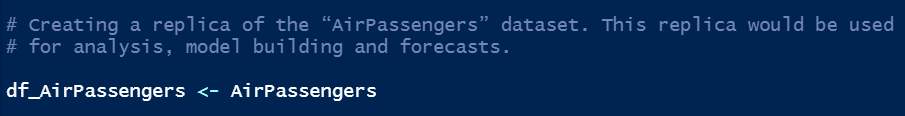
**1. TIME SERIES ANALYSIS**

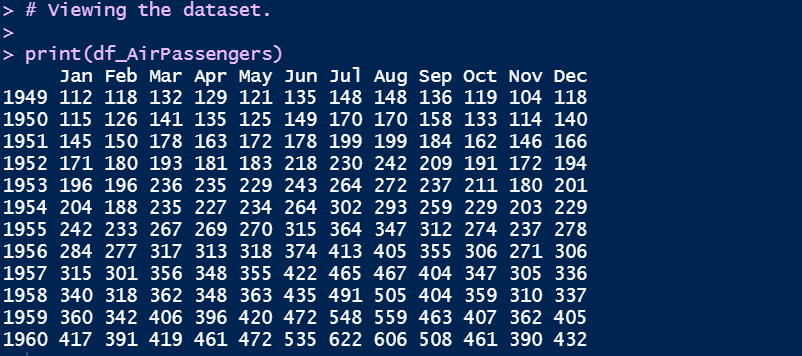
The objective of this section is to analyse the R Studio’s inbuilt dataset “AirPassengers” and build Time Series Models.

**Preliminary analysis**

Before model building, we need to perform preliminary analysis to understand the data.

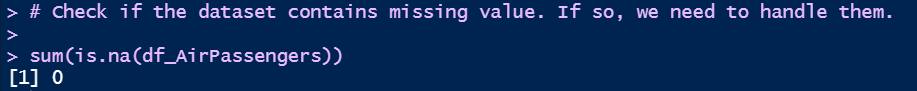


**Viewing the dataset.**



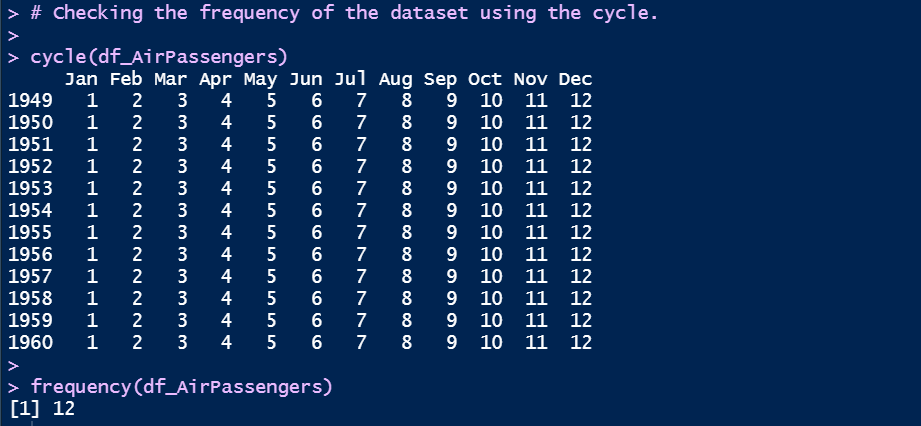
The data starts from Jan in the year 1949 and ends in the year 1960 in the month of December.

**Check if the dataset contains missing value.** If so, we need to handle them.

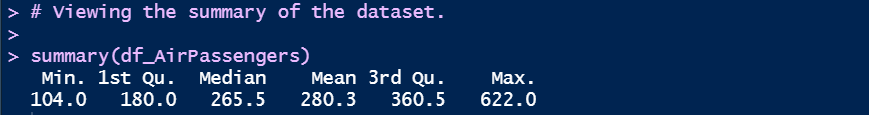


There are no missing values in the dataset.

**Checking the frequency of the dataset using the cycle.**

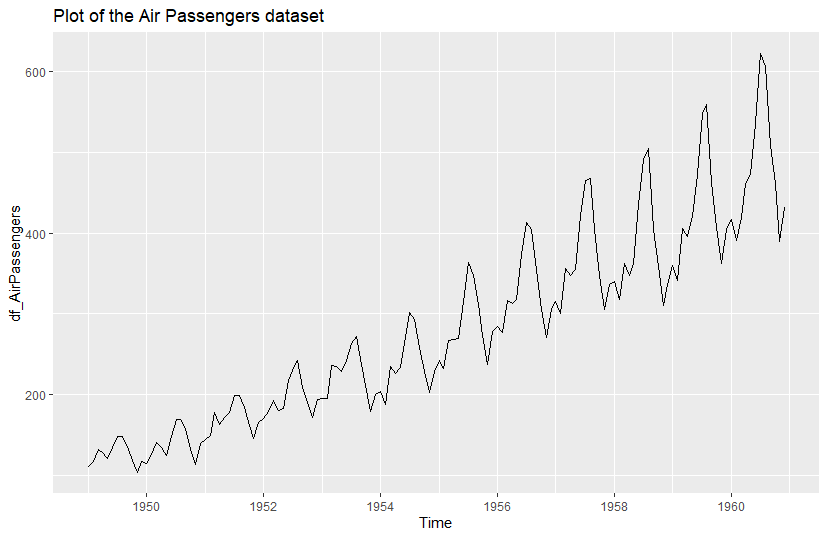


**Viewing the summary of the dataset.**



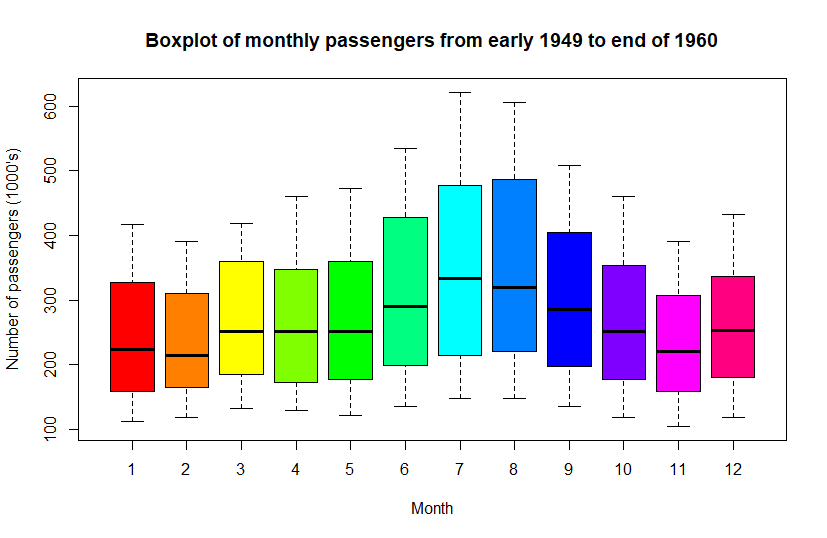
The minimum number of passengers is 104 and the maximum is 622.

**Visualising the data using autoplot () function.**



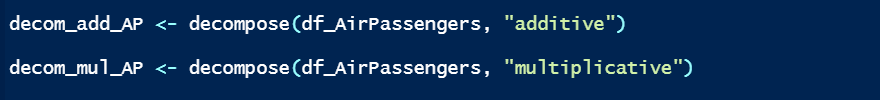
From the plot, we can infer that the number of passengers increases over time for each year. There appears to be an upward linear trend owing to the high demand for air travel. This dataset seems to be a multiplicative time series due to the increase in seasonality over the years.

**Visualising the data using a boxplot to check for seasonal impacts.**

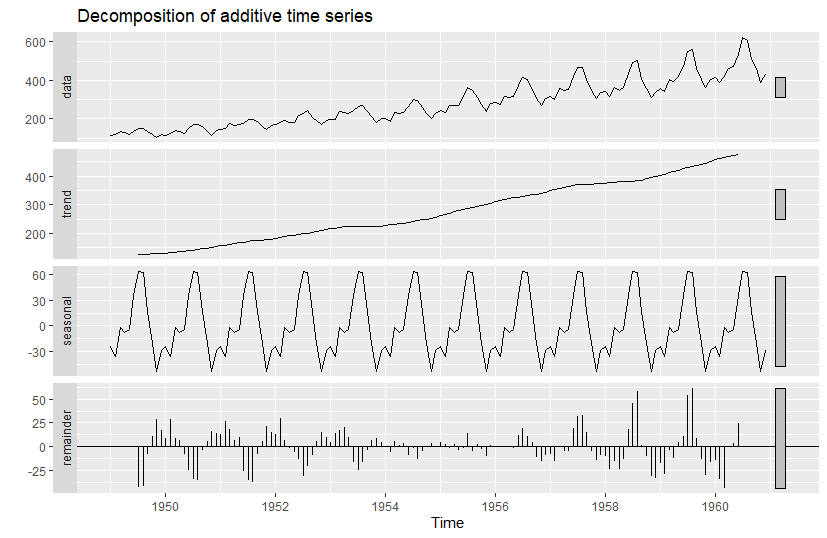


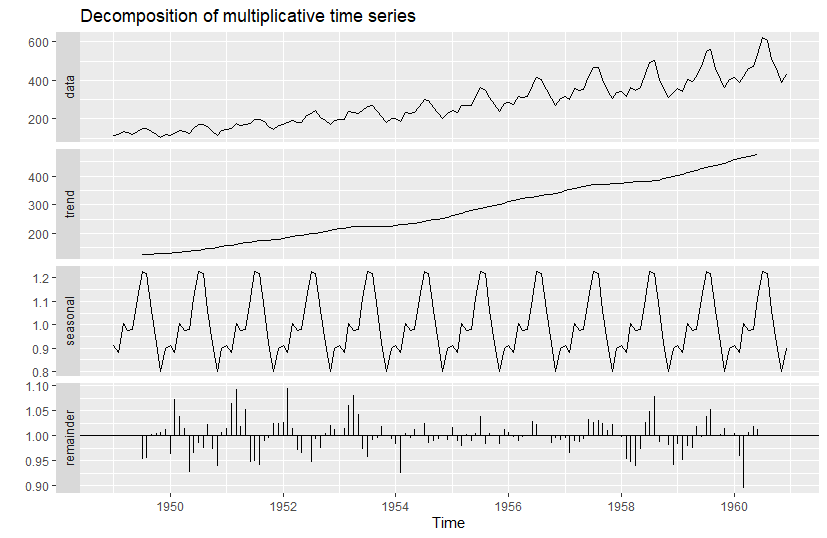
From the boxplot we can infer that the number of passengers is quite high in the months of June, July, August, and September. These observations possess the highest variability in the data. This stipulates that there is a seasonality over a cycle of 12 months (1 year). This also means that most of the people travel by air to other places during the summer season to enjoy their vacation. There are no outliers or missing values in the dataset which means that there is no need for data cleaning.

**Decomposing the dataset into additive and multiplicative components.**

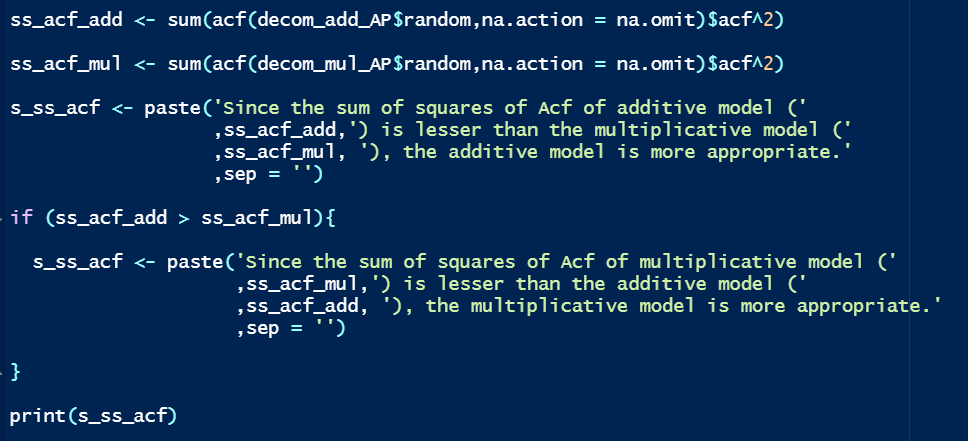


**Visualizing the additive and multiplicative components.**



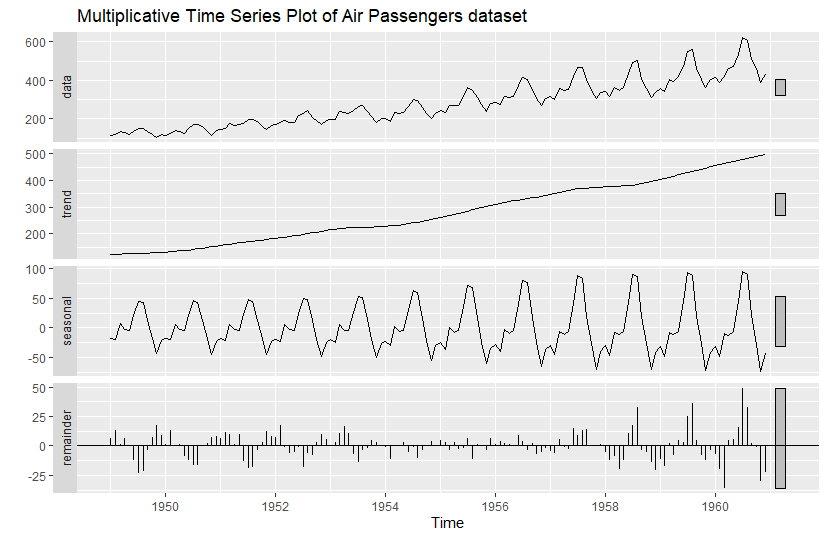


From the decomposed plots we can observe the linear increase in trend with the seasonality. However, it is difficult to assess the decomposition type just by observing the plot of the components. We need to find the correlation between the data and the residuals (random component). This can be done by applying the acf () function on the random components. Since there could be negative values, we find the sum of squares of the acf from both the models. The model type with the least value is the more appropriate.



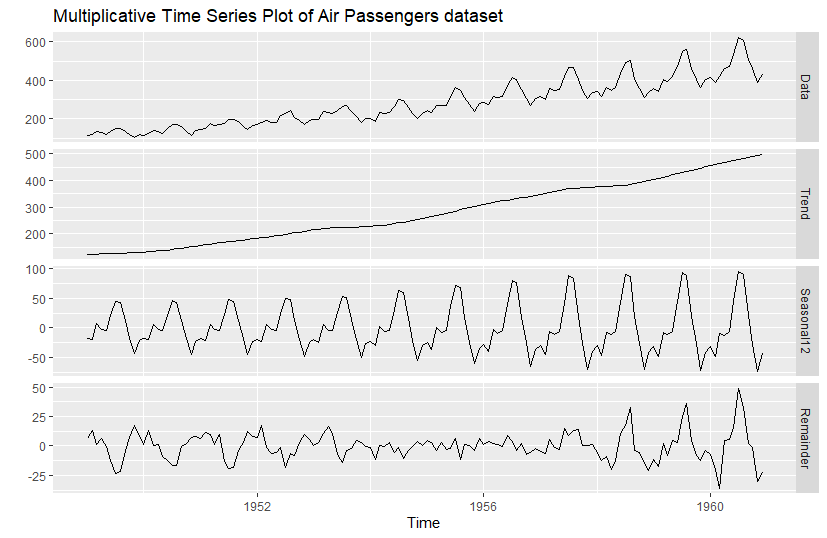
Since the sum of squares of Acf of multiplicative model (1.918) is lesser than the additive model (4.0847), the multiplicative model is more appropriate.

A time series plot using stl () function is created to understand which component has the highest impact on it.



From the plot, we can see that the trend component has the highest impact on the time series since it has the highest variability among all the components.

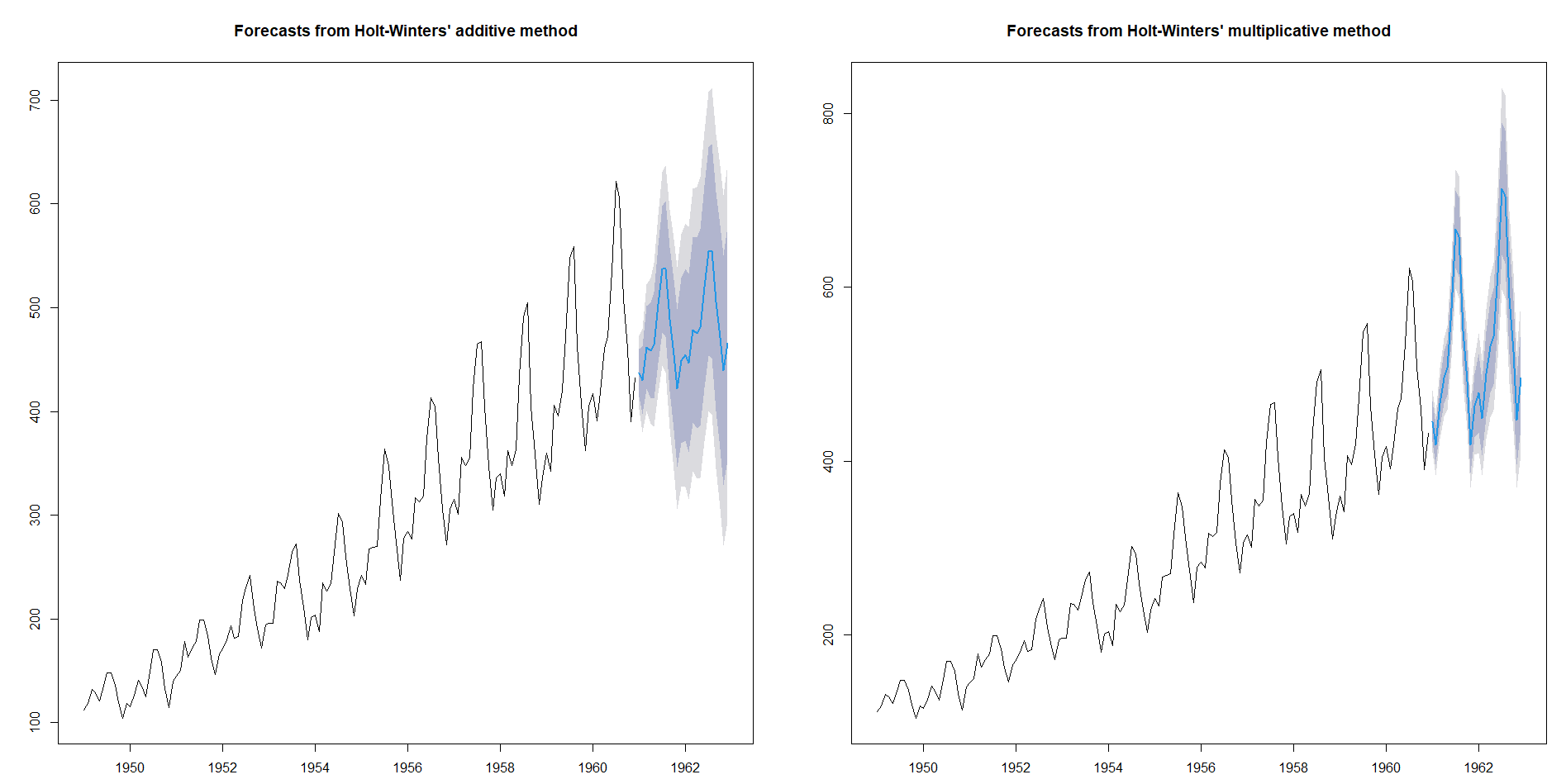
Too much variability in trend will impact future predictions and it must be addressed. This can be done by first applying log transformations on the dataset to reduce the variability and then taking the first difference of the log transformed values. This will remove the variability and trend.



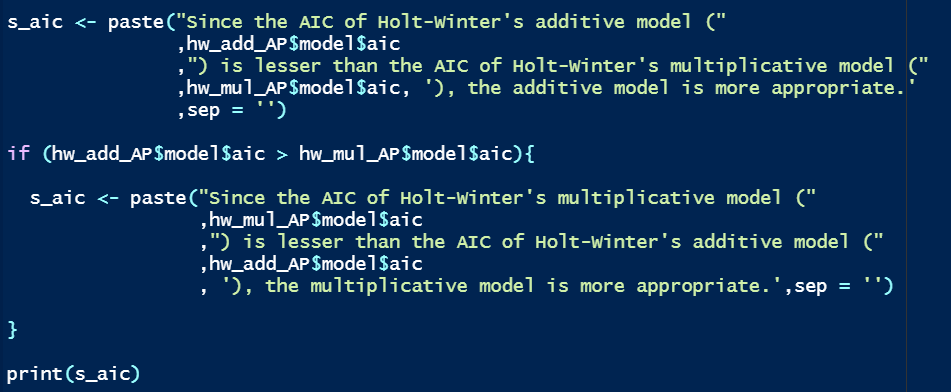
The Periodicity is 12 from the label of the Seasonal component.

**Time series modelling**

Simple Exponential Smoothing (SES) is used for time series that have neither trend nor seasonality. Double Exponential Smoothing is used for time series that have trend but no seasonal component. Triple Exponential Smoothing is used for time series that have both the trend and seasonal components. Since the time series plot of Air Passengers dataset has both the components, we can implement Triple Exponential Smoothing models such as Holt-Winter's model.



The forecast from the multiplicative model appears to be better than the additive model. However, we need to confirm that using Goodness of Fit (GOF) measures. The GOF of Holt-Winter's model is determined by Akaike’s Information Criterion (AIC), Akaike’s Corrected Information Criterion (AICC) and Bayesian Information Criterion (BIC). When comparing two models using AIC, the model with the lower AIC value is the better model.



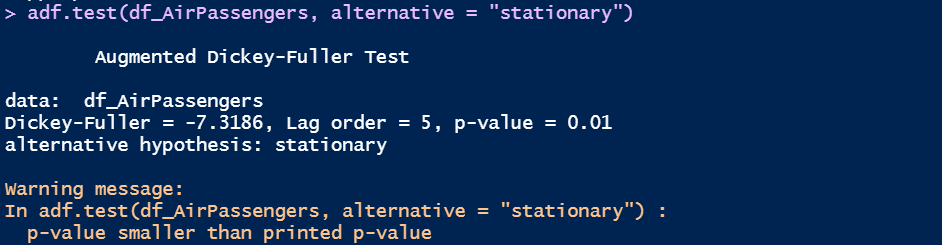
Since the AIC of Holt-Winter's multiplicative model (1405.654) is lesser than the AIC of Holt-Winter's additive model (1565.871), the multiplicative model is more appropriate. This conclusion is consistent with what was expected in the preliminary analysis.

Next, we need to implement Auto Regressive Integrated Moving Average Models (ARIMA). Stationarity is an important pre-requisite for building ARIMA models.

Augmented Dickey-Fuller (ADF) Test can be used for testing the stationarity of a time series.

“H0: Time Series is not stationary.”

“HA: There is stationarity in the time series.”

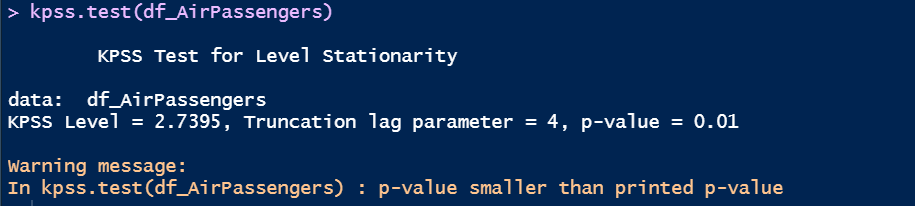


Since p-value (0.01) is less than alpha (0.05), we reject the null hypothesis.

