V&D Baseline Missing

```r
# Read in the SMS data using the readxl package
SMS_Data <- read_excel("SMS Data.09.03.24.xlsx", skip = 1)


# Filter out Participation Status = Done and Study Status = All Participation Complete or Compensation
SMS_Data <- SMS_Data %>%
  filter(`Participation Status` == "Done",
         `Study Status` %in% c("All Participation Complete", "Compensation Sent"))

# Count the number of unique StudyID
uniqueStudyID <- SMS_Data %>%
  distinct(StudyID)

uniqueStudyID_count <- nrow(uniqueStudyID)

# Print the uniqueStudyID count
uniqueStudyID_count
```

```
## [1] 28
```

```r
#check the uniqueStudyIDs
uniqueStudyID
```

```
## # A tibble: 28 x 1
##    StudyID
##      <dbl>
##  1    4826
##  2   12452
##  3   13098
##  4   14195
##  5   18682
##  6   19003
##  7   19599
##  8   19686
##  9   21854
## 10   38178
## # i 18 more rows
```

```r
# Read and clean the V&S Baseline CSV file
V_S_Baseline <- read_csv("V&S Baseline Survey_September 9, 2024_10.27.csv")
```

```
## Rows: 71 Columns: 631
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (631): startDate, endDate, status, ipAddress, progress, duration, finish...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
V_S_Baseline <- V_S_Baseline %>%
  select(StudyID, everything())


V_S_Baseline <- V_S_Baseline %>%
  # Remove rows with NA StudyID and convert StudyID to numeric and integer
  filter(!is.na(StudyID) & !is.na(as.numeric(StudyID))) %>%
  mutate(StudyID = as.integer(as.numeric(StudyID))) %>%
  # Filter rows where StudyID matches uniqueStudyID and finished is 'True'
  filter(StudyID %in% uniqueStudyID$StudyID, finished == "True")
```

```
## Warning: There was 1 warning in ‘filter()‘.
## i In argument: ‘!is.na(StudyID) & !is.na(as.numeric(StudyID))‘.
## Caused by warning:
## ! NAs introduced by coercion
```

```
# Print the number of unique StudyIDs after filtering
n_distinct(V_S_Baseline$StudyID)
```

```
## [1] 27
```

```
V_S_Baseline$StudyID
```

```
##  [1] 19003 18682 43465 73026 89171 59503 40761 49314 59154 85321 38178 38190
## [13] 12452 68129 54010 69258 80549 19599 14195 39925 19686 13098 45107 97466
## [25] 86313 21854 41740
```

```
# Check for StudyID values in uniqueStudyID that are not in V_S_Baseline
missing_ids <- setdiff(uniqueStudyID$StudyID, V_S_Baseline$StudyID)

# Display the missing StudyIDs
missing_ids
```

```
## [1] 4826
```

```
# Check for StudyID values in V_S_Baseline that are not in uniqueStudyID
different_ids <- setdiff(V_S_Baseline$StudyID, uniqueStudyID$StudyID)

# Display the different StudyIDs
different_ids
```

```
## integer(0)
```

There is one different id from test survey which is 4826 And different_ids shows integer(0), it means that there are no StudyID values in V_S_Baseline that are not present in uniqueStudyID. In other words, all the StudyID values in V_S_Baseline are also found in uniqueStudyID.

```
# Drop unnecessary columns
V_S_Baseline <- V_S_Baseline %>%
  select(-startDate, -endDate, -status, -ipAddress, -recipientLastName, -recipientFirstName,
         -recipientEmail, -externalDataReference, -locationLatitude, -locationLongitude,
         -distributionChannel, -userLanguage)
```

