Optimising the workflow

Using R and R Markdown to generate reproducible statistical reports in the higher education sector

Iryna Schlackow

iryna.schlackow@admin.ox.ac.uk

University of Oxford

03 September 2024

Overview

- Background: context, challenges and need for a streamlined solution
- Getting help from R (data cleaning, table generation, tidying up output for publication)
- Getting help from R Markdown (referencing, text)
- (Further) benefits of automation
- Q&A

All data, settings and interpretations in this talk are entirely fictitious

Acknowledgements to Swati Kanoi, Hil Vandormael, Peter Chetwynd and the Oxford University's Admissions and Outreach team for generating the first version of the reports and inspiring discussions

`R Markdown` can be (almost?) replaced with `Quarto` throughout the talk

Background: context and admissions process challenges

Aim: fair access to higher education

Image from https://oxfordvisit.com/articles/harry-potter-filming-locations-great-hall-christ-church/1

- Universities/courses can be very competitive and cannot accept everyone
- Admissions measures used to aid selection: predicted A-level grades, admissions tests/interviews
- Even with a fair selection process, inequalities exist prior to application: those from disadvantaged backgrounds less likely to get grades, be prepared for an admissions test or apply
- Can we quantify the admissions process from the fair access point of view?

Background: technical challenges

- During each admissions cycle, tutors look at the application cohort, eg:
  - number and sociodemographic profile of applicants
  - distribution of admissions measures
  - comparisons with previous years and/or across the university
  - statistics by characteristics of interest (gender, deprivation status, schooling)
- Ca. 40 descriptive statistics reports are produced each year:
  - very similar in structure… but not the same!

Example report: one admissions metric

Herbology

1 Admissions performance in the 2024 UCAS cycle

The information in the tables in this section will give you additional reference points and can be used to identify students with high potential within the applicant cohort to your course.

Table 1.1: Summary of applicants' performance on admissions measures

	Number of GCSE A*s	
Townsend index	Average (SD)	No. of applicants
Quintile 1 (least deprived)	9.12 (3.54)	45
Quintile 2	8.43 (3.79)	52
Quintile 3	7.97 (5.82)	83
Quintile 4	6.80 (4.57)	51
Quintile 5 (most deprived)	6.11 (5.12)	52

Example report: two admissions metrics

Defence against the Dark Arts

1 Admissions performance in the 2024 UCAS cycle

The information in the tables in this section will give you additional reference points and can be used to identify students with high potential within the applicant cohort to your course.

Table 1.1: Summary of applicants' performance on admissions measures

	Number of GCSE A*s		Admissions test	
Townsend index	Average (SD)	No. of applicants	Average (SD)	No. of applicants
Quintile 1 (least deprived)	8.01 (3.65)	60	68.12 (13.54)	58
Quintile 2	7.32 (3.90)	67	66.43 (13.79)	63
Quintile 3	6.86 (5.93)	98	64.97 (15.82)	93
Quintile 4	5.79 (4.68)	66	61.80 (14.50)	63
Quintile 5 (most deprived)	5.34 (5.23)	67	60.11 (15.12)	62

Example report: comparison across admissions years

3 Application stage - comparison of the selection decisions (2023 vs 2024)

The tables below show a comparison in terms of number (and percentage) of applicants for different applicant characteristics between last year and this year. The rows in bold emphasize characteristics relevant to access.

Table 3.2: Applications stage – Townsend index

	2023		2024		
Townsend index	No. of applicants	%	No. of applicants	%	% change from 2023
Q1 (least deprived)	77	22%	85	23%	1.1%
Q2	67	19%	65	17%	-1.4%
Q3	98	27%	102	27%	-0.2%
Q4	66	18%	71	19%	0.4%
Q5 (most deprived)	**50**	**14%**	**53**	**14%**	**0.1%**

The results from Table 3.2 in this section indicate the following:

1. The number of applications **increased** from 358 to 376
2. The **Townsend Q1 / Q5 ratio** at application for the course is **1.6/1 compared to 2.1/1** for the wider University

