Frequency Analysis of Notes From Portland Harbor Model Workshops

Curtis C. Bohlen

2023-01-18

Contents

Introduction

CBEP recently received a grant from NSF's CIVIC Innovation Challenge to work on developing hydrodynamic models that address community needs in Portland Harbor. As part of the project, CBEP hosted three community workshops in November of 2022.

Facilitators produced both "live" notes during the meeting – visible to all on a screen at the front of the meeting room – and detailed meeting transcripts. CBEP staff then reviewed those notes paragraph by paragraph, and coded each paragraph in terms of six characteristics:

- Potential users and uses of hydrodynamic models,
- Data or information needs identified by community members,
- Implied extensions of the initial Casco Bay Model required to fully address those data needs, and
- Ideas for improving communications of model results (e.g., communications channels and user interface design),
- Specifications for model performance or data criteria such as resolution, geographic coverage or ability to conduct simulations.
- Suggestions about monitoring or data collection that could improve information availability.

If a paragraph or live note included something relevant to one or more of these categories, we summarized the related idea, and then assigned each paragraph or comment to categories. In this way we can look at what ideas were expressed most commonly during the workshops.

Of course, not all paragraphs include information related to each of the six types of information, so there is not a perfect one-to-one correspondence between categories.

In this R Notebook, I explore these data in terms of frequency with which certain ideas came up, and cross-correlations among ideas.

Load Packages

```
library(tidyverse)
#> -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
#> v ggplot2 3.4.0      v purrr   1.0.1
#> v tibble  3.1.8      v dplyr   1.0.10
#> v tidyr   1.2.1      v stringr 1.5.0
#> v readr   2.1.3      v forcats 0.5.2
#> -- Conflicts ------------------------------------------ tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
library(ggmosaic)
library(readxl)
library(networkD3)

theme_set(theme_classic())
```

Create Figures Folder

```
dir.create(file.path(getwd(), 'figures'), showWarnings = FALSE)
```

Load Data

```
the_data <- read_excel("Data_Export_Query.xlsx" ) %>%
  mutate(ID = as.integer(ID)) %>%
  rename_with(function(x) sub(" Category_Category", '_Category', x)) %>%
  rename_with(function(x) sub(" ", '_', x))
head(the_data)
#> # A tibble: 6 x 16
#>      ID Category   Day   Comment User_~1 Inter~2 Inter~3 Data_~4 Data_~5 Exten~6
#>   <int> <chr>      <chr> <chr>   <chr>   <chr>   <chr>   <chr>     <dbl> <chr>
#> 1     1 Live Comm~ Day ~ How ca~ Urban ~ Simple  Exampl~ <NA>         NA Draina~
#> 2     2 Live Comm~ Day ~ Use of~ Harbor~ <NA>    <NA>    Waves         4 <NA>
#> 3     2 Live Comm~ Day ~ Use of~ Marine~ <NA>    <NA>    Waves         4 <NA>
#> 4     2 Live Comm~ Day ~ Use of~ Urban ~ <NA>    <NA>    Waves         4 <NA>
#> 5     3 Live Comm~ Day ~ MS4 pr~ Water ~ Inform~ Shared~ <NA>         NA Discha~
#> 6     3 Live Comm~ Day ~ MS4 pr~ Water ~ Inform~ Shared~ <NA>         NA Draina~
#> # ... with 6 more variables: Extension_Timing <dbl>, Monitoring_Category <chr>,
#> #   Monitoring_Data_Group <dbl>, Performance_Category <chr>,
#> #   Performance_Type <dbl>, Performance_Timing <dbl>, and abbreviated variable
#> #   names 1: User_Category, 2: Interface_Category, 3: Interface_Group,
#> #   4: Data_Group, 5: Data_Timing, 6: Extension_Category
```

Our coding was generated in a somewhat sloppy `Access` database, and because of the way SQL works, it is easier to replace numerical values for some groups here, in `R`, rather than before we exported the data from `Access`. I read in the dictionaries here.

```
timing_table <- read_excel("Timing Category.xlsx",
    col_types = c("numeric", "text", "text"))
data_types_table <- read_excel("Data Type.xlsx",
    col_types = c("numeric", "text", "text"))
```

Numerical Values to Strings

And finally I correct the data table to all text entries.

```
the_data <- the_data %>%
  mutate(Data_Timing = timing_table$Timing[match(Data_Timing,
                                                 timing_table$ID)],
         Extension_Timing = timing_table$Timing[match(Extension_Timing,
                                                      timing_table$ID)],
         Performance_Timing = timing_table$Timing[match(Performance_Timing,
                                                        timing_table$ID)]) %>%
  mutate(Monitoring_Data_Group = data_types_table$Group[match(Monitoring_Data_Group,
                                                      data_types_table$ID)],
         Performance_Type = data_types_table$Group[match(Performance_Type,
                                                      data_types_table$ID)])
```

```
the_data %>%
  filter(! is.na(Monitoring_Data_Group)) %>%
  select(contains("Monitoring"))
```

```
#> # A tibble: 35 x 2
#>    Monitoring_Category Monitoring_Data_Group
#>    <chr>               <chr>
#>  1 Data Type           Temperature
#>  2 Data Type           Temperature
#>  3 Validation          Water Quality
#>  4 Validation          Water Quality
#>  5 Validation          Water Quality
#>  6 Validation          Water Quality
#>  7 Validation          Water Quality
#>  8 Validation          Other
#>  9 Validation          Water Quality
#> 10 Validation          Water Level
#> # ... with 25 more rows
```

