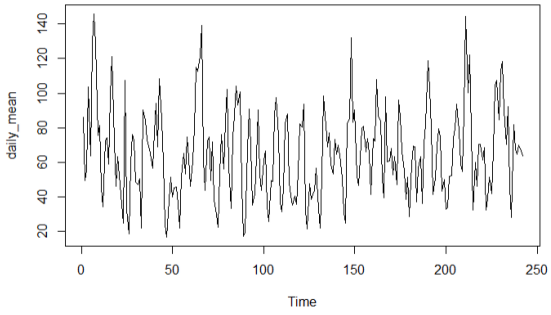
Time Series and Forecasting - Coursework Assignment

Student Number: 20416586

Student Name: Yutian Sun

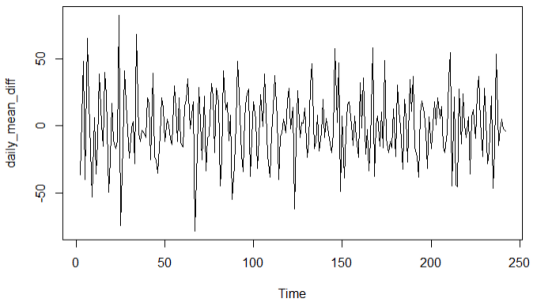
Question 1

Firstly I import the data from "a23_nox.csv"and read out the daily_mean_nox column. Then produce a time plot of the data.



It can be seen from the time plot that the value fluctuates up and down around to the zero, which means the data is not stationary. So we have to define the time series daily_mean_diff as the first difference of the time series daily_mean.

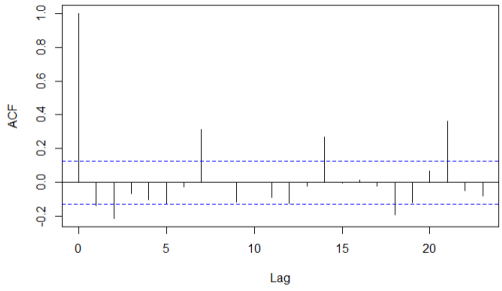
A time plot of the first-differenced daily_mean dataset is shown below.

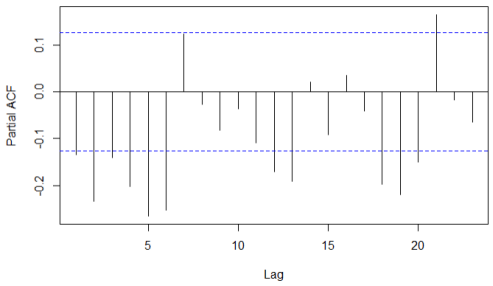


After produce the time plot again, we can see that the time plot has a constant mean and appears to show constant variability over time. It means that the first-difference of the data is stationary now.

Then plot the sample ACF and PACF against the lag for the daily_mean_diff time series.

A plot of the sample ACF and the sample PACF against the lag is shown below.



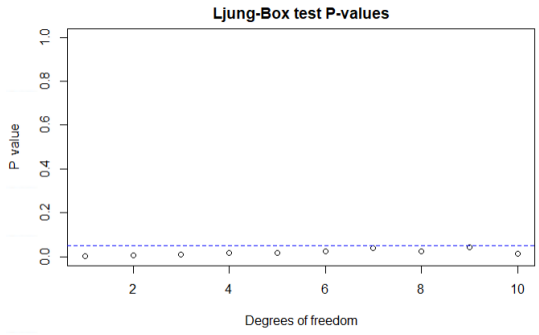


We are not sure whether it is an AR model or a MA model because neither the ACF plot nor the PCF plot has a sharp cut-off, and both decay slowly, both images show tails off. Also, this time series data is stationary and not a white noise process now. Therefore, I choose the ARMA model.

After that, we see that the sample ACF appears to drop to zero after lag 2, we might begin to think that the observed data originate from an ARMA(6, 2) or ARMA(6, 1) process. It was eventually found that ARMA (9, 1) has the smallest AIC value. Finally, we'll examine the Ljung-Box test P-values with respect to the model residuals extracted after build the model.

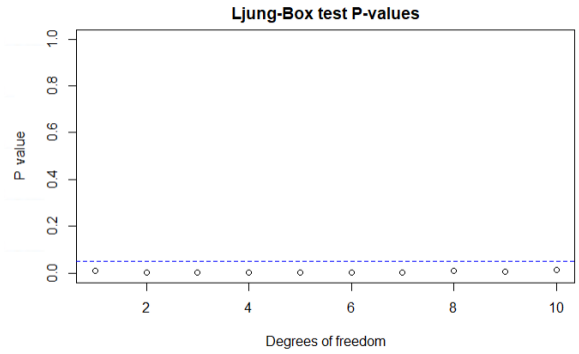
ARMA(6, 2)

```
Call:
arima(x = daily_mean_diff, order = c(6, 0, 2), method = "ML")

sigma^2 estimated as 501.1:  log likelihood = -1091.78, aic = 2203.56
```



