



WORLDWIDE
CLINICAL TRIALS

PHUSE EU Connect,
November 2020

TT05: Using R from end to end: CDISC SDTM and ADaM into .xpt and output generation

Antonio Rodríguez

✉ antonio.rodriguez@worldwide.com

📶 Online

Presentation Content

- Introduction



Presentation Content

- Introduction
- Step 0: Get data into R



Presentation Content

- Introduction
- Step 0: Get data into R
- Step 1: SDTM creation



Presentation Content

- Introduction
- Step 0: Get data into R
- Step 1: SDTM creation
- Step 2: ADaM creation



Presentation Content

- Introduction
- Step 0: Get data into R
- Step 1: SDTM creation
- Step 2: ADaM creation
- Step 3: Outputs creation



Presentation Content

- Introduction
- Step 0: Get data into R
- Step 1: SDTM creation
- Step 2: ADaM creation
- Step 3: Outputs creation
- Conclusion





WORLDWIDE
CLINICAL TRIALS

Introduction

Background

Food and Drug Administration (FDA) pointed out in the clarification statement on 6th May 2015

Statistical Software Clarifying Statement

FDA does not require use of any specific software for statistical analyses, and statistical software is not explicitly discussed in Title 21 of the Code of Federal Regulations [e.g., in 21CFR part 11]. However, the software package(s) used for statistical analyses should be fully documented in the submission, including version and build identification.

Alternative Software



We are going to focus on R, one of the multiple statistical software out there.

R is a free software environment for statistical computing, data manipulation and charting, widely used in the scientific world.

R Potential

To illustrate that R is suitable for clinical trials, we are going to use it to create CDISC standards SDTM and ADaM and also we are going to create a Table of the TFL as an example.

Shopping list

- | | |
|--------------------|----------------------|
| ▶ haven: v2.2.0 | ▶ SASxport: v1.6.0.1 |
| ▶ arsenal: v3.4.0 | ▶ xtable: v1.8-4 |
| ▶ openxlsx: v4.1.4 | ▶ rtf: v0.4-14 |



WORLDWIDE
CLINICAL TRIALS

Step 0: Get data into R

Usual Case

The usual case is having data on a Electronic Data Capture (EDC) that export in a SAS compatible format. In that case, we can use *haven* package to load the data.

```
1 #install.packages("haven") #First time only
2 #load package
3 library(haven)
4 #load .sas7bdat & associated .sas7bcat
5 path<-system.file("iris",
6                   "iris.sas7bdat",package="haven")
7 read_sas(path)
```

Mock Data Case

We do not have data of a real study to work with, so we are going to use the *arsenal* package to use the *mockstudy* data (around 1500 rows).

```
1 #install.packages("arsenal") #First time only
2 library(arsenal)
3 mockstudy[1,]
4 > case      age      arm      sex      race      fu.time      fu.stat
5 1 110754      67 F: FOLFOX      Male Caucasian      922      2
6 ps      hgb      bmi      alk.phos      ast      mdquality.S      age.ord
7 1      0      11.5 25.09861      160      35      NA      60-69
```



WORLDWIDE
CLINICAL TRIALS

Step 1: SDTM creation

Mock Data to SDTM

To show that we can use R for SDTM creation, we are going only to produce *DM* domain.

As you may spotted from the previous sample of data, we are not having all the required fields for DM, so we are going to simulate the ones missing.

