

Superiority in Nature: Typst in Production of TLFs

Tao Zhang, Phastar

ABSTRACT

Typst is a novel typesetting system developed in Rust, aimed at providing a robust alternative to both sophisticated systems like LaTeX and more user-friendly platforms such as Word. This paper introduces Typst as a replacement of SAS ODS and PROC REPORT to generate TLF outputs from pre-assembled datasets. The proposed approach leverages Typst to enhance the production of TLFs by offering better default layouts with automatic column width distribution and pagination, fast PDF generation, and seamless integration into existing SAS processes. The introduction of Typst simplifies the TLF production process, enhancing automation without compromising flexibility, thereby streamlining the workflow of TLF production.

INTRODUCTION

The ODS RTF/PDF destination and PROC REPORT offer a wide range of functionalities that enable SAS programmers to create high-quality tables and listings. However, the default layout of PROC REPORT is often not acceptable. Ensuring that the output presents data clearly and aesthetically usually requires significant effort. For instance, determining the appropriate column width often involves a process of continuous trial and error. Writing PROC REPORT code can be tedious and time-consuming, and the resulting programs can become complex and difficult to maintain. Programs often need to be rerun after updates to the shell or data, which may require further adjustments to column widths. In an industry increasingly focused on automation, it is challenging to automate processes based on PROC REPORT. For example, if we develop a listing automation tool based on PROC REPORT, users would still need to manually provide the column widths.

Typst is a novel typesetting system developed in Rust, aimed at providing a robust alternative to both sophisticated systems like LaTeX and more user-friendly platforms such as Word. It is open source and designed to be easy to learn, fast, and versatile. A good typesetting system is a perfect match for the production of TLFs. In this paper, we present a new Typst-based approach as a replacement for SAS ODS and PROC REPORT. The proposed approach leverages Typst to enhance the production of TLFs by offering better default layouts with automatic column width distribution and pagination, fast PDF generation, and seamless integration into existing SAS processes. We will first give a brief introduction to Typst and then cover the new TLF approach.

HOW TO USE TYPST

Typst is a markup language. We can use simple syntax to accomplish common layout tasks. The markup language is backed by a tightly integrated scripting language with built-in and user-defined functions. Typst compiler takes text files with markup in them and outputs PDFs. Here are the steps to produce a simple PDF using Typst.

Step 1: Download the compiler from <https://github.com/typst/typst/releases/>.

Name	Date modified	Type	Size
 typst.exe	3/16/2024 2:15 AM	Application	33,096 KB

Figure 1: The Typst compiler

Step 2: Create a text file and save as a .typ file.

```
t hello.typ x
paper > t hello.typ
1 Hello Typst!
2
```

Figure 2: Typst source file

Step 3: Run the compile command: `typst compile hello.typ`, it will produce `hello.pdf`.

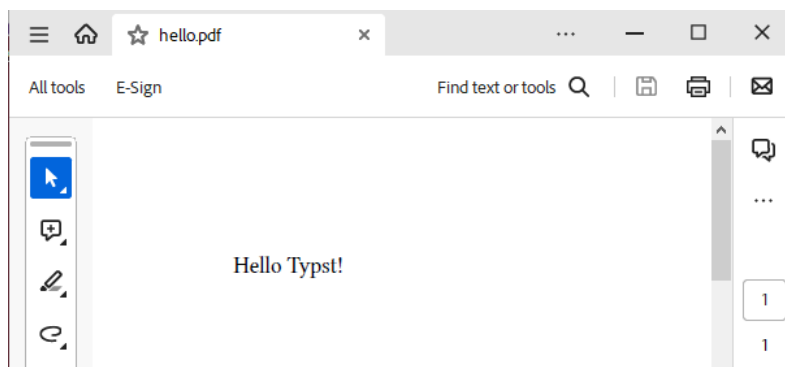


Figure 3: The PDF produced by Typst compiler

If we use VS Code and install two extensions (Typst LSP and Typst Preview), we can have nice syntax highlighting and a live preview. Figure 4 shows my screen while I am composing this paper.

