Data Appendix to "Name of your paper"

Avery Hammond

Contents


```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

1 Appendix description

Your Data Appendix should begin with a brief statement explaining its purpose like the following one.

This Data Appendix documents the data used in "Papter Title". It was prepared in a Rmarkdown document that contains both the documentation and the R code used to prepare the data used in the final estimation. It also includes descriptive statistics for both the original data and the final dataset, with a discussion of any issues of note.

The datasets used directly by the final analysis are saved in `processed-data/` at the end of this file.

Note: this document structure will require you to re-run steps of your analysis multiple times. If your code takes a long time, please come talk with me about strategies to reduce run time or save earlier results.

2 Instructions for Use

This document includes instructions for how to create your Data Appendix. Outside of this section, instruction paragraphs are listed in *italics* (like the first paragraph above). Instructions should be removed before submission.

To start creating your own data appendix, follow these steps:

1. Replace the title and author in the section at the top of the file (called the YAML).
2. Commit your changes with a message like "customizing data appendix".
3. Delete this instruction section of the document.
4. Remove any other instructions in italics and examples from the completed sections of the document.

Remember that you will submit your assignment by committing and then pushing your versions to your repository. I encourage you to commit your changes often as you work, but there are three specific points at which you need to both submit and push changes, corresponding to course deadlines:

1. You must submit a version with the original data section completed by the Data Appendix 1 deadline. This will include the .Rmd file, the .pdf file, and the html data summary files stored in the output folder.
2. You must submit a version with all parts completed by the Data Appendix 2 deadline.

3. You must submit a final version of this document that is consistent with your final paper by the final project deadline.

While creating your data appendix, refer regularly to the assignment descriptions posted on Moodle.

A few tips:

- When creating a list like this one, be sure to put an empty line above the list. If you don't do this, your entries won't be formatted a list.
- Make sure you have empty lines above and below section and subsection headings.
- When creating numbered lists, you can number all items in your list with 1. Rmarkdown will number them sequentially when it creates your final document.

#3 Raw data *Each dataset you use will have its own documentation section. The next subsection in this document (Dataset description) is a template. You can copy this section and paste it into your document each time you need to add a section for a new dataset. Note that each line in the Dataset description section* ***must*** *end with two spaces.* This section documents the datasets used in this analysis.

##3.1 Historical Pollutant and Weather Data **Citation:** Texas Commission on Environmental Quality. (2003-2004). "Historical Pollutant and Weather Data." Retrieved from https://www.tceq.texas.gov/airquality/monops/historical_data.html#red **Date Downloaded:** March 10, 2020 **Filename(s):** raw_data/camswx_200x.csv raw_data/co_200x.csv raw_data/nox_200x.csv raw_data/oz_200x.csv raw_data/pm25x_200x.csv raw_data/so2_200x.csv *If you have a large number of files you can use a patten (see visit data below)* **Unit of observation:** parts per billion, meters/second, degrees compass, decrees Celsius **Dates covered:** 01/01/2003-12/31/2004

2.0.1 To obtain a copy

Interested users should visit the Historical Pollutant and Weather Data on the Texas Commission on Environmental Quality website at https://www.tceq.texas.gov/airquality/monops/historical_data.html#red. To download data from 2003 and 2004, users should click on the corresponding years for each column, which will download the data setfor each column as a CSV file.

##3.2 Metro Rail Station Data

Citation: City of Houston. (2019). "Metro Rail Station (current)." Retrieved from https://cohgis-mycity.opendata.arcgis.com/datasets/coh-metro-rail-station-current-1?geometry=-95.616%2C29.649%2C-95.181%2C29.858 **Date Downloaded:** March 10, 2020 **Filename(s):** raw_data/COH_METRO_RAIL_STATION_current.csv **Unit of observation:** degrees longitude, degrees latitude, year in service **Dates covered:** 2004

2.0.2 To obtain a copy

Interested users should visit the Metro Rail Stations section of the City of Houston GIS data website at https://cohgis-mycity.opendata.arcgis.com/datasets/1dc7a23374ac44cdae8553044bfeaf22_14?geometry=-95.618%2C29.649%2C-95.179%2C29.858. Users should select "Download," and then select "Spreadsheet" beneath "Full Dataset," which will download the metro rail station data as a csv file.

2.0.3 Variable descriptions

Create a bullet list with the name of each variable in the dataset followed by any information the user would need to understand it.

- **variable_name:** Variable description.
- **variable_name2:** Description of second variable.

2.0.4 Data import code and summary

Once you've described the variables, enter an R chunk by selecting Code -> Insert Chunk, or Ctrl+Alt+I, give it a name to describe the dataset you are importing. After importing, export a dataframe summary using the command.

