ISYE 6402 Homework 4

Background

For this data analysis, you will analyze the daily and weekly domestic passenger count arriving in Hawaii airports. File *DailyDomestic.csv* contains the *daily* number of passengers between May 2019 and February 2023 File *WeeklyDomestic.csv* contains the *weekly* number of passengers for the same time period. Here we will use different ways of fitting the ARIMA model while dealing with trend and seasonality.

```
library(lubridate)
library(mgcv)
library(tseries)
library(car)
```

Instructions on reading the data

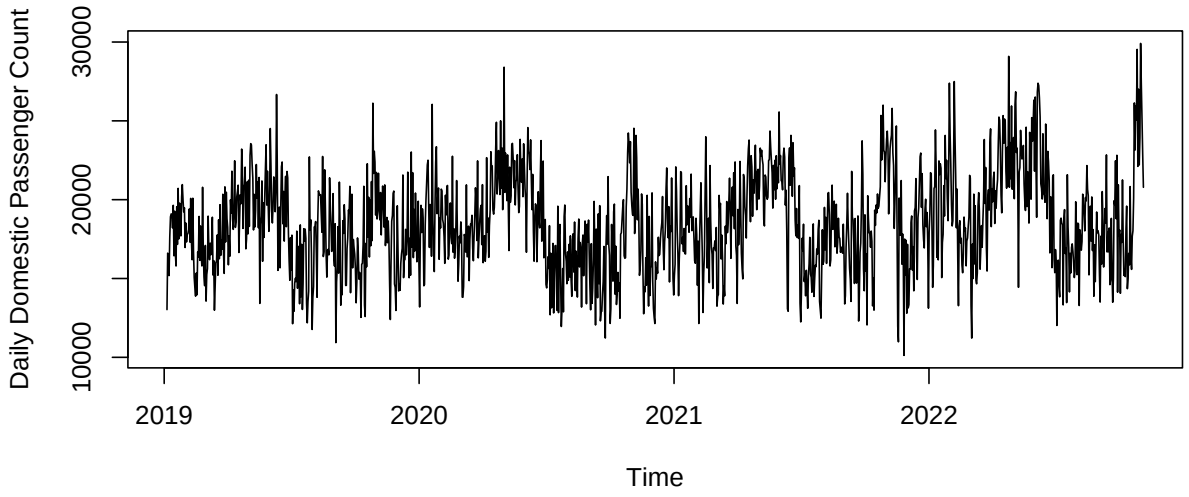
To read the data in `R`, save the file in your working directory (make sure you have changed the directory if different from the R working directory) and read the data using the `R` function `read.csv()`

```
daily <- read.csv("DailyDomestic.csv", head = TRUE)
daily$date <- as.Date(daily$date)
weekly <- read.csv("WeeklyDomestic.csv", head = TRUE)
weekly$week <- as.Date(weekly$week)
```

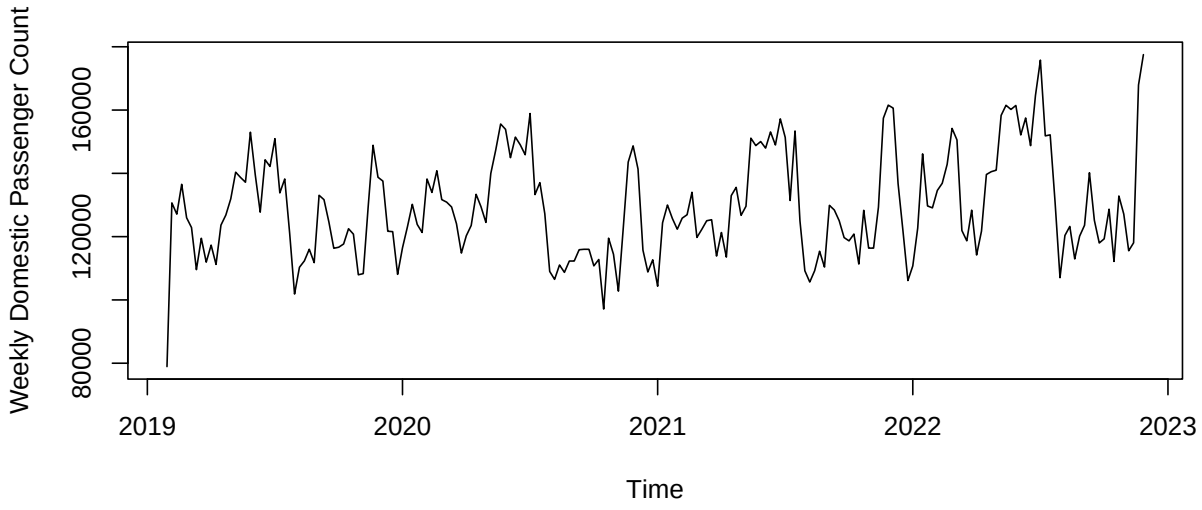
Question 1. Trend and seasonality estimation

1a. Plot the daily and weekly domestic passenger count separately. Do you see a strong trend and seasonality?

```
daily_ts <- ts(daily$domestic, start = c(2019, 05, 1), frequency = 365.25)
ts.plot(daily_ts,ylab="Daily Domestic Passenger Count")
```




```
weekly_ts <- ts(weekly$domestic, start = c(2019, 05, 1), frequency = 52)
ts.plot(weekly_ts,ylab="Weekly Domestic Passenger Count")
```



Response It is difficult to decipher, but upon visual inspection, there does not appear to be a strong trend or seasonality in either plot, but there might be some cyclical behavior.

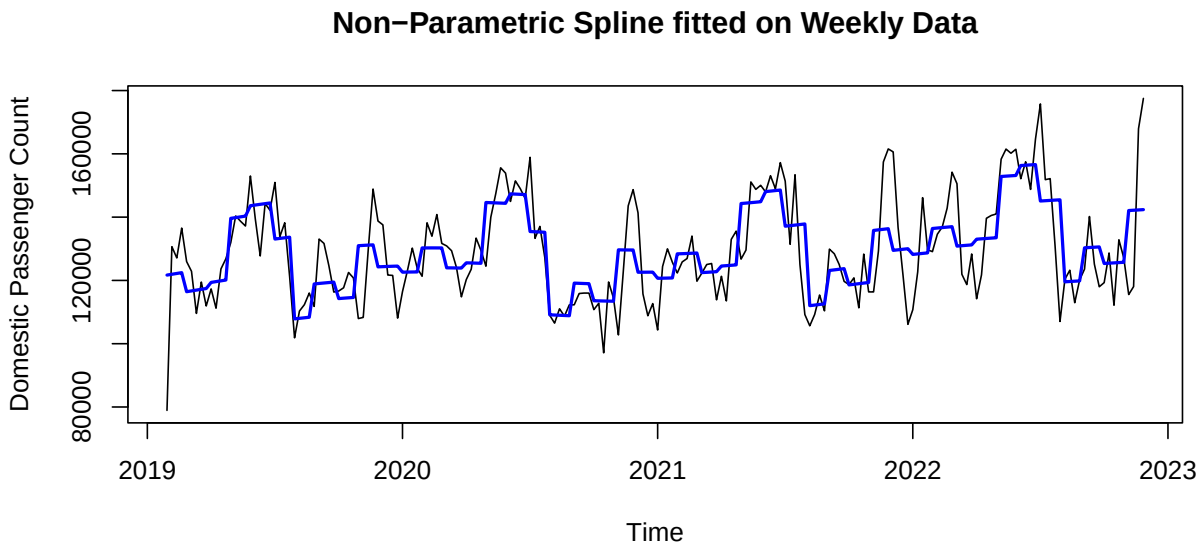
1b. (Trend and seasonality) Fit the *weekly* domestic passenger count with a non-parametric trend using splines and monthly seasonality using ANOVA. Is the seasonality significant? Plot the fitted values together with the original time series. Plot the residuals and the ACF of the residuals. Comment on how the model fits and on the appropriateness of the stationarity assumption of the residuals.

```
## X-axis points converted to 0-1 scale, common in nonparametric regression
time.pts = c(1:length(weekly_ts))
```

