STAT8111-Assignment 2

Aditya Sagave

Student ID: 47541164

Question 1

```
pbc_data <- read_csv(here::here("pbc.csv"))
```

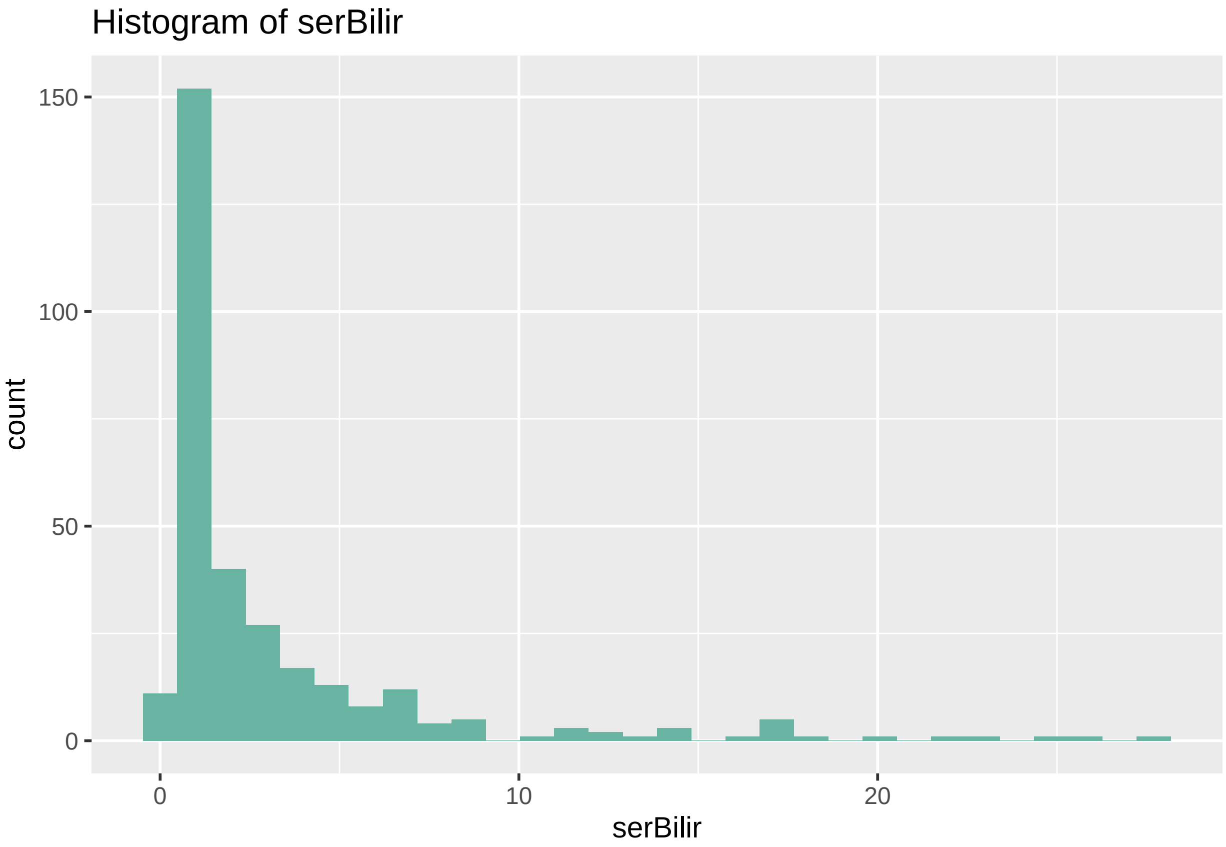
```
## Rows: 312 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): sex, hepatomegaly
## dbl (6): age, serBilir, albumin, alkaline, prothrombin, histologic
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(pbc_data,10)
```

```
## # A tibble: 10 x 8
##      age sex    hepatomegaly serBilir albumin alkaline prothrombin histologic
##    <dbl> <chr>  <chr>           <dbl>   <dbl>    <dbl>       <dbl>      <dbl>
##  1  58.8 female Yes              14.5    2.6      1718        12.2          4
##  2  56.4 female Yes               1.1    4.14     7395        10.6          3
##  3  70.1 male   No                1.4    3.48      516        12            4
##  4  54.7 female Yes               1.8    2.54     6122        10.3          4
##  5  38.1 female Yes               3.4    3.53      671        10.9          3
##  6  66.3 female Yes               0.8    3.98      944        11            3
##  7  55.5 female Yes               1      4.09      824         9.7          3
##  8  53.1 female No                0.3    4        4651        11            3
##  9  42.5 female No                3.2    3.08     2276        11            2
## 10  70.6 female No               12.6    2.74      918        11.5          4
```

```
ggplot(pbc_data, aes(x = serBilir)) +
  geom_histogram(fill="#69b3a2") +
  labs(title = "Histogram of serBilir", x = "serBilir")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



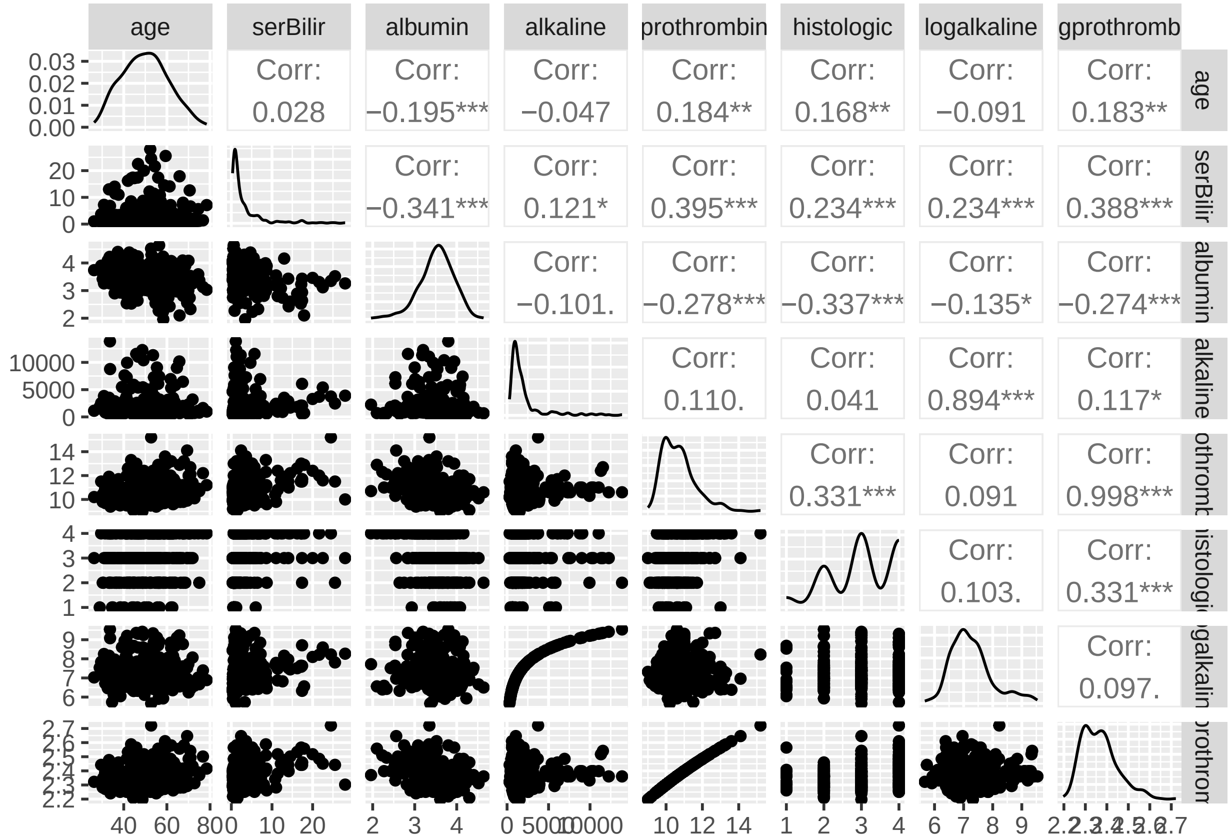
Answer 1.a: The right-skewed plot of "serBilir" suggests it may be best described by a Gamma or Inverse Gaussian Distribution, both of which are parameterized by two parameters.

Answer 1.b: When evaluating the alkaline predictor, we noticed a strong right skew in its distribution. We experimented with both logarithmic and square root transformations. Logarithm transformation effectively normalized the distribution, while square root did not. Similarly, the prothrombin variable exhibited right skewness, and applying a log transformation successfully brought it closer to a normal distribution.

```
# Apply a natural logarithm (base e) transformation to prothrombin
pbc_data$logalkaline <- log(pbc_data$alkaline)
pbc_data$logprothrombin <- log(pbc_data$prothrombin)
pbc_data$logserBilir <- log(pbc_data$serBilir)
```

Answer 1.c

```
ggpairs(pbc_data, columns = c("age", "serBilir", "albumin", "alkaline", "prothrombin", "histologic", "l
```



In our analysis, we explored the relationships between continuous covariates and the target variable serBilir. Here are our key findings:

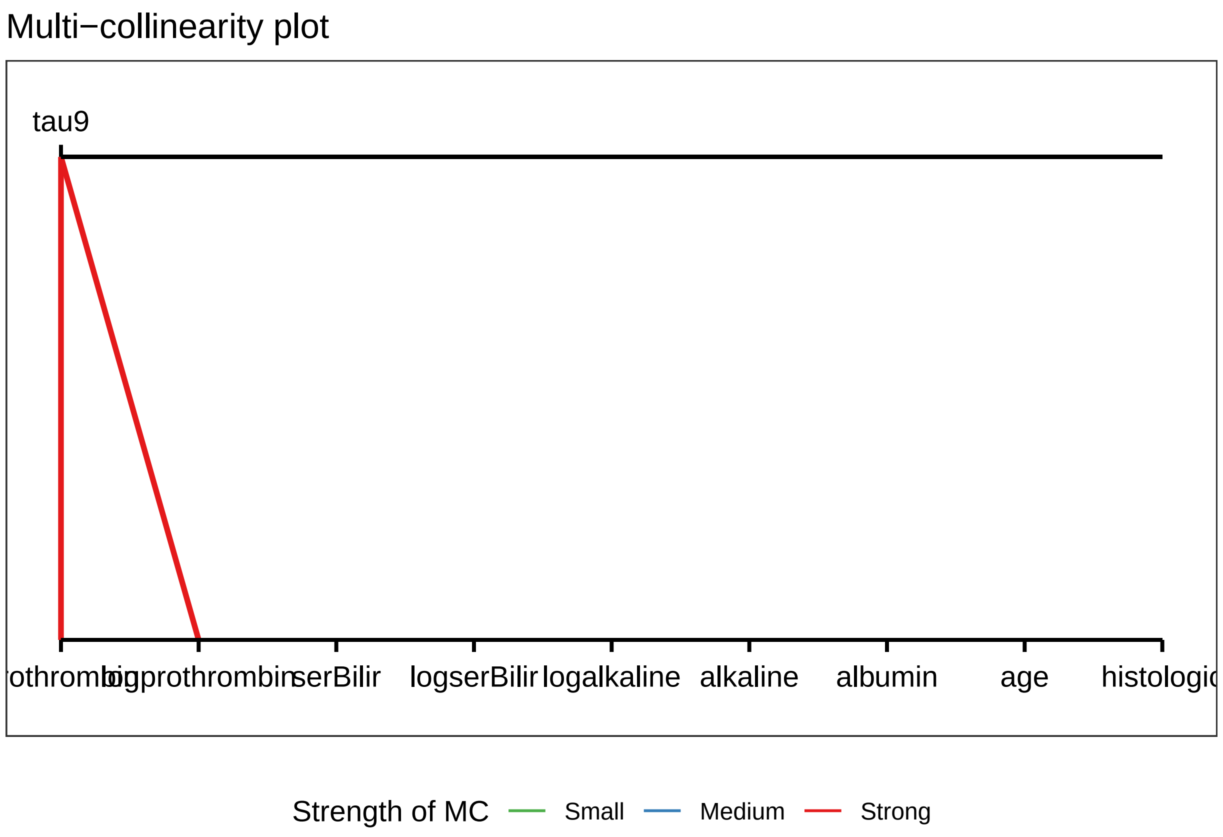
Predictors of serBilir:

1. albumin: Strong negative correlation (-0.341) with serBilir, suggesting lower bilirubin levels with higher albumin levels.

2. prothrombin: Strong positive correlation (0.395) with serBilir, indicating higher bilirubin levels as prothrombin time increases.

3. logprothrombin: Strong positive correlation (0.388) with serBilir, indicating higher bilirubin levels as logprothrombin time increases.