#A Warning about Uniqueness We have to be careful here, because each note or comment can be represented in this data table multiple times. Each paragraph in the meeting transcript might imply several different users, for example. But if there are multiple users and multiple data types, the records got duplicated (in part) in the SQL query. So for any analysis, we need to test for uniqueness of the data. always

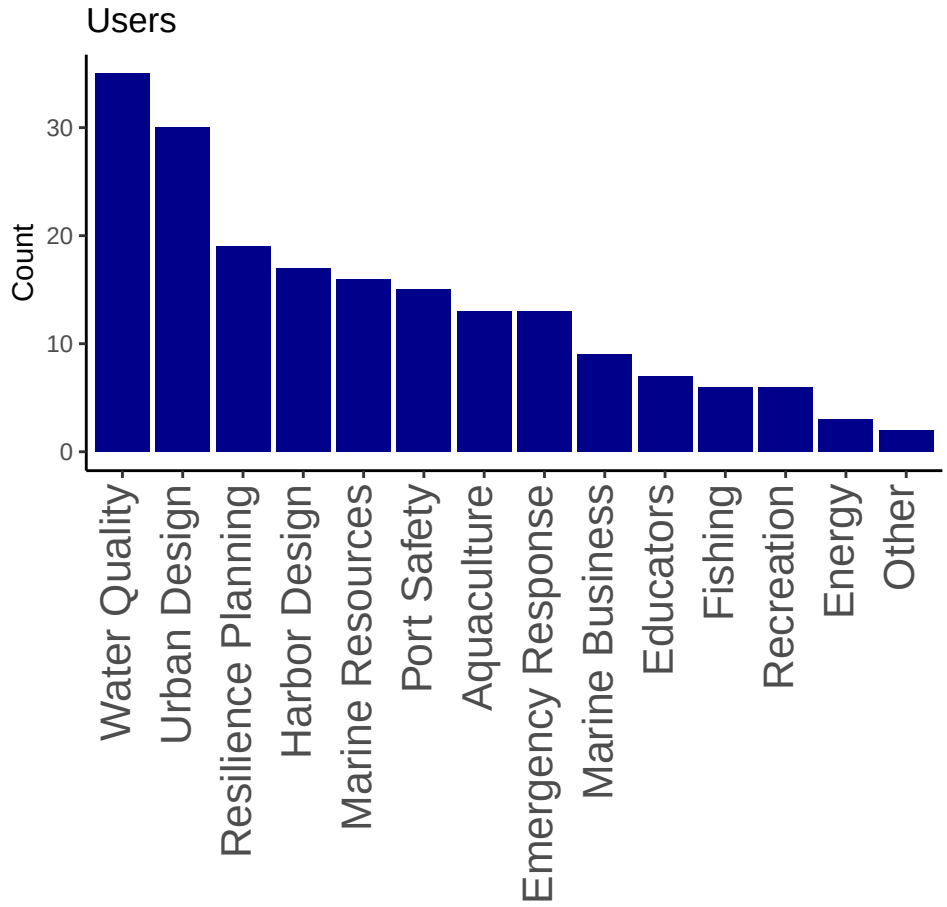
We actually have over 375 records, built out of just over 200 unique comments.

```
cat("All rows in the data:\t\t")
#> All rows in the data:
nrow(the_data)
#> [1] 376

cat("Unique comments reviewed:\t")
#> Unique comments reviewed:
the_data %>%
  select(ID) %>%
  unique() %>%
  nrow()
#> [1] 207
```

Users

```
tmp <- the_data %>%
  select(ID, User_Category) %>%
  unique()
tst <- xtabs(~User_Category, tmp) %>%
  sort(decreasing = TRUE) %>%
  as_tibble()


cat("Number of Unique User Records:\t")
#> Number of Unique User Records:
sum(tst$n)
#> [1] 191
```

```
tst %>%
  mutate(User_Category = fct_reorder(User_Category, n, .desc = TRUE)) %>%

  ggplot(aes(User_Category, n)) +
  geom_col(fill = "blue4") +
  theme(axis.text.x = element_text(angle = 90, size = 16,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle('Users')
```



