How Did I Do That?: Redesigning A Workflow In R

Edgar Zamora

Big Bend Community College

June 5, 2021

Agenda

Identifying the Gaps 🔎

Connecting to databases with `{dbplyr}` 💾

Importing and transforming 🤠

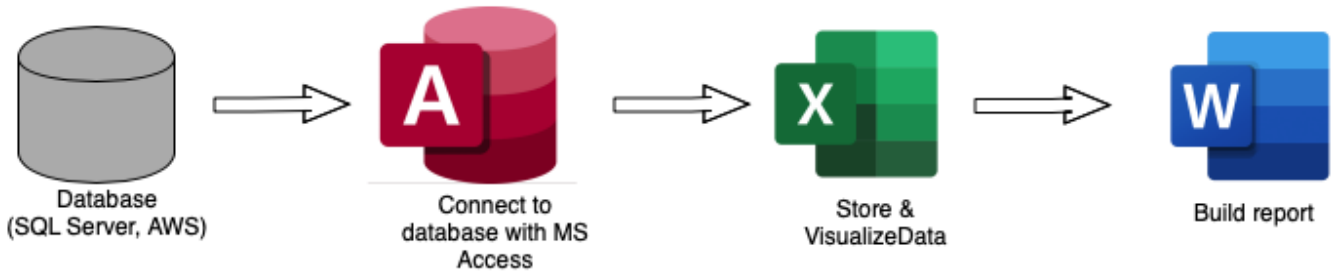
Building the report 📄 🏀

Conclusion 🙆

Identifying the Gaps 🔎

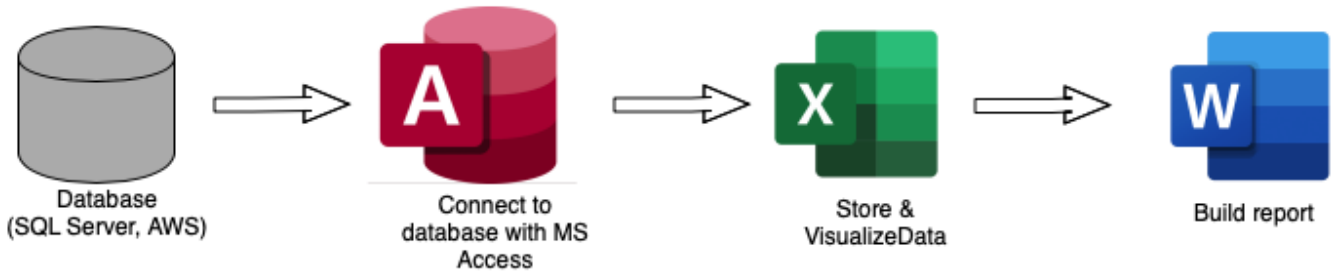
Identifying the Gaps 🔎

Previous Workflow



Identifying the Gaps 🔎

Previous Workflow



Redesigned Workflow

Connecting to databases with `{dbplyr}` 💾

Making a connection

- `con` is the connection that will be used to access the database.
- Multiple connections can be created but you will have to have different names for each connection.

```
library(dbplyr)
library(DBI)
library(odbc)

# ODBC connection (e.g. Access)
con <- dbConnect(odbc::odbc(), "name_of_connection")

# RSQLite
con <- DBI::dbConnect(RSQLite::SQLite(), dbname = ":memory:")
```

{dbplyr} continued

Writing some R | Results

- Most `dplyr` verbs can be translated into their SQL equivalents.
- When writing `dplyr` verbs you can use them before and/or after collecting the results.
  - The size of the database will help guide your decision.

```
dbplyr::tbl(con, "Class") %>%
  filter(YEAR == "B90",
         DEPT_DIV %in% c("POLS&", "CHEM&", "ENGL&")) %>%
  select(YEAR, DEPT_DIV, COURSE_NUM, ENR_TOTAL) %>%
  group_by(YEAR, DEPT_DIV) %>%
  summarise(total_enr = sum(ENR_TOTAL, na.rm = T)) %>%
  dplyr::collect() %>%
  janitor::clean_name()
```

