second_semester

Prabhat Ale

2024-05-31

Q.8) Question no was changed due to some mistake in numbering in the paper

```
library(car)
```

```
## Loading required package: carData
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data <- Arrests
```

taking colour, age, sex, employed and citizen as independent variable and released as dependent variable

Converting the target variable into numeric values

and then again converting them as factor variables.

```
data$released <- ifelse(data$released == "Yes", 1, 0)

data$released <- as.factor(data$released)

final_data <- select(data, "released", "colour", "age",
                     "sex", "employed", "citizen")

table(final_data$colour)
```

```
##
## Black White
##  1288  3938
```

Converting categorical variables colour into binary intergers and then as factors

```
final_data$colour <- ifelse(final_data$colour == "Black", 0, 1)
final_data$colour <- as.factor(final_data$colour)
```

Converting categorical variables gender female as 0 and male as 1

```
final_data$sex <- ifelse(final_data$sex == "Female", 0, 1)
final_data$sex <- as.factor(final_data$sex)
```

Converting categorical variables employed no as 0 and yes as 1

```
final_data$employed <- ifelse(final_data$employed == "No", 0, 1)
final_data$employed <- as.factor(final_data$employed)
```

Converting categorical variables citizen no as 0 and yes as 1

```
final_data$citizen <- ifelse(final_data$citizen == "No", 0, 1)
final_data$citizen <- as.factor(final_data$citizen)
```

Divide the data into train and test sets using 80:20 random splits

set.seed(22)

```
ind <- sample(2, size = nrow(final_data), replace = T, prob = c(0.8, 0.2))

train.data <- final_data[ind == 1,] test.data <- final_data[ind == 2, ]

library(ggplot2)


create_plot <- function(x, y, title) { ggplot(data, aes(x = !!x, y = !!y)) + geom_point(color = 'red') + labs(title = title) + theme_bw() }

plot1<- create_plot(final_data$released, final_data$colour, "released vs colour") plot2<- create_plot(final_data$released, final_data$age, "released vs age") plot3<- create_plot(final_data $released, final_data$sex, "released vs sex") plot4<- create_plot(final_data$released, final_data$ employed, "released vs employed") plot5<- create_plot(final_data$released, final_data$citizen, "released vs citizen")

library(gridExtra)

grid.arrange(plot1, plot2, plot3, plot4, plot5, nrow = 3, ncol = 2)
```

From this graph, we can see that there exist non linear relationship between the target variable and independent variable.

Apply multivariate logistic regression

```
model.lr <- glm(released~., data = final_data, family = 'binomial' )

summary(model.lr)
```

Checking whether there exist mulitcollinearity in the features.

TO check it, we have VIF(Variance Inflation Factor).

IF the VIF value of the feautures in a model is greater than 2, then we can remove such feature as it suffers from multicolinearity issue.

```
vif(model.lr)
```

Since, the vif of all of these features are less than 2, we can say that these features are not collinear to each other.

Making predictions on the test datasets

```
predictions <- predict(model.lr, test.data)
```

```
predictions <- ifelse(predictions > 0.5, 1, 0)
```

Confusion Matrix

```
library(caret)
```

```
confusionMatrix(as.factor(predictions), test.data$released)
```

The accuracy of the logistic regression model is 0.8366 and specificity is 0.98 and sensitivity is 0.05.

naive bayes algorithm

```
library(e1071)
```

```
model.nb <- naiveBayes(released~., data = train.data)
```

```
predictions <- predict(model.nb, test.data)
```

```
confusionMatrix(as.factor(predictions), test.data$released)
```

The accuracy of the naive bayes model is 0.8366 and specificity is 0.98 and sensitivity is 0.05.

Both model yielded the same results for this data

Q.no 9)

Loading iris datasets

```
iris_data <- iris
```

```
head(iris_data)
```

```
iris_data
```

We need to combine the murder, assault and rape variables and create a latent variable

i.e. “criminality” score using these three variables.

```
library(dplyr)
```

Removing the last label of the iris datasets which contains flower species.

Scaling is required for PCA so scaling the data as well after removing the target columns

iris_data.1 <- iris_data[, -5] %>% scale

importing factorextra and FactoMineR for computing eigenvalues and PCA

library(factoextra) library(FactoMineR)

Checking the strucutre of iris datasets

str(iris_data.1)

Visualizing top 6 rows

head(iris_data.1)

res.pca <- PCA(iris_data.1, graph = F)

res.pca$eig

Here Eigenvalue > 1 = Only one component

According to Kaiser's criteria, the principal component is the only component which has an eigne value greater than 1.

