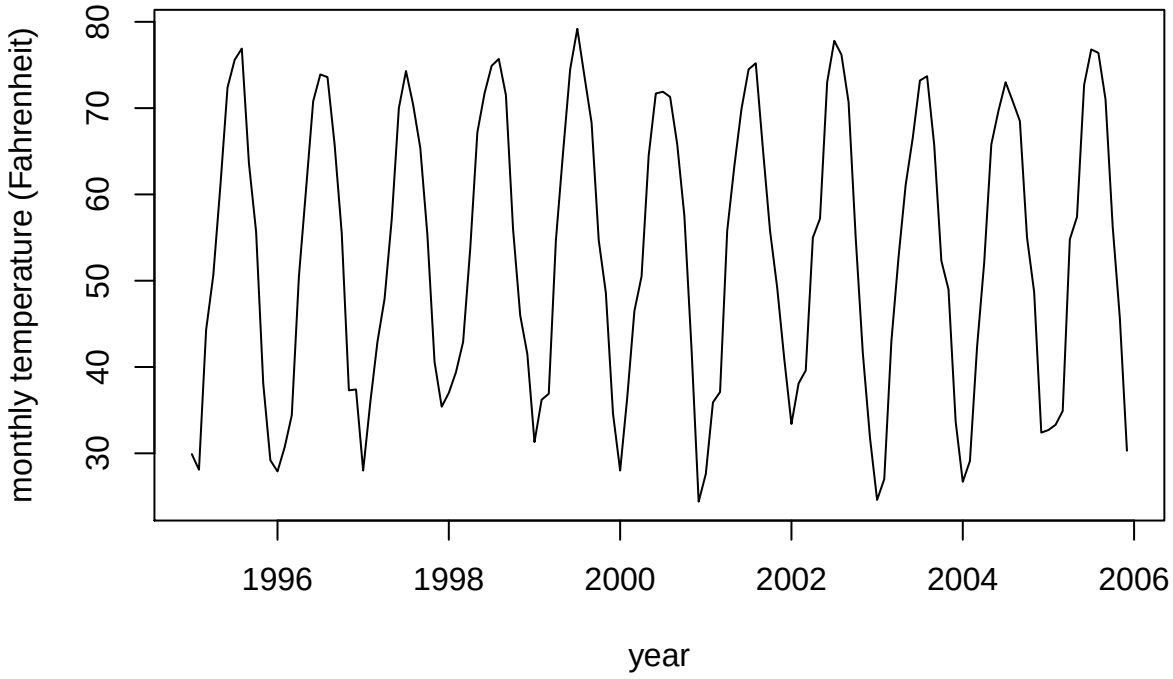
STAT 6550 HW3: R Exercises

Blain Morin

2/23/2021

We are going to perform the classical decomposition algorithm (CDA) upon a time series of monthly temperatures recorded in Columbus, OH from January 1995 to December 2005. The units of measurement is Fahrenheit.



Part (a) Summarize the plot of the time series – what features of the data are present?

