

No API, No Problem

Automating content delivery with R and Excel

Joshua Goldberg | Hasib Neaz

 @GoldbergData

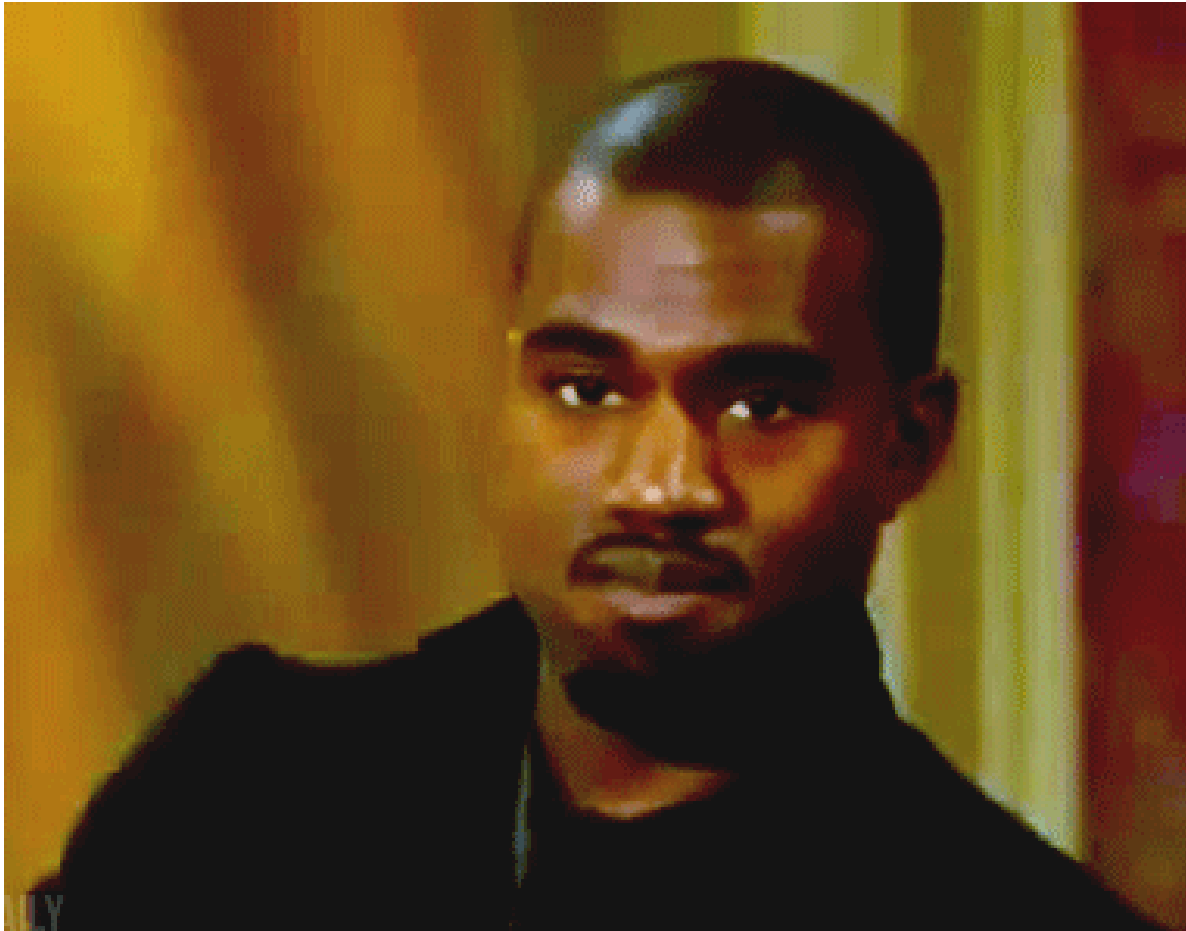
slides at github.com/GoldbergData/no-api-no-problem

We are in the future...



So we can move on, right?





