STATISTICAL MODELLING ASS 2

NEEMA NDANU

2024-07-18

## QUESTION 1 [show your R codes, R output, description of the process and Interpretation both in HTML and Word document] (10 Marks)

### (a) Using appropriate statistical data data, compare a log model, poison model , negative binomial and Quasi-Poisson Do they differ significantly explanatory variables therein considering potential over-dispersion and the categorical nature of of the response variable?(10 marks)

# Load the necessary libraries  
library(readxl)

## Warning: package 'readxl' was built under R version 4.3.3

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

library(MASS) # For Negative Binomial model  
library(pscl) # For various model diagnostics

## Warning: package 'pscl' was built under R version 4.3.3

## Classes and Methods for R originally developed in the  
## Political Science Computational Laboratory  
## Department of Political Science  
## Stanford University (2002-2015),  
## by and under the direction of Simon Jackman.  
## hurdle and zeroinfl functions by Achim Zeileis.

library(dplyr) # For data manipulation

## Warning: package 'dplyr' was built under R version 4.3.2

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':  
##   
## select

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

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

library(sandwich) # For robust standard errors

## Warning: package 'sandwich' was built under R version 4.3.3

library(lmtest) # For coeftest function

## Warning: package 'lmtest' was built under R version 4.3.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.3.2

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

library(car) # For VIF function

## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

# Load the data  
data <- read.csv("C:\\Users\\HP\\Desktop\\R- CODES\\WineQT.csv")  
  
# Inspect the data structure and summary  
str(data)

## 'data.frame': 1143 obs. of 13 variables:  
## $ fixed.acidity : num 7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 6.7 ...  
## $ volatile.acidity : num 0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.58 ...  
## $ citric.acid : num 0 0 0.04 0.56 0 0 0.06 0 0.02 0.08 ...  
## $ residual.sugar : num 1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 1.8 ...  
## $ chlorides : num 0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.097 ...  
## $ free.sulfur.dioxide : num 11 25 15 17 11 13 15 15 9 15 ...  
## $ total.sulfur.dioxide: num 34 67 54 60 34 40 59 21 18 65 ...  
## $ density : num 0.998 0.997 0.997 0.998 0.998 ...  
## $ pH : num 3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.28 ...  
## $ sulphates : num 0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.54 ...  
## $ alcohol : num 9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 9.2 ...  
## $ quality : int 5 5 5 6 5 5 5 7 7 5 ...  
## $ Id : int 0 1 2 3 4 5 6 7 8 10 ...

summary(data)

## fixed.acidity volatile.acidity citric.acid residual.sugar   
## Min. : 4.600 Min. :0.1200 Min. :0.0000 Min. : 0.900   
## 1st Qu.: 7.100 1st Qu.:0.3925 1st Qu.:0.0900 1st Qu.: 1.900   
## Median : 7.900 Median :0.5200 Median :0.2500 Median : 2.200   
## Mean : 8.311 Mean :0.5313 Mean :0.2684 Mean : 2.532   
## 3rd Qu.: 9.100 3rd Qu.:0.6400 3rd Qu.:0.4200 3rd Qu.: 2.600   
## Max. :15.900 Max. :1.5800 Max. :1.0000 Max. :15.500   
## chlorides free.sulfur.dioxide total.sulfur.dioxide density   
## Min. :0.01200 Min. : 1.00 Min. : 6.00 Min. :0.9901   
## 1st Qu.:0.07000 1st Qu.: 7.00 1st Qu.: 21.00 1st Qu.:0.9956   
## Median :0.07900 Median :13.00 Median : 37.00 Median :0.9967   
## Mean :0.08693 Mean :15.62 Mean : 45.91 Mean :0.9967   
## 3rd Qu.:0.09000 3rd Qu.:21.00 3rd Qu.: 61.00 3rd Qu.:0.9978   
## Max. :0.61100 Max. :68.00 Max. :289.00 Max. :1.0037   
## pH sulphates alcohol quality   
## Min. :2.740 Min. :0.3300 Min. : 8.40 Min. :3.000   
## 1st Qu.:3.205 1st Qu.:0.5500 1st Qu.: 9.50 1st Qu.:5.000   
## Median :3.310 Median :0.6200 Median :10.20 Median :6.000   
## Mean :3.311 Mean :0.6577 Mean :10.44 Mean :5.657   
## 3rd Qu.:3.400 3rd Qu.:0.7300 3rd Qu.:11.10 3rd Qu.:6.000   
## Max. :4.010 Max. :2.0000 Max. :14.90 Max. :8.000   
## Id   
## Min. : 0   
## 1st Qu.: 411   
## Median : 794   
## Mean : 805   
## 3rd Qu.:1210   
## Max. :1597

