Frequency Analysis of Notes From Portland HArbor Model Workshops

Curtis C. Bohlen

2022-12-09

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 capabilities 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 cn look at what ideas were expressed most commonly during the workshops.

Of course, not all paragraphs include information related to each of the five 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   0.3.5
#> 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(readxl)
library(networkD3)

theme_set(theme_classic())
```

Load Data

```
the_data <- read_excel("Export_Data_Query.xlsx" ) %>%
  mutate(ID = as.integer(ID))
head(the_data)
#> # A tibble: 6 x 15
#>      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~ Shore ~ <NA>    <NA>    <NA>         NA Waters~
#> 2     2 Live Comm~ Day ~ Use of~ Marine~ <NA>    <NA>    Waves         4 <NA>
#> 3     2 Live Comm~ Day ~ Use of~ Shore ~ <NA>    <NA>    Waves         4 <NA>
#> 4     3 Live Comm~ Day ~ MS4 pr~ Water ~ <NA>    <NA>    <NA>         NA Waters~
#> 5     4 Live Comm~ Day ~ Consid~ <NA>    <NA>    <NA>    <NA>         NA <NA>
#> 6     5 Live Comm~ Day ~ Unders~ Shore ~ <NA>    <NA>    <NA>         NA Waters~
```

```
#> # ... with 5 more variables: Extension_Timing <dbl>, Monitoring_Category <chr>,
#> #   Monitoring_Data_Group <chr>, Performance_Apply <chr>,
#> #   Performance_Criterion <chr>, 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"))
```

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)])
```

#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 400 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] 452

cat("Unique comments reviewed:\t")
#> Unique comments reviewed:
the_data %>%
  select(ID) %>%
  unique() %>%
  nrow()
#> [1] 206
```

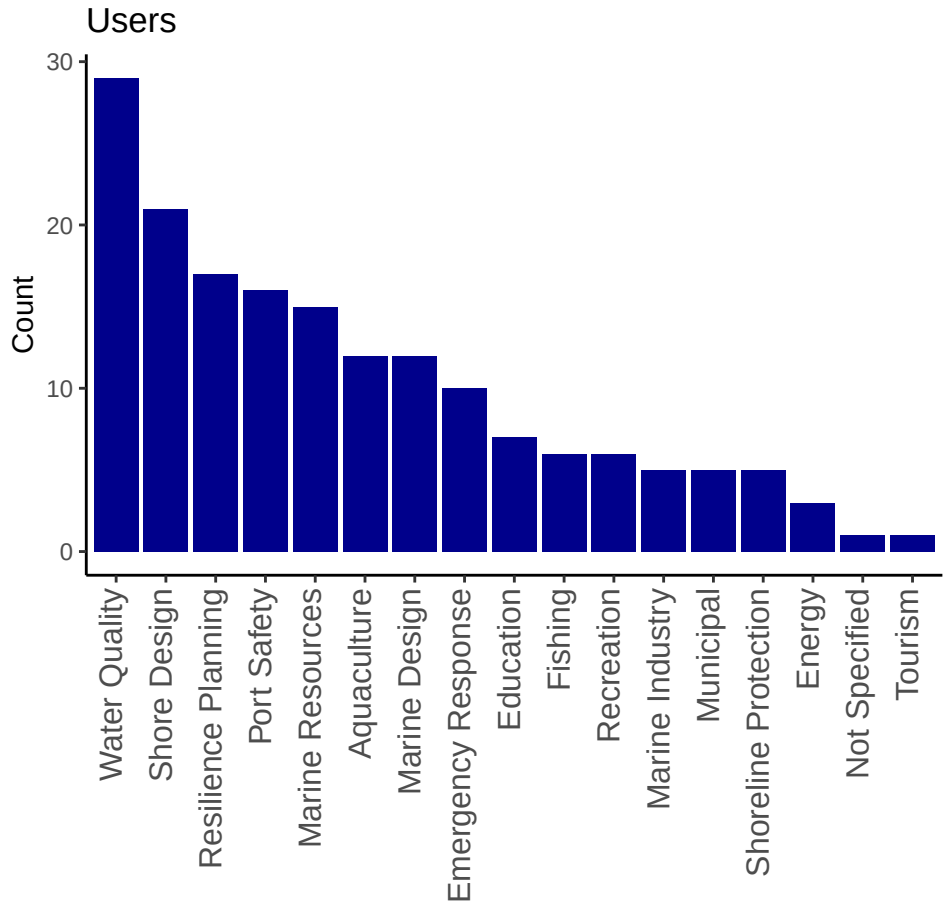
Users