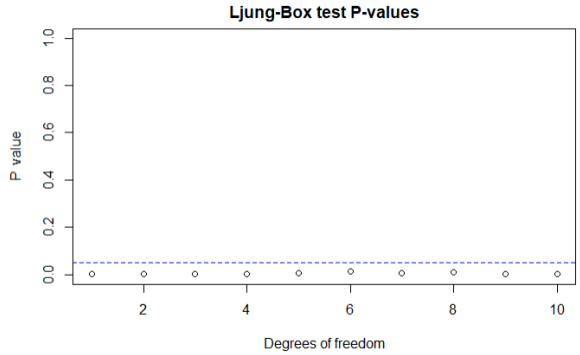
ARMA(6, 1)

```
Call:
arima(x = daily_mean_diff, order = c(6, 0, 1), method = "ML")

sigma^2 estimated as 505.6:  log likelihood = -1092.79, aic = 2203.58
```



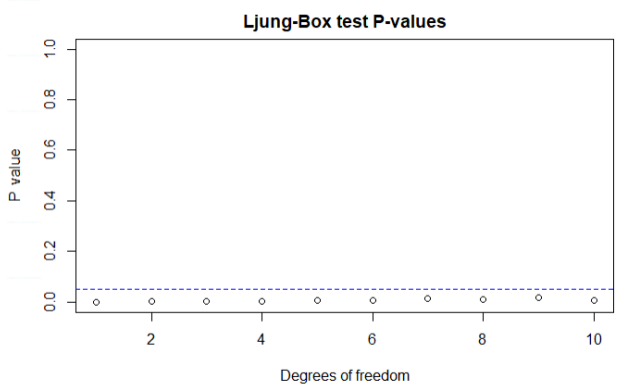
ARMA(7, 2)

```
Call:
arima(x = daily_mean_diff, order = c(7, 0, 2), method = "ML")

sigma^2 estimated as 497.5:  log likelihood = -1091.03,  aic = 2204.06
```



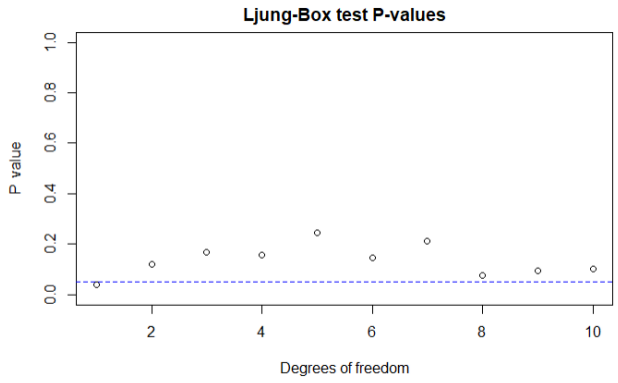
ARMA(7, 1)

```
Call:
arima(x = daily_mean_diff, order = c(7, 0, 1), method = "ML")

sigma^2 estimated as 503.8:  log likelihood = -1092.37,  aic = 2204.73
```



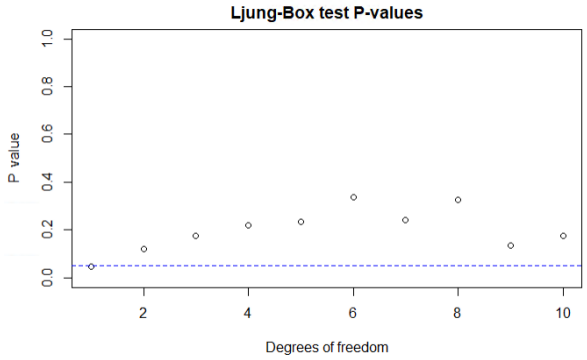
ARMA(8, 2)

```
Call:
arima(x = daily_mean_diff, order = c(8, 0, 2), method = "ML")

sigma^2 estimated as 440.1:  log likelihood = -1078.55,  aic = 2181.09
```



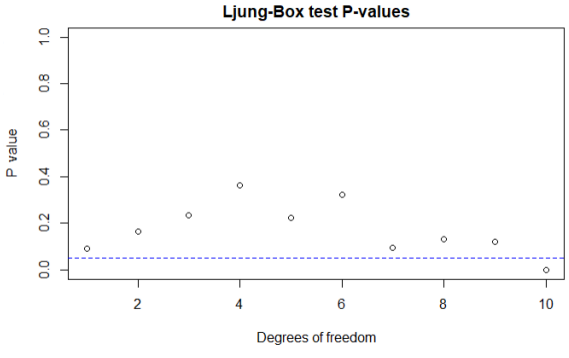
ARMA(8, 1)

```
Call:
arima(x = daily_mean_diff, order = c(8, 0, 1), method = "ML")

Coefficients:
         ar1      ar2     ar3      ar4      ar5      ar6     ar7      ar8      ma1  intercept
      0.5915  -0.1197  0.0228  -0.0841  -0.0363  -0.0039  0.2921  -0.2413  -1.0000     0.0133
s.e.  0.0625   0.0708  0.0712   0.0724   0.0727   0.0736  0.0734   0.0653   0.0165     0.0338

sigma^2 estimated as 448.8:  log likelihood = -1080.55   aic = 2183.11
```