# Ensure quality is treated as a factor  
data$quality <- as.factor(data$quality)  
  
# Check for missing values  
missing\_values <- sapply(data, function(x) sum(is.na(x)))  
print(missing\_values)

## fixed.acidity volatile.acidity citric.acid   
## 0 0 0   
## residual.sugar chlorides free.sulfur.dioxide   
## 0 0 0   
## total.sulfur.dioxide density pH   
## 0 0 0   
## sulphates alcohol quality   
## 0 0 0   
## Id   
## 0

# Impute missing values  
# Impute numeric columns with the mean  
data <- data %>%   
 mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))  
  
# Impute categorical columns with the mode (most frequent value)  
impute\_mode <- function(x) {  
 if(is.factor(x) || is.character(x)) {  
 # Get the most frequent value  
 return(ifelse(is.na(x), as.character(names(sort(table(x), decreasing = TRUE))[1]), x))  
 }  
 return(x)  
}  
  
data <- data %>%   
 mutate(across(where(~ is.factor(.) || is.character(.)), ~ impute\_mode(.)))  
  
# Remove rows with any remaining missing values (if necessary)  
data\_clean <- na.omit(data)  
  
# Check if there are still any missing values  
missing\_values\_clean <- sapply(data\_clean, function(x) sum(is.na(x)))  
print(missing\_values\_clean)

## fixed.acidity volatile.acidity citric.acid   
## 0 0 0   
## residual.sugar chlorides free.sulfur.dioxide   
## 0 0 0   
## total.sulfur.dioxide density pH   
## 0 0 0   
## sulphates alcohol quality   
## 0 0 0   
## Id   
## 0

### EXPLANATION

The code reads the dataset, checks and imputes missing values, and ensures that the data is clean and ready for analysis Most of the variables exhibit a right-skewed distribution that is because their mean being either slightly higher or significantly higher than the median. Variables such as pH, quality, and density have distributions that are closer to normal. For these variables, the mean is either slightly lower than or around the median, indicating a more symmetric distribution.

# Load necessary libraries  
library(corrplot) # For correlation plots

## Warning: package 'corrplot' was built under R version 4.3.3

## corrplot 0.92 loaded

# Convert quality to numeric if not already done  
data\_clean$quality\_numeric <- as.numeric(data\_clean$quality)  
  
# Select numeric columns excluding quality\_numeric  
numeric\_columns <- sapply(data\_clean, is.numeric)  
numeric\_columns["quality\_numeric"] <- FALSE  
  
numeric\_data <- data\_clean[, numeric\_columns]  
  
# Compute correlation matrix  
cor\_matrix <- cor(numeric\_data)  
cor\_matrix

## fixed.acidity volatile.acidity citric.acid residual.sugar  
## fixed.acidity 1.00000000 -0.250728322 0.67315725 0.171830535  
## volatile.acidity -0.25072832 1.000000000 -0.54418694 -0.005751097  
## citric.acid 0.67315725 -0.544186937 1.00000000 0.175814854  
## residual.sugar 0.17183054 -0.005751097 0.17581485 1.000000000  
## chlorides 0.10788857 0.056336259 0.24531249 0.070863112  
## free.sulfur.dioxide -0.16483079 -0.001962479 -0.05758910 0.165338797  
## total.sulfur.dioxide -0.11062837 0.077747722 0.03687111 0.190790035  
## density 0.68150088 0.016511520 0.37524326 0.380146952  
## pH -0.68516260 0.221491518 -0.54633914 -0.116958936  
## sulphates 0.17459183 -0.276078597 0.33123176 0.017474504  
## alcohol -0.07505485 -0.203909273 0.10625034 0.058420606  
## quality 0.12197010 -0.407393513 0.24082084 0.022001931  
## Id -0.27582627 -0.007891570 -0.13901077 -0.046344405  
## chlorides free.sulfur.dioxide total.sulfur.dioxide  
## fixed.acidity 0.10788857 -0.164830793 -0.11062837  
## volatile.acidity 0.05633626 -0.001962479 0.07774772  
## citric.acid 0.24531249 -0.057589104 0.03687111  
## residual.sugar 0.07086311 0.165338797 0.19079003  
## chlorides 1.00000000 0.015280458 0.04816316  
## free.sulfur.dioxide 0.01528046 1.000000000 0.66109287  
## total.sulfur.dioxide 0.04816316 0.661092872 1.00000000  
## density 0.20890071 -0.054150318 0.05017483  
## pH -0.27775907 0.072803706 -0.05912572  
## sulphates 0.37478389 0.034445122 0.02689368  
## alcohol -0.22991709 -0.047094832 -0.18816480  
## quality -0.12408453 -0.063259641 -0.18333915  
## Id -0.08809907 0.095267572 -0.10738941  
## density pH sulphates alcohol  
## fixed.acidity 0.68150088 -0.68516260 0.17459183 -0.07505485  
## volatile.acidity 0.01651152 0.22149152 -0.27607860 -0.20390927  
## citric.acid 0.37524326 -0.54633914 0.33123176 0.10625034  
## residual.sugar 0.38014695 -0.11695894 0.01747450 0.05842061  
## chlorides 0.20890071 -0.27775907 0.37478389 -0.22991709  
## free.sulfur.dioxide -0.05415032 0.07280371 0.03444512 -0.04709483  
## total.sulfur.dioxide 0.05017483 -0.05912572 0.02689368 -0.18816480  
## density 1.00000000 -0.35277462 0.14313929 -0.49472690  
## pH -0.35277462 1.00000000 -0.18549903 0.22532220  
## sulphates 0.14313929 -0.18549903 1.00000000 0.09442113  
## alcohol -0.49472690 0.22532220 0.09442113 1.00000000  
## quality -0.17520792 -0.05245303 0.25771026 0.48486621  
## Id -0.36392568 0.13290421 -0.10395366 0.23808652  
## quality Id  
## fixed.acidity 0.12197010 -0.27582627  
## volatile.acidity -0.40739351 -0.00789157  
## citric.acid 0.24082084 -0.13901077  
## residual.sugar 0.02200193 -0.04634440  
## chlorides -0.12408453 -0.08809907  
## free.sulfur.dioxide -0.06325964 0.09526757  
## total.sulfur.dioxide -0.18333915 -0.10738941  
## density -0.17520792 -0.36392568  
## pH -0.05245303 0.13290421  
## sulphates 0.25771026 -0.10395366  
## alcohol 0.48486621 0.23808652  
## quality 1.00000000 0.06970824  
## Id 0.06970824 1.00000000

