Blatt6

Tobias

13 8 2021

```
library(datasets)
library(SynchWave)
```

```
## Loading required package: fields

## Loading required package: spam

## Loading required package: dotCall64

## Loading required package: grid

## Spam version 2.6-0 (2020-12-14) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## See https://github.com/NCAR/Fields for
##  an extensive vignette, other supplements and source code
```

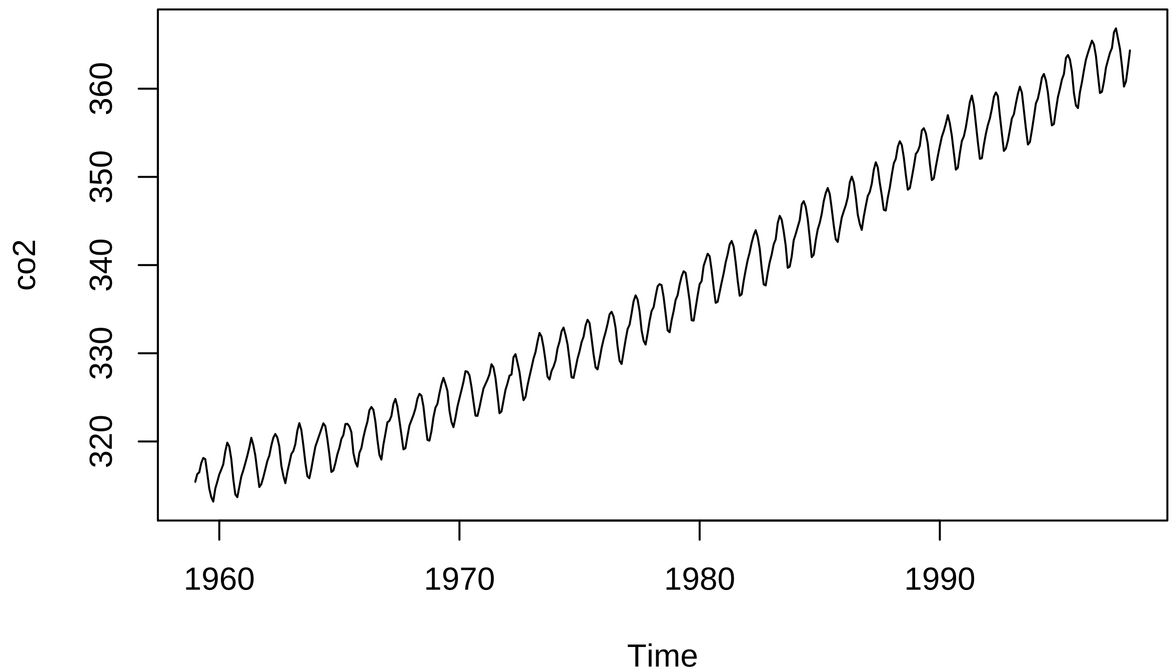
```
data('co2')
#attach(co2)
```

Aufgabe 6.1

a)

Ternd: Grundrichtung, langfristige Entwiklung(Glatte Komponente/Linearer/Polynominaler(bsw. Parabel) Trend)
Saisonkomponente: kurzfristige Entwiklung innerhalb der einzelnen Jahre durch saisonbedingte Schwankungen
Residuen: Summe aller Fehler -> nicht erklärte Schwankungen in der Zeitreihe