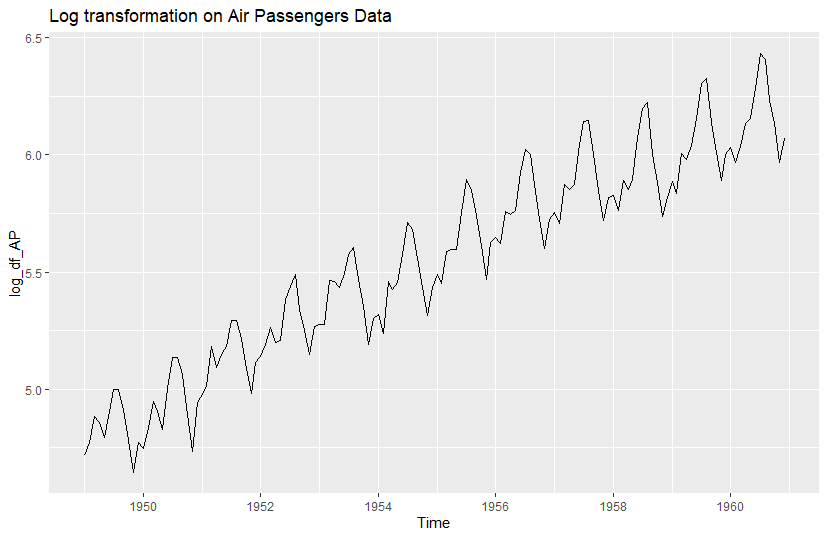
This means that there is no evidence of non-stationarity. However, we need to confirm that using Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. This is test is used for checking the level/trend stationarity of a time series.

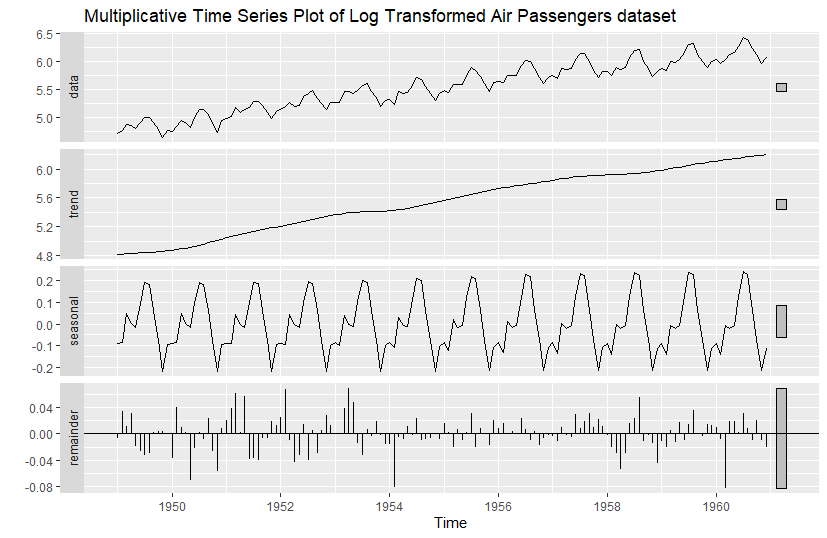
“H0: Time Series is level/trend stationary.”

“HA: Time Series is not stationary.”

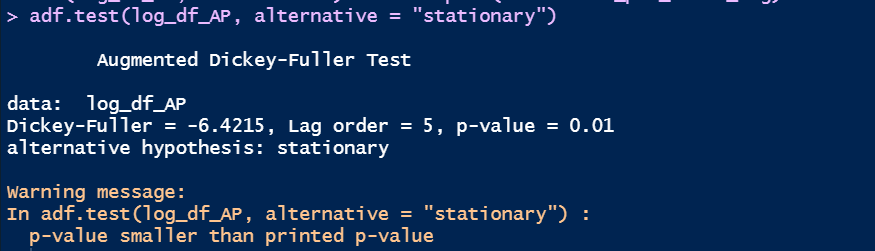


Since p-value (0.01) is less than alpha (0.05), we reject the null hypothesis. This means that there is no evidence of level/trend stationarity. This result is inconsistent from the ADF test result. Hence, we must transform the data to induce stationarity. We can apply log transformations to remove the variability.



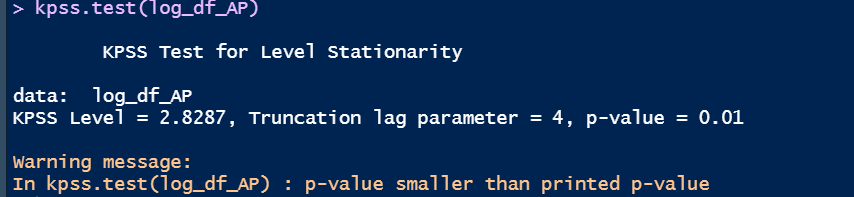


The seasonality appears to have almost constant variability. We perform the ADF and KPSS tests on the log transformed data.



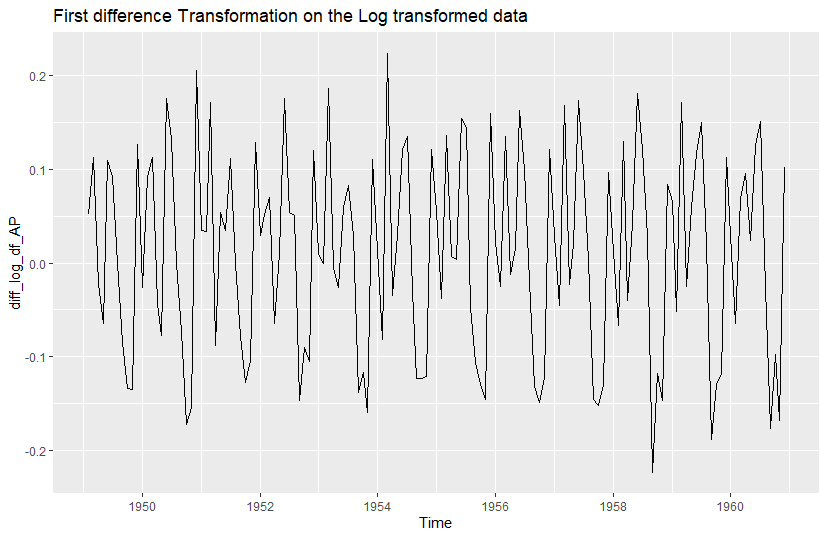
Since p-value (0.01) is less than alpha (0.05), we reject the null hypothesis.

This means that there is no evidence of non-stationarity after log transformation.

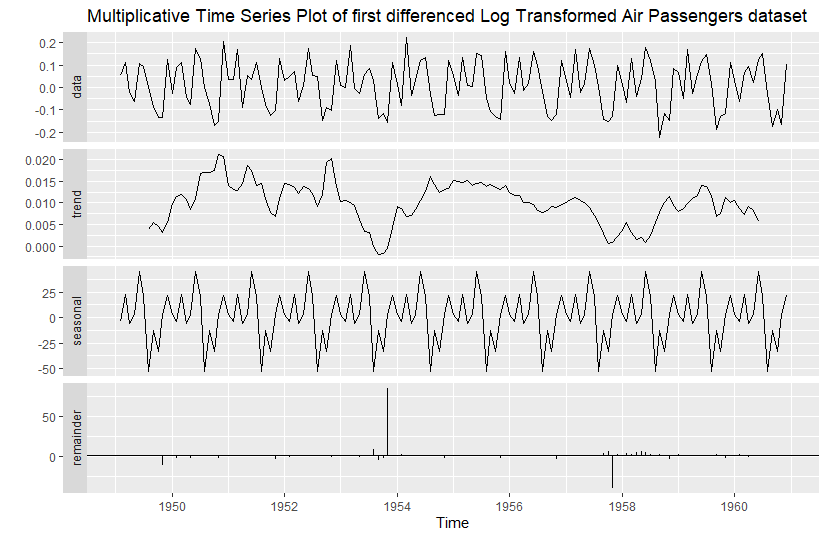


Since p-value (0.01) is less than alpha (0.05), we reject the null hypothesis.

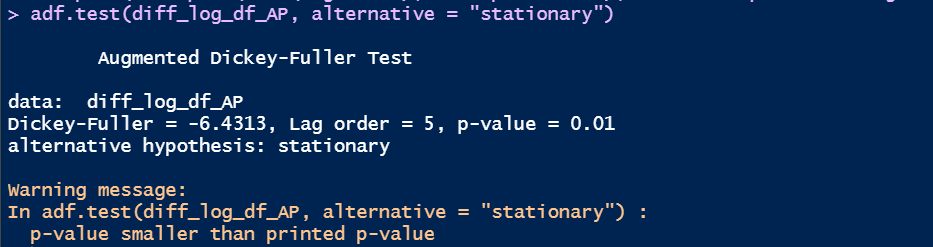
This means that there is no evidence of level/trend stationarity. This result is inconsistent from the ADF test result. Hence, we must transform the data once more to induce stationarity. We apply a first difference transform to remove the trend.



We can see that the plot almost resembles random noise.

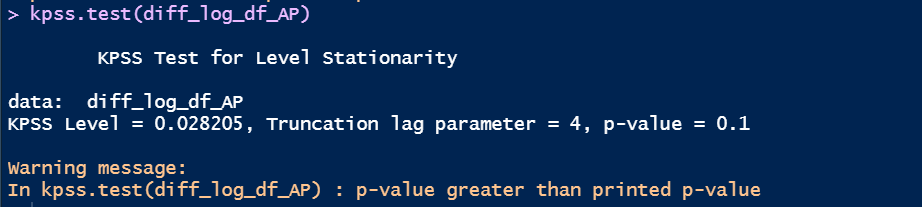


We can see that there is no pattern in the trend and variability in the seasonality has been removed. The time series appears to be stationary. However, we need to perform the ADF and KPSS tests to confirm this. We perform the ADF and KPSS tests on the first differenced log transformed data.



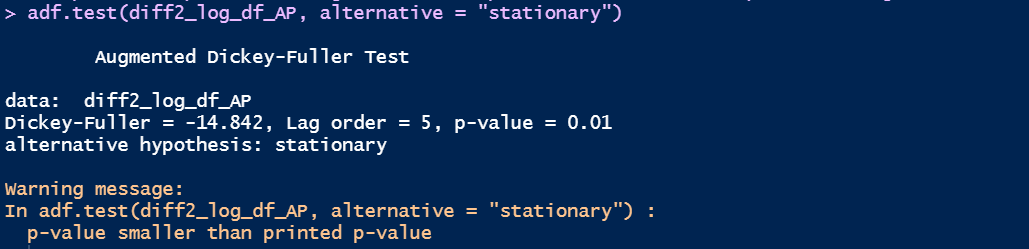
Since p-value (0.01) is less than alpha (0.05), we reject the null hypothesis.

This means that there is no evidence of non-stationarity after first differenced log transformation.



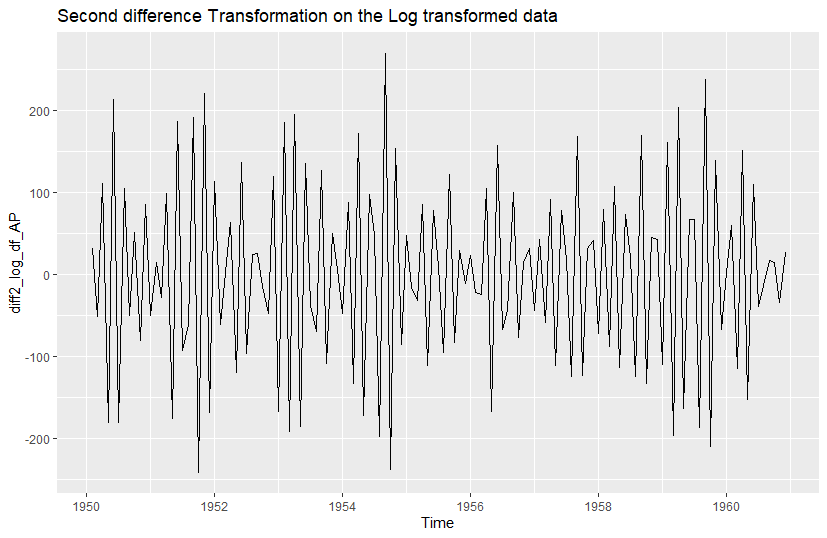
Since p-value (0.1) is more than alpha (0.05), we fail to reject the null hypothesis. This means that there is no evidence to show that the transformed time series is non-stationary. This result is consistent with the ADF test result.

However, the seasonality drastically dominates the time series. Hence, we need to perform second differencing to eliminate this. We apply a second difference transform to remove the seasonality.

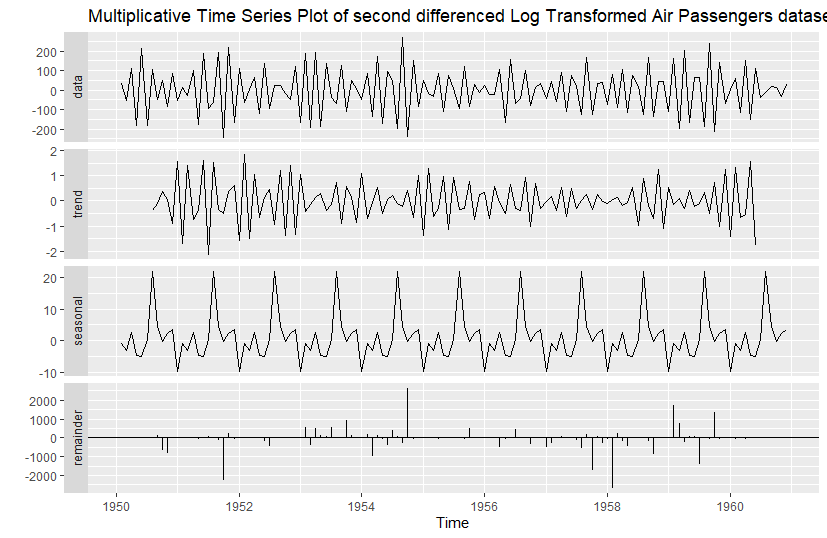


Since p-value (0.01) is less than alpha (0.05), we reject the null hypothesis.

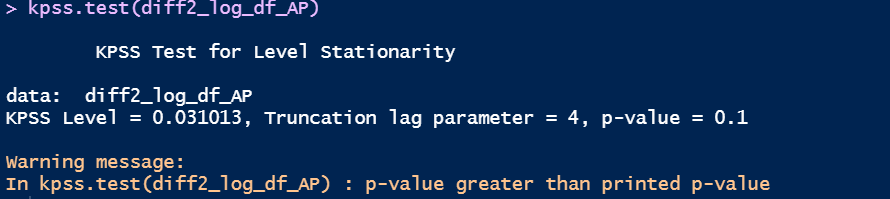
This means that there is no evidence of non-stationarity after second differenced log transformation.



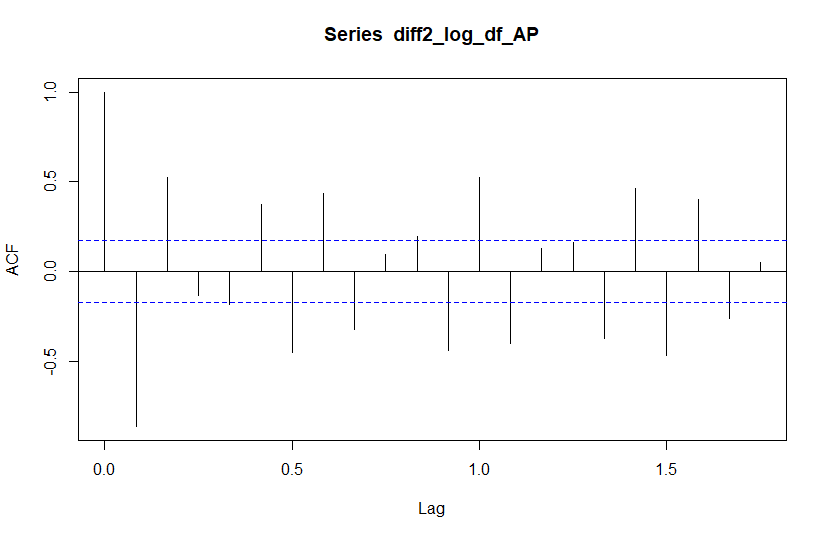
We can see that the plot closely resembles random noise.



We can see that the trend resembles white noise and the time series appears to be almost stationary. However, we need to perform the ADF and KPSS tests to validate this. We perform the ADF and KPSS tests on the second differenced log transformed data.

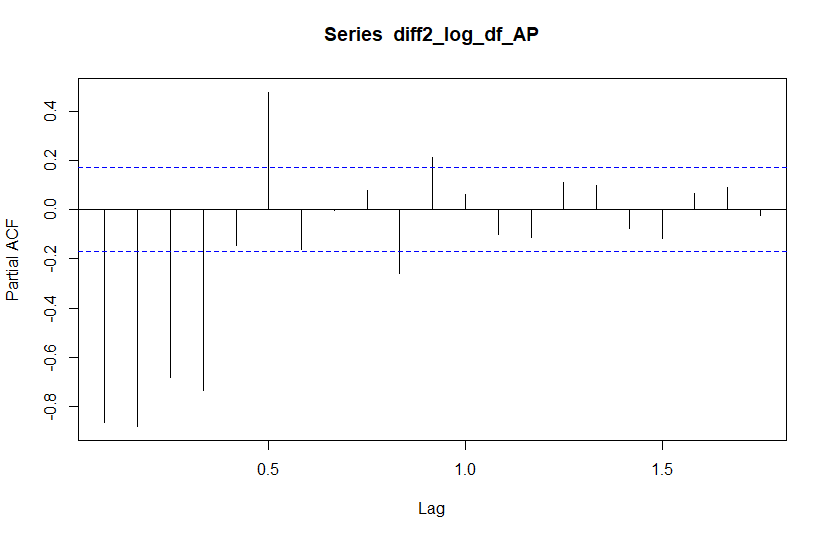


Since p-value (0.1) is more than alpha (0.05), we fail to reject the null hypothesis. This means that there is no evidence to show that the transformed time series is non-stationary. This result is consistent with the ADF test result and the seasonality no longer dominates the time series. The stationarity has been achieved and hence, we can build ARIMA models. We plot the correlograms for analysing the transformed time series.



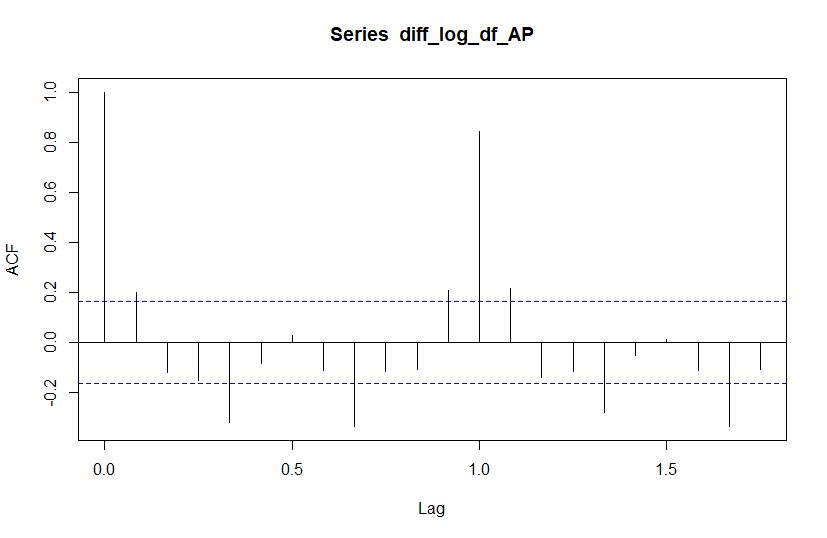
From the ACF plot of the stationary time series having neither trend nor stationarity, we can see that cut off takes places after the first lag.

This means d = 1. There is a positive peak and negative peak after which there is a decreased sinusoidal pattern in the peaks. Hence, we can say p = 2.

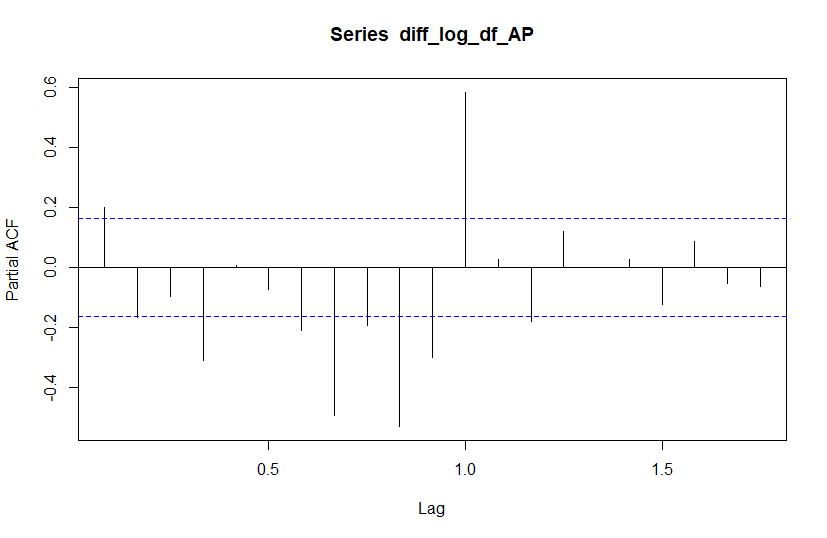


From the PACF plot of the stationary time series having neither trend nor stationarity, we can see that cut off takes places after the first positive lag.

Hence, q = 1.

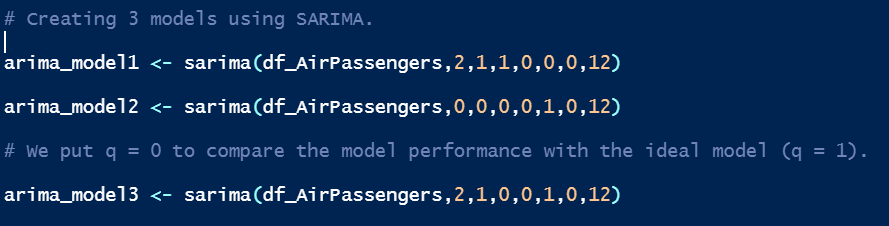


From the ACF plot of the stationary time series having no trend but stationarity, we can see that cut off takes places after the first lag. This means D = 1. There is a sinusoidal pattern which means there is no AR process. Hence, P = 0.