# Add the quality column to the matrix  
# First, get correlations between each numeric variable and the quality  
cor\_with\_quality <- sapply(numeric\_data, function(x) cor(x, data\_clean$quality\_numeric))  
  
# Add correlations with quality to the matrix  
cor\_matrix <- cbind(cor\_matrix, quality = cor\_with\_quality)  
cor\_matrix <- rbind(cor\_matrix, quality = c(cor\_with\_quality, 1))

### EXPLANATION

The above code helps us to understand how the variables are correlated to each other and form the code we got the following info: There is inter variable relationship such as:

1. fixed.acidity and citric.acid have a high positive correlation (0.67315725), indicating they tend to increase or decrease together.
2. total.sulfur.dioxide and free.sulfur.dioxide have a high positive correlation (0.66109287), showing a strong linear relationship between them.

Some of the overall trends:

1. Variables like citric.acid, residual.sugar, and sulphates show positive correlations with quality, which might imply that higher values of these variables are associated with higher quality.
2. Variables like volatile.acidity and chlorides show negative correlations with quality, suggesting that higher values of these variables are associated with lower quality.

# Fit the varaious modles  
# Log-linear model (using GLM with log link)  
log\_linear\_model <- glm(quality\_numeric ~ alcohol + citric.acid + sulphates, family = poisson(link = "log"), data = data\_clean)  
summary(log\_linear\_model)

##   
## Call:  
## glm(formula = quality\_numeric ~ alcohol + citric.acid + sulphates,   
## family = poisson(link = "log"), data = data\_clean)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.18440 0.15296 1.206 0.2280   
## alcohol 0.08856 0.01380 6.419 1.37e-10 \*\*\*  
## citric.acid 0.14947 0.08239 1.814 0.0696 .   
## sulphates 0.21369 0.09276 2.304 0.0212 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 204.50 on 1142 degrees of freedom  
## Residual deviance: 145.86 on 1139 degrees of freedom  
## AIC: 3762.2  
##   
## Number of Fisher Scoring iterations: 4

# Fit a Poisson regression model  
pois\_model <- glm(quality\_numeric ~ alcohol + citric.acid + sulphates,   
 family = poisson(), data = data\_clean)  
summary(pois\_model)

##   
## Call:  
## glm(formula = quality\_numeric ~ alcohol + citric.acid + sulphates,   
## family = poisson(), data = data\_clean)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.18440 0.15296 1.206 0.2280   
## alcohol 0.08856 0.01380 6.419 1.37e-10 \*\*\*  
## citric.acid 0.14947 0.08239 1.814 0.0696 .   
## sulphates 0.21369 0.09276 2.304 0.0212 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 204.50 on 1142 degrees of freedom  
## Residual deviance: 145.86 on 1139 degrees of freedom  
## AIC: 3762.2  
##   
## Number of Fisher Scoring iterations: 4