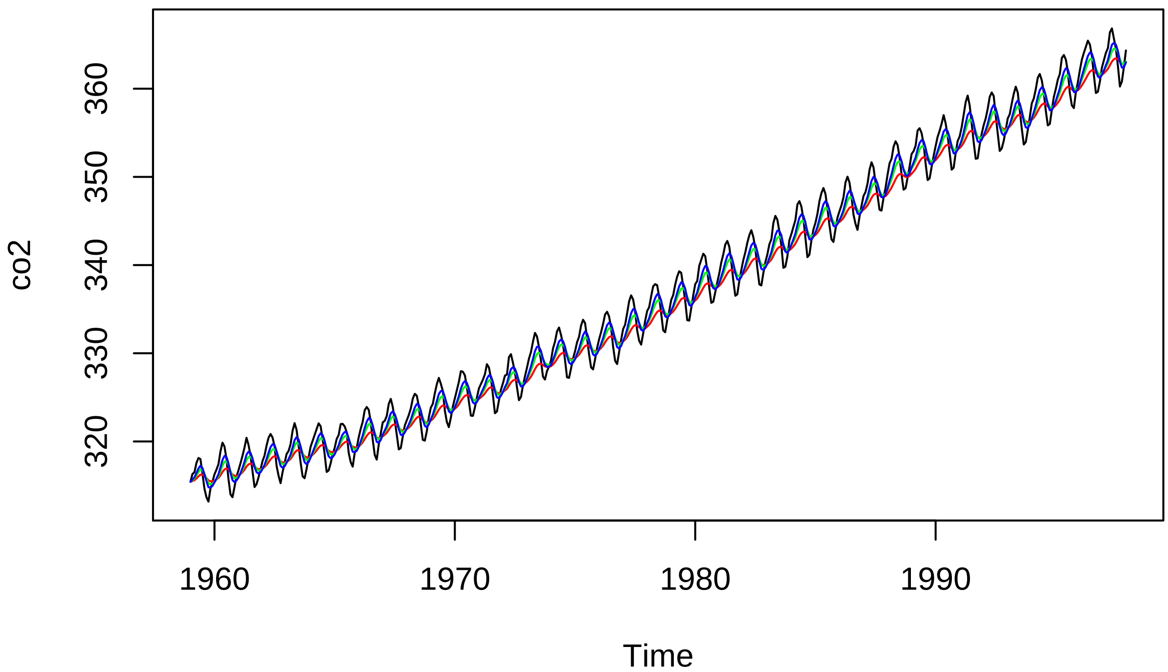
```
ts.plot(co2) #so soll es wohl sein (ts unnötig hier)
```



b)

```
eglat = function(zeitreihe,a){
  g = ts(zeitreihe)[1]
  for(i in 2:length(zeitreihe)){
    g = ts(zeitreihe)[i]*a+(1-a)*g
    zeitreihe[i]=g
  }
  return(zeitreihe)
}
```

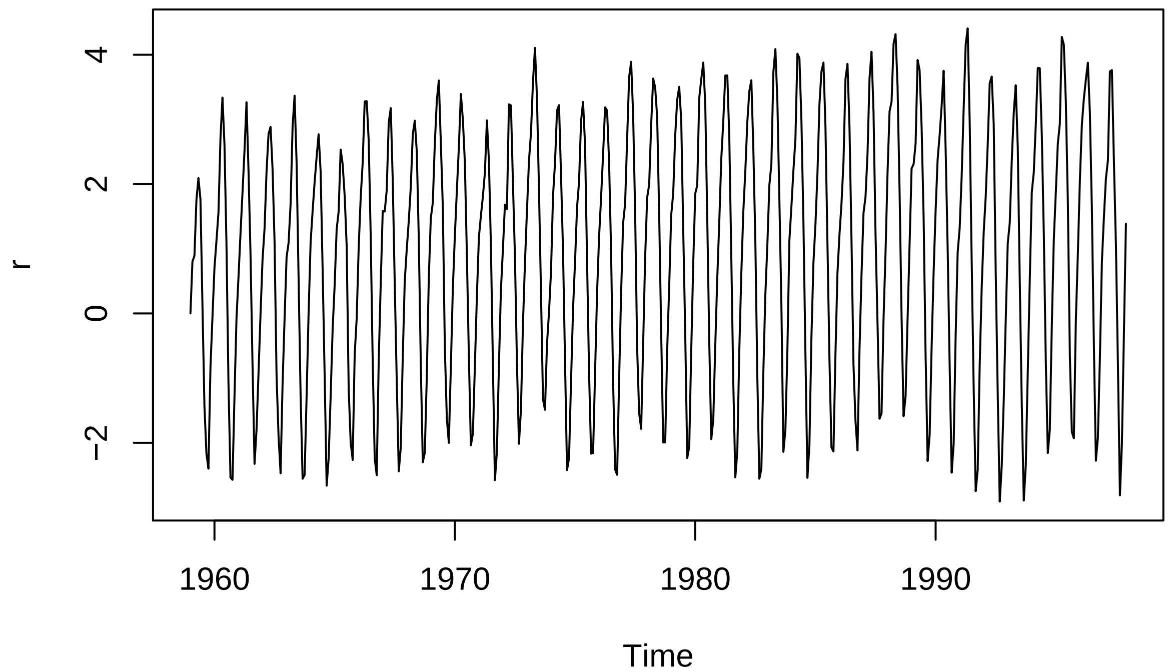
```
glat1 = eglat(co2,0.1)
glat2 = eglat(co2,0.2)
glat3 = eglat(co2,0.3)
plot(co2)
lines(glat1,type = 'l',col='red')
lines(glat2,type = 'l',col='green')
lines(glat3,type = 'l',col='blue')
```



c)

