

- Install and Load Libraries
- Obtain Data
 - Connect to Database
 - Obtain Dataset
- Explore and prepare data
 - Investigate Data
 - Create DateTime Objects
 - Any NAs?
 - Data Documentation
 - Dataframes, Functions, and Objects
 - Visualizations
 - Total Energy Consumption for Each Submeter
 - Energy consumption on Mondays at 8:00pm
 - Energy consumption, Christmas Week 2008
 - Energy consumption by weekdays
 - Time of Day
 - Total Power, grouped by various time intervals, for each submeter and combined submeters
 - Zero Power Days
 - dataframes for forecasting
 - dataframes for decomposing
- submeter 1, mornings, 2007,2008,2009 only (full time periods only)
- convert to time series
- Mornings, by week, instead of day
- Mornings, by week, as time series
 - Energy over Time for each Submeter (and Frequency)
 - Submeter 1
 - Submeter 2
 - Submeter 3
 - Total Energy Consumption for Each Submeter
 - Exploring Different Intervals in 2008
 - Day of January 9, 2008
 - Christmas Week, 2008
 - Weekdays in 2008
 - Weekdays All Years
 - Submeter 1
 - Submeter 2
 - Submeter 3
 - Time of Day Visualizations
 - Total Usage by Time of Day
 - Total Usage by Time of Day, and Submeter
 - Total Usage by Weekday

Install and Load Libraries

```
#install packages
install.packages("RMySQL")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
library(RMySQL)

#If there's trouble with installing RMySQL, try this:
#install.packages('RMySQL', dependencies=TRUE, repos='http://cran.rstudio.com/')
#chooseCRANmirror() I tried chooseCRANmirror() with selection 65. It somehow works.
```

Hide

```
install.packages("dplyr ")
```

```
Warning in install.packages :
  package 'dplyr ' is not available (for R version 3.6.1)
```

Hide

```
install.packages("RMySQL")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/RMySQL_0.10.19.tgz'
Content type 'application/x-gzip' length 1760084 bytes (1.7 MB)
=====
downloaded 1.7 MB
```

```
The downloaded binary packages are in
  /var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages
```

Hide

```
install.packages("ggplot2")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("tidyr")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("lubridate")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/lubridate_1.7.4.tgz'
Content type 'application/x-gzip' length 1512972 bytes (1.4 MB)
=====
downloaded 1.4 MB
```

The downloaded binary packages are in
/var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages

Hide

```
install.packages("ggplot2")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/ggplot2_3.2.1.tgz'
Content type 'application/x-gzip' length 3973186 bytes (3.8 MB)
=====
downloaded 3.8 MB
```

The downloaded binary packages are in
/var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages

Hide

```
install.packages("plotly")
```

```
Error in install.packages : Updating loaded packages
```

Hide

```
install.packages("RcppArmadillo")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/RcppArmadillo_0.9.850.1.0.tgz'
Content type 'application/x-gzip' length 1776828 bytes (1.7 MB)
=====
downloaded 1.7 MB
```

The downloaded binary packages are in
/var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages

Hide

```
install.packages("tidyr")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/tidyr_1.0.2.tgz'
Content type 'application/x-gzip' length 1020461 bytes (996 KB)
=====
downloaded 996 KB
```

The downloaded binary packages are in
/var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages

Hide

```
install.packages("plotly")
```

Error in install.packages : Updating loaded packages

Hide

```
install.packages("forecast", dependency = TRUE)
```

Error in install.packages : Updating loaded packages

Hide

```
install.packages("imputeTS")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/imputeTS_3.0.tgz'
Content type 'application/x-gzip' length 2409150 bytes (2.3 MB)
=====
downloaded 2.3 MB
```

The downloaded binary packages are in
/var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages

Hide

```
install.packages("plotly")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/plotly_4.9.2.tgz'
Content type 'application/x-gzip' length 2987045 bytes (2.8 MB)
=====
downloaded 2.8 MB
```

The downloaded binary packages are in
/var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages

[Hide](#)

```
install.packages("forecast", dependency = TRUE)
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/forecast_8.11.tgz'
Content type 'application/x-gzip' length 2483677 bytes (2.4 MB)
=====
downloaded 2.4 MB
```

The downloaded binary packages are in
/var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages

[Hide](#)

```
install.packages("ggfortify")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.6/ggfortify_0.4.8.tgz'
Content type 'application/x-gzip' length 2364813 bytes (2.3 MB)
=====
downloaded 2.3 MB
```

The downloaded binary packages are in
/var/folders/hm/2md7sccd0479bw81zsh0yyq80000gn/T//Rtmp2fWG7D/downloaded_packages

[Hide](#)

```
# load libraries
library(dplyr)
library(ggplot2)
library(ggfortify)
library(tidyr)
library(lubridate)
library(scales)
library(plotly)
library(forecast)
library(imputeTS)
```

[Hide](#)

```
# Set specific options of for libraries
## Only use scientific notation for values greather than set amount
options(scipen=10000000)
```

[Hide](#)

```
#confirm libraries
(.packages())
```

```
[1] "imputeTS" "forecast" "scales" "lubridate" "tidyr" "ggfortify" "dplyr"
[8] "plotly" "ggplot2" "RMySQL" "DBI" "stats" "graphics" "grDevices"
[15] "utils" "datasets" "methods" "base"
```

Obtain Data

Connect to Database

Hide

```
# Create a database connection
con = dbConnect(MySQL(), user='deepAnalytics', password='Sqltask1234!', dbname='dataanalytics2018', host='data-analytics-2018.cbrosir2cswx.us-east-1.rds.amazonaws.com')
```

Hide

```
#summary of connection
summary(con)
```

```
<MySQLConnection:0,0>
  User:   deepAnalytics
  Host:   data-analytics-2018.cbrosir2cswx.us-east-1.rds.amazonaws.com
  Dbnme:  dataanalytics2018
  Connection type: data-analytics-2018.cbrosir2cswx.us-east-1.rds.amazonaws.com via TCP/IP

Results:
```

Hide

```
dbGetInfo(con)
```

```
$host
[1] "data-analytics-2018.cbrosir2cswx.us-east-1.rds.amazonaws.com"

$user
[1] "deepAnalytics"

$dbname
[1] "dataanalytics2018"

$conType
[1] "data-analytics-2018.cbrosir2cswx.us-east-1.rds.amazonaws.com via TCP/IP"

$serverVersion
[1] "5.6.10"

$protocolVersion
[1] 10

$threadId
[1] 185355

$rsId
list()
```

Obtain Dataset

[Hide](#)

```
# List the tables contained in the database.
my_tables <- dbListTables(con)
my_tables
```

```
[1] "iris"      "yr_2006" "yr_2007" "yr_2008" "yr_2009" "yr_2010"
```

[Hide](#)

```
# there are 6 tables: "iris"      "yr_2006" "yr_2007" "yr_2008" "yr_2009" "yr_2010"
```

[Hide](#)

```
# Lists attributes contained in a table
list_db_fields_custom_function<- function (x) {dbListFields(con,x)}
lapply(my_tables,list_db_fields_custom_function)
```

```

[[1]]
[1] "id"          "SepalLengthCm" "SepalWidthCm" "PetalLengthCm" "PetalWidthCm"
[6] "Species"

[[2]]
[1] "id"          "Date"          "Time"
[4] "Global_active_power" "Global_reactive_power" "Global_intensity"
[7] "Voltage"      "Sub_metering_1" "Sub_metering_2"
[10] "Sub_metering_3"

[[3]]
[1] "id"          "Date"          "Time"
[4] "Global_active_power" "Global_reactive_power" "Global_intensity"
[7] "Voltage"      "Sub_metering_1" "Sub_metering_2"
[10] "Sub_metering_3"

[[4]]
[1] "id"          "Date"          "Time"
[4] "Global_active_power" "Global_reactive_power" "Global_intensity"
[7] "Voltage"      "Sub_metering_1" "Sub_metering_2"
[10] "Sub_metering_3"

[[5]]
[1] "id"          "Date"          "Time"
[4] "Global_active_power" "Global_reactive_power" "Global_intensity"
[7] "Voltage"      "Sub_metering_1" "Sub_metering_2"
[10] "Sub_metering_3"

[[6]]
[1] "id"          "Date"          "Time"
[4] "Global_active_power" "Global_reactive_power" "Global_intensity"
[7] "Voltage"      "Sub_metering_1" "Sub_metering_2"
[10] "Sub_metering_3"

```

Hide

tables for the years 2006 -2010 have the same attributes. Column names are the same.

Hide

We are only using Date, Time and Submeters for our analysis.

```

yr_2006SELECT <- dbGetQuery(con, "SELECT Date, Time, Sub_metering_1, Sub_metering_2, Sub
_metering_3 FROM yr_2006")
yr_2007SELECT <- dbGetQuery(con, "SELECT Date, Time, Sub_metering_1, Sub_metering_2, Sub
_metering_3 FROM yr_2007")
yr_2008SELECT <- dbGetQuery(con, "SELECT Date, Time, Sub_metering_1, Sub_metering_2, Sub
_metering_3 FROM yr_2008")
yr_2009SELECT <- dbGetQuery(con, "SELECT Date, Time, Sub_metering_1, Sub_metering_2, Sub
_metering_3 FROM yr_2009")
yr_2010SELECT <- dbGetQuery(con, "SELECT Date, Time, Sub_metering_1, Sub_metering_2, Sub
_metering_3 FROM yr_2010")

```


Explore and prepare data

Note: MySQL tables are read into R as data.frames, but without coercing character or logical data into factors. Similarly while exporting data.frames, factors are exported as character vectors. Integer columns are usually imported as R integer vectors, except for cases such as BIGINT or UNSIGNED INTEGER which are coerced to R's double precision vectors to avoid truncation (currently R's integers are signed 32-bit quantities). Time variables are imported/exported as character data, so you need to convert these to your favorite date/time representation.

Investigate Data

[Hide](#)

```
# Function to explore tables. Prints out structure, summary, head and tail of data for every table.
investigateDF <- function(df) {list(str(df), summary(df), head(df), tail(df))}
```

[Hide](#)

```
# Investigates tables from 2006 to 2010
investigateDF(yr_2006SELECT)
```

```
'data.frame': 21992 obs. of 5 variables:
 $ Date      : chr  "2006-12-16" "2006-12-16" "2006-12-16" "2006-12-16" ...
 $ Time      : chr  "17:24:00" "17:25:00" "17:26:00" "17:27:00" ...
 $ Sub_metering_1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_2: num  1 1 2 1 1 2 1 1 1 2 ...
 $ Sub_metering_3: num  17 16 17 17 17 17 17 17 17 16 ...

[[1]]
NULL

[[2]]
      Date      Time      Sub_metering_1  Sub_metering_2  Sub_metering_3
Length:21992 Length:21992 Min. : 0.000 Min. : 0.000 Min. : 0.00
Class :character Class :character 1st Qu.: 0.000 1st Qu.: 0.000 1st Qu.: 0.00
Mode :character Mode :character Median : 0.000 Median : 0.000 Median : 0.00
Mean : 1.249 Mean : 2.215 Mean : 7.41
3rd Qu.: 0.000 3rd Qu.: 1.000 3rd Qu.:17.00
Max. :77.000 Max. :74.000 Max. :20.00

[[3]]
```

Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
1 2006-12-16	17:24:00	0	1	17
2 2006-12-16	17:25:00	0	1	16
3 2006-12-16	17:26:00	0	2	17

Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
4 2006-12-16	17:27:00	0	1	17
5 2006-12-16	17:28:00	0	1	17
6 2006-12-16	17:29:00	0	2	17

6 rows

[[4]]

Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
21987 2006-12-31	23:54:00	0	0	0
21988 2006-12-31	23:55:00	0	0	0
21989 2006-12-31	23:56:00	0	0	0
21990 2006-12-31	23:57:00	0	0	0
21991 2006-12-31	23:58:00	0	0	0
21992 2006-12-31	23:59:00	0	0	0

6 rows

Hide

investigateDF(yr_2007SELECT)

```
'data.frame': 521669 obs. of 5 variables:
 $ Date      : chr  "2007-01-01" "2007-01-01" "2007-01-01" "2007-01-01" ...
 $ Time      : chr  "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...
 $ Sub_metering_1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_2: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_3: num  0 0 0 0 0 0 0 0 0 0 ...
```

[[1]]

NULL

[[2]]

Date	Time	Sub_metering_1	Sub_metering_2	Sub_metering_3
Length:521669	Length:521669	Min. : 0.000	Min. : 0.000	Min. : 0.000
Class :character	Class :character	1st Qu.: 0.000	1st Qu.: 0.000	1st Qu.: 0.000
Mode :character	Mode :character	Median : 0.000	Median : 0.000	Median : 0.000
		Mean : 1.232	Mean : 1.638	Mean : 5.795
		3rd Qu.: 0.000	3rd Qu.: 1.000	3rd Qu.:17.000
		Max. :78.000	Max. :78.000	Max. :20.000

[[3]]

Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
1 2007-01-01	00:00:00	0	0	0
2 2007-01-01	00:01:00	0	0	0
3 2007-01-01	00:02:00	0	0	0
4 2007-01-01	00:03:00	0	0	0
5 2007-01-01	00:04:00	0	0	0
6 2007-01-01	00:05:00	0	0	0
6 rows				

```
[[4]]
```

	Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
521664	2007-12-31	23:54:00	0	0	18
521665	2007-12-31	23:55:00	0	0	18
521666	2007-12-31	23:56:00	0	0	18
521667	2007-12-31	23:57:00	0	0	18
521668	2007-12-31	23:58:00	0	0	18
521669	2007-12-31	23:59:00	0	0	18
6 rows					

