**Supplemental Material 2:** Coding scripts, data analysis, plots for manuscript, and additional plots not used in the final manuscript



# missing_waypt

*Gretchen Galliano,MD*

*December 19, 2018*

```
#clear the environment remove the current dataframes
 rm(list=ls())
#check for the presence of R packages needed for the script
packages <- c("readr", "plyr", "lubridate", "dplyr", "knitr", "openxlsx", "ggplot2","ggrepel","g
ridExtra","naniar")

#install the missing packages needed for this script. Delete if install packages not wanted

if ( length(missing_pkgs <- setdiff(packages, rownames(installed.packages()))) > 0) {
  message("Installing missing package(s): ", paste(missing_pkgs, collapse = ", "))
  install.packages(missing_pkgs)
}
```

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] compiler_3.5.1  magrittr_1.5    tools_3.5.1     htmltools_0.3.6
##  [5] yaml_2.2.0      Rcpp_1.0.0      stringi_1.2.4   rmarkdown_1.11
##  [9] knitr_1.21      stringr_1.3.1   xfun_0.4        digest_0.6.18
## [13] evaluate_0.12
```

```
#read the csv files in the working directory into the environment
#files are in the same directory
filenames <- gsub("\\.csv$","", list.files(pattern="\\.csv$"))
```

```
#delete the last row of onedat abd twodat due to AP LIS formatting of the datatable
#extraline that give date of the report
onedat <- onedat[1:(nrow(onedat)-1),]
twodat <- twodat[1:(nrow(twodat)-1),]
#assign the column names to one dat and two dat to be identical
colnames(onedat) <- c("ID", "code", "collect","created","grswrk","histop","dist","signoff", "siz
e")
colnames(twodat) <- c("ID", "code", "collect","created","grswrk","histop","dist","signoff", "siz
e")

#join the two dataframes together
waypt <- full_join(onedat, twodat)
```

```
## Joining, by = c("ID", "code", "collect", "created", "grswrk", "histop", "dist", "signoff", "s
ize")
```

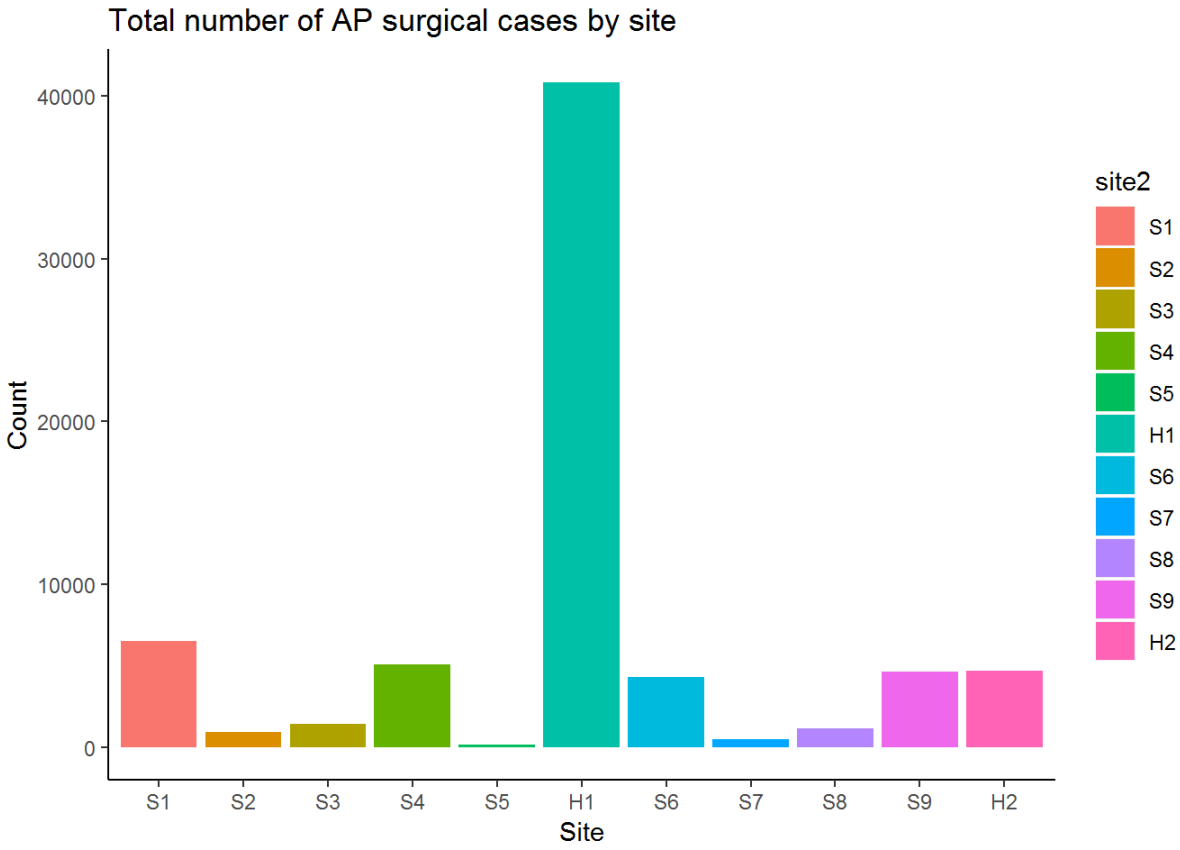
```
#remove the onedat and twodat dataframes from the environment not needed
rm(onedat)
rm(twodat)
```