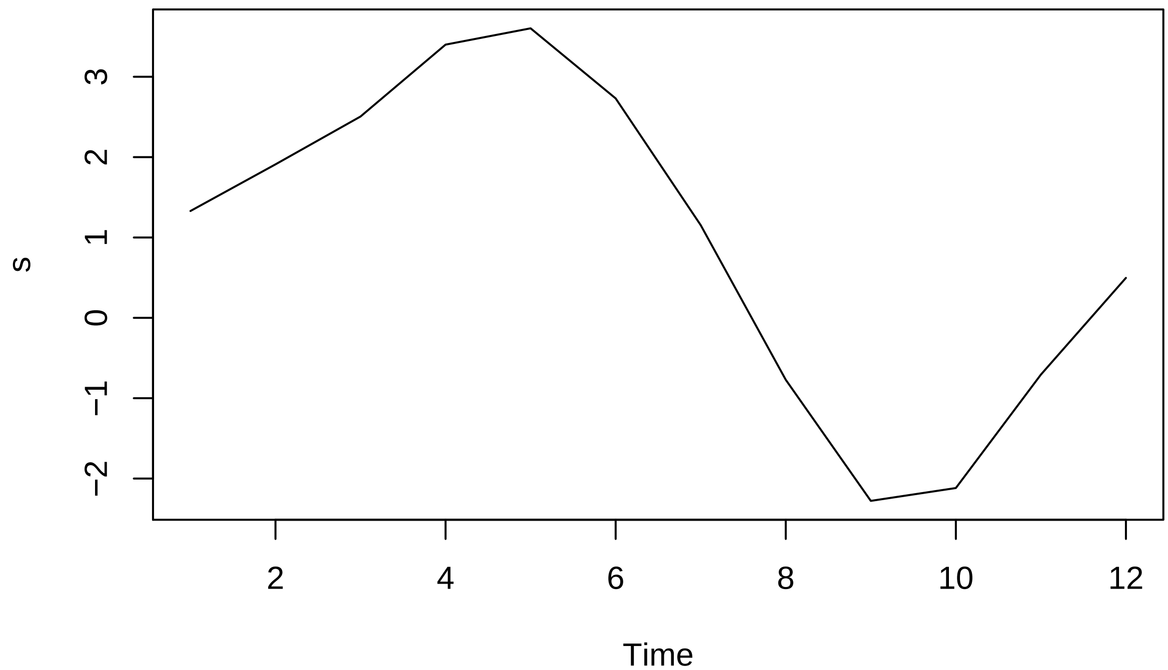
```r
r = co2-glat1 #Zeitreihe vom Trent befreit
m = 1997-1958 #volle Jahre
l = 12 #periodenlänge(12Monate)
```

```r
plot(r)
```



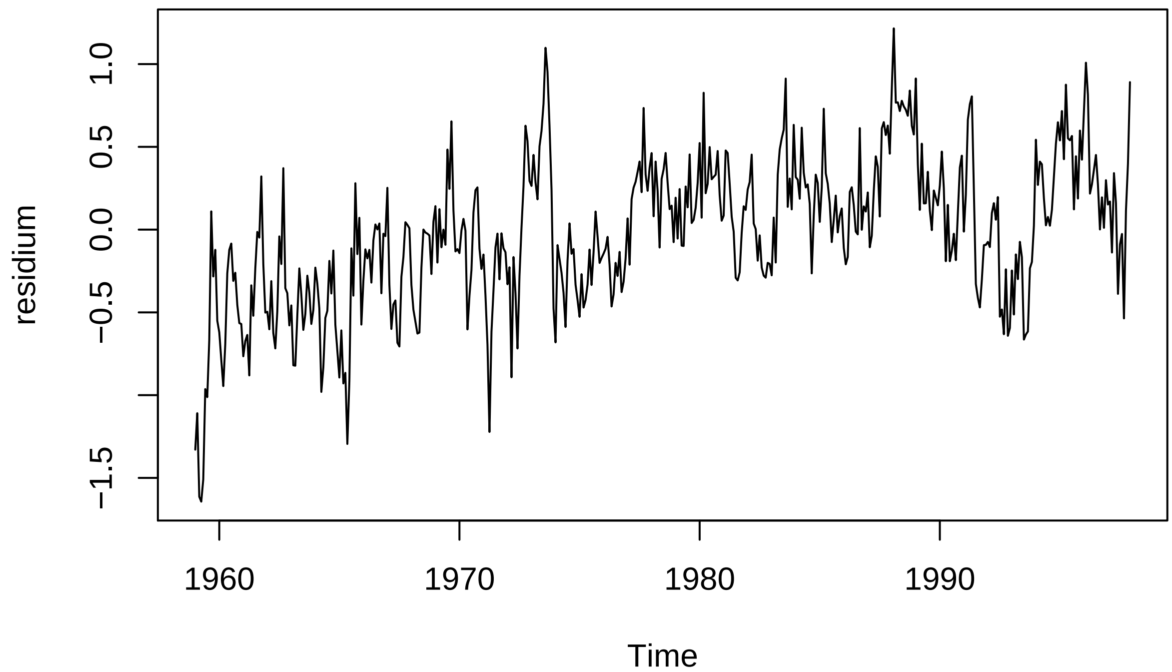

```
saison = function(zeitreihe,m,l){
  ges = c()
  for(j in 1:l){
    t = c()
    for(i in 0:m){
      t = append(t,zeitreihe[j+i*l])
    }
    ges = append(ges,1/m*sum(t))
  }
  return(ges)
}
```

```
s = saison(r,38,12)
ts.plot(s,type = 'l')
```



