Energies_R-code_6-30-19.R

rcruicks

Sun Jun 30 07:50:23 2019

```
# Robert F Cruickshank et al., Energies: R-Code to quantify & simulate RBSA TCL Appliances
# Paper: Quantifying the Opportunity Limits of Automatic Residential Electric Load Shaping
# Intent: Compare stochastic vs. spectral analysis fits & simulations w/ AR, ARIMA, & WARM
# Code credits: R. Balaji, C. Torrence, and G. Compo.  See fig numbers from paper at right
# App. A RBSA Fields by category: Appliances, HVAC, Lighting, Plug, Electric, Other, Temp

rm(list=ls()); library(fBasics); library(stats); library(sm)  # Clear vars; load libraries
```

```
## Loading required package: timeDate

## Loading required package: timeSeries

## Package 'sm', version 2.2-5.4: type help(sm) for summary information
```

```
# fbasics::rowStats, sm::sm.density, stats::skewness.  Beware, some stats calc differently
# e.g., in 'moments' package, Pearson 4th moment different than default timeDate::kurtosis
setwd("~/Documents/Energies")                                # Working Dir, Output in ./Figures
# Load RBSA DHW data, https://neea.org/img/documents/rbsam_y1_part-1-of-4.zip...   9 files
# Unzip files. Each has disimilar fields, so: 1) cut & check fields, and then 2) cat files
# e.g., house,time,DHW_1,DHW_2: bash$ cut -f1,2,41,207 RBSAM_Y1_PART\ 1\ OF\ 4.TXT > 2Q12
# Repeat above for 9 files then cat 2Q12 3Q12 4Q12 1Q13 2Q13 3Q13 4Q13 1Q14 2Q14 > All_TCL
# All_TCL=read.table(file='~/Downloads/RBSA_web_download/All_TCL',sep='\t')# Combined file
# colnames(All_TCL)=c('siteid','time','DHW_1','DHW_2')                          # Set colnames
# All_TCL[,2]=as.POSIXct(strptime(All_TCL[,2],'%d %b %y:%H:%M:%S')) # Format readable time
# All=All_TCL[!is.na(All_TCL$time),]                          # Remove rows missing time entries
# All_TCL=All_TCL[with(All_TCL,order(siteid, DHW_1)),]  # Sort ascending order: site, time
# range(RAll_TCL15$time);length(unique(All_TCL$time))# 19 months, 81,798 unique timestamps
# length(unique(All_TCL$siteid))   103 Unique houses; some houses more samples than others
# save('~/Documents/Energies/All_TCL')                                # Save for fast retrieval
load('~/Documents/Energies/All_TCL')                                         # Load data 98.4 MB

# Select RBSA appliances that show diversity in time and energy use per interval
png(file='./Figures/Fig1.png',width=2070,height=1100,pointsize=44)                     # Fig1
par(mfrow=c(3,2),mar=c(3,3.4,1.1,0.6))                       #  3 rows, 2 columns, adjusted margins
X=ALL_TCL[which(ALL_TCL$siteid==24684 & ALL_TCL$time>="2012-07-03 00:00:00" &     # 1 house
                   ALL_TCL$time<"2012-07-05 00:00:00"),"Refrig"]                    # 2 days
barplot(X,border=NA,col=0,space=0); mtext('kWh/15-min',2,line=2,cex=.7)         # Blank plot
axis(1,at=seq(1,192,4),labels=NA)                                                # Hourly ticks
grid(nx=NA,ny=NULL,lty=1); abline(v=seq(1,193,12),col='lightgray')         # Background grid
axis(1,at=seq(1,193,12),labels=seq(0,48,3),las=3); abline(v=97,lty=2,lwd=5);  # Split days
par(new=T);barplot(X,border=NA,col=1,space=0)                                    # Colored bars
title('(a) Refrigerator 24684',adj=0)                                        # Appliance and house

X=ALL_TCL[which(ALL_TCL$siteid==24203 & ALL_TCL$time>="2012-07-03 00:00:00" &     # 1 house
                   ALL_TCL$time<"2012-07-05 00:00:00"),"Refrig"]                    # 2 days
barplot(X,border=NA,col=0,space=0); mtext('kWh/15-min',2,line=2,cex=.7)         # Blank plot
axis(1,at=seq(1,192,4),labels=NA)                                                # Hourly ticks
grid(nx=NA,ny=NULL,lty=1); abline(v=seq(1,193,12),col='lightgray')         # Background grid
axis(1,at=seq(1,193,12),labels=seq(0,48,3),las=3); abline(v=97,lty=2,lwd=5);  # Split days
```

```
par(new=T);barplot(X,border=NA,col=1,space=0)                              # Colored bars
title('Refrigerator 24203',adj=0);grid(nx=NA,ny=NULL,lty=1)

X=ALL_TCL[which(ALL_TCL$siteid==24203 & ALL_TCL$time>="2012-07-03 00:00:00" &     # 1 house
                         ALL_TCL$time<"2012-07-05 00:00:00"),"Freezer"]            # 2 days
barplot(X,border=NA,col=0,space=0); mtext('kWh/15-min',2,line=2,cex=.7)       # Blank plot
axis(1,at=seq(1,192,4),labels=NA)                                           # Hourly ticks
grid(nx=NA,ny=NULL,lty=1); abline(v=seq(1,193,12),col='lightgray')       # Background grid
axis(1,at=seq(1,193,12),labels=seq(0,48,3),las=3); abline(v=97,lty=2,lwd=5);  # Split days
par(new=T);barplot(X,border=NA,col=4,space=0)                              # Colored bars
title('(b) Freezer 24203',adj=0)

X=ALL_TCL[which(ALL_TCL$siteid==23267 & ALL_TCL$time>="2012-07-03 00:00:00" &     # 1 house
                         ALL_TCL$time<"2012-07-05 00:00:00"),"Freezer"]            # 2 days
barplot(X,border=NA,col=0,space=0); mtext('kWh/15-min',2,line=2,cex=.7)       # Blank plot
axis(1,at=seq(1,192,4),labels=NA)                                           # Hourly ticks
grid(nx=NA,ny=NULL,lty=1); abline(v=seq(1,193,12),col='lightgray')       # Background grid
axis(1,at=seq(1,193,12),labels=seq(0,48,3),las=3); abline(v=97,lty=2,lwd=5);  # Split days
par(new=T);barplot(X,border=NA,col=4,space=0)                              # Colored bars
title('Freezer 23267',adj=0)

X=ALL_TCL[which(ALL_TCL$siteid==24684 & ALL_TCL$time>="2012-07-03 00:00:00" &     # 1 house
                         ALL_TCL$time<"2012-07-05 00:00:00"),"DHW_1"]              # 2 days
barplot(X,border=NA,col=0,space=0); mtext('kWh/15-min',2,line=2,cex=.7)       # Blank plot
axis(1,at=seq(1,192,4),labels=NA)                                           # Hourly ticks
grid(nx=NA,ny=NULL,lty=1); abline(v=seq(1,193,12),col='lightgray')       # Background grid
axis(1,at=seq(1,193,12),labels=seq(0,48,3),las=3); abline(v=97,lty=2,lwd=5);  # Split days
par(new=T);barplot(X,border=NA,col=2,space=0)                              # Colored bars
title('(c) DHW 24684',adj=0) ;mtext('Hour',1,cex=.7,line=2)