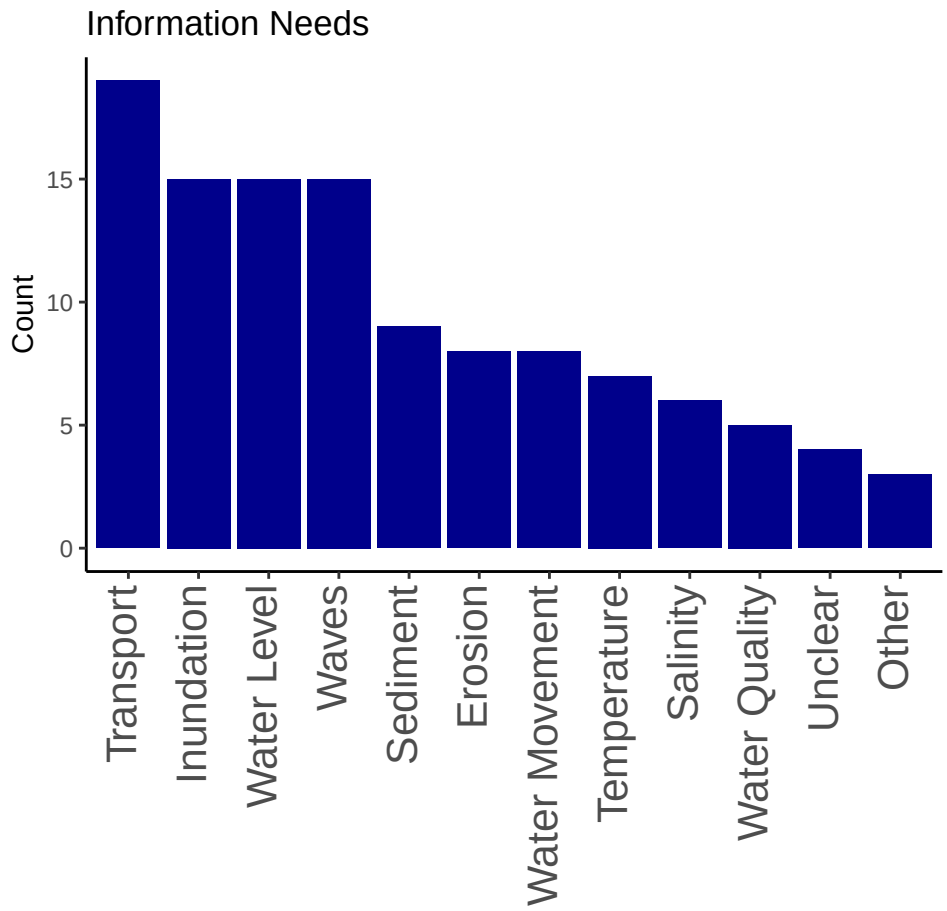

```
ggsave('figures/users.png', type='cairo',
         width = 6, height = 6)
```

Data Types Requested

```
tmp <- the_data %>%
  select(ID, Data_Group) %>%
  unique()
```

```
tst <- xtabs(~Data_Group, tmp) %>%
  sort(decreasing = TRUE) %>%
  as_tibble()

cat("Number of Unique Data Records:\t")
#> Number of Unique Data Records:
sum(tst$n)
#> [1] 114
```

```
tmp %>%
  filter(! is.na(Data_Group))  %>%
  mutate(Data_Group = fct_infreq(Data_Group)) %>%

  ggplot(aes(Data_Group)) +
  geom_bar(fill = "blue4") +
  theme(axis.text.x = element_text(angle = 90, size = 16,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle("Information Needs")
```

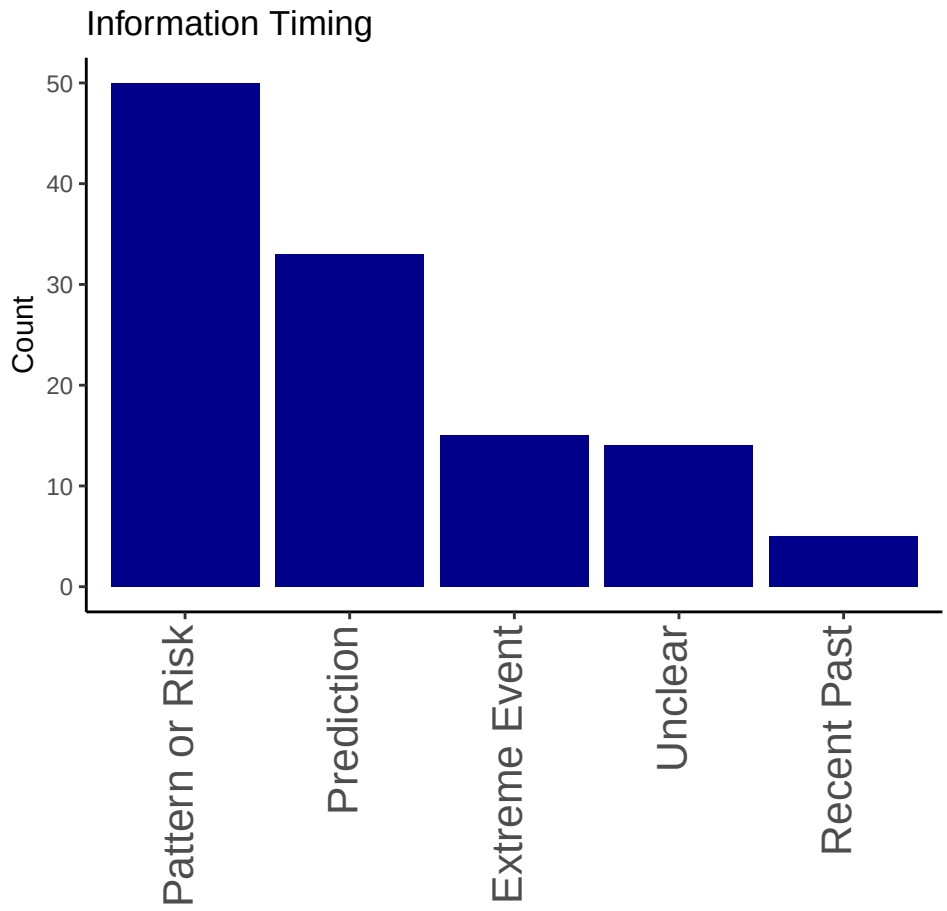


```
ggsave('figures/data.png', type='cairo',
         width = 6, height = 6)
```

