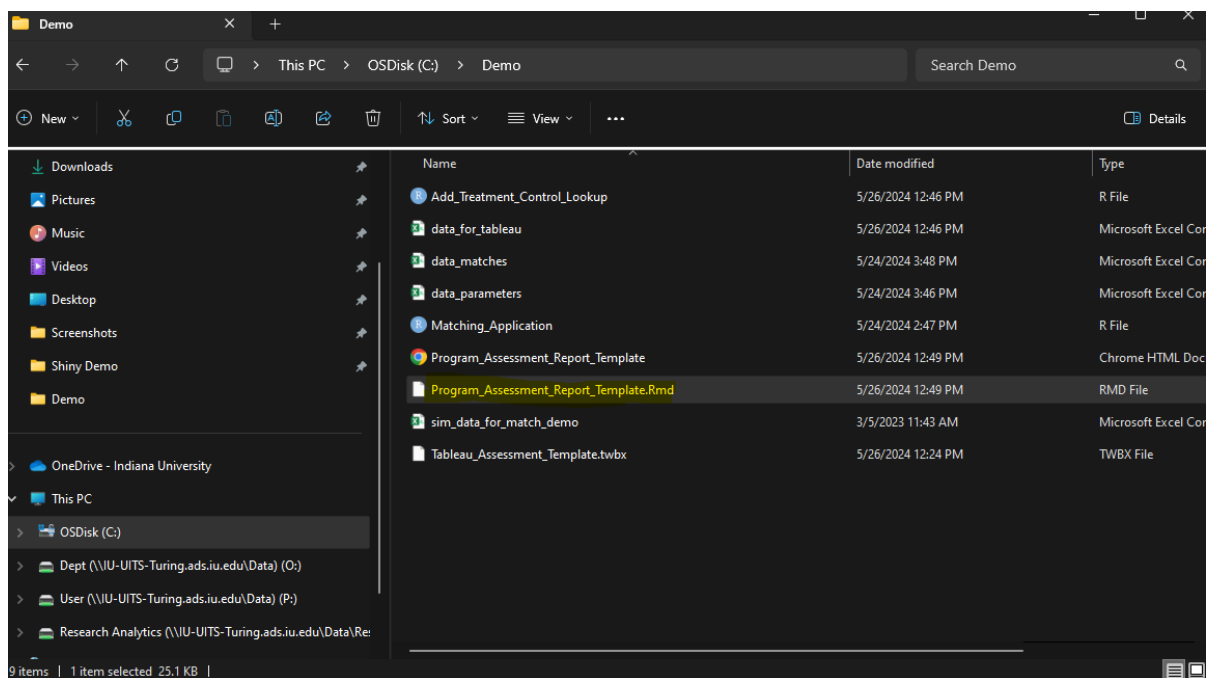
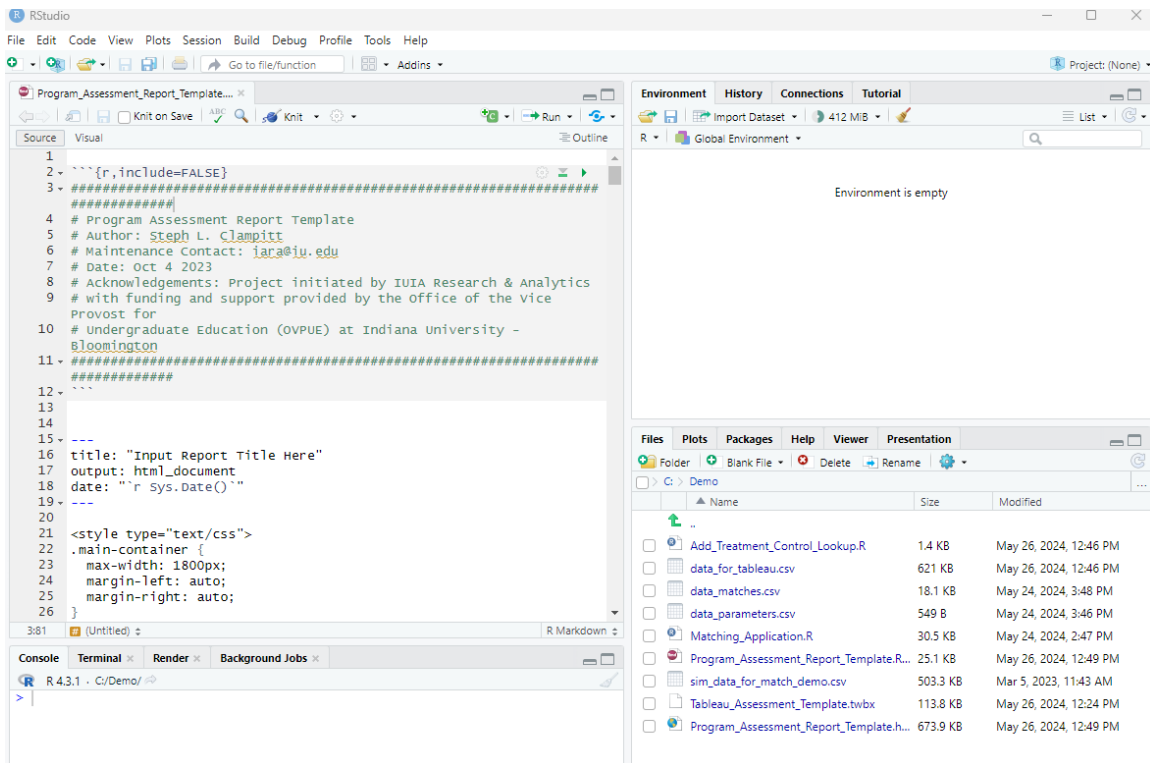