Your boss wants 49 Excel sheets



Data dump

sales_data

```
## # A tibble: 9,994 x 21
##   Person `Order ID` `Order Date`      `Ship Date`      `Ship Mode`
##   <chr>  <chr>      <dtm>          <dtm>          <chr>
## 1 Cassa... CA-2016-1... 2016-11-08 00:00:00 2016-11-11 00:00:00 Second Cla...
## 2 Cassa... CA-2016-1... 2016-11-08 00:00:00 2016-11-11 00:00:00 Second Cla...
## 3 Anna ... CA-2016-1... 2016-06-12 00:00:00 2016-06-16 00:00:00 Second Cla...
## 4 Cassa... US-2015-1... 2015-10-11 00:00:00 2015-10-18 00:00:00 Standard C...
## 5 Cassa... US-2015-1... 2015-10-11 00:00:00 2015-10-18 00:00:00 Standard C...
## 6 Anna ... CA-2014-1... 2014-06-09 00:00:00 2014-06-14 00:00:00 Standard C...
## 7 Anna ... CA-2014-1... 2014-06-09 00:00:00 2014-06-14 00:00:00 Standard C...
## 8 Anna ... CA-2014-1... 2014-06-09 00:00:00 2014-06-14 00:00:00 Standard C...
## 9 Anna ... CA-2014-1... 2014-06-09 00:00:00 2014-06-14 00:00:00 Standard C...
## 10 Anna ... CA-2014-1... 2014-06-09 00:00:00 2014-06-14 00:00:00 Standard C...
## # ... with 9,984 more rows, and 16 more variables: `Customer ID` <chr>,
## #   `Customer Name` <chr>, Segment <chr>, Country <chr>, City <chr>,
## #   State <chr>, `Postal Code` <dbl>, Region <chr>, `Product ID` <chr>,
## #   Category <chr>, `Sub-Category` <chr>, `Product Name` <chr>,
## #   Sales <dbl>, Quantity <dbl>, Discount <dbl>, Profit <dbl>
```



XLConnect created by Mirai Solutions GmbH

Excel interface

What can XLConnect do?

Excel interface

What can XLConnect do?

- Create/load Excel workbooks

Excel interface

What can XLConnect do?

- Create/load Excel workbooks
- Add/read data from Excel

Excel interface

What can XLConnect do?

- Create/load Excel workbooks
- Add/read data from Excel
- Format worksheets (some limitations)

Excel interface

What can XLConnect do?

- Create/load Excel workbooks
- Add/read data from Excel
- Format worksheets (some limitations)
- Include images

Excel interface

What can XLConnect do?

- Create/load Excel workbooks
- Add/read data from Excel
- Format worksheets (some limitations)
- Include images
- And much more... see [here](#)

Excel interface

What can XLConnect do?