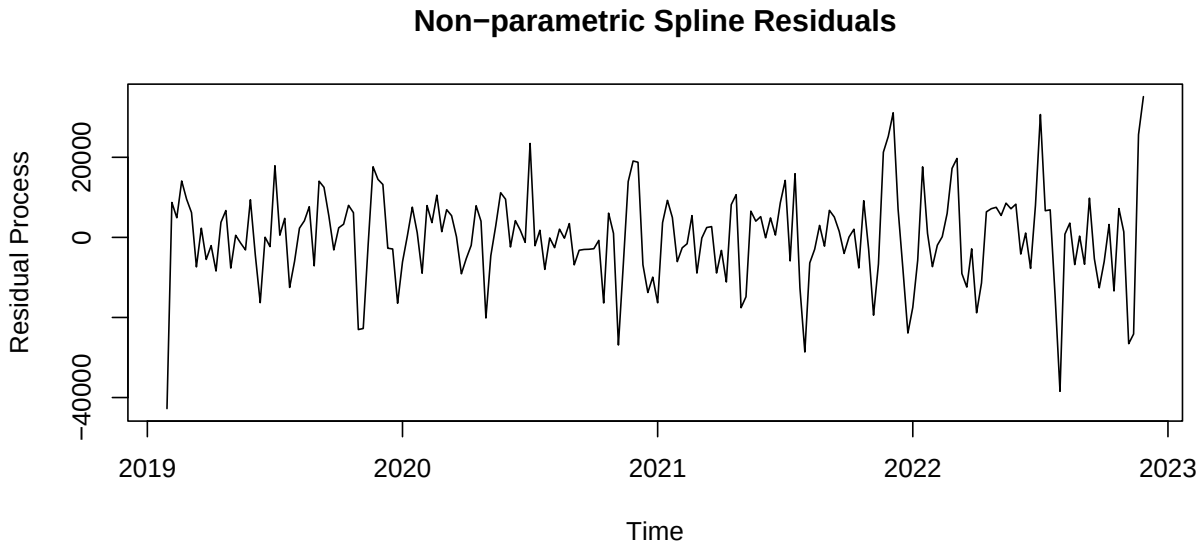
```r
time.pts = c(time.pts - min(time.pts))/max(time.pts)

## Splines Trend Estimation
#gam.fit = gam(weekly_ts~s(time.pts))
#weekly.fit.gam = ts(fitted(gam.fit),start=c(2019, 05, 1),frequency=52)
#ts.plot(weekly_ts,ylab="Domestic Passenger Count", main = "Spline")
#lines(weekly.fit.gam,lwd=2,col="red")

month = as.factor(format(weekly$week, "%b"))

#gam.fit = gam(weekly_ts~s(time.pts)+season(weekly_ts)-1)
gam.fit = gam(weekly_ts~s(time.pts)+month-1)

gam.fit.spline = ts((fitted(gam.fit)),start=c(2019, 05, 1),frequency=52)

summary(gam.fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## weekly_ts ~ s(time.pts) + month - 1
##
## Parametric coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## monthApr   123879       3502   35.38   <2e-16 ***
## monthAug   146112       2840   51.46   <2e-16 ***
## monthDec   121932       2932   41.59   <2e-16 ***
## monthFeb   132869       3070   43.28   <2e-16 ***
## monthJan   116608       2804   41.58   <2e-16 ***
## monthJul   126851       2932   43.26   <2e-16 ***
## monthJun   125166       2972   42.11   <2e-16 ***
## monthMar   125847       3370   37.34   <2e-16 ***
## monthMay   131439       2808   46.81   <2e-16 ***
## monthNov   111564       2922   38.18   <2e-16 ***
## monthOct   137565       2838   48.48   <2e-16 ***
## monthSep   149165       2920   51.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F  p-value
## s(time.pts) 3.899  4.737 6.721 2.16e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.468   Deviance explained = 99.2%
## GCV = 1.5551e+08  Scale est. = 1.4315e+08  n = 200
```

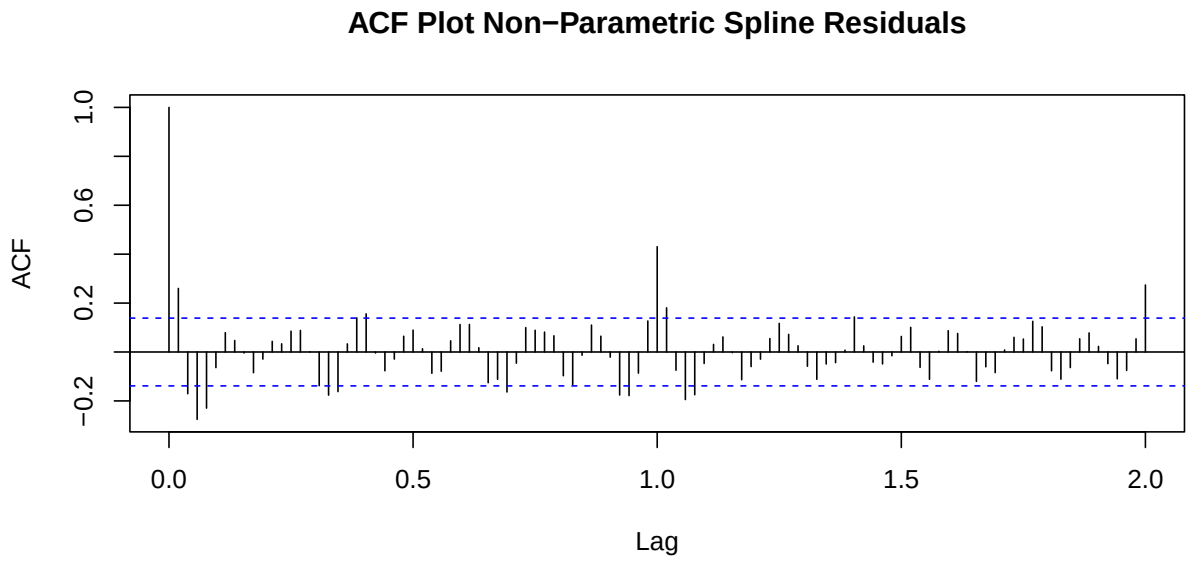
```r
ts.plot(weekly_ts,ylab="Domestic Passenger Count", main = "Non-Parametric Spline fitted on Weekly Data")
lines(gam.fit.spline,lwd=2,col="blue")
```




```
dif.fit.gam = ts((weekly_ts-fitted(gam.fit)),start=c(2019, 05, 1),frequency=52)
ts.plot(dif.fit.gam,ylab="Residual Process", main = "Non-parametric Spline Residuals")
```




```
acf(dif.fit.gam, lag.max = 52*2, main="ACF Plot Non-Parametric Spline Residuals")
```




```
residuals_weekly = weekly_ts-fitted(gam.fit)
```

Response:

Based on the coefficients computed, and the p-values associated with each month, it appears that the coefficients of the monthly seasonality are significant, with highly significant p-values.

For the acf plot of the residuals, there appears to be some autocorrelation that is significant, based on the dashed blue 95% confidence interval line. In addition, below the line, there also appears to be some small level of seasonality, as the autocorrelation alternates. The autocorrelation does not drop to zero immediately, so the third condition of stationarity is not met, and we cannot say that the residuals are plausibly stationary.