We can see that the comp 1 has an eigenvalue of 2.91 which is greater than 1

and more than 72.9% of variance is explained by this component alone.

For computing var_explained we compute pca using prcomp in iris datasets

pca.1 <- prcomp(iris_data.1) summary(pca.1)

So how many components to retain

var_explained = pca.1$sdev^2 / sum(pca.1$sdev^2)

Creating a Scree Plot

library(ggplot2)

qplot(c(1:4), var_explained) + geom_line() + xlab("Principal Component") + ylab("Variance Explained") + ggtitle("Scree Plot") + ylim(0,1)

It will be wise to use Kaiser's rule and Scree plot to decide how many components

to retain for the problem in hand! We can use scree plot's suggestion for now.

Three components should be used PC1, PC2 and PC2 if we use scree plot.

PC1 = SIngle component using Kaiser's criteria of EV > 1

This is "confirmative"

d)

library(psych) ## We can try PCA with "VARIMAX" rotation

Rotated PCA with variance maximization

fa <- psych::principal(iris_data.1, nfactors = 3, rotate = "varimax")

fa

After apply varimax rotation, we got three eigen values whose

eigenvector is greater than 1. So,

biplot(fa, labels = rownames(iris_data.1))

Note:

PCA must be used to produce “orthogonal” components

PCA with “varimax” rotation also produced “orthogonal” components

PCA with “varimax” rotation can not be interpreted as a “true” PCA.

If we need to get a latent variable with “correlated” components then we

must use other oblique rotation methods and PCA not longer applies here.

Thus, we must use principal axes factoring (PFA) or factor analysis in such situations

The common oblique rotation are:

1) Promax 2) Equimax, etc

10)

```
library(dplyr)
```

Removing the last label of the iris datasets which contains flower species.

Scaling is required for Kmeans clustering as it depends on euclidean distances

```
iris_data.1 <- iris_data[, -5] %>% scale
```

Importing these two libraries for forming clusters

```
library(cluster)
```

Checking structure of iris_data

```
str(iris_data.1)
```

a) Fitting k-means clustering model to training datasets with k = 2 and k = 3

set.seed(44)

with k = 2

kmeans.res <- kmeans(iris_data.1, centers = 2, nstart = 20)

kmeans.res

Confusion Matrix

cm <- table(iris_data$Species, kmeans.res$cluster)

cm

Accuracy

(accuracy <- sum(diag(cm))/sum(cm))

We get an accuracy of 66% when we used k = 2 in kmeans clustering

with k = 3

set.seed(44)

kmeans.res <- kmeans(iris_data.1, centers = 3, nstart = 20)

kmeans.res

Confusion Matrix

cm <- table(iris_data$Species, kmeans.res$cluster)

cm

Accuracy

(accuracy <- sum(diag(cm))/sum(cm))

We get an accuracy of 83.3% when we use k = 3

plot(iris_data[c(“Sepal.Length”, “Sepal.Width”)], col = kmeans.res$cluster, main = “K-means with 3 clusters”)

From this graph , we can see that based on Sepal.Length and Sepal.Width, the flower species can be classified.

One flower have sepal.Width greater than 3 and sepal length usually smaller. Due to such characteristics, they seems to

be classified and 3 separate clusters are formed.

Visualizing clusters

y_kmeans <- kmeans.res$cluster library(cluster) clusplot(iris_data[, c("Sepal.Length", "Sepal.Width")], y_kmeans, lines = 0, shade = TRUE, color = TRUE, labels = 2, plotchar = FALSE, span = TRUE, main = paste("Cluster iris"), xlab = 'Sepal.Length', ylab = 'Sepal.Width')

We can clearly see three clusters formation from the given figure. Cluster2 and Cluster3 seems to overlap at some points

but cluster 1 is separated from these two clusters.

#Q.7)

data <- airquality

Let's perform goodness of fit on Temp Variable. Since Temp is a continuous variable and the number of dataset is greater than 100.

So, we can use Kolmogrov spirnov test for goodness of fit.

ks.test(data, data$Temp)

Since the p-value is less than 0.05 we can say that the data is not normally distributed.