X=ALL_TCL[which(ALL_TCL$siteid==24495 & ALL_TCL$time>="2012-07-03 00:00:00" &     # 1 house
                         ALL_TCL$time<"2012-07-05 00:00:00"),"DHW_1"]              # 2 days
barplot(X,border=NA,col=0,space=0); mtext('kWh/15-min',2,line=2,cex=.7)       # Blank plot
axis(1,at=seq(1,192,4),labels=NA)                                           # Hourly ticks
grid(nx=NA,ny=NULL,lty=1); abline(v=seq(1,193,12),col='lightgray')       # Background grid
axis(1,at=seq(1,193,12),labels=seq(0,48,3),las=3); abline(v=97,lty=2,lwd=5);  # Split days
par(new=T);barplot(X,border=NA,col=2,space=0)                              # Colored bars
title('DHW 24495',adj=0); mtext('Hour',1,cex=.7,line=2)
dev.off()
```

```
## pdf
##   2
```

```
# WARM Wavelet decomposition.   Fig2 must be annotated with blue text and arrows --> Fig2a
png(file='Figures/Fig2.png',width=2070,height=1650,pointsize=44)                   # Fig2
par(mfrow=c(4,2),mar=c(4,2.4,1.1,3.6))   #  3 rows, 2 columns, margin leaves gap in middle
comp1=1.2*sin(seq(0,4*pi,length.out=250))                       # Illustrative 1st component
comp2=sin(seq(0,8*pi,length.out=250))                           # Illustrative 2nd component
noise=runif(250,-1,1); noise2=runif(250,-1,1)                                        # Noise
data=noise+comp1+comp2; sim=noise2+comp1+comp2             # Time series data and simulation
plot(data,type='l',ylim=c(-4,4),xlab=NA,ylab=NA,yaxt='n',main='Time Series Data')    # Plot
axis(2,at=seq(-4,4,2))                                                               # Y-axis
par(mar=c(4,5.4,1.1,0.6))                                         # Margin leaves gap in middle
plot(sim,type='l',ylim=c(-4,4),xlab=NA,ylab=NA,main='Time Series Simulation') # Right plot
```

```
par(mar=c(4,2.4,1.1,3.6))                                    # Margin leaves gap in middle
plot(comp1,type='l',ylim=c(-2,2),xlab=NA,ylab=NA,main='Component 1')          # Left plot
par(mar=c(4,5.4,1.1,0.6))                                    # Margin leaves gap in middle
plot(comp1,type='l',ylim=c(-2,2),xlab=NA,ylab=NA,main='Component 2')         # Right plot

par(mar=c(4,2.4,1.1,3.6))                                    # Margin leaves gap in middle
plot(comp2,type='l',ylim=c(-2,2),xlab=NA,ylab=NA,main='Component 2')          # Left plot
par(mar=c(4,5.4,1.1,0.6))                                    # Margin leaves gap in middle
plot(comp2,type='l',ylim=c(-2,2),xlab=NA,ylab=NA,main='Component 2')         # Right plot

par(mar=c(4,2.4,1.1,3.6))                                    # Margin leaves gap in middle
plot(noise,type='l',ylim=c(-2,2),xlab=NA,ylab=NA,main='Noise')                # Left plot
par(mar=c(4,5.4,1.1,0.6))                                    # Margin leaves gap in middle
plot(noise2,type='l',ylim=c(-2,2),xlab=NA,ylab=NA,main='Noise')              # Right plot
dev.off()
```

```
## pdf
##   2
```

```
# Annual energy use of all RBSA TCL appliances
Stats=as.data.frame(matrix(rev(unique(ALL_TCL$siteid)), # Setup for stats per unique house
                           nrow=length(rev(unique(ALL_TCL$siteid))),ncol=1))
colnames(Stats)=c('siteid')                                                   # 1st column
Stats$DHW_1=NA; Stats$DHW_2=NA; Stats$Freezer=NA    # Init for loop processing; note order
Stats$Refrig=NA; Stats$Refrig_2=NA; Stats$Freezer_2=NA
for (i in rev(unique(ALL_TCL$siteid))) {                                # Summary by siteid
  for (j in 3:dim(ALL_TCL)[2]) {                                # Summary by TCL appliance
    if (sum(!is.na(ALL_TCL[ALL_TCL$siteid==i,j]))>0){           # Check observations exist
      if (sum(ALL_TCL[ALL_TCL$siteid==i,j],na.rm=T)>0){ # Check for non-zero onbservations
        Stats[which(Stats$siteid==i),j-1]<-mean(ALL_TCL[ALL_TCL$siteid==i,j],na.rm=T)#mean
} } } }                              # j-1 because dropped siteid; close loops and if tests

Stats[,-1]=8760*4*Stats[,-1]# Multiply up to kWh/yr; no sum because missing obs. Drop site
png(file='Figures/Fig3.png',width=2070,height=1100,pointsize=44); par(mfrow=c(1,6)) # Fig3
boxplot(Stats$DHW_1,main='DHW_1',ylim=range(Stats$DHW_1,Stats$DHW_2,na.rm=T),lwd=4, # Plot
        xlab=paste0(length(Stats$DHW_1[!is.na(Stats$DHW_1)]),' units\n',     # xlab footer
                round(length(ALL_TCL$DHW_1[!is.na(ALL_TCL$DHW_1)])/1e3),' ksamples\n',
                round(100*length(which(ALL_TCL$DHW_1>0))/
                          length(ALL_TCL$DHW_1[!is.na(ALL_TCL$DHW_1)])),'% DC'),
        ylab='kWh/yr',cex.main=1.4,cex.lab=1.4,cex.axis=1.4); grid(nx=NA,ny=NULL,lwd=4)
points(mean(Stats$DHW_1,na.rm=T),col='red',pch=19,cex=2)                             # Mean

boxplot(Stats$DHW_2,main='DHW_2',ylim=range(Stats$DHW_1,Stats$DHW_2,na.rm=T),lwd=4, # Same
        xlab=paste0(length(Stats$DHW_2[!is.na(Stats$DHW_2)]),' units\n',
                  round(length(ALL_TCL$DHW_2[!is.na(ALL_TCL$DHW_2)])/1e3),' ksamples\n',
                  round(100*length(which(ALL_TCL$DHW_2>0))/
                           length(ALL_TCL$DHW_2[!is.na(ALL_TCL$DHW_2)])),'% DC'),
        ylab='kWh/yr', cex.main=1.4,cex.lab=1.4,cex.axis=1.4);grid(nx=NA,ny=NULL,lwd=4)
points(mean(Stats$DHW_2,na.rm=T),col='red',pch=19,cex=2)                             # Mean

boxplot(Stats$Refrig,main='Refrig_1',ylim=range(Stats$Refrig,Stats$Refrig_2,na.rm=T),
        lwd=4,xlab=paste0(length(Stats$Refrig[!is.na(Stats$Refrig)]),' units\n',
                round(length(ALL_TCL$Refrig[!is.na(ALL_TCL$Refrig)])/1e3),' ksamples\n',
                round(100*length(which(ALL_TCL$Refrig>0))/
```

```
                            length(ALL_TCL$Refrig[!is.na(ALL_TCL$Refrig)])),'% DC'),
        ylab='kWh/yr', cex.main=1.4,cex.lab=1.4,cex.axis=1.4);grid(nx=NA,ny=NULL,lwd=4)
points(mean(Stats$Refrig,na.rm=T),col='red',pch=19,cex=2)                        # Mean

boxplot(Stats$Refrig_2,main='Refrig_2',ylim=range(Stats$Refrig,Stats$Refrig_2,na.rm=T),
        lwd=4, xlab=paste0(length(Stats$Refrig_2[!is.na(Stats$Refrig_2)]),' units\n',
