SAS Vs R in Pharma: : **CHEAT**

Introduction

This cheat sheet mainly focus on data manipulation techniques frequently used in pharmaceutical industry. Run the below codes while starting R.

```
install.packages("tidyverse", “lubridate", "flextable","officer")
library(tidyverse,lubridate,flextable,officer)
```

Data Inputs

```
data DM;
infile datalines delimiter=',';
    input subjid $  strata sex $ armcd $
age height weight ;
datalines;
101, 1 , M, B, 43, 150, 75
102, 2 , F, A, 53, 178, 65
103, 2 , F, B, 67, 157, 64
104, 1 , M, A, 34, 168, 72
105, 2 , M, B, 76, 145, 61
;
run;
```

```
subjid      <- c('101','102','103', '104','105')
strata      <- c(1,2,2,1,2)
sex         <- c('M','F','F','M','M')
armcd       <- c('B','A','B','A','B')
age         <- c(43,53,67,34,76)
height      <- c(150,178,157,168,145)
weight      <- c(75,65,64,72,61)
DM          <- data.frame(subjid,strata,sex,
                  armcd,age,height, weight)
View(DM)
```

```
data VS;
infile datalines delimiter=',';
    input subjid $  strata armcd $ visit $
visitnum paramcd  $ aval ;
datalines;
101, 1, B, visit  1,  100, SYSBP, 120
101, 1, B, visit 19, 1900, SYSBP, 128
101, 1, B, visit  1,  100, DIABP,  65
101, 1, B, visit 19, 1900, DIABP,  78
102, 2, A, visit  1,  100, SYSBP, 156
102, 2, A, visit 19, 1900, SYSBP, 127
102, 2, A, visit  1,  100, DIABP,  74
102, 2, A, visit 19, 1900, DIABP,  72
105, 2, B, visit  1,  100, SYSBP, 136
105, 2, B, visit 19, 1900, SYSBP, 125
105, 2, B, visit  1,  100, DIABP,  59
105, 2, B, visit 19, 1900, DIABP,  64
;
run;
```

```
subjid     <- c('101','101','101','101',
                '102','102','102','102',
                 '105','105','105','105')
strata     <-  c(1,1,1,1,2,2,2,2,2,2,2,2)
armcd      <- c('B','B','B','B','A','A',
                 'A','A','B','B','B','B')
visit      <-  c('Visit 1','Visit 19','Visit 1', 'Visit 19',
                'Visit 1','Visit 19','Visit 1', 'Visit 19',
                 'Visit 1','Visit 19','Visit 1', 'Visit 19')
visitnum   <- c(100,1900,100,1900,100,1900,
                100,1900,100,1900,100,1900)
paramcd    <- c('SYSBP','SYSBP','DIABP','DIABP',
                 'SYSBP','SYSBP','DIABP','DIABP',
                'SYSBP','SYSBP','DIABP','DIABP')
aval        <- c(120,128,65,78,156,127,
                  74,72,136,125,59,64)
VS         <- data.frame(subjid,strata,armcd,visit,
               visitnum,paramcd,aval)
View(VS)
```

```
data EX;
infile datalines delimiter=',';
input subjid $ visitnum visit $ exstdtc
$23-49 ;
datalines;
101,    100, visit  1, 2021-12-22T08:25
101,  1900, visit 19, 2021-12-29T08:55
104,    100, visit  1, 2021-12-16T11:02
104,  1900, visit 19, 2022-01-06T13:45
;
run;
```

```
subjid     <- c('101','101','104','104')
Visitnum <- c(100,1900, 100,1900)
visit        <- c('Visit 1','Visit 19', 'Visit 1', 'Visit 19')
exstdtc   <- c("2021-12-22T08:25",
                   "2021-12-29T08:55",
                   "2021-12-16T11:02",
                   "2022-01-06T13:45")
EX    <- data.frame(subjid,visitnum,visit,exstdtc)
View(EX)
```

Variable operation

Variable Sorting

```
proc sort data=VS
             out=ADVS_SRT1;
  by subjid descending paramcd
                        visitnum;
run;
```

```
ADVS_SRT1 <- VS %>%
  arrange(subjid,desc(paramcd),
                       visitnum)
View(ADVS_SRT1)
```

Data Filtering

```
data ADSL_FL1;
  set DM;
  if strata = 2;
run;
```

```
ADSL_FL <- DM %>%
  filter(strata==2 )
View(ADSL_FL)
```

```
data ADSL_FL2;
  set DM;
  if strata = 2 & armcd = 'A';
run;
```

```
ADSL_FL2 <- DM %>%
  filter(strata==2 & armcd=='A')
View(ADSL_FL2)
```

```
data ADSL_FL3;
  set DM;
  if subjid in ('101','102');
run;
```

```
ADSL_FL3 <- ADSL %>%
  filter(subjid %in% c('101', '102'))
View(ADSL_FL3)
```

