# MTH303 Coursework

Student ID:2143444

## Part A

### 1.1

### Coding:

```
## Task 1.1: Histogram and transformation for LOS

# Read data
readmission <- read.csv("readmission.csv")

# Histogram of LOS
hist(readmission$LOS,
     breaks = 15,
     main   = "Histogram of LOS",
     xlab   = "Length of stay (days)")

# Define transformed response
readmission$log_LOS <- log(readmission$LOS)

# Histogram of log(LOS)
hist(readmission$log_LOS,
     breaks = 15,
     main   = "Histogram of log(LOS)",
     xlab   = "log(Length of stay)")
     install.packages("e1071")
library(e1071)

skewness(readmission$LOS)
skewness(readmission$log_LOS)
```

### Plots:

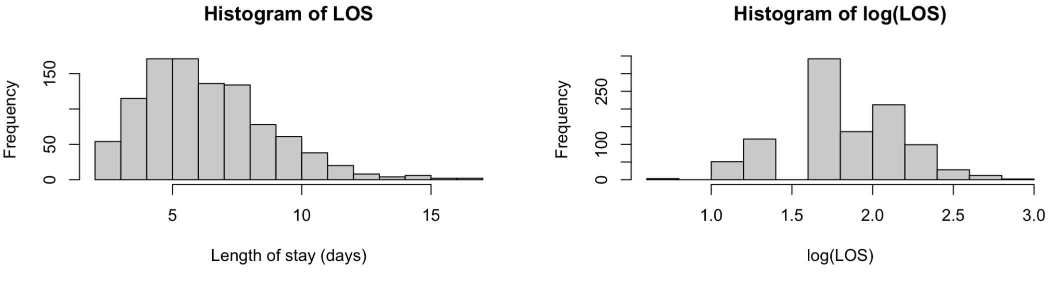

Figure 1: Histograms of LOS and log(LOS)

**Justification / Arguments:**

I first plotted a histogram of the original LOS variable (length of stay in days), as shown in Figure 1. LOS takes integer values between 2 and 17 days, with most patients staying in hospital for about 4 to 8 days and a few patients staying much longer. The distribution is clearly right-skewed: there is a long right tail with relatively few long stays, and the sample skewness is approximately 0.8. Because LOS is strictly positive and moderately right-skewed, a log transformation is appropriate to stabilise the variance and make the distribution closer to normal. Therefore, I created a transformed response variable `log_LOS` = log(`LOS`). The histogram of `log_LOS` in Figure 1 is much more symmetric, with sample skewness close to zero (around −0.15), so I will use `log_LOS` as the response variable in all subsequent modelling in Part A.

```
> skewness(readmission$LOS)
[1] 0.8099948
> skewness(readmission$log_LOS)
[1] -0.1539507
```

### 1.2

**Coding:**

```
## Task 1.2: Plot transformed LOS by a categorical predictor

# Use DRG.Class as the categorical predictor and draw boxplots
boxplot(log_LOS ~ DRG.Class,
        data = readmission,
        main = "Boxplot of log(LOS) by DRG.Class",
        xlab = "DRG.Class",
        ylab = "log(Length of stay)")

# Group means of log(LOS) by DRG.Class
tapply(readmission$log_LOS, readmission$DRG.Class, mean)
```

**Plots:**

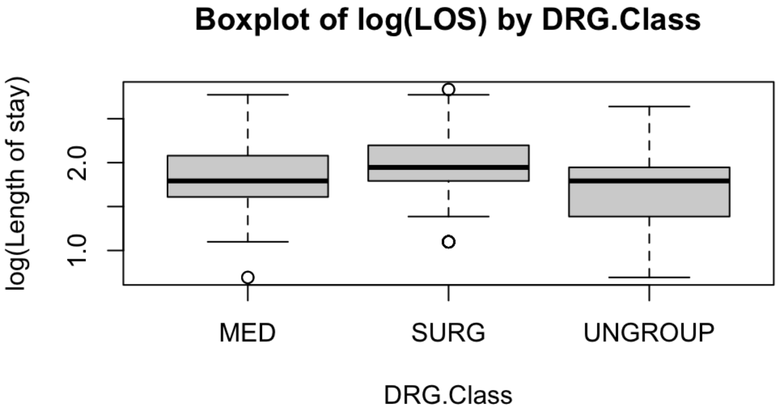

Figure 2: Boxplot of log(LOS) by DRG.Class

**Justification / Arguments:**

I chose `DRG.Class` as a categorical predictor and plotted a boxplot of `log_LOS` by `DRG.Class`, as shown in Figure 2. The boxplot shows clear differences in the median and spread of `log_LOS` across diagnostic groups, and some classes have noticeably longer stays and more variability than others. This suggests that `DRG.Class` is an important factor for explaining variation in the transformed length of stay, so it is reasonable to include `DRG.Class` as a categorical predictor in the subsequent regression models.

