Catherine Nderitu

Portfolio Goal Setting:

For my portfolio, I chose to create an interactive applet that explains a supervised statistical learning method using real-world data. The applet focuses on linear regression, one of the foundational concepts of statistical modeling. It allows users to upload their own dataset or use a pre-loaded dataset to visualize how linear regression works. Users can interactively adjust the model parameters, see the corresponding changes in the fitted line, and explore the statistical measures like R-squared and residual plots. This applet was developed using the R programming language and the Shiny package.

Course Objectives

Objective 1 : Describe probability as a foundation of statistical modeling, including inference and maximum likelihood estimation

The provided example demonstrates the application of a linear regression model to predict car miles per gallon (mpg) based on predictor variables such as car weight (wt), horsepower (hp), and quarter-mile time (qsec) using the ‘mtcars’ dataset. The model is fitted using maximum likelihood estimation (MLE), which leverages probability to quantify uncertainty and estimate model parameters that best explain the observed data. Probability serves as the foundation for statistical modeling, enabling us to understand uncertainty and make predictions based on data.

```
# Load necessary libraries
library(stats)
library(GGally)
```

```
## Loading required package: ggplot2
```

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

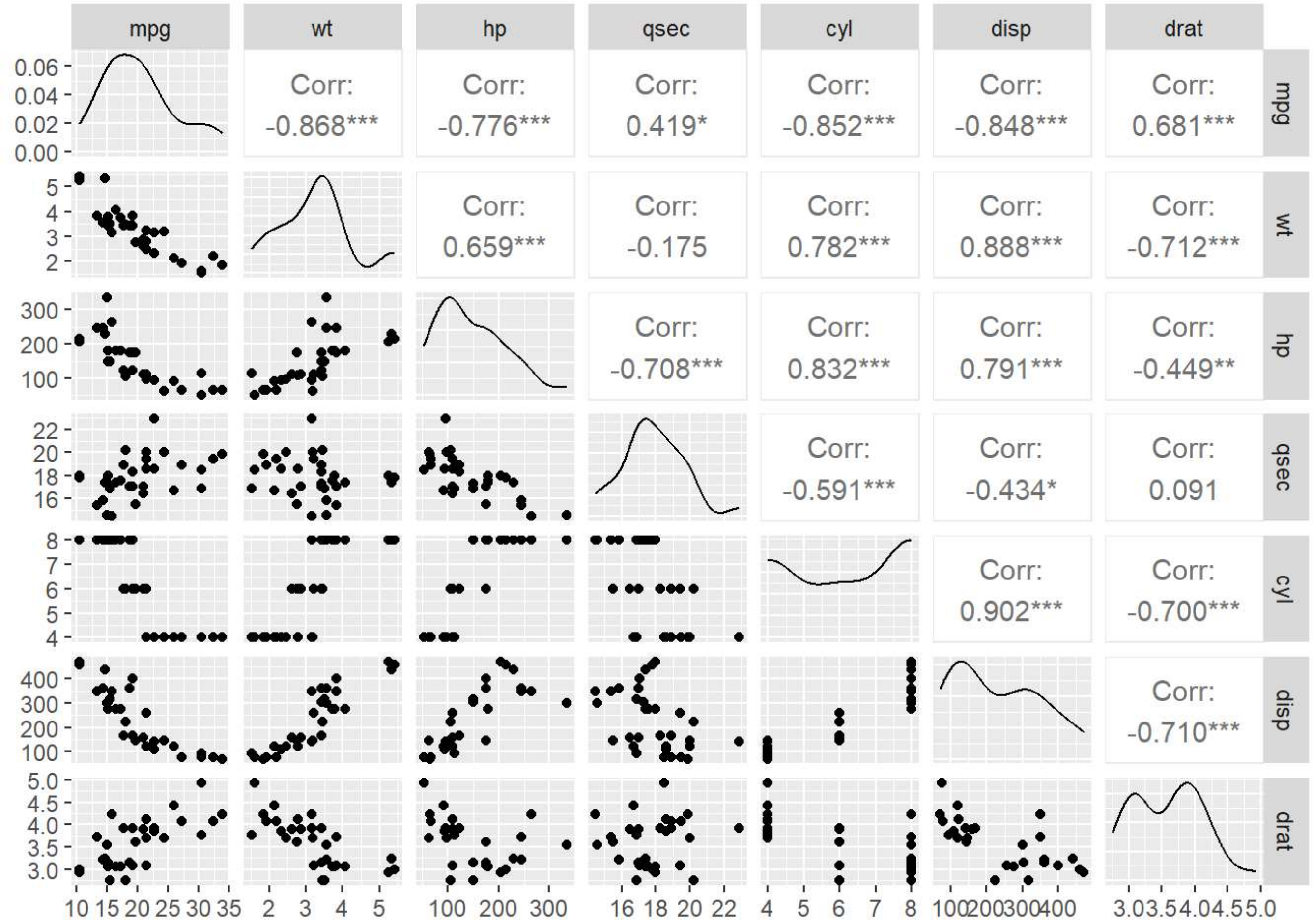
```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.2     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ lubridate 1.9.2     ✔ tibble    3.2.1
## ✔ purrr     1.0.1     ✔ tidyr     1.3.0
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```r
library(tidymodels)
```

```
## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.0 ──
## ✔ broom        1.0.5     ✔ rsample      1.1.1
## ✔ dials        1.2.0     ✔ tune         1.1.1
## ✔ infer        1.0.4     ✔ workflows    1.1.3
## ✔ modeldata    1.1.0     ✔ workflowsets 1.0.1
## ✔ parsnip      1.1.0     ✔ yardstick    1.2.0
## ✔ recipes      1.0.6
## ── Conflicts ───────────────────────────────────────── tidymodels_conflicts() ──
## ✖ scales::discard() masks purrr::discard()
## ✖ dplyr::filter()   masks stats::filter()
## ✖ recipes::fixed()  masks stringr::fixed()
## ✖ dplyr::lag()      masks stats::lag()
## ✖ yardstick::spec() masks readr::spec()
## ✖ recipes::step()   masks stats::step()
## • Search for functions across packages at https://www.tidymodels.org/find/
```

```r
#plots to check the relationship between response and predictor variables
mtcars %>%
  select(mpg,wt,hp,qsec,cyl,disp,drat) %>%
  ggpairs()
```