```
plot(mcvis(pbc_data[ ,!(colnames(pbc_data) %in% c("sex", "hepatomegaly"))]))
```



Collinearity:

1. Strong positive collinearity (0.894) between alkaline and logalkaline.
2. Strong positive collinearity (0.998) between prothrombin and logprothrombin
3. Negative collinearity (-0.336) between albumin and histologic.

Answer 1.d)

```
frequency_sex <- table(pbc_data$sex)
frequency_hepatomegaly <- table(pbc_data$hepatomegaly)
frequency_histologic <- table(pbc_data$histologic)

# Display the frequency tables
frequency_sex
```

```
##
## female   male
##    276     36
```

```
frequency_hepatomegaly
```

```
##
##  No Yes
## 152 160
```

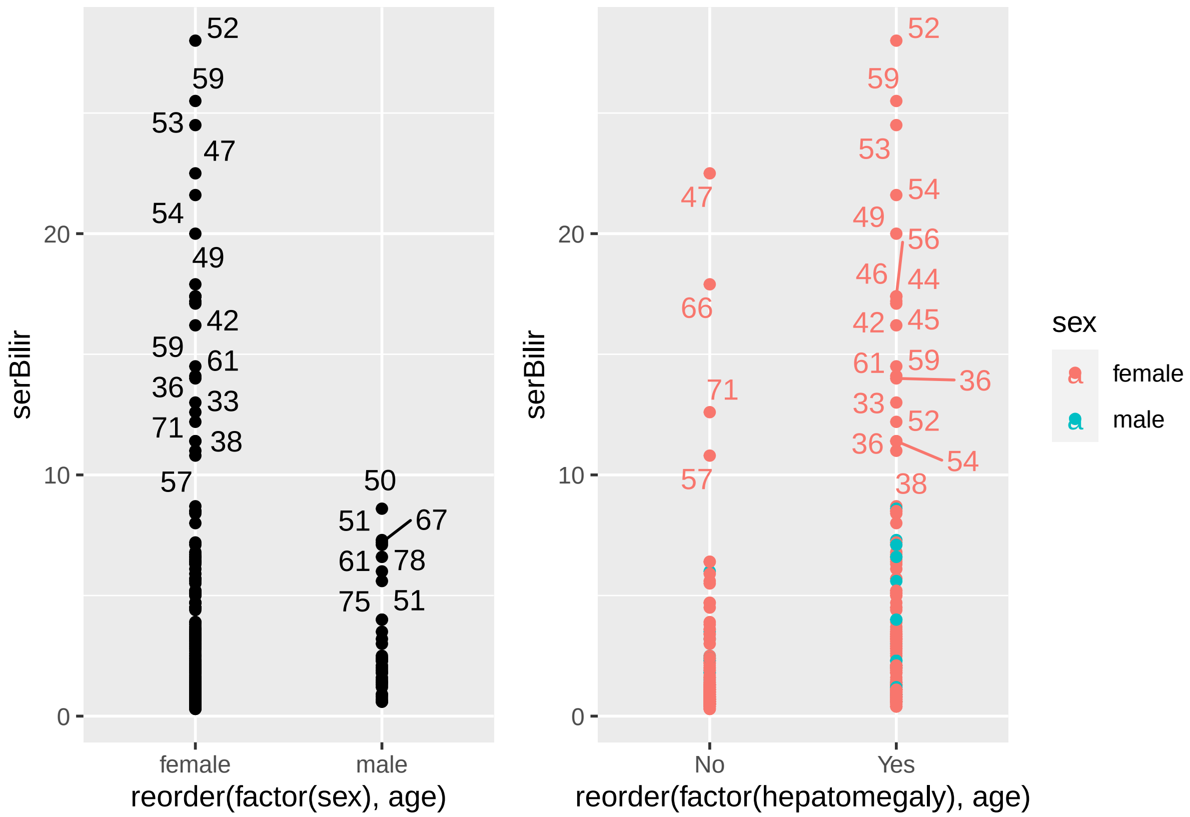
```
frequency_histologic
```

```
##
```

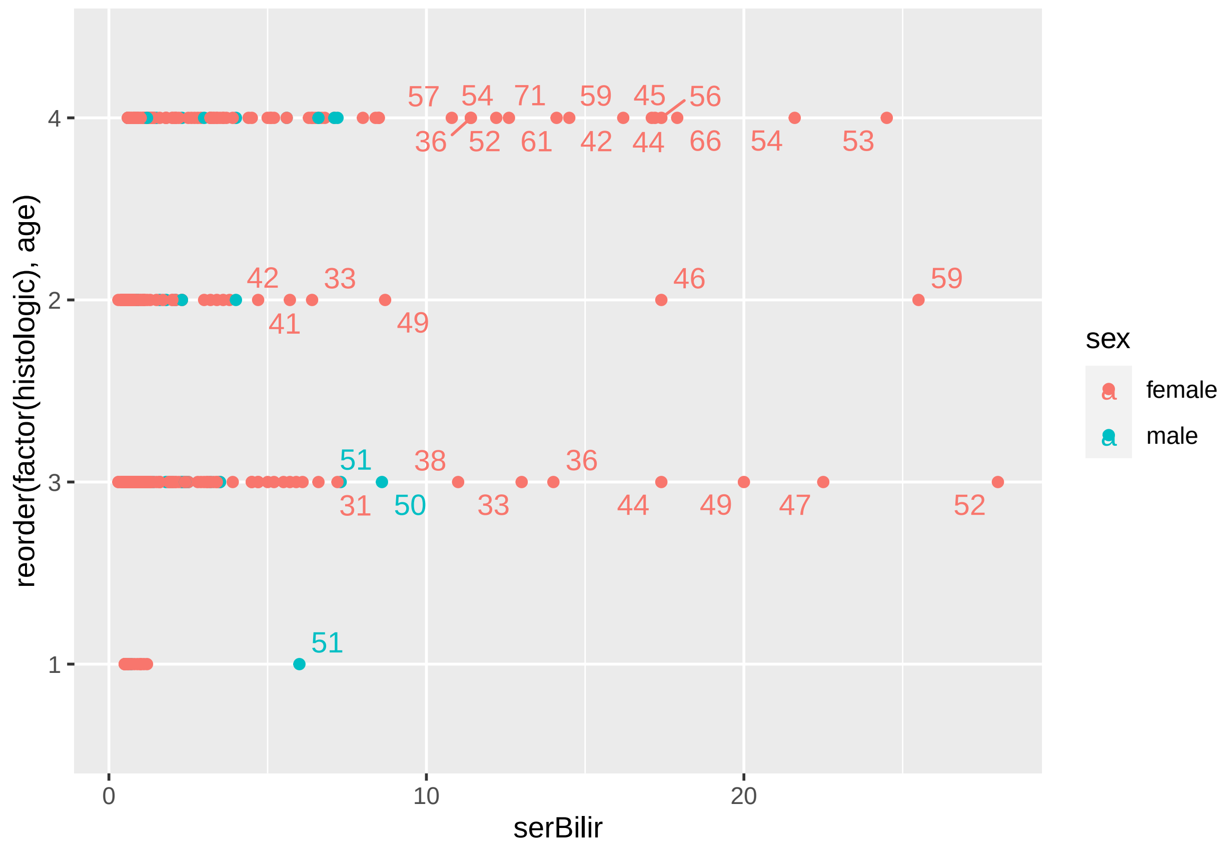
```
##   1   2   3   4
##  16  67 120 109
```

Answer 1.e)

```
#Create a scatter plot for 'serBilir' vs. 'sex'
p1 <- ggplot(pbc_data, aes(x = reorder(factor(sex), age), y = serBilir, label = round(age,))) +
  geom_point()+
  geom_text_repel()

p2 <- ggplot(pbc_data, aes(x = reorder(factor(hepatomegaly), age), y = serBilir,color=sex, label = rounc
  geom_point()+
  geom_text_repel()

p3 <- ggplot(pbc_data, aes(y = reorder(factor(histologic), age), x = serBilir,color=sex, label = round(a
  geom_point()+
  geom_text_repel()

p1 + p2
```



```
p3
```



In the serBilir vs. gender plot, we observe that most cases involve females, particularly those aged 30-70, with less than 10 mg/dl of serBilir.

In the serBilir vs. hepatomegaly plot, females are predominantly affected, especially those over 35 years old, showing a higher risk of hepatomegaly.

In the serBilir vs. histologic plot, there are more female records, and among females, levels 3 and 4 of histologic stage are most common when plotted against serBilir.

Answer 1.f)