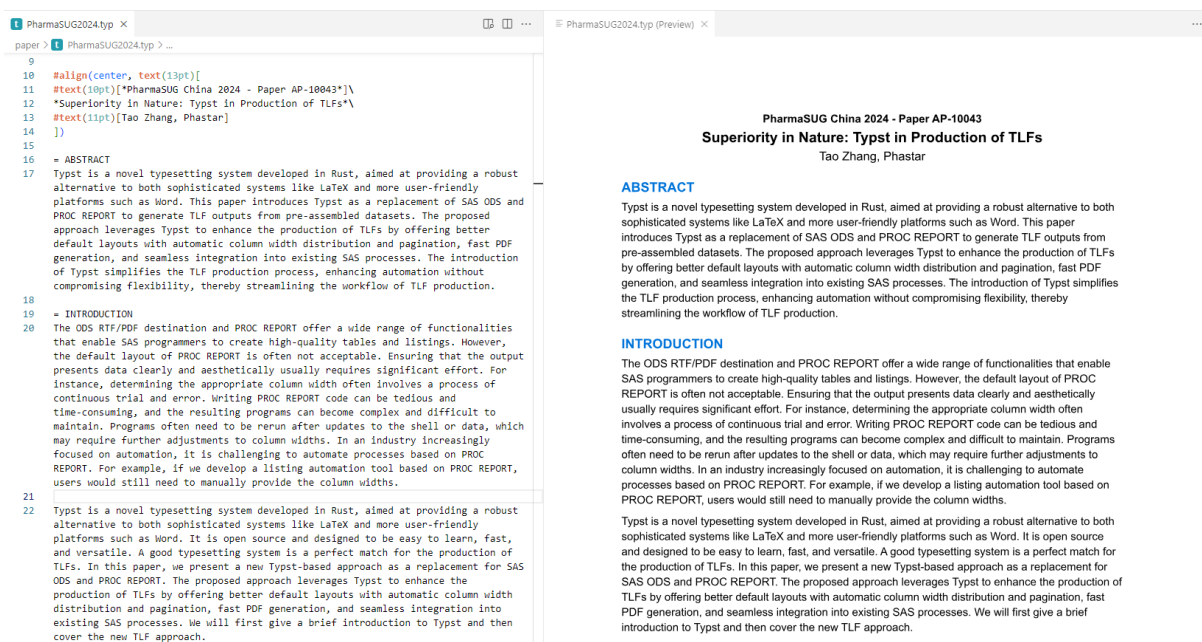


Figure 4: Typst live preview via VS Code

COMMON LAYOUT TASKS IN TYPST

Now that we know how to use Typst, we will demonstrate its strengths by showing how to achieve common TLF related layout tasks in Typst.

PAGE SETUP

It is very easy to customize pages, margins, headers, footers, and page numbers in Typst.

In the below code, we first set the font and font size. Then we setup the page with a header and footer and display the page number within the header. `#h(1fr)` is used to insert 1 fraction of the remaining horizontal space. Typst supports the following length units: points: 72pt, millimeters: 254mm, centimeters: 2.54cm, inches: 1in, relative to font size: 2.5em. The appearance of the page is shown in Figure 5.

```
#set text(font: "Times New Roman", size: 15pt)
#set page(
  width: 11in,
  height: 8.5in,
  margin: (top: 1in, left: 0.375in, bottom: 0.75in, right: 0.375in),
  header: [Study code: xxx#h(1fr)Page #counter(page).display("1 of 1", both: true)],
  footer: [Program: path/to/program#h(1fr)Executed: 2024-06-28T12:02:11]
)
```