              round(length(ALL_TCL$Refrig_2[!is.na(ALL_TCL$Refrig_2)])/1e3),' ksamples\n',
              round(100*length(which(ALL_TCL$Refrig_2>0))/
                        length(ALL_TCL$Refrig_2[!is.na(ALL_TCL$Refrig_2)])),'% DC'),
        ylab='kWh/yr', cex.main=1.4,cex.lab=1.4,cex.axis=1.4);grid(nx=NA,ny=NULL,lwd=4)
points(mean(Stats$Refrig_2,na.rm=T),col='red',pch=19,cex=2)                        # Mean

boxplot(Stats$Freezer,main='Freezer_1',ylim=range(Stats$Freezer,Stats$Freezer_2,na.rm=T)
        ,lwd=4,xlab=paste0(length(Stats$Freezer[!is.na(Stats$Freezer)]),' units\n',
             round(length(ALL_TCL$Freezer[!is.na(ALL_TCL$Freezer)])/1e3),' ksamples\n',
                  round(100*length(which(ALL_TCL$Freezer>0))/
                         length(ALL_TCL$Freezer[!is.na(ALL_TCL$Freezer)])),'% DC'),
        ylab='kWh/yr', cex.main=1.4,cex.lab=1.4,cex.axis=1.4);grid(nx=NA,ny=NULL,lwd=4)
points(mean(Stats$Freezer,na.rm=T),col='red',pch=19,cex=2)                        # Mean

boxplot(Stats$Freezer_2,main='Freezer_2',ylim=range(Stats$Freezer,Stats$Freezer_2,na.rm=T),
        lwd=4,xlab=paste0(length(Stats$Freezer_2[!is.na(Stats$Freezer_2)]),' units\n',
            round(length(ALL_TCL$Freezer_2[!is.na(ALL_TCL$Freezer_2)])/1e3),' ksamples\n',
            round(100*length(which(ALL_TCL$Freezer_2>0))/
                      length(ALL_TCL$Freezer_2[!is.na(ALL_TCL$Freezer_2)])),'% DC'),
        ylab='kWh/yr', cex.main=1.4,cex.lab=1.4,cex.axis=1.4);grid(nx=NA,ny=NULL,lwd=4)
points(mean(Stats$Freezer_2,na.rm=T),col='red',pch=19,cex=2)                      # Mean
dev.off()
```

```
## pdf
##   2
```

```
# Diversity bounds in DHW heater Electricy consumption for small sets of homes, some w/Gas
DHW_Usage=ALL_TCL[,1:3]                                          # Select site, time, DHW_1
DHW_Usage=DHW_Usage[which(DHW_Usage$time=='2012-04-01 00:00:00'),]  # Pick 15-min interval
nsim=1000; nsam=6; distrib=matrix(NA,ncol=nsam,nrow=nsim)  # Loop params to create distrib
for(sim in 1:nsim){             # Outer loop is # of simulations.  Inner Loop is # of homes
  for(sam in 1:nsam) distrib[sim,sam]<-mean(sample(DHW_Usage$DHW_1,2^sam),na.rm=T) #Inloop
}     # This samples Gas & Elect DWH heaters; gas DHW heaters always return zero elect use

png(file='Figures/Fig4.png',width=2070,height=1100,pointsize=44)                   # Fig4
par(mfrow=c(1,1)); par(mar=c(5.1,4.1,1.6,0.4))
boxplot(distrib,ylim=c(0,1.1),ylab='Mean Energy/Home [kWh/15-min]',lwd=4,        # 6 Plots
        xaxt='n',xlab='Homes Sampled')
legend('topright','Mean of all water heaters',col='red',lty=2,lwd=4,bty='n') # Elect & Gas
axis(1,at=1:nsam,labels=c(2,4,8,16,32,64)); grid(nx=NA,ny=NULL,lwd=4)     # Num simul homes
boxplot(distrib,ylim=c(0,1.1),ylab='Mean Energy/Home [kWh/15-min]',lwd=4, # Plot over grid
        xaxt='n',xlab='Homes Sampled',add=T)
abline(h=mean(DHW_Usage$DHW_1,na.rm=T),col='red',lty=2,lwd=4)    # Add mean line dashed red
dev.off()
```

```
## pdf
##   2
```

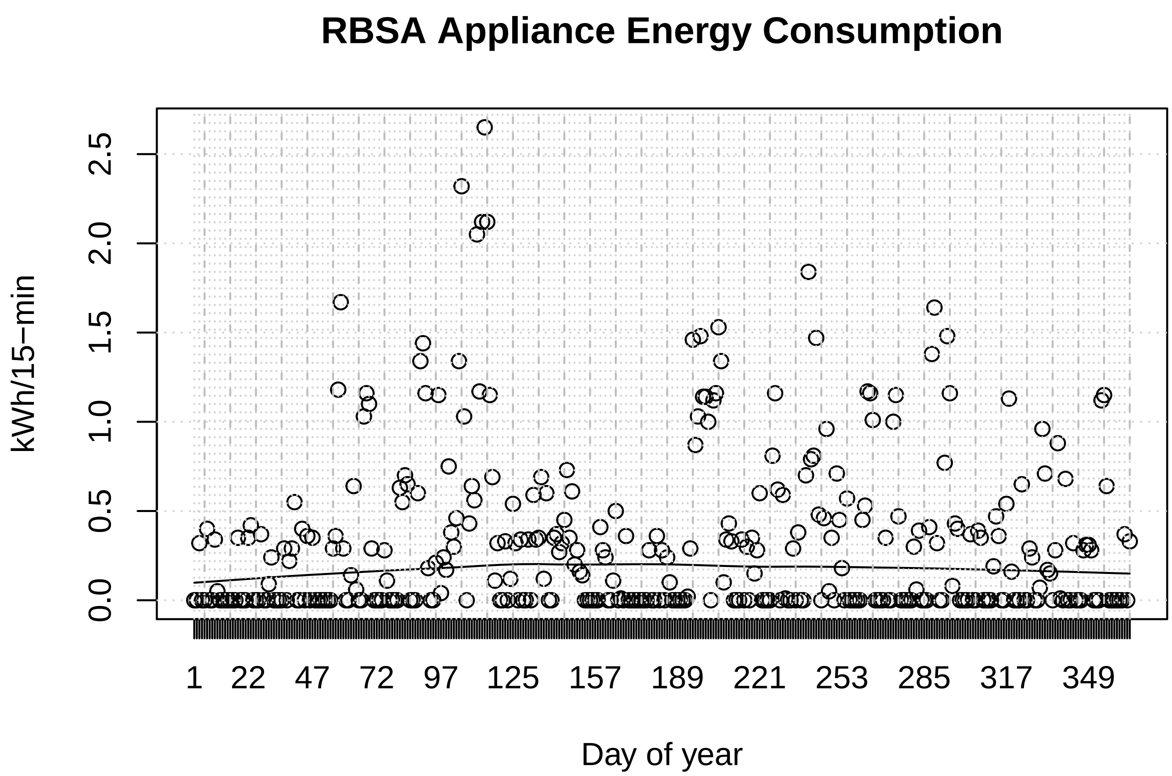
```
# Observed annual energy use of single DHW heater
DHW_1yr=head(ALL_TCL[ALL_TCL$siteid==10388,3],365*96) # Grab 1st yr of data for appliances
tmp=matrix(DHW_1yr,ncol=96,nrow=365)
appliance=matrix(1,nrow=365,ncol=24)
for (i in seq(4,96,4)) appliance[,i/4]=rowSums(tmp[,(i-3):i])      # Sum 15-min obs to 1-hr
title='RBSA Appliance Energy Consumption'
rownames(appliance)=1:(dim(appliance)[1])                                      # 365 days
colnames(appliance)=1:24                                                       # 24 Hours
nyrs=length(appliance[,1])                                               # Set max loop value

png(file='Figures/Fig5.png',width=2070,height=1100,pointsize=44)                    # Fig5
par(mfrow=c(1,6),mar=c(3,4.4,4.1,0.6))
boxplot(rowMeans(appliance),main='Mean',ylab='kWh/15-min',cex.main=1.4,cex.lab=1.4, # Mean
        cex.axis=1.4,lwd=4); grid(nx=NA,ny=NULL,lwd=4)
boxplot(rowVars(appliance),main='Var',ylab='[kWh/15-min]^2',cex.main=1.4,cex.lab=1.4,# Var
        cex.axis=1.4,lwd=4); grid(nx=NA,ny=NULL,lwd=4)
boxplot(rowSkewness(appliance),main='Skew',ylab='Dimensionless',cex.main=1.4,       # Skew
        cex.lab=1.4,cex.axis=1.4,lwd=4); grid(nx=NA,ny=NULL,lwd=4)
boxplot(rowKurtosis(appliance),main='Kurt',ylab='Dimensionless',cex.main=1.4,       # Kurt
        cex.lab=1.4,cex.axis=1.4,lwd=4); grid(nx=NA,ny=NULL,lwd=4)
boxplot(rowMaxs(appliance),main='Max',ylab='kWh/15-min',cex.main=1.4,cex.lab=1.4,    # Max
        cex.axis=1.4,lwd=4); grid(nx=NA,ny=NULL,lwd=4)
boxplot(rowSums(appliance),main='Sum',ylab='kWh/Day',cex.main=1.4,cex.lab=1.4,       # Sum
        cex.axis=1.4,lwd=4); grid(nx=NA,ny=NULL,lwd=4)
dev.off()
```

```
## pdf
##   2
```

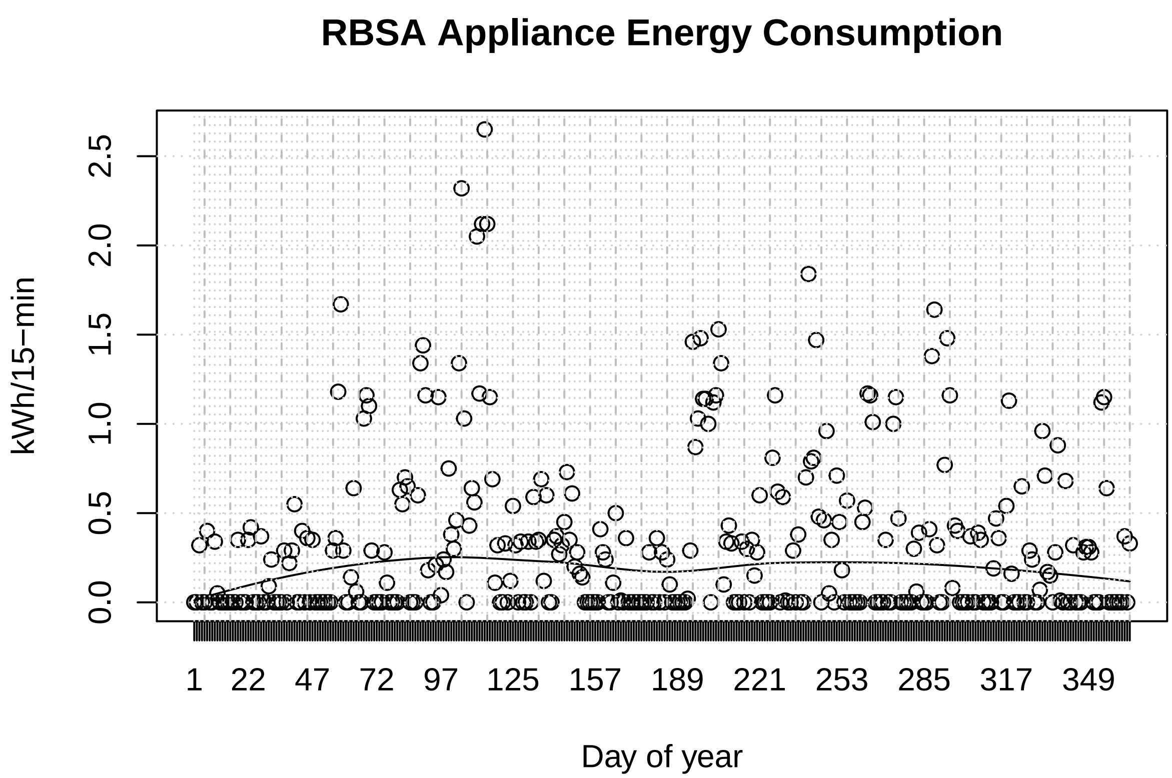
```
# Annual distribution of use by hour of day
png(file='Figures/Fig6.png',width=2070,height=1100,pointsize=44)                    # Fig6
par(mfrow=c(1,1),mar=c(5.1,4.2,1.6,0.5))
boxplot(appliance,ylab='kWh/hr',xaxt='n',xlab='Hour of Day',lwd=4)    # 24 hourly boxplots
axis(1,at=1:24,labels=colnames(appliance),las=3)         # las=3 rotates labels vertically
grid(nx=NA,ny=NULL,lwd=4)
boxplot(appliance,ylab='kWh/hr',xaxt='n',xlab='Hour of Day',lwd=4,add=T)  # Plot over grid
lines(colMeans(appliance),col='red',lwd=4)    # Mean is 1st moment, a non-resistant measure
legend('topleft','Mean',col='red',lty=1,bty='n',lwd=4)
dev.off()                                   # sth moment = (x1^s + x2^s + . . . + xn^s)/n
```

```
## pdf
##   2
```

```
# Hour 5 Analysis of 1 DHW heater starting with traditional curve fits (figs not in paper)
X=appliance[,5]; unit='kWh/15-min'                                           # Hr 5 usage
scatter.smooth(X,xlab='Day of year',ylab=unit,xaxt='n',main=title)# Not satisfying; no fit
axis(1,at=1:365,labels=rownames(appliance)); grid(nx=NA,ny=NULL)
abline(v=seq(1,365,2),lty=3,col='lightgray'); abline(v=seq(5,365,10),lty=2,col='gray')
```




```
scatter.smooth(X,xlab='Day of year',ylab=unit,degree=2,xaxt='n',main=title)      # Bimodal?