```
m1 <- glm(serBilir ~ round(age,), data = pbc_data, family = Gamma(link = "log"))
m2 <- glm(serBilir ~ factor(sex), data = pbc_data, family = Gamma(link = "log"))
m3 <- glm(serBilir ~ factor(hepatomegaly), data = pbc_data, family = Gamma(link = "log"))
m4 <- glm(serBilir ~ albumin, data = pbc_data, family = Gamma(link = "log"))
m5 <- glm(serBilir ~ alkaline, data = pbc_data, family = Gamma(link = "log"))
m6 <- glm(serBilir ~ prothrombin, data = pbc_data, family = Gamma(link = "log"))
m7 <- glm(serBilir ~ histologic, data = pbc_data, family = Gamma(link = "log"))
m8 <- glm(serBilir ~ logalkaline, data = pbc_data, family = Gamma(link = "log"))
m9 <- glm(serBilir ~ logprothrombin, data = pbc_data, family = Gamma(link = "log"))

summary(m4)
```

```
##
## Call:
## glm(formula = serBilir ~ albumin, family = Gamma(link = "log"),
##     data = pbc_data)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9137  -0.9815  -0.6162   0.1247   3.3401
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.6451     0.6203   7.488 7.31e-13 ***
## albumin      -1.0157     0.1750  -5.804 1.60e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.679069)
##
##     Null deviance: 373.71  on 311  degrees of freedom
## Residual deviance: 312.39  on 310  degrees of freedom
## AIC: 1297.7
##
## Number of Fisher Scoring iterations: 5
```

```r
# Create an empty dataframe to store results
results_df <- data.frame(Model = character(0), P_values = numeric(0), AIC = numeric(0))

# List of model names
model_names <- c("m1", "m2", "m3", "m4", "m5", "m6", "m7", "m8", "m9")

# Iterate through the models and extract Pr(>|t|) and AIC
for (i in 1:length(model_names)) {
  model <- eval(parse(text = model_names[i]))  # Get the model by its name

  # Extract Pr(>|t|) and AIC and store in the dataframe
  results_df <- rbind(results_df, data.frame(Model = model_names[i], P_values = round(summary(model)$coe
}

# Print the results dataframe
print(results_df)
```

```
##   Model P_values      AIC
## 1    m1  0.01085 1362.530
## 2    m2  0.00000 1362.506
## 3    m3  0.00000 1300.311
## 4    m4  0.00000 1297.731
## 5    m5  0.00000 1350.137
## 6    m6  0.00000 1287.458
## 7    m7  0.57979 1323.034
## 8    m8  0.00041 1324.588
## 9    m9  0.00000 1287.958
```

20% Significance leve i.e p<0.2

Model m1 (Age): Age is statistically significant (p-value = 0.01085) in predicting serBilir levels. The AIC is 1362.530.

Model m2 (Sex): Sex is highly statistically significant (p-value = 0.00000) in predicting serBilir levels. The AIC is 1362.506.

Model m3 (Hepatomegaly): Hepatomegaly is highly statistically significant (p-value = 0.00000) in predicting serBilir levels. The AIC is 1300.311.

Model m4 (Albumin): Albumin is highly statistically significant (p-value = 0.00000) in predicting serBilir levels. The AIC is 1297.731.

Model m5 (Alkaline): Alkaline is highly statistically significant (p-value = 0.00000) in predicting serBilir levels. The AIC is 1350.137.

Model m6 (Prothrombin): Prothrombin is highly statistically significant (p-value = 0.00000) in predicting serBilir levels. The AIC is 1287.458.

Model m8 (Log Alkaline): Log-transformed alkaline is statistically significant (p-value = 0.00041) in predicting serBilir levels. The AIC is 1324.588.

Model m9 (Log Prothrombin): Log-transformed prothrombin is highly statistically significant (p-value = 0.00000) in predicting serBilir levels. The AIC is 1287.958.

Answer 1.g)

Model m6 (Prothrombin): AIC = 1287.458

Model m9 (Log Prothrombin): AIC = 1287.958

Model m4 (Albumin): AIC = 1297.731

Model m3 (Hepatomegaly): AIC = 1300.311

Model m8 (Log Alkaline): AIC = 1324.588

Model m7 (Histologic): AIC = 1323.034

Model m5 (Alkaline): AIC = 1350.137

Model m1 (Age): AIC = 1362.530

Model m2 (Sex): AIC = 1362.506

```
m10<-glm(serBilir~prothrombin, data=pbc_data, family = Gamma(link = "log"))

m11<-glm(serBilir~prothrombin + albumin, data=pbc_data, family = Gamma(link = "log"))

m12<-glm(serBilir~prothrombin + albumin + factor(hepatomegaly), data=pbc_data, family = Gamma(link = "lo

m13<-glm(serBilir~prothrombin + albumin + factor(hepatomegaly) + logalkaline, data=pbc_data, family = Ga

m14<-glm(serBilir~prothrombin + albumin + factor(hepatomegaly) + logalkaline + histologic, data=pbc_data

m15<-glm(serBilir~prothrombin + albumin + factor(hepatomegaly) + logalkaline + histologic + age, data=pb

m16<-glm(serBilir~prothrombin + albumin + factor(hepatomegaly) + logalkaline + histologic + age + factor


# Create an empty dataframe to store results
results_df <- data.frame(Model = character(0), P_values = numeric(0), AIC = numeric(0))

# List of model names
model_names <- c("m10", "m11", "m12", "m13", "m14", "m15", "m16")

# Iterate through the models and extract Pr(>|t|) and AIC
for (i in 1:length(model_names)) {
  model <- eval(parse(text = model_names[i]))  # Get the model by its name

  # Extract Pr(>|t|) and AIC and store in the dataframe
  results_df <- rbind(results_df, data.frame(Model = model_names[i], P_values = round(summary(model)$coe
}

# Print the results dataframe
print(results_df)
```

```
##   Model P_values      AIC
## 1   m10  0.00000 1287.458
## 2   m11  0.82798 1251.637
## 3   m12  0.61151 1223.627
## 4   m13  0.00127 1193.201
## 5   m14  0.00091 1192.984
## 6   m15  0.00635 1192.169
## 7   m16  0.01238 1192.632
```

20% Significance level i.e p<0.2

- Model m10 (Prothrombin): Prothrombin is statistically significant (p-value = 0.00000) in predicting serBilir levels. The AIC is 1287.458.
- Model m13 (Prothrombin + Albumin + Hepatomegaly + Log Alkaline): This is statistically significant (p-value = 0.00127) in predicting serBilir levels. The AIC is 1193.201.
- Model m14 (Prothrombin + Albumin + Hepatomegaly + Log Alkaline + Histologic): This is statistically significant (p-value = 0.00091) in predicting serBilir levels. The AIC is 1192.984.

We've decided to choose Model m14 as our final model due to its statistical significance (p-value = 0.00091) and its lowest AIC value of 1192.984. While there's another model with a slightly lower AIC value, it's advisable to opt for the model with fewer covariates. This choice minimizes the potential for significant changes in results, as each covariate can have a substantial impact on the final outcome.

Answer 1.h)

```
summary(m14)
```

```
## 
## Call:
## glm(formula = serBilir ~ prothrombin + albumin + factor(hepatomegaly) + 
##     logalkaline + histologic, family = Gamma(link = "log"), data = pbc_data)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -1.8501  -0.8258  -0.3807   0.2080   2.8177  
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             -3.74097    1.11659  -3.350 0.000908 ***
## prothrombin              0.27544    0.06492   4.243 2.93e-05 ***
## albumin                 -0.53146    0.14723  -3.610 0.000358 ***
## factor(hepatomegaly)Yes  0.48745    0.12984   3.754 0.000208 ***
## logalkaline              0.42220    0.07924   5.328 1.93e-07 ***
## histologic               0.09112    0.07646   1.192 0.234312    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Gamma family taken to be 0.9846936)
## 
```

