

# SCE Usage

*Oliver de Pug*

## Read Data

Import weather data.

```
weather <- read_csv("NOAA_LCD_123110to103118.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   STATION = col_character(),
##   STATION_NAME = col_character(),
##   DATE = col_character(),
##   REPORTTPYE = col_character(),
##   HOURLYSKYCONDITIONS = col_character(),
##   HOURLYVISIBILITY = col_character(),
##   HOURLYPRESENTWEATHERTYPE = col_character(),
##   HOURLYDewPointTempF = col_character(),
##   HOURLYDewPointTempC = col_character(),
##   HOURLYWindSpeed = col_character(),
##   HOURLYWindDirection = col_character(),
##   HOURLYPressureChange = col_logical(),
##   HOURLYPrecip = col_character(),
##   DAILYAverageRelativeHumidity = col_logical(),
##   DAILYAverageDewPointTemp = col_logical(),
##   DAILYAverageWetBulbTemp = col_logical(),
##   DAILYWeather = col_character(),
##   DAILYPrecip = col_character(),
##   DAILYSnowfall = col_logical(),
##   DAILYSnowDepth = col_logical()
##   # ... with 26 more columns
## )

## See spec(...) for full column specifications.

## Warning: 617 parsing failures.
##   row          col          expected actual          file
## 4639 HOURLYDRYBULBTEMPF      no trailing characters      s 'NOAA_LCD_123110to103118.csv'
## 4639 HOURLYDRYBULBTEMPC      no trailing characters      s 'NOAA_LCD_123110to103118.csv'
## 5106 MonthlyDaysWithGT32Temp no trailing characters      s 'NOAA_LCD_123110to103118.csv'
## 5106 MonthlyDaysWithLT0Temp no trailing characters      s 'NOAA_LCD_123110to103118.csv'
## 7214 HOURLYDRYBULBTEMPF      no trailing characters      s 'NOAA_LCD_123110to103118.csv'
## .....
## See problems(...) for more details.
```

```
#names(weather)
head(weather)
```

```
## # A tibble: 6 x 90
##   STATION STATION_NAME ELEVATION LATITUDE LONGITUDE DATE REPORTTPYE
##   <chr>   <chr>           <dbl>   <dbl>   <dbl> <chr> <chr>
## 1 WBAN:0~ ONTARIO INT~      289.    34.1    -118. 12/3~ FM-15
```

```

## 2 WBAN:0~ ONTARIO INT~      289.      34.1      -118. 12/3~ FM-15
## 3 WBAN:0~ ONTARIO INT~      289.      34.1      -118. 12/3~ FM-15
## 4 WBAN:0~ ONTARIO INT~      289.      34.1      -118. 12/3~ FM-15
## 5 WBAN:0~ ONTARIO INT~      289.      34.1      -118. 12/3~ FM-15
## 6 WBAN:0~ ONTARIO INT~      289.      34.1      -118. 