## 2.1

**Coding:**

```
## Task 2.1: Baseline multiple linear regression model m0

# Fit baseline model: log_LOS as response, others as predictors
m0 <- lm(log_LOS ~ Age + ER + HCC.Riskscore +
           Gender + Race + DRG.Class + DRG.Complication,
         data = readmission)

# Model summary
summary(m0)
```

**Justification / Arguments:**

In Task 2.1 I fitted a baseline multiple linear regression model for the transformed response `log_LOS`. The model includes all available predictors as main effects: Age, ER, HCC.Riskscore, Gender, Race, DRG.Class and DRG.Complication, using the `lm()` function in R and naming the fitted object `m0`. The `summary(m0)` output gives the estimated coefficients and overall fit measures such as $R^2$ and the residual standard error.

```
    Call:
lm(formula = log_LOS ~ Age + ER + HCC.Riskscore + Gender + Race + 
    DRG.Class + DRG.Complication, data = readmission)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.85977 -0.14377  0.01904  0.15131  0.58795 

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 1.3646151  0.0552695  24.690  < 2e-16 ***
Age                         0.0022755  0.0006089   3.737 0.000197 ***
ER                         -0.0007695  0.0087181  -0.088 0.929687    
HCC.Riskscore               0.2750984  0.0090378  30.439  < 2e-16 ***
GenderM                     0.0071409  0.0140391   0.509 0.611116    
RaceHispanic                0.0238210  0.0273524   0.871 0.384025    
RaceOthers                  0.0261312  0.0309304   0.845 0.398407    
RaceWhite                   0.0091821  0.0199600   0.460 0.645598    
DRG.ClassSURG               0.0496636  0.0350064   1.419 0.156302    
DRG.ClassUNGROUP           -0.1628789  0.0275548  -5.911 4.68e-09 ***
DRG.ComplicationMedicalNoC -0.1930594  0.0217614  -8.872  < 2e-16 ***
DRG.ComplicationOther      -0.1009406  0.0334560  -3.017 0.002617 **
DRG.ComplicationSurgMCC.CC  0.0574757  0.0391219   1.469 0.142112
DRG.ComplicationSurgNoC    -0.1012829  0.0383826  -2.639 0.008452 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2198 on 986 degrees of freedom
Multiple R-squared:  0.6399, Adjusted R-squared:  0.6351
F-statistic: 134.8 on 13 and 986 DF,  p-value: < 2.2e-16
```

### 2.2

**Coding:**

```
## Task 2.2: Reduced model based on 5% significance level

# Based on summary(m0), remove ER, Gender, Race
m_red <- lm(log_LOS ~ Age + HCC.Riskscore +
              DRG.Class + DRG.Complication,
            data = readmission)

# Summary of reduced model
summary(m_red)

# Compare the two models (nested F-test)
anova(m_red, m0)
```

**Justification / Arguments:**

Based on the coefficient table of the baseline model `m0`, I removed predictors that were clearly not significant at the 5% level. In particular, ER, Gender and all Race dummies had large p-values, while Age, HCC.Riskscore, DRG.Class and DRG.Complication showed clear evidence of association with `log_LOS`. I therefore refitted a reduced model `m_red` with only these four predictors. When I compare `m_red` with `m0`, the residual standard error and adjusted $R^2$ are almost the same (the adjusted $R^2$ for `m_red` is only slightly higher), and the F-test from `anova(m_red, m0)` is not significant. This means that the overall goodness-of-fit is not meaningfully improved by keeping ER, Gender and Race, so `m_red` achieves essentially the same fit as `m0` but with fewer predictors and a simpler interpretation.

```
> summary(m_red)

Call:
lm(formula = log_LOS ~ Age + HCC.Riskscore + DRG.Class + DRG.Complication,
    data = readmission)

Residuals:
     Min       1Q   Median       3Q      Max
-0.86437 -0.14805  0.01997  0.15244  0.58257
```

```
Coefficients:
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 1.3812964  0.0499020  27.680  < 2e-16 ***
Age                         0.0022317  0.0006044   3.693 0.000234 ***
HCC.Riskscore               0.2748734  0.0089810  30.606  < 2e-16 ***
DRG.ClassSURG               0.0489407  0.0347682   1.408 0.159555    
DRG.ClassUNGROUP           -0.1630129  0.0274323  -5.942 3.89e-09 ***
DRG.ComplicationMedicalNoC -0.1924350  0.0216665  -8.882  < 2e-16 ***
DRG.ComplicationOther      -0.1008555  0.0332695  -3.031 0.002497 ** 
DRG.ComplicationSurgMCC.CC  0.0595320  0.0387660   1.536 0.124937    
DRG.ComplicationSurgNoC    -0.1009040  0.0382095  -2.641 0.008401 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2194 on 991 degrees of freedom
Multiple R-squared:  0.6393, Adjusted R-squared:  0.6364 
F-statistic: 219.6 on 8 and 991 DF,  p-value: < 2.2e-16

> # Compare the two models (nested F-test)
> anova(m_red, m0)
Analysis of Variance Table

Model 1: log_LOS ~ Age + HCC.Riskscore + DRG.Class + DRG.Complication
Model 2: log_LOS ~ Age + ER + HCC.Riskscore + Gender + Race + DRG.Class + 
    DRG.Complication
  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1    991 47.719                           
2    986 47.649  5  0.069831 0.289 0.9192
```