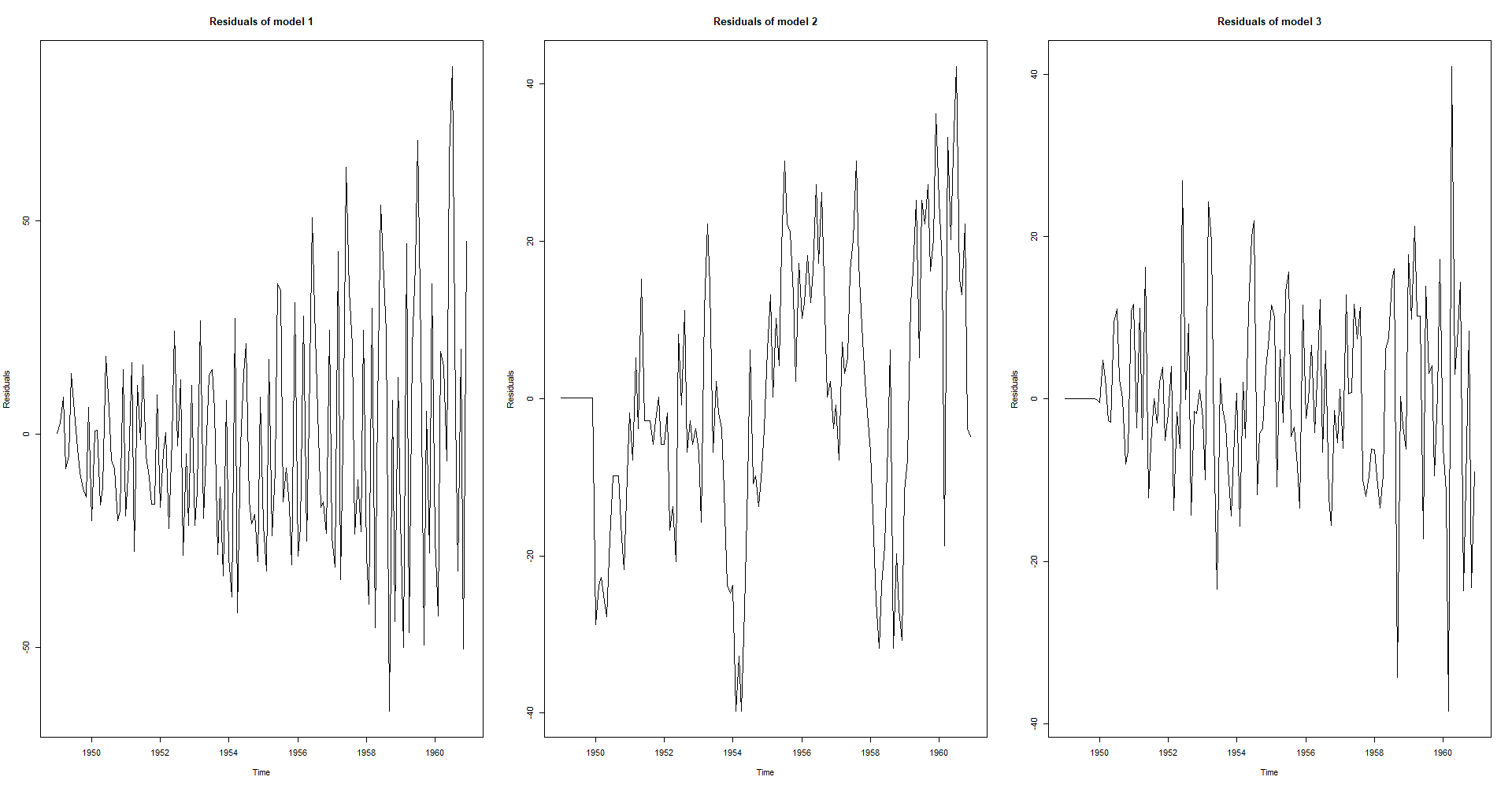


From the PACF plot of the stationary time series having no trend but stationarity, we can see that there is no cut-off. Hence, Q = 0.

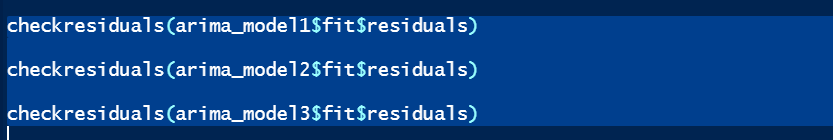
**Creating 3 models using SARIMA.**

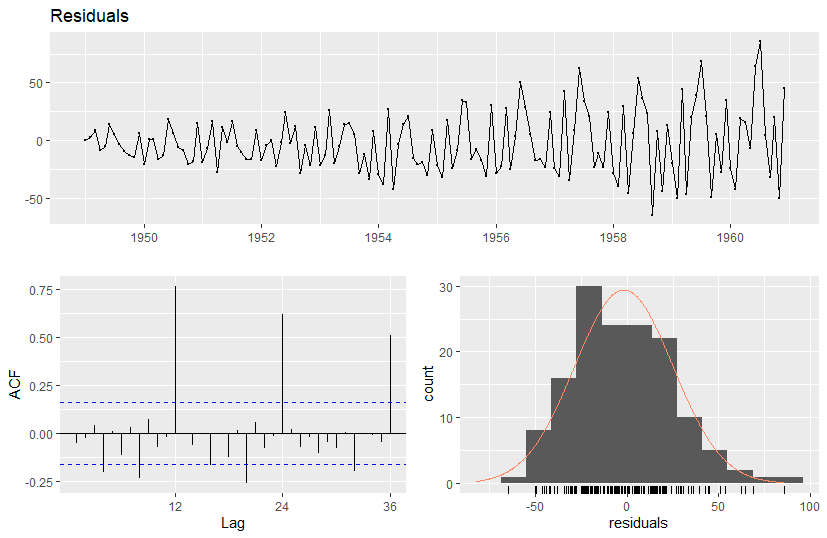


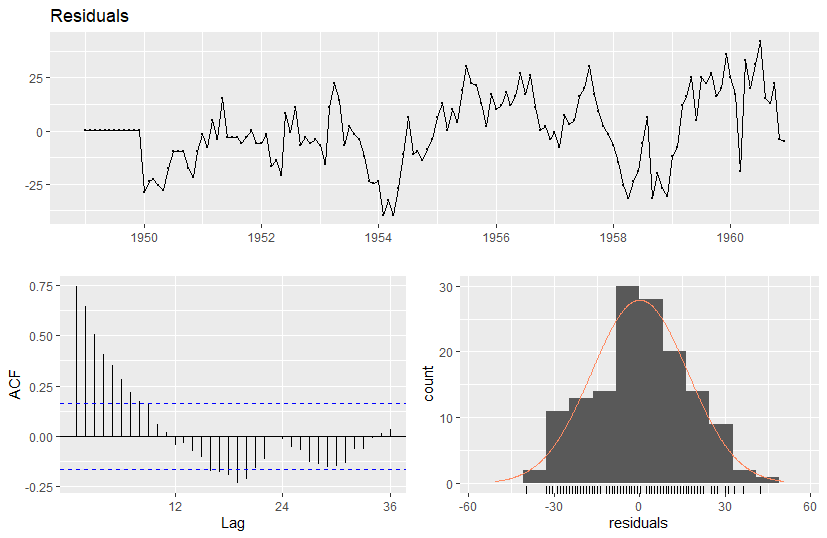
**Plotting the residuals of the models.**

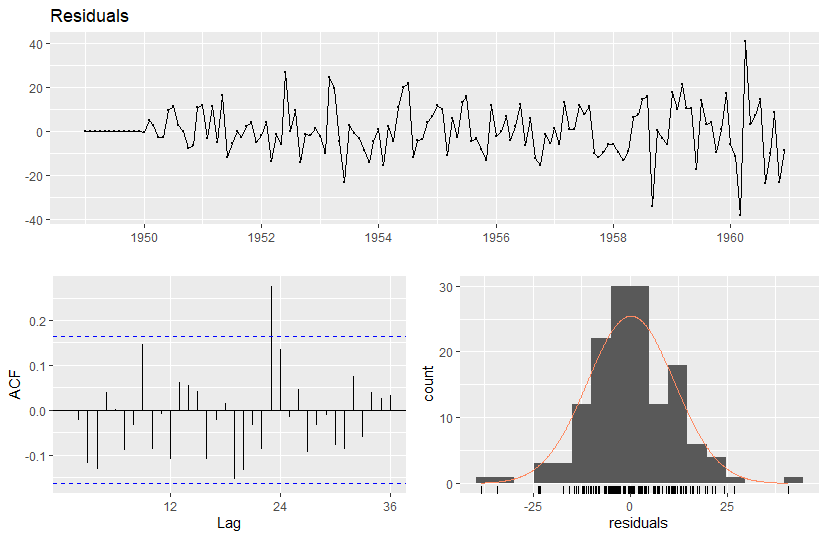


**Checking the normality of the residuals.**





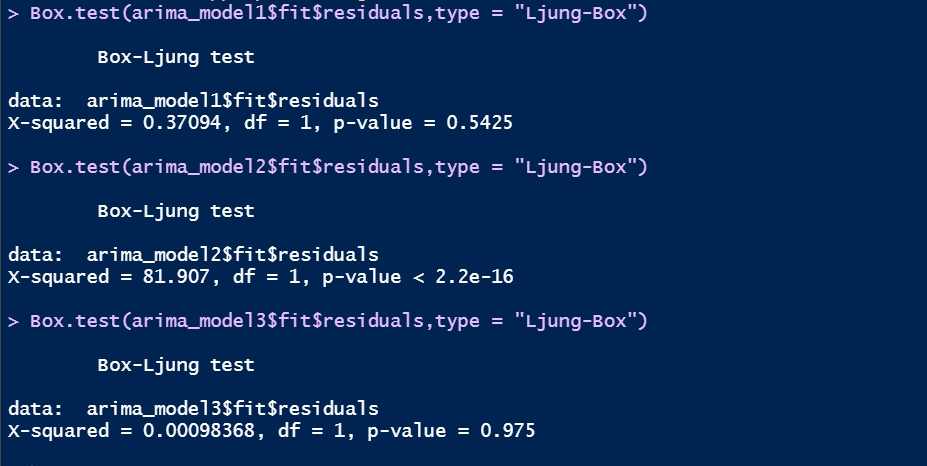




We need to perform the Ljung-Box test on the model residuals.

“H0: The residuals of the time series are independent.”

“HA: The residuals of the time series are correlated.”



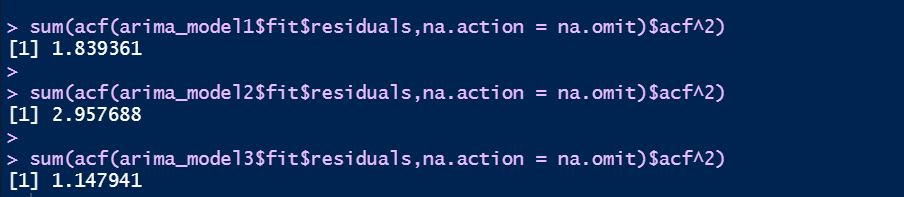
The p-value (2.2e-16) for the second model is less than alpha (0.05).

This means we reject the null hypothesis and can conclude that the residuals are correlated in the second model. In the other models, p-value is significant.

Hence, we can say that there is no evidence of dependence in the residuals.

We need to find the correlation between the data and the residuals. This can be done by applying the acf () function on the random components. Since there could be negative values, we find the sum of squares of the acf from both the models.

The model type with the least value is the more appropriate.



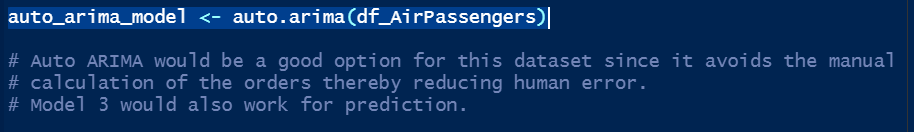
We can see that the model with least Sum of Squares is the third model.

We can also verify this by checking the AIC values. The model with the lowest AIC is the best model.



We can see that the 3rd model has the lowest AIC among all. Hence, it is the best model.

**Finding best ARIMA model using auto.arima function**



Auto ARIMA would be a good option for this dataset since it avoids the manual calculation of the orders thereby reducing human error. Model 3 would also work for prediction. We can compare the classical model with the ARIMA model using AIC as GOF.

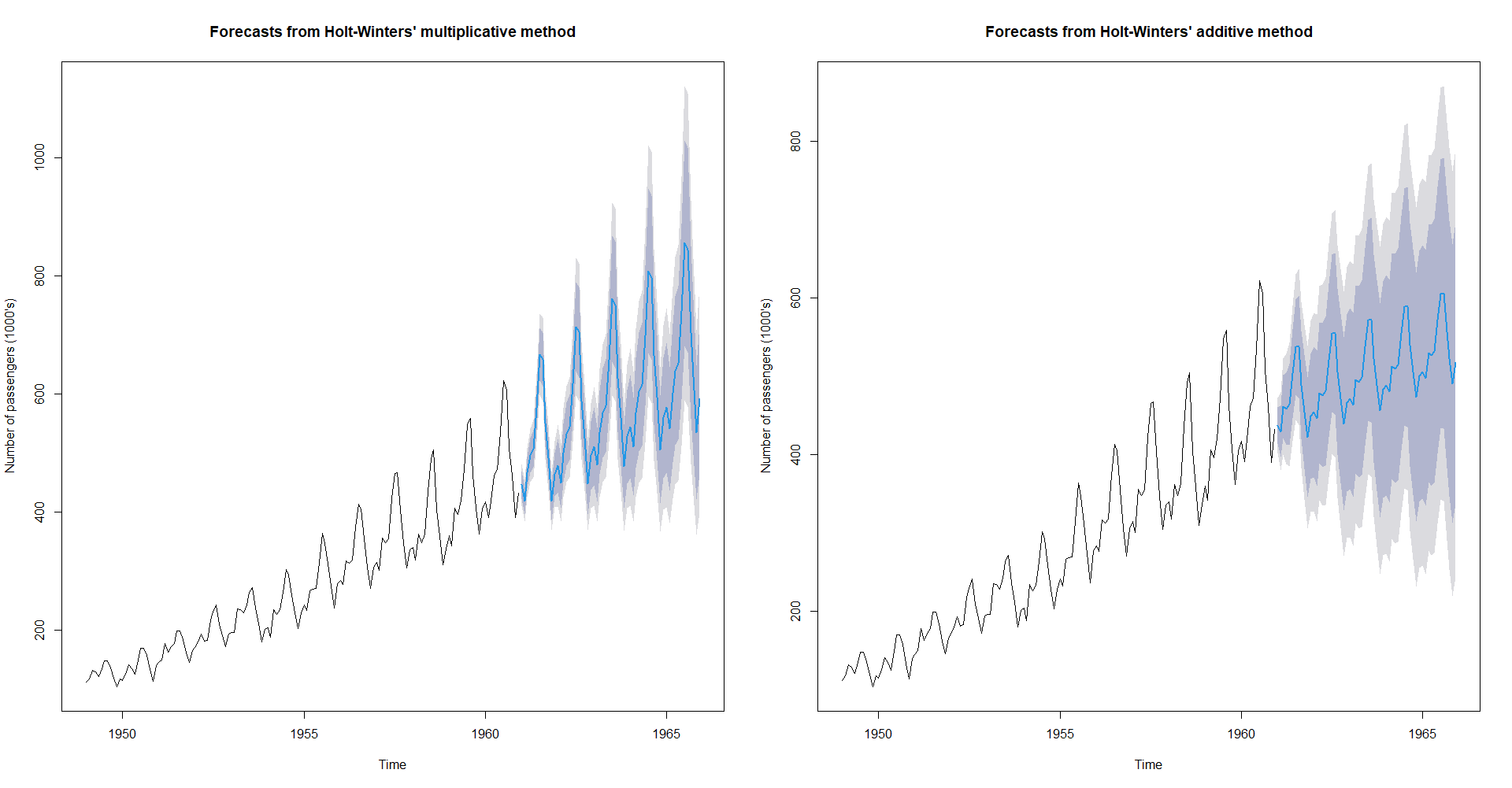


The AIC of both the ARIMA models is less than AIC of classical model (Holt-Winter's). Therefore, the ARIMA model is better for this dataset.

**Forecast**

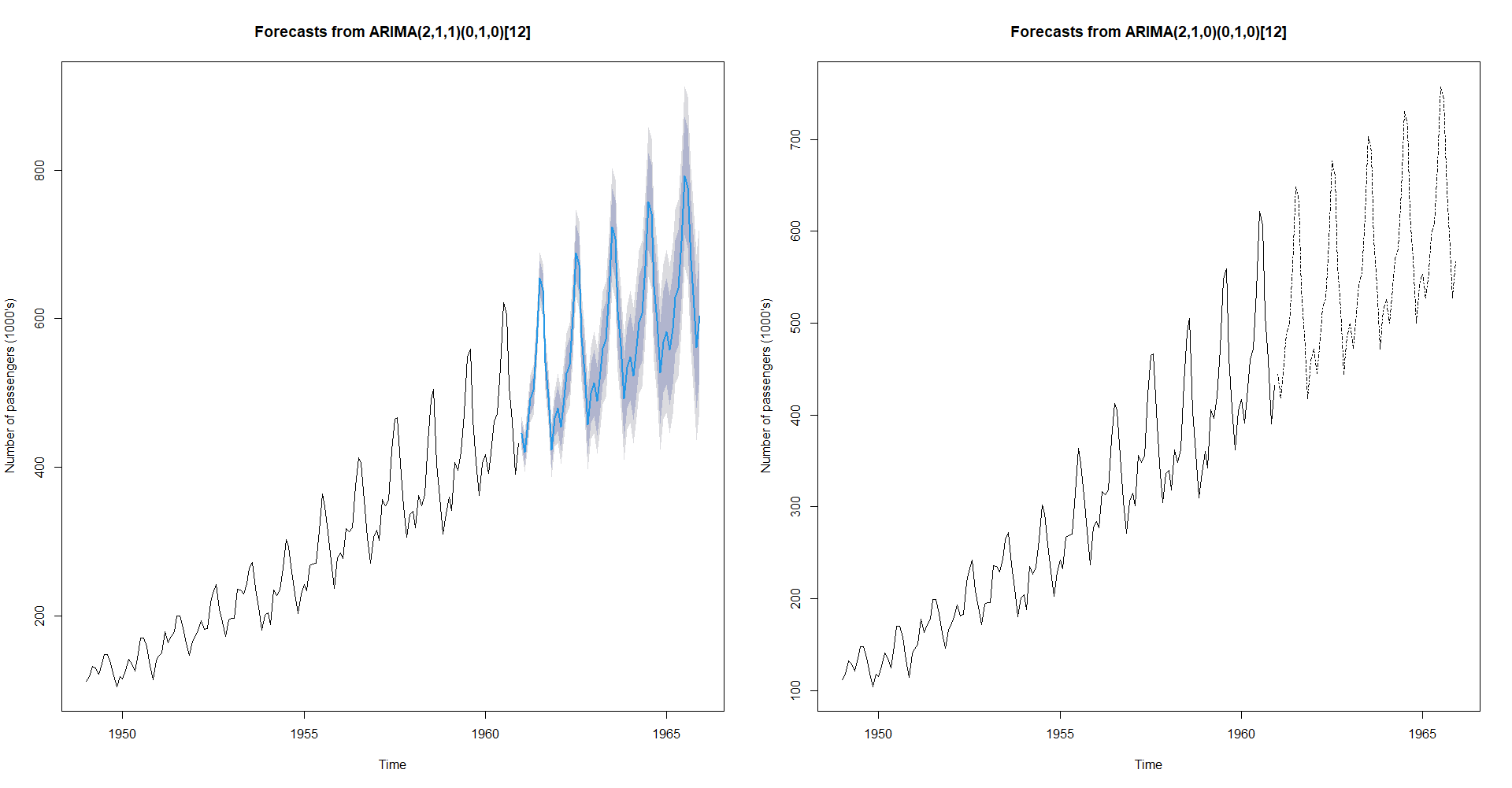
The model building is complete, and we need to proceed with forecasting.

**Forecasting for the next 5 periods using the classical models.**



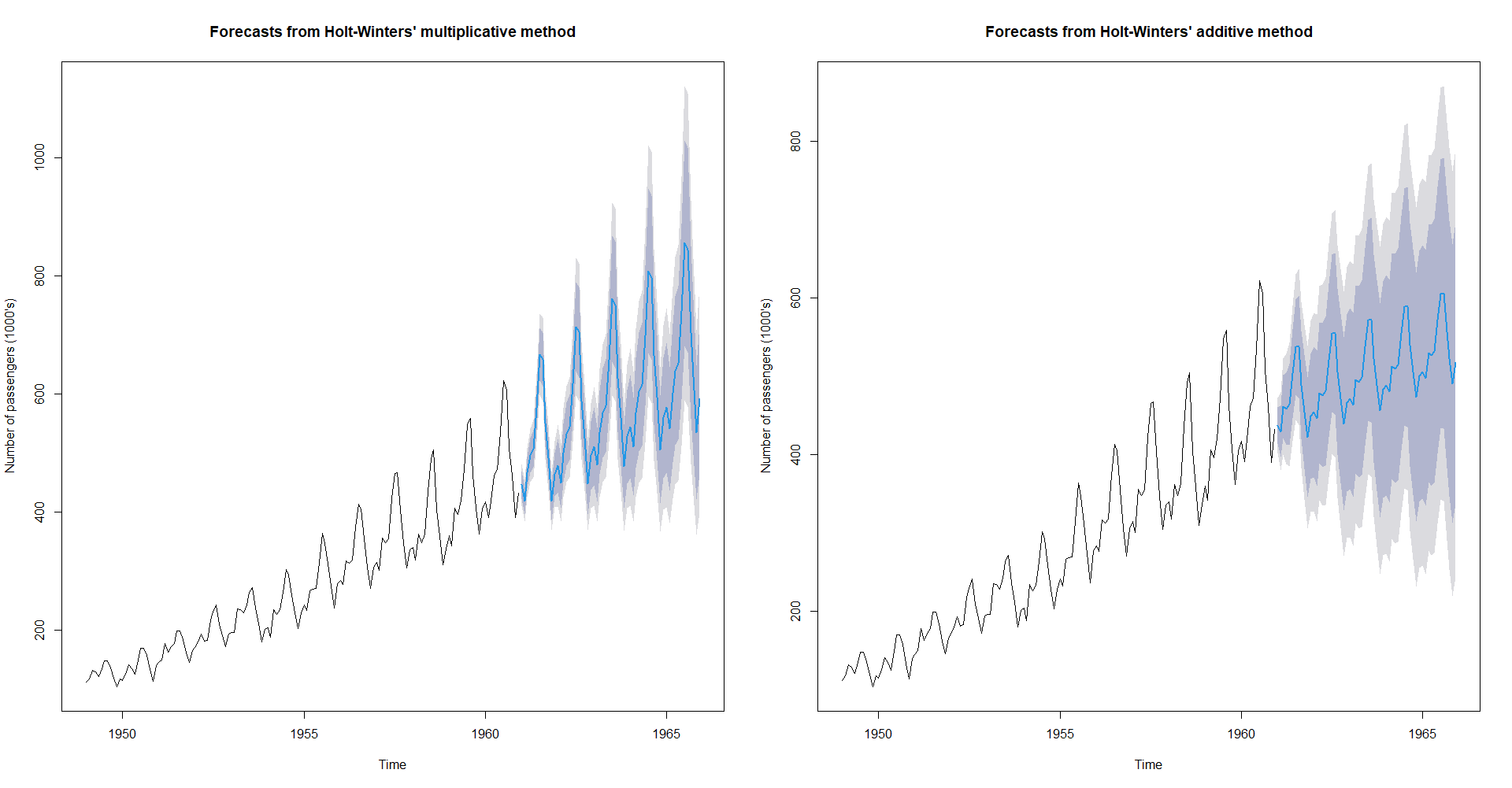
The forecast of the multiplicative model appears better than the forecast of the additive model.