```
#convert the date and time vectors for each column to posixct objects



waypt$collect <- mdy_hms(waypt$collect, tz ="America/Chicago")

waypt$created <- mdy_hms(waypt$created, tz ="America/Chicago")

waypt$grswrk <- mdy_hms(waypt$grswrk, tz ="America/Chicago")

waypt$histop <- mdy_hms(waypt$histop, tz ="America/Chicago")

waypt$dist <- mdy_hms(waypt$dist, tz ="America/Chicago")

waypt$signoff <- mdy_hms(waypt$signoff, tz ="America/Chicago")

#next code chunk hidden to deidentify hospital sites

# first created a hospital site factor variable based on the first two letters in the accession

#waypt <- mutate(waypt, site = as.factor(substr(ID, 1,2)))


#then create new deidentified factor variable for publication plots example is below with false
 site names
#on left

#waypt <- waypt %>% mutate(site2 = recode(site, "XS"= "S1",
#                                                "YS"= "S2",
#                                               "ZS" = "S3",
#                                                "AS" = "S4",
#                                               "FS"= "S5",
#                                               "ES" = "H1",
#                                               "JS" = "S6",
#                                               "OS" = "S7",
#                                               "IS" = "S8",
#                                               "TS" = "S9",
#                                               "RS"= "H2"))
```

```
#filtered deidentified database for base summary function
summary(pub)
```

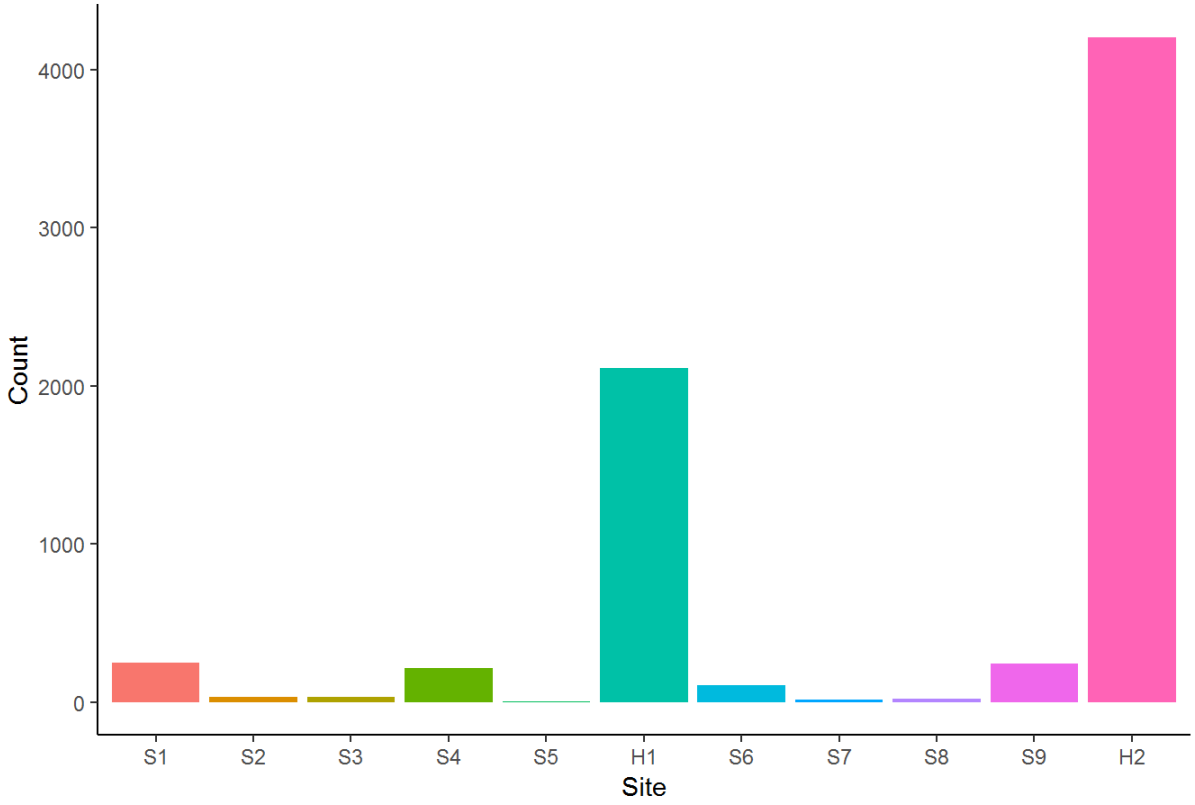
```
##      site2           code              collect
##  H1     :40874   Length:70085       Min.   :2018-01-01 01:01:00
##  S1     : 6524   Class :character   1st Qu.:2018-04-03 09:16:00
##  S4     : 5061   Mode  :character   Median :2018-06-27 11:28:00
##  H2     : 4688                      Mean   :2018-06-27 13:11:40
##  S9     : 4635                      3rd Qu.:2018-09-21 13:51:00
##  S6     : 4319                      Max.   :2018-12-15 22:23:00
##  (Other): 3984
##     created                         grswrk
##  Min.   :2018-01-02 07:44:30   Min.   :2018-01-02 08:45:04
##  1st Qu.:2018-04-03 14:20:06   1st Qu.:2018-04-04 14:21:59
##  Median :2018-06-27 16:30:53   Median :2018-06-28 16:55:26
##  Mean   :2018-06-28 02:39:19   Mean   :2018-06-29 03:29:14
##  3rd Qu.:2018-09-24 07:27:20   3rd Qu.:2018-09-24 14:47:00
##  Max.   :2019-01-02 06:39:36   Max.   :2019-01-02 06:40:36
##                                NA's   :1982
##      histop                          dist
##  Min.   :2018-01-02 08:45:04   Min.   :2018-01-03 05:21:00
##  1st Qu.:2018-04-05 21:01:46   1st Qu.:2018-04-06 14:12:03
##  Median :2018-07-01 14:27:08   Median :2018-07-03 01:09:53
##  Mean   :2018-06-30 22:20:36   Mean   :2018-07-01 17:49:13
##  3rd Qu.:2018-09-26 05:39:02   3rd Qu.:2018-09-26 23:24:24
##  Max.   :2019-01-02 06:40:36   Max.   :2019-01-02 20:22:22
##  NA's   :2408                  NA's   :6641
##     signoff
##  Min.   :2018-01-03 08:28:56
##  1st Qu.:2018-04-09 10:38:27
##  Median :2018-07-03 12:24:26
##  Mean   :2018-07-03 03:06:02
##  3rd Qu.:2018-09-27 17:29:44
##  Max.   :2019-01-18 11:09:43
##  NA's   :1
```