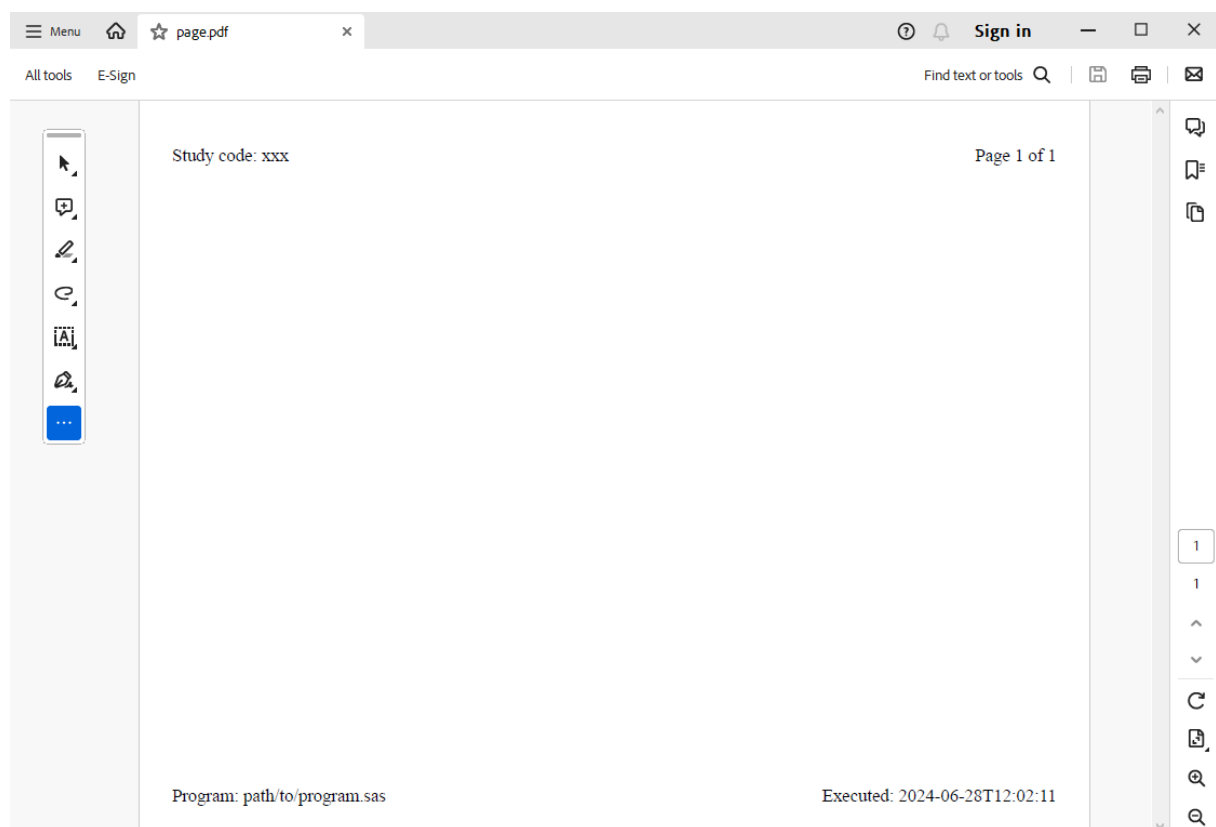


Figure 5: Page setup

TABLE

We can use the function `table` to create tables. This function is very powerful. It can achieve all kinds of table layout and the code is simple.