axis(1,at=1:365,labels=rownames(appliance)); grid(nx=NA,ny=NULL)
abline(v=seq(1,365,2),lty=3,col='lightgray');abline(v=seq(5,365,10),lty=2,col='gray')
```



```r
# Start Wavelet Analysis. A few extra figs & params not in paper included for background
source("./wavelet_function_r.txt")                              # Torrence & Compo's wavelet code
DT=1                                                             # DT is timestep for annual data
DJ=0.25                         # Spacing between discrete scales; smallest Scale S0 = 2*DT
pad=1                                       # Pads data with zeros at begin and end of time
zz=wavelet(X,DT,pad,DJ,6)                                               # 6 is Morlet wavelet
str(zz)                                     # Wave, Period, Scale, Power, Cone of Influence
```

```
## List of 5
##  $ wave  : cplx [1:31, 1:365] 0.0116+0.0644i -0.0172+0.1355i -0.1364+0.1786i ...
##  $ period: num [1:31] 2.07 2.46 2.92 3.47 4.13 ...
##  $ scale : num [1:31] 2 2.38 2.83 3.36 4 ...
##  $ power : num [1:31, 1:365] 0.00429 0.01866 0.05049 0.05034 0.0158 ...
##  $ coi   : num [1:365] 0.73 1.46 2.19 2.92 3.65 ...
```

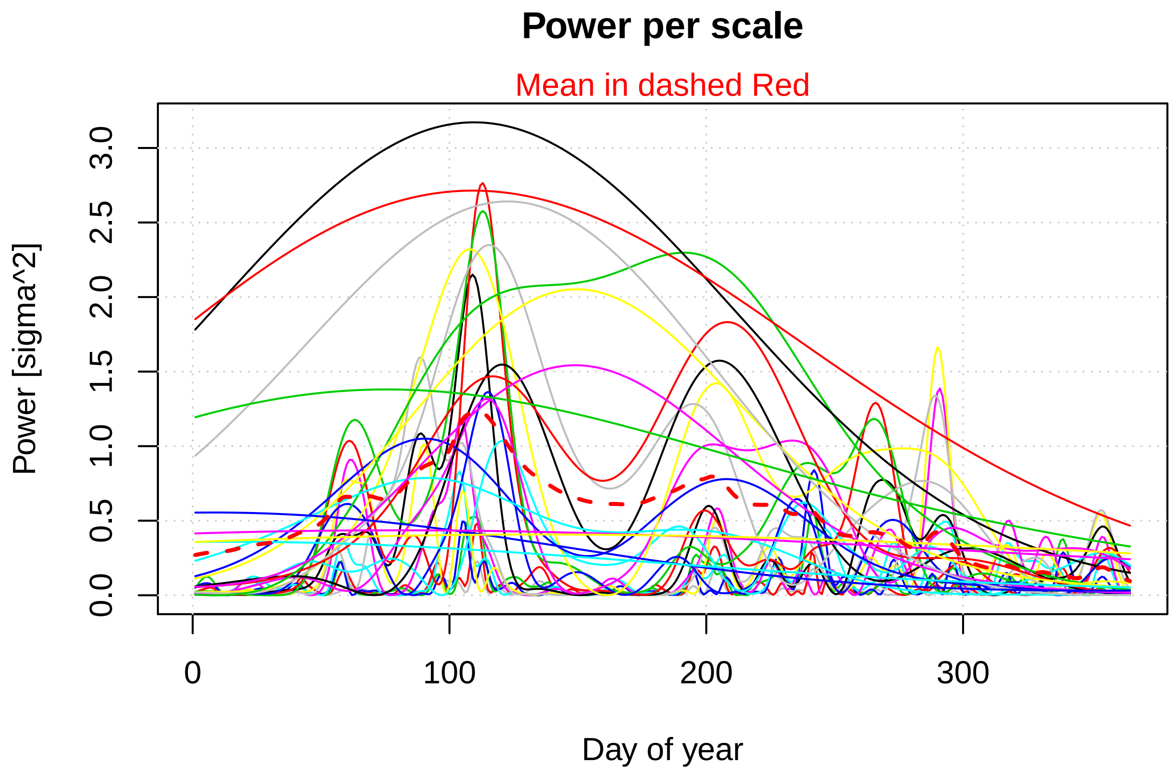
```r
names(zz)
```

```
## [1] "wave"   "period" "scale"  "power"  "coi"
```

```r
lapply(names(zz),function(x) {range(Re(zz[[x]]))})
```

```
## [[1]]
## [1] -1.643492  1.776421
##
## [[2]]
## [1]   2.066087 374.001750
##
## [[3]]
## [1]   2.0000 362.0387
##
## [[4]]
## [1] 1.452618e-05 3.171832e+00
##
## [[5]]
## [1]   0.7304722 133.6764069
```

```r
plot(zz$power[1,],type='l',col='gray',ylim=range(zz$power),ylab='Power [sigma^2]',
     xlab='Day of year',main='Power per scale'); grid()
for (i in 2:length(zz$power[,1])) lines(zz$power[i,],col=i)            # 1 trace per scale
lines(colMeans(zz$power),col='red',lwd=2,lty=2); mtext('Mean in dashed Red',col='red')
```




```
ns=length(zz$scale)                          # Number of (J+1) Scales; J=(log2(N*DT/S0))/DJ
z2=matrix(0,nrow=ns,ncol=nyrs)                                                 # Initialize
for(i in 1:ns){
  z2[i,]=(sqrt(DT)*DJ)*Re(zz$wave[i,])/((sqrt(zz$scale[i]))*.776*(pi^(-1/4)))    # Source?
}
Xrecon=apply(z2,2,sum) + mean(X)                          # Sum columns and add mean of obs
png(file='Figures/Fig7.png',width=2100,height=1100,pointsize=44)                      # Fig7
par(mar=c(5.1,4.1,1.6,0.4))
plot(X[1:182],type='n',xlab='5 AM hour of Day',ylab='kWh/hr',xaxt='n',lwd=4)  # Blank Plot
axis(1,at=seq(0,182,7),labels=NA)
axis(1,at=seq(0,182,14),labels=seq(0,182,14)); grid(nx=NA,ny=NULL,lwd=4)    # Axis and grid
par(new=T);plot(X[1:182],type='h',xlab='5 AM hour of Day',ylab='kWh/hr',xaxt='n',lwd=4)
points(Xrecon[1:182],col='red',cex=.5,pch=19)              # Overlay of Xrecon shows good fit
legend('topright',c('Observed','Wavelet reconstructed'), pch=c('|','•'),col=c('black','red'),
       bty='n')
# Xrecon & X should be similar and show that the wavelet decomposition adds up to original
dev.off()
```

```
## pdf
##   2
```

```
# House Add/Shed calc. As alternative to flat file, shows RBSA data coming from SQLite DB
library(DBI)                                             # Communicate between R & SQLite
library(RSQLite)                  # SQLite driver implementing the R/S-Plus database (DBI) API
library(lubridate)                                                      # Date/Time handling
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
```

```
##     date
```

```
Yr1_Interval_15_min_db = dbConnect(SQLite(), dbname="RBSA_METER_DATA_1.sqlite")    # 2Q12-
Yr1_sites <- dbGetQuery(Yr1_Interval_15_min_db,"select siteid from RBSA_METER_DATA")# 1Q13
Yr1_sites=unique(Yr1_sites); Yr1_sites=Yr1_sites[,1]              # List of unique sites
site=13019                                                          # Have a look at 1 house
sqlcmd <- paste("select * from RBSA_METER_DATA where siteid =", site)# All data on 1 house
Yr1_Usage <- data.frame(dbGetQuery(Yr1_Interval_15_min_db,sqlcmd))    # 35037 of 35040 obs
a=Yr1_Usage[1,]; b=is.na(a); b1=which(b==FALSE); Yr1_Usage=Yr1_Usage[,b1]  # 247-->69 vars
Yr1_Intervals=dmy_hms(Yr1_Usage[,2], tz="PST8PDT")                    # POSIXct time format
```

```
## Warning: 1 failed to parse.
```

```
Yr1_Service=Yr1_Usage[c(grep("Service",colnames(Yr1_Usage)))] # Energy use at svc entrance
Yr1_TCL=Yr1_Usage[c(grep("Refrig|Freezer|DHW_1|DHW_2",colnames(Yr1_Usage)))]   # Grep TCLs
Yr1_Usage=cbind(Yr1_Service,Yr1_TCL)           # Service and varaible # of TCLs per house
maxvals=lapply(Yr1_Usage, max)                 # Maximum value observed per TCL over time
Yr1_Usage$Increase=0; Yr1_Usage$Decrease=0                            # Initialize for loop
for (i in colnames(Yr1_TCL)){                    # Add/Shed cals (aka increase/decrease)
#  print(maxvals[i])            # Verbose prints to checks increase/decrease per timestep
  for (j in 1:24){
#    print(Yr1_Usage[j,i])
    if(Yr1_Usage[j,i]== 0) {Yr1_Usage$Increase[j]=Yr1_Usage$Increase[j]+maxvals[[i]]
#      print("Increase"); print(Yr1_Usage$Increase[j])
    } else {Yr1_Usage$Decrease[j]=Yr1_Usage$Decrease[j]+Yr1_Usage[j,i]
#      print("Decrease"); print(Yr1_Usage$Decrease[j])
} } }                                                                          # End loop
Yr1_Usage$Increase=Yr1_Usage$Service+Yr1_Usage$Increase            # Every 15-min for year
Yr1_Usage$Decrease=Yr1_Usage$Service-Yr1_Usage$Decrease
# print(head(Yr1_Usage,1)) ; print(tail(Yr1_Usage,1))                         # Have a look

png(file='Figures/Fig8.png',width=2070,height=1100,pointsize=44)                    # Fig8
par(mfrow=c(1,1),mar=c(5.1,4.1,1.6,0.4))
ts.plot(Yr1_Usage[1:24,c(1,5,6)],gpars=list(col=c("black","red","green"),xaxt='n',   # Svc
                                    yaxt='n', xlab="",lwd=4,ylim=c(0,.3)))
lines(Yr1_Usage[1:24,2],col='blue',lwd=4,lty=2)                                 # Freezer
lines(Yr1_Usage[1:24,3],col='cyan',lwd=4,lty=2)                                  # Refrig
lines(Yr1_Usage[1:24,4],col='magenta',lwd=4,lty=2)                             # Refrig_2
axis(2,cex.axis=1)
title(xlab="First 6 hours of 1 April 2012 at 15-min intervals", ylab="kWh/15-min", # Title
      cex.lab=1, cex.main=2)
axis(1,Yr1_Intervals, at=seq(1,24,4), lwd.tick=2, label=seq(0,5,1), las=3, cex.axis=1)
axis(1,Yr1_Intervals, at=seq(1,24,1), lwd.tick=.5, label=FALSE) # 15-min ticks, 1hr labels
grid(nx=NA,ny=NULL,lwd=4);abline(v=seq(1,24,4),lty=3,lwd=4,col='lightgray')         # Grid
legend("topright", legend = colnames(Yr1_Usage)[c(5,1,6,4,3,2)],     # Plot groups in order
       title = 'House            Appliances',
       col=c("red","black","green","magenta","cyan","blue"),
       pch = 19, bty="n", cex=1, inset=c(.025,-0.01), ncol=2, x.intersp=1.5, y.intersp=.9)
dev.off()
```

```
## pdf
##   2
```

```
# Aggregate Add/Shed across 14 homes
Yr1_Service_Sum=0; Yr1_Increase_Sum=0; Yr1_Decrease_Sum=0                           # Init
for (site in c(12975,13222,14102,14153,14273,14545,20215,20469,20775,20998,21355,21799,
```

```
              22138,24203)){                     # These 14 houses chosen for visual clarity
  # print(site)
  sqlcmd <- paste("select * from RBSA_METER_DATA where siteid =", site)   # Get house data
  Yr1_Usage <- data.frame(dbGetQuery(Yr1_Interval_15_min_db,sqlcmd))       # Same as above
  a=Yr1_Usage[1,]; b=is.na(a); b1=which(b==FALSE); Yr1_Usage=Yr1_Usage[,b1]# Same as above
  Yr1_Intervals=dmy_hms(Yr1_Usage[,2], tz="PST8PDT")                        # Same as above
  Yr1_Service=Yr1_Usage[c(grep("Service",colnames(Yr1_Usage)))]            # Same as above
  Yr1_TCL=Yr1_Usage[c(grep("Refrig|Freezer|DHW_1|DHW_2",colnames(Yr1_Usage)))]# Same above