Simulate Data - Part 1

```
1 n=length(mockstudy[,1])
2 #Repeat string for number of rows present mock study
3 STUDYID=rep("MCK-001",n)
4 DOMAIN=rep("DM",n)
5 AGEU=rep("YEARS",n)
6 COUNTRY=rep("USA",n)
7 #AGE: Use mockstudy age column
8 AGE=mockstudy$age
9 #SEX: Use mock study sex first letter
10 SEX=substr(mockstudy$sex, 1,1)
11 #SUBJID: Use mockstudy case column as subjid adding 0 until
12 #length is 6
13 SUBJID=formatC(mockstudy$case, width=6, format="d", flag="0")
14 #USUBJID: Concatenate STUDYID with SUBJID separated by "-"
15 USUBJID=paste(STUDYID,SUBJID, sep="-")
16 #SITEID: Create a random site 3 digits between 101 and 120
17 SITEID=as.character(sample(101:120, n, replace=TRUE))
```

Simulate Data - Part 2 (Treatments)

```
1 #ARM: Use the three arm variable of mockstudy
2 ARM      = mockstudy$arm
3 #ARMCD: Set to first letter of ARM
4 ARMCD = substr( ARM , 1 , 1)
5 ACTARM = ARM
6 ACTARMCD = ARMCD
```

Simulate Data - Part 3 (Race)

```

1 table(mockstudy$race, exclude = NULL)
2 > African-Am      Asian      Caucasian      Hawaii/Pacific
3      115          18      1288              5
4      Hispanic      Native-Am/Alaska      Other      <NA>
5      54              5              7              7
6 RACE = ifelse(is.na(mockstudy$race),"UNKNOWN",
7             ifelse(mockstudy$race=="Caucasian"|
8                   mockstudy$race=="Hispanic","WHITE",
9             ifelse(mockstudy$race=="African-Am",
10                  "BLACK OR AFRICAN AMERICAN",
11             ifelse(mockstudy$race=="Asian","ASIAN",
12             ifelse( mockstudy$race=="Hawaii/Pacific",
13                  "NATIVE HAWAIIAN OR OTHER PACIFIC ISLANDER",
14             ifelse(mockstudy$race=="Native-Am/Alaska",
15                  "AMERICAN INDIAN OR ALASKA NATIVE","OTHER"))))))))

```

Simulate Data - Part 4 (Date Variables)

```
1 #RFSTDTC: Random date of 2010 year
2 RFSTDTC =as.Date("2010-01-01")
3           + sample(1:365, n,replace=TRUE)
4 #RFXSTDTC: 5 to 10 days after RFSTDTC
5 RFXSTDTC= RFSTDTC +sample(5:10, n,replace=TRUE)
6 #RFENDTC: RFXSTDTC + follow-up time in days
7 RFENDTC = RFXSTDTC + mockstudy$fu.time
8 #DTHFL: Follow-up status = 1 set to "Y", else "N"
9 DTHFL = ifelse(mockstudy$fu.stat == 1, "Y", "N")
10 #DTHDTC: Fill Death date when needed with RFENDTC, else missing
11 DTHDTC =as.Date(ifelse(mockstudy$fu.stat == 1, RFENDTC, NA),
12               origin ="1970-01-01")
13 #RFXENDTC: If not death, between 20-30 days before RFENDTC
14 RFXENDTC = RFENDTC + ifelse( mockstudy$fu.stat == 1, 0,
15                             sample(-30:-20, n, replace = TRUE))
16 RFICDTC  = RFSTDTC
17 RFPENDTC = RFENDTC
```