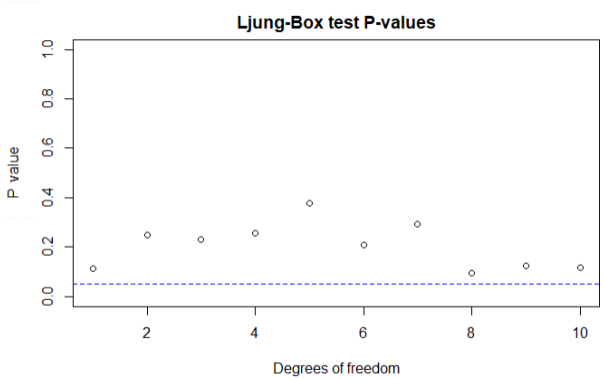
ARMA(9, 2)

```
Call:
arima(x = daily_mean_diff, order = c(9, 0, 2), method = "ML")


sigma^2 estimated as 444.3:  log likelihood = -1079.44,  aic = 2184.88
```



ARMA(9, 1)

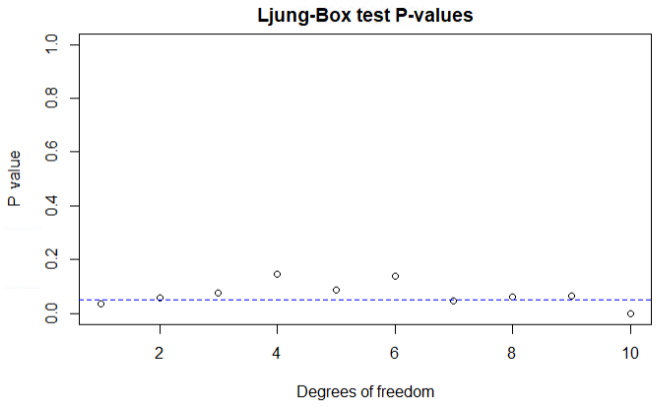
```
Call:
arima(x = daily_mean_diff, order = c(9, 0, 1), method = "ML")

Coefficients:
         ar1      ar2     ar3      ar4      ar5      ar6     ar7      ar8      ar9      ma1  intercept
      0.5663  -0.0909  0.0218  -0.0884  -0.0452  -0.0037  0.2781  -0.1794  -0.1068  -1.0000     0.0159
s.e.  0.0641   0.0727  0.0707   0.0720   0.0724   0.0731  0.0735   0.0757   0.0671   0.0191     0.0304

sigma^2 estimated as 443.5:  log likelihood = -1079.3,  aic = 2182.59
```



The p-values of ARMA (9, 1) are all greater than 0. Another attempt at ARMA (10, 1) finds that the p-values are no longer all greater than 0, so the optimal model ARMA (9, 1) is finally be taken.

ARMA(10, 1)

```
Call:
arima(x = daily_mean_diff, order = c(10, 0, 1), method = "ML")

sigma^2 estimated as 442.4:  log likelihood = -1079.04   aic = 2184.07
```



In the ARMA(8, 1) and ARMA(9, 1) models, they both have three test statistics greater than 2, implying that three elements are retained in the model, that means the ARMA(9, 1) model has fewer parameters so, despite the tiny increase in AIC, we should probably choose to fit an ARMA(8, 1) model here.

The equation of the final fitted model should be:

$X_t=0.5915X_{t-1}-0.1197X_{t-2}+0.0228X_{t-3}-0.0841X_{t-4}-0.0363X_{t-5}-0.0039X_{t-6}+0.2921X_{t-7}-0.2413X_{t-8}+Z_t-Z_{t-1}$

R code

#1

#import data

```{r}
a23_nox<-read.csv("a23_nox.csv")
daily_mean<-ts(a23_nox$daily_mean_nox)
ts.plot(daily_mean)
```

#first-differenced data to make the data stationary

```{r}
daily_mean_diff<-diff(daily_mean)
ts.plot(daily_mean_diff)
```

#plot ACF and PACF to decide which model is more reasonable

```{r}
acf(daily_mean_diff)
pacf(daily_mean_diff)
```