  Yr1_Usage=cbind(Yr1_Service,Yr1_TCL)                                      # Same as above
  Yr1_Usage=head(Yr1_Usage,n=96)                                 # Trim to first 96 readings
  maxvals=lapply(Yr1_Usage, max)                                  # Find max looking forward

  Yr1_Usage$Increase=0
  Yr1_Usage$Decrease=0
  for (i in colnames(Yr1_TCL)){
    # print(maxvals[i])
    for (j in 1:96){
      #print(Yr1_Usage[j,i])
      if(Yr1_Usage[j,i]== 0) {Yr1_Usage$Increase[j]=Yr1_Usage$Increase[j]+maxvals[[i]];
      # print("Increase"); print(Yr1_Usage$Increase[j])
      } else {Yr1_Usage$Decrease[j]=Yr1_Usage$Decrease[j]+Yr1_Usage[j,i]
      #print("Decrease"); print(Yr1_Usage$Decrease[j])
  } } }                                                                          # End loop

  Yr1_Service_Sum=Yr1_Service_Sum+Yr1_Service # Add multpile houses
  Yr1_Increase_Sum=Yr1_Service_Sum+(Yr1_Increase_Sum+Yr1_Usage$Increase)
  Yr1_Decrease_Sum=Yr1_Service_Sum-(Yr1_Decrease_Sum+Yr1_Usage$Decrease)
  Yr1_Sum=cbind(Yr1_Increase_Sum,Yr1_Service_Sum,Yr1_Decrease_Sum)
  colnames(Yr1_Sum)=c("Increase","Service","Decrease")
  Yr1_Usage$Increase=Yr1_Usage$Service+Yr1_Usage$Increase
  Yr1_Usage$Decrease=Yr1_Usage$Service-Yr1_Usage$Decrease
  # print(head(Yr1_Usage,1)); print(tail(Yr1_Usage,1))          # View add/shed per house
  # ts.plot(Yr1_Sum[1:96,],gpars=list(col=c("red","black","green"),xaxt='n',yaxt='n',
  # xlab="", lwd=3))
  # axis(2, cex.axis=1.5)
  # title(xlab="1 April 2012 at 15-min intervals", ylab="kWh/15-minutes",
  #       main="Sum of 12975,13222,14102,14153,14273,14545,20215,20469,20775,20998,21355,
  #             21799,22138,24203", cex.lab=1.5, cex.main=1.5)
  # axis(1,Yr1_Intervals,at=seq(1,96,4),lwd.tick=2,label=seq(0,23,1),las=3,cex.axis=1.5)
  # axis(1,Yr1_Intervals,at=seq(1,96,1),lwd.tick=.5,label=FALSE)
  # legend("topleft",legend=colnames(Yr1_Sum),col=c("red","black","green"),pch = 1,bty="n",
  #        cex=1.4, inset=c(.025,-0.01), ncol=1, x.intersp=1)
  # cat ("Press [enter] to continue")
  # User_input <- readline()
}
```

```
## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.
```

```
## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.

## Warning: 4 failed to parse.
```

```
png(file='Figures/Fig9.png',width=2070,height=1100,pointsize=44)                     # Fig 9
par(mfrow=c(1,1),mar=c(5.1,4.1,1.6,0.4))
ts.plot(Yr1_Sum[1:96,],gpars=list(col=c("red","black","green"),xaxt='n',yaxt='n',xlab="",
                                  lwd=4),type='n')                              # Blank Plot
axis(2, cex.axis=1)
title(xlab="1 April 2012 at 15-min intervals",ylab="kWh/15-min",cex.lab=1,cex.main=1)
axis(1,Yr1_Intervals,at=seq(1,96,4),lwd.tick=2,label=seq(0,23,1),las=3,cex.axis=1.1)  # Hr
axis(1,Yr1_Intervals,at=seq(1,96,1),lwd.tick=.5,label=FALSE)                  # 15-min ticks
grid(nx=NA,ny=NULL,lwd=4);abline(v=seq(1,96,4),lty=3,lwd=4,col='lightgray')          # Grid
par(new=T); ts.plot(Yr1_Sum[1:96,],gpars=list(col=c("red","black","green"),xaxt='n',
                                              yaxt='n',xlab="",lwd=4))    # Plot over grid
legend("topright",legend=colnames(Yr1_Sum),col=c("red","black","green"),pch = 19,bty="n",
       cex=1,inset=c(.025,-0.01),ncol=1,x.intersp=1)                                # Legend
dev.off()
```

```
## pdf
##   2
```

```
# Wavelet plot of DHW_1 in 1 house. Modified from R Balaji's code
options("max.contour.segments"= 3000000)
# Above is 1D Wavelet transform with optional significance testing [WAVE,PERIOD,SCALE,COI]
data=matrix(scan("22284_DHW.txt"), ncol=2, byrow=T)  # Data in 2 cols: year 1st & vals 2nd
lvl=.95; lvl2=.9                                             # 95% and 90% Conf Intervals
XL=as.vector(data[,2])                                                             # Vals
fin=wavelet(data[,2], .25, 1, .25, 6)                          # fin = result of wavelet
source("ken_sig.txt")                                            # Significance function
zz=noise(lvl2, XL,"r",  fin, length(data[,2]))   # r for red noise background, w for white
```

```
##       ar1
## 0.4968443
```

```
xx=noise(lvl,XL, "r",  fin, length(data[,2]))
```

```
##       ar1
## 0.4968443
```

```
source("wave_plot_8-10-18.R")                      # Fig10 created herein; add arrow to Fig10a

# Hourly Usage Nonparametric Density Estimation
# Simulation via phase randomization... Easy & straightforward.  Modified R Balaji's code
X=appliance[,5]                                                                  # Hr 5 usage
zz=wavelet(X,DT,pad,DJ,6)                                              # 6 is Morlet wavelet
wavphas=zz$wave
for(i in 1:ns) wavphas[i,]= Arg(zz$wave[i,])           # Arg is angle in polar coordinates
png(file='Figures/Fig11.png',width=2100,height=1100,pointsize=44)                  # Fig11
par(mar=c(5.1,4.1,1.6,0.4))
sm.density(X,ylim=c(0,1.5),xlab='kWh',type='b')                              # Obs density
sm.density(Xrecon,add=T,col='red')                         # Wavelet reconstructed denisty
index=1:nyrs; nsim=25                                     # Simulation length & iterations
xmaysim=matrix(0,nrow=nsim,ncol=nyrs)                                         # Initialize
tmpdy=matrix(NA,nsim,86)                           # Initialize storage of sm.denisty output
for(isim in 1:nsim){                                                       # Simulation loop
  index1=sample(index)                                         # Sample all w/o replacement
    z2=matrix(0,nrow=ns,ncol=nyrs)                                              # Initialize
    for(i in 1:ns){                                                # ns = number of scales
    z2[i,]=(sqrt(DT)*DJ)*Re(Mod(zz$wave[i,])*cos(wavphas[i,index1]))/((sqrt(zz$scale[i]))*
                                                                           .776*(pi^(-1/4)))
  }
    xmaysim[isim,]=apply(z2,2,sum) + mean(X)              # Sum columns and add mean of obs
    sm.density(xmaysim[isim,],add=T,col="grey")     # Add density trace for each simulation
    tmpdy[isim,]=sm.density(xmaysim[isim,],eval.points=seq(-0.75,3.5,.05),
                            display='none')$estimate    # Add density trace for each simul
}
grid(lwd=4)
sm.density(X,add=T,lwd=4); sm.density(Xrecon,add=T,col='red',lwd=4)        # Plot over grid
legend=c('Empirical PDF','Wavelet reconstructed PDF','Phase randomization simulations')
legend('topright',legend,lwd=4,col=c('black','red','gray'),bty='n',y.intersp=2)
dev.off()
```

```
## pdf
##   2
```

```
## Fit AR to each component and simulate
png(file='Figures/Fig12.png',width=2100,height=1100,pointsize=44)                  # Fig12
par(mar=c(5.1,4.1,1.6,0.4))
sm.density(X,ylim=c(0,1.5),xlab='kWh')                                        # Obs density
tmpary=matrix(NA,nsim,86)                          # Initialize storage of sm.denisty output
armaysim=matrix(0,nrow=nsim,ncol=ns)                                          # Initialize
for(isim in 1:nsim){
  zs=c()
  for(i in 1:ns){
    Y=z2[i,] - mean(z2[i,])
    arbest=ar(Y,order.max=25)
    p=order(arbest$aic)[1]
    zar=ar(Y,order=p)                                                             # AR Fit
    if(isim==1 & i==1) str(zar)
    zs1=arima.sim(n=nyrs,list(ar=c(zar$ar)),sd=sqrt(zar$var.pred)) + mean(z2[i,])
    zs=cbind(zs,zs1)                               # Some traces don't cover full range of usage
  }
  armaysim[isim,]=apply(zs,2,sum) + mean(X)         # Changed from rowsum(1) to colsum (2)
```

```
  sm.density(armaysim[isim,],add=T,col="grey",lwd=4)
  tmpary[isim,]=sm.density(armaysim[isim,],
                         eval.points=seq(-0.75,3.5,.05),display='none')$estimate    # Sim
  }
```

