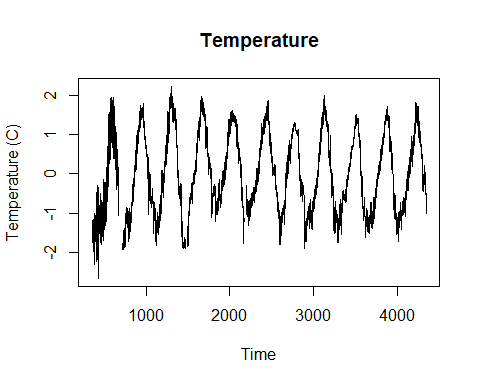
2021\_Wavelets

#Import historical data #ALEXANDER ##Read in Data

# Water temperature - response variable  
# Hourly temperature time series dataset from Alexander Pond called "APH" (Alexander Pond Historical)   
# Includes data from 16 December 2010 to 24 November 2021  
# Data collected by Mid Klamath Watershed Council, see metadata for more details  
  
APH <- read.csv("Alexander\_Historical\_2.csv")  
APH$date <- lubridate::mdy\_hm(APH$Date\_Time) #convert dates to POSIXct format and bin by hour  
  
#Check for missing data  
missing\_data <- APH[!complete.cases(APH),]   
missing\_data   
  
#Bin data by hour  
APH$hour <- lubridate::round\_date(APH$date, unit="hour")   
head(APH) #check the dataset start date, use for "hour" sequence  
tail(APH) #check the dataset end date, use for "hour" sequence  
  
#Create hourly sequence to ensure all missing data is accounted for  
hour <- seq(mdy\_h('12/16/2010 13'),mdy\_h('11/24/2021 14'),by = "hour") #Create an object that goes hour by hour for the entire time series   
hour <- as.data.frame(hour)   
APH <- left\_join(hour, APH)

## Joining, by = "hour"

missing\_data <- APH[!complete.cases(APH),]   
missing\_data   
  
#z score s  
APH$zTemp <- zscore(APH$Temp)  
  
#Convert to time series  
APH\_ts <- ts(APH$zTemp, start = c(351, 13), frequency = 24) # This time series starts on 16 Dec 2010 at~13:00, so it starts on day 351 at hour 13 and the frequency is 24 (24 hours per day)  
ts.plot(APH\_ts,main="Temperature",ylab = "Temperature (C)", xlab = "Time")



##Interpolate missing data

#Run ARIMA to interpolate missing data  
y <- APH\_ts  
date\_s <- APH$hour  
y\_na <- ifelse(is.na(y),0,NA)  
  
fit <- auto.arima(y,trace=TRUE) #fit limited number of models (faster)

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(2,0,2)(1,1,1)[24] with drift : Inf  
## ARIMA(0,0,0)(0,1,0)[24] with drift : -180159.6  
## ARIMA(1,0,0)(1,1,0)[24] with drift : -455692.7  
## ARIMA(0,0,1)(0,1,1)[24] with drift : -151172.4  
## ARIMA(0,0,0)(0,1,0)[24] : -180160.3  
## ARIMA(1,0,0)(0,1,0)[24] with drift : -450186.9  
## ARIMA(1,0,0)(2,1,0)[24] with drift : -463544.8  
## ARIMA(1,0,0)(2,1,1)[24] with drift : Inf  
## ARIMA(1,0,0)(1,1,1)[24] with drift : Inf  
## ARIMA(0,0,0)(2,1,0)[24] with drift : -184292.9  
## ARIMA(2,0,0)(2,1,0)[24] with drift : Inf  
## ARIMA(1,0,1)(2,1,0)[24] with drift : -346161  
## ARIMA(0,0,1)(2,1,0)[24] with drift : -152991.4  
## ARIMA(2,0,1)(2,1,0)[24] with drift : -350220.9  
## ARIMA(1,0,0)(2,1,0)[24] : -463546.7  
## ARIMA(1,0,0)(1,1,0)[24] : -455694.7  
## ARIMA(1,0,0)(2,1,1)[24] : Inf  
## ARIMA(1,0,0)(1,1,1)[24] : Inf  
## ARIMA(0,0,0)(2,1,0)[24] : -184292.5  
## ARIMA(2,0,0)(2,1,0)[24] : Inf  
## ARIMA(1,0,1)(2,1,0)[24] : -346162.9  
## ARIMA(0,0,1)(2,1,0)[24] : -152986.9  
## ARIMA(2,0,1)(2,1,0)[24] : -350222.8  
##   
## Now re-fitting the best model(s) without approximations...  
##   
## ARIMA(1,0,0)(2,1,0)[24] : Inf  
## ARIMA(1,0,0)(2,1,0)[24] with drift : Inf  
## ARIMA(1,0,0)(1,1,0)[24] : Inf  
## ARIMA(1,0,0)(1,1,0)[24] with drift : Inf  
## ARIMA(1,0,0)(0,1,0)[24] with drift : Inf  
## ARIMA(2,0,1)(2,1,0)[24] : Inf  
## ARIMA(2,0,1)(2,1,0)[24] with drift : Inf  
## ARIMA(1,0,1)(2,1,0)[24] : -461214  
##   
## Best model: ARIMA(1,0,1)(2,1,0)[24]

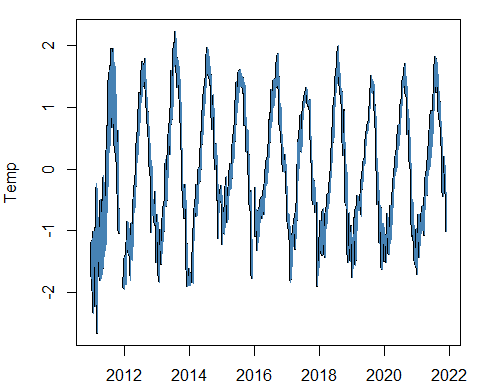
summary(fit) #Take a closer look at the best fitted model