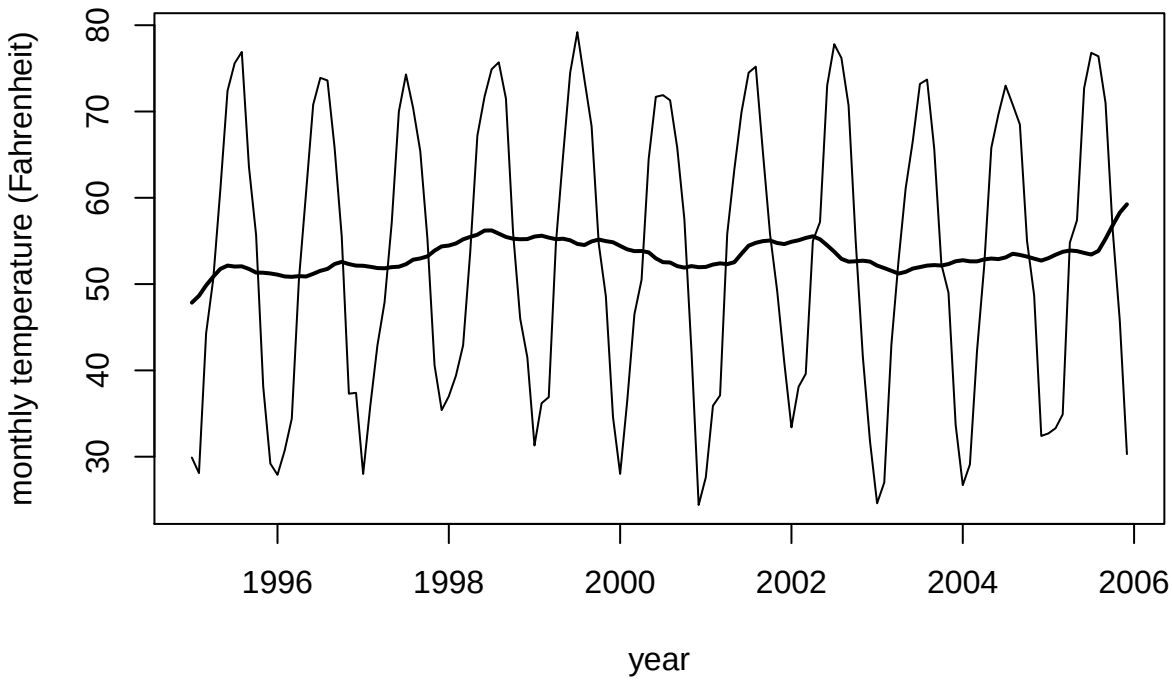
- The series visually appears to be stationary
- Trend: There does not seem to be an obvious trend
- Seasonality: There appears to be seasonality because there seems to be a periodic yearly pattern.
- Correlation: There is correlation because nearby points seem more similar to each other than point further away.

```r
## We will now carry out the classical decomposition algorithm.

## ======================================================================
## ** Step 1 of the algorithm **
## Estimate the trend by smoothing the time series.
## ======================================================================

## let us choose the period to be d=12 months
d <- 12

## half.d is d/2
half.d <- d/2

## smooth the temp time series using the 'filter' command
our.filter <- c(0.5, rep(1,d-1), 0.5)/d

## display 'our.filter' to see the values makes sense!
our.filter

## 'n' is the length of time series
n <- length(temp)

## We will not filter the variable 'temp',
## but a new variable called 'longer.temp'
longer.temp <- c(temp[(d/2):1], temp, temp[n:(n-d/2+1)])
longer.temp

## now filter 'longer.temp'.
temp.MA <- filter(longer.temp, our.filter)[(d/2+1):(d/2+n)]
temp.MA
```

Part (b) Try to explain why we have the variable 'longer.temp'. What does this variable contain, and what does the variable allow us to do in the 'filter' command?

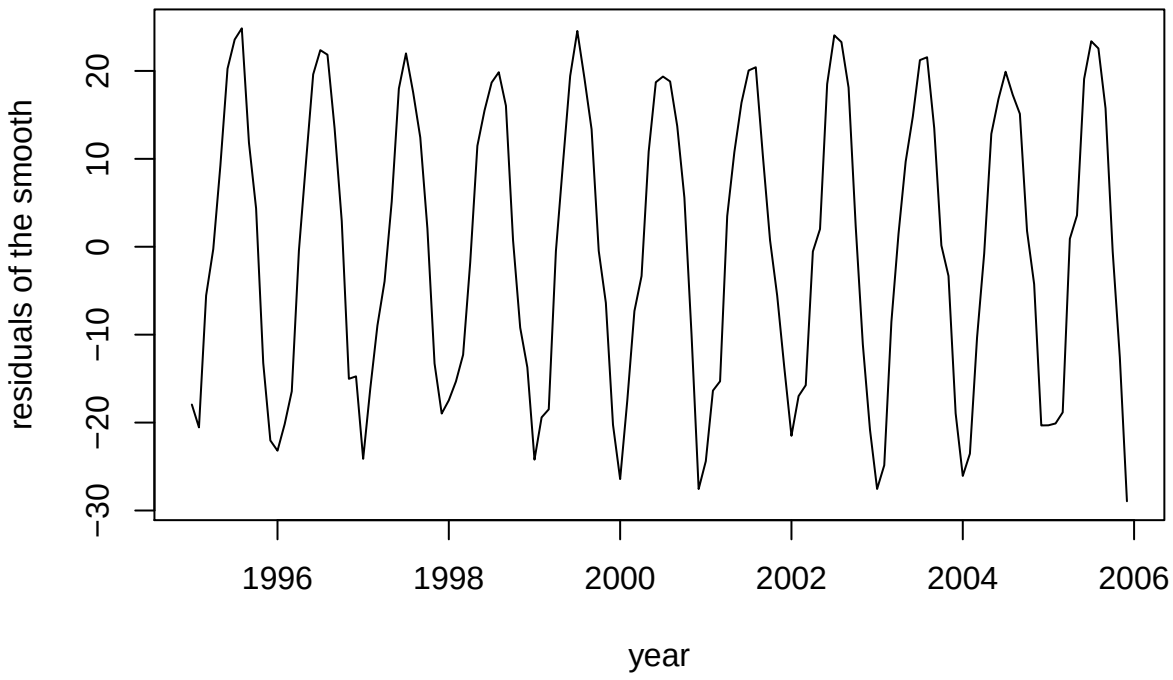
The 'longer.temp' variable artificially adds six months to the beginning and end of the original time series using values from the first and last month, respectively. Doing this allows us to produce estimates of the average for the entire range of the original series.