### 2.3

**Coding:**

```
## Task 2.3: Diagnostic plots for the reduced model m_red

# Standard diagnostic plots
par(mfrow = c(2, 2))
plot(m_red)
par(mfrow = c(1, 1))
```

**Plots:**

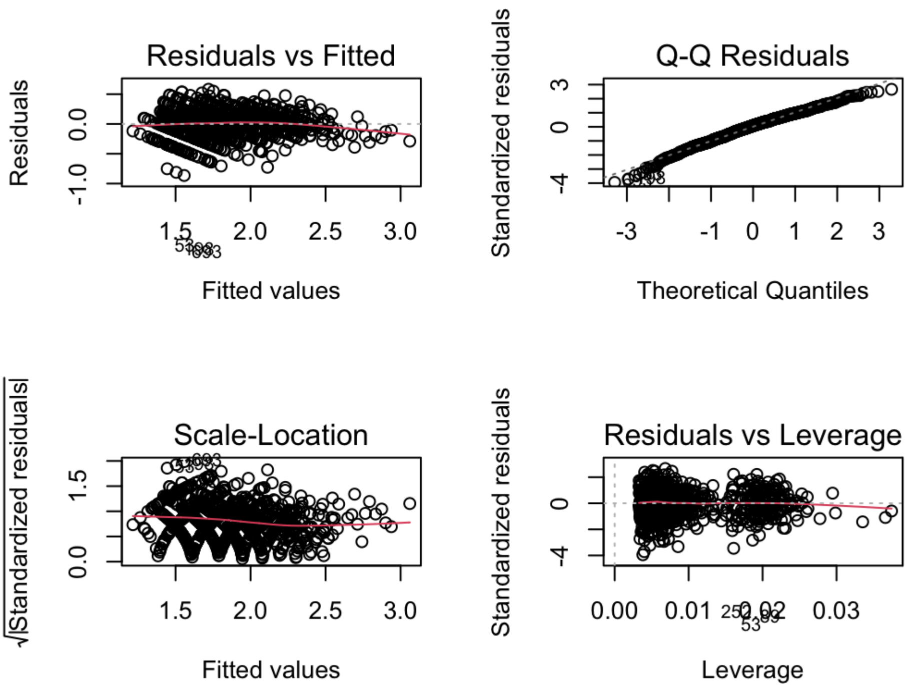

Figure 3: Standard diagnostic plots for the reduced model `m_red`.

**Justification / Arguments:**

The four standard diagnostic plots for the reduced model `m_red`, shown in Figure 3, do not show any serious problems. In the residuals versus fitted plot, the residuals are roughly centred around zero and there is no strong curved pattern, so a linear relationship between the predictors and `log_LOS` seems reasonable. The Normal Q–Q plot shows that most points lie close to the straight line, with only small deviations in the tails, which suggests that the normality assumption for the errors is acceptable. The Scale–Location plot indicates that the spread of the residuals is fairly constant across the fitted values, so there is no clear evidence of strong heteroscedasticity. Finally, the residuals versus leverage plot shows a few observations with slightly higher leverage, but their Cook's distances are small, so no single observation appears to have an excessive influence on the model fit. Overall, these diagnostics suggest that the basic regression assumptions for `m_red` are reasonably satisfied, and the model provides an adequate description of the transformed length of stay.

## 2.4

**Coding:**

```
## Task 2.4: Detect unusual observations for m_red

# Standardised residuals
r_std <- rstandard(m_red)

# Leverage (hat values)
h <- hatvalues(m_red)
```

```
# Cook’s distance
cook <- cooks.distance(m_red)

# Sample size n and number of predictors p (excluding intercept)
n <- nobs(m_red)
p <- length(coef(m_red)) - 1

# Average leverage: hbar = (p + 1) / n  (same as in the handout)
hbar <- (p + 1) / n

## Benchmarks (following the LRM R programming notes)

# (a) Outliers: |standardised residual| > 3
out_idx <- which(abs(r_std) > 3)

# (b) High leverage: h_i > 4 * hbar
lev_threshold <- 4 * hbar
lev_idx <- which(h > lev_threshold)

# (c) Influential points: Cook’s distance > 4 / (n - p - 1)
cook_threshold <- 4 / (n - p - 1)
inf_idx <- which(cook > cook_threshold)

# Inspect indices
out_idx        # potential outliers
lev_idx        # high leverage points
inf_idx        # influential points
```

**Justification / Arguments:**

I used the reduced model m_red to check for unusual observations, following the rules in the linear regression programming notes. First, I computed standardised residuals and treated any case with $|r_i| > 3$ as a potential outlier. This gave a small number of observations with much larger residuals than the rest of the data. Second, I calculated the leverage values $h_i$ and compared them with the average leverage $\bar{h} = (p + 1)/n$. Using the rule $h_i > 4\bar{h}$, I found a few high-leverage points that have more extreme combinations of predictor values.

