Report

The given dataset provides information about houses located in California that can be utilised to predict their prices. It contains a total of eight variables: longitude, latitude, housingMedianAge, totalRooms, totalBedrooms, population, hosueholds, and medianIncome. Furthermore, all these variables are numeric, with no missing values.

Task 1

By considering only medianHouseValue and medianIncome as response and input variables, respectively, the relationship between the two can be examined. To do this, a linear model can be fitted, making three assumptions: there is a linear relationship between input and output, errors are independent of each other, and errors are normally distributed with mean zero and constant variance. Whether these assumptions hold true can be determined by diagnostic plots of the linear model, `lm(medianHousValue ~ medianIncome, data=train)`. First, a residual plot can be used to examine the assumptions whether the errors are independent and normally distributed with mean zero and constant variance. As shown in Figure 1, residuals are clustered around zero, although they are not perfectly centred at zero. However, as there are no obvious patterns, the plot indicates that there are no problems with the model fit. The model can also be checked if it has constant variance by examining the shape of the plot, which thins out towards the edge of the range, meaning there are fewer chances for large residuals to occur near the edges. However, this does not guarantee that the model satisfies the assumptions. Another way to examine the errors is to use a Q-Q plot, which assesses whether the residuals are normally distributed. If the points in the Q-Q plot lie in a straight line, it represents that the residuals are normally distributed. The Q-Q plot in Figure 1 shows an S-shaped line, which indicates that the residuals are not normally distributed. Additionally, we can visualise the relationship between medianHousValue and medianIncome by creating a scatter plot. The plot shows that as medianIncome increases, medianHouseValue also increases, although the relationship does not seem to be perfectly linear.

In summary, the data does not fit well as a linear model due to the non-constant variance and non-normal distributed errors.

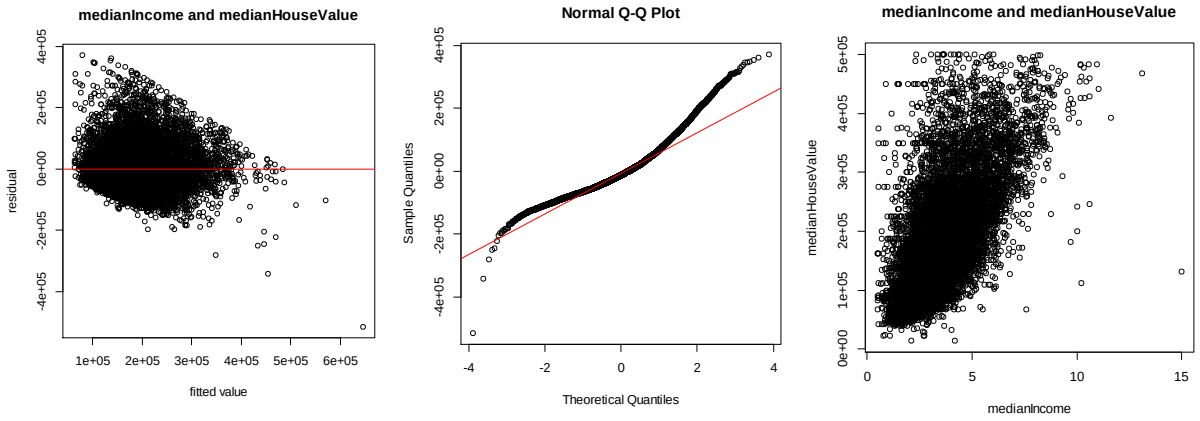

Figure 1 medianIncome and medianHouseValue

To analyse the model further, the summary(m) function can be used. This function provides valuable information about the model, such as coefficients of the linear model, R-squared value, adjusted R-squared value, and p-value. Using the previously obtained values, the 95% confidence interval of the intercept can be determined. The following code calculates the confidence interval:

```
alpha <- 0.05
n <- nrow(train)
p <- 1
t <- qt(1-alpha/2, n-p-1)
cat("[", 43720.2-1919.1*t, ", ", 43720.2+1919.1*t, "]₩n", sep="")
```

