STAT202 Lab 3: Multiple Linear Regression START

David Ewing

Due on 7 August 2024

#Introduction

This document contains the analysis for Assignment 3. It explores the aquatic\_toxicity dataset using multiple linear regression to understand the relationships between various predictors and the response variable LC50. The analysis will follow these steps:

Step 0: setup loading libraries:

# Introduction

This document contains the analysis for Lab 2. It explores biometric data using multiple linear regression to understand the relationships between various predictors and the response variable weight\_kg. The analysis will follow these steps:”

# Step 0: setup

loading libraries:

set.seed(82171165) #set seed   
  
knitr::opts\_chunk$set(  
 echo = TRUE, # Show all code by default  
 message = TRUE, # Include package messages  
 warning = TRUE # Include warnings if they occur  
)  
  
  
  
library(conflicted)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2

library(readxl)

## Warning: package 'readxl' was built under R version 4.4.2

library(readr)   
library(performance)

## Warning: package 'performance' was built under R version 4.4.2

library(GGally)

## Warning: package 'GGally' was built under R version 4.4.2

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

library(flextable)

## Warning: package 'flextable' was built under R version 4.4.2

library(broom)  
library(skimr)

## Warning: package 'skimr' was built under R version 4.4.2

library(data.table)  
library(lmtest)

## Warning: package 'lmtest' was built under R version 4.4.2

## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:data.table':  
##   
## yearmon, yearqtr  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(leaps)

## Warning: package 'leaps' was built under R version 4.4.2

conflict\_prefer("filter", "dplyr"); conflict\_prefer("select", "dplyr")

## [conflicted] Will prefer dplyr::filter over any other package.  
## [conflicted] Will prefer dplyr::select over any other package.

# Step 1: Read aquatic\_toxicity.xlsx

# Load the data  
toxic <- read\_excel("../data/aquatic\_toxicity.xlsx")

# Step 2: Select a random sample of 500 rows

set.seed(82171165)  
  
toxic\_rows <- nrow(toxic)  
toxic\_sample <- toxic |> slice\_sample(n = 500) # without replacement  
my\_toxic <- toxic\_sample  
  
# Summarise missing values   
# View the first few rows of the sample  
  
skim\_toxic <- skim(my\_toxic) |>  
 select(skim\_variable, n\_missing)  
skim\_toxic

## # A tibble: 9 × 2  
## skim\_variable n\_missing  
## <chr> <int>  
## 1 tpsa 0  
## 2 saacc 0  
## 3 h\_050 0  
## 4 mlogp 0  
## 5 rdchi 0  
## 6 gats1p 0  
## 7 nn 0  
## 8 c\_040 0  
## 9 lc50 0

head(toxic\_sample)

## # A tibble: 6 × 9  
## tpsa saacc h\_050 mlogp rdchi gats1p nn c\_040 lc50  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 0 0 2.85 1.99 0.938 0 0 3.53  
## 2 93.2 118. 1 -0.147 2.36 1.64 2 0 5.9   
## 3 45.8 50.7 0 2.24 2.12 0.942 1 0 4.01  
## 4 64.2 92.5 0 3.61 4.24 1.18 4 2 4.84  
## 5 133. 75.2 0 0.906 3.03 0.941 3 1 7.86  
## 6 35.2 43.9 2 1.25 2.05 1.08 1 0 5.64