[Hide](#)

```
investigateDF(yr_2008SELECT)
```

```
'data.frame': 526905 obs. of 5 variables:
 $ Date      : chr  "2008-01-01" "2008-01-01" "2008-01-01" "2008-01-01" ...
 $ Time      : chr  "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...
 $ Sub_metering_1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_2: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_3: num  18 18 18 18 18 17 18 18 18 18 ...
```

```
[[1]]
```

```
NULL
```

```
[[2]]
```

Date	Time	Sub_metering_1	Sub_metering_2	Sub_metering_3
Length:526905	Length:526905	Min. : 0.00	Min. : 0.000	Min. : 0.000
Class :character	Class :character	1st Qu.: 0.00	1st Qu.: 0.000	1st Qu.: 0.000
Mode :character	Mode :character	Median : 0.00	Median : 0.000	Median : 1.000
		Mean : 1.11	Mean : 1.256	Mean : 6.034
		3rd Qu.: 0.00	3rd Qu.: 1.000	3rd Qu.:17.000
		Max. :80.00	Max. :76.000	Max. :31.000

```
[[3]]
```

Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
1 2008-01-01	00:00:00	0	0	18
2 2008-01-01	00:01:00	0	0	18
3 2008-01-01	00:02:00	0	0	18
4 2008-01-01	00:03:00	0	0	18
5 2008-01-01	00:04:00	0	0	18
6 2008-01-01	00:05:00	0	0	17

6 rows

```
[[4]]
```

	Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
526900	2008-12-31	23:54:00	0	0	0
526901	2008-12-31	23:55:00	0	0	0
526902	2008-12-31	23:56:00	0	0	0
526903	2008-12-31	23:57:00	0	0	0
526904	2008-12-31	23:58:00	0	0	0
526905	2008-12-31	23:59:00	0	0	0

6 rows

Hide

```
investigateDF(yr_2009SELECT)
```

```
'data.frame': 521320 obs. of 5 variables:
 $ Date      : chr  "2009-01-01" "2009-01-01" "2009-01-01" "2009-01-01" ...
 $ Time      : chr  "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...
 $ Sub_metering_1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_2: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_3: num  0 0 0 0 0 0 0 0 0 0 ...

[[1]]
NULL

[[2]]
      Date      Time      Sub_metering_1  Sub_metering_2  Sub_metering_3
Length:521320 Length:521320 Min. : 0.000 Min. : 0.000 Min. : 0.000
Class :character Class :character 1st Qu.: 0.000 1st Qu.: 0.000 1st Qu.: 0.000
Mode :character  Mode :character Median : 0.000 Median : 0.000 Median : 1.000
Mean : 1.137 Mean : 1.136 Mean : 6.823
3rd Qu.: 0.000 3rd Qu.: 1.000 3rd Qu.:18.000
Max. :82.000 Max. :77.000 Max. :31.000

[[3]]
```

Date	Time	Sub_metering_1	Sub_metering_2	Sub_metering_3
<chr>	<chr>	<dbl>	<dbl>	<dbl>
1 2009-01-01	00:00:00	0	0	0
2 2009-01-01	00:01:00	0	0	0
3 2009-01-01	00:02:00	0	0	0
4 2009-01-01	00:03:00	0	0	0
5 2009-01-01	00:04:00	0	0	0
6 2009-01-01	00:05:00	0	0	0

6 rows

```
[[4]]
```

	Date	Time	Sub_metering_1	Sub_metering_2	Sub_metering_3
	<chr>	<chr>	<dbl>	<dbl>	<dbl>
521315	2009-12-31	23:54:00	0	0	18
521316	2009-12-31	23:55:00	0	0	18

	Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
521317	2009-12-31	23:56:00	0	0	19
521318	2009-12-31	23:57:00	0	0	18
521319	2009-12-31	23:58:00	0	0	18
521320	2009-12-31	23:59:00	0	0	19
6 rows					

[Hide](#)

```
investigateDF(yr_2010SELECT)
```

```
'data.frame':  457394 obs. of  5 variables:
 $ Date      : chr  "2010-01-01" "2010-01-01" "2010-01-01" "2010-01-01" ...
 $ Time      : chr  "00:00:00" "00:01:00" "00:02:00" "00:03:00" ...
 $ Sub_metering_1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_2: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_3: num  18 18 19 18 18 19 18 18 19 18 ...
[[1]]
NULL

[[2]]
      Date      Time      Sub_metering_1  Sub_metering_2  Sub_metering_3
Length:457394  Length:457394  Min.   : 0.0000  Min.   : 0.000  Min.   : 0.000
Class :character  Class :character  1st Qu.: 0.0000  1st Qu.: 0.000  1st Qu.: 1.000
Mode  :character  Mode  :character  Median : 0.0000  Median : 0.000  Median : 1.000
      Mean   : 0.9875  Mean   : 1.102  Mean   : 7.244
      3rd Qu.: 0.0000  3rd Qu.: 1.000  3rd Qu.:18.000
      Max.   :88.0000  Max.   :80.000  Max.   :31.000

[[3]]
```

	Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
1	2010-01-01	00:00:00	0	0	18
2	2010-01-01	00:01:00	0	0	18
3	2010-01-01	00:02:00	0	0	19
4	2010-01-01	00:03:00	0	0	18
5	2010-01-01	00:04:00	0	0	18
6	2010-01-01	00:05:00	0	0	19
6 rows					

```
[[4]]
```

	Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
457389	2010-11-26	20:57:00	0	0	0
457390	2010-11-26	20:58:00	0	0	0
457391	2010-11-26	20:59:00	0	0	0
457392	2010-11-26	21:00:00	0	0	0
457393	2010-11-26	21:01:00	0	0	0
457394	2010-11-26	21:02:00	0	0	0

6 rows

NA

Hide

```
#Combine tables into one dataframe (using dplyr)
df2006_2010 <- bind_rows(yr_2006SELECT,yr_2007SELECT,yr_2008SELECT,yr_2009SELECT,yr_2010
SELECT)
investigateDF(df2006_2010)
```

```
'data.frame': 2049280 obs. of 5 variables:
 $ Date      : chr  "2006-12-16" "2006-12-16" "2006-12-16" "2006-12-16" ...
 $ Time      : chr  "17:24:00" "17:25:00" "17:26:00" "17:27:00" ...
 $ Sub_metering_1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_2: num  1 1 2 1 1 2 1 1 1 2 ...

 $ Sub_metering_3: num  17 16 17 17 17 17 17 17 17 16 ...
[[1]]
NULL

[[2]]
      Date      Time      Sub_metering_1  Sub_metering_2  Sub_metering_3
Length:2049280 Length:2049280 Min. : 0.000 Min. : 0.000 Min. : 0.000
Class :character Class :character 1st Qu.: 0.000 1st Qu.: 0.000 1st Qu.: 0.000
Mode :character  Mode :character Median : 0.000 Median : 0.000 Median : 1.000
                                Mean : 1.122 Mean : 1.299 Mean : 6.458
                                3rd Qu.: 0.000 3rd Qu.: 1.000 3rd Qu.:17.000
                                Max. :88.000 Max. :80.000 Max. :31.000

[[3]]
```

	Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
--	---------------	---------------	-------------------------	-------------------------	-------------------------

Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
1 2006-12-16	17:24:00	0	1	17
2 2006-12-16	17:25:00	0	1	16
3 2006-12-16	17:26:00	0	2	17
4 2006-12-16	17:27:00	0	1	17
5 2006-12-16	17:28:00	0	1	17
6 2006-12-16	17:29:00	0	2	17
6 rows				

[[4]]

	Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
2049275	2010-11-26	20:57:00	0	0	0
2049276	2010-11-26	20:58:00	0	0	0
2049277	2010-11-26	20:59:00	0	0	0
2049278	2010-11-26	21:00:00	0	0	0
2049279	2010-11-26	21:01:00	0	0	0
2049280	2010-11-26	21:02:00	0	0	0
6 rows					

NA

Create DateTime Objects

Hide

```
# Combine Date and Time attribute values in a new attribute column
df2006_2010 <-cbind(df2006_2010,paste(df2006_2010$Date,df2006_2010$Time), stringsAsFactors=FALSE)
head(df2006_2010)
```

Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
1 2006-12-16	17:24:00	0	1	17
2 2006-12-16	17:25:00	0	1	16

Date <chr>	Time <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
3 2006-12-16	17:26:00	0	2	17
4 2006-12-16	17:27:00	0	1	17
5 2006-12-16	17:28:00	0	1	17
6 2006-12-16	17:29:00	0	2	17

6 rows | 1-6 of 6 columns

Hide

```
# Remove Date and Time columns
#drop date and time, which are not time zone adjusted, and rename dateT
df2006_2010 <- df2006_2010[ , !(names(df2006_2010) %in% c("Date","Time"))]
head(df2006_2010)
```

	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>	paste(df2006_2010\$Date, df2006_2010\$Time) <chr>
1	0	1	17	2006-12-16 17:24:00
2	0	1	16	2006-12-16 17:25:00
3	0	2	17	2006-12-16 17:26:00
4	0	1	17	2006-12-16 17:27:00
5	0	1	17	2006-12-16 17:28:00
6	0	2	17	2006-12-16 17:29:00

6 rows

Hide

```
# Give the new attribute in the 6th column a header name
colnames(df2006_2010)[4] <- "DateTime"
head(df2006_2010)
```

	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>	DateTime <chr>
1	0	1	17	2006-12-16 17:24:00
2	0	1	16	2006-12-16 17:25:00
3	0	2	17	2006-12-16 17:26:00
4	0	1	17	2006-12-16 17:27:00
5	0	1	17	2006-12-16 17:28:00
6	0	2	17	2006-12-16 17:29:00

6 rows

Hide

```
# Move the DateTime attribute within the dataset
df2006_2010 <- df2006_2010[,c(ncol(df2006_2010), 1:(ncol(df2006_2010)-1))]
head(df2006_2010)
```

DateTime <chr>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
1 2006-12-16 17:24:00	0	1	17
2 2006-12-16 17:25:00	0	1	16
3 2006-12-16 17:26:00	0	2	17
4 2006-12-16 17:27:00	0	1	17
5 2006-12-16 17:28:00	0	1	17
6 2006-12-16 17:29:00	0	2	17

6 rows

Hide

```
# Convert DateTime from character to POSIXct
df2006_2010$DateTime <- as.POSIXct(df2006_2010$DateTime, "%Y/%m/%d %H:%M:%S")
```

```
unknown timezone '%Y/%m/%d %H:%M:%S'unknown timezone '%Y/%m/%d %H:%M:%S'unknown timezone
'%Y/%m/%d %H:%M:%S'unknown timezone '%Y/%m/%d %H:%M:%S'
```

Hide

```
attr(df2006_2010$DateTime, "tzone") <- "Europe/Paris"
```

```
#Verify
str(df2006_2010)
```

```
'data.frame': 2049280 obs. of 4 variables:
 $ DateTime      : POSIXct, format: "2006-12-16 18:24:00" "2006-12-16 18:25:00" "2006-12-
-16 18:26:00" ...
 $ Sub_metering_1: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Sub_metering_2: num  1 1 2 1 1 2 1 1 1 2 ...
 $ Sub_metering_3: num  17 16 17 17 17 17 17 17 17 16 ...
```

Hide

```
head(df2006_2010)
```

	DateTime <S3: POSIXct>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>
1	2006-12-16 18:24:00	0	1	17
2	2006-12-16 18:25:00	0	1	16
3	2006-12-16 18:26:00	0	2	17
4	2006-12-16 18:27:00	0	1	17
5	2006-12-16 18:28:00	0	1	17
6	2006-12-16 18:29:00	0	2	17
6 rows				

[Hide](#)

```
# Create "year, quarter, month, week, weekday, day, dateTZ(different than original date
[chr string] with time zone applied], hour, and minute attributes
df2006_2010$year <- year(df2006_2010$DateTime)
df2006_2010$quarter <- quarter(df2006_2010$DateTime)
df2006_2010$month <- month(df2006_2010$DateTime)
df2006_2010$week <- week(df2006_2010$DateTime)
df2006_2010$weekday <- weekdays(df2006_2010$DateTime)
df2006_2010$day <- day(df2006_2010$DateTime)
df2006_2010$date <- date(df2006_2010$DateTime)
df2006_2010$hour <- hour(df2006_2010$DateTime)
df2006_2010$minute <- minute(df2006_2010$DateTime)
```

[Hide](#)

```
# verify new attributes
head(df2006_2010)
```

	DateTime <S3: POSIXct>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>	y... <dbl>	quarter <int>	m... <dbl>
1	2006-12-16 18:24:00	0	1	17	2006	4	12
2	2006-12-16 18:25:00	0	1	16	2006	4	12
3	2006-12-16 18:26:00	0	2	17	2006	4	12
4	2006-12-16 18:27:00	0	1	17	2006	4	12
5	2006-12-16 18:28:00	0	1	17	2006	4	12
6	2006-12-16 18:29:00	0	2	17	2006	4	12
6 rows 1-9 of 13 columns							

[Hide](#)

```
tail(df2006_2010)
```

	DateTime <S3: POSIXct>	Sub_metering_1 <dbl>	Sub_metering_2 <dbl>	Sub_metering_3 <dbl>	y... <dbl>	quarter <int>
2049275	2010-11-26 21:57:00	0	0	0	2010	4
2049276	2010-11-26 21:58:00	0	0	0	2010	4
2049277	2010-11-26 21:59:00	0	0	0	2010	4
2049278	2010-11-26 22:00:00	0	0	0	2010	4
2049279	2010-11-26 22:01:00	0	0	0	2010	4
2049280	2010-11-26 22:02:00	0	0	0	2010	4

6 rows | 1-9 of 13 columns

Any NAs?

Hide

```
sum(is.na(df2006_2010))
```

```
[1] 0
```

Hide

```
# no missing values
```

Data Documentation

Source: <http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption#>
(<http://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption#>)

Abstract: Measurements of electric power consumption in one household with a one-minute sampling rate over a period of almost 4 years. Different electrical quantities and some sub-metering values are available.