Data Timing

```
tmp <- the_data %>%
  select(ID, Data_Timing) %>%
  unique()
tst <- xtabs(~Data_Timing, tmp) %>%
  sort(decreasing = TRUE) %>%
  as_tibble()


cat("Number of Unique Timing Records:\t")
#> Number of Unique Timing Records:
sum(tst$n)
#> [1] 117
```

```
tst %>%
  mutate(Data_Timing = fct_reorder(Data_Timing, n, .desc = TRUE)) %>%

  ggplot(aes(Data_Timing, n)) +
  geom_col(fill = "blue4") +
  theme(axis.text.x = element_text(angle = 90, size = 16,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle('Information Timing')
```



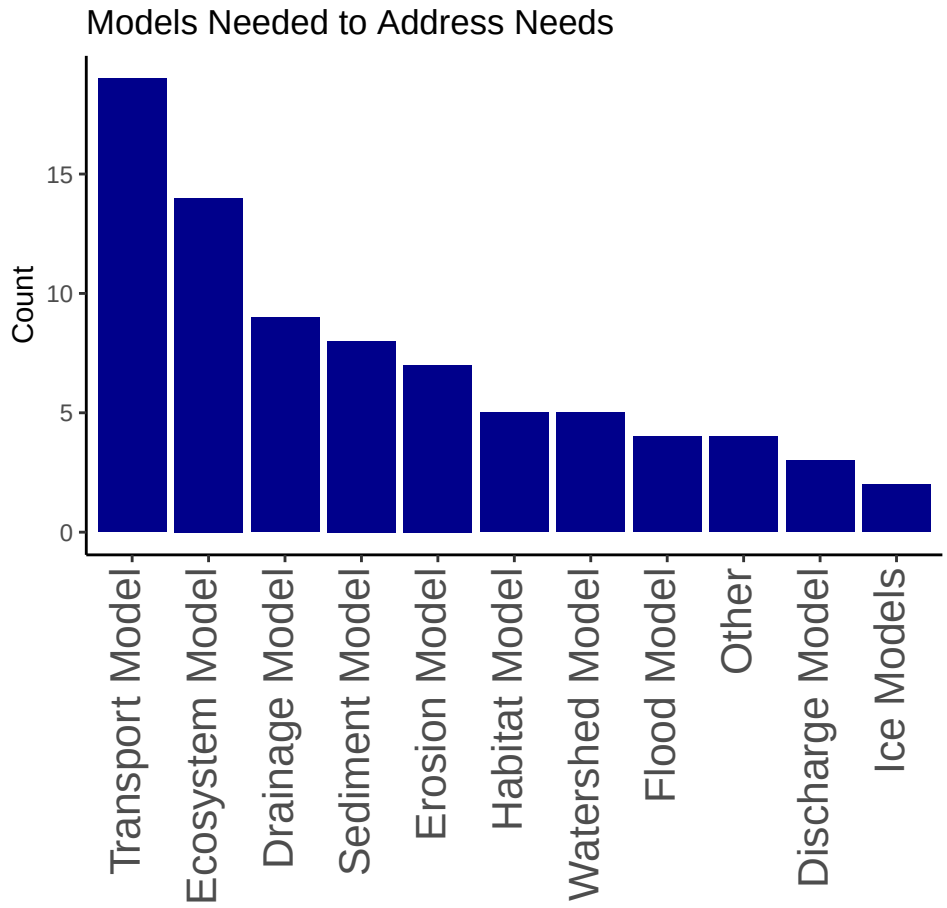

```
ggsave('figures/timing.png', type='cairo',
         width = 6, height = 6)
```

Model Extensions

```
tmp <- the_data %>%
  select(ID, Extension_Category) %>%
  unique()
tst <- xtabs(~Extension_Category, tmp) %>%
  sort(decreasing = TRUE) %>%
  as_tibble()

cat("Number of Unique Extension Records:\t")
#> Number of Unique Extension Records:
sum(tst$n)
#> [1] 80
```

```
tmp %>%
  filter(! is.na(Extension_Category))  %>%
```

```
  mutate(Extension_Category = fct_infreq(Extension_Category)) %>%

  ggplot(aes(Extension_Category)) +
  geom_bar(fill = "blue4") +
  theme(axis.text.x = element_text(angle = 90, size = 16,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle("Models Needed to Address Needs")
```




```
ggsave('figures/models.png', type='cairo',
         width = 6, height = 6)
```