12/3~ FM-15
## # ... with 83 more variables: HOURLYSKYCONDITIONS <chr>,
## #   HOURLYVISIBILITY <chr>, HOURLYPRESENTWEATHERTYPE <chr>,
## #   HOURLYDRYBULBTEMPF <dbl>, HOURLYDRYBULBTEMPC <dbl>,
## #   HOURLYWETBULBTEMPF <dbl>, HOURLYWETBULBTEMPC <dbl>,
## #   HOURLYDewPointTempF <chr>, HOURLYDewPointTempC <chr>,
## #   HOURLYRelativeHumidity <dbl>, HOURLYWindSpeed <chr>,
## #   HOURLYWindDirection <chr>, HOURLYWindGustSpeed <dbl>,
## #   HOURLYStationPressure <dbl>, HOURLYPressureTendency <dbl>,
## #   HOURLYPressureChange <lgl>, HOURLYSeaLevelPressure <dbl>,
## #   HOURLYPrecip <chr>, HOURLYAltimeterSetting <dbl>,
## #   DAILYMaximumDryBulbTemp <dbl>, DAILYMinimumDryBulbTemp <dbl>,
## #   DAILYAverageDryBulbTemp <dbl>, DAILYDeptFromNormalAverageTemp <dbl>,
## #   DAILYAverageRelativeHumidity <lgl>, DAILYAverageDewPointTemp <lgl>,
## #   DAILYAverageWetBulbTemp <lgl>, DAILYHeatingDegreeDays <dbl>,
## #   DAILYCoolingDegreeDays <dbl>, DAILYSunrise <dbl>, DAILYSunset <dbl>,
## #   DAILYWeather <chr>, DAILYPrecip <chr>, DAILYSnowfall <lgl>,
## #   DAILYSnowDepth <lgl>, DAILYAverageStationPressure <dbl>,
## #   DAILYAverageSeaLevelPressure <lgl>, DAILYAverageWindSpeed <dbl>,
## #   DAILYPeakWindSpeed <chr>, PeakWindDirection <chr>,
## #   DAILYSustainedWindSpeed <dbl>, DAILYSustainedWindDirection <dbl>,
## #   MonthlyMaximumTemp <dbl>, MonthlyMinimumTemp <dbl>,
## #   MonthlyMeanTemp <dbl>, MonthlyAverageRH <lgl>,
## #   MonthlyDewpointTemp <lgl>, MonthlyWetBulbTemp <lgl>,
## #   MonthlyAvgHeatingDegreeDays <lgl>, MonthlyAvgCoolingDegreeDays <lgl>,
## #   MonthlyStationPressure <dbl>, MonthlySeaLevelPressure <dbl>,
## #   MonthlyAverageWindSpeed <lgl>, MonthlyTotalSnowfall <lgl>,
## #   MonthlyDeptFromNormalMaximumTemp <dbl>,
## #   MonthlyDeptFromNormalMinimumTemp <dbl>,
## #   MonthlyDeptFromNormalAverageTemp <dbl>,
## #   MonthlyDeptFromNormalPrecip <chr>, MonthlyTotalLiquidPrecip <chr>,
## #   MonthlyGreatestPrecip <lgl>, MonthlyGreatestPrecipDate <lgl>,
## #   MonthlyGreatestSnowfall <lgl>, MonthlyGreatestSnowfallDate <lgl>,
## #   MonthlyGreatestSnowDepth <lgl>, MonthlyGreatestSnowDepthDate <lgl>,
## #   MonthlyDaysWithGT90Temp <dbl>, MonthlyDaysWithLT32Temp <chr>,
## #   MonthlyDaysWithGT32Temp <dbl>, MonthlyDaysWithLT0Temp <dbl>,
## #   MonthlyDaysWithGT001Precip <lgl>, MonthlyDaysWithGT010Precip <lgl>,
## #   MonthlyDaysWithGT1Snow <lgl>, MonthlyMaxSeaLevelPressureValue <lgl>,
## #   MonthlyMaxSeaLevelPressureDate <dbl>,
## #   MonthlyMaxSeaLevelPressureTime <dbl>,
## #   MonthlyMinSeaLevelPressureValue <lgl>,
## #   MonthlyMinSeaLevelPressureDate <dbl>,
## #   MonthlyMinSeaLevelPressureTime <dbl>,
## #   MonthlyTotalHeatingDegreeDays <dbl>,
## #   MonthlyTotalCoolingDegreeDays <dbl>,
## #   MonthlyDeptFromNormalHeatingDD <dbl>,
## #   MonthlyDeptFromNormalCoolingDD <dbl>,
## #   MonthlyTotalSeasonToDateHeatingDD <lgl>,
## #   MonthlyTotalSeasonToDateCoolingDD <lgl>