## Series: y   
## ARIMA(1,0,1)(2,1,0)[24]   
##   
## Coefficients:  
## ar1 ma1 sar1 sar2  
## 0.9796 0.0360 -0.2904 -0.2942  
## s.e. 0.0007 0.0024 0.0032 0.0032  
##   
## sigma^2 estimated as 0.0004044: log likelihood=230612  
## AIC=-461214 AICc=-461214 BIC=-461166.6  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set 9.161849e-06 0.02010788 0.008321832 1.233202 6.684451 0.1293441  
## ACF1  
## Training set -0.3061128

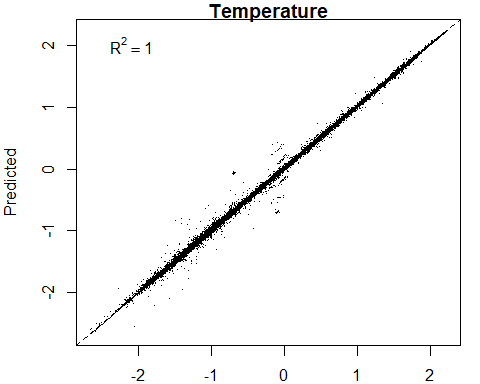
forecast\_fit <- forecast(y,model=fit) #Predict values using the calibration dataset

##Plot interpolated data

#Plot the observed and interpolated temperature regimes  
par(mar=c(2,4,1,1))   
plot(date\_s,y,xlab="Time",ylab="Temp",lwd=2,type="l")   
lines(date\_s,forecast\_fit$fitted,col="steelblue")

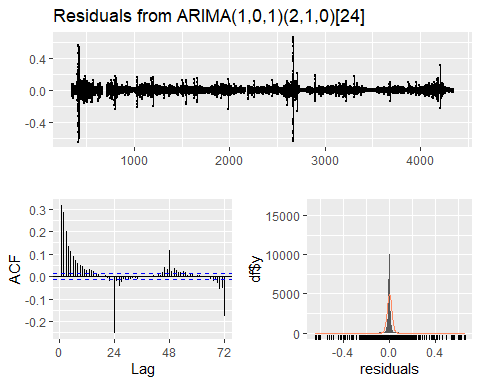


#Plot predicted versus observed values  
scatter.smooth(y,forecast\_fit$fitted,xlab="Observed",ylab="Predicted",pch=".",main = "Temperature")  
abline(0,1,lty=2)  
R2 = round(cor.test(y,forecast\_fit$fitted,na.rm=T)$estimate^2,2)  
mtext(side=3,line=-2,adj=0.1,bquote(R^2 == .(R2)))



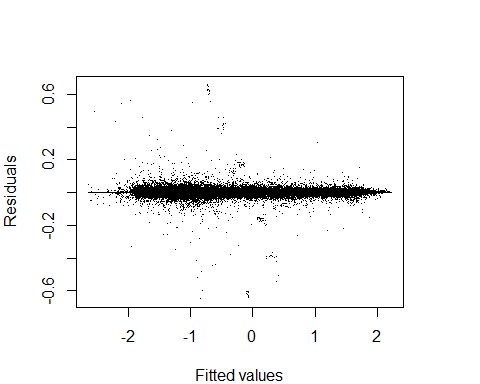
##Check residuals of interpolated data

checkresiduals(fit) #also gives the results for the Ljung\_Box test with H0 = randomly distributed errors (white noise)



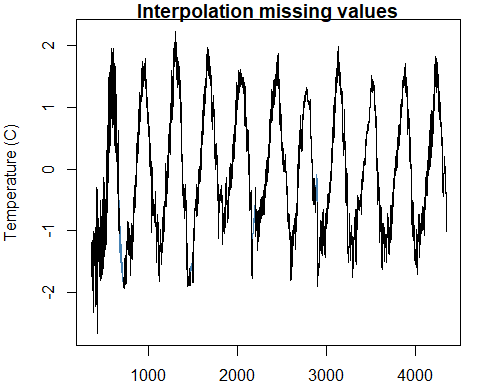
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(1,0,1)(2,1,0)[24]  
## Q\* = 180154, df = 44, p-value < 2.2e-16  
##   
## Model df: 4. Total lags used: 48

#plot residuals versus fitted values (=check for heterosedasticity)  
par(mar=c(4,4,4,4))  
scatter.smooth(forecast\_fit$fitted,forecast\_fit$residuals,pch = ".",ylab="Residuals",xlab="Fitted values")



##Smooth interpolated data

#Interpolate missing values using a Kalman filter (=smoother)  
y\_inter <- na\_kalman(y,model=fit$model)   
  
#Plot the results  
par(mar=c(2,4,1,1))  
plot(y\_inter,xlab="",ylab="Temperature (C)",col="steelblue",main="Interpolation missing values")  
lines(y,col="black")



##Format interpolated time series into a dataframe

#Put the interpolated temperature dataset (y\_inter) into a dataset with the correct dates   
x <- as.data.frame(y\_inter) #change y\_inter from a ts object to a dataframe  
x$ID <- seq.int(nrow(x)) #add a unique ID  
date <- as.data.frame(APH$hour) #make the unique "hour" index from APH into a separate dataframe  
date$ID <- seq.int(nrow(date)) #create unique ID that aligns with x  
y\_inter\_df <- merge(x,date,"ID") #merge the two dataframes  
colnames(y\_inter\_df)<-c("ID","temp","date") #rename columns  
saveRDS(y\_inter\_df,"y\_inter\_df.rds")

##Check for NAs

missing\_data <- y\_inter\_df[!complete.cases(y\_inter\_df),]   
missing\_data

## [1] ID temp date  
## <0 rows> (or 0-length row.names)

##Wavelet

APH.w <- analyze.wavelet(y\_inter\_df, "temp",  
 loess.span = 0,  
 dt = 1/24,   
 make.pval = TRUE, n.sim = 1000,  
 date.format = "%Y-%m-%d-%h")