With an estimate of the intercept being 43720.2 and the standard error being 1919.1, the 95% confidence interval of the intercept can be calculated as [39958.33, 47482.13]

Task 2

Now, a linear model can be fitted by considering all eight variables by lm(medianHouseValue ~ ., data=train). As done in task 1, the model can be examined by visualising with a residual plot and Q-Q plot and improved further. Figure 2 shows that the residual plot indicates errors are approximately centred at zero, but it does not demonstrate a constant variance. The Q-Q plot displays a straight line with lighter tails. Compared to the model with only one variable, this model shows more independent and normally distributed errors, indicating a better fit.

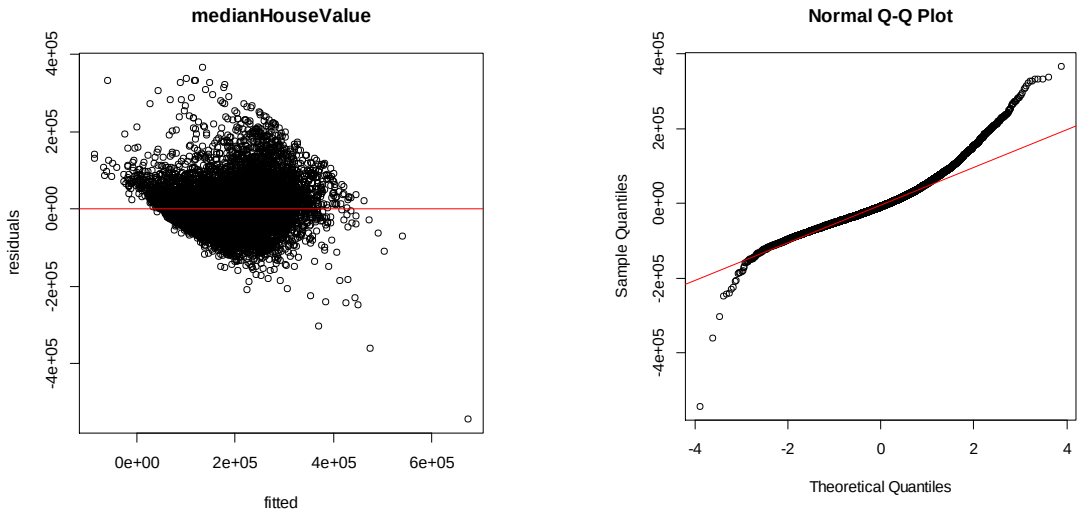

Figure 2 medianHouseValue

One of the methods to improve the model is through transformation. By transforming the data, the variance can be adjusted to be more constant. As illustrated in Figure 2, the variance increases as the value of medianHousevalus increases, which means it needs to be stabilised. Since the assumption of constant variance is violated, transforming the data can improve the model. However, as transformations can affect the mean, it is necessary to verify the model fit for the transformed data through visualisations. In this regard, three different transformations are applied to find the best model fit. First, the transformation of $y' = \sqrt{y}$ is applied. By fitting a model, `lm(sqrt(medianHouseValue) ~ ., data=train)`, an improved model can be presented. As shown in Figure 3, the plots illustrate the range of variance is smaller, and the errors are more generally distributed after transformation.