```
years = c("2003","2004")

download_and_read_data <- function(filename){

  if (!file.exists(file.path("raw-data",str_c(filename, ".csv")))) {

    destfile = file.path("raw-data",str_c(filename, ".zip"))

    download.file(url = str_c("https://www.tceq.texas.gov/assets/public/compliance/monops/air/ozonehist/

    unzip(destfile, exdir = "raw-data", junkpaths = T)
  }
  this_data <- read_csv(file.path("raw-data", str_c(filename, ".csv")))
}
```

```
oz_data <- lapply(str_c("oz_",years), download_and_read_data) %>%
  bind_rows() %>%
  select(airs, ST_CODE, AQCR, date, contains("OZ1hr")) %>%
  select(-OZ1hrvh, -OZ1hrvd, -OZ1hrpk, -OZ1hrav) %>%
  group_by(airs, ST_CODE, AQCR, date) %>%
  pivot_longer(cols = contains("OZ1"), names_to = "hour", values_to = "ozone")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   OZunit = col_character(),
##   OZmetxt = col_character()
## )

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

## Warning: 1082 parsing failures.
##   row     col expected actual                   file
## 17625 OZ1hrvh a double     ** 'raw-data/oz_2003.csv'
## 17626 OZ1hrvh a double     ** 'raw-data/oz_2003.csv'
## 17627 OZ1hrvh a double     ** 'raw-data/oz_2003.csv'
## 17628 OZ1hrvh a double     ** 'raw-data/oz_2003.csv'
```

```
## 17629 OZ1hrvh a double     ** 'raw-data/oz_2003.csv'
## ..... ....... ........ ...... ......................
## See problems(...) for more details.

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   OZunit = col_character(),
##   OZmetxt = col_character(),
##   poc = col_logical(),
##   units = col_logical()
## )
## See spec(...) for full column specifications.

## Warning: 2609 parsing failures.
##   row     col           expected actual                   file
## 17435 OZ1hrvh a double              **  'raw-data/oz_2004.csv'
## 17435 units   1/0/T/F/TRUE/FALSE    ppb 'raw-data/oz_2004.csv'
## 17436 OZ1hrvh a double              **  'raw-data/oz_2004.csv'
## 17436 units   1/0/T/F/TRUE/FALSE    ppb 'raw-data/oz_2004.csv'
## 17437 OZ1hrvh a double              **  'raw-data/oz_2004.csv'
## ..... ....... .................. ...... ......................
## See problems(...) for more details.
```

```
stargazer::stargazer(oz_data)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Mar 11, 2020 - 14:21:26

Table 1:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max

```
co_data <- lapply(str_c("co_",years), download_and_read_data) %>%
  bind_rows() %>%
  select(airs, ST_CODE, AQCR, date, contains("CO1hr")) %>%
  select(-CO1hrvh, -CO1hrvd, -CO1hrpk, -CO1hrav) %>%
  group_by(airs, ST_CODE, AQCR, date) %>%
  pivot_longer(cols = contains("CO1"), names_to = "hour", values_to = "carbon monoxide")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   COunit = col_character(),
##   COmetxt = col_character()
## )

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

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   COunit = col_character(),
##   COmetxt = col_character()
## )

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

