Lesson 7 Solutions

Stefanie Molin

April 26, 2017

Let's do some practice problems to challenge your understanding of [redacted] and review the material from the prior lessons.

1. Write a query to a file to collect the last 30 days of stats for the client of your choice. Make the query simple (like the one we used in the lesson) and dynamic. Use [redacted] to read the query into a string and query Vertica once you have modified the query. Use `dplyr` to pivot the data up as you wish, then save the dataframe to a CSV.

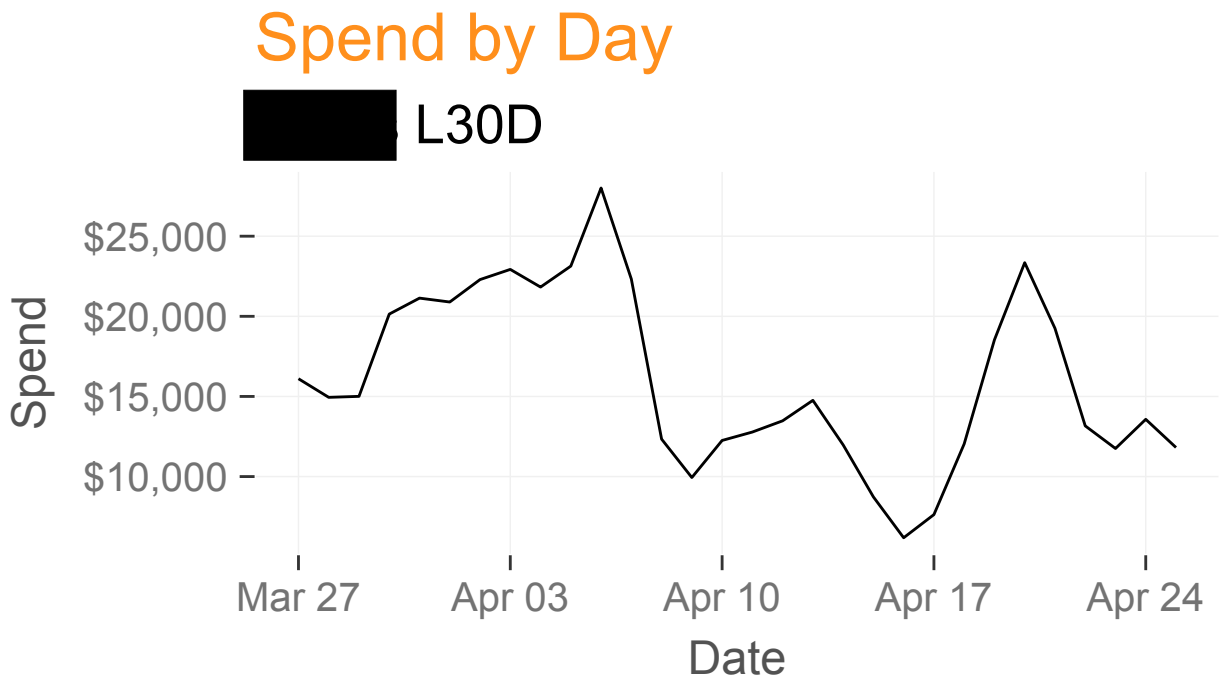
```
# load libraries
library([redacted])
library(dplyr)

# parameters
startDate <- Sys.Date() - 30
[redacted]_client_id <- 8050

# read query from a file
query <- read_query_from_file("client_stats_query.sql")

# query for last 30 days of client stats for[redacted]
[redacted] <- pull_data_vertica(sprintf(query, startDate, [redacted]_client_id), cluster = "pa4",
                                username = username, password = password)

# pivot dataframe
[redacted]_pivot <- [redacted] %>%
  select(day, displays, clicks, revenue, pc_conv = post_click_conversions,
         pc_sales = post_click_sales) %>%
  group_by(day) %>%
  summarize(total_clicks = sum(clicks, na.rm = TRUE),
            total_imps = sum(displays, na.rm = TRUE),
            spend = sum(revenue, na.rm = TRUE),
            conv = sum(pc_conv, na.rm = TRUE)) %>%
  mutate(ctr = total_clicks/total_imps, cpc = spend/total_clicks) %>%
  arrange(day)

# see first few rows
head([redacted]_pivot, 3)
```

```
## # A tibble: 3 × 7
##          day total_clicks total_imps    spend  conv         ctr       cpc
##        <chr>        <dbl>      <dbl>    <dbl> <dbl>       <dbl>     <dbl>
## 1 2017-03-27       [redacted]
## 2 2017-03-28
## 3 2017-03-29
```

```
# write to a CSV
csv_name <- "[redacted].csv"
write.csv(x = [redacted]_pivot, file = csv_name, row.names = FALSE)
```

2. Using the dataframe you obtained in (1), (a) make a [redacted] themed ggplot showing spend by day. (b) Save the graph as a PNG. (c) Send both the PNG and the CSV from (1) to yourself using [redacted].

```
# load library
library(ggplot2)

# define aesthetics
(plot <- [redacted]_pivot %>%
  ggplot(aes(x = as.Date(day), y = spend)) +

  # add line
  geom_line() +

  # format labels
  xlab("Date") +
  scale_y_continuous("Spend", labels = scales::dollar) +

  # add title
  ggtitle("Spend by Day", subtitle = "[redacted] L30D") +

  # [redacted] themes and colors
  [redacted]::scale_color[redacted]() +
  [redacted]::theme_[redacted]_default())
```




```
# save the plot as an image
plot_name <- "[redacted].png"
ggsave(plot_name, plot)

# send plot and CSV
send_email(username = username, password = password, recipient = "[redacted]",
           attachment_files = c(csv_name, plot_name), subject = "[redacted] Training Ex 2")
```

3. Turn the dataframe and ggplot from (1) and (2) into a PowerPoint using [redacted] and `ReporteRs`. Send yourself the deck.

```
# create ppt object
ppt <- [redacted]::ppt[redacted]_template()

# load ReporteRs library
library(ReporteRs)

# add slides and content
# add title slide
ppt <- addSlide(ppt, "main_title")
ppt <- addTitle(ppt, "[redacted]Performance Last 30 Days", width = 8)

# add slide for graph
ppt <- ppt %>%
  addSlide("blank_slide") %>%
  addTitle("Spend by Day") %>%
  addPlot(fun = print, x = plot)

# add slide for metrics by day
ppt <- ppt %>%
  addSlide("blank_slide") %>%
  addTitle("Daily Stats") %>%
  addFlexTable(FlexTable([redacted]_pivot))

# add an end slide
ppt <- addSlide(ppt, "end_slide")

# save the ppt
filename <- "[redacted]Metrics Last 30 Days.pptx"
writeDoc(ppt, file = filename)

# send email with PPT
send_email(username = username, recipient = "[redacted]",
           attachment_files = filename, password = password,
           subject = "[redacted] PPT: Solution 3",
           body = "[redacted]PPT from [redacted]Training.")
```

.