```

Import SCE data.

```
sce060116to073017 <- read_csv("SCE060116to073017.csv", skip=13)

## Parsed with column specification:
## cols(
##   `Energy consumption time period` = col_character(),
##   `Usage(Real energy in kilowatt-hours)` = col_character(),
##   `Reading quality` = col_character()
## )

sce061517to071218 <- read_csv("SCE061517to071218.csv", skip=13)

## Parsed with column specification:
## cols(
##   `Energy consumption time period` = col_character(),
##   `Usage(Real energy in kilowatt-hours)` = col_character(),
##   `Reading quality` = col_character()
## )

sce090117to092618 <- read_csv("SCE090117to092618.csv", skip=13)

## Parsed with column specification:
## cols(
##   `Energy consumption time period` = col_character(),
##   `Usage(Real energy in kilowatt-hours)` = col_character(),
##   `Reading quality` = col_character()
## )

SCE <- unique(bind_rows(sce060116to073017, sce061517to071218, sce090117to092618))
table(is.na(SCE[,1]), is.na(SCE[,2]))

##
##          FALSE  TRUE
## FALSE 20352   848
##  TRUE     0     1
```

## Select and Create Variables

Use **stringr** and **lubridate** to create date and time variables from the combined information.

```
### Shorten names to make them easier to type
names(SCE)

## [1] "Energy consumption time period"
## [2] "Usage(Real energy in kilowatt-hours)"
## [3] "Reading quality"

names(SCE) <- c("UsagePeriod", "UsageKWH", "ReadQual")

### Convert character date to POSIXct and get rid of ReadQual=NA
SCE <- SCE %>%
  filter(!is.na(UsageKWH)) %>% # Dump obs with no usage information
  rowwise() %>% # rowwise() takes care of our working with a list-variable
  mutate(
    UsageKWH = as.numeric(UsageKWH), # Force to numeric
    Period = str_sub(UsagePeriod, 1, 19), # Pull the starting date-time
    Date = as.POSIXct(Period, format="%Y-%m-%d %H:%M:%S") #Convert to POSIXct
```

```
) %>%
  select(Date, UsageKWH) # Dump unwanted variables
```

## Warning: NAs introduced by coercion

Now set up a date and time for the **weather** data.

```
weather <- weather %>%
  filter(!is.na(HOURLYDRYBULBTEMPF)) %>%
  rowwise() %>%
  mutate(Date = floor_date(as.POSIXct(DATE, format="%m/%d/%Y %H:%M"), "hour")
) %>%
  select(Date, DryTemp=HOURLYDRYBULBTEMPF, Precip=HOURLYPrecip, WindSpeed=HOURLYWindSpeed, Humidity=H
```

## Combine Usage and Weather Information

```
Electricity <- SCE %>%
  left_join(weather, by="Date")
head(Electricity)
```