```
stargazer::stargazer(co_data)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Mar 11, 2020 - 14:21:27

Table 2:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max

```
so2_data <- lapply(str_c("so2_",years), download_and_read_data) %>%
  bind_rows() %>%
  select(airs, ST_CODE, AQCR, date, contains("SO21hr")) %>%
  select(-SO21hrvh, -SO21hrvd, -SO21hrpk, -SO21hrav) %>%
  group_by(airs, ST_CODE, AQCR, date) %>%
  pivot_longer(cols = contains("SO21"), names_to = "hour", values_to = "sulfur dioxide")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   SO2unit = col_character(),
##   SO2metxt = col_character()
## )

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

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   SO2unit = col_character(),
##   SO2metxt = col_character()
## )

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

```
stargazer::stargazer(so2_data)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Mar 11, 2020 - 14:21:28

Table 3:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max

```
nox_data <- lapply(str_c("nox_",years), download_and_read_data) %>%
  bind_rows() %>%
  select(airs, ST_CODE, AQCR, date, contains("NOX1hr")) %>%
  select(-NOX1hrvh, -NOX1hrvd, -NOX1hrpk, -NOX1hrav) %>%
  group_by(airs, ST_CODE, AQCR, date) %>%
  pivot_longer(cols = contains("NOX1"), names_to = "hour", values_to = "nitrous oxides")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   NOunit = col_character(),
##   NOmetxt = col_character(),
##   NO2unit = col_character(),
##   NO2metxt = col_character(),
##   NOXunit = col_character(),
##   NOXmetxt = col_character()
## )

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

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   NOunit = col_character(),
##   NOmetxt = col_character(),
##   NO2unit = col_character(),
##   NO2metxt = col_character(),
##   NOXunit = col_character(),
##   NOXmetxt = col_character()
## )

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

```
stargazer::stargazer(nox_data)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Mar 11, 2020 - 14:21:29

Table 4:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max

```
pm25_data <- lapply(str_c("pm25x_",years), download_and_read_data) %>%
  bind_rows() %>%
  select(AIRS, ST_CODE, AQCR, date, contains("PM251hr")) %>%
  select(-PM251hrvh, -PM251hrvd, -PM251hrpk, -PM251hrav) %>%
  group_by(AIRS, ST_CODE, AQCR, date) %>%
  pivot_longer(cols = contains("PM251"), names_to = "hour", values_to = "PM2.5")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   PM25unit = col_character(),
##   PM25metxt = col_logical()
## )

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

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   PM25unit = col_character(),
##   PM25metxt = col_logical()
## )

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

```
stargazer::stargazer(pm25_data)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Mar 11, 2020 - 14:21:30