## Starting wavelet transformation...  
## ... and simulations...   
## | | | 0% | | | 1% | |= | 1% | |= | 2% | |== | 2% | |== | 3% | |== | 4% | |=== | 4% | |=== | 5% | |==== | 5% | |==== | 6% | |===== | 6% | |===== | 7% | |===== | 8% | |====== | 8% | |====== | 9% | |======= | 9% | |======= | 10% | |======= | 11% | |======== | 11% | |======== | 12% | |========= | 12% | |========= | 13% | |========= | 14% | |========== | 14% | |========== | 15% | |=========== | 15% | |=========== | 16% | |============ | 16% | |============ | 17% | |============ | 18% | |============= | 18% | |============= | 19% | |============== | 19% | |============== | 20% | |============== | 21% | |=============== | 21% | |=============== | 22% | |================ | 22% | |================ | 23% | |================ | 24% | |================= | 24% | |================= | 25% | |================== | 25% | |================== | 26% | |=================== | 26% | |=================== | 27% | |=================== | 28% | |==================== | 28% | |==================== | 29% | |===================== | 29% | |===================== | 30% | |===================== | 31% | |====================== | 31% | |====================== | 32% | |======================= | 32% | |======================= | 33% | |======================= | 34% | |======================== | 34% | |======================== | 35% | |========================= | 35% | |========================= | 36% | |========================== | 36% | |========================== | 37% | |========================== | 38% | |=========================== | 38% | |=========================== | 39% | |============================ | 39% | |============================ | 40% | |============================ | 41% | |============================= | 41% | |============================= | 42% | |============================== | 42% | |============================== | 43% | |============================== | 44% | |=============================== | 44% | |=============================== | 45% | |================================ | 45% | |================================ | 46% | |================================= | 46% | |================================= | 47% | |================================= | 48% | |================================== | 48% | |================================== | 49% | |=================================== | 49% | |=================================== | 50% | |=================================== | 51% | |==================================== | 51% | |==================================== | 52% | |===================================== | 52% | |===================================== | 53% | |===================================== | 54% | |====================================== | 54% | |====================================== | 55% | |======================================= | 55% | |======================================= | 56% | |======================================== | 56% | |======================================== | 57% | |======================================== | 58% | |========================================= | 58% | |========================================= | 59% | |========================================== | 59% | |========================================== | 60% | |========================================== | 61% | |=========================================== | 61% | |=========================================== | 62% | |============================================ | 62% | |============================================ | 63% | |============================================ | 64% | |============================================= | 64% | |============================================= | 65% | |============================================== | 65% | |============================================== | 66% | |=============================================== | 66% | |=============================================== | 67% | |=============================================== | 68% | |================================================ | 68% | |================================================ | 69% | |================================================= | 69% | |================================================= | 70% | |================================================= | 71% | |================================================== | 71% | |================================================== | 72% | |=================================================== | 72% | |=================================================== | 73% | |=================================================== | 74% | |==================================================== | 74% | |==================================================== | 75% | |===================================================== | 75% | |===================================================== | 76% | |====================================================== | 76% | |====================================================== | 77% | |====================================================== | 78% | |======================================================= | 78% | |======================================================= | 79% | |======================================================== | 79% | |======================================================== | 80% | |======================================================== | 81% | |========================================================= | 81% | |========================================================= | 82% | |========================================================== | 82% | |========================================================== | 83% | |========================================================== | 84% | |=========================================================== | 84% | |=========================================================== | 85% | |============================================================ | 85% | |============================================================ | 86% | |============================================================= | 86% | |============================================================= | 87% | |============================================================= | 88% | |============================================================== | 88% | |============================================================== | 89% | |=============================================================== | 89% | |=============================================================== | 90% | |=============================================================== | 91% | |================================================================ | 91% | |================================================================ | 92% | |================================================================= | 92% | |================================================================= | 93% | |================================================================= | 94% | |================================================================== | 94% | |================================================================== | 95% | |=================================================================== | 95% | |=================================================================== | 96% | |==================================================================== | 96% | |==================================================================== | 97% | |==================================================================== | 98% | |===================================================================== | 98% | |===================================================================== | 99% | |======================================================================| 99% | |======================================================================| 100%  
## Class attributes are accessible through following names:  
## series loess.span dt dj Wave Phase Ampl Power Power.avg Power.pval Power.avg.pval Ridge Period Scale nc nr coi.1 coi.2 axis.1 axis.2 date.format date.tz

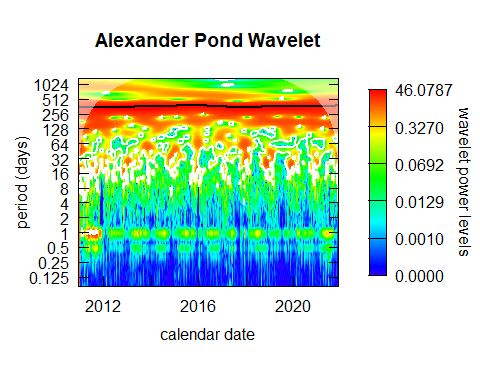
str(APH.w)