Copy-pasting approach

- error-prone;
- different faculty requirements;
- not scalable: single database update / requirement change require re-generation of the whole report suite

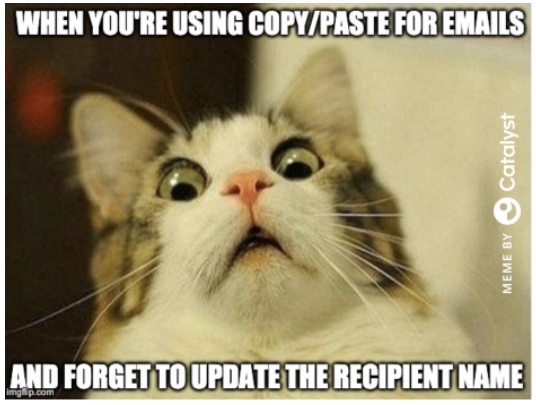


Image from https://www.csmemes.io/memes/copy-paste

Can the workflow be made more efficient?

What parts of the workflow can be automated?

Example automation

Defence against the Dark Arts

1 Admissions performance in the 2024 UCAS cycle

The information in the tables in this section will give you additional reference points and can be used to identify students with high potential within the applicant cohort to your course.

Table 1.1: Summary of applicants' performance on admissions measures

	Number of GCSE A*s		Admissions test	
Townsend index	Average (SD)	No. of applicants	Average (SD)	No. of applicants
Quintile 1 (least deprived)	8.01 (3.65)	60	68.12 (13.54)	58
Quintile 2	7.32 (3.90)	67	66.43 (13.79)	63
Quintile 3	6.86 (5.93)	98	64.97 (15.82)	93
Quintile 4	5.79 (4.68)	66	61.80 (14.50)	63
Quintile 5 (most deprived)	5.34 (5.23)	67	60.11 (15.12)	62

Example automation

Defence against the Dark Arts

1 Admissions performance in the 2024 UCAS cycle

The information in the tables in this section will give you additional reference points and can be used to identify students with high potential within the applicant cohort to your course.

Table 1.1: Summary of applicants' performance on admissions measures

	Number of GCSE A*s		Admissions test	
Townsend index	Average (SD)	No. of applicants	Average (SD)	No. of applicants
Quintile 1 (least deprived)	8.01 (3.65)	60	68.12 (13.54)	58
Quintile 2	7.32 (3.90)	67	66.43 (13.79)	63
Quintile 3	6.86 (5.93)	98	64.97 (15.82)	93
Quintile 4	5.79 (4.68)	66	61.80 (14.50)	63
Quintile 5 (most deprived)	5.34 (5.23)	67	60.11 (15.12)	62

Example automation

3 Application stage - comparison of the selection decisions (2023 vs 2024)

The tables below show a comparison in terms of number (and percentage) of applicants for different applicant characteristics between last year and this year. The rows in bold emphasize characteristics relevant to access.

Table 3.2: Applications stage - Townsend index

	2023		2024		
Townsend index	No. of applicants	%	No. of applicants	%	% change from 2023
Q1 (least deprived)	77	22%	85	23%	1.1%
Q2	67	19%	65	17%	-1.4%
Q3	98	27%	102	27%	-0.2%
Q4	66	18%	71	19%	0.4%
Q5 (most deprived)	**50**	**14%**	**53**	**14%**	**0.1%**

The results from Table 3.2 in this section indicate the following:

1. The number of applications **increased** from 358 to 376
2. The **Townsend Q1 / Q5 ratio** at application for the course is **1.6/1 compared to 2.1/1** for the wider University

How can  and R Markdown help?

Data cleaning and prepation

- Link data sources (admissions, sociodemographics)
- Rename columns in a generic way (helps with future test names changes/locations)
- Transform data into correct format, eg character into numeric; continuous into categories
- Wide format: 1 line per person; characteristics in columns.