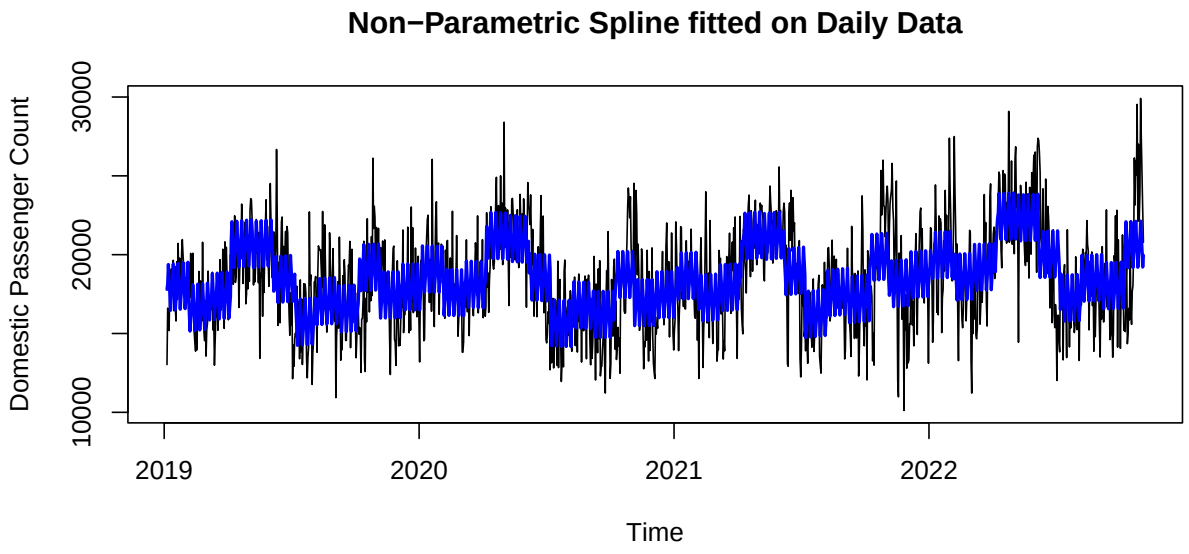
1c. (Trend and seasonality) This time fit the *daily* domestic passenger count with a non-parametric trend using splines, monthly and day-of-the-week seasonality using ANOVA. Plot the fitted values together with the original time series. Are the seasonal effects significant? Plot the residuals and the ACF of the residuals. Comment on how the model fits and on the appropriateness of the stationarity assumption of the residuals.

```
time.pts = c(1:length(daily_ts))
time.pts = c(time.pts - min(time.pts))/max(time.pts)


month = as.factor(format(daily$date, "%b"))
weekday = as.factor(weekdays(daily$date))

gam.fit = gam(daily_ts~s(time.pts)+month+weekday-1)

gam.fit.spline = ts((fitted(gam.fit)),start=c(2019, 05, 1),frequency=365.25)

summary(gam.fit)
```

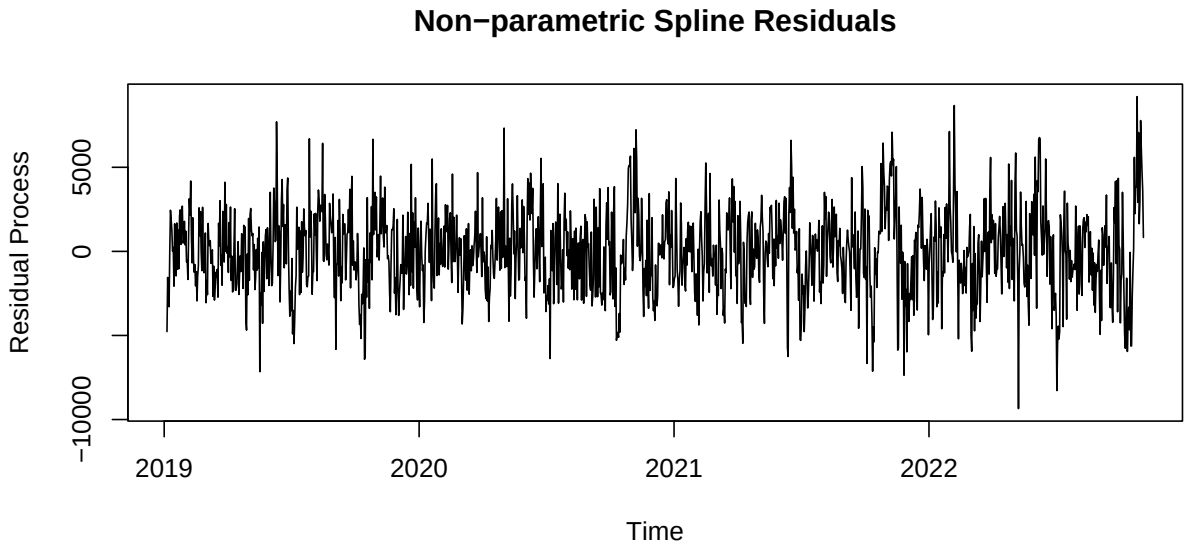
```
##
## Family: gaussian
## Link function: identity
##
```

```
## Formula:
## daily_ts ~ s(time.pts) + month + weekday - 1
##
## Parametric coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## monthApr           19524.5      314.7  62.037  < 2e-16 ***
## monthAug           22925.1      281.9  81.319  < 2e-16 ***
## monthDec           18810.0      282.2  66.666  < 2e-16 ***
## monthFeb           20816.8      295.9  70.340  < 2e-16 ***
## monthJan           18269.2      284.2  64.276  < 2e-16 ***
## monthJul           19767.0      281.3  70.281  < 2e-16 ***
## monthJun           19205.1      288.4  66.590  < 2e-16 ***
## monthMar           19036.4      313.8  60.664  < 2e-16 ***
## monthMay           20665.9      287.2  71.949  < 2e-16 ***
## monthNov           17602.8      284.4  61.898  < 2e-16 ***
## monthOct           20423.4      280.5  72.802  < 2e-16 ***
## monthSep           22794.4      283.9  80.287  < 2e-16 ***
## weekdayMonday      -2925.7      250.3 -11.687  < 2e-16 ***
## weekdaySaturday     -907.1      250.3  -3.624 0.000301 ***
## weekdaySunday      -1791.3      250.3  -7.156 1.35e-12 ***
## weekdayThursday    -1213.4      250.3  -4.847 1.39e-06 ***
## weekdayTuesday     -2182.4      250.4  -8.717  < 2e-16 ***
## weekdayWednesday   -1571.1      250.3  -6.276 4.64e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df    F p-value
## s(time.pts) 4.811  5.692 17.2  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.362   Deviance explained = 98.2%
## GCV = 6.368e+06  Scale est. = 6.2643e+06  n = 1400
```

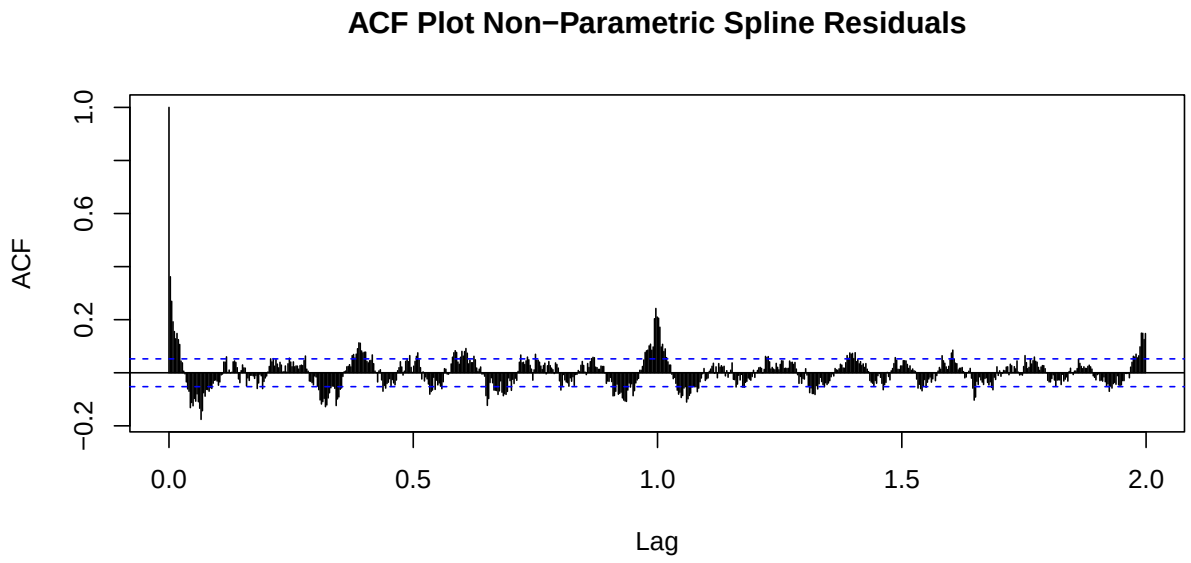
```r
ts.plot(daily_ts,ylab="Domestic Passenger Count", main = "Non-Parametric Spline fitted on Daily Data")
lines(gam.fit.spline,lwd=2,col="blue")
```




```
dif.fit.gam = ts((daily_ts-fitted(gam.fit)),start=c(2019, 05, 1),frequency=365.25)
ts.plot(dif.fit.gam,ylab="Residual Process", main = "Non-parametric Spline Residuals")
```




```
acf(dif.fit.gam, lag.max = 365.25*2, main="ACF Plot Non-Parametric Spline Residuals")
```




```
residuals_daily = daily_ts-fitted(gam.fit)
```

Response:

The day of the week and month seasonality coefficients are significant. Like in the previous question, the model does have a strong autocorrelation at a lag of 1 (one year).