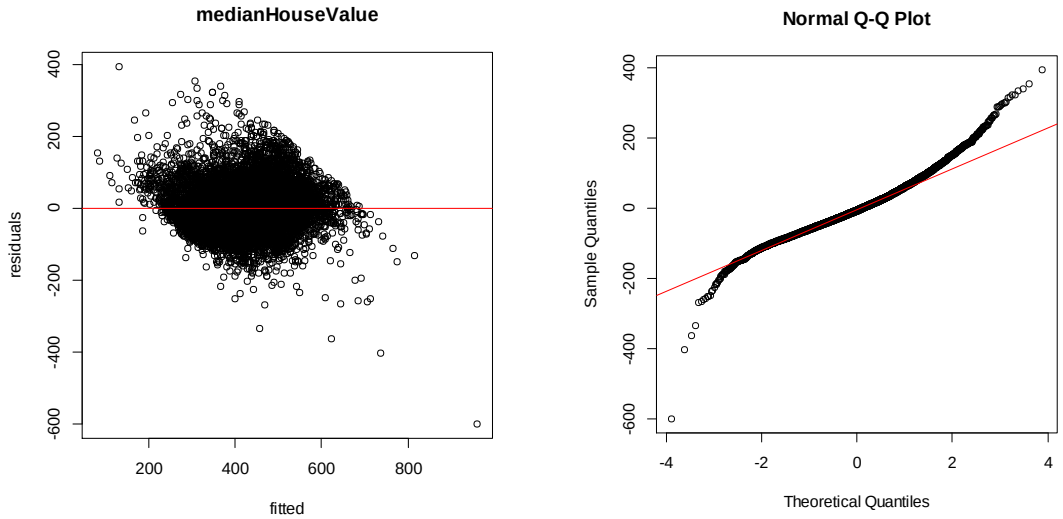

Figure 3 Transformation 1

Second, the transformation of $y' = \log(y)$ is implemented. Figure 4 illustrates a better model fit with `lm(log(medianHouseValue) ~ ., data=train)`. The plots exhibit a smaller range of variance and errors

that are closer to the straight line in comparison to the previous transformation, indicating a better model fit.

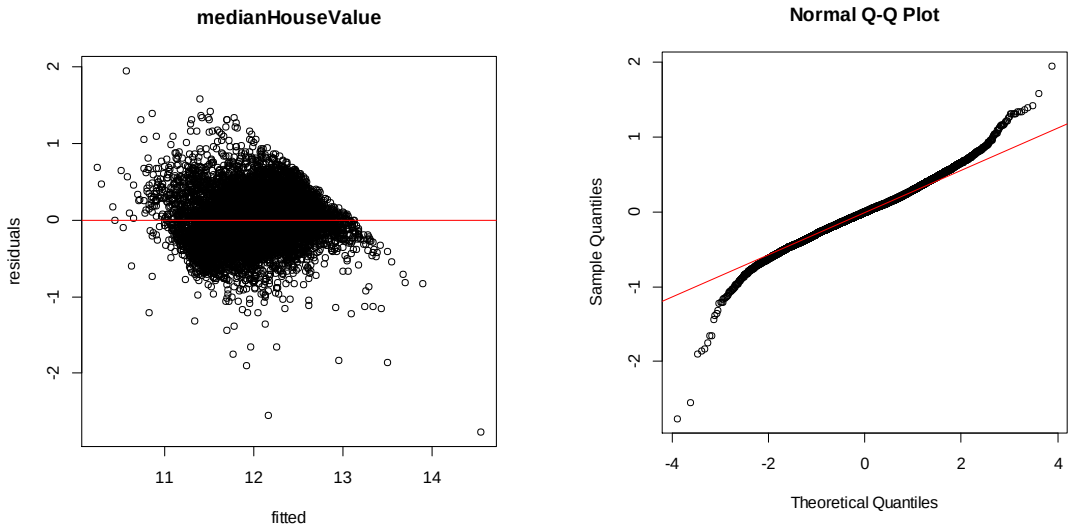

Figure 4 Transformation 2

Finally, a power transformation is applied to the dataset to improve model fit.

```
y <- train$medianHouseValue
gm <- exp(mean(log(y)))
lambda <- seq(0.1, 1, length.out=101)
rss <- numeric(length(lambda))
for (i in seq_along(lambda)){
    li <- lambda[i]
    y.prime <- (y^li-1)/(li*gm^(li-1))
    mi <- lm(y.prime ~ longitude + latitude + housingMedianAge + totalRooms + totalBedrooms
                + households + medianIncome, data=train)
    rss[i] <- sum(resid(mi)^2)
}
plot(lambda, rss, type='l')

n <- nrow(train)
p <- ncol(train)-1
cutoff <- min(rss)*(1+qt(0.975, n-p-1)^2/(n-p-1))
abline(h=cutoff)
range(lambda[which(rss<=cutoff)])
```