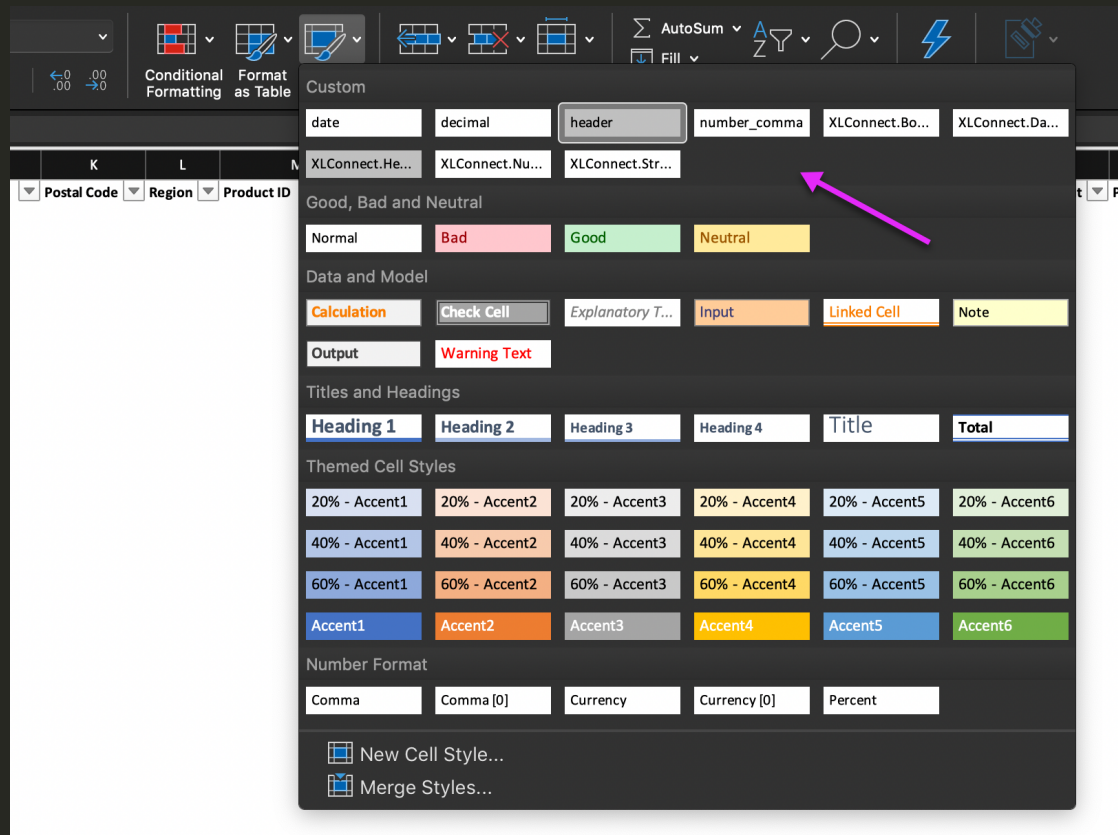
- Create/load Excel workbooks
- Add/read data from Excel
- Format worksheets (some limitations)
- Include images
- And much more... see [here](#)

All programatically!

Make a template

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit
2																				
3																				
4																				
5																				
6																				
7																				
8																				
9																				
10																				
11																				
12																				
13																				
14																				
15																				
16																				
17																				
18																				
19																				
20																				
21																				
22																				
23																				
24																				
25																				
26																				
27																				
28																				
29																				
30																				
31																				
32																				
33																				
34																				
35																				
36																				
37																				
38																				
39																				
40																				
41																				
42																				
43																				
44																				
45																				
46																				
47																				
48																				
49																				
50																				
51																				
52																				
53																				

Create cell styles



Load/create workbook and styles

```
wb <- loadWorkbook("template.xlsx")
```

```
date_cellstyle <- getCellStyle(wb, "date")  
num_comma_cellstyle <- getCellStyle(wb, "number_comma")  
dec_cellstyle <- getCellStyle(wb, "decimal")  
  
setStyleAction(wb, XLC$STYLE_ACTION.NONE)
```

Write data/format data

```
writeWorksheet(wb, data = data,
               sheet_name,
               startRow = 2,
               header = FALSE)

walk(
  2:(row_count + 1),
  ~ setCellStyle(
    wb,
    sheet = sheet_name,
    row = .x,
    col = 2:3,
    cellstyle = date_cellstyle
  )
)

walk(
  2:(row_count + 1),
  ~ setCellStyle(
    wb,
    sheet = sheet_name,
    row = .x,
    col = c(sales, discount, profit),
    cellstyle = num_comma_cellstyle
  )
)
```

Add a plot to Excel

```
createName(wb, "plot", formula = glue("Summary!{idx2cref(c(1, 6))}") %>% as.character)

p <- data %>%
  group_by(Segment, Category) %>%
  summarise_if(is.numeric, ~ sum(.x, na.rm = TRUE)) %>%
  gather(key = key, value = value, -c("Segment", "Category", "Postal Code")) %>%
  filter(!key %in% c("Quantity", "Discount")) %>%
  ggplot(aes(Category, value, fill = key)) +
  geom_col(position = "dodge") +
  scale_fill_brewer(name = "Metric", type = "qual", palette = 2) +
  scale_y_continuous(label = scales::comma) +
  facet_wrap( ~ Segment) +
  ggthemes::theme_tufte() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Profit/Sales by Segment/Category",
       y = NULL)

ggsave(p, filename = "segment-category.png", width = 4, height = 3)

addImage(wb, filename = "segment-category.png", name = "plot", originalSize = TRUE)

saveWorkbook(wb, file = file_name)
```