Below code demonstrates how to create a simple table in Typst. We call the `table` function and tell the function there are 9 columns and provide the content for each of the table cells. The contents of

table.header and table.footer will be repeated on each page if there are multiple pages. The parameter stroke and table.hline call are used to control borders. Note that Typst automatically distribute the space among the columns. This fantastic feature is very helpful in the production of TLFs. If we want to control the column widths we can pass different values to the parameter columns. See <https://typst.app/docs/guides/table-guide/> for details.

```
#table(
  columns: 9,
  stroke: none,

  table.header(
    table.hline(stroke: 1pt + black),
    "Subject", "Site", "Region", "Age(years)", "Sex", "Race",
    "Ethnic", "Baseline weight(kg)", "Baseline height(cm)",
    table.hline(stroke: 1pt + black),
  ),

  "101-301", "101", "North America", "57", "F", "White",
  "Not Hispanic or Latino", "58.1", "160.8",
  "102-303", "102", "North America", "61", "M", "White",
  "Not Reported", "104.9", "188",
  "111-302", "111", "North America", "43", "M", "White",
  "Not Hispanic or Latino", "85.5", "182.5",

  table.footer(
    table.hline(stroke: 1pt + black),
  ),
)
```

Subject	Site	Region	Age(years)	Sex	Race	Ethnic	Baseline weight(kg)	Baseline height(cm)
101-301	101	North America	57	F	White	Not Hispanic or Latino	58.1	160.8
102-303	102	North America	61	M	White	Not Reported	104.9	188
111-302	111	North America	43	M	White	Not Hispanic or Latino	85.5	182.5

USER-DEFINED FUNCTIONS

There are lots of built-in functions in Typst we can use to create and manipulate contents on the document. We can also define our own functions. Two examples are given below.

Achieve decimal alignment using a user-defined function dalign.

First, we define the function using below code.

```
#let dalign(txt, width1: 1fr, width2: 1fr) = {
  let txtparts = txt.replace(regex("[^\\d-]"), it => "\u{bb}" + it.text).split("\u{bb}")
  table(
    inset: 0pt,
    columns: (width1, width2),
    align: (right, left),
    txtparts.first(), txtparts.slice(1).join()
  )
}
```

Inside a call to table, we call the function dalign we just defined. The outputs are presented on the right side of the code. The decimal points are nicely aligned.

```
#set table(stroke: 1pt + rgb(0, 0, 0, 0%))
#table(
  dalign("0"),
  dalign("0.01"),
  dalign("0.001"),
  dalign("0.0001"),
  dalign("100"),
)
```

0
0.01
0.001
0.0001
100

We achieve decimal alignment by putting a two-column table inside table cells. In below code, we set the transparency of the borders to 25% so we can clearly see that we are having a table inside each cell. We can do this trick when we want to achieve any kind of alignment. For example, we can put a table as the footnote of a table so we can have the item numbers nicely aligned.

```
#set table(stroke: 1pt + rgb(0, 0, 0, 25%))
#table(
  dalign("0"),
  dalign("0.01"),
  dalign("0.001"),
  dalign("0.0001"),
  dalign("100"),
)
```

0
0.01
0.001
0.0001
100

Pack text together using a user-defined function packtext.

First, we define the function using below code.

```
#let packtext(list, indent: 0pt, sep: "/", indent1: 1.25em, indent2: 0.625em) = {
  let mod_list = if list.len() > 1 {list.map(it => it+sep)} else {list.map(it => it)}
  if list.len() > 1 {mod_list.last() = list.last()}

  let list_enum = mod_list.enumerate()
  table(
    stroke: none,
    inset: (left: indent, rest: 0pt),
    [#list_enum.map(
      it => par(
        first-line-indent: it.at(0)*indent1,
        hanging-indent: it.at(0)*indent1+indent2,
        it.at(1)).join()]
    )
}
```

Inside a call to table, we call the function packtext we just defined. The output are presented on the right side of the code. The three parts are packed together with a separator and 2 levels of indentation.

```
#table(
  columns: 6cm,
  packtext(
    ("Body system",
     "Dictionary-derived term",
     "Reported term for the adverse event"))
)
```

Body system/ Dictionary-derived term/ Reported term for the adverse event
--

TLF PRODUCTION USING TYPST

Typst is versatile and the code is easy to generate from SAS. We can develop SAS macros to generate Typst code and call the compiler to produce PDF files. We have put together a working set of macros.