Part (c) Summarise the plot of the smooth – what do you see? Do you think the variance of the trend estimate is constant with time? Explain.

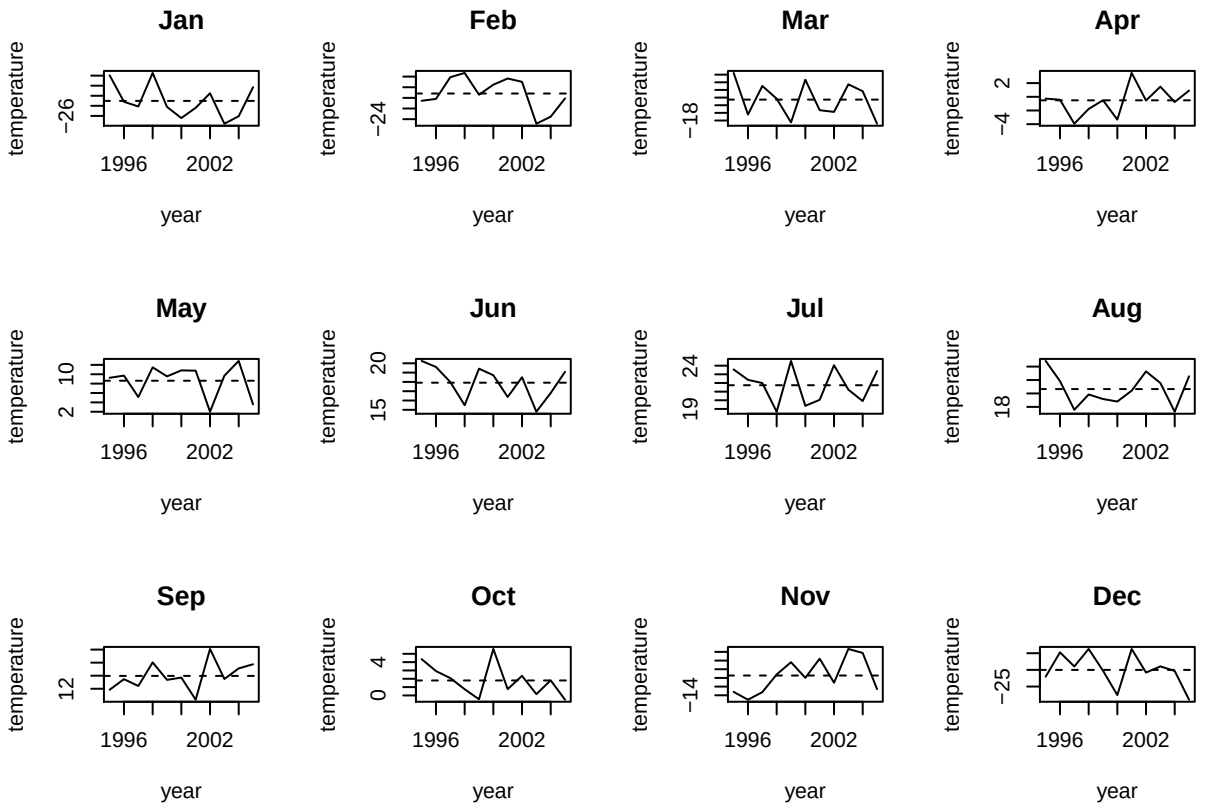


Our MA filter stays fairly stable around 51 degress. It seems that the variance is constant over time because (as we see from the stability of the MA line) the series appears to be stationary.

```
## ======================================================================
## ** Step 2 of the algorithm **
## Estimate the seasonal component.
## ======================================================================

## calculate and plot the residuals of the smooth.
temp.MA.resids <- temp - temp.MA

## plot the time series (first plot)
plot(year, temp.MA.resids, type="l",
     xlab="year", ylab="residuals of the smooth")
```