```
## List of 14
##  $ order     : int 0
##  $ ar        : num(0)
##  $ var.pred  : num 0.00148
##  $ x.mean    : num 2.69e-19
##  $ aic       : Named num [1:2] 0 1.74
##   ..- attr(*, "names")= chr [1:2] "0" "1"
##  $ n.used    : int 365
##  $ n.obs     : int 365
##  $ order.max : num 1
##  $ partialacf: num [1, 1, 1] -0.0265
##  $ resid     : num [1:365] -0.00513 -0.00744 -0.02023 0.00844 -0.00552 ...
##  $ method    : chr "Yule-Walker"
##  $ series    : chr "Y"
##  $ frequency : num 1
##  $ call      : language ar(x = Y, order.max = p)
##  - attr(*, "class")= chr "ar"
```

```
grid(lwd=4)
sm.density(X,add=T,lwd=4); sm.density(Xrecon,add=T,col='red',lwd=4)      # Add X and Xrecon
legend=c('Empirical PDF','Wavelet reconstructed PDF','AR simulations')
legend('topright',legend,lwd=4,col=c('black','red','gray'),bty='n',y.intersp=2)
dev.off()
```

```
## pdf
##   2
```

```
## Fit ARIMA to each component and simulate
png(file='Figures/Fig13.png',width=2100,height=1100,pointsize=44)                     # Fig13
par(mar=c(5.1,4.2,1.6,0.5))
sm.density(X,ylim=c(0,1.5),xlab='kWh',lwd=4); grid(lwd=4)
tmparimay=matrix(NA,nsim,86)                          # Initialize storage of sm.denisty output
arimamaysim=matrix(0,nrow=nsim,ncol=ns)
for(isim in 1:nsim){
  zs=c()                                                                                # Init
  for(i in 1:ns){
    Y=z2[i,] - mean(z2[i,])
    arbest=ar(Y,order.max=25)
    p=order(arbest$aic)[1]                                               # Which P to choose?
    p=2
    zar=arima(Y,order=c(p,0,0),include.mean=TRUE,method="ML")            # Max Likelihood fit
    # if(isim==1 & i==1) str(zar)                                            # Print str(zar)
    zs1=arima.sim(n=nyrs,list(ar=c(zar$coef[1:p])),sd=sqrt(zar$sigma2)) + mean(z2[i,])
    zs=cbind(zs,zs1)                                # Some traces don't cover full range of usage
  }
  arimamaysim[isim,]=apply(zs,2,sum) + mean(X)       # Changed from rowsum(1) to colsum (2)
  sm.density(arimamaysim[isim,],add=T,col="grey",lwd=4)     # ARIMA simulation more compute
  tmparimay[isim,]=sm.density(arimamaysim[isim,],eval.points=seq(-0.75,3.5,.05),
                              display='none')$estimate    # Add density trace for each sim
  }
```

```
sm.density(X,add=T,lwd=4); sm.density(Xrecon,add=T,col='red',lwd=4)     # Add X and Xrecon
legend=c('Empirical PDF','Wavelet reconstructed PDF','ARIMA simulations')
legend('topright',legend,lwd=4,col=c('black','red','gray'),bty='n',y.intersp=2)
dev.off()
```

```
## pdf
##   2
```

```
# Compairion of Phase Randomization and ARIMA (AR not inlcuded as graph would be too busy)
png(file='Figures/Fig14.png',width=2100,height=1100,pointsize=44)                  # Fig14
par(mar=c(5.1,4.1,1.6,0.4))
boxplot(tmparimay,ylim=c(0,1.5),xaxt='n',xlab='kWh',col='gray',            # ARIMA boxplots
        ylab='Probability density function',lwd=3)
grid(nx=NA,ny=NULL,lwd=4); abline(v=seq(6,86,10),col='lightgray',lwd=4,lty=3)       # Grid
boxplot(tmparimay,ylim=c(0,1.5),xaxt='n',xlab='kWh',col='gray',     # Plot over grid
        ylab='Probability density function',lwd=3,add=T)
boxplot(tmpdy,ylim=c(0,1.5),xaxt='n',col='red',add=T,lwd=3) # Phase Randomization boxplots
axis(1,at=seq(6,86,10),labels=seq(-0.5,3.5,.5))
lines(unlist(sm.density(X,eval.points=seq(-0.75,3.5,.05),display='none')$estimate),  # Obs
      lwd=4)
legend=c('Empirical PDF','Phase randomization simulations','ARIMA simulations')
legend('topright',legend,lwd=4,col=c('black','red','gray'),bty='n',y.intersp=2)
dev.off()
```

```
## pdf
##   2
```

```
# Annual Statistics of Observed vs. Simulated hourly usage
# Mean and variance distributions
oldpar=par(no.readonly=T);par(mfrow=c(1,6),mar=c(5.1,4.1,4.1,2.1))       # Save par params
png(file='Figures/Fig15.png',width=2100,height=1100,pointsize=44)                 # Fig 15
par(mfrow=c(1,6),mar=c(2.1,4.8,4.5,0.8))
boxplot(rowMeans(xmaysim),main='Mean\nPhase',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(mean(X),rowMeans(xmaysim),rowMeans(armaysim),rowMeans(arimamaysim)))
points(mean(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                   # Mean
boxplot(rowMeans(armaysim),main='Mean\nAR',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(mean(X),rowMeans(xmaysim),rowMeans(armaysim),rowMeans(arimamaysim)))
points(mean(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                   # Mean
boxplot(rowMeans(arimamaysim),main='Mean\nARIMA',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(mean(X),rowMeans(xmaysim),rowMeans(armaysim),rowMeans(arimamaysim)))
points(mean(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                   # Mean

boxplot(rowVars(xmaysim),main='Var\nPhase',ylab='[kWh/hr]^2',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(var(X),rowVars(xmaysim),rowVars(armaysim),rowVars(arimamaysim)))
points(var(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                     # Var
boxplot(rowVars(armaysim),main='Var\nAR',ylab='[kWh/hr]^2',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(var(X),rowVars(xmaysim),rowVars(armaysim),rowVars(arimamaysim)))
points(var(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                     # Var
boxplot(rowVars(arimamaysim),main='Var\nARIMA',ylab='[kWh/hr]^2',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
```

```
        ylim=range(var(X),rowVars(xmaysim),rowVars(armaysim),rowVars(arimamaysim)))
points(var(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                    # Var
dev.off()
```

```
## pdf
##   2
```

```
# Skew and Kurtosis distributions
png(file='Figures/Fig16.png',width=2100,height=1100,pointsize=44)
par(mfrow=c(1,6),mar=c(2.1,4.8,4.5,0.8))
boxplot(rowSkewness(xmaysim),main='Skew\nPhase',ylab='Dimensionless',cex.main=1.4,
        cex.lab=1.4,cex.axis=1.4,lwd=4,
        ylim=range(skewness(X),rowSkewness(xmaysim),rowSkewness(armaysim),
                   rowSkewness(arimamaysim)))
points(skewness(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)           # Skewness

boxplot(rowSkewness(armaysim),main='Skew\nAR',ylab='Dimensionless',cex.main=1.4,
        cex.lab=1.4,cex.axis=1.4,lwd=4,
        ylim=range(skewness(X),rowSkewness(xmaysim),rowSkewness(armaysim),
                   rowSkewness(arimamaysim)))
points(skewness(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)           # Skewness

boxplot(rowSkewness(arimamaysim),main='Skew\nARIMA',ylab='Dimensionless',cex.main=1.4,
        cex.lab=1.4,cex.axis=1.4,lwd=4,
        ylim=range(skewness(X),rowSkewness(xmaysim),rowSkewness(armaysim),
                   rowSkewness(arimamaysim)))
points(skewness(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)           # Skewness

boxplot(rowKurtosis(xmaysim),main='Kurt\nPhase',ylab='Dimensionless',cex.main=1.4,
        cex.lab=1.4,cex.axis=1.4,lwd=4,
        ylim=range(kurtosis(X),rowKurtosis(xmaysim),rowKurtosis(armaysim),
                   rowKurtosis(arimamaysim)))
points(kurtosis(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)           # Kurtosis

boxplot(rowKurtosis(armaysim),main='Kurt\nAR',ylab='Dimensionless',cex.main=1.4,
        cex.lab=1.4,cex.axis=1.4,lwd=4,
        ylim=range(kurtosis(X),rowKurtosis(xmaysim),rowKurtosis(armaysim),
                   rowKurtosis(arimamaysim)))
points(kurtosis(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)           # Kurtosis
boxplot(rowKurtosis(arimamaysim),main='Kurt\nARIMA',ylab='Dimensionless',cex.main=1.4,
        cex.lab=1.4,cex.axis=1.4,lwd=4,
        ylim=range(kurtosis(X),rowKurtosis(xmaysim),rowKurtosis(armaysim),
                   rowKurtosis(arimamaysim)))
points(kurtosis(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)            # Kurtosis
dev.off()
```

```
## pdf
##   2
```

```
# Max, min and sum distributions
png(file='Figures/Fig17.png',width=2100,height=1100,pointsize=44)
par(mfrow=c(1,9),mar=c(2.1,4.3,4.5,1.5))
boxplot(rowMaxs(xmaysim),main='Max\nPhase',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(max(X),rowMaxs(xmaysim),rowMaxs(armaysim),rowMaxs(arimamaysim)))
```

```r
points(max(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Max
boxplot(rowMaxs(armaysim),main='Max\nAR',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(max(X),rowMaxs(xmaysim),rowMaxs(armaysim),rowMaxs(arimamaysim)))
points(max(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Max

boxplot(rowMaxs(arimamaysim),main='Max\nARIMA',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(max(X),rowMaxs(xmaysim),rowMaxs(armaysim),rowMaxs(arimamaysim)))
points(max(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Max

boxplot(rowMins(xmaysim),main='Min\nPhase',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(min(X),rowMins(xmaysim),rowMins(armaysim),rowMins(arimamaysim)))
points(min(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Min

boxplot(rowMins(armaysim),main='Min\nAR',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(min(X),rowMins(xmaysim),rowMins(armaysim),rowMins(arimamaysim)))
points(min(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Min

boxplot(rowMins(arimamaysim),main='Min\nARIMA',ylab='kWh/hr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(min(X),rowMins(xmaysim),rowMins(armaysim),rowMins(arimamaysim)))
points(min(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Min

boxplot(rowSums(xmaysim),main='Sum\nPhase',ylab='kWh/yr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(sum(X),rowSums(xmaysim),rowSums(armaysim),rowSums(arimamaysim)))
points(sum(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Sum
boxplot(rowSums(armaysim),main='Sum\nAR',ylab='kWh/yr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(sum(X),rowSums(xmaysim),rowSums(armaysim),rowSums(arimamaysim)))
points(sum(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Sum

boxplot(rowSums(arimamaysim),main='Sum\nARIMA',ylab='kWh/yr',cex.main=1.4,cex.lab=1.4,
        cex.axis=1.4,lwd=4,
        ylim=range(sum(X),rowSums(xmaysim),rowSums(armaysim),rowSums(arimamaysim)))
points(sum(X),col='red',pch=19,cex=2); grid(nx=NA,ny=NULL,lwd=4)                 # Sum
dev.off()
```

```
## pdf
##   2
```