Attribute Information:

sub_metering_1: energy sub-metering No. 1 (in). It corresponds to the kitchen, containing mainly a dishwasher, an oven and a microwave (hot plates are not electric but gas powered).

sub_metering_2: energy sub-metering No. 2 (in watt-hour of active energy). It corresponds to the laundry room, containing a washing-machine, a tumble-drier, a refrigerator and a light.

sub_metering_3: energy sub-metering No. 3 (in watt-hour of active energy). It corresponds to an electric water-heater and an air-conditioner.

Dataframes, Functions, and Objects

This section is a list of dataframes. I intend to keep most dataframes in one section and call on the object as needed. This way, I can keep my analysis more concise and can easily keep tracks of what objects have already been created.

[Hide](#)

```
# We do not have data for a full year of 2006 and 2010. We will use years 2007–2009 for
time series analysis that either require complete time intervals or would benefit from
the simplicity of complete intervals.
df2007_2009 <- df2006_2010 %>%
  filter( year > 2006 & year < 2010)
```

Visualizations

Total Energy Consumption for Each Submeter

Bar and Pie chart

[Hide](#)

```
# Create a data frame of the sum of energy consumption for each submeter
sum_of_submeters <- data.frame(Kitchen = sum(df2006_2010$Sub_metering_1), LaundryRoom =
  sum(df2006_2010$Sub_metering_2), WaterHeater_AC = sum(df2006_2010$Sub_metering_3))

# Transpose data frame so that submeters and total energy use are each in a column
sum_of_submeters_long = gather(sum_of_submeters, key = "Submeters") %>%
  group_by(Submeters) %>%
  summarize(Total_Energy_Usage = sum(value, na.rm = TRUE))
```

Energy consumption on Mondays at 8:00pm

[Hide](#)

```
## Subset 2008, January 9, every 10 minutes
houseDay20080109_10min <- filter(df2006_2010, year == 2008 & month == 1 & day == 9 & (mi
nute == 0 | minute == 10 | minute == 20 | minute == 30 | minute == 40 | minute == 50))
```

Energy consumption, Christmas Week 2008

[Hide](#)

```
## Subset of Week 52 in 2008, every 4 hours
week52_2018_4hours <- filter(df2006_2010, year == 2008 & week==52 & (hour == 0 | hour ==
4 | hour == 8 | hour == 12 | hour == 16 | hour == 20 ))
```

Energy consumption by weekdays

[Hide](#)

```
#Weekdays
weekdaysAllYears <- df2006_2010 %>%
  group_by(year, weekday, date ) %>%
  summarize(Kitchen = sum(Sub_metering_1),
            LaundryRoom = sum(Sub_metering_2),
            WaterHeaterAC = sum(Sub_metering_3))

weekdays07_09 <- weekdaysAllYears %>%
  filter(year !=2006 & year != 2010)
```

Hide

```
# Total Energy Consumption for each weekday
weekdaysAllYearsTotals <- weekdaysAllYears %>%
  group_by(weekday) %>%
  summarize(Kitchen = sum(Kitchen),
            LaundryRoom = sum(LaundryRoom),
            WaterHeaterAC = sum(WaterHeaterAC)) %>%
  mutate(totalEnergy = Kitchen+LaundryRoom+WaterHeaterAC)
```

Hide

```
#Mondays
mondays <- weekdaysAllYears %>%
  filter(weekday == "Monday")
tuesdays <- weekdaysAllYears %>%
  filter(weekday == "Tuesday")
wednesdays <- weekdaysAllYears %>%
  filter(weekday == "Wednesday")
thursdays <- weekdaysAllYears %>%
  filter(weekday == "Thursday")
fridays <- weekdaysAllYears %>%
  filter(weekday == "Friday")
saturdays <- weekdaysAllYears %>%
  filter(weekday == "Saturday")
sundays <- weekdaysAllYears %>%
  filter(weekday == "Sunday")
```

Weekdays in 2008

Hide

```
thursdays08 <- thursdays %>%
  filter(year == 2008)
sundays08 <- sundays %>%
  filter(year == 2008)
```

Time of Day

Paris Sun Times <https://www.timeanddate.com/sun/france/paris?month=1&year=2020>
<https://www.timeanddate.com/sun/france/paris?month=1&year=2020>)

I will break time into four intervals: morning, afternoon, evening, night

I would look at historical weather trends to get an idea about what particular times represent our time intervals but historical weather data costs money and is out of the scope of this analysis. Instead, I browsed travel sites to get an idea of what time of the day the Frech typically start and end.

I decided to set morning from 6:00am - 11:59pm (in our dataset, it's from hours 6 to 11). I read on TripAdvisor that bakeries typically at 7am, and it's ideal to get a croissant with coffeee. Therefore, I will start morning (with a wakeup time starting) at 6:00am. I may alter this time interval as I learn more about the dataset and see if I can find a distinct pattern between night and morning for this particular family.

Afternoon is from 12:00pm - 4:59pm (hours 12 to 16). Evening follows from 5:00pm - 11:59pm (hours 17 to 23). It is set this late because I read on TripAdvisor that most restaurants are open until midnight and that Paris has a very active nightlife. Night time is the remaining time between 12:00am to 5:59am (hours 0 to 5).

Hide

```
# create new column with time of day in it
allYearsTimeofday <- df2006_2010 %>%
  mutate(timeOfDay = case_when(
    hour %in% c(6,7,8,9,10,11) ~ "1. morning",
    hour %in% c(12,13,14,15,16) ~ "2. afternoon",
    hour %in% c(17,18,19,20,21,22,23,24) ~ "3. evening",
    hour %in% c(0,1,2,3,4,5) ~ "4. night"))
```

Piechart for total use for all years by time of Day

Hide

```
allYearsTimeofdayTotals <- allYearsTimeofday %>%
  group_by(timeOfDay) %>%
  summarize(Kitchen = sum(Sub_metering_1),
            LaundryRoom = sum(Sub_metering_2),
            WaterHeaterAC = sum(Sub_metering_3)) %>%
  mutate(totalEnergy = Kitchen+LaundryRoom+WaterHeaterAC)
```

Total Power, grouped by various time intervals, for each submeter and combined submeters

Hide

```
# Submeters by Minute
submetersByMinute <- df2006_2010 %>%
  mutate(TotalEnergy_perMinute= (Sub_metering_1+Sub_metering_2+Sub_metering_3))
submetersByMinute$minute_index <- seq.int(nrow(submetersByMinute))
```

Hide

```
# Submeters by Hour
submetersByHour <- df2006_2010 %>%
  group_by(year,date,hour) %>%
  summarize(Kitchen = sum(Sub_metering_1),
            LaundryRoom = sum(Sub_metering_2),
            WaterHeater_AC = sum(Sub_metering_3))%>%
  mutate(TotalEnergy_perHour= (Kitchen+LaundryRoom+WaterHeater_AC))
submetersByHour$hour_index <- seq.int(nrow(submetersByHour))
```

Hide

```
# We will use 2007 to 2009 to reduce granularity and to compare equal period intervals
submetersByHour24_Total <- df2007_2009 %>%
  group_by(hour) %>%
  summarize(Kitchen = sum(Sub_metering_1),
            LaundryRoom = sum(Sub_metering_2),
            WaterHeater_AC = sum(Sub_metering_3))%>%
  mutate(TotalEnergy_perHour= (Kitchen+LaundryRoom+WaterHeater_AC))
```

Hide

```
submetersByHour24_Average <- df2007_2009 %>%
  group_by(year,hour) %>%
  summarize(Kitchen = mean(Sub_metering_1),
            LaundryRoom = mean(Sub_metering_2),
            WaterHeater_AC = mean(Sub_metering_3))%>%
  mutate(MeanEnergy_perHour= (Kitchen+LaundryRoom+WaterHeater_AC)) %>%
  round(0)
```

Hide

```
submetersByHour24wMonth_Average <- df2007_2009 %>%
  filter(year != 2010 & year != 2007 & year != 2008 ) %>%
  group_by(year,month,hour) %>%
  summarize(Kitchen = mean(Sub_metering_1),
            LaundryRoom = mean(Sub_metering_2),
            WaterHeater_AC = mean(Sub_metering_3))%>%
  mutate(MeanEnergy_perHour= (Kitchen+LaundryRoom+WaterHeater_AC)) %>%
  round(0)
```

Hide

```
# Submeters by Day
submetersByDay <- df2006_2010 %>%
  group_by(date) %>%
  summarize(Kitchen = sum(Sub_metering_1),
            LaundryRoom = sum(Sub_metering_2),
            WaterHeater_AC = sum(Sub_metering_3)) %>%
  mutate(TotalEnergy_perDay= (Kitchen+LaundryRoom+WaterHeater_AC))
submetersByDay$day_index <- seq.int(nrow(submetersByDay))
```

Hide


```
# total weekly usage for each submeter
submetersByWeek <- df2006_2010 %>%
  group_by(year, week ) %>%
  summarize(Kitchen = sum(Sub_metering_1),
            LaundryRoom = sum(Sub_metering_2),
            WaterHeater_AC = sum(Sub_metering_3)) %>%
  mutate(TotalEnergy_perWeek= (Kitchen+LaundryRoom+WaterHeater_AC))
submetersByWeek$week_index <- seq.int(nrow(submetersByWeek))
```

Hide

```
# total monthly usage for each submeter
submetersByMonth <- df2006_2010 %>%
  group_by(year, month ) %>%
  summarize(Kitchen = sum(Sub_metering_1),
            LaundryRoom = sum(Sub_metering_2),
            WaterHeater_AC = sum(Sub_metering_3)) %>%
  mutate(TotalEnergy_perMonth= (Kitchen+LaundryRoom+WaterHeater_AC))
submetersByMonth$month_index <- seq.int(nrow(submetersByMonth))
```

Hide

```
# total quarterly usage for each submeter
submetersByQuarter <- df2006_2010 %>%
  group_by(year, quarter ) %>%
  summarize(Kitchen = sum(Sub_metering_1),
            LaundryRoom = sum(Sub_metering_2),
            WaterHeater_AC = sum(Sub_metering_3)) %>%
  mutate(TotalEnergy_perQuarter= (Kitchen+LaundryRoom+WaterHeater_AC))
submetersByQuarter$quarter_index <- seq.int(nrow(submetersByQuarter))
```

Zero Power Days

Hide

```
zeroPowerDays_Kitchen <- df2007_2009 %>%
  group_by(year, month, weekday, date) %>%
  summarize(Kitchen = sum(Sub_metering_1), LaundryRoom = sum(Sub_metering_2), WaterHeater_AC = sum(Sub_metering_3)) %>%
  filter(Kitchen == 0) %>%
  select(-LaundryRoom, -WaterHeater_AC)
```

Hide

```
zeroPowerDays_Kitchen_Frequency_Weekdays <- zeroPowerDays_Kitchen %>%
  group_by(year, weekday) %>%
  summarize(Total = n())
```

Hide

```
zeroPowerDays_Laundry <- df2007_2009 %>%
  group_by(year, month, weekday, date) %>%
  summarize(Kitchen = sum(Sub_metering_1), LaundryRoom = sum(Sub_metering_2), WaterHeater_AC = sum(Sub_metering_3)) %>%
  filter(LaundryRoom == 0) %>%
  select(-Kitchen, -WaterHeater_AC)
```

[Hide](#)

```
zeroPowerDays_WaterHeater_AC <- df2006_2010 %>%
  group_by(year, month, weekday, date) %>%
  summarize(Kitchen = sum(Sub_metering_1), LaundryRoom = sum(Sub_metering_2), WaterHeater_AC = sum(Sub_metering_3)) %>%
  filter(WaterHeater_AC == 0) %>%
  select(-Kitchen, -LaundryRoom)
```

[Hide](#)

```
#Create time objects of submeter_1:
kitchenHourly_ts <- ts(submetersByHour[,4], start = c(2006, 12), frequency = 8760)
kitchenHourlyMeanMonthly_ts <- ts(submetersByHour24wMonth_Average[,4], start = c(2007, 1), frequency = 12*24)

#submeter1Weekly_ts <- ts(submetersByWeek[,3], start = c(2007, 1), frequency = 52)
#submeter1Monthly_ts <- ts(submetersByMonth[,3], start = c(2007, 1), frequency = 12)
#submeter1Quarterly_ts <- ts(submetersByQuarter[,3], start = c(2007, 1), frequency = 4)
```

[Hide](#)

```
#Create time objects of submeter_2
laundryRoomHourly_ts <- ts(submetersByHour[,4], start = c(2006, 12), frequency = 8760)
#submeter2Weekly_ts <- ts(submetersByWeek[,4], start = c(2007, 1), frequency = 52)
#submeter2Monthly_ts <- ts(submetersByMonth[,4], start = c(2007, 1), frequency = 12)
#submeter2Quarterly_ts <- ts(submetersByQuarter[,4], start = c(2007, 1), frequency = 4)
```

[Hide](#)

```
#Create time objects of submeter_3
waterHeater_AC_ts <- ts(submetersByHour[,5], start = c(2007, 1), frequency = 8760)
#submeter3Weekly_ts <- ts(submetersByWeek[,5], start = c(2007, 1), frequency = 52)
#submeter3Monthly_ts <- ts(submetersByMonth[,5], start = c(2007, 1), frequency = 12)
#submeter3Quarterly_ts <- ts(submetersByQuarter[,5], start = c(2007, 1), frequency = 4)
```