Simulate Data - Part 5 (Final)

```

1 RFSTDTC =format(RFSTDTC ,"%Y- %m- %d")
2 RFENDTC =format(RFENDTC ,"%Y-%m-%d")
3 RFXSTDTC =format(RFXSTDTC ,"%Y- %m-%d")
4 RFXENDTC =format(RFXENDTC ,"%Y-%m-%d")
5 RFICDTC =format(RFICDTC,"%Y-%m-%d")
6 RFPENDTC =format(RFPENDTC ,"%Y-%m-%d")
7 DTHDTC =format(DTHDTC ,"%Y-%m-%d")
8 SDTM.DM =data.frame(DOMAIN, STUDYID, USUBJID, SUBJID,
9                      RFSTDTC, RFENDTC, RFXSTDTC, RFXENDTC,RFICDTC,
10                     RFPENDTC, DTHDTC, DTHFL, SITEID, AGE, AGEU,
11                     SEX, RACE, ARMCD, ARM, ACTARMCD, ACTARM, COUNTRY)
12 SDTM.DM[1,]
13   DOMAIN STUDYID      USUBJID SUBJID   RFSTDTC   RFENDTC
14 1      DM MCK-001 MCK-001-110754 110754 2010-01-01 2012-07-17
15   RFXSTDTC RFXENDTC   RFICDTC RFPENDTC   DTHDTC DTHFL SITEID
16 1 2010-01-07 2012-06-20 2010-01-01 2012-07-17      NA    N    104
17   AGE AGEU   SEX RACE ARMCD      ARM ACTARMCD ACTARM COUNTRY
18 1   67 YEARS    M WHITE    F F: FOLFOX      F F: FOLFOX    USA

```

Metadata

Now that we have the data ready, we need to apply all metadata (e.g. labels, sorting of variable in the dataset, sort the dataset by key variables, etc.).

There are different approaches about how to handle those. We are going to use an Excel with the Pinnacle 21 specifications structure.

A	B	C	D	E	F	G	H	I
Dataset	Description	Class	Structure	Purpose	Key Variables	Repeating	Reference Data	Comment
TA	Trial Arms	Trial Design	One record per planned Element per Arm	Tabulation	STUDYID, ARMCD, TAETORD	No	Yes	
TE	Trial Elements	Trial Design	One record per planned Element	Tabulation	STUDYID, ETCOD	No	Yes	
TI	Trial Inclusion/Exclusion Criteria	Trial Design	One record per I/E criterion	Tabulation	STUDYID, IETESTCD	No	Yes	

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Order	Dataset	Variable	Label	Data Type	Length	Significant Digits	Format	Mandatory	Codelist	Origin	Pages	Method	Predecessor	Role	Comment
1	DM	STUDYID	Study Identifier												
2	DM	DOMAIN	Domain Abbreviation												
3	DM	USUBJID	Unique Subject Identifier												
4	DM	SUBJID	Subject Identifier for the Study												

Excel on R

There are different ways to handle Excels from R, such as the base *read.table*, but we are going to use *openxlsx* instead.

```
1 library(openxlsx)
2 #Point where the file is located
3 xlsxFile = "C:/Users/USER_NAME/Desktop/Define.xlsx"
4 #Load subset of sheets of Domains we are interested
5 Datasets = subset(read.xlsx(xlsxFile,
6                             sheet = "Datasets"), Dataset == "DM")
7 Variables = subset(read.xlsx(xlsxFile,
8                             sheet = "Variables"), Dataset == "DM")
```

Sorting Variables and Data

```
1 #uppercase columna names (R is key sensitive)
2 names(SDTM.DM)= toupper(names(SDTM.DM))
3 #Sort SDTM.DM, using Key Variables list
4   #Split by", "
5 Key =unlist(strsplit(
6           as.character(Datasets["Key.Variables"]),","))
7 #Remove any space in vars
8 Key = gsub(" ", "",Key)
9 #Get row index ordered
10 index<-with(SDTM.DM,order(eval(parse(text=Key))))
11   #Change row position base on index
12 SDTM.DM = SDTM.DM[index, ]
13 #Resort variable columns inside data frame
14 DM = SDTM.DM[, Variables$Variable]
```

SASxport Package

By default, R do not have labels for data frames or variables, so we use *SASxport* package to do that and be able to create *.xpt* files.

```
1 #install.packages("SASxport") it's needed the first time
2 library( SASxport )
3 #Apply Variables labels
4 label(DM)=as.list(Variables$Label)
5 #Apply Dataset label
6 label(DM) = Datasets$Description
```

Export to .XPT

```

1 #Point file to be created
2 xptdm ="C:/Users/USER_NAME/Desktop/DM.xpt"
3 #Store xpt file
4 write.xport(DM,
5             file= xptdm ,
6             autogen.formats=FALSE,
7             sasVer="5.0"
8 )

