Preprocessing Package

2024-12-23

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(ggplot2)
library(rlang)
library(reshape2)
library(moments)
library(pheatmap)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(dplyr)
```

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

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

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

```
library(vcd)
```

```
## Loading required package: grid
```

```
library(grid)

convert_tofactor=function(data) {
  for (col in names(data)) {
    if (!is.numeric(data[[col]])) {
```

```
      data[[col]]=as.factor(data[[col]])
    }
  }
  return(data)
}
univar_breakdown=function(var) {
  if (is.numeric(var)) {     #Numeric
    cat("Numeric Univariate Breakdown:\n")
    sum=summary(var) #datasummary
     name=deparse(substitute(var))
    his=hist(var,main =paste("Histogram of",name),xlab="values",col="blue")
   return(list(summary=sum,histogram=his))

  }
  else { #categorical breakdown
    cat("Categorical Univariate Breakdown:\n")
    result=lapply(names(var),function(name) {
      freq=table(var[[name]])
      plot=barplot(freq,main=paste("Barplot of",name),col="blue",xlab="Values",ylab="Frequency")
    })

  }
  return(list(table=result$freq,plot=result$plot))
}
one_hot_code=function(data,variable) { #must provide list of variables if u want multiple must be in pa
  one_hot_list=lapply(variable,function(var) {
    one_hot=model.matrix(~.-1,data=data[, c(variable,colnames(data)[length(data)])])
    return(one_hot)
  })
  result=cbind(one_hot_list,data)
  result=result[,!duplicated(colnames(result))]
  return(result)
}
clean_input_medi=function(data) {
  data=lapply(data,function(col) {
    if (is.numeric(col) && any(is.na(col))) {
    col[is.na(col)]=median(col,na.rm=TRUE)
    }
    return(col)
  })
return(data)
}
bivar_breakdown=function(x,y,data) { #format for multiple c("a","b")
  var_x=data[x]
  var_y=data[y]
 if(is.numeric(unlist(var_x)) & is.numeric(unlist(var_y))) {
    x_sum=summary(var_x)
    y_sum=summary(var_y)
    plots <- lapply(names(var_x), function(var_x) {
  lapply(names(var_y), function(var_y) {
    # Create scatterplot
    ggplot(data, aes_string(x = var_x, y = var_y)) +
      geom_point(color = "blue", size = 3) +
```

```
      labs(title = paste("Scatterplot of", var_x, "vs", var_y),
           x =var_x,
           y = var_y) +
      theme_minimal()
  })
})
    return(list(summary_of_x=x_sum,summary_of_y=y_sum,plot=plots))
  }
  else if (!is.numeric(unlist(var_x)) & !is.numeric(unlist(var_y))) {
    tab=matrix(NA,nrow = length(var_x),ncol = length(var_y))
    for(i in 1:length(var_x)) {
     for(j in 1:length(var_y)) {
     tab <-table(unlist(var_x[i]), unlist(var_y[j]))

     }
    }
      plots <- lapply(names(var_x), function(var_x) {
  lapply(names(var_y), function(var_y) {
    # Create Grouped barplot
    ggplot(data, aes_string(x = var_x, fill = var_y)) +
      geom_bar(position="stack") +
      labs(title = paste("Barplot of", var_x, "vs", var_y),
           x =var_x, y="Count",
           fill = var_y) +
      theme_minimal()
  })
})

return(list(table=tab,plot=plots))
  }
 else {
   grp=lapply(split(var_y,var_x),summary)
     plots <- lapply(names(var_x), function(var_x) {
  lapply(names(var_y), function(var_y) {
    # Create Grouped barplot
    ggplot(data, aes_string(x = var_x, y = var_y)) +
      geom_bar(stat="identity",fill="blue") +
      labs(title = paste("Barplot of", var_x, "vs", var_y),
           x =var_x, y="Count",
             ) +
      theme_minimal()
  })
})

     return(list(table=grp,plot=plots))
 }
}
distribution_check=function(data) {
    results <- list(
    normal = ks.test(data, "pnorm", mean = mean(data), sd = sd(data)),
    uniform = ks.test(data, "punif", min = min(data), max = max(data)),
    exponential = ks.test(data, "pexp", rate = 1 / mean(data)),
    gamma = ks.test(data, "pgamma", shape = 2, rate = 1 / mean(data)),  # Example shape and rate
```

```
    beta = ks.test((data - min(data)) / (max(data) - min(data)), "pbeta", shape1 = 2, shape2 = 5)
  )

  # Summarize results
  summary_results <- sapply(results, function(test) {
    c(statistic = test$statistic, p_value = test$p.value)
  })
  skew=skewness(data,na.rm = TRUE)
  kurt=kurtosis(data,na.rm=TRUE)
  return(list(results=summary_results,skew=skew,kurt=kurt))
}
multivar_breakdown=function(data) {
  if(is.numeric(unlist(data))==TRUE) {
    #cov matrix
    cov_matrix=cov(data) #unscale
    scale_cov=cov(scale(data))
    #cov plot
    #pca
    pca=prcomp(data)
    pca_summary=summary(pca)
    #pca plots
    explained_variance=pca$sdev^2/sum(pca$sdev^2)
    screeplot=plot(explained_variance,type="b",pch=19,col="red",xlab="Principle Component",ylab="Propor
    biplot=biplot(pca,main="PCA Biplot") #biplot
    #scatterplots
    scatter_grid=ggpairs(data,title = "Scatterplot Matrix",upper = list(continous="cor"),lower=list(con

    return(list(cov_matrix=cov_matrix,scaled_matrix=scale_cov, pca_summary=pca_summary,screeplot=screep
  }
  else if (all(sapply(data,is.factor))==TRUE| all(sapply(data,is.character))==TRUE) {
    cat_vars=names(data)
    freq=data %>% group_by(across(all_of(cat_vars))) %>% summarize(Frequency=n(),.groups = "drop") %>% r
    name=deparse(substitute(data))
    mos=mosaic(table(data),main=paste("Mosaic Plot of", name),shading=TRUE,legend=TRUE)#mosaic
      return(list(table=freq,graph=mos))
  }
  else{
     cat_vars=names(data)[sapply(data,function(x) is.factor(x) | is.character(x))]
    freq=data %>% group_by(across(all_of(cat_vars))) %>% summarize(Count=n(),.groups = "drop")
    return(list(table=freq))
  }
}
```