Even with applying the day of the week seasonality to the model, there appears to be seasonal autocorrelation, and so the residuals are still not plausibly stationary.

Question 2. ARMA fitting and residual analysis

2a. (ARMA fitting) Fit a ARMA model with both AR and MA orders of 6 without intercept using the residual processes from Question 1b and 1c for the daily and weekly domestic passenger count, respectively. What are the coefficients of the fitted models? Are the fitted ARMA models causal? (Hint: Set include.mean = FALSE if using arima(). Use polyroot() to find the roots of a polynomial.)

```
modarma_weekly = arima(residuals_weekly, order=c(6,0,6), method="ML", include.mean = FALSE)
coefs_weekly = coef(modarma_weekly)
print(coefs_weekly)
```

```
##         ar1         ar2         ar3         ar4         ar5         ar6
## -0.31966876 -0.01156879 -0.87944011 -0.06847642  0.39483372 -0.31257392
##         ma1         ma2         ma3         ma4         ma5         ma6
##  0.61455733 -0.06264098  0.55079820 -0.07272817 -0.81005159 -0.04125763
```

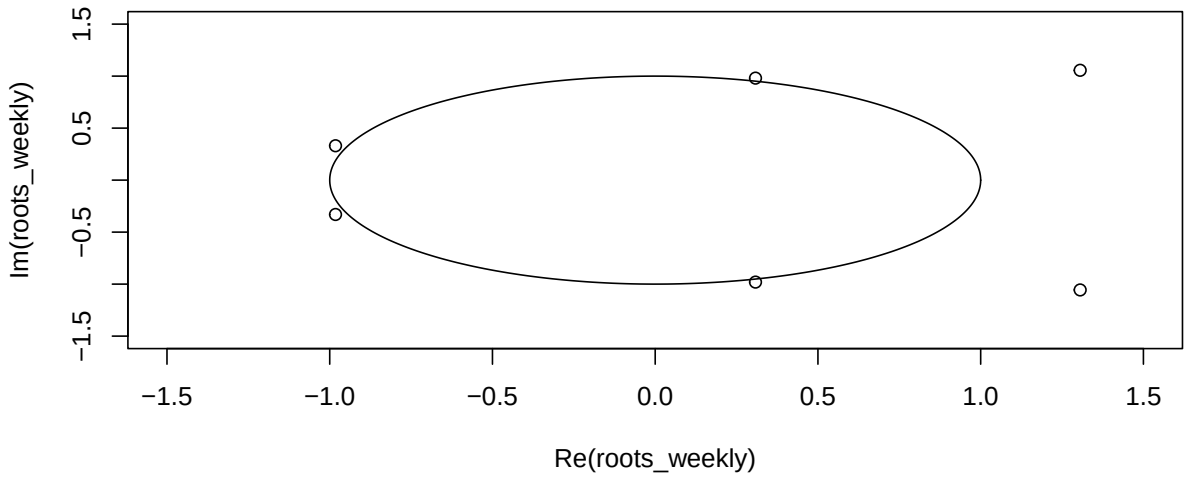
```
roots_weekly = polyroot(c(1, -modarma_weekly$model$phi))
print(roots_weekly)
```

```
## [1]  0.3081752+0.9805227i -0.9822365+0.3304380i -0.9822365-0.3304380i
## [4]  0.3081752-0.9805227i  1.3056459+1.0559982i  1.3056459-1.0559982i
```

```r
modulus_weekly = Mod(roots_weekly)
print(modulus_weekly)
```

```
## [1] 1.027812 1.036329 1.036329 1.027812 1.679239 1.679239
```

```r
plot(roots_weekly, xlim=c(-1.5, 1.5), ylim=c(-1.5, 1.5))
lines(complex(arg=seq(0, 2*pi, len=300)))
```




```r
modarma_daily = arima(residuals_daily, order=c(6,0,6), method="ML", include.mean = FALSE)
coefs_daily = coef(modarma_daily)
print(coefs_daily)
```

```
##         ar1         ar2         ar3         ar4         ar5         ar6
## -0.22760642  0.36377304 -0.11006576 -0.63762998 -0.13657295  0.79808111
##         ma1         ma2         ma3         ma4         ma5         ma6
##  0.49920070 -0.09346976  0.21719009  0.77802413  0.41208947 -0.55600433
```

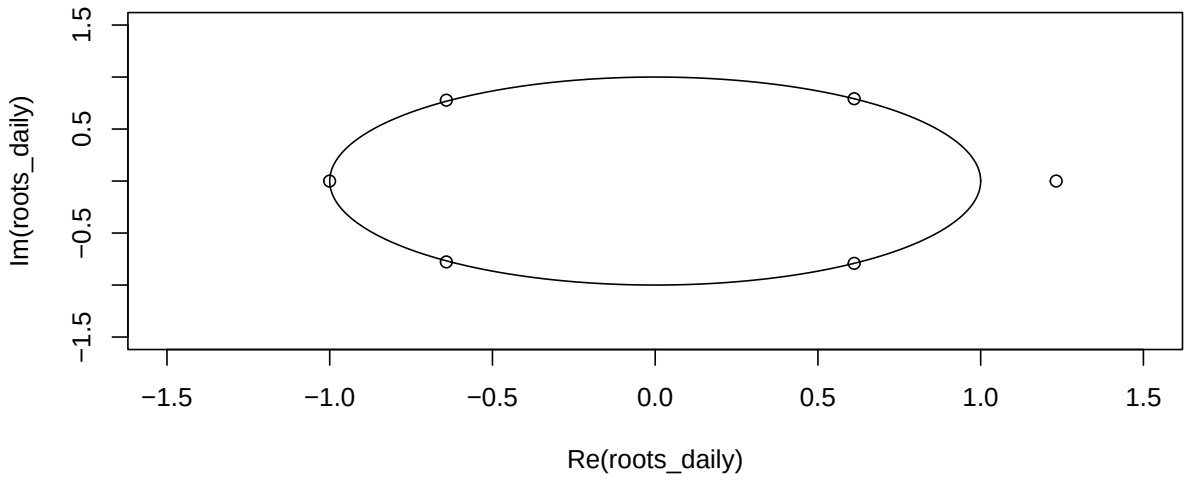
```r
roots_daily = polyroot(c(1, -modarma_daily$model$phi))
print(roots_daily)
```

```
## [1]  0.6113880+0.791383i -0.6416425+0.777748i -0.6416425-0.777748i
## [4]  0.6113880-0.791383i  1.2319992+0.000000i -1.0003636-0.000000i
```

```r
modulus_daily = Mod(roots_daily)
print(modulus_daily)
```

```
## [1] 1.000041 1.008264 1.008264 1.000041 1.231999 1.000364
```

```r
plot(roots_daily, xlim=c(-1.5, 1.5), ylim=c(-1.5, 1.5))
lines(complex(arg=seq(0, 2*pi, len=300)))
```



Response

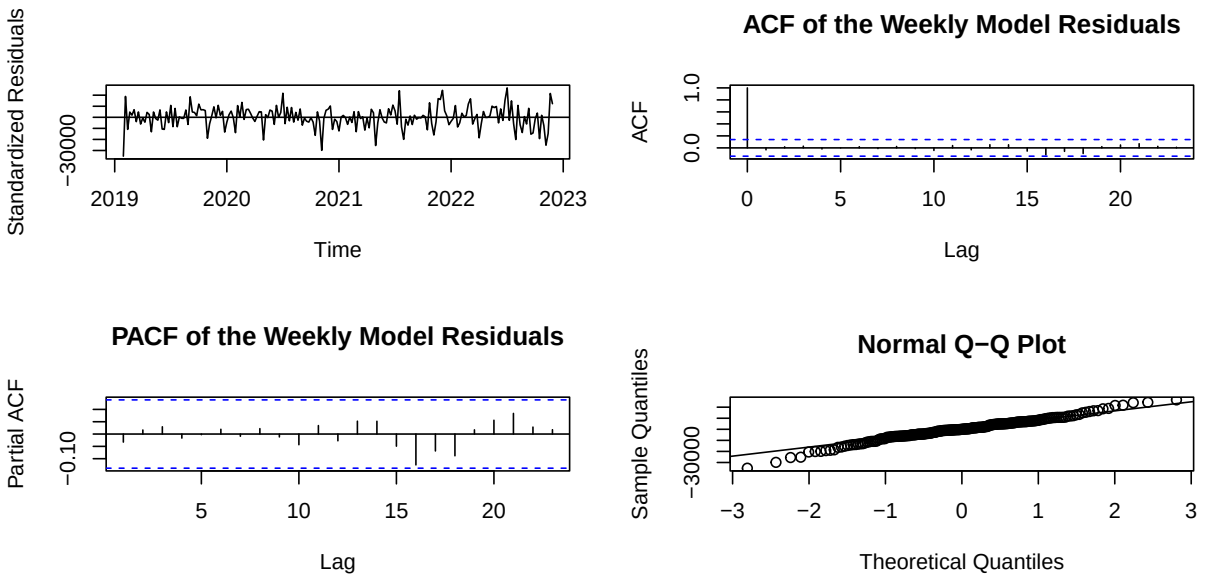
The coefficients (phi and theta) are given above for both datasets, and the roots of the AR part of the model are plotted above.