In this example the important predictor variable for mpg is wt with a pvalue <0.05.As wt increases the mpg decreases.One unit increase in wt leads to 4.38 decrease in mpg. The model becomes mpg = 16.53 - 4.38 (wt). The model estimates are the maximum likelihood estimators of the multiple linear regression coefficients

```
# Created a parsnip specification for a linear model
lm_spec <- linear_reg() %>%
set_mode("regression") %>%
set_engine("lm")

lm_spec
```

```
## Linear Regression Model Specification (regression)
##
## Computational engine: lm
```

```
# Fit our specification to our data
#To check which predictor variables are associated with the response
#We use the linearly related predictor variables
mlr_mod <- lm_spec %>%
fit(mpg ~ wt+hp+qsec+disp+drat, data = mtcars)
# View our model output
tidy(mlr_mod)
```

```
## # A tibble: 6 × 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept) 16.5       11.0        1.51  0.144
## 2 wt          -4.39       1.24      -3.53  0.00158
## 3 hp          -0.0206     0.0153    -1.35  0.189
## 4 qsec         0.640      0.459      1.39  0.175
## 5 disp         0.00872    0.0112     0.779 0.443
## 6 drat         2.02       1.31       1.54  0.136
```

To assess our model fit, we can use R^2 (the coefficient of determination), the proportion of variability in the response variable that is explained by the explanatory variable.With R-squared of 0.84 it means that, approximately 84% of the variability in the dependent variable mpg can be explained by the independent variable wt

```
glance(mlr_mod)
```

```
## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
##       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1     0.849         0.820  2.56      29.2 6.89e-10     5  -72.1  158.  169.
## # ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Objective 2 : Determine and apply the appropriate generalized linear model for a specific data context

Logistic Regresion GLM

In this example I used the “Default” dataset to perform a logistic regression analysis. The objective was to predict the likelihood of default (binary outcome) based on the predictor variables: “student,” “balance,” and “income.” I first loaded the necessary libraries and displayed a summary of the dataset.

I then fitted a multiple logistic regression model using the “glm” function from the “tidyverse” package. I created two plots to assess the model’s fit. The first plot shows the deviance residuals against the fitted values, helping to visualize the distribution of residuals and identify any patterns or trends. The second plot displays the deviance residuals against the row numbers,in order to detect potential outliers or influential observations.

Overall, this example provided a comprehensive analysis of the logistic regression model a member of generalized linear models, from fitting the model to evaluating its fit using residual plots. It helped understand how the predictor variables contribute to the likelihood of default in the “Default” dataset.