```
##     Null deviance: 373.71  on 311  degrees of freedom
## Residual deviance: 227.18  on 306  degrees of freedom
## AIC: 1193
##
## Number of Fisher Scoring iterations: 8
```

Equation for our final model m14 will be:

$$ser\hat{B}ilir_i = \beta_0 + \beta_1 * \text{prothrombin} + \beta_2 * \text{albumin} + \beta_3 * \text{factor(hepatomegaly)} + \beta_4 * \text{logalkaline} + \beta_5 * \text{histologic}$$

$$ser\hat{B}ilir_i = -3.740 + 0.275 * \text{prothrombin} - 0.531 * \text{albumin} + 0.487 * \text{factor(hepatomegaly)} + 0.422 * \text{logalkaline} + 0.091 * \text{histologic}$$

$Where:$

-serBilir is the serum bilirubin levels, which is the dependent variable we are trying to predict.

-β_0 is the intercept.

-β_1 is the coefficient for the Prothrombin variable.

-β_2 is the coefficient for the Albumin variable.

-β_3 is the coefficient for the Hepatomegaly variable.

-β_4 is the coefficient for the Log-transformed Alkaline variable.

-β_5 is the coefficient for the Histologic variable.

Answer 1.i)

- **Prothrombin (Beta = 0.2754)**: Higher Prothrombin levels (for a one-unit increase) are associated with an approximately 0.2754 increase in the serum bilirubin levels.
- **Albumin (Beta = -0.5315)**: Higher Albumin levels (for a one-unit increase) are linked to an approximately 0.5315 decrease in the serum bilirubin levels.
- **Hepatomegaly (Yes) (Beta = 0.4875)**: The presence of Hepatomegaly is associated with an approximately 0.4875 increase in the serum bilirubin levels.
- **Log-transformed Alkaline (Beta = 0.4222)**: Higher Log-transformed Alkaline levels (for a one-unit increase) correspond to an approximately 0.4222 increase in the serum bilirubin levels.
- **Histologic (Beta = 0.0911)**: An increase in the Histologic stage of the disease is related to a slight increase of approximately 0.0911 in the serum bilirubin levels.

Answer 1.j)

Patients who typically exhibit elevated levels of serum bilirubin, based on the final model (Model m14), are characterized by the following:

- **Higher Prothrombin Levels**: Patients with higher levels of Prothrombin tend to have elevated serum bilirubin levels. Prothrombin is positively associated with serum bilirubin.
- **Lower Albumin Levels**: Patients with lower Albumin levels are more likely to have elevated serum bilirubin levels. Albumin is negatively associated with serum bilirubin.
- **Presence of Hepatomegaly**: Patients with Hepatomegaly (enlarged liver) are more likely to exhibit elevated serum bilirubin levels. Hepatomegaly is associated with higher bilirubin levels.

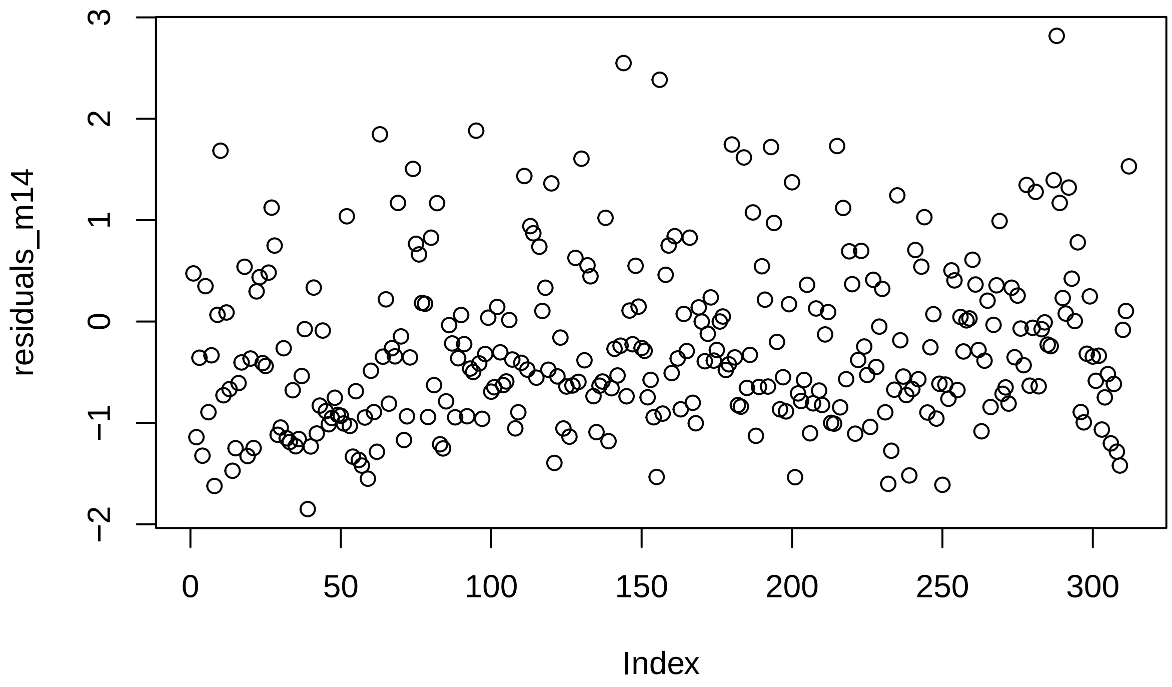
- **Higher Log-transformed Alkaline Levels**: Patients with higher Log-transformed Alkaline levels tend to have elevated serum bilirubin levels. Log-transformed Alkaline is positively associated with serum bilirubin.
- **Histologic Stage**: Patients with a higher Histologic stage of the disease may have slightly elevated serum bilirubin levels. The relationship is positive but relatively modest.

Answer 1.k)

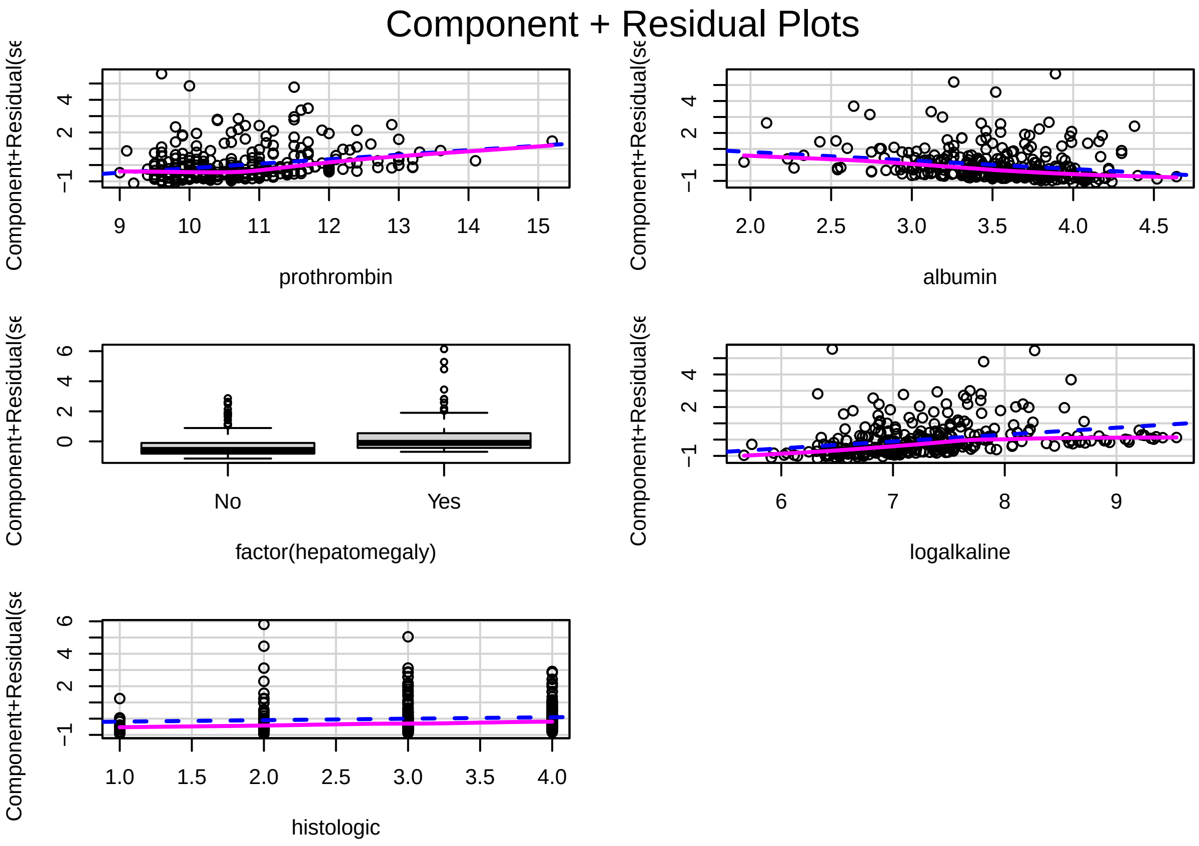
```
# Obtain the scaled deviance for Model m14
scaled_deviance_m14 <- summary(m14)$deviance / m14$df.residual
scaled_deviance_m14
```

```
## [1] 0.7424326
```

```
# Obtain the residuals for Model m14
residuals_m14 <- resid(m14)

# Plot residuals for Model m14
plot(residuals_m14)
```



```
# Partial residual plots
crPlots(m14)
```



- A value close to 1 indicates a good fit, while values significantly greater than 1 suggest overdispersion, and values less than 1 indicate underdispersion. In our model (m14), the scaled deviance of approximately 0.7424 suggests a reasonably good fit, indicating that the model adequately explains the variability in the data.
- Random scatter of residuals in the plot for Model m14 suggests a good fit with no significant issues in capturing relationships between predictors and the response variable.
- The partial residual plots of all predictors show a significant relationships with the response variable. There is no significant amount of deviation from the reference line. Also, component+residual(serBilir) vs factor(hepatomegaly) it suggests that a enlarged liver also increases the risk of increasing serBilir levels in the body.

Question 2

Answer 2.a)

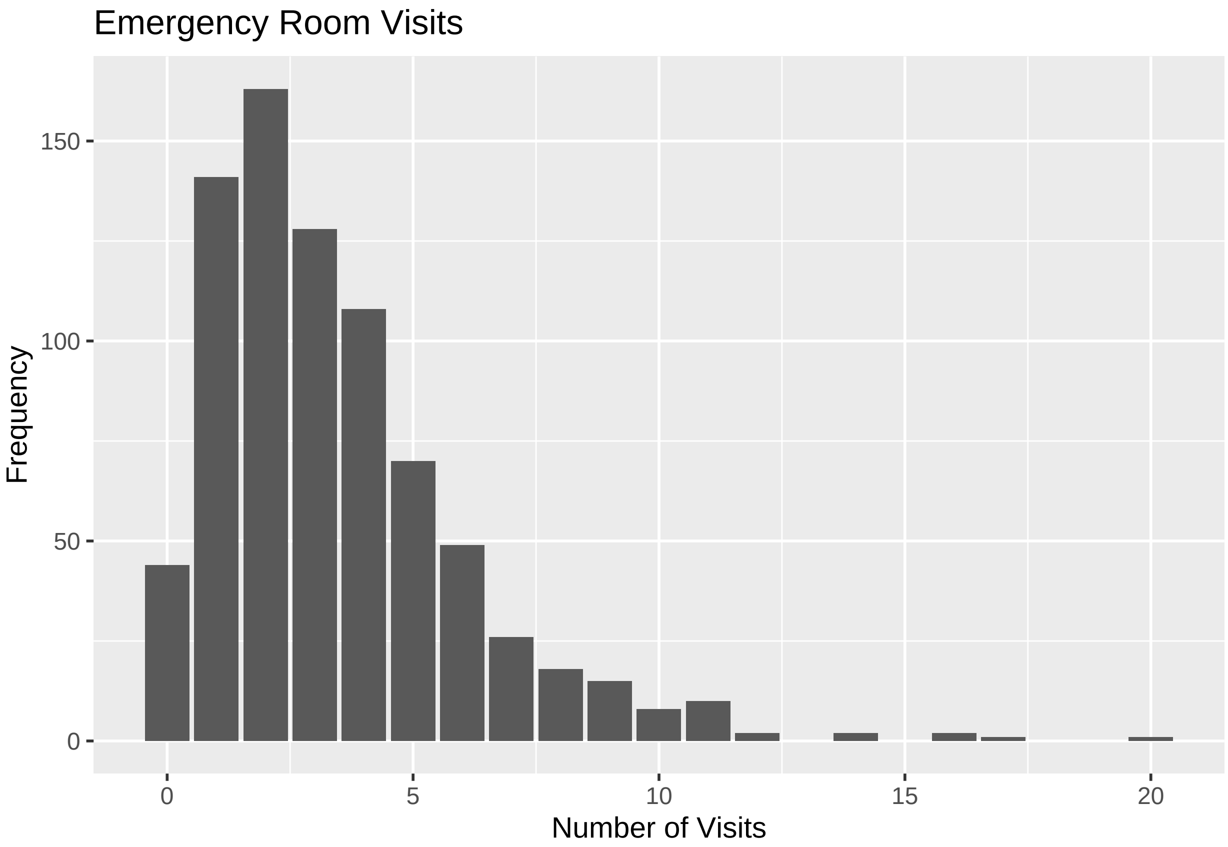
```
erData <- read_csv(here::here("Heart_Disease.csv"))
```

```
## Rows: 788 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (10): ID, Total_Cost, Age, Gender, Interventions, Drug, ER_visits, Compl...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(erData,10)
```

```
## # A tibble: 10 x 10
##       ID Total_Cost   Age Gender Interventions  Drug ER_visits Complications
##    <dbl>      <dbl> <dbl>  <dbl>         <dbl> <dbl>     <dbl>         <dbl>
##  1     1       179.    63      0             2     1         4             0
##  2     2       319     59      0             2     0         6             0
##  3     3      9311.    62      0            17     0         2             0
##  4     4       281.    60      1             9     0         7             0
##  5     5     18727.    55      0             5     2         7             0
##  6     6       453.    66      0             1     0         3             0
##  7     7       323.    64      1             2     0         3             0
##  8     8      3874.    45      1             3     0         5             0
##  9     9      3244.    68      0             6     2         5             0
## 10    10       226.    64      1             3     0         2             0
## # i 2 more variables: Comorbidities <dbl>, Duration <dbl>
```

```
# Create a bar chart for ER_visits
ggplot(erData, aes(x = ER_visits)) +
  geom_bar() +
  labs(title = "Emergency Room Visits") +
  xlab("Number of Visits") +
  ylab("Frequency")
```



Looks like the ER_visits is Right Skewed.

To check overdispersion in the datset we need to fit a poisson model and check the ratio of residual deviance and degrees of freedom, the rule of thumb says, the ratio should be ess than 1.10.

