IXN D — Exploratory Data Analysis

ANONYMIZED

2023-09-08

Contents

This notebook includes *exploratory data analysis* (primarily through visualization) for data collected for the IXN-D project. These analyses were conducted by [TODO-X-ANONYMIZED] who can be reached at [TODO-X-ANONYMIZED].

Data are imported for analysis from two sources:

1. **coded utterances** — CSV file `CLEAN_coded_utterances.csv` constitutes output of content analysis coding process, which transforms linguistic output of (experimental) task transcripts into discretized units of meaning, each associated with (no more than 2) detail-code and a corresponding topic-code, indicating the topic & (detail) subject to which the utterance referred. Importantly, this sheet *also* contains a (manual) identification of *what representations* the participant was using *while making an utterance*. These data were entered by the analyst during the content analysis process.
2. **telemetry representations** — CSV file `arf_CLEAN_telemetry_representations.csv` constitutes output of telemetry wrangling scripts, which transform raw log-telemetry data from the (experimental) analysis tasks into a flat file indicating what representations (including code and data visualizations) were generated by participants at what times

Through the data import process we cast a number of dataframe variables as FACTORS to be used during analysis, and the construct the following dataframes:

1. **Raw data** [for reference, troubleshooting]
    1. `df_raw_utterances` raw utterance data from `coded_utterances.csv`
    2. `df_raw_telemetry` raw (telemetry) representation data from `telemetry_representations.csv`
2. **Wrangled data** [for analysis]
    1. `df_coded` utterance data (from `df_raw_utterances`) . Each row corresponds to one unique [utterance+detail code] pair. In most cases, utterances are unique. In a limited number of cases where utterances are *dual-coded* (the utterance contained more than one unit of meaning but could not be further discretized into smaller utterances while maintaining readability),the utterance then appears on two rows, once for each associated detail-code.
    2. `df_codedrep` utterance + (manual) representation data (from `df_raw_utterances`). This dataframe takes the `df_coded` df and pivots it to a long format where each row/observation refers to one unique [utterance+representation] combination. Many utterances were generated via reference to more than one representation (again, these representations were manually added by the data analyst doing content analysis coding, who made note of what representations the participant scrolled to, interacted with, and/or made reference to with mouse gestures).
    3. `df_telemetry` log-telemetry data (from `df_raw_telemetry`). Each row indicates a unique representation (code output, table, or visualization) generated during the (experimental) analysis tasks, and captured by the logging utility attached to the Jupyter notebook. Each row represents a unique representation, or *version* of the representation (i.e. if an interaction technique, or additional encoding is added to an existing representation in a code cell, this is tracked as a new representation.)

```
#IMPORT (wrangled) data
df_raw_utterances <- read_csv("data/CLEAN_coded_utterances.csv")
df_raw_telemetry <- read_csv("data/arf_CLEAN_telemetry_representations.csv")

##PRIMARY UTTERANCE DF
#NOT unique utterances, 1 obs for each utterance+detail-code
df_coded <- df_raw_utterances %>%
```

```
  #rename and factorize cols
  mutate(
    #UNIQUE IDS
    sid = factor(SID), #unique ID for utterance+detail-code
    pid = factor(PID, levels = c( #define level order so happiness first
      #HAPPINESS-FIRST
      "bjs827ee1u", "3r2sh20ei", "4728sjuiz","7ACC0B75","92ghd48xe","iurmer289", "s294hoei",
      #SPACE-FIRST
      "j2719eertu2","lkin27js09b","li832lin23","7382kwtue","E1D39056","8v892iige")),
    #create unique ID for utterances
    uid = factor(as.numeric(factor(paste(pid,factor(Utterance))))), #construct a unique ID for utteranc
    #recode lower case and order based on true task order
    TASK = factor(recode(Condition, "Static"="static", "Interactive"="ixn" )),
    TASK = factor(TASK, levels = c("static", "ixn")), #reorder factor levels
    #rename Notebook as DATASET
    DATASET = factor(recode(Notebook, "Happiness"="happiness", "Space"="space")),
    #create temp dataset order var
    data_order = factor(paste(TASK,"_",DATASET)), #create an order var
    data_order = recode(data_order, "ixn _ happiness"="space-first",
                                    "ixn _ space"="happiness-first",
                                    "static _ happiness"="happiness-first",
                                    "static _ space"="space-first"),
    utterance = Utterance,
    reps_group = factor(Final_Group),
    reps_all = factor(`All representations`),
    #rename flags
    flag_story = `Dylan Flag Storytelling`,
    flag_correction = `Dylan Flag Correction`,
    flag_simultaneous = `Dylan Flag Simultaneous Characterization`,
    #recode and order TOP LEVEL CODES
    code_topic = factor(Highlevel),
    code_topic = recode(code_topic, "ANALYSIS PROCESS" = "PROCESS"),
    code_topic = factor(code_topic, levels = c("PROCESS","DATASET","VARIABLE","RELATIONSHIP")),
    code_datatype = factor(`Data Type`),
    code_detail = factor(`Utterance Type`),
    #collapse two detail codes due to sparsity (less than 3 obs)
    #collapse dist.var -> dist shape
    #collapse rel faceted -> rel strength
    code_detail = recode(code_detail,
                         "distribution variance (sd, var)" = "distribution shape [shape, skew, kurtosis]
                         "relationship faceted distribution characterization" = "relationship strength a
                         ),
    #reorder factor for good graphs
    code_detail = factor(code_detail, levels = c(
    "distribution outlier (variable)",
    "distribution range [min, max]",
    "distribution shape [shape, skew, kurtosis]",
    "data size",
    "variable metadata",
    "data provenance",
    "data orientation",
    "missing data",
    "relationship range constriction",
```

```
    "relationship form (linearity/non-linearity)",
    "relationship cluster(s)/subgroup/ unexpected",
    "relationship existence / non-existence",
    "outlier (relationship)",
    "relationship strength and/or direction",
    "plan of action",
    "representation comment")),
    timestamp = adj_timestamp,
    ixn = factor(interaction_used), #was interaction used?
    PNUM = factor(PNUM,levels = c("P6", "P9", "P10", "P2", "P4", "P12","P13",
                                  "P5", "P7", "P8", "P3", "P1","P11")),

    ) %>%
  dplyr::select(sid,pid,PNUM,uid,TASK,DATASET,timestamp,ixn,code_topic,code_detail,code_datatype,
         flag_story, flag_correction, flag_simultaneous, utterance, reps_group, reps_all, data_order) %
  arrange(data_order)

#REPLACE NA in logicals to FALSE
df_coded$flag_story[is.na(df_coded$flag_story)] <- FALSE
df_coded$flag_correction[is.na(df_coded$flag_correction)] <- FALSE
df_coded$flag_simultaneous[is.na(df_coded$flag_simultaneous)] <- FALSE



##NOW WRANGLE TIME INFO
# #CALCULATE RELATIVE TASK TIMES
df_coded <- df_coded %>% mutate(
  time = hms::as_hms(timestamp)
) %>% group_by(pid, TASK) %>%
  # dplyr::summarise( .groups="keep",
  mutate(
    task_start = hms::as_hms(min(time)),
    task_end = hms::as_hms(max(time)),
    task_mins = round(difftime(task_end,task_start, units="mins"),1),
    task_second = task_end - task_start,
    relative_time_s = timestamp-task_start,
    relative_time = as.double(relative_time_s)
  ) %>% ungroup()
# %>%  dplyr::select(pid,PNUM, code_topic,code_detail, TASK,DATASET,timestamp,task_start,relative_time_


##JOINED REPRESENATIONS DURING UTTERANCES DF
#START with REPRESENTATIONS associated with UTTERANCES
#ROW is unique utterance + rep combination
#many-many relationship between utterances and representations
df_codedrep <- df_coded %>%
  dplyr::select(-data_order,-utterance, -flag_story, -flag_correction) %>%
  mutate(
    #replace "data_dictionary" with dictionary
    #rename "Multi-view Chart"
    reps_group  = factor(str_replace(reps_group, "data_dictionary", "dictionary")),
    reps_group  = factor(str_replace(reps_group, "Multi-view Chart", "multiviewchart")),
    reps_multi  = str_detect(reps_group,"_")) %>%  #flag multiple-representations
```

```
  separate_longer_delim(reps_group, delim = "_") %>% #pivot longer based _ in reps_group
  mutate (
    REP = factor(reps_group),
    #create simplified condensed detail reps
    rep_simple = recode(REP,

          "multiviewchart" = "multi-view chart",
          "heatmap" = "heatmap",
          "pairplot" = "pairplot",
          "stripplot" = "stripplot",
          "lineplot" = "lineplot",
          "scatterplot" = "scatterplot",
          "barplot" = "barplot",
          "double-profiler" = "profile",
          "profile" = "profile",
          "hist" = "histogram",
          "python" = "CODE",
          "dictionary" = "TABLE",
          "describe" = "TABLE",
          "dataframe" = "TABLE",
          "info" = "TABLE",
          "columns" = "TABLE",
          "none"  ="NONE"

          )) %>%
  dplyr::select(-reps_group) %>%  #drop reps_group column since now separated
  mutate(
    rep_type = recode(REP,
                      "hist" = "CHART",
                      "profile" = "CHART",
                      "scatterplot" = "CHART",
                      "barplot" = "CHART",
                      "stripplot" = "CHART",
                      "lineplot" = "CHART",
                      "heatmap" = "CHART",
                      "pairplot" = "CHART",
                      "multiviewchart" = "CHART",
                      "double-profiler" = "CHART",
                      "profile" = "CHART",
                      "python" = "CODE",
                      "dictionary" = "CODE",
                      "describe" = "CODE",
                      "dataframe" = "CODE",
                      "info" = "CODE",
                      "columns" = "CODE",
                      "none"  ="NONE"
  ))

##PRIMARY REPRESENATION DF
#ROW ==?
df_telemetry <- df_raw_telemetry %>%
  mutate(
```

```
  #PARTICIPANT DATA
  pid = factor(pID, levels = c( #define level order so happiness first
    #HAPPINESS-FIRST
    "bjs827ee1u", "3r2sh20ei", "4728sjuiz","7ACC0B75","92ghd48xe","iurmer289", "s294hoei",
    #SPACE-FIRST
    "j2719eertu2","lkin27js09b","li832lin23","7382kwtue","E1D39056","8v892iige")),
  PNUM = factor(PNUM,levels = c("P6", "P9", "P10", "P2", "P4", "P12","P13",
                                "P5", "P7", "P8", "P3", "P1","P11")),



  TASK = factor(TASK, levels = c("static", "ixn")), #reorder factor levels
  DATASET = factor(session, levels = c("happiness","space")),
  #create temp dataset order var
  data_order = factor(paste(TASK,"_",DATASET)), #create an order var
  data_order = recode(data_order, "ixn _ happiness"="space-first",
                                  "ixn _ space"="happiness-first",
                                  "static _ happiness"="happiness-first",
                                  "static _ space"="space-first"),
  timestamp = timestamp,
  time_elapsed = time_elapsed,
  technique = factor(`Interaction_Techniques`),
  IXN = (technique!="none"),

  code = cell_content,
  REP = factor(merged_output_type),
  REP = factor(str_replace(REP, "Multi-view Chart", "multiviewchart"))) %>%
filter( #EXCLUDE SOME REPS
  #none and other are python code not of the tabular forms we look for
  REP %nin% c("none","error","markdown","other")
) %>% mutate(
  REP = recode_factor(REP,
              "double-profiler" = "profile",
              "countplot" = "barplot"),
  REP = factor(REP), #reset factor levels
  rep_type = recode(REP,
                    # "dictionary" = "TABLE",
                    "describe" = "TABLE",
                    "dataframe" = "TABLE",
                    "info" = "TABLE",
                    "columns" = "TABLE",
                    "hist" = "CHART",
                    "profile" = "CHART",
                    "double-profiler" = "CHART",
                    "countplot" = "CHART",
                    "barplot" = "CHART",
                    "scatterplot" = "CHART",
                    "lineplot" = "CHART",
                    "stripplot" = "CHART",
                    "pairplot" = "CHART",
                    "heatmap" = "CHART",
                    "multiviewchart" = "CHART"
                    )) %>%
```

```
  dplyr::select(pid,PNUM,TASK,DATASET,data_order,timestamp,time_elapsed,technique,IXN,REP,rep_type,cell
```

The following variables, some common across dataframes, are especially important to the subsequent analyses:

1. `PID` is a (unique) random identifier for participant
2. `PNUM` is the corresponding 'participant number' (e.g. P1, P2, etc) for each `PID` as used in the paper
3. `TASK` refers to the experimental task (within-subjects; repeated measures) either `static` or `ixn` (interactive)
4. `DATASET` refers to the dataset attached to the corresponding `TASK`, either `happiness` (had a numeric-continuous outcome variable) or `space` (had a nomial-categorical outcome variable)
5. `data_order` indicates the dataset counterbalancing order for the corresponding participant, either `space-first` or `happiness-first`. Most graphs are *sorted* by `data_order`
6. `code_topic` is the outcome variable indicating the highest level code applied during content analysis
7. `code_detail` is the outcome variable indicating the lowest-level (detail) code applied during content analysis
8. `ixn` is a boolean flag (for utterance dfs only) indicating whether the participant was *actively engaging with an interactive visualization* while making the utterance. (i.e. inspecting/reading an interactive graph without using the interaction feature is coded as `FALSE`)
9. `relative_time` indicates the time (in seconds) after the start of the task at which the utterance occurs
10. `rep_simple` is an outcome variable indicating what kind of representation what was used while the utterance was made (detail of visualization type + one category for code output + one category for code output that yields a table, such as df.describe(), df.info(), etc.)
11. `rep_type` is an outcome variable indicating the high-level category for what kind of representation was used while the utterance was made: `CHART`, `CODE` (including tables) or `NONE`
12. `REP` is used in the `df_telemetry` df to refer to the detail kind of representation
13. `rep_type` is used in the `df_telemetry` df to refer to high level kind of representation, but does not include code cells *other* than those that generate tables. (Also does not contain NONE as a valid value because NONE refers to utterances made without reference to a representation, and the telemetry data contains only representations, not utterances)

PROFILE DATA

PROFILE UTTERANCE-CODE

There are 742 rows in the `df_coded` dataset, where each row represents an utterance+coding (i.e. utterance + detail code). There are 662 *unique* utterances. The difference indicates utterances that were *dual-coded* (i.e. two detail-level codes). No more than two codes were applied to a single utterance. For the purposes of analysis, dual-coded utterances will be treated as two utterances, as they have two distinct (but lexically insepeperable) units of meaning.

The following profile reflects a dataframe where individual observations refer to unique utterance+detail_code.

```
df_coded%>% summarytools::dfSummary(
              plain.ascii  = FALSE,
              graph.magnif = 0.75,
              style        = "grid",
              tmp.img.dir  = "temp",
              missing.col = FALSE,
              method = "render"
)
```

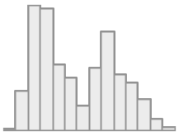
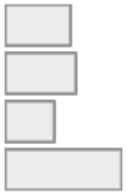
Data Frame Summary

df_coded **Dimensions:** 742 x 25
Duplicates: 0

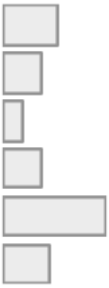
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	sid [factor]	1. 0 2. 1 3. 2 4. 3 5. 4 6. 5 7. 6 8. 7 9. 8 10. 9 [732 others]	1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 732 (98.7%)		742 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
						
2	pid [factor]	1. bjs827ee1u 2. 3r2sh20ei 3. 4728sjuiz 4. 7ACC0B75 5. 92ghd48xe 6. iurmer289 7. s294hoei 8. j2719eertu2 9. lkin27js09b 10. li832lin23 [3 others]	29 (3.9%) 103 (13.9%) 43 (5.8%) 28 (3.8%) 56 (7.5%) 87 (11.7%) 88 (11.9%) 82 (11.1%) 48 (6.5%) 51 (6.9%) 127 (17.1%)		742 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
						
3	PNUM [factor]	1. P6 2. P9 3. P10 4. P2 5. P4 6. P12 7. P13 8. P5 9. P7 10. P8 [3 others]	29 (3.9%) 103 (13.9%) 43 (5.8%) 28 (3.8%) 56 (7.5%) 87 (11.7%) 88 (11.9%) 82 (11.1%) 48 (6.5%) 51 (6.9%) 127 (17.1%)		742 (100.0%)	0 (0.0%)
						
4	uid [factor]	1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 [652 others]	2 (0.3%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 2 (0.3%) 1 (0.1%) 1 (0.1%) 730 (98.4%)		742 (100.0%)	0 (0.0%)

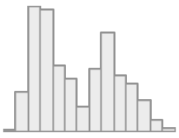
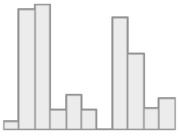
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
5	TASK [factor]	1. static 2. ixn	403 (54.3%) 339 (45.7%)		742 (100.0%)	0 (0.0%)
6	DATASET [factor]	1. happiness 2. space	431 (58.1%) 311 (41.9%)		742 (100.0%)	0 (0.0%)
7	timestamp [hms, difftime]	min : 622 med : 2857 max : 6900 units : secs	622 distinct values		742 (100.0%)	0 (0.0%)
8	ixn [factor]	1. FALSE 2. TRUE	633 (85.3%) 109 (14.7%)		742 (100.0%)	0 (0.0%)
9	code_topic [factor]	1. PROCESS 2. DATASET 3. VARIABLE 4. RELATIONSHIP	160 (21.6%) 176 (23.7%) 122 (16.4%) 284 (38.3%)		742 (100.0%)	0 (0.0%)

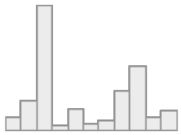
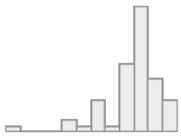
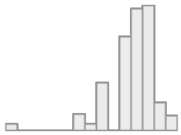
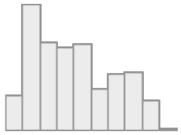
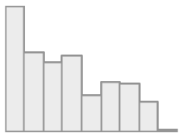
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
10	code_detail [factor]	1. distribution outlier (var 2. distribution range [min, 3. distribution shape [shape 4. data size 5. variable metadata 6. data provenance 7. data orientation 8. missing data 9. relationship range constr 10. relationship form (linear [6 others]	9 (1.2%) 33 (4.4%) 80 (10.8%) 9 (1.2%) 64 (8.6%) 11 (1.5%) 16 (2.2%) 76 (10.2%) 8 (1.1%) 15 (2.0%) 421 (56.7%)		742 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
11	code_datatype [factor]	1. distribution (continuous 2. distribution (categorical 3. relationship (categorical 4. relationship (categorical 5. relationship (continuous 6. relationship (multivariat	76 (17.8%) 54 (12.7%) 28 (6.6%) 55 (12.9%) 146 (34.3%) 67 (15.7%)		426 (57.4%)	316 (42.6%)
12	flag_story [logical]	1. FALSE 2. TRUE	700 (94.3%) 42 (5.7%)		742 (100.0%)	0 (0.0%)
13	flag_correction [logical]	1. FALSE 2. TRUE	733 (98.8%) 9 (1.2%)		742 (100.0%)	0 (0.0%)
14	flag_simultaneous [logical]	1. FALSE 2. TRUE	682 (91.9%) 60 (8.1%)		742 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
15	utterance [character]	1. [Talking about the profil 2. actually, let me see if p 3. Although we have like les 4. And are they within range 5. And confidence in governm 6. And just I want to see ho 7. And so it looks like it s 8. And then if I had more ti 9. Because it does seem like 10. Data frame. Got a bunch o [652 others]	2 (0.3%) 2 (0.3%) 2 (0.3%) 2 (0.3%) 2 (0.3%) 2 (0.3%) 2 (0.3%) 2 (0.3%) 2 (0.3%) 2 (0.3%) 722 (97.3%)		742 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
16	reps_group [factor]	1. barplot 2. columns 3. columns_data_dictionary 4. data_dictionary 5. data_dictionary_dataframe 6. data_dictionary_describe 7. dataframe 8. dataframe_describe 9. dataframe_heatmap 10. dataframe_pairplot [15 others]	16 (2.2%) 4 (0.5%) 1 (0.1%) 56 (7.5%) 1 (0.1%) 9 (1.2%) 76 (10.2%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 576 (77.6%)		742 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
17	reps_all [factor]	1. affect_corruption_brush_7 2. Age_CryoSleep_scatterplot 3. age_CryoSleep_ShoppingMal 4. Age_RoomService_scatterpl 5. age_roomservice_scatterpl 6. Age_RoomService_scatterpl 7. Age_ShoppingMall_scatterp 8. altair_profile_contVars_j 9. alx_barplot_df_homeplanet 10. alx_barplot_df_homeplanet [245 others]	4 (0.6%) 3 (0.4%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 2 (0.3%) 1 (0.1%) 693 (97.7%)		709 (95.6%)	33 (4.4%)
18	data_order [factor]	1. space-first 2. happiness-first	308 (41.5%) 434 (58.5%)		742 (100.0%)	0 (0.0%)
19	time [hms, difftime]	min : 622 med : 2857 max : 6900 units : secs	622 distinct values		742 (100.0%)	0 (0.0%)
20	task_start [hms, difftime]	min : 622 med : 2123 max : 5349 units : secs	26 distinct values		742 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
21	task_end [hms, difftime]	min : 1811 med : 3815 max : 6900 units : secs	26 distinct values		742 (100.0%)	0 (0.0%)
22	task_mins [difftime]	min : 6.5 med : 24.1 max : 28.5 units : mins	26 distinct values		742 (100.0%)	0 (0.0%)
23	task_second [difftime]	min : 388 med : 1449 max : 1712 units : secs	26 distinct values		742 (100.0%)	0 (0.0%)
24	relative_time_s [difftime]	min : 0 med : 552 max : 1712 units : secs	506 distinct values		742 (100.0%)	0 (0.0%)
25	relative_time [numeric]	Mean (sd) : 616.1 (459) min < med < max: 0 < 552 < 1712 IQR (CV) : 794.8 (0.7)	506 distinct values		742 (100.0%)	0 (0.0%)

[TODO-X-ANONYMIZED] has reviewed data profile for missing data and correct factorization.

PROFILE UTTERANCE-REPRESENTATION

There are 760 rows in the `df_codedrep` dataset, where each row represents an utterance+representation (i.e. utterance + detail-level rep). There are 662 *unique* utterances (*should match* ***`df_coded`*** *dataframe*). The difference indicates utterances that used *multiple representations* (i.e. two graphs, or a graph and a table).

The following profile reflects a dataframe where individual observations refer to unique utterance+representation.

```
df_codedrep%>% summarytools::dfSummary(
              plain.ascii  = FALSE,
              graph.magnif = 0.75,
              style        = "grid",
              tmp.img.dir  = "temp",
              missing.col = FALSE,
              method = "render"
)
```

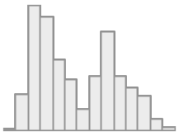
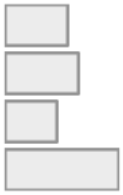
Data Frame Summary

df_codedrep **Dimensions:** 760 x 24
Duplicates: 0

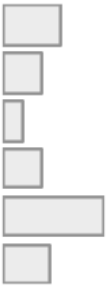
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	sid [factor]	1. 0 2. 1 3. 2 4. 3 5. 4 6. 5 7. 6 8. 7 9. 8 10. 9 [732 others]	1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 750 (98.7%)		760 (100.0%)	0 (0.0%)

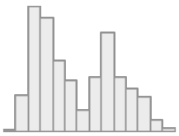
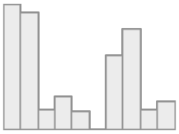
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
						
2	pid [factor]	1. bjs827ee1u 2. 3r2sh20ei 3. 4728sjuiz 4. 7ACC0B75 5. 92ghd48xe 6. iurmer289 7. s294hoei 8. j2719eertu2 9. lkin27js09b 10. li832lin23 [3 others]	29 (3.8%) 107 (14.1%) 43 (5.7%) 28 (3.7%) 63 (8.3%) 87 (11.4%) 88 (11.6%) 87 (11.4%) 48 (6.3%) 51 (6.7%) 129 (17.0%)		760 (100.0%)	0 (0.0%)

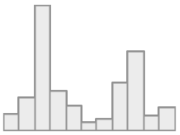
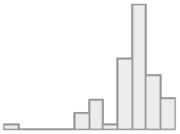
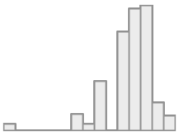
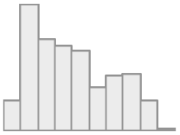
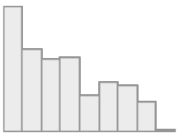
No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
						
3	PNUM [factor]	1. P6 2. P9 3. P10 4. P2 5. P4 6. P12 7. P13 8. P5 9. P7 10. P8 [3 others]	29 (3.8%) 107 (14.1%) 43 (5.7%) 28 (3.7%) 63 (8.3%) 87 (11.4%) 88 (11.6%) 87 (11.4%) 48 (6.3%) 51 (6.7%) 129 (17.0%)		760 (100.0%)	0 (0.0%)
						
4	uid [factor]	1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10 [652 others]	2 (0.3%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 2 (0.3%) 1 (0.1%) 1 (0.1%) 748 (98.4%)		760 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
5	TASK [factor]	1. static 2. ixn	418 (55.0%) 342 (45.0%)		760 (100.0%)	0 (0.0%)
6	DATASET [factor]	1. happiness 2. space	441 (58.0%) 319 (42.0%)		760 (100.0%)	0 (0.0%)
7	timestamp [hms, difftime]	min : 622 med : 2829.5 max : 6900 units : secs	622 distinct values		760 (100.0%)	0 (0.0%)
8	ixn [factor]	1. FALSE 2. TRUE	651 (85.7%) 109 (14.3%)		760 (100.0%)	0 (0.0%)
9	code_topic [factor]	1. PROCESS 2. DATASET 3. VARIABLE 4. RELATIONSHIP	161 (21.2%) 182 (23.9%) 131 (17.2%) 286 (37.6%)		760 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
10	code_detail [factor]	1. distribution outlier (var 2. distribution range [min, 3. distribution shape [shape 4. data size 5. variable metadata 6. data provenance 7. data orientation 8. missing data 9. relationship range constr 10. relationship form (linear [6 others]	10 (1.3%) 40 (5.3%) 81 (10.7%) 9 (1.2%) 67 (8.8%) 11 (1.4%) 17 (2.2%) 78 (10.3%) 8 (1.1%) 15 (2.0%) 424 (55.8%)		760 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
11	code_datatype [factor]	1. distribution (continuous 2. distribution (categorical 3. relationship (categorical 4. relationship (categorical 5. relationship (continuous 6. relationship (multivariat	84 (19.2%) 54 (12.4%) 28 (6.4%) 57 (13.0%) 147 (33.6%) 67 (15.3%)		437 (57.5%)	323 (42.5%)
12	flag_simultaneous [logical]	1. FALSE 2. TRUE	699 (92.0%) 61 (8.0%)		760 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
13	reps_all [factor]	1. affect_corruption_brush_7 2. Age_CryoSleep_scatterplot 3. age_CryoSleep_ShoppingMal 4. Age_RoomService_scatterpl 5. age_roomservice_scatterpl 6. Age_RoomService_scatterpl 7. Age_ShoppingMall_scatterp 8. altair_profile_contVars_j 9. alx_barplot_df_homeplanet 10. alx_barplot_df_homeplanet [245 others]	4 (0.6%) 3 (0.4%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 1 (0.1%) 2 (0.3%) 1 (0.1%) 711 (97.8%)		727 (95.7%)	33 (4.3%)
14	time [hms, difftime]	min : 622 med : 2829.5 max : 6900 units : secs	622 distinct values		760 (100.0%)	0 (0.0%)
15	task_start [hms, difftime]	min : 622 med : 2123 max : 5349 units : secs	26 distinct values		760 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
16	task_end [hms, difftime]	min : 1811 med : 3283 max : 6900 units : secs	26 distinct values		760 (100.0%)	0 (0.0%)
17	task_mins [difftime]	min : 6.5 med : 24.1 max : 28.5 units : mins	26 distinct values		760 (100.0%)	0 (0.0%)
18	task_second [difftime]	min : 388 med : 1449 max : 1712 units : secs	26 distinct values		760 (100.0%)	0 (0.0%)
19	relative_time_s [difftime]	min : 0 med : 531.5 max : 1712 units : secs	506 distinct values		760 (100.0%)	0 (0.0%)
20	relative_time [numeric]	Mean (sd) : 610.9 (456.4) min < med < max: 0 < 531.5 < 1712 IQR (CV) : 761 (0.7)	506 distinct values		760 (100.0%)	0 (0.0%)
21	reps_multi [logical]	1. FALSE 2. TRUE	724 (95.3%) 36 (4.7%)		760 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
22	REP [factor]	1. barplot 2. columns 3. dataframe 4. describe 5. dictionary 6. double-profiler 7. heatmap 8. hist 9. info 10. lineplot [7 others]	16 (2.1%) 5 (0.7%) 83 (10.9%) 34 (4.5%) 67 (8.8%) 23 (3.0%) 20 (2.6%) 6 (0.8%) 13 (1.7%) 36 (4.7%) 457 (60.1%)		760 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
23	rep_simple [factor]	1. barplot 2. TABLE 3. profile 4. heatmap 5. histogram 6. lineplot 7. multi-view chart 8. NONE 9. pairplot 10. CODE [2 others]	16 (2.1%) 202 (26.6%) 133 (17.5%) 20 (2.6%) 6 (0.8%) 36 (4.7%) 59 (7.8%) 43 (5.7%) 51 (6.7%) 60 (7.9%) 134 (17.6%)		760 (100.0%)	0 (0.0%)
24	rep_type [factor]	1. CHART 2. CODE 3. NONE	455 (59.9%) 262 (34.5%) 43 (5.7%)		760 (100.0%)	0 (0.0%)

[TODO-X-ANONYMIZED] has reviewed data profile for missing data and correct factorization.

PROFILE REPRESENTATIONS

There are 504 rows in the `df_telemetry` dataset, where each row represents a unique representation generated by a participant during an analysis task, as captured by the logging utility.

There are 504 *unique* representation-versions. The difference between this and the number of representations referenced in the`df_codedrep` is a result of the orientation and source of the data: `df_telemetry` may contain representations that are not used in the course of making an utterance, and `df_codedrep` may include representations that are referenced multiple times in the course of multiple utterances. There is no variable on which representations from `df_telemetry` and `df_codedrep` can joined.

The following profile reflects a dataframe where individual observations refer to unique representation_versions.

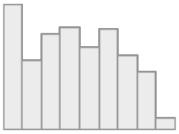
```
df_telemetry%>% summarytools::dfSummary(
              plain.ascii  = FALSE,
              graph.magnif = 0.75,
              style        = "grid",
              tmp.img.dir  = "temp",
              missing.col = FALSE,
              method = "render"
)
```

Data Frame Summary

df_telemetry **Dimensions:** 504 x 12
Duplicates: 16

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
1	pid [factor]	1. bjs827ee1u 2. 3r2sh20ei 3. 4728sjuiz 4. 7ACC0B75 5. 92ghd48xe 6. iurmer289 7. s294hoei 8. j2719eertu2 9. lkin27js09b 10. li832lin23 [3 others]	43 (8.5%) 41 (8.1%) 33 (6.5%) 19 (3.8%) 31 (6.2%) 27 (5.4%) 31 (6.2%) 39 (7.7%) 15 (3.0%) 65 (12.9%) 160 (31.7%)		504 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
2	PNUM [factor]	1. P6 2. P9 3. P10 4. P2 5. P4 6. P12 7. P13 8. P5 9. P7 10. P8 [3 others]	43 (8.5%) 41 (8.1%) 33 (6.5%) 19 (3.8%) 31 (6.2%) 27 (5.4%) 31 (6.2%) 39 (7.7%) 15 (3.0%) 65 (12.9%) 160 (31.7%)		504 (100.0%)	0 (0.0%)
3	TASK [factor]	1. static 2. ixn	232 (46.0%) 272 (54.0%)		504 (100.0%)	0 (0.0%)
4	DATASET [factor]	1. happiness 2. space	301 (59.7%) 203 (40.3%)		504 (100.0%)	0 (0.0%)
5	data_order [factor]	1. space-first 2. happiness-first	279 (55.4%) 225 (44.6%)		504 (100.0%)	0 (0.0%)
6	timestamp [numeric]	Min : 1.68e+12 Mean : 1683018480492.8 Max : 1.69e+12	1.68e+12 : 340 (69.8%) 1.69e+12 : 147 (30.2%)		487 (96.6%)	17 (3.4%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
7	time_elapsed [numeric]	Mean (sd) : 745818.9 (466847.7) min < med < max: 1948 < 736053 < 1673757 IQR (CV) : 776174 (0.6)	398 distinct values		398 (79.0%)	106 (21.0%)
8	technique [factor]	1. filter_brush 2. filter_brush+filter_slide 3. filter_brush+filter_slide 4. filter_brush+filter_slide 5. filter_slider 6. filter_slider+highlight_b 7. filter_slider+highlight_b 8. filter_slider+highlight_b 9. filter_slider+highlight_p 10. filter_slider+pan_zoom+to [11 others]	11 (2.2%) 2 (0.4%) 2 (0.4%) 1 (0.2%) 11 (2.2%) 5 (1.0%) 2 (0.4%) 10 (2.0%) 2 (0.4%) 4 (0.8%) 454 (90.1%)		504 (100.0%)	0 (0.0%)
9	IXN [logical]	1. FALSE 2. TRUE	337 (66.9%) 167 (33.1%)		504 (100.0%)	0 (0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
						
10	REP	1. profile	42 (8.3%)		504	0
	[factor]	2. barplot	43 (8.5%)		(100.0%)	(0.0%)
		3. columns	30 (6.0%)			
		4. dataframe	101 (20.0%)			
		5. describe	20 (4.0%)			
		6. heatmap	7 (1.4%)			
		7. hist	19 (3.8%)			
		8. info	6 (1.2%)			
		9. lineplot	18 (3.6%)			
		10. multiviewchart	43 (8.5%)			
		[3 others]	175 (34.7%)			
						
11	rep_type	1. CHART	347 (68.8%)		504	0
	[factor]	2. TABLE	157 (31.2%)		(100.0%)	(0.0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Graph	Valid	Missing
						
12	cell_content [character]	1. df.head() 2. df 3. df.columns 4. df.describe() 5. df.info() 6. columns = df.columns # re 7. df.isna() 8. df.isnull() 9. alx.pairplot(df) 10. alx.lineplot(df[df.countr [374 others]	19 (3.8%) 18 (3.6%) 10 (2.0%) 9 (1.8%) 7 (1.4%) 5 (1.0%) 5 (1.0%) 5 (1.0%) 4 (0.8%) 3 (0.6%) 419 (83.1%)		504 (100.0%)	0 (0.0%)

[TODO-X-ANONYMIZED] has reviewed data profile for missing data and correct factorization.