Finally, I looked at Cook's distance to assess influence. With the benchmark CookD $> 4/(n - p - 1)$, several observations were flagged as influential, meaning that they have a noticeable joint effect of large residual and high leverage. Some cases are both outliers and influential, while others only have high leverage. Although there are a few observations that deserve closer inspection, their Cook's distances are still well below 1, so no single case completely dominates the fit of m_red.

```
> out_idx        # potential outliers
 53 198 642 693 995
 53 198 642 693 995
> lev_idx        # high leverage points
 77 224
 77 224
```

```
> inf_idx        # influential points
 53  89 119 126 198 206 217 221 224 244 252 256 280 309 385 411 441 442 452 459 497
 53  89 119 126 198 206 217 221 224 244 252 256 280 309 385 411 441 442 452 459 497
526 553 579 625 642 656 658 666 677 693 694 698 710 714 716 740 741 745 746 758 766
526 553 579 625 642 656 658 666 677 693 694 698 710 714 716 740 741 745 746 758 766
865 890 965 995
865 890 965 995
```

### 2.5

**Coding:**

```
## Task 2.5: Assess multicollinearity for m_red using VIF

# Need car package for VIF

library(car)

vif(m_red)
```

**Justification / Arguments:**

I used variance inflation factors (VIFs) to check multicollinearity in the reduced model `m_red`. For the two continuous predictors (Age and HCC.Riskscore), the VIF values are close to 1, which means they are almost uncorrelated with the other predictors. For the factor variables `DRG.Class` and `DRG.Complication`, I looked at the adjusted measures $\text{GVIF}^{1/(2\cdot\text{Df})}$, which are also only slightly above 1 and well below the common cut-off of 5. This indicates that there is no serious multicollinearity in the model.

In general, strong multicollinearity would make it difficult to interpret individual coefficients and would inflate their standard errors, so some variables could appear non-significant even if they are important. In this dataset the VIFs are small, so this problem should be limited: the estimated coefficients and their standard errors should be reasonably stable, and the main conclusions from the regression are unlikely to change dramatically due to collinearity.

```
> vif(m_red)
                     GVIF Df GVIF^(1/(2*Df))
Age              1.001285  1        1.000642
HCC.Riskscore    1.228184  1        1.108235
DRG.Class        5.610205  2        1.539022
DRG.Complication 6.877287  4        1.272557
```

### 2.6

**Coding:**

```
## Task 2.6: Stepwise selection from the baseline model m0
# Stepwise selection using AIC (both directions)
m_sel <- step(m0,
              direction = "both",  # can also be "backward"
              k = 2)              # k = 2 corresponds to AIC

# Summary of selected model
summary(m_sel)

# Compare with reduced model m_red from Task 2.2
summary(m_red)

AIC(m0, m_red, m_sel)
```

**Justification / Arguments:**

In Task 2.6 I used automatic model selection based on AIC. Starting from the full model m0, I ran step(m0, direction = "both") so that the procedure could either drop or add terms at each step. At each move, the function compares nearby models and keeps the one with the lowest AIC, where a lower AIC means a better trade-off between goodness of fit and model complexity. The final selected model m_sel keeps Age, HCC.Riskscore, DRG.Class and DRG.Complication, and drops ER, Gender and Race. This is exactly the same set of predictors as in the manually reduced model m_red, and their summaries and AIC values are almost identical. This suggests that both the data-driven AIC procedure and the simpler significance-based approach lead to the same parsimonious model for log_LOS.

```
> summary(m_sel)

Call:
lm(formula = log_LOS ~ Age + HCC.Riskscore + DRG.Class + DRG.Complication,
    data = readmission)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.86437 -0.14805  0.01997  0.15244  0.58257 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)                1.3812964  0.0499020  27.680  < 2e-16 ***
Age                        0.0022317  0.0006044   3.693 0.000234 ***
HCC.Riskscore              0.2748734  0.0089810  30.606  < 2e-16 ***
DRG.ClassSURG              0.0489407  0.0347682   1.408 0.159555    
DRG.ClassUNGROUP          -0.1630129  0.0274323  -5.942 3.89e-09 ***
DRG.ComplicationMedicalNoC -0.1924350  0.0216665  -8.882  < 2e-16 ***
DRG.ComplicationOther     -0.1008555  0.0332695  -3.031 0.002497 ** 
DRG.ComplicationSurgMCC.CC  0.0595320  0.0387660   1.536 0.124937    
DRG.ComplicationSurgNoC   -0.1009040  0.0382095  -2.641 0.008401 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2194 on 991 degrees of freedom
Multiple R-squared:  0.6393, Adjusted R-squared:  0.6364 
F-statistic: 219.6 on 8 and 991 DF,  p-value: < 2.2e-16

> # Compare with reduced model m_red from Task 2.2
> summary(m_red)

Call:
lm(formula = log_LOS ~ Age + HCC.Riskscore + DRG.Class + DRG.Complication,
    data = readmission)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.86437 -0.14805  0.01997  0.15244  0.58257 

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                 1.3812964  0.0499020  27.680  < 2e-16 ***
Age                         0.0022317  0.0006044   3.693 0.000234 ***
HCC.Riskscore               0.2748734  0.0089810  30.606  < 2e-16 ***
DRG.ClassSURG               0.0489407  0.0347682   1.408 0.159555    
DRG.ClassUNGROUP           -0.1630129  0.0274323  -5.942 3.89e-09 ***
DRG.ComplicationMedicalNoC -0.1924350  0.0216665  -8.882  < 2e-16 ***
DRG.ComplicationOther      -0.1008555  0.0332695  -3.031 0.002497 ** 
DRG.ComplicationSurgMCC.CC  0.0595320  0.0387660   1.536 0.124937    
DRG.ComplicationSurgNoC    -0.1009040  0.0382095  -2.641 0.008401 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2194 on 991 degrees of freedom
Multiple R-squared:  0.6393,	Adjusted R-squared:  0.6364 
F-statistic: 219.6 on 8 and 991 DF,  p-value: < 2.2e-16

> AIC(m0, m_red, m_sel)
      df       AIC
m0    15 -176.0215
m_red 10 -184.5570
m_sel 10 -184.5570
```

