1.

전처리 이전

	DUID PANEL # + ENCRYPTED DU IDENTIFIER	PID PERSON NUMBER PORTION OF PERSID	DUPERSID PERSON ID (DUID + PID)	PANEL PANEL NUMBER	DATAYEAR SURVEY DATA YEAR	FAMID31 FAMILY ID (STUDENT MERGED IN) - R3/1
1	2460002	101	2460002101	24	2022	A
2	2460006	101	2460006101	24	2022	A
3	2460006	102	2460006102	24	2022	A
4	2460010	101	2460010101	24	2022	A
5	2460018	101	2460018101	24	2022	A
6	2460024	101	2460024101	24	2022	A
7	2460026	101	2460026101	24	2022	A
8	2460026	103	2460026103	24	2022	A
9	2460029	101	2460029101	24	2022	A
10	2460029	102	2460029102	24	2022	A

Showing 1 to 10 of 22,431 entries, 1420 total columns

전처리 이후

	bmi	age	gender	race	education	health	limitation	region	private	visits_hosp	diabetes	stroke	cancer	income
1	35.7	77	0	3	6	7	0	3	0	1	1	1	1	22.000
2	27.4	51	0	2	16	8	0	2	1	0	0	0	0	45.000
3	20.8	58	1	2	16	7	0	5	0	0	0	0	0	47.405
4	43.3	53	0	2	12	6	0	2	1	1	0	0	1	86.500
5	23.7	69	1	3	16	5	0	2	1	1	0	0	1	11.500
6	22.3	37	1	2	16	6	0	5	0	0	0	0	0	38.989
7	27.4	37	0	1	16	7	0	5	0	0	0	0	0	163.360
8	18.9	81	0	2	13	7	0	2	0	1	0	0	0	50.800
9	35.8	75	0	2	14	5	0	2	0	1	1	0	0	22.500
10	36.4	60	0	2	16	5	0	4	1	1	0	0	0	84.175

Showing 1 to 10 of 10,494 entries, 14 total columns

2.a

```
> # 모델 결과 요약
> summary(fit1)

Call:
glm(formula = visits_hosp ~ bmi + age + gender + race + education +
    health + limitation + region + private + diabetes + stroke +
    cancer + income, family = binomial(link = "logit"), data = d2022_1)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.8727636  0.2063184 -18.771  < 2e-16 ***
bmi          0.0145390  0.0035074   4.145 3.39e-05 ***
age          0.0203464  0.0015426  13.190  < 2e-16 ***
gender      -0.4084043  0.0455597  -8.964  < 2e-16 ***
race2        0.6564477  0.0785218   8.360  < 2e-16 ***
race3        0.4493645  0.0964018   4.661 3.14e-06 ***
race4        0.3106272  0.1263946   2.458 0.013987 *
race5        0.6396252  0.1516212   4.219 2.46e-05 ***
education    0.0694073  0.0094631   7.335 2.23e-13 ***
health6      0.0403552  0.0590808   0.683 0.494575
health7      0.1385430  0.0611431   2.266 0.023459 *
health8      0.3390040  0.0866716   3.911 9.18e-05 ***
health9     -0.1259653  0.1872800  -0.673 0.501199
limitation1  0.4696704  0.0620127   7.574 3.63e-14 ***
region3      0.2729724  0.0711239   3.838 0.000124 ***
region4     -0.5122166  0.0681964  -7.511 5.87e-14 ***
region5      0.0904287  0.0702862   1.287 0.198242
private      0.0532504  0.0511386   1.041 0.297739
diabetes     0.2513947  0.0636752   3.948 7.88e-05 ***
stroke       0.2327072  0.0940908   2.473 0.013390 *
cancer       0.6149931  0.0609450  10.091  < 2e-16 ***
income       0.0010913  0.0004444   2.455 0.014073 *
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13439  on 10493  degrees of freedom
Residual deviance: 11993  on 10472  degrees of freedom
AIC: 12037

Number of Fisher Scoring iterations: 4

>
> # 모델 해석을 위한 오즈비 계산
> exp(coef(fit1))  # 각 계수의 오즈비
(Intercept)         bmi         age      gender
 0.02080081  1.01464524  1.02055479  0.66471011
      race2       race3       race4       race5
 1.92793153  1.56731587  1.36428052  1.89577016
  education     health6     health7     health8
 1.07187270  1.04118054  1.14859908  1.40354903
    health9 limitation1     region3     region4
 0.88164541  1.59946699  1.31386399  0.59916600
    region5     private    diabetes      stroke
 1.09464343  1.05469367  1.28581751  1.26201191
     cancer      income
 1.84964391  1.00109187
>
```