{dbplyr} continued

Writing some R | Results

- After using `collect()` we see the traditional tibble format for the dataframe.
- Depending on how your database is setup, you may have to fix up some of the column names.
  - This is quickly down with the `clean_names()` function from the `{janitor}` package.

```
#`summarise()` has grouped output by 'YEAR'. You can override using the `.groups` argument.
# A tibble: 3 x 3
# Groups:   YEAR [1]
  YEAR  DEPT_DIV total_enr
  <chr> <chr>        <dbl>
1 B90   CHEM&           527
2 B90   ENGL&          1436
3 B90   POLS&           239
```

Importing and visualizing 📊

More {dplyr} | Visualizing

```
class_summary <- tbl(con, "Class") %>%
  filter(YEAR %in% c("B89", "B90"), QUARTER %in% c("1", "2")) %>%
  select(YEAR, QUARTER, DEPT_DIV, ENR_TOTAL) %>%
  collect() %>%
  janitor::clean_names() %>%
  group_by(year, quarter, dept_div) %>%
  summarise(
    total_enr = sum(enr_total, na.rm = T)
  ) %>%
  ungroup() %>%
  group_by(year, quarter) %>%
  arrange(desc(total_enr)) %>%
  slice(1:5) %>%
  mutate(quarter = case_when(quarter == "2" ~ "Fall",
                             TRUE ~ "Winter"),
         year = case_when(year == "B89" ~ "2018-19",
                          TRUE ~ "2019-20")) %>%
  arrange(year, quarter)
```

Importing and visualizing 📊

More {dplyr} | Visualizing

- If your report includes tables, there are many packages (e.g `{gt}`, `{flextable}`, `{kable}`...) that can help.

```
class_summary %>%
  pivot_wider(names_from = c(year, quarter),
              values_from = total_enr) %>%
  arrange(dept_div) %>%
  gt() %>%
  fmt_missing(
    columns = 2:5,
    missing_text = 0
  )
```

- Visualizing data using data from a database connection is no different than working with the `mtcars` dataset.

```
class_summary %>%
  ggplot(aes(dept_div, total_enr, fill = quarter)) +
  geom_col(position = position_dodge()) +
  labs(
    x = "Department",
    y = "Enrollment"
  ) +
  scale_y_continuous(limits = c(0, 600)) +
  facet_wrap(~year) +

  theme_classic()
```

Building the report  

- To report out I used the {rmarkdown} and {bookdown} packages. Each offers consistent and reproducible ways to report out.
- {bookdown} builds on {rmarkdown} so if you use the latter learning {bookdown} will be straight forward.
- {bookdown} offers the ability to create a single document instead of the traditional format. Using `pdf_document2()` in the YAML metadata will produce this format.

Reporting syntax

Metadata | YAML

```
---
  title: "How Did I Do That?: Redesigning A Workflow In R"
  site: bookdown:bookdown_site
  knit: "bookdown:render_book"
  mainfont: Calibri Light
  indent: yes
  urlcolor: blues
---
```

Reporting syntax

Metadata | YAML

```
bookdown::pdf_document2: # allows for a single document
  toc: FALSE
  includes:
    in_header: preable.tex #where you include your latex specific packages
  latex_engine: xelatex # will need to have {tidytext} installed
  dev: "cairo_pdf"
```

Conclusion 🤦

- There is going to be upfront work but the payoff is tremendous.
- The ability to avoid human error is reduced since you do not need to jump between programs.
- Next time will simply consist of rerunning your code instead of spending time trying to remember what you did.
- Ability take your workflow further by incorporating `{purrr}`, `{shiny}`, and package development.

Acknowledgements & Resources

- `{xaringan}`: Isabella Velásquez
- `{dbplyr}`:
  - Vebash Naidoo
  - GitHub Repo
- SQL:
  - Vebash Naidoo
  - tidyquery
  - Visualizing `*_joins()` with Garrick Aden-Buie
- Sharla Gelfand - RStudioConf

Thank You!!!