What are the comments associated with the "Other" Category?

```
the_data %>%
  filter(Extension_Category == "Other") %>%
  select(ID, Comment) %>%
  unique()%>%
```

```
  pull(Comment)
#> [1] "Saltwater intrusion into isolated aquifers - impacts on local (peninsular and island) aquifers"
#> [2] "Kelp, shellfish, other aquatic resources - are rapidly expanding in Casco Bay - so re impacted
#> [3] "* Saltwater intrusion into isolated aquifers - impacts on local (peninsular and island) aquifer
#> [4] "* Modeling of saltwater intrusion can benefit from better understanding of nearshore coastal wa
```

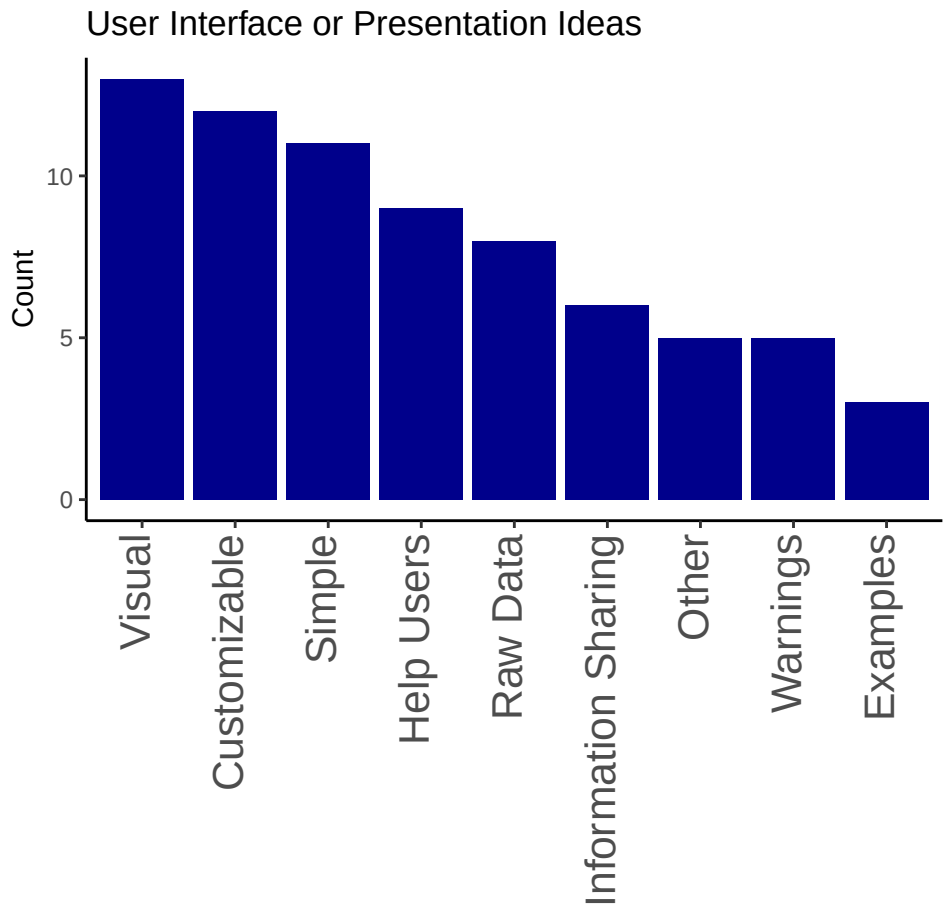
So the model extensions I classified as “Other” include: * Developing decision support tools for aquaculture siting and permitting; and

- Modelling impact of rising seas on groundwater.

User Interface Ideas

```
tmp <- the_data %>%
  select(ID, Interface_Category) %>%
  unique()
tst <- xtabs(~Interface_Category, tmp) %>%
  sort(decreasing = TRUE) %>%
  as_tibble()

cat("Number of Unique Interface Records:\t")
#> Number of Unique Interface Records:
sum(tst$n)
#> [1] 72
```

```
tmp %>%
  filter(! is.na(Interface_Category))  %>%
  mutate(Interface_Category = fct_infreq(Interface_Category)) %>%

  ggplot(aes(Interface_Category)) +
  geom_bar(fill = "blue4") +

  theme(axis.text.x = element_text(angle = 90, size = 16,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle("User Interface or Presentation Ideas")
```




```
ggsave('figures/interfaces.png', type='cairo',
         width = 6, height = 6)
```

What are the comments associated with the "Other" Category?

```
the_data %>%
  filter(Interface_Category == "Other") %>%
  select(Comment) %>%
  unique() %>%
  pull()
#> [1] "What happens to Rt. 1 across marsh area? - need high resolution."
#> [2] "What are decisions people need to make?  Understand those, so can tailor info shared better."
#> [3] "Sedimentation rates - inform decisions about dredging.  Location specific. Also sources of sedi
#> [4] "Where does effluent from wastewater facilities go (old info based on dye studies) - can model p
#> [5] "From chat:  The one meter depth resolution just discussed with regard to wastewater effluent mo
```