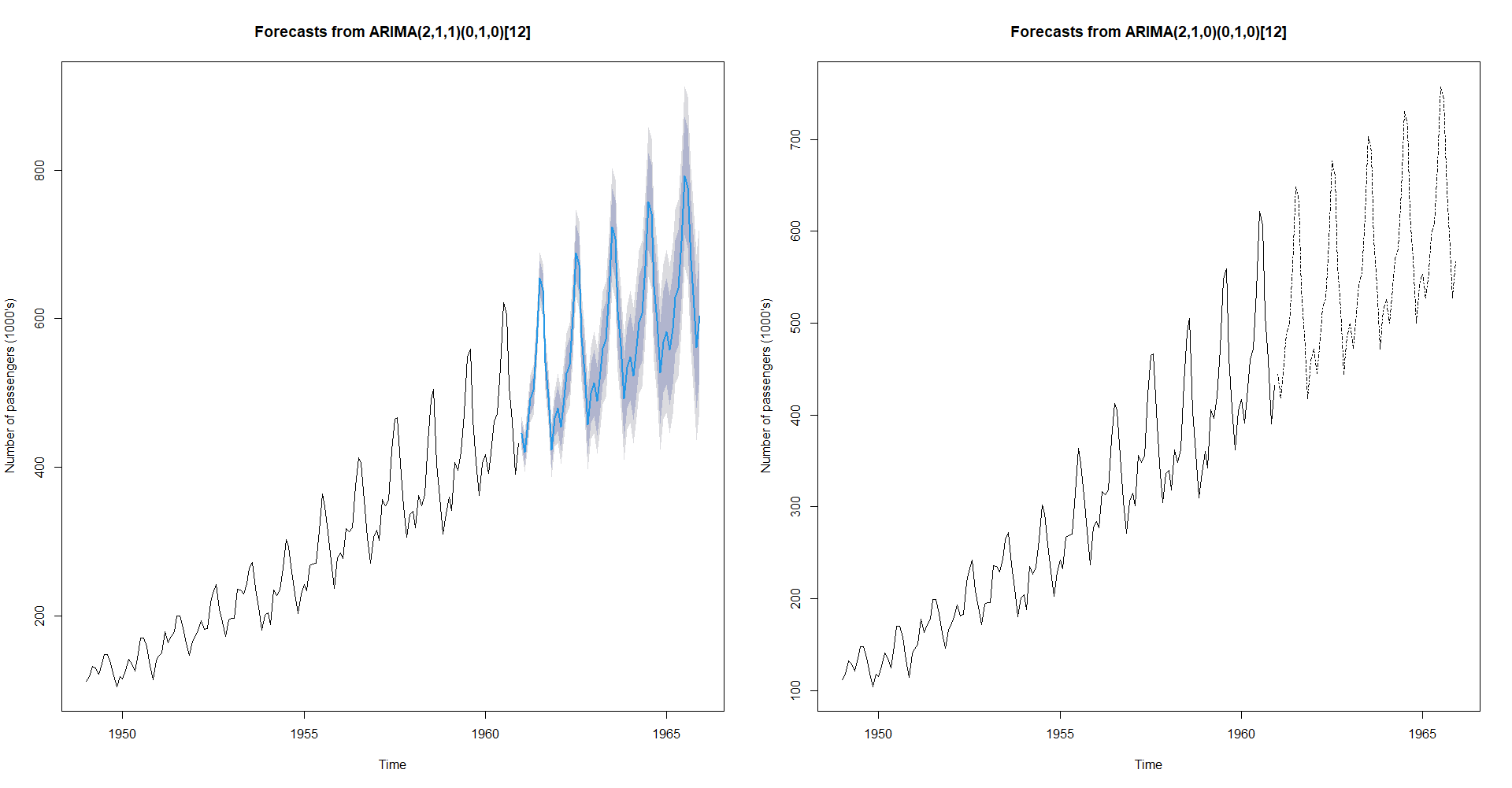
**Forecasting for the next 5 periods using the ARIMA models.**



The forecast of the auto.arima model (ARIMA(2,1,1)(0,1,0)[12]) appears better than the forecast of the model chosen from analysis.

**Comparing the forecasts results from the ARIMA and classical models.**

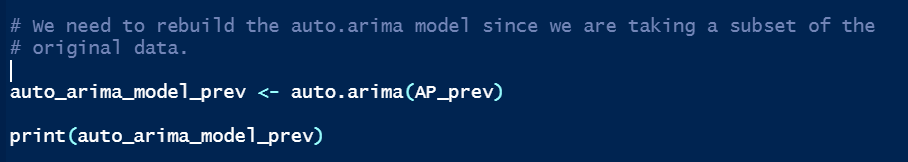




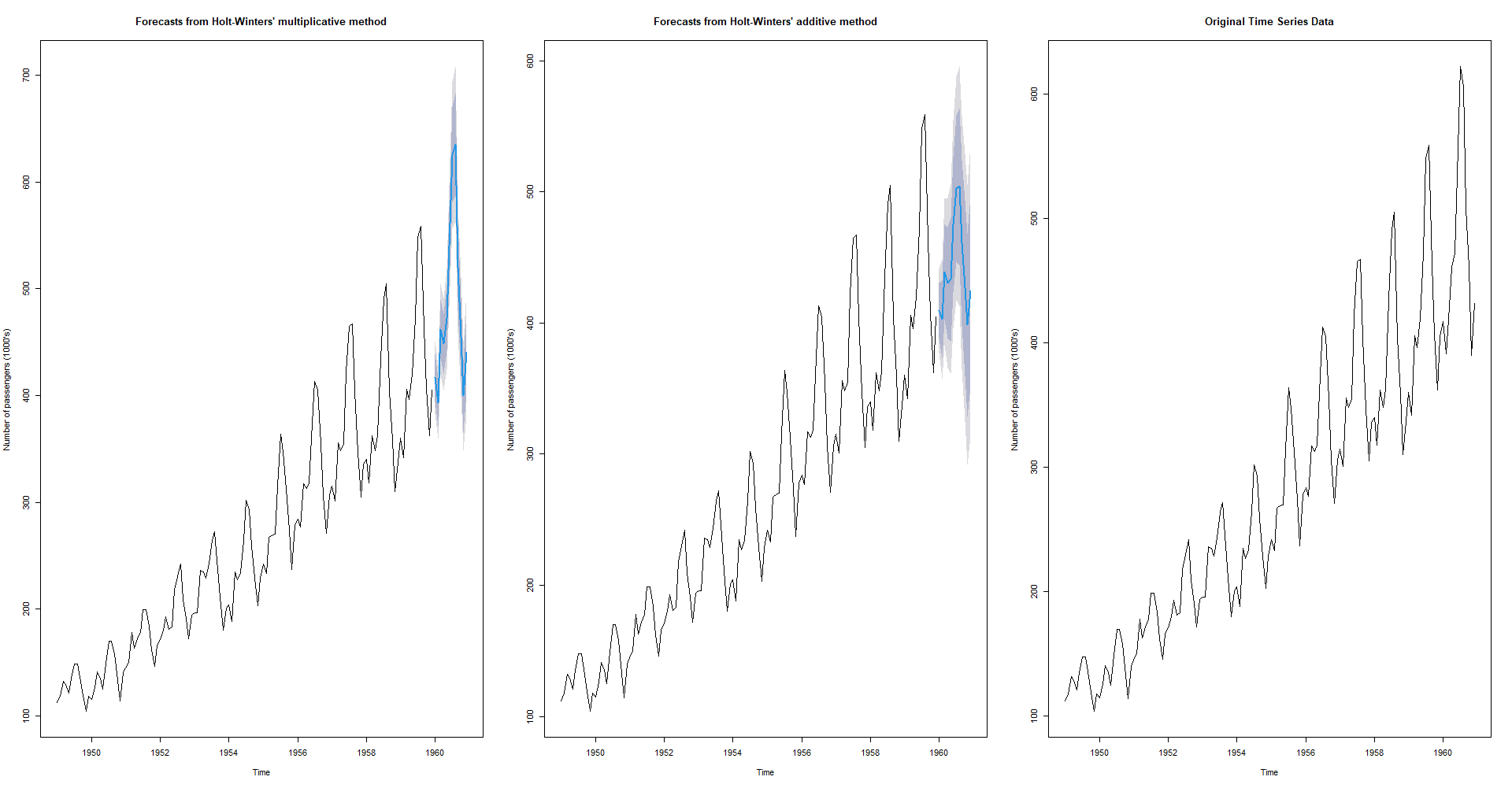
From the plots we can say that the ARIMA models yields better results than the classical models.

The best model is the auto.arima model (ARIMA(2,1,1)(0,1,0)[12]).

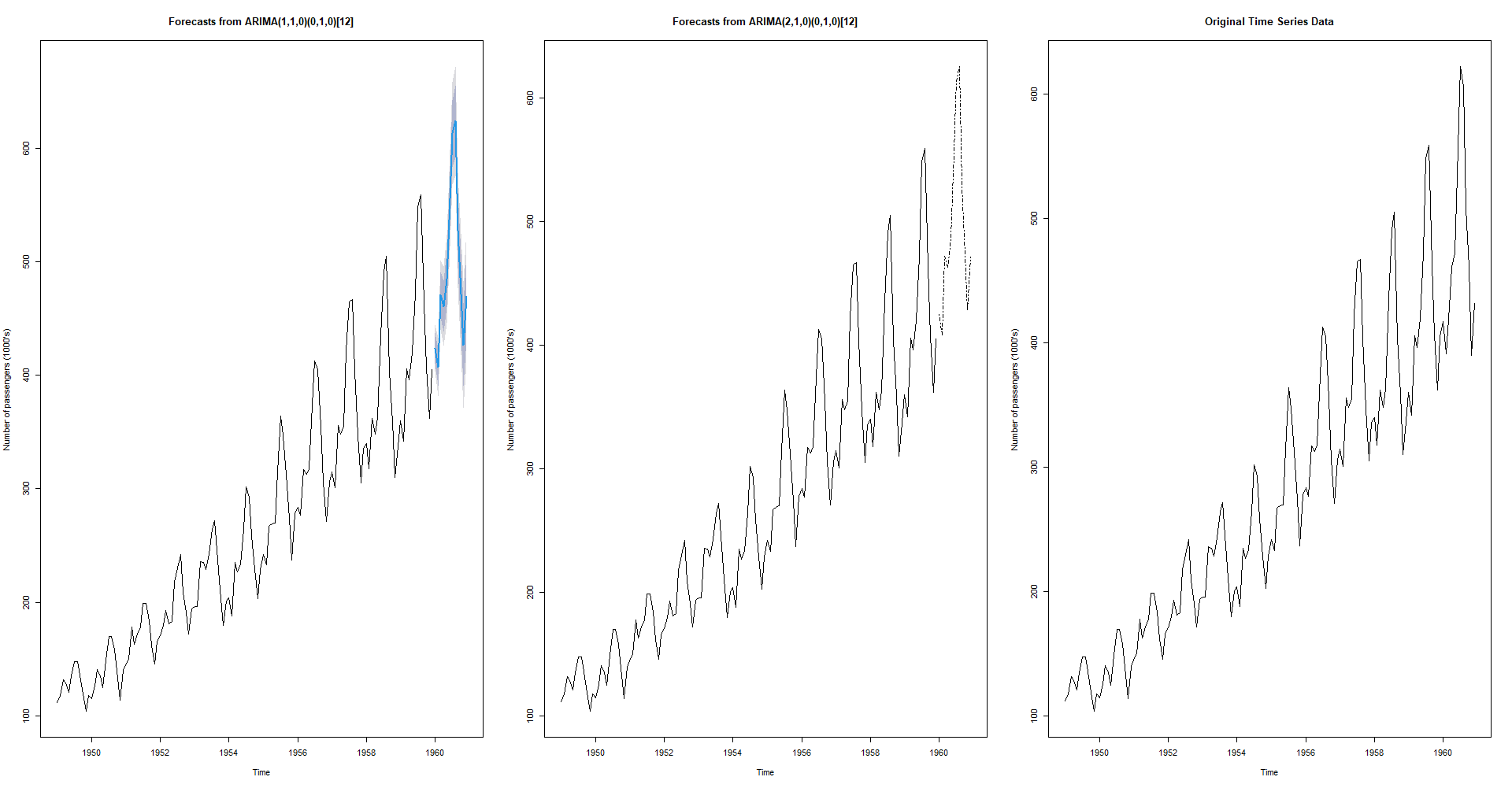
We can check the accuracy by forecasting the last periods of the time series and comparing them with the actual values. We need to rebuild the auto.arima model since we are taking a subset of the original data.



**Plotting forecasts of classical models.**



**Plotting forecasts of ARIMA models.**



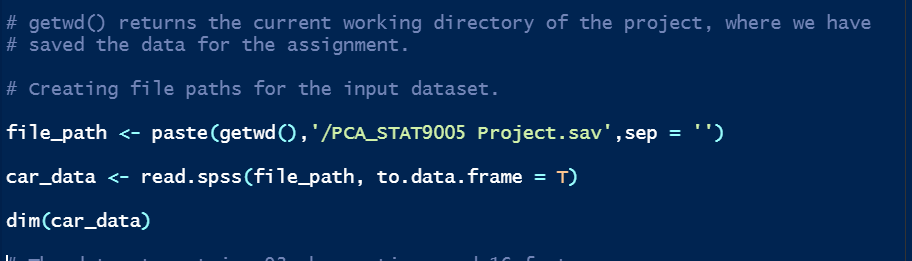
From the plots we can say that the forecasts from ARIMA model are better than the classical models.

We can use ARIMA(2,1,0)(0,1,0)[12] or ARIMA(2,1,1)(0,1,0)[12] for prediction.

**2. PCA and FACTOR ANALYSIS**

Here, we would be analysing a Car Model dataset using Principal Component Analysis (PCA) and Factor Analysis (FA). However, before proceeding with PCA or FA, we need to perform the preliminary data analysis to observe the patterns in data.

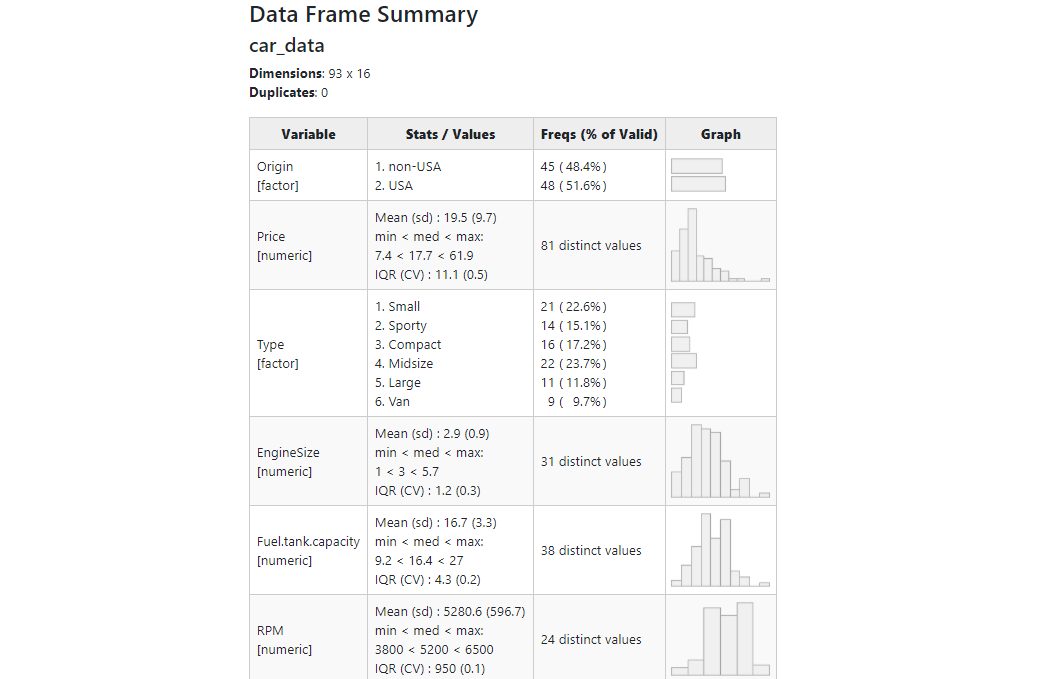
**Preliminary analysis**



The dataset contains 93 observations and 16 features.

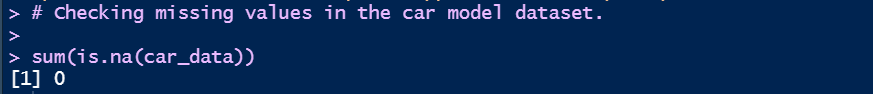
We now view the descriptive summary of the Car Model Dataset.

Since there are too many columns, the results are limited for better readability.



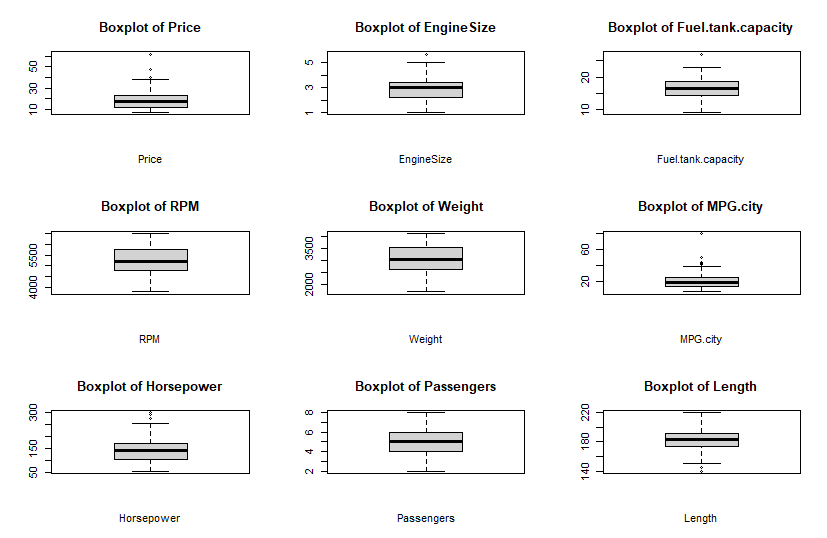
The histograms of numeric variables Price, EngineSize, Fuel.tank.capacity, MPG.city, Horsepower and Length are skewed to the right, while the other numeric values are skewed to the left. We also need to remove some columns that are not essential for the analysis.

Next, we check for missing values in the dataset.



There are no missing values in the dataset.

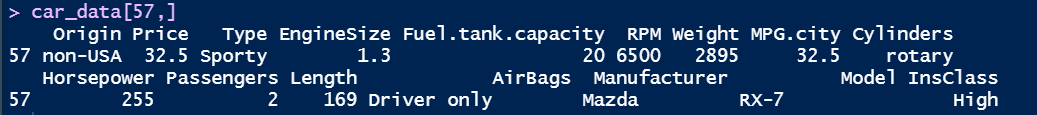
We generate boxplots to check if there are any outliers.



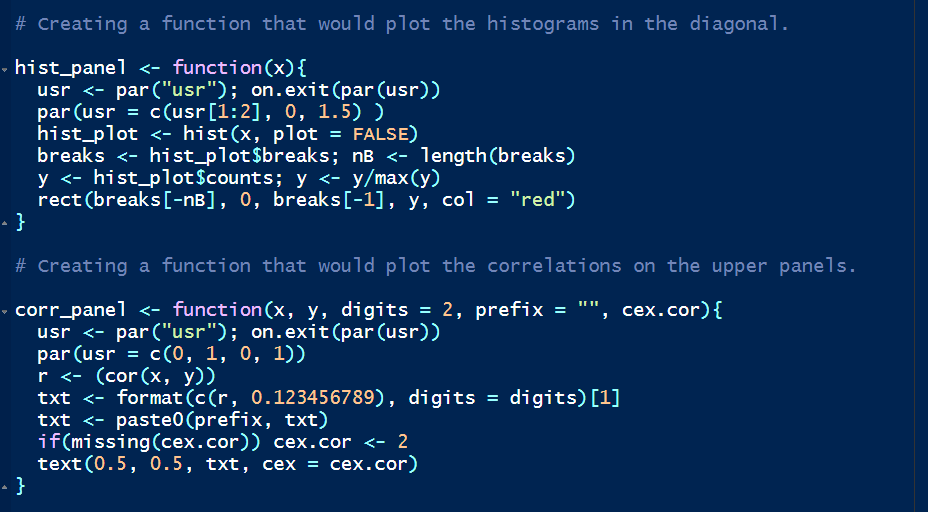
From the boxplots we can see that there are very less outliers in the data.

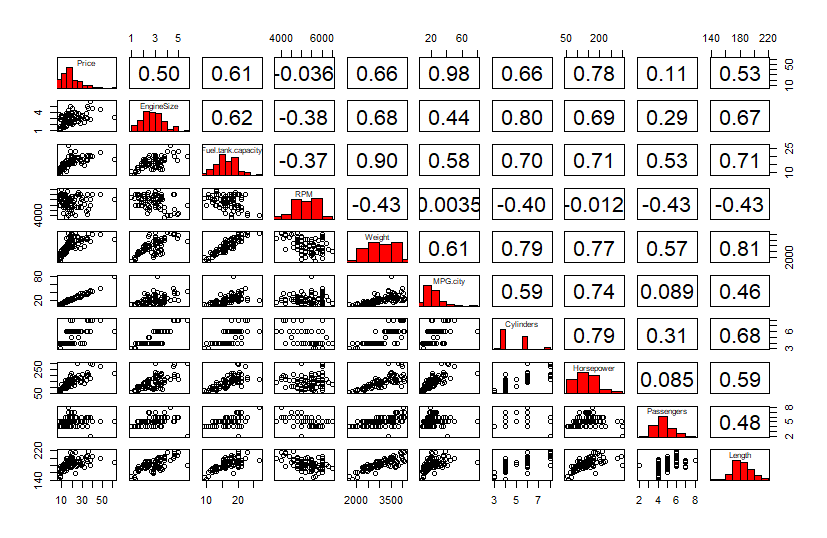
For performing Principal Component Analysis (PCA) and Factor Analysis (FA), we need to ensure that majority of the variables are numerical. Hence, we need to manipulate the data so that we have minimal categorical values.

The 57th observation in the "Cylinders" variable is inconsistent with the other observations. It holds the value "rotary" while the other observations hold numeric values. Hence, the 57th observation is to be removed. We only need the categorical information provided by "Manufacturer" and "Model".