```
# Load necessary libraries
library(ISLR2)       # For accessing the "Default" dataset
library(tidyverse)   # For data manipulation and visualization
library(tidymodels)  # For building generalized linear models

# Load the "Default" dataset
data <- ISLR2::Default

# View summary of the dataset
summary(data)
```

```
##  default    student       balance           income     
##  No :9667   No :7056   Min.   :   0.0   Min.   :  772  
##  Yes: 333   Yes:2944   1st Qu.: 481.7   1st Qu.:21340  
##                        Median : 823.6   Median :34553  
##                        Mean   : 835.4   Mean   :33517  
##                        3rd Qu.:1166.3   3rd Qu.:43808  
##                        Max.   :2654.3   Max.   :73554
```

```
# Find the total number of observations in the dataset
nrow(data)
```

```
## [1] 10000
```

```
# Fit the multiple logistic regression model
mult_log_mod <- glm(default ~ student + balance + income, data = data, family = "binomial")

# Display the coefficients and their statistics
tidy(mult_log_mod)
```

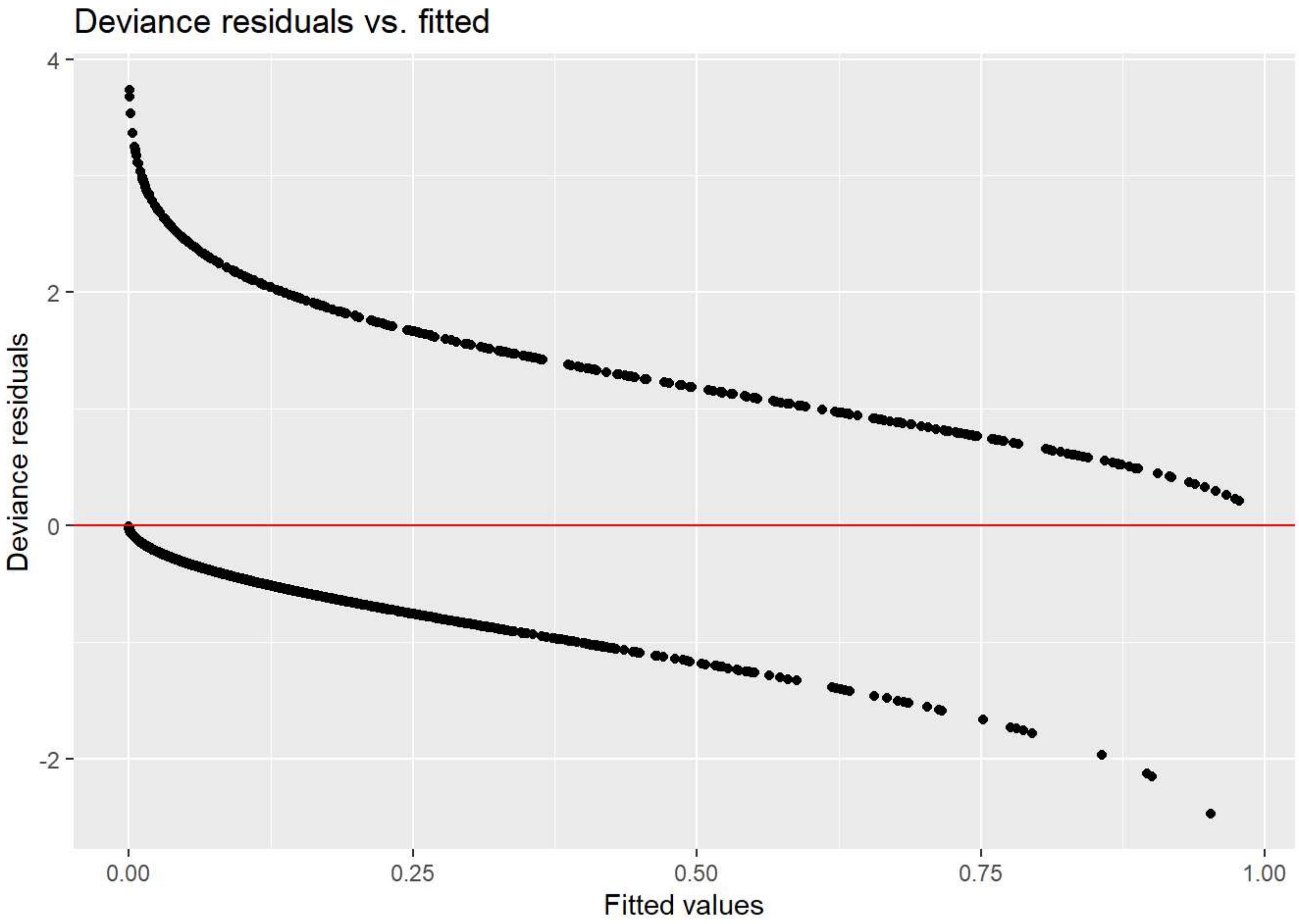
```
## # A tibble: 4 × 5
##   term            estimate  std.error statistic   p.value
##   <chr>              <dbl>      <dbl>     <dbl>     <dbl>
## 1 (Intercept) -10.9        0.492        -22.1   4.91e-108
## 2 studentYes   -0.647      0.236         -2.74  6.19e-  3
## 3 balance       0.00574    0.000232      24.7   4.22e-135
## 4 income        0.00000303 0.00000820     0.370 7.12e-  1
```