```
#create a theme layer to remove grid lines on the plot (optional)

theme_obj <- theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
                   panel.background = element_blank(), axis.line=element_line(colour = "black"))

#plot of total cases in the data set by hospital site ID
waypt %>%
  group_by(site2) %>%
  ggplot(aes(x=site2, fill =site2)) + geom_bar()+
  labs(y="Count", x = "Site")+
  ggtitle("Total number of AP surgical cases by site")+
  theme_obj
```




```
#exploration of missing values also called NA
#filter includes NA values on grossing, histoprep, distribution, or signout

# "OR" statements, captures all cases with missing data

NA_prep <- waypt%>% filter((is.na(grswrk)) | (is.na(histop)) | (is.na(dist) | (is.na(signoff))))

#plot of all missing actions by site NA_prep
NA_prep %>%
  group_by(site2) %>%
  ggplot(aes(x=site2, fill =site2)) + geom_bar()+
  labs(y="Count", x = "Site")+
  theme(legend.position = "none")+
  ggtitle(" Figure 3. Cases missing any process timestamp by hospital site")+
  theme_obj
```

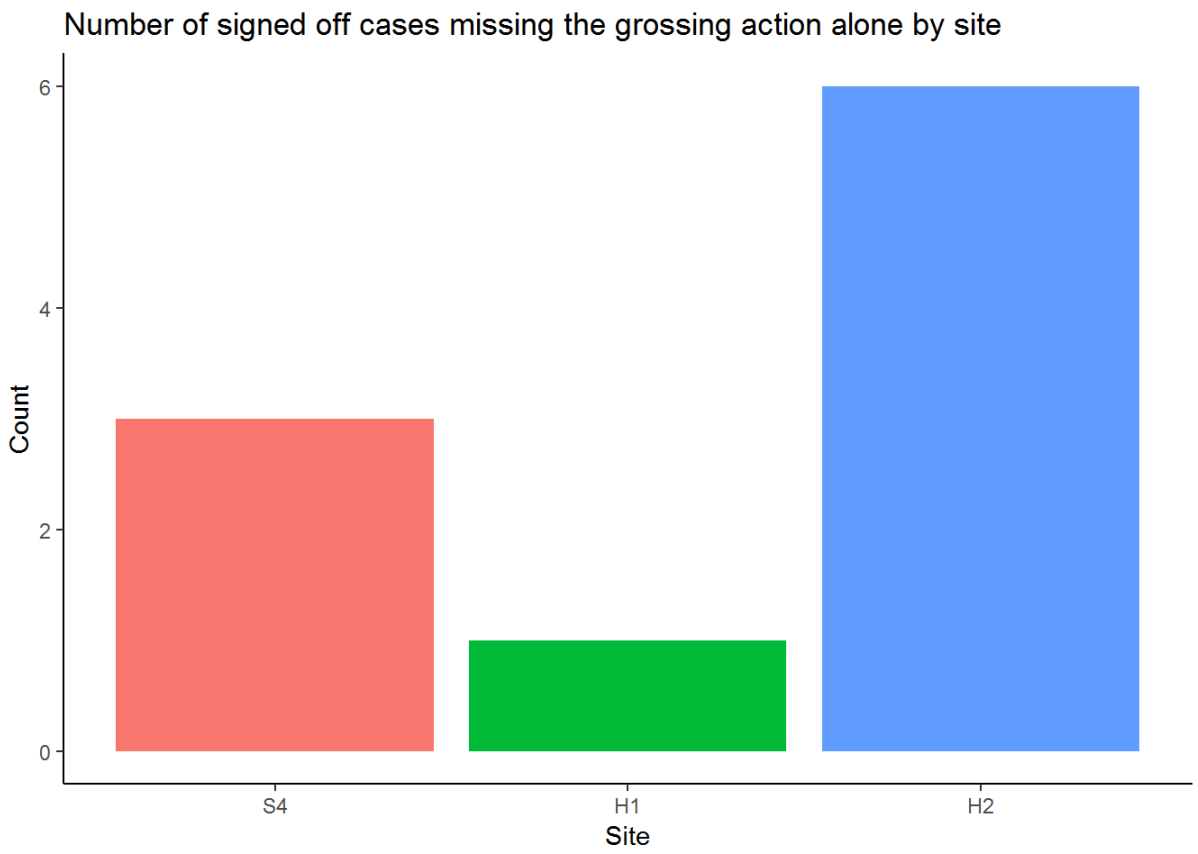
Figure 3. Cases missing any process timestamp by hospital site



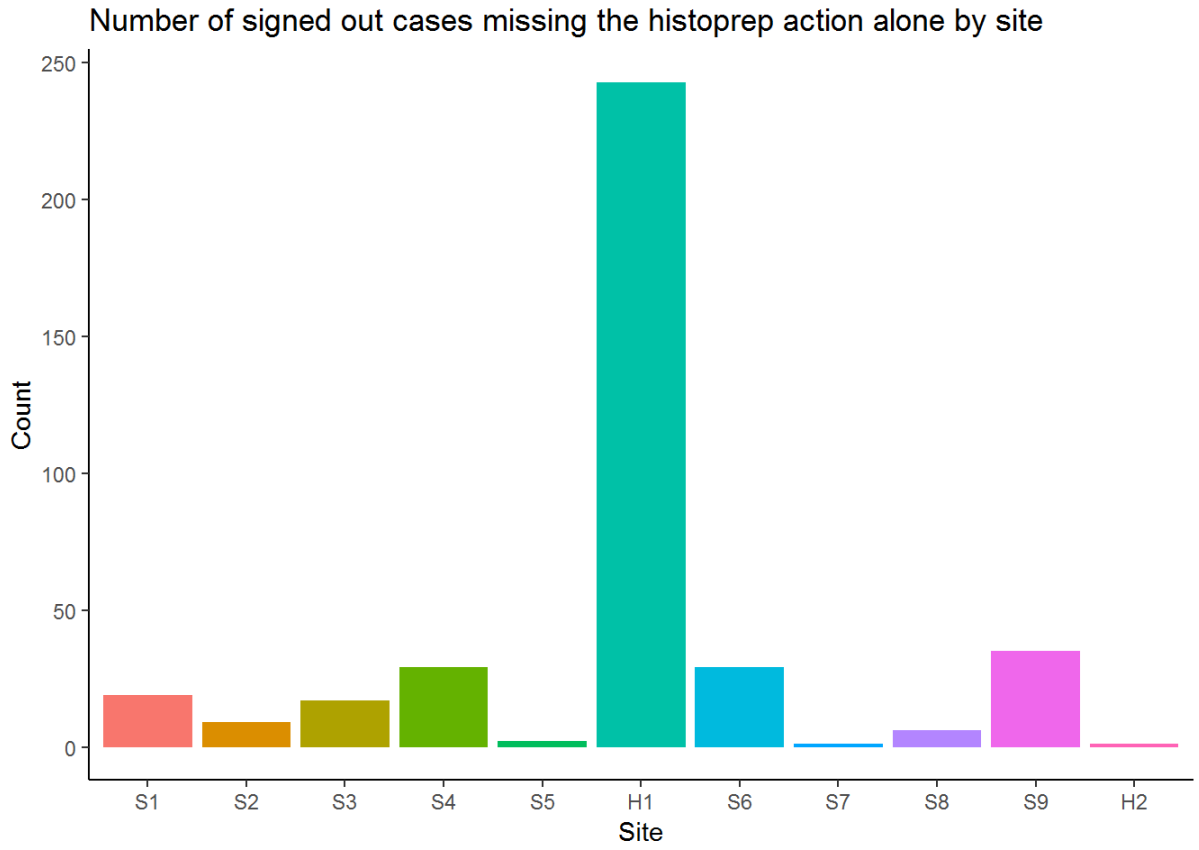