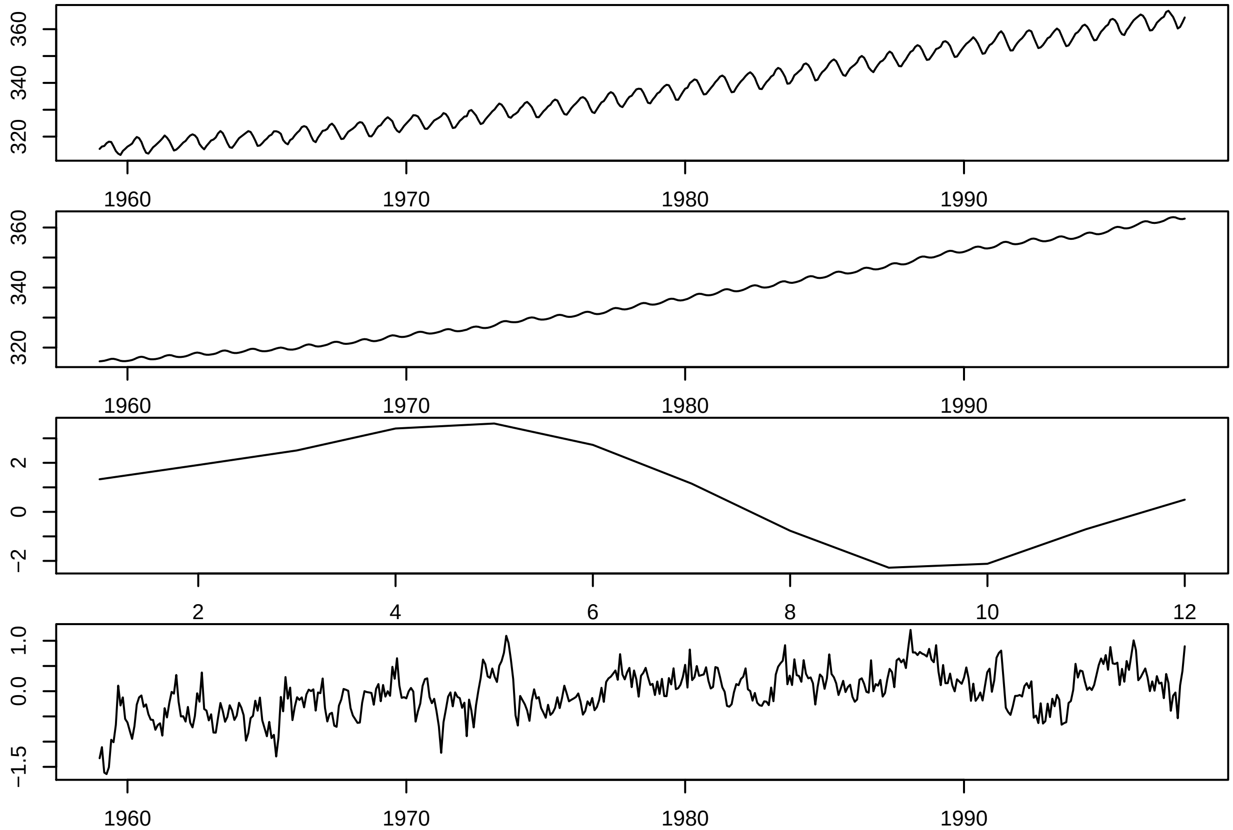
d)

```
residium = co2-(glat1+s)
ts.plot(residium)
```