# Step 3: Estimate the correlations between all variables

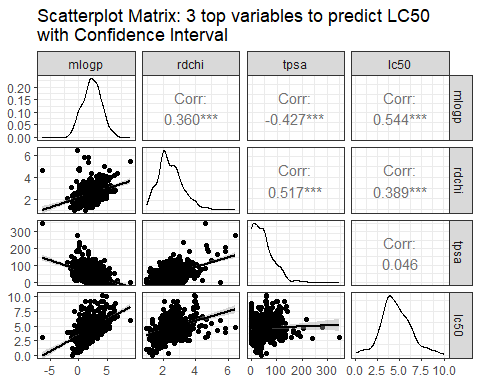
cors <- cor(my\_toxic) # Estimate   
  
cors

## tpsa saacc h\_050 mlogp rdchi gats1p  
## tpsa 1.0000000 0.8469383 0.62198171 -0.4266096 0.51671900 0.17175217  
## saacc 0.8469383 1.0000000 0.74314224 -0.3704348 0.56607663 0.21332189  
## h\_050 0.6219817 0.7431422 1.00000000 -0.4649171 0.26223979 0.05570727  
## mlogp -0.4266096 -0.3704348 -0.46491709 1.0000000 0.35965466 -0.37241844  
## rdchi 0.5167190 0.5660766 0.26223979 0.3596547 1.00000000 0.04810484  
## gats1p 0.1717522 0.2133219 0.05570727 -0.3724184 0.04810484 1.00000000  
## nn 0.5950754 0.4914496 0.45280345 -0.2669030 0.33988338 0.07360678  
## c\_040 0.4447388 0.4919657 0.18932603 -0.1047519 0.41756944 0.14116757  
## lc50 0.0464166 -0.1071164 -0.21235360 0.5435629 0.38927278 -0.30997109  
## nn c\_040 lc50  
## tpsa 0.59507541 0.44473881 0.04641660  
## saacc 0.49144958 0.49196570 -0.10711645  
## h\_050 0.45280345 0.18932603 -0.21235360  
## mlogp -0.26690303 -0.10475193 0.54356289  
## rdchi 0.33988338 0.41756944 0.38927278  
## gats1p 0.07360678 0.14116757 -0.30997109  
## nn 1.00000000 0.31542078 -0.07261928  
## c\_040 0.31542078 1.00000000 0.01760953  
## lc50 -0.07261928 0.01760953 1.00000000

# Step 4: Top variables to predict LC50

# 3 variables with the highest correlation with LC50  
lc50\_sort <- sort(cors["lc50", ], decreasing = TRUE)  
lc50\_sort <- lc50\_sort[lc50\_sort != 1]  
top\_3 <- names(lc50\_sort[1:3])  
  
lc50\_matrix <- my\_toxic[, c(top\_3, "lc50")] |>  
 ggpairs(  
 lower = list(continuous = wrap("smooth", method = "lm", se = TRUE)),  
 title = "Scatterplot Matrix: 3 top variables to predict LC50\nwith Confidence Interval"  
 ) +  
 theme\_bw()

lc50\_matrix



A positive correlation between **mlogp** and **LC50**; with weaker positive relationships for **rdchi** and **LC50**; and an insiginficatn association between **tpsa** and **LC50**. Each relationship appear generally linear but variability, particularly in **rdchi**, could affect predictability. THere has been no attempt to remove outliers.

# Step 5: multiple linear regression model to predict LC50

lc50\_arg <- as.formula(paste("lc50 ~ ", paste(top\_3, collapse = " + ")))  
m1 <- lm(lc50\_arg, data = my\_toxic)  
summary(m1)

##   
## Call:  
## lm(formula = lc50\_arg, data = my\_toxic)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.8829 -0.8113 -0.1803 0.6748 5.0817   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.680400 0.190861 14.044 < 2e-16 \*\*\*  
## mlogp 0.741600 0.056631 13.095 < 2e-16 \*\*\*  
## rdchi -0.217494 0.125996 -1.726 0.0849 .   
## tpsa 0.015881 0.002363 6.720 4.99e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.29 on 496 degrees of freedom  
## Multiple R-squared: 0.3938, Adjusted R-squared: 0.3901   
## F-statistic: 107.4 on 3 and 496 DF, p-value: < 2.2e-16

tidy(m1)

## # A tibble: 4 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 2.68 0.191 14.0 5.85e-38  
## 2 mlogp 0.742 0.0566 13.1 7.31e-34  
## 3 rdchi -0.217 0.126 -1.73 8.49e- 2  
## 4 tpsa 0.0159 0.00236 6.72 4.99e-11

# Step 6: Fit All Regression Subsets using adjusted R-squared

all\_models <- regsubsets(lc50 ~ ., data = my\_toxic)  
plot(all\_models, scale = "adjr2",   
 main = expression("Regression Subsets (Adjusted " ~ R^2 ~ ")"))