Model Performance

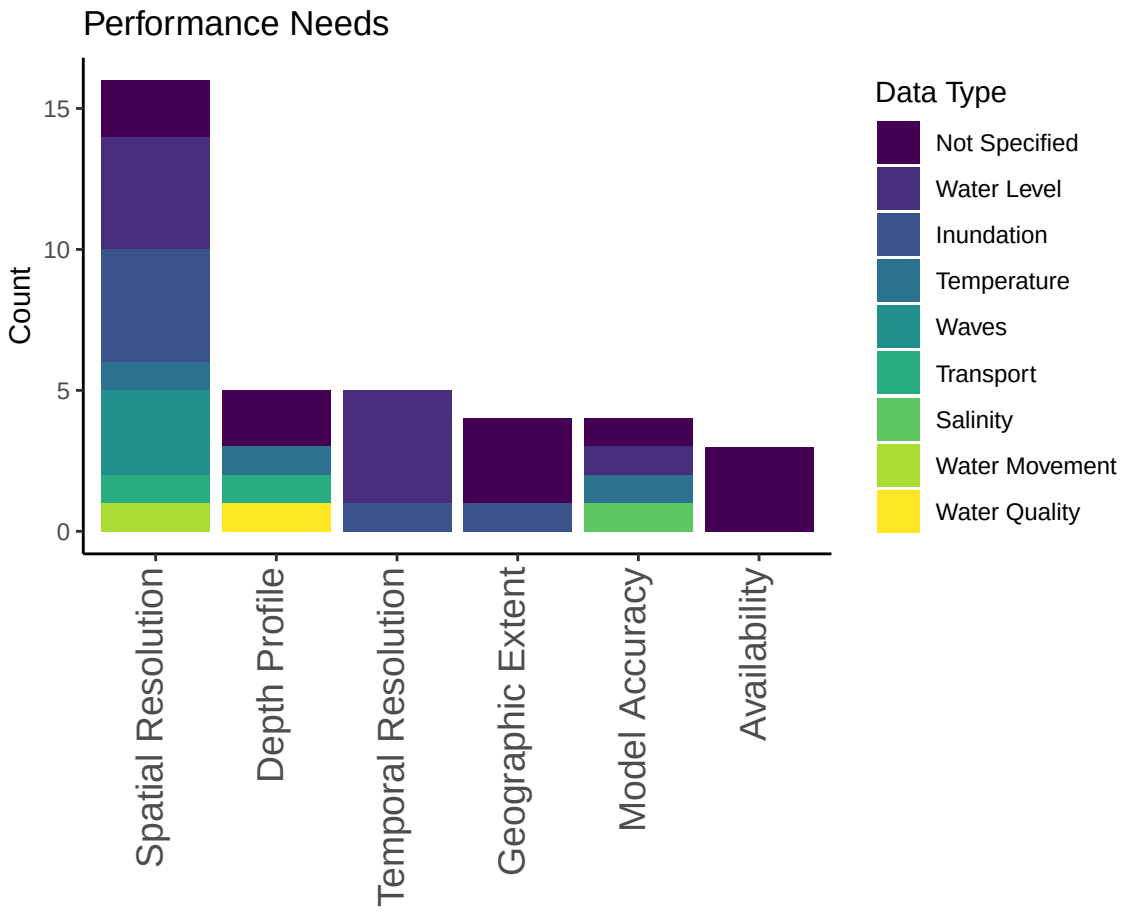
```
tmp <- the_data %>%
  select(ID, Performance_Category, Performance_Type, Performance_Timing) %>%
  unique() %>%
  filter(if_any(starts_with('Data_'), ~!is.na(.)))

xtabs(~  Performance_Type + Performance_Category, data = tmp) %>%
  as_tibble() %>%
  pivot_wider(names_from = Performance_Type, values_from = n) %>%
  mutate(row_tot = rowSums(select(., `Inundation`:`Waves`))) %>%
  arrange(desc(row_tot)) %>%
  knitr::kable()
```

Performance_Category	Inundation	Not Specified	Salinity	Temperature	Transport	Water Level	Water Movement	Water Quality	Waves	row_tot
Spatial Resolution	4	2	0	1	1	4	1	0	3	16
Depth Profile	0	2	0	1	1	0	0	1	0	5
Temporal Resolution	1	0	0	0	0	4	0	0	0	5
Geographic Extent	1	3	0	0	0	0	0	0	0	4
Model Accuracy	0	1	1	1	0	1	0	0	0	4
Availability	0	3	0	0	0	0	0	0	0	3

In initial draft graphics, the "timimg" category is important, but confusing because it also occurs as a separate category.