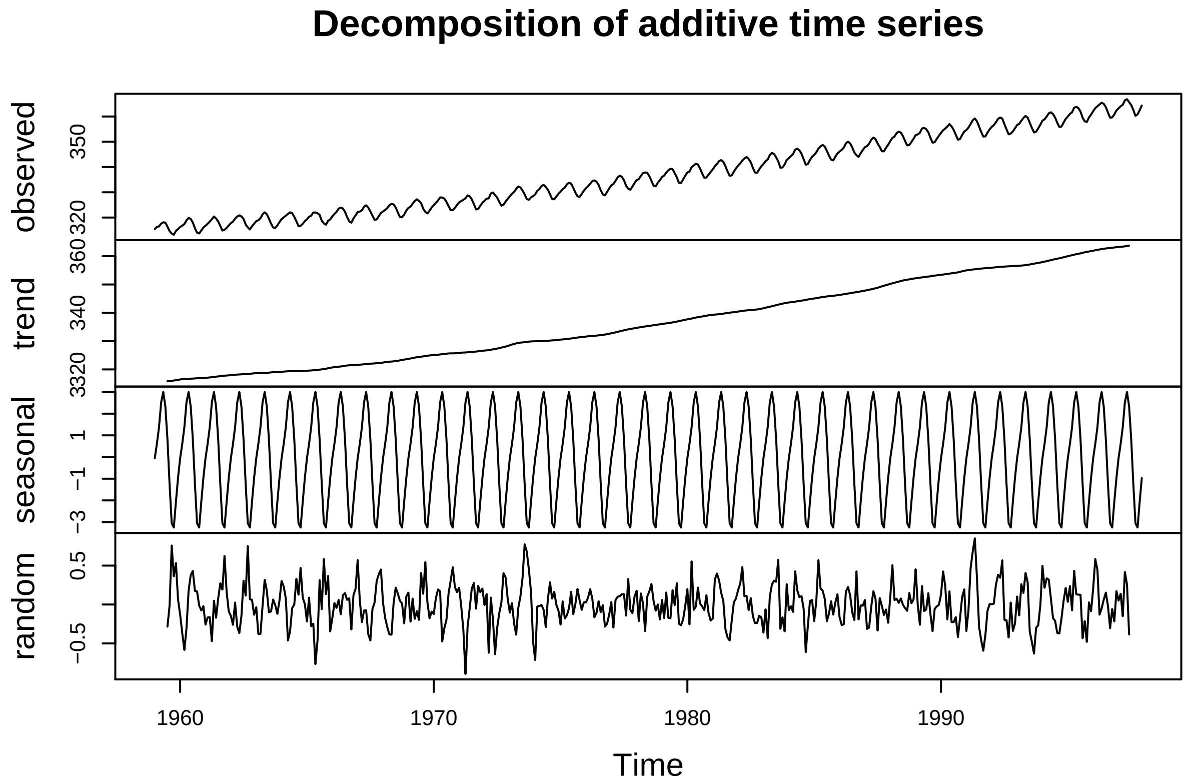
e)

```
par(mfrow=c(4,1), mar=c(1,1,1,0.5), oma=c(1,1,0.5,0.5))
plot(co2)
plot(glat1)
plot(s,type = 'l')
plot(residium)
```



f)

```
h = decompose(co2,type='additiv')
plot(h)
```



Aufgabe 6.2

a)

Im Frequenzbereich geht die Ortsinformation verlohren, es werden nur die zu überlagernden Frequenzen abgebildet.
Im Ortsbereich ist nicht zu erkennen welche Frequenzen das Signal hat, dafür ist die genaue Lage ersichtlich.

b)

