Using GGPlot in R

Mike Levine

June 29, 2019

GGPlot is a powerful graphics tool in R for making beautiful graphs. In this tutorial, let's use a dataset from the popular #MakeoverMonday competition. The dataset was originally published in Week 4 of 2019 through the program.

10 Downing Street, located in Westminster, London, is the British equivalent to the White House. The building is over 300 years old and houses the Prime Minister of the United kingdom, The Spouse of the Prime Minister and Family, and the Chief Mouser to the Cabinet Office (the Cabinet Office's cat–yes, you read that right). The building has gone through a number of renovations since its construction in 1694. Most notably, renovations etween 1960 and 1990 were aimed at preventing the building from decaying beyond repair.

Let's start by setting our working directory, pulling in our dataset and looking at the first five rows of it.

```
setwd("C:/Users/mdlev/OneDrive/Documents/Data Projects and Datasets/Data
Vizes/Energy Use at 10 Downing St")

dat <- read.csv("places.9_2018_elec.csv", header=T)

head(dat)
##          X X0.30 X1.00 X1.30 X2.00 X2.30 X3.00 X3.30 X4.00 X4.30 X5.00
## 1 1/1/2018  58.5  55.7  55.2  50.7  47.7  46.6  47.7  47.8  48.2  50.0
## 2 1/2/2018  49.3  47.4  48.0  50.5  49.7  52.0  51.3  49.4  47.9  48.9
## 3 1/3/2018  50.6  49.5  48.9  49.7  47.4  48.7  47.9  49.6  48.5  50.3
## 4 1/4/2018  50.7  49.4  49.0  51.7  52.6  53.2  51.6  51.9  52.8  53.0
## 5 1/5/2018  50.3  48.9  51.7  51.0  49.8  49.4  49.0  47.5  49.2  50.6
## 6 1/6/2018  53.2  53.2  49.3  50.7  52.2  51.7  52.1  51.4  52.5  52.4
##   X5.30 X6.00 X6.30 X7.00 X7.30 X8.00 X8.30 X9.00 X9.30 X10.00 X10.30
## 1  48.1  47.5  47.5  47.0  47.2  48.3  47.9  47.6  47.9   48.5   48.7
## 2  54.3  58.0  56.4  55.9  62.1  66.3  67.9  69.8  75.4   76.4   74.1
## 3  55.7  59.6  63.3  63.9  66.9  69.5  74.4  73.2  73.9   75.4   77.1
## 4  56.0  56.4  58.9  59.0  59.1  64.4  70.6  72.0  74.9   78.8   75.4
## 5  52.2  51.1  55.0  57.2  58.6  65.4  66.0  70.5  75.1   75.9   76.1
## 6  56.4  55.6  52.8  53.3  53.3  54.6  57.0  58.5  59.5   58.1   57.9
##   X11.00 X11.30 X12.00 X12.30 X13.00 X13.30 X14.00 X14.30 X15.00 X15.30
## 1   48.2   48.8   49.2   48.9   48.9   45.8   46.7   48.2   49.8   48.1
## 2   76.2   77.3   73.9   74.6   71.5   71.9   70.2   71.8   69.1   65.7
## 3   76.7   74.9   76.7   76.6   76.8   75.6   72.6   71.1   73.1   72.3
## 4   76.1   74.5   73.3   73.8   73.2   73.5   70.5   70.1   72.4   70.7
## 5   73.9   75.5   74.3   75.3   76.5   73.9   73.2   69.5   69.6   72.7
## 6   59.7   60.5   59.3   58.9   58.5   59.6   57.3   57.7   58.5   56.9
##   X16.00 X16.30 X17.00 X17.30 X18.00 X18.30 X19.00 X19.30 X20.00 X20.30
## 1   48.6   48.5   48.7   48.2   50.9   50.9   52.1   51.8   51.7   49.0
## 2   66.6   67.5   64.1   64.0   58.8   58.0   59.0   58.9   57.5   54.7
## 3   72.0   73.5   71.5   67.7   65.4   62.3   60.5   58.6   60.5   57.4
## 4   67.4   65.4   66.6   65.6   62.7   63.9   63.7   61.5   59.3   61.2
## 5   73.2   71.8   72.3   68.8   67.9   66.0   64.2   64.9   60.4   60.1
## 6   57.9   57.3   54.4   55.8   54.9   56.8   58.4   55.9   55.8   58.5
##   X21.00 X21.30 X22.00 X22.30 X23.00 X23.30 X0.00
## 1   48.0   48.2   51.1   50.8   50.8   49.4    NA
## 2   51.5   51.6   49.3   49.5   49.0   49.1  49.5
## 3   58.0   55.0   53.5   52.2   50.9   50.2  49.6
## 4   56.4   54.3   53.2   52.8   51.8   52.3  52.2
## 5   58.1   57.8   55.5   52.5   51.1   53.5  53.9
## 6   55.9   56.3   57.8   54.2   56.4   55.8  55.7
```

Looking at our dataset, we can see that it's organized with dates in the first column and half-hour increments in energy use (in kW). Let's create a heatmap by day of the week for average energy use.