## List of 22  
## $ series :'data.frame': 95933 obs. of 2 variables:  
## ..$ date: POSIXct[1:95933], format: "2010-12-16 13:00:00" "2010-12-16 14:00:00" ...  
## ..$ temp: num [1:95933] -1.48 -1.45 -1.44 -1.45 -1.47 ...  
## $ loess.span : num 0  
## $ dt : num 0.0417  
## $ dj : num 0.05  
## $ Wave : cplx [1:280, 1:95933] -0.197+0.042i -0.226+0.052i -0.254+0.063i ...  
## $ Phase : num [1:280, 1:95933] 2.93 2.91 2.9 2.88 2.85 ...  
## $ Ampl : num [1:280, 1:95933] 0.714 0.808 0.896 0.972 1.034 ...  
## $ Power : num [1:280, 1:95933] 0.51 0.653 0.803 0.945 1.069 ...  
## $ Power.avg : num [1:280] 0.00223 0.00229 0.00217 0.00192 0.00158 ...  
## $ Power.pval : num [1:280, 1:95933] 0.848 0.845 0.839 0.832 0.816 0.809 0.799 0.802 0.789 0.783 ...  
## $ Power.avg.pval: num [1:280] 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ridge : num [1:280, 1:95933] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Period : num [1:280] 0.0833 0.0863 0.0893 0.0925 0.0957 ...  
## $ Scale : num [1:280] 0.0796 0.0824 0.0853 0.0883 0.0914 ...  
## $ nc : int 95933  
## $ nr : int 280  
## $ coi.1 : num [1:95937] 0.979 0.979 1 1.042 1.083 ...  
## $ coi.2 : num [1:95937] 10.39 -3.61 -20.63 -4.02 -3.02 ...  
## $ axis.1 : num [1:95933] 1 1.04 1.08 1.12 1.17 ...  
## $ axis.2 : num [1:280] -3.58 -3.53 -3.48 -3.43 -3.38 ...  
## $ date.format : chr "%Y-%m-%d-%h"  
## $ date.tz : NULL  
## - attr(\*, "class")= chr "analyze.wavelet"

saveRDS(APH.w,"APH.w.rds")

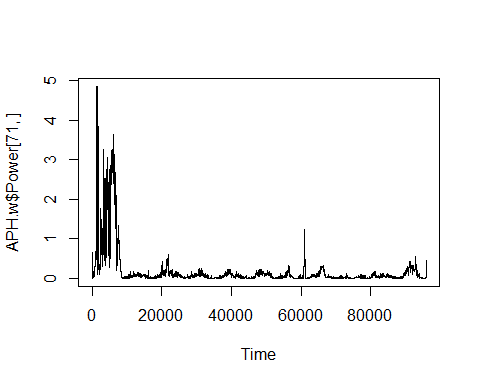
##Wavelet Image

wt.image(APH.w, color.key = "quantile", n.levels = 250, main = "Alexander Pond Wavelet",   
 legend.params = list(lab = "wavelet power levels", mar = 6.5, lab.line=4, label.digits = 4), label.time.axis=TRUE,show.date=TRUE,  
 periodlab = "period (days)")



##Pull 24 hr period

plot.ts(APH.w$Power[71,])

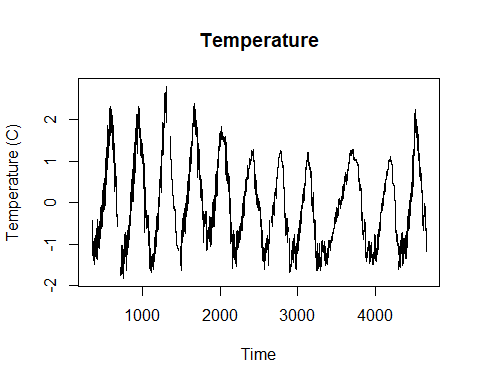


#STENDER ##Read in Data

# Water temperature - response variable  
# Hourly temperature time series dataset from Stender Pond called "SPH" (Stender Pond Historical)   
# Includes data from 16 December 2010 to 24 November 2021  
# Data collected by Mid Klamath Watershed Council, see metadata for more details  
  
SPH <- read.csv("Stender\_Historical\_2.csv")  
SPH$date <- lubridate::mdy\_hm(SPH$Date\_Time) #convert dates to POSIXct format and bin by hour  
  
#Check for missing data  
missing\_data <- SPH[!complete.cases(SPH),]   
missing\_data   
  
#Bin data by hour  
SPH$hour <- lubridate::round\_date(SPH$date, unit="hour")  
head(SPH) #check the dataset start date, use for "hour" sequence  
tail(SPH) #check the dataset end date, use for "hour" sequence  
  
#Create hourly sequence to ensure all missing data is accounted for  
hour <- seq(mdy\_h('12/16/2010 13'),mdy\_h('11/24/2021 14'),by = "hour") #Create an object that goes hour by hour for the entire time series   
hour <- as.data.frame(hour)   
SPH <- left\_join(hour, SPH)

## Joining, by = "hour"

missing\_data <- SPH[!complete.cases(SPH),]   
missing\_data   
  
#z score   
SPH$zTemp <- zscore(SPH$Temp)  
  
#Convert to time series  
SPH\_ts <- ts(SPH$zTemp, start = c(351, 13), frequency = 24) # This time series starts on 16 Dec 2010 at~13:00, so it starts on day 351 at hour 13 and the frequency is 24 (24 hours per day)  
ts.plot(SPH\_ts,main="Temperature",ylab = "Temperature (C)", xlab = "Time")



##Interpolate missing data

#Run ARIMA to interpolate missing data  
y2 <- SPH\_ts  
date\_s2 <- SPH$hour  
y\_na2 <- ifelse(is.na(y2),0,NA)  
  
fit2 <- auto.arima(y2,trace=TRUE) #fit limited number of models (faster)