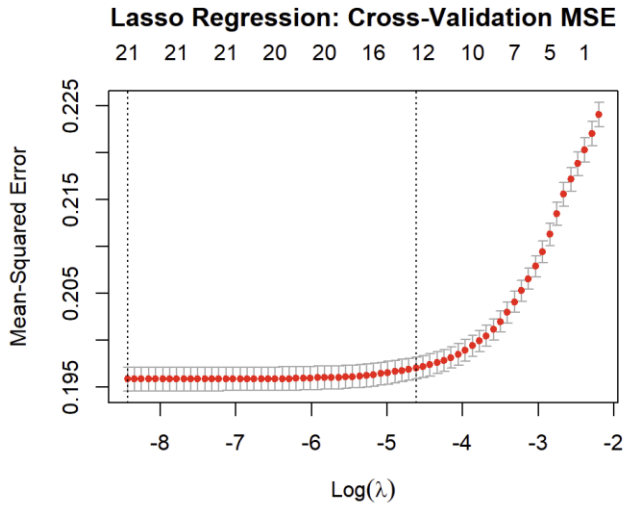
2.b

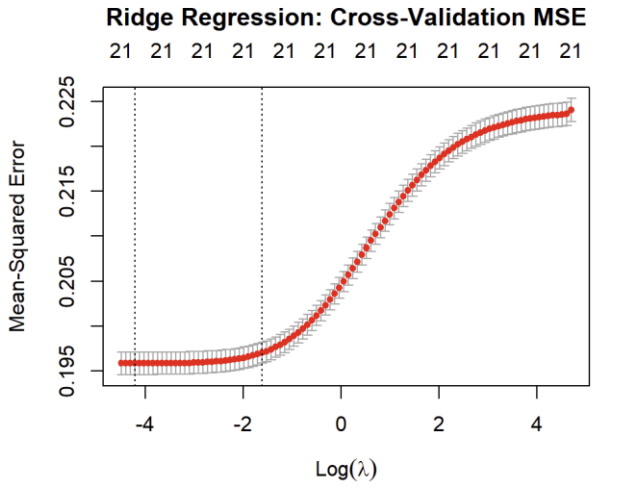
```
> #lasso
> set.seed(123)
> lasso_cv <- cv.glmnet(x, y, alpha = 1, nfolds = 20, family = "gaussian")
> lasso_lambda_min <- lasso_cv$lambda.min
> lasso_coef <- coef(lasso_cv, s = "lambda.min")
> lasso_num_vars <- sum(lasso_coef != 0) - 1  # Intercept 제외
>
>
> cat("Lasso 최적 lambda:", lasso_lambda_min, "\n")
Lasso 최적 lambda: 0.0002179719
> cat("Lasso 선택된 변수 수:", sum(lasso_coef != 0) - 1, "\n") # Intercept 제외
Lasso 선택된 변수 수: 21
>
>
> #ridge
> set.seed(123)
> ridge_cv <- cv.glmnet(x, y, alpha = 0, nfolds = 20, family = "gaussian")
> ridge_lambda_min <- ridge_cv$lambda.min
> ridge_coef <- coef(ridge_cv, s = "lambda.min")
> ridge_num_vars <- sum(ridge_coef != 0) - 1
>
>
> cat("Ridge 최적 lambda:", ridge_lambda_min, "\n")
Ridge 최적 lambda: 0.01467856
> cat("Ridge 선택된 변수 수:", sum(ridge_coef != 0) - 1, "\n")
Ridge 선택된 변수 수: 21
```