```
summary(poisson_model <- glm(ER_visits ~ Age, data = erData, family = poisson))
```

```
##
## Call:
## glm(formula = ER_visits ~ Age, family = poisson, data = erData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7150  -0.9497  -0.2824   0.6929   6.1356
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.804579   0.173198   4.645 3.39e-06 ***
## Age         0.007244   0.002915   2.485    0.013 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1485.0  on 787  degrees of freedom
## Residual deviance: 1478.7  on 786  degrees of freedom
## AIC: 3692.1
##
## Number of Fisher Scoring iterations: 5
```

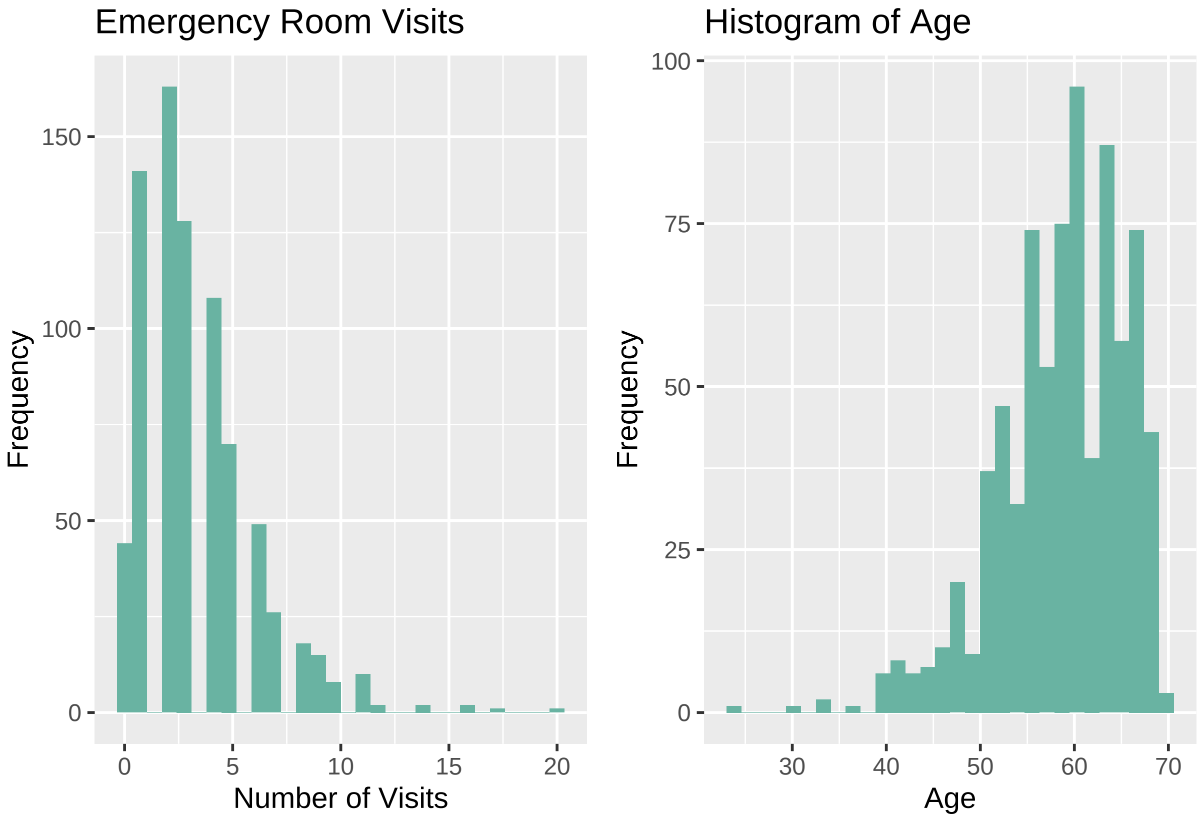
In this case, our residual deviance is 1478.7 for 786 degrees of freedom. The rule of thumb is that the ratio of deviance to df should be 1, but it is 1.88, indicating severe overdispersion.

Answer 1.b)

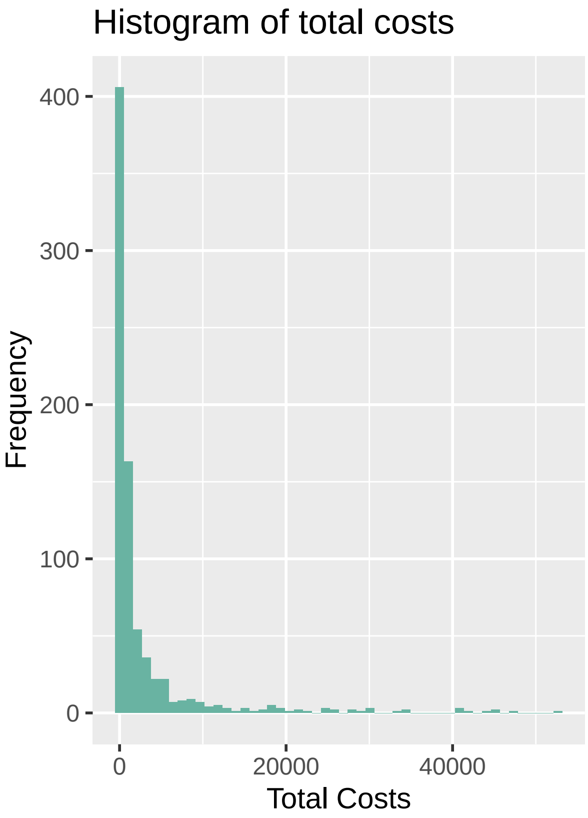
```
p1 <- ggplot(erData, aes(x = ER_visits)) +
  geom_histogram(fill="#69b3a2") +
  labs(title = "Emergency Room Visits") +
  xlab("Number of Visits") +
  ylab("Frequency")

p2 <- ggplot(erData, aes(x = Age)) +
  geom_histogram(fill="#69b3a2") +
  labs(title = "Histogram of Age") +
  xlab("Age") +
  ylab("Frequency")

p3 <- ggplot(erData, aes(x = Total_Cost)) +
  geom_histogram(bins = 50, fill="#69b3a2") +
  labs(title = "Histogram of total costs") +
  xlab("Total Costs") +
  ylab("Frequency")

p4 <- ggplot(erData, aes(x = Comorbidities)) +
  geom_histogram(bins = 20, fill="#69b3a2") +
  labs(title = "Histogram for Comorbidities") +
  xlab("Comorbidities") +
  ylab("Frequency")
```

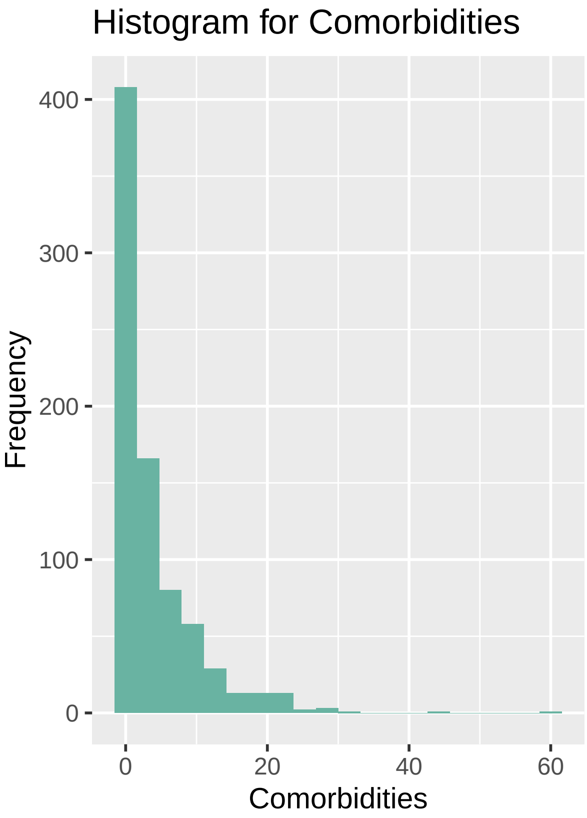
```
p1 + p2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
p3 + p4
```



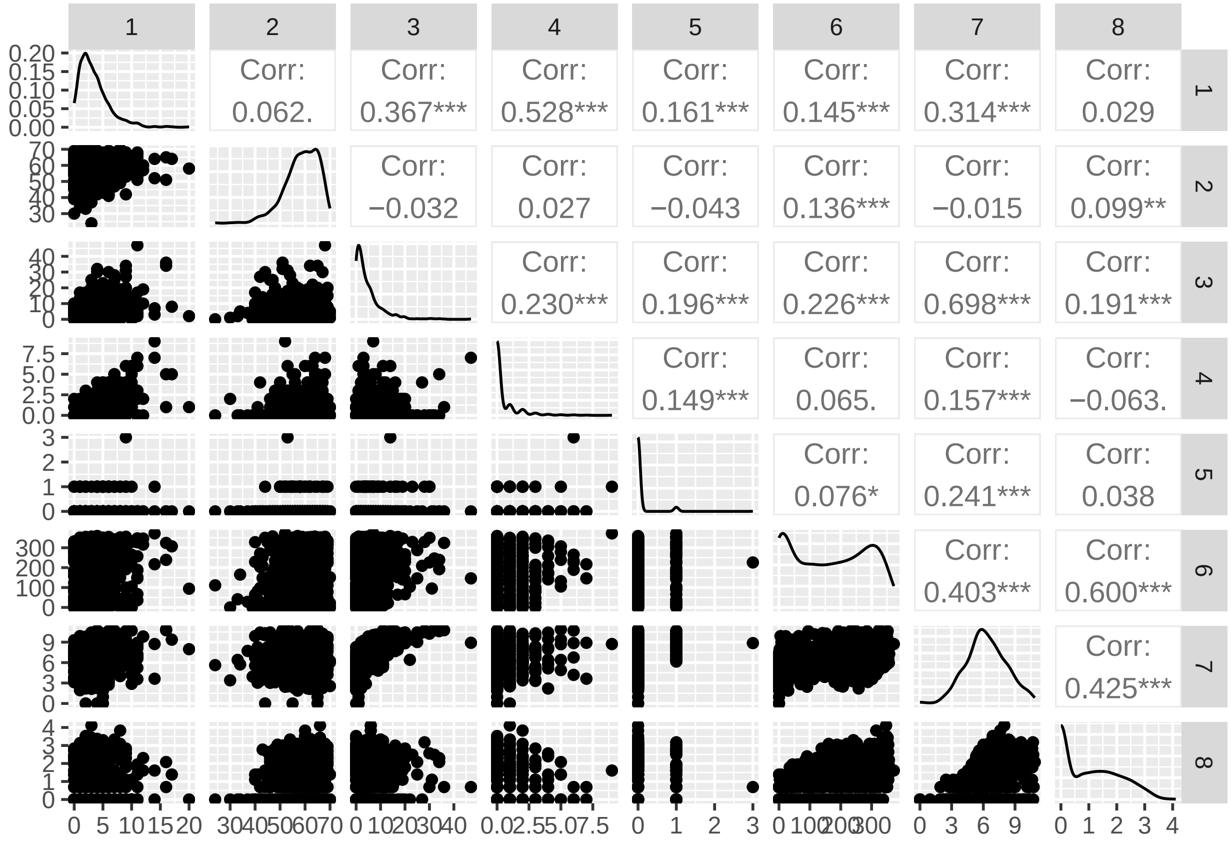


- In many cases, ER_visits and Age are count or continuous variables and don't typically require transformations within a count regression context like Poisson or Negative Binomial regression. While already treating them as counts (whole numbers), regression models can handle them directly.
- Taking the natural logarithm (ln) of Total_Cost and Comorbidities is a common transformation to make the data more symmetric and address issues related to heteroscedasticity.