Guide for Program Assessment RMarkdown Report Template

Important: to use the program assessment report template, you must have exported a parameter and matching data file from the matching application (see user guide for program assessment). Additionally, there must be at least one outcome variable specified in the parameter file so that this template knows which outcomes to analyze by the treatment group.

1. Double Click on the “Program_Assessment_Report_Template.Rmd” in file explorer to open the RMarkdown template in RStudio (should look like the second picture below).





2. Navigate to part of the code where the parameters and matching files are specified (lines 50-54). Update the file path to be the file paths (portions highlighted below) where your parameter and matching files are located on your machine.

- a. Note that file paths for PCs in R must have forward back slashes or double back slashes
 - i. If having difficulty setting a path to a file, simple google searches on setting file paths in R should help. Or, you can use the File -> Open File menus in the RStudio environment to navigate to files, and the path will be loaded into your console. Then, you can paste that path into the code.

```
#####
# Input datasets from matching app
# Parameters - Match Parameters CSV
# Matches - Matched Data CSV
#####

# Change File Path to Parameters File from Matching Application
parameters <- read.csv("C:\\Demo\\data_parameters.csv")

# Change File Path to Matched Data File from Matching Application
matches <- read.csv("C:\\Demo\\data_matches.csv")
```

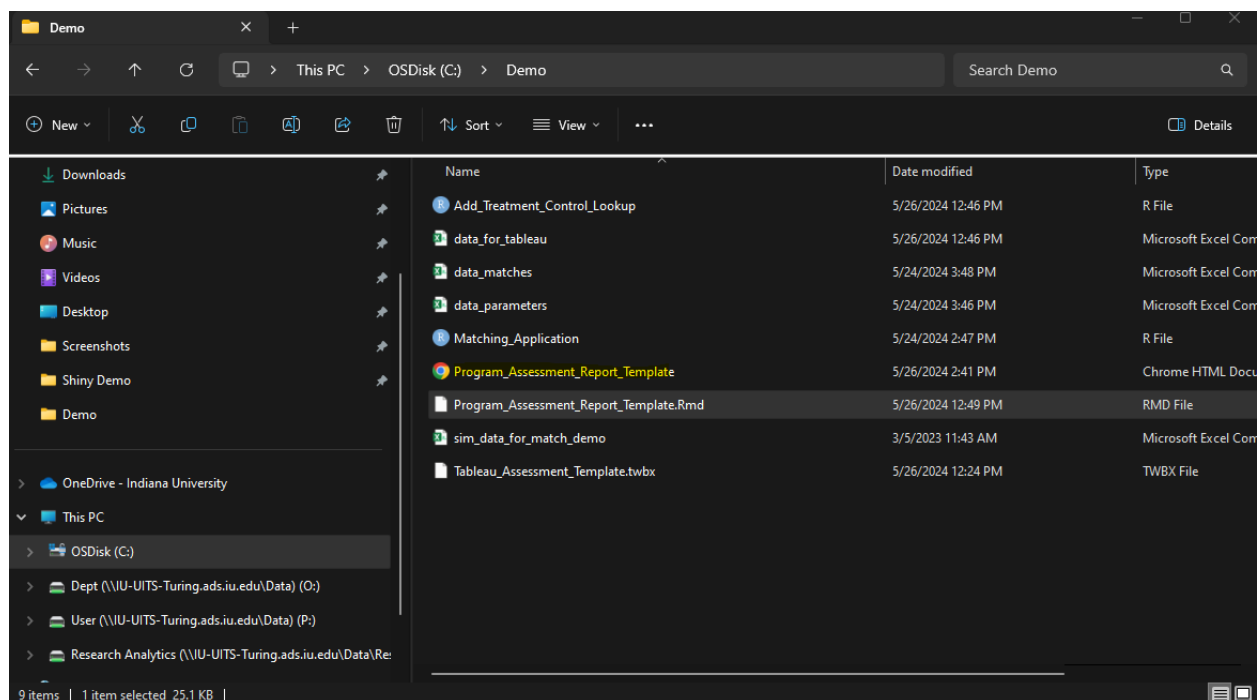
3. Click on the “Knit” button (blue yarn icon at the top of the RStudio task bar). This will load in the data from the two files and create a basic html report file at the same location as the .Rmd file is saved.