## Part B

### 3.1

#### Coding:

```
Task 3.1: Baseline logistic GLM for readmission

# Proportion of readmissions
prop.table(table(readmission$Readmission.Status))

# Fit baseline logistic regression model g0
g0 <- glm(Readmission.Status ~ Age + ER + HCC.Riskscore +
            Gender + Race + DRG.Class + DRG.Complication,
          family = binomial(link = "logit"),
          data   = readmission)
```

```
# Model summary
summary(g0)
```

### 3.2

**Coding:**

```
Task 3.2: Reduced logistic model based on 10% level

# From summary(g0), variables not significant at 10%:
# Age, Gender, Race
# Remove them and fit reduced model drop_g
drop_g <- glm(Readmission.Status ~ ER + HCC.Riskscore +
                DRG.Class + DRG.Complication,
              family = binomial(link = "logit"),
              data   = readmission)

# Summary of reduced model
summary(drop_g)

# Likelihood ratio test comparing drop_g and g0 (nested models)
anova(drop_g, g0, test = "Chisq")

# Compare AIC
AIC(g0, drop_g)
```

**Justification / Arguments:**

In Task 3.2 I simplified the baseline logistic model by using the p-values from summary(g0) with a 10% significance level. In the full model g0, HCC.Riskscore, ER, DRG.Class and several levels of DRG.Complication have relatively small p-values, while Age, Gender and all Race dummies have large p-values and look unimportant in this specification. Treating each factor as a whole, I removed Age, Gender and Race and refitted a reduced model drop_g with the formula Readmission.Status   ER + HCC.Riskscore + DRG.Class + DRG.Complication.

When I compare the goodness-of-fit between g0 and drop_g, the residual deviance and AIC in the two summaries are very similar, with the reduced model slightly preferred. This means that keeping Age, Gender and Race does not meaningfully improve the fit, so drop_g achieves essentially the same goodness-of-fit as g0 but with fewer predictors and a simpler interpretation.

```
> summary(g0)
Call:
glm(formula = Readmission.Status ~ Age + ER + HCC.Riskscore +
    Gender + Race + DRG.Class + DRG.Complication, family = binomial(link = "logit"),
    data = readmission)

Coefficients:
```

```
                           Estimate Std. Error z value Pr(>|z|)    
(Intercept)                -1.40492    0.53853  -2.609 0.009086 ** 
Age                         0.00502    0.00592   0.848 0.396428    
ER                          0.15010    0.08506   1.765 0.077627 .  
HCC.Riskscore               0.49300    0.09069   5.436 5.44e-08 ***
GenderM                    -0.18710    0.13750  -1.361 0.173578    
RaceHispanic               -0.02098    0.26870  -0.078 0.937771    
RaceOthers                 -0.29816    0.30825  -0.967 0.333411    
RaceWhite                   0.10117    0.19513   0.518 0.604115    
DRG.ClassSURG               0.78929    0.33853   2.332 0.019723 *  
DRG.ClassUNGROUP            0.96664    0.26881   3.596 0.000323 ***
DRG.ComplicationMedicalNoC -0.75042    0.21048  -3.565 0.000363 ***
DRG.ComplicationOther      -0.67090    0.33079  -2.028 0.042544 *  
DRG.ComplicationSurgMCC.CC -0.23305    0.37846  -0.616 0.538033    
DRG.ComplicationSurgNoC    -0.62694    0.36778  -1.705 0.088260 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1361.9  on 999  degrees of freedom
Residual deviance: 1243.1  on 986  degrees of freedom
AIC: 1271.1

Number of Fisher Scoring iterations: 4

> summary(drop_g)
Call:
glm(formula = Readmission.Status ~ ER + HCC.Riskscore + DRG.Class + 
    DRG.Complication, family = binomial(link = "logit"), data = readmission)

Coefficients:
                           Estimate Std. Error z value Pr(>|z|)    
(Intercept)                -1.07068    0.24376  -4.392 1.12e-05 ***
ER                          0.14079    0.08429   1.670 0.094885 .  
HCC.Riskscore               0.50109    0.09043   5.541 3.01e-08 ***
DRG.ClassSURG               0.78321    0.33588   2.332 0.019710 *  
DRG.ClassUNGROUP            0.95902    0.26753   3.585 0.000337 ***
DRG.ComplicationMedicalNoC -0.76443    0.20963  -3.647 0.000266 ***
DRG.ComplicationOther      -0.69363    0.32790  -2.115 0.034396 *  
DRG.ComplicationSurgMCC.CC -0.27994    0.37479  -0.747 0.455116    
DRG.ComplicationSurgNoC    -0.63931    0.36553  -1.749 0.080291 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1361.9  on 999  degrees of freedom
Residual deviance: 1248.2  on 991  degrees of freedom
AIC: 1266.2
```

```
Number of Fisher Scoring iterations: 4
```