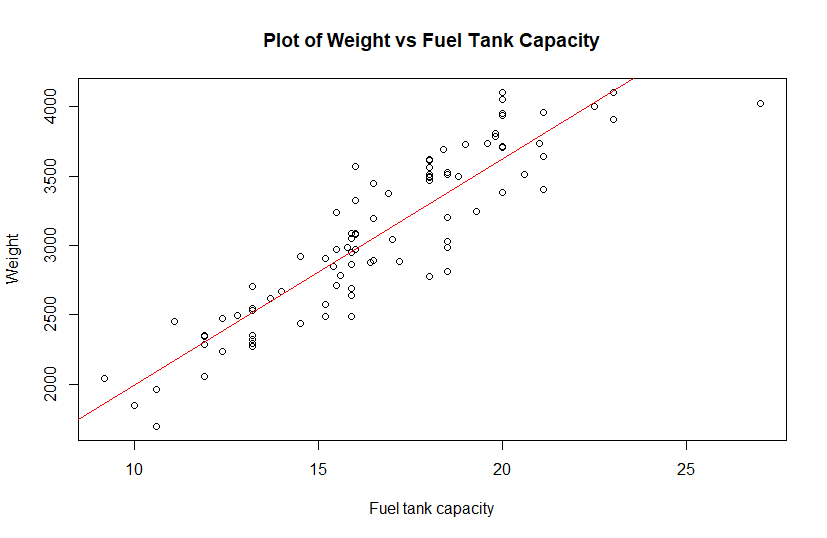


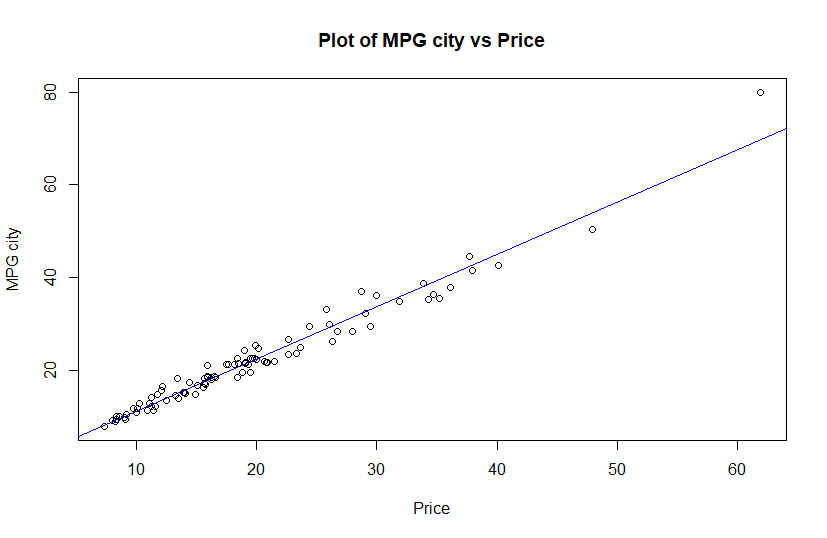
The pairs command will generate the scatterplots between pairs of variables of a dataframe. We will create two functions where we modify the pairs plot such that the upper triangle will hold the correlation coefficients, the lower triangle will hold the scatterplots and the diagonal elements would hold the histograms of each numeric variable.

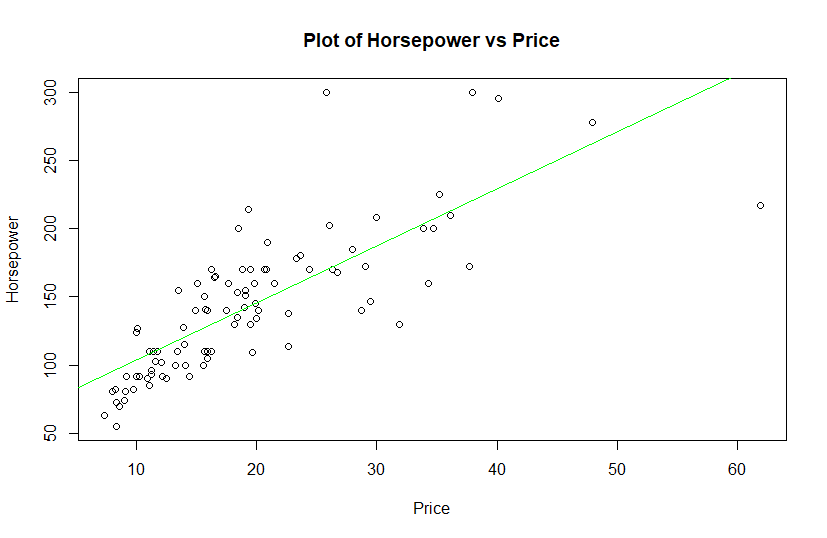




We can see that there are strong correlations between pairs of variables ("Fuel.tank.capacity", "Weight"), ("Price", "MPG.city") and ("Price", "Horsepower"). We can visualize this by fitting the regression lines in the scatterplots.

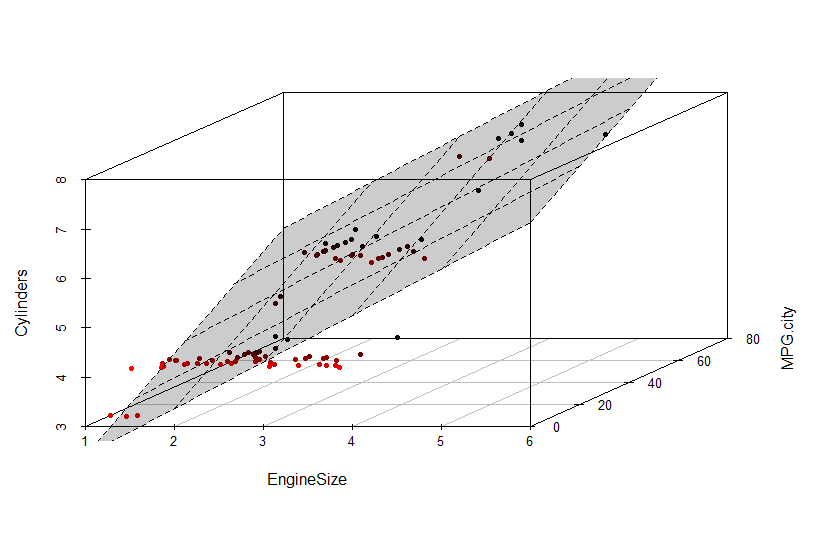




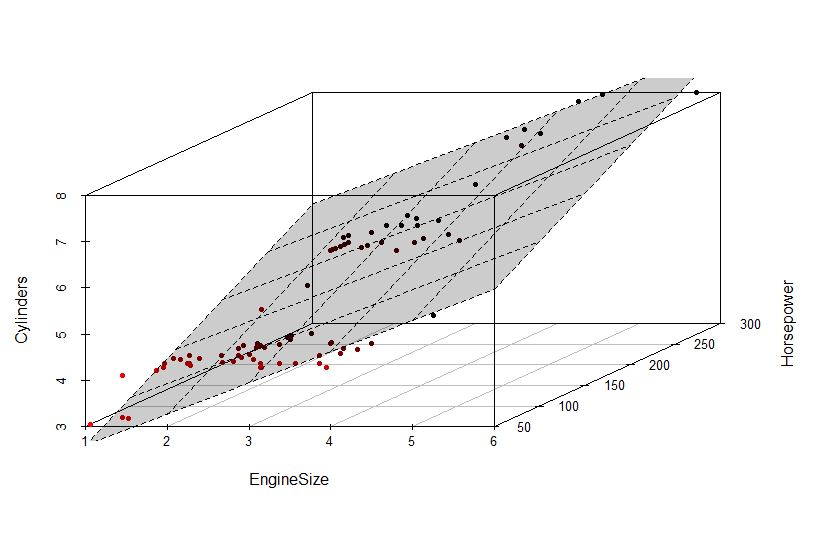


Next, we generate some 3D plots to understand the multilinear relationship between the attributes “Cylinders”, “EngineSize”, “MPG.city”, “Horsepower” and “Fuel.tank.capacity”.

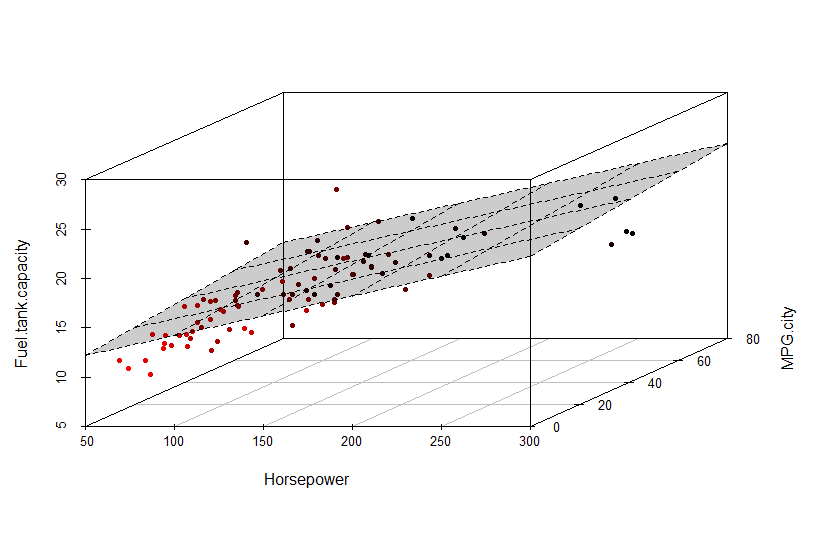
**3D plot of Cylinders Vs EngineSize + MPG.city**



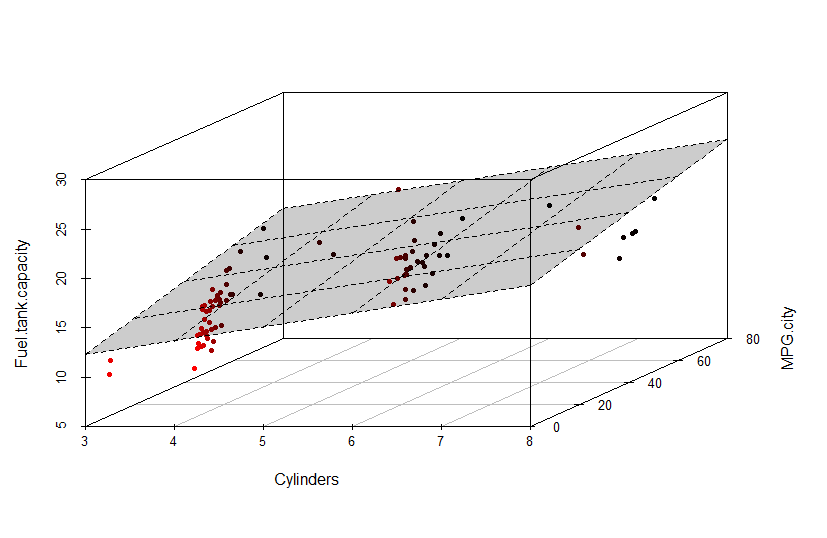
**3D plot of Cylinders Vs EngineSize + Horsepower**



**3D plot of Fuel.tank.capacity Vs Horsepower + MPG.city**



**3D plot of Fuel.tank.capacity Vs Cylinders + MPG.city**



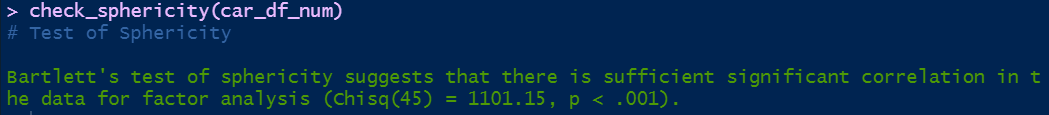
From the plots we can say that the relationships between the variables are significant as the regression line covers most of the points. We can say that the latent variables are EngineSize, MPG.city and Horsepower.

Next, we can perform “Bartlett's test of sphericity” to check if we can perform data reduction techniques.

“Bartlett's test of sphericity”:

“H0 = Variables are uncorrelated i.e. correlation matrix is an identity matrix.”

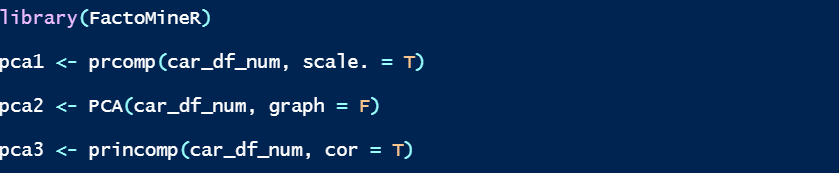
“HA = Variables are correlated and suitable for factor analysis i.e. correlation matrix is not an identity matrix.”

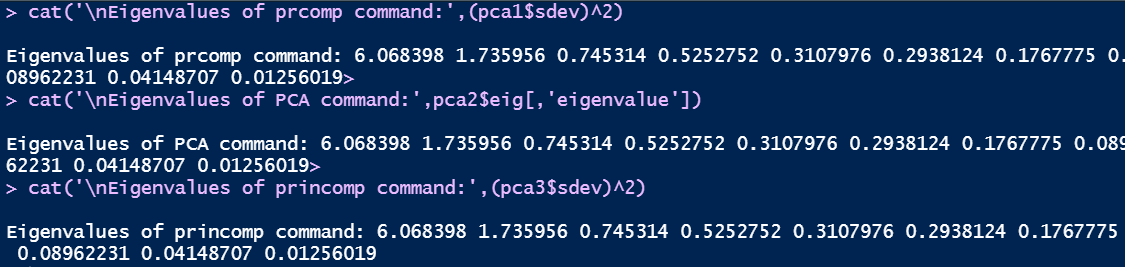


Since p-value is less than alpha (0.05), we reject the Null Hypothesis. There is sufficient significant correlation in the data for factor analysis (Chisq(45) = 1101.15, p < .001).

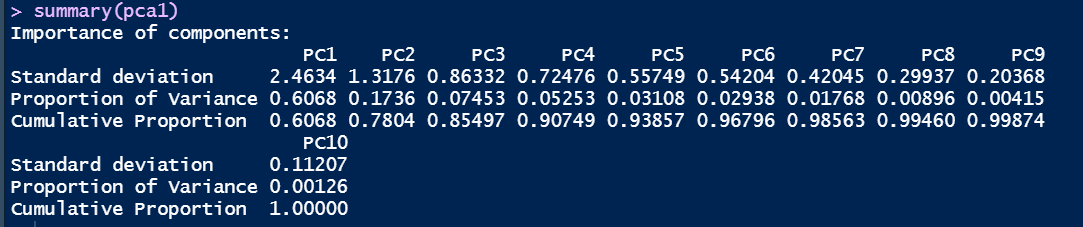
**Principal Components**

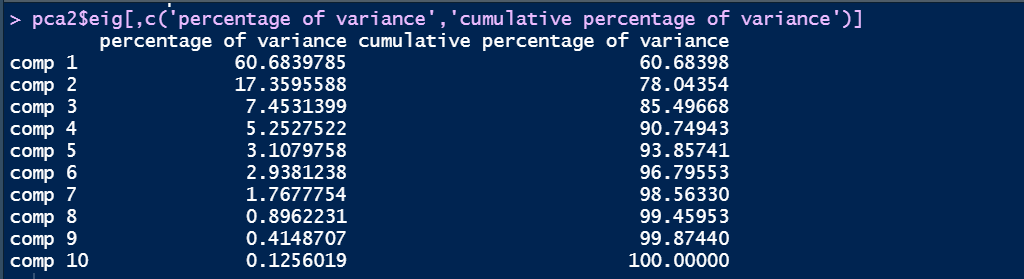
PCA can be done in R using 3 commands namely prcomp, princomp, and PCA. We need to invoke the 'FactoMineR' library for using PCA command.

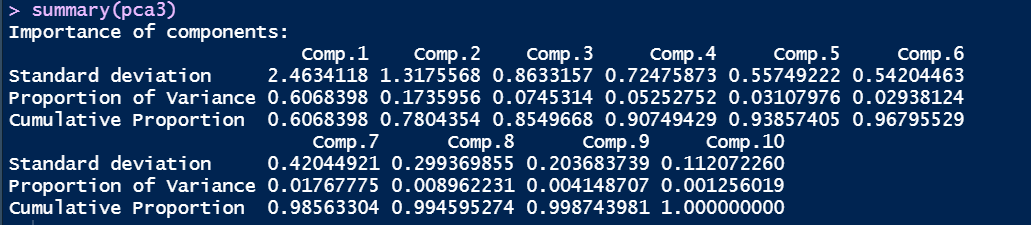




The eigenvalues of the 3 commands are same. If the eigenvalues are same, then the eigenvectors would be same as well. We can view the amount of information by each PCA component using the summary.

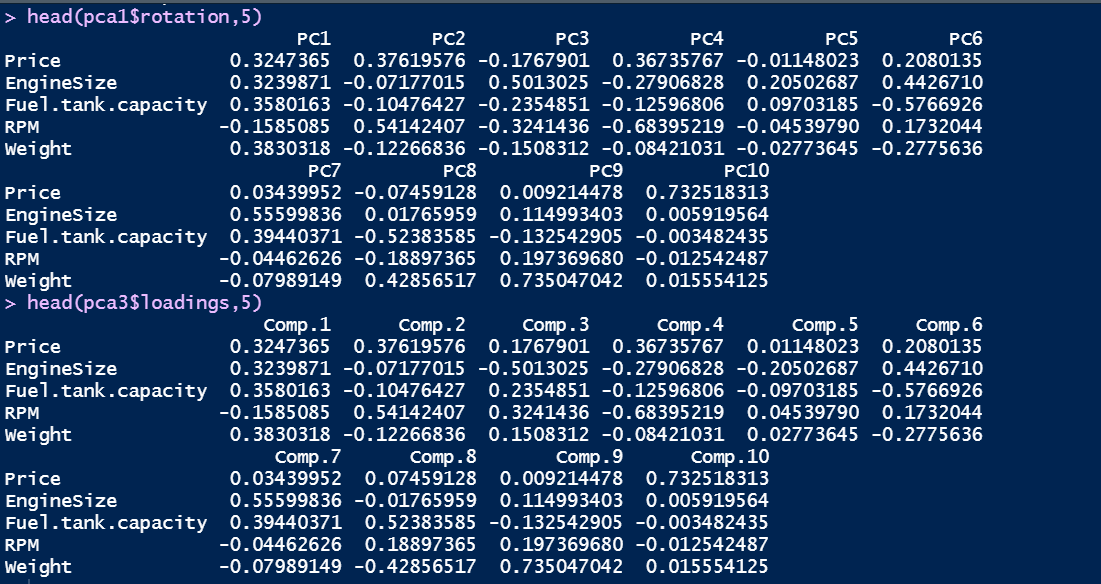


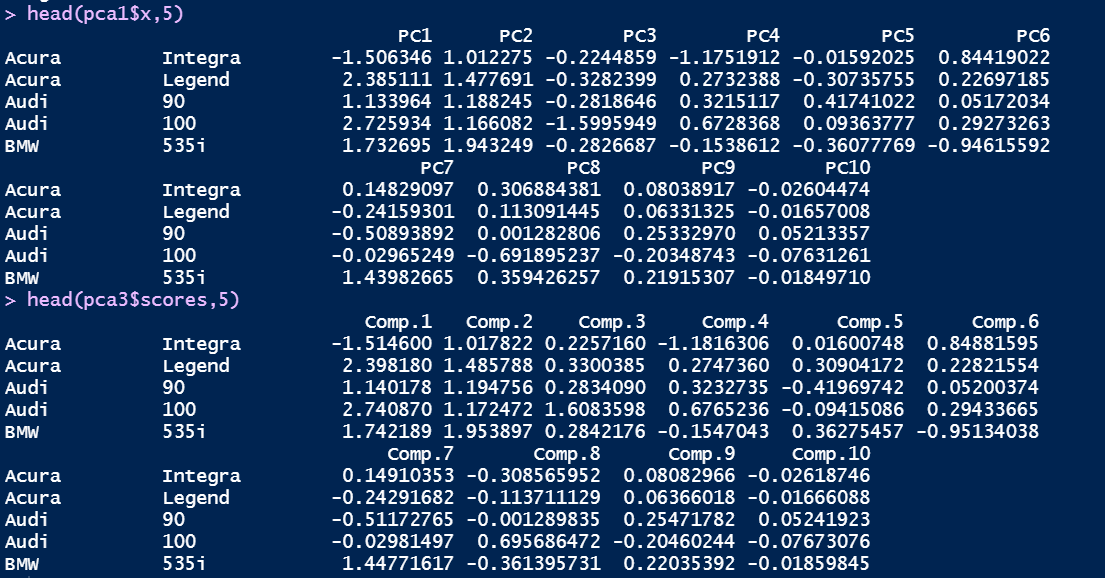




We can see that the results from the 3 commands are identical.

Next, we need to check the factor loadings and PCA Scores. Factor loadings and scores cannot be computed from result of PCA command.



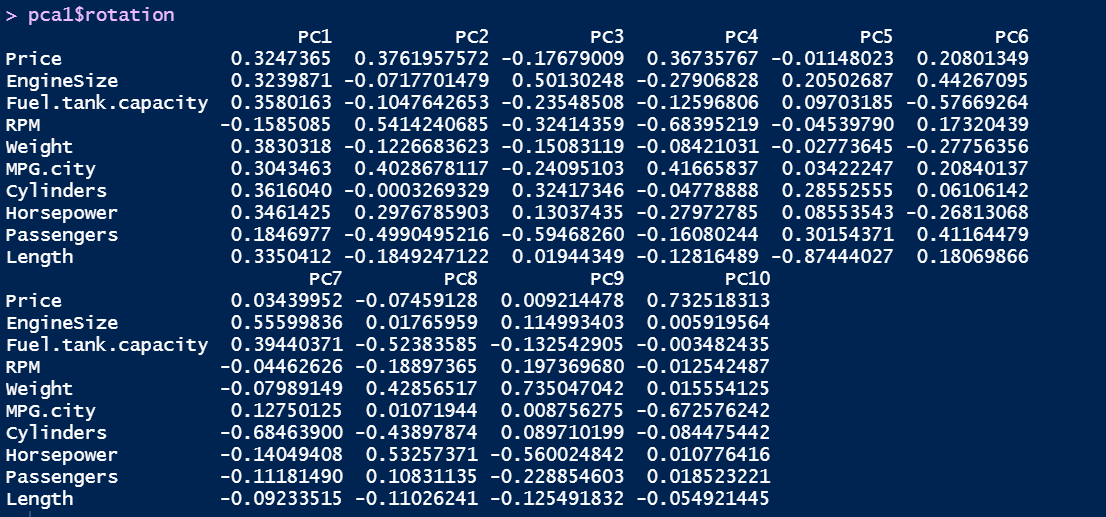


The factor loading results of prcomp and princomp commands are identical.

The PCA scores are almost identical between the results of the commands.