[Hide](#)

dataframes for forecasting

```
# Every monday at 8:00pm
house070809weekly <- filter(df2007_2009, weekday == "Monday" & hour == 20 & minute == 1)
tsSM1_070809weekly <- ts(house070809weekly$Sub_metering_1, frequency=52, start=c(2007,1))
tsSM2_070809weekly <- ts(house070809weekly$Sub_metering_2, frequency=52, start=c(2007,1))
tsSM3_070809weekly <- ts(house070809weekly$Sub_metering_3, frequency=52, start=c(2007,1))
```

dataframes for decomposing

submeter 1, mornings, 2007,2008,2009 only (full time periods only)

```
sub1ByMorning070809 <- df2006_2010_timeofDay %>% filter(year != 2006 & year != 2010 , TimeOfDay == "1. morning")
sub1ByMorning0809 <- df2006_2010_timeofDay %>% filter(year != 2006 & year != 2007 & year != 2010, TimeOfDay == "1. morning")
```

convert to time series

```
sub1ByMorning070809_ts <- ts(sub1ByMorning070809 Kitchen, frequency = 366, start = c(2007, 1))
sub1ByMorning0809_ts <- ts(sub1ByMorning0809 Kitchen, frequency=363, start=c(2008,1))
```

Mornings, by week, instead of day

```
subsByWeek070809 <- df2006_2010_timeofDay %>% filter(year != 2006 & year != 2010 , TimeOfDay == "1. morning") %>% group_by(year,week) %>% summarize(Kitchen = sum(Kitchen), LaundryRoom = sum(LaundryRoom), WaterHeaterAC =sum(WaterHeaterAC))
```

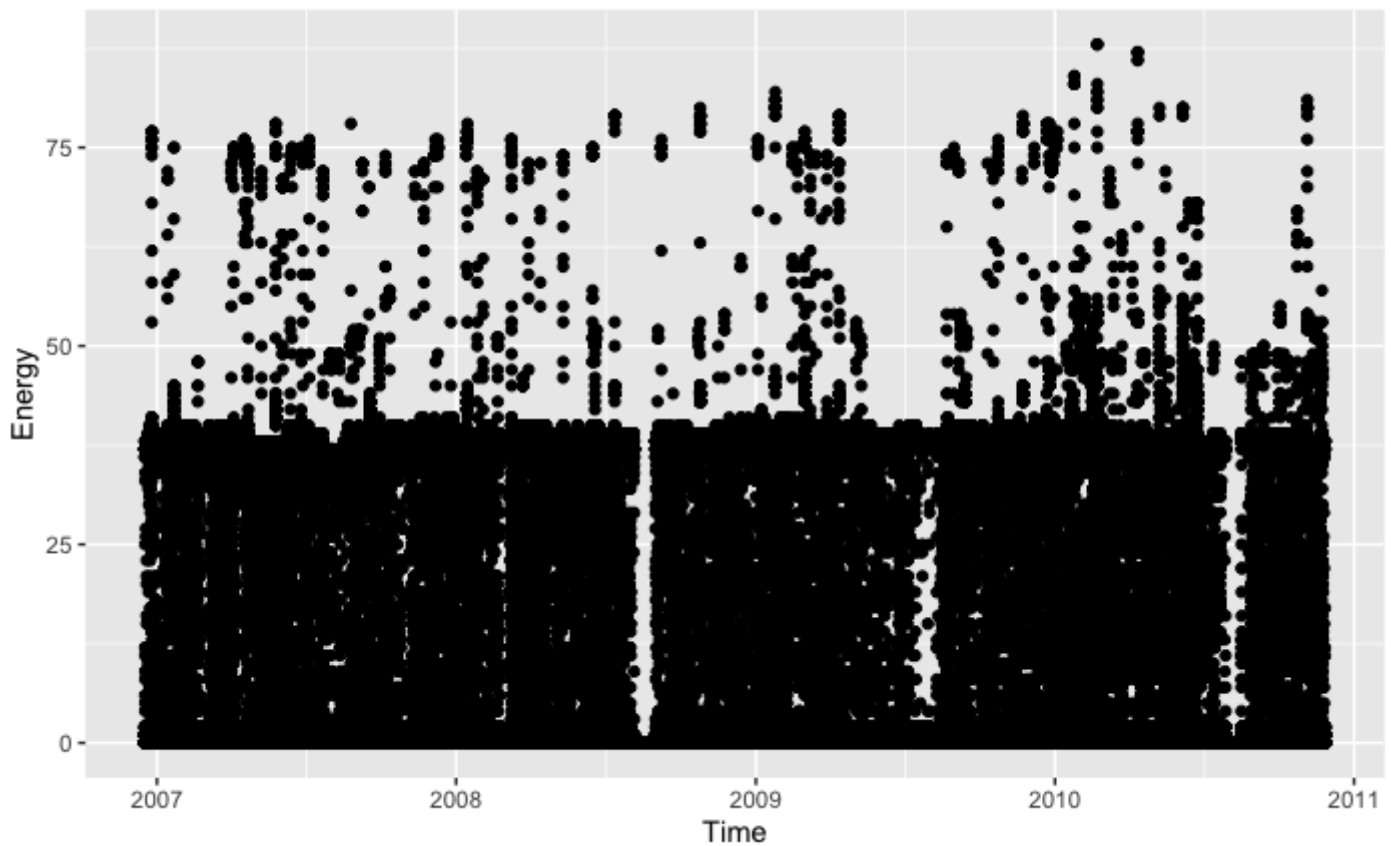
Mornings, by week, as time series

```
sub1ByWeek070809_ts <- ts(subsByWeek070809 Kitchen, frequency = 52, start = c(2007, 1))
sub2ByWeek070809_ts <- ts(subsByWeek070809 LaundryRoom, frequency=52, start=c(2007,1))
sub3ByWeek070809_ts <- ts(subsByWeek070809 WaterHeaterAC, frequency=52, start=c(2007,1))
```

Energy over Time for each Submeter (and Frequency)

Submeter 1

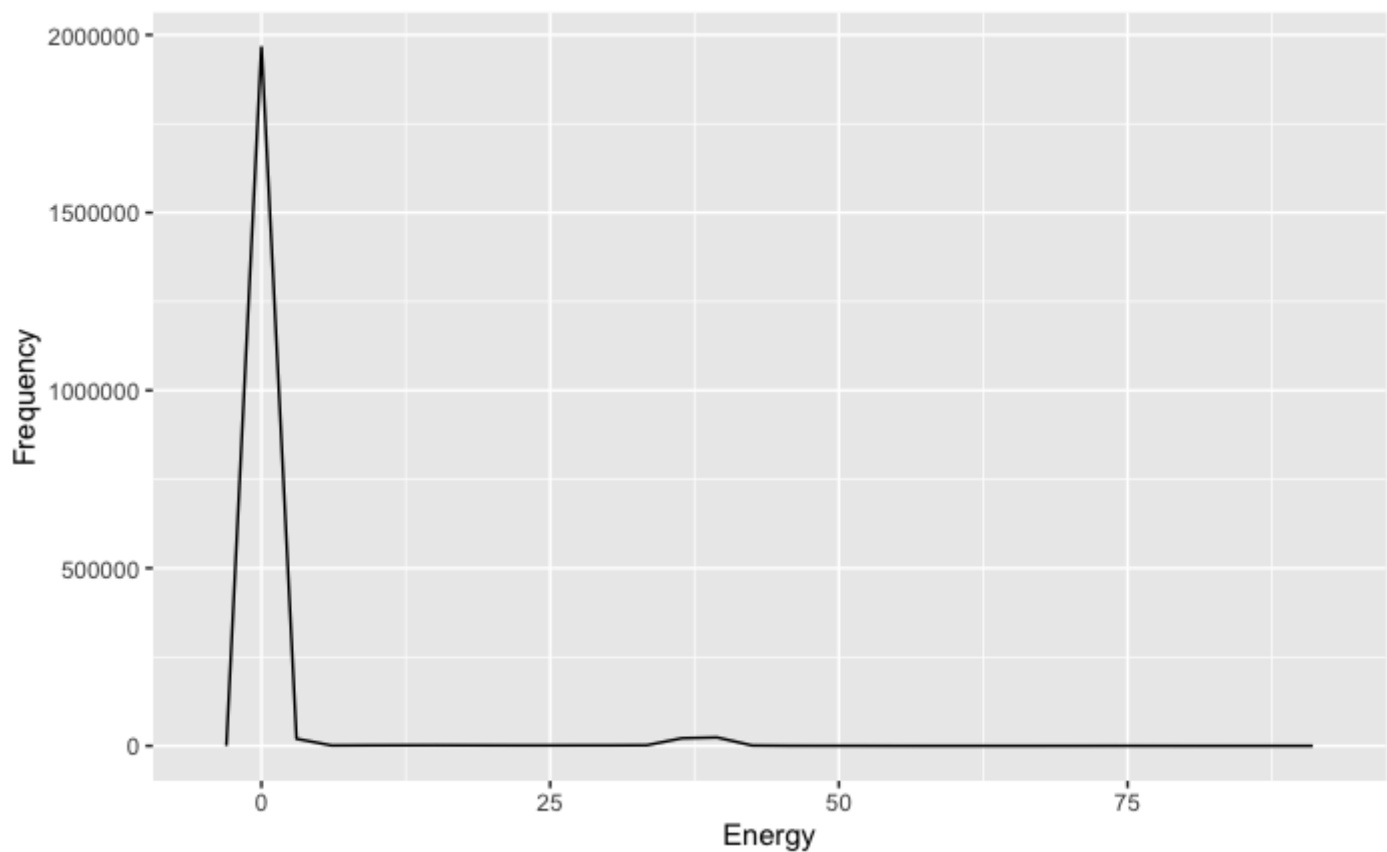
```
# Submeter 1 Over Time
ggplot(data=df2006_2010, aes(x=DateTime, y=Sub_metering_1)) + geom_point()+ylab("Energy")
)+ xlab("Time")
```



Most observations are between 0 and 40, with occasional high usage. There are two primary breaks in the data, where there is minimum usage. Perhaps, this is vacation time.

Hide

```
# Frequency Plot
ggplot(data=df2006_2010, aes(x=Sub_metering_1)) + geom_freqpoly()+ylab("Frequency")+ xlab("Energy")
```


[Hide](#)

```
# Frequency Table for Submeter 1
frequency_submeter1 <- data.frame(table(df2006_2010$Sub_metering_1))
names(frequency_submeter1)[names(frequency_submeter1) == "Var1"] <- "Energy"
names(frequency_submeter1)[names(frequency_submeter1) == "Freq"] <- "Frequency"
arrange(frequency_submeter1, -frequency_submeter1$Frequency)
```

Energy <fctr>	Frequency <int>
0	1880175
1	84936
2	19017
38	16119
37	14892
39	6503
36	5270
35	1359
40	1159
32	802