### 3.3

**Coding:**

```
## Task 3.3: Compare residual types and choose one for diagnostics

# statmod package is needed for quantile residuals
library(statmod)

## 1. Compute three types of residuals
dev_res   <- residuals(drop_g, type = "deviance")
pear_res  <- residuals(drop_g, type = "pearson")
q_res     <- qresid(drop_g)   # quantile residuals (as in the handout)

## 2. Compare densities with N(0,1)

# Grid for standard normal density
x_grid <- seq(-4, 4, length = 200)
dnorm_grid <- dnorm(x_grid)

par(mfrow = c(1, 3))

# (a) Deviance residuals
plot(density(dev_res),
     main = "Density of deviance residuals",
     xlab = "Deviance residual")
lines(x_grid, dnorm_grid, lty = 2)  # dashed: standard normal density

# (b) Pearson residuals
plot(density(pear_res),
     main = "Density of Pearson residuals",
     xlab = "Pearson residual")
lines(x_grid, dnorm_grid, lty = 2)

# (c) Quantile residuals
plot(density(q_res),
     main = "Density of quantile residuals",
     xlab = "Quantile residual")
lines(x_grid, dnorm_grid, lty = 2)

par(mfrow = c(1, 1))

## 3. Q{Q plots of the three residual types

par(mfrow = c(1, 3))

qqnorm(dev_res,  main = "Q-Q: deviance residuals")
qqline(dev_res)
```

```
qqnorm(pear_res, main = "Q-Q: Pearson residuals")
qqline(pear_res)

qqnorm(q_res,    main = "Q-Q: quantile residuals")
qqline(q_res)

par(mfrow = c(1, 1))

# Fitted linear predictor eta = logit(p_hat)
eta_hat <- predict(drop_g, type = "link")

par(mfrow = c(1, 2))

# (a) Quantile residuals vs linear predictor
plot(eta_hat, q_res,
     xlab = "Fitted linear predictor (eta)",
     ylab = "Quantile residuals",
     main = "Quantile residuals vs fitted eta")
abline(h = 0, lty = 2)

# (b) Q{Q plot of quantile residuals
qqnorm(q_res, main = "Q-Q plot of quantile residuals")
qqline(q_res)

par(mfrow = c(1, 1))
```

**Plots:**

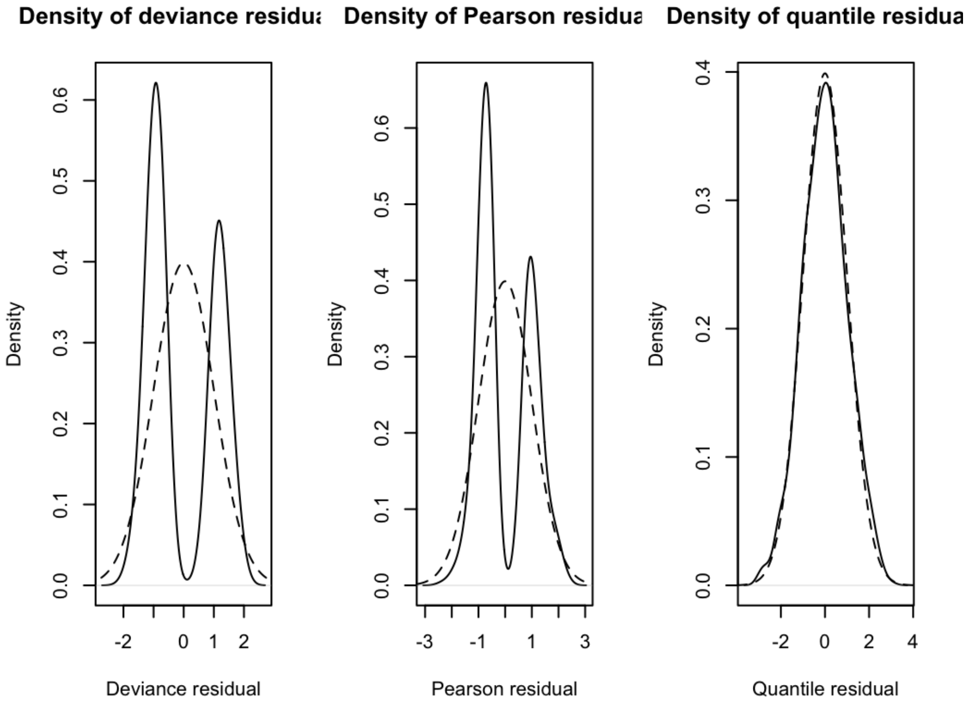

Figure 4: Density plots for deviance, Pearson and quantile residuals.