# Fit a Negative Binomial model  
negbin\_model <- glm.nb(quality\_numeric ~ alcohol + citric.acid + sulphates, data = data\_clean)

## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =  
## control$trace > : iteration limit reached  
## Warning in theta.ml(Y, mu, sum(w), w, limit = control$maxit, trace =  
## control$trace > : iteration limit reached

summary(negbin\_model)

##   
## Call:  
## glm.nb(formula = quality\_numeric ~ alcohol + citric.acid + sulphates,   
## data = data\_clean, init.theta = 526417.0391, link = log)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.18440 0.15296 1.206 0.2280   
## alcohol 0.08856 0.01380 6.419 1.37e-10 \*\*\*  
## citric.acid 0.14947 0.08239 1.814 0.0697 .   
## sulphates 0.21369 0.09276 2.304 0.0212 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Negative Binomial(526417) family taken to be 1)  
##   
## Null deviance: 204.50 on 1142 degrees of freedom  
## Residual deviance: 145.85 on 1139 degrees of freedom  
## AIC: 3764.2  
##   
## Number of Fisher Scoring iterations: 1  
##   
##   
## Theta: 526417   
## Std. Err.: 3439264   
## Warning while fitting theta: iteration limit reached   
##   
## 2 x log-likelihood: -3754.199

### EXPLANATAION

The above code does the following: -

* For log -linear model

1. Formula: The model predicts quality\_numeric (the response variable) as a function of alcohol, citric.acid, and sulphates (the predictor variables).
2. Family: The model uses the Poisson distribution with a log link function, which is suitable for count data or rates.

* For poisson model

1. For the above code we code ‘family = poisson()’ specifies that the response variable follows a Poisson distribution, which is appropriate for count data or rates. The Poisson model assumes that the log of the expected value of the response variable is a linear combination of the predictors.
2. Poisson regression assumes the dispersion parameter is 1, which means the variance equals the mean.

* For neagtive binomial

1. The code ‘glm.nb()’ is a function fits a Generalized Linear Model with a Negative Binomial distribution. The Negative Binomial model is used for count data that may exhibit overdispersion (where the variance exceeds the mean), which the Poisson model cannot handle.
2. The theta parameter represents the dispersion of the Negative Binomial model. A large theta value often suggests that the data is overdispersed compared to a Poisson model. The standard error is large, which might be related to the convergence warnings.

In all the models we have generated above we get the following insights from the code :

1. The intercept represents the value of quality\_numeric when all predictors are zero. Since its p-value is greater than 0.05, it is not statistically significant.
2. For a one-unit increase in alcohol, the value of quality\_numeric increases by 0.08856. The effect is highly significant, as indicated by the very small p-value.
3. For a one-unit increase in citric.acid, the value of quality\_numeric increases by 0.14947. This effect is not significant, as the p-value is above to the typical 0.05 threshold.
4. For a one-unit increase in sulphates, the value of quality\_numeric increases by 0.21369. This effect is statistically significant at the 0.05 level.
5. The null deviance measures how well the null model (intercept only) fits the data. It serves as a baseline for comparing the fit of the current model.
6. The number of fisher scoring iterations: is the number of iterations the algorithm took to converge to the final parameter estimates. Here all the number of iterations are the same except that of negative binomial which is 1.
7. AIC provides a measure of model quality, balancing fit and complexity. Lower AIC values indicate a better model, but it should be compared across models to be meaningful.

# Fit a Quasi-Poisson model  
quasipoisson\_model <- glm(quality\_numeric ~ alcohol + citric.acid + sulphates,   
 family = quasipoisson(), data = data\_clean)  
summary(quasipoisson\_model)

##   
## Call:  
## glm(formula = quality\_numeric ~ alcohol + citric.acid + sulphates,   
## family = quasipoisson(), data = data\_clean)  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.184405 0.053765 3.430 0.000626 \*\*\*  
## alcohol 0.088560 0.004849 18.262 < 2e-16 \*\*\*  
## citric.acid 0.149469 0.028959 5.161 2.89e-07 \*\*\*  
## sulphates 0.213692 0.032604 6.554 8.46e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for quasipoisson family taken to be 0.1235472)  
##   
## Null deviance: 204.50 on 1142 degrees of freedom  
## Residual deviance: 145.86 on 1139 degrees of freedom  
## AIC: NA  
##   
## Number of Fisher Scoring iterations: 4

### EXPLANATION

The Quasi-Poisson model was used to fit the data, addressing overdispersion where the variance exceeds the mean, which is common in count data. This model employs the quasipoisson() family, adjusting the Poisson distribution to account for this overdispersion.

The estimated dispersion parameter for this model was 0.1235472, indicating a certain degree of overdispersion.