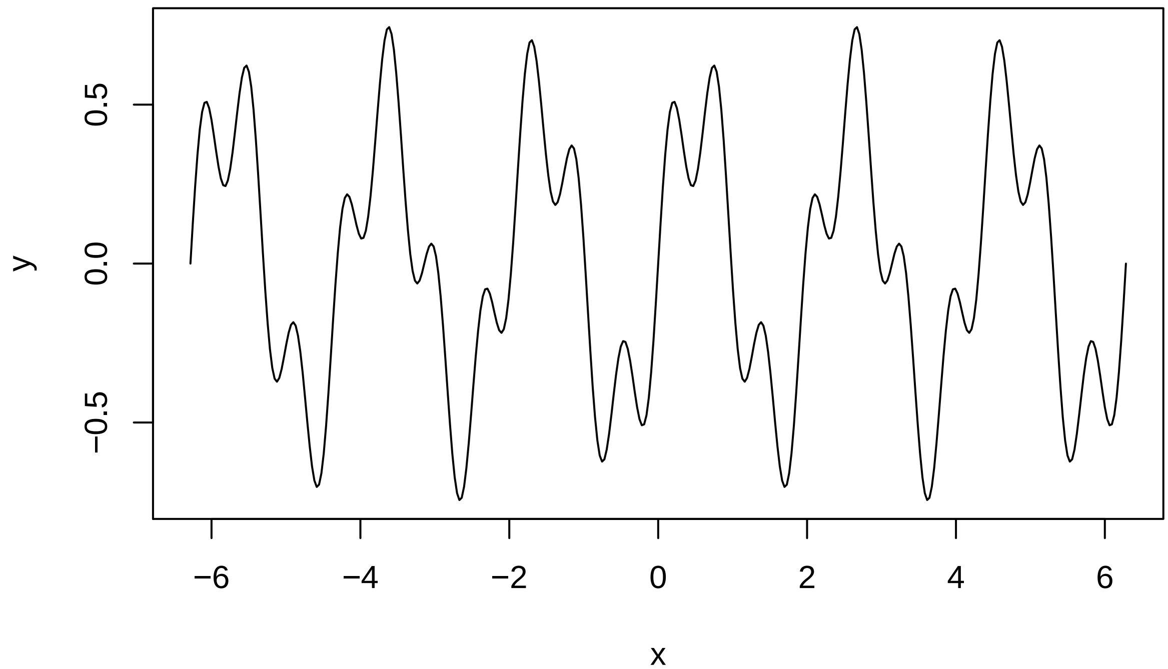
Ein Signal ist nur denn rekonstruierbar, wenn die Abtastfrequenz mehr als doppelt so hoch ist wie die höchste Frequenz im Signal.
Sonst aliassing

c)

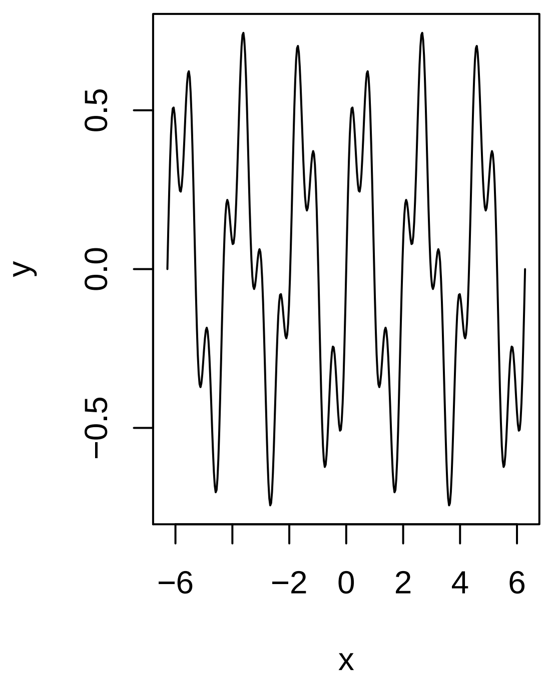
Tiefpassfilter -> Läst tiefe Frequenzen durch und filtert hohe weg.
Filter im Ortsbereich -> Multiplikation im Frequenzbereich

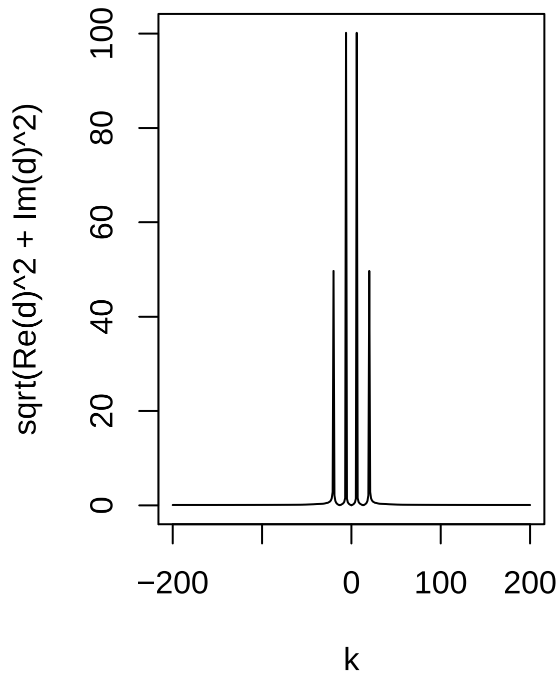
d)

```
x = seq(-2*pi,2*pi,pi/100)
y = 0.5*sin(3*x)+0.25*sin(10*x)
plot(x,y,type = 'l')
```




```
f = fft(y)
d = fftshift(f) #SynchWave
par(mfrow=c(1,2))
plot(x,y,type = 'l')
#plot(Re(f),type = 'l') #mittenfrequenz|Symetrich um die höchste Frequenz in der Mitte.
#plot(Im(f),type = 'l')

k = seq(from = -200, to = 200, by = 1)
plot(k,sqrt(Re(d)^2+Im(d)^2),type='l') #Intensitätsbild
```