```
# Take the natural logarithm of Total_Cost and Comorbidities
erData$logtotal_cost <- log(erData$Total_Cost + 1)
erData$logcomorbidities <- log(erData$Comorbidities + 1)
```

Answer 2.c)

```
# Calculate the correlation matrix
correlation_matrix <- cor(erData[, c("ER_visits", "Total_Cost", "Age", "Interventions", "Drug", "Complic

# Visualize the correlation matrix as a heatmap
ggpairs(erData, columns = c("ER_visits", "Age", "Interventions", "Drug", "Complications", "Duration", "
```



By referring to the cor-relation plot, we can tell that the covariates Interventions, Drug and logtotal_cost are likely to be the predictors of ER_visits because they have strong positive correlations w.r.t ER_visits.

- Interventions and Drug both have positive correlations with ER_visits. This suggests that as the number of interventions or the number of tracked drugs prescribed increases, the number of emergency room visits tends to increase.
- Higher total costs of claims tend to be associated with more emergency room visits.
- Complications and Duration also show a weak positive relationship with ER_visits.

Answer 2.d)

```
# Fit a Poisson regression model with Drug as the covariate
poisson_model <- glm(ER_visits ~ Interventions, data = erData, family = poisson)

# Summarize the model
summary(poisson_model)
```

```
##
## Call:
## glm(formula = ER_visits ~ Interventions, family = poisson, data = erData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8505  -1.2041  -0.2742   0.6036   6.4580
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.02940    0.02505   41.10   <2e-16 ***
## Interventions  0.03724    0.00255   14.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1485.0  on 787  degrees of freedom
## Residual deviance: 1310.5  on 786  degrees of freedom
## AIC: 3523.9
##
## Number of Fisher Scoring iterations: 5
```

```
# plot(simulateResiduals(poisson_model, n = 1000))
```

- Lets have a look at the ratio of Residual deviance and df, the ratio of Residual deviance and df of the poisson model is 1.667 which is larger than 1.10 and according to the rule of thumb anything above 1.10 is considered as overdispersion.
- Also the the simulated residual plot also tells us that there are issues while fitting the poisson model, there is Quantile deviation detected at 25% and 50% quantile lines.
- In poisson model we assume that the variance of the counts is equal to the mean, but here its different. Overdispersion can lead to underestimated standard errors and incorrect inferences.
- If there is a significant number of cases with zero emergency room visits, a Poisson model may not handle this well
- One alternative to the Poisson model that can address these issues is Negative Binomial regression.
- Negative Binomial regression accommodates overdispersion by allowing the variance to be greater than the mean. It introduces an additional parameter that captures the extra variability in the data.
- Negative Binomial regression can also handle excess zeros effectively.

Answer 2.e

```
# Load necessary libraries
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.2.3
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:patchwork':
##
##     area
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
# Fit Negative Binomial models for each covariate
m1 <- glm.nb(ER_visits ~ Age, data = erData)
m2 <- glm.nb(ER_visits ~ factor(Gender), data = erData)
m3 <- glm.nb(ER_visits ~ Interventions, data = erData)
m4 <- glm.nb(ER_visits ~ sqrt(Drug), data = erData)
m5 <- glm.nb(ER_visits ~ Complications, data = erData)
m6 <- glm.nb(ER_visits ~ sqrt(Comorbidities), data = erData)
m7 <- glm.nb(ER_visits ~ sqrt(Duration), data = erData)
m8 <- glm.nb(ER_visits ~ logtotal_cost, data = erData)
m9 <- glm.nb(ER_visits ~ logcomorbidities, data = erData)
```

```
summary(m3)
```

```
## 
## Call:
## glm.nb(formula = ER_visits ~ Interventions, data = erData, init.theta = 5.421463787, 
##     link = log)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.4684  -0.9127  -0.2170   0.4478   4.4384  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)   1.022054   0.032472   31.48   <2e-16 ***
## Interventions 0.038598   0.003841   10.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for Negative Binomial(5.4215) family taken to be 1)
## 
##     Null deviance: 925.02  on 787  degrees of freedom
## Residual deviance: 824.80  on 786  degrees of freedom
## AIC: 3404.4
## 
## Number of Fisher Scoring iterations: 1
## 
## 
##               Theta:  5.421 
##           Std. Err.:  0.714 
## 
##  2 x log-likelihood:  -3398.425
```

```
# Create an empty dataframe to store results
results_df <- data.frame(Model = character(0), P_values = numeric(0), AIC = numeric(0), LogLikelihood =

# List of model names
model_names <- c("m1", "m2", "m3", "m4", "m5", "m6", "m7", "m8", "m9")

# Iterate through the models and extract Pr(>|t|), AIC, and Log-likelihood
for (i in 1:length(model_names)) {
  model <- eval(parse(text = model_names[i]))  # Get the model by its name

  # Extract Pr(>|t|), AIC, and Log-likelihood and store in the dataframe
  log_likelihood <- -2 * logLik(model)  # Calculate log-likelihood
  results_df <- rbind(results_df, data.frame(Model = model_names[i], P_values = round(summary(model)$co
}

# Print the results dataframe
print(results_df)
```

```
##   Model P_values      AIC LogLikelihood
## 1    m1  0.00067 3494.270      3488.270
## 2    m2  0.00000 3487.684      3481.684
## 3    m3  0.00000 3404.425      3398.425
## 4    m4  0.00000 3295.303      3289.303
## 5    m5  0.00000 3480.276      3474.276
## 6    m6  0.00000 3496.954      3490.954
## 7    m7  0.00000 3475.486      3469.486
## 8    m8  0.00000 3415.898      3409.898
## 9    m9  0.00000 3497.025      3491.025
```

Answer2.f)

```
m10 <- glm.nb(ER_visits ~ Age + factor(Gender) + Interventions + Drug + Complications + Comorbidities +

m11<- glm.nb(ER_visits ~ Age + factor(Gender) + Interventions + Drug + Complications + Comorbidities +

m12 <- glm.nb(ER_visits ~ Age + factor(Gender) + Interventions + Drug + Complications + Comorbidities +

m13 <- glm.nb(ER_visits ~ Age + factor(Gender) + Interventions + Drug + Complications + Comorbidities,

m14 <- glm.nb(ER_visits ~ Age + factor(Gender) + Interventions + Drug + Complications, data = erData)

m15 <- glm.nb(ER_visits ~ Age + factor(Gender) + Interventions + Drug, data = erData)

m16 <- glm.nb(ER_visits ~ Age + Interventions + Drug, data = erData)

m17 <- glm.nb(ER_visits ~ Interventions + Drug, data = erData)


# Create an empty dataframe to store results
results_df <- data.frame(Model = character(0), P_values = numeric(0), AIC = numeric(0), LogLikelihood =

# List of model names
model_names <- c("m10", "m11", "m12", "m13", "m14", "m15", "m16", "m17")

# Iterate through the models and extract Pr(>|t|), AIC, and Log-likelihood
for (i in 1:length(model_names)) {
  model <- eval(parse(text = model_names[i]))  # Get the model by its name

  # Extract Pr(>|t|), AIC, and Log-likelihood and store in the dataframe
  log_likelihood <- -2 * logLik(model)  # Calculate log-likelihood
  results_df <- rbind(results_df, data.frame(Model = model_names[i], P_values = round(summary(model)$co
}

# Print the results dataframe
print(results_df)
```

```
##   Model P_values      AIC LogLikelihood
## 1   m10  0.56937 3241.560      3219.560
## 2   m11  0.54226 3242.453      3222.453
## 3   m12  0.02529 3253.001      3235.001
## 4   m13  0.02390 3252.916      3236.916
## 5   m14  0.02524 3251.270      3237.270
## 6   m15  0.01986 3250.845      3238.845
## 7   m16  0.01062 3259.943      3249.943
## 8   m17  0.00000 3262.168      3254.168
```