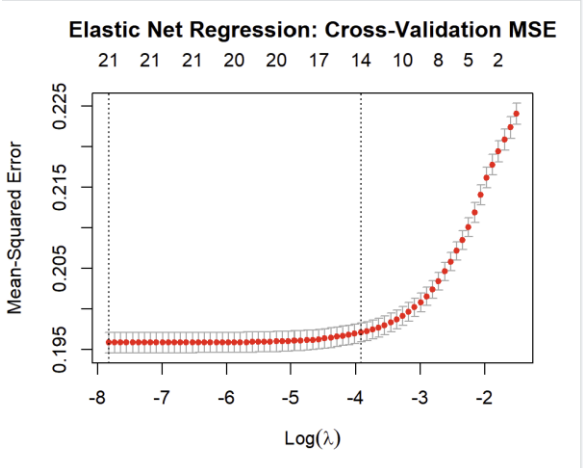
```
> #elastic net
> set.seed(123)
> elastic_cv <- cv.glmnet(x, y, alpha = 0.5, nfolds = 20, family = "gaussian")
> elastic_lambda_min <- elastic_cv$lambda.min
> elastic_coef <- coef(elastic_cv, s = "lambda.min")
> elastic_num_vars <- sum(elastic_coef != 0) - 1
>
>
> cat("Elastic Net 최적 lambda:", elastic_lambda_min, "\n")
Elastic Net 최적 lambda: 0.0003972158
> cat("Elastic Net 선택된 변수 수:", sum(elastic_coef != 0) - 1, "\n")
Elastic Net 선택된 변수 수: 21
>
>
> #MSE 비교
> cat("Lasso 최소 MSE:", min(lasso_cv$cvm), "\n")
Lasso 최소 MSE: 0.1958869
> cat("Ridge 최소 MSE:", min(ridge_cv$cvm), "\n")
Ridge 최소 MSE: 0.1958701
> cat("Elastic Net 최소 MSE:", min(elastic_cv$cvm), "\n")
Elastic Net 최소 MSE: 0.1958863
>
```

2.c.

```
> cat("Lasso 최적 lambda:", lasso_lambda_min, "선택된 변수 수:", lasso_num_vars, "\n")
Lasso 최적 lambda: 0.0002179719 선택된 변수 수: 21
> cat("Ridge 최적 lambda:", ridge_lambda_min, "선택된 변수 수:", ridge_num_vars, "\n")
Ridge 최적 lambda: 0.01467856 선택된 변수 수: 21
> cat("Elastic Net 최적 lambda:", elastic_lambda_min, "선택된 변수 수:", elastic_num_vars, "\n")
Elastic Net 최적 lambda: 0.0003972158 선택된 변수 수: 21
```







3.a

```
> summary(logistic_model)

Call:
glm(formula = train_y ~ ., family = binomial, data = train_x)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.184438   0.280077 -22.081  < 2e-16 ***
bmi          0.090300   0.004753  18.996  < 2e-16 ***
age          0.045841   0.002377  19.285  < 2e-16 ***
gender       0.239655   0.066513   3.603 0.000314 ***
education   -0.068830   0.011298  -6.092 1.11e-09 ***
visits_hosp  0.216492   0.068579   3.157 0.001595 **
income      -0.005099   0.000892  -5.717 1.09e-08 ***
stroke       0.537424   0.112113   4.794 1.64e-06 ***
cancer      -0.296692   0.087861  -3.377 0.000733 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7182.9  on 8395  degrees of freedom
Residual deviance: 6119.8  on 8387  degrees of freedom
AIC: 6137.8

Number of Fisher Scoring iterations: 5
```

```
> # 혼동행렬 생성
> conf_matrix <- confusionMatrix(as.factor(pred_class), as.factor(test_y))
>
> # 오분류율(Misclassification Rate) 계산
> misclassification_rate <- 1 - conf_matrix$overall["Accuracy"]
>
> # 결과 출력
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1744  300
         1   33   21

               Accuracy : 0.8413
                 95% CI : (0.8249, 0.8567)
    No Information Rate : 0.847
    P-Value [Acc > NIR] : 0.7767

                  Kappa : 0.0711

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.98143
            Specificity : 0.06542
         Pos Pred Value : 0.85323
         Neg Pred Value : 0.38889
             Prevalence : 0.84700
         Detection Rate : 0.83127
   Detection Prevalence : 0.97426
      Balanced Accuracy : 0.52342

       'Positive' Class : 0

> cat("Out-of-Sample Misclassification Rate:", misclassification_rate, "\n")
Out-of-Sample Misclassification Rate: 0.1587226
>
```