```
# Display the exponentiated coefficients for interpretation
tidy(mult_log_mod, exponentiate = TRUE) %>%
  knitr::kable(digits = 3)
```

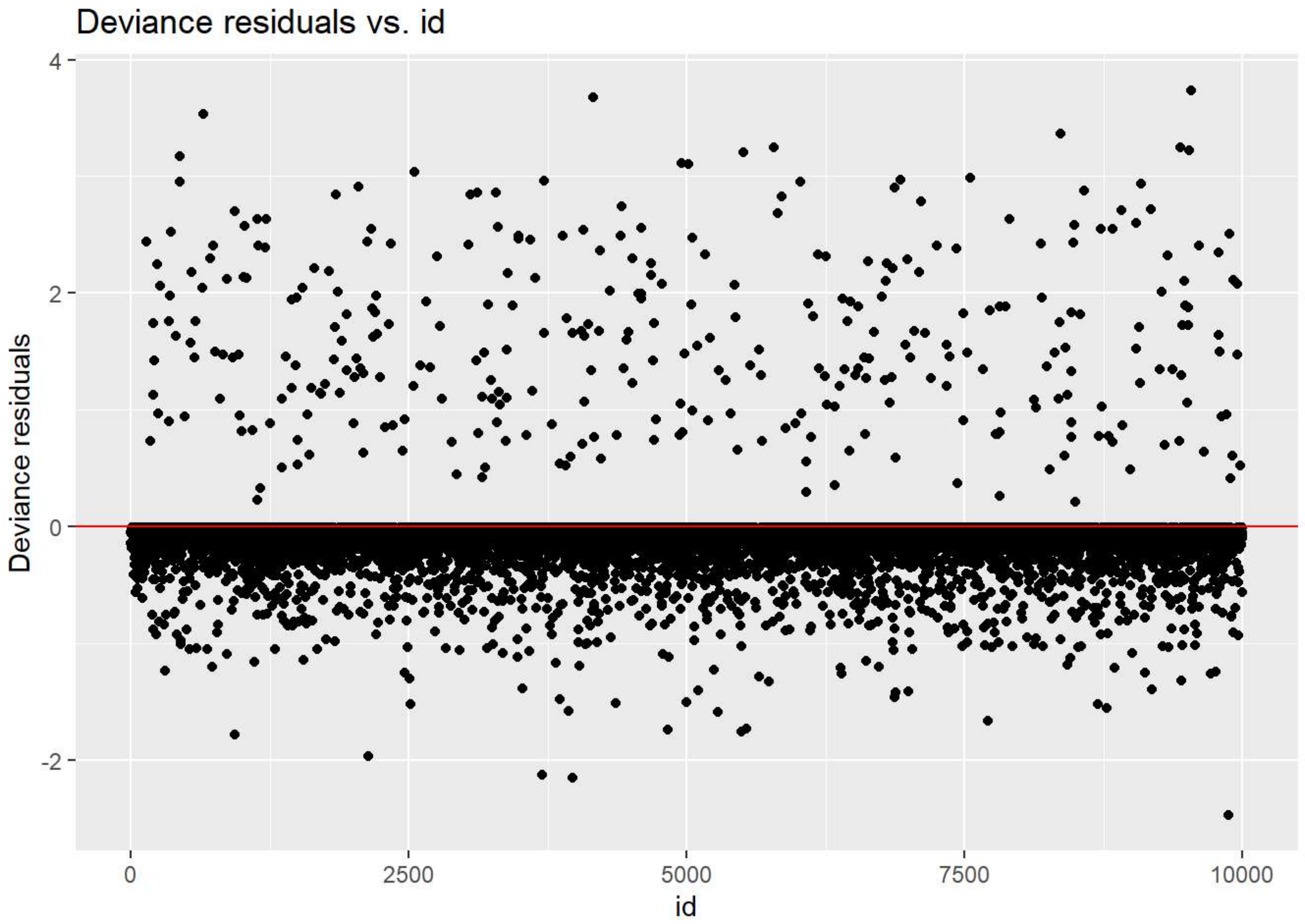
term	estimate	std.error	statistic	p.value
(Intercept)	0.000	0.492	-22.080	0.000
studentYes	0.524	0.236	-2.738	0.006

term	estimate	std.error	statistic	p.value
balance	1.006	0.000	24.738	0.000
income	1.000	0.000	0.370	0.712

```r
# To store residuals and create a row number variable
mult_log_aug <- augment(mult_log_mod, type.predict = "response",
                        type.residuals = "deviance") %>%
                        mutate(id = row_number())

