CSCI E-106:Assignment 2

Shreya Bajpai

2024-09-20

Problem 1

Refer to the Grade point average Data. The director of admissions of a small college selected 120 students at random from the new freshman class in a study to determine whether a student's grade point average (GPA) at the end of the freshman year (Y) can be predicted from the ACT test score (X). (40 points)

```r
library(ggplot2)
library(lattice)
library(caret)
gpa_data <- read.csv('/Users/shreyabajpai/CSCI E-106 - Data Modeling/CSCI-E-106-Assignment-2-Files/Grade
summary(gpa_data)
```

```
##        Y               X
##  Min.   :0.500   Min.   :14.00
##  1st Qu.:2.689   1st Qu.:21.00
##  Median :3.078   Median :25.00
##  Mean   :3.074   Mean   :24.73
##  3rd Qu.:3.593   3rd Qu.:28.00
##  Max.   :4.000   Max.   :35.00
```

a-) Create train and test data sets. Use 70% of the data for train data set and use remaining data (30% of data) for test data set (use set.seed(1023)). (5points)

There are 120 observations of grade point average (GPA) (**Y**) and ACT test score (**X**) in the **gpa_data** data frame.

1. First, we set a seed using set.seed(), which ensures that we can generate the same training and test sets every time we run the code.
2. Then, we use the sample() function to randomly assign rows from gpa_data to either the training or test set. About 70% of the rows are marked as TRUE (for training) and 30% as FALSE (for testing), based on the probability setting prob=c(0.7, 0.3).
3. The replace=TRUE argument ensures the selection is done randomly, without applying the probabilities sequentially, so that we truly generate, at random, a training and test data set.

When the sample vector is ready, we leverage it to define our training_data and testing_data data frames by verifying which logical is populated ('True' - Training Data, 'False' - Testing Data). When we capture the data frames now built, we can see they contain data from gpa_data as per the defined percentages in the question (83/120 ~ 0.69167; 38/120 ~ 0.3167).

```r
# Make code replicable
set.seed(1023)

# Generate a vector, sample, which disperses gpa_data into 70% and 30% for the purpose of splitting the
sample <- sample(c(TRUE, FALSE), nrow(gpa_data), replace=TRUE, prob=c(0.7,0.3))
```

```r
# Using the sample vector, we load the values held in sample when 'TRUE' (sample) to be the population
training_data <- gpa_data[sample, ]
testing_data <- gpa_data[!sample, ]
str(training_data)
```

```
## 'data.frame':    83 obs. of  2 variables:
##  $ Y: num  3.9 3.88 2.54 3.03 3.87 ...
##  $ X: int  21 14 22 21 31 27 29 26 24 33 ...
```

```r
str(testing_data)
```

```
## 'data.frame':    37 obs. of  2 variables:
##  $ Y: num  3.78 2.96 3.54 3.08 3.01 ...
##  $ X: int  28 32 30 24 24 25 28 28 25 22 ...
```

b-) Build a regression model on the train data set. Write down the stand error of the slope and intercept. Plot regression line. Comment on the model performance.(10 points)

We build our model and evaluate the model's performance before diving into exploring the standard error of the slope and intercept and plotting the regression line.

First, we fit the linear regression model with the training data, where Y is the dependent (response) variable, GPA, and X is the independent (predictor) variable, ACT Score. Next, we use the predict() function to generate predicted values for each observation in the training dataset based on the relationship between X and Y estimated by the linear regression model, fit_gpa_trn_data. Then, to observe the performance of the model, we create a ModelGPA data frame to compare the actual observed values of Y (obs) with the predicted values (pred) for each instance in our training data. Finally, we use Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Multiple R^2 as the performance metrics for our regression model.

```r
fit_gpa_trn_data <- lm(Y ~ X, data=training_data)

# Get the predicted values
PredictedGPA<-predict(fit_gpa_trn_data,training_data)

# Create a data frame with 2 columns:rm actual and predicted values
ModelGPA<-data.frame(obs = training_data$Y, pred=PredictedGPA)

# Calculate Performance Statistics
summary(ModelGPA)
```

```
##       obs              pred      
##  Min.   :0.500   Min.   :2.561  
##  1st Qu.:2.571   1st Qu.:2.862  
##  Median :3.072   Median :3.024  
##  Mean   :3.030   Mean   :3.030  
##  3rd Qu.:3.581   3rd Qu.:3.164  
##  Max.   :4.000   Max.   :3.535  
```

```r
#Calculate MSE and RMSE
MSETraining <- (mean((training_data$Y-PredictedGPA)^2))
RMSETraining <- sqrt(mean((training_data$Y-PredictedGPA)^2))

#Calculate MAE
MAETraining <- mean(abs(training_data$Y-PredictedGPA))

# Calculate R-squared for the training data
```

```
SST_Train <- sum((training_data$Y - mean(training_data$Y))^2)  # Total sum of squares
SSE_Train <- sum((training_data$Y - PredictedGPA)^2)  # Residual sum of squares
RSqrTrain <- 1 - (SSE_Train / SST_Train)

cat("The Mean Squared Error (MSE): ", MSETraining, "\n")
```

```
## The Mean Squared Error (MSE):  0.4333239
```

```
cat("The Root Mean Squared Error (RMSE): ", RMSETraining, "\n")
```

```
## The Root Mean Squared Error (RMSE):  0.6582734
```

```
cat("The Mean Absolute Error (MAE): ", MAETraining, "\n")
```

```
## The Mean Absolute Error (MAE):  0.4960106
```

```
cat("The Multiple R Squared (R^2): ", RSqrTrain, "\n")
```

```
## The Multiple R Squared (R^2):  0.09518177
```

```
#I will leverage the function below, as Hakan did, going forward, the calculations above were to ensure
defaultSummary(ModelGPA)
```

```
##       RMSE   Rsquared        MAE
## 0.65827339 0.09518177 0.49601059
```

Based on the summary() output, the model performs reasonably well in predicting central tendencies but may not be capturing variability in the data, particularly at the lower and higher ends of the observed range.