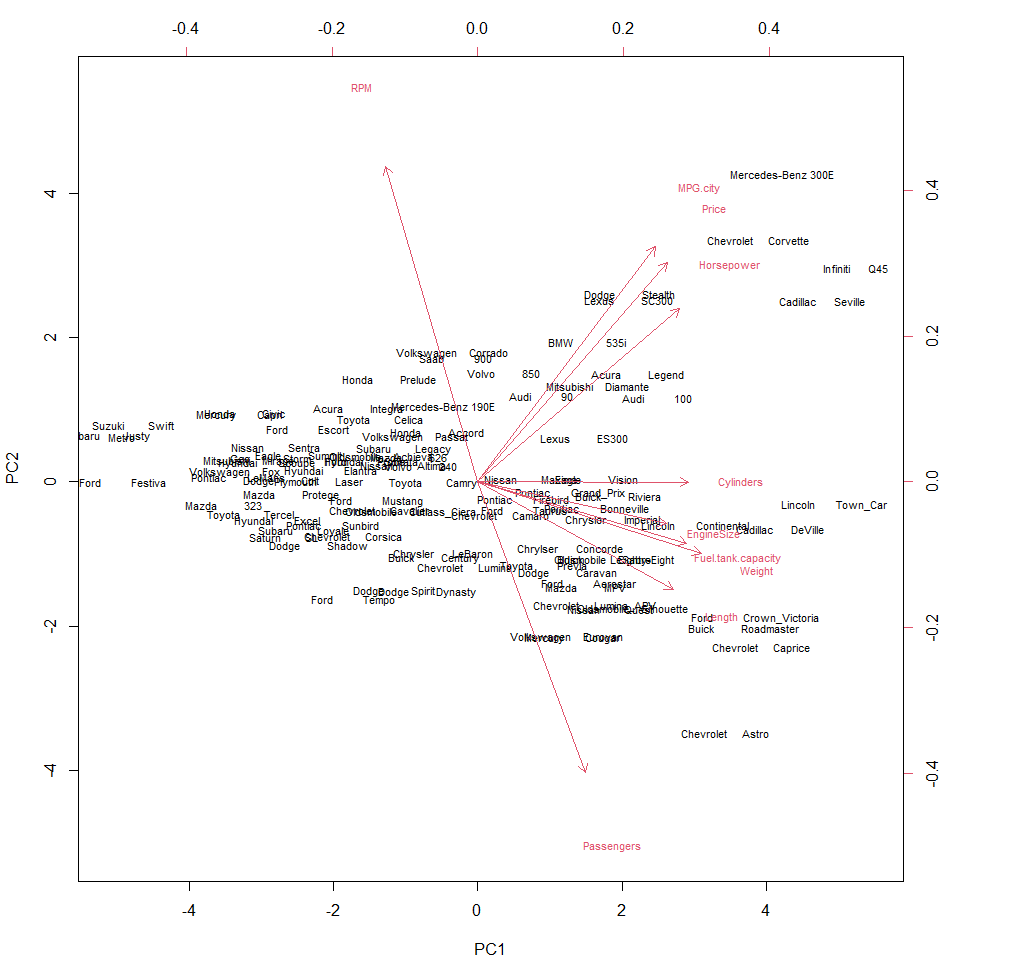
Factor Loadings are calculated using the eigenvalues and the slope co-ordinates. Factor Loading = u(sqrt(lambda))/sqrt(var(X)\*lambda), where u is the slope, lambda is the eigenvalue and X is the input matrix. PCA scores are calculated by multiplying the input matrix and the matrix formed using the eigenvectors of the correlation matrix that is derived from the input matrix.

The correlation between the original variables and principal components can be studied by viewing at the factor loadings. Since the factor loading results of prcomp and princomp commands are identical, we will be using the result of prcomp command.



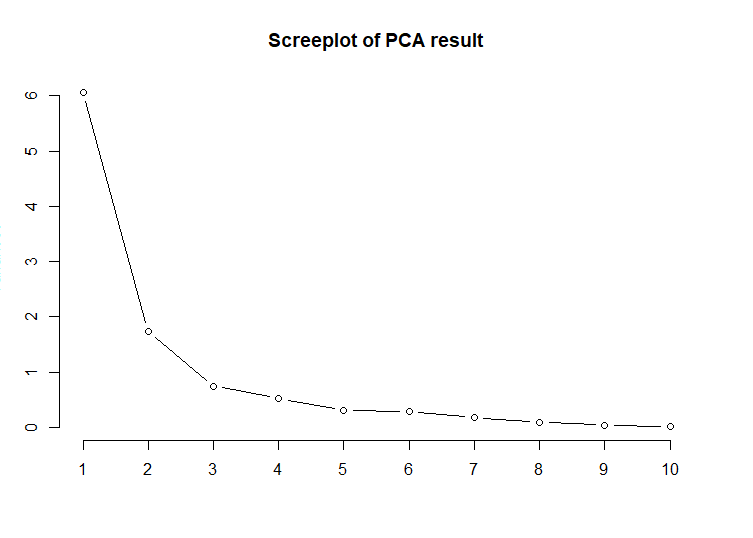
The correlations are very weak or almost moderate between the original variables and the principal components. Only few of the correlations are strong.

We use biplot to understand the variables represented by PC1 and PC2.



The variables Cylinders, EngineSize, Fuel.tank.capacity, Weight and Length are more related to PC1 while the rest are more related to PC2.

Next, we need to generate a screeplot to check the number of components to be retained.



From the screeplot we can say that it is best to retain 3 components.

After the second component, the graph almost saturates. This is because most of the information is explained by the first two principal components.

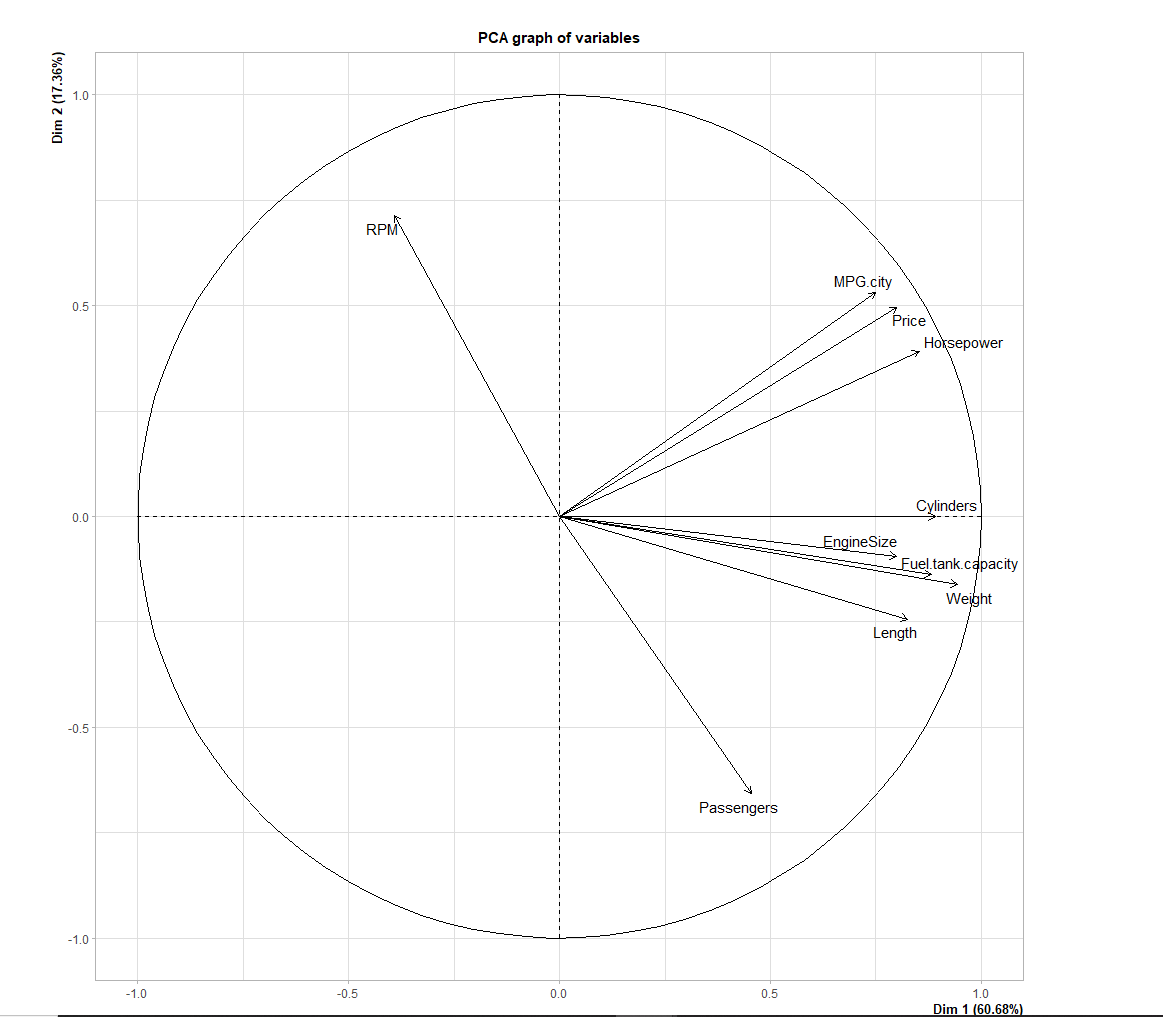
This decision is supported by the factor loadings and biplot.

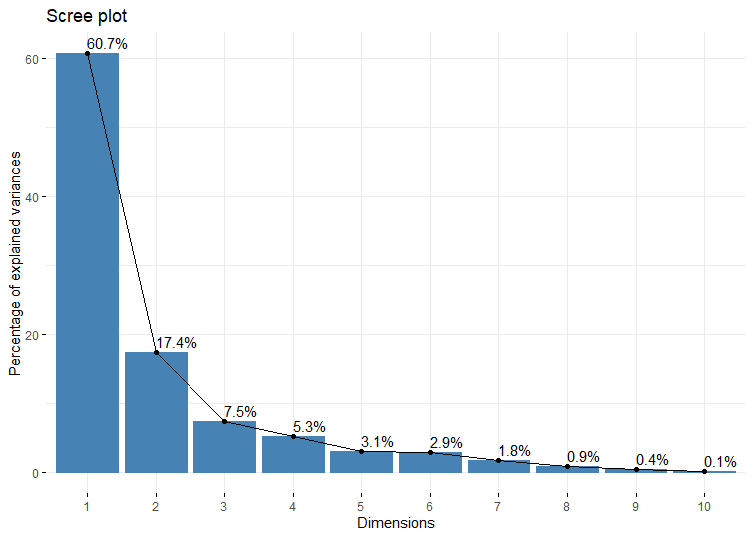
We now generate another biplot which captures the relation between the observations and the PCA scores.

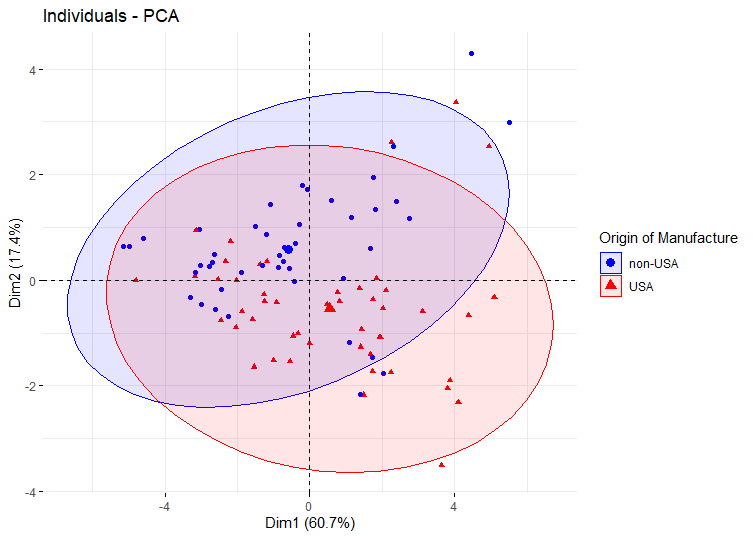


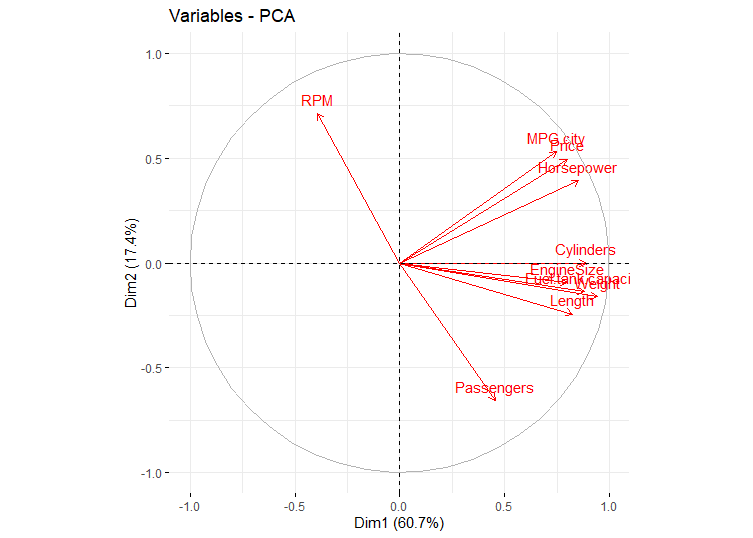
Most of the observations are pointed towards PC1 than PC2. This is due to the strong correlation between PC1 and observations.

Next, we need to generate a screeplot, a correlation circle of the dataset and a biplot that represents the car models by “Origin” – “USA” and “NonUSA”.





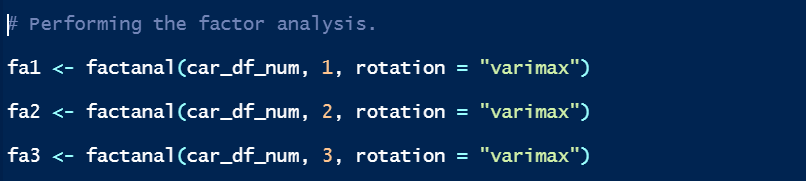


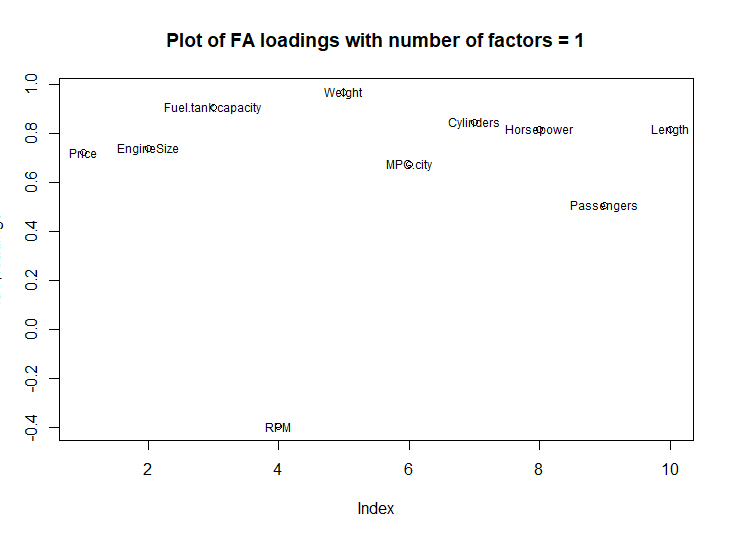


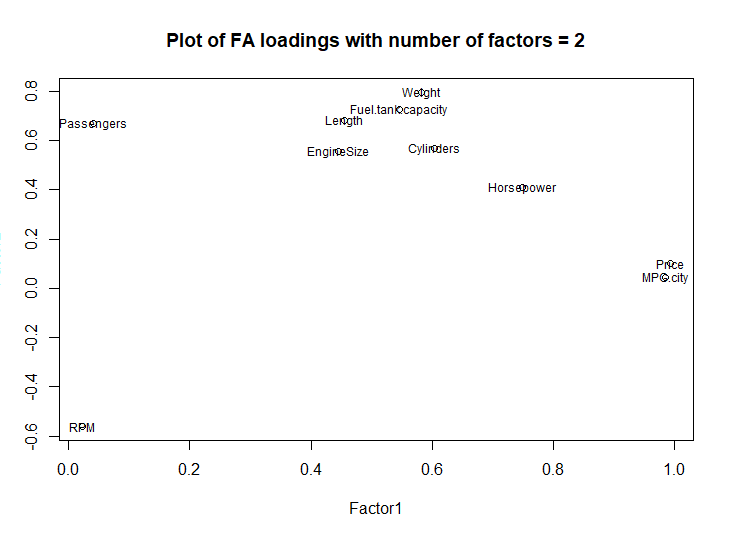
There is a slight difference in the biplots between the USA manufacturers and the rest of carmakers. For the cars manufactured in USA, the variables MPG.city, Price and Horsepower are the significant components. For the other manufacturers, the variables Cylinders, EngineSize, Weight, Fuel.tank.capacity and Length are the significant components.

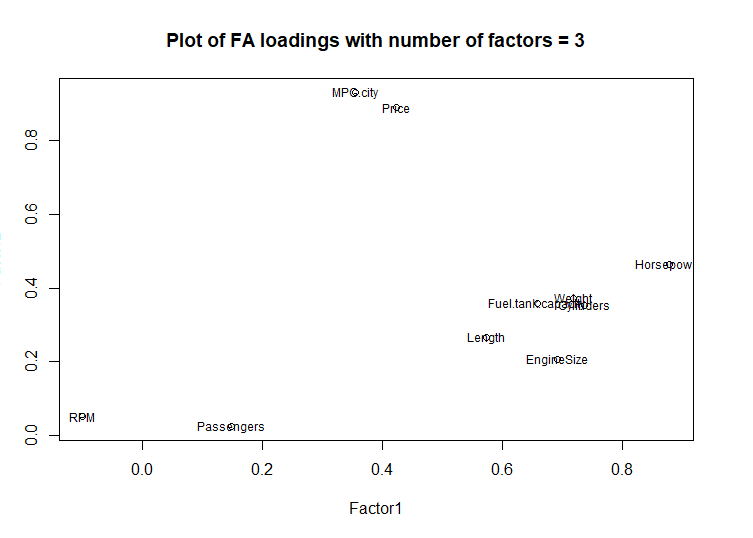
**Factor Analysis**

After PCA we will perform Factor Analysis (FA) and check if PCA is different from FA.







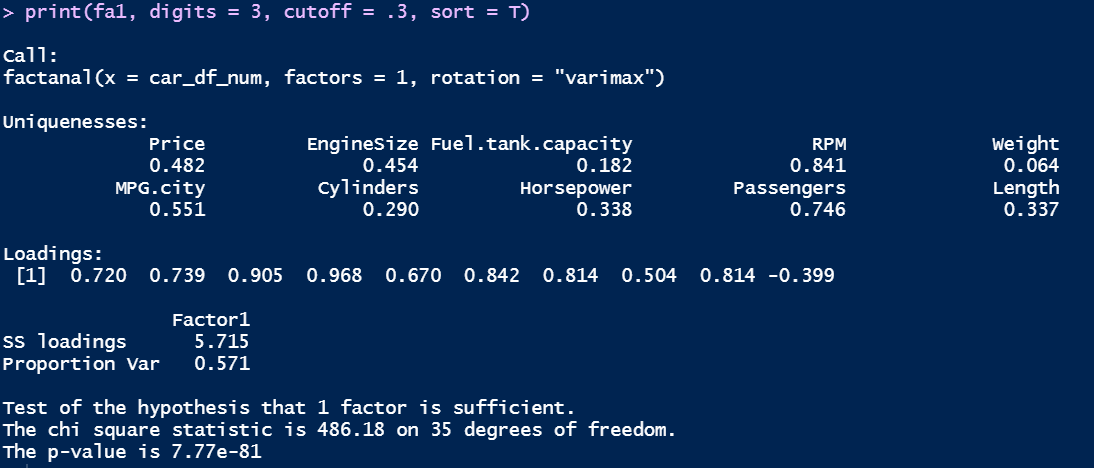


When number of factors = 1, most of the weights are higher than 0.3.

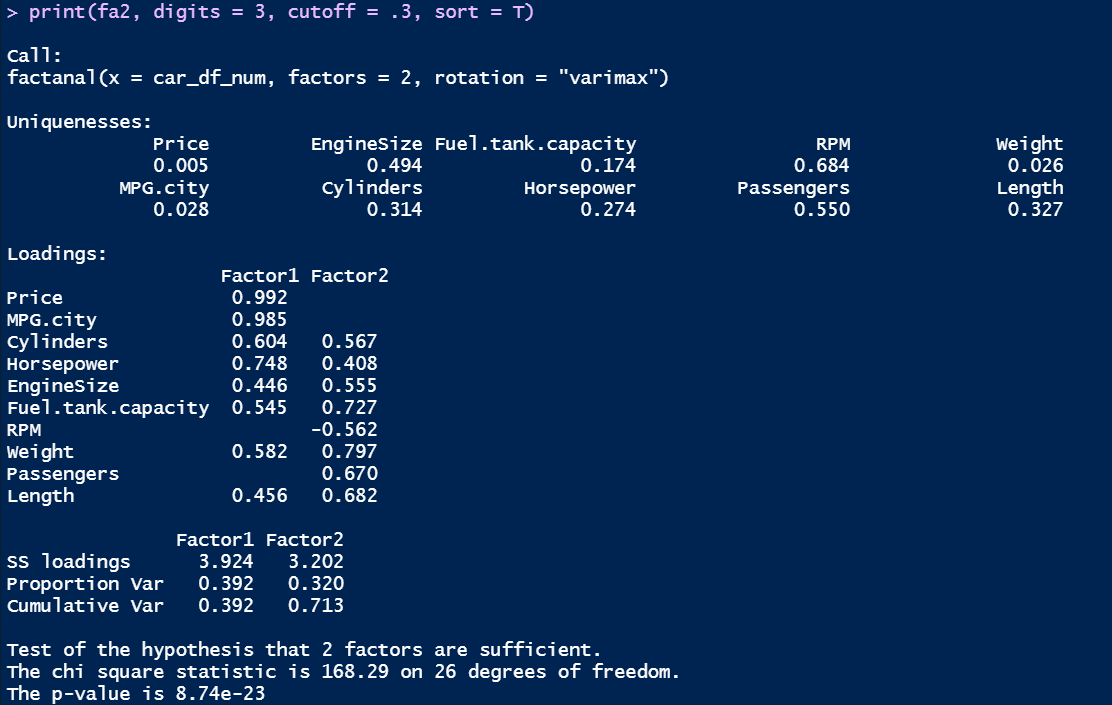
When number of factors = 2, only some of the weights are higher than 0.3.

When number of factors = 3, most of the weights are lesser than 0.3.

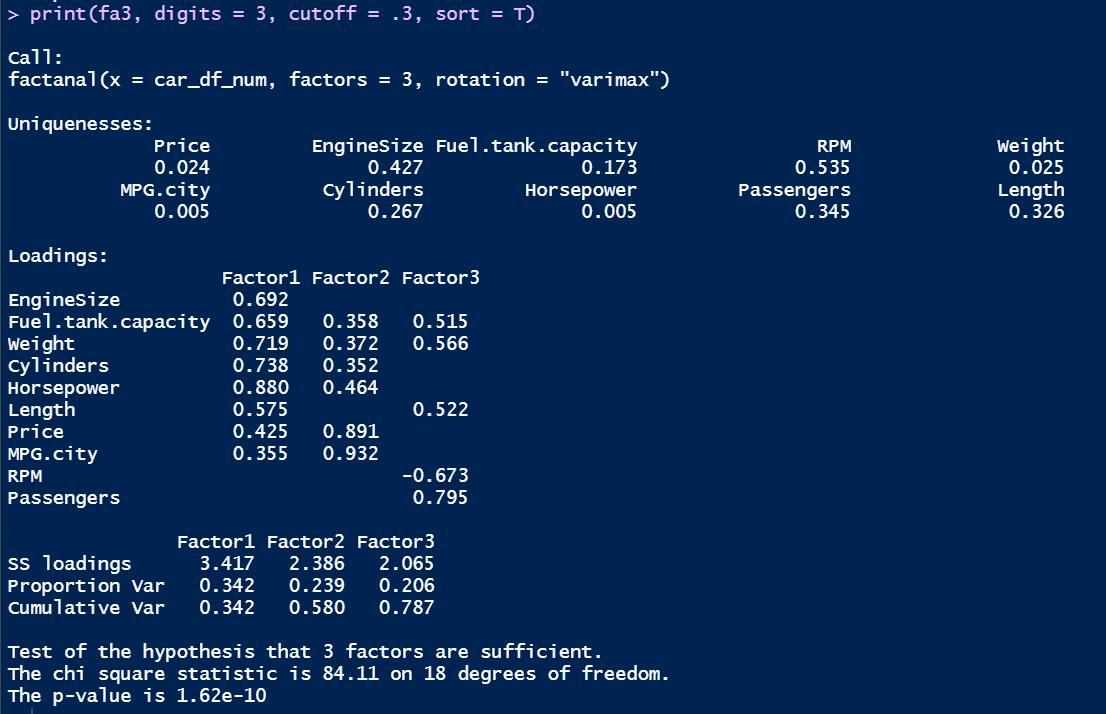
The weights are to be checked when the values are higher than 0.3.