Table 5:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max

```
weather_data <- lapply(str_c("camswx_",years), download_and_read_data) %>%
  bind_rows() %>%
  select(AIRS, ST_CODE, AQCR, date, contains("WSR1hr"), contains("TMP1hr")) %>%
  select(-WSR1hrvh, -WSR1hrvd, -WSR1hrpk, -WSR1hrav) %>%
  group_by(AIRS, ST_CODE, AQCR, date) %>%
  pivot_longer(cols = contains("WSR1"), names_to = "hour", values_to = "weather")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   WSRunit = col_character(),
##   WSRmetxt = col_character(),
```

```
##   WDRunit = col_character(),
##   WDRmetxt = col_character(),
##   WSAunit = col_character(),
##   WSAmetxt = col_character(),
##   WDA1hr0 = col_logical(),
##   WDA1hr1 = col_logical(),
##   WDA1hr2 = col_logical(),
##   WDA1hr3 = col_logical(),
##   WDA1hr4 = col_logical(),
##   WDA1hr5 = col_logical(),
##   WDA1hr6 = col_logical(),
##   WDA1hr7 = col_logical(),
##   WDA1hr8 = col_logical(),
##   WDA1hr9 = col_logical(),
##   WDA1hr10 = col_logical(),
##   WDA1hr11 = col_logical(),
##   WDA1hr12 = col_logical()
##   # ... with 18 more columns
## )

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

## Warning: 10142 parsing failures.
##   row     col           expected actual                       file
## 10748 WDA1hr0 1/0/T/F/TRUE/FALSE  83.9  'raw-data/camswx_2003.csv'
## 10748 WDA1hr1 1/0/T/F/TRUE/FALSE  121.5 'raw-data/camswx_2003.csv'
## 10748 WDA1hr2 1/0/T/F/TRUE/FALSE  284.4 'raw-data/camswx_2003.csv'
## 10748 WDA1hr3 1/0/T/F/TRUE/FALSE  275.6 'raw-data/camswx_2003.csv'
## 10748 WDA1hr4 1/0/T/F/TRUE/FALSE  279.4 'raw-data/camswx_2003.csv'
## ..... ....... .................. ...... ........................
## See problems(...) for more details.

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   date = col_character(),
##   WSRunit = col_character(),
##   WSRmetxt = col_character(),
##   WDRunit = col_character(),
##   WDRmetxt = col_character(),
##   WSAunit = col_character(),
##   WSAmetxt = col_character(),
##   WDA1hr0 = col_logical(),
##   WDA1hr1 = col_logical(),
##   WDA1hr2 = col_logical(),
##   WDA1hr3 = col_logical(),
##   WDA1hr4 = col_logical(),
##   WDA1hr5 = col_logical(),
##   WDA1hr6 = col_logical(),
##   WDA1hr7 = col_logical(),
##   WDA1hr8 = col_logical(),
##   WDA1hr9 = col_logical(),
##   WDA1hr10 = col_logical(),
```

```
##   WDA1hr11 = col_logical(),
##   WDA1hr12 = col_logical()
##   # ... with 18 more columns
## )
## See spec(...) for full column specifications.

## Warning: 10223 parsing failures.
##   row     col           expected actual                       file
## 10523 WDA1hr0 1/0/T/F/TRUE/FALSE  15.8  'raw-data/camswx_2004.csv'
## 10523 WDA1hr1 1/0/T/F/TRUE/FALSE  348.4 'raw-data/camswx_2004.csv'
## 10523 WDA1hr2 1/0/T/F/TRUE/FALSE  176.4 'raw-data/camswx_2004.csv'
## 10523 WDA1hr3 1/0/T/F/TRUE/FALSE  249.9 'raw-data/camswx_2004.csv'
## 10523 WDA1hr4 1/0/T/F/TRUE/FALSE  261.9 'raw-data/camswx_2004.csv'
## ..... ....... .................. ...... ........................
## See problems(...) for more details.
```

```
stargazer::stargazer(weather_data)
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Wed, Mar 11, 2020 - 14:21:37

Table 6:

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max

```
export_summary_table(dfSummary(dataset_name))
```

While it will make your resulting file long, you should not modify the chunk options to suppress printing of code and output. I would likely not include this in the documentation for an actual paper I was submitting, but including them here will let me read your code and the output message from R and may help identify data import concerns early in the process. Since these files will exist only electronically, their length is less of a concern. If you like to print out files to proofread and want me to help you shorten the printed versions, let me know. We can temporarily modify the chunk options for printing and restore them before you submit the assignment.

3 Data Processing and Combination

This section should include a discussion of the processing and merging steps needed to create your basic data. The code to implement these steps should be included in chunks in this section. Once the final merged data has been created, you should use the dfSummary function again to summarize the data you will be using. You should also save a file containing all the objects you will use in your final analysis to the processed_data folder.

4 Analysis Variables

This section should include a description of all the variables that are used in your final analysis. At the end of the section, you should save all of these variables in the processed_data folder of your repository.

5 Discussion of Data

This section should include a discussion of any data patterns you notice based on the summaries created in the code above.