# Assess model fit

# Plot residuals vs fitted values
ggplot(data = mult_log_aug, aes(x = .fitted, y = .resid)) +
geom_point() +
geom_hline(yintercept = 0, color = "red") +
labs(x = "Fitted values",
     y = "Deviance residuals",
     title = "Deviance residuals vs. fitted")
```



```
# Plot residuals vs row number
ggplot(data = mult_log_aug, aes(x = id, y = .resid)) +
geom_point() +
geom_hline(yintercept = 0, color = "red") +
labs(x = "id",
     y = "Deviance residuals",
     title = "Deviance residuals vs. id")
```



Objective3 : Conduct model selection for a set of candidate models

Subset selection method

In the example below I performed best subsets regression on the “mtcars” dataset to find the best model with the optimal number of predictor variables.I explored different metrics, such as R-squared, Adjusted R-squared, Cp, and BIC, to evaluate the performance of models with varying numbers of predictors.I code generated plots to visualize the relationships between these metrics and the number of variables in the model. Ultimately,I identified the model with the highest Adjusted R-squared and the model with the minimum Cp and BIC values. Additionally, I retrieved the coefficients of the model with 6 variables, providing insights into the significant predictors for that model.

```
# Load required libraries
library(ISLR2)
library(leaps)

# Check column names, dimensions, and missing values in mtcars dataset
names(mtcars)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

```
dim(mtcars)
```

```
## [1] 32 11
```

```
sum(is.na(mtcars$mpg))
```

```
## [1] 0
```

```
# Perform best subsets regression with up to 19 variables
regfit.full <- regsubsets(mpg ~ ., data = mtcars, nvmax = 19)

# Summarize the results of the best subsets regression
reg.summary <- summary(regfit.full)

# View the R-squared values for different models with varying number of variables
reg.summary$rsq
```

```
##  [1] 0.7528328 0.8302274 0.8496636 0.8578510 0.8637377 0.8667078 0.8680976
##  [8] 0.8687064 0.8689448 0.8690158
```

```
# Create a 2x2 plot to visualize the relationship between RSS and number of variables,
# and between Adjusted R-squared and number of variables
par(mfrow = c(2, 2))
plot(reg.summary$rss, xlab = "Number of Variables", ylab = "RSS", type = "l")
plot(reg.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")

# Identify the index of the model with the highest Adjusted R-squared value and plot a red point
# to mark the maximum value
which.max(reg.summary$adjr2)
```

```
## [1] 5
```

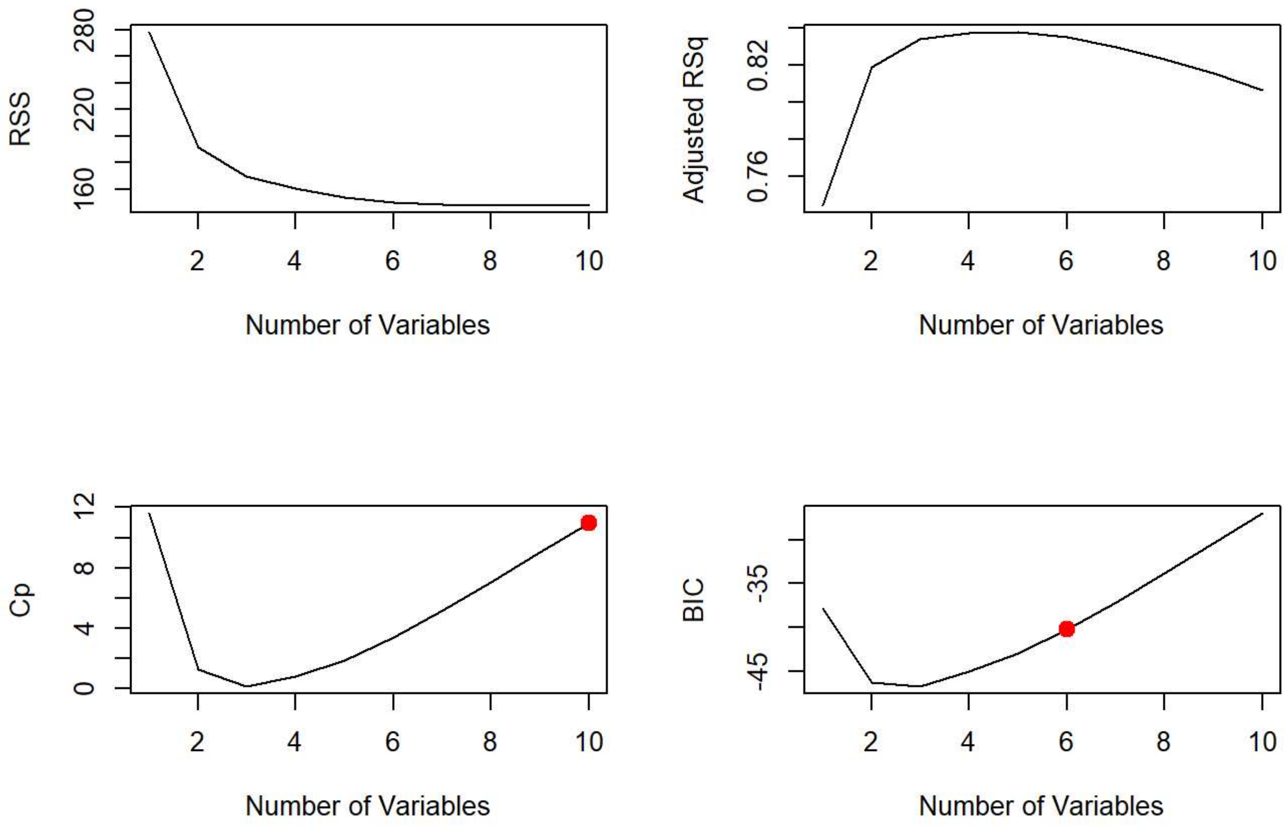
```
points(11, reg.summary$adjr2[11], col = "red", cex = 2, pch = 20)

# Create a plot to visualize the relationship between Cp and number of variables,
# and identify the index of the model with the minimum Cp value and mark it with a red point
plot(reg.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
which.min(reg.summary$cp)
```

```
## [1] 3
```