## Source: local data frame [6 x 7]

## Groups: <by row>

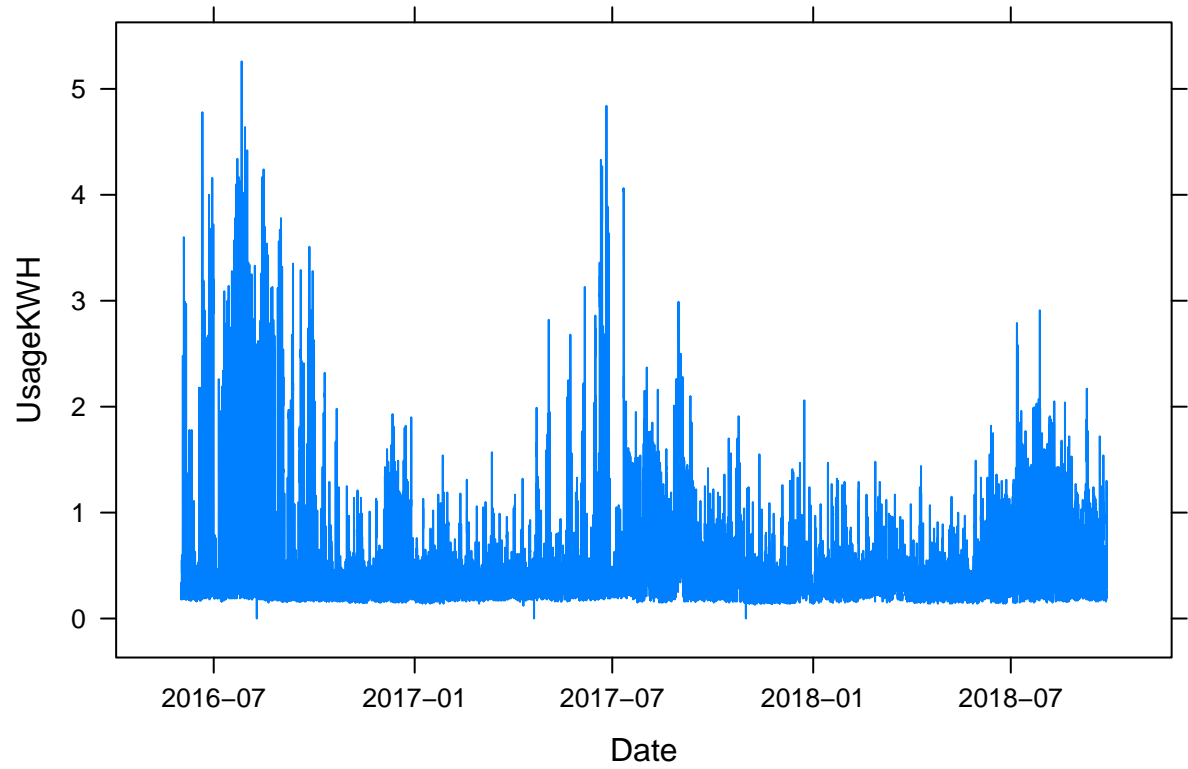
##

## # A tibble: 6 x 7

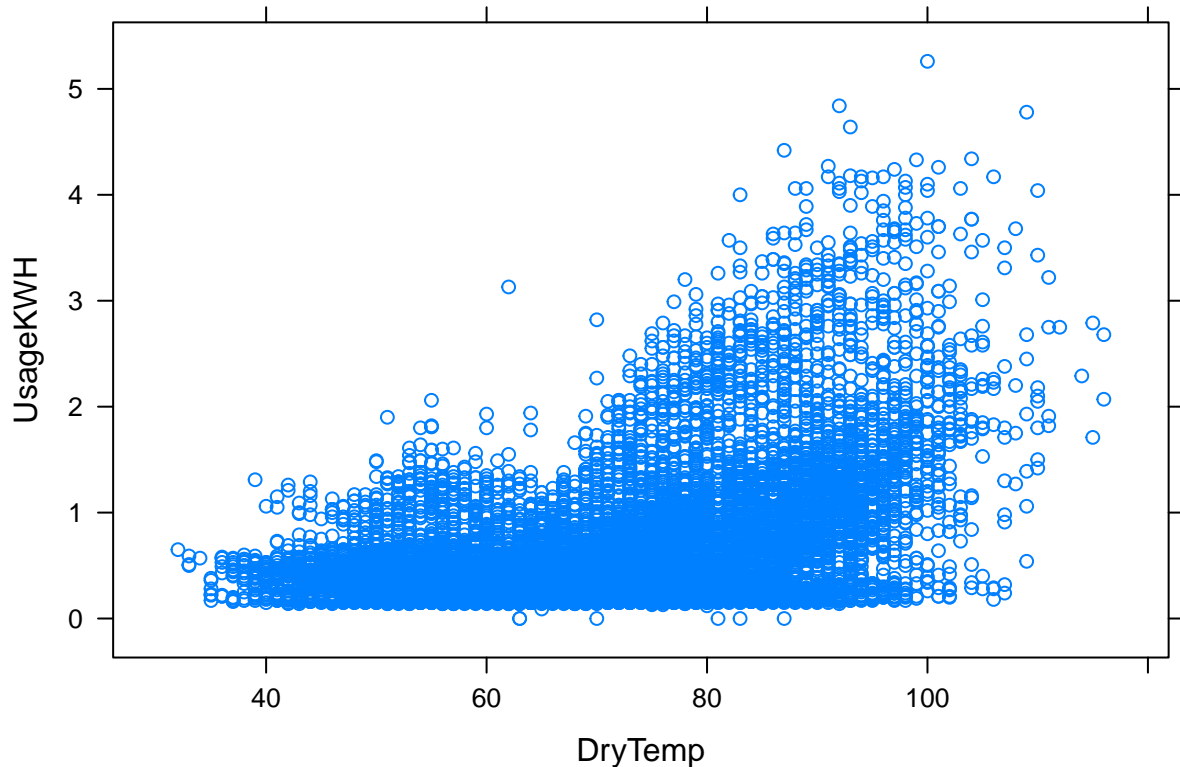
##	Date	UsageKWH	DryTemp	Precip	WindSpeed	Humidity	WetTemp
##	<dtm>	<dbl>	<dbl>	<chr>	<chr>	<dbl>	<dbl>
## 1	2016-06-01 00:00:00	0.2	60	<NA>	6	84	57
## 2	2016-06-01 00:00:00	0.2	60	<NA>	5	84	57
## 3	2016-06-01 00:00:00	0.2	59	0	5	87	57
## 4	2016-06-01 01:00:00	0.2	60	0	7	86	58
## 5	2016-06-01 02:00:00	0.18	61	0	0	84	58
## 6	2016-06-01 03:00:00	0.19	60	0	5	86	58