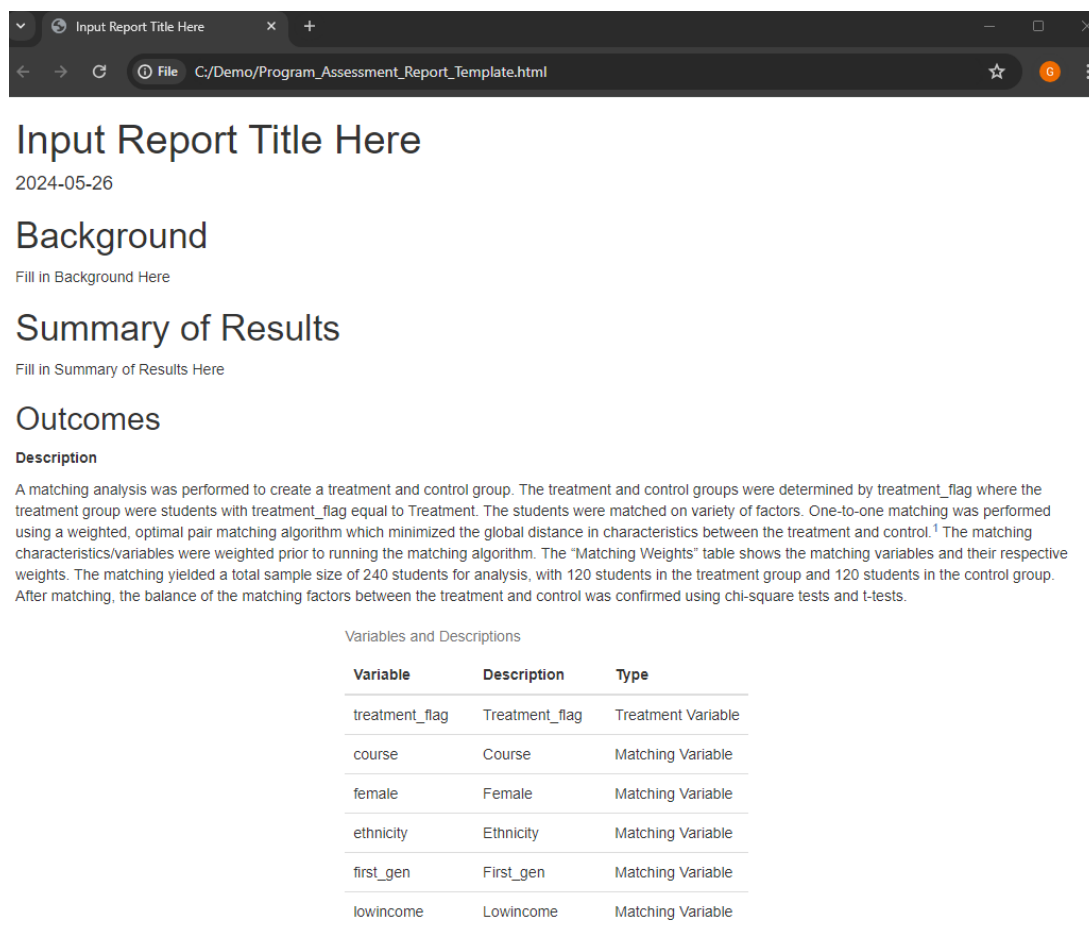
```

27 </style>
28
29 # Background
30
31 Fill in Background Here
32
33 # Summary of Results
34
35 Fill in Summary of Results Here
36
37 ```{r setup for text, include=FALSE}
38 library(dplyr)
39 library(Hmisc)
40 library(english)
41 library(stringr)
42 library(ggplot2)
43
44 #####
45 # Input datasets from matching app
46 # Parameters - Match Parameters CSV
47 # Matches - Matched Data CSV
48 #####
49
50 # Change File Path to Parameters File from Matching Application
51 parameters <- read.csv("C:\\Demo\\data_parameters.csv")
52
53 # Change File Path to Matched Data File from Matching Application
54 matches <- read.csv("C:\\Demo\\data_matches.csv")
55
56
57 #set up dataset with variable descriptions (if description missing, replace with VALUE)
58 parameters<-parameters%>%
59   dplyr::mutate(description=case_when(description=="NA"~VALUE,
60                                     is.na(description)~VALUE,
61                                     TRUE~description))
62
63 #set up dataset to be used with variable names and descriptions
64 description_table <- select(parameters, VALUE, description, PARAMETER) %>%
65   subset(PARAMETER %in% c("Treatment Variable", "Matching Variable", "Outcome Variable"))
66
67 description_table$description <- capitalize(description_table$description)
68
69 #####
70 #Set up list of outcome variables
71 #####
72 outcome_variables <- parameters%>%
73   dplyr::filter(PARAMETER == "Outcome Variable")%>%
74   dplyr::select(description)
75