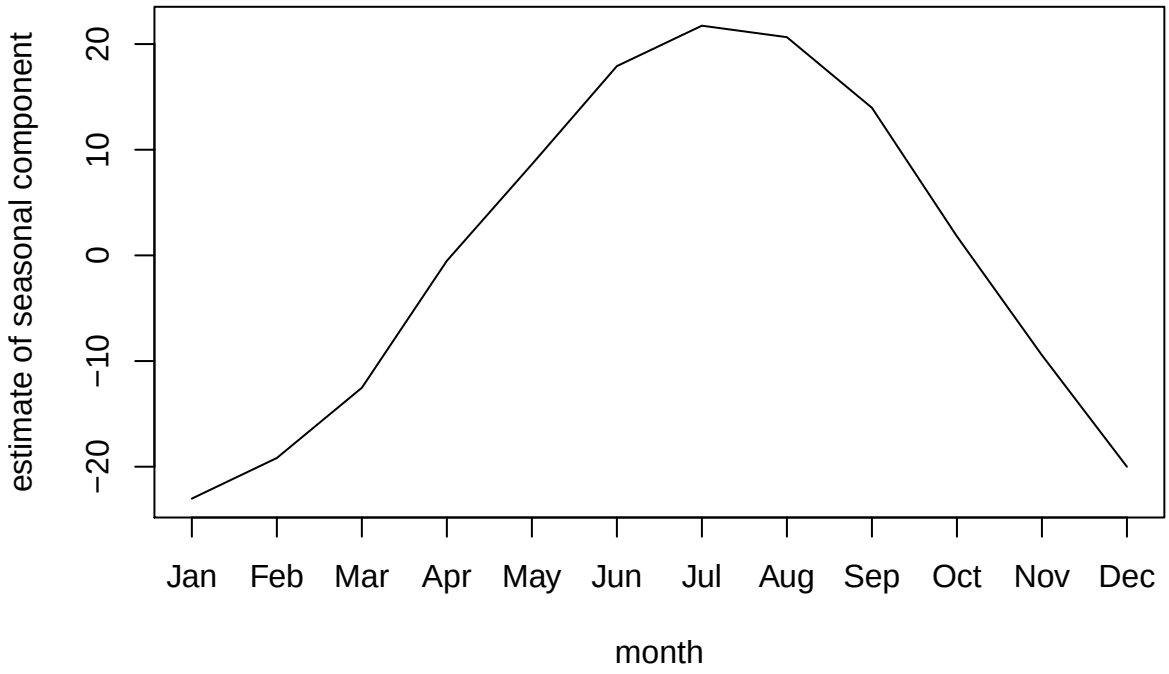

```
## we define a variable to hold the names of the months
months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
            "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")

## Now we need to estimate the average seasonal components.
## There should be 'd=12' of them.
## We first form 'temp.MA.resids' into a matrix with 'd' rows.
temp.matrix <- matrix(temp.MA.resids, nrow=d)
temp.matrix

## create the plot of temperatures by month (second plot)
## in a panel of size 3 rows by 4 columns.
par(mfrow=c(3,4))
for (m in 1:12) {
  plot(1995:2005, temp.matrix[m,], type="l",
       xlab="year", ylab="temperature", main=months[m])
  abline(h=mean(temp.matrix[m,]), lty=2)
}
```



Part (d) Describe the two plots above. Do we have evidence of seasonality in the time series? Explain.

Yes, there appears to be seasonality. We see that there is a clear pattern in the residual plot (which seems stationary). We confirm this observation when looking at the monthly average plot. We see that there are visually significant differences across months. Moreover, these differences across months are greater than within month.