```
#filter for cases missing process steps and still open (not signed off- NSO)
NA_prep_nso <- NA_prep%>% filter(is.na(signoff))


#Filter for signed off cases missing at least one process step (signed off- SO)
NA_prep_so <- NA_prep%>% filter(!is.na(signoff))
```

```
#open Excel file writing package
library(openxlsx)
#Create file for case deep dive of signed off cases missing a process step
write.xlsx(NA_prep_so, "naprepso.xlsx")
```

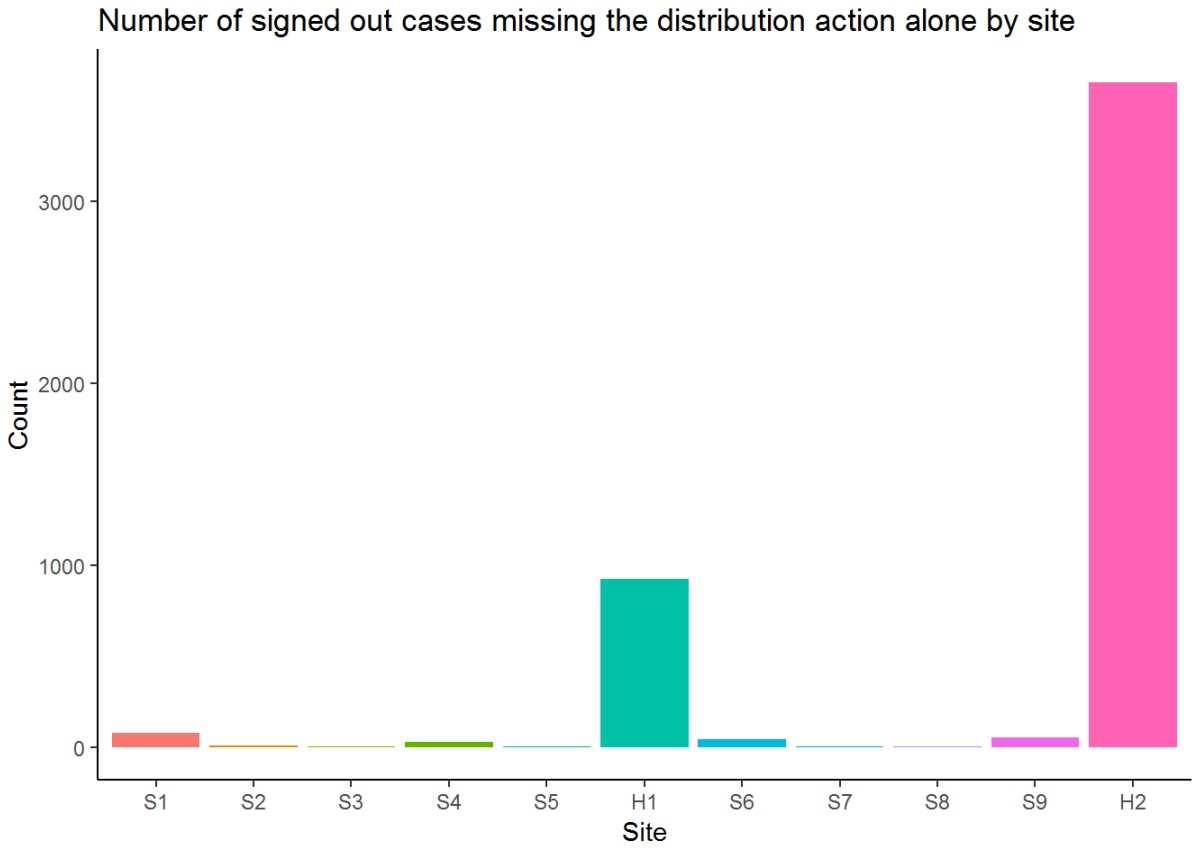
```
#Signed off cases missing a grossing action alone

NA_prep_gr <- NA_prep_so %>% filter((is.na(grswrk) & !(is.na(histop)) & !(is.na(dist))))