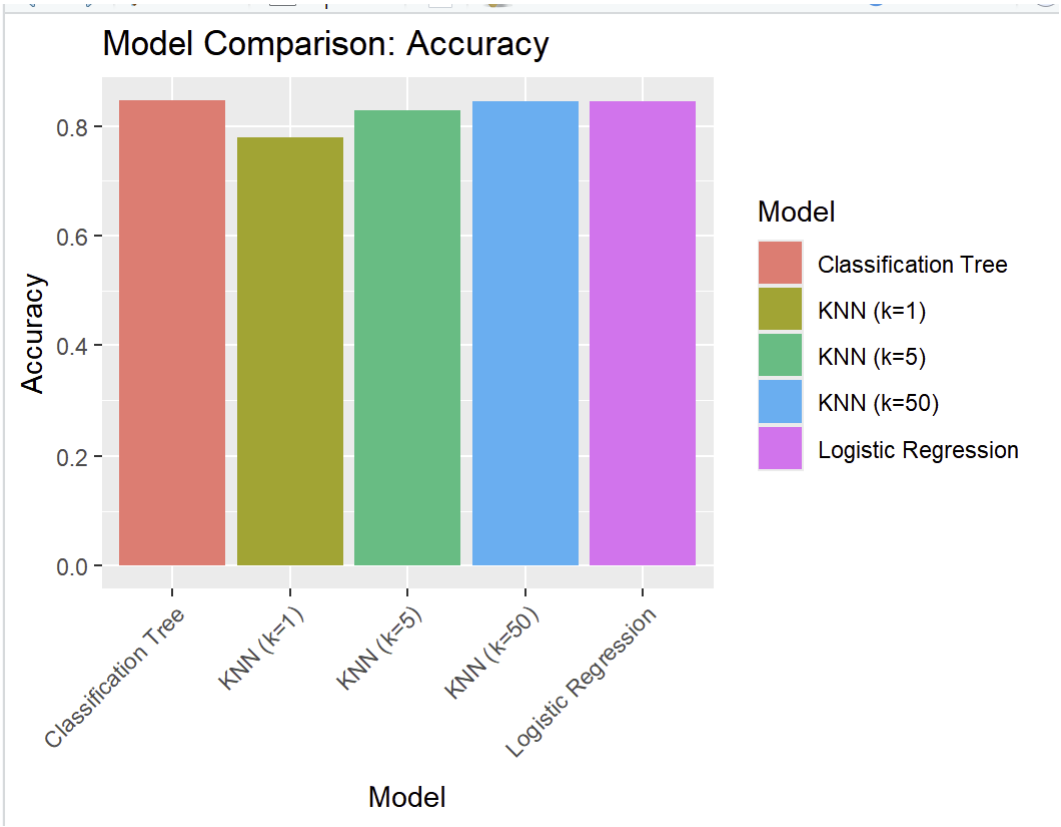
3.b.