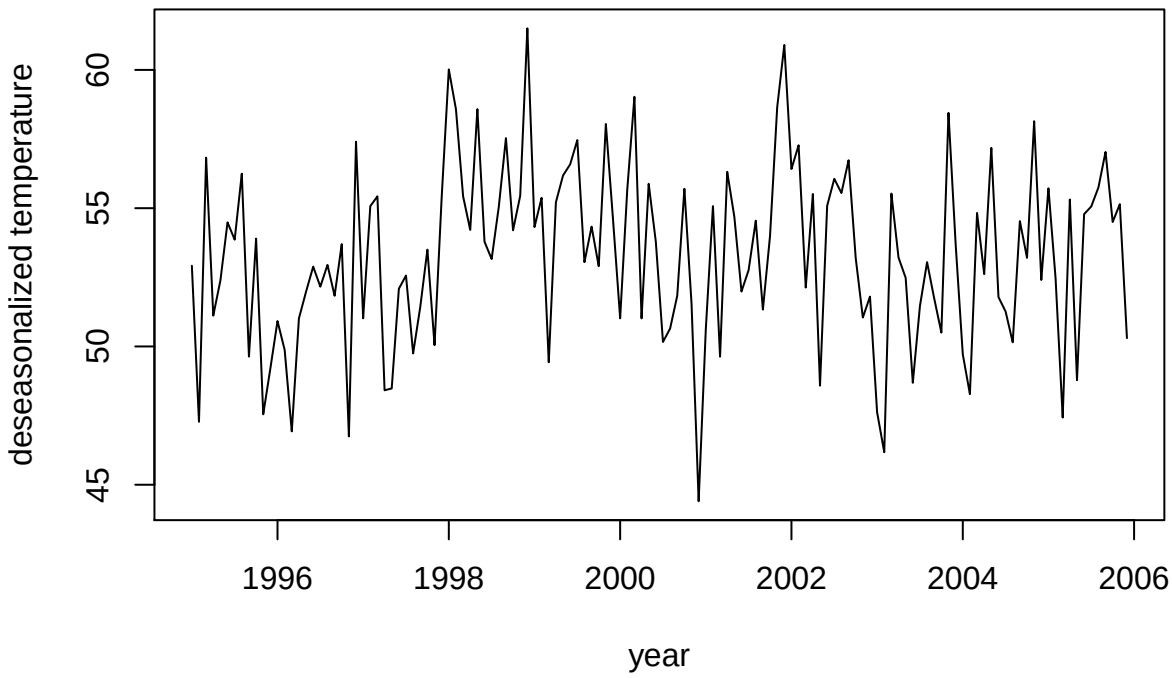
Part (e) Describe the estimate of the seasonal component in words.



From the seasonal component plot, we expect lower than average temperatures from about October to April, with January expected to have the lowest average temp (about 23 degrees below average). We expect higher temperatures from about May to September , with July expected to have the highest average temp (about 22 degrees above average).

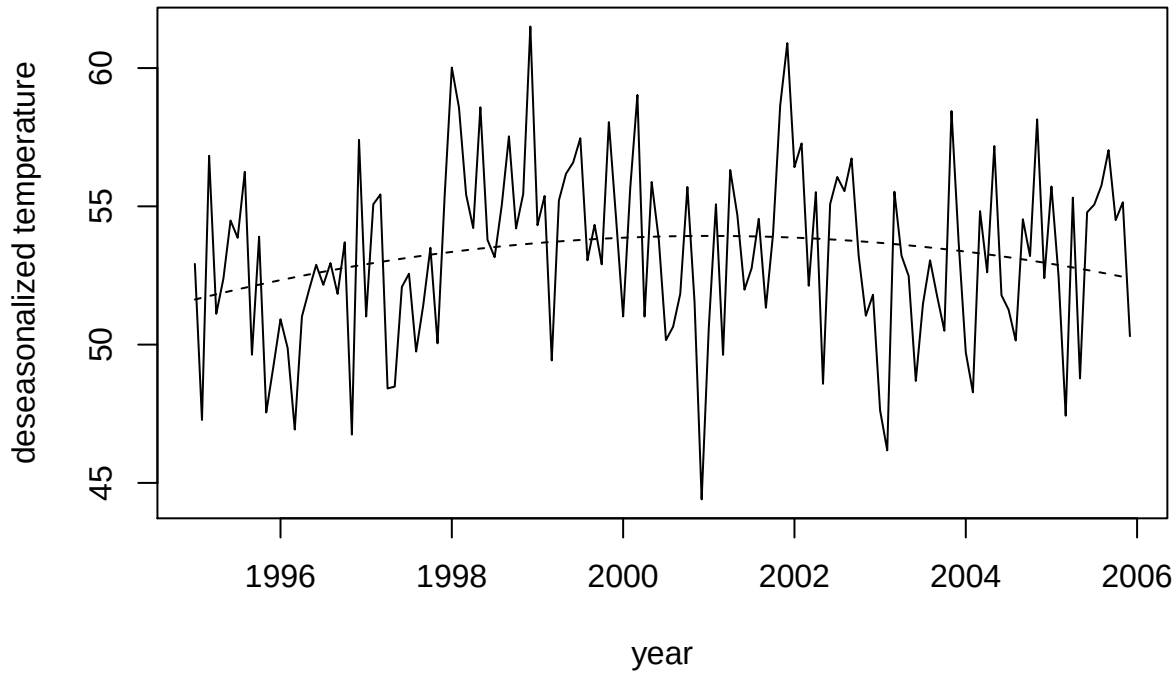
Part (f) Describe this plot:

```
## ========================================================================
## ** Step 3 of the algorithm **
## Deseasonalize the time series and estimate the trend parametrically.
## ========================================================================

## remove the seasonality
deseasonalized <- temp - s.hat

## plot the deseasonalized time series.
plot(year, deseasonalized, type="l",
     xlab="year", ylab="deseasonalized temperature")
```