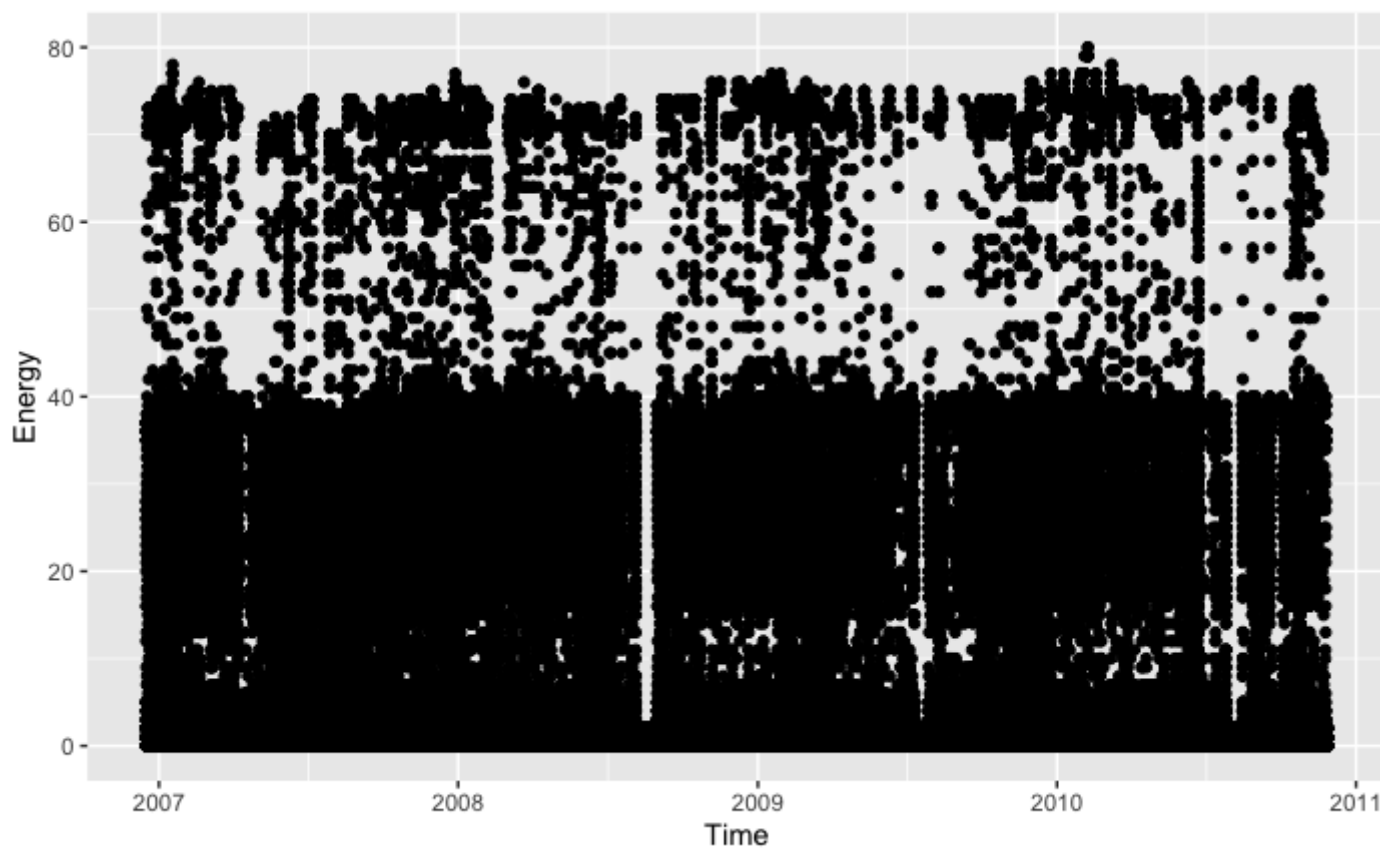
1-10 of 88 rows

Previous **1** 2 3 4 5 6 ... 9 Next

Submeter 2

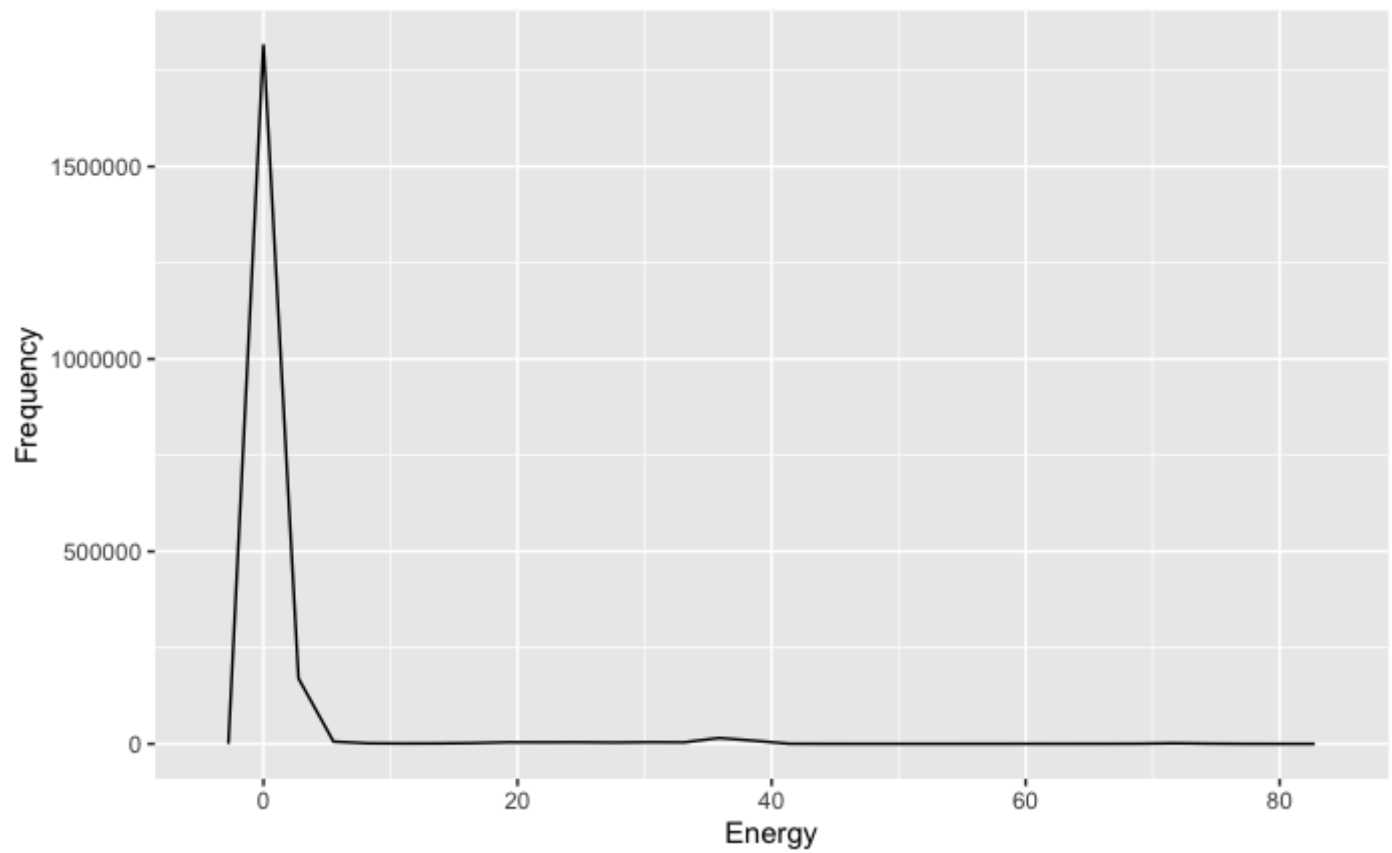
Hide

```
# Submeter 2 Over Time
ggplot(data=df2006_2010, aes(x=DateTime, y=Sub_metering_2)) + geom_point()+ylab("Energy")
)+ xlab("Time")
```



Hide

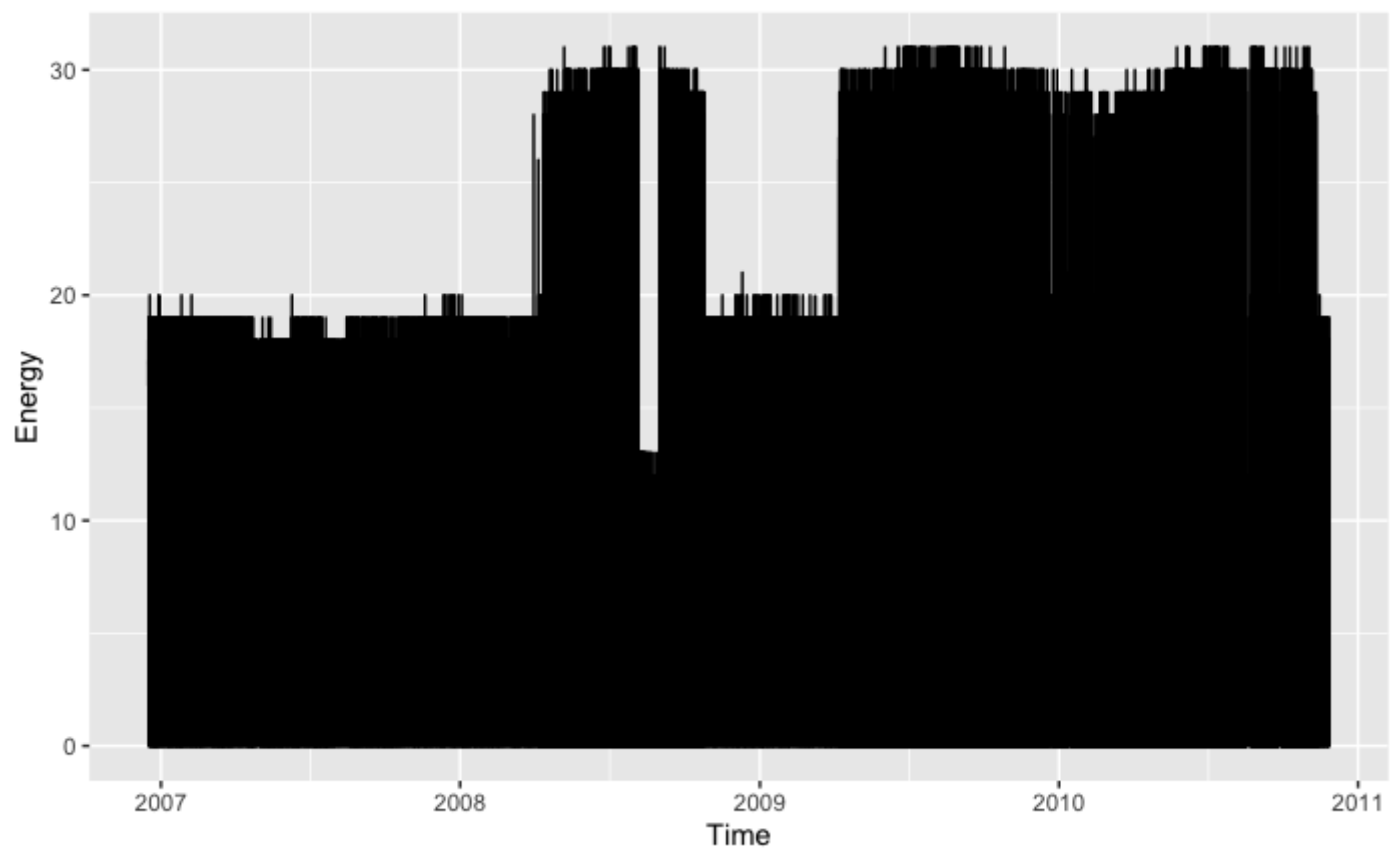
```
# Frequency
ggplot(data=df2006_2010, aes(x=Sub_metering_2)) + geom_freqpoly()+ylab("Frequency")+ xlab("Energy")
```



Submeter 3

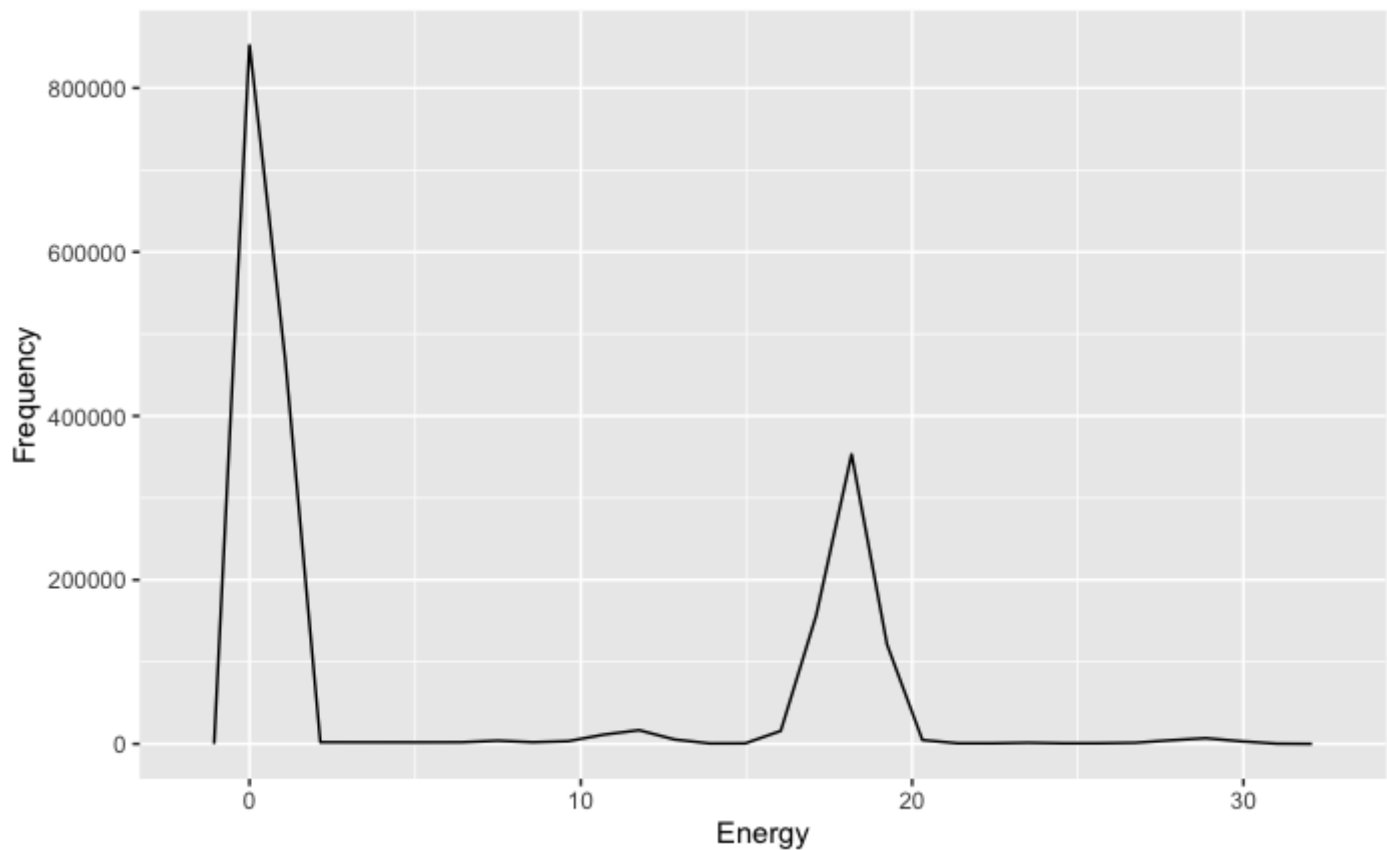
Hide

```
# Submeter 2 Over Time
ggplot(data=df2006_2010, aes(x=DateTime, y=Sub_metering_3)) + geom_line()+ylab("Energy")
+ xlab("Time")
```