For the daily dataset, it shows that the process is not clearly stationary (as five of the points are very close to the circle), and that it is also not clearly causal, since all of the points are so close to the circle. For causality, they need to be outside the circle. The modulus of the roots are indeed extremely close to 1, as the console output shows.

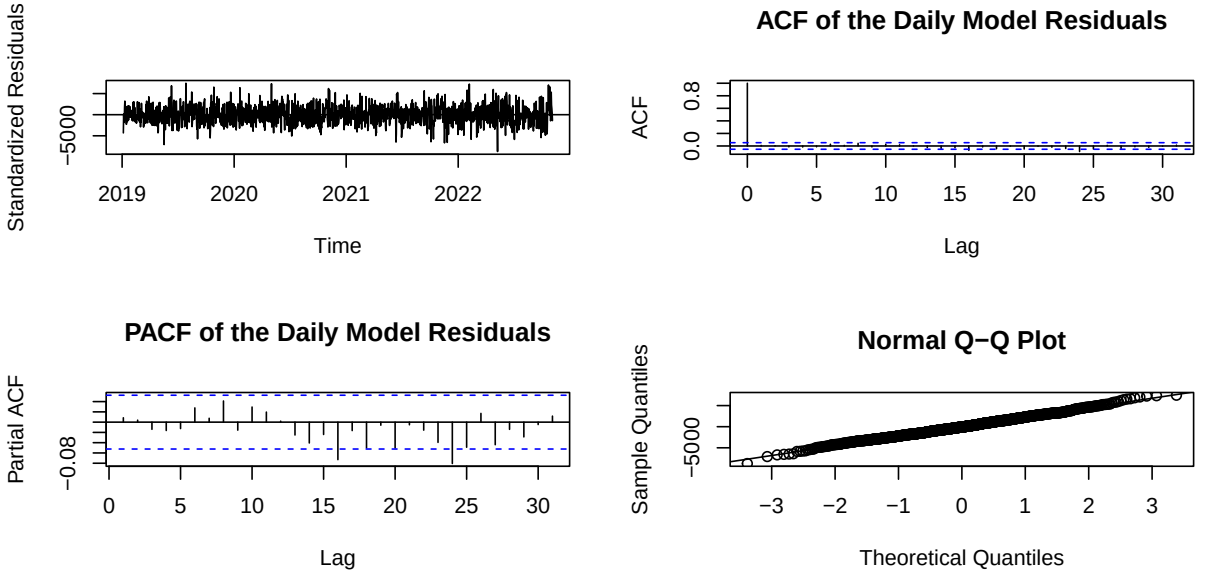
For the weekly dataset, it shows that the process is not clearly stationary (as four of the points are very close to the circle), and that it is also not clearly causal, since the points are just so close to the circle. For causality, they need to be outside the circle.

2b. (Residual analysis) Plot the residual processes of the two fitted models in Question 2a. Display the ACF, PACF, QQ-plot of these residual processes. Do the residual processes satisfies the assumptions of the R implementation?

```r
par(mfrow=c(2,2))
plot(resid(modarma_weekly), ylab='Standardized Residuals')
abline(h=0)
acf(as.vector(resid(modarma_weekly)), main='ACF of the Weekly Model Residuals')
pacf(as.vector(resid(modarma_weekly)), main='PACF of the Weekly Model Residuals')
qqnorm(resid(modarma_weekly))
qqline(resid(modarma_weekly))
```




```
par(mfrow=c(2,2))
plot(resid(modarma_daily), ylab='Standardized Residuals')
abline(h=0)
acf(as.vector(resid(modarma_daily)), main='ACF of the Daily Model Residuals')
pacf(as.vector(resid(modarma_daily)), main='PACF of the Daily Model Residuals')
qqnorm(resid(modarma_daily))
qqline(resid(modarma_daily))
```



Response

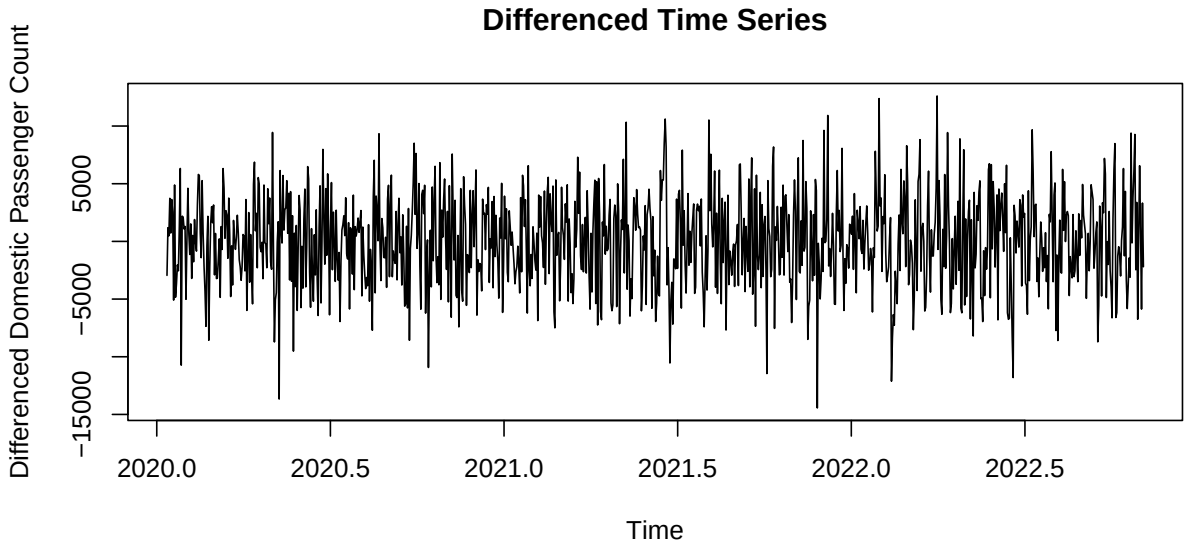
The residual ACF plot for the weekly dataset does not show a pattern, and all values are the within the confidence interval. The Q-Q norm plot shows a slight tail on the left, but isn't very significant for the normality assumption when using the MLE method. The variance of the residuals from the standardized residual plot is constant.

For the daily dataset, there are a few places where there is autocorrelation, which may not satisfy the requirement for stationarity. The Q-Q norm plot, however, does not show any tail, and aligns quite well with the expected diagonal line, which shows normality. This is necessary for the MLE method used when creating the models.