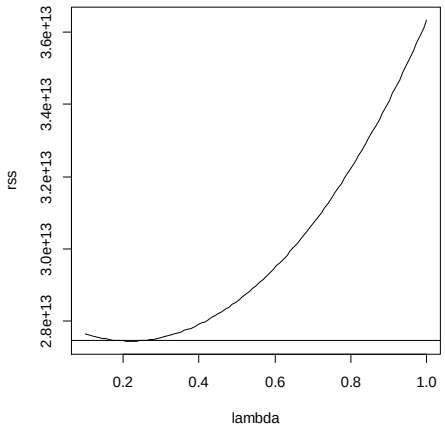

Figure 5 Transformation 3_1

The code above helps in finding the lambda value for the best model using a graph, as displayed in Figure 5. The cod also provides the range of suggested lambda values, [0.199, 0.244], based on the cutoff. Hence, the lambda value can be chosen as 0.22, which is

well within the reasonable range.

```
lambda <- 0.22
y.prime <- (y^lambda-1)/(lambda*gm^(lambda-1))
```

Once the lambda value is determined, the power transformation function can be used to define the transformed y values, as indicated in the code above. After defining these values, the model can be established using the linear regression equation `lm(y.prime ~ longitude + latitude + housingMedianAge + totalRooms + totalBedrooms + households + medianIncome, data=train)`. Compared to the other two transformations previously done, Figure 6 shows a smaller variance range and errors that are closer to the straight line.

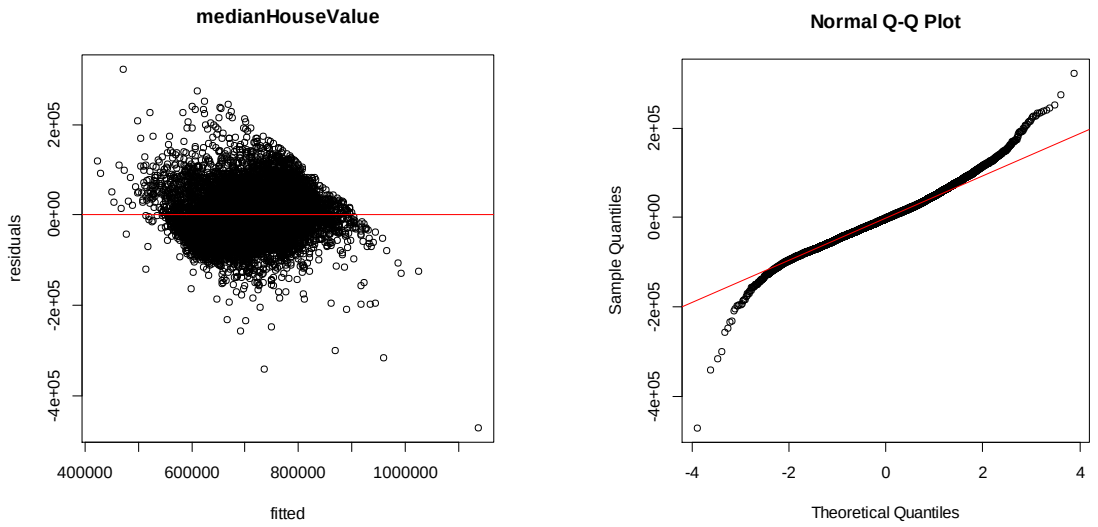

Figure 6 Transformation 3_2

When evaluating a model fit, it is important to look at both visual representations like residual plots and Q-Q plots and the adjusted R-squared values. While the R-squared value increases as the number of input variables increases, the adjusted R-square value can be used as a better measure of model fit. In general, a higher adjusted R-squared value indicates a better model fit. To find the adjusted R-squared values for each transformed model, summary(m) can be used, which yields values of 0.6357, 0.6379, and 0.6409, respectively.