```

Study Identifier	Domain Abbreviation	Unique Subject Identifier	Subject Identifier for the Study	Subject Reference Start Date/Time	Subject Reference End Date/Time	Date/Time of First Study Treatment	Date/Time of Last Study Treatment	Date/Time of Informed Consent	Date/Time of End of Participation	Date/Time of Death	Subject Death Flag	Study Site Identifier	Age	Age Units	Sex	Race	Planned Arm Code	Description of Planned Arm	Actual Arm Code
MCK-001	DM	MCK-001-076170	076170	2010-01-01	2010-06-09	2010-01-11	2010-05-11	2010-01-01	2010-06-09		N	103	52	YEARS	M	WHITE	A	A: IFL	A
MCK-001	DM	MCK-001-076240	076240	2010-01-01	2012-01-01	2010-01-09	2011-12-12	2010-01-01	2012-01-01		N	112	65	YEARS	M	WHITE	A	A: IFL	A
MCK-001	DM	MCK-001-076431	076431	2010-01-01	2011-02-11	2010-01-08	2011-01-21	2010-01-01	2011-02-11		N	117	62	YEARS	F	WHITE	A	A: IFL	A
MCK-001	DM	MCK-001-076712	076712	2010-01-01	2011-01-19	2010-01-06	2010-12-22	2010-01-01	2011-01-19		N	112	52	YEARS	M	WHITE	A	A: IFL	A
MCK-001	DM	MCK-001-076780	076780	2010-01-01	2011-03-20	2010-01-09	2011-02-26	2010-01-01	2011-03-20		N	115	59	YEARS	F	WHITE	A	A: IFL	A
MCK-001	DM	MCK-001-077066	077066	2010-01-01	2010-04-15	2010-01-11	2010-03-20	2010-01-01	2010-04-15		N	117	61	YEARS	F	WHITE	A	A: IFL	A
MCK-001	DM	MCK-001-077516	077516	2010-01-01	2011-07-19	2010-01-06	2011-06-27	2010-01-01	2011-07-19		N	106	67	YEARS	M	WHITE	A	A: IFL	A





WORLDWIDE
CLINICAL TRIALS

Step 2: ADaM creation

ADaM workflow

The process to create *ADAM.ADSL* is really similar to the one that we already show to *SDTM.DM*.

We are going to create the needed variables and apply again all the metadata from an excel with the same structure, so a lot of the code is recycled and can be easily automatised by a function.

Derive ADaM Data

For *ADAM.ADSL* most of variables are conditionally required, and with mock data is a bit harder to decided which ones should be included. As minimum example we created those

```
1 #Load predecessor dataset
2 ADAM.ADSL = DM
3 #Create new vars
4 ADAM.ADSL$FASFL = "Y"
5 ADAM.ADSL$SAFFL = "Y"
6 ADAM.ADSL$TRT01P = ADAM.ADSL$ARM
7 ADAM.ADSL$TRTSDT = as.Date(ADAM.ADSL$RFXSTDTC)
8 ADAM.ADSL$TRTEDT = as.Date(ADAM.ADSL$RFXENDTC)
```

Metadata (Part 1)

```

1 #Point where the file is located
2 xlsx File="C:/Users/USER_NAME/Desktop/Define.xlsx"
3 #Load subset of each of the sheets in the Domain we are interested.
4 Datasets =subset(read.xlsx(xlsxFile, sheet = "Datasets"),
5                   Dataset == "ADSL")
6 Variables =subset(read.xlsx(xlsxFile, sheet = "Variables"),
7                     Dataset == "ADSL")
8 #upper case columna names (R is key sensitive)
9 names(ADAM.ADSL)= toupper(names(ADAM.ADSL))
10 #Sort ADAM.ADSL, using Key Variables list
11   #Split by ","
12 Key =unlist(strsplit(
13             as.character(Datasets["Key.Variables"]),","))
14 #Remove any space in vars
15 Key = gsub(" ", "", Key)
16 #Get row index ordered
17 index <- with(ADAM.ADSL,order(eval(parse(text=Key))))
18 #Change row position base on index
19 ADAM.ADSL = ADAM.ADSL[index, ]
20 #Resort variable columns inside data frame
21 ADSL = ADAM.ADSL[,Variables$Variable]