- The mean of the observed values is 3.030, and the median is 3.072, suggesting that the data is fairly centered around these values, with no extreme skew (Min: 0.500 Max: 4.000); however, the predicted values from the model range from 2.561 to 3.535, meaning the model's predictions are not capturing the full range of the observed data. *Notably*, the minimum predicted value (2.561) is much higher than the minimum observed value (0.500), and the maximum predicted value (3.535) is lower than the maximum observed value (4.000).
- For the 1st quartile (25th percentile), the observed value is 2.571, while the predicted value is slightly higher at 2.862. This indicates that the model tends to overpredict values at the lower end of the distribution.
- For the 3rd quartile (75th percentile), the observed value is 3.581, while the predicted value is 3.164, indicating that the model is underpredicting higher values.

In contrast, the performance metrics paint a different picture about the regression model.

Error-Based Statistics Review: The MSE, *the squared differences between the predicted and actual values*, of 0.4333 indicates moderate error in the model, given the observed values range from 0.500 to 4.000. The RMSE, *the average difference between the model's predicted values and the actual values*, suggests predictions deviate from actual GPA values by about 0.658 points, reflecting similar moderate error. The MAE, *absolute difference between the predicted and actual values*, of 0.4960, a deviation too high when considering GPA's limited range to begin with, reinforces that the model may not fully capture data nuances, particularly as seen in the 1st and 3rd quartiles. Notably, MAE is less sensitive to outliers than MSE and RMSE, so we cannot attribute the presence of outliers as causing this level of error the model makes between the actual and predicted values.

Coefficient of Determination: The R^2 value of 9.52% is concerning, as it indicates that the ACT Score explains only a small portion of the variance in GPA. This low value, combined with the moderate errors exhibited by the model from other metrics, suggests the model does not effectively capture the underlying relationship between GPA and ACT Score.

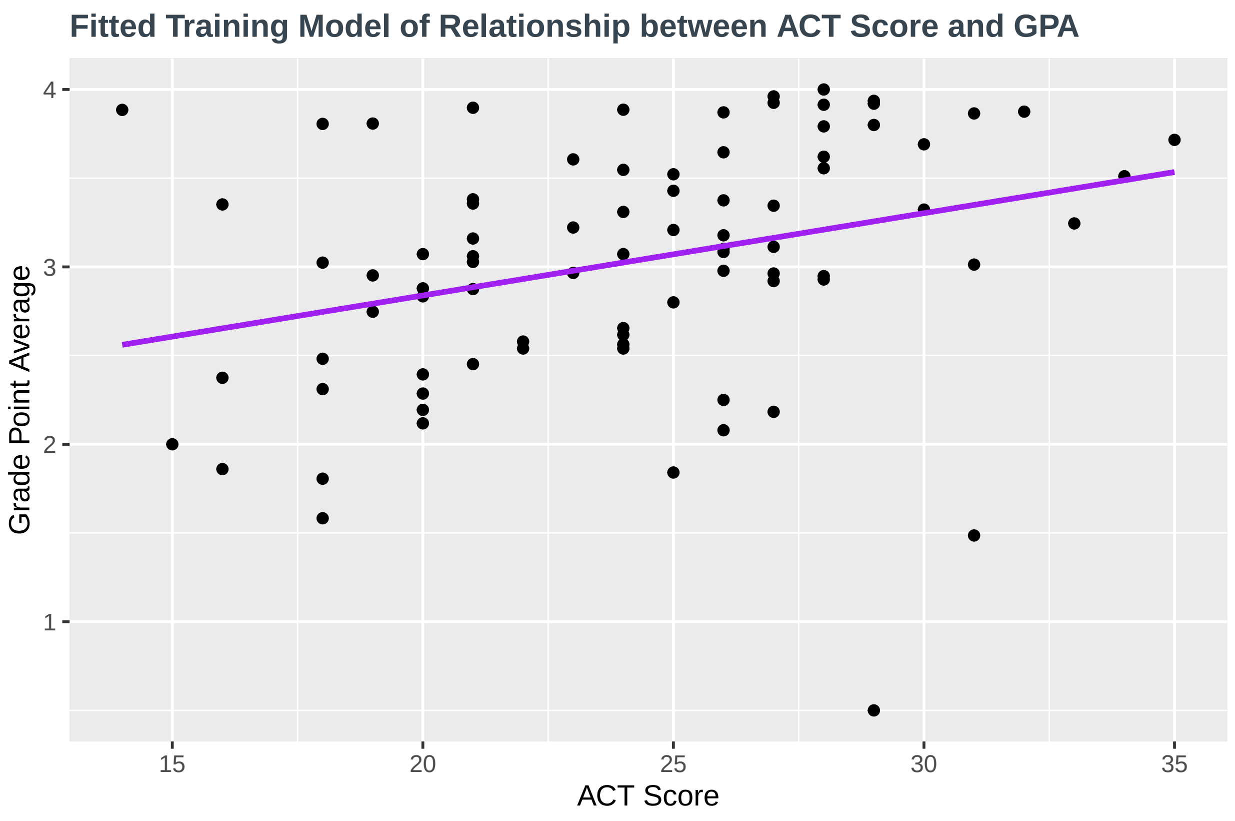
Now, we plot the regression line of the model on the trained dataset:

```r
library(ggpubr)
summary(fit_gpa_trn_data)
```

```
##
## Call:
## lm(formula = Y ~ X, data = training_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.75636 -0.36108  0.04006  0.47559  1.32434
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.91133    0.39017   4.899 4.86e-06 ***
## X            0.04638    0.01589   2.919  0.00454 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6664 on 81 degrees of freedom
## Multiple R-squared:  0.09518,    Adjusted R-squared:  0.08401
## F-statistic: 8.521 on 1 and 81 DF,  p-value: 0.004545
```

```r
library(ggplot2)
ggplot(data=training_data, aes(x = X, y = Y)) +
    geom_point(color = "black") +
    labs(title = "Fitted Training Model of Relationship between ACT Score and GPA",
    caption = "Source: Grade point average.csv",
    x = "ACT Score",
    y = "Grade Point Average") +
    theme(plot.title = element_text(color = "#36454F", size = 12, face = "bold")) + geom_smooth(color='
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Source: Grade point average.csv

The *standard error of the slope* (Value of Slope Estimate: 0.04638) is 0.01589 and the *standard error of the intercept* (Value of Intercept Estimate: 1.91133) is 0.39017. These standard errors provide an indication of the variability of the coefficient estimates, with lower standard errors corresponding to more precise estimates of the intercept and slope of the regression line. Although the slope is positive and statistically significant, the practical impact is minimal. For each additional ACT point, the predicted GPA only increases by 0.04638, which suggests that the ACT score alone is not a strong predictor of GPA in this context. Visually, the linear regression line does not seem to appropriately reduce residuals of the dispersed data points and we delve into that below.

Another means to examine the model performance of this linear regression model is to dive into key metrics of the model's output:

1. $\mathbf{R^2}$ is the proportion of variance in the dependent variable, GPA, that is explained by the independent variable, ACT Score. In our output, this value is 0.09518, which indicates that only 9.52% of the variance in GPA is explained by the ACT score. This indicates that the model is not explaining much of the variance in GPA by only utilizing ACT Score.
2. **Adjusted** $\mathbf{R^2}$ adjusts the R^2 for the number of predictors in the model, which for our case, is only one predictor, which is why this value, 0.08401, is close to R^2. This further suggests that our model's predictive power is limited, suggesting there may be other variables influencing the GPA which are not included in our data set.
3. The **F-statistic** tests whether the independent variable, ACT Score, explains a significant portion of the variance of the dependent variable, GPA. The *p-value* is small (0.004545) indicating the model is statistically significant. While there may be a statistically significant relationship between ACT score and GPA, the strength of this relationship (as indicated by R^2) is weak.
4. The **coefficients** of the regression model

$$GPA = 1.91133 + 0.04638 \times (X_i)$$

   indicate that for every one point increase in a student's ACT score, the model predicts a 0.04638 point increase in the GPA. This slope suggests a positive relationship between ACT score and GPA, but the impact is relatively small. The intercept states that for a student who might have received a 0 on the ACT score, the model predicts their GPA to be 1.91, but this might not make sense in a real-world context.

5. The **Residual Standard Error (RSE)** is a measure of the average amount that the observed values deviate from the regression line. For our model, the RSE is 0.6664, which is small, thereby theoretically a better fit; however, GPA ranges from 0 to 4, so an RSE this small indicates a degree of error in the model's prediction.

b-) Are the slope and intercept statistically significant? Write down your null and alternative hypothesis and p value of the test. (10 points)

First, let us examine the null and alternative hypothesis of the slope (ACT Score):

H_o : $\beta_1 = 0$ The slope is equal to 0. The ACT score has no effect on GPA.

H_a : $\beta_1 \neq 0$ The slope is not equal to 0. The ACT score has an effect on GPA.