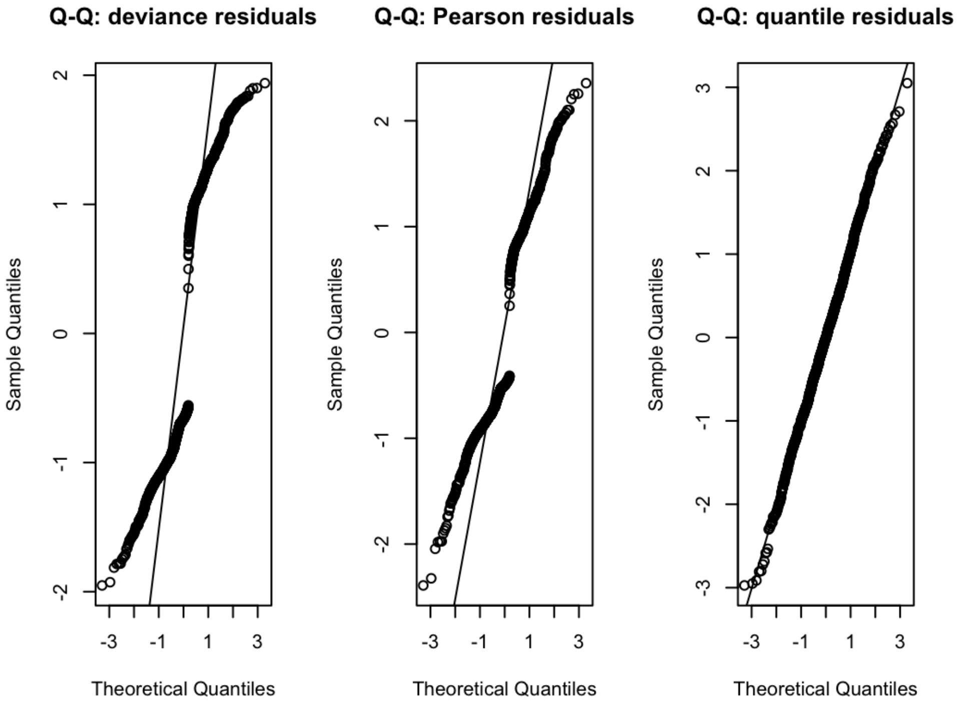

Figure 5: Q–Q plots for deviance, Pearson and quantile residuals.

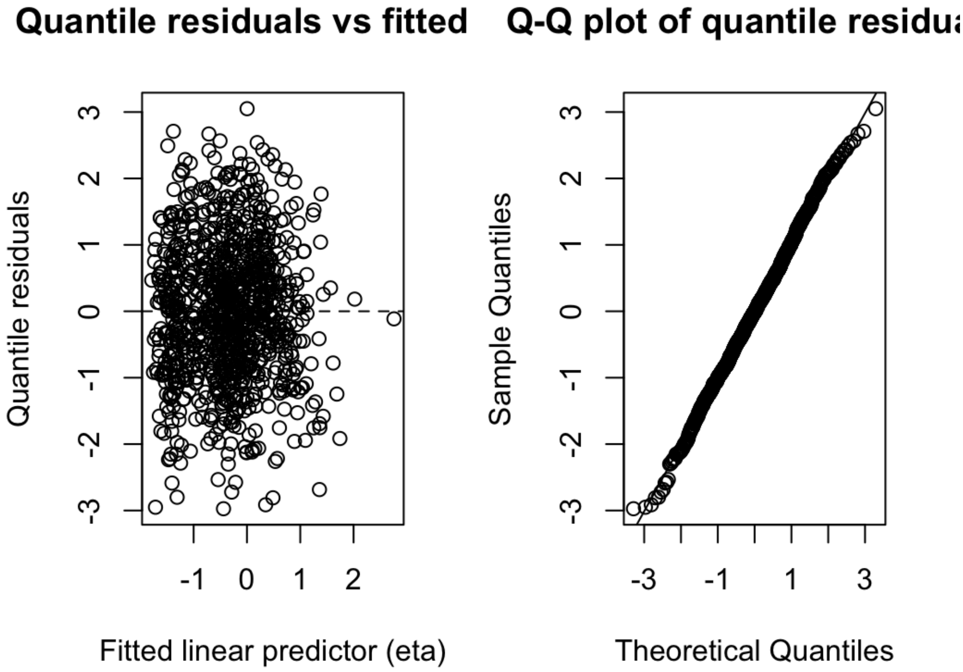

Figure 6: Quantile residuals for drop_g: residuals vs fitted eta and Q–Q plot.

**Justification / Arguments:**

For the logistic model drop_g, I compared three types of residuals, following the GLM programming notes: deviance residuals, Pearson residuals and quantile residuals. I first plotted the density functions of all three residuals together with the standard normal density, as shown in Figure 4. The density curves for the deviance and Pearson residuals are clearly not normal. The corresponding Q–Q plots in Figure 5 show strong departures from the straight line, especially in both tails. This indicates that, under the binomial model, these two residual types do not behave in an approximately normal way, so they are not ideal for checking the random component here.

