Quarto Tutorial 3

Alier Reng

April 24, 2022

Table of contents

1 Introduction

> Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

1.1 Definitions

Before getting started, let's explain what each option in the `yaml` means.

- `toc` adds the table of contents to your document.

- `number-sections` adds number to the section headings when sets to `true`.
- `Latex` equations are rendered using `MathJax`; however, you can change this to other options, as shown above.
- `highlight-style` is used to style code outputs.
- `code-overflow` controls the width of source code. When sets to `wrap`, the source code wraps around and vice versa.

There are numerous options to style and format your document, so we recommend reading the documentation on the Quarto website.

1.2 Loading the Libraries

Here we will load `pandas`, `seaborn`, and `matplotlib`.

```
# Loading packages
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Formatting
sns.set_context('notebook')
sns.set_style('white')
```

1.3 Importing the Dataset

```
# Import the data
ss_2008_census_df = pd.read_csv('../00_data/ss_2008_census_data_raw.csv')

# Inspect the first 5 rows
ss_2008_census_df.head()
```

	Region	Region Name	Region - RegionId	Variable	Variable Name	Age	Age Nam
0	KN.A2	Upper Nile	SS-NU	KN.B2	Population, Total (Number)	KN.C1	Total
1	KN.A2	Upper Nile	SS-NU	KN.B2	Population, Total (Number)	KN.C2	0 to 4
2	KN.A2	Upper Nile	SS-NU	KN.B2	Population, Total (Number)	KN.C3	5 to 9
3	KN.A2	Upper Nile	SS-NU	KN.B2	Population, Total (Number)	KN.C4	10 to 14
4	KN.A2	Upper Nile	SS-NU	KN.B2	Population, Total (Number)	KN.C5	15 to 19

```
# Inspect the last 5 rows
ss_2008_census_df.tail()
```

	Region	Region Name	Region - RegionId	Variable	V
448	KN.A11	Eastern Equatoria	SS-EE	KN.B8	P
449	KN.A11	Eastern Equatoria	SS-EE	KN.B8	P
450	NaN	NaN	NaN	NaN	N
451	Source:	National Bureau of Statistics, South Sudan	NaN	NaN	N
452	Download URL:	http://southsudan.opendataforafrica.org/fvjqdp...	NaN	NaN	N

Above, we see that the three last rows contain `nas` (missing values). One is the data source where we obtained this dataset, and the other is the data URL.

2 Cleaning and Transforming the Data

2.1 Checking for Missing Values

Now that we have imported our dataset, we will clean and manipulate it. First, we will reconfirm the missing values and proceed with our data wrangling process.

```
# Check for missing values
ss_2008_census_df.isna().sum()
```

	0
Region	1
Region Name	1
Region - RegionId	3
Variable	3
Variable Name	3
Age	3
Age Name	3
Scale	3
Units	3
2008	3

2.2 Wrangling the Data Using `Method Chaining`

```
# Select desired columns
cols = ['Region Name', 'Variable Name', 'Age Name', '2008']

# Rename columns
cols_names = {'Region Name':'state',
              'Variable Name':'gender',
```

```
              'Age Name':'age_cat',
              '2008':'population'}

# Create new age categories
new_age_cats = {'0 to 4':'0-14',
                '5 to 9':'0-14',
                '10 to 14':'0-14',
                '15 to 19':'15-29',
                '20 to 24':'15-29',
                '25 to 29':'15-29',
                '30 to 34':'30-49',
                '35 to 39':'30-49',
                '40 to 44':'30-49',
                '45 to 49':'30-49',
                '50 to 54':'50-64',
                '55 to 59':'50-64',
                '60 to 64':'50-64',
                '65+':'>= 65'
                }


# Clean the data
df = (ss_2008_census_df
      [cols]
      .rename(columns = cols_names)
      .query('~age_cat.isna()')
      .assign(gender = lambda x:x['gender'].str.split('\s+').str[1],
              age_cat = lambda x:x['age_cat'].replace(new_age_cats),
              population = lambda x:x['population'].astype('int')
      )
      .query('gender != "Total" & age_cat != "Total"')
      # .drop(columns = 'pop_cat', axis = 'column')
      .groupby(['state', 'gender', 'age_cat'])['population']
      .sum()
      .reset_index()
     )

# Inspect the first 5 rows
df.head()
```

	state	gender	age_cat	population
0	Central Equatoria	Female	0-14	221216
1	Central Equatoria	Female	15-29	166887
2	Central Equatoria	Female	30-49	101676
3	Central Equatoria	Female	50-64	23460
4	Central Equatoria	Female	>= 65	8596

3 Summarizing Census Data

3.1 Population by State

```
# Calculate census data by state
st_df = (df
         .groupby(['state'])['population']
         .sum()
         .reset_index()
         .sort_values('population',
                      ascending=False,
                      ignore_index=True)
         )

# Display the outpout
st_df
```

	state	population
0	Jonglei	1358602
1	Central Equatoria	1103557
2	Warrap	972928
3	Upper Nile	964353
4	Eastern Equatoria	906161
5	Northern Bahr el Ghazal	720898
6	Lakes	695730
7	Western Equatoria	619029
8	Unity	585801
9	Western Bahr el Ghazal	333431

3.2 Population by State and Gender

```
# Calculate census data by state and gender
gender_df = (df
```

```
          .groupby(['state', 'gender'])['population']
          .sum()
          .reset_index()
          .sort_values('population',
                       ascending=False,
                       ignore_index=True)
          )

# Display the outpout
gender_df.head()
```

	state	gender	population
0	Jonglei	Male	734327
1	Jonglei	Female	624275
2	Central Equatoria	Male	581722
3	Upper Nile	Male	525430
4	Central Equatoria	Female	521835

3.3 Population by State, Gender, and Age Category

```
# Calculate census data by state, gender, and age category
age_df = (df
          .groupby(['state', 'gender', 'age_cat'])['population']
          .sum()
          .reset_index()
          .sort_values(['state','population'],
                       ascending = [True, False],
                       ignore_index = True)
          )

# Display the outpout
age_df.head(5)
```

	state	gender	age_cat	population
0	Central Equatoria	Male	0-14	242247
1	Central Equatoria	Female	0-14	221216
2	Central Equatoria	Male	15-29	179903
3	Central Equatoria	Female	15-29	166887
4	Central Equatoria	Male	30-49	119355

4 Closing Remarks

This tutorial demonstrates creating a Quarto document with various yaml options to style and format the output. We hope you will find this tutorial helpful. Please let us know if there are any topics you want us to do a tutorial on.

With that said, our next tutorial will be on R.