```
#Extract intercept coefficient and standard error
slp_actgpa <- coef(summary(fit_gpa_trn_data))["X", "Estimate"]
slp_actgpa_se <- coef(summary(fit_gpa_trn_data))["X", "Std. Error"]

#Calculate t-statistic for slope
t_stat_slp_act_gpa <- slp_actgpa / slp_actgpa_se

#Calculate df
df_actgpa <- fit_gpa_trn_data$df.residual

#Calculate two-sided p-value
p_val_slp_actgpa <- 2 * pt(-abs(t_stat_slp_act_gpa), df=df_actgpa)

#Calculate critical t-value for a 95% confidence interval (two-sided)
alpha <- 0.05
t_crit_val_slp_actgpa <- qt(1 - alpha / 2, df=df_actgpa)

#Output the t-statistic and p-value
cat("T-statistic for Slope:", t_stat_slp_act_gpa, "\n")
```

```
## T-statistic for Slope: 2.919031
```

```
cat("p-value:", p_val_slp_actgpa, "\n")
```

```
## p-value: 0.004544645
```

```
cat("Critical t-value (95% CI):" , t_crit_val_slp_actgpa, "\n")
```

```
## Critical t-value (95% CI): 1.989686
```

Since the slope's t-statistic (2.919031) is greater than the critical t-value (1.989686) and the p-value (0.004544645) is smaller than 0.05, we can conclude that the slope is statistically significant at the 95% confidence level, which means that ACT score does have a significant impact on GPA.

Then, let us examine the null and alternative hypothesis of the y-intercept (ACT Score):

H_o : $\beta_0 = 0$ The intercept is equal to 0. When the ACT score is 0, the GPA is 0.

H_a : $\beta_0 \neq 0$ The intercept is not equal to 0. When the ACT score is 0, the GPA is not 0.

```
#Extract intercept coefficient and standard error
intcp_actgpa <- coef(summary(fit_gpa_trn_data))["(Intercept)", "Estimate"]
intcp_actgpa_se <- coef(summary(fit_gpa_trn_data))["(Intercept)", "Std. Error"]
```

```r
#Calculate t-statistic for intercept
t_stat_intcp_actgpa <- intcp_actgpa / intcp_actgpa_se

#Calculate df
df_actgpa <- fit_gpa_trn_data$df.residual

#Calculate two-sided p-value
p_val_intcp_actgpa <- 2 * pt(-abs(t_stat_intcp_actgpa), df=df_actgpa)

#Calculate critical t-value for a 95% confidence interval (two-sided)
alpha <- 0.05
t_crit_val_intcp_actgpa <- qt(1 - alpha / 2, df=df_actgpa)

#Output the t-statistic and p-value
cat("T-statistic for Intercept:", t_stat_intcp_actgpa, "\n")
```