EXPLORE UTTERANCES

Utterances are the lowest-level discrete units of meaning transcribed from the EDA Task transcripts. Utterances are coded at two levels of analysis: (1) **topic-code** gives a *high level* topic of the participant's verbalization, (2) **detail-code** gives the *lower level* detail of the subject.

In the following subsections we explore the distribution of *number of utterances* based on TASK, DATASET, and PARTICIPANT, before describing the distribution of utterances *through the timecourse* of the TASK.

[Number of] Utterances

FIRST we explore the distribution of utterances by Analysis Task, Dataset, Participant and Time, irrespective of what the utterance was about (topic, detail).

RQ: How much did participants talk aloud during EDA? When did they talk aloud?

Answer: *Inspection of frequency tables and visualizations suggests that the most substantial determinant of how many utterances an individual made is individual participant-level differences, rather than structural differences imposed by the TASK or DATASET. This is not altogether unexpected given the fact that across both tasks (static/interactive) and datasets the structure of the experimental task was the same*

by TASK

```
print("BY TASK")
```

[1] “BY TASK”

```
freq(df_coded$TASK,
     cumul      = FALSE,
     headings   = FALSE,
     report.nas = FALSE,
     plain.ascii = FALSE)
```

	Freq	%
static	403	54.31
ixn	339	45.69
Total	742	100.00

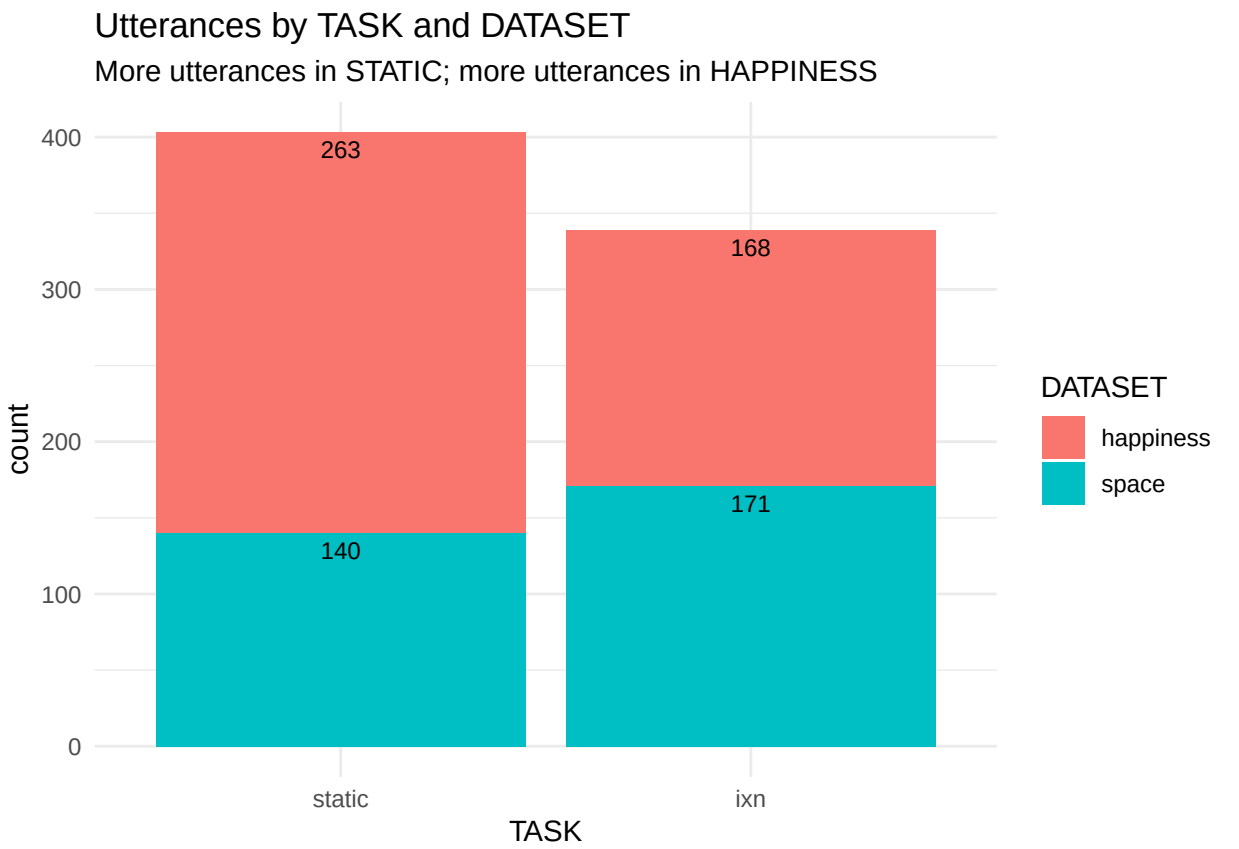
by TASK and DATASET

```
#COUNT BY TASK AND DATASET
ctable(x = df_coded$TASK,
       y = df_coded$DATASET,
       prop = "t")
```

Cross-Tabulation, Total Proportions
TASK * DATASET
Data Frame: df_coded

DATASET	happiness	space	Total
TASK			
static	263 (35.4%)	140 (18.9%)	403 (54.3%)
ixn	168 (22.6%)	171 (23.0%)	339 (45.7%)
Total	431 (58.1%)	311 (41.9%)	742 (100.0%)

```
#DF SUMMARIZED BY TASK + DATASET
df_summary <- df_coded %>%
  group_by(TASK,DATASET) %>%
  dplyr::summarise(
    c = n()
  )

#STACKED BAR BY TASK
ggplot(df_summary, aes(x = TASK, y=c, fill= DATASET)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  # scale_fill_brewer(type="qual", palette = 4) +
  labs( title = "Utterances by TASK and DATASET",
        subtitle = "More utterances in STATIC; more utterances in HAPPINESS",
        x= "TASK", y = "count") + theme_minimal()
```




```
# + theme(legend.position = "blank")
```

by PARTICIPANT

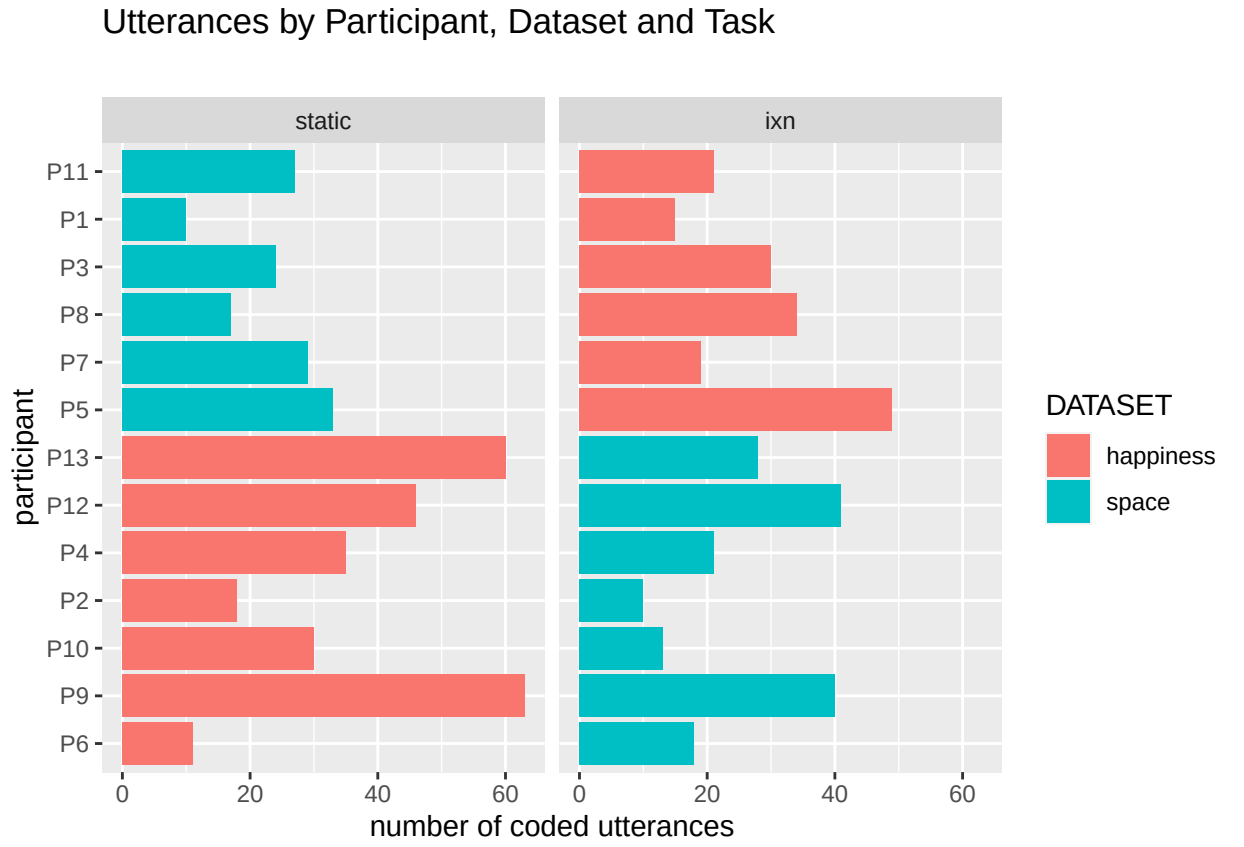
```
#COUNT BY PARTICIPANT AND TASK
ctable(x = df_coded$PNUM,
       y = df_coded$TASK,
       prop = "r")
```

Cross-Tabulation, Row Proportions
PNUM * TASK
Data Frame: df_coded

TASK	static	ixn	Total
PNUM			
P6	11 (37.9%)	18 (62.1%)	29 (100.0%)
P9	63 (61.2%)	40 (38.8%)	103 (100.0%)
P10	30 (69.8%)	13 (30.2%)	43 (100.0%)
P2	18 (64.3%)	10 (35.7%)	28 (100.0%)
P4	35 (62.5%)	21 (37.5%)	56 (100.0%)
P12	46 (52.9%)	41 (47.1%)	87 (100.0%)
P13	60 (68.2%)	28 (31.8%)	88 (100.0%)
P5	33 (40.2%)	49 (59.8%)	82 (100.0%)

P7	29 (60.4%)	19 (39.6%)	48 (100.0%)
P8	17 (33.3%)	34 (66.7%)	51 (100.0%)
P3	24 (44.4%)	30 (55.6%)	54 (100.0%)
P1	10 (40.0%)	15 (60.0%)	25 (100.0%)
P11	27 (56.2%)	21 (43.8%)	48 (100.0%)
Total	403 (54.3%)	339 (45.7%)	742 (100.0%)

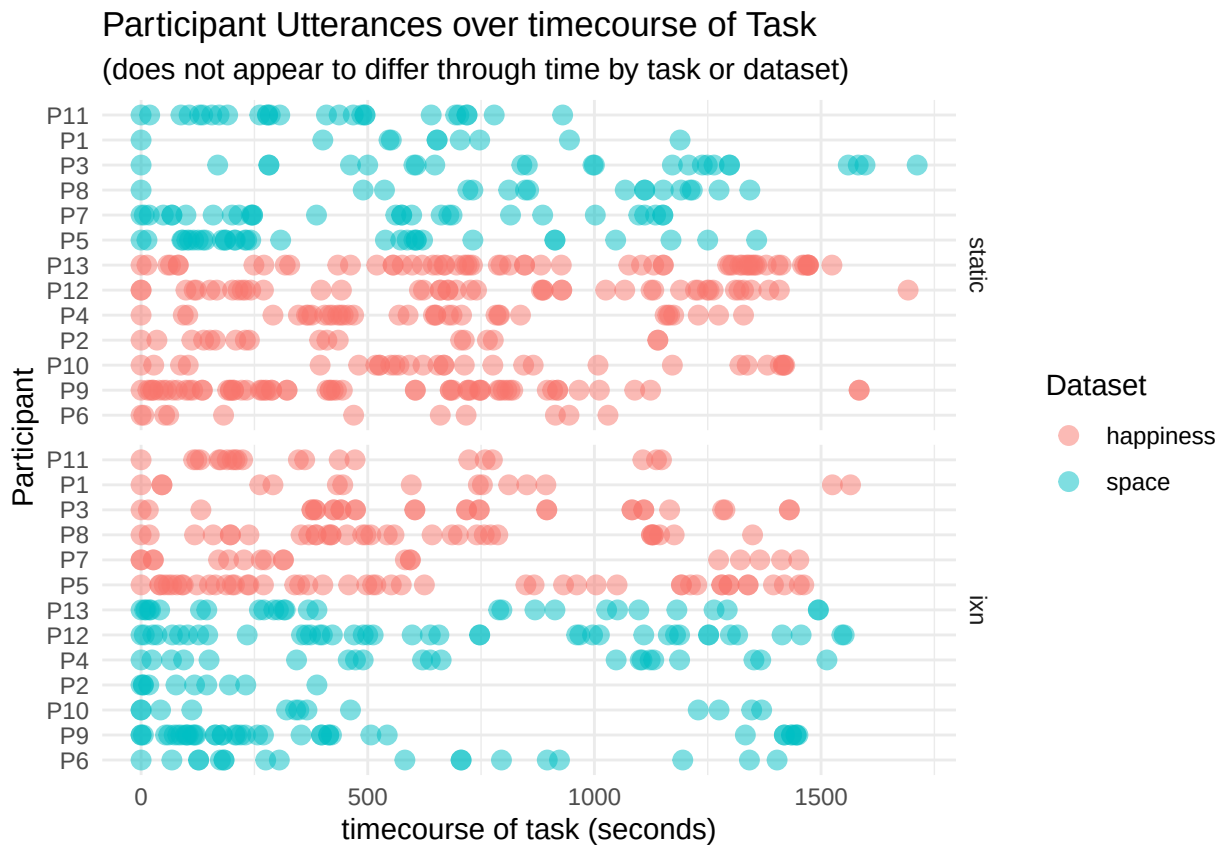
```
#UTTERANCES by PARTICPANT facet TASK color DATASET
gf_bar( PNUM ~., fill = ~ DATASET, data = df_coded) %>%
  gf_facet_grid(.~TASK) +
  labs(
    title = "Utterances by Participant, Dataset and Task",
    subtitle = "",
    x = "number of coded utterances",
    y = "participant",
    fill = "DATASET"
  )
```



through TIME

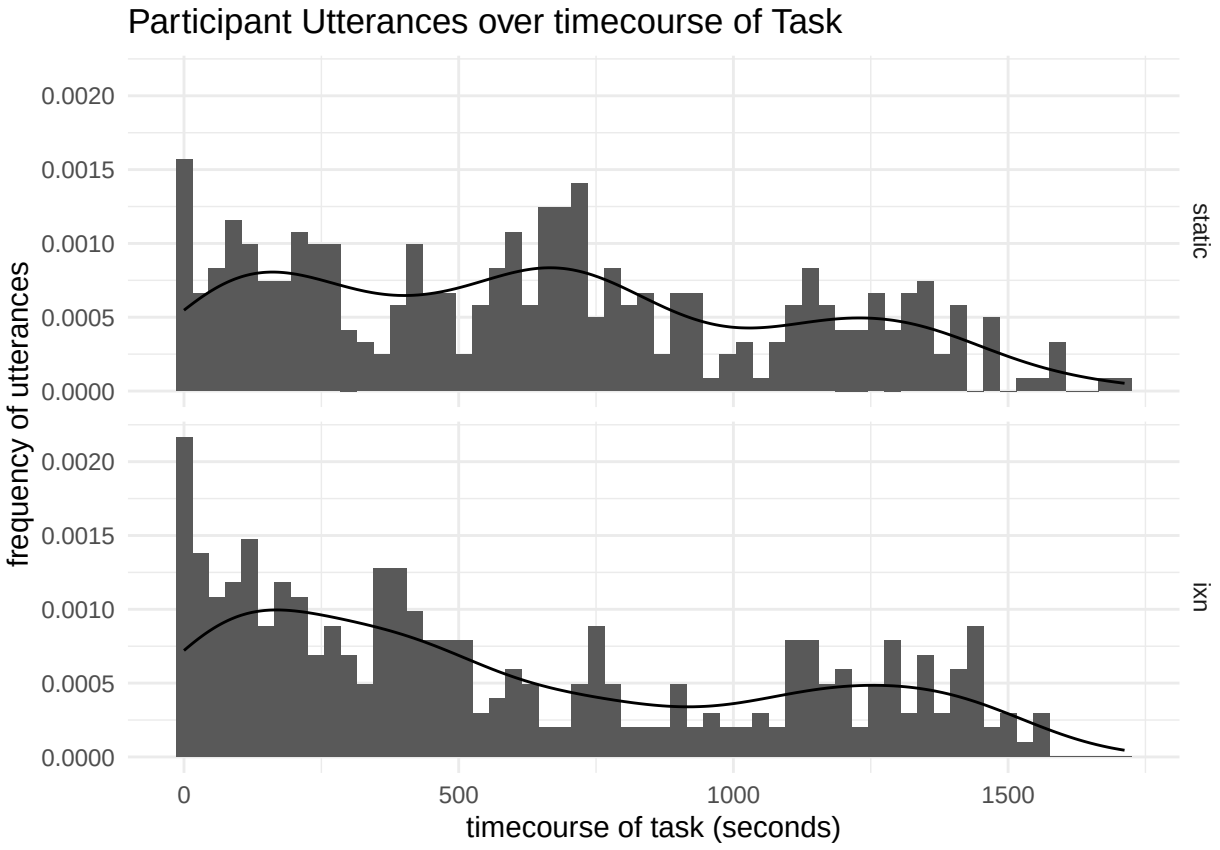
```
#DOTPLOT-PARTICIPANT-facet-TASK-color-DATASET
ggplot(df_coded, aes(x=relative_time, y = PNUM, color = DATASET)) +
```

```
  geom_point(alpha=0.5, size=3) +
  facet_grid(df_coded$TASK) +
  # scale_color_brewer(type="qual", palette = 3) +
  theme_minimal() + labs(
    title = "Participant Utterances over timecourse of Task",
    subtitle = "(does not appear to differ through time by task or dataset) ",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "Dataset"
  )
```




```
#HISTOGRAMS BY TASK
ggplot(df_coded, aes(x = relative_time)) +
  geom_histogram(binwidth = 30,aes(y=..density..)) +
  geom_density()+
  facet_grid(df_coded$TASK) +
  theme_minimal() + labs(
    title = "Participant Utterances over timecourse of Task",
    x= "timecourse of task (seconds)", y = "frequency of utterances",
  ) + theme_minimal() + theme(legend.position = "blank")
```

```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
```



[TOPIC of] Utterances

NEXT we explore the distribution of utterances coded by high level TOPIC, across Analysis Task, Dataset, Participant and Time.

RQ: What kinds of things did participants talk aloud during EDA? Did they progress through any 'topical phases' over the course of the task? Or are topics equally distributed across analysis time?

Answer: *Inspection of frequency tables and visualizations suggests that:*

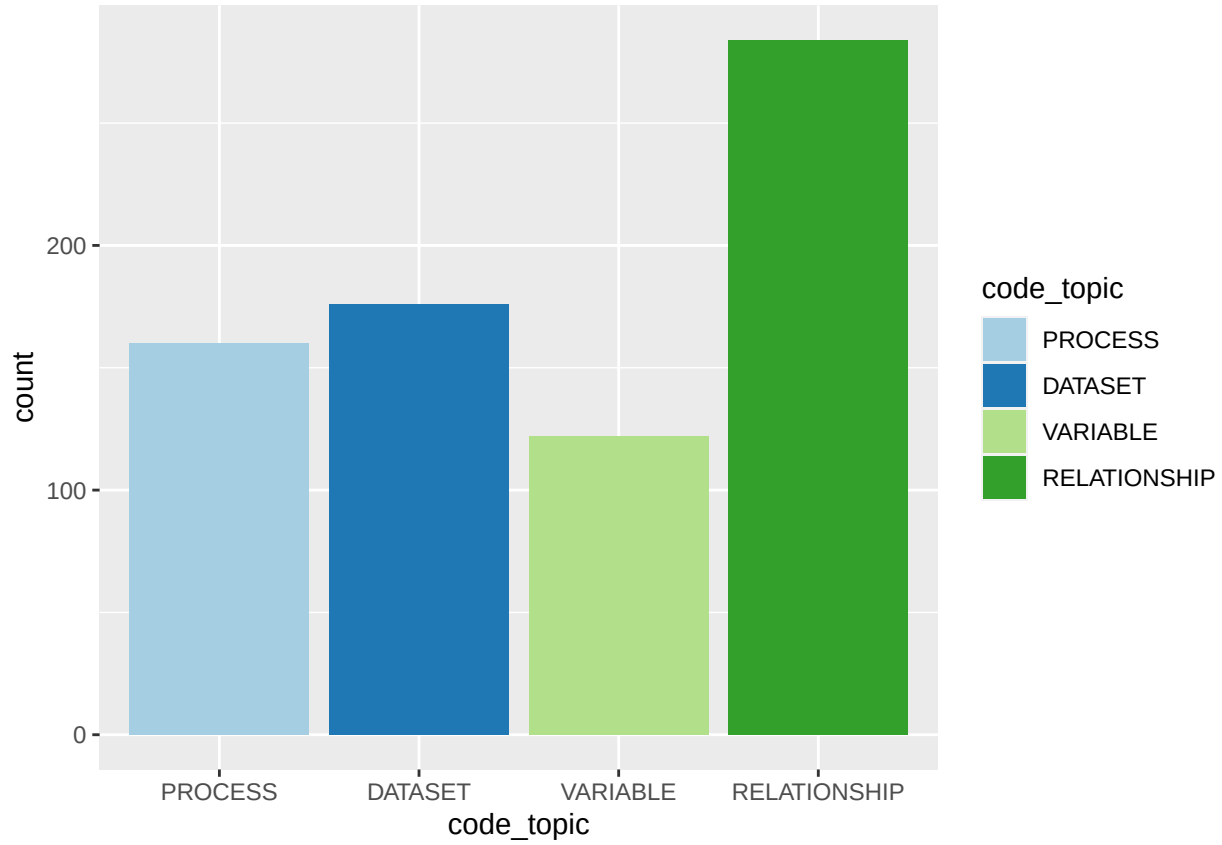
1. *Individual differences continue to play an important role*
2. *There do not appear to be strong TASK/DATASET effects on topic that are consistent across participants.*
3. *PROCESS and RELATIONSHIP topics are more evenly distributed across the timecourse of analysis, while DATASET AND VARIABLE topics are more tightly clustered near the beginning of the analysis. This pattern of distribution is sensical given what we know about EDA, and is consistent with the intuition that patterns of thought during EDA are likely more iterative and situational than we think (or model).*

TOPICS

```
freq(df_coded$code_topic,
     cumul      = FALSE,
     headings   = TRUE,
     report.nas = FALSE,
     plain.ascii = FALSE)
```

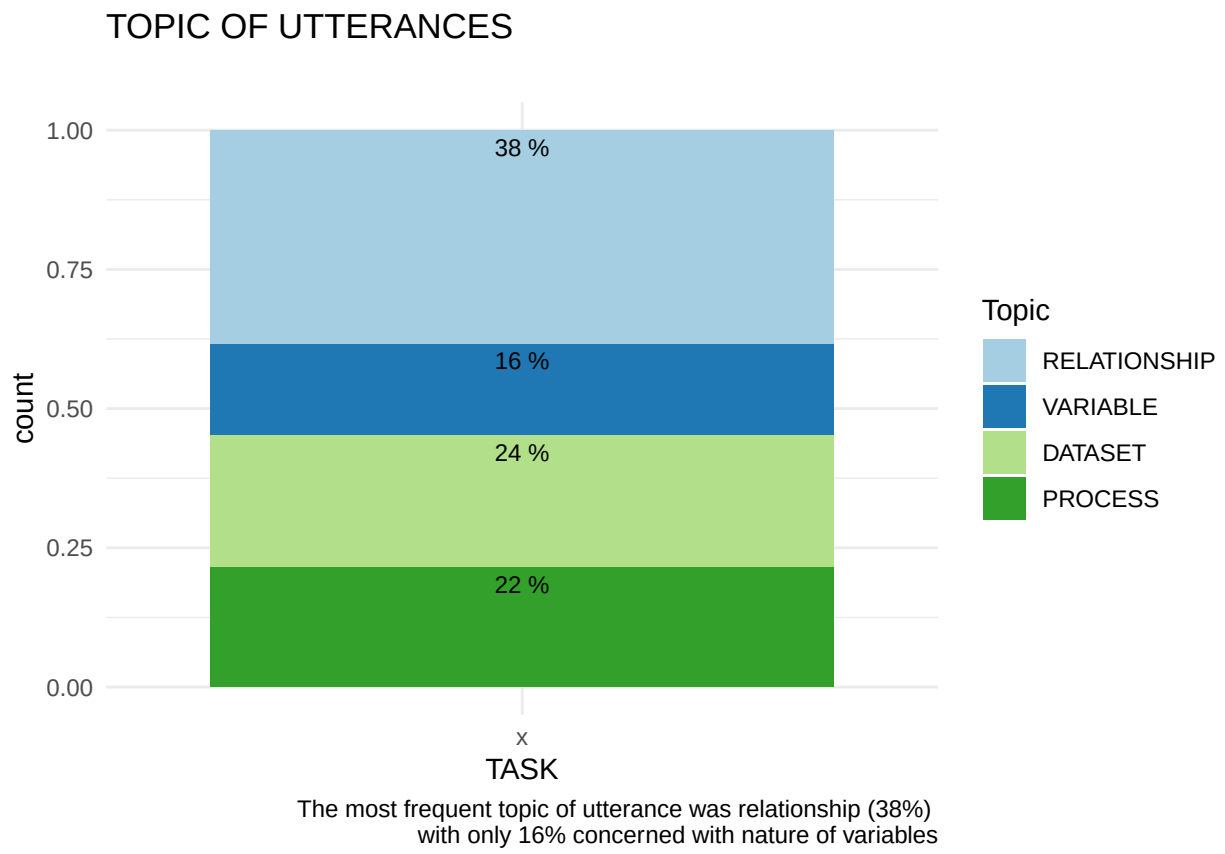
```
## ### Frequencies
## #### df_coded$code_topic
## **Type:** Factor
##
##                  Freq        %
## ------------------- ------ --------
##         **PROCESS**    160    21.56
##         **DATASET**    176    23.72
##        **VARIABLE**    122    16.44
##    **RELATIONSHIP**    284    38.27
##           **Total**    742   100.00
```

```
gf_bar(~code_topic, fill = ~code_topic, position="stack", data = df_coded) +
  scale_fill_brewer(type="qual", palette = 3)
```




```
df_summary <- df_coded %>%
  group_by(code_topic) %>%
  summarise(n = n()) %>%
```

```
  mutate(freq = n / sum(n),
         dumm =  factor("x"))

#TOPIC
ggplot(df_summary, aes(y=freq, x = dumm, fill= fct_rev(code_topic))) +
  geom_col() +
  geom_text(aes(label=paste(round(freq*100,0),"%")), size = 3, hjust = 0.5, vjust = 1.5, position = "sta
  scale_fill_brewer(type="qual", palette = 3) +
  labs( title = "TOPIC OF UTTERANCES",
        subtitle = "",
        caption = "The most frequent topic of utterance was relationship (38%) \n with only 16% concerne
        x= "TASK", y = "count", fill = "Topic") + theme_minimal()
```



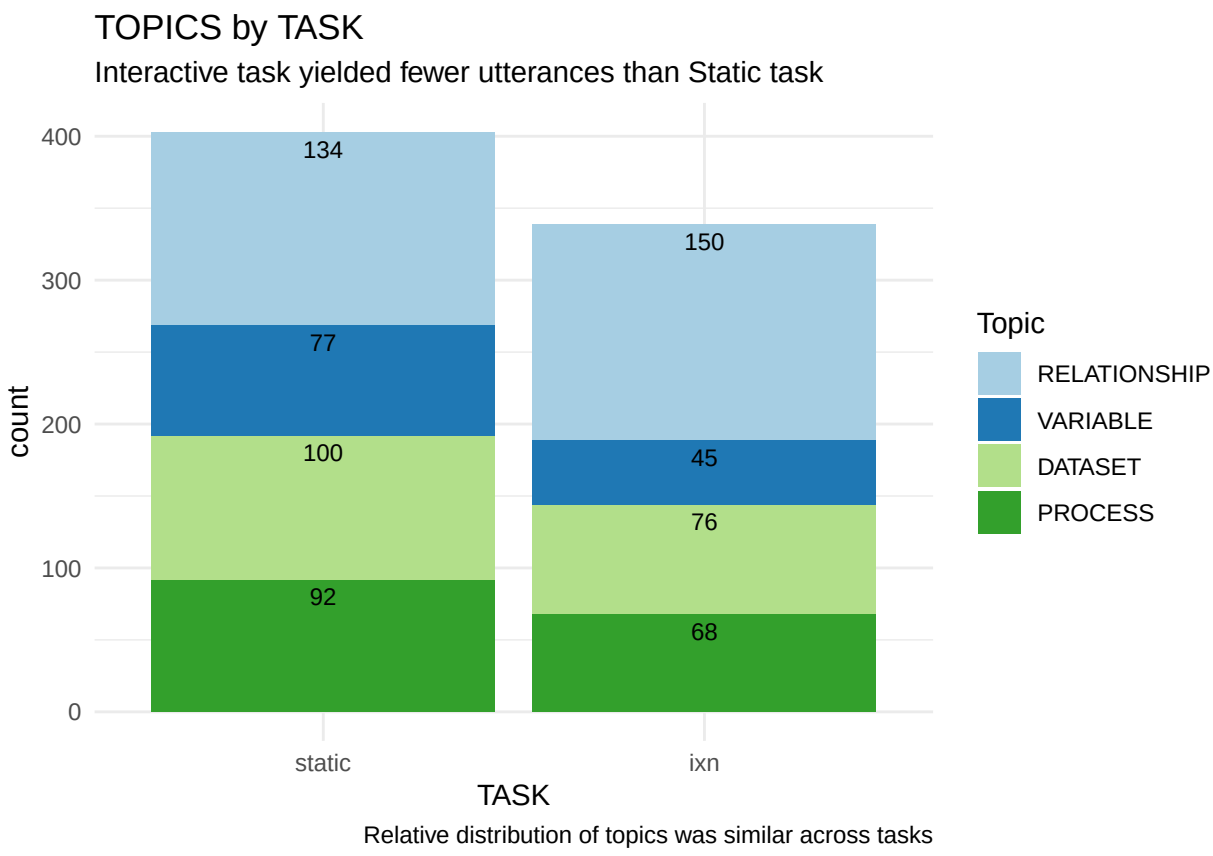
by TASK

```
#COUNT BY TASK
ctable(x = df_coded$code_topic,
       y = df_coded$TASK,
       prop = "r")
```

Cross-Tabulation, Row Proportions
code_topic * TASK
Data Frame: df_coded

TASK	static	ixn	Total
code_topic			
PROCESS	92 (57.5%)	68 (42.5%)	160 (100.0%)
DATASET	100 (56.8%)	76 (43.2%)	176 (100.0%)
VARIABLE	77 (63.1%)	45 (36.9%)	122 (100.0%)
RELATIONSHIP	134 (47.2%)	150 (52.8%)	284 (100.0%)
Total	403 (54.3%)	339 (45.7%)	742 (100.0%)

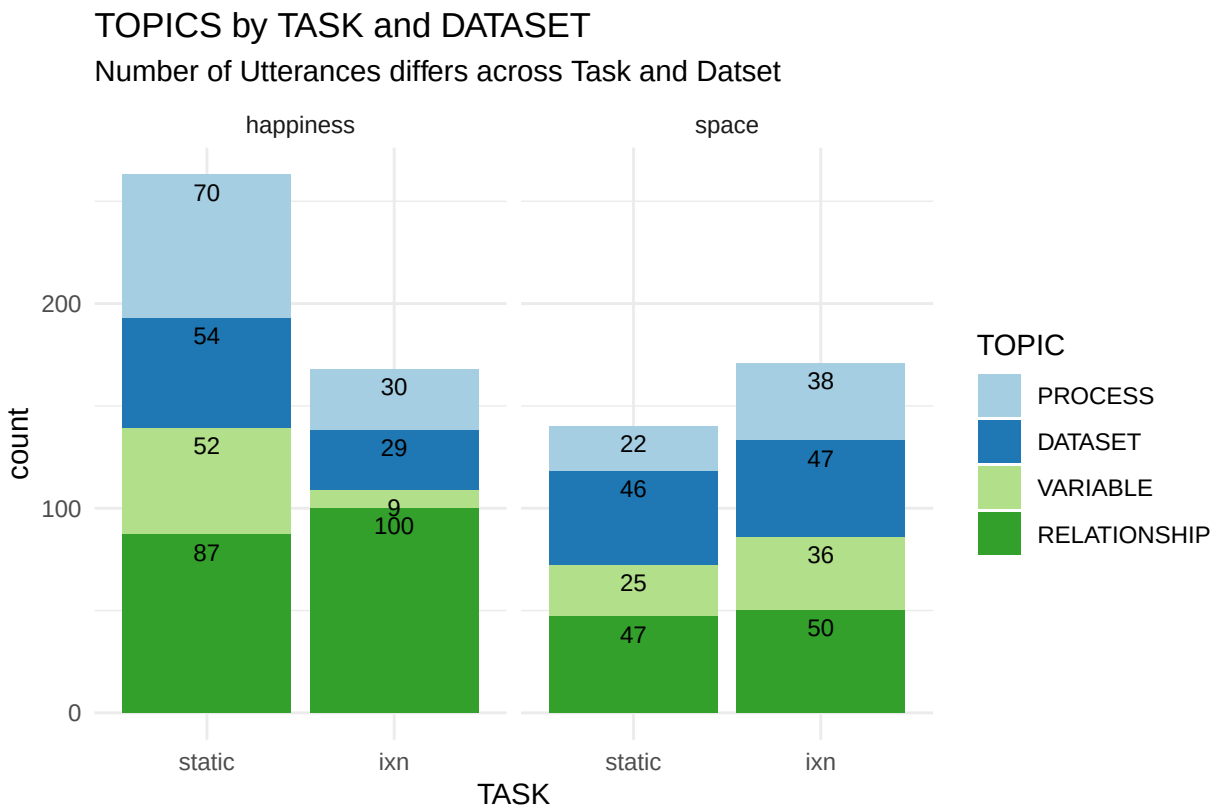
```
#DF SUMMARIZED BY TASK + DATASET
df_summary <- df_coded %>%
  group_by(code_topic, TASK) %>%
  dplyr::summarise(
    c = n()
  )

#STACKED BAR BY TASK
ggplot(df_summary, aes(x = TASK, y=c, fill= fct_rev(code_topic))) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_brewer(type="qual", palette = 3) +
  labs( title = "TOPICS by TASK",
        subtitle = "Interactive task yielded fewer utterances than Static task",
        caption="Relative distribution of topics was similar across tasks",
        x= "TASK", y = "count", fill="Topic") + theme_minimal()
```



```
# + theme(legend.position = "blank")
```

by TASK and DATASET

```
#DF SUMMARIZED BY TASK + DATASET
df_summary <- df_coded %>%
  group_by(code_topic, TASK,DATASET) %>%
  dplyr::summarise(
    c = n()
  )

#PAPER FIGURE HERE
#STACKED BAR BY TASK FACET DATASET
(p <- ggplot(df_summary, aes(x = TASK, y=c, fill= (code_topic))) +
  facet_wrap(df_summary$DATASET) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  # scale_fill_manual(values=c(
  #   "#7FADCB", ##ACA2BA",
  #   "#A2D8D0", ##33A02C", #"#C2E2B7",
  #   "#B2DF8A", #AED8D2",
  #   "#ACA2BA" #AECDE1"
  # ))  +
  scale_fill_brewer(type="qual", palette = 3) +
  labs( title = "TOPICS by TASK and DATASET",
        subtitle = "Number of Utterances differs across Task and Datset",
        caption = "Relative distribution of Topics is similar across factors excepting IXN-Happiness",
        x= "TASK", y = "count", fill="TOPIC") + theme_minimal())
```




```
# + theme(legend.position = "blank")

ggsave(p, file="figures/UTTERANCE_topics_by_factors.png")
```