Chaning the Month variable as a factor variable

data$Month <- as.factor(data$Month) # Calculating the mean of Wind by Month

If the sample size for each category is less than 100, it’s appropriate to use the Shapiro-Wilk test for normality, especially if you’re interested in testing the normality assumption within each group separately.

```
library(dplyr)

per_month_count <- data %>% group_by(Month) %>% summarize(count= n())

per_month_count
```

Since Temp variable has around 30 to 31 data points per each month so it is appropriate to use Shapiro-Wilk test for normality test.

Shapiro Wilk Test Of Normality

H0: The sample data comes from a normally distributed population.

H1: The sample data does not come from a normally distributed population.

If the p-value obtained from the Shapiro-Wilk test is less than or equal to α (p ≤ 0.05),

you reject the null hypothesis. This suggests that the sample data significantly deviates

from a normal distribution.

If the p-value is greater than α (p > 0.05), you fail to reject the null hypothesis.

This suggests that there is no significant evidence to say that the sample data is not

normally distributed.

Function to perform Shapiro-Wilk test for normality within each group

shapiro_within_month <- function(data) { result <- tapply(data$Temp, data$Month, shapiro.test) return(result) }

Perform Shapiro-Wilk test for each month

shapiro_results <- shapiro_within_month(data)

View the results

```
print(shapiro_results)
```

Since the p-value is greater than 0.05 for every months, then we can say that the data comes from a normal distribution.

Thus, we can say that Temp variable by each month follows a normal distribution.

c) Perform a goodness-of-fit test on the Wind variable by the Month variable to check if the variances of mpg are equal or not on the “am” variable categories

Levene’s test for homogeneity of variances

H0: The variances of wind speed are equal across all categories of the “Month” variable.

H1: The variances of wind speed are not equal across all categories of the “Month” variable.

```
library(car) # Perform Levene’s test levene_result <- leveneTest(Temp ~ Month, data = data)
```

View the test result

```
print(levene_result) ## Since the p-value is 0.03 which is less than 0.05 so we reject the null hypothesis. ## That means the variances of Temp variable are not equal across all categories of the “Month” variable.
```

d) Discuss which one-way ANOVA must be used to compare the “Temp” variable with the “Month” variable categories based on the results obtained above.

Since, temp variable is not normally distributed across each month and the variances of temp are

not equal across all cateogries of the “Month” variable, we can apply Classical One-Way Anova test.

H0: There are no differences in the means of the “temp” variable in different months.

H1: There is a difference in the means of the “temp” variable in different months.

```
summary(aov(Temp ~ Month, data = data))
```

Since the p-value (0.0275) is less than the common significance level of 0.05, we reject the null hypothesis.

This suggests that there is a statistically significant difference in mean temp across different months.

This means, post-hoc test or pairwise comparison is required!

For classical 1-way ANOVA, Tukey HSD is the best post-hoc test!

This will help in identifying which months have significantly different temp speeds from each other.

H0: There is no significant difference in temp between the two months being compared.

H1: There is a significant difference in temp between the two months being compared.

When the p-value is greater than 0.05, we fail to reject the null hypothesis. This means

that we do not have sufficient evidence to conclude that there is a statistically

significant difference in mean temp between the two groups.

When the p-value is less than 0.05, we reject the null hypothesis. This means that we

have sufficient evidence to conclude that there is a statistically significant difference

in mean temp between the two groups.

```
TukeyHSD (aov(Temp ~ Month, data = data))
```

Since, adjusted p value for months (9-6) and (8-7) is greater than 0.05, that means the

there is no significant difference in temp betweeen these two

months.

summary(lm(Temp ~ Month, data = data))

Q.6)

set.seed(22) dataset <- data.frame(age = sample(c(18:99), 250, replace = T), sex = sample(c("male", "female"), 250, replace = T), education_lvl = sample(c("No education", "Primary", "Secondary", "Beyond Secondary"), 250, replace = T), socio_economic_status = sample(c("Low", "Middle", "High"), 250, replace = T), body_mass_index = sample(c(14:38), 250, replace = T))

library(ggplot2)

ggplot(data = dataset, aes(age, body_mass_index)) + geom_point(size = 2, color = "blue") + labs(title = "GGPLOT For age and body mass index") + theme_classic()