```
## T-statistic for Intercept: 4.898765
```

```r
cat("p-value:", p_val_intcp_actgpa, "\n")
```

```
## p-value: 4.85857e-06
```

```r
cat("Critical t-value (95% CI):" , t_crit_val_intcp_actgpa, "\n")
```

```
## Critical t-value (95% CI): 1.989686
```

Since the absolute value of the t-statistic (4.898765) is much greater than the critical t-value (1.989686) and the p-value (4.85857e-06) is much smaller than the typical threshold (0.05), we can reject the null hypothesis and conclude that the intercept is statistically significant at the 95% confidence level. This is not semantically meaningful, as ACT scores cannot be 0 so there cannot be a corresponding significant base level of GPA.

c-) What is the error variance of the regression line? (5 points)

Error variance of a regression line refers to the variance of the residuals (errors), and it quantifies how much the observed values deviate from the predicted values. In a simple linear regression, the error variance can be estimated by the Residual Standard Error (RSE) or by the Mean Squared Error (MSE).

From our model output, we know Residual Standard Error (RSE) = 0.6664, so to calculate Error Variance, we square this value based on the formula:

$ErrorVariance = (ResidualStandardError)^2 = (0.6664)^2 = 0.4441.$

This value, 0.4441, represents the average squared distance between the actual GPA values and the predicted GPA values from the regression model. It informs us how much the actual GPAs tend to deviate from the values predicted by the model. A lower error variance means the model's predictions are closer to the actual values, while a higher variance indicates more variability in the errors and a less accurate model.

Note: When we calculate Error Variance via 'R' code, there is a slight variation, but we know this can be due to R storing values internally with many more significant digits than are displayed in outputs.

```r
print(summary(fit_gpa_trn_data)$sigma**2)
```

```
## [1] 0.4440232
```

d-) Check the model performance on the test data set. Is the model performance consistent. (10 points)

```r
# Get the predicted values based on the model built above
PredictedTestGPA<-predict(fit_gpa_trn_data, newdata=testing_data)
```

```r
# Create a data frame with 2 columns:rm actual and predicted values
ModelTestGPA<-data.frame(obs = testing_data$Y, pred=PredictedTestGPA)

# Calculate Performance Statistics
summary(ModelTestGPA)
```

```
##       obs             pred
##  Min.   :2.069   Min.   :2.746
##  1st Qu.:2.867   1st Qu.:3.024
##  Median :3.083   Median :3.117
##  Mean   :3.173   Mean   :3.121
##  3rd Qu.:3.591   3rd Qu.:3.256
##  Max.   :3.956   Max.   :3.396
```

```r
#Calculate MSE and RMSE
MSETesting <- (mean((testing_data$Y-PredictedTestGPA)^2))
RMSETesting <- sqrt(mean((testing_data$Y-PredictedTestGPA)^2))

cat("The Mean Squared Error (MSE): ", MSETesting, "\n")
```

```
## The Mean Squared Error (MSE):  0.2707591
```

```r
rbind(defaultSummary(ModelGPA), defaultSummary(ModelTestGPA))
```

```
##           RMSE    Rsquared       MAE
## [1,] 0.6582734 0.095181768 0.4960106
## [2,] 0.5203452 0.002823154 0.4148962
```

Let us compare the performance of the model by examining its performance on the testing and training datasets.

1. **Range of Values** In the training data, the observed values range from 0.500 to 4.000, but the predicted values range from 2.561 to 3.535. The model underestimates the high values and overestimates the low values in the training data. In the test data, the observed values range from 2.069 to 3.956, and the predicted values range from 2.746 to 3.396. The model also underpredicts high values and overpredicts low values but within a narrower range in the testing dataset, suggesting a better fit. The model shows, on both data sets, a consistent pattern of underpredicting high values and overpredicting low values.

2. **Quartiles** The model performs similarly on both the training and testing datasets:

- The 1st quartile in both datasets shows slight overprediction in both training and test data models (*Training*: 2.571 (obs) vs 2.862 (pred), *Testing*: 2.867 (obs) vs 3.024 (pred)).
- The median and mean prediction in both datasets is quite close to the actual median (*Training*: $Median_{pred} = 3.072$, $Median_{obs} = 3.024$, $\mu_{pred} = 3.030$, $\mu_{obs} = 3.030$, *Testing*: $Median_{pred} = 3.083$, $Median_{obs} = 3.117$, $\mu_{pred} = 3.173$, $\mu_{obs} = 3.121$), suggesting a good prediction of central values in both models with the performance in the training data being slightly better when compared to the performance in the testing data. .
- The 3rd quartile shows underprediction in both models (*Training*: 3.581 (obs) vs 3.164 (pred), *Testing*: 3.591 (obs) vs 3.256 (pred)). The data model generally performs better in terms of quartiles on the testing dataset, with predicted values being closer to the observed values than in the data model's performance on the training dataset.

3. **Performance Metrics**

- The MSE of the data model on testing data is 0.2707591, which is lower than the MSE of the data model on training data (0.4333239), suggesting that the test model has a better fit and smaller errors compared to the training model, thereby we can conclude, at first glance, it generalizes well to unseen data and doesn't overfit.

- The RMSE of the data model on testing data is 0.5203452, which is lower than the RMSE of the data model on the training data (0.6582734), and this metric is easier to interpret as it represents the average error magnitude in the same units as GPA, so we know that the model results in lower errors in the predicting the GPA of the testing data when compared to the errors in predicting the GPA of the training data. However, a lower RMSE of our model on test data could also indicate that the test data might be easier to predict than the training data or that our model is slightly overfitting to the training data.
- The MAE for the test data is 0.4148962, also lower than the MAE for the training data (0.4960106), suggesting that the test model makes smaller prediction errors on average compared to the training model. While this might suggest decent performance on the test set, it could also mean the test set is not as challenging as the training data.
- The R^2 drop when comparing the model's performance on the training (0.09518177) versus testing (0.002823154) datasets indicates that the model is not generalizing well and performs poorly on unseen data, suggesting overfitting. The model is struggling to capture the relationship between the predictor and the target variables on both the training and test data.

The model exhibits bias in predicting extreme values as it tends to underpredict high values and overpredict low values, and while the model misleads us by managing to limit error and perform better on the test data compared to the training data, with lower MSE, RMSE and MAE values, the very low R^2 present in both the data sets proves that the model is not fitting the data well and may not be complex enough to explain the variance of GPA by ACT Score alone.

Problem 2

Refer to the CDI data set. The number of active physicians in a CDI (Y) is expected to be related to total population, number of hospital beds, and total personal income. (60 points)

```r
cdi_data <- read.csv('/Users/shreyabajpai/CSCI E-106 - Data Modeling/CSCI-E-106-Assignment-2-Files/CDI 
cdi_desc_data <- read.csv('/Users/shreyabajpai/CSCI E-106 - Data Modeling/CSCI-E-106-Assignment-2-Files/
```

a-) Regress the number of active physicians in turn on each of the three predictor variables. State the estimated SINGLE regression functions. (10 points)

The formula for single regression function with one dependent and and one independent variable is: $\hat{Y} = \beta_0 + \beta_1 X_1$. In our data set, $\hat{Y}$ is the predicted number of active physicians, and, for each individual linear regression, β_1 is the total population, the number of hospital beds and total personal income in each respective regression function.