In contrast, the quantile residuals have a much more symmetric density and almost lie on top of the standard normal density, with only small differences in the extreme tails. Their Q–Q plot is close to a straight line over most of the range, with only mild deviations at the very ends. Quantile residuals are constructed to be approximately normal when the binomial random component and logit link are correctly specified, so this behaviour is more consistent with the GLM assumptions. Based on these plots, I chose quantile residuals for further diagnostics. The quantile residual diagnostics in Figure 6 do not show any strong pattern in the residuals versus fitted linear predictor, and the Q–Q plot remains close to the reference line. Overall, these diagnostics based on quantile residuals support the appropriateness of the binomial random component for `drop_g`, although there may still be some minor deviations in the tails.

### 3.4

**Coding:**

```
## Task 3.4: Detect outliers (|standardised deviance residual| > 2.5)
##          and refit final model fin_g

## 1. Compute standardised deviance residuals
r_std_dev <- rstandard(drop_g, type = "deviance")   # as in the handout

## 2. Use benchmark |r_i| > 2.5 to flag outliers
cutoff  <- 2.5
out_idx <- which(abs(r_std_dev) > cutoff)

# Number and indices of outliers
length(out_idx)          # number of outliers
out_idx                  # row indices
readmission[out_idx, ]   # optional: inspect these observations

## 3. Remove outliers to obtain reduced dataset
if (length(out_idx) == 0) {
  # No observation exceeds 2.5: use original data
  readmission_clean <- readmission
} else {
  readmission_clean <- readmission[-out_idx, ]
}

nrow(readmission_clean)  # sample size after deletion

## 4. Refit the same logistic model on the reduced dataset to obtain fin_g
fin_g <- glm(Readmission.Status ~ ER + HCC.Riskscore +
               DRG.Class + DRG.Complication,
             family = binomial(link = "logit"),
             data   = readmission_clean)

summary(fin_g)

## 5. Compare fin_g with original model drop_g
AIC(drop_g, fin_g)
anova(drop_g, fin_g, test = "Chisq")
```

### 3.5

### Coding:

```
Task 3.5: Predictive probability for a new patient

# Inspect factor levels in the final model (for constructing newdata)
fin_g$xlevels

# Construct a new patient:
# ER = 2, HCC.Riskscore = 1.5,
# DRG.Class = second level (e.g. "SURG"),
# DRG.Complication = fourth level (e.g. "SurgMCC.CC")
new_patient <- data.frame(
  ER            = 2,
  HCC.Riskscore = 1.5,
  DRG.Class     = factor(fin_g$xlevels$‘DRG.Class‘[2],
                         levels = fin_g$xlevels$‘DRG.Class‘),
  DRG.Complication = factor(fin_g$xlevels$‘DRG.Complication‘[4],
                            levels = fin_g$xlevels$‘DRG.Complication‘)
)

new_patient

# Use final model fin_g to predict readmission probability
pred_prob <- predict(fin_g, newdata = new_patient, type = "response")
pred_prob
```

### Justification / Arguments:

I created a new observation representing a relatively high-risk surgical encounter. The new patient has `ER = 2` previous emergency-room visits and `HCC.Riskscore = 1.5`, which lies in the upper part of the observed risk-score range. For the categorical covariates I chose `DRG.Class = SURG` and `DRG.Complication = SurgMCC.CC`, corresponding to a surgical case with recorded major complications or comorbidities. These values are all within the data dictionary.

Using the final fitted model `fin_g` and the R command `predict(fin_g, newdata = new_patient, type = "response")`, the predicted probability of readmission for this patient is about 0.61. In other words, the model suggests that this type of surgical patient has roughly a 61% chance of being readmitted after discharge.

```
> new_patient
  ER HCC.Riskscore DRG.Class DRG.Complication
1  2           1.5      SURG       SurgMCC.CC
> pred_prob
        1
0.6143877
```

## Part C

In Part A I plotted a histogram of LOS and found right skew, so I applied a log transformation; the transformed response `log_LOS` was closer to normal. I then used `DRG.Class` as a categorical predictor and drew boxplots of `log_LOS` by `DRG.Class`. Next I fitted a baseline multiple linear regression model, removed variables not significant at the 5% level to obtain the reduced model `m_red`, and compared goodness-of-fit measures. The fit was very similar, but `m_red` used fewer predictors. Standard diagnostic plots for `m_red` looked reasonable, and I checked for outliers, multicollinearity and confirmed the model using AIC.

In Part B I built a logistic GLM `g0` for readmission, removed variables not significant at the 10% level to obtain `drop_g`, and found no meaningful loss of fit. I compared residual types, chose quantile residuals, and the results supported the binomial random component for `drop_g`. I found no strong outliers, so `fin_g` coincided with `drop_g`, and I used `fin_g` to predict the readmission probability for a new patient. A practical implication is that such models can flag high-risk patients for closer follow-up. A limitation is that the analysis uses a single observational dataset, so unmeasured confounding may bias the results.