##   
## Fitting models using approximations to speed things up...  
##   
## ARIMA(2,0,2)(1,1,1)[24] with drift : Inf  
## ARIMA(0,0,0)(0,1,0)[24] with drift : -214977.1  
## ARIMA(1,0,0)(1,1,0)[24] with drift : Inf  
## ARIMA(0,0,1)(0,1,1)[24] with drift : -293411.7  
## ARIMA(0,0,0)(0,1,0)[24] : -214976.4  
## ARIMA(0,0,1)(0,1,0)[24] with drift : -293136.8  
## ARIMA(0,0,1)(1,1,1)[24] with drift : -294401.7  
## ARIMA(0,0,1)(1,1,0)[24] with drift : -294313.2  
## ARIMA(0,0,1)(2,1,1)[24] with drift : -294559.4  
## ARIMA(0,0,1)(2,1,0)[24] with drift : -294201.1  
## ARIMA(0,0,1)(2,1,2)[24] with drift : -294587.1  
## ARIMA(0,0,1)(1,1,2)[24] with drift : -294402.4  
## ARIMA(0,0,0)(2,1,2)[24] with drift : -183150.8  
## ARIMA(1,0,1)(2,1,2)[24] with drift : Inf  
## ARIMA(0,0,2)(2,1,2)[24] with drift : -368593.8  
## ARIMA(0,0,2)(1,1,2)[24] with drift : -368442.3  
## ARIMA(0,0,2)(2,1,1)[24] with drift : -368159.5  
## ARIMA(0,0,2)(1,1,1)[24] with drift : -368452.1  
## ARIMA(1,0,2)(2,1,2)[24] with drift : Inf  
## ARIMA(0,0,3)(2,1,2)[24] with drift : -417015.9  
## ARIMA(0,0,3)(1,1,2)[24] with drift : -416732  
## ARIMA(0,0,3)(2,1,1)[24] with drift : -416091.3  
## ARIMA(0,0,3)(1,1,1)[24] with drift : -416666  
## ARIMA(1,0,3)(2,1,2)[24] with drift : Inf  
## ARIMA(0,0,4)(2,1,2)[24] with drift : -449608.3  
## ARIMA(0,0,4)(1,1,2)[24] with drift : -448985.7  
## ARIMA(0,0,4)(2,1,1)[24] with drift : -448763.6  
## ARIMA(0,0,4)(1,1,1)[24] with drift : -448800.3  
## ARIMA(1,0,4)(2,1,2)[24] with drift : Inf  
## ARIMA(0,0,5)(2,1,2)[24] with drift : -471490.3  
## ARIMA(0,0,5)(1,1,2)[24] with drift : -471272.3  
## ARIMA(0,0,5)(2,1,1)[24] with drift : -470562.8  
## ARIMA(0,0,5)(1,1,1)[24] with drift : -471100  
## ARIMA(1,0,5)(2,1,2)[24] with drift : Inf  
## ARIMA(0,0,5)(2,1,2)[24] : -471474.5  
##   
## Now re-fitting the best model(s) without approximations...  
##   
## ARIMA(0,0,5)(2,1,2)[24] with drift : Inf  
## ARIMA(0,0,5)(2,1,2)[24] : Inf  
## ARIMA(0,0,5)(1,1,2)[24] with drift : Inf  
## ARIMA(0,0,5)(1,1,1)[24] with drift : -544057.7  
##   
## Best model: ARIMA(0,0,5)(1,1,1)[24] with drift

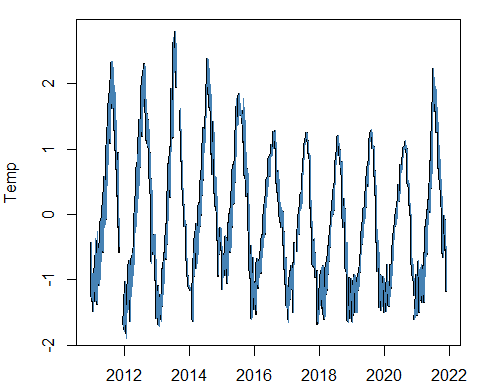
summary(fit2) #Take a closer look at the best fitted model

## Series: y2   
## ARIMA(0,0,5)(1,1,1)[24] with drift   
##   
## Coefficients:  
## ma1 ma2 ma3 ma4 ma5 sar1 sma1 drift  
## 1.7500 2.1459 1.8663 1.1753 0.4220 -0.5075 0.2327 0  
## s.e. 0.0037 0.0064 0.0065 0.0044 0.0024 0.0147 0.0173 0  
##   
## sigma^2 estimated as 0.0002425: log likelihood=272037.9  
## AIC=-544057.7 AICc=-544057.7 BIC=-543971.8  
##   
## Training set error measures:  
## ME RMSE MAE MPE MAPE MASE  
## Training set -2.014778e-06 0.01556897 0.0106746 0.195905 5.954043 0.1806112  
## ACF1  
## Training set 0.1827068

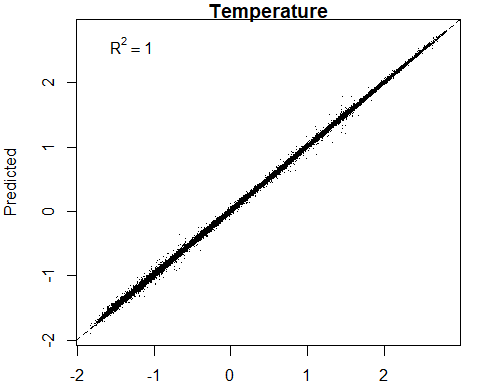
forecast\_fit2 <- forecast(y2,model=fit2) #Predict values using the calibration dataset

##Plot interpolated data

#Plot the observed and interpolated temperature regimes  
par(mar=c(2,4,1,1))   
plot(date\_s2,y2,xlab="Time",ylab="Temp",lwd=2,type="l")   
lines(date\_s2,forecast\_fit2$fitted,col="steelblue")

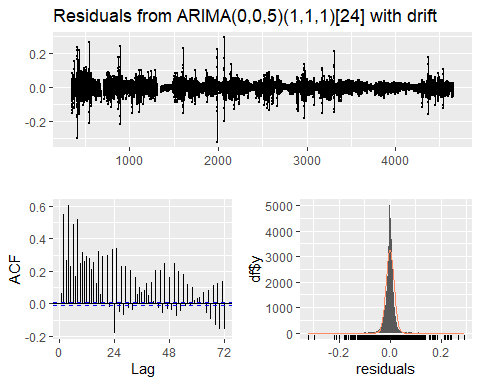


#Plot predicted versus observed values  
scatter.smooth(y2,forecast\_fit2$fitted,xlab="Observed",ylab="Predicted",pch=".",main = "Temperature")  
abline(0,1,lty=2)  
R2 = round(cor.test(y2,forecast\_fit2$fitted,na.rm=T)$estimate^2,2)  
mtext(side=3,line=-2,adj=0.1,bquote(R^2 == .(R2)))