```r
fit_act_phys_totpop <- lm(Number.of.active.physicians ~ Total.population, data=cdi_data)
summary(fit_act_phys_totpop)
```

```
##
## Call:
## lm(formula = Number.of.active.physicians ~ Total.population,
##     data = cdi_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1969.4  -209.2   -88.0    27.9  3928.7
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.106e+02  3.475e+01  -3.184  0.00156 **
## Total.population  2.795e-03  4.837e-05  57.793  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 610.1 on 438 degrees of freedom
## Multiple R-squared:  0.8841, Adjusted R-squared:  0.8838
## F-statistic:  3340 on 1 and 438 DF,  p-value: < 2.2e-16
```

Using the coefficients from the model output above, we transcribe the estimated regression function for Number of Active Physicians on Total Population to be:

$\hat{Y} = -110.6 + 0.002795 \times X_1$

We can extrapolate the values of the y-intercept (β_0) to indicate that when the total population is zero, the model predicts there would be -110.6 active physicians (which doesn't make practical sense) and the value of the slope (β_1) to indicate that for every unit increase in the total population, the predicted number of active physicians increases by approximately 0.002795. Since this value is positive, we can assume that as the total population increases, the number of active physicians is expected to increase. The p-values for the y-intercept and the slope are statistically significant, with the y-intercept's p-value (0.00246) and the slope's p-value (<2e-16) being much smaller than 0.05, indicating that the relationship between the Total Population and the Number of Active Physicians is highly significant.

```r
fit_act_phys_numofbeds <- lm(Number.of.active.physicians ~ Number.of.hospital.beds, data=cdi_data)
summary(fit_act_phys_numofbeds)
```

```
##
## Call:
## lm(formula = Number.of.active.physicians ~ Number.of.hospital.beds,
##     data = cdi_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3133.2  -216.8   -32.0    96.2  3611.1
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -95.93218   31.49396  -3.046  0.00246 **
## Number.of.hospital.beds   0.74312    0.01161  63.995  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 556.9 on 438 degrees of freedom
## Multiple R-squared:  0.9034, Adjusted R-squared:  0.9032
## F-statistic:  4095 on 1 and 438 DF,  p-value: < 2.2e-16
```

Using the coefficients from the model output above, we transcribe the estimated regression function for Number of Active Physicians on Number of Hospital Beds to be:

$\hat{Y} = -95.93218 + 0.74312 \times X_1$

We can extrapolate the values of the y-intercept (β_0) to indicate that when the number of hospital beds is zero, the model predicts there would be -95.93218 (~96) active physicians (which doesn't make practical sense) and of the slope (β_1) to indicate that for every unit increase in the number of hospital beds, the predicted number of active physicians increases by approximately 0.74312. Since this value is positive, we can assume that as the number of hospital beds increases, the number of active physicians is expected to increase. The p-values for the y-intercept and the slope are statistically significant, with the y-intercept's p-value (0.00246) and the slope's p-value (<2e-16) being much smaller than 0.05, indicating that the relationship between the Number of Hospital Beds and the Number of Active Physicians is highly significant.

```r
fit_act_phys_totpi <- lm(Number.of.active.physicians ~ Total.personal.income, data=cdi_data)
summary(fit_act_phys_totpi)
```