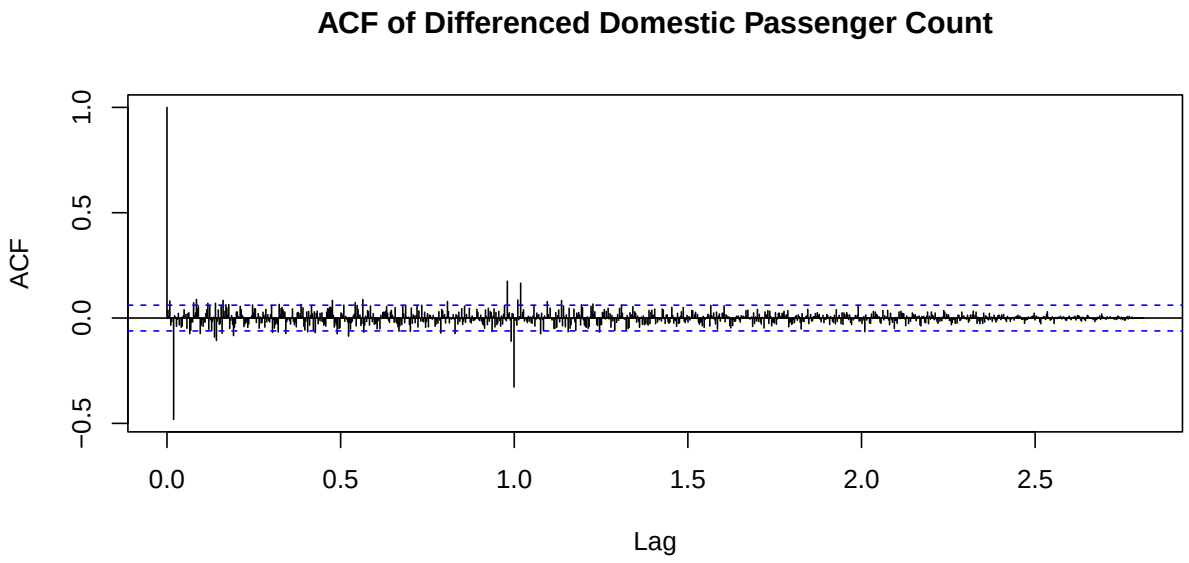
Question 3. ARMA fitting and model selection: Differenced daily domestic passenger count

3a. (Differencing for seasonality) Difference the daily domestic passenger count by 7 days, then again by 365 days. Plot the differenced time series, its ACF and PACF. Does this looks like a pure AR/MA process from the ACF/PACF plot?

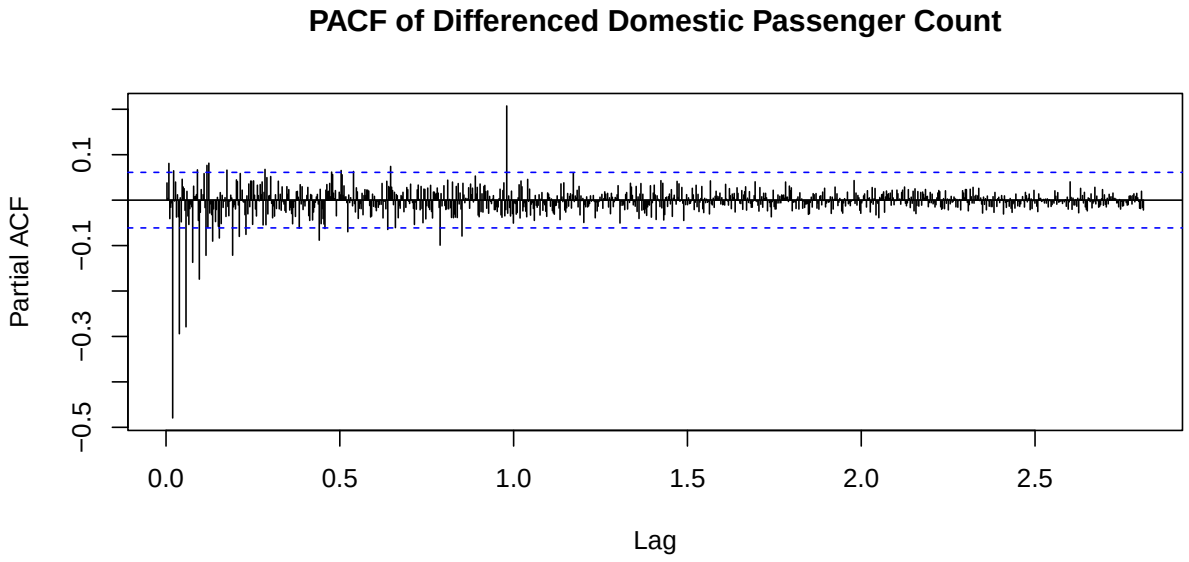
```
ddaily7_ts = diff(daily_ts,lag=7)
ddaily365_ts = diff(ddaily7_ts,lag=365)

#par(mfrow=c(2,1))
ts.plot(ddaily365_ts,ylab="Differenced Domestic Passenger Count", main = "Differenced Time Series")
```




```
acf(ddaily365_ts, main='ACF of Differenced Domestic Passenger Count',lag.max=365*3)
```




```
pacf(ddaily365_ts, main='PACF of Differenced Domestic Passenger Count',lag.max=365*3)
```



Response:

It does not look like a pure AR or MA process based on the ACF and PACF plots. For a pure AR process, the PACF should be zero for lags greater than p=6. This does not seem to be the case based on the PACF plot, as there are several lines after p=6 that are significant.

For a pure MA process, the ACF should be zero for lags greater than q=6. This does not seem to be the case based on the PACF plot, as there are several lines after q=6 that are significant, especially the spike at the lags shown at around 1 year.

3b. (ARMA fitting and order selection). Fit an ARMA model without intercept using the differenced daily data with AR and MA order up to 8. Select the best ARMA model using AICc. What is the order for the selected model and what is its AICc?

```
n = length(ddaily365_ts)
norder = 8
p=c(1:norder)-1;q = c(1:norder)-1
aic = matrix(0, norder, norder)
for (i in 1:norder){
    for(j in 1:norder){
        modij = arima(ddaily365_ts, order=c(p[i], 0, q[j]), method='ML')
        aic[i,j] = modij$aic-2*(p[i]+q[j]+1)*n/(n-p[i]-q[j]-2)}}

aicv = as.vector(aic)
#plot(aicv, ylab="AIC values")
indexp = rep(c(1:norder), norder)
indexq = rep(c(1:norder), each=norder)
indexaic = which(aicv == min(aicv))
porder = indexp[indexaic]-1
qorder = indexq[indexaic]-1

final_model = arima(ddaily365_ts, order = c(porder,0,qorder), method='ML')

cat('P Order:', porder, '\n')
```

```
## P Order: 7
```

```
cat('Q Order:', qorder, '\n')
```

```
## Q Order: 7
```

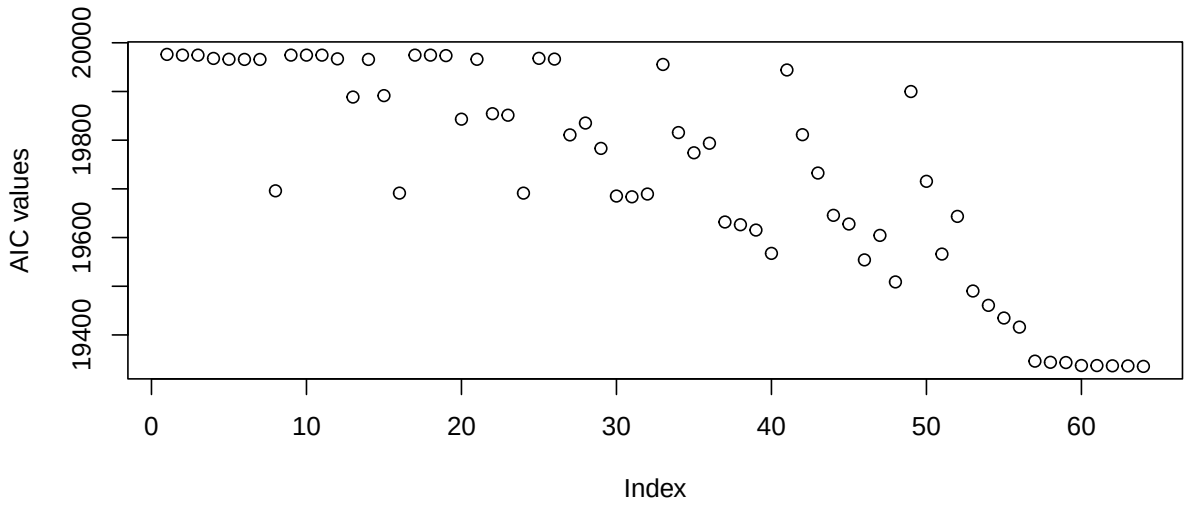
```
cat('AIC Value', aicv[indexaic], '\n')
```

```
## AIC Value 19335.37
```

```
cat('Final Model Coefficients', final_model$coef, '\n')
```

```
## Final Model Coefficients 0.05547448 -0.02018606 0.07831951 0.00546044 -0.02060109 0.01094547 0.03102
```

```
plot(aicv, ylab="AIC values")
```