```
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] 171
```

```
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 = 12,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle('Users')
```



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

```
the_data %>%
  filter(Interface_Category == "Other") %>%
```

```
  select(Comment) %>%
  unique() %>%
  as.list() %>%
  unlist
#>                                                                                                  Comment1
#> "Teaching tools - interactive interface, simple visualizations, accessible via GIS overlays, etc."
#>                                                                                                  Comment2
#>                          "Collect data on water movement, then add layer upon layer of other data."
```

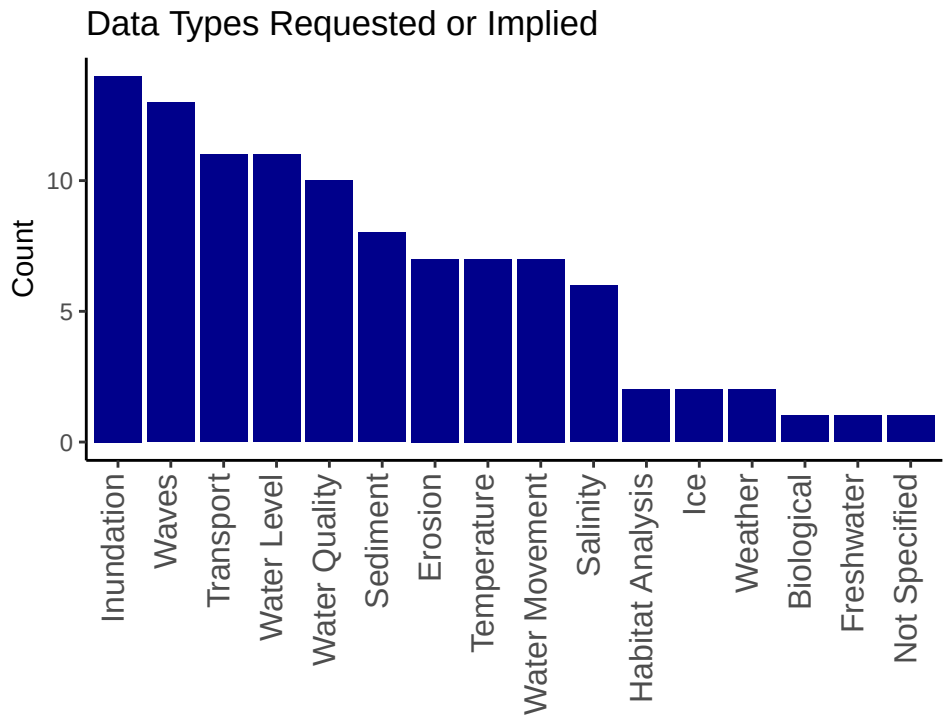
Data Types Requested