NOTE: Here we see an interesting pattern in the IXN-HAPPINESS dataset, where there are far fewer utterances about VARIABLES and more utterances about RELATIONSHIPS. Analysts believe this is likely a result of using INTERACTIVE-PROFILE visualization, where participants seemed to skip over characterization of the univariate distributions in favor of observing or speculating about the potential relationships *between* variables while actively scrubbing across one of the profiler histograms.

by PARTICIPANT

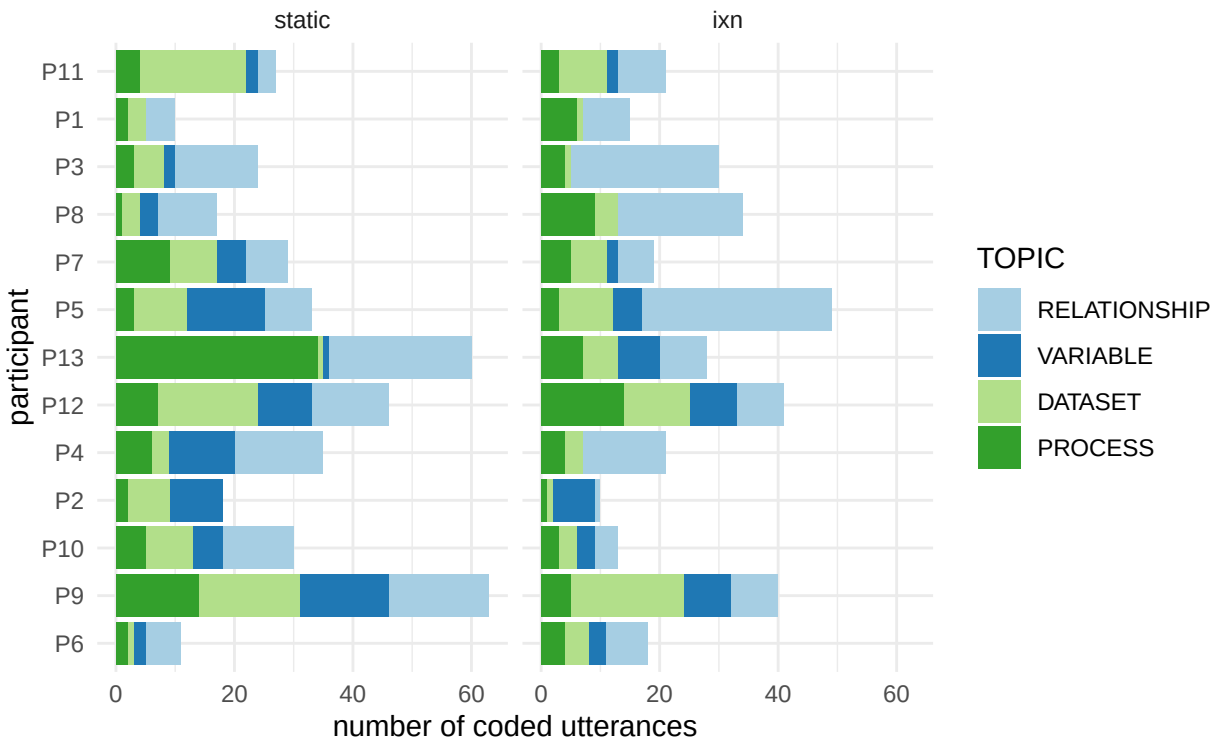
```
#COUNT BY PARTICIPANT
ctable(x = df_coded$PNUM,
       y = df_coded$code_topic,
       prop = "r")
```

Cross-Tabulation, Row Proportions
PNUM * code_topic
Data Frame: df_coded

code_topic	PROCESS	DATASET	VARIABLE	RELATIONSHIP	Total
PNUM					

P6	6 (20.7%)	5 (17.2%)	5 (17.2%)	13 (44.8%)	29 (100.0%)
P9	19 (18.4%)	36 (35.0%)	23 (22.3%)	25 (24.3%)	103 (100.0%)
P10	8 (18.6%)	11 (25.6%)	8 (18.6%)	16 (37.2%)	43 (100.0%)
P2	3 (10.7%)	8 (28.6%)	16 (57.1%)	1 (3.6%)	28 (100.0%)
P4	10 (17.9%)	6 (10.7%)	11 (19.6%)	29 (51.8%)	56 (100.0%)
P12	21 (24.1%)	28 (32.2%)	17 (19.5%)	21 (24.1%)	87 (100.0%)
P13	41 (46.6%)	7 (8.0%)	8 (9.1%)	32 (36.4%)	88 (100.0%)
P5	6 (7.3%)	18 (22.0%)	18 (22.0%)	40 (48.8%)	82 (100.0%)
P7	14 (29.2%)	14 (29.2%)	7 (14.6%)	13 (27.1%)	48 (100.0%)
P8	10 (19.6%)	7 (13.7%)	3 (5.9%)	31 (60.8%)	51 (100.0%)
P3	7 (13.0%)	6 (11.1%)	2 (3.7%)	39 (72.2%)	54 (100.0%)
P1	8 (32.0%)	4 (16.0%)	0 (0.0%)	13 (52.0%)	25 (100.0%)
P11	7 (14.6%)	26 (54.2%)	4 (8.3%)	11 (22.9%)	48 (100.0%)
Total	160 (21.6%)	176 (23.7%)	122 (16.4%)	284 (38.3%)	742 (100.0%)

```
#PAPER FIGURE HERE
#TOPICS by PARTICPANT facet TASK
(p <- gf_bar( PNUM ~., fill = ~ fct_rev(code_topic), data = df_coded) %>%
  gf_facet_grid(.~TASK) +
  scale_fill_brewer(type="qual", palette = 3) +
  labs(
    title = "Utterances by Participant, Dataset and Task",
    subtitle = "",
    x = "number of coded utterances",
    y = "participant",
    fill = "TOPIC"
  ) + theme_minimal())
```

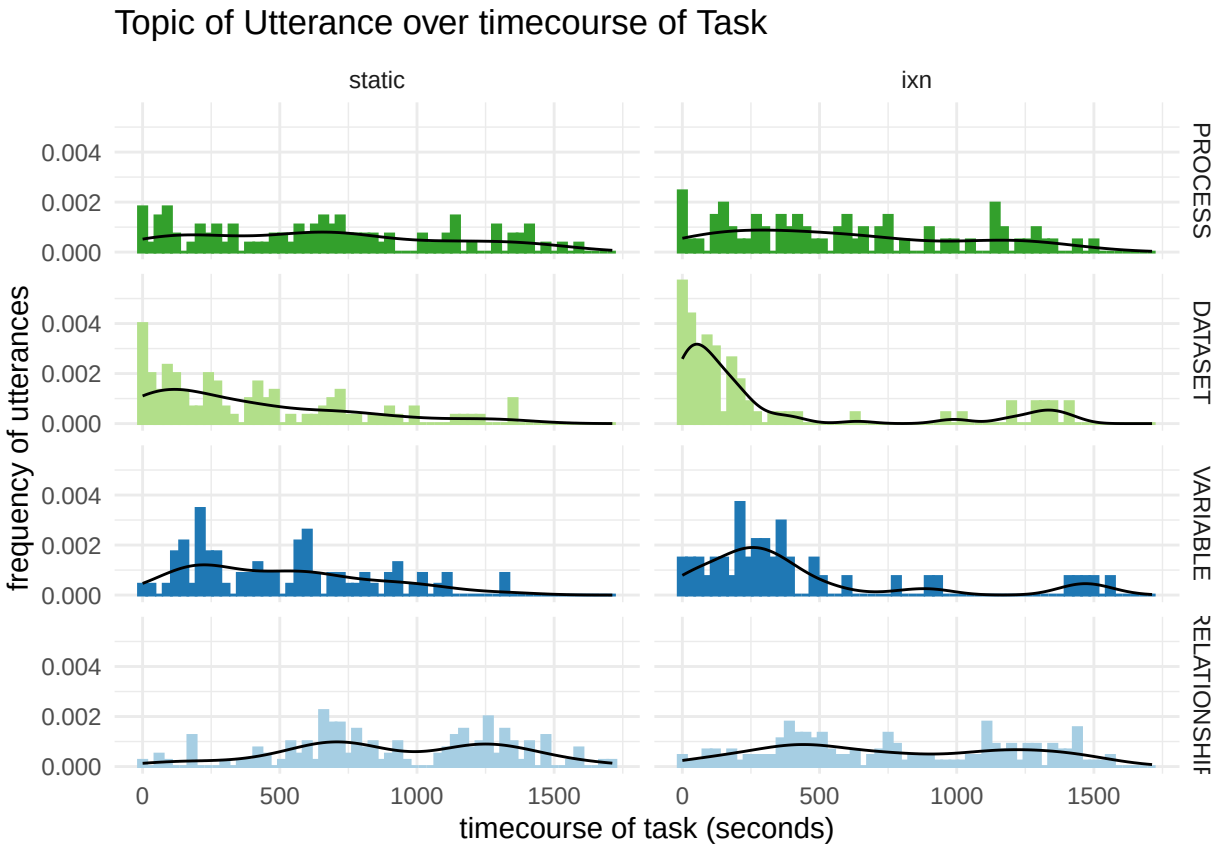



```
ggsave(p, file="figures/UTTERANCE_topics_by_participant.png")



# #TOPICS by PARTICPANT facet TASK
# gf_bar( PNUM ~., fill = ~ fct_rev(code_topic), data = df_coded) %>%
#   gf_facet_grid(DATASET~TASK) +
#   scale_fill_brewer(type="qual", palette = 3) +
#   labs(
#     title = "Utterances by Participant, Dataset and Task",
#     subtitle = "",
#     x = "number of coded utterances",
#     y = "participant",
#     fill = "DATASET"
#   )
```

NOTE: There appear to be substantial individual differences in the number of utterances participants generate, and to some extent to the high level *topic* of the utterances. It is possible, however, this is also related to individuals' relative familiarity/interest in exploring data with numeric vs. nominal outcome variables. Both data analysts noted participants (in general) seemed to struggle more in choosing and interpreting representations aimed at representing relationships with nominal variables.

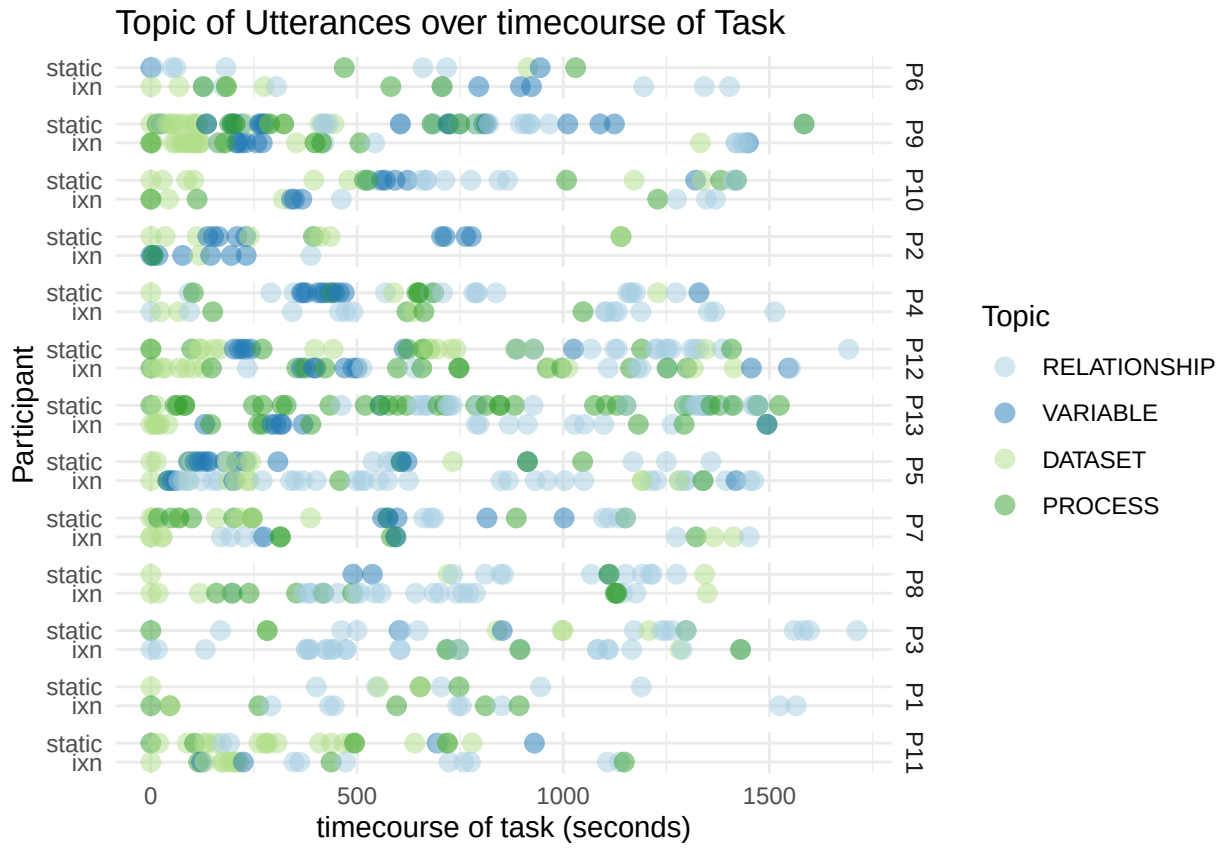
by TIME

```
#HISTOGRAMS BY TASK
ggplot(df_coded, aes(x = relative_time)) +
  geom_histogram(binwidth = 30,aes(y=..density.., fill = fct_rev(code_topic), color = fct_rev(code_topic
  geom_density()+
  facet_grid(df_coded$code_topic ~ df_coded$TASK) +
  scale_fill_brewer(type="qual", palette = 3) +
  scale_color_brewer(type="qual", palette = 3) +
  theme_minimal() + labs(
    title = "Topic of Utterance over timecourse of Task",
    x= "timecourse of task (seconds)", y = "frequency of utterances",
    fill = "Topic"
  ) + theme_minimal() + theme(legend.position = "blank")
```




```
#DOTPLOT - PARTICIPANT FACET
# (p <- ggplot(df_coded, aes(x=relative_time, y = PNUM, color=fct_rev(code_topic))) +
#   geom_point(alpha=0.5, size=3) +
#   facet_grid(df_coded$TASK) +
#   scale_color_brewer(type="qual", palette = 3) +
#   theme_minimal() + labs(
#     title = "Topic of Utterances over timecourse of Task",
#     x= "timecourse of task (seconds)", y = "Task",
#     color = "Topic"
#   ))
```

```
# ggsave(p, file="figures/UTTERANCE_topics_by_time_FACET.png")

#PAPER FIGURE HERE
#DOTPLOT TASK STACKED
(p <- ggplot(df_coded, aes(x=relative_time, y = fct_rev(TASK), color=fct_rev(code_topic))) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df_coded$PNUM) +
  # facet_grid(df_coded$TASK ~ df_coded$DATASET) +
  scale_color_brewer(type="qual", palette = 3) +
  theme_minimal() + labs(
    title = "Topic of Utterances over timecourse of Task",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "Topic"
))
```




```
ggsave(p, file="figures/UTTERANCE_topics_by_time_STACK.png")
```

[DETAIL of] Utterances

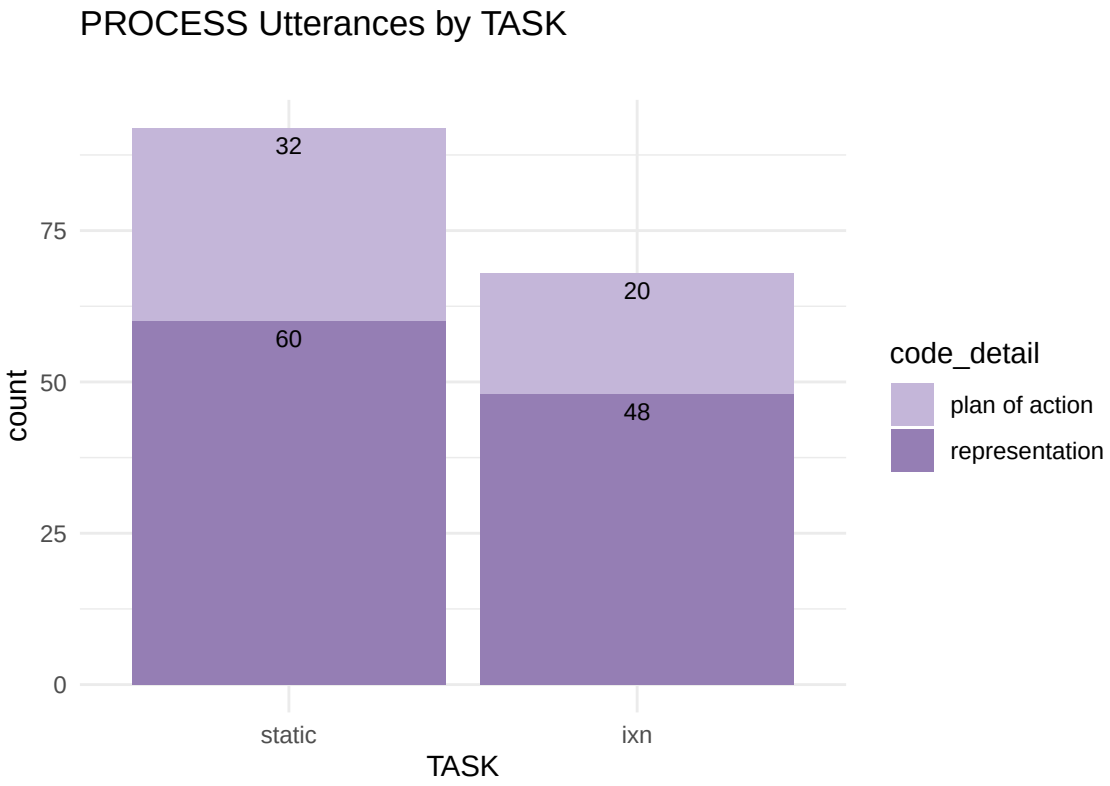
NEXT we explore the distribution of specific detail utterances across Analysis Task, Dataset, Participant and Time.

We organize these sections according to the high-level topic codes: (ANALYSIS) PROCESS, DATASET, VARIABLE, RELATIONSHIP

***RQ: What specific things did participants talk aloud during EDA? Are there any details folks only* mention during static(v)interactive, or nominal(v)numeric tasks? Any substantial changes in proportion by TASK or DATASET?**

PROCESS UTTERANCES

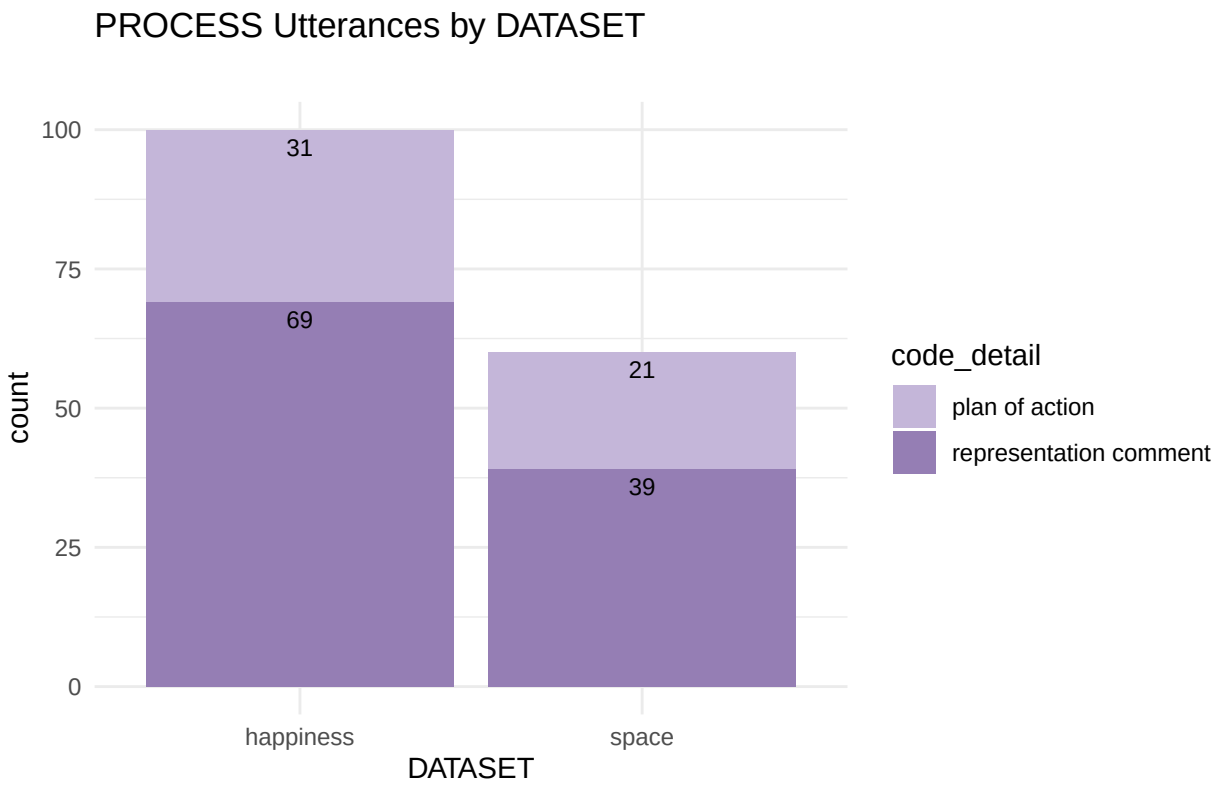
```
#PREP DATA FRAMES
df_process <- df_coded %>%
  filter(code_topic=="PROCESS") %>%
  dplyr::select(pid,PNUM,TASK,DATASET,code_detail)

df_time_process <- df_coded %>%
  filter(code_topic=="PROCESS") %>%
  dplyr::select(pid,PNUM,TASK,DATASET,relative_time,code_detail)

df_summary_task <- df_process %>%
  group_by(code_detail, TASK) %>%
  dplyr::summarise(c = n())

df_summary_dataset <- df_process %>%
  group_by(code_detail, DATASET) %>%
  dplyr::summarise(c = n())


#DETAILS BY TASK
ggplot(df_summary_task, aes(x = TASK, y=c, fill= code_detail)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_manual(values = c("#C4B6D9","#957EB4"))+
  # scale_fill_brewer(type="seq", palette = "PuRd") +
  labs( title = "PROCESS Utterances by TASK",
        subtitle = "",
        caption = "weak to moderate difference in PROCESS utterances by TASK, \n but these do not seem s
        x= "TASK", y = "count") + theme_minimal()
```



weak to moderate difference in PROCESS utterances by TASK,
but these do not seem substantial when broken into the two categories

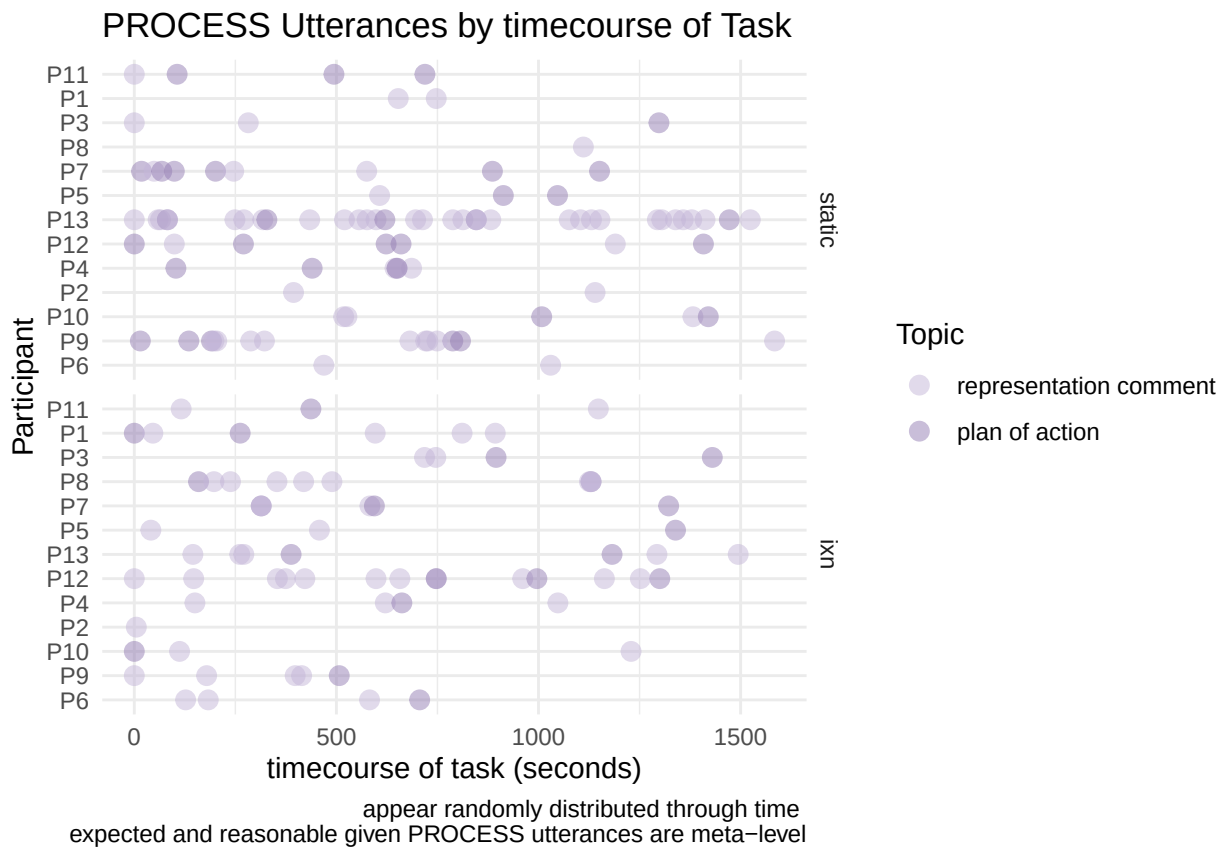
PROCESS Utterances

```
#DETAILS BY DATASET
ggplot(df_summary_dataset, aes(x = DATASET, y=c, fill= code_detail)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_manual(values = c("#C4B6D9","#957EB4"))+
  # scale_fill_brewer(type="seq", palette = "PuRd") +
  labs( title = "PROCESS Utterances by DATASET",
        subtitle = "",
        caption = "much more substantial differences in PROCESS utterances by DATASET, \n consistent wi
        x= "DATASET", y = "count") + theme_minimal()
```



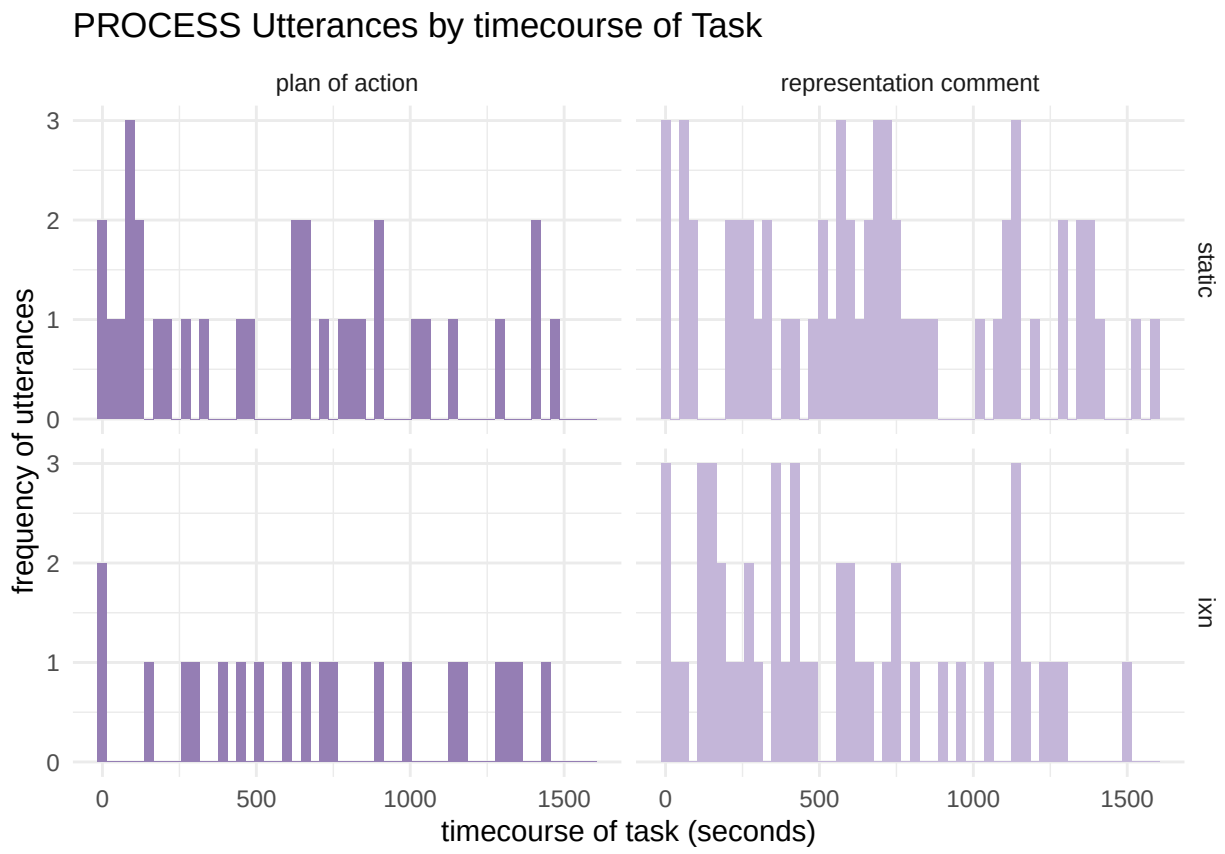
much more substantial differences in PROCESS utterances by DATASET, with intution Ps had more to say about the numeric (vs) nominal outcome variable

```
#DETAILS DOTPLOT
ggplot(df_time_process, aes(x=relative_time, y = PNUM, color=fct_rev(code_detail))) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df_time_process$TASK) +
  scale_color_manual(values = c("#C4B6D9","#957EB4"))+
  # scale_color_brewer(type="seq", palette = "PuRd") +
  theme_minimal() + labs(
    title = "PROCESS Utterances by timecourse of Task",
    caption = "appear randomly distributed through time \n expected and reasonable given PROCESS uttera
    x= "timecourse of task (seconds)", y = "Participant",
    color = "Topic"
  )
```

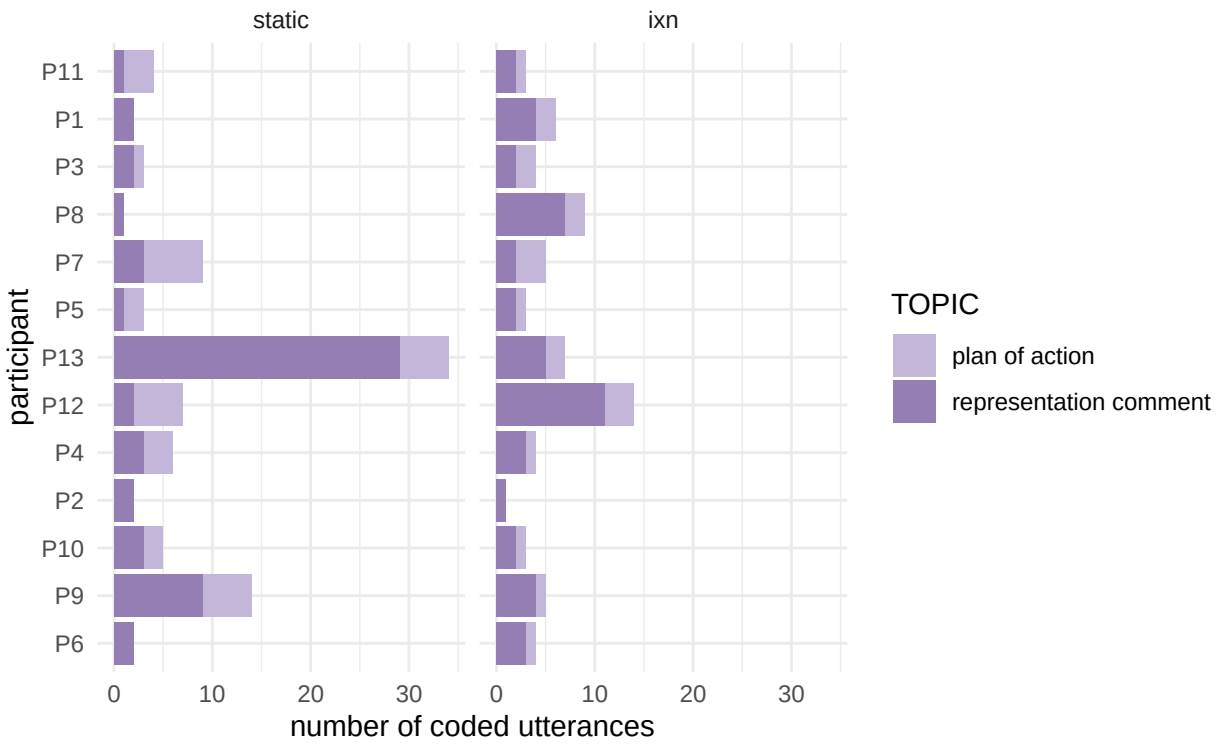


appear randomly distributed through time
expected and reasonable given PROCESS utterances are meta−level

```
#DETAIL HISTOGRAMS BY TASK
ggplot(df_time_process, aes(x = relative_time, fill = fct_rev(code_detail))) +
  geom_histogram(binwidth = 30) +
  facet_grid(df_time_process$TASK ~ df_time_process$code_detail ) +
  # scale_fill_brewer(type="seq", palette = "PuRd") +
  scale_fill_manual(values = c("#C4B6D9","#957EB4"))+
  theme_minimal() + labs(
    title = "PROCESS Utterances by timecourse of Task",
    x= "timecourse of task (seconds)", y = "frequency of utterances"
  ) + theme_minimal() + theme(legend.position = "blank")
```




```
#PAPER FIGURE HERE
#PROCESSES by PARTICPANT facet TASK
(p <- gf_bar( PNUM ~., fill = ~ (code_detail), data = df_process) %>%
  gf_facet_grid(.~TASK) +
  scale_fill_manual(values = c("#C4B6D9","#957EB4"))+
  # scale_fill_brewer(palette = "PRGn", direction=-1) +
  # scale_fill_brewer(type="seq", palette = "PuRd") +
  labs(
    title = "PROCESS Utterances by Participant and Task",
    subtitle = "",
    # caption = "TODO explore P13 representation comments",
    x = "number of coded utterances",
    y = "participant",
    fill = "TOPIC"
  ) + theme_minimal()
)
```



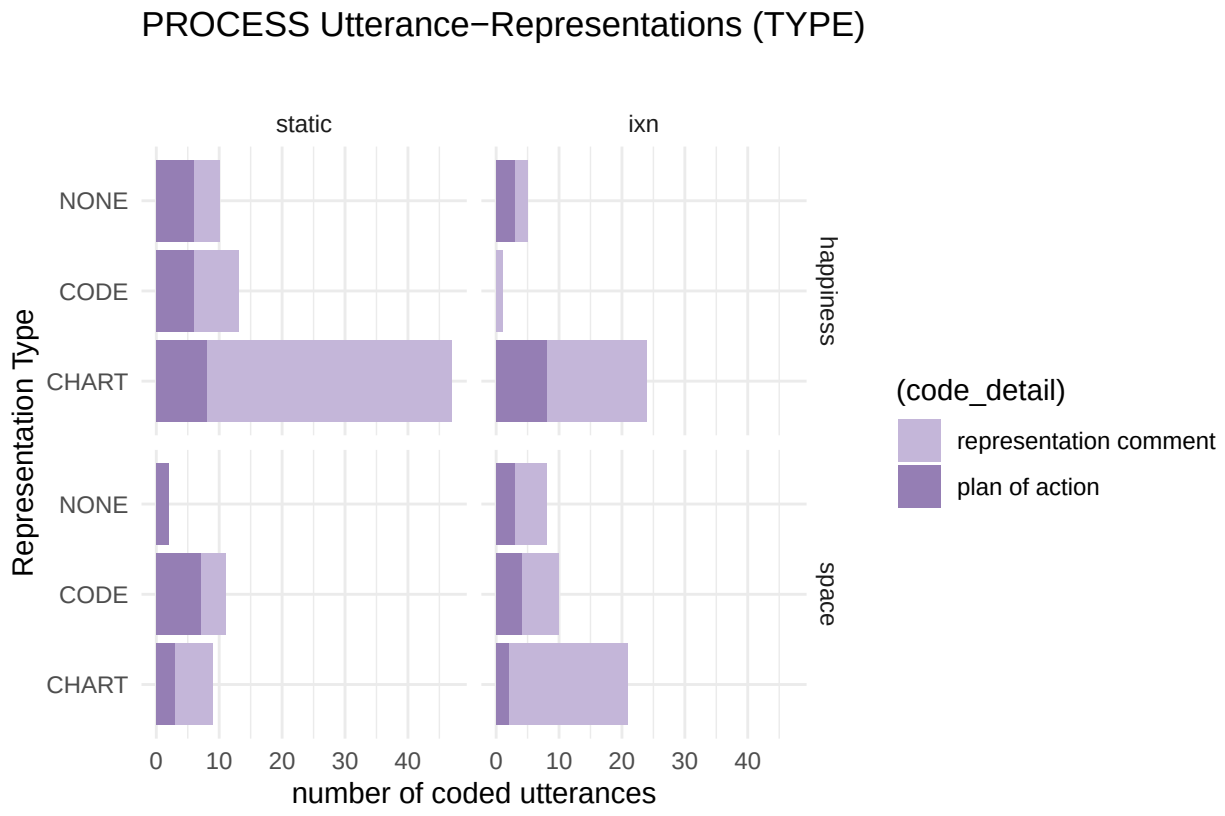

```
ggsave(p, file="figures/UTTERANCE_detail_PROCESS_participants.png", width=6, height=4)
```

NOTE: Across TASKS and DATASET participants had more to say about Representations than their *Plans of Action*. Both kinds of PROCESS utterances were more or less evenly distributed across the timecourse of the tasks.