Level 1 macros:

- `sastyp_build`: build PDF from Typst code in SAS datasets
- `sastyp_page`: produce Typst code for page settings and global code
- `sastyp_table`: produce Typst code for a table from a SAS dataset
- `sastyp_align`: produce Typst code that call the user-defined function `dalign`
- `sastyp_packtext`: produce Typst code that call the user-defined function `packtext`

Level 2 macro:

- `typrpt`: call level 1 macros to produce TLFs

The level 1 macros are designed for general purpose. We can develop any other applications based on the level 1 macros. For example, we can develop tools which produce narratives, patient profiles and review's guides for eCRT package. This paper will only cover TLFs. We give some examples of the usage of `%typrpt`.

Example 1: A simple listing

Below SAS program first prepare a dataset which contains each column for a listing and call `%get_tfh` to get titles, footnotes and column headers from a central repository. Then it calls the `%typrpt` macro to produce `dm100.pdf`. This is the most simple case of `%typrpt`. The macro automatically handle the layout of the listing. Figure 6 shows the output.

```
%let id=dm100;

data dm100;
  set adam.adsl;
  where ittfl = 'Y';
  col1 = subjid;
  col2 = siteid;
  col3 = region1;
  if age ne . then col4 = put(age, best.-l);
  col5 = sex;
  array race[*] race1-race7;
  array race_dis[*] $200 raced_1-raced_7;
  do i=1 to dim(race);
    if race[i] = 'Y' then race_dis[i] = vlabel(race[i]);
  end;
  col6 = catx(';', of race_dis[*]);
  col7 = ethnic;
  if wtbl ne . then col8 = put(wtbl, best.-l);
  if htbl ne . then col9 = put(htbl, best.-l);
  if bmibl ne . then col10 = put(bmibl, best.-l);
  keep col;;
run;

%get_tfh(from=.\study_tfh.xlsx, id=&id.);

%typrpt(
  data=dm100
  ,out=&id..pdf
);
```

Listing 16.2.4.1
Demographic and baseline characteristics
Intent-to-treat analysis set

Subject	Site	Region	Age(years)	Sex	Race	Ethnic	Baseline weight(kg)	Baseline height(cm)	Baseline body mass index(kg/ m2)
101-301	101	North America	57	F	White	Not Hispanic or Latino	58.1	160.8	20.8
102-303	102	North America	61	M	White	Not Reported	104.9	188	29.7
111-302	111	North America	43	M	White	Not Hispanic or Latino	85.5	182.5	25.2
111-305	111	North America	86	F	Other	Not Hispanic or Latino	75.7	165	27.8
141-301	141	North America	51	F	White	Not Hispanic or Latino	71.3	155.6	29.9
150-302	150	North America	71	M	White	Unknown	62.5	170	22.3
160-301	160	North America	69	F	White	Not Hispanic or Latino	48.9	152.4	20.9
457-302	457	Europe	68	M	White	Not Reported	50	168	17.7

F Female; M Male; m2 Square metre.