#compare AIC values to determine appropriate p, q values

```{r}
#model_a23<-arima(daily_mean_diff,order=c(6,0,2),method="ML")
#model_a23
#model_a23<-arima(daily_mean_diff,order=c(6,0,1),method="ML")
#model_a23
#model_a23<-arima(daily_mean_diff,order=c(7,0,2),method="ML")
#model_a23
#model_a23<-arima(daily_mean_diff,order=c(7,0,1),method="ML")
#model_a23
#model_a23<-arima(daily_mean_diff,order=c(8,0,2),method="ML")
#model_a23
model_a23<-arima(daily_mean_diff,order=c(8,0,1),method="ML")
```

model_a23
#model_a23<-arima(daily_mean_diff,order=c(9,0,2),method="ML")
#model_a23
#model_a23<-arima(daily_mean_diff,order=c(9,0,1),method="ML")
#model_a23
#model_a23<-arima(daily_mean_diff,order=c(10,0,1),method="ML")
#model_a23
```

#extract the model residuals and calculate the Ljung-Box test P-values
```{r}
#resid.model_a23<-residuals(model_a23)
#ARMA62.LB<-LB_test(resid.model_a23,max.k=18,p=6,q=2)
#plot(ARMA62.LB$deg_freedom,ARMA62.LB$LB_p_value,xlab="Degrees of freedom",ylab="P  value",main="Ljung-Box test P-values",ylim=c(0,1))
#abline(h=0.05,col="blue",lty=2)

#resid.model_a23<-residuals(model_a23)
#ARMA61.LB<-LB_test(resid.model_a23,max.k=17,p=6,q=1)
#plot(ARMA61.LB$deg_freedom,ARMA61.LB$LB_p_value,xlab="Degrees of freedom",ylab="P  value",main="Ljung-Box test P-values",ylim=c(0,1))
#abline(h=0.05,col="blue",lty=2)

#resid.model_a23<-residuals(model_a23)
#ARMA72.LB<-LB_test(resid.model_a23,max.k=19,p=7,q=2)
#plot(ARMA72.LB$deg_freedom,ARMA72.LB$LB_p_value,xlab="Degrees of freedom",ylab="P  value",main="Ljung-Box test P-values",ylim=c(0,1))
#abline(h=0.05,col="blue",lty=2)

#resid.model_a23<-residuals(model_a23)
#ARMA71.LB<-LB_test(resid.model_a23,max.k=18,p=7,q=1)
#plot(ARMA71.LB$deg_freedom,ARMA71.LB$LB_p_value,xlab="Degrees of freedom",ylab="P  value",main="Ljung-Box test P-values",ylim=c(0,1))
#abline(h=0.05,col="blue",lty=2)

```
#resid.model_a23<-residuals(model_a23)
#ARMA82.LB<-LB_test(resid.model_a23,max.k=20,p=8,q=2)
#plot(ARMA82.LB$deg_freedom,ARMA82.LB$LB_p_value,xlab="Degrees of
freedom",ylab="P    value",main="Ljung-Box test P-values",ylim=c(0,1))
#abline(h=0.05,col="blue",lty=2)

#resid.model_a23<-residuals(model_a23)
#ARMA81.LB<-LB_test(resid.model_a23,max.k=19,p=8,q=1)
#plot(ARMA81.LB$deg_freedom,ARMA81.LB$LB_p_value,xlab="Degrees of
freedom",ylab="P    value",main="Ljung-Box test P-values",ylim=c(0,1))
#abline(h=0.05,col="blue",lty=2)

#resid.model_a23<-residuals(model_a23)
#ARMA92.LB<-LB_test(resid.model_a23,max.k=21,p=9,q=2)
#plot(ARMA92.LB$deg_freedom,ARMA92.LB$LB_p_value,xlab="Degrees of
freedom",ylab="P    value",main="Ljung-Box test P-values",ylim=c(0,1))
#abline(h=0.05,col="blue",lty=2)

resid.model_a23<-residuals(model_a23)
ARMA91.LB<-LB_test(resid.model_a23,max.k=20,p=9,q=1)#k=10+p+q
plot(ARMA91.LB$deg_freedom,ARMA91.LB$LB_p_value,xlab="Degrees of
freedom",ylab="P    value",main="Ljung-Box test P-values",ylim=c(0,1))
abline(h=0.05,col="blue",lty=2)

#resid.model_a23<-residuals(model_a23)
#ARMA101.LB<-LB_test(resid.model_a23,max.k=21,p=10,q=1)#k=10+p+q
#plot(ARMA101.LB$deg_freedom,ARMA101.LB$LB_p_value,xlab="Degrees of
freedom",ylab="P    value",main="Ljung-Box test P-values",ylim=c(0,1))
#abline(h=0.05,col="blue",lty=2)
```