Data cleaning performed on the whole dataset, with data on all subjects

	A	B	C	D	E	F	G	H
1	student_id	subject	gender	school	townsend5	gcse	test1	test2
2	1	History of mag	M	wizard	Quintile 1 (lea	5.27954768	79.01929786	NA
3	2	History of mag	M	wizard	Quintile 5 (mo	9.337527977	51.69189462	NA
4	3	Defence again	M	wizard	Quintile 2	7.540460468	81.3589797	NA
5	4	Herbology	F	muggle	Quintile 5 (mo	6.35798785	NA	NA
6	5	Defence again	M	wizard	Quintile 3	7.14821682	71.0535352	NA
7	6	Defence again	F	muggle	Quintile 3	9.035023146	62.77405622	NA
8	7	History of mag	M	muggle	Quintile 4	6.357793325	57.09717157	NA
9	8	History of mag	M	muggle	Quintile 5 (mo	5.129904447	71.37380142	NA
10	9	Herbology	F	muggle	Quintile 4	6.807521728	NA	NA

Generating tables

```
df %>%
  group_by(townsend5) %>%
  filter(!is.na(gcse)) %>%
  summarise(var_mean = mean(gcse),
            var_sd = sd(gcse),
            var_n = n())
```

```
## # A tibble: 5 x 4
##   townsend5                    var_mean var_sd var_n
##   <fct>                           <dbl>  <dbl> <int>
## 1 Quintile 1 (least deprived)      7.29   1.30    60
## 2 Quintile 2                       7.42   1.56    60
## 3 Quintile 3                       7.43   1.53    64
## 4 Quintile 4                       7.33   1.52    67
## 5 Quintile 5 (most deprived)       7.56   1.24    57
```

Parameterising

```
group_by_var <- "townsend5"
measure_var <- "gcse"

df %>%
  filter(!is.na(!!sym(measure_var))) %>%
  group_by(!!sym(group_by_var))%>%
  summarise(var_mean = mean(!!sym(measure_var)),
            var_sd = sd(!!sym(measure_var)),
            var_n = n())
```

```
## # A tibble: 5 x 4
##   townsend5                    var_mean var_sd var_n
##   <fct>                           <dbl>  <dbl> <int>
## 1 Quintile 1 (least deprived)      7.29   1.30    60
## 2 Quintile 2                       7.42   1.56    60
## 3 Quintile 3                       7.43   1.53    64
## 4 Quintile 4                       7.33   1.52    67
## 5 Quintile 5 (most deprived)       7.56   1.24    57
```

Translating repetitive operations into functions

```
.get_mean_by_var <- function(df, group_by_var, measure_var) {
  retval <- df %>%
    filter(!is.na(!!sym(group_by_var))) %>%
    group_by(!!sym(group_by_var)) %>%
    summarise(var_mean = mean(!!sym(measure_var)),
              var_sd = sd(!!sym(measure_var)),
              var_n = n()) %>%
    rename(group := !!quo_name(group_by_var))
  return(retval)
}
```

```
.get_mean_by_var(df = df, group_by_var = "townsend5", measure_var = "gcse")
```

```
## # A tibble: 5 x 4
##   group                        var_mean var_sd var_n
##   <fct>                           <dbl>  <dbl> <int>
## 1 Quintile 1 (least deprived)      7.29   1.30    60
## 2 Quintile 2                       7.42   1.56    60
## 3 Quintile 3                       7.43   1.53    64
## 4 Quintile 4                       7.33   1.52    67
## 5 Quintile 5 (most deprived)       7.56   1.24    57
```

Translating repetitive operations into functions

```
.get_mean_by_var <- function(df, group_by_var, measure_var) {
  retval <- df %>%
    filter(!is.na(!!sym(group_by_var))) %>%
    group_by(!!sym(group_by_var)) %>%
    summarise(var_mean = mean(!!sym(measure_var)),
              var_sd = sd(!!sym(measure_var)),
              var_n = n()) %>%
    rename(group := !!quo_name(group_by_var))
  return(retval)
}
```

```
.get_mean_by_var(df = df, group_by_var = "gender", measure_var = "test1")
```

```
## # A tibble: 2 x 4
##   group var_mean var_sd var_n
##   <chr>    <dbl>  <dbl> <int>
## 1 F         75.3   15.3   140
## 2 M         76.0   14.3   168
```