Data operations (keep, drop, and rename)

```
data ADSL_DO;
  set DM;
  keep subjid armcd ;
  drop age;
  rename subjid=usubjid;
run;
```

```
ADSL_DO <- DM %>%
 select(subjid,age,armcd)%>%
 select(-age) %>%
 rename(usubjid=subjid)
View(ADSL_DO)
```

Variable creation

```
data ADSL_MT1;
  set DM;
  height_m= height/100;
  BMI=weight/(height_m**2);
run;
```

```
ADSL_MT1 <- DM %>%
  mutate(height_m=height/100) %>%
  mutate(BMI=weight/height_m^2)
View(ADSL_MT1)
```

Remove duplicate records

```
proc sort data=VS
    out=ADVS_SRT1 nodupkey;
  by subjid  paramcd;
run;
```

```
ADVS_SRT1 <- VS %>%
  arrange( subjid , paramcd )%>%
  group_by ( subjid, paramcd) %>%
  slice( 1 )
view(ADVS_SRT1)
```

Data Transformation

Data transpose

(long to wide)

```
proc transpose data=VS
                     out=ADVS_TR;
   by subjid strata armcd visit;
   id paramcd;
   var aval;
run;
*Sort the dataset before transpose
```

```
ADVS_TR <-VS %>%
 pivot_wider(
   names_from=paramcd,
   values_from=aval)
view(ADVS_TR)
```

(wide to long)

```
proc transpose data=ADVS_TR
                     out=ADVS_TR2;
   by subjid  strata armcd visit;
   var SYSBP DIABP;
run;
*Sort the dataset before transpose
```

```
ADVS_TR2 <- ADVS_TR %>%
 pivot_longer(
   cols=SYSBP:DIABP,
   names_to='paramcd',
   values_to='aval')
view(ADVS_TR2)
```

Data Appending

```
data ADVS_APD;
    set VS EX;
run;
```

```
ADVS_APD <- bind_rows(VS,EX)
View(ADVS_APD)
```

Data merging

single-dataset

```
proc sql;
    create table ADVS_IJ as
    select distinct a.*, b.exstdtc
    from VS as a
    inner join EX as b
    on a.subjid = b.subjid and
    a.visitnum= b.visitnum;
quit;
```

```
ADVS_IJ <- VS %>%
 inner_join(EX,
          by = c("subjid","visitnum"))
view(ADVS_IJ)
```

Multiple-dataset

```
proc sql;
 create table ADSL_FJ as
      select distinct a.*, b.visitnum,
      b.paramcd,b.aval,c.exstdtc
      from DM as a
      full join VS as b
      on a.subjid = b.subjid
      full join EX as c
     on a.subjid = c.subjid;
 quit;
```

```
ADVS_FJ <- VS %>%
 full_join(EX,
         by = c("subjid","visitnum")) %>%
 full_join(DM, by = "subjid")
view(ADVS_FJ)
```

Views and opinions expressed are those of the author and not necessarily Novo Nordisk * RStudio® is a trademark of RStudio, Inc. • CC BY SA Bharath kumar• bkq@novonordisk..com • Updated: 2022-11

SAS Vs R in Pharma: : **CHEAT**

Character operation

Variable conversion:

Numeric to character:

```
data ADVS_CHAR;
   set VS;
   avalc=put(aval,8.);
run;
 *SAS has formats to handle digits
```

```
ADVS_CHAR <- VS %>%
  mutate(avalc=as.character(aval))
view(ADVS_CHAR)
```

Character to numeric

```
data ADVS_NUM;
   set ADVS_CHAR;
   aval_num=input(avalc, 8.);
run;
  *SAS has various informats
```

```
ADVS_NUM  <- ADVS_CHAR %>%
 mutate(aval_num=as.numeric(avalc))
View(ADVS_NUM)
```

String operations:

```
data ADVS_STR1;
  set VS;
  substring=substr(visit,7,2);
  scanstring=scan(visit,2);
run;
```

```
ADSL_STR1 <- VS %>%
  mutate(substring=
         str_sub(visit,7,9)) %>%
  mutate(scanstring=
         (word(visit,2,sep=' ')))
View(ADSL_STR1)
```

If and else if command

```
data ADSL_IF;
  set DM;
  length age_r $12;
  if age < 18
         then age_r= "<18";
  else if 18 <= age <=64
         then age_r = "18-64";
  else if age > 64
        then age_r = ">65";
run;
```

```
ADSL_IF <- DM %>%
  mutate(age_r=(
   ifelse(age < 18 ,"<18",
   ifelse(age >= 18 & age <=64,
          "18-64",">65"))))
view(ADSL_IF)
```