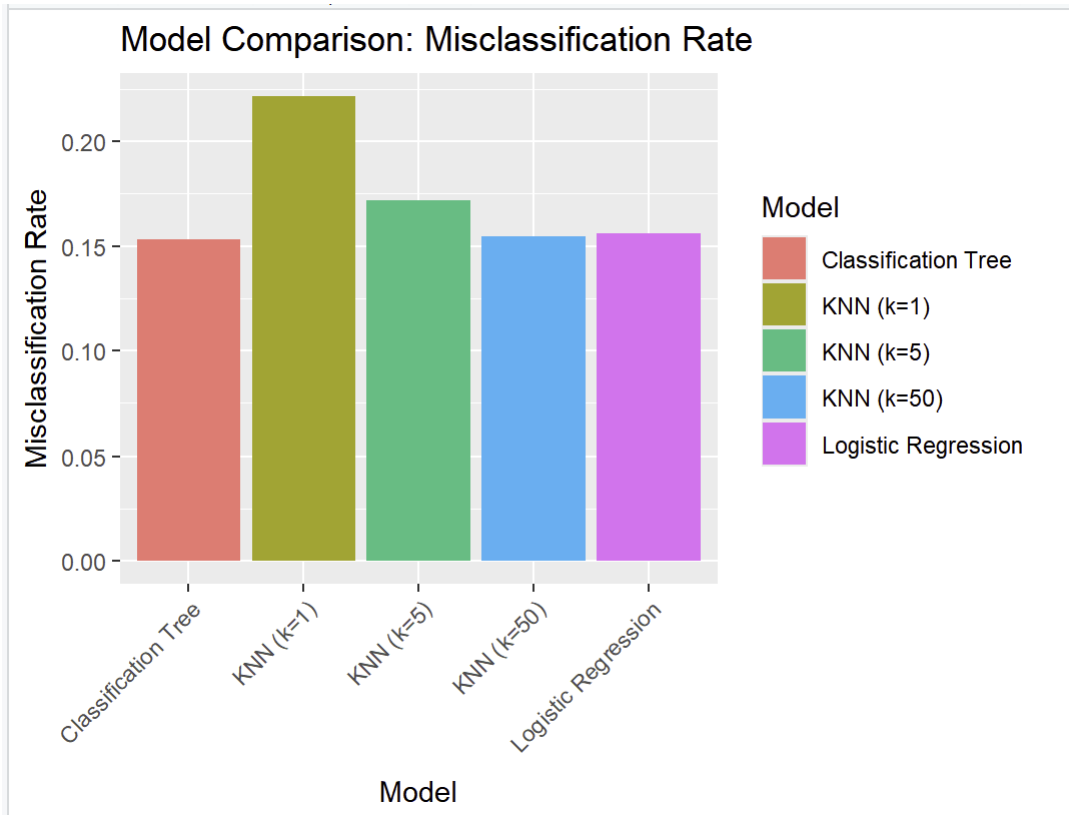
```
> # k = 1
> set.seed(123)
> pred_knn1 <- knn(train = train_x, test = test_x, cl = train_y, k = 1)
> conf_matrix1 <- confusionMatrix(pred_knn1, test_y)
> misclassification_rate1 <- 1 - conf_matrix1$overall["Accuracy"]
>
> # k = 5
> set.seed(123)
> pred_knn5 <- knn(train = train_x, test = test_x, cl = train_y, k = 5)
> conf_matrix5 <- confusionMatrix(pred_knn5, test_y)
> misclassification_rate5 <- 1 - conf_matrix5$overall["Accuracy"]
>
> # k = 50
> set.seed(123)
> pred_knn50 <- knn(train = train_x, test = test_x, cl = train_y, k = 50)
> conf_matrix50 <- confusionMatrix(pred_knn50, test_y)
> misclassification_rate50 <- 1 - conf_matrix50$overall["Accuracy"]
>
> # 결과 출력
> cat("KNN (k = 1) Misclassification Rate:", misclassification_rate1, "\n")
KNN (k = 1) Misclassification Rate: 0.2216397
> cat("KNN (k = 5) Misclassification Rate:", misclassification_rate5, "\n")
KNN (k = 5) Misclassification Rate: 0.1720686
> cat("KNN (k = 50) Misclassification Rate:", misclassification_rate50, "\n")
KNN (k = 50) Misclassification Rate: 0.1553861
>
```