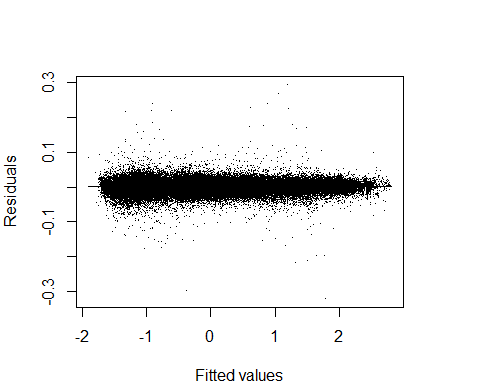
##Check residuals of interpolated data

#Check residuals of the interpolation process  
checkresiduals(fit2) #also gives the results for the Ljung\_Box test with H0 = randomly distributed errors (white noise)



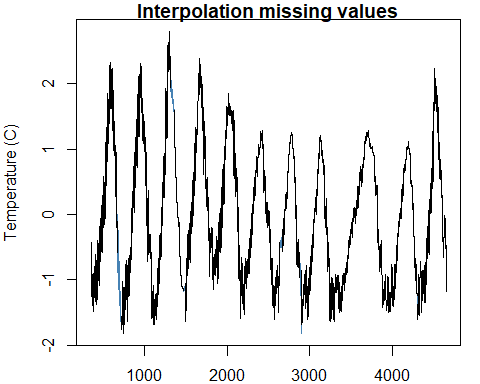
##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,0,5)(1,1,1)[24] with drift  
## Q\* = 741908, df = 40, p-value < 2.2e-16  
##   
## Model df: 8. Total lags used: 48

#plot residuals versus fitted values (=check for heterosedasticity)  
par(mar=c(4,4,4,4))  
scatter.smooth(forecast\_fit2$fitted,forecast\_fit2$residuals,pch = ".",ylab="Residuals",xlab="Fitted values")



##Smooth interpolated data

#Interpolate missing values using a Kalman filter (=smoother)  
y\_inter2 <- na\_kalman(y2,model=fit2$model)   
  
#Plot the results  
par(mar=c(2,4,1,1))  
plot(y\_inter2,xlab="",ylab="Temperature (C)",col="steelblue",main="Interpolation missing values")  
lines(y2,col="black")



##Format interpolated time series into a dataframe

#Put the interpolated temperature dataset (y\_inter) into a dataset with the correct dates (  
x2 <- as.data.frame(y\_inter2) #change y\_inter from a ts to a dataframe  
x2$ID <- seq.int(nrow(x2)) #add a unique ID, check it is correct length  
date2 <- as.data.frame(SPH$hour) #make the unique "hour" index from SPH into a separate dataframe  
date2$ID <- seq.int(nrow(date2)) #give that a unique ID that aligns with x  
y\_inter2\_df <- merge(x2,date2,"ID") #merge the two dataframes  
colnames(y\_inter2\_df)<-c("ID","temp","date") #rename columns  
saveRDS(y\_inter2\_df,"y\_inter2\_df.rds")

##Check for NAs

missing\_data <- y\_inter2\_df[!complete.cases(y\_inter2\_df),]   
missing\_data

## [1] ID temp date  
## <0 rows> (or 0-length row.names)

##Wavelet

#run the wavelet  
SPH.w <- analyze.wavelet(y\_inter2\_df, "temp",  
 loess.span = 0,  
 dt = 1/24,   
 make.pval = TRUE, n.sim = 1000,  
 date.format = "%Y-%m-%d-%h")

## Starting wavelet transformation...  
## ... and simulations...   
## | | | 0% | | | 1% | |= | 1% | |= | 2% | |== | 2% | |== | 3% | |== | 4% | |=== | 4% | |=== | 5% | |==== | 5% | |==== | 6% | |===== | 6% | |===== | 7% | |===== | 8% | |====== | 8% | |====== | 9% | |======= | 9% | |======= | 10% | |======= | 11% | |======== | 11% | |======== | 12% | |========= | 12% | |========= | 13% | |========= | 14% | |========== | 14% | |========== | 15% | |=========== | 15% | |=========== | 16% | |============ | 16% | |============ | 17% | |============ | 18% | |============= | 18% | |============= | 19% | |============== | 19% | |============== | 20% | |============== | 21% | |=============== | 21% | |=============== | 22% | |================ | 22% | |================ | 23% | |================ | 24% | |================= | 24% | |================= | 25% | |================== | 25% | |================== | 26% | |=================== | 26% | |=================== | 27% | |=================== | 28% | |==================== | 28% | |==================== | 29% | |===================== | 29% | |===================== | 30% | |===================== | 31% | |====================== | 31% | |====================== | 32% | |======================= | 32% | |======================= | 33% | |======================= | 34% | |======================== | 34% | |======================== | 35% | |========================= | 35% | |========================= | 36% | |========================== | 36% | |========================== | 37% | |========================== | 38% | |=========================== | 38% | |=========================== | 39% | |============================ | 39% | |============================ | 40% | |============================ | 41% | |============================= | 41% | |============================= | 42% | |============================== | 42% | |============================== | 43% | |============================== | 44% | |=============================== | 44% | |=============================== | 45% | |================================ | 45% | |================================ | 46% | |================================= | 46% | |================================= | 47% | |================================= | 48% | |================================== | 48% | |================================== | 49% | |=================================== | 49% | |=================================== | 50% | |=================================== | 51% | |==================================== | 51% | |==================================== | 52% | |===================================== | 52% | |===================================== | 53% | |===================================== | 54% | |====================================== | 54% | |====================================== | 55% | |======================================= | 55% | |======================================= | 56% | |======================================== | 56% | |======================================== | 57% | |======================================== | 58% | |========================================= | 58% | |========================================= | 59% | |========================================== | 59% | |========================================== | 60% | |========================================== | 61% | |=========================================== | 61% | |=========================================== | 62% | |============================================ | 62% | |============================================ | 63% | |============================================ | 64% | |============================================= | 64% | |============================================= | 65% | |============================================== | 65% | |============================================== | 66% | |=============================================== | 66% | |=============================================== | 67% | |=============================================== | 68% | |================================================ | 68% | |================================================ | 69% | |================================================= | 69% | |================================================= | 70% | |================================================= | 71% | |================================================== | 71% | |================================================== | 72% | |=================================================== | 72% | |=================================================== | 73% | |=================================================== | 74% | |==================================================== | 74% | |==================================================== | 75% | |===================================================== | 75% | |===================================================== | 76% | |====================================================== | 76% | |====================================================== | 77% | |====================================================== | 78% | |======================================================= | 78% | |======================================================= | 79% | |======================================================== | 79% | |======================================================== | 80% | |======================================================== | 81% | |========================================================= | 81% | |========================================================= | 82% | |========================================================== | 82% | |========================================================== | 83% | |========================================================== | 84% | |=========================================================== | 84% | |=========================================================== | 85% | |============================================================ | 85% | |============================================================ | 86% | |============================================================= | 86% | |============================================================= | 87% | |============================================================= | 88% | |============================================================== | 88% | |============================================================== | 89% | |=============================================================== | 89% | |=============================================================== | 90% | |=============================================================== | 91% | |================================================================ | 91% | |================================================================ | 92% | |================================================================= | 92% | |================================================================= | 93% | |================================================================= | 94% | |================================================================== | 94% | |================================================================== | 95% | |=================================================================== | 95% | |=================================================================== | 96% | |==================================================================== | 96% | |==================================================================== | 97% | |==================================================================== | 98% | |===================================================================== | 98% | |===================================================================== | 99% | |======================================================================| 99% | |======================================================================| 100%  
## Class attributes are accessible through following names:  
## series loess.span dt dj Wave Phase Ampl Power Power.avg Power.pval Power.avg.pval Ridge Period Scale nc nr coi.1 coi.2 axis.1 axis.2 date.format date.tz