Program: path/to/program.sas

Executed: 2024-06-30T20:25:52

Figure 6: Output of %typrpt for dm100

Example 2: An AE listing with text packed together and a footnote page

In below SAS code, col4 is produced by calling the macro %sastyp_packtext. We suppress repeating values of col1, col2 and col3 by setting the parameter idcols. And the parameters fpage and fpagenote are used to create a footnote page. Figure 7 shows the footnote page and Figure 8 shows the second page of ae100.pdf.

```
%let id=ae100;

* prepare the text to report for each column, use '\n' to insert linebreak;
data ae100;
  set adam.ae;
  if _n_ <=10;
  col1=strip(scan(usubjid, 2, '/'));
  col2=procase(aphase);
  col3=strip(put(age,best.))||'/\n' || strip(sex)||'/\n' || strip(procase(race));
  col5=put(aestdy,4.)||'/\n' || put(adurn,4.);
  col6=strip(put(atoxgrn,best.));
  col7=arel;
  col8=aacn;
  col9=arel;
  col10=aeout;
  space_after = "10pt";
  proc sort;
  by usubjid trtsdt astdt aedecod;
run;

%sastyp_packtext(
```

```
inds=ae100
,invars=aebodsys aedecod aeterm
,outvar=col4
);

data body;
  retain col1-col10;
  set ae100;
  keep col: space_after;
run;

%get_tfh(from=.\study_tfh.xlsx, id=&id.);

%typrpt(
  data=body
  ,idcols=col1 col2 col3
  ,fpage=Y
  ,fpagenote=Note: footnotes are listed on the 1st page.
  ,out=ae100.pdf
);
```

Study code: xxx
Dry Run 1

Confidential

Listing 16.2.7.1
All Adverse events
Safety analysis set

Page 1 of 4
DCO: 2024-06-26

DNC = Do Not Change; Y = Yes

- [a] Coded using MedDRA version xxx. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequae doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.
 - [b] Study day relative to first dose date. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.
 - [c] This is footnote c Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequae doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.
 - [d] This is footnote d Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequae doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.
 - [e] This is footnote e Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aequae doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguere possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et.
-

Figure 7: The footnote page of ae100.pdf

The content of the above footnote page is provided as below in the central repository. The `#lorem()` call is just to add some test text. We can control almost everything on the page by inserting Typst code directly. This ensures flexibility.

```
#table(columns: (auto, 1fr), align: (right, left),
table.cell(colspan: 2, align: left, text("DNC = Do Not Change; Y = Yes")),
[[a]], [Coded using MedDRA version xxx. #lorem(40)],
[[b]], [Study day relative to first dose date. #lorem(20)],
[[c]], [This is footnote c #lorem(50)],
[[d]], [This is footnote d #lorem(40)],
[[e]], [This is footnote e #lorem(80)],
)
```

Study code: xxx
Dry Run 1

Confidential

Page 2 of 4
DCO: 2024-06-26

Listing 16.2.7.1
All Adverse events
Safety analysis set

Subject identifier	Treatment period	Age/ Sex/ Race	Body system/ Dictionary-derived term [a]/ Reported term for the adverse event	Study day of start of adverse event [b]/ Duration of adverse event (days)	This is a Spanning Header				
					CTCAE grade	Serious event	Action taken	Causality	Outcome
E1301002	On Treatment	47/ F/ Asian	Blood and lymphatic system disorders/ Anaemia/ anemia	9/ 35	2	Y	DNC	Y	RECOVERED/ RESOLVED WITH SEQUELAE
			Metabolism and nutrition disorders/ Hyperuricaemia/ hyperuricemia	12/ 59	1	N	DNC	N	RECOVERED/ RESOLVED
			Investigations/ Blood creatinine increased/ Creatinine increased	16/ 6	1	Y	DNC	Y	RECOVERED/ RESOLVED
			Investigations/ Blood urea increased/ Increased urea	16/ 6	1	Y	DNC	Y	RECOVERED/ RESOLVED

Note: footnotes are listed on the 1st page.

Program: path/to/program.sas

Executed: 2024-06-28T10:07:00

Figure 8: The second page of ae100.pdf

The content of the spanning header is provided as below in the central repository.

```
#table.cell(colspan: 5, align: center, stroke: (bottom: 1pt+black))[This is a Spanning Header]
```

Example 3: An AE by SOC/PT count table

In below SAS code, %sastyp_packtext is called to create indentation on col1. We call %sastyp_align to set decimal alignment for the number columns. Three special variables space_before_page, row_before_page and space_after are used. space_before_page is used to insert some space below the column header. row_before_page is used to add continue text when an SOC is displayed across multiple pages. space_after is used to insert some space after each SOC. Other special variables include: page, subtitle, space_before, row_before, row_after and imagepath.