Unlike other models, the Quasi-Poisson model does not provide an Akaike Information Criterion (AIC) value because AIC is not typically defined for quasi-likelihood models.

The analysis revealed that all three predictors—alcohol, citric.acid, and sulphates—are statistically significant, with positive coefficients suggesting that increases in these predictors are associated with higher values of the response variable, quality\_numeric.

Notably, the intercept remains consistent across all models, including the Quasi-Poisson model. This model differs from others by showing that all explanatory variables are significant, unlike the results obtained from the alternative models.

# Compare models using AIC  
aic\_values <- c(  
 log\_linear = AIC(log\_linear\_model),  
 poisson = AIC(pois\_model),  
 negbin = AIC(negbin\_model),  
 quasipoisson = AIC(quasipoisson\_model)  
)  
print(aic\_values)

## log\_linear poisson negbin quasipoisson   
## 3762.192 3762.192 3764.199 NA

### EXPLANATION

The log-linear and Poisson models have the same AIC value of 3762.192, suggesting they fit the data equally well in terms of AIC.

The Negative Binomial model has a slightly higher AIC value of 3764.199, indicating a slightly worse fit compared to the log-linear and Poisson models.

The Quasi-Poisson model cannot be compared using AIC because AIC is not defined for quasi-likelihood models.

# Create a list of models  
fm <- list(log\_linear\_model, pois\_model, negbin\_model, quasipoisson\_model)  
  
# Compare standard errors of coefficients  
se\_comparison <- sapply(fm, function(x) sqrt(diag(vcov(x))))  
print(se\_comparison)

## [,1] [,2] [,3] [,4]  
## (Intercept) 0.15296100 0.15296100 0.15296157 0.053764697  
## alcohol 0.01379668 0.01379668 0.01379673 0.004849433  
## citric.acid 0.08238943 0.08238943 0.08238973 0.028959296  
## sulphates 0.09275797 0.09275797 0.09275833 0.032603764

### EXPLANATION

The standard errors for the coefficients (Intercept, alcohol, citric.acid, and sulphates) are similar across the models, indicating comparable variability in coefficient estimates.

However, the Quasi-Poisson model has noticeably smaller standard errors than the other models, suggesting it estimates coefficients with less variability. This means that the Quasi-Poisson model may provide the most precise estimates of the coefficients.

# Compare log-likelihood and degrees of freedom  
logLik\_comparison <- rbind(  
 LogLik = sapply(fm, function(x) if (inherits(x, "glm")) logLik(x) else NA),  
 Df = sapply(fm, function(x) if (inherits(x, "glm")) attr(logLik(x), "df") else NA)  
)  
  
print(logLik\_comparison)

## [,1] [,2] [,3] [,4]  
## LogLik -1877.096 -1877.096 -1877.099 NA  
## Df 4.000 4.000 5.000 5

# EXPLANATION

The provided code snippet compares log-likelihood values and degrees of freedom across different regression models to assess fit and complexity. Log-likelihood values reflect model fit, while degrees of freedom represent the number of parameters.

The Negative Binomial model shows a slightly different fit compared to the Log-linear and Poisson models, with both the Negative Binomial and Quasi-Poisson models incorporating an additional parameter compared to the Log-linear and Poisson models.

## QUESTION 2 [show your R codes, R output, description of the process and Interpretation] (10 Marks)

### (a) Using appropriate statistical data, perform non-parametric regression analysis using any technique. Interpret the output

# Load the necessary libraries  
library(readxl)  
library(ggplot2)  
library(caret)

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.3.3

library(kknn)

## Warning: package 'kknn' was built under R version 4.3.3

##   
## Attaching package: 'kknn'

## The following object is masked from 'package:caret':  
##   
## contr.dummy

# Load the data  
data <- read.csv("C:\\Users\\HP\\Desktop\\R- CODES\\housing.csv")  
  
# Inspect the data structure and summary  
str(data)

## 'data.frame': 20640 obs. of 10 variables:  
## $ longitude : num -122 -122 -122 -122 -122 ...  
## $ latitude : num 37.9 37.9 37.9 37.9 37.9 ...  
## $ housing\_median\_age: int 41 21 52 52 52 52 52 52 42 52 ...  
## $ total\_rooms : int 880 7099 1467 1274 1627 919 2535 3104 2555 3549 ...  
## $ total\_bedrooms : int 129 1106 190 235 280 213 489 687 665 707 ...  
## $ population : int 322 2401 496 558 565 413 1094 1157 1206 1551 ...  
## $ households : int 126 1138 177 219 259 193 514 647 595 714 ...  
## $ median\_income : num 8.33 8.3 7.26 5.64 3.85 ...  
## $ median\_house\_value: int 452600 358500 352100 341300 342200 269700 299200 241400 226700 261100 ...  
## $ ocean\_proximity : chr "NEAR BAY" "NEAR BAY" "NEAR BAY" "NEAR BAY" ...