## Now We Play

```
xyplot(UsageKWH~Date, data=Electricity, type="l")
```



```
xyplot(UsageKWH~DryTemp, data=Electricity)
```



## Did It Work?

We have assumed that all of the above actually worked. We are not really 90% done here. We should have been checking to see if each step was giving us what we thought we were getting.

```
table(is.na(SCE$UsageKWH), is.na(SCE$Date))
```

```
##
##      FALSE  TRUE
## FALSE 20351   0
##  TRUE    0    1
```

```
### Check obs per day
Electricity %>%
  mutate(period.day = floor_date(Date, "day")) %>%
  group_by(period.day) %>%
  summarize(DailyKWH = sum(UsageKWH),
            hours = n()
            ) %>%
  filter(hours != 24)
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
## # A tibble: 358 x 3
##   period.day      DailyKWH hours
##   <dtm>          <dbl> <int>
## 1 2016-06-01 00:00:00    8.02   29
## 2 2016-06-02 00:00:00   18.3    29
```

```
## 3 2016-06-06 00:00:00    11.9    29
## 4 2016-06-07 00:00:00    11.9    29
## 5 2016-06-09 00:00:00    13.2    29
## 6 2016-06-10 00:00:00    17.2    32
## 7 2016-06-11 00:00:00     8.58   31
## 8 2016-06-12 00:00:00     9.54   25
## 9 2016-06-13 00:00:00     7.46   26
## 10 2016-06-14 00:00:00     9.02   31
## # ... with 348 more rows
```

```
### Check obs per day
weather %>%
  mutate(period.day = floor_date(Date, "day")) %>%
  group_by(period.day) %>%
  summarize(
    hours = n()
  ) %>%
  filter(hours != 24)
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
## # A tibble: 1,271 x 2
##   period.day      hours
##   <dtm>          <int>
## 1 2011-01-02 00:00:00    28
## 2 2011-01-03 00:00:00    32
## 3 2011-01-08 00:00:00    33
## 4 2011-01-09 00:00:00    28
## 5 2011-01-10 00:00:00    27
## 6 2011-01-14 00:00:00    25
## 7 2011-01-20 00:00:00    25
## 8 2011-01-26 00:00:00    25
## 9 2011-01-29 00:00:00    25
## 10 2011-01-30 00:00:00    38
## # ... with 1,261 more rows
```

```
### Check obs per day
SCE %>%
  mutate(period.day = floor_date(Date, "day")) %>%
  group_by(period.day) %>%
  summarize(DailyKWH = sum(UsageKWH),
    hours = n()
  ) %>%
  filter(hours != 24)
```

```
## Warning: Grouping rowwise data frame strips rowwise nature
```

```
## # A tibble: 6 x 3
##   period.day      DailyKWH hours
##   <dtm>          <dbl> <int>
## 1 2016-11-06 00:00:00     9.56    25
## 2 2017-03-12 00:00:00     9.95    23
## 3 2017-11-05 00:00:00     9.27    25
## 4 2018-02-13 00:00:00     8.15    23
## 5 2018-03-11 00:00:00    10.4    23
## 6 NA              NA        1
```

Apparently, things did not work. It is time to do some wrangling.