Around 57.1% of the variance is explained by factor 1 when number of factors=1.

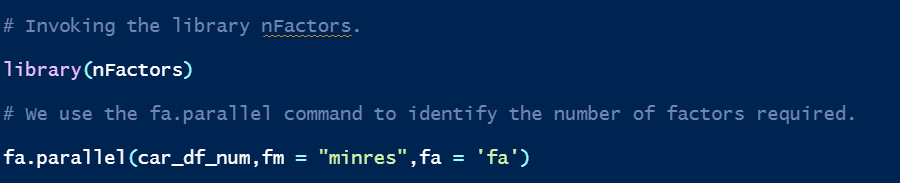


Around 39.1% of the variance is explained by factor 1 and 32% of the variance is explained by factor 2 when number of factors = 2.

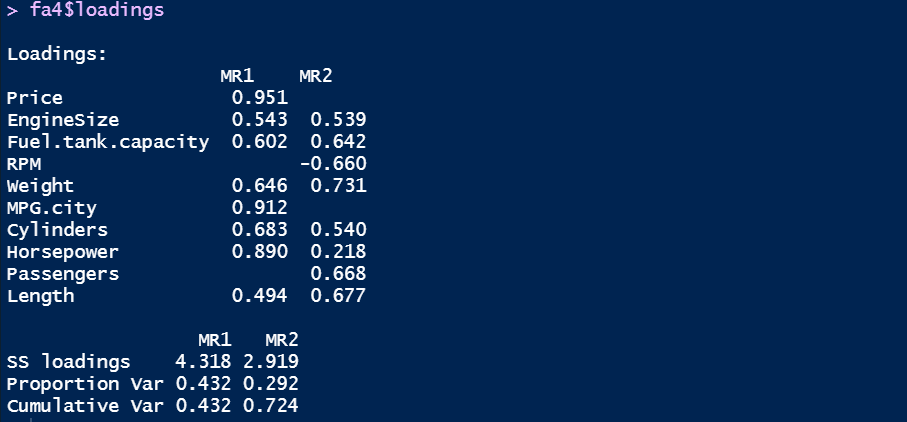


Around 34.1% of the variance is explained by factor 1, 23% of the variance is explained by factor 2 and 21% of the variance is explained by factor 3 when number of factors = 3. As the number of factors increases the variability explained by the first factor decreases.

Instead of analysing the number of factors to be created, we can use the “nFactors” library to find the number of factors required.

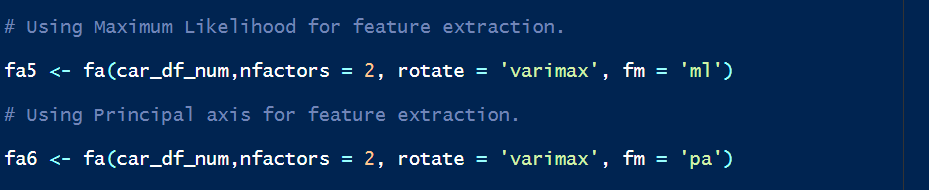


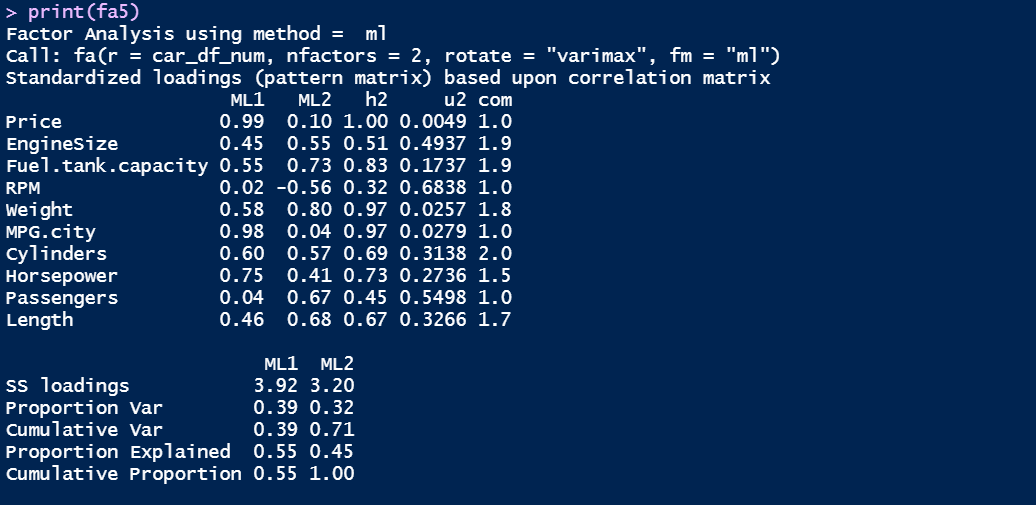
We can see that we need only 2 factors and no additional components. This is consistent with the PCA results.

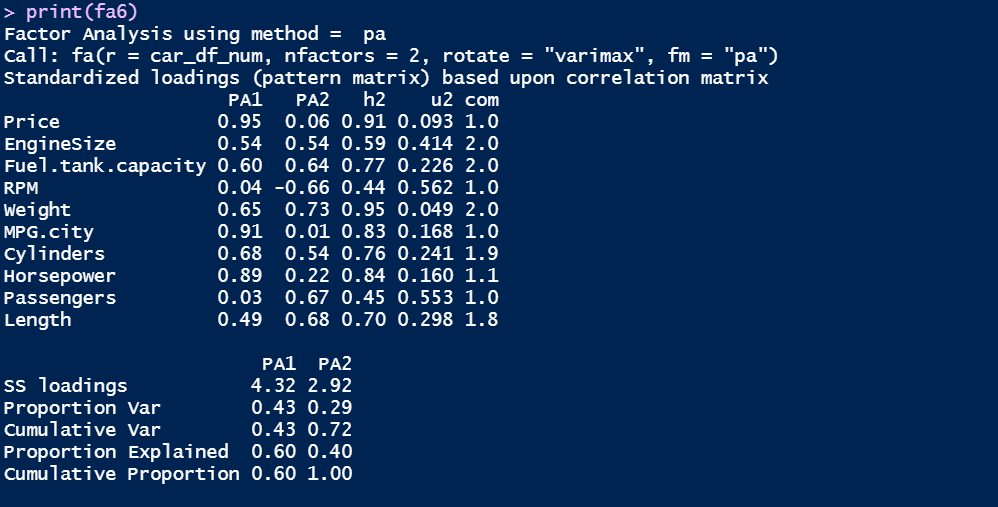


The loadings are almost like the factanal results. Majority of the loadings are above 0.3 and need to be re-investigated. Here, around 43% of the variance is explained by factor 1 and 30% of the variance is explained by factor 2. This is a better result than factanal command.

Next, we need to use the “fa” command to extract factors using methods such as “Maximum Likelihood”, “Principal Axis”, etc.

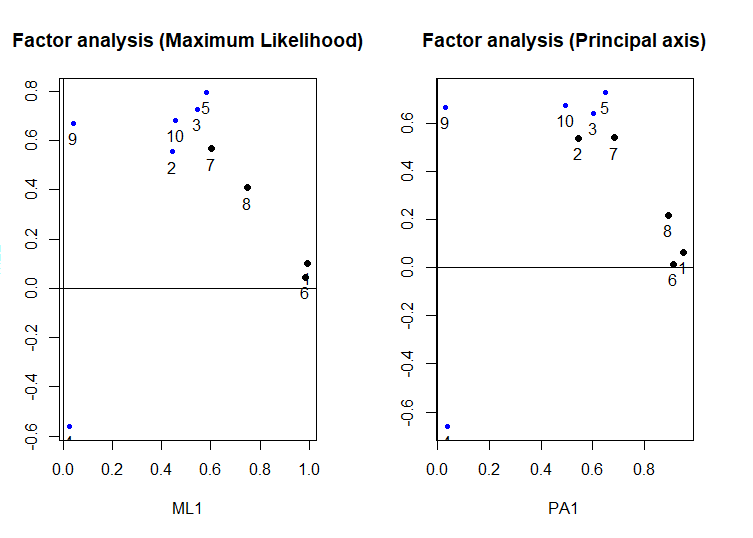


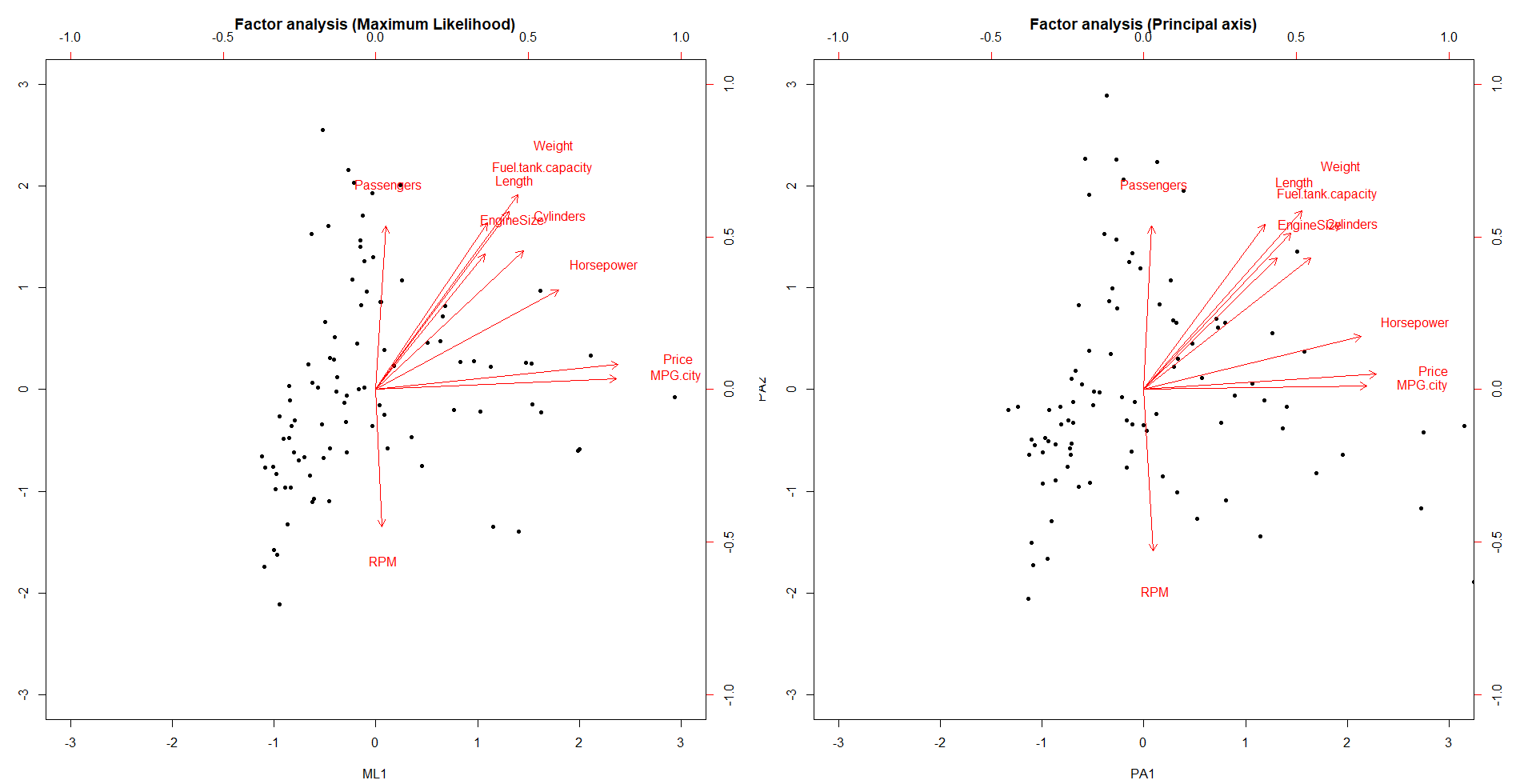




We can see that the factor standardized values are almost similar between the two methods.

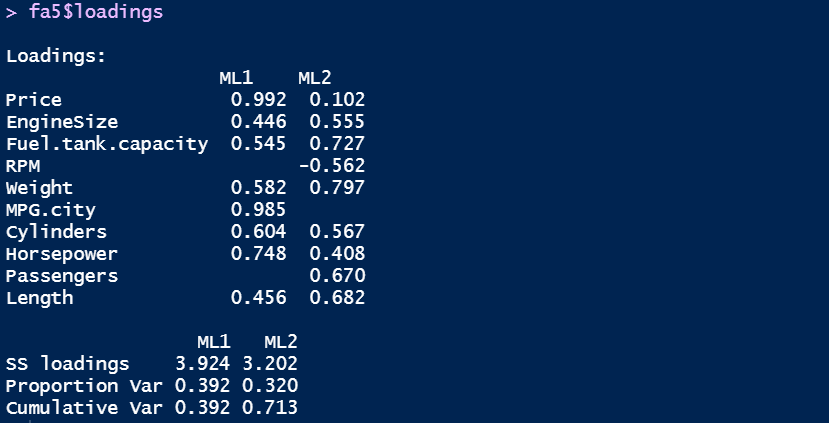
However, we need to verify this visually as well using Biplots and FA plots.

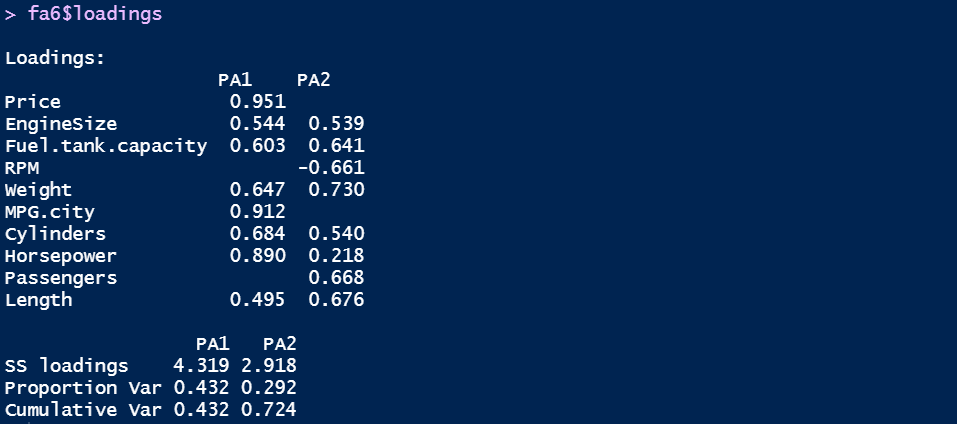




The plots are not much different between the two models.

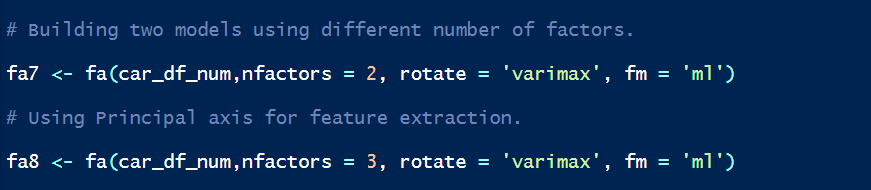
Lastly, we need to look at the factor loadings for assessing similarities.

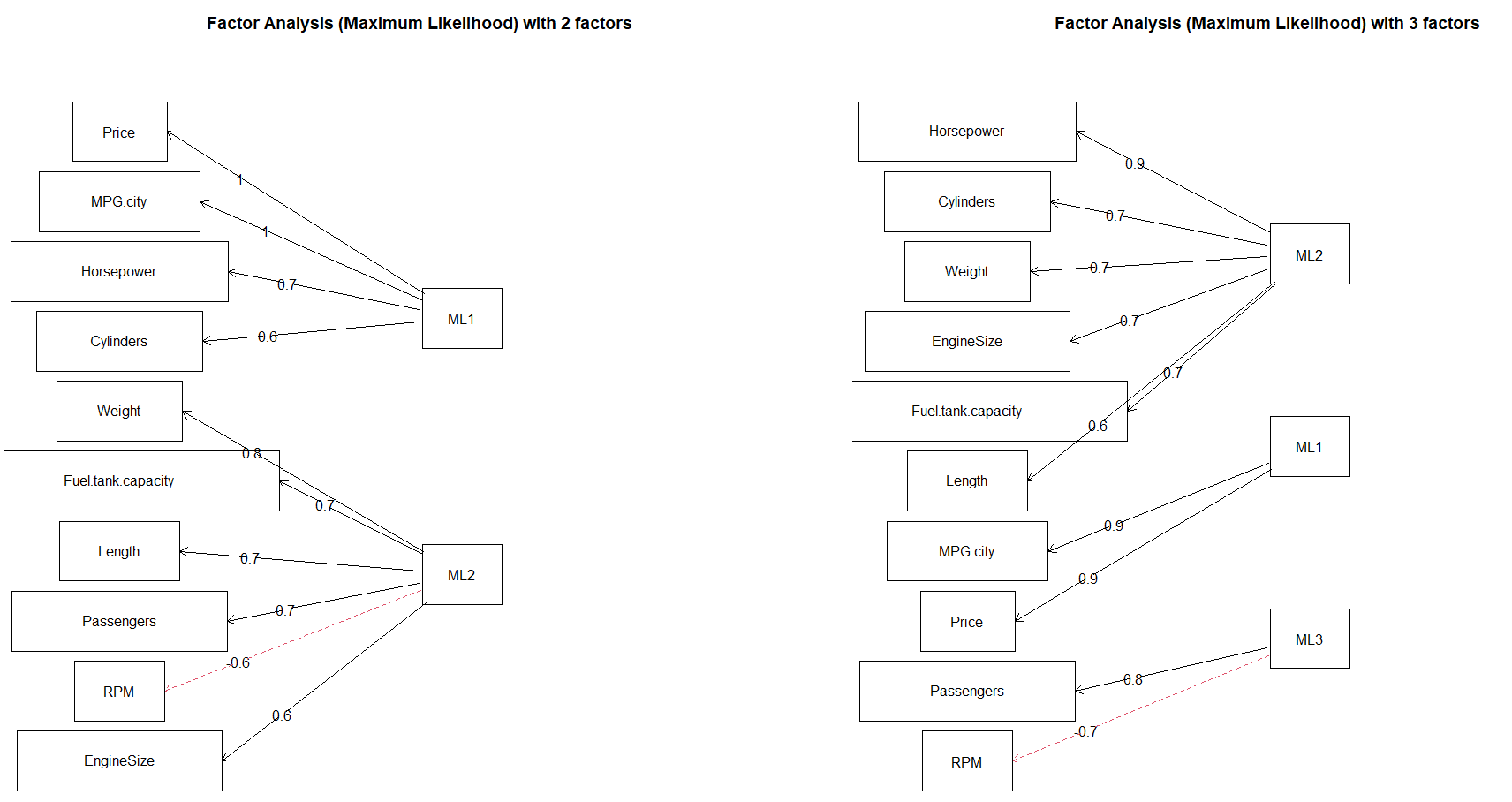




There is a minor difference in the factor loadings when different methods are used.

Now, we build path diagrams to understand the structure of the factor loadings.



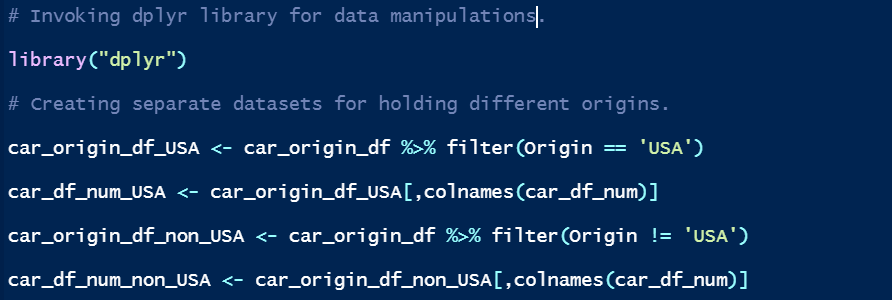


As the number of factors increases, the weights of each variable decrease slightly. There is no correlation in the factors, and we can see that this is akin to PCA.

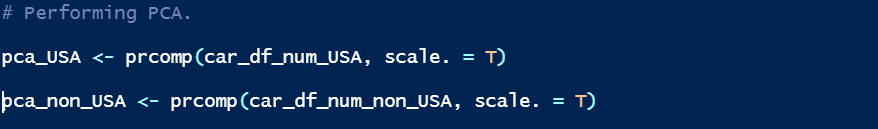
The prcomp command would be used as final PCA model. The fa command would be used as final FA model.

**Second Analysis**

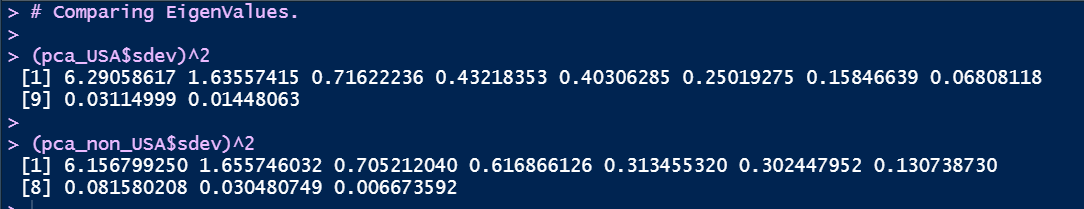
We need to split the original dataset into two categories: 1. With origin as USA, 2. With origin other than USA (Non-USA).



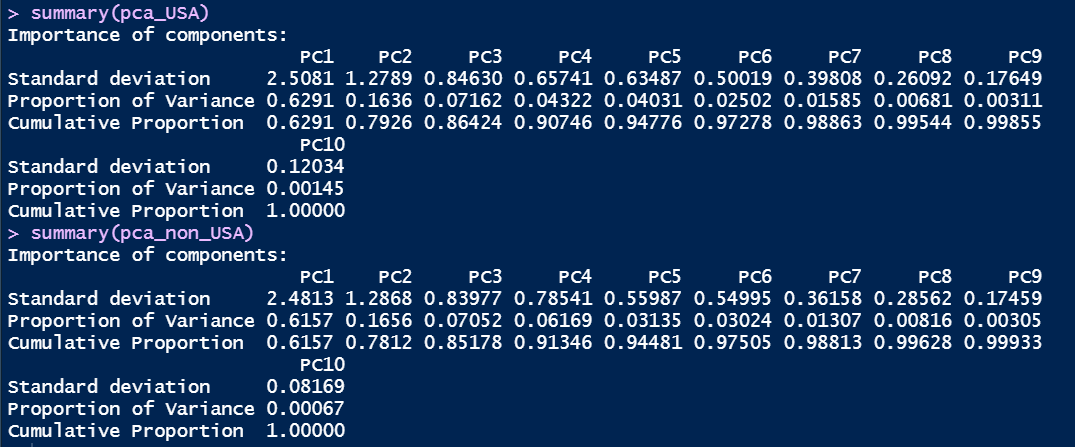
First, we perform PCA on the two datasets.