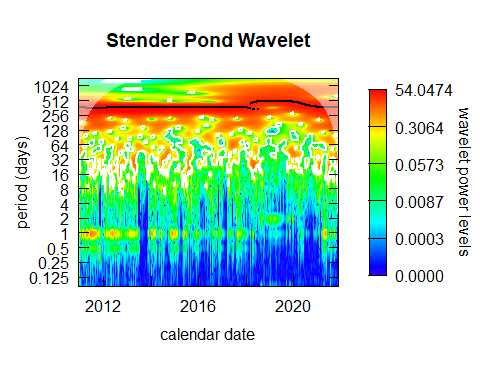
str(SPH.w) # Output

## List of 22  
## $ series :'data.frame': 103302 obs. of 2 variables:  
## ..$ date: POSIXct[1:103302], format: "2010-12-16 13:00:00" "2010-12-16 14:00:00" ...  
## ..$ temp: num [1:103302] -0.434 -0.587 -0.575 -0.587 -0.609 ...  
## $ loess.span : num 0  
## $ dt : num 0.0417  
## $ dj : num 0.05  
## $ Wave : cplx [1:282, 1:103302] -0.0357+0.0195i -0.0411+0.0237i -0.0463+0.0283i ...  
## $ Phase : num [1:282, 1:103302] 2.64 2.62 2.59 2.56 2.53 ...  
## $ Ampl : num [1:282, 1:103302] 0.144 0.165 0.186 0.205 0.223 ...  
## $ Power : num [1:282, 1:103302] 0.0208 0.0273 0.0345 0.0421 0.0495 ...  
## $ Power.avg : num [1:282] 0.000116 0.000136 0.000152 0.000162 0.000167 ...  
## $ Power.pval : num [1:282, 1:103302] 0.989 0.991 0.99 0.988 0.987 0.988 0.982 0.982 0.99 0.984 ...  
## $ Power.avg.pval: num [1:282] 1 1 1 1 1 1 1 1 1 1 ...  
## $ Ridge : num [1:282, 1:103302] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Period : num [1:282] 0.0833 0.0863 0.0893 0.0925 0.0957 ...  
## $ Scale : num [1:282] 0.0796 0.0824 0.0853 0.0883 0.0914 ...  
## $ nc : int 103302  
## $ nr : int 282  
## $ coi.1 : num [1:103306] 0.979 0.979 1 1.042 1.083 ...  
## $ coi.2 : num [1:103306] 10.49 -3.61 -20.63 -4.02 -3.02 ...  
## $ axis.1 : num [1:103302] 1 1.04 1.08 1.12 1.17 ...  
## $ axis.2 : num [1:282] -3.58 -3.53 -3.48 -3.43 -3.38 ...  
## $ date.format : chr "%Y-%m-%d-%h"  
## $ date.tz : NULL  
## - attr(\*, "class")= chr "analyze.wavelet"

saveRDS(SPH.w,"SPH.w.rds")

##Wavelet Image

wt.image(SPH.w, color.key = "quantile", n.levels = 250, main = "Stender Pond Wavelet",   
 legend.params = list(lab = "wavelet power levels", mar = 6.5, lab.line=4, label.digits = 4), label.time.axis=TRUE,show.date=TRUE,  
 periodlab = "period (days)")