```
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] 103
```

```
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 = 12,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle("Data Types Requested or Implied")
```



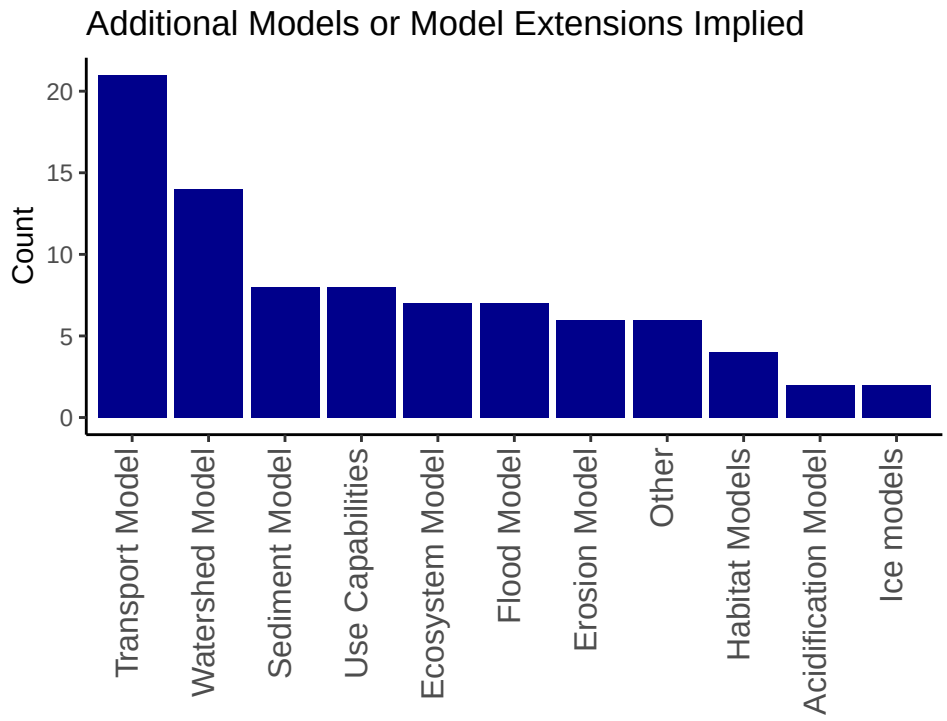
Model Extensions

```
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] 85
```

```
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 = 12,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle("Additional Models or Model Extensions Implied")
```



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

```
the_data %>%
  filter(Extension_Category == "Other") %>%
  select(ID, Comment) %>%
  unique()
#> # A tibble: 6 x 2
#>      ID Comment
#>   <int> <chr>
#> 1    53 Water level at tidal restrictions - relevant to habitat restoration and~
#> 2    65 Eutrophication zones - can predict?  Inform decisions about aquaculture~
#> 3    66 Modeling growth and impacts of aquaculture
#> 4    69 Saltwater intrusion into isolated aquifers - impacts on local (peninsul~
#> 5   109 Kelp, shellfish, other aquatic resources - are rapidly expanding in Cas~
#> 6   145 * Saltwater intrusion into isolated aquifers - impacts on local (penins~
```

So the model extensions I classified as "Other" include:

- Calculating tidal datums at specific locations,
- Finding ways to model areas at high risk for eutrophication,
- Developing decision support tools for aquaculture siting and permitting; and
- Modelling impact of rising seas on groundwater.

Another ambiguous category are the comments I read as implying something that would require expanding the capabilities of the model in other ways:

```
the_data %>%
  filter(Extension_Category == "Use Capabilities") %>%
  select(ID, Comment) %>%
  unique()
#> # A tibble: 8 x 2
#>      ID Comment
#>   <int> <chr>
#> 1    53 Water level at tidal restrictions - relevant to habitat restoration and~
#> 2    87 Planning and decision making - a tool for good visualizations that can ~
#> 3    91 Forensics for Coast Guard accident investigations, also spills / releas~
#> 4    97 Digital twins of ecosystems (Germany, Japan)?
#> 5   111 We need ways to share uses of model for community benefits - visualizat~
#> 6   138 Collect data on water movement, then add layer upon layer of other data.
#> 7   173 * Digital twins of ecosystems (Germany, Japan)?
#> 8   183 Can you demonstrate how the model develops conclusions - work backwards~
```

These include:

- Incorporating sea level rise into forecasts of tidal elevations or tidal datums at habitat restoration locations or locations of vulnerable infrastructure
- Developing versions of the model that can be used in a "what if" or exploratory manner to build shared understanding when seeking policy solutions.
- Retention of model output – at least "now cast" and short-term forecasts – to allow forensic analysis of accidents and spills.
- Ensure the model or its output can be coordinated with other data and other models.
- Using the model to conduct simulations to improve understanding of ocean process and mechanism
- Generating "digital twins" of the harbor. The concept of a digital twin generally implies the ability to run various kinds of "what if" analyses.

User Interface Ideas

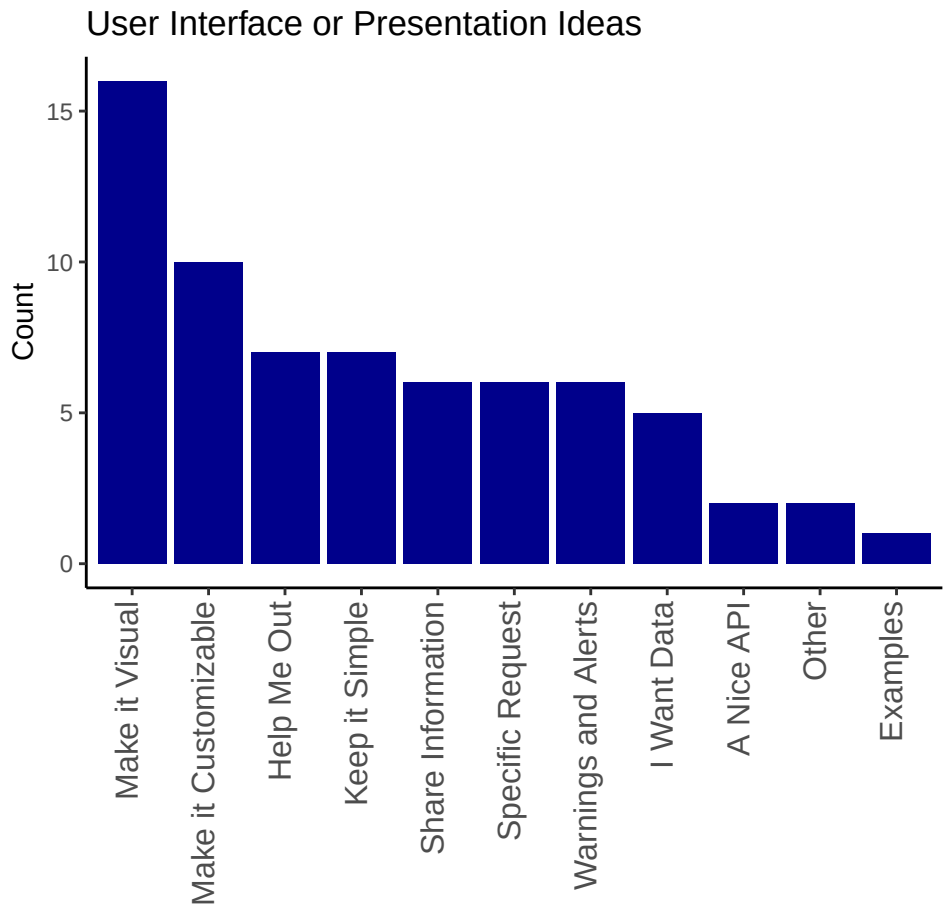
```
tmp <- the_data %>%
  select(ID, Interface_Category) %>%
  unique()
tst <- xtabs(~Interface_Category, tmp) %>%
  sort(decreasing = TRUE) %>%
  as_tibble()

cat("Number of Unique Interface Records:\t")
#> Number of Unique Interface Records:
sum(tst$n)
#> [1] 68
```