Es ist zu erkennen, dass die Funktion aus zwei Frequenzen besteht wobei die tiefe frequenz eine größere Amplitude besitzt.
Duch das fftshiften ist die Mittenfrequenz in der mitte und die Frequenz nimmt nach Außen hin zu.
Es ist um die Nullpunkt symetrisch.

e)

$W(j,s) = \sum_{k=0}^{n-1} f(k) \cdot \overline{\Psi}(\frac{k-j}{s})$ wobei s die Skalierung und j die Translation der Funktion Ψ ist.
Die Funktion Ψ kann gewählt werden in der Vorlesung -> Mexikanischer Hut.
Die Wavelet-Transformation ist in der Vorlesung als Kreuzkorrelation der gegebenen Funktion mit der Wavelet-Funktion vorgestellt worden.
(Faltung der Wavelet-Funktion mit der gegebenen Funktion(multiplikation im Frequenzbereich) -> Hoher Wert bei hoher Übereinstimmung.)

Es gibt auch noch die Haar-Wavelet Transformation(Funktionspaare ->(1)Mittelwertmatrix, (2)Abweichungsmatrix),(3)wider von vorn)

f)

Die Wavelet-Transformation ergibt ein Grauwertbild welches anzeigt wo die Wavelet-Funktion und die gegebene Funktion übereinstimmt.
Es ist dehmentsprechender der Ort zu erkennen wo die beiden Funktionen sich gleichen.(Funktion kann begrenzte Welle mit Frequenz sein)

Auf der y-Achse ist sie Skalierung(s) der Wavelet-Funktion(Ψ) aufgetragen.

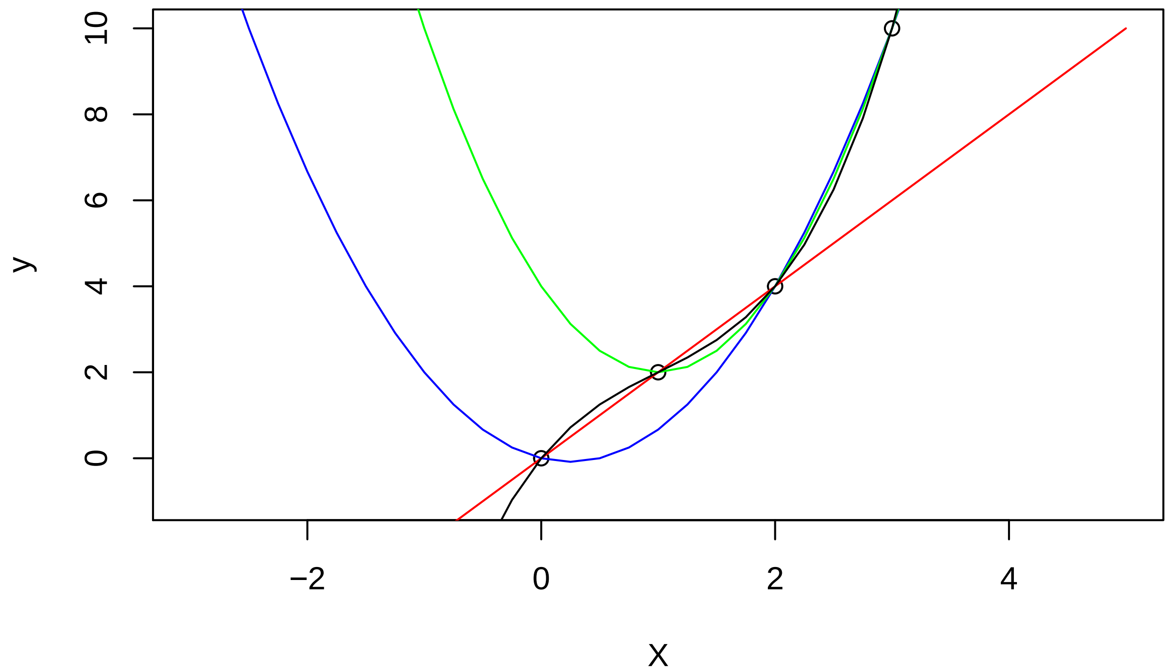
Bei der Fourier-Transformation bekommt man alle im Singal enthaltenen Frequenzen aber Ortsunabhängig.(Globale Aussage aber nicht lokal)