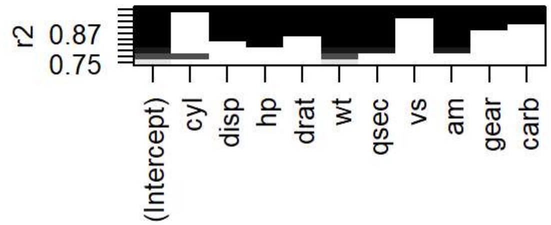
```
points(10, reg.summary$cp[10], col = "red", cex = 2, pch = 20)

# Identify the index of the model with the minimum BIC value and plot a red point
# to mark the minimum value
which.min(reg.summary$bic)
```

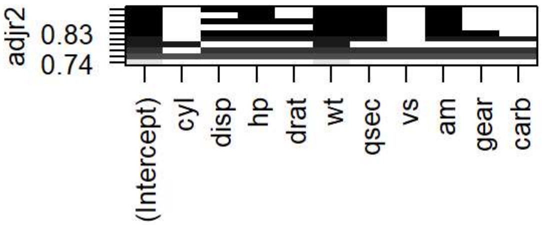
```
## [1] 3
```

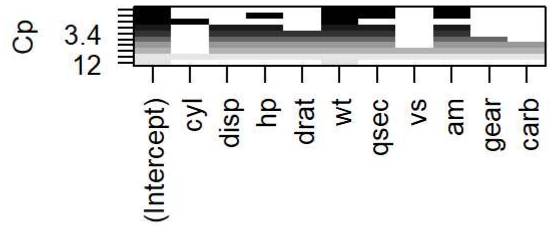
```
plot(reg.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")
points(6, reg.summary$bic[6], col = "red", cex = 2, pch = 20)
```

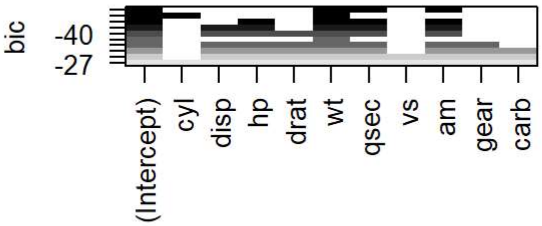


```
# Create plots to visualize the relationship between R-squared, Adjusted R-squared, Cp, and BIC
# for all models in the best subsets regression
plot(regfit.full, scale = "r2")
plot(regfit.full, scale = "adjr2")
plot(regfit.full, scale = "Cp")
plot(regfit.full, scale = "bic")
```