```
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 = 12,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle("User Interface or Presentation Ideas")
```

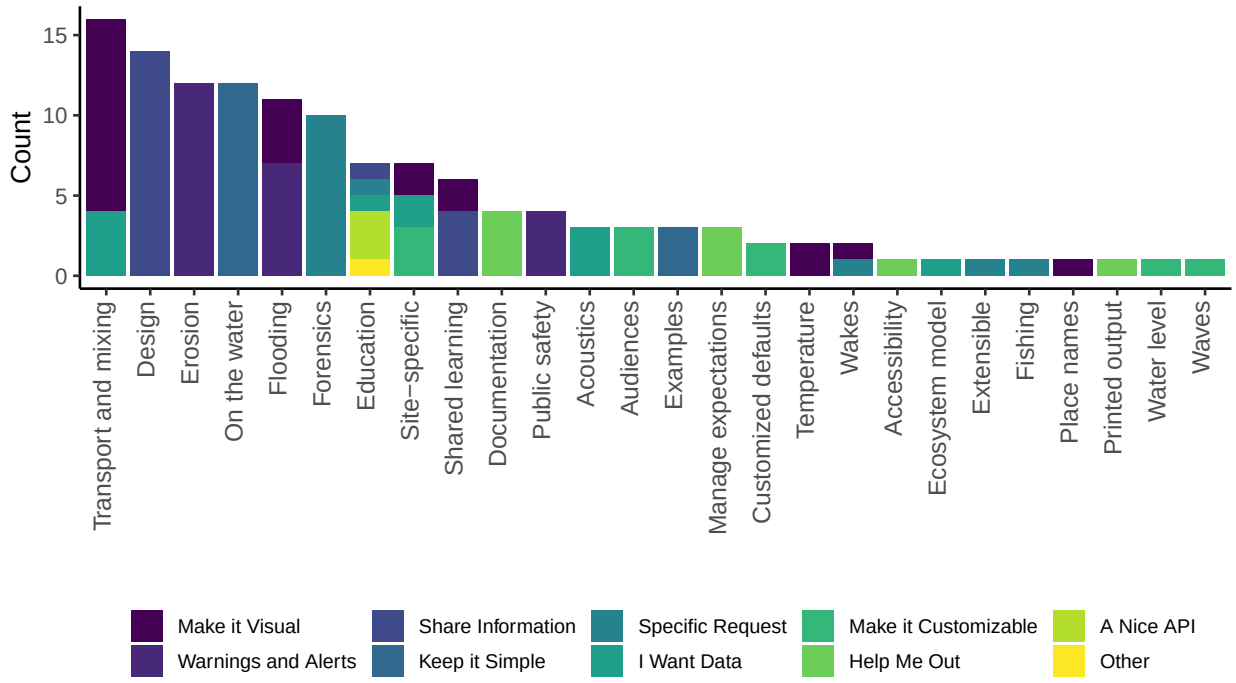


What kind of things fall into each category?

```
the_data %>%
  filter(! is.na(Interface_Category)) %>%
  filter(Interface_Group != "General") %>%
  unique() %>%
  mutate(Interface_Group = fct_infreq(Interface_Group),
         Interface_Category = fct_infreq(Interface_Category)) %>%
  ggplot(aes(Interface_Group)) +
  geom_bar(aes(fill = Interface_Category)) +
  theme(axis.text.x = element_text(angle = 90, size = 10,
                                   hjust = 1, vjust = 0.25),
        legend.text = element_text(size=8),
```

```
        legend.key.size = unit(0.5, 'cm'),
        legend.position="bottom") +
  scale_fill_viridis_d(name = "") +