##Pull 24 hr period

SPH.w$Period

## [1] 8.333333e-02 8.627208e-02 8.931446e-02 9.246412e-02 9.572486e-02  
## [6] 9.910059e-02 1.025954e-01 1.062134e-01 1.099590e-01 1.138367e-01  
## [11] 1.178511e-01 1.220071e-01 1.263097e-01 1.307640e-01 1.353754e-01  
## [16] 1.401494e-01 1.450918e-01 1.502084e-01 1.555055e-01 1.609894e-01  
## [21] 1.666667e-01 1.725442e-01 1.786289e-01 1.849282e-01 1.914497e-01  
## [26] 1.982012e-01 2.051907e-01 2.124268e-01 2.199180e-01 2.276734e-01  
## [31] 2.357023e-01 2.440143e-01 2.526194e-01 2.615280e-01 2.707508e-01  
## [36] 2.802988e-01 2.901835e-01 3.004168e-01 3.110110e-01 3.219788e-01  
## [41] 3.333333e-01 3.450883e-01 3.572578e-01 3.698565e-01 3.828995e-01  
## [46] 3.964024e-01 4.103815e-01 4.248535e-01 4.398360e-01 4.553468e-01  
## [51] 4.714045e-01 4.880286e-01 5.052389e-01 5.230561e-01 5.415016e-01  
## [56] 5.605976e-01 5.803670e-01 6.008336e-01 6.220220e-01 6.439576e-01  
## [61] 6.666667e-01 6.901766e-01 7.145156e-01 7.397130e-01 7.657989e-01  
## [66] 7.928047e-01 8.207629e-01 8.497071e-01 8.796719e-01 9.106935e-01  
## [71] 9.428090e-01 9.760571e-01 1.010478e+00 1.046112e+00 1.083003e+00  
## [76] 1.121195e+00 1.160734e+00 1.201667e+00 1.244044e+00 1.287915e+00  
## [81] 1.333333e+00 1.380353e+00 1.429031e+00 1.479426e+00 1.531598e+00  
## [86] 1.585609e+00 1.641526e+00 1.699414e+00 1.759344e+00 1.821387e+00  
## [91] 1.885618e+00 1.952114e+00 2.020955e+00 2.092224e+00 2.166006e+00  
## [96] 2.242390e+00 2.321468e+00 2.403335e+00 2.488088e+00 2.575830e+00  
## [101] 2.666667e+00 2.760706e+00 2.858063e+00 2.958852e+00 3.063196e+00  
## [106] 3.171219e+00 3.283052e+00 3.398828e+00 3.518688e+00 3.642774e+00  
## [111] 3.771236e+00 3.904229e+00 4.041911e+00 4.184449e+00 4.332013e+00  
## [116] 4.484781e+00 4.642936e+00 4.806669e+00 4.976176e+00 5.151660e+00  
## [121] 5.333333e+00 5.521413e+00 5.716125e+00 5.917704e+00 6.126391e+00  
## [126] 6.342438e+00 6.566104e+00 6.797657e+00 7.037376e+00 7.285548e+00  
## [131] 7.542472e+00 7.808457e+00 8.083822e+00 8.368897e+00 8.664026e+00  
## [136] 8.969562e+00 9.285873e+00 9.613338e+00 9.952352e+00 1.030332e+01  
## [141] 1.066667e+01 1.104283e+01 1.143225e+01 1.183541e+01 1.225278e+01  
## [146] 1.268488e+01 1.313221e+01 1.359531e+01 1.407475e+01 1.457110e+01  
## [151] 1.508494e+01 1.561691e+01 1.616764e+01 1.673779e+01 1.732805e+01  
## [156] 1.793912e+01 1.857175e+01 1.922668e+01 1.990470e+01 2.060664e+01  
## [161] 2.133333e+01 2.208565e+01 2.286450e+01 2.367082e+01 2.450556e+01  
## [166] 2.536975e+01 2.626441e+01 2.719063e+01 2.814950e+01 2.914219e+01  
## [171] 3.016989e+01 3.123383e+01 3.233529e+01 3.347559e+01 3.465610e+01  
## [176] 3.587825e+01 3.714349e+01 3.845335e+01 3.980941e+01 4.121328e+01  
## [181] 4.266667e+01 4.417130e+01 4.572900e+01 4.734163e+01 4.901113e+01  
## [186] 5.073950e+01 5.252883e+01 5.438125e+01 5.629900e+01 5.828438e+01  
## [191] 6.033978e+01 6.246766e+01 6.467057e+01 6.695118e+01 6.931220e+01  
## [196] 7.175649e+01 7.428698e+01 7.690671e+01 7.961882e+01 8.242657e+01  
## [201] 8.533333e+01 8.834261e+01 9.145800e+01 9.468326e+01 9.802226e+01  
## [206] 1.014790e+02 1.050577e+02 1.087625e+02 1.125980e+02 1.165688e+02  
## [211] 1.206796e+02 1.249353e+02 1.293411e+02 1.339024e+02 1.386244e+02  
## [216] 1.435130e+02 1.485740e+02 1.538134e+02 1.592376e+02 1.648531e+02  
## [221] 1.706667e+02 1.766852e+02 1.829160e+02 1.893665e+02 1.960445e+02  
## [226] 2.029580e+02 2.101153e+02 2.175250e+02 2.251960e+02 2.331375e+02  
## [231] 2.413591e+02 2.498706e+02 2.586823e+02 2.678047e+02 2.772488e+02  
## [236] 2.870260e+02 2.971479e+02 3.076268e+02 3.184753e+02 3.297063e+02  
## [241] 3.413333e+02 3.533704e+02 3.658320e+02 3.787330e+02 3.920890e+02  
## [246] 4.059160e+02 4.202306e+02 4.350500e+02 4.503920e+02 4.662751e+02  
## [251] 4.827182e+02 4.997413e+02 5.173646e+02 5.356094e+02 5.544976e+02  
## [256] 5.740520e+02 5.942959e+02 6.152536e+02 6.369505e+02 6.594125e+02  
## [261] 6.826667e+02 7.067409e+02 7.316640e+02 7.574661e+02 7.841781e+02  
## [266] 8.118321e+02 8.404613e+02 8.701001e+02 9.007841e+02 9.325501e+02  
## [271] 9.654365e+02 9.994825e+02 1.034729e+03 1.071219e+03 1.108995e+03  
## [276] 1.148104e+03 1.188592e+03 1.230507e+03 1.273901e+03 1.318825e+03  
## [281] 1.365333e+03 1.413482e+03

plot.ts(SPH.w$Power[71,])