We see that the de-sesonalized data has a much smaller ranger than the original set. However, we see visually that there is a significant amount of noise.

Part (g) First write down the statistical model that has been fit to the deseasonalized data. Next, write down the estimated trend model. Explain why can we not assess the statistical significance of the regression parameters from this model output.



The statistical model is:

$$Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 - \hat{s}_t + \eta_t$$

The estimated model is:

$$\widehat{deseason} Y_t = -254,500 + 254.4t - 0.06358t^2$$

We cannot assess statistical significance because these ordinary least squares estimates assume that η_t is identically and independently distributed. However, as we observed above there seems to be dependence in the data.

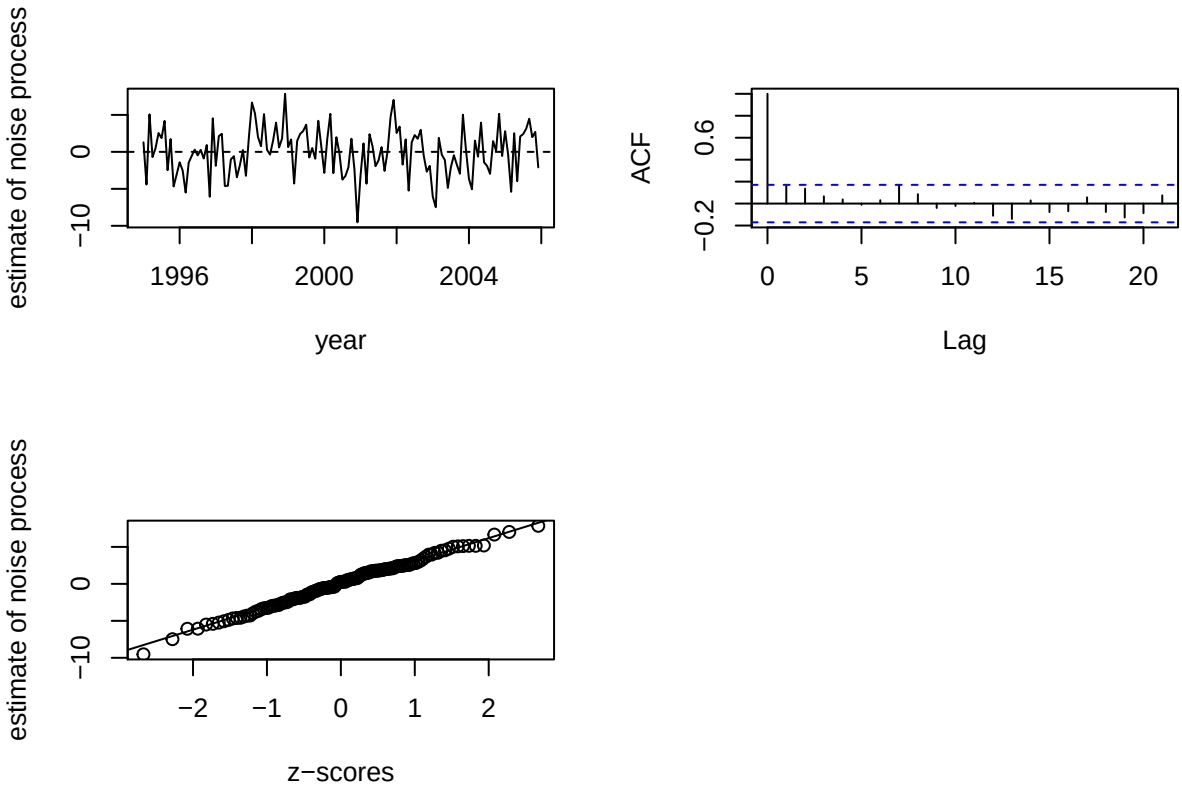
Part (h) Using the figures, summarize the residuals. In your answer make sure you respond to the following:

- **Do the residuals seem to be drawn from a stationary process?**
- **Do you believe that the residuals are a sample of IID noise?**
- **Do the residuals seem to be drawn from a normal distribution?**

```r
## ========================================================================
## ** Step 4 of the algorithm **
## Calculate the residuals and look at the ACF.
## ========================================================================

## calculate the residuals
resids <- resid(lm.deseasonalized)

## Plot the residual series with a horizontal line at y=0.
par(mfrow=c(2, 2))
plot(year, resids, type="l", xlab="year", ylab="estimate of noise process")
abline(h=0, lty=2)

## plot the acf
acf(resids, main="")

## carry out the Ljung-Box test
Box.test(resids, 22, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  resids
## X-squared = 25.455, df = 22, p-value = 0.2758
```

```r
## draw a Q-Q plot of the residuals.
qqnorm(resids, xlab="z-scores", ylab="estimate of noise process", main="")
qqline(resids)
```



- Looking at the noise process plot, the residuals seem to be drawn from a stationary process (neither the mean or variance depend on time).
- From the ACF plot and the Box-Ljung test, there is not enough evidence to suggest that the noise is not IID. The p-value of the Box-Ljung test is .275, which is not significant and thus does not reject the null hypothesis that the errors are IID. Moreover, in the ACF plot, none of the bars visually appears to be outside of the 95% blue interval. This again supports our other observations that there is not enough evidence to reject that the noise is IID.
- Looking at the qqplot, the residuals line up on the reference line pretty well. Visually, I would conclude that the errors appear to be drawn from a normal distribution.