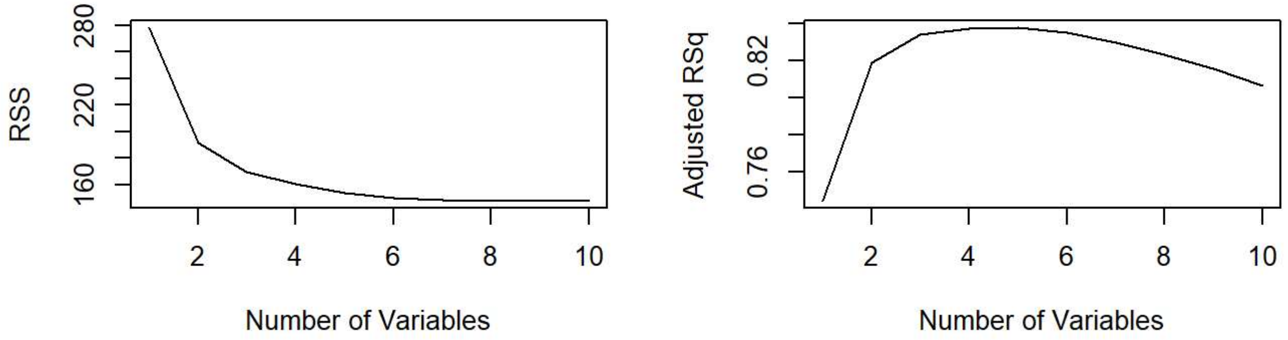




```
# View the coefficients of the model with 6 variables
coef(regfit.full, 6)
```

```
## (Intercept)        disp          hp        drat          wt        qsec 
## 10.71061639  0.01310313 -0.02179818  1.02065283 -4.04454214  0.99072948 
##          am 
##  2.98468801
```

```
 par (mfrow = c(2, 2))
plot (reg.summary$rss , xlab = " Number of Variables ",
ylab = " RSS ", type = "l")
plot (reg.summary$adjr2 , xlab = " Number of Variables ",
ylab = " Adjusted RSq ", type = "l")
```



Forward and backward selection

In this example I performed forward stepwise regression and backward stepwise regression using best subsets on the “mtcars” dataset with the response variable “mpg” and predictor variables from the dataset.I used regsubsets function from the “leaps” package to perform the best subsets regression.

After fitting the forward and backward stepwise regression models, I used the summary function to view the results, which include information about the selected variables and their coefficients for each model with different numbers of variables (up to 19).

I then retrieved the coefficients of the model with 7 variables from the full model, the forward stepwise model, and the backward stepwise model, respectively.

In summary, conducted model selection by exploring different combinations of predictor variables and provided insights into the best models with a specific number of variables, helping to identify the most relevant predictors for explaining the variation in the “mpg” response variable.

```
# Perform forward stepwise regression using best subsets
regfit.fwd <- regsubsets(mpg ~ ., data = mtcars, nvmax = 19, method = "forward")

# View summary of forward stepwise regression results
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(mpg ~ ., data = mtcars, nvmax = 19, method = "forward")
## 10 Variables  (and intercept)
##      Forced in Forced out
## cyl      FALSE      FALSE
## disp     FALSE      FALSE
## hp       FALSE      FALSE
## drat     FALSE      FALSE
## wt       FALSE      FALSE
## qsec     FALSE      FALSE
## vs       FALSE      FALSE
## am       FALSE      FALSE
## gear     FALSE      FALSE
## carb     FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: forward
##           cyl disp hp  drat wt  qsec vs  am  gear carb
## 1  ( 1 )  " " " "  " " " "  "*" " "  " " " " " "  " " 
## 2  ( 1 )  "*" " "  " " " "  "*" " "  " " " " " "  " " 
## 3  ( 1 )  "*" " "  "*" " "  "*" " "  " " " " " "  " " 
## 4  ( 1 )  "*" " "  "*" " "  "*" " "  " " "*" " "  " " 
## 5  ( 1 )  "*" " "  "*" " "  "*" "*"  " " "*" " "  " " 
## 6  ( 1 )  "*" "*"  "*" " "  "*" "*"  " " "*" " "  " " 
## 7  ( 1 )  "*" "*"  "*" "*"  "*" "*"  " " "*" " "  " " 
## 8  ( 1 )  "*" "*"  "*" "*"  "*" "*"  " " "*" "*"  " " 
## 9  ( 1 )  "*" "*"  "*" "*"  "*" "*"  " " "*" "*"  "*" 
## 10  ( 1 ) "*" "*"  "*" "*"  "*" "*"  "*" "*" "*"  "*"
```

```
# Perform backward stepwise regression using best subsets
regfit.bwd <- regsubsets(mpg ~ ., data = mtcars, nvmax = 19, method = "backward")

# View summary of backward stepwise regression results
summary(regfit.bwd)
```

```
## Subset selection object
## Call: regsubsets.formula(mpg ~ ., data = mtcars, nvmax = 19, method = "backward")
## 10 Variables  (and intercept)
##      Forced in Forced out
## cyl      FALSE      FALSE
## disp     FALSE      FALSE
## hp       FALSE      FALSE
## drat     FALSE      FALSE
## wt       FALSE      FALSE
## qsec     FALSE      FALSE
## vs       FALSE      FALSE
## am       FALSE      FALSE
## gear     FALSE      FALSE
## carb     FALSE      FALSE
## 1 subsets of each size up to 10
## Selection Algorithm: backward
##           cyl disp hp  drat wt  qsec vs  am  gear carb
## 1  ( 1 )  " " " "  " " " "  "*" " "  " " " " " "  " " 
## 2  ( 1 )  " " " "  " " " "  "*" "*"  " " " " " "  " " 
## 3  ( 1 )  " " " "  " " " "  "*" "*"  " " "*" " "  " " 
## 4  ( 1 )  " " " "  "*" " "  "*" "*"  " " "*" " "  " " 
## 5  ( 1 )  " " "*"  "*" " "  "*" "*"  " " "*" " "  " " 
## 6  ( 1 )  " " "*"  "*" "*"  "*" "*"  " " "*" " "  " " 
## 7  ( 1 )  " " "*"  "*" "*"  "*" "*"  " " "*" "*"  " " 
## 8  ( 1 )  " " "*"  "*" "*"  "*" "*"  " " "*" "*"  "*" 
## 9  ( 1 )  " " "*"  "*" "*"  "*" "*"  "*" "*" "*"  "*" 
## 10  ( 1 ) "*" "*"  "*" "*"  "*" "*"  "*" "*" "*"  "*"
```

```
# Retrieve coefficients of the model with 7 variables from the full model
coef(regfit.full, 7)
```

```
## (Intercept)        disp          hp        drat          wt        qsec 
##  9.19762837  0.01551976 -0.02470716  0.81022794 -4.13065054  1.00978651 
##          am        gear 
##  2.58979984  0.60644020
```

```
# Retrieve coefficients of the model with 7 variables from the forward stepwise model
coef(regfit.fwd, 7)
```

```
## (Intercept)         cyl        disp          hp        drat          wt 
## 15.30918956 -0.34192099  0.01458808 -0.02057733  0.81836607 -3.99345102 
##        qsec          am 
##  0.85996253  2.72022025
```

```
# Retrieve coefficients of the model with 7 variables from the backward stepwise model
coef(regfit.bwd, 7)
```

```
## (Intercept)        disp          hp        drat          wt        qsec 
##  9.19762837  0.01551976 -0.02470716  0.81022794 -4.13065054  1.00978651 
##          am        gear 
##  2.58979984  0.60644020
```