PROCESS Representations THIS SECTION covers representations EXPLICITLY LINKED to UTTERANCES. Does not include ALL representations generated, but rather, what representations were being used when the participant generated utterances.

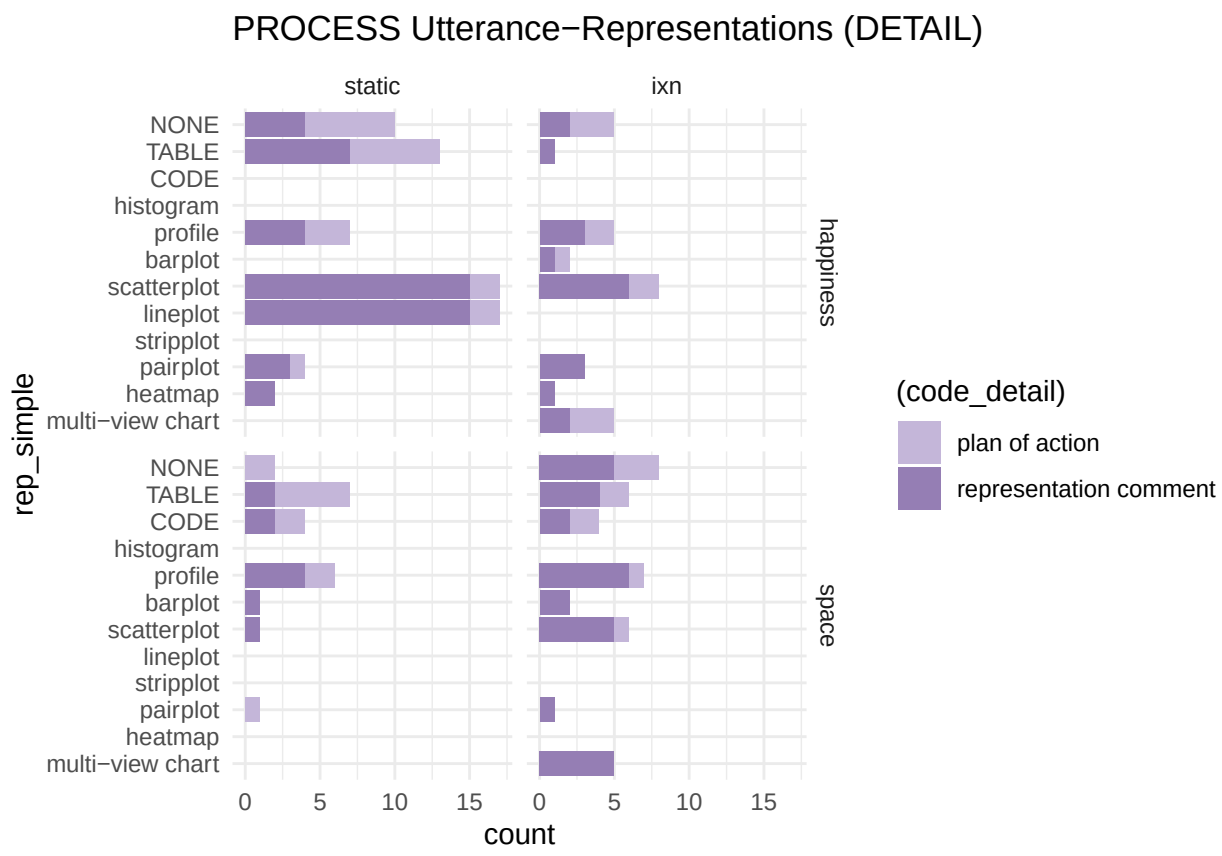
```
#FILTER JOINED DATAFRAME
df <- df_codedrep %>% filter(code_topic =="PROCESS")


#DETAIL REP TYPE
(p <- gf_bar( rep_type ~., fill = ~ fct_rev(code_detail), data = df) %>%
  gf_facet_grid(DATASET~TASK) +
  scale_fill_manual(values = c("#C4B6D9","#957EB4"))+
  # scale_fill_brewer(type="seq", palette = "PuRd") +
  labs(
    title = "PROCESS Utterance-Representations (TYPE) ",
    subtitle = "",
    caption = "",
    x = "number of coded utterances",
    y = "Representation Type",
```

```
    fill = "(code_detail)"
  ) + theme_minimal()
)
```

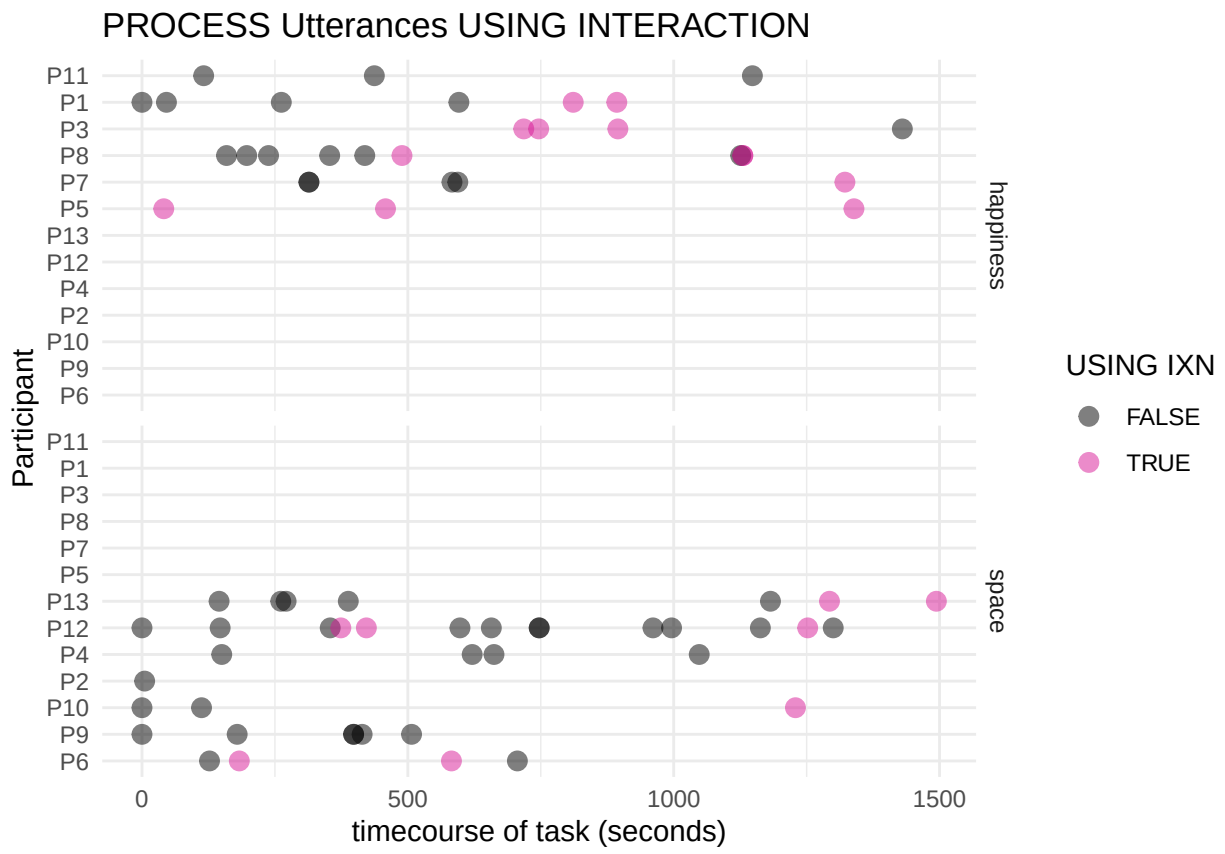



```
#PAPER FIGURE
#DETAIL REP DETAIL
(p <- gf_bar(  ~ rep_simple, fill = ~(code_detail), data = df) %>%
  gf_facet_grid( DATASET ~ TASK) +
  # scale_fill_brewer(type="qual", palette = 1, direction = -1) +
  scale_fill_manual(values = c("#C4B6D9","#957EB4"))+
  coord_flip() +
  scale_x_discrete(limits = c(
            "multi-view chart",
            "heatmap",
            "pairplot",
            "stripplot",
            "lineplot",
            "scatterplot",
            "barplot",
            "profile",
            "histogram",
            "CODE",
            "TABLE",
            "NONE"
  ))+
```

```
  theme_minimal() + labs(
    title = "PROCESS Utterance-Representations (DETAIL)"
))
```




```
ggsave(p, file="figures/UTTERANCE-REP_detail_PROCESS_factors.png", width=6, height=4)


#FILTER ON ONLY ACTIVELY USING IXN
df <- df_codedrep %>% filter(code_topic == "PROCESS") %>% filter(TASK=="ixn")

#PAPER FIGURE
#DOTPLOT-IXN
( p <- ggplot(df, aes(x=relative_time, y = PNUM, color = ixn)) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df$DATASET) +
  # scale_color_brewer(type="qual", palette = 3) +
  scale_color_manual(values=c("black","#D81897")) +
  theme_minimal() + labs(
    title = "PROCESS Utterances USING INTERACTION",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "USING IXN"
  ))
```

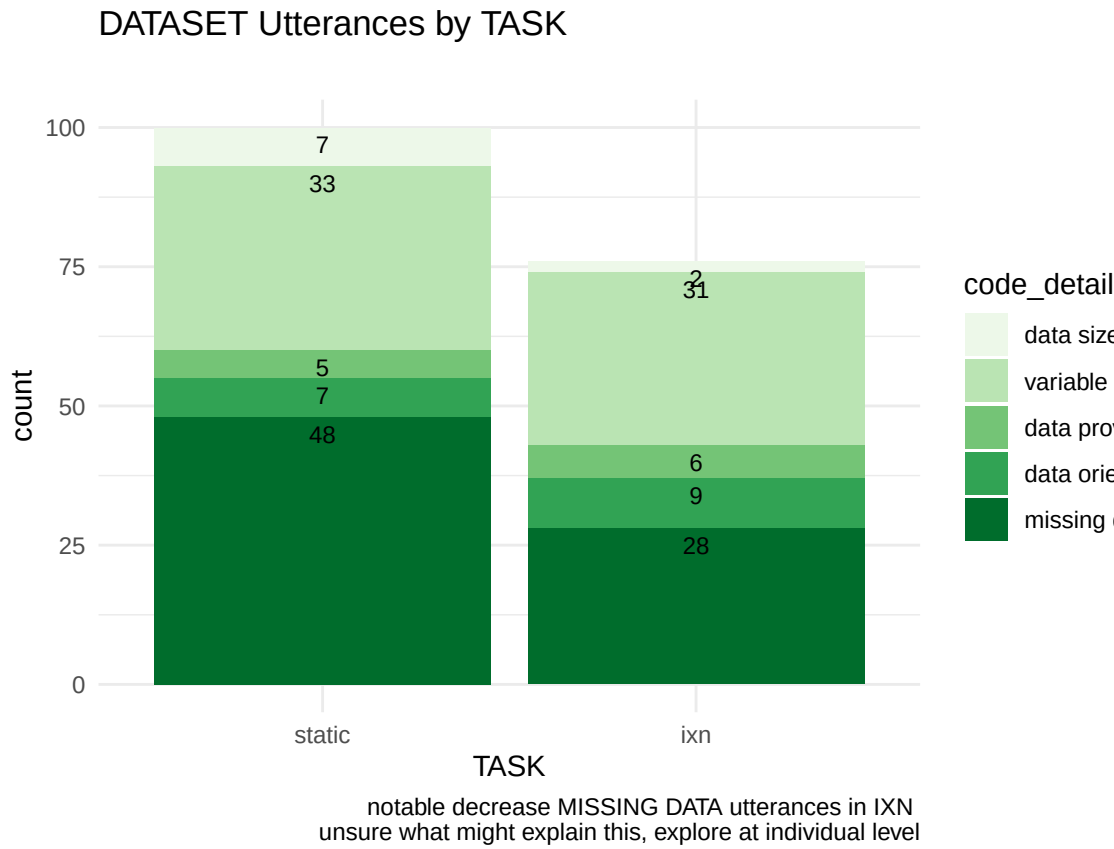



```
ggsave(p, file="figures/IXN_PROCESS_time.png", width=6, height=4)
```

DATASET UTTERANCES

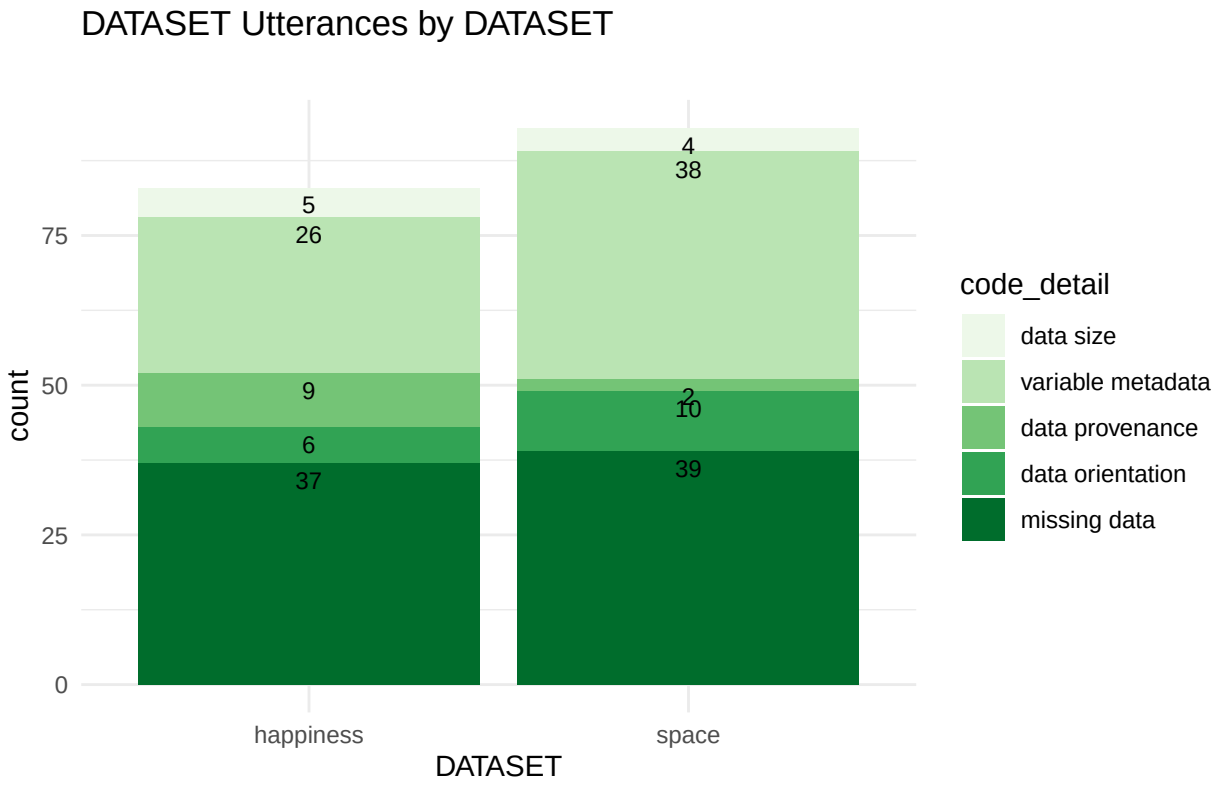
```
#PREP DATA FRAMES
df_dataset <- df_coded %>%
  filter(code_topic=="DATASET") %>%
  dplyr::select(pid,PNUM,TASK,DATASET,code_detail)

df_time_dataset <- df_coded %>%
  filter(code_topic=="DATASET") %>%
  dplyr::select(pid,PNUM,TASK,DATASET,relative_time,code_detail)

df_summary_task <- df_dataset %>%
  group_by(code_detail, TASK) %>%
  dplyr::summarise(c = n())

df_summary_dataset <- df_dataset %>%
  group_by(code_detail, DATASET) %>%
  dplyr::summarise(c = n())


#DETAILS BY TASK
```

```
ggplot(df_summary_task, aes(x = TASK, y=c, fill= code_detail)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_brewer(type="seq", palette = "Greens") +
  # scale_fill_brewer(type="seq", palette = 4) +
  labs( title = "DATASET Utterances by TASK",
        subtitle = "",
        caption = "notable decrease MISSING DATA utterances in IXN \n unsure what might explain this, e
        x= "TASK", y = "count") + theme_minimal()
```

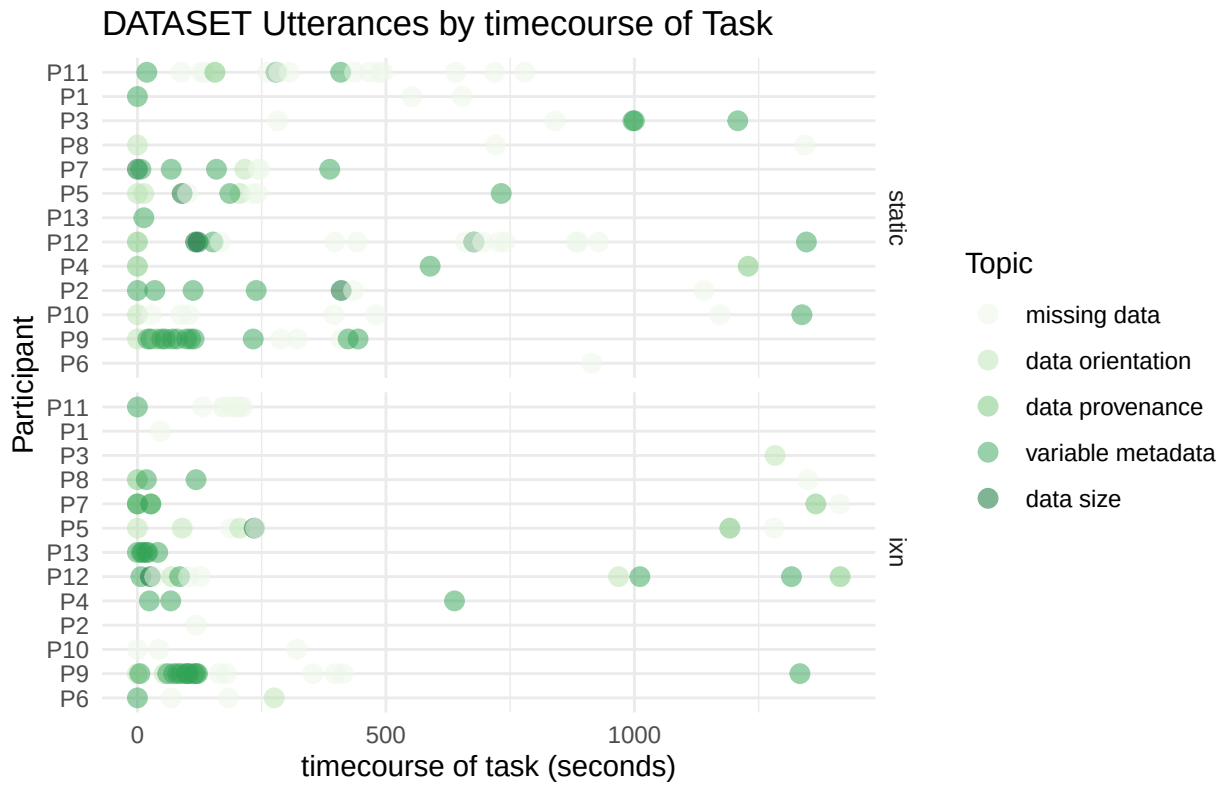


DATASET Utterances

```
#DETAILS BY DATASET
ggplot(df_summary_dataset, aes(x = DATASET, y=c, fill= code_detail)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_brewer(type="seq", palette = "Greens") +
  # scale_fill_brewer(type="seq", palette = 4) +
  labs( title = "DATASET Utterances by DATASET",
        subtitle = "",
        caption = "minor differences by DATASET  reasonable given DATASET \n  represenations are typical
        x= "DATASET", y = "count") + theme_minimal()
```

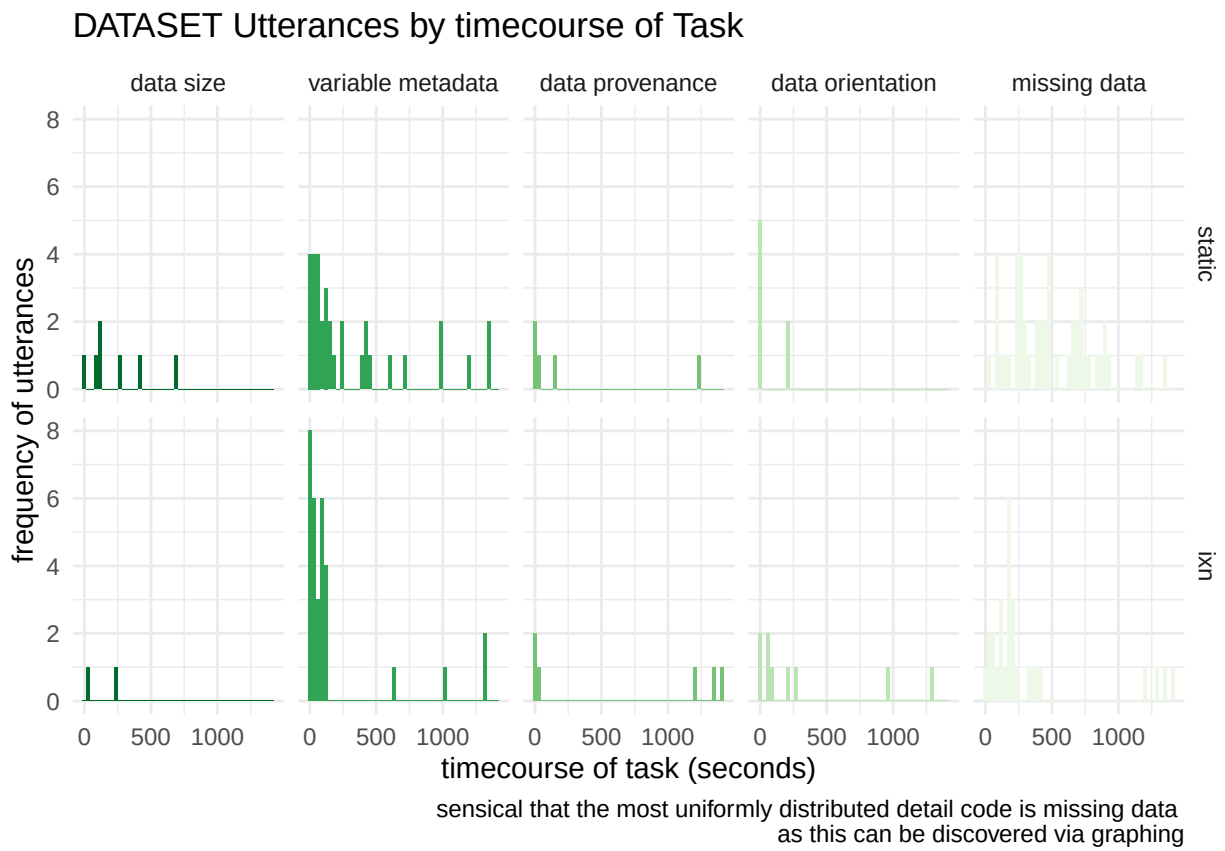



```
#DETAILS DOTPLOT
ggplot(df_time_dataset, aes(x=relative_time, y = PNUM, color=fct_rev(code_detail))) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df_time_dataset$TASK) +
  scale_color_brewer(type="seq", palette = "Greens") +
  # scale_color_brewer(type="seq", palette = 4) +
  theme_minimal() + labs(
    title = "DATASET Utterances by timecourse of Task",
    x= "timecourse of task (seconds)", y = "Participant",
    caption = "notable sparsity in center of timecourse, reasonable as EDA normative behavior \n is to c
    color = "Topic"
  )
```

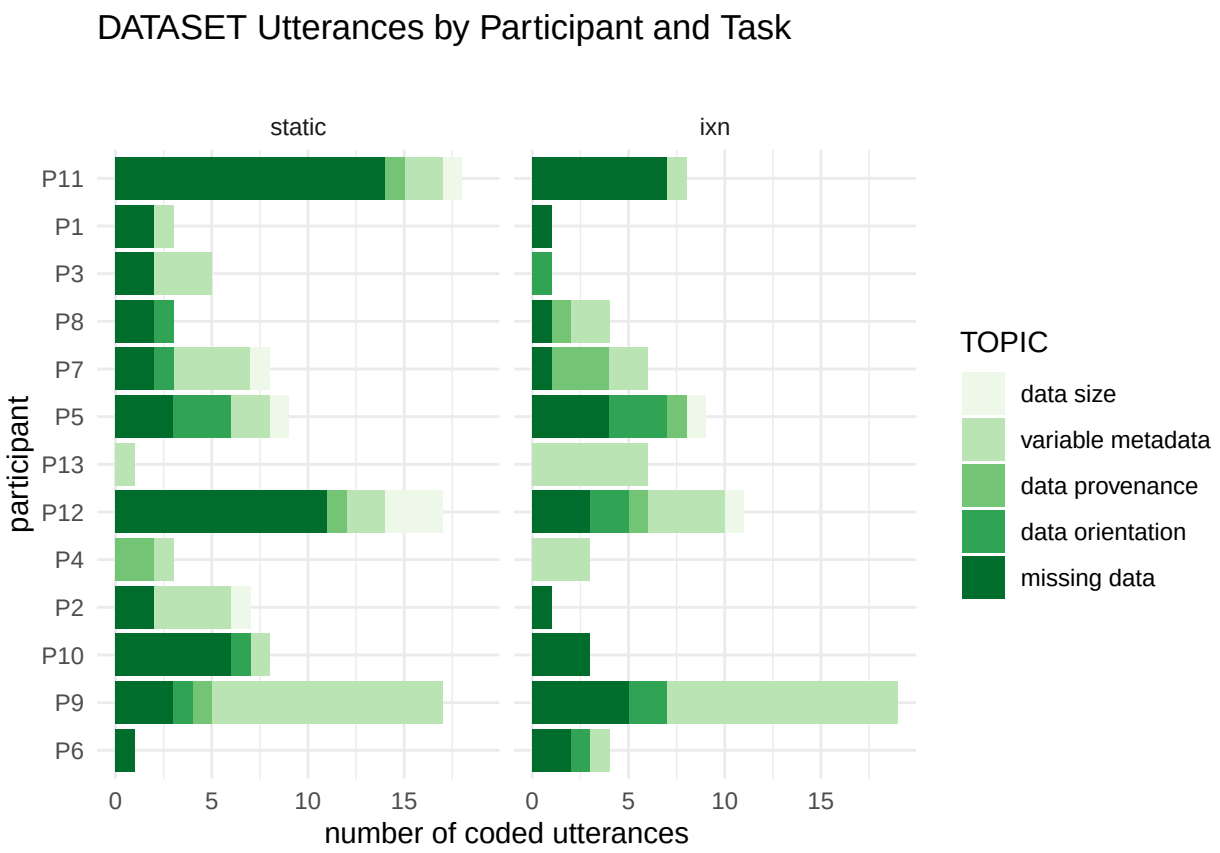


notable sparsity in center of timecourse, reasonable as EDA normative behavior
is to consider dataframe shape and missing data at the start of an analysis

```
#DETAIL HISTOGRAMS BY TASK
ggplot(df_time_dataset, aes(x = relative_time, fill = fct_rev(code_detail))) +
  geom_histogram(binwidth = 30) +
  facet_grid(df_time_dataset$TASK ~ df_time_dataset$code_detail ) +
  scale_fill_brewer(type="seq", palette = "Greens") +
  # scale_fill_brewer(type="seq", palette = 4) +
  theme_minimal() + labs(
    title = "DATASET Utterances by timecourse of Task",
    x= "timecourse of task (seconds)", y = "frequency of utterances",
        caption = "sensical that the most uniformly distributed detail code is missing data \n as this c
  ) + theme_minimal() + theme(legend.position = "blank")
```



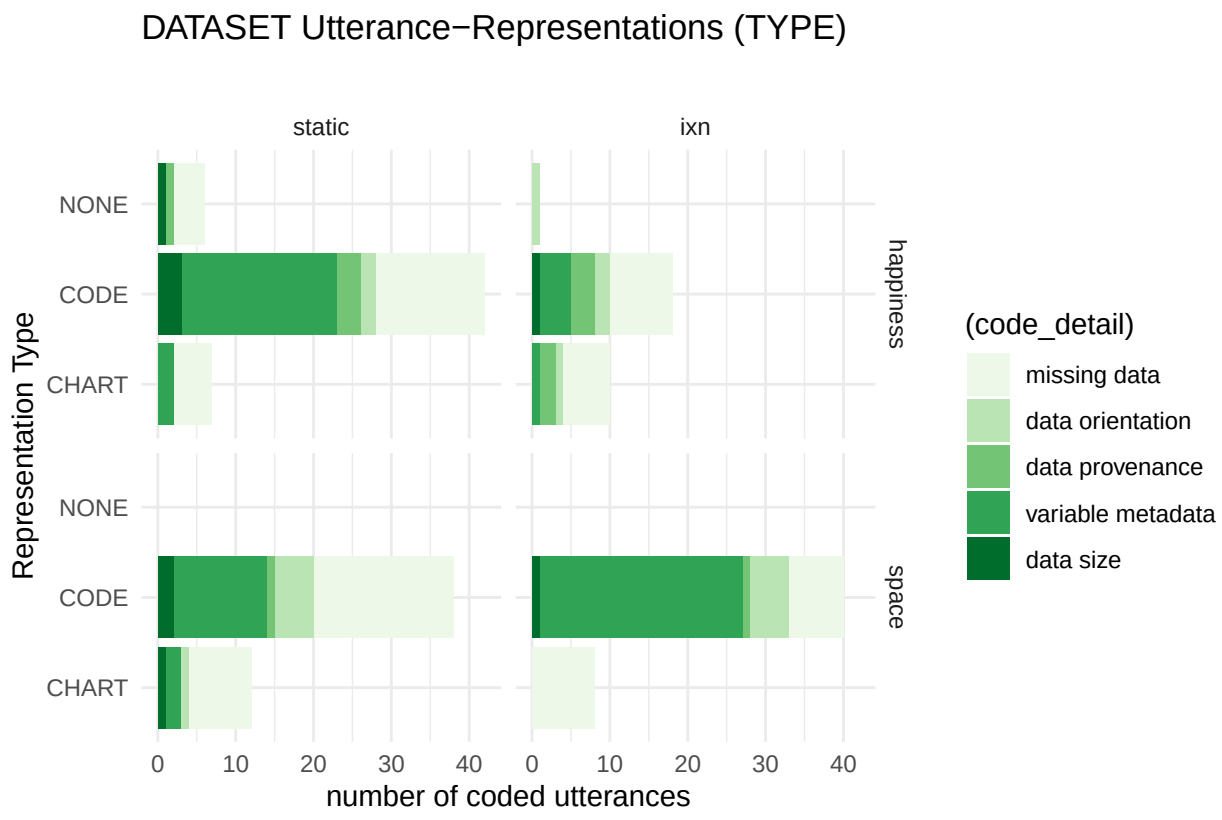

```
#PAPER FIGURE HERE
#DATASET UTTERANCES by PARTICPANT facet TASK
(p <- gf_bar( PNUM ~., fill = ~ (code_detail), data = df_dataset) %>%
  gf_facet_grid(.~TASK) +
  scale_fill_brewer(type="seq", palette = "Greens") +
  labs(
    title = "DATASET Utterances by Participant and Task",
    subtitle = "",
    x = "number of coded utterances",
    y = "participant",
    fill = "TOPIC",
    # caption = "P11, P12 contribute to MISSING DATA \n P9 contributes largely to variable metadata \n
  ) + theme_minimal()
)
```