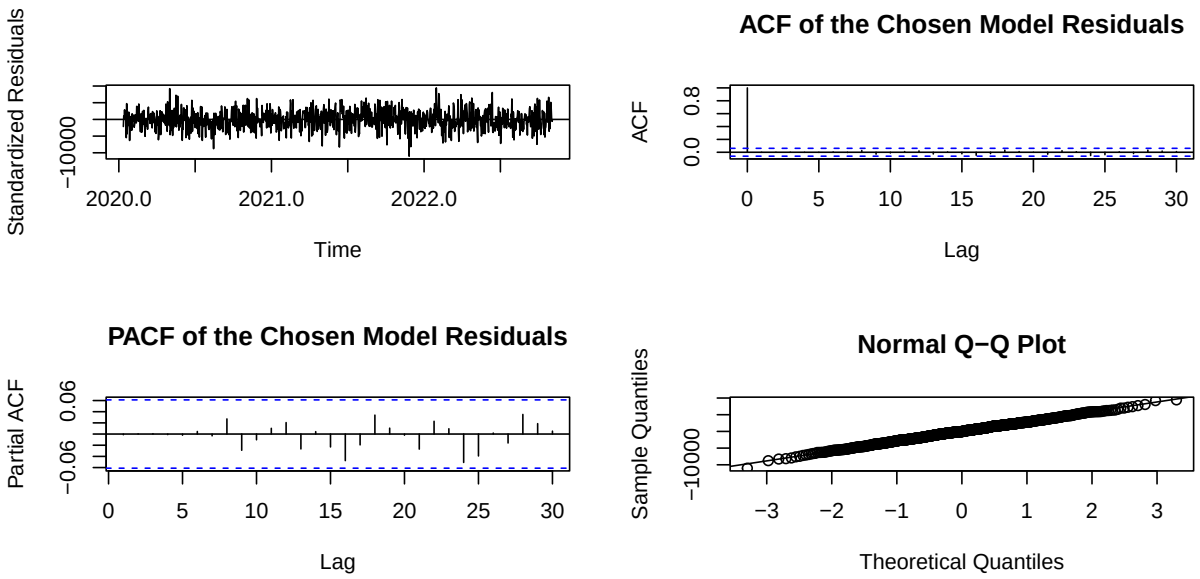

```
#arma_diff = arima(ddaily365_ts, order=c(8,0,8), method="ML", include.mean = FALSE)
#coefs_weekly = coef(modarma_weekly)
#print(coefs_weekly)
```

Response:

As printed, the model chosen has p=q=7, and the AICc value is 19333.37.

3c. (Residual analysis) Plot the residual process of the selected model in Question 3b. Display the ACF, PACF and QQ plot of the residual process. How does this model compare to the one on daily passenger count data in Question 2b?

```
par(mfrow=c(2,2))
plot(resid(final_model), ylab='Standardized Residuals')
abline(h=0)
acf(as.vector(resid(final_model)), main='ACF of the Chosen Model Residuals')
pacf(as.vector(resid(final_model)), main='PACF of the Chosen Model Residuals')
qqnorm(resid(final_model))
qqline(resid(final_model))
```



Response:

In contrast to the ACF and PACF plots of the daily dataset in question 2b, the ACF and PACF values are all within the 95% confidence band, so this satisfies stationarity assumption 3. Similar to question 2b, the Q-Q plot aligns well to the expected diagonal line, and the variance is constant as shown by the residuals plot.

3d. (Testing uncorrelated residuals) Use the Ljung-Box Test to decide whether the residuals of the selected ARMA model in Question 3b are correlated.

```
Box.test(final_model$resid, lag = (porder+qorder+1), type="Ljung-Box", fitdf=(porder+qorder))
```

```
##
##  Box-Ljung test
##
## data:  final_model$resid
## X-squared = 3.6799, df = 1, p-value = 0.05507
```

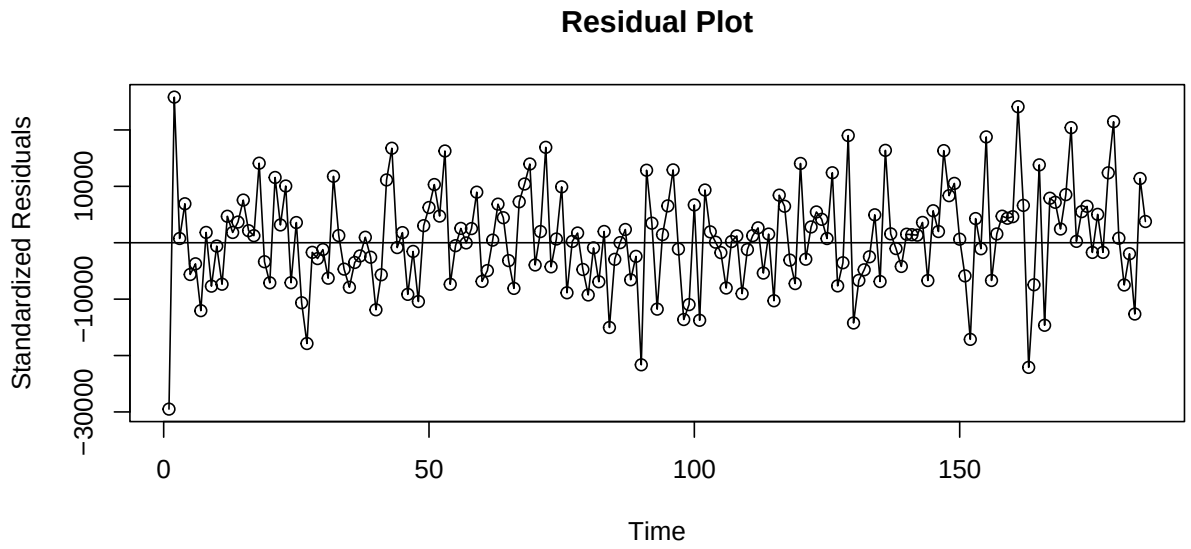
Response:

The P value is large (at the 0.05 significance level), indicating that the null hypothesis of uncorrelated residuals is plausible. Thus, according to these tests, the model performs well in modeling the temporal correlation in the time series process since the residuals are uncorrelated.

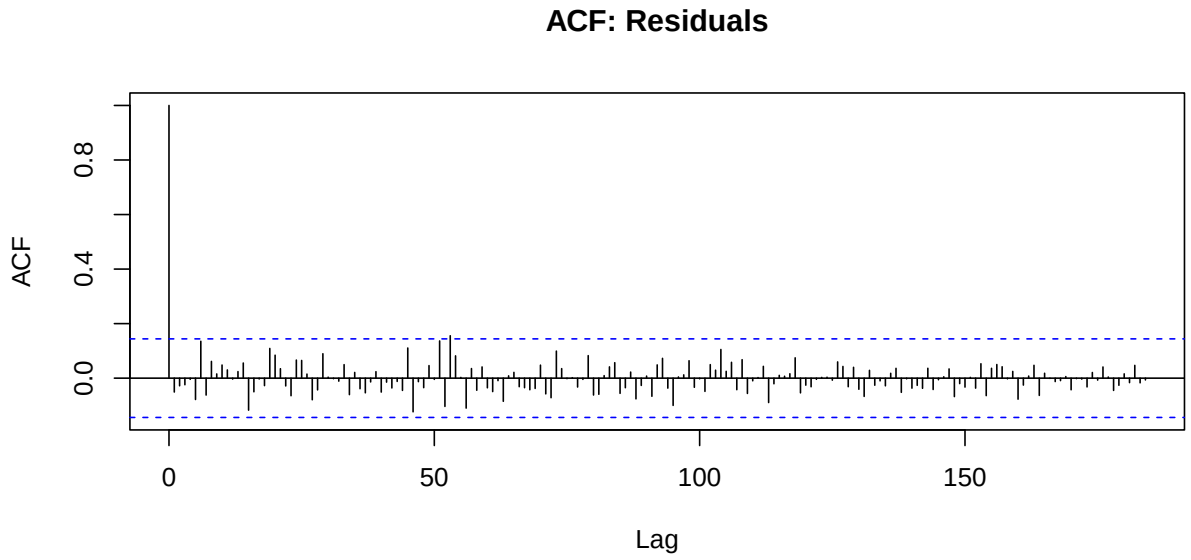
Question 4. Seasonal ARMA model and forecasting: Weekly domestic passenger count

4a. (Seasonal ARMA) Use the first 185 data points of weekly domestic passenger count as training data. Fit a seasonal ARMA model with intercept, where the non-seasonal model is ARMA(1,1) and the seasonal model is AR(1) with a period of 52 weeks. Plot the residual process and the ACF of the residual process. Comment on the appropriateness of the fitted model.