Question 2

Executive Summary

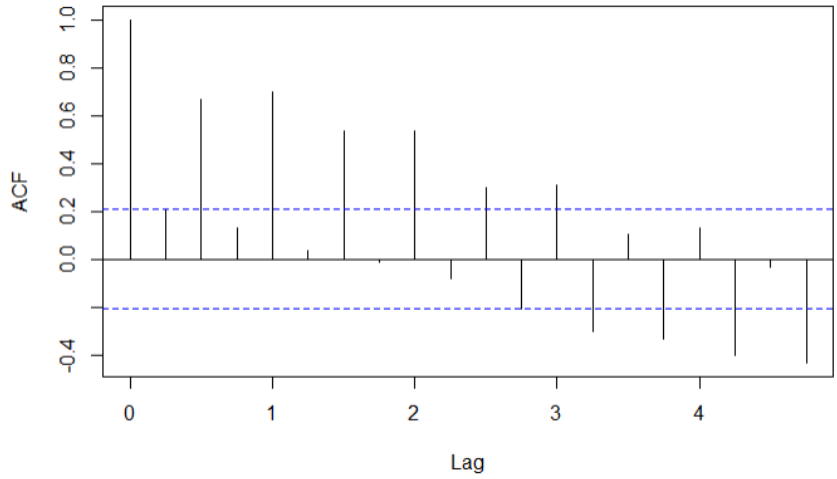
The purpose of this report is to identify a suitable time series forecasting model to forecast the numbers of new cars registered in England in the future. I will use the data of new cars registered in England from Q1-Q4 2001 to Q1-Q3 2022 as sample data for constructing the prediction model. I will firstly to choose the type of time series forecasting model based on currently data by observing the pattern of ACF plot and PACF plot, and also use auto.arima() function. Then I will fit the forecasting model to the optimum by observing the AIC values and test the model for accuracy and reasonableness of the forecasting effect by using Ljung-Box test. Finally use suitable seasonal time forecasting model ARIMA(1, 0, 1)x$(0, 1, 1)_4$ to forecast the number of new cars that will be registered in Q4 2022 and Q1, Q2, Q3 2023.

1 Main Purpose

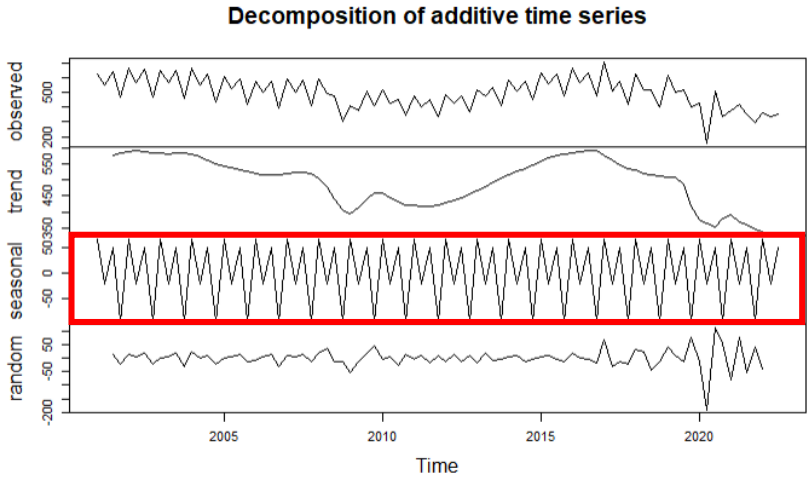
The main aim of my work is to analyse a dataset that contains the number of new cars registered in England and constructing a suitable time series model to forecast some future numbers of new cars registered.

2 Model Construction

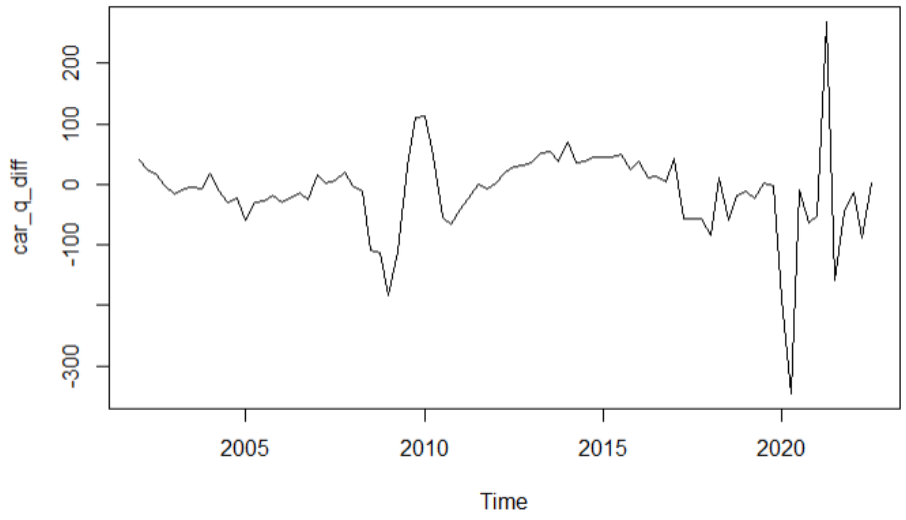
Firstly I create a time series in four quarterly increments. Then plot the ACF graph of original data, we can see that there is a clear seasonality.



There is a clear pattern of variation in the seasonal factors decomposed be found that can show seasonality by observe the graph named "Decomposition of additive time series".



Also, the data is weak stationary after differencing.