Based on the residual plot, Q-Q plot, and the adjusted R-squared value, the model with power transformation appears to be a good model. The residual plot shows that the residuals are clustered around zero with a smaller range of variance, while the Q-Q plot indicates that the points are approximately aligned on a straight line.

As a further analysis, multicollinearity can be checked, which is present when an independent variable is highly correlated with one or more other independent variables. The following code detects multicollinearity in the model:

```
X <- model.matrix(y.prime ~ longitude + latitude + housingMedianAge + totalRooms
                  + totalBedrooms + households + medianIncome, data=train
s <- svd(X, nu=0)
s$d[1]/s$d[length(s$d)]
round(s$v[, 9], 3)
```

If the condition number is greater than 30, the model is considered to have multicollinearity. Additionally, the singular vectors of X calculated as 1.000, 0.011, 0.009, 0.000, 0.000, 0.000, 0.000, 0.000, and 0.002 suggest that the variables, longitude and latitude significantly impact the model. By removing each variable to observe their impacts on the adjusted R-squared, it is shown that removing either the longitude or latitude variable results in a reduction of about 20% in the adjusted R-squared, while removing other variables causes a decrease of less than 1%. Therefore, it is necessary to include the variables longitude and latitude in the model.

Then, automatic model selection can be applied when dealing with a model that has many variables. For instance, a model with eight variables has 2^8=256 different choices for which variables to include in the model. As the number of variables increases, it becomes impractical to try all possible models to find the best one. Therefore, an automatic model selection algorithm can be used instead. This algorithm automatically selects a model with the largest adjusted R-squared value. To illustrate, consider the following code:

```
m <- regsubsets(y.prime ~ longitude + latitude + housingMedianAge + totalRooms
        + totalBedrooms + households + medianIncome, data=train, method="exhaustive", nvmax=8)
round(summary(m)$adjr2, 4)
s <- summary(m)
s$which[which.max(s$adjr2), ]
```

The code above gives the adjusted R-squared values of models with different numbers of variables included. In this case, the adjusted R-squared values are 0.4172, 0.4495, 0.5963, 0.6013, 0.6297, 0.6361, 0.6395, and 0.6409 in order. Thus, it can be concluded that the model with all eight variables included has the highest adjusted R-squared value, which is 0.6409. Additionally, the last code, `s$which[which.max(s$adjr2), ]`, shows which variables should be included for the best model by indicating TRUE and FALSE for each variable. In this case, all eight variables are indicated by TRUE.

In conclusion, the best model derived from the process is the power-transformed model that includes all eight variables. This model gives a linear equation of $-2945000-41840\,x_1-42090x_2+673.1x_3-8.271x_4+85.15x_5-30.54x_6+54.48x_7+35060x_8$,eightwhere variables used in this equations are longitude, latitude, housingMedianAge, totalRooms, totalBedrooms, population, households, and medianIncome respectively.

Task 3

The model defined above can be used to predict medianHouseValue in the test dataset. The following code predicts the medianHouseValue of the test dataset by training the model with the training dataset.

```
m <- lm(y.prime ~ longitude + latitude + housingMedianAge + totalRooms + totalBedrooms
        + households + medianIncome, data=train)
test_actual = test$medianHouseValue
test_preds = predict(m, test)
mse(test_actual, test_preds)
```

The adjusted R-squared value of this model is 0.6409, and the mean squared error value is 273070576362. The MSE represents the average squared difference between actual and predicted house prices, which is $273070576362. Furthermore, the adjusted R-squared value indicates that the variables in the dataset can explain 64.09% of the variation. Although the result is still unsatisfactory, the model has been improved through various methods, resulting in a significant increase in the adjusted R-squared value.