summary(data)

## longitude latitude housing\_median\_age total\_rooms   
## Min. :-124.3 Min. :32.54 Min. : 1.00 Min. : 2   
## 1st Qu.:-121.8 1st Qu.:33.93 1st Qu.:18.00 1st Qu.: 1448   
## Median :-118.5 Median :34.26 Median :29.00 Median : 2127   
## Mean :-119.6 Mean :35.63 Mean :28.64 Mean : 2636   
## 3rd Qu.:-118.0 3rd Qu.:37.71 3rd Qu.:37.00 3rd Qu.: 3148   
## Max. :-114.3 Max. :41.95 Max. :52.00 Max. :39320   
##   
## total\_bedrooms population households median\_income   
## Min. : 1.0 Min. : 3 Min. : 1.0 Min. : 0.4999   
## 1st Qu.: 296.0 1st Qu.: 787 1st Qu.: 280.0 1st Qu.: 2.5634   
## Median : 435.0 Median : 1166 Median : 409.0 Median : 3.5348   
## Mean : 537.9 Mean : 1425 Mean : 499.5 Mean : 3.8707   
## 3rd Qu.: 647.0 3rd Qu.: 1725 3rd Qu.: 605.0 3rd Qu.: 4.7432   
## Max. :6445.0 Max. :35682 Max. :6082.0 Max. :15.0001   
## NA's :207   
## median\_house\_value ocean\_proximity   
## Min. : 14999 Length:20640   
## 1st Qu.:119600 Class :character   
## Median :179700 Mode :character   
## Mean :206856   
## 3rd Qu.:264725   
## Max. :500001   
##

# Check for missing values  
sum(is.na(data$total\_bedrooms))

## [1] 207

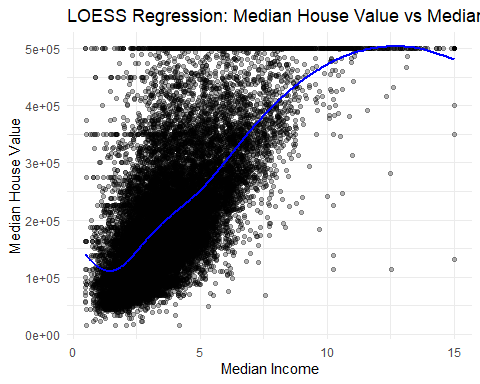
# Omit the data   
data <- na.omit(data)  
  
# Impute missing values with median  
data$total\_bedrooms[is.na(data$total\_bedrooms)] <- median(data$total\_bedrooms, na.rm = TRUE)

### EXPLANATION

The 207 indictates missing values in the total\_bedrooms variable (before handling missing values). Missing values are checked and handled by either removing rows with missing values or imputing them with the median.

# Perform LOESS non-parametric regression  
loess\_fit <- loess(median\_house\_value ~ median\_income, data = data, span = 0.5)  
  
# Predict using the LOESS model  
data$predicted\_house\_value\_loess <- predict(loess\_fit, data)  
  
# Plot the results  
ggplot(data, aes(x = median\_income, y = median\_house\_value)) +  
 geom\_point(alpha = 0.3) +  
 geom\_line(aes(y = predicted\_house\_value\_loess), color = 'blue', size = 1) +  
 labs(title = "LOESS Regression: Median House Value vs Median Income",  
 x = "Median Income",  
 y = "Median House Value") +  
 theme\_minimal()

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



### EXPLANATION

In the graph, each data point represents a specific location, likely a region or neighborhood. The concentration of points in the lower left corner indicates that areas with lower median incomes generally have lower median house values.

As we move towards the upper right, the data points become more spread out, showing that higher median incomes are associated with higher median house values.

The smooth LOESS (Locally Estimated Scatterplot Smoothing) curve overlaid on the scatter plot illustrates the underlying trend. It starts from the lower left (low median income, low house value) and curves upward to the upper right (high median income, high house value).

The positive slope of the LOESS curve suggests that median house values tend to increase with median income, although there is noticeable variability around this trend.