Functional

```
make_plot()
```

```
style()
```

```
build_excel()
```

Functional

```
make_plot()
```

```
style()
```

```
build_excel()
```

```
walk(sales_split, function(split) {  
  build_excel(split, "reports")  
})
```

Functional
















```
make_plot()
```

```
style()
```

```
build_excel()
```

```
walk(sales_split, function(split) {  
  build_excel(split, "reports")  
})
```

```
plan(multiprocess)  
furrr::future_map(sales_split, ~ build_excel(.x, "reports"))
```

	..	
<input type="checkbox"/>	 Alabama.xlsx	99 KB
<input type="checkbox"/>	 Arizona.xlsx	119.6 KB
<input type="checkbox"/>	 Arkansas.xlsx	98.6 KB
<input type="checkbox"/>	 California.xlsx	345 KB
<input type="checkbox"/>	 Colorado.xlsx	117.5 KB
<input type="checkbox"/>	 Connecticut.xlsx	102.8 KB
<input type="checkbox"/>	 Delaware.xlsx	104.1 KB
<input type="checkbox"/>	 District of Columbia.xlsx	89.3 KB
<input type="checkbox"/>	 Florida.xlsx	140.9 KB
<input type="checkbox"/>	 Georgia.xlsx	123.4 KB
<input type="checkbox"/>	 Idaho.xlsx	95.1 KB
<input type="checkbox"/>	 Illinois.xlsx	160.9 KB
<input type="checkbox"/>	 Indiana.xlsx	110.2 KB
<input type="checkbox"/>	 Iowa.xlsx	91.2 KB
<input type="checkbox"/>	 Kansas.xlsx	93 KB
<input type="checkbox"/>	 Kentucky.xlsx	109.9 KB



Sales by Segment and Category

Row Labels	Total Sales \$	Total Profit \$	Count
Consumer	11,683	1,629	42
Furniture	3,584	246	8
Office Supplies	1,578	359	24
Technology	6,521	1,024	10
(blank)			
Corporate	1,366	429	16
Furniture			
Office Supplies	1,293	420	15
Technology	74	8	1
(blank)			
Home Office	497	40	3
Furniture	320	17	2
Office Supplies			
Technology	177	23	1
(blank)			
Grand Total	13,546	2,098	61

Profit/Sales by Segment/Category California



Compose a message

```
subject = "Sales State Coverage Report"

message <- paste0(
  '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
    <title>Sales Coverage</title>
    <style type="text/css">
    </style>
  </head>
  <body>
    Hello,
  </p>
  <p>Attached is sales data for your state coverage.</p>
  <p>A visualization and summary are available on the second tab.</p>
  <p>Let us know if you have questions.</p>
  <p>Kind regards,</p>
  <p>Analysis Group,</p>
  </body>
  </html>'
)
```

mailR/sendmailR

```
send.mail(  
  from = "user@domain.com",  
  to = "user@domain.com",  
  subject = subject,  
  body = message,  
  html = TRUE,  
  smtp = list(  
    host.name = "smtp.gmail.com",  
    port = 465,  
    user.name = rstudioapi::askForPassword(),  
    passwd = rstudioapi::askForPassword(),  
    ssl = TRUE  
  ),  
  attach.files = "file_path",  
  authenticate = TRUE,  
  send = TRUE  
)
```

Nice to know

- Java/rJava are required to use XLConnect and mailR/sendmailR.
- Latest version of java: <https://java.com/en/download/>.
- If you have trouble installing Java on Mac, see <https://www.chrisjmendez.com/2018/11/16/installing-xlconnect-and-rjava-on-macos/>.
- It's typically required you set your JAVA_HOME, such as
`Sys.setenv(JAVA_HOME = "C:\\\\Program Files\\\\Java\\\\jdk1.8.0_102")`.
The path is relative to your machine, so you must locate your directory and version of Java.
- Useful: `vignette("XLConnect")`.
- XLConnect runs on memory until `saveWorkbook()` is called. If you experience out-of-memory errors, try allocating more memory with
`options(java.parameters = "-Xmx1024m")`. See vignette above.