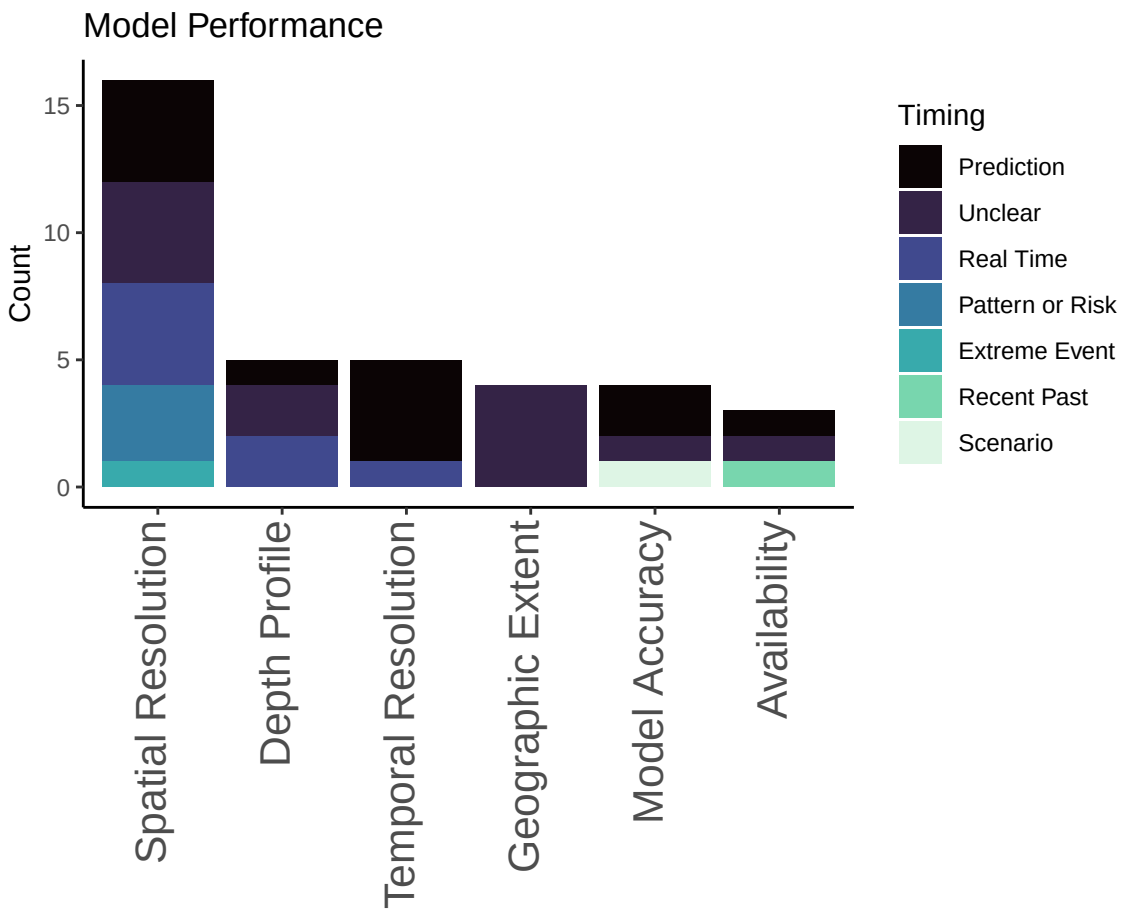
```
tmp <- the_data %>%
  select(ID, Performance_Category, Performance_Type, Performance_Timing) %>%
  unique() %>%
  filter(if_any(starts_with('Data_'), ~!is.na(.))) %>%
  filter(! is.na(Performance_Category),
           Performance_Category != "Timing" )  %>%
  mutate(Performance_Category = fct_infreq(Performance_Category),
         Performance_Type = fct_infreq(Performance_Type),
         Performance_Timing = fct_infreq(Performance_Timing))
```

```
ggplot(tmp, aes(Performance_Category)) +
  geom_bar(aes(fill = Performance_Type)) +
  theme(axis.text.x = element_text(angle = 90, size = 14,
                                   hjust = 1, vjust = 0.25)) +
  scale_fill_viridis_d(name = 'Data Type') +
  ylab('Count') +
  xlab("") +
  ggtitle("Performance Needs")
```




```
ggsave('figures/Performance.png', type='cairo',
         width = 6, height = 5)
```

```
ggplot( tmp, aes(Performance_Category)) +
  geom_bar(aes(fill = Performance_Timing)) +
  theme(axis.text.x = element_text(angle = 90, size = 16,
                                   hjust = 1, vjust = 0.25)) +
  scale_fill_viridis_d(name = 'Timing', option = "G") +
  ylab('Count') +
  xlab("") +
  ggtitle("Model Performance")
```