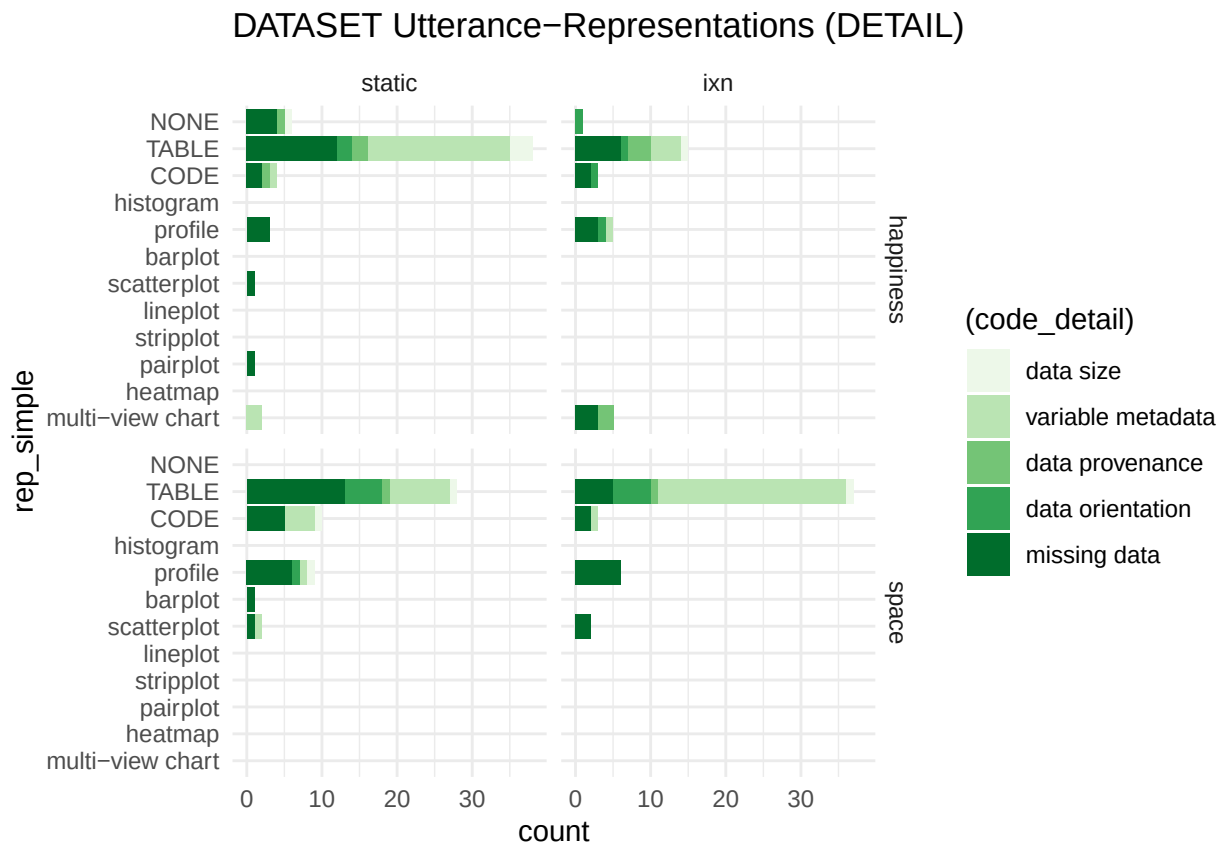

```
ggsave(p, file="figures/UTTERANCE_detail_DATASET_participants.png",width=6, height=4)
```

DATASET Representations THIS SECTION covers representations EXPLICITLY LINKED to UTTERANCES. Does not include ALL representations generated, but rather, what representations were being used when the participant generated utterances.

```
df <- df_codedrep %>% filter(code_topic =="DATASET")

#DETAIL REP TYPE
(p <- gf_bar( rep_type ~., fill = ~ fct_rev(code_detail), data = df) %>%
  gf_facet_grid(DATASET~TASK) +
  scale_fill_brewer(type="seq", palette = "Greens") +
  # scale_fill_brewer(type="seq", palette = "PuRd") +
  labs(
    title = "DATASET Utterance-Representations (TYPE) ",
    subtitle = "",
    caption = "",
    x = "number of coded utterances",
    y = "Representation Type",
    fill = "(code_detail)"
  ) + theme_minimal()
)
```

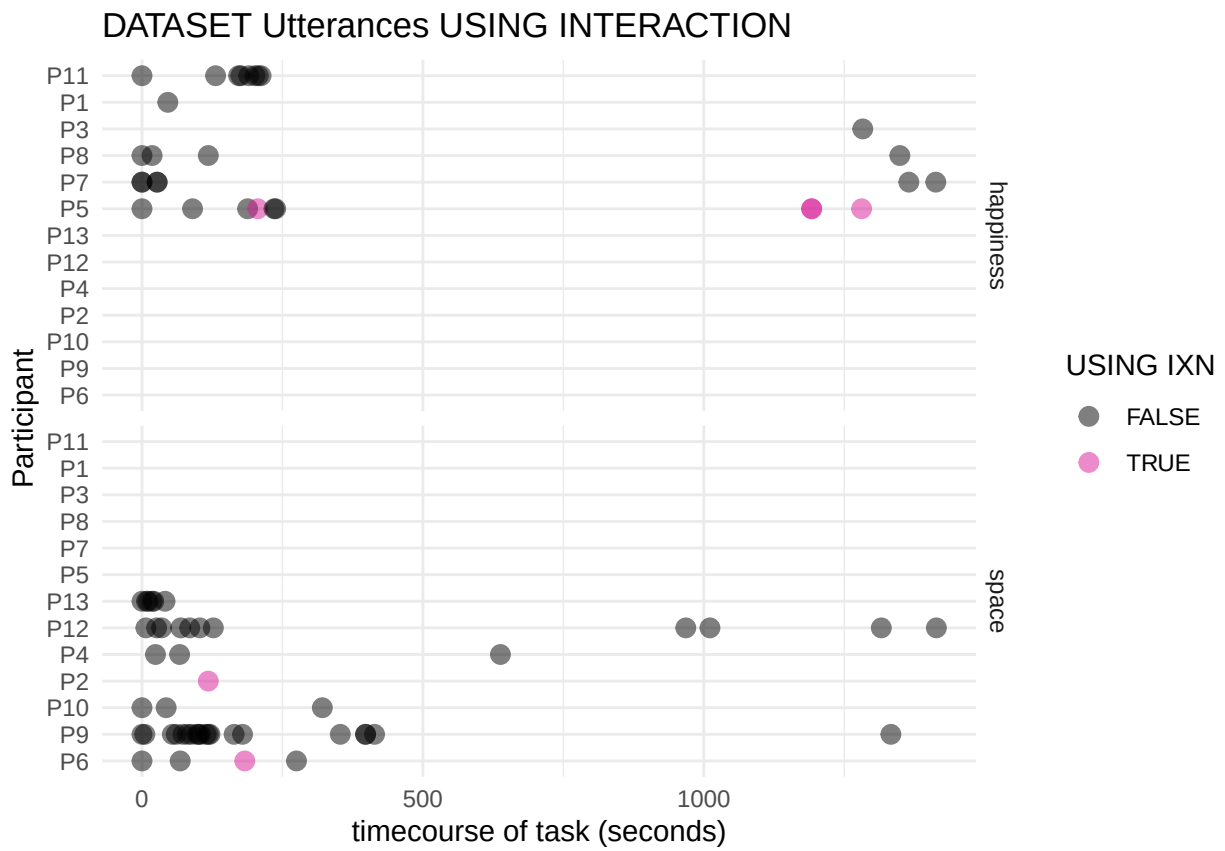



```
#PAPER FIGURE
(p <- gf_bar(  ~ rep_simple, fill = ~(code_detail), data = df) %>%
  gf_facet_grid( DATASET ~ TASK) +
  scale_fill_brewer(type="seq", palette = "Greens") +
  coord_flip() +
  scale_x_discrete(limits = c(
            "multi-view chart",
            "heatmap",
            "pairplot",
            "stripplot",
            "lineplot",
            "scatterplot",
            "barplot",
            "profile",
            "histogram",
            "CODE",
            "TABLE",
            "NONE"
  ))+
  theme_minimal() + labs(
    title = "DATASET Utterance-Representations (DETAIL)"
))
```




```
ggsave(p, file="figures/UTTERANCE-REP_detail_DATASET_factors.png", width=6, height=4)
```

```
df <- df_codedrep %>% filter(code_topic == "DATASET") %>% filter(TASK=="ixn")

#PAPER FIGURE
#DOTPLOT-IXN
( p <- ggplot(df, aes(x=relative_time, y = PNUM, color = ixn)) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df$DATASET) +
  scale_color_manual(values=c("black","#D81897")) +
  theme_minimal() + labs(
    title = "DATASET Utterances USING INTERACTION",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "USING IXN"
  ))
```

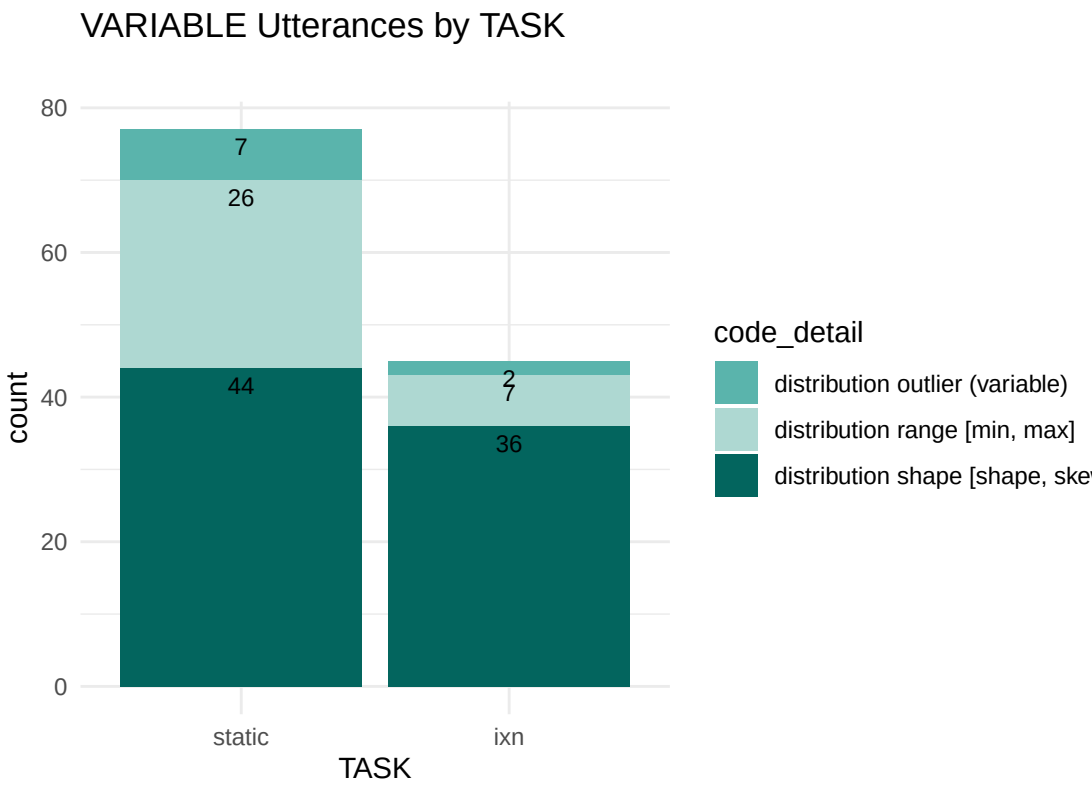



```
ggsave(p, file="figures/IXN_DATASET_time.png", width=6, height=4)
```

VARIABLE UTTERANCES

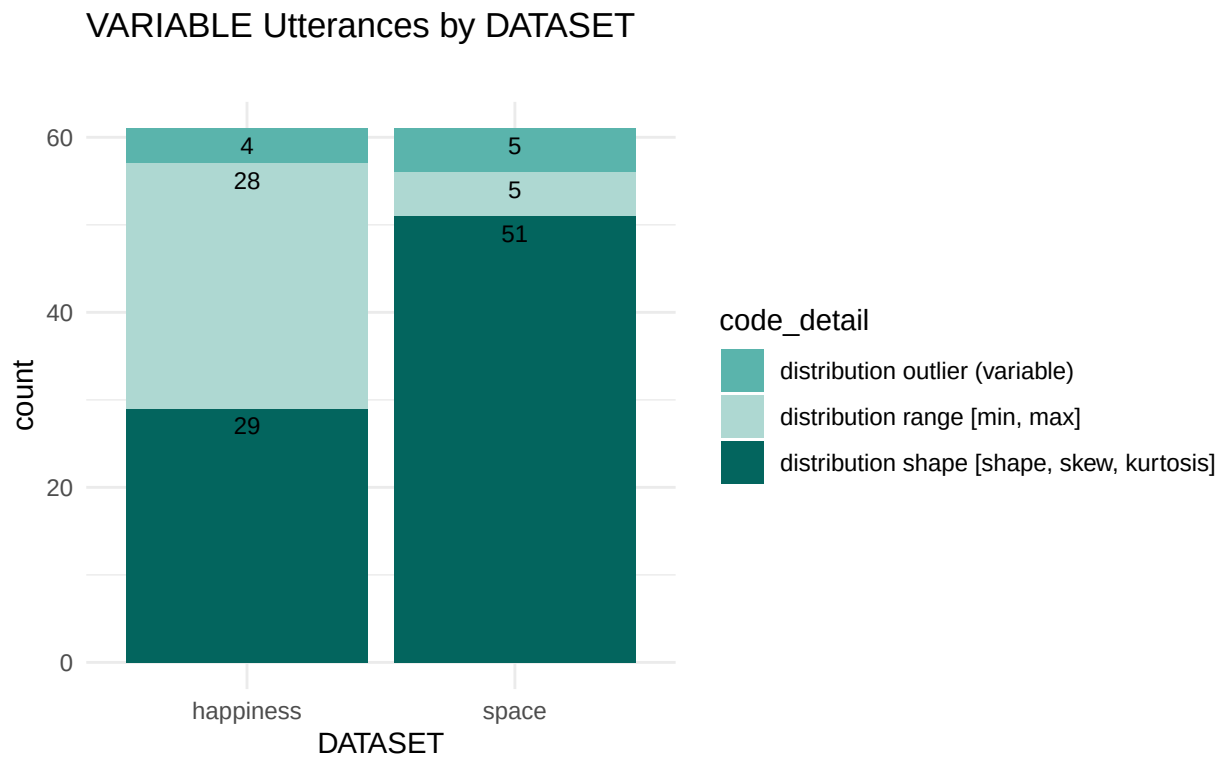
```
#PREP DATA FRAMES
df_variable <- df_coded %>%
  filter(code_topic=="VARIABLE") %>%
  dplyr::select(pid,PNUM,TASK,DATASET,code_detail)

df_time_variable <- df_coded %>%
  filter(code_topic=="VARIABLE") %>%
  dplyr::select(pid,PNUM,TASK,DATASET,relative_time,code_detail)

df_summary_task <- df_variable %>%
  group_by(code_detail, TASK) %>%
  dplyr::summarise(c = n())

df_summary_dataset <- df_variable %>%
  group_by(code_detail, DATASET) %>%
  dplyr::summarise(c = n())


#DETAILS BY TASK
```

```
ggplot(df_summary_task, aes(x = TASK, y=c, fill= code_detail)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_manual(values=c("#5AB4AC","#AED8D2","#03655E"))+
  # scale_fill_brewer(type="seq", palette = 5) +
  labs( title = "VARIABLE Utterances by TASK",
        subtitle = "",
        caption = "notably more RANGE obs in STATIC than INTERACTIVE \n TODO consider collapsing distri
        x= "TASK", y = "count") + theme_minimal()
```



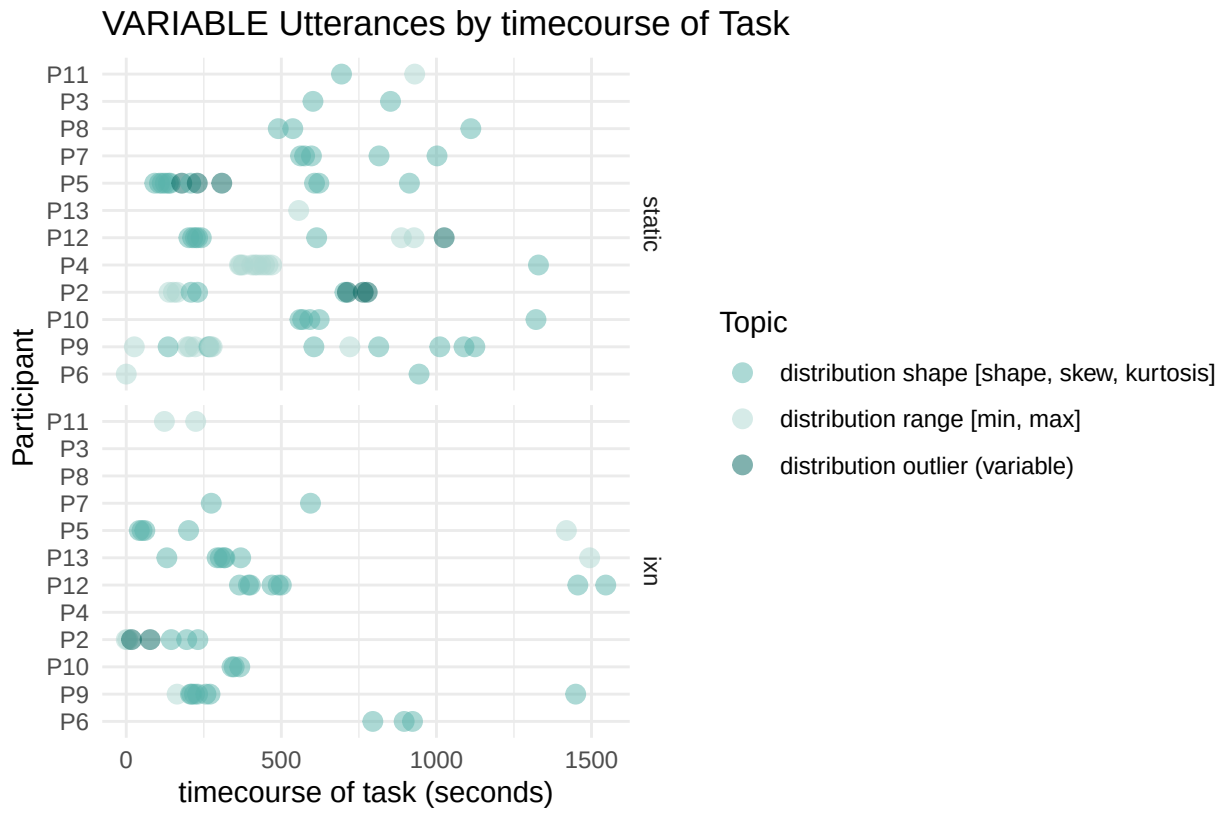
VARIABLE Utterances

```
#DETAILS BY DATASET
ggplot(df_summary_dataset, aes(x = DATASET, y=c, fill= code_detail)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  # scale_fill_brewer(type="seq", palette = 5) +
  scale_fill_manual(values=c("#5AB4AC","#AED8D2","#03655E"))+
  labs( title = "VARIABLE Utterances by DATASET",
        caption = "Notably more SHAPE in SPACE than HAPPINESS \n notably fewer RANGE in SPACE than HAPP
        subtitle = "",
        x= "DATASET", y = "count") + theme_minimal()
```



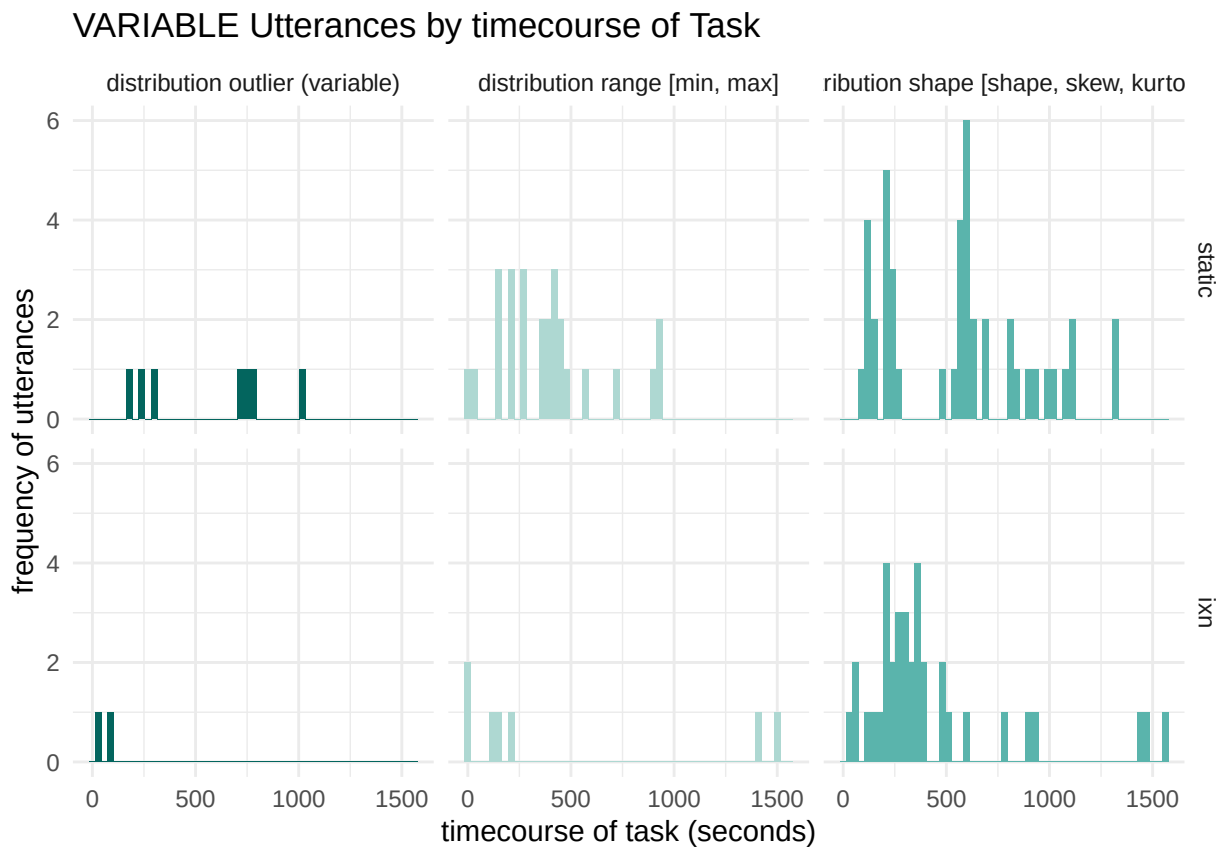
Notably more SHAPE in SPACE than HAPPINESS
notably fewer RANGE in SPACE than HAPPINESS
TODO are shape and range normalized across variable types?

```
#DETAILS DOTPLOT
ggplot(df_time_variable, aes(x=relative_time, y = PNUM, color=fct_rev(code_detail))) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df_time_variable$TASK) +
  # scale_color_brewer(type="seq", palette = 5) +
  scale_color_manual(values=c("#5AB4AC","#AED8D2","#03655E"))+
  theme_minimal() + labs(
    title = "VARIABLE Utterances by timecourse of Task",
    caption = "IXN appears more BIMODAL than STATIC where the distribution is more uniform",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "Topic"
  )
```

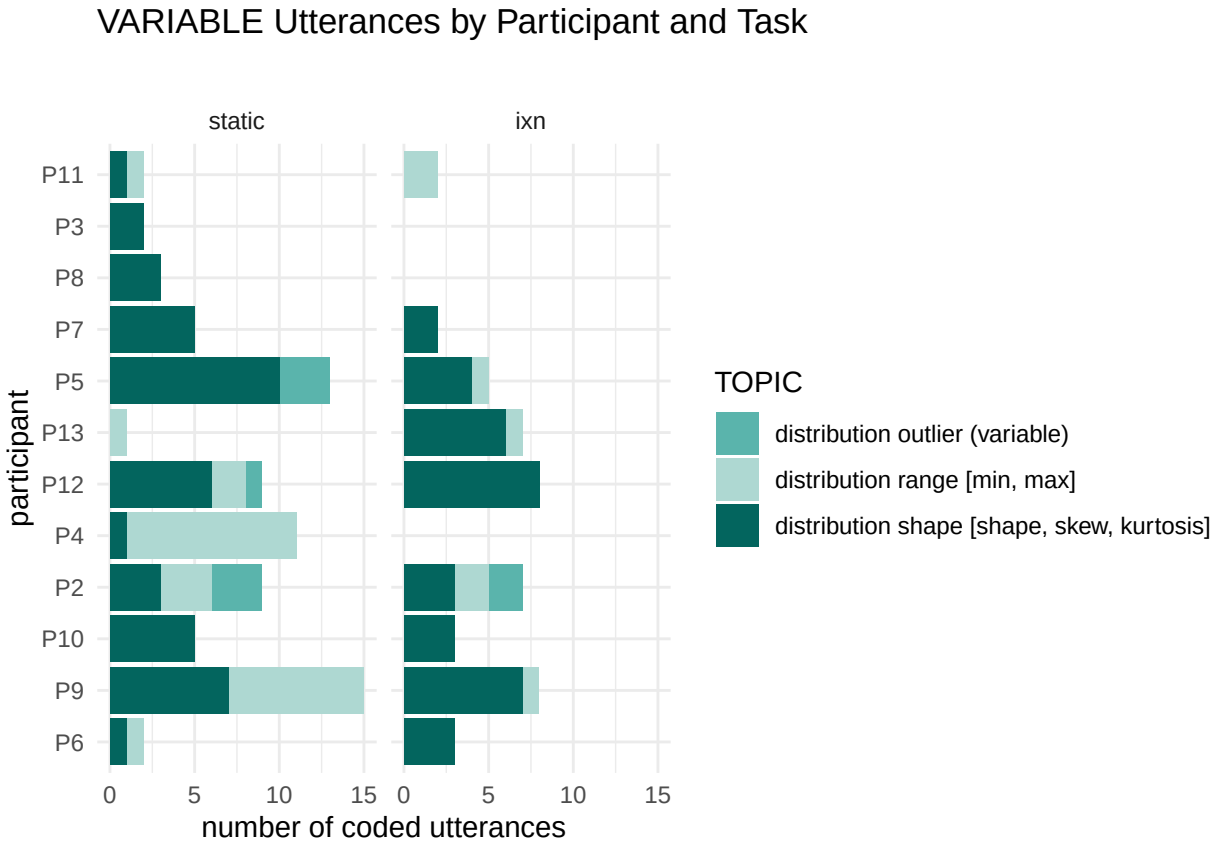


MODAL than STATIC where the distribution is more uniform

```
#DETAIL HISTOGRAMS BY TASK
ggplot(df_time_variable, aes(x = relative_time, fill = fct_rev(code_detail))) +
  geom_histogram(binwidth = 30) +
  facet_grid(df_time_variable$TASK ~ df_time_variable$code_detail ) +
  # scale_fill_brewer(type="seq", palette = 5) +
  scale_fill_manual(values=c("#5AB4AC","#AED8D2","#03655E"))+
  theme_minimal() + labs(
    title = "VARIABLE Utterances by timecourse of Task",
    x= "timecourse of task (seconds)", y = "frequency of utterances"
  ) + theme_minimal() + theme(legend.position = "blank")
```



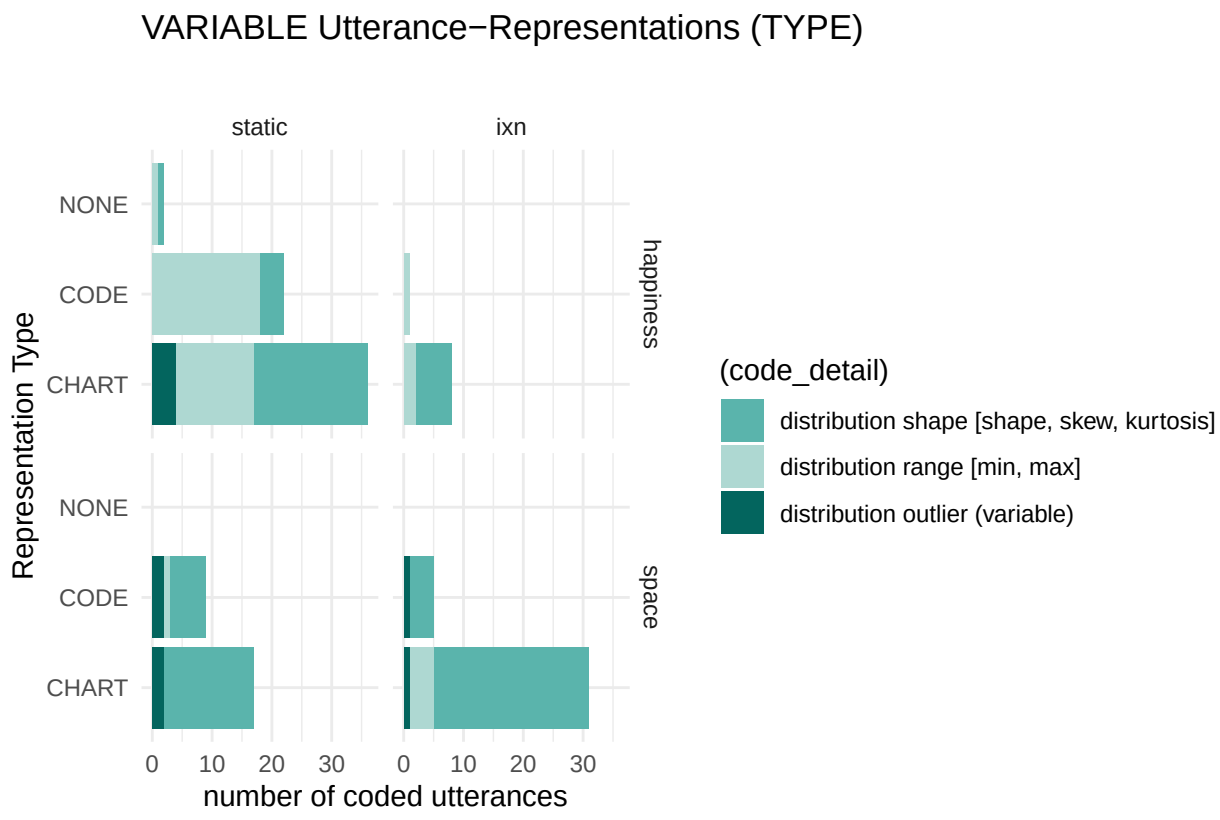

```
#PAPER FIGURE HERE
#VARIABLE UTTERANCES by PARTICPANT facet TASK
(p <- gf_bar( PNUM ~., fill = ~ (code_detail), data = df_variable) %>%
  gf_facet_grid(.~TASK) +
  # scale_fill_brewer(type="seq", palette = 5) +
  scale_fill_manual(values=c("#5AB4AC","#AED8D2","#03655E"))+
  labs(
    title = "VARIABLE Utterances by Participant and Task",
    subtitle = "",
    x = "number of coded utterances",
    y = "participant",
    fill = "TOPIC",
    # caption = "substantial individual differences, most everyone made some comments \n at some point
  ) + theme_minimal()
)
```



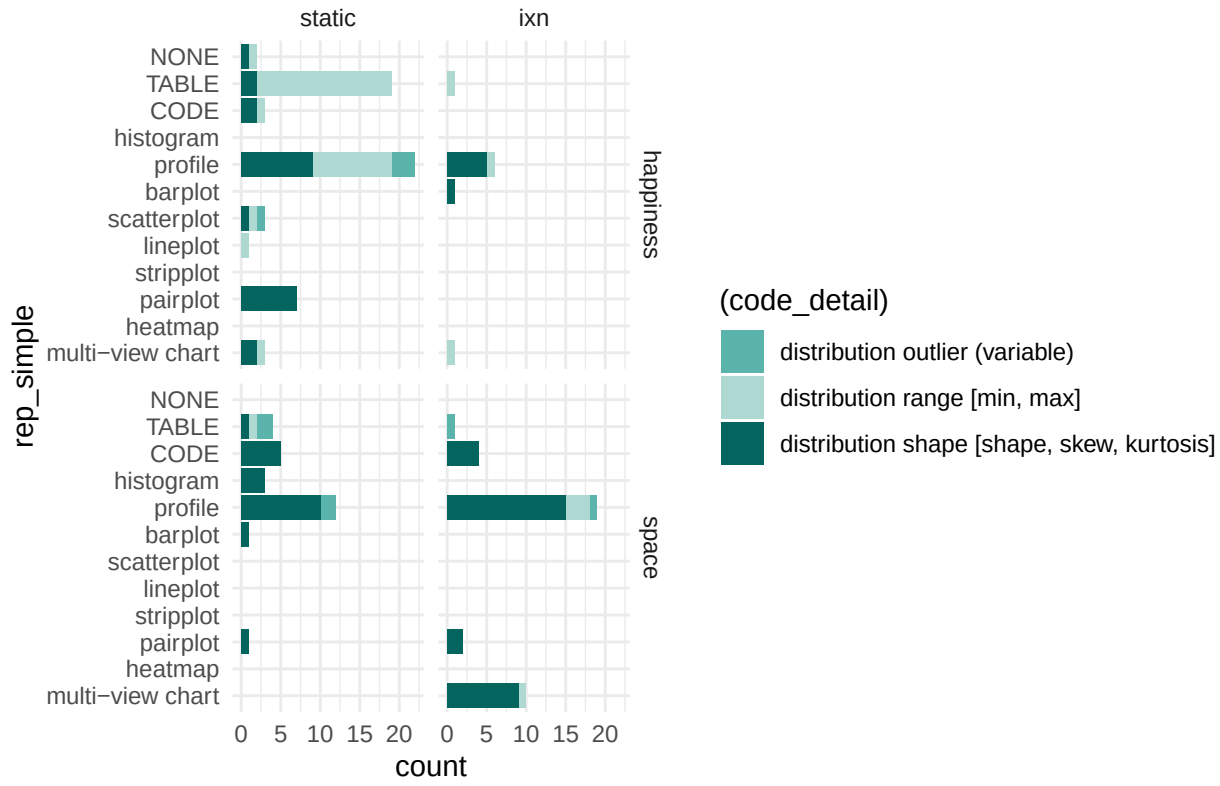

```
ggsave(p, file="figures/UTTERANCE_detail_VARIABLE_participants.png", width=6, height=4)
```

VARIABLE Representations THIS SECTION covers representations EXPLICITLY LINKED to UTTERANCES. Does not include ALL representations generated, but rather, what representations were being used when the participant generated utterances.