Hide

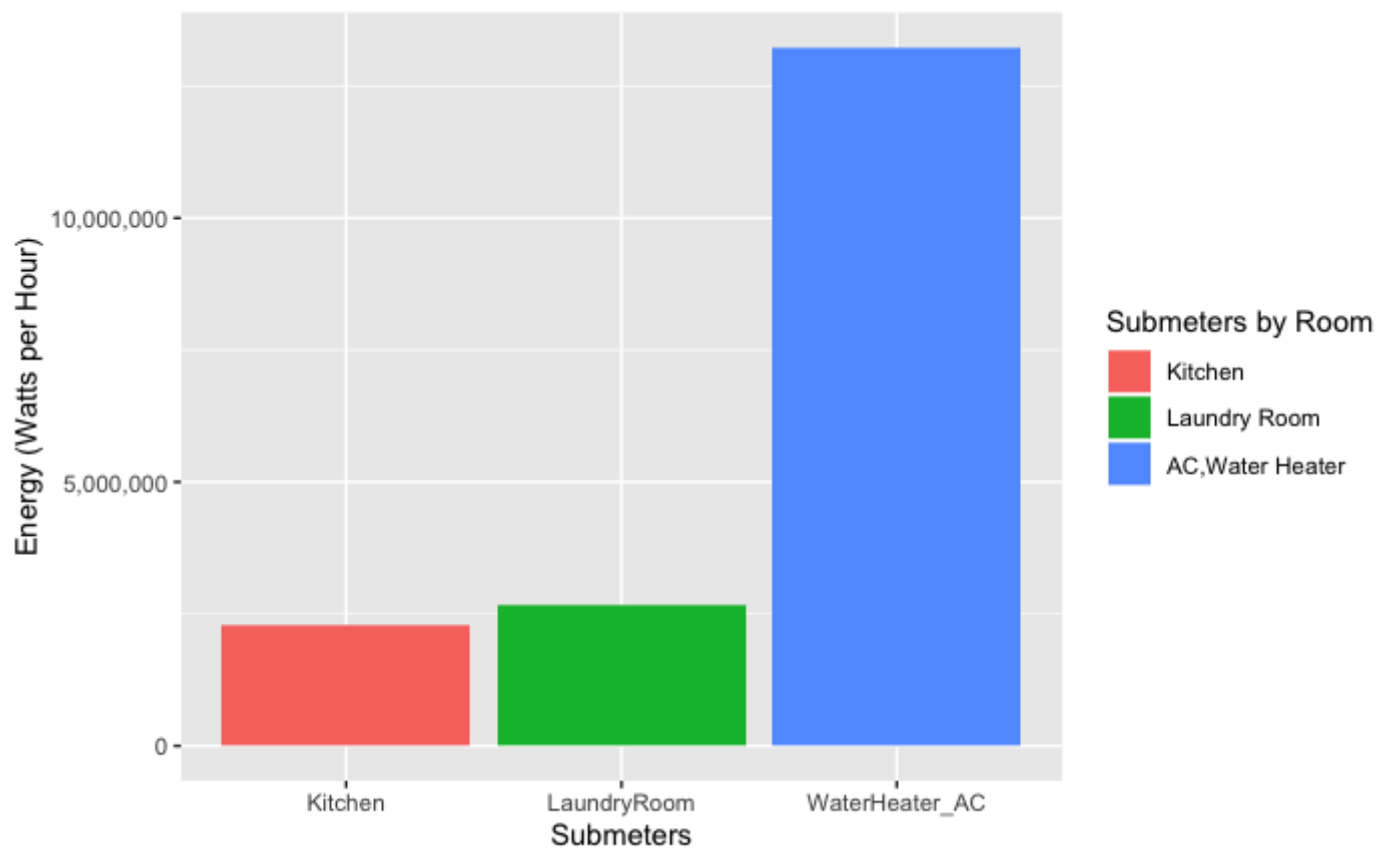
```
# Frequency
ggplot(data=df2006_2010, aes(x=Sub_metering_3)) + geom_freqpoly()+ylab("Frequency")+ xlab("Energy")
```

Total Energy Consumption for Each Submeter

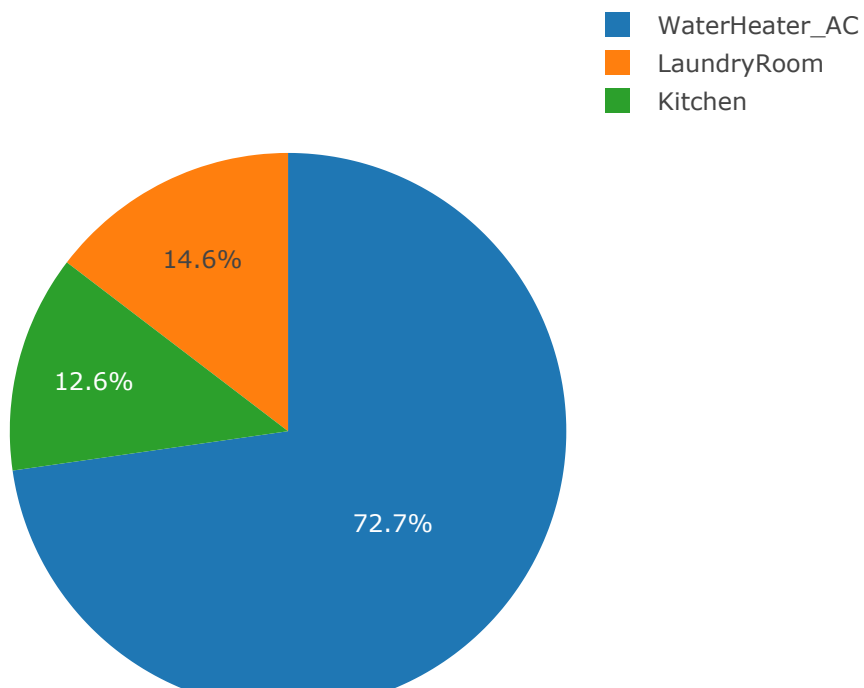
Hide

```
# Plot
ggplot(data = sum_of_submeters_long, aes(x = Submeters, y = Total_Energy_Usage, fill = S
ubmeters)) +
  geom_col()+
  scale_y_continuous(label=comma)+
  ylab("Energy (Watts per Hour)")+
  scale_fill_discrete(name = "Submeters by Room", labels = c("Kitchen", "Laundry Room",
"AC,Water Heater"))
```



Hide

```
# pie chart of total usage
plot_ly(sum_of_submeters_long, labels = ~Submeters, values = ~Total_Energy_Usage, type =
'pie') %>%
  layout(title = '',
    xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
    yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
```



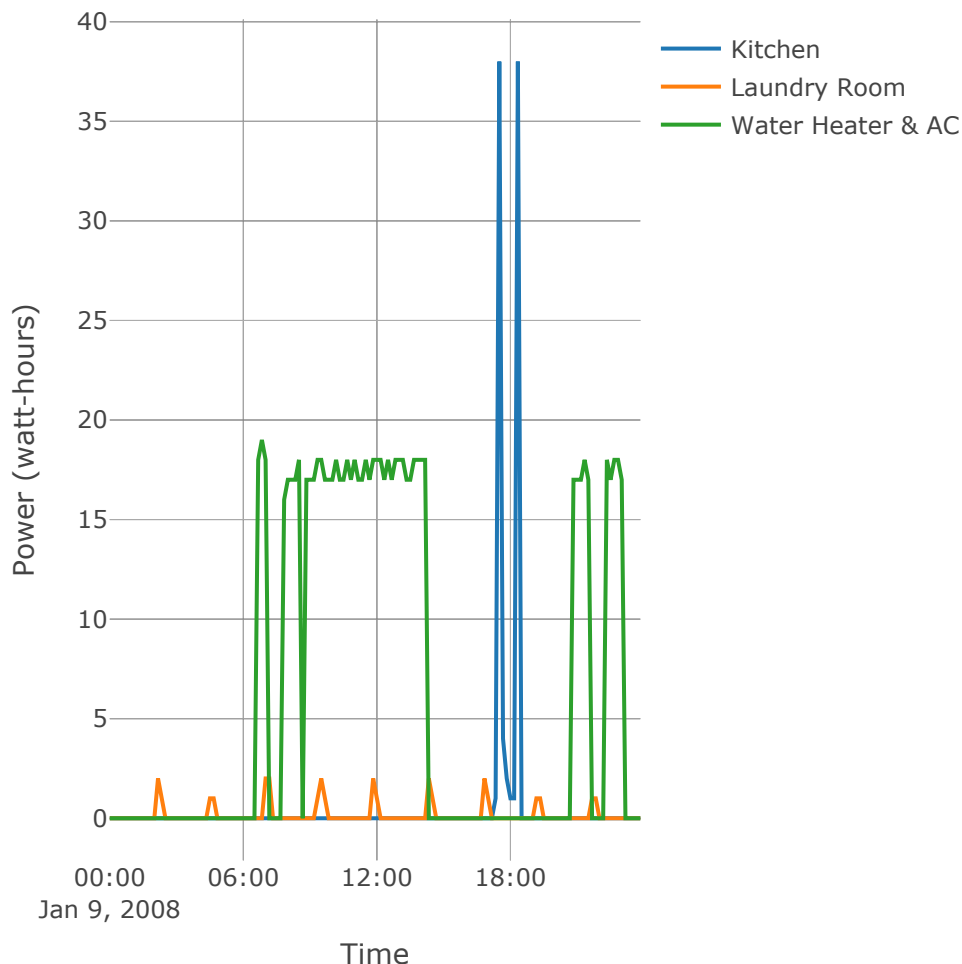
Exploring Different Intervals in 2008

Which time interval is best for addressing granularity?

Day of January 9, 2008

Hide

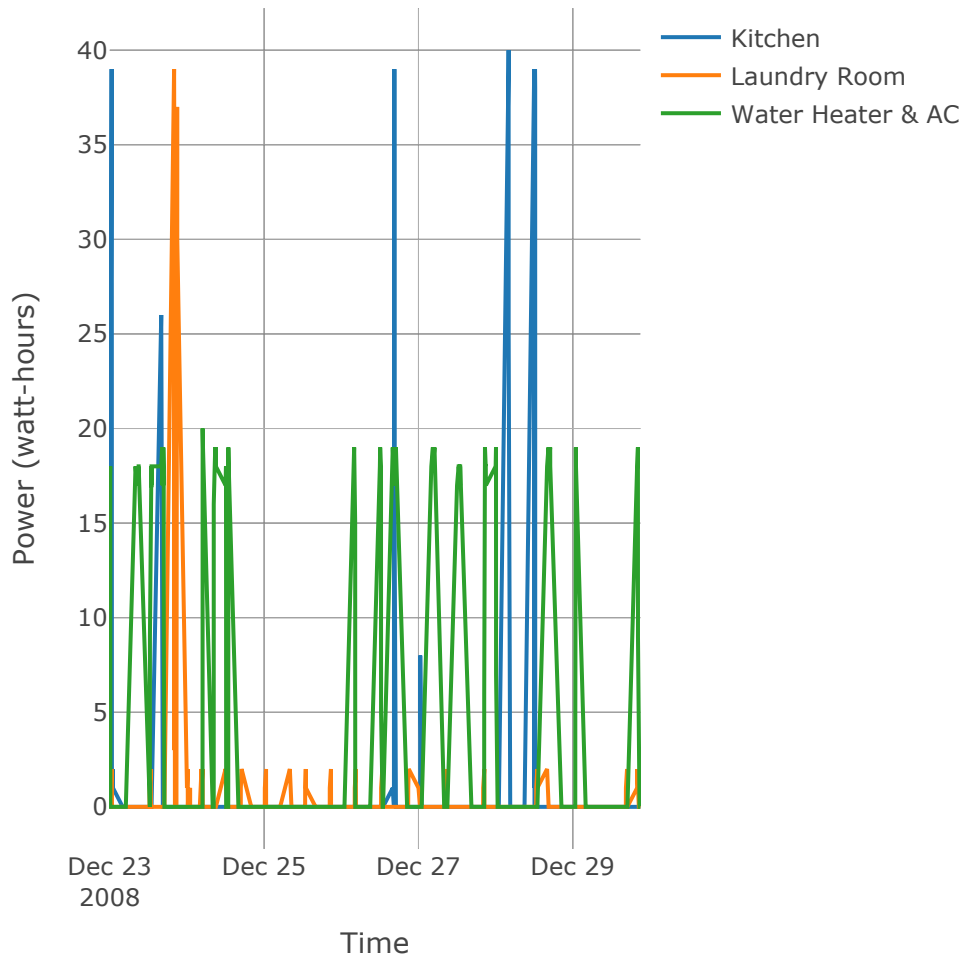
```
## Plot 2008/01/08- Submeters, 10-minute Frequencies
plot_ly(houseDay20080109_10min, x = ~houseDay20080109_10min$DateTime, y = ~houseDay20080109_10min$Sub_metering_1, name = 'Kitchen', type = 'scatter', mode = 'lines') %>%
  add_trace(y = ~houseDay20080109_10min$Sub_metering_2, name = 'Laundry Room', mode = 'lines') %>%
  add_trace(y = ~houseDay20080109_10min$Sub_metering_3, name = 'Water Heater & AC', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list(title = "Power (watt-hours)"))
```



Christmas Week, 2008

[Hide](#)

```
#Plot Week 52 in 2008, every 4 hours
plot_ly(week52_2018_4hours, x = week52_2018_4hours$DateTime, y = ~week52_2018_4hours$Sub_metering_1, name = 'Kitchen', type = 'scatter', mode = 'lines') %>%
  add_trace(y = ~week52_2018_4hours$Sub_metering_2, name = 'Laundry Room', mode = 'lines'
) %>%
  add_trace(y = ~week52_2018_4hours$Sub_metering_3, name = 'Water Heater & AC', mode = 'lines') %>%
  layout( xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))
```