NA_prep_gr %>%
  group_by(site2) %>%
  ggplot(aes(x=site2, fill =site2)) + geom_bar()+
  labs(y="Count", x = "Site")+
  ggtitle("Number of signed off cases missing the grossing action alone by site")+
   theme(legend.position = "none")+
  theme_obj
```



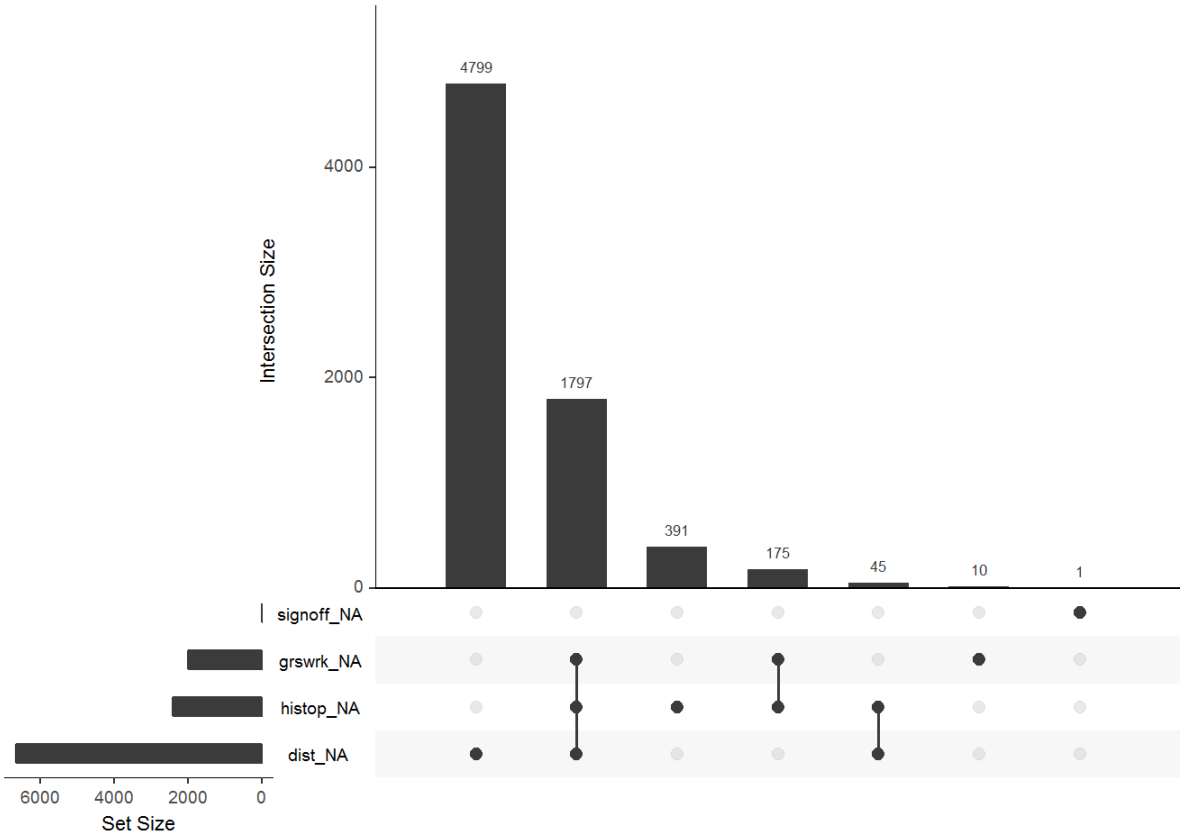

```
#Filter for Signed cases missing histoprep action only
NA_prep_hist <- NA_prep_so %>% filter((!(is.na(grswrk)) & (is.na(histop)) & (!(is.na(dist)))))

NA_prep_hist %>%
  group_by(site2) %>%
  ggplot(aes(x=site2, fill =site2)) + geom_bar()+
  labs(y="Count", x = "Site")+
  ggtitle("Number of signed out cases missing the histoprep action alone by site")+
  theme(legend.position = "none")+
  theme_obj
```




```
#Filter for signed off cases missing the distribution step alone
NA_prep_dist <- NA_prep_so %>% filter((!(is.na(grswrk)) & (!(is.na(histop))) & (is.na(dist))))