Aufgabe 6.3

a)

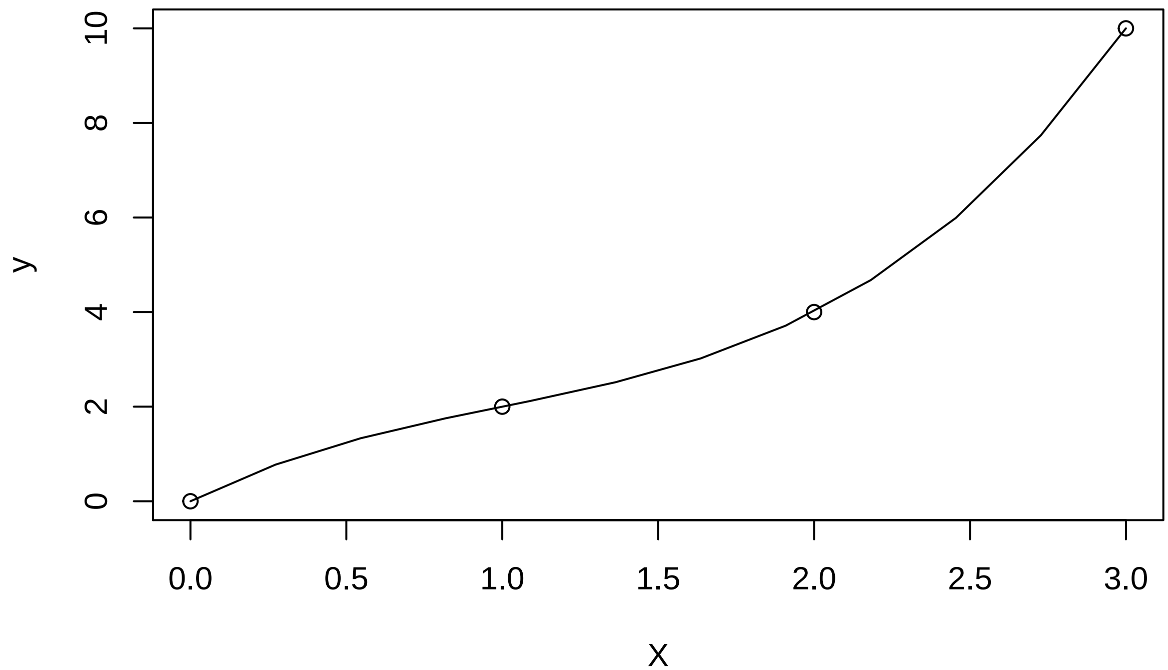
Funktion hat Form $f(t) = \sum_{k=0}^{2}(c_k \cdot e_k(t))$ mit $e_0 = 1$, $e_1 = t$ und $e_2 = t^2$
LGS mit Punkten -> Nicht Lösbar (nur approximierte Näherung)
($f(t) = \sum_{k=0}^{3}(c_k \cdot e_k(t))$ und $e_0 = 1$, $e_1 = t$, $e_2 = t^2$ und $e_3 = t^3$ -> Lösbar)

```
X = c(0,1,2,3)
y = c(0,2,4,10)
plot(X,y,xlim=c(-3,5),ylim=c(-1,10))


x = seq(from = -3, to = 5, by = 0.25)

c0=0
c1=2
c2=0

funktion  = c0*1+c1*x+c2*x*x
lines(x,funktion,col='red')

c0=0
c1=-2/3
c2=4/3

funktion  = c0*1+c1*x+c2*x*x
lines(x,funktion,col='blue')

c0=4
c1=-4
c2=2

funktion  = c0*1+c1*x+c2*x*x
lines(x,funktion,col='green')

c0=0
c1=10/3
c2=-2
c3=2/3

funktion  = c0*1+c1*x+c2*x*x+c3*x*x*x
lines(x,funktion)
```



b) & c)

```
plot(X,y)
lines(spline(X,y))
```



d)

Interpolation: findet Funktion die die gegebenen Werte annimmt.
Approximation: findet Funktion die den gegebenen Werten möglichst nahe kommt.