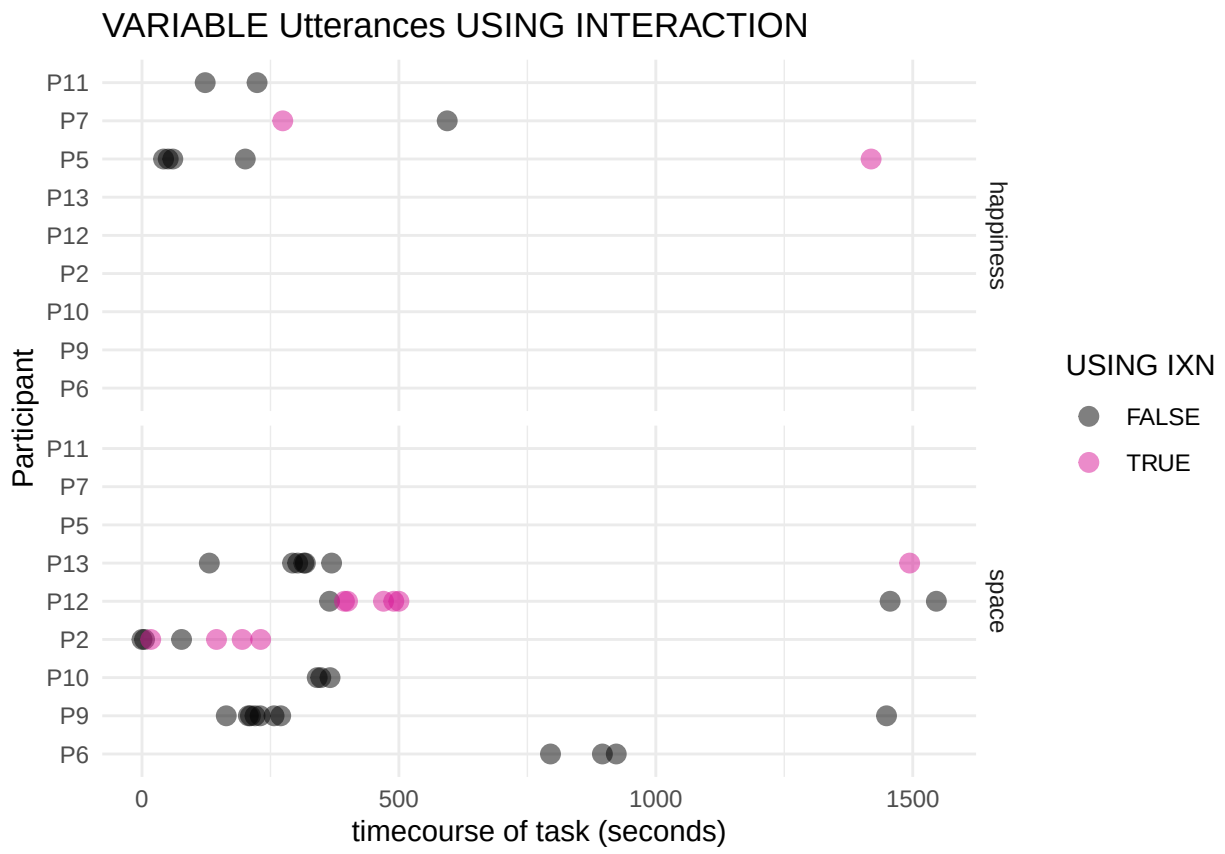
```
df <- df_codedrep %>% filter(code_topic =="VARIABLE")


#DETAIL REP TYPE
gf_bar( rep_type ~., fill = ~ fct_rev(code_detail), data = df) %>%
 gf_facet_grid(DATASET~TASK) +
 scale_fill_manual(values=c("#5AB4AC","#AED8D2","#03655E"))+
  labs(
    title = "VARIABLE Utterance-Representations (TYPE) ",
    subtitle = "",
    caption = "",
    x = "number of coded utterances",
    y = "Representation Type",
    fill = "(code_detail)"
  ) + theme_minimal()
```




```
(p <- gf_bar(  ~ rep_simple, fill = ~(code_detail), data = df) %>%
  gf_facet_grid( DATASET ~ TASK) +
  scale_fill_manual(values=c("#5AB4AC","#AED8D2","#03655E"))+
  coord_flip() +
  scale_x_discrete(limits = c(
            "multi-view chart",
            "heatmap",
            "pairplot",
            "stripplot",
            "lineplot",
            "scatterplot",
            "barplot",
            "profile",
            "histogram",
            "CODE",
            "TABLE",
            "NONE"
  ))+
  theme_minimal() + labs(
    title = "VARIABLE Utterance-Representations (DETAIL)"
))
```




```
ggsave(p, file="figures/UTTERANCE-REP_detail_VARIABLE_factors.png", width=6, height=4)


df <- df_codedrep %>% filter(code_topic == "VARIABLE") %>% filter(TASK=="ixn")

#PAPER FIGURE
#DOTPLOT-IXN
( p <- ggplot(df, aes(x=relative_time, y = PNUM, color = ixn)) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df$DATASET) +
  scale_color_manual(values=c("black","#D81897")) +
  theme_minimal() + labs(
    title = "VARIABLE Utterances USING INTERACTION",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "USING IXN"
  ))
```

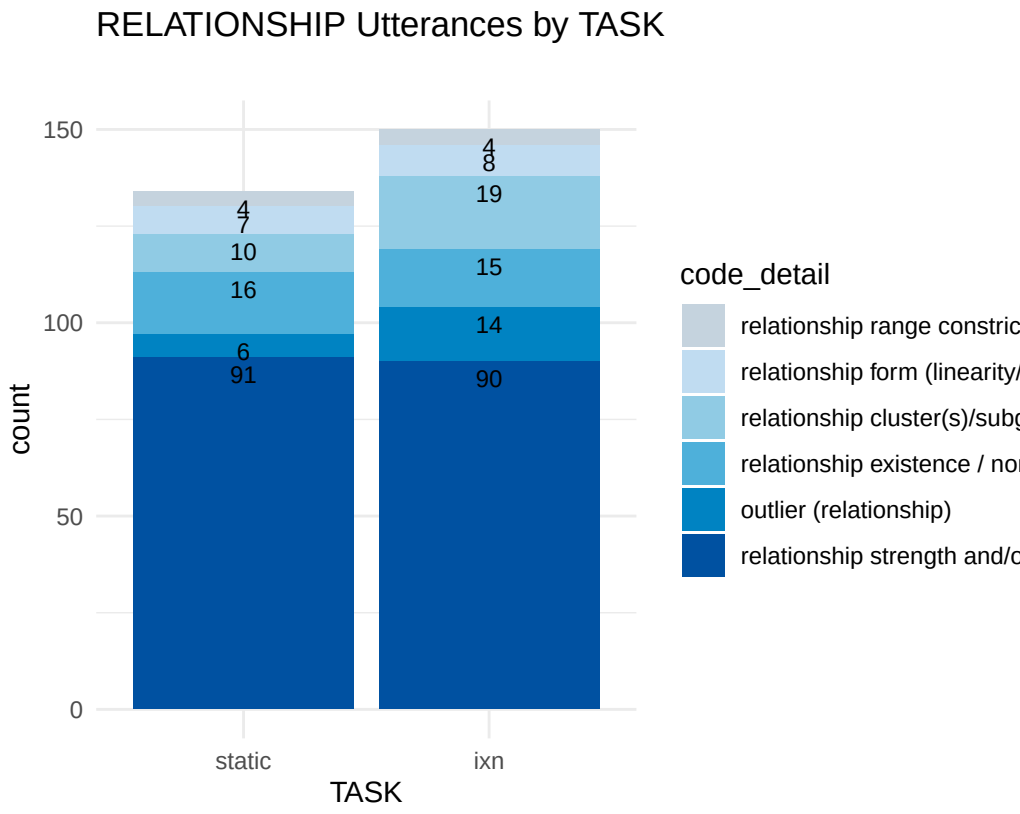



```
ggsave(p, file="figures/IXN_VARIABLE_time.png", width=6, height=4)
```

RELATIONSHIP UTTERANCES

```
#PREP DATA FRAMES
df_relationship <- df_coded %>%
  filter(code_topic=="RELATIONSHIP") %>%
  dplyr::select(pid,PNUM,TASK,DATASET,code_detail)

df_time_relationship <- df_coded %>%
  filter(code_topic=="RELATIONSHIP") %>%
  dplyr::select(pid,PNUM,TASK,DATASET,relative_time,code_detail)

df_summary_task <- df_relationship %>%
  group_by(code_detail, TASK) %>%
  dplyr::summarise(c = n())

df_summary_relationship <- df_relationship %>%
  group_by(code_detail, DATASET) %>%
  dplyr::summarise(c = n())


#DETAILS BY TASK
```

```
ggplot(df_summary_task, aes(x = TASK, y=c, fill= code_detail)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  # scale_fill_brewer(type="seq", palette = 3) +
  scale_fill_manual(values=c(
                "#C5D3DE",
                "#C0DCF1",
                "#90CBE4",
                "#4EB0DA",
                "#0083C2",
                "#0051A1"))+
  labs( title = "RELATIONSHIP Utterances by TASK",
        subtitle = "",
        caption = "TODO think about proportion of existence and strength/direction",
        x= "TASK", y = "count") + theme_minimal()
```

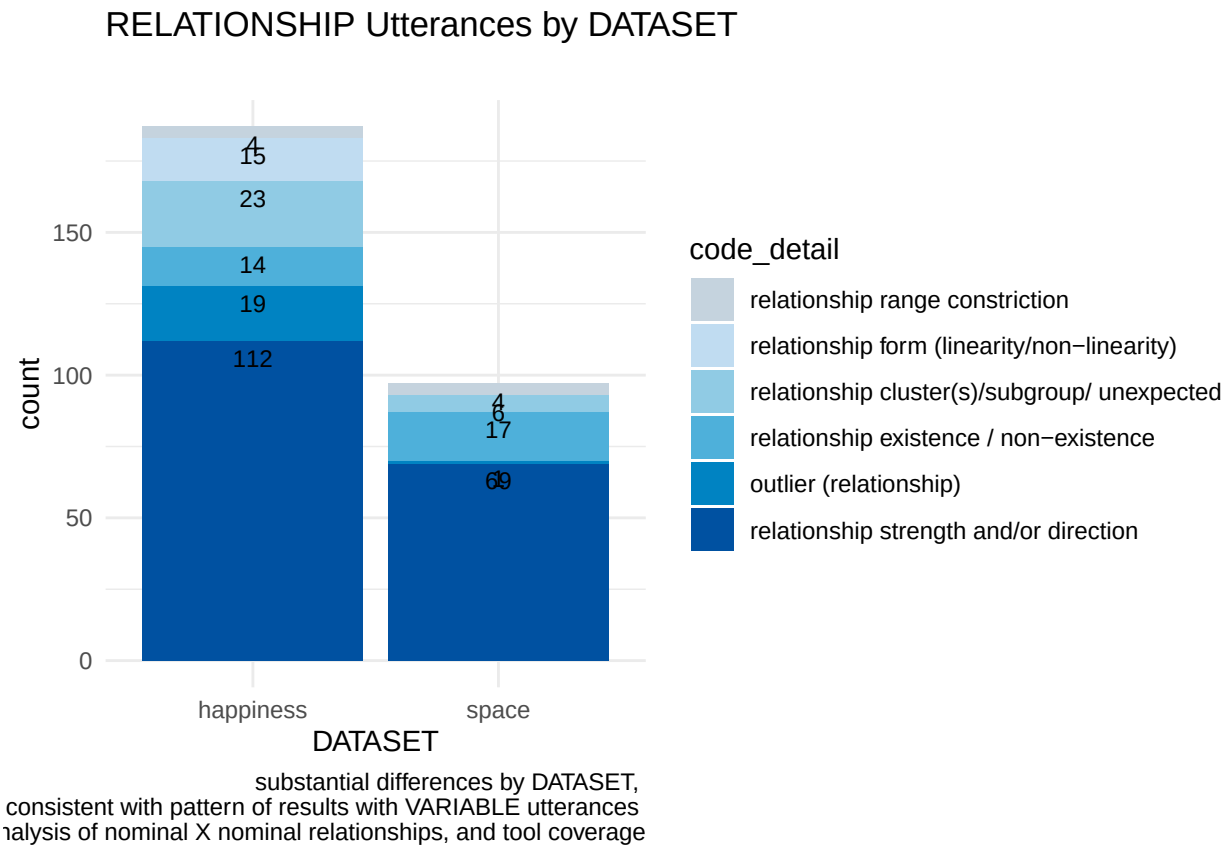


TODO think about proportion of existence and strength/direction

RELATIONSHIP Utterances

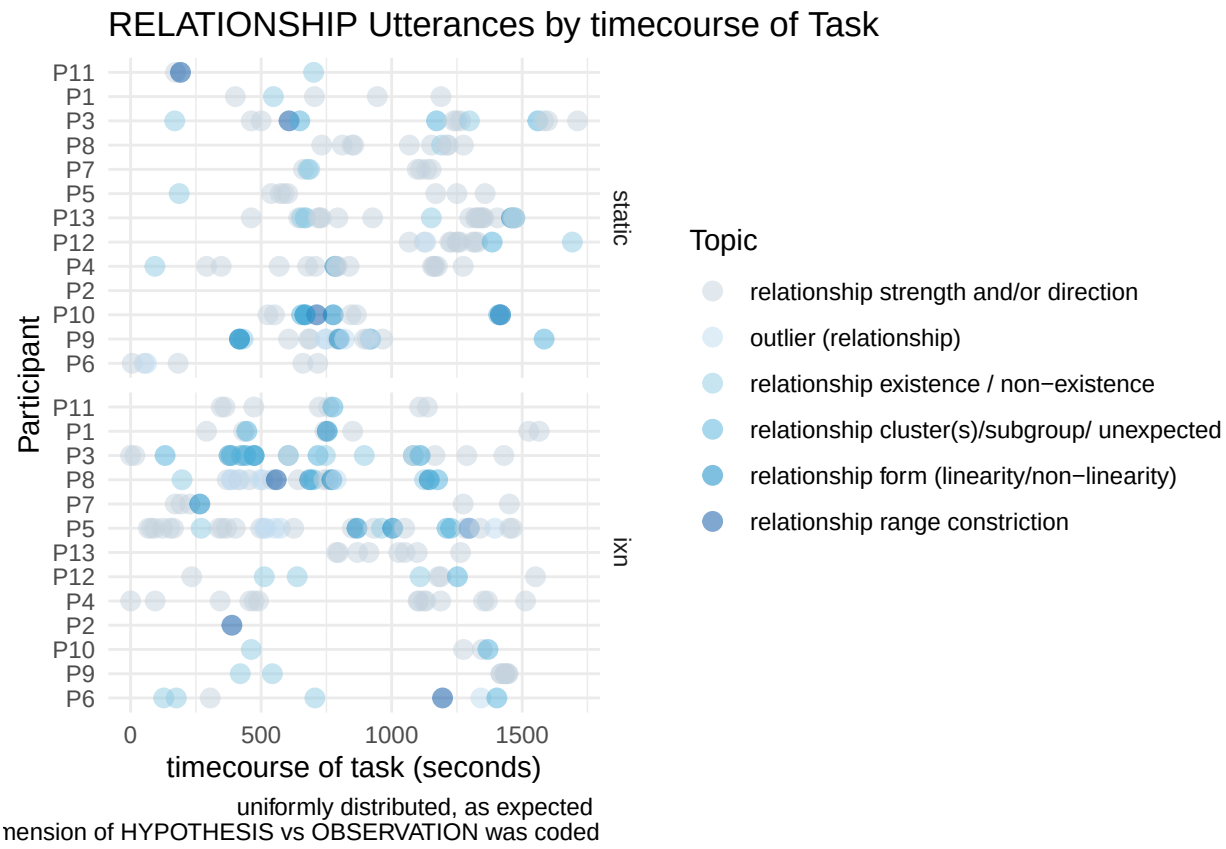
```
#DETAILS BY DATASET
ggplot(df_summary_relationship, aes(x = DATASET, y=c, fill= code_detail)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  # scale_fill_brewer(type="seq", palette = 3) +
  scale_fill_manual(values=c(
                "#C5D3DE",
                "#C0DCF1",
                "#90CBE4",
```

```
              "#4EB0DA",
              "#0083C2",
              "#0051A1"))+
  labs( title = "RELATIONSHIP Utterances by DATASET",
        subtitle = "",
        caption = "substantial differences by DATASET, \n consistent with pattern of results with VARIA
        x= "DATASET", y = "count") + theme_minimal()
```




```
#DETAILS DOTPLOT
ggplot(df_time_relationship, aes(x=relative_time, y = PNUM, color=fct_rev(code_detail))) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df_time_relationship$TASK) +
  # scale_color_brewer(type="seq", palette = 3) +
  scale_color_manual(values=c(
              "#C5D3DE",
              "#C0DCF1",
              "#90CBE4",
              "#4EB0DA",
              "#0083C2",
              "#0051A1"))+
  theme_minimal() + labs(
    title = "RELATIONSHIP Utterances by timecourse of Task",
    caption = "uniformly distributed, as expected \n may see variance if dimension of HYPOTHESIS vs OBS
    x= "timecourse of task (seconds)", y = "Participant",
    color = "Topic"
```

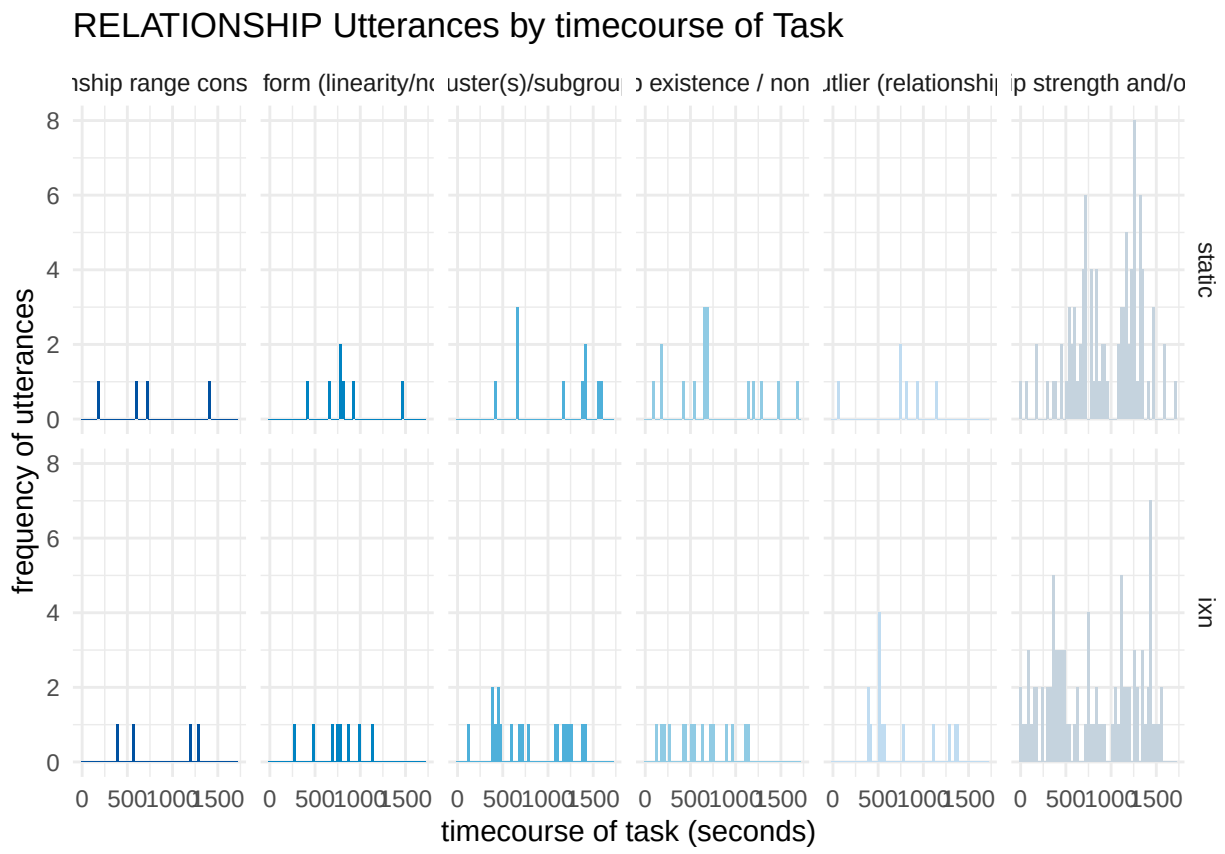
```
  )
```



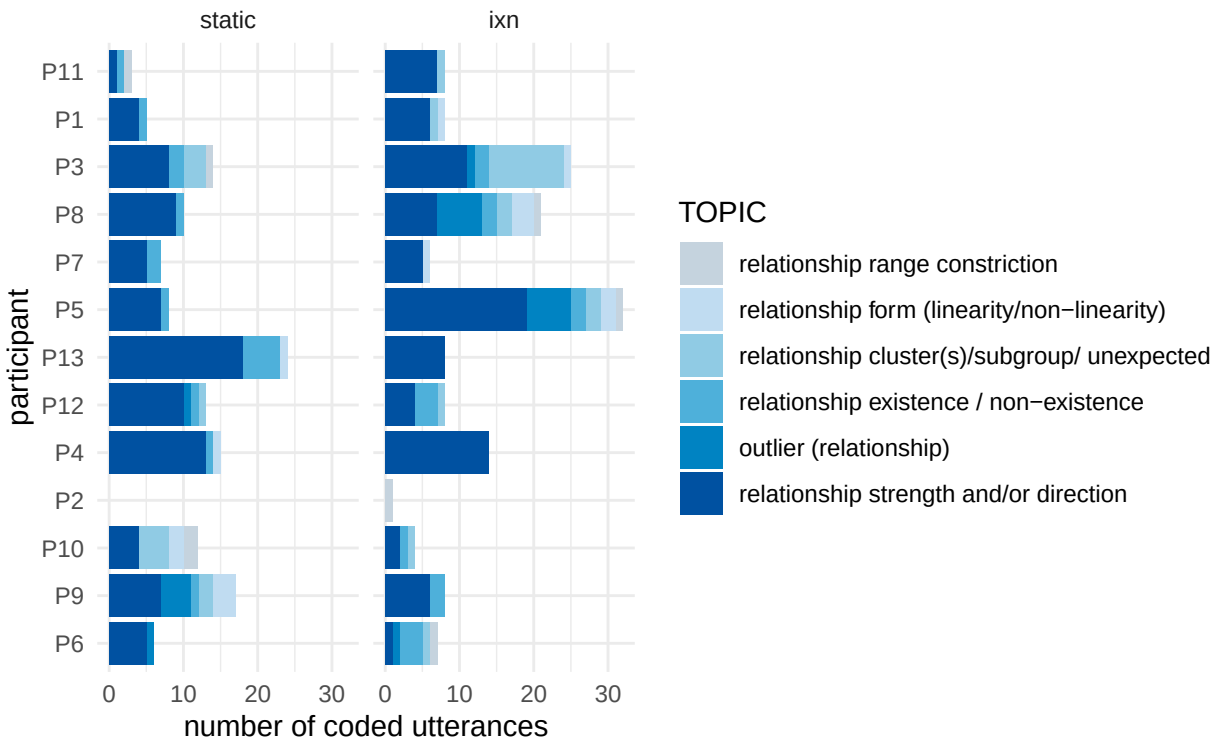
uniformly distributed, as expected

nension of HYPOTHESIS vs OBSERVATION was coded

```
#DETAIL HISTOGRAMS BY TASK
ggplot(df_time_relationship, aes(x = relative_time, fill = fct_rev(code_detail))) +
  geom_histogram(binwidth = 30) +
  facet_grid(df_time_relationship$TASK ~ df_time_relationship$code_detail ) +
  # scale_fill_brewer(type="seq", palette = 3) +
  scale_fill_manual(values=c(
              "#C5D3DE",
              "#C0DCF1",
              "#90CBE4",
              "#4EB0DA",
              "#0083C2",
              "#0051A1"))+
  theme_minimal() + labs(
    title = "RELATIONSHIP Utterances by timecourse of Task",
    x= "timecourse of task (seconds)", y = "frequency of utterances"
  ) + theme_minimal() + theme(legend.position = "blank")
```




```
#PAPER FIGURE HERE
#RELATIONSHIP UTTERANCES by PARTICPANT facet TASK
(p <- gf_bar( PNUM ~., fill = ~ (code_detail), data = df_relationship) %>%
  gf_facet_grid(.~TASK) +
  # scale_fill_brewer(type="seq", palette = 1) +
  scale_fill_manual(values=c(
              "#C5D3DE",
              "#C0DCF1",
              "#90CBE4",
              "#4EB0DA",
              "#0083C2",
              "#0051A1"))+
  labs(
    title = "RELATIONSHIP Utterances by Participant and Task",
    subtitle = "",
    x = "number of coded utterances",
    y = "participant",
    fill = "TOPIC"
  ) + theme_minimal()
)
```

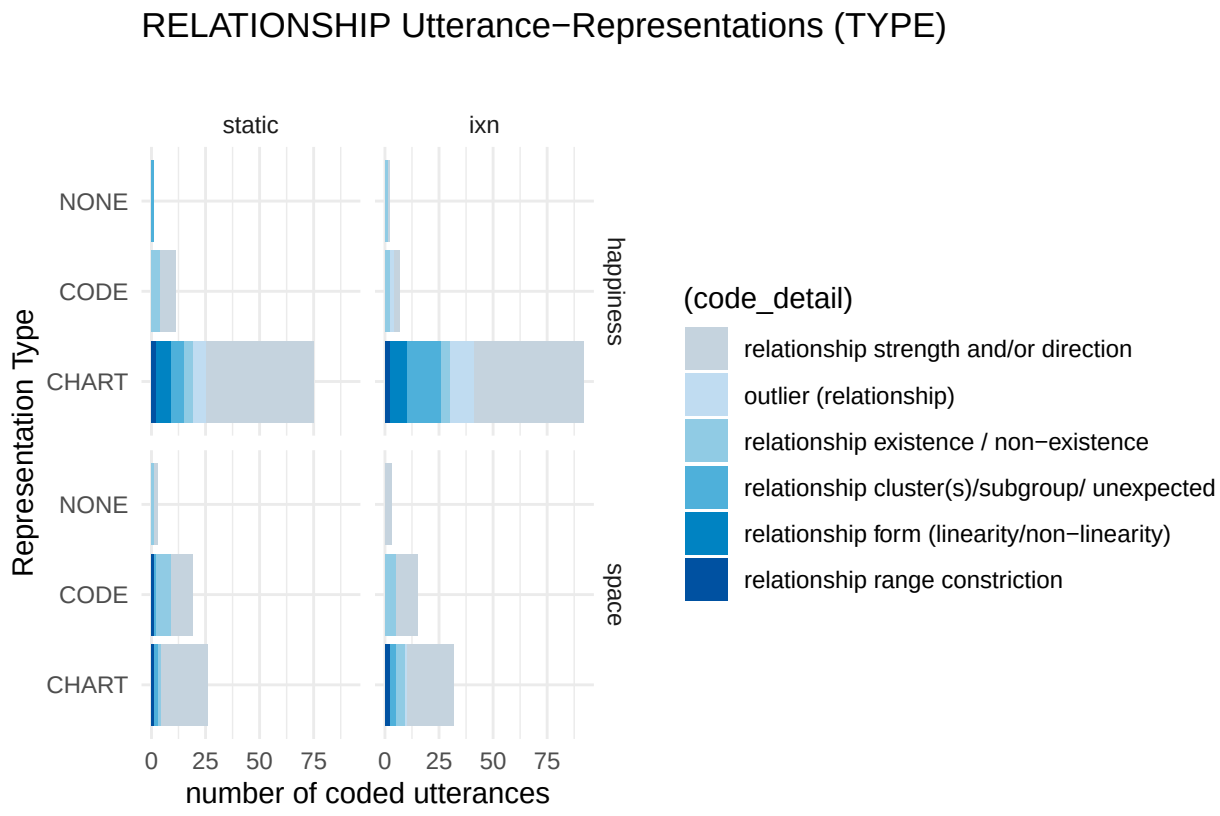



```
ggsave(p, file="figures/UTTERANCE_detail_RELATIONSHIP_participants.png", width=8, height=4)
```

RELATIONSHIP Representations THIS SECTION covers representations EXPLICITLY LINKED to UTTERANCES. Does not include ALL representations generated, but rather, what representations were being used when the participant generated utterances.

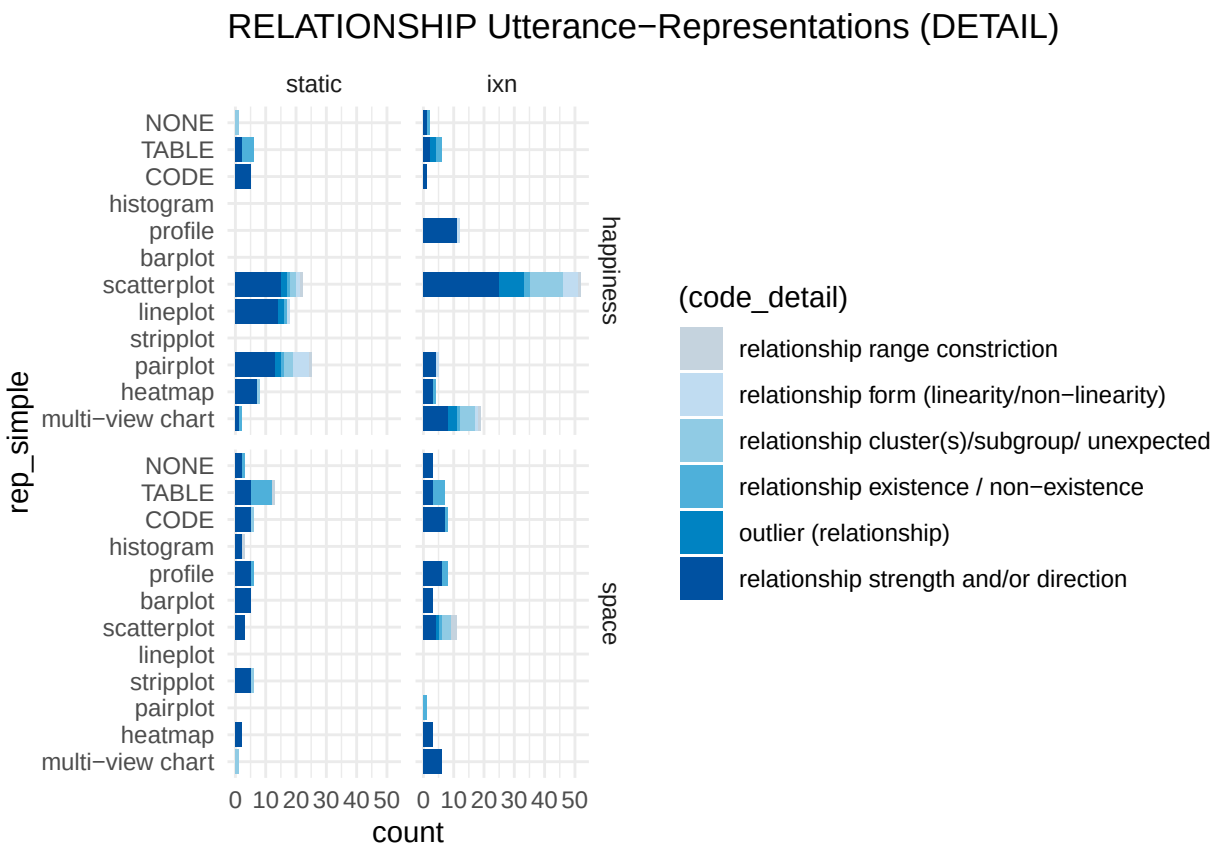
```
df <- df_codedrep %>% filter(code_topic =="RELATIONSHIP")

#DETAIL REP TYPE
gf_bar( rep_type ~., fill = ~ fct_rev(code_detail), data = df) %>%
  gf_facet_grid(DATASET~TASK) +
  scale_fill_manual(values=c(
                "#C5D3DE",
                "#C0DCF1",
                "#90CBE4",
                "#4EB0DA",
                "#0083C2",
                "#0051A1"))+
  labs(
    title = "RELATIONSHIP Utterance-Representations (TYPE) ",
    subtitle = "",
    caption = "",
    x = "number of coded utterances",
    y = "Representation Type",
```

```
    fill = "(code_detail)"
  ) + theme_minimal()
```



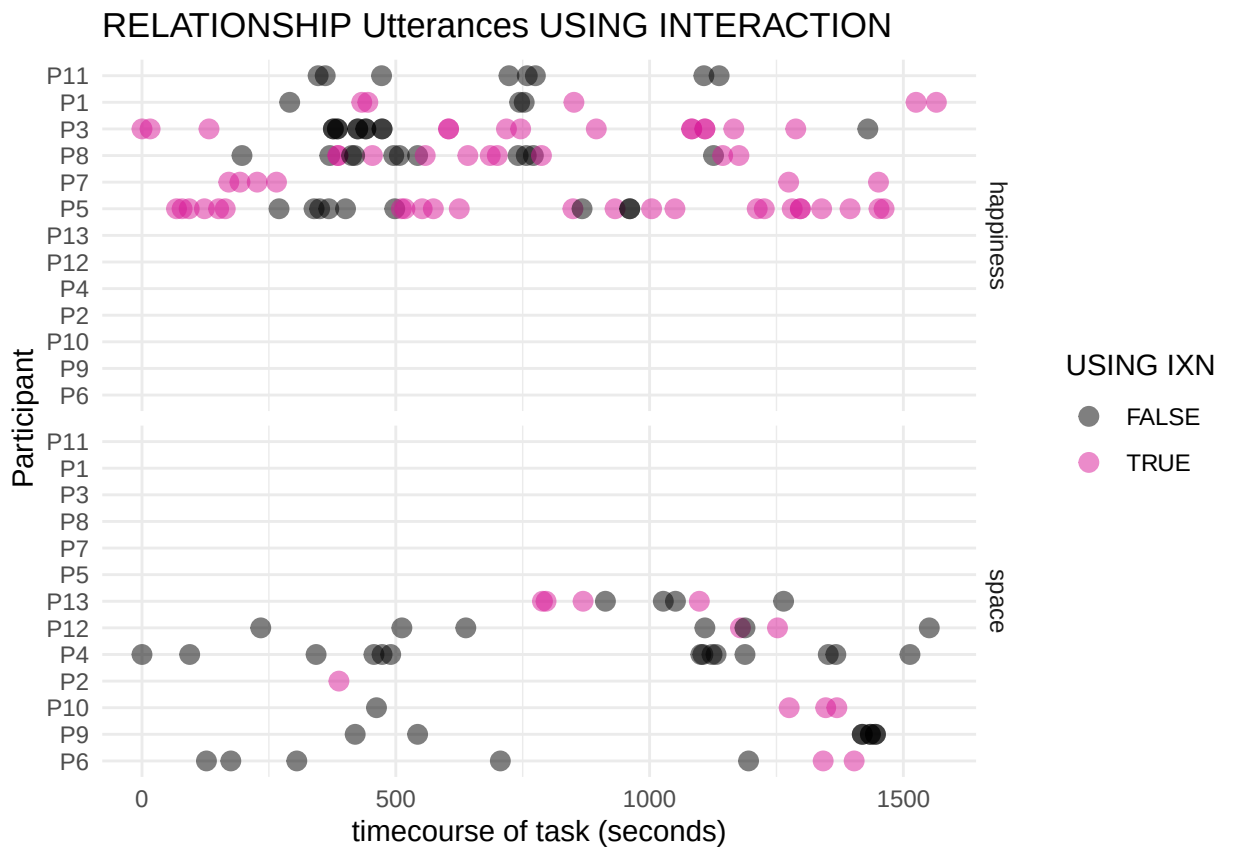

```
#PAPER FIGURE
(p <- gf_bar(  ~ rep_simple, fill = ~(code_detail), data = df) %>%
  gf_facet_grid( DATASET ~ TASK) +
 scale_fill_manual(values=c(
              "#C5D3DE",
              "#C0DCF1",
              "#90CBE4",
              "#4EB0DA",
              "#0083C2",
              "#0051A1"))+
  coord_flip() +
  scale_x_discrete(limits = c(
           "multi-view chart",
           "heatmap",
           "pairplot",
           "stripplot",
           "lineplot",
           "scatterplot",
           "barplot",
           "profile",
           "histogram",
           "CODE",
```

```
            "TABLE",
            "NONE"
  ))+
  theme_minimal() + labs(
    title = "RELATIONSHIP Utterance-Representations (DETAIL)"
))
```




```
ggsave(p, file="figures/UTTERANCE-REP_detail_RELATIONSHIP_factors.png", width=8, height=4)
```

```
df <- df_codedrep %>% filter(code_topic == "RELATIONSHIP") %>% filter(TASK=="ixn")

#PAPER FIGURE
#DOTPLOT-IXN
( p <- ggplot(df, aes(x=relative_time, y = PNUM, color = ixn)) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df$DATASET) +
  scale_color_manual(values=c("black","#D81897")) +
  theme_minimal() + labs(
    title = "RELATIONSHIP Utterances USING INTERACTION",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "USING IXN"
  ))
```




```
ggsave(p, file="figures/IXN_RELATIONSHIP_time.png", width=6, height=4)
```

EXPLORE UTTERANCE REPRESENTATIONS

*NOTE: A finer-grained view of representations used when making particular kinds of utterances is addressed in the DETAIL of Utterances subsections of EXPLORE UTTERANCES (directly above). Here we address only an overview of the *type* of representations created by the structural factors in the study.