Remove leading/trailing spaces and Concatenation

```
data ADVS_RB;
 set VS;
 group_t=strip(subjid)||"/"||
        strip(armcd)||"/"||
        strip(strata);
run;
```

```
ADVS_RB <- VS %>%
  mutate(group_t=paste(
   trimws(subjid),"/",
   trimws(armcd),"/",
   trimws(strata)))
view(ADVS_RB)
```

*See Date/time section for handling in-between spaces

Plotting

```
proc sgplot data=DM;
   scatter x=height y=weight;
   xaxis values=
            (140 to 180 by 10);
   yaxis values=
            (50 to 80 by 10);
run;
```

```
ggplot(data=DM,
      aes(x=height, y=weight)) +
  geom_point()+
  lims(x=c(140,180),y=c(50,80)) +
  ggtitle("Height Vs. weight") +
  theme_classic()
```

```
proc sgplot data=ADSL_IF;
    vbar age_r;
run;
```

```
ggplot(data=ADSL_IF,
      aes(x=age_r)) +
  geom_bar()+
  xlab("Age category")
theme_classic()
```

```
proc sgpanel data=VS;
    panelby  paramcd subjid;
    scatter x=visit
         y=aval/group=armcd;
    series  x=visit
        y=aval/group=armcd;
run;
```

```
ggplot(data=VS,
      aes(x=visitnum,
          y=aval,colour=armcd))+
  geom_point()+
  geom_line()+
  facet_wrap(~ paramcd + subjid)+
  scale_x_continuous(
   labels=c("Visit1","Visit19"),
   breaks=c(100, 1900))
```

Data Summary

Summary

```
proc summary data=
            ADVS_SRT;
    by paramcd visitnum visit;
    var aval;
    output out=summary;
run;
*Sort the data with “by”
variables before summarise
```

```
ADVS_SM <-VS %>%
 group_by(paramcd,armcd,visit)%>%
 summarise(mean=mean(aval),
         sd= sd(aval),
         min=min(aval),
         max=max(aval),
         n=length(aval))
View(ADVS_SM)
```

frequency

```
proc freq data=DM;
    table armcd*strata
        / out=ADSL_FREQ;
run;
```

```
ADSL_FQ <- DM %>%
 count(armcd,strata)
view(ADSL_FQ)
ADSL_FREQ <- ADSL_FQ %>%
 mutate(percent=n/(sum(n)))
View(ADSL_FREQ)
```

Date/time operations

```
data ADEX_DTM;
   set EX;
   format ADTM datetime18.  ADT date9.
          ATM Time5.;
   ADTM = input(exstdtc, e8601DT.);
   ADT  = datepart(ADTM);
   ATM  = timepart(ADTM);
   visit=compress(visit);
run;
```

```
ADEX_DTM <- EX %>%
   mutate(ADTM=ymd_hm(exstdtc)) %>%
   mutate(ADT=date(ADTM)) %>%
   mutate(hours=hour(ADTM)) %>%
   mutate(mins=minute(ADTM)) %>%
   mutate(ATM=paste(hours,":",mins))
View(ADEX_DTM)
```

```
proc transpose data=ADEX_DTM
            out=ADEX_DTM1;
   by subjid;
   id visit;
   var ADT;
run;
```

```
ADEX_DTM1 <- ADEX_DTM %>%
   mutate(visit_=str_replace_all(
          visit," ", "")) %>%
   select(subjid,visit_,ADT) %>%
   spread(visit_,ADT)
```

```
data ADEX_DUR;
   set ADEX_DTM1;
   ADUR=visit19 - visit1;
run;
```

```
ADEX_DTM2 <- ADEX_DTM1 %>%
   mutate(diff=difftime(
         Visit19,Visit1,unit='days'))%>%
   mutate(adur=as.numeric(word(diff,1))) %>%
   select(subjid,Visit1,Visit19,adur)
```

Reporting

```
proc report data=ADVS_RB headline
split='#' spacing=0;
   columns (group_t  paramcd visit aval);
   define group_t/ 'subjid/Armcd/strata'
                     order=data ;
   define paramcd/order;
   define visit/order;
   define aval/order;
run;
```

```
report<- ADVS_RB %>%
 select(group_t,paramcd,visit, aval)%>%
 rename("Usubjid/Armcd/Strata"=group_t,
     "Parameter"=paramcd,
     "Visit"=visit,
     "Value"=aval)%>%
 regulartable()%>%
 autofit()
report <- merge_v(report)
report <-valign(report,valign="top")
```

Data import and export

```
proc import datafile ="ADSL.csv"
      out = ADSL dbms= csv;
run;
```

```
read.csv(ADSL , "ADSL.csv")
```

```
proc export data = ADSL

outfile = "ADSL.csv" dbms = csv replace;

run;
```

```
write.csv(ADSL ,"ADSL.csv”)
```

Views and opinions expressed are those of the author and not necessarily Novo Nordisk * RStudio® is a trademark of RStudio, Inc. • CC BY SA Bharath kumar• bkq@novonordisk..com • Updated: 2022-11