To do this, we'll need to first create a new column with the day of the week for each row. We'll accomplish this using the lubridate library, like so:

```
library(lubridate)
dat$X <- as.Date(dat$X, "%m/%d/%Y")
dat$X <- wday(dat$X, label = TRUE)
head(dat$X)
## [1] Mon Tue Wed Thu Fri Sat
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

Now if we look at the first five rows in our dataset, we'll see that our 'X' variable has been converted to be the day of the week for each date.

```
head(dat)
##     X X0.30 X1.00 X1.30 X2.00 X2.30 X3.00 X3.30 X4.00 X4.30 X5.00 X5.30
## 1 Mon  58.5  55.7  55.2  50.7  47.7  46.6  47.7  47.8  48.2  50.0  48.1
## 2 Tue  49.3  47.4  48.0  50.5  49.7  52.0  51.3  49.4  47.9  48.9  54.3
## 3 Wed  50.6  49.5  48.9  49.7  47.4  48.7  47.9  49.6  48.5  50.3  55.7
## 4 Thu  50.7  49.4  49.0  51.7  52.6  53.2  51.6  51.9  52.8  53.0  56.0
## 5 Fri  50.3  48.9  51.7  51.0  49.8  49.4  49.0  47.5  49.2  50.6  52.2
## 6 Sat  53.2  53.2  49.3  50.7  52.2  51.7  52.1  51.4  52.5  52.4  56.4
##   X6.00 X6.30 X7.00 X7.30 X8.00 X8.30 X9.00 X9.30 X10.00 X10.30 X11.00
## 1  47.5  47.5  47.0  47.2  48.3  47.9  47.6  47.9   48.5   48.7   48.2
## 2  58.0  56.4  55.9  62.1  66.3  67.9  69.8  75.4   76.4   74.1   76.2
## 3  59.6  63.3  63.9  66.9  69.5  74.4  73.2  73.9   75.4   77.1   76.7
## 4  56.4  58.9  59.0  59.1  64.4  70.6  72.0  74.9   78.8   75.4   76.1
## 5  51.1  55.0  57.2  58.6  65.4  66.0  70.5  75.1   75.9   76.1   73.9
## 6  55.6  52.8  53.3  53.3  54.6  57.0  58.5  59.5   58.1   57.9   59.7
##   X11.30 X12.00 X12.30 X13.00 X13.30 X14.00 X14.30 X15.00 X15.30 X16.00
## 1   48.8   49.2   48.9   48.9   45.8   46.7   48.2   49.8   48.1   48.6
## 2   77.3   73.9   74.6   71.5   71.9   70.2   71.8   69.1   65.7   66.6
## 3   74.9   76.7   76.6   76.8   75.6   72.6   71.1   73.1   72.3   72.0
## 4   74.5   73.3   73.8   73.2   73.5   70.5   70.1   72.4   70.7   67.4
## 5   75.5   74.3   75.3   76.5   73.9   73.2   69.5   69.6   72.7   73.2
## 6   60.5   59.3   58.9   58.5   59.6   57.3   57.7   58.5   56.9   57.9
##   X16.30 X17.00 X17.30 X18.00 X18.30 X19.00 X19.30 X20.00 X20.30 X21.00
## 1   48.5   48.7   48.2   50.9   50.9   52.1   51.8   51.7   49.0   48.0
## 2   67.5   64.1   64.0   58.8   58.0   59.0   58.9   57.5   54.7   51.5
## 3   73.5   71.5   67.7   65.4   62.3   60.5   58.6   60.5   57.4   58.0
## 4   65.4   66.6   65.6   62.7   63.9   63.7   61.5   59.3   61.2   56.4
## 5   71.8   72.3   68.8   67.9   66.0   64.2   64.9   60.4   60.1   58.1
## 6   57.3   54.4   55.8   54.9   56.8   58.4   55.9   55.8   58.5   55.9
##   X21.30 X22.00 X22.30 X23.00 X23.30 X0.00
## 1   48.2   51.1   50.8   50.8   49.4    NA
## 2   51.6   49.3   49.5   49.0   49.1  49.5
## 3   55.0   53.5   52.2   50.9   50.2  49.6
## 4   54.3   53.2   52.8   51.8   52.3  52.2
## 5   57.8   55.5   52.5   51.1   53.5  53.9
## 6   56.3   57.8   54.2   56.4   55.8  55.7
```

In order to create our heatmap, we first need to reformat our data. Right now, it's in a 'wide' format. We need it to be in a 'long' format. In Excel, this could take quite a while to do! In R, it's a matter of one line of code.