# QUESTION 3 [show your R codes, R output, description of the process and Interpretation] (20 Marks)

### (a) Using appropriate statistical data, perform any two appropriate analysis that can be used in observations that clustered within groups. Interpret the output

# Load the necessary libraries  
library(readxl)  
library(lme4) # For mixed-effects models

## Warning: package 'lme4' was built under R version 4.3.3

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.3.2

library(geepack) # For GEE models

## Warning: package 'geepack' was built under R version 4.3.3

library(car) # For ANOVA and other statistical functions  
library(dplyr) # For data manipulation  
library(cluster) # For hierarchical clustering

## Warning: package 'cluster' was built under R version 4.3.3

library(factoextra) # For clustering visualization

## Warning: package 'factoextra' was built under R version 4.3.3

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

# Load the data  
data <- read.csv("C:\\Users\\HP\\Desktop\\R- CODES\\diabetes.csv")  
  
# Inspect the data structure and summary  
str(data)

## 'data.frame': 768 obs. of 9 variables:  
## $ Pregnancies : int 6 1 8 1 0 5 3 10 2 8 ...  
## $ Glucose : int 148 85 183 89 137 116 78 115 197 125 ...  
## $ BloodPressure : int 72 66 64 66 40 74 50 0 70 96 ...  
## $ SkinThickness : int 35 29 0 23 35 0 32 0 45 0 ...  
## $ Insulin : int 0 0 0 94 168 0 88 0 543 0 ...  
## $ BMI : num 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...  
## $ DiabetesPedigreeFunction: num 0.627 0.351 0.672 0.167 2.288 ...  
## $ Age : int 50 31 32 21 33 30 26 29 53 54 ...  
## $ Outcome : int 1 0 1 0 1 0 1 0 1 1 ...

summary(data)

## Pregnancies Glucose BloodPressure SkinThickness   
## Min. : 0.000 Min. : 0.0 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 1.000 1st Qu.: 99.0 1st Qu.: 62.00 1st Qu.: 0.00   
## Median : 3.000 Median :117.0 Median : 72.00 Median :23.00   
## Mean : 3.845 Mean :120.9 Mean : 69.11 Mean :20.54   
## 3rd Qu.: 6.000 3rd Qu.:140.2 3rd Qu.: 80.00 3rd Qu.:32.00   
## Max. :17.000 Max. :199.0 Max. :122.00 Max. :99.00   
## Insulin BMI DiabetesPedigreeFunction Age   
## Min. : 0.0 Min. : 0.00 Min. :0.0780 Min. :21.00   
## 1st Qu.: 0.0 1st Qu.:27.30 1st Qu.:0.2437 1st Qu.:24.00   
## Median : 30.5 Median :32.00 Median :0.3725 Median :29.00   
## Mean : 79.8 Mean :31.99 Mean :0.4719 Mean :33.24   
## 3rd Qu.:127.2 3rd Qu.:36.60 3rd Qu.:0.6262 3rd Qu.:41.00   
## Max. :846.0 Max. :67.10 Max. :2.4200 Max. :81.00   
## Outcome   
## Min. :0.000   
## 1st Qu.:0.000   
## Median :0.000   
## Mean :0.349   
## 3rd Qu.:1.000   
## Max. :1.000

# Check for missing values  
sum(is.na(data))

## [1] 0

# Handle missing values by replacing NaN and Inf with NA  
data <- data %>% mutate(across(everything(), ~ replace(.x, is.nan(.x) | is.infinite(.x), NA)))  
  
# Remove rows with any NA values  
data <- na.omit(data)  
  
# Simulate clustering  
set.seed(123)  
data$cluster <- as.factor(sample(1:10, size = nrow(data), replace = TRUE))  
  
# Inspect the simulated clustering  
table(data$cluster)

##   
## 1 2 3 4 5 6 7 8 9 10   
## 68 68 75 69 68 71 91 88 82 88

### EXPLANATION

The code handles missing values by replacing NaN and Inf with NA and removing rows with NA values, and simulate clustering by adding a cluster ID variable to each observation.It then displays the number of observations in each cluster.

# Fit a GEE model  
gee\_model <- geeglm( Outcome ~ Glucose + BloodPressure, id = cluster, family = gaussian, data = data)  
  
# Summarize the model  
summary(gee\_model)

##   
## Call:  
## geeglm(formula = Outcome ~ Glucose + BloodPressure, family = gaussian,   
## data = data, id = cluster)  
##   
## Coefficients:  
## Estimate Std.err Wald Pr(>|W|)   
## (Intercept) -0.4835390 0.0760926 40.381 2.09e-10 \*\*\*  
## Glucose 0.0069745 0.0005404 166.588 < 2e-16 \*\*\*  
## BloodPressure -0.0001546 0.0008067 0.037 0.848   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation structure = independence   
## Estimated Scale Parameters:  
##   
## Estimate Std.err  
## (Intercept) 0.1777 0.007582  
## Number of clusters: 693 Maximum cluster size: 4

### EXPLANATION

The model has an intercept value of -0.4835390, representing the expected outcome when all predictors are zero.