We now compare the eigenvalues from the two PCA’s.

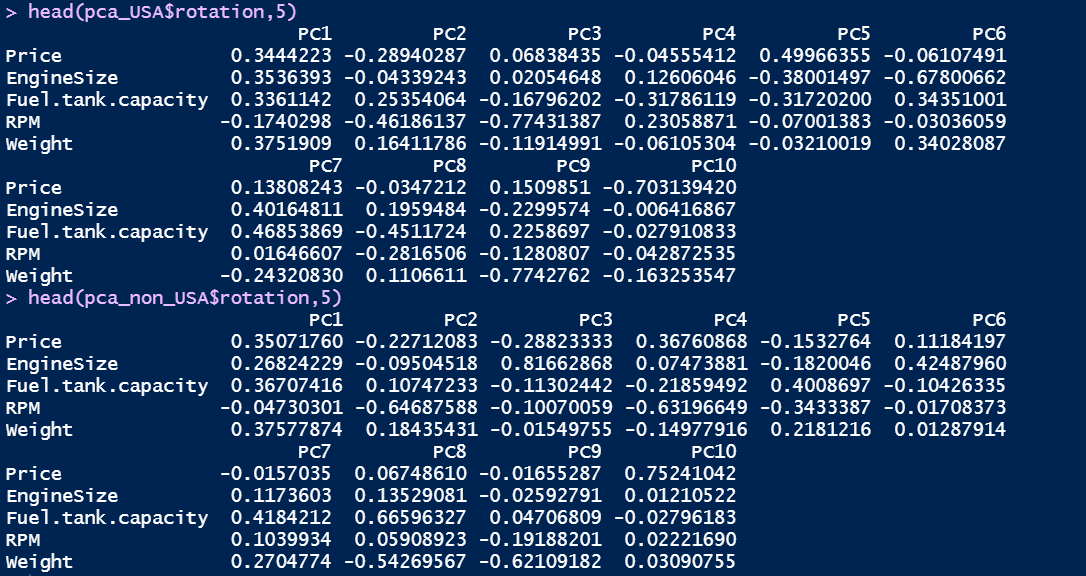


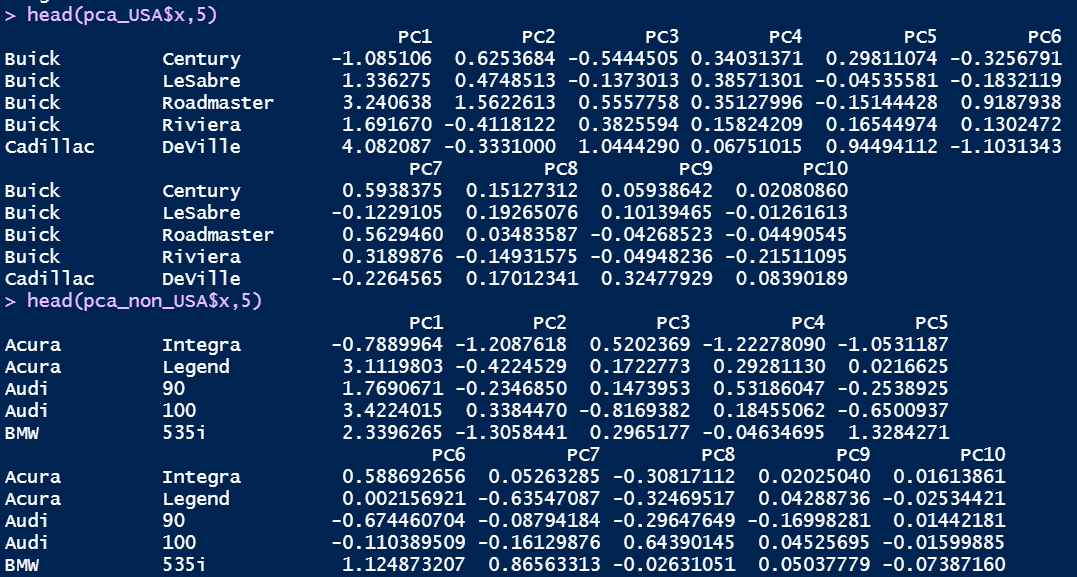
The eigenvalues are different for the different Origins. This applies for the eigenvectors as well. Next, we need to compare the amount of information between the two PCA models.



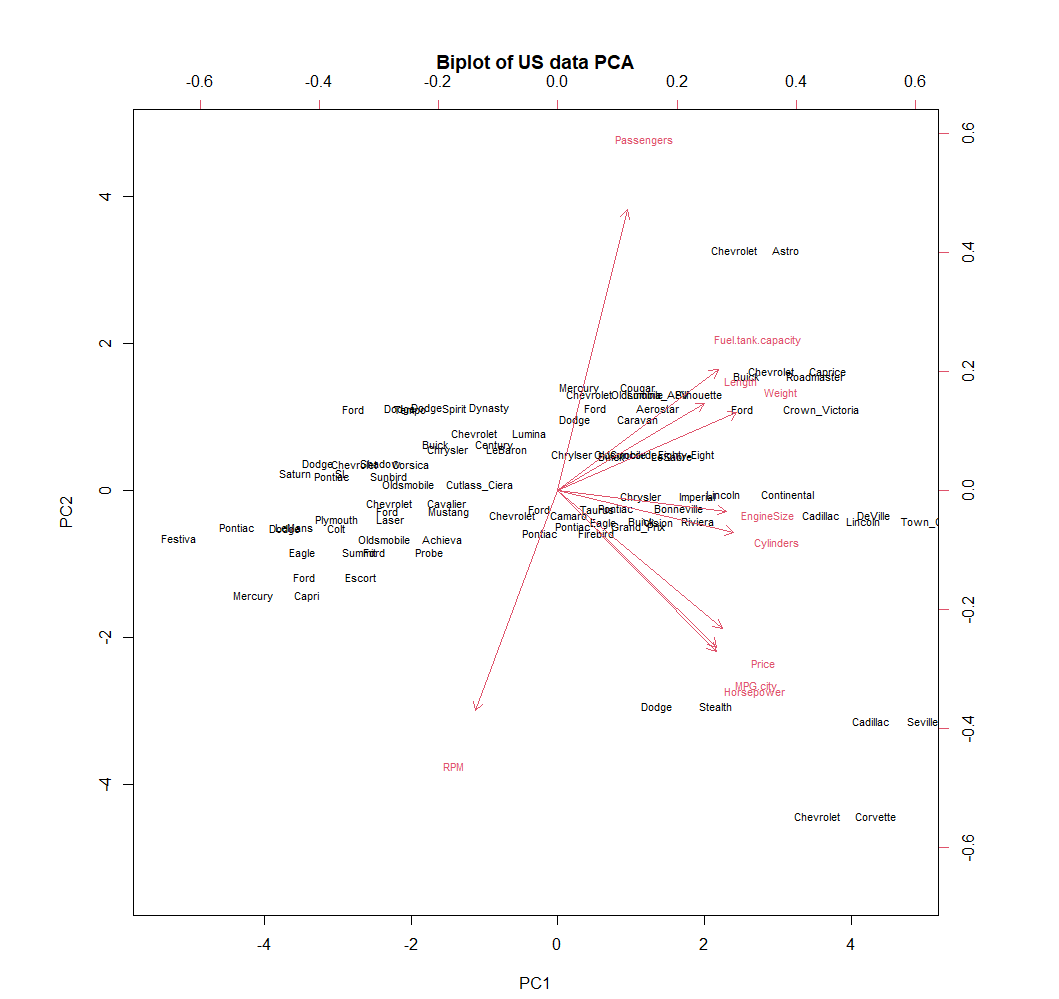
The amount of information varies between the different origins. Since we can observe differences in the other values, we need to check if there is a change in the factor loadings and scores as well.

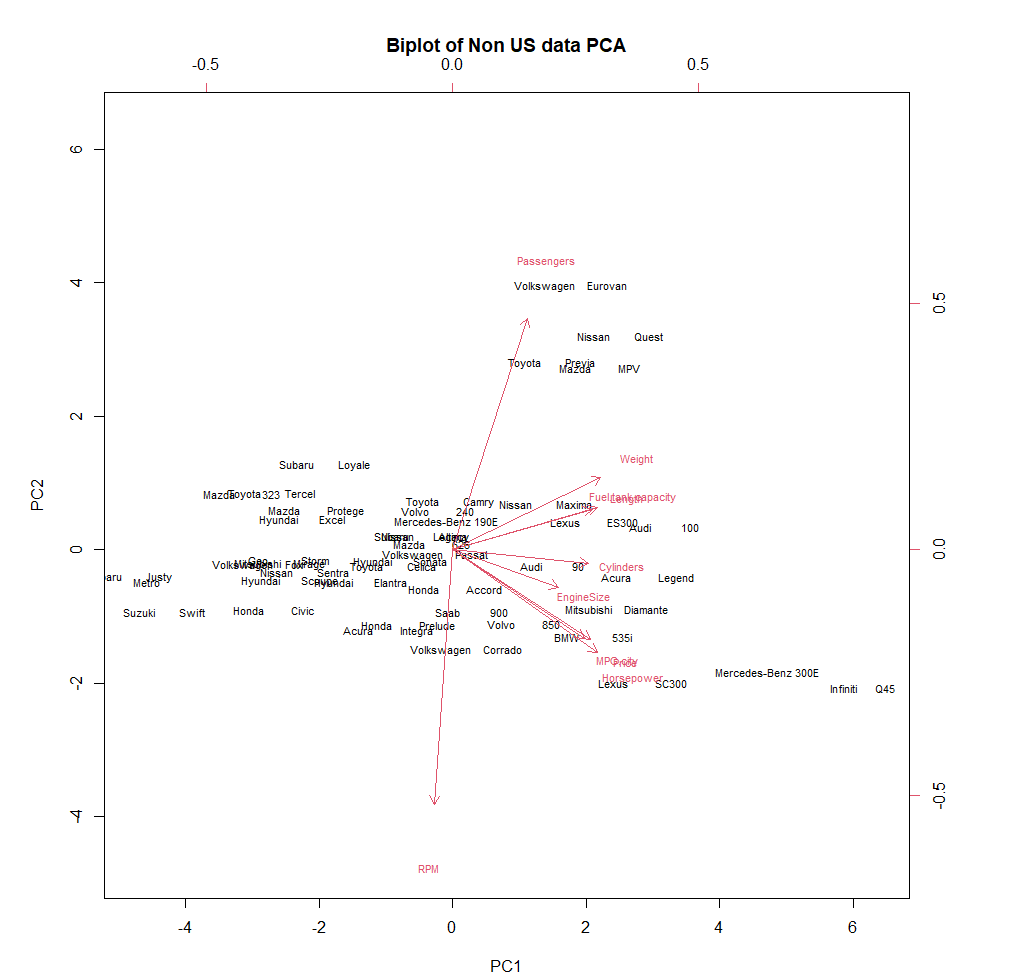






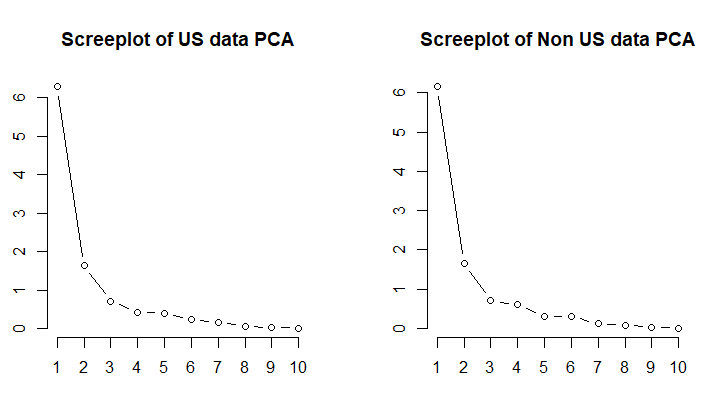
Most of the factor loadings and PCA scores of USA data are varying significantly from the values of Non-USA data. Next, we need to check the biplots to understand if there is any difference.





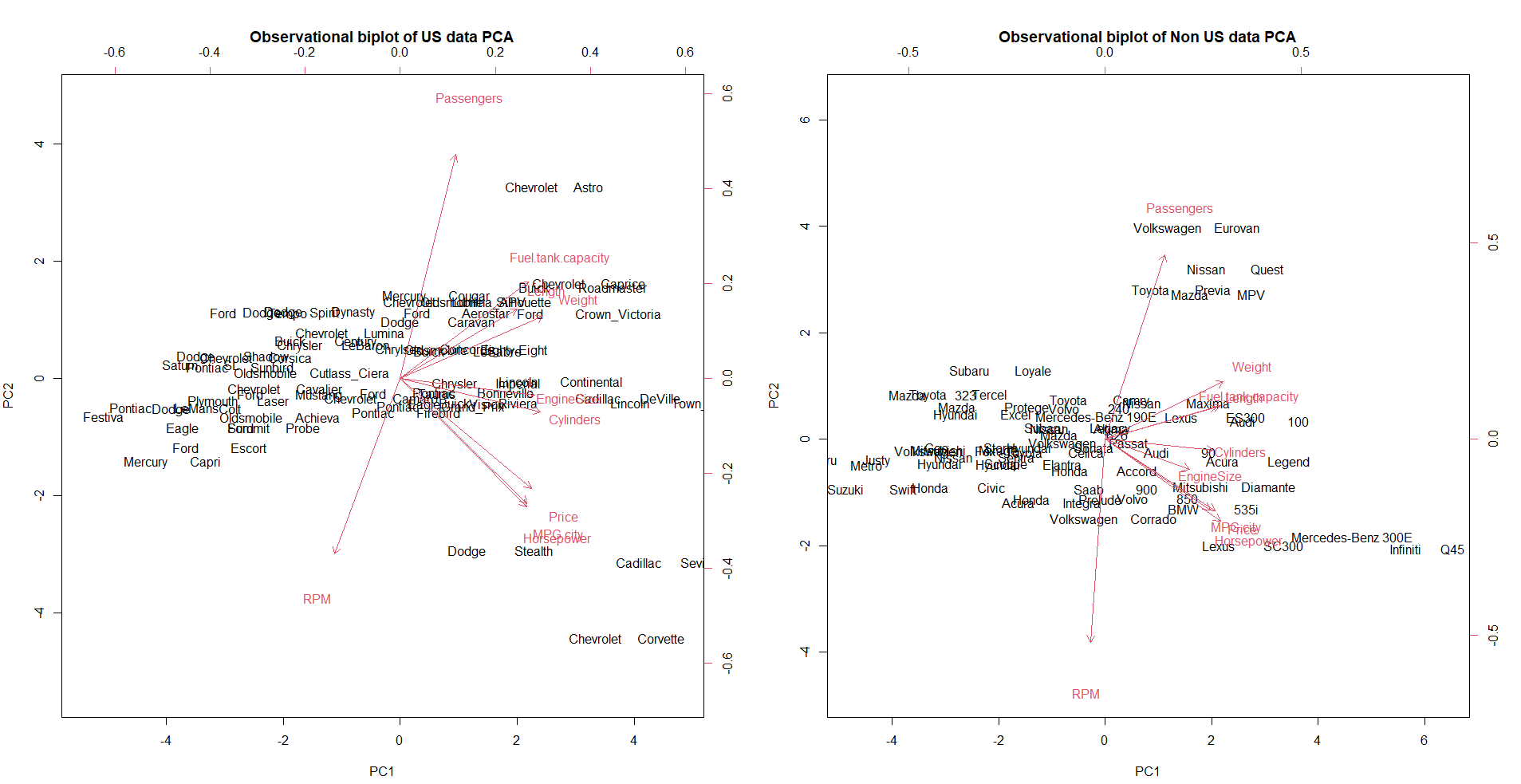
In both the biplots, the variables Passengers and RPM are more related to PC2 while the rest are more related to PC1.

Next, we need to generate screeplots to determine the number of components to be retained.



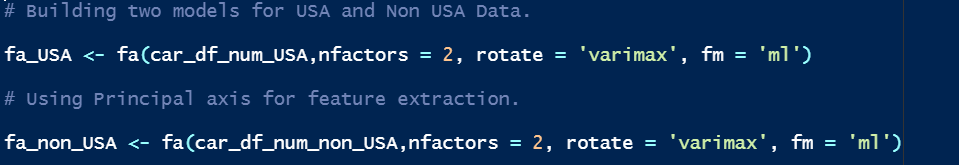
From the screeplots we can say that it is best to retain 3 components. After the second component, the graph almost saturates. The screeplots are identical between the PCA of the two datasets.

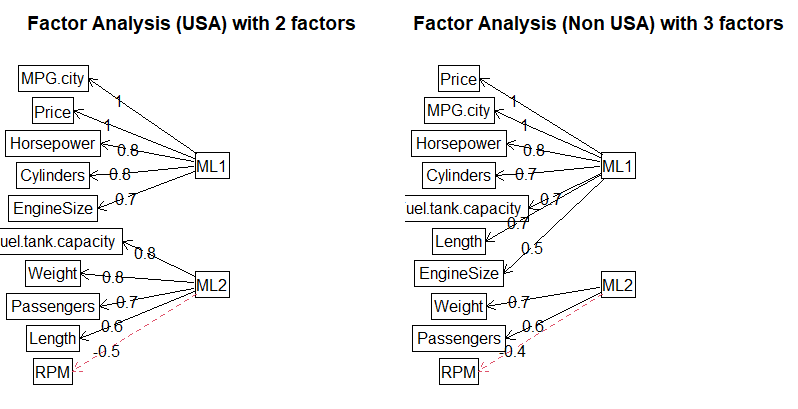
Lastly, we compare the observational biplots of the two PCA models.



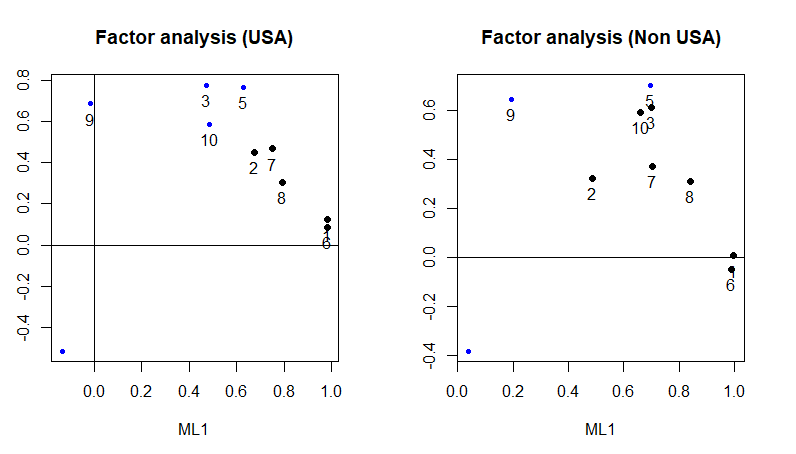
In both the biplots, most of the observations are pointed towards PC1 than PC2. This is due to the strong correlation between PC1 and observations.

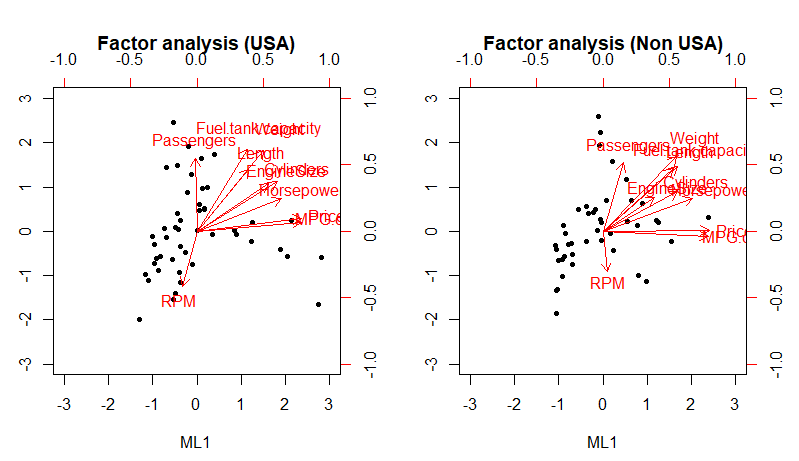
PCA is completed. Next, we perform FA to check the difference between the datasets.



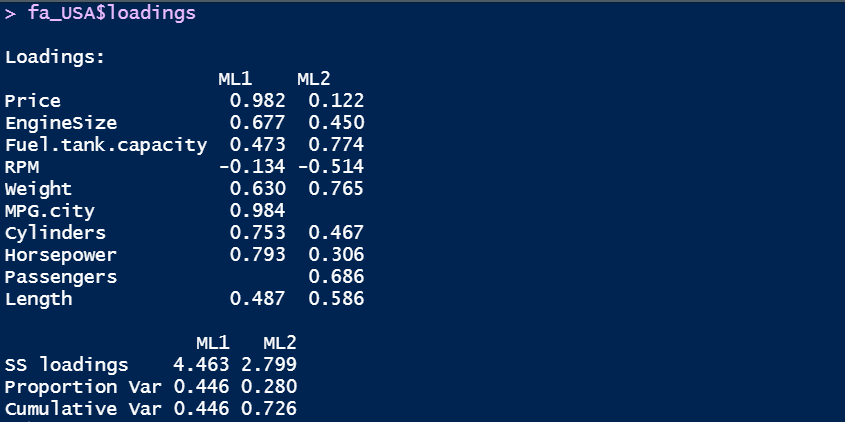


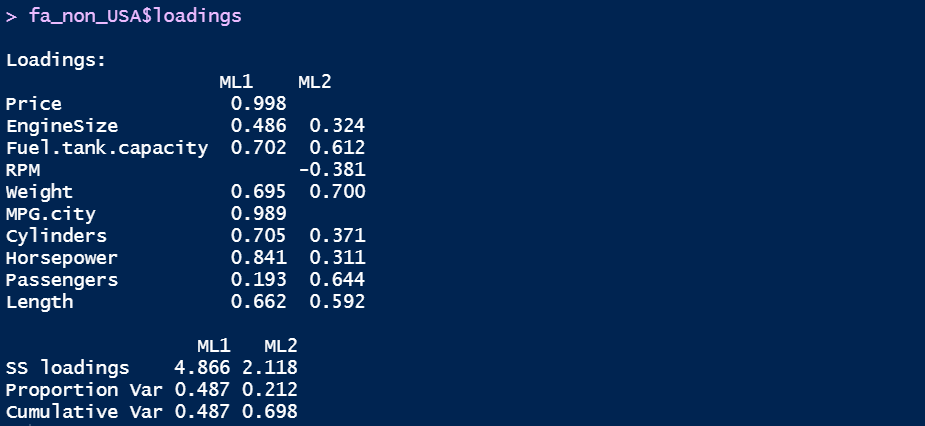
The path diagrams are similar between the USA and Non-USA data.





The plots are slightly similar between the USA and Non-USA data.





There is a minor difference in the factor loadings due to the difference in data. There are minor differences in the results of the two different groups. There are major differences with the results obtained previously for the dataset.

**References:**

Dr. Francisco’s lecture notes and code shared in labs were used as reference for solving this assignment.