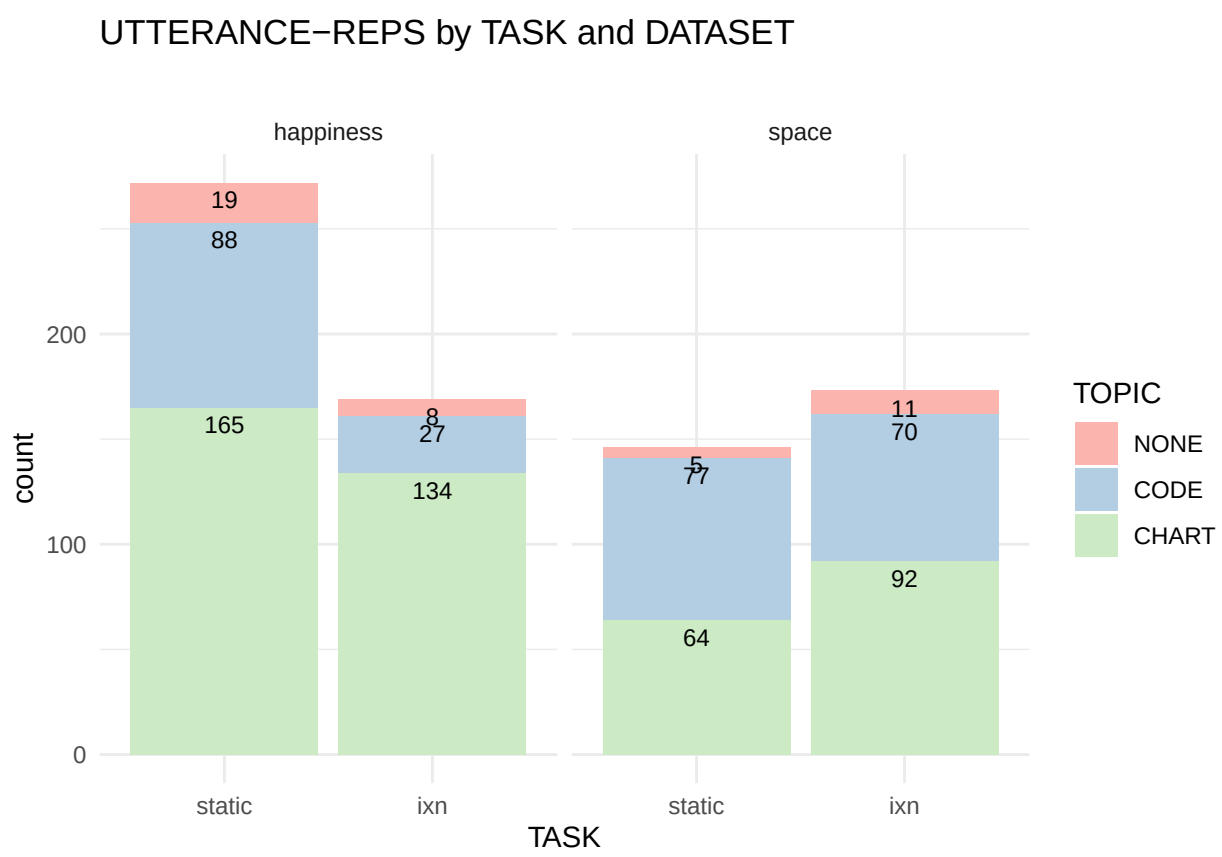
This section covers representations EXPLICITLY LINKED to UTTERANCES. Does not include ALL representations generated, but rather, what representations were being used when the participant generated utterances.

[CATEGORY of] Representation

by TASK and DATASET

```
#DF SUMMARIZED BY TASK + DATASET
df_summary <- df_codedrep %>%
  group_by(rep_type, TASK,DATASET) %>%
  dplyr::summarise(
    c = n()
  )
```

```
#STACKED BAR BY TASK FACET DATASET
ggplot(df_summary, aes(x = TASK, y=c, fill= fct_rev(rep_type))) +
  facet_wrap(df_summary$DATASET) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_brewer(type="qual", palette = 4) +
  labs( title = "UTTERANCE-REPS by TASK and DATASET",
        subtitle = "",
        x= "TASK", y = "count", fill="TOPIC") + theme_minimal()
```




```
# + theme(legend.position = "blank")
```

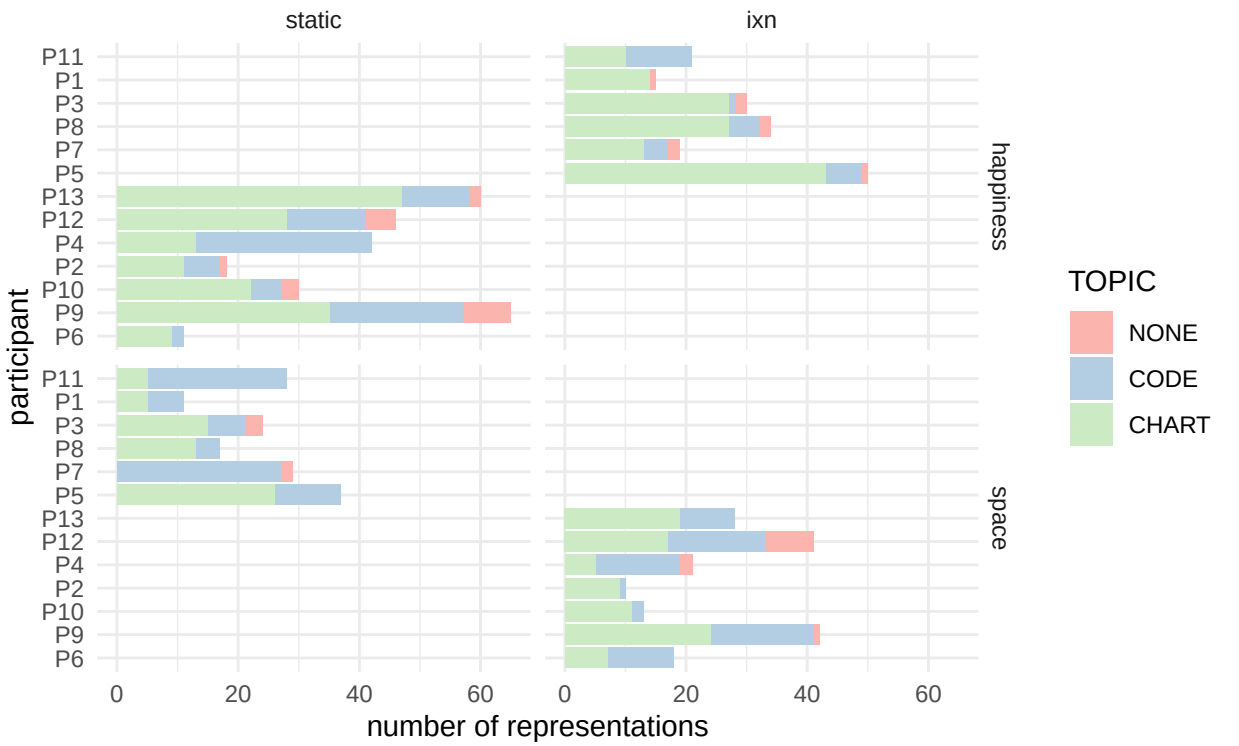
by PARTICIPANT

```
#COUNT BY PARTICIPANT
ctable(x = df_codedrep$PNUM,
       y = df_codedrep$rep_type,
       prop = "r")
```

Cross-Tabulation, Row Proportions
PNUM * rep_type
Data Frame: df_codedrep

rep_type	CHART	CODE	NONE	Total
PNUM				
P6	16 (55.2%)	13 (44.8%)	0 (0.0%)	29 (100.0%)
P9	59 (55.1%)	39 (36.4%)	9 (8.4%)	107 (100.0%)
P10	33 (76.7%)	7 (16.3%)	3 (7.0%)	43 (100.0%)
P2	20 (71.4%)	7 (25.0%)	1 (3.6%)	28 (100.0%)
P4	18 (28.6%)	43 (68.3%)	2 (3.2%)	63 (100.0%)
P12	45 (51.7%)	29 (33.3%)	13 (14.9%)	87 (100.0%)
P13	66 (75.0%)	20 (22.7%)	2 (2.3%)	88 (100.0%)
P5	69 (79.3%)	17 (19.5%)	1 (1.1%)	87 (100.0%)
P7	13 (27.1%)	31 (64.6%)	4 (8.3%)	48 (100.0%)
P8	40 (78.4%)	9 (17.6%)	2 (3.9%)	51 (100.0%)
P3	42 (77.8%)	7 (13.0%)	5 (9.3%)	54 (100.0%)
P1	19 (73.1%)	6 (23.1%)	1 (3.8%)	26 (100.0%)
P11	15 (30.6%)	34 (69.4%)	0 (0.0%)	49 (100.0%)
Total	455 (59.9%)	262 (34.5%)	43 (5.7%)	760 (100.0%)

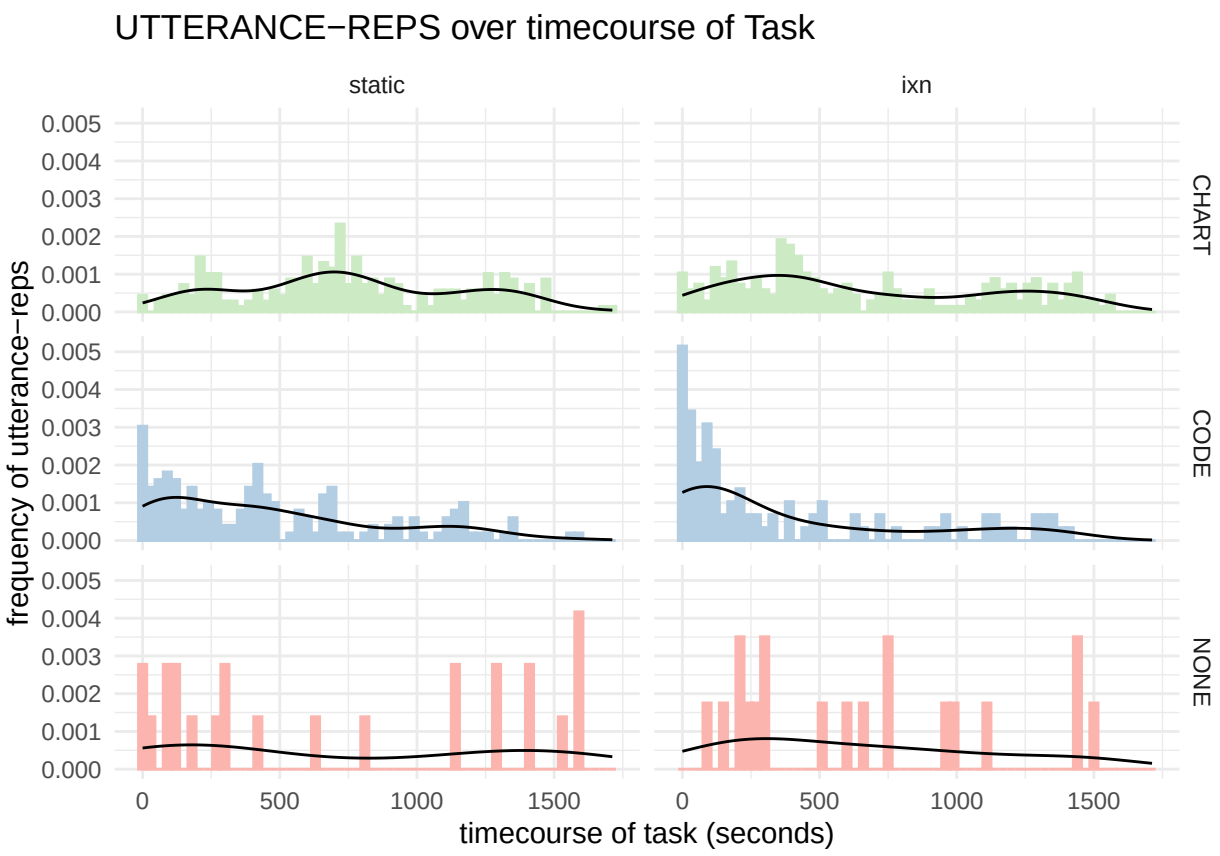
```
#TOPICS by PARTICPANT facet TASK
(p <- gf_bar( PNUM ~., fill = ~ fct_rev(rep_type), data = df_codedrep) %>%
  gf_facet_grid(DATASET~TASK) +
  scale_fill_brewer(type="qual", palette = 4) +
  labs(
    title = "Utterance-Represenations by Participant, Dataset and Task",
    subtitle = "",
    x = "number of representations",
    y = "participant",
    fill = "TOPIC"
  ) + theme_minimal())
```



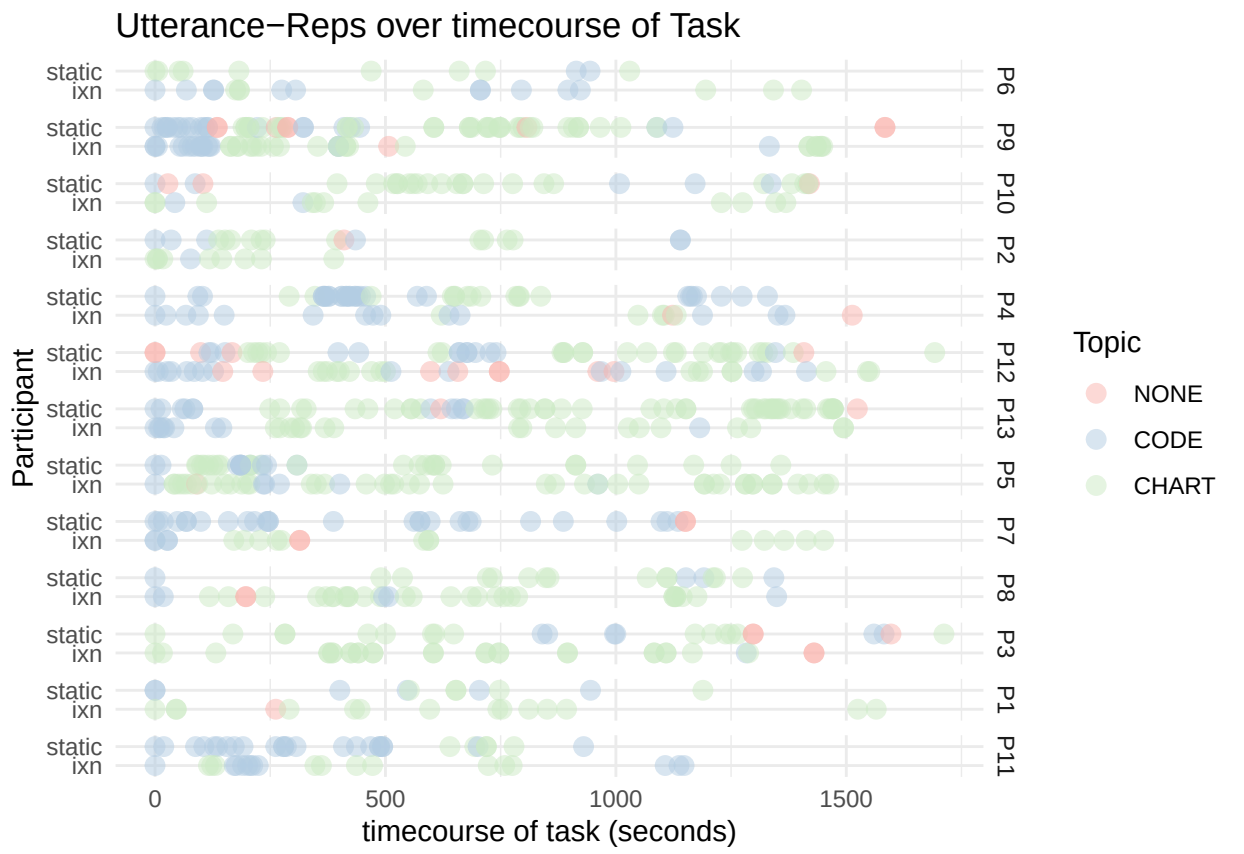

```
# ggsave(p, file="figures/utterance_reptypes_by_count.png")
```

by TIME

```
#HISTOGRAMS BY TASK
ggplot(df_codedrep, aes(x = relative_time)) +
  geom_histogram(binwidth = 30,aes(y=..density.., fill = fct_rev(rep_type), color = fct_rev(rep_type)))
  geom_density()+
  facet_grid(df_codedrep$rep_type ~ df_codedrep$TASK) +
  scale_fill_brewer(type="qual", palette = 4) +
  scale_color_brewer(type="qual", palette = 4) +
  theme_minimal() + labs(
    title = "UTTERANCE-REPS over timecourse of Task",
    x= "timecourse of task (seconds)", y = "frequency of utterance-reps",
    fill = "Topic"
  ) + theme_minimal() + theme(legend.position = "blank")
```




```
#DOTPLOT - FACET TASK
# (p <- ggplot(df_codedrep, aes(x=relative_time, y = PNUM, color=fct_rev(rep_type))) +
#   geom_point(alpha=0.5, size=3) +
#   facet_grid(df_codedrep$TASK) +
#   scale_color_brewer(type="qual", palette = 4) +
#   theme_minimal() + labs(
#     title = "Utterance-Reps over timecourse of Task",
#     x= "timecourse of task (seconds)", y = "Task",
#     color = "Topic"
#   ))
# ggsave(p, file="figures/UTTREP_classes_by_time_FACET.png")

#DOTPLOT STACKED TASKS
(p <- ggplot(df_codedrep, aes(x=relative_time, y = fct_rev(TASK), color=fct_rev(rep_type))) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df_codedrep$PNUM) +
  # facet_grid(df_codedrep$TASK ~ df_codedrep$DATASET) +
  scale_color_brewer(type="qual", palette = 4) +
  theme_minimal() + labs(
    title = "Utterance-Reps over timecourse of Task",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "Topic"
))
```



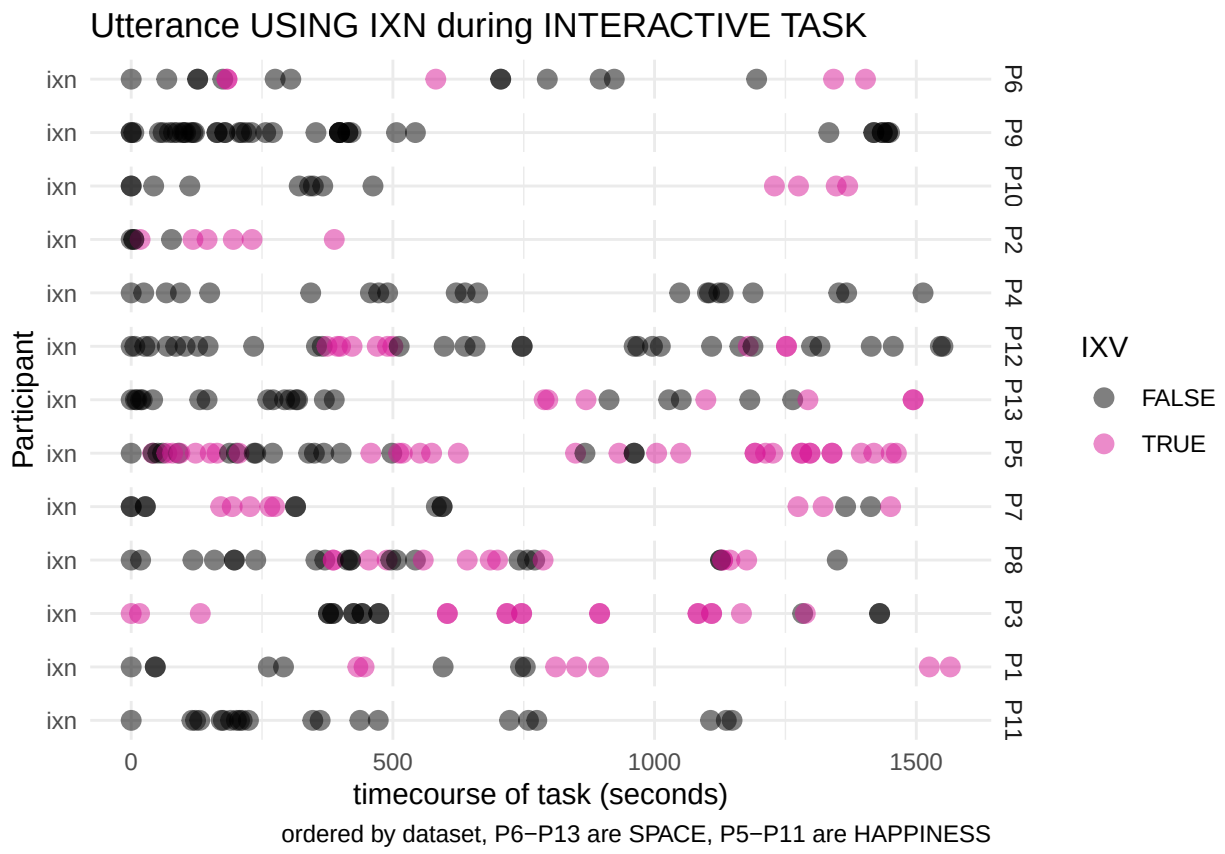

```
ggsave(p, file="figures/UTTREP_classes_by_time_STACK.png")
```

```
## Saving 6.5 x 4.5 in image
```

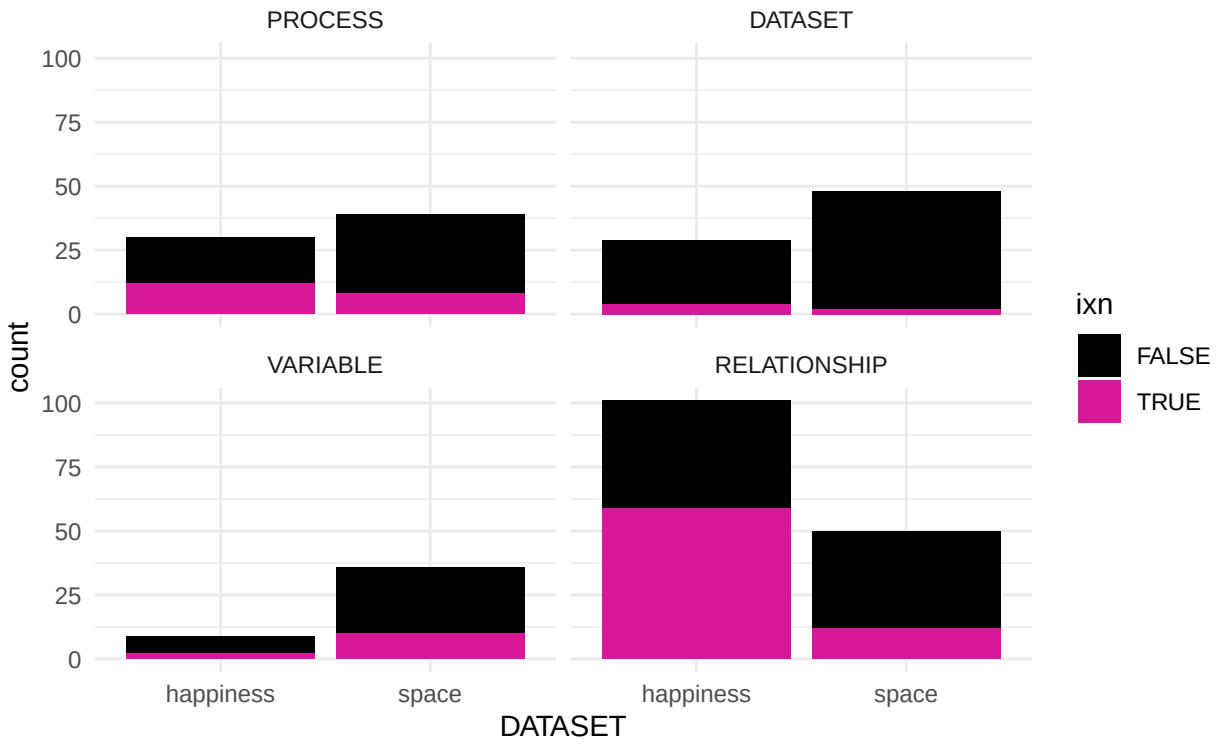
[INTERACTING with] Representations

This section explores the the relatively small number of utterancess that are flagged as occuring when interaction was ACTIVELY BEING USED SHOULD be a quick but effective.

```
#DOTPLOT
df <- df_codedrep %>% filter(TASK=="ixn")
ggplot(df, aes(x=relative_time, y = fct_rev(TASK), color=ixn)) +
  geom_point(alpha=0.5, size=3) +
  facet_grid(df$PNUM) +
  # facet_grid(df_codedrep$TASK ~ df_codedrep$DATASET) +
  scale_color_manual(values=c("black","#D81897")) +
  theme_minimal() + labs(
    title = "Utterance USING IXN during INTERACTIVE TASK",
    caption = "ordered by dataset, P6-P13 are SPACE, P5-P11 are HAPPINESS",
    x= "timecourse of task (seconds)", y = "Participant",
    color = "IXV"
)
```

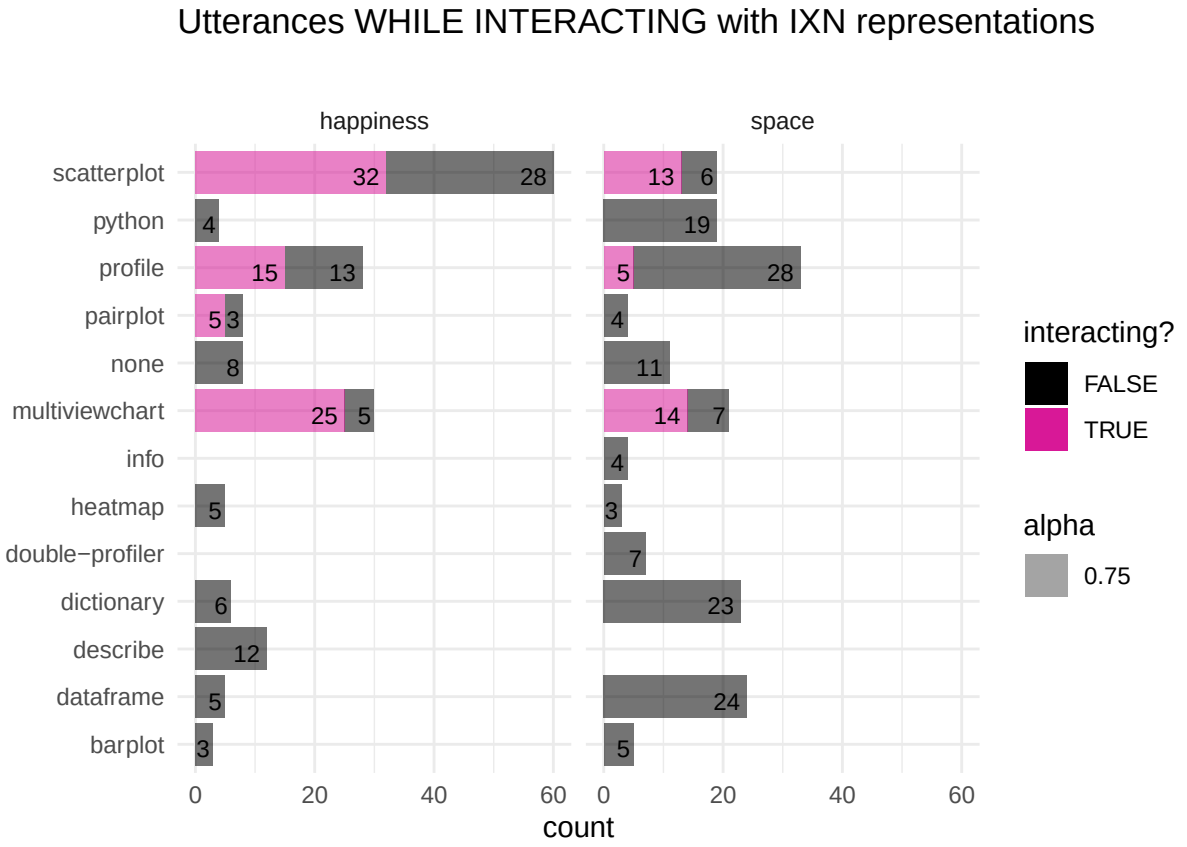



```
#DF SUMMARIZED BY TASK + DATASET
df_summary <- df_codedrep %>%
  filter(TASK == "ixn") %>%  #can only occur during interactive task
  group_by(DATASET,code_topic,ixn) %>%
  dplyr::summarise( .groups="keep",
    c = n()
  )

##no need to summarize by rep_type because only vis can be ixn
## consider middle level summary of uni vs bi vs multivariate vis


#STACKED BAR BY TASK
ggplot(df_summary, aes(x = DATASET, y=c, fill= ixn)) +
  geom_col() +
  facet_wrap(df_summary$code_topic) +
  # geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_manual(values=c("black","#D81897")) +
  labs( title = "KINDS of utterances made WHILE INTERACTING with ixn visualization",
        subtitle = "",
        x= "DATASET", y = "count") + theme_minimal()
```




```
# + theme(legend.position = "blank")
```

```
df <- df_codedrep %>% dplyr::select(PNUM, TASK, DATASET, code_topic, code_detail, REP, rep_type, relati


df_summary <- df_codedrep %>%
  filter(TASK == "ixn") %>%  #can only occur during interactive task
  group_by(DATASET,REP,ixn) %>%
  summarise( .groups = "keep",
    c = n()
  )

#STACKED BAR BY TASK FACET DATASET
ggplot(df_summary, aes(x = REP, y=c, fill= ixn, alpha = 0.75)) +
  facet_wrap(df_summary$DATASET) +
  geom_col() +
  coord_flip()+
  geom_text(aes(label=c),  alpha = 1, size = 3, hjust = 1.25, vjust = 0.75, position = "stack") +
  scale_fill_manual(values=c("black","#D81897")) +
  # scale_fill_brewer(type="qual", palette = 4) +
  labs( title = "Utterances WHILE INTERACTING with IXN representations",
        subtitle = "",
        x= "", y = "count", fill="interacting?") + theme_minimal()
```




```
# + theme(legend.position = "blank")
```

EXPLORE TELEMETRY REPRESENTATIONS

Representations are computationally-generated visual-spatial artifacts that participants use during the EDA tasks. These include data visualizations, but also tabular code outputs, or other data structures returned by Python code.

NOTE that there are three uses of representations we explore in this project:

1. Representations *used or referenced* during the course of making an utterance. These representations are explored in the [DETAIL of] Utterances sections (above).
2. Representations *used interactively* during the course of making an utterance. These include charts to which an interaction encoding is added via Altair Express, but *only* when they are *actively engaged* in while making an utterance. These are also explored in the [DETAIL of] Utterances sections (above).
3. Representations created by participants during the analysis task. These are not necessarily associated with utterances. These are the representations we explore in this section, and they are derived from log-telemetry data.