Creating tables of publication quality

Package `flextable` converts a table output into a Word format

```
output_means <- .get_mean_by_var(df = df, group_by_var = "townsend5", measure_var = "gcse")
flextable(output_means)
```

group	var_mean	var_sd	var_n
Quintile 1 (least deprived)	7.289597	1.296666	60
Quintile 2	7.418129	1.555758	60
Quintile 3	7.428153	1.531850	64
Quintile 4	7.326366	1.516620	67
Quintile 5 (most deprived)	7.555323	1.239044	57

Creating tables of publication quality

To tidy up Word tables, `flextable` + `officer` could be used (`kable` + `kableExtra` for html):

- merge cells
- add header rows and footnotes
- change font, size, alignment (individual cell, row, column or whole table)
- colour-code cells (absolutely or conditionally)
- change all/some table borders
- construct by specifying *layers*, connected by `%>%`

Repetitive operations can be translated into functions

group	var_mean	var_sd	var_n
Quintile 1 (least deprived)	7.289597	1.296666	60
Quintile 2	7.418129	1.555758	60
Quintile 3	7.428153	1.531850	64
Quintile 4	7.326366	1.516620	67
Quintile 5 (most deprived)	7.555323	1.239044	57

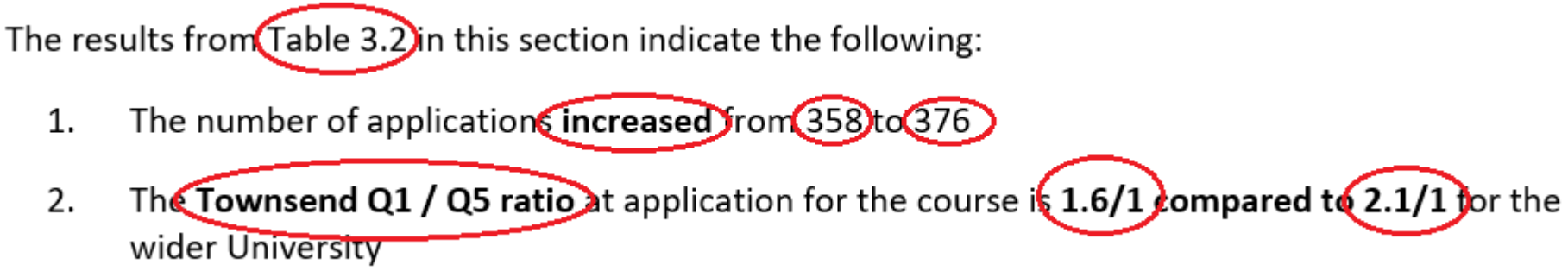
Applicants' performance on admissions measures

	Number of GCSE A*s	
Townsend quintile	Average (SD)	No. of applicants
Quintile 1 (least deprived)	7.29 (1.30)	60
Quintile 2	7.42 (1.56)	60
Quintile 3	7.43 (1.53)	64
Quintile 4	7.33 (1.52)	67
Quintile 5 (most deprived)	7.56 (1.24)	57

Creating tables of publication quality

```
.display_mean_by_var <- function(output_means, group_by_var_lab, var_lab) {
  df_f <- flextable(output_means,
                    col_keys = c("group", "var_mean", "var_n")) %>%
    colformat_double(digits = 2) %>%
    add_header_row(values = c(" ", var_lab),
                   colwidths = c(1, 2)) %>%
    mk_par(j = 1, value = as_paragraph(group)) %>%
    mk_par(j = 2, value = as_paragraph(as_chunk(sprintf("%.2f", var_mean)), "  ",
                                       as_bracket(sprintf("%.2f", var_sd)))) %>%
    set_header_labels(group = group_by_var_lab,
                      var_mean = "Average (SD)",
                      var_n = "No. of applicants") %>%
    align(align = 'center', part = 'header') %>%
    vline(j = c(1), part = 'all', border = fp_border_default()) %>%
    hline(part = 'body', border = fp_border_default(color = 'gray', width = 0.5)) %>%
    bg(bg = 'wheat', part = 'header') %>%
    set_table_properties(layout = 'autofit') %>%
    set_caption(caption = "Applicants' performance on admissions measures",
                style = 'Table Caption')
  return(df_f)
}
```