```
summary(m15)
```

```
##
## Call:
## glm.nb(formula = ER_visits ~ Age + factor(Gender) + Interventions +
##     Drug, data = erData, init.theta = 10.3866107, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1414  -0.9319  -0.2043   0.4887   4.4756
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     0.474173   0.203600   2.329 0.019862 *
## Age             0.007206   0.003402   2.118 0.034188 *
## factor(Gender)1 0.174000   0.051686   3.367 0.000761 ***
## Interventions   0.028515   0.003480   8.194 2.53e-16 ***
## Drug            0.216396   0.016606  13.031  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(10.3866) family taken to be 1)
##
##     Null deviance: 1117.34  on 787  degrees of freedom
## Residual deviance:  814.61  on 783  degrees of freedom
## AIC: 3250.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  10.39
##           Std. Err.:  1.98
##
##  2 x log-likelihood:  -3238.845
```

Answer 2.g)

We are going to select m15 as our final model because it is significant (20% significance value is satisfied) and is the smallest AIC value.

the Equation is as folows:

$\hat{ERvisits} = \beta_0 + \beta_1 \cdot Age + \beta_2 \cdot factor(Gender) + \beta_3 \cdot Interventions + \beta_4 \cdot Drug + \epsilon$

$$\hat{ERvisits} = 0.474 + 0.007 \cdot Age + 0.174 \cdot factor(Gender) + 0.028 \cdot Interventions + 0.216 \cdot Drug + \epsilon$$

Where: -$\hat{ERvisits}$ represents the number of emergency room visits.

-β_0 is the intercept.

-β_1 is the coefficient associated with the “Age” variable.

-β_2 is the coefficient associated with the “Gender” variable (“Gender” is coded as a factor with two levels, e.g., 0 and 1).

-β_3 is the coefficient associated with the “Interventions” variable.

-β_4 is the coefficient associated with the “Drug” variable.

Answer 2.h)

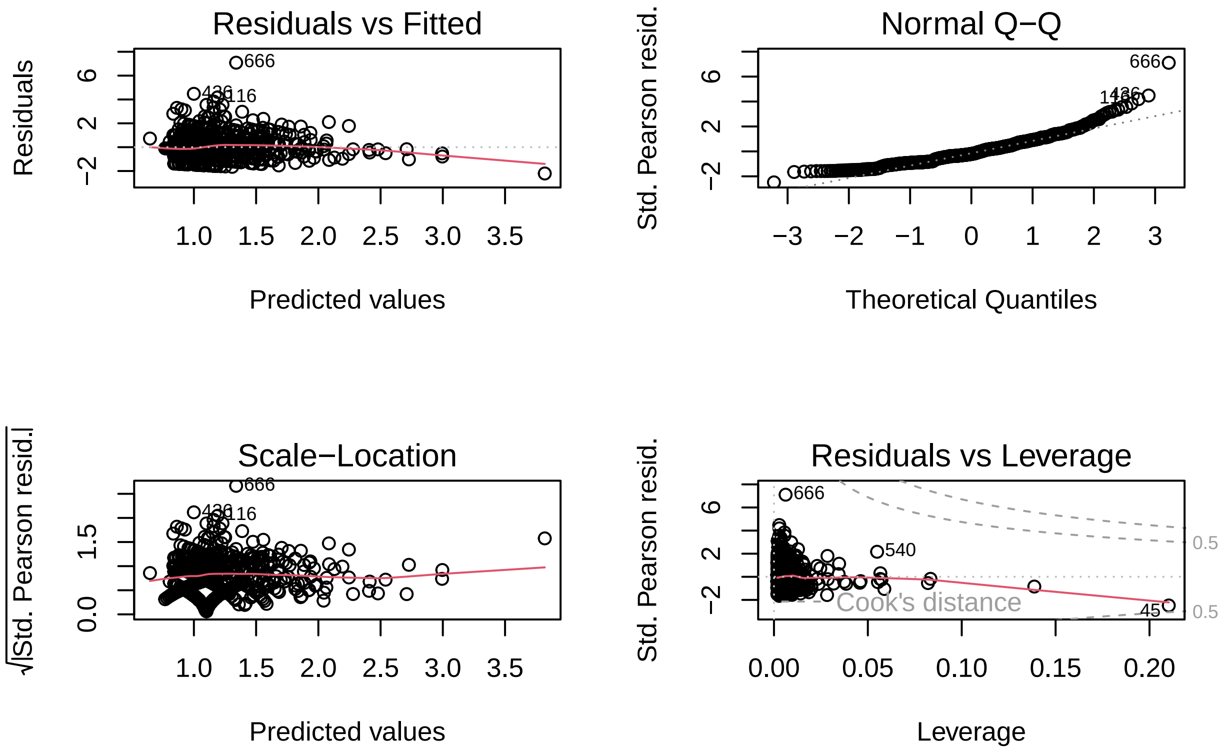
- **Intercept (0.474)**: The expected number of emergency room visits for a hypothetical subscriber with age, gender, interventions, and drug counts all at zero is 0.474.
- **Age (0.007)**: For each one-year increase in age, the expected number of emergency room visits increases by 0.007. Older subscribers tend to have slightly more visits on average.
- **Gender (0.174)**: Males (coded as 1) have a higher expected number of visits than females (coded as 0), with a difference of 0.174 visits on average.
- **Interventions (0.028)**: Each additional intervention or procedure is associated with an increase of 0.028 in the expected number of visits.
- **Drug (0.216)**: Each additional prescribed drug is linked to a 0.216 increase in the expected number of visits.

Answer 2.i)

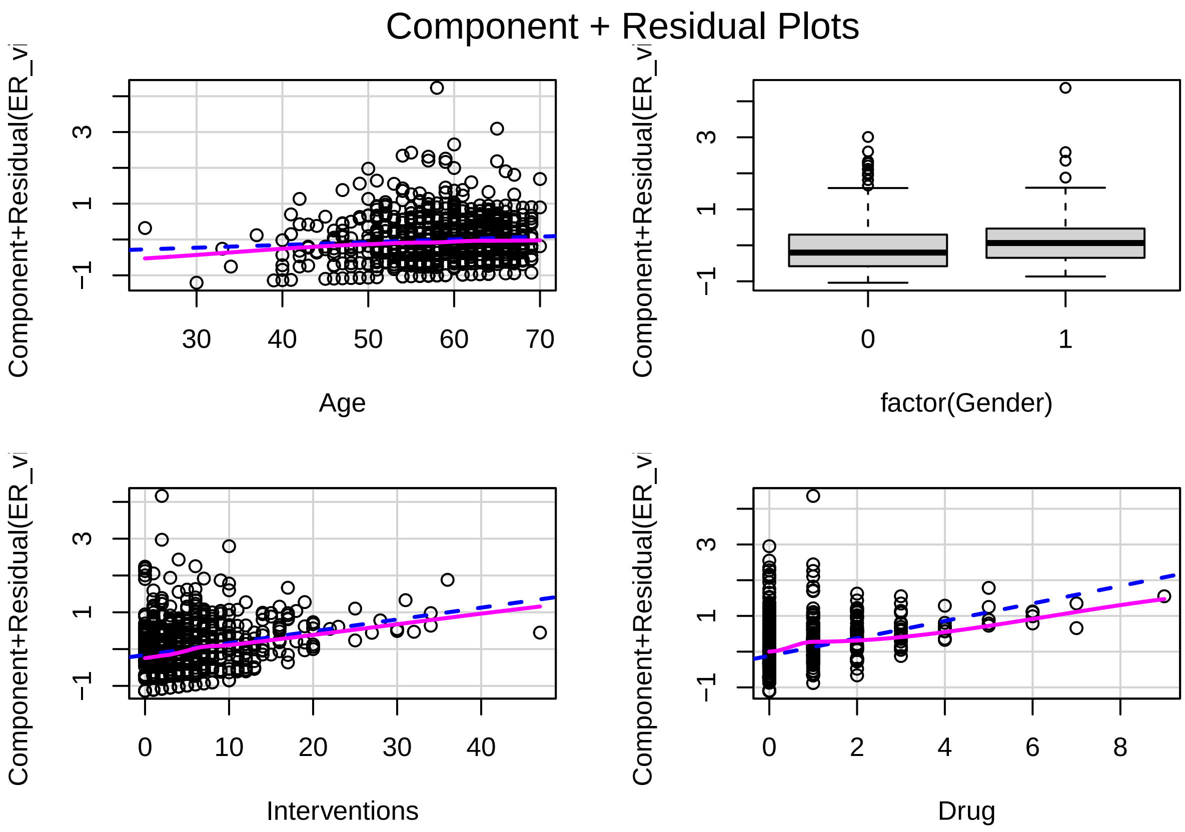
```r
# Fit the final model (m15)
final_model <- glm.nb(ER_visits ~ Age + factor(Gender) + Interventions + Drug, data = erData)

# Calculate the scaled deviance
scaled_deviance <- deviance(final_model) / df.residual(final_model)

# Print the scaled deviance
scaled_deviance
```

```
## [1] 1.040367
```

```r
par(mfrow = c(2,2))
plot(m15)
```



```
# Create partial residual plots for each predictor
crPlots(m15)
```

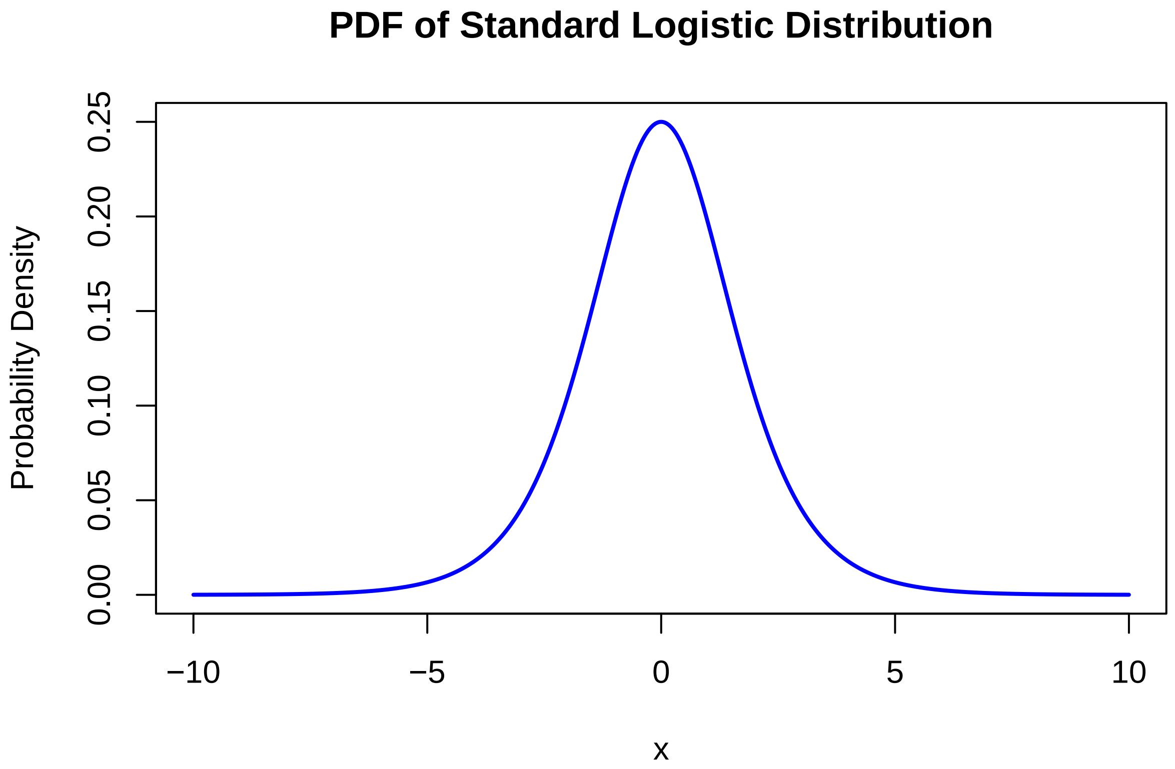


- The scaled deviance is close to 1 which suggests that the model is a good fit.

Question 3

Answer 3.a)