```

4. Double click on the html file in file explorer. This will open the basic default report in a browser.





5. This creates a bare bones report, summarizing the data in the parameter and matching files that were exported as a part of the matching application. You can further customize aspects of the report in several ways, and some options are outlined below. After each change is made, you can save the file and re-click on the “Knit” button to update and write out a new report with changes.
 - a. Change Report Title
 - i. Find the “title: Input Report Title Here” portion of the code (line 16) and update the text between the quotation marks to be the title of the report that you want.
 - b. Background Information
 - i. Find the section of the R markdown code that is tagged with “# Background” (beginning at line 29). Replace the “Fill in Background Here” default text with sentences or paragraphs providing background of the report (i.e. purpose of the analysis, the consumer of the analysis, what data were accessed, description of research questions, etc.)
 - c. Summary of Results
 - i. Find the section of the R markdown code that is tagged with “# Summary of Results” (beginning at line 33). Replace the “Fill in Summary of Results Here” default text with sentences or paragraphs describing the results or implications, etc.

d. Variable friendly names and Matching Variable Group Types

- i. Additional aspects of the report can be customized from the parameter file itself, specifically with the “description” and “var type” columns.

PARAMETER	VALUE	WEIGHT_OR_LEVEL	DATA_TYPE	description	var_type
Unique Row Identifier	id_var	NA	NA	NA	NA
Treatment Variable	treatment_flag	Treatment	NA	NA	NA
Matching Variable	course		8 character	NA	NA
Matching Variable	female		2 character	NA	NA
Matching Variable	ethnicity		2 character	NA	NA
Matching Variable	first_gen		2 character	NA	NA
Matching Variable	lowincome		2 character	NA	NA
Matching Variable	high_school_gpa		4 double	NA	NA
Outcome Variable	grade	NA	double	NA	NA
Matching Algorithm	optimal	NA	NA	NA	NA
Matching Ratio		1 NA	NA	NA	NA

- ii. Use the description column to tag variable names with “friendly names.” This will translate the technical variable names into non-technical names for the consumer of the report.
- iii. Use the var_type column to tag matching variable names into groups, if desired. This will update the description section to organize the matching variable text into groups to make the paragraph easier to read and understand.
- iv. Filled in Example for ii & iii

PARAMETER	VALUE	WEIGHT_OR_LEVEL	DATA_TYPE	description	var_type
Unique Row Identifier	id_var	NA	NA	Student ID	Id
Treatment Variable	treatment_flag	Treatment	NA	Intervention Indicator	Treatment
Matching Variable	course		8 character	Course Name (Entry Biology or Upper Level Chemistry)	Course
Matching Variable	female		2 character	Student's Sex	Demographic
Matching Variable	ethnicity		2 character	Student's Race/Ethnicity	Demographic
Matching Variable	first_gen		2 character	Student's First Generation Status	Demographic
Matching Variable	lowincome		2 character	Student's Pell Recipient Status	Demographic
Matching Variable	high_school_gpa		4 double	Student's High School GPA	Academic Performance
Outcome Variable	grade	NA	double	Student's Grade (GPA) in the Course	Outcome
Matching Algorithm	optimal	NA	NA	NA	NA
Matching Ratio		1 NA	NA	NA	NA

e. Adding Sections

- i. Any section can be added to the report with additions of chunks of code.
- ii. For example, the below code chunk would add a linear regression model to summarize how a student’s course grade depends on sex, ethnicity, first generation status, low-income status, high school gpa, and treatment/control indicator.

```

```{r, include=TRUE}

##Example of Linear Regression Model for Course Grade (simulated data set)

linear.reg.model<-lm(grade~factor(female)+
 factor(ethnicity)+
 factor(first_gen)+
 factor(lowincome)+
 high_school_gpa+
 factor(treatment_flag),
 data=matches)|
summary(linear.reg.model)
```

```

- iii. For tips on creating and customizing RMarkdown files, there are a variety of resources online, both written articles in blogs and training videos.