NA_prep_dist %>%
  group_by(site2) %>%
  ggplot(aes(x=site2, fill =site2)) + geom_bar()+
  labs(y="Count", x = "Site")+
   ggtitle("Number of signed out cases missing the distribution action alone by site")+
  theme(legend.position = "none")+
  theme_obj
```



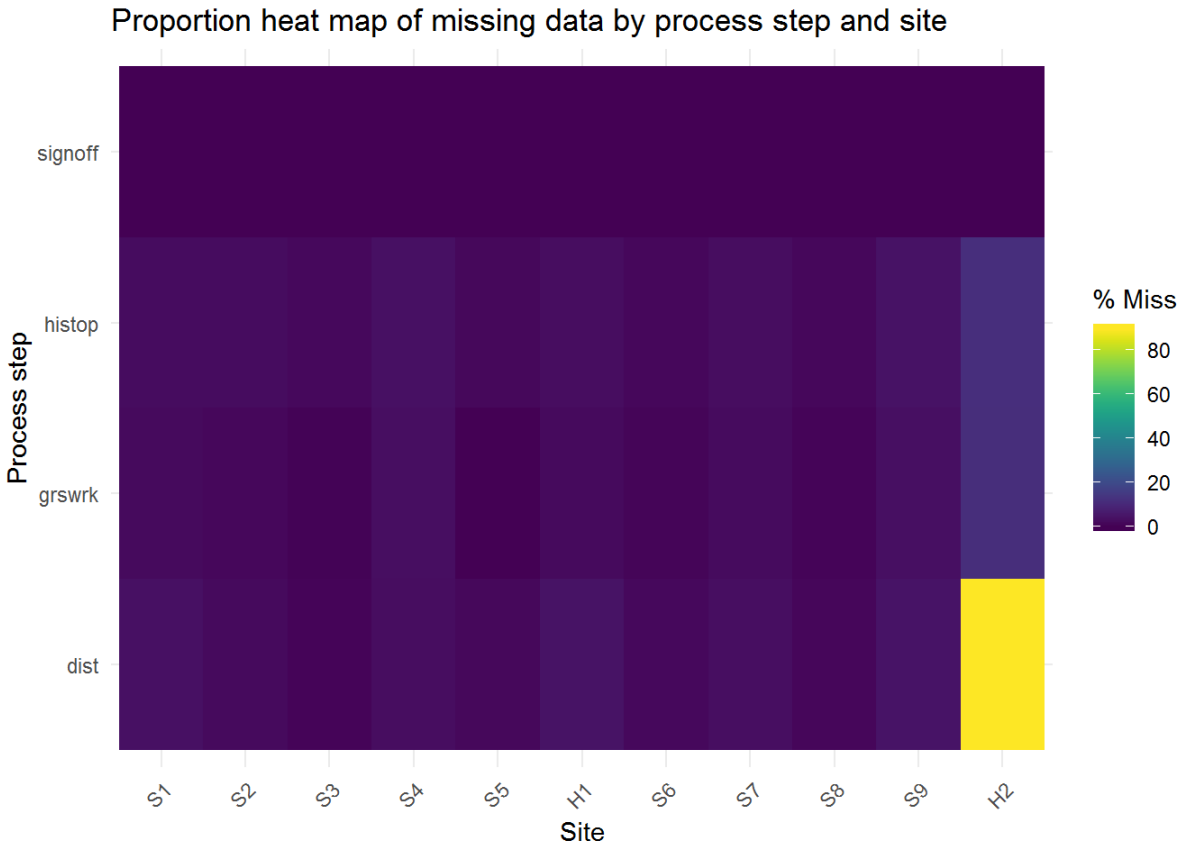

```
#Intersection plot of missing data for system cases by action process step
library(naniar)

gg_miss_upset(waypt%>% select(collect, created, grswrk, histop, dist, signoff),
              nsets = 6,
              nintersects = NA)
```



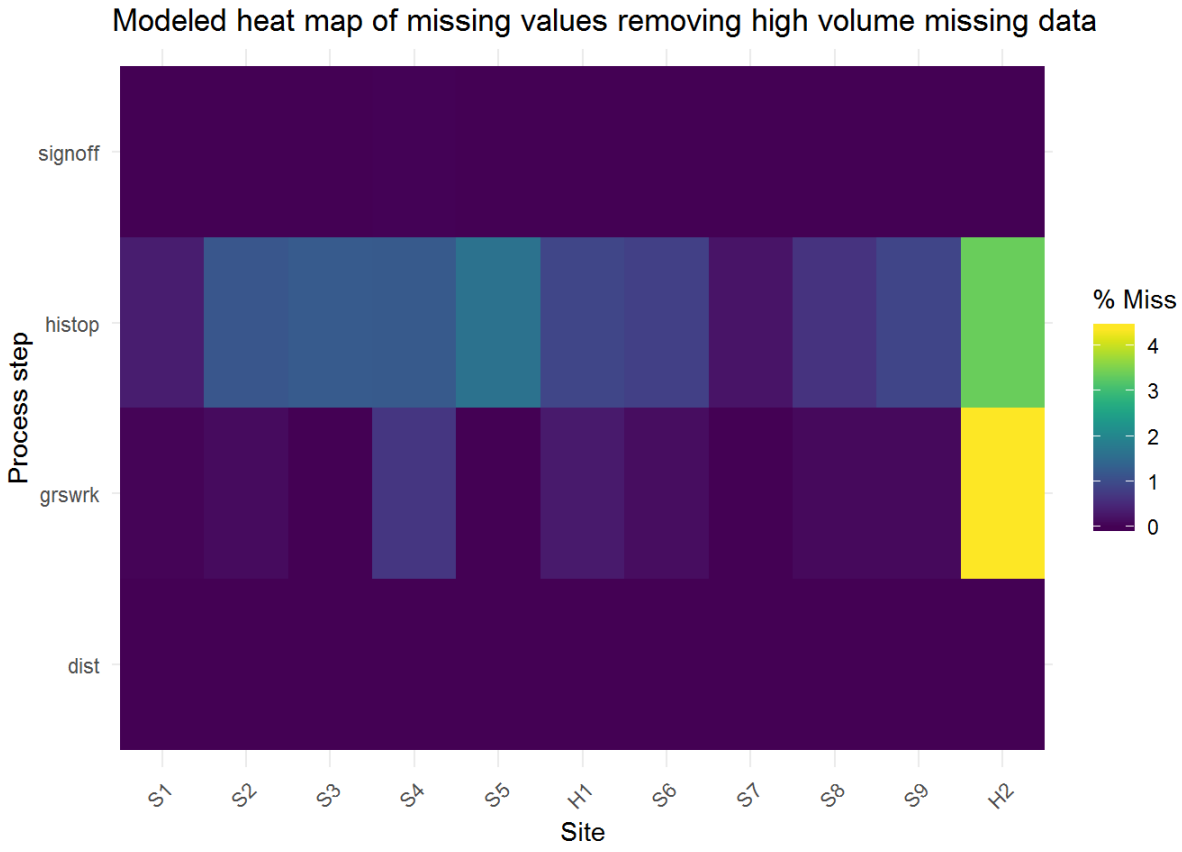