```r
# Appendix A: RBSA Field Categories
# 1. Appliance categories:
# Cwash: Clothes washer energy use in kWh.
# Cwash_2: Additional clothes washer energy use in kWh.
# Dryer: Clothes dryer energy use in kWh. Includes heating element, drum motor, and
# exhaust fan energy.
# Dryer_g_e: Gas dryer electric energy use in kWh. Applies only to dryer motor and fan
#   energy. Gas consumption not measured.
# Dwash: Dishwasher energy use in kWh.
# Freezer: Stand-alone freezer energy use in kWh.
```

```
# Freezer_2: Additional stand-alone freezer energy use in kWh.
# Oven: Electric cooking stove (oven and range) energy use in kWh.
# Oven_2: Additional electric cooking stove (oven and range) energy use in kWh.
# Refrig: Primary refrigerator energy use in kWh. All refrigerators are combination
#   refrigerator/freezers. Primary is defined as the refrigerator in the kitchen or space
#   nearest kitchen.
# Refrig_2: Secondary refrigerator energy use in kWh. All refrigerators are combination
#   refrigerator/freezers. Secondary is defined as a refrigerator located in areas other
#   than the primary refrigerator.
#
# 2. Domestic Hot Water Heating Categories:
# DHW_1: Primary electric resistance tank water heater energy in kWh.
# DHW_2: Secondary electric resistance tank water heater energy in kWh.
# DHW_HP: Heat pump water heater energy use in kWh.
# DHW_g_Therms: Primary gas tank water heater energy use in therms. Calculated from
#   measured ontime multiplied by known gas flow rate. Gas flow is rate measured by a
#   one-time "clocking" of the gas utility meter on site.
#
# 3. Heating, Ventilating and Air Conditioning Categories (HVAC):
# AC: Central air conditioner outdoor unit energy use in kWh.
# Boilr_g_Therms: Gas boiler energy use in therms. Calculated from measured boiler ontime
#   multiplied by known gas flow rate. Gas flow is rate measured by a one-time "clocking"
#   of the gas utility meter on site.
# Boilr_g_e: Hydronic loop electric pump energy use in kWh.
# DHP: Ductless heat pump total energy use in kWh. Includes both outdoor and indoor units.
# ER: Zonal electric resistance heater energy use in kWh.
# ER_2: Additional zonal electric resistance heater energy use in kWh.
# ER_3: Additional zonal electric resistance heater energy use in kWh.
# ER_4: Additional zonal electric resistance heater energy use in kWh.
# ER_5: Additional zonal electric resistance heater energy use in kWh.
# Furn: Electric furnace resistance heating element energy use in kWh
# Furn_AH: Central forced air system air handler energy use in kWh. Includes air handlers
#   for gas furnaces, electric resistance furnaces, central air conditioners & heat pumps.
# Furn_g_s1_Therms: Gas furnace first stage heating energy use in therms. Calculated from
#   measured boiler ontime multiplied by known gas flow rate. Gas flow is rate measured by
#   a one-time "clocking" of the gas utility meter on site.
# Furn_g_s2_Therms: Gas furnace second stage heating energy use in therms. Calculated from
#   measured boiler ontime multiplied by known gas flow rate. Gas flow is rate measured by
#   a one-time "clocking" of the gas utility meter on site.
# GSHP_in: Ground source heat pump indoor unit energy use in kWh.
# GSHP_out: Ground source heat pump outdoor unit energy use in kWh.
# HP_VLT:Heat pump vapor line temperature measured in Fahrenheit.
# HP_in: Air source heat pump system auxilliary resistance element energy use in kWh.
#   Located indoor, at the air handler, the elements provide additional heat.
# HP_in_2: Additional air source heat pump system auxilliary resistance element energy use
#   in kWh. Located indoor, at the air handler, the elements provide additional heat.
# HP_out: Air source heat pump outdoor unit energy use. Records energy use for both
#   heating and cooling.
# HP_out_2: Additional air source heat pump outdoor unit energy use. Records energy use
#   for both heating and cooling.
# PTAC: Packaged terminal air conditioner energy use in kWh.
# PTHP: Packaged terminal heat pump energy use in kWh.
#
```

```
# 4. Lighting Categories:
# BR: Bedroom light fixture group energy use in kWh.
# BR_2: Additional bedroom light fixture group energy use in kWh.
# BR_2_ontime: Additional bedroom light fixture group hours of ontime.
# BR_3: Additional bedroom light fixture group energy use in kWh.
# BR_3_ontime: Additional bedroom light fixture group hours of ontime.
# BR_4: Additional bedroom light fixture group energy use in kWh.
# BR_4_ontime: Additional bedroom light fixture group hours of ontime.
# BR_5: Additional bedroom light fixture group energy use in kWh.
# BR_5_ontime: Additional bedroom light fixture group hours of ontime.
# BR_6: Additional bedroom light fixture group energy use in kWh.
# BR_6_ontime: Additional bedroom light fixture group hours of ontime.
# BR_7: Additional bedroom light fixture group energy use in kWh.
# BR_7_ontime: Additional bedroom light fixture group hours of ontime.
# BR_master: Master bedroom light fixture group energy use in kWh.
# BR_master_2: Additional Master bedroom light fixture group energy use in kWh.
# BR_master_2_ontime: Additional Master bedroom light fixture group hours of ontime.
# BR_master_3: Additional Master bedroom light fixture group energy use in kWh.
# BR_master_3_ontime: Additional Master bedroom light fixture group hours of ontime.
# BR_master_4: Additional Master bedroom light fixture group energy use in kWh.
# BR_master_4_ontime: Additional Master bedroom light fixture group hours of ontime.
# BR_master_ontime: Master bedroom light fixture group hours of ontime.
# BR_ontime: Bedroom light fixture group hours of ontime.
# Basement: Basement light fixture group energy use in kWh.
# Basement_2: Additional basement light fixture group energy use in kWh.
# Basement_2_ontime: Additional basement light fixture group hours of ontime.
# Basement_3: Additional basement light fixture group energy use in kWh.
# Basement_3_ontime: Additional basement light fixture group hours of ontime.
# Basement_4: Additional basement light fixture group energy use in kWh.
# Basement_4_ontime: Additional basement light fixture group hours of ontime.
# Basement_ontime: basement light fixture group hours of ontime.
# Bath: Bathroom light fixture group energy use in kWh.
# Bath_2: Additional bathroom light fixture group energy use in kWh.
# Bath_2_ontime: Additional bathroom light fixture group hours of ontime.
# Bath_3: Additional bathroom light fixture group energy use in kWh.
# Bath_3_ontime: Additional bathroom light fixture group hours of ontime.
# Bath_4: Additional bathroom light fixture group energy use in kWh.
# Bath_4_ontime: Additional bathroom light fixture group hours of ontime.
# Bath_5: Additional bathroom light fixture group energy use in kWh.
# Bath_5_ontime: Additional bathroom light fixture group hours of ontime.
# Bath_master: Master bathroom light fixture group energy use in kWh.
# Bath_master_2: Additional master bathroom light fixture group energy use in kWh.
# Bath_master_2_ontime: Additional master bathroom light fixture group hours of ontime.
# Bath_master_3: Additional master bathroom light fixture group energy use in kWh.
# Bath_master_3_ontime: Additional master bathroom light fixture group hours of ontime.
# Bath_master_4: Additional master bathroom light fixture group energy use in kWh.
# Bath_master_4_ontime: Additional master bathroom light fixture group hours of ontime.
# Bath_master_ontime: Master bathroom light fixture group hours of ontime.
# Bath_ontime: Bathroom light fixture group hours of ontime.
# Closet: Closet light fixture group energy use in kWh.
# Closet_2: Additional closet light fixture group energy use in kWh.
# Closet_2_ontime: Additional closet light fixture group hours of ontime.
# Closet_3: Additional closet light fixture group energy use in kWh.
```