```

Metadata (Part 2)

```

1 #Apply Variables labels
2 label(ADSL)=as.list(Variables$Label)
3 #Apply Dataset label
4 label(ADSL) = Datasets$Description
5 #Point file to be created
6 xptadsl ="C:/Users/USER_NAME/Desktop/ADSL.xpt"
7 #Store xpt file
8 write.xport(ADSL,
9             file= xptadsl ,
10             autogen.formats=FALSE,
11             sasVer="5.0"
12 )

```

	Study Identifier	Unique Subject Identifier	Subject Identifier for the Study	Study Site Identifier	Age	Age Units	Sex	Race	Full Analysis Set Population Flag	Safety Population Flag	Planned Arm Code	Description of Planned Arm	Planned Treatment for Period 01	Subject Reference Start Date/Time	Subject Reference End Date/Time	Date/Time of First Study Treatment	Date of First Exposure to Treatment	Date/Time of Last Study Treatment	Date/Time of Informed Consent
1	MCK-001	MCK-001-076170	076170	111	52	YEARS	M	WHITE	Y	Y	A	A: IFL	A: IFL	2010-03-20	2010-08-26	2010-03-30	18351	2010-07-27	2010-03-20
2	MCK-001	MCK-001-076240	076240	104	65	YEARS	M	WHITE	Y	Y	A	A: IFL	A: IFL	2010-08-24	2012-08-23	2010-09-01	18506	2012-07-28	2010-08-24
3	MCK-001	MCK-001-076431	076431	106	62	YEARS	F	WHITE	Y	Y	A	A: IFL	A: IFL	2010-09-22	2011-11-05	2010-10-02	18537	2011-10-08	2010-09-22
4	MCK-001	MCK-001-076712	076712	106	52	YEARS	M	WHITE	Y	Y	A	A: IFL	A: IFL	2010-06-20	2011-07-09	2010-06-26	18439	2011-06-19	2010-06-20
5	MCK-001	MCK-001-076780	076780	120	59	YEARS	F	WHITE	Y	Y	A	A: IFL	A: IFL	2010-10-01	2011-12-18	2010-10-09	18544	2011-11-20	2010-10-01



WORLDWIDE
CLINICAL TRIALS

Step 3: Outputs creation

Table generation

We are going to create a table, since the process to create a Table or Listing is basically the same.

L^AT_EX

- ▶ No editable (*.pdf*)
- ▶ Broadly used in science

RTF

- ▶ Editable
- ▶ Habitual delivery format

Both are free and based on plain text, so they are light.

Table generation

We are going to create a table, since the process to create a Table or Listing is basically the same.

L^AT_EX

- ▶ No editable (*.pdf*)
- ▶ Broadly used in science

L^AT_EX

Also extremely flexible
(Done on L^AT_EX)

format

Both are free and based on plain text, so they are light.

Categorical stats

R is really easy to use to display basic stats

```
1 table (ADSL$SEX ,ADSL$ARM)
2      A: IFLF:      FOLFOX G: IROX
3      F      151      280      152
4      M      277      411      228
```

But they do not show the typical display of percentages, big Ns, etc. so we are going to create an small function to display this kind of data in the usual way.