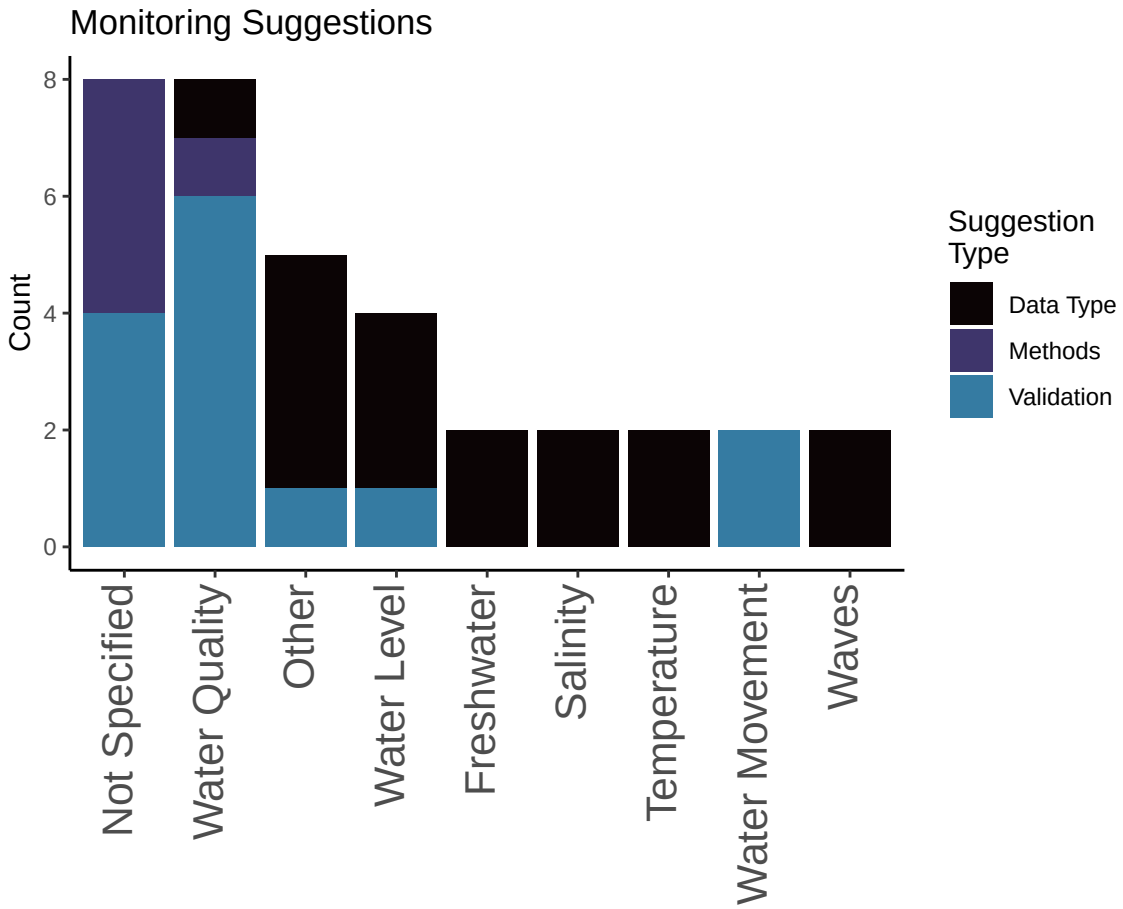


Monitoring Suggestions

```
tmp <- the_data %>%
  select(ID, Monitoring_Data_Group) %>%
  unique()
tst <- xtabs(~Monitoring_Data_Group, tmp) %>%
  sort(decreasing = TRUE) %>%
  as_tibble()

cat("Number of Unique Monitoring Records:\t")
#> Number of Unique Monitoring Records:
sum(tst$n)
#> [1] 29
```

```
tmp <-
  the_data %>%
  select(ID, Monitoring_Category, Monitoring_Data_Group) %>%
  filter(! is.na(Monitoring_Data_Group),)  %>%
  mutate(Monitoring_Data_Group = fct_infreq(Monitoring_Data_Group))
```

```
ggplot(tmp, aes(Monitoring_Data_Group, fill = Monitoring_Category)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90, size = 16,
                                   hjust = 1, vjust = 0.25)) +
  scale_fill_viridis_d(name = 'Suggestion\nType', option = "G", end = 0.5) +
  ylab('Count') +
  xlab("") +
  ggtitle("Monitoring Suggestions")
```




```
ggsave('figures/monitoring.png', type='cairo',
         width = 6, height = 5)
```

What are the "Not Specified" Comments?

```
the_data %>%
  filter(Monitoring_Data_Group == "Not Specified") %>%
  select(ID, Comment) %>%
  unique()
#> # A tibble: 8 x 2
```

```
#>      ID Comment
#>   <int> <chr>
#> 1    67 Link monitoring efforts to model runs - validation. What data to collec~
#> 2    77 How to develop constituencies for tools, data - groups that represent t~
#> 3    96 Data availability - role of autonomous vehicles in water to gather data~
#> 4   131 How can you collect data on all of the freshwater run-off occurring in ~
#> 5   133 This is a very complex environment. The 10-meter grid may not be suffic~
#> 6   153 * How to develop constituencies for tools, data - groups that represent~
#> 7   172 * Data availability - role of autonomous vehicles in water to gather da~
#> 8   180 Can the model compare its forecasts with the actual weather and conditi~
```

These principally constitute comments on model validation and monitoring methods.