Automating text and inline calculations with R Markdown



The results from Table 3.2 in this section indicate the following:

1. The number of applications **increased** from 358 to 376
2. The **Townsend Q1 / Q5 ratio** at application for the course is **1.6/1 compared to 2.1/1** for the wider University

```
1. The number of applications **'r if(n_prev_year < n_this_year) {str_c("increased from
", n_prev_year, " to ", n_this_year)} else if (n_prev_year > n_this_year)
{str_c("decreased from ", n_prev_year, " to ", n_this_year)} else {"stayed the
same"}'**.
```

1. The number of applications **decreased from 358 to 356**.

Regressions

Which applicant characteristics are associated with their on-course outcomes?

Table 2.3: Predictive validity for applicants at shortlisting stage

	Exam grade (Model 1)	Exam grade (Model 2)
Number of observations	638	620
Adjusted R-squared	35%	44%
GCSE A*s	1.14 (0.09)***	1.15 (0.09)***
School: muggle (ref: wizard)	-1.00 (0.36)**	-1.00 (0.39)*
Female gender (ref: male)		-0.05 (0.23)

Footnote: Multivariate regressions illustrating strength of relationship between shortlisting measures and on-course attainment. Please refer to Methods section for detail on how to interpret the results.

- estimates presented with standard errors
- numbers rounded to a pre-defined number of digits; trailing zero(es)
- asterisks indicate significance of associations (eg $***$ = $p<0.001$)
- significant coefficients colour-coded depending on direction of effect
- table incorporated into the .docx and automatically re-generated every time

Scalability scenario

Suppose we want to add an extra explanatory variable: gender (female/male)

Scalability scenario

Suppose we want to add an extra explanatory variable: gender (female/male)

```
xvars <- c("gcse", "school")
yvar <- "grade"

f <- as.formula(str_c(yvar, "~",
                      str_c(xvars, collapse = "+")))
glm(f, data = df_tofit)
```

	Exam grade (Model 1)
Number of observations	638
Adjusted R-squared	35%
GCSE A*s	1.14 (0.09)***
School: muggle (ref: wizard)	-1.00 (0.36)**

Scalability scenario

Suppose we want to add an extra explanatory variable: gender (female/male)

```
xvars <- c("gcse", "school")
yvar <- "grade"

f <- as.formula(str_c(yvar, "~",
                      str_c(xvars, collapse = "+")))
glm(f, data = df_tofit)
```

```
xvars <- c("gcse", "school", "gender")
yvar <- "grade"

f <- as.formula(str_c(yvar, "~",
                      str_c(xvars, collapse = "+")))
glm(f, data = df_tofit)
```

	Exam grade (Model 1)
Number of observations	638
Adjusted R-squared	35%
GCSE A*s	1.14 (0.09)***
School: muggle (ref: wizard)	-1.00 (0.36)**

	Exam grade (Model 2)
Number of observations	620
Adjusted R-squared	44%
GCSE A*s	1.15 (0.09)***
School: muggle (ref: wizard)	-1.00 (0.39)*
Female gender (ref: male)	-0.05 (0.23)

Summary

- Time investment at the start but leads to less error-prone and scalable outputs
- Potential scalability scenarios:
    - numerous repetitive outputs
    - new requirements for statistics to be generated / layout / colour scheme(s)
    - re-arrangement of items in the numbered lists
    - change of location and/or name of the admissions test
    - re-arrangement of tables/figures across the report (re-referencing)
    - ...
- Reproducibility of data and analysis code
- Anything repetitive can (should) be parameterised / translated into functions
- Emphasis on *automation*: Quarto vs R Markdown; integration with Python, Tableau, Power BI...

Thank you! Any questions?

Keep calm and carry on coding

GIF from https://media.giphy.com/media/13HgwGsXF0aiGY/giphy.gif