The intercept is highly significant with a Wald test p-value of 2.09e-10. The Glucose coefficient is also highly significant, with a Wald test p-value < 2e-16. In contrast, the BloodPressure coefficient is not significant, with a p-value of 0.848.

The code snippet correlation structure = independence specifies that an independence correlation structure was used for clustering.

The model includes 693 clusters, with the largest cluster containing 4 observations. The estimated scale parameter is 0.1777, indicating unexplained variability in the outcome.

The GEE model, which accounts for clustering, finds Glucose to be a significant predictor, while BloodPressure is not, and the residual variability is captured by the scale parameter.

# Fit a simplified linear mixed-effects model  
model <- lmer(Outcome ~ Glucose + BloodPressure + (1 | cluster), data = data,control = lmerControl(optimizer = "bobyqa"))  
  
summary(model)

## Linear mixed model fit by REML ['lmerMod']  
## Formula: Outcome ~ Glucose + BloodPressure + (1 | cluster)  
## Data: data  
## Control: lmerControl(optimizer = "bobyqa")  
##   
## REML criterion at convergence: 883.8  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.128 -0.730 -0.349 0.886 3.544   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## cluster (Intercept) 0.00158 0.0397   
## Residual 0.17703 0.4207   
## Number of obs: 768, groups: cluster, 10  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) -0.491869 0.076409 -6.44  
## Glucose 0.007032 0.000482 14.57  
## BloodPressure -0.000123 0.000797 -0.15  
##   
## Correlation of Fixed Effects:  
## (Intr) Glucos  
## Glucose -0.652   
## BloodPressr -0.604 -0.153

### EXPLANATION

The following are the insights form the above code:

1. Degrees of Freedom Calculation: With 768 observations and 3 fixed effects (including the intercept), the degrees of freedom are 768 − 3 = 765
2. Critical t-value: For a two-tailed test with 𝛼= 0.05 and 765 degrees of freedom, the critical t-value is approximately 1.96.

We can interpret the t-Values as follows:

1. Intercept: -6.44, indicating statistical significance.
2. Glucose Coefficient: 14.57, indicating high significance.
3. Blood Pressure Coefficient: -0.15, indicating it is not significant.

For the Correlations we can interpret it as follows:

1. Between the intercept and Glucose coefficient: -0.652.
2. Between Glucose and Blood Pressure coefficients: -0.153.

Lastly for the model description we have the following insights:

1. The model fits a linear mixed-effects model with Glucose and Blood Pressure as fixed effects and a random intercept for each cluster.
2. It uses 768 clusters, with the largest cluster containing 10 observations. The summary output includes the REML criterion, scaled residuals, variances of random effects, and fixed effects estimates with their standard errors and t-values.
3. Glucose is a significant predictor of the outcome, while Blood Pressure is not. The scale parameter indicates the residual variability not explained by the model.

# Perform ANOVA test to compare fixed effects  
anova\_result <- Anova(model, type = 3)  
print(anova\_result)

## Analysis of Deviance Table (Type III Wald chisquare tests)  
##   
## Response: Outcome  
## Chisq Df Pr(>Chisq)   
## (Intercept) 41.44 1 1.2e-10 \*\*\*  
## Glucose 212.42 1 < 2e-16 \*\*\*  
## BloodPressure 0.02 1 0.88   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

### EXPLANATION

The p-value associated with the chi-square statistic reflects the probability of observing a chi-square statistic as extreme as, or more extreme than, the one obtained if the null hypothesis is true.

For the intercept, the p-value is highly significant (p < 0.001), indicating a strong contribution to the model, as the Wald chi-square test statistic is very large.

Similarly, the Glucose variable is highly significant (p < 0.001), with a very large chi-square statistic and extremely small p-value, highlighting its importance as a predictor of the outcome.

In contrast, the Blood Pressure variable is not significant (p = 0.88), with a very small chi-square statistic and a p-value much larger than 0.05, suggesting it does not significantly contribute to the model.

# Inspect random effects  
random\_effects <- ranef(model)  
print(random\_effects)

## $cluster  
## (Intercept)  
## 1 -0.028230  
## 2 0.006730  
## 3 0.010782  
## 4 0.048136  
## 5 0.020404  
## 6 -0.013585  
## 7 -0.013079  
## 8 0.010551  
## 9 -0.039511  
## 10 -0.002197  
##   
## with conditional variances for "cluster"

### EXPLANATION

The random effects represent deviations from the overall intercept for each cluster.

For instance, in the model, the first cluster has an intercept that is -0.028230 units below the overall intercept, while the fourth cluster’s intercept is 0.048136 units above the overall intercept.

With 10 clusters, each has its own random effect for the intercept, reflecting how much the intercept for each cluster deviates from the overall model intercept.