 아무 차수 아)
o HRA_hr_NC1 <- NbClust(HRA_hr, distance="euclidean", min.nc=2, max.nc=6, method="average") ; 최소 ~ 최대 6을 설정하고 최적의 군집개수 산출
o HRA_hr_HCA <- cutree(HRA_CA_hr, 9) ; 덴드로그램에서 9개만큼 잘라주기 (군집분석방식)
L HRA_hr_NC1에서 나온 숫자를
o result_hr <- aggregate(HRA_hr, by=list(cluster=HRA_hr_HCA), mean)
; 군집별 차이와 특징 도출
→ library(NbClust) 먼저하기

2-비계층군집분석
o HRA_hr_K <- scale(HRA_hr_k) ; 변수의 표준화한 데이터 만들기
o HRA_hr_NC2 <- NbClust(HRA_hr_K, distance="euclidean", min.nc=2, max.nc=6, method="kmeans") ; 최적의 군집개수 구하기
o HRA_hr_KCA <- kmeans(HRA_hr_K, centers=(K), nstart=25)
; 위에서 구한 최적의 군집 개수 K를 적용하여 비계층적 군집분석 실행
o result_hr2 <- aggregate(HRA_hr, by=list(cluster=HRA_hr_KCA$cluster), mean) ; 군집별 차이와 특징 도출
우선 library(NbClust) 실행

<knn>
o cancer_z <- as.data.frame(scale(cancer[2:31])) ; 1열인 종속변수를 제외하고 나머지 IV변수들을 표준화해서 데이터 생성 [-1] : 1행을 제외
o ind <- sample(2, nrow(cancer_z), replace=T, prob=c(0.7, 0.3)) ; test와 train을 구분하는 ind 객체생성 [행의개수 복원추출 [0.7, 0.3비율
o cancer_train <- cancer_z[ind=1,]
o cancer_test <- cancer_z[ind=2,] > train과 test 나뉘어 생성

No. Date.

- grid1 <- expand.grid(k=3:10) : k범위를 3~10 에서설정
  - 위의 library(caret) 먼저하기
- control <- trainControl(method="repeatedcv", number=10, repeats=3) : 훈련방법제정
- knn.train <- train(diagnosis~., data=cancer_train, method="knn", trControl=control, tuneGrid=grid1) : 최적의 K값 선정
  - 위에 모든변수 다사용 "~."
  - library(caret)
- varImp(knn.train, scale=F) : 변수중요도확인
  - 앞서 scale 적용했으면
- pred.test <- predict(knn.train, newdata=cancer_test) ;
  생성한 knn.train을 test에 데이터에 적용해서 성능판단
- confusionMatrix(pred.test, cancer_test$diagnosis) : 실제값 vs 예측값
  - library(caret)
  - pred.test : 예측값
  - cancer_test$diagnosis : 실제값
- kknn.train <- train.kknn(diagnosis~., data=cancer_train, kmax=25, distance=2, kernel="cc" "다양한 10가지 방법 적성"))
  : 최적의 K값과 방법찾기
  - distance=2 : 유클리디안거리
- kknn.pred.test <- predict(kknn.train, newdata=cancer_test)
  : 개선된 모델과 실제값적용해서비교
- confusionMatrix(kknn.pred.test, cancer_test$diagnosis)
  : 개선된 모델로 실제값 vs 예측값

<SVM실습>

- linear.svm <- tune.svm(diagnosis~., data=cancer_svm_train, kernel="linear", cost=c(0.1, 0.25, 0.5, 0.75, 1, 2, 3, 4, 5, 7, 10))
  : 선형(linear) 방식으로 svm 훈련 → 옵션(파라미터) 값지정
  - 위의 library(e1071) 먼저하기
- linear.test <- predict(linear.svm$best.model, newdata=cancer_svm_test) : 생성한 linear의 최적의 모델을 test에 적용해서 성능판단
- confusionMatrix(linear.test, cancer_svm_test$diagnosis)
  : Accuracy, kappa외같은 값들확인 → 예측값 vs 실제값
  - library(caret) 먼저하기

=> 나머지 3가지 방법동일하게진행