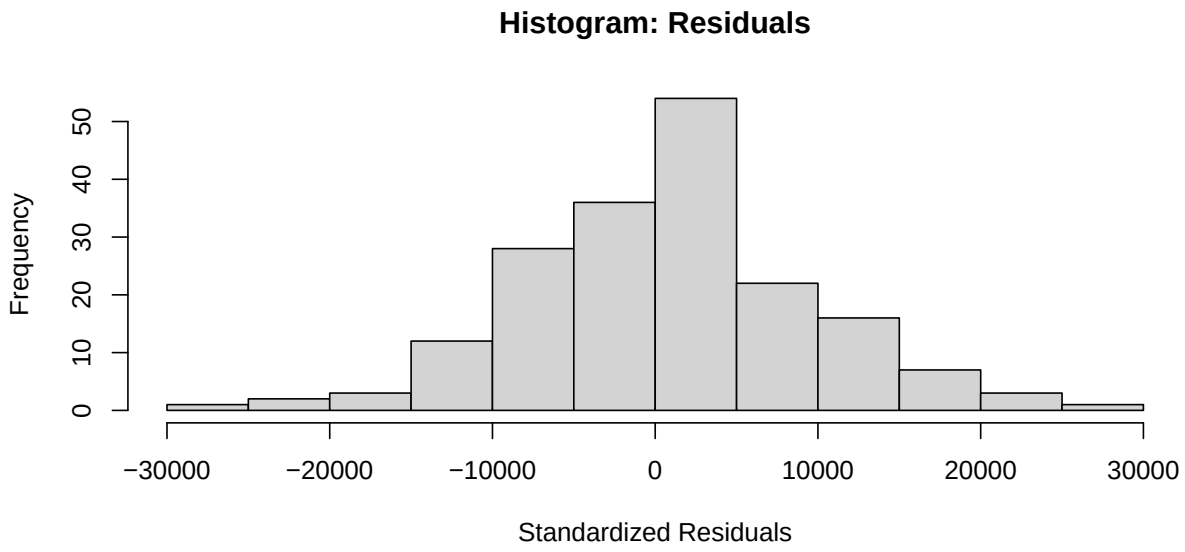
```r
mod_forecast = arima(weekly_ts[1:185], order= c(1, 0, 1), seasonal = list(order=c(1, 0, 0), period=52),

plot(resid(mod_forecast), ylab='Standardized Residuals', type='o', main="Residual Plot")
abline(h=0)
```



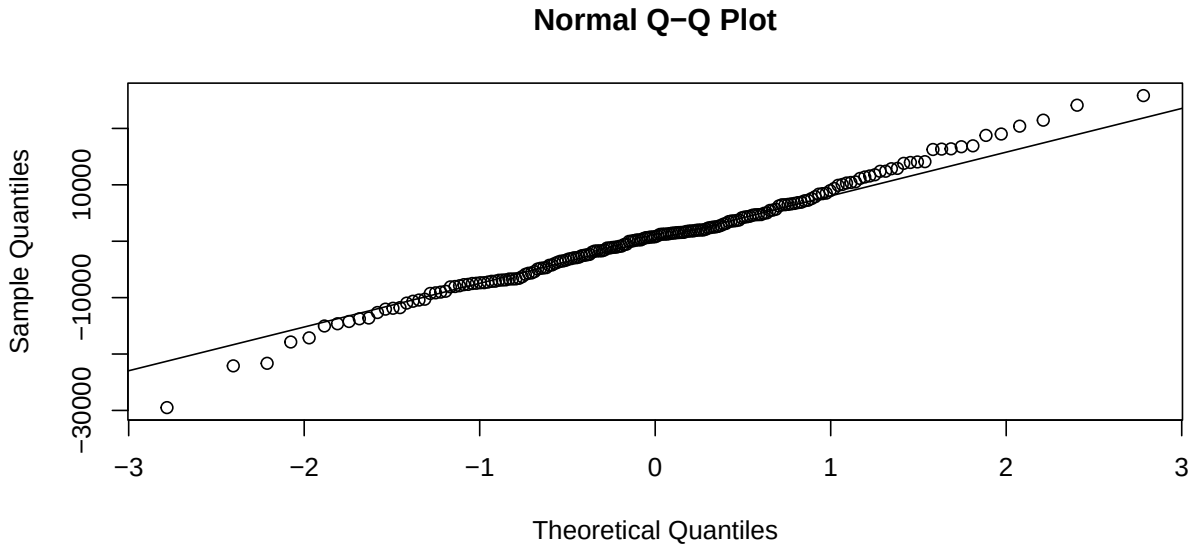

```r
acf(as.vector(resid(mod_forecast)), lag.max=365*2, main="ACF: Residuals")
```




```r
hist(resid(mod_forecast), xlab='Standardized Residuals', main='Histogram: Residuals')
```




```
qqnorm(resid(mod_forecast))
qqline(resid(mod_forecast))
```

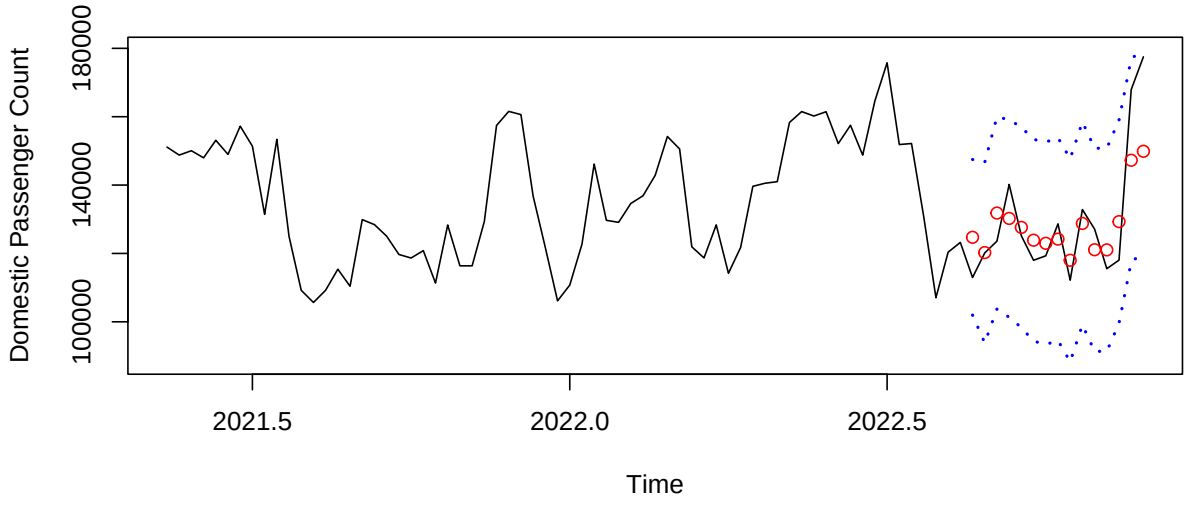


Response:

From the residual plot, it appears that the model has constant variance. From the ACF plot, there is only one outlier, but otherwise there appears to be no autocorrelation. This means that my statement regarding yearly seasonality may be correct, as modeling yearly seasonality removed autocorrelation from the residuals. The histogram of the standardized residuals, and the Q-Q plot shows a tiny amount of skewness, but nothing severe to indicate that the MLE method is not appropriate.

4b. (Forecasting) Use the fitted model in Question 4a to predict the total passenger count of the remainder of the weeks. Plot the 99% confidence interval. Compare with the actual observation. Does the actual observation fell into the 99% confidence interval?

```r
n = length(weekly_ts); nfit = n-15
out_pred = as.vector(predict(mod_forecast, n.ahead=15))


time_vals = time(weekly_ts)
ubound = out_pred$pred+2.58*out_pred$se
lbound = out_pred$pred-2.58*out_pred$se
ymin=min(lbound)
ymax=max(ubound)
plot(time_vals[120:n], weekly_ts[120:n], type="l", ylim=c(ymin,ymax), xlab="Time", ylab="Domestic Passe
points(time_vals[(nfit+1):n], out_pred$pred, col="red")
lines(time_vals[(nfit+1):n], ubound, lty=3, lwd=2, col="blue")
lines(time_vals[(nfit+1):n], lbound, lty=3, lwd=2, col="blue")
```



Response

As shown in the plot above, the actual observations fall within the 99% confidence interval. In addition, from visual inspection, they follow closely to the actual observations, and they even follow some of the volatility, especially for the last two point predictions.