Figure 9 shows the first page and Figure 10 shows the second page of ae200.pdf.

```
%let id=ae200;

data all_1;
  set count_any count_soc count_pt;*Code that produces these datasets are not included;
  length col1 $200;
  if section = 1 then col1 = "Any TEAE";
  if level = 1 then col1 = aebodsys;
  if level = 2 then col1 = aedecod;
  proc sort;
  by section aebodsys level aedecod;
run;

data all_2;
  set all_1;
  by section aebodsys level aedecod;

  if level = 2 then indent = '10pt';
  space_before_page = "10pt";
  if not first.aebodsys then row_before_page = strip(aebodsys)||' (cont.)';
  if last.aebodsys then space_after = "15pt";
run;

%sastyp_packtext(inds=all_2,invars=col1,indent=indent)*;

%sastyp_align(inds=all_2,invars=n_1 pcnt_1 n_2 pcnt_2 n_9 pcnt_9, width1=25pt, width2=25pt)*;

data final;
  retain col1 n_1 pcnt_1 n_2 pcnt_2 n_9 pcnt_9;
  set all_2;
  keep col1 n_1 pcnt_1 n_2 pcnt_2 n_9 pcnt_9 space_before_page row_before_page space_after;
run;

%get_tfh(from=.\study_tfh.xlsx, id=&id.);

%typrpt(
  data=final
  ,align=left center center center center center center
  ,out=ae200.pdf
);
```

Table 14.1.3.4
Treatment-Emergent Adverse Events by System Organ Class and Preferred Term
Safety Set

System Organ Class (SOC) Preferred Term (PT)	Drug 1 N = 6		Drug 2 N = 2		Total N = 8	
	n	%	n	%	n	%
Any TEAE	5	83.3	2	100	7	87.5
Blood and lymphatic system disorders	0	0	1	50.0	1	12.5
Anaemia	0	0	1	50.0	1	12.5
Gastrointestinal disorders	2	33.3	1	50.0	3	37.5
Abdominal distension	0	0	1	50.0	1	12.5
Abdominal pain	1	16.7	1	50.0	2	25.0
Ascites	0	0	1	50.0	1	12.5
Constipation	1	16.7	1	50.0	2	25.0
Diarrhoea	1	16.7	1	50.0	2	25.0
Flatulence	0	0	1	50.0	1	12.5
Ileus	0	0	1	50.0	1	12.5
Large intestine perforation	0	0	1	50.0	1	12.5
Nausea	1	16.7	0	0	1	12.5
Stomatitis	0	0	1	50.0	1	12.5

N = Number of participants in the Safety Set. n = Number of participants with observed data. TEAE = Treatment-Emergent Adverse Event.
Note 1: The Safety Set consists of participants who have received at least one dose of study drug.
Note 2: Percentages are calculated based on the total number of participants in the Safety Set.

Figure 9: The first page of ae200.pdf

Table 14.1.3.4
Treatment-Emergent Adverse Events by System Organ Class and Preferred Term

System Organ Class (SOC) Preferred Term (PT)	Safety Set					
	Drug 1 N = 6		Drug 2 N = 2		Total N = 8	
	n	%	n	%	n	%
Gastrointestinal disorders (cont.)						
Vomiting	1	16.7	1	50.0	2	25.0
General disorders and administration site conditions						
Chills	1	16.7	0	0	1	12.5
Influenza like illness	1	16.7	0	0	1	12.5
Pyrexia	0	0	1	50.0	1	12.5
Secretion discharge	0	0	1	50.0	1	12.5
Infections and infestations						
COVID-19	0	0	1	50.0	1	12.5
Hordeolum	1	16.7	0	0	1	12.5
Investigations						
Alanine aminotransferase increased	1	16.7	0	0	1	12.5
Aspartate aminotransferase increased	0	0	1	50.0	1	12.5
Blood creatine increased	0	0	1	50.0	1	12.5