  ylab('Count') +
  xlab("")
```




```
#ggsave(ggsave('Interface_ideas_bars.png', type='cairo',
 #        width = 10, height = 5))
```

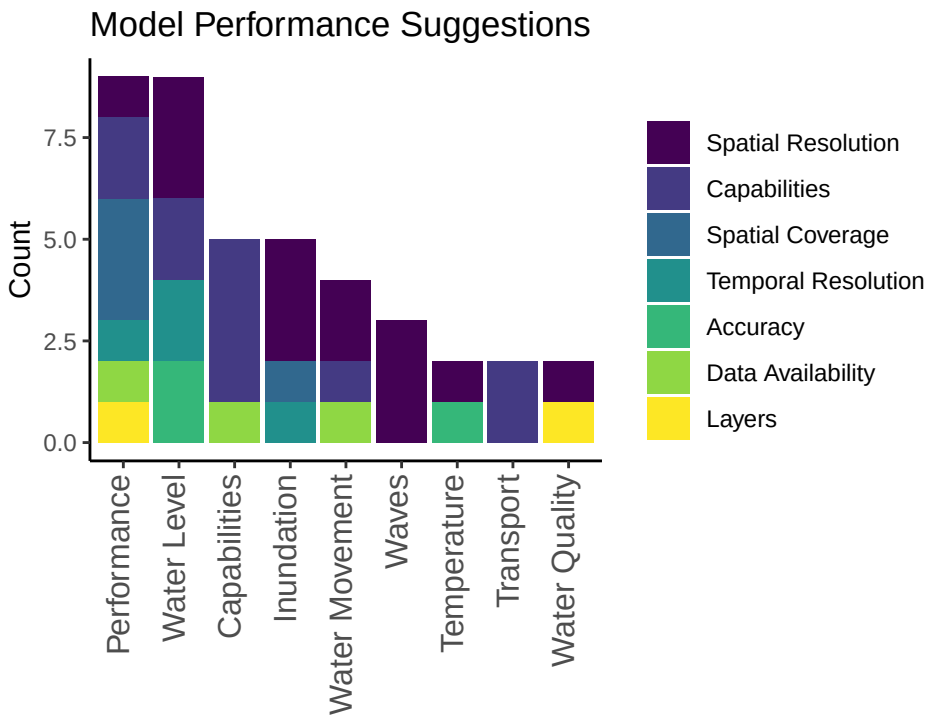
Model Performance Goals

```
tmp <- the_data %>%
  select(ID, Performance_Apply, Performance_Criterion) %>%
  unique()
xtabs(~Performance_Criterion + Performance_Apply, data = tmp) %>%
  as_tibble() %>%
  pivot_wider(names_from = Performance_Criterion, values_from = n) %>%
  mutate(row_tot = rowSums(select(., `Accuracy`:`Temporal Resolution`))) %>%
  arrange(desc(row_tot))
#> # A tibble: 9 x 9
#>   Performance_A~1 Accur~2 Capab~3 Data ~4 Layers Spati~5 Spati~6 Tempo~7 row_tot
#>   <chr>             <int>   <int>   <int>  <int>   <int>   <int>   <int>   <dbl>
#> 1 Performance           0       2       1      1       3       1       1       9
#> 2 Water Level           2       2       0      0       0       3       2       9
```

```
#> 3 Capabilities          0       4       1       0       0       0       0       5
#> 4 Inundation            0       0       0       0       1       3       1       5
#> 5 Water Movement        0       1       1       0       0       2       0       4
#> 6 Waves                 0       0       0       0       0       3       0       3
#> 7 Temperature           1       0       0       0       0       1       0       2
#> 8 Transport             0       2       0       0       0       0       0       2
#> 9 Water Quality         0       0       0       1       0       1       0       2
#> # ... with abbreviated variable names 1: Performance_Apply, 2: Accuracy,
#> #   3: Capabilities, 4: `Data Availability`, 5: `Spatial Coverage`,
#> #   6: `Spatial Resolution`, 7: `Temporal Resolution`
```

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

  ggplot(aes(Performance_Apply)) +
  geom_bar(aes(fill = Performance_Criterion)) +
  theme(axis.text.x = element_text(angle = 90, size = 12,
                                   hjust = 1, vjust = 0.25)) +
  scale_fill_viridis_d(name = '') +
  ylab('Count') +
  xlab("") +
  ggtitle("Model Performance Suggestions")
```



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 Extension Records:\t")
#> Number of Unique Extension Records:
sum(tst$n)
#> [1] 23
```

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

  ggplot(aes(Monitoring_Data_Group)) +
  geom_bar(fill = "blue4") +
  theme(axis.text.x = element_text(angle = 90, size = 12,
                                   hjust = 1, vjust = 0.25)) +
  ylab('Count') +
  xlab("") +
  ggtitle("Additional Models or Model Extensions Implied")
```