```
#Table of missing data by process step  to identify areas to evaluate further

plotsel <- waypt %>% select(collect, created, grswrk, histop, dist, signoff, site2)

miss_var_summary(plotsel)
```

```
## # A tibble: 7 x 3
##   variable n_miss pct_miss
##   <chr>     <int>    <dbl>
## 1 dist       6641  9.48
## 2 histop     2408  3.44
## 3 grswrk     1982  2.83
## 4 signoff       1  0.00143
## 5 collect       0  0
## 6 created       0  0
## 7 site2         0  0
```

```
#Proportional heat map of missing data
#no need to plot created and collected information because none are missing in above table
plotsel <- waypt %>% select(grswrk, histop, dist, signoff, site2)
gg_miss_fct(x = plotsel,fct = site2) + labs(title = "Proportion heat map of missing data by proc
ess step and site")+
  labs(y="Process step", x = "Site")
```




```
#model the proportion map to visualize difference if data completed
#such as all distriubtion actions get completed (remove missing data)
#and remove gross only
plotmod <- waypt %>% filter  ( (!is.na(dist)) & ((!(grepl('GROSS',code)))) )
plotmod <- plotmod %>% select (grswrk,histop,dist, signoff, site2)

gg_miss_fct(x = plotmod, fct = site2) +  labs(title = "Modeled heat map of missing values removi
ng high volume missing data")+
    labs(y="Process step", x = "Site")
```


Modeled heat map of missing values removing high volume missing data
Process step
signoff
histop
grswrk
dist
S1
S2
S3
S4
S5
H1
S6
S7
S8
S9
H2
Site
% Miss
4
3
2
1
0

```
#create random samples for case level deep dives for large subsets
#library(openxlsx)

#sample of 100 completed cases for look back and write excel file

#na_dist_rand <- NA_prep_dist[sample(nrow(NA_prep_dist), 100), ]
#write.xlsx(na_dist_rand, "distrand.xlsx")

#rmnawaypt <- na.omit(waypt)


#rmnawaypt <- rmnawaypt[sample(nrow(rmnawaypt), 100), ]

#write.xlsx(rmnawaypt, "rmnarand.xlsx")

#random sample of 100 of cases after august after IS upgrade

#rmnawaypt2 <- na.omit(waypt)

#rmnawaypt2 <- rmnawaypt2 %>% filter(rmnawaypt2$co_month > "Aug")

#rmnawaypt2 <- rmnawaypt2[sample(nrow(rmnawaypt2), 100), ]

#write.xlsx(rmnawaypt2, "rand_a_aug.xlsx")


#random sample of 100 of cases  without distriubtion step after august after IS upgrade



#dist2 <- NA_prep_dist %>% filter(NA_prep_dist$co_month > "Aug")

#dist2 <- dist2[sample(nrow(dist2), 100), ]

#write.xlsx(dist2, "dist2.xlsx")

#stats of missing data

#read in table from case deep dive
#combine any process variation and no process variation and compare with chi square
#not reported in paper

dat =("
type    pe  is  lev1    lev2    lev3
proc_var    168 0   0   32  0
      random    135 116 182 165 133
        ")

matriz = as.matrix(read.table(textConnection(dat),
                              header=TRUE,
                              row.names=1))
```

```
chisq.test(matriz,
           correct=TRUE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  matriz
## X-squared = 328.34, df = 4, p-value < 2.2e-16
```

```
#fishers exact
fisher.test(matriz, simulate.p.value=TRUE)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based
##  on 2000 replicates)
##
## data:  matriz
## p-value = 0.0004998
## alternative hypothesis: two.sided
```

```
#chi square of grouped proc variation
#seperated into insignificant variation and significant variation
datmat =("type  no_var  ins_var sign_var
proc_var    135 298 298
       all_rand 168 0   32
")

datmat=as.matrix(read.table(textConnection(datmat),
                            header=TRUE,
                            row.names=1))

chisq.test(datmat,
           correct=TRUE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  datmat
## X-squared = 315.92, df = 2, p-value < 2.2e-16
```

```
fisher.test(datmat)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  datmat
## p-value < 2.2e-16
## alternative hypothesis: two.sided
```

```
#after the IS changes
dat2= ("type    no_var  ins_var sign_var
proc_var    78  180 15
all_rand    98  0   2
")

dat2=as.matrix(read.table(textConnection(dat2),
                          header=TRUE,
                          row.names=1))

fisher.test(dat2)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  dat2
## p-value < 2.2e-16
## alternative hypothesis: two.sided
```