N = Number of participants in the Safety Set. n = Number of participants with observed data. TEAE = Treatment-Emergent Adverse Event.
Note 1: The Safety Set consists of participants who have received at least one dose of study drug.
Note 2: Percentages are calculated based on the total number of participants in the Safety Set.
Program: path/to/program.sas Executed: 2024-06-28T08:19:45

Figure 10: The second page of ae200.pdf

Example 4: A figure

In this example, we call %typrpt to present a figure. We create the figure and save the figure to a .png file. %typrpt can add the figure to the final PDF file since the path is provided by the special variable imagepath. Figure 11 shows the output of f100.pdf.

```
%let id=f100;

ods _all_ close;
ods listing;
ods graphics on / outputfmt=png imagename="f100" reset=index border=off;
proc sgplot data= summx_cnt;
  where folderseq<=53;
  styleattrs datacolors=(vigb vipk teal) datacontrastcolors=(cx13478c cxcc1b3b cx008080);
  vbar foldername / response=n group=trtname groupdisplay=cluster dataskin=none;
  xaxis label="Study Visit" display=(nolabel) values=(&vis1.);
  yaxis label="Number of subjects (n)" grid max=&max.;
  keylegend / title="Treatment";
  xaxistable n / title="Treatment" location=inside colorgroup=trtname;
run;

data figures;
  imagepath="f100.png";
run;

%get_tfh(from=.\study_tfh.xlsx, id=&id.);

%typrpt(
  data=figures
```

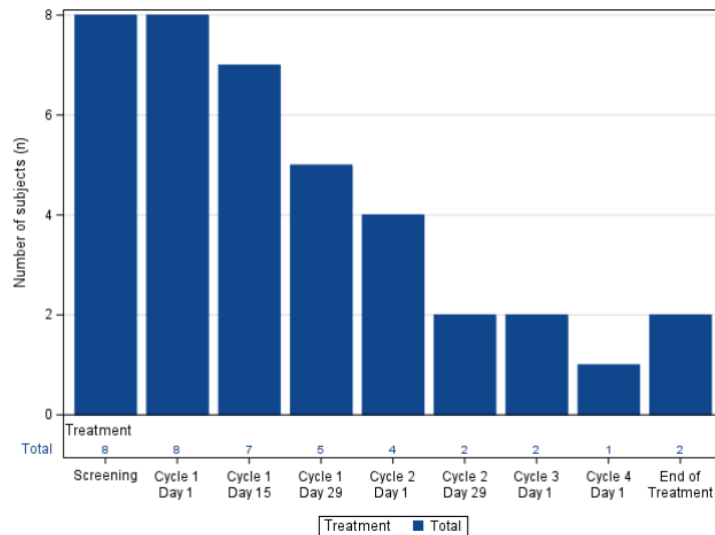
```
,out=f100.pdf
);
```

Study code: xxx
Dry Run 1

Confidential

Page 1 of 1
DCO: 2024-06-26

Figure 14.1.1
Bar Graph for Study Visit
Intent-to-Treat Set



The Intent-to-Treat Set consists of enrolled participants who have been randomized to a study drug.

Note 1: 'n' represents number of subjects with non-missing screening and visit values.

Program: path/to/program.sas

Executed: 2024-06-28T12:02:11

Figure 11: Output of %typrpt for f100

CONCLUSION

Through these examples, we can conclude that the introduction of Typst simplifies the TLF production process, enhancing automation without compromising flexibility, thereby streamlining the workflow of TLF production. This approach can be easily integrated into existing process for TLFs and the macros would serve well as the report module of other automation tools.

REFERENCES

Typst documentation available at <https://typst.app/docs>.

ACKNOWLEDGMENTS

The author would like to thank Yong Cao and Lu Zhang for their great advice.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Tao Zhang

Phastar

tao.zhang@phastar.com

www.phastar.com

Any brand and product names are trademarks of their respective companies.