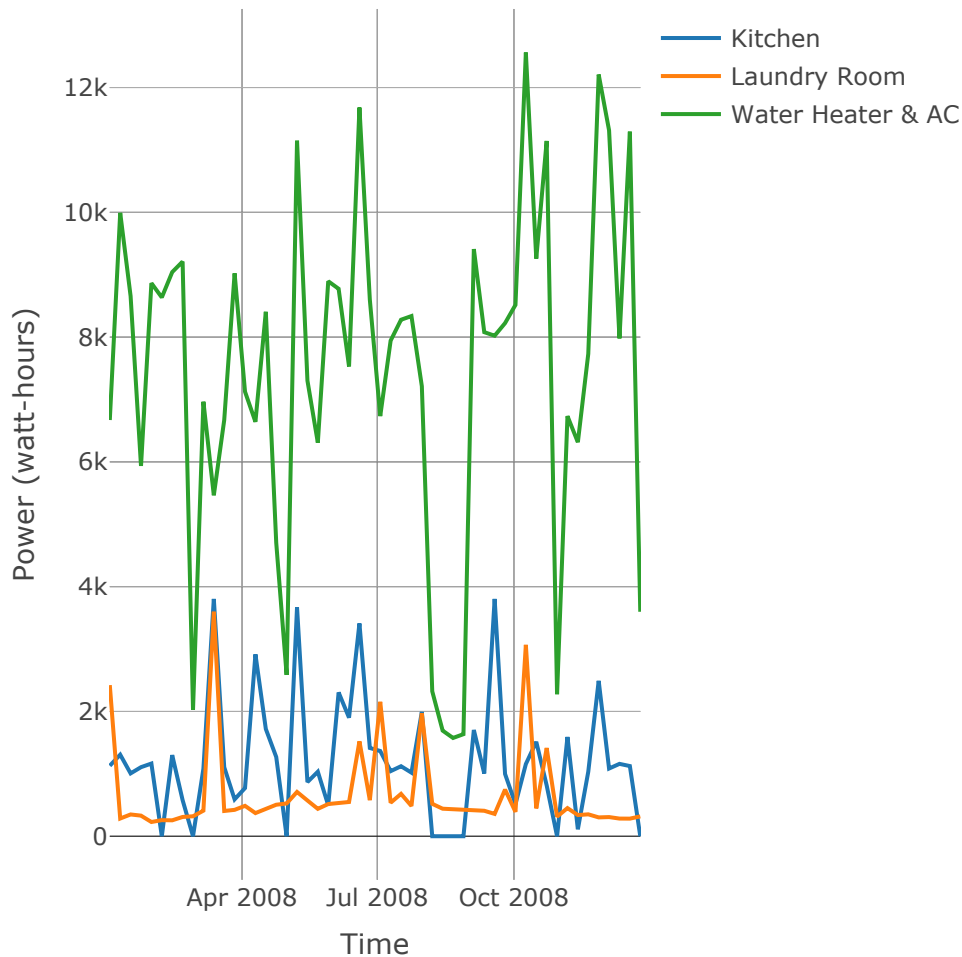


Weekdays in 2008

Thursdays

[Hide](#)

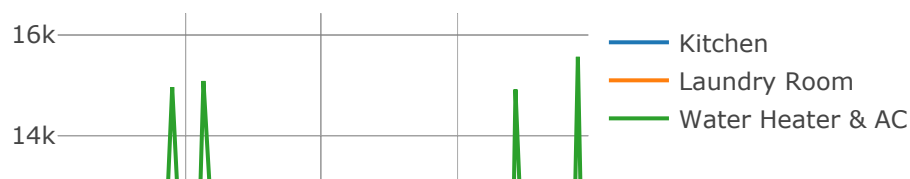
```
#Plot Thursdays in 2008
plot_ly(thursdays08, x = thursdays08$date, y = ~thursdays08$Kitchen, name = 'Kitchen', type = 'scatter', mode = 'lines') %>%
  add_trace(y = ~thursdays08$LaundryRoom, name = 'Laundry Room', mode = 'lines') %>%
  add_trace(y = ~thursdays08$WaterHeaterAC, name = 'Water Heater & AC', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list(title = "Power (watt-hours)"))
```

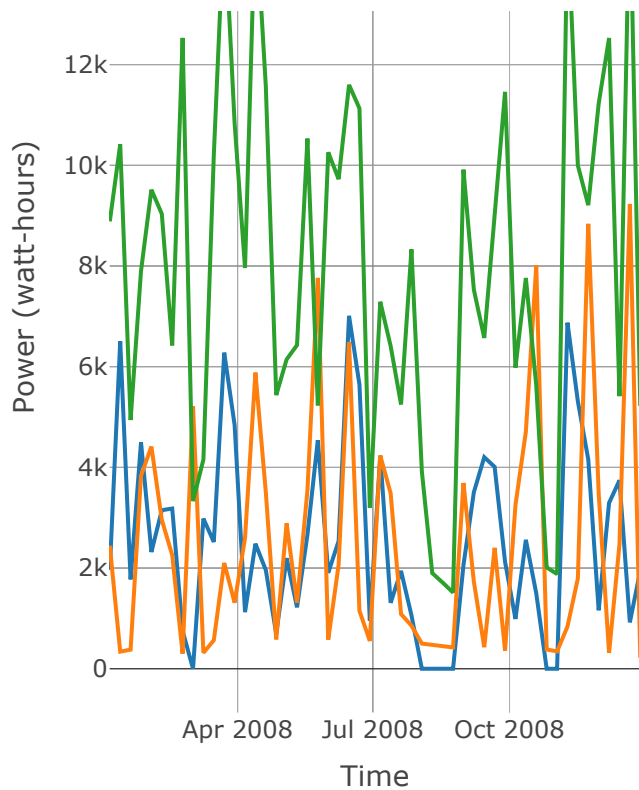


sundays

Hide

```
#Plot sundays in 2008
plot_ly(sundays08, x = sundays08$date, y = ~sundays08$Kitchen, name = 'Kitchen', type = 'scatter', mode = 'lines') %>%
  add_trace(y = ~sundays08$LaundryRoom, name = 'Laundry Room', mode = 'lines') %>%
  add_trace(y = ~sundays08$WaterHeaterAC, name = 'Water Heater & AC', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list(title = "Power (watt-hours)"))
```





Weekdays All Years

Submeter 1

Hide

```
#make individual weekday plots for Submeter 1
mondays1_plot <- plot_ly(mondays, x = mondays$date, y = ~mondays$Kitchen, name = 'mondays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

tuesdays1_plot <- plot_ly(tuesdays, x = tuesdays$date, y = ~tuesdays$Kitchen, name = 'tuesdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

wednesdays1_plot <- plot_ly(wednesdays, x = wednesdays$date, y = ~wednesdays$Kitchen, name = 'wednesdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

thursdays1_plot <- plot_ly(thursdays, x = thursdays$date, y = ~thursdays$Kitchen, name = 'Thursdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

fridays1_plot <- plot_ly(fridays, x = fridays$date, y = ~fridays$Kitchen, name = 'fridays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

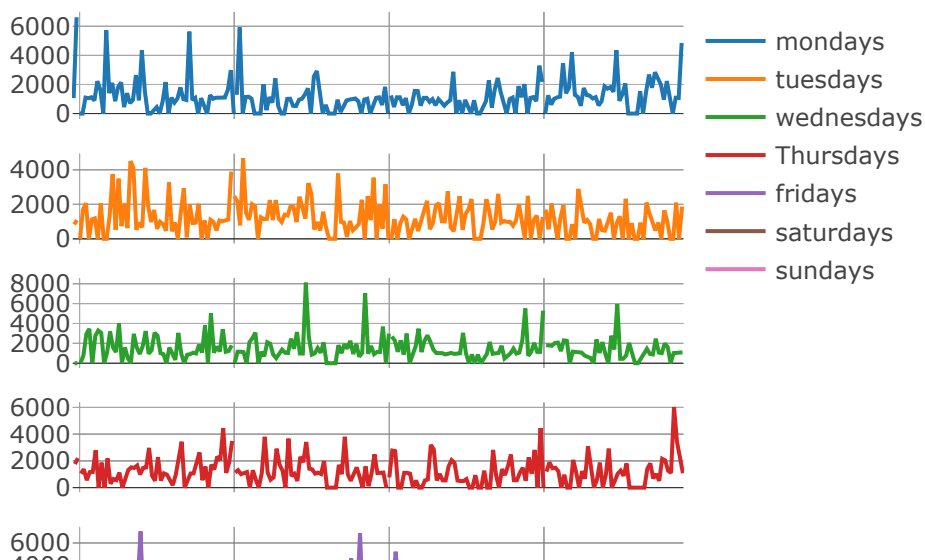
saturdays1_plot <- plot_ly(saturdays, x = saturdays$date, y = ~saturdays$Kitchen, name = 'saturdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

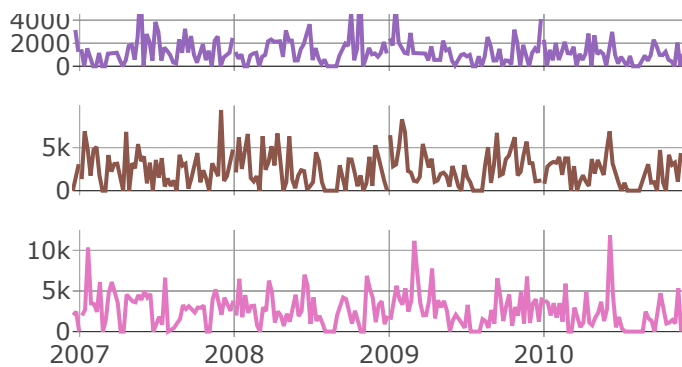
sundays1_plot <- plot_ly(sundays, x = sundays$date, y = ~sundays$Kitchen, name = 'sundays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))
```

Hide

```
weekdays_plot_kitchen <- subplot(mondays1_plot, tuesdays1_plot, wednesdays1_plot, thursdays1_plot, fridays1_plot, saturdays1_plot, sundays1_plot, nrow = 7, shareX = TRUE, shareY = TRUE, titleY = FALSE, titleX = FALSE)
```

```
weekdays_plot_kitchen
```





Submeter 2

Hide

```
#make individual weekday plots for Submeter 2
mondays2_plot <- plot_ly(mondays, x = mondays$date, y = ~mondays$LaundryRoom, name = 'mondays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

tuesdays2_plot <- plot_ly(tuesdays, x = tuesdays$date, y = ~tuesdays$LaundryRoom, name = 'tuesdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

wednesdays2_plot <- plot_ly(wednesdays, x = wednesdays$date, y = ~wednesdays$LaundryRoom, name = 'wednesdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

thursdays2_plot <- plot_ly(thursdays, x = thursdays$date, y = ~thursdays$LaundryRoom, name = 'Thursdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

fridays2_plot <- plot_ly(fridays, x = fridays$date, y = ~fridays$LaundryRoom, name = 'fridays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

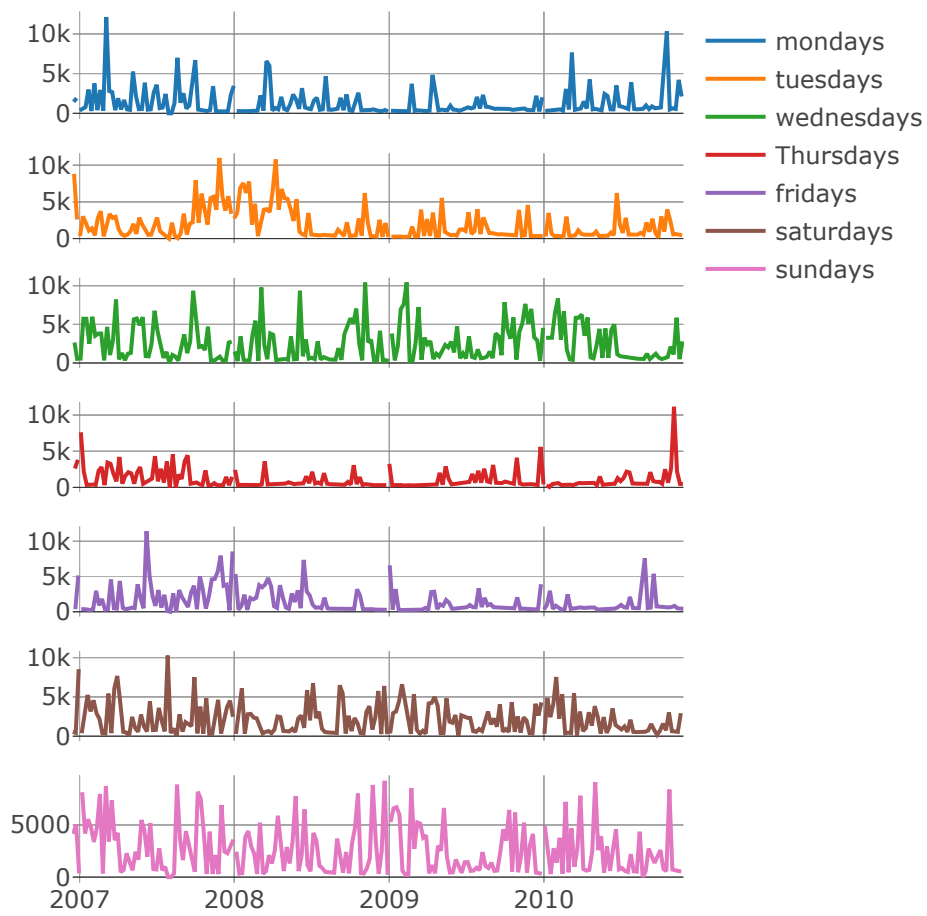
saturdays2_plot <- plot_ly(saturdays, x = saturdays$date, y = ~saturdays$LaundryRoom, name = 'saturdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

sundays2_plot <- plot_ly(sundays, x = sundays$date, y = ~sundays$LaundryRoom, name = 'sundays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))
```

Hide

```
weekdays_plot_LaundryRoom <- subplot(mondays2_plot, tuesdays2_plot, wednesdays2_plot, thursdays2_plot, fridays2_plot, saturdays2_plot, sundays2_plot, nrows = 7, shareX = TRUE, shareY = TRUE, titleY = FALSE, titleX = FALSE)

weekdays_plot_LaundryRoom
```