3.c.

```
> print(conf_matrix)
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1777  321
         1    0    0

               Accuracy : 0.847
                 95% CI : (0.8309, 0.8621)
    No Information Rate : 0.847
    P-Value [Acc > NIR] : 0.5149

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.000
            Specificity : 0.000
         Pos Pred Value : 0.847
         Neg Pred Value :   NaN
             Prevalence : 0.847
         Detection Rate : 0.847
   Detection Prevalence : 1.000
      Balanced Accuracy : 0.500

       'Positive' Class : 0

> cat("Out-of-Sample Misclassification Rate:", misclassification_rate, "\n")
Out-of-Sample Misclassification Rate: 0.1530029
>
```

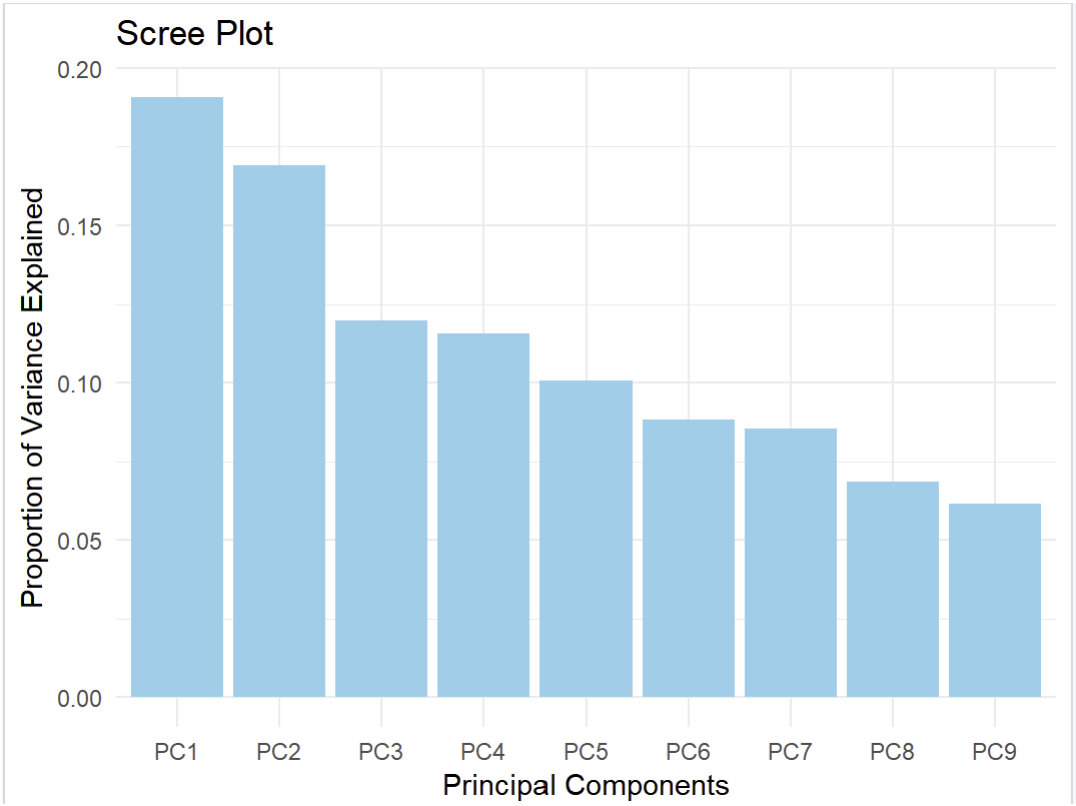
3.d.