```
##
## Call:
## lm(formula = Number.of.active.physicians ~ Total.personal.income,
##     data = cdi_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1926.6  -194.5   -66.6    44.2  3819.0
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -48.39485   31.83333   -1.52    0.129
## Total.personal.income   0.13170    0.00211   62.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 569.7 on 438 degrees of freedom
## Multiple R-squared:  0.8989, Adjusted R-squared:  0.8987
## F-statistic:  3895 on 1 and 438 DF,  p-value: < 2.2e-16
```

Using the coefficients from the model output above, we transcribe the estimated regression function for Number of Active Physicians on Total Personal Income to be:

$\hat{Y} = -48.39485 - 0.13170 \times X_1$

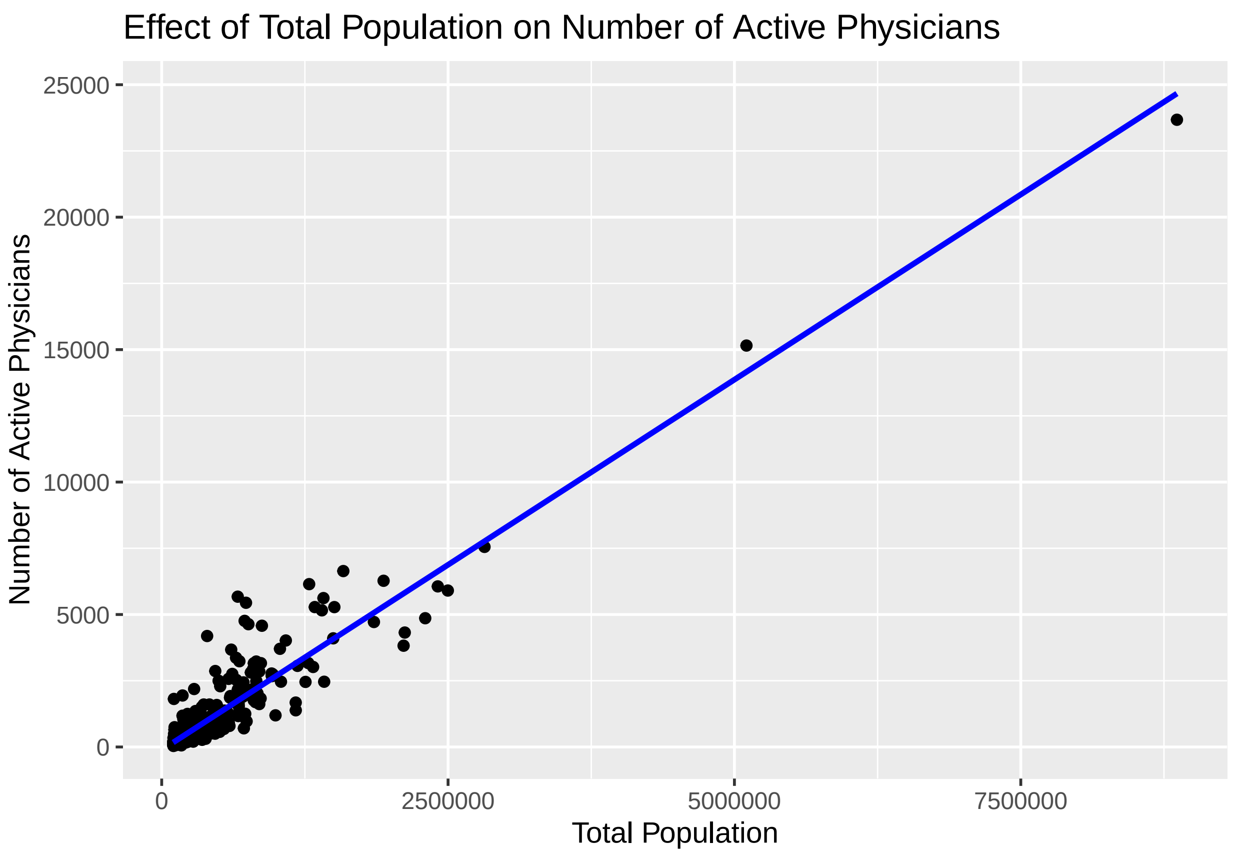
We can extrapolate the values of the y-intercept (β_0) to indicate that when total personal income is zero, the model predicts there would be -48.39485 (~49) active physicians (which doesn't make practical sense) and of the slope (β_1) to indicate that for every unit increase in total personal income, the predicted number of active physicians increases by approximately 0.13170. Since this value is positive, we can assume that as the total personal income increases, the number of active physicians is expected to increase. The p-values for only the slope is statistically significant, with the slope's p-value (<2e-16) being much smaller than 0.05, indicating that the relationship between the total personal income and the number of active physicians is highly significant.

b-) Plot EACH of the three estimated regression functions and data on separate graphs. Does a linear SINGLE regression relation appear to provide a good fit for each of the three predictor variables? (10 points)

For our first plot, we plot the regression the number of active physicians on the total population.

```r
plot_pop_act_phys <- ggplot(cdi_data, aes(x = Total.population, y = Number.of.active.physicians)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Effect of Total Population on Number of Active Physicians",
       x = "Total Population", y = "Number of Active Physicians")
print(plot_pop_act_phys)
```

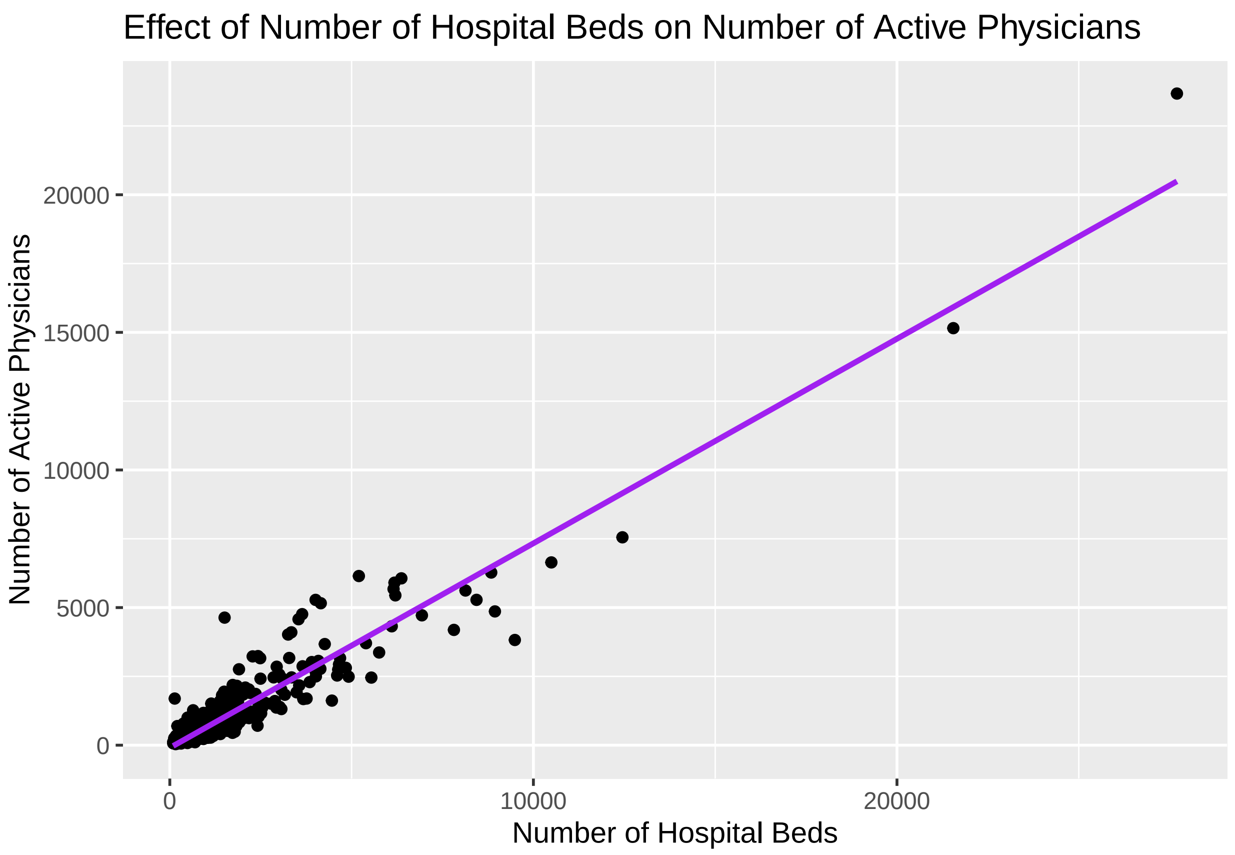
```
## `geom_smooth()` using formula = 'y ~ x'
```



For our second plot, we plot the regression the number of active physicians on the number of hospital beds.