Appendix: All R code

```
knitr::opts_chunk$set(echo = FALSE, warning = FALSE, error = FALSE)

temp <-
  c(29.9, 28.1, 44.3, 50.6, 61.0, 72.4, 75.6, 76.9, 63.6, 55.7, 38.1, 29.2,
    27.9, 30.7, 34.4, 50.5, 60.6, 70.8, 73.9, 73.6, 65.8, 55.5, 37.3, 37.4,
    28.0, 35.9, 42.9, 47.9, 57.1, 70.0, 74.3, 70.4, 65.4, 55.3, 40.6, 35.4,
    37.0, 39.4, 42.9, 53.7, 67.2, 71.7, 74.9, 75.7, 71.5, 56.0, 46.0, 41.5,
    31.3, 36.2, 36.9, 54.7, 64.8, 74.5, 79.2, 73.7, 68.3, 54.7, 48.6, 34.6,
    28.0, 36.5, 46.5, 50.5, 64.5, 71.7, 71.9, 71.3, 65.8, 57.5, 42.1, 24.4,
    27.6, 35.9, 37.1, 55.8, 63.3, 69.9, 74.5, 75.2, 65.3, 55.8, 49.2, 40.9,
    33.4, 38.1, 39.6, 55.0, 57.2, 73.0, 77.8, 76.2, 70.7, 55.0, 41.6, 31.8,
    24.6, 27.0, 43.0, 52.7, 61.1, 66.6, 73.2, 73.7, 65.7, 52.3, 49.0, 33.7,
```

```
    26.7, 29.1, 42.3, 52.1, 65.8, 69.7, 73.0, 70.8, 68.5, 55.0, 48.7, 32.4,
    32.7, 33.3, 34.9, 54.8, 57.4, 72.7, 76.8, 76.4, 71.0, 56.3, 45.7, 30.3)

## set up the time variable, 'year'
year <- seq(from=1995, by=1/12, length=length(temp))

## plot the time series.
plot(year, temp, type="l",
     xlab="year", ylab="monthly temperature (Fahrenheit)")


## We will now carry out the classical decomposition algorithm.

## ======================================================================
## ** Step 1 of the algorithm **
## Estimate the trend by smoothing the time series.
## ======================================================================

## let us choose the period to be d=12 months
d <- 12

## half.d is d/2
half.d <- d/2

## smooth the temp time series using the 'filter' command
our.filter <- c(0.5, rep(1,d-1), 0.5)/d

## display 'our.filter' to see the values makes sense!
our.filter

## 'n' is the length of time series
n <- length(temp)

## We will not filter the variable 'temp',
## but a new variable called 'longer.temp'
longer.temp <- c(temp[(d/2):1], temp, temp[n:(n-d/2+1)])
longer.temp

## now filter 'longer.temp'.
temp.MA <- filter(longer.temp, our.filter)[(d/2+1):(d/2+n)]
temp.MA


## plot the time series.
plot(year, temp, type="l",
     xlab="year", ylab="monthly temperature (Fahrenheit)")

## overlay the smooth
lines(year, temp.MA, type="l", lwd=2)


## ======================================================================
## ** Step 2 of the algorithm **
```

```r
## Estimate the seasonal component.
## ================================================================

## calculate and plot the residuals of the smooth.
temp.MA.resids <- temp - temp.MA

## plot the time series (first plot)
plot(year, temp.MA.resids, type="l",
     xlab="year", ylab="residuals of the smooth")

## we define a variable to hold the names of the months
months <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun",
            "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")

## Now we need to estimate the average seasonal components.
## There should be 'd=12' of them.
## We first form 'temp.MA.resids' into a matrix with 'd' rows.
temp.matrix <- matrix(temp.MA.resids, nrow=d)
temp.matrix

## create the plot of temperatures by month (second plot)
## in a panel of size 3 rows by 4 columns.
par(mfrow=c(3,4))
for (m in 1:12) {
  plot(1995:2005, temp.matrix[m,], type="l",
       xlab="year", ylab="temperature", main=months[m])
  abline(h=mean(temp.matrix[m,]), lty=2)
}


## now average across the years to get the 's.hat.stars'
s.hat.stars <- rowMeans(temp.matrix)
s.hat.stars

## plot the 's.hat.stars', without the x-axis
#par(mfrow=c(1,1))
#plot(s.hat.stars, type="l", xaxt="n")

## now we'll label the x-axis
#axis(side=1, at=1:12, labels=months)


## now subtract off the mean of the 's.hat.stars' so that we have a
## seasonal component that adds up to zero (has average zero).
s.hat <- s.hat.stars - mean(s.hat.stars)

## plot the 's.hat', without the x-axis
plot(s.hat, type="l", xaxt="n",
     xlab="month", ylab="estimate of seasonal component")

## now we'll label the x-axis
axis(side=1, at=1:12, labels=months)
```

```r
## ========================================================================
## ** Step 3 of the algorithm **
## Deseasonalize the time series and estimate the trend parametrically.
## ========================================================================

## remove the seasonality
deseasonalized <- temp - s.hat

## plot the deseasonalized time series.
plot(year, deseasonalized, type="l",
     xlab="year", ylab="deseasonalized temperature")


## We now fit a quadratic line model to the data.
lm.deseasonalized <- lm(deseasonalized ~ year + I(year^2))

## Summarize the model obtained.
summary(lm.deseasonalized)

## Plot the series and the estimated regression line.
plot(year, deseasonalized, type="l",
     xlab="year", ylab="deseasonalized temperature")
lines(year, fitted(lm.deseasonalized), lty=2)


## ========================================================================
## ** Step 4 of the algorithm **
## Calculate the residuals and look at the ACF.
## ========================================================================

## calculate the residuals
resids <- resid(lm.deseasonalized)

## Plot the residual series with a horizontal line at y=0.
par(mfrow=c(2, 2))
plot(year, resids, type="l", xlab="year", ylab="estimate of noise process")
abline(h=0, lty=2)

## plot the acf
acf(resids, main="")

## carry out the Ljung-Box test
Box.test(resids, 22, type="Ljung-Box")

## draw a Q-Q plot of the residuals.
qqnorm(resids, xlab="z-scores", ylab="estimate of noise process", main="")
qqline(resids)
```