Submeter 3

Hide

```
#make individual weekday plots for Submeter 3
mondays3_plot <- plot_ly(mondays, x = mondays$date, y = ~mondays$WaterHeaterAC, name =
'mondays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

tuesdays3_plot <- plot_ly(tuesdays, x = tuesdays$date, y = ~tuesdays$WaterHeaterAC, name =
'tuesdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

wednesdays3_plot <- plot_ly(wednesdays, x = wednesdays$date, y = ~wednesdays$WaterHeater
AC, name = 'wednesdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"),yaxis = list (title = "Power (watt-hours)"))

thursdays3_plot <- plot_ly(thursdays, x = thursdays$date, y = ~thursdays$WaterHeaterAC,
name = 'Thursdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"),yaxis = list (title = "Power (watt-hours)"))

fridays3_plot <- plot_ly(fridays, x = fridays$date, y = ~fridays$WaterHeaterAC, name =
'fridays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"),yaxis = list (title = "Power (watt-hours)"))

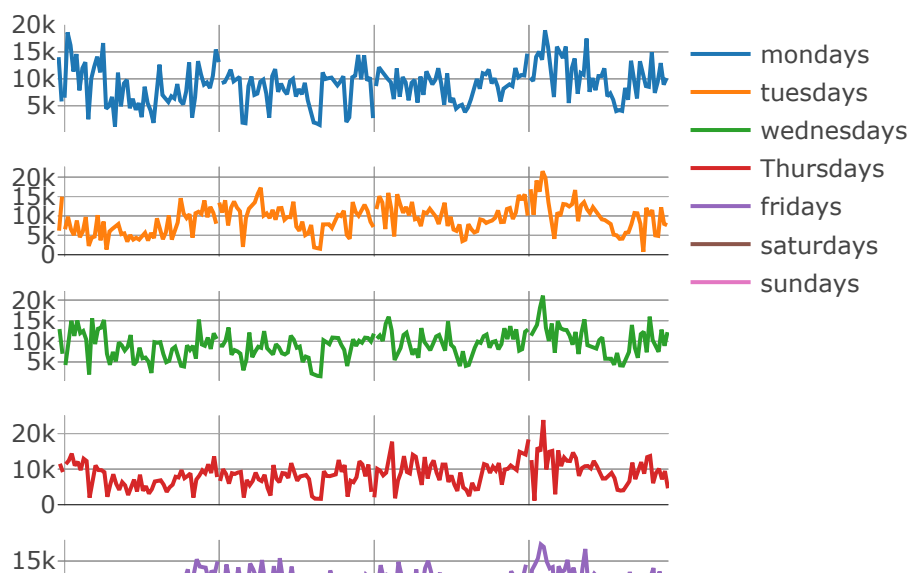
saturdays3_plot <- plot_ly(saturdays, x = saturdays$date, y = ~saturdays$WaterHeaterAC,
name = 'saturdays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"), yaxis = list (title = "Power (watt-hours)"))

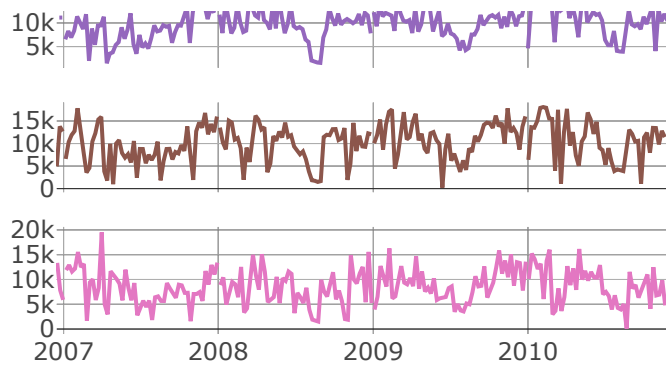
sundays3_plot <- plot_ly(sundays, x = sundays$date, y = ~sundays$WaterHeaterAC, name =
'sundays', type = 'scatter', mode = 'lines') %>%
  layout(xaxis = list(title = "Time"),yaxis = list (title = "Power (watt-hours)"))
```

Hide

```
weekdays_plot_WaterHeaterAC <- subplot(mondays3_plot, tuesdays3_plot, wednesdays3_plot,
thursdays3_plot, fridays3_plot, saturdays3_plot, sundays3_plot, nrows =7, shareX = TRUE,
shareY = TRUE, titleY = FALSE, titleX = FALSE)
```

```
weekdays_plot_WaterHeaterAC
```



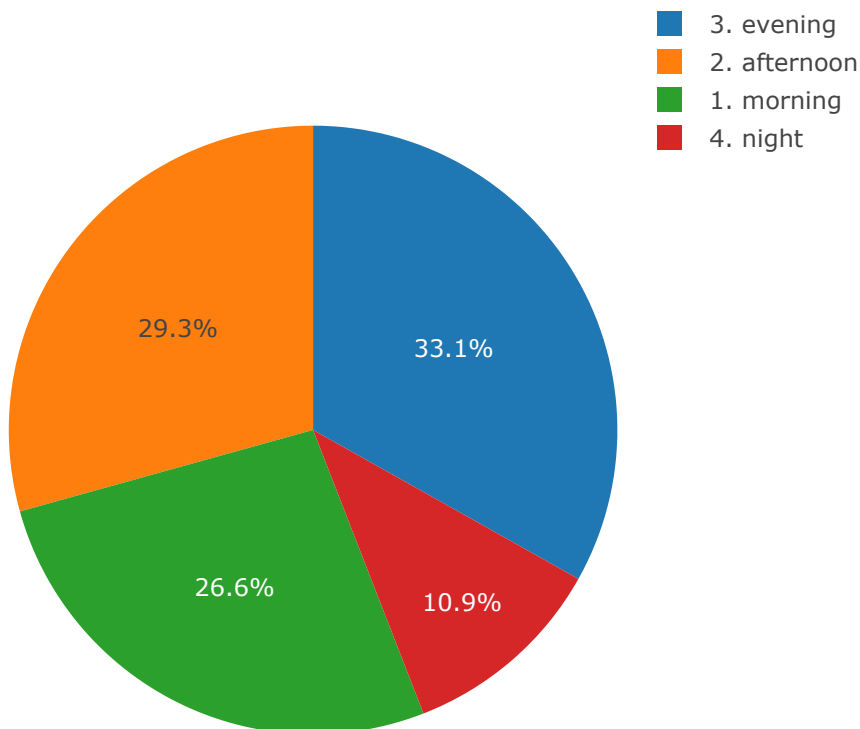


Time of Day Visualizations

Total Usage by Time of Day

Hide

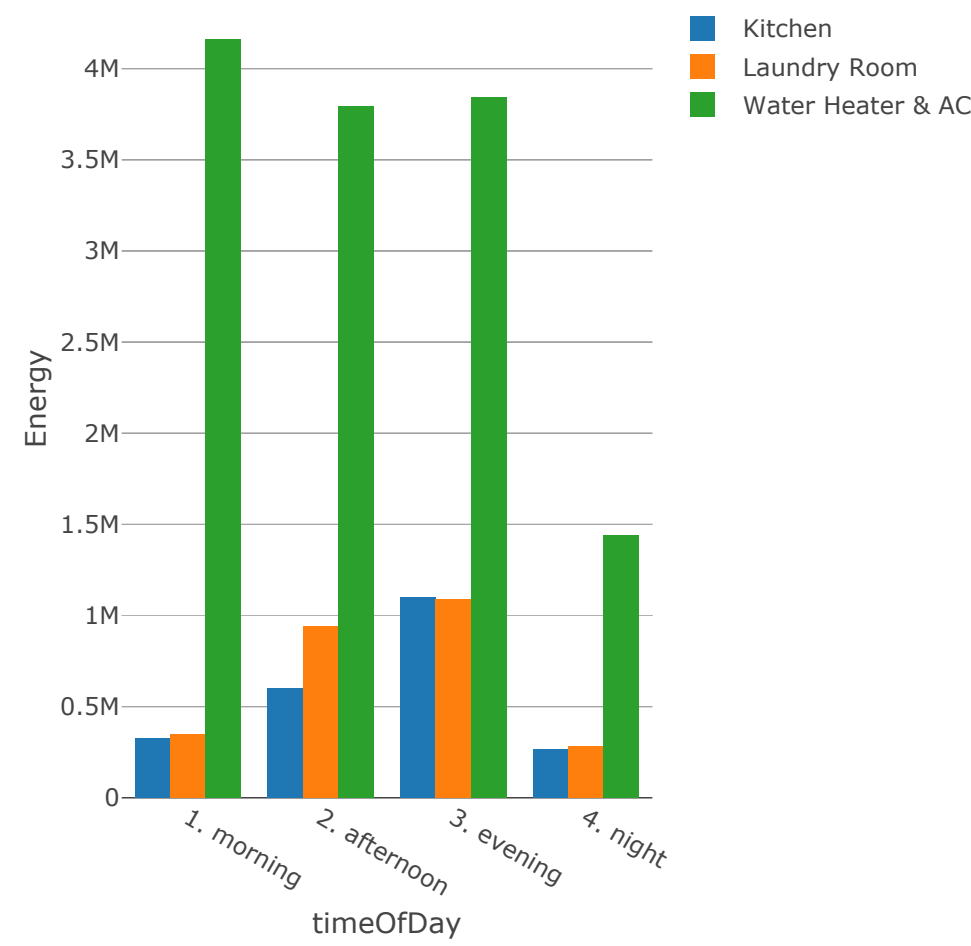
```
# pie chart "TimeOfDay Total Usage"
plot_ly(allYearsTimeofDayTotals, labels = ~timeOfDay, values = ~totalEnergy, type = 'pie') %>%
  layout(title = '',
    xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
    yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE))
```



Total Usage by Time of Day, and Submeter

Hide

```
# bar chart for Time of Day, Total Usage, by each submeter
plot_ly(allYearsTimeOfDayTotals, x = ~timeOfDay, y = ~Kitchen, type = 'bar', name = 'Kitchen') %>%
  add_trace(y = ~LaundryRoom, name = 'Laundry Room') %>%
  add_trace(y = ~WaterHeaterAC, name = 'Water Heater & AC') %>%
  layout(yaxis = list(title = 'Energy'), barmode = 'group')
```



Total Usage by Weekday

Hide

```
# View dataframe for weekday, 2006-2010, Sum of All Usage
weekdaysAllYearsTotals
```

weekday<chr>	Kitchen<dbl>	LaundryRoom<dbl>	WaterHeaterAC<dbl>	totalEnergy<dbl>
Friday	254548	299050	2021938	2575536
Monday	235899	265556	1842356	2343811

weekday <chr>	Kitchen <dbl>	LaundryRoom <dbl>	WaterHeaterAC <dbl>	totalEnergy <dbl>
Saturday	490027	433751	2098915	3022693
Sunday	530394	536137	1739338	2805869
Thursday	244990	220302	1760543	2225835
Tuesday	243890	386935	1896362	2527187
Wednesday	299387	519300	1875715	2694402

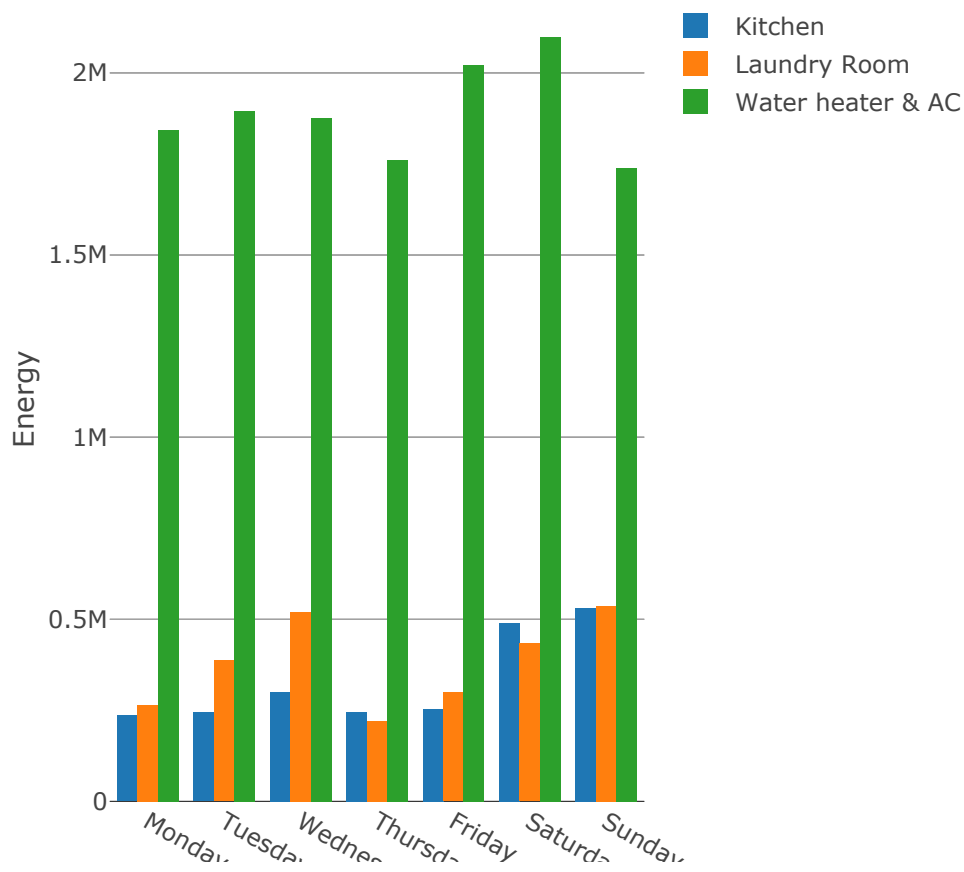
7 rows

Hide

```
# Reorder Weekday column
weekdaysAllYearsTotals$weekday <- factor(weekdaysAllYearsTotals$weekday, levels = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))
```

Hide

```
# plot bar graph grouped by time of day, stratified with the 3 submeters
plot_ly(weekdaysAllYearsTotals, x = ~weekday, y = ~Kitchen, type = 'bar', name = 'Kitchen') %>%
  add_trace(y = ~LaundryRoom, name = 'Laundry Room') %>%
  add_trace(y = ~WaterHeaterAC, name = 'Water heater & AC') %>%
  layout(yaxis = list(title = 'Energy'), barmode = 'group')
```



~y ~y ~sday ~dy . ~dy ~y
weekday