```
library(reshape2)

dat2 <- melt(dat)
## Using X as id variables
```

Using the 'melt' function from the 'reshape2' library, we're able to quicky transform our dataset like so:

```
head(dat2)
##     X variable value
## 1 Mon    X0.30  58.5
## 2 Tue    X0.30  49.3
## 3 Wed    X0.30  50.6
## 4 Thu    X0.30  50.7
## 5 Fri    X0.30  50.3
## 6 Sat    X0.30  53.2
```

Next, let's aggregate our 'value' column into averages by our 'X' and 'variable' columns. We'll also rename our columns to be 'Day,' 'Time' and 'Kw.'

```
dat3 <- aggregate(dat2, by=list(dat2$X,dat2$variable), FUN=mean, na.rm=TRUE, 
warnings=FALSE)
dat3$X <- NULL
dat3$variable <- NULL
colnames(dat3)[colnames(dat3) == 'Group.1'] <- 'Day'
colnames(dat3)[colnames(dat3) == 'Group.2'] <- 'Time'
colnames(dat3)[colnames(dat3) == 'value'] <- 'Kw'
```

```
head(dat3)
##   Day  Time       Kw
## 1 Sun X0.30 52.96667
## 2 Mon X0.30 53.26923
## 3 Tue X0.30 54.53077
## 4 Wed X0.30 55.00000
## 5 Thu X0.30 55.06154
## 6 Fri X0.30 53.08333
```

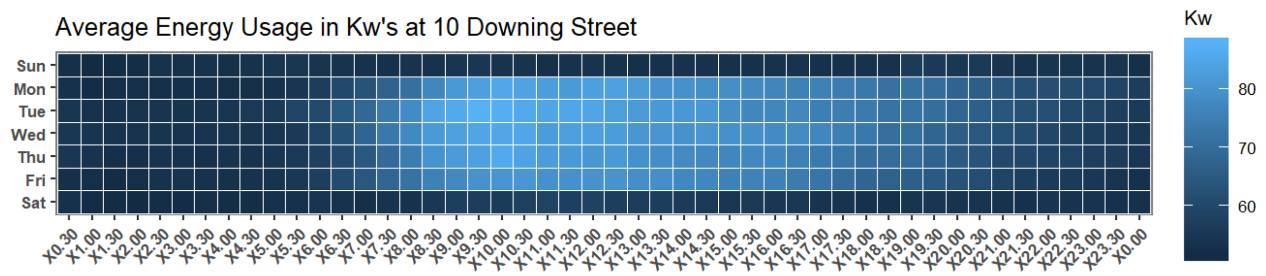
Now let’s make sure our ‘Day’ variable is leveled correctly as a factor. If we don’t do this, our heatmap will look strangely out of order. Fortunately, it is leveled correctly.

```
is.factor(dat3$Day)
## [1] TRUE
as.factor(dat3$Day)
## Levels: Sun < Mon < Tue < Wed < Thu < Fri < Sat
```

Now that we’ve reformatted our data, let’s create our heatmap.

```
library(ggplot2)
heatmap <- ggplot(dat3, aes(x = Time, y = Day, fill = Kw)) +
  ggtitle("Average Energy Usage in Kw's at 10 Downing Street") +
  geom_tile(colour = "white", size = 0.25) +
  labs(x = "", y = "") +
  scale_y_discrete(limits = rev(levels(dat3$Day))) +
  coord_fixed() +
  theme_dark(base_size = 8) +
  theme(
  axis.text = element_text(face = "bold"),
  axis.ticks = element_line(size = 0.4),
  plot.background = element_blank(),
  panel.border = element_blank()
  )

  heatmap + theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


Average Energy Usage in Kw's at 10 Downing Street
Kw
80
70
60
Sun
Mon
Tue
Wed
Thu
Fri
Sat
X0.30 X1.00 X1.30 X2.00 X2.30 X3.00 X3.30 X4.00 X4.30 X5.00 X5.30 X6.00 X6.30 X7.00 X7.30 X8.00 X8.30 X9.00 X9.30 X10.00 X10.30 X11.00 X11.30 X12.00 X12.30 X13.00 X13.30 X14.00 X14.30 X15.00 X15.30 X16.00 X16.30 X17.00 X17.30 X18.00 X18.30 X19.00 X19.30 X20.00 X20.30 X21.00 X21.30 X22.00 X22.30 X23.00 X23.30 X0.00