Model Comparison: Accuracy
Accuracy
0.0
0.2
0.4
0.6
0.8
Classification Tree
KNN (k=1)
KNN (k=5)
KNN (k=50)
Logistic Regression
Model
Model
Classification Tree
KNN (k=1)
KNN (k=5)
KNN (k=50)
Logistic Regression


Model Comparison: Misclassification Rate
Misclassification Rate
0.00
0.05
0.10
0.15
0.20
Classification Tree
KNN (k=1)
KNN (k=5)
KNN (k=50)
Logistic Regression
Model
Model
Classification Tree
KNN (k=1)
KNN (k=5)
KNN (k=50)
Logistic Regression

4.b.

```
> # PCA 결과 요약
> summary_pca <- summary(pca_result)
> print(summary_pca)
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6
Standard deviation     1.3104 1.2335 1.0387 1.0202 0.9523 0.89139
Proportion of Variance 0.1908 0.1691 0.1199 0.1157 0.1008 0.08829
Cumulative Proportion  0.1908 0.3598 0.4797 0.5954 0.6961 0.78442
                           PC7     PC8     PC9
Standard deviation     0.87683 0.78626 0.74376
Proportion of Variance 0.08543 0.06869 0.06147
Cumulative Proportion  0.86984 0.93853 1.00000
>
> # 첫 번째와 두 번째 주성분의 설명된 분산 비율
> explained_variance <- summary_pca$importance[2, 1:2]
> cat("첫 번째 주성분이 설명하는 변동 비율:", explained_variance[1], "\n")
첫 번째 주성분이 설명하는 변동 비율: 0.19079
> cat("두 번째 주성분이 설명하는 변동 비율:", explained_variance[2], "\n")
두 번째 주성분이 설명하는 변동 비율: 0.16905
>
> # 누적 설명 비율
> cumulative_variance <- summary_pca$importance[3, 1:2]
> cat("첫 두 주성분의 누적 설명 비율:", cumulative_variance[2], "\n")
첫 두 주성분의 누적 설명 비율: 0.35984
>
```



4.c.

```
> pca_result
Standard deviations (1, .., p=9):
[1] 1.3103935 1.2334674 1.0387167 1.0202138 0.9522880 0.8913905 0.8768299
[8] 0.7862648 0.7437650

Rotation (n x k) = (9 x 9):
                  PC1         PC2         PC3         PC4         PC5
diabetes    -0.4093126 -0.19053141  0.45458689 -0.09067645 -0.01112381
bmi         -0.1608793 -0.25348264  0.50420682 -0.54946410 -0.08449603
age         -0.5480333  0.21786877 -0.03554678  0.17968416 -0.11672954
gender       0.1230439  0.08633379  0.59901606  0.59523180 -0.25702858
education    0.1797520  0.58264267  0.11091285 -0.25170400  0.21433266
visits_hosp -0.3603704  0.31091522 -0.07642761 -0.34846910 -0.01434468
income       0.2201437  0.55089262  0.34732585 -0.06732708  0.09791160
stroke      -0.3435586 -0.01852992  0.07338166  0.29921850  0.82650851
cancer      -0.4030229  0.32335309 -0.18676049  0.15629915 -0.41737267
                  PC6         PC7         PC8         PC9
diabetes    -0.62084750 -0.12183412 -0.34448508 -0.2494006
bmi          0.42073979  0.35225448  0.15567777  0.1457695
age         -0.25285408  0.07334747  0.45484298  0.5734281
gender       0.30832108 -0.24621840 -0.13397352  0.1568042
education   -0.09805048  0.17260018 -0.52839430  0.4336086
visits_hosp  0.33300812 -0.72356923 -0.02033300 -0.1054845
income      -0.18179395  0.08718414  0.52488704 -0.4440076
stroke       0.26546219  0.13513706 -0.03103315 -0.1178812
cancer       0.23345626  0.46461116 -0.27600722 -0.3911025
>
```

4.d.

```
> # 결과 확인
> print(contribution_df)
               Variable       PC1        PC2
diabetes       diabetes 14.894295  7.5173661
bmi                 bmi  5.854167 10.0010899
age                 age 19.942141  8.5959544
gender           gender  4.477391  3.4062768
education     education  6.540914 22.9880112
visits_hosp visits_hosp 13.113359 12.2670772
income           income  8.010711 21.7353214
stroke           stroke 12.501602  0.7310931
cancer           cancer 14.665420 12.7578099
>
```