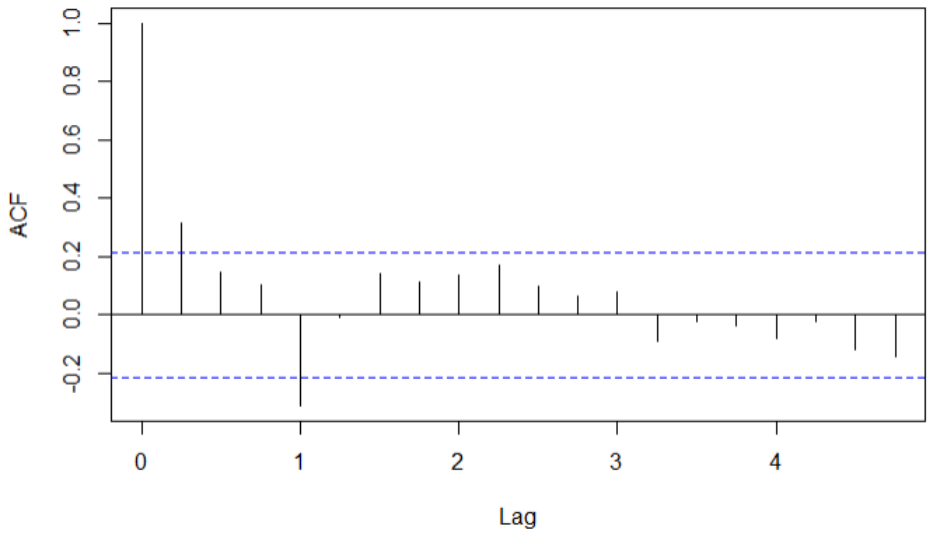


Hence, we can use the SARIMA(ARIMA(p, d, q)x(P, D, Q)$_s$) model for forecasting.

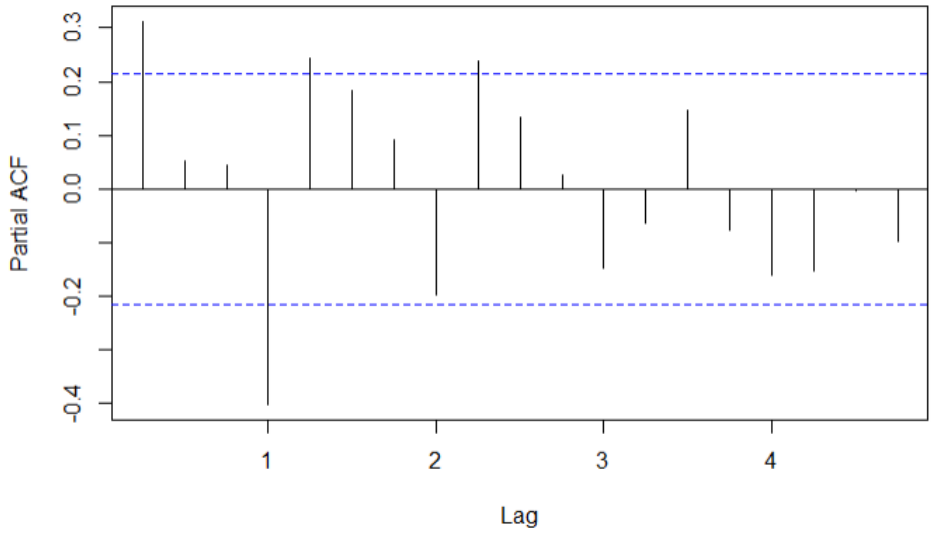
Secondly, we need to define the time series car_q_diff as the first difference of the time series car_q.

The ACF and PACF plots are then plotted after seasonal differencing of the car_q_diff time series. It can be seen that there is no longer seasonality.

A plot of the sample ACF against the lag is shown below.



A plot of the sample PACF against the lag is shown below.



According to the ACF plot, the ACF value with lag1 is outside the confidence interval and the ACF value with a lag of 4 is inside the confidence interval, so Q may be euqal to 1.

According to the PACF plot, the PACF value with lag1 is outside the confidence interval and the PACF value with a lag of 4 is inside the confidence interval, so P may be equal to 1.

On the other hand, the ACF plot shows a rapid convergence to 0 in lag1, which means q may be equal to 1.

Also, the PACF plot shows a rapid convergence to 0 in lag2, which means p may be equal to 2.