```
# Generate a range of values for the x-axis
x <- seq(-10, 10, by = 0.01)

# Calculate the pdf of the standard logistic distribution
pdf <- dlogis(x)

# Plot the pdf
plot(x, pdf, type = "l", col = "blue", lwd = 2,
     xlab = "x", ylab = "Probability Density",
     main = "PDF of Standard Logistic Distribution")
```



Answer 3.b)

To show that the standard logistic distribution is symmetric about 0, we need to demonstrate that its probability density function (pdf) is symmetric with respect to the vertical line at$x = 0$. In other words, we want to confirm that$f(x) = f(-x)$ for all values of$x$.

The pdf of the standard logistic distribution is given by:

$$f(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

Now, let's calculate$f(-x)$:

$$f(-x) = \frac{e^{x}}{(1 + e^{x})^2}$$

We can see that$f(-x)$ is the same as$f(x)$, except for the sign in the exponent. To demonstrate symmetry, we'll compare$f(x)$ and$f(-x)$ directly:

Let's compare these two expressions:

$$f(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

$$f(-x) = \frac{e^{x}}{(1 + e^{x})^2}$$

If we substitute$-x$ for$x$ in$f(x)$, we see that$f(x)$ and$f(-x)$ are indeed equal:

$$f(-x) = \frac{e^x}{(1+e^x)^2} = f(x)$$

This demonstrates that the pdf of the standard logistic distribution is symmetric about0, as$f(x) = f(-x)$ for all values of$x$. This symmetry is a characteristic of the standard logistic distribution.

Answer 3.c)

To show that$P(Y_i = 1|X_i = x_i, \beta) = \frac{1}{1+e^{-x_i^T \beta}}$, we can use the logistic model defined in part (1) of the question, where the relationship between the latent variableΨ_i andX_i is given by:

$$\Psi_i = x_i^T \beta + \epsilon_i$$

AndY_i is defined as:

$$Y_i = \begin{cases} 1, & \text{if } \Psi_i \geq 0 \\ 0, & \text{if } \Psi_i < 0 \end{cases}$$

We want to find$P(Y_i = 1|X_i = x_i, \beta)$, which is the probability that$Y_i = 1$ given$X_i = x_i$ and the model parametersβ. This is equivalent to finding the probability that$\Psi_i \geq 0$.

So, we need to find$P(\Psi_i \geq 0|X_i = x_i, \beta)$. Using the cumulative distribution function (CDF) of the logistic distribution, which was provided in the question as:

$$F_\epsilon(u) = \frac{1}{1+e^{-u}}$$

We can express the probability that$\Psi_i \geq 0$ as follows:

$$P(\Psi_i \geq 0|X_i = x_i, \beta) = 1 - P(\Psi_i < 0|X_i = x_i, \beta)$$

Now, we'll use the logistic model to expressΨ_i in terms ofx_i andβ:

$$\Psi_i = x_i^T \beta + \epsilon_i$$

To find$P(\Psi_i < 0|X_i = x_i, \beta)$, we subtract$\Psi_i$ from both sides of the inequality:

$$-x_i^T \beta \leq -\epsilon_i$$

Now, we'll apply the CDF of the logistic distribution to both sides of this inequality:

$$F_\epsilon(-x_i^T \beta) \leq F_\epsilon(-\epsilon_i)$$

Using the provided CDF formula:

$$\frac{1}{1+e^{x_i^T \beta}} \leq \frac{1}{1+e^{\epsilon_i}}$$

Now, subtracting both sides from 1 and simplifying:

$$1 - \frac{1}{1+e^{x_i^T \beta}} \geq 1 - \frac{1}{1+e^{\epsilon_i}}$$

This is equivalent to:

$$\frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}} \geq \frac{e^{\epsilon_i}}{1+e^{\epsilon_i}}$$

Now, we can see that$P(\Psi_i < 0|X_i = x_i, \beta)$ is the probability that the logistic distribution with parameter$x_i^T\beta$ is less than 0, which is:

$$P(\Psi_i < 0|X_i = x_i, \beta) = \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$$

Finally, using the complement rule for probabilities, we find$P(\Psi_i \geq 0|X_i = x_i, \beta)$:

$$P(\Psi_i \geq 0|X_i = x_i, \beta) = 1 - \frac{e^{x_i^T\beta}}{1+e^{x_i^T\beta}}$$

Simplifying the right side:

$$= \frac{1+e^{x_i^T\beta} - e^{x_i^T\beta}}{1+e^{x_i^T\beta}} = \frac{1}{1+e^{x_i^T\beta}}$$

This shows that$P(Y_i = 1|X_i = x_i, \beta) = \frac{1}{1+e^{x_i^T\beta}}$, which is the logistic function and represents the probability that$Y_i = 1$ given$X_i = x_i$ and the model parametersβ.

Answer 3.d)

Yes, based on the information provided in part (1) of the question, you can define a Generalized Linear Model (GLM). The GLM framework consists of three essential components:

1. **Linear Predictor (Systematic Component)**:

   $$\Psi_i = x_i^T\beta + \epsilon_i$$

   Here,Ψ_i is the linear predictor,x_i^T represents the transpose of the vector of covariatesX_i, andβ represents a vector of coefficients. This part represents the systematic component of the GLM, where we model the relationship between the covariates and the unobserved latent variableΨ_i using a linear model.

2. **Link Function**: The link function connects the linear predictor (Ψ_i) to the expected value of the response variable (Y_i). In this case, the link function is not explicitly mentioned, but based on the context, it appears that the link function is the logistic function (logit link). The logistic function is commonly used for binary response variables and is given as:

   $$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$$

   In this context,μ represents the probability ofY_i being 1, and the linear predictorΨ_i is related toμ through this link function.

3. **Probability Distribution**: The probability distribution for the response variableY_i is also not explicitly mentioned, but it's implied that it's a binary distribution, which can be modeled using the Bernoulli distribution. The relationship betweenΨ_i andY_i is defined as:

   $$Y_i = 1, \text{ if } \Psi_i \geq 0; \quad Y_i = 0, \text{ if } \Psi_i < 0$$

   This indicates that the response variableY_i follows a Bernoulli distribution, where the probability of success ($Y_i = 1$) is determined by the logistic function.

In summary, based on the given information, you can define a GLM with a logistic link function for modeling binary response data. The link function relates the linear predictorΨ_i to the probability ofY_i being 1, and the response variableY_i follows a Bernoulli distribution.

Answer 3.e)

In this case, when the noise componentϵ_i follows a standard normal distribution ($\epsilon_i \sim N(0,1)$), we can write down the expression for$P(Y_i = 1|X_i = x_i, \beta)$ as follows:

Recall the latent variable model from part (1):

$$\Psi_i = x_i^T \beta + \epsilon_i$$

And the definition ofY_i:

$$Y_i = \begin{cases} 1, & \text{if } \Psi_i \geq 0 \\ 0, & \text{if } \Psi_i < 0 \end{cases}$$

To find$P(Y_i = 1|X_i = x_i, \beta)$, we need to compute the probability thatΨ_i is greater than or equal to 0, given$X_i = x_i$ andβ. In other words, we want to find$P(\Psi_i \geq 0|X_i = x_i, \beta)$.

Sinceϵ_i is normally distributed with mean 0 and variance 1 ($\epsilon_i \sim N(0,1)$),ϵ_i itself is distributed according to the standard normal distribution.

Now, we can use the properties of normal distribution to express$P(\Psi_i \geq 0|X_i = x_i, \beta)$ as follows:

$$P(\Psi_i \geq 0|X_i = x_i, \beta) = P(x_i^T \beta + \epsilon_i \geq 0|X_i = x_i, \beta)$$

Sinceϵ_i follows a standard normal distribution,$x_i^T \beta + \epsilon_i$ follows a normal distribution with mean$x_i^T \beta$ and variance 1. Therefore, we can standardize this distribution by subtracting the mean and dividing by the standard deviation:

$$P\left(\frac{x_i^T \beta + \epsilon_i - (x_i^T \beta)}{1} \geq \frac{0 - (x_i^T \beta)}{1}\right) = P(\epsilon_i \geq -x_i^T \beta)$$

Now, we can use the cumulative distribution function (CDF) of the standard normal distribution to find this probability:

$$P(\epsilon_i \geq -x_i^T \beta) = 1 - P(\epsilon_i < -x_i^T \beta)$$

Finally, we can find the probability$P(Y_i = 1|X_i = x_i, \beta)$ by simplifying the right-hand side:

$$P(Y_i = 1|X_i = x_i, \beta) = 1 - \Phi(-x_i^T \beta)$$

Where$\Phi(\cdot)$ is the CDF of the standard normal distribution.

So, the expression for$P(Y_i = 1|X_i = x_i, \beta)$ whenϵ_i follows a standard normal distribution is:

$$P(Y_i = 1|X_i = x_i, \beta) = 1 - \Phi(-x_i^T \beta)$$

This represents the probability ofY_i being 1 given$X_i = x_i$ andβ under the assumption of a standard normal distribution for the noise componentϵ_i.

Answer 3.f)

Yes, based on the information provided in part (e) of the question, we can identify a Generalized Linear Model (GLM). Let's summarize the key components:

1. **Linear Predictor (Systematic Component)**: $\Psi_i = x_i^T \beta + \epsilon_i$

   Here,Ψ_i is the linear predictor,x_i^T represents the transpose of the vector of covariatesX_i, andβ represents a vector of coefficients. This part represents the systematic component of the GLM, where we model the relationship between the covariates and the unobserved latent variableΨ_i using a linear model.

2. **Link Function**: The link function connects the linear predictor (Ψ_i) to the expected value of the response variable (Y_i). In this case, the link function is not explicitly mentioned, but based on the context and the standard normal distribution assumption forϵ_i, we can infer that the link function is the logistic function (logit link). The logistic function is commonly used for binary response variables and is given as:

   $$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$$

   In this context,μ represents the probability ofY_i being 1, and the linear predictorΨ_i is related toμ through this link function.

3. **Probability Distribution**: The probability distribution for the response variableY_i is not explicitly mentioned, but we can infer that it follows a Bernoulli distribution. The relationship betweenΨ_i andY_i is defined as:

   $$Y_i = 1, \text{ if } \Psi_i \geq 0; \quad Y_i = 0, \text{ if } \Psi_i < 0$$

   This indicates that the response variableY_i follows a Bernoulli distribution with parameterμ, whereμ is determined by the logistic function.

In summary, based on the given information in part (e), we can identify a Generalized Linear Model (GLM) with a logistic link function (logit link) for modeling binary response data. The link function relates the linear predictorΨ_i to the probability ofY_i being 1, and the response variableY_i follows a Bernoulli distribution. This is a common setup for modeling binary outcomes in GLMs.