Categorical stats in macro

```

1 CatbyCol<-function(treat, Cat){
2   ncat =length(levels(as.factor(Cat)))
3   #Get N for percentages calc
4   BigN =table(treat)
5   #Contingency table
6   Cont =table(Cat,treat)
7   #Percentages calculations
8   BigN_m =rep(1,ncat) %*% t(BigN)
9   Perc =format(round( Cont*100/BigN_m, 1),nsmall = 1)
10  #Convert Cont to characters
11  Cont_c=format(Cont,nsmall=0)
12  #Merge them
13  data_t=matrix(paste(Cont_c,'(',Perc,'\\%)',sep =""),
14                ,nrow= ncat )
15  colnames(data_t) = colnames(Cont)
16  rownames(data_t) = rownames(Cont)
17  return(data_t)
18 }
19 Cat = CatbyCol(treat = ADSL$ARM,Cat = ADSL$SEX)
20 Cat
21   A: IFL          F: FOLFOX          G: IROX
22 F "151(35.3\\%)"  "280(40.5\\%)"  "152(40.0\\%)"
23 M "277(64.7\\%)"  "411(59.5\\%)"  "228(60.0\\%)"

```

Numerical stats

For numerical stats, we have also some handy functions to give an idea of the distribution but it do not provide all the typical stats (*e.g.* n,std)

```
1 summary(ADSL$AGE)
```

```
2      Min . 1st Qu.      Median      Mean 3 rd Qu.      Max.  
3    19.00    52.00    61.00    59.99    68.00    88.00
```

So we are going to create an small function too for displaying data in the usual way

Numerical stats in macro

```

1 SumByCol<-function(treat,Num, dp){
2   treat =as.factor(treat)
3   Num =as.numeric(Num)
4   ntreatcol=length(levels(treat))
5   nob =by(!is.na( Num ), treat ,sum) [1: ntreatcol ]
6   mean=format(round(by(Num,treat,mean)[1:ntreatcol],dp + 1), nsmall = dp + 1)
7   std =format(round(by(Num,treat,sd)[1:ntreatcol],dp + 2), nsmall = dp + 2)
8   Mean_std=paste(mean,"(", std ,")", sep="")
9   Median =format(round(by( Num , treat ,median) [1: ntreatcol ], dp + 1), nsmall = dp + 1)
10  Q1 =format(round(by(Num,treat,FUN =function(x)quantile(x)[2])[1:ntreatcol],dp + 1),
11             nsmall = dp + 1)
12  Q3 =format(round(by(Num,treat,FUN =function(x)quantile(x)[4])[1:ntreatcol],dp + 1),
13             nsmall = dp + 1)
14  Quantiles =paste( Q1 , Q3, sep =",")
15  Min =format(round(by(Num,treat,min)[1:ntreatcol],dp), nsmall = dp)
16  Max =format(round(by(Num,treat,max)[1:ntreatcol],dp), nsmall = dp)
17  Min_Max=paste( Min , Max , sep =",") data_t=t(data.frame( nob , Mean_std , Median , Quantiles , Min_Max ))
18  row.names(data_t) =c("\\quad n","\\quad Mean (SD)",
19                      "\\quad Median",
20                      "\\quad Q1, Q3",
21                      "\\quad Min, Max")
22  return(data_t)
23 }
24 Num = SumByCol(Num = ADSL$AGE,treat = ADSL$ARM,dp = 0)
25 Num
26
27      A: IFL      F: FOLFOX      G: IROX
28 \\quad n      "428"      "691"      "380"
29 \\quad Mean (SD) "59.7(11.36)" "60.3(11.63)" "59.8(11.50)"
30 \\quad Median      "61.0"      "61.0"      "61.0"
31 \\quad Q1, Q3      "53.0,68.0" "52.0,69.0" "52.0,68.0"
32 \\quad Min, Max      "27,88"      "19,88"      "26,85"