Objective 4 : Communicate the results of statistical models to a general audience

The applet is designed to be user-friendly and accessible to a general audience. It provides clear visualizations, easy-to-understand sliders for model parameters, and interactive components that engage users in the learning process. By being intuitive and informative, the applet effectively communicates the concepts of linear regression and its results to a broader audience.

Objective 5 : Use programming software (i.e., R) to fit and assess statistical models

This applet is entirely built using R and the Shiny package. It showcases my proficiency in using R for fitting and assessing statistical models. The applet employs various R packages for data manipulation, model fitting, and visualization. Additionally, the applet demonstrates my ability to present statistical concepts in an interactive and engaging way using R programming.

Reflection and Learning growth

Reflection on Learning and Growth: Throughout the semester, I have gained a deeper understanding of probability and its essential role in statistical modeling. I initially struggled with grasping the concept of maximum likelihood estimation, but through practice, in-class activities, and discussions with peers, I now feel much more confident in this area. Additionally, I have developed a solid grasp of generalized linear models and their applications in different data contexts.

I found model selection to be challenging, as it required balancing between simplicity and accuracy in the models. However, participating in class activities and mini-competitions helped me develop a systematic approach to model selection and identify the trade-offs involved.

Regarding communication, I have made an effort to improve my ability to present complex statistical concepts to a general audience effectively. The interactive applet I created for this portfolio is a testament to that growth. I learned how to simplify complex ideas and create engaging visualizations to facilitate understanding.

As for using programming software, I came into the course with some experience in R, but I am delighted with how much I’ve learned and utilized R to fit, assess, and visualize various statistical models.

Reflection on Active Participation in the Course Community: Throughout the semester, I actively participated in the course community by regularly attending class, actively engaging in team discussions, and contributing to online forums. I enjoyed collaborating with my peers on mini-competitions and in-class activities, as it provided an opportunity to learn from different perspectives and approaches.

To foster a supportive community, I made an effort to be responsive to my peers’ questions and encouraged collaboration within our teams. I also shared resources and helpful materials whenever I came across them, which contributed to a positive learning environment.

Overall, this course has been a journey of growth and self-discovery. I am grateful for the supportive community and the opportunity to apply my knowledge to real-world data through various projects and competitions. I feel much more confident in my statistical modeling abilities and look forward to applying these skills in future endeavors.