Then the result is shown below, the preliminary model is ARIMA(2, 0, 1)×$(1, 1, 1)_4$. We need to observe the value of AIC in the table below, because it is a measure of the goodness of fit of a statistical model, the smaller value it have, the better the model. And the AIC of this model is 915.03. We are not sure whether it is the smallest AIC value, so the following step is still needed.

```
Call:
arima(x = car_q, order = c(2, 0, 1), seasonal = list(order = c(1, 1, 1), period = 4))

Coefficients:
         ar1     ar2      ma1     sar1     sma1
      0.9687  0.0137  -0.5304  0.0052  -0.9342
s.e.  0.2072  0.1873   0.1699  0.1320   0.1544

sigma^2 estimated as 2881:  log likelihood = -451.52,  aic = 915.03
```

For greater accuracy, I also use the auto.arima() function to help me further determine the final model. The results are shown in the table below.

```
Series: car_q
ARIMA(1,0,1)(0,1,1)[4]

Coefficients:
         ar1      ma1     sma1
      0.9826  -0.5387  -0.9318
s.e.  0.0504   0.0968   0.1388

sigma^2 estimated as 2991:  log likelihood=-451.52
AIC=911.04   AICc=911.56   BIC=920.72
```

```
Call:
arima(x = car_q_diff, order = c(1, 0, 1), seasonal = list(order = c(0, 1, 1),
    period = 4))

Coefficients:
         ar1      ma1     sma1
      0.4576  -0.1580  -1.0000
s.e.  0.2645   0.2836   0.0711

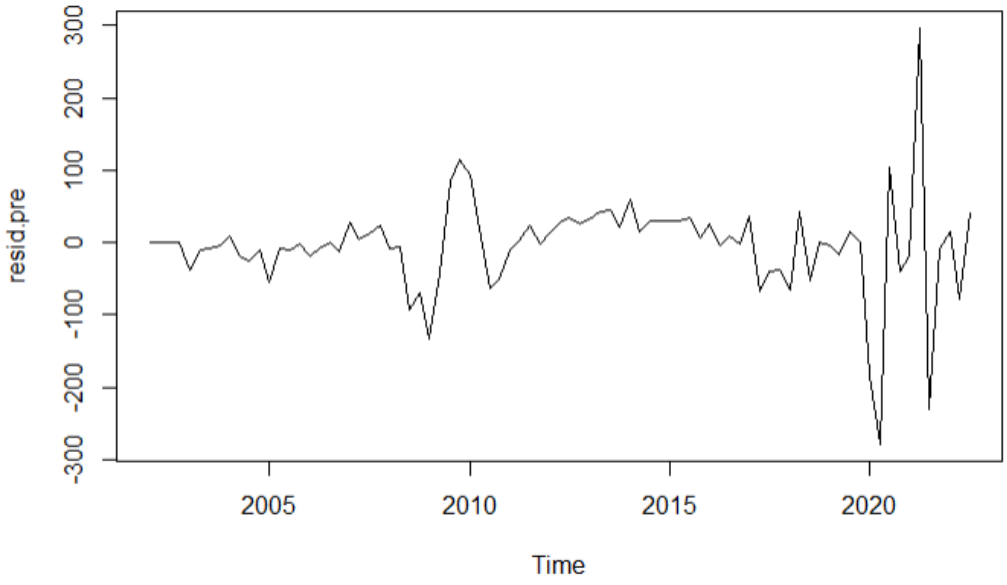
sigma^2 estimated as 4903:  log likelihood = -453.83,  aic = 915.65
```

We note the very slight reduction in AIC for this model (AIC =915.65) compared to the ARIMA(2, 0, 1)×(1, 1, 1)$_4$ model (AIC =915.03). Perhaps this might cause us to favour the ARIMA(2, 0, 1)×(1, 1, 1)$_4$ model over the ARIMA(1, 0, 1)×(0, 1, 1)$_4$ model. To examine this further, we perform a test of the hypotheses $H_0 : \theta_1 = 0$ versus $H_1 : \theta_1 \neq 0$. The test statistic is $|0.0137/0.1873| = 0.07$. Clearly, 0.07 is less than 2, so we would retain H_0 at the 5% level. Also, we perform a test of the hypotheses $H_0 : \theta_2 = 0$ versus $H_1 : \theta_2 \neq 0$. The test statistic is $|0.0052/0.1320| = 0.04$. Clearly, 0.04 is less than 2, so we would retain H0 at the 5% level. All that means the ARIMA(2, 0, 1)x(1, 1, 1)4 model has fewer parameters so, despite the tiny increase in AIC, we'd probably choose to fit an ARIMA(1, 0, 1)x(0, 1, 1)$_4$ model

here. Therefore, the model ARIMA(1, 0, 1)x(0, 1, 1)$_4$ is more suitable to be a predictive model.

3 check model fit

Finally, we can examine the Ljung-Box test p-values with respect to the model residuals extracted after build the model.



we can see that the residuals plot has a constant variability over time.

As it shown in the result, p-value euqals to 0.9721 and greater than the test statistic 0.05, accept the alternative hypothis, which means the process is white noise.

```
        Box-Ljung test

data:  resid.pre
X-squared = 0.0023164, df = 1, p-value = 0.9616
```

All the above results showed that the model fits well, and we have already obtained a suitable prediction model: ARIMA(1, 0, 1)x(0, 1, 1)$_4$.