```

Format everything

```

1 #First line of header concatenate Variable text with ARM
2 Header1 =c("Variable",colnames(Cat))
3 #Second line have the Level + (N=XX)
4 N=format(table( ADSL $ ARM ))
5 BigN =paste("N = ",N,")", sep="")
6 Header 2 =c("\\quad Level", BigN)

7
8 #Generate sex labels
9 level =c("\\quad Female", "\\quad Male")
10 #Append to our data
11 Cat_new=matrix(c(level,Cat),byrow = FALSE,ncol= 4)
12 #Generate label for the variable
13 label_row=c("Gender", "", "", "")
14 #Append all together
15 Cat_new=matrix(c(label_row,t(Cat_new)),byrow = TRUE,ncol= 4)

16
17 #Same for Num
18 level =rownames(Num)
19 Num_new=matrix(c(level,Num),byrow = FALSE,ncol= 4)
20 label_row=c("Age (Years)", "", "", "")
21 Num_new=matrix(c(label_row,t(Num_new)),byrow = TRUE,ncol= 4)

```

xtable package

The *xtable* package convert any matrix or data frame into a table to be introduced to L^AT_EX

We are going to join the headers and the different summaries and create an output with it.

```

1 library(xtable)
2 enter =c("", "", "", "")
3 out =matrix(c(Header1,Header2,t(Cat_new),enter,t(Num_new)),
4                                     byrow = TRUE ,ncol= 4)
5 Out_table= xtable ( out , align =c("l","l","c","c","c"))
6 print(Out_table,include.colnames= FALSE,
7       include.rownames= FALSE,
8       hline =c(0,2,nrow(out)),
9       sanitize.text.function= force)

```

Final output in L^AT_EX

Variable Level	A: IFL (N = 428)	F: FOLFOX (N = 691)	G: IROX (N = 380)
Gender			
Female	151 (35.3%)	280 (40.5%)	152 (40.0%)
Male	277 (64.7%)	411 (59.5%)	228 (60.0%)
Age (Years)			
n	428	691	380
Mean (SD)	59.7 (11.36)	60.3 (11.63)	59.8 (11.50)
Median	61.0	61.0	61.0
Q1, Q3	53.0, 68.0	52.0, 69.0	52.0, 68.0
Min, Max	27, 88	19, 88	26, 85



RTF approach

We can do the same to get a RTF output. To do so we are going to use the last package of the day, *rtf* package.

Rewriting prior macros to not use special latex code we can do the following

```

1 library(rtf)
2 Headers =paste(Header1,Header2,sep = "\\line")
3 enter =c("", "", "", "")
4 out =matrix(c(t(Cat_new),enter,t(Num_new)),
5             byrow = TRUE ,ncol= 4)
6 rtf = RTF ("Sample.doc")
7 colnames( out ) =Headers
8 addTable(rtf,out,col.justify =c("L","C","C","C"),
9         header.col.justify =c("L","C","C","C"))
10 done(rtf)

```

Final output in RTF

Variable Level	A: IFL (N = 428)	F: FOLFOX (N = 691)	G: IROX (N = 380)
Gender			
Female	151 (35.3%)	280 (40.5%)	152 (40.0%)
Male	277 (64.7%)	411 (59.5%)	228 (60.0%)
Age (Years)			
n	428	691	380
Mean (SD)	59.7 (11.36)	60.3 (11.63)	59.8 (11.50)
Median	61.0	61.0	61.0
Q1, Q3	53.0, 68.0	52.0, 69.0	52.0, 68.0
Min, Max	27, 88	19, 88	26, 85



WORLDWIDE
CLINICAL TRIALS

Conclusion

Conclusion

R has a great potential and is able already to perform the needed task on a clinical trial, with more contributions of the community, this can only improve. With this paper we aim to highlight the potential of R, and maybe give them a place into the tool belt and allow them to coexist. For any new trend, rather than contract a new expensive package, it maybe worth to look for the existing packages, you may get surprised.





WORLDWIDE
CLINICAL TRIALS



Thanks for your
attention!

Any question?

Contact details: antonio.rodriguez@worldwide.com