```r
plot_numofbeds_act_phys <- ggplot(cdi_data, aes(x = Number.of.hospital.beds, y = Number.of.active.physic
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "purple") +
  labs(title = "Effect of Number of Hospital Beds on Number of Active Physicians",
       x = "Number of Hospital Beds", y = "Number of Active Physicians")
print(plot_numofbeds_act_phys)
```

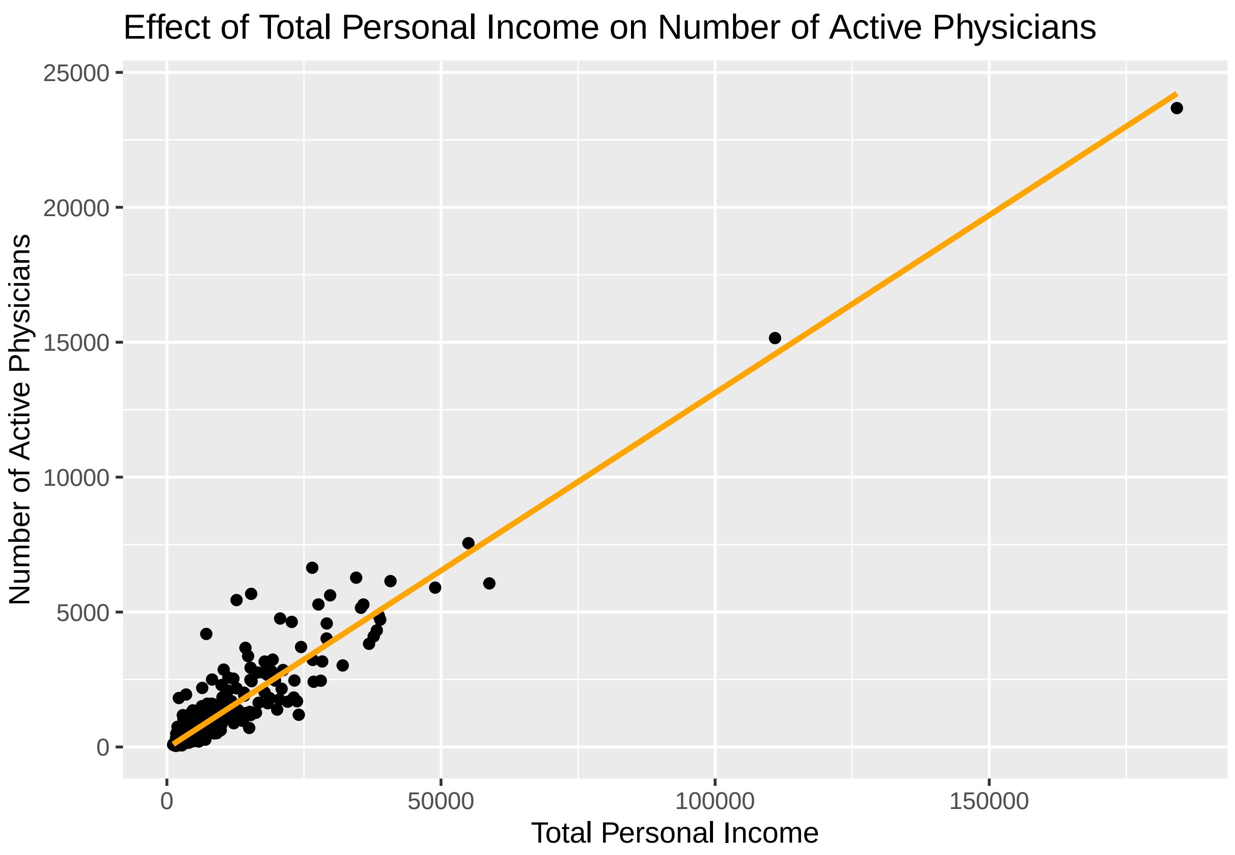
```
## `geom_smooth()` using formula = 'y ~ x'
```



For our third plot, we plot the regression the number of active physicians on the total personal income.

```
plot_totpi_act_phys <- ggplot(cdi_data, aes(x = Total.personal.income, y = Number.of.active.physicians)
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "orange") +
  labs(title = "Effect of Total Personal Income on Number of Active Physicians",
       x = "Total Personal Income", y = "Number of Active Physicians")
print(plot_totpi_act_phys)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



All three predictor variables are strong predictors of the number of active physicians (seen by the high R^2 values, with lowest being for Total Population (0.8841) and highest being for Number of Hospital Beds (0.9034)); however, if I were to choose, the predictor variable that not only explains the highest variability in the number of active physicians (of the ones we explored) but has the least Residual Standard Error (556.9 compared to the other two predictor variables, Total Personal Income (569.7) and Total Population (610.1)) in the **Number of Hospital Beds**. While it is not perfect, given the errors in prediction ranging from -3133.2 to 3611.1, it is the best option of the ones we plotted.

c-) Calculate MSE for each of the three predictor variables. Which predictor variable leads to the smallest variability around the fitted regression line? Which variable would you use the estimate Y and why? (15 points)

```r
# Compute residuals for each model
res_totpop <- residuals(fit_act_phys_totpop)
resi_numofbeds <- residuals(fit_act_phys_numofbeds)
resi_totpi <- residuals(fit_act_phys_totpi)

# Compute MSE for each model
mse_totpop <- mean(res_totpop^2)
mse_numofbeds <- mean(resi_numofbeds^2)
mse_totpi <- mean(resi_totpi^2)

# Display MSE values
mse_totpop
```

```
## [1] 370511.7
```

```r
mse_numofbeds
```

```
## [1] 308781.9
```

```r
mse_totpi
```

```
## [1] 323064.2
```