4 Conclusion and Associated Uncertainties

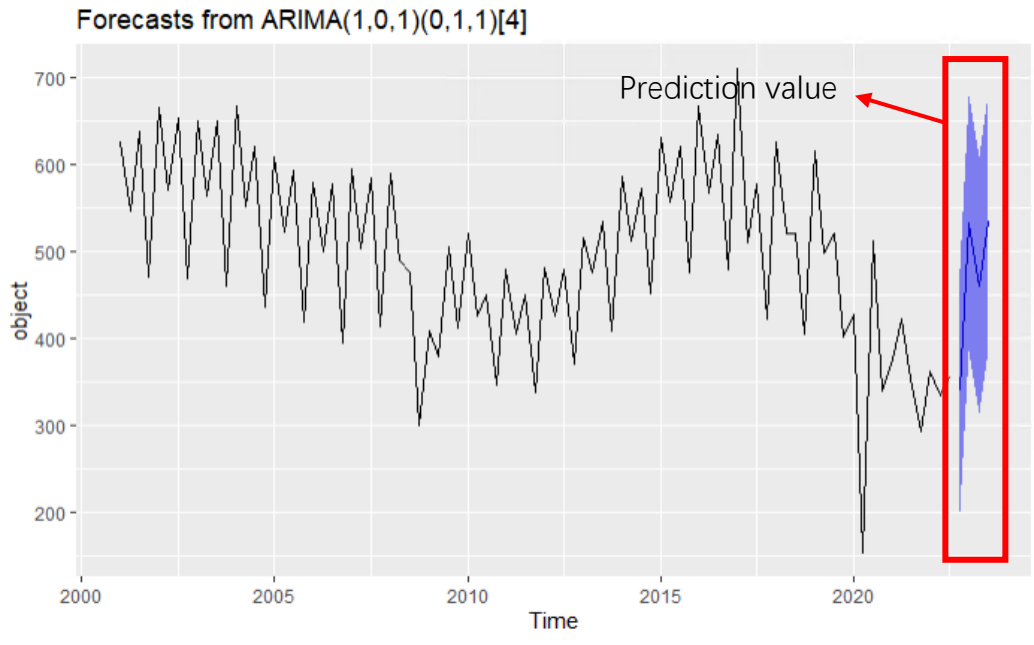
a) Conclusion

Follow the table, the number of new car registrations in Q4 2022 will be 339, the maximum number is 480 and the minumum number is 199 at the 95% confidence interval. The number of new car registrations in Q1 2023 will be 533, the maximum number is 679 and the minumum number is 386 at the 95% confidence interval. The number of new car registrations in Q2 2023 will be 461, the maximum number is 608 and the minumum number is 313 at the 95% confidence interval. The number of new car registrations in Q3 2023 will be 534, the maximum number is 682 and the minumum number is 386 at the 95% confidence interval.

	Point Forecast <dbl>	Lo 95 <dbl>	Hi 95 <dbl>
2022 Q4	339.3118	198.8631	479.7605
2023 Q1	532.8399	386.3632	679.3166
2023 Q2	460.6013	312.8692	608.3335
2023 Q3	534.4499	386.4586	682.4412

4 rows

The blue shaded area is the 95% confidence interval. We can also note that all the predicted values fall within the confidence interval, it may indicating that the model is correct and the prediction is good.



b) Uncertain Factors

As there are many factors that affect the registration of new cars, including people's wage levels, congestion, willingness to buy, etc. The prediction model so constructed is not entirely accurate, so it is not possible to determine whether these factors will affect the outcome of the number predicted by the model. In addition, different confidence intervals also affect our judgement of the comparison of the prediction and actual results.

R code

#2

#import data

```{r}
eng_car_reg<-read.csv("eng_car_reg.csv")
car_q<-ts(eng_car_reg$no_new_reg,start=c(2001,1),frequency=4)
#ts.plot(car_q)
acf(car_q)
line_c <- decompose(car_q) #Decomposition of time series data
plot(line_c)
```

#first-differenced data to make the data stationary and plot ACF and PACF to decide which model is more reasonable

```{r}
car_q_diff<-diff(car_q,lag=4)
acf(car_q_diff)
pacf(car_q_diff)
plot(car_q_diff)
```

#Fit the SARIMA model after observing ACF and PACF plot

```{r}
pre<-arima(car_q,order=c(2,0,1),seasonal = list(order=c(1,1,1),period=4))
pre
```

```{r}
#install.packages('forecast')
library(forecast)
```

#use the auto.arima() function to determine the final model

```{r}
auto.arima(car_q)
```

#Fit the ARIMA(1, 0, 1)x(0, 1, 1)4. model
```{r}
pre_f<-arima(car_q_diff,order=c(1,0,1),seasonal = list(order=c(0,1,1),period=4))
pre_f
```

#Plotting the residuals to check if the residuals are white noise
```{r}
resid.pre<-residuals(pre_f)
plot(resid.pre)
Box.test(resid.pre, type = "Ljung-Box")
```

#Forecast for the next 4 quarters and plot the forecasted values
```{r}
pre_car<-forecast(car_q,h=4,model=pre_f,level=95)
pre_car
autoplot(pre_car)
```