```
# Closet_3_ontime: Additional closet light fixture group hours of ontime.
# Closet_ontime: Closet light fixture group hours of ontime.
# Den: Den light fixture group energy use in kWh.
# Den_2: Additional den light fixture group energy use in kWh.
# Den_2_ontime: Additional den light fixture group hours of ontime.
# Den_3: Additional den light fixture group energy use in kWh.
# Den_3_ontime: Additional den light fixture group hours of ontime.
# Den_ontime: Den light fixture group hours of ontime.
# Dining: Dining light fixture group energy use in kWh.
# Dining_2: Additional dining light fixture group energy use in kWh.
# Dining_2_ontime: Additional dining light fixture group hours of ontime.
# Dining_ontime: Additional dining light fixture group hours of ontime.
# Entryway: Interior entryway light fixture group energy use in kWh.
# Entryway_2: Additional interior entryway light fixture group energy use in kWh.
# Entryway_2_ontime: Additional interior entryway light fixture group hours of ontime.
# Entryway_ontime: Interior entryway light fixture group hours of ontime.
# Exterior: Exterior light fixture group energy use in kWh.
# Exterior_2: Additional exterior light fixture group energy use in kWh.
# Exterior_2_ontime: Additional exterior light fixture group hours of ontime.
# Exterior_4: Additional exterior light fixture group energy use in kWh.
# Exterior_4_ontime: Additional exterior light fixture group hours of ontime.
# Exterior_ontime: Exterior light fixture group hours of ontime.
# Family: Family room light fixture group energy use in kWh.
# Family_2: Additional family room light fixture group energy use in kWh.
# Family_2_ontime: Additional family room light fixture group hours of ontime.
# Family_3: Additional family room light fixture group energy use in kWh.
# Family_3_ontime: Additional family room light fixture group hours of ontime.
# Family_4: Additional family room light fixture group energy use in kWh.
# Family_4_ontime: Additional family room light fixture group hours of ontime.
# Family_5: Additional family room light fixture group energy use in kWh.
# Family_5_ontime: Additional family room light fixture group hours of ontime.
# Family_6: Additional family room light fixture group energy use in kWh.
# Family_6_ontime: Additional family room light fixture group hours of ontime.
# Family_ontime: Family room light fixture group hours of ontime.
# Garage: Garage light fixture group energy use in kWh.
# Garage_2: Additional garage light fixture group energy use in kWh.
# Garage_2_ontime: Additional garage light fixture group energy hours of ontime.
# Garage_ontime: Garage light fixture group energy hours of ontime.
# Hall: Hallway light fixture group energy use in kWh.
# Hall_2: Additional hallway light fixture group energy use in kWh.
# Hall_2_ontime: Additional hallway light fixture group hours of ontime.
# Hall_3: Additional hallway light fixture group energy use in kWh.
# Hall_3_ontime: Additional hallway light fixture group hours of ontime.
# Hall_4: Additional hallway light fixture group energy use in kWh.
# Hall_4_ontime: Additional hallway light fixture group hours of ontime.
# Hall_ontime: Hallway light fixture group hours of ontime.
# Kitchen: Kitchen light fixture group energy use in kWh.
# Kitchen_2: Additional kitchen light fixture group energy use in kWh.
# Kitchen_2_ontime: Additional kitchen light fixture group hours of ontime.
# Kitchen_3: Additional kitchen light fixture group energy use in kWh.
# Kitchen_3_ontime: Additional kitchen light fixture group hours of ontime.
# Kitchen_4: Additional kitchen light fixture group energy use in kWh.
# Kitchen_4_ontime: Additional kitchen light fixture group hours of ontime.
```

```
# Kitchen_6: Additional kitchen light fixture group energy use in kWh.
# Kitchen_6_ontime: Additional kitchen light fixture group hours of ontime.
# Kitchen_ontime: Kitchen light fixture group hours of ontime.
# Laundry: Laundry room light fixture group energy use in kWh.
# Laundry_2: Additional laundry room light fixture group energy use in kWh.
# Laundry_2_ontime: Additional laundry room light fixture group hours of ontime.
# Laundry_3: Additional laundry room light fixture group energy use in kWh.
# Laundry_3_ontime: Additional laundry room light fixture group hours of ontime.
# Laundry_ontime: Laundry room light fixture group hours of ontime.
# Living: Living room light fixture group energy use in kWh.
# Living_2: Additional living room light fixture group energy use in kWh.
# Living_2_ontime: Additional living room light fixture group hours of ontime.
# Living_3: Additional living room light fixture group energy use in kWh.
# Living_3_ontime: Additional living room light fixture group hours of ontime.
# Living_4: Additional living room light fixture group energy use in kWh.
# Living_4_ontime: Additional living room light fixture group hours of ontime.
# Living_5: Additional living room light fixture group energy use in kWh.
# Living_5_ontime: Additional living room light fixture group hours of ontime.
# Living_ontime: Living room light fixture group hours of ontime.
# Office: Office room light fixture group energy use in kWh.
# Office_2: Additional office room light fixture group energy use in kWh.
# Office_2_ontime: Additional office room light fixture group hours of ontime.
# Office_3: Additional office room light fixture group energy use in kWh.
# Office_3_ontime: Additional office room light fixture group hours of ontime.
# Office_4: Additional office room light fixture group energy use in kWh.
# Office_4_ontime: Additional office room light fixture group hours of ontime.
# Office_ontime: Office room light fixture group hours of ontime.
#
# 5. Other Categories
# Other: Unspecified device energy use in kWh.
# OtherRoom: Unspecified room light fixture group energy use in kWh.
# OtherRoom_2: Additional unspecified room light fixture group energy use in kWh.
# OtherRoom_2_ontime: Additional unspecified room light fixture group hours of ontime.
# OtherRoom_3: Additional unspecified room light fixture group energy use in kWh.
# OtherRoom_3_ontime: Additional unspecified room light fixture group hours of ontime.
# OtherRoom_4: Additional unspecified room light fixture group energy use in kWh.
# OtherRoom_4_ontime: Additional unspecified room light fixture group hours of ontime.
# OtherRoom_5: Additional unspecified room light fixture group energy use in kWh.
# OtherRoom_5_ontime: Additional unspecified room light fixture group hours of ontime.
# OtherRoom_ontime: Unspecified room light fixture group hours of ontime.
# Other_2: Additional unspecified device energy use in kWh.
# Other_3: Additional unspecified device energy use in kWh.
# Septic: Septic pump energy use in kWh.
# Spa: Hot tub energy use in kWh.
# Sump: Sump pump energy use in kWh.
# Well: Well pump energy use in kWh.
#
# 6. Plug Categories
# AC_windo: Window air conditioner energy use in kWh.
# CBox: Cable box energy use in kWh. The cable box label is reserved for devices with that
#   single function and excludes those that are combination units such as CBox_and_DVR.
# CBox_2: Additional cable box energy use in kWh.
# CBox_3: Additional cable box energy use in kWh.
```

```
# CBox_and_DVR: Cable box and digital video recorder combination unit energy use in kWh.
# CBox_and_DVR_2: Additional cable box and digital video recorder combination unit energy
#   use in kWh.
# CPU: Computer tower energy use in kWh (monitors and accessories excluded).
# CPU_2: Additional computer tower energy use in kWh.
# CPU_3: Additional computer tower energy use in kWh.
# CPU_4: Additional computer tower energy use in kWh.
# Comp: Computer energy use in kWh. A computer is defined an all-in-one device which has
#   an integrated display, central processing unit, memory, and storage. Examples include
#   laptops and iMacs.
# Comp_2: Additional computer energy use in kWh. A computer is defined an all-in-one
#   device which has an integrated display, central processing unit, memory, and storage.
#   Examples include laptops and iMacs.
# Comp_3: Additional computer energy use in kWh. A computer is defined an all-in-one
#   device which has an integrated display, central processing unit, memory, and storage.
#   Examples include laptops and iMacs.
# Comp_acc: Computer accessory energy use in kWh. Examples include  printer, scanner,
#   speakers, modem, router.
# Comp_acc_2: Additional computer accessory energy use in kWh. Examples include  printer,
#   scanner, speakers, modem, router.
# Comp_acc_3: Additional computer accessory energy use in kWh. Examples include  printer,
#   scanner, speakers, modem, router.
# Comp_acc_4: Additional computer accessory energy use in kWh. Examples include  printer,
#   scanner, speakers, modem, router.
# Comp_and_acc: Computer and accessory energy use in kWh that were monitored on same
#   channel.
# Comp_and_acc_2: Additional computer and accessory energy use in kWh that were monitored
#   on same channel.
# DVD: Digital video disc player energy use in kWh.
# DVD_2: Additional digital video disc player energy use in kWh.
# DVR: Digital video recorder energy use in kWh.
# DVR_2: Additional Digital video recorder energy use in kWh.
# Game_cons: Gaming console energy use in kWh.
# Game_cons_2: Additional gaming console energy use in kWh.
# Game_cons_3: Additional gaming console energy use in kWh.
# Microw: Microwave oven energy use in kWh.
# Monitor: Computer monitor energy use in kWh.
# Stereo: Stereo energy use in kWh.
# Stereo_acc: Stereo accessory energy use in kWh. Examples include subwoofers.
# TV: Primary television energy use in kWh. Primary is defined as TV with the most on time
#   in the house.
# TV_2: Secondary television energy use in kWh. Secondary is defined as TV with the second
#   most on time in the house.
# TV_3: Tertiary television energy use in kWh. Tertiary is defined as TV with the third
#   most on time in the house.
# TV_acc: Television accessory energy use in kWh. Accessory other than cable box, DVR, DVD
#   player, or gaming console.
# TV_acc_2: Additional television accessory energy use in kWh. Accessory other than cable
#   box, DVR, DVD player, or gaming console.
# TV_and_acc: Television and television accessory, measured jointly, energy use in kWh.
#   Accessory other than cable box, DVR, DVD player, or gaming console.
#
# 7. Electrical Service Categories:
```

```
# Serv_A: Electricity use for one leg of service drop in kWh.
# Serv_A_2: Electricity use for one leg of additional service drop in kWh.
# Serv_B: Electricity use for one leg of service drop in kWh.
# Serv_B_2: Electricity use for one leg of additional service drop in kWh.
# Serv_C: Special case measurement for both legs of electric service drop energy in kWh.
# For siteID=13912, no A and B legs exist so C is total Service. For siteID=22138,
# Serv_C gets summed with Serv_A & Serv_B for total Service.
# Service: Total house electricity use in kWh. The sum of all service legs.
# Subp: Electrical subpanel energy use in kWh
# Subp_2: Additional electrical subpanel energy use in kWh.
# Subp_A: Electricity use in kWh for one leg of subpanel.
# Subp_B: Electricity use in kWh for one leg of subpanel.
#
# 8. Temperature Categories:
# IDT: Indoor temperature measured in Fahrenheit. Sensor located next to central/main
#   thermostat.
# ODT: Outdoor temperature measured in Fahrenheit. Sensor placed on north face of house.
# WST: Outdoor temperature in Fahrenheit reported from weather station nearest to site.
#
# End Appendix A
```