Since MSE shows us how much the actual values differ from the predicted ones (*average squared deviation of the observed values from the predicted values*), a lower MSE means the predictor does a better job at explaining the variability in the dependent variable and leaves less unexplained error. Looking at the results for the three independent variables, we can see that Number of Beds has the lowest MSE (3.0878192×10^5). This tells us that it does the best job of explaining the variability in Number of Active Physicians compared to the other predictors. So, if we had to pick just one variable to estimate the outcome, Number of Hospital Beds would be the best choice, as it gives us the most accurate predictions, with smaller residuals and a better fit overall.

d-) Calculate the Mean Absolute Error (MAE) for each of the three predictor variables. You can calculate the MAE by using the formula mean(abs(predicted Ys - actual Ys)). Which variable would you use the estimate Y and why? (15 points)

```r
# Compute MAE by taking the mean of the absolute residuals
mae_totpop <- mean(abs(res_totpop))
mae_numofbeds <- mean(abs(resi_numofbeds))
mae_totpi <- mean(abs(resi_totpi))

# Display MAE for each predictor with text
cat("MAE for Total.population: ", mae_totpop, "\n")
```

```
## MAE for Total.population:  338.9397
```

```r
cat("MAE for Number.of.hospital.beds: ", mae_numofbeds, "\n")
```

```
## MAE for Number.of.hospital.beds:  314.637
```

```r
cat("MAE for Total.personal.income: ", mae_totpi, "\n")
```

```
## MAE for Total.personal.income:  314.5328
```

The Mean Absolute Error (MAE) tells us how far off, on average, the predictions are from the actual values. A lower MAE means a better fit, as it reflects smaller errors overall. So, let's look at the MAE results to see which variable performs best. The MAE for Total.population is the highest (338.9396578), meaning this variable tends to give higher errors compared to the other two predictors, Number.of.hospital.beds and Total.personal.income. Interestingly, the MAEs for Number.of.hospital.beds (314.636962) and Total.personal.income (314.5327771) are very close, with Total.personal.income having a slight edge. While Total.personal.income has the smallest MAE, the model I would use to predict the dependent variable would be Number of Beds, as it minimizes both the MSE and the MAE.

d-) For the regression model you choose, Are the slope and intercept statistically significant? Write down your null and alternative hypothesis and p value of the test.(10 points)

I choose to regress Number of Physicians on Number of Beds.

Slope (β_1)

$H_o \; : \beta_1 = 0$ The slope is equal to 0 and is not statistically significant, meaning there is no relationship between the number of hospital beds and the number of active physicians.

$H_a : \beta_1 \neq 0$ The slope is not equal to 0 and is statistically significant, meaning there is a relationship between the number of hospital beds and the number of active physicians.

```r
#Extract intercept coefficient and standard error
slp_nop <- coef(summary(fit_act_phys_numofbeds))["Number.of.hospital.beds", "Estimate"]
```

```r
slp_nop_se <- coef(summary(fit_act_phys_numofbeds))["Number.of.hospital.beds", "Std. Error"]

#Calculate t-statistic for slope
t_stat_nop_slp <- slp_nop / slp_nop_se

#Calculate df
df_nop <- fit_act_phys_numofbeds$df.residual

#Calculate two-sided p-value
p_val_slp_nop <- 2 * pt(-abs(t_stat_nop_slp), df=df_nop)

#Calculate critical t-value for a 95% confidence interval (two-sided)
alpha <- 0.05
t_crit_val_slp_nop <- qt(1 - alpha / 2, df=df_nop)

#Output the t-statistic and p-value
cat("T-statistic for slope:", t_stat_nop_slp, "\n")
```

```
## T-statistic for slope: 63.99487
```

```r
cat("p-value:", p_val_slp_nop, "\n")
```

```
## p-value: 2.1383e-224
```

```r
cat("Critical t-value (95% CI):", t_crit_val_slp_nop, "\n")
```

```
## Critical t-value (95% CI): 1.965395
```

Given that the absolute t-statistic (63.99487) exceeds the critical t-value (1.965395) and the p-value (2.1383e-224) is significantly less than 0.05, we reject the null hypothesis. This leads us to conclude that the slope is statistically significant at the 95% confidence level, suggesting a robust relationship between the number of hospital beds and the number of active physicians.

Intercept (β_0)

$H_o \ : \beta_0 = 0$ The intercept is not statistically significant, meaning the intercept is 0. When the Number of Hospital Beds is 0, the Number of Physicians is 0.

$H_a : \beta_0 \neq 0$ The intercept is statistically significant, meaning the intercept is not 0. When the Number of Hospital Beds is 0, the Number of Physicians is not 0.

```r
#Extract intercept coefficient and standard error
intcp_nop <- coef(summary(fit_act_phys_numofbeds))["(Intercept)", "Estimate"]
intcp_nop_se <- coef(summary(fit_act_phys_numofbeds))["(Intercept)", "Std. Error"]

#Calculate t-statistic for intercept
t_stat_intcp_nop <- intcp_nop / intcp_nop_se

#Calculate df
df_nop <- fit_act_phys_numofbeds$df.residual

#Calculate two-sided p-value
p_val_intcp_nop <- 2 * pt(-abs(t_stat_intcp_nop), df=df_nop)

#Calculate critical t-value for a 95% confidence interval (two-sided)
alpha <- 0.05
t_crit_val_intcp_nop <- qt(1 - alpha / 2, df=df_nop)
```

```r
#Output the t-statistic and p-value
cat("T-statistic for intercept:", t_stat_intcp_nop, "\n")
```

```
## T-statistic for intercept: -3.04605
```

```r
cat("p-value:", p_val_intcp_nop, "\n")
```

```
## p-value: 0.002458724
```

```r
cat("Critical t-value (95% CI):", t_crit_val_intcp_nop, "\n")
```

```
## Critical t-value (95% CI): 1.965395
```

Since the absolute t-statistic ($|-3.04605| = 3.04605$) surpasses the critical t-value (1.965395) and the p-value (0.002458724) is below the 0.05 threshold, we reject the null hypothesis. This indicates that the intercept is statistically significant at the 95% confidence level. The negative t-statistic for the intercept suggests that the expected value of the response variable, the number of physicians, would be negative in a hypothetical scenario where there are zero hospital beds. However, such a scenario is highly unlikely to occur in practice.