Here for the missing bc of skip logic , code to -98.

```
# Convert QID120 column to character
V_S_Baseline$QID120 <- as.character(V_S_Baseline$QID120)

# Replace NA with "-98" (as character) in the entire dataframe
V_S_Baseline[is.na(V_S_Baseline)] <- "-98"

# Check the first few rows to confirm
head(V_S_Baseline)
```

```
## # A tibble: 6 x 619
##   StudyID progress duration finished recordedDate  ‘_recordId‘ QID10 QID11 QID12
##     <int> <chr>    <chr>    <chr>    <chr>         <chr>       <chr> <chr> <chr>
## 1   19003 100      49570    True     2024-04-11 1~ R_6Jwt902U~ 11 -~ more~ 5 - ~
## 2   18682 100      2624     True     2024-04-25 1~ R_2ziruOJV~ 6 - ~ more~ 5 - ~
## 3   43465 100      3621     True     2024-05-04 0~ R_3Tq67LLf~ 7 - ~ 30 -~ less~
## 4   73026 100      2374     True     2024-06-23 1~ R_57HW3bGU~ 1 - ~ 30 -~ 5 - ~
## 5   89171 100      12037    True     2024-06-24 0~ R_6kbRsmce~ 0 - ~ 30 -~ less~
## 6   59503 100      157041   True     2024-06-27 0~ R_6NE93TsU~ More~ more~ 50 -~
## # i 610 more variables: QID25 <chr>, QID26 <chr>, QID27 <chr>, QID28 <chr>,
## #   QID92 <chr>, QID120 <chr>, QID13_TEXT <chr>, QID14 <chr>, QID21 <chr>,
## #   QID121_7 <chr>, QID127_7 <chr>, QID127_12 <chr>, QID128_7 <chr>,
## #   QID128_13 <chr>, QID128_14 <chr>, QID148_7 <chr>, QID148_14 <chr>,
## #   QID148_15 <chr>, QID148_16 <chr>, QID149_7 <chr>, QID149_15 <chr>,
## #   QID149_16 <chr>, QID149_17 <chr>, QID149_18 <chr>, QID150_7 <chr>,
## #   QID150_16 <chr>, QID150_17 <chr>, QID150_18 <chr>, QID150_19 <chr>, ...
```

don’t know options recode to -97

```
# Define the values to replace
replace_values <- c("Don’t know")


# Replace specified values with -97 in columns QID258 and QID259
for (col in c('QID230', 'QID238', 'QID98', 'QID240', 'QID99', 'QID233',"QID258", "QID259")) {
  V_S_Baseline[[col]][V_S_Baseline[[col]] %in% replace_values] <- -97
}
```

```
# Define the values to replace
replace_values <- c("Don't know")


# Replace specified values with -97 in columns QID258 and QID259
for (col in c("QID258", "QID259")) {
  V_S_Baseline[[col]][V_S_Baseline[[col]] %in% replace_values] <- -97
}
```

```
# Replace values to missing codes: -99 (completely missing), -98 (skipped due to logic), -97 (don't kno
missing_codes <- c("-99", "-98", "-97")

# Create a function to summarize missing codes by StudyID
missing_summary <- V_S_Baseline %>%
  select(StudyID, starts_with("QID")) %>%
  gather(key = "Question", value = "Response", -StudyID) %>%
  filter(Response %in% missing_codes) %>%
  group_by(StudyID, Response) %>%
  summarise(Count = n()) %>%
  spread(key = Response, value = Count, fill = 0) %>%
  rename(`-99` = `-99`, `-98` = `-98`, `-97` = `-97`)
```

```
## ‘summarise()‘ has grouped output by ’StudyID’. You can override using the
## ‘.groups‘ argument.
```

```
# Display the summary table
kable(missing_summary)
```

StudyID	-97	-98	-99
12452	0	128	3
13098	1	179	3
14195	0	164	2
18682	1	164	0
19003	0	140	2
19599	2	143	2
19686	0	145	0
21854	0	167	0
38178	0	141	1
38190	1	197	15
39925	0	160	9
40761	1	156	5
41740	0	154	6
43465	0	157	2
45107	1	172	1
49314	1	191	12
54010	1	152	0
59154	0	165	2
59503	0	180	7
68129	1	181	6
69258	0	155	2
73026	0	164	6
80549	0	139	2
85321	0	175	9
86313	0	152	7
89171	0	171	2
97466	0	166	12