In the following subsections we start by exploring the distribution of *kinds of representations* participants generated based on TASK, DATASET, and PARTICIPANT. [TELEMETRY] Representations

RQ: How many representations did participants generate? Of what kinds? At what times during the process of analysis? (These are the representations coming from telemetry. Not connected to utterances explicitly. Code only reps are excluded, but tabular output of code cells (eg. .info(), .describe(), etc. included)

[Number of] Representations

by TASK

```
print("BY TASK")
```

[1] “BY TASK”

```
freq(df_telemetry$TASK,
     cumul      = FALSE,
     headings   = FALSE,
     report.nas = FALSE,
     plain.ascii = FALSE)
```

	Freq	%
static	232	46.03
ixn	272	53.97
Total	504	100.00

by TASK and DATASET

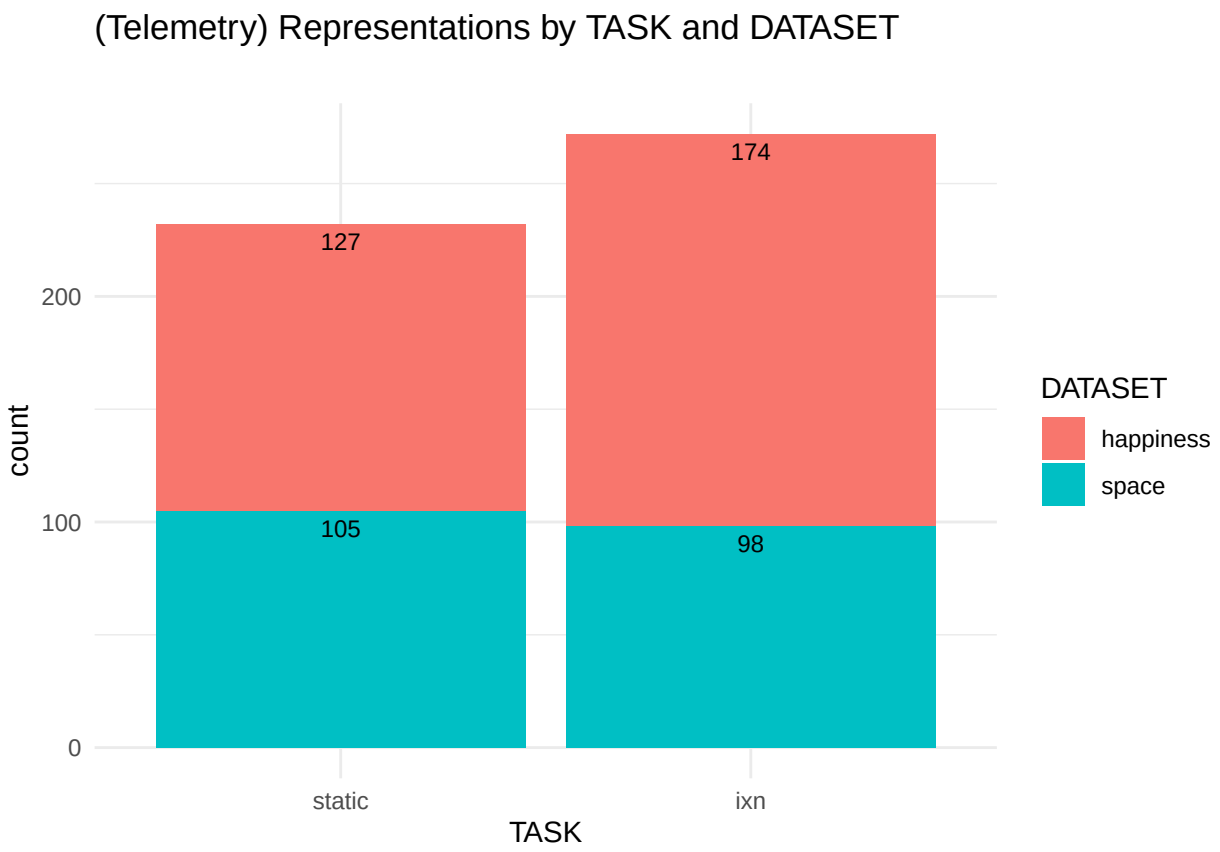
```
#COUNT BY TASK AND DATASET
ctable(x = df_telemetry$TASK,
       y = df_telemetry$DATASET,
       prop = "t")
```

```
## Cross-Tabulation, Total Proportions
## TASK * DATASET
## Data Frame: df_telemetry
##
## -------- --------- ------------- ------------- --------------
##             DATASET     happiness         space          Total
##     TASK
##   static             127 (25.2%)   105 (20.8%)   232 ( 46.0%)
##      ixn             174 (34.5%)    98 (19.4%)   272 ( 54.0%)
##    Total             301 (59.7%)   203 (40.3%)   504 (100.0%)
## -------- --------- ------------- ------------- --------------
```

```
#DF SUMMARIZED BY TASK + DATASET
df_summary <- df_telemetry %>%
  group_by(TASK,DATASET) %>%
  dplyr::summarise(
    c = n()
  )
```

```
## `summarise()` has grouped output by 'TASK'. You can override using the
## `.groups` argument.
```

```
#STACKED BAR BY TASK AND DATASET
ggplot(df_summary, aes(x = TASK, y=c, fill= DATASET)) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  # scale_fill_brewer(type="qual", palette = 4) +
  labs( title = "(Telemetry) Representations by TASK and DATASET",
        subtitle = "",
        x= "TASK", y = "count") + theme_minimal()
```




```
# + theme(legend.position = "blank")
```

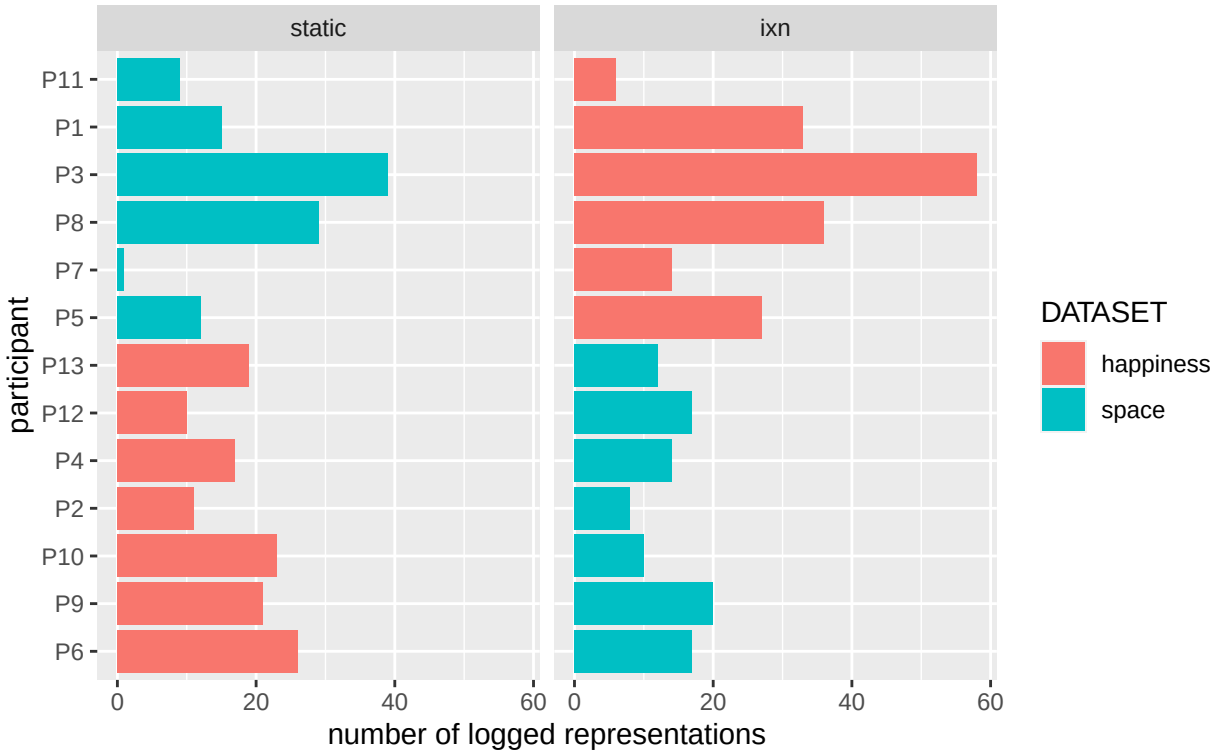
by PARTICIPANT

```
#COUNT BY PARTICIPANT AND TASK
ctable(x = df_telemetry$PNUM,
       y = df_telemetry$TASK,
       prop = "r")
```

Cross-Tabulation, Row Proportions
PNUM * TASK
Data Frame: df_telemetry

TASK	static	ixn	Total
PNUM			
P6	26 (60.5%)	17 (39.5%)	43 (100.0%)
P9	21 (51.2%)	20 (48.8%)	41 (100.0%)
P10	23 (69.7%)	10 (30.3%)	33 (100.0%)
P2	11 (57.9%)	8 (42.1%)	19 (100.0%)
P4	17 (54.8%)	14 (45.2%)	31 (100.0%)
P12	10 (37.0%)	17 (63.0%)	27 (100.0%)
P13	19 (61.3%)	12 (38.7%)	31 (100.0%)
P5	12 (30.8%)	27 (69.2%)	39 (100.0%)
P7	1 (6.7%)	14 (93.3%)	15 (100.0%)
P8	29 (44.6%)	36 (55.4%)	65 (100.0%)
P3	39 (40.2%)	58 (59.8%)	97 (100.0%)
P1	15 (31.2%)	33 (68.8%)	48 (100.0%)
P11	9 (60.0%)	6 (40.0%)	15 (100.0%)
Total	232 (46.0%)	272 (54.0%)	504 (100.0%)

```
#UTTERANCES by PARTICPANT facet TASK color DATASET
gf_bar( PNUM ~., fill = ~ DATASET, data = df_telemetry) %>%
  gf_facet_grid(.~TASK) +
  labs(
    title = "(Telemetry) Representations by Participant, Dataset and Task",
    subtitle = "",
    x = "number of logged representations",
    y = "participant",
    fill = "DATASET"
  )
```



through TIME

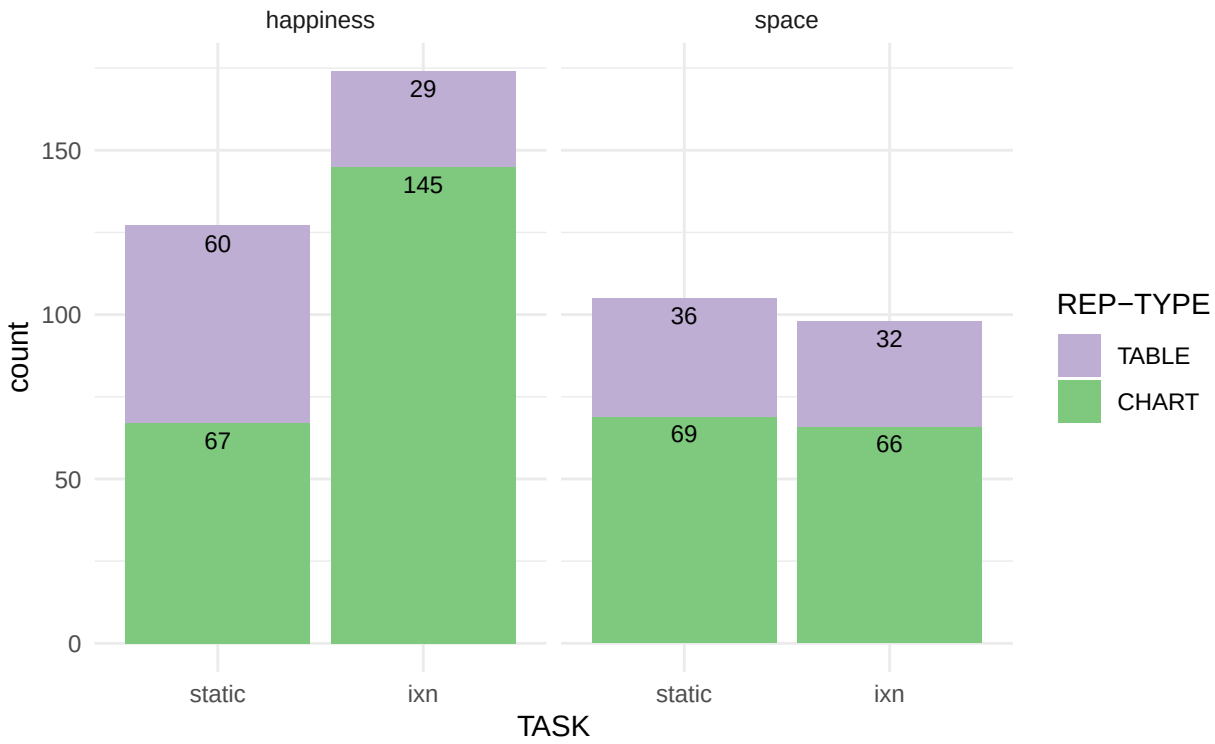
```
##TODO-INVESTIGATE TELEMETRY ROWS MISSING TIMESTAMP

# #DOTPLOT-PARTICIPANT-facet-TASK-color-DATASET
# ggplot(df_telemetry, aes(x=time_elapsed, y = PNUM, color = DATASET)) +
#   geom_point(alpha=0.5, size=3) +
#   facet_grid(df_telemetry$TASK) +
#   # scale_color_brewer(type="qual", palette = 3) +
#   theme_minimal() + labs(
#     title = "Participant Utterances over timecourse of Task",
#     subtitle = "",
#     x= "timecourse of task (seconds)", y = "Participant",
#     color = "Dataset"
#   )
#
#
# #HISTOGRAMS BY TASK
# ggplot(df_telemetry, aes(x = time_elapsed)) +
#   geom_histogram(binwidth = 30,aes(y=..density..)) +
#   geom_density()+
#   facet_grid(df_telemetry$TASK) +
#   theme_minimal() + labs(
#     title = "(Telemetry) Representations over timecourse of Task",
```

```
#     x= "timecourse of task (seconds)", y = "frequency of utterances",
#   ) + theme_minimal() + theme(legend.position = "blank")
```

[CATEGORY of] Representation

by TASK and DATASET

```
#DF SUMMARIZED BY TASK + DATASET
df_summary <- df_telemetry %>%
  group_by(rep_type, TASK,DATASET) %>%
  dplyr::summarise(
    c = n()
  )

#PAPER FIGURE HERE
#STACKED BAR BY TASK FACET DATASET
(p <- ggplot(df_summary, aes(x = TASK, y=c, fill= fct_rev(rep_type))) +
  facet_wrap(df_summary$DATASET) +
  geom_col() +
  geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
  scale_fill_brewer(type="qual", palette = 1, direction = -1) +
  labs( title = "(TELEMETRY) REPRESENTATIONS by TASK and DATASET",
        subtitle = "",
        x= "TASK", y = "count", fill="REP-TYPE") + theme_minimal()
)
```




```
ggsave(p, file="figures/TELEMETRY_classes_by_factors.png")
# + theme(legend.position = "blank")

# #HIGH LEVEL
# gf_bar(  ~ rep_type, fill = ~rep_type, position="stack", data = df_telemetry) %>%
#   gf_facet_grid( DATASET ~ TASK) +
#   scale_fill_brewer(type="qual", palette = 1) +
#   theme_minimal() + labs(
#     title = "[TELEMETRY]  Representations HIGH"
#   )
```

by PARTICIPANT

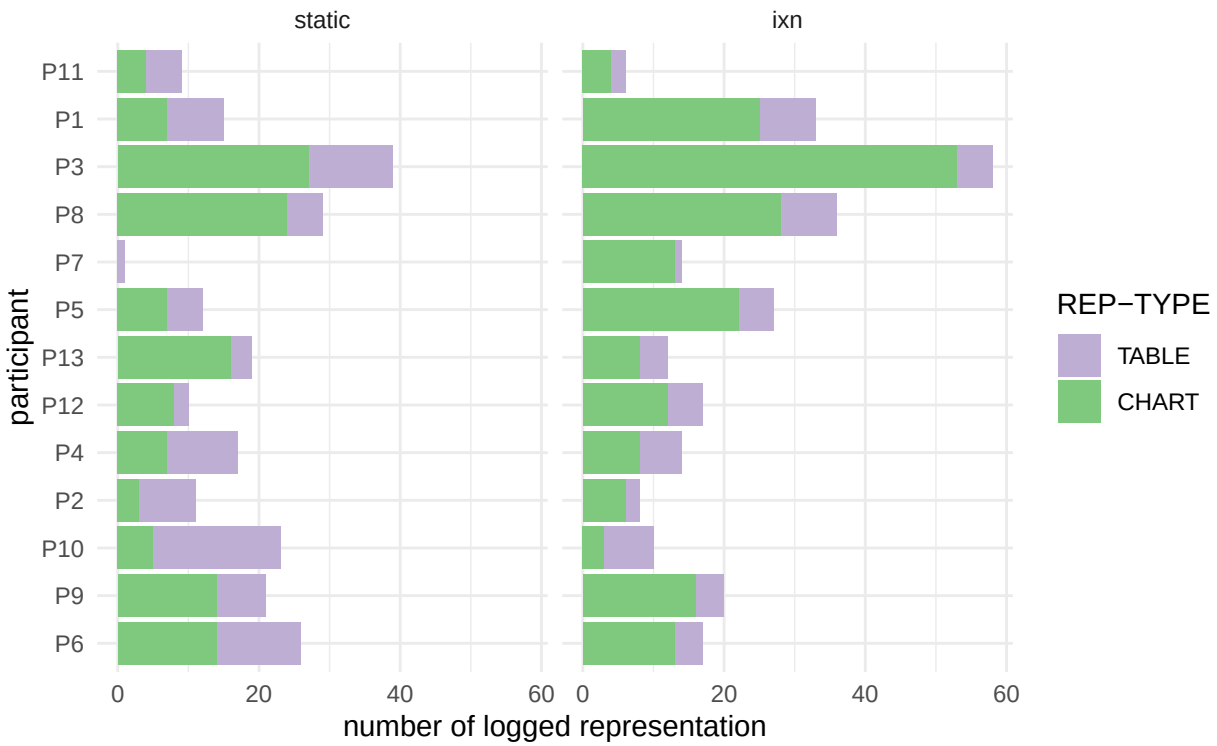
```
#COUNT BY PARTICIPANT
ctable(x = df_telemetry$PNUM,
       y = df_telemetry$rep_type,
       prop = "r")
```

Cross-Tabulation, Row Proportions
PNUM * rep_type
Data Frame: df_telemetry

rep_type	CHART	TABLE	Total

PNUM			
P6	27 (62.8%)	16 (37.2%)	43 (100.0%)
P9	30 (73.2%)	11 (26.8%)	41 (100.0%)
P10	8 (24.2%)	25 (75.8%)	33 (100.0%)
P2	9 (47.4%)	10 (52.6%)	19 (100.0%)
P4	15 (48.4%)	16 (51.6%)	31 (100.0%)
P12	20 (74.1%)	7 (25.9%)	27 (100.0%)
P13	24 (77.4%)	7 (22.6%)	31 (100.0%)
P5	29 (74.4%)	10 (25.6%)	39 (100.0%)
P7	13 (86.7%)	2 (13.3%)	15 (100.0%)
P8	52 (80.0%)	13 (20.0%)	65 (100.0%)
P3	80 (82.5%)	17 (17.5%)	97 (100.0%)
P1	32 (66.7%)	16 (33.3%)	48 (100.0%)
P11	8 (53.3%)	7 (46.7%)	15 (100.0%)
Total	347 (68.8%)	157 (31.2%)	504 (100.0%)

```
#CATEGORY by PARTICPANT facet TASK
(p <- gf_bar( PNUM ~., fill = ~ fct_rev(rep_type), data = df_telemetry) %>%
  gf_facet_grid(.~TASK) +
  scale_fill_brewer(type="qual", palette = 1, direction = -1) +
  labs(
    title = "(Telemetry) Representations by Participant, Dataset and Task",
    subtitle = "",
    x = "number of logged representation",
    y = "participant",
    fill = "REP-TYPE"
  ) + theme_minimal())
```




```
ggsave(p, file="figures/TELEMETRY_classes_by_participants.png")
```

[TYPE of] Representation

by TASK and DATASET

```
#COUNT BY TASK
ctable(x = df_telemetry$REP,
       y = df_telemetry$TASK,
       prop = "t")
```

```
## Cross-Tabulation, Total Proportions
## REP * TASK
## Data Frame: df_telemetry
##
## ---------------- ------ ------------ ------------ --------------
##                    TASK       static          ixn          Total
##             REP
##          profile          25 ( 5.0%)   17 ( 3.4%)    42 (  8.3%)
##          barplot           8 ( 1.6%)   35 ( 6.9%)    43 (  8.5%)
##          columns          21 ( 4.2%)    9 ( 1.8%)    30 (  6.0%)
##        dataframe          59 (11.7%)   42 ( 8.3%)   101 ( 20.0%)
##         describe          13 ( 2.6%)    7 ( 1.4%)    20 (  4.0%)
```

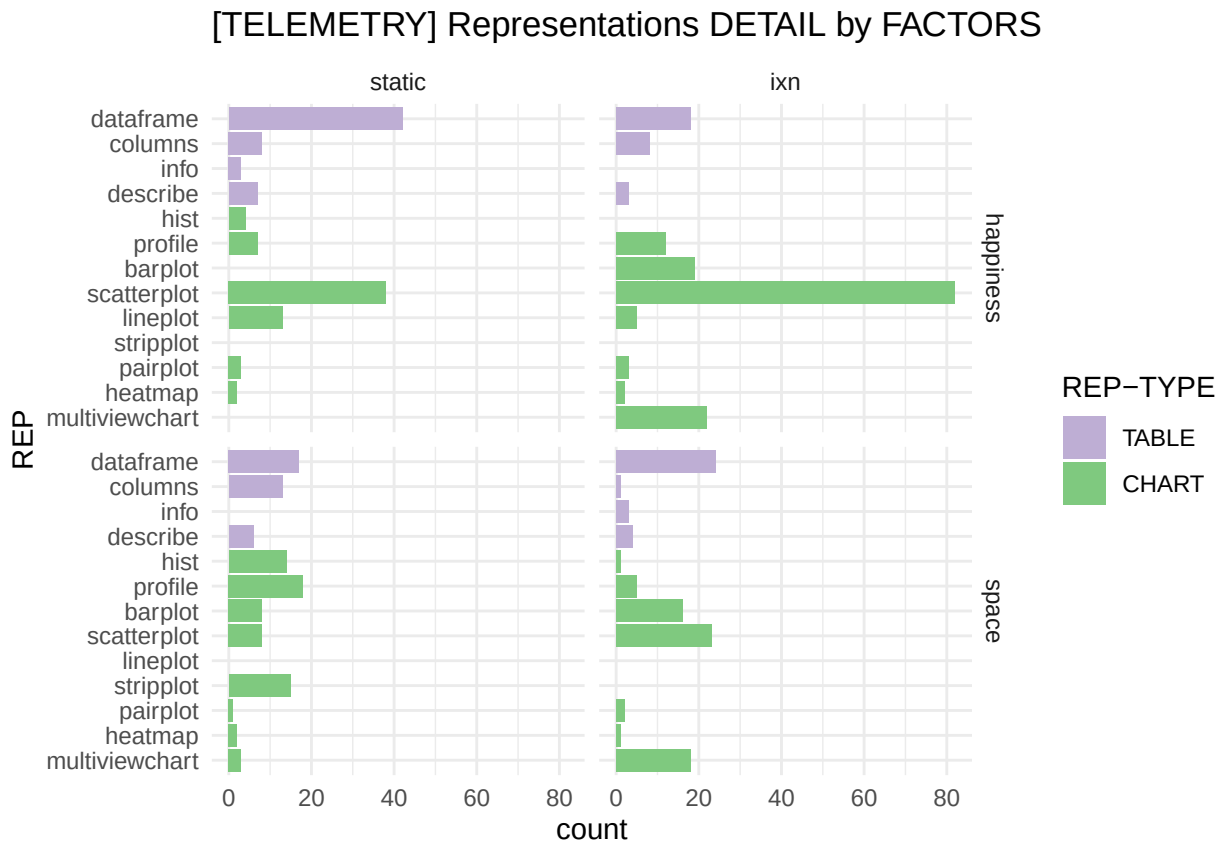
```
##          heatmap            4 ( 0.8%)     3 ( 0.6%)     7 (  1.4%)
##             hist           18 ( 3.6%)     1 ( 0.2%)    19 (  3.8%)
##             info            3 ( 0.6%)     3 ( 0.6%)     6 (  1.2%)
##         lineplot           13 ( 2.6%)     5 ( 1.0%)    18 (  3.6%)
##   multiviewchart            3 ( 0.6%)    40 ( 7.9%)    43 (  8.5%)
##         pairplot            4 ( 0.8%)     5 ( 1.0%)     9 (  1.8%)
##      scatterplot           46 ( 9.1%)   105 (20.8%)   151 ( 30.0%)
##        stripplot           15 ( 3.0%)     0 ( 0.0%)    15 (  3.0%)
##            Total          232 (46.0%)   272 (54.0%)   504 (100.0%)
## ---------------- ------ ------------- ------------- --------------
```

```
#COUNT BY DATASET
ctable(x = df_telemetry$REP,
       y = df_telemetry$DATASET,
       prop = "t")
```

```
## Cross-Tabulation, Total Proportions
## REP * DATASET
## Data Frame: df_telemetry
##
## ---------------- --------- ------------- ------------- --------------
##                     DATASET     happiness         space          Total
##             REP
##          profile              19 ( 3.8%)    23 ( 4.6%)    42 (  8.3%)
##          barplot              19 ( 3.8%)    24 ( 4.8%)    43 (  8.5%)
##          columns              16 ( 3.2%)    14 ( 2.8%)    30 (  6.0%)
##        dataframe              60 (11.9%)    41 ( 8.1%)   101 ( 20.0%)
##         describe              10 ( 2.0%)    10 ( 2.0%)    20 (  4.0%)
##          heatmap               4 ( 0.8%)     3 ( 0.6%)     7 (  1.4%)
##             hist               4 ( 0.8%)    15 ( 3.0%)    19 (  3.8%)
##             info               3 ( 0.6%)     3 ( 0.6%)     6 (  1.2%)
##         lineplot              18 ( 3.6%)     0 ( 0.0%)    18 (  3.6%)
##   multiviewchart              22 ( 4.4%)    21 ( 4.2%)    43 (  8.5%)
##         pairplot               6 ( 1.2%)     3 ( 0.6%)     9 (  1.8%)
##      scatterplot             120 (23.8%)    31 ( 6.2%)   151 ( 30.0%)
##        stripplot               0 ( 0.0%)    15 ( 3.0%)    15 (  3.0%)
##            Total             301 (59.7%)   203 (40.3%)   504 (100.0%)
## ---------------- --------- ------------- ------------- --------------
```

```
#PAPER FIGURE HERE
#DETAIL
(p <- gf_bar(  ~ REP, fill = ~fct_rev(rep_type), data = df_telemetry) %>%
  gf_facet_grid( DATASET ~ TASK) +
  scale_fill_brewer(type="qual", palette = 1, direction = -1) +
  coord_flip() +
  scale_x_discrete(limits = c(
                           "multiviewchart",
                           "heatmap",
                           "pairplot",
                           "stripplot",
                           "lineplot",
                           "scatterplot",
                           "barplot",
```

```
                            "profile",
                            "hist",
                            "describe",
                            "info",
                            "columns",
                            "dataframe"

  ))+
  theme_minimal() + labs(
    title = "[TELEMETRY] Representations DETAIL by FACTORS",
    fill = "REP-TYPE"
))
```

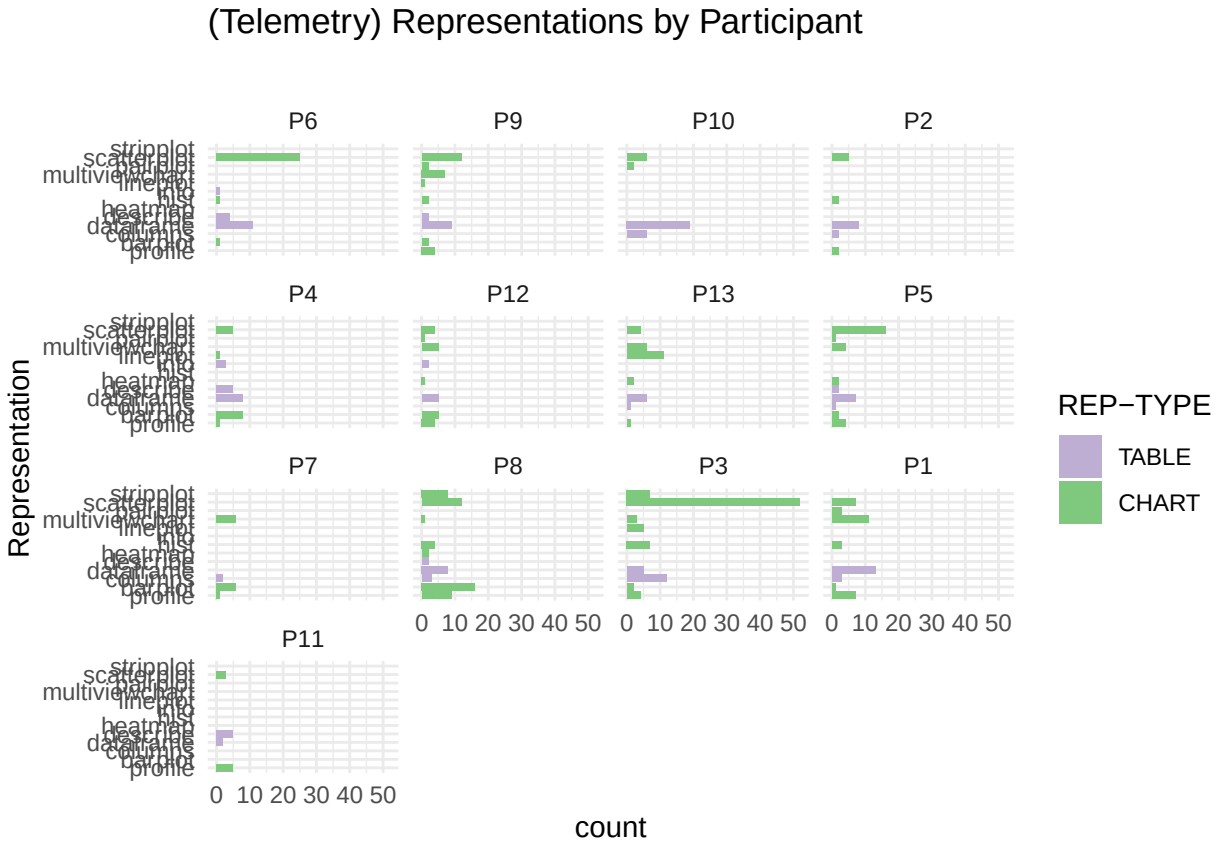



```
ggsave(p, file="figures/TELEMETRY_detail_by_factors.png", width = 7, height = 5, units ="in")


#flipped
# gf_bar(  ~ REP, fill = ~fct_rev(rep_type), data = df_telemetry) %>%
#   gf_facet_grid( TASK ~ DATASET) +
#   scale_fill_brewer(type="qual", palette = 1, direction = -1) +
#   coord_flip() +
#   scale_x_discrete(limits = c(
#                             "multiviewchart",
#                             "heatmap",
#                             "pairplot",
```

```
#                             "stripplot",
#                             "lineplot",
#                             "scatterplot",
#                             "barplot",
#                             "profile",
#                             "hist",
#                             "describe",
#                             "info",
#                             "columns",
#                             "dataframe"
#
#   ))+
#   theme_minimal() + labs(
#     title = "[TELEMETRY]  Representations DETAIL by FACTORS"
# )

# #DF BY REP
# df_summary <- df_telemetry %>%
#   group_by(REP) %>%
#   dplyr::summarise(
#     c = n()
#   )

# #STACKED BAR
# ggplot(df_summary, aes(x = REP, y=c, fill= REP)) +
#   geom_col() +
#   geom_text(aes(label=c), size = 3, hjust = 0.5, vjust = 1.5, position = "stack") +
#   # scale_fill_brewer(type="qual", palette = 4) +
#   coord_flip()+
#   labs( title = "(Telemetry) Representations DETAIL",
#         subtitle = "",
#         x= "TASK", y = "count") + theme_minimal()
# # + theme(legend.position = "blank")
```

by PARTICIPANT

```
#REPRESENTATIONS by PARTICPANT
(p <- gf_bar( REP ~., fill = ~ fct_rev(rep_type), data = df_telemetry) %>%
  gf_facet_wrap(~ PNUM ) +
  scale_fill_brewer(type="qual", palette = 1, direction = -1) +
  labs(
    title = "(Telemetry) Representations by Participant",
    subtitle = "",
    y = "Representation",
    fill = "REP-TYPE"
  ) + theme_minimal()
)
```



MODELLING

(limited) Data modelling to explore post-hoc hypotheses.

```
#DEFINE DATAFRAME
df <- df_coded %>% select(pid, uid, TASK, DATASET)

# #MOSAIC PLOT
# mosaic(formula = ~DATASET + TASK,
#        data = df,
#        main = "Proportion of Utterances by TASK and DATASET",
#        sub = "u = 734 utterance-codes",
#        labeling = labeling_values,
#        labeling_args = list(set_varnames = c(graph = "TASK",
#                             datset = "DATASET")))
```

Predicting NUMBER of UTTERANCES

How much variance in number of utterances is explained DATASET, TASK and PARTICIPANT?

OLS Mixed Effects Model

```
#DEFINE DATAFRAME
df <- df_coded %>% group_by(pid, DATASET, TASK) %>%
  dplyr::summarise( .groups = "keep",
    n_utterances = n()
  )

#NUMBER UTTERANCES predicted by DATASET + TASK | participatnt--> MIXED LINEAR REGRESSION
print("LMER, UTTERANCES ~ DATASET + TASK")
```

```
## [1] "LMER, UTTERANCES ~ DATASET + TASK"
```

```
mm1 <- lmer(n_utterances ~ DATASET + TASK+ (1|pid), data = df)
paste("Model")
```

```
## [1] "Model"
```

```
summ(mm1)
```

Observations	26
Dependent variable	n_utterances
Type	Mixed effects linear regression

AIC	198.45
BIC	204.74
Pseudo-R^2 (fixed effects)	0.12
Pseudo-R^2 (total)	0.68

Fixed Effects					
	Est.	S.E.	t val.	d.f.	p
(Intercept)	35.11	4.17	8.42	20.26	0.00
DATASETspace	-8.90	3.33	-2.67	11.00	0.02
TASKixn	-4.24	3.33	-1.27	11.00	0.23

p values calculated using Satterthwaite d.f.

Random Effects		
Group	Parameter	Std. Dev.
pid	(Intercept)	11.12
Residual		8.47

```
paste("Partition Variance")
```

```
## [1] "Partition Variance"
```

Grouping Variables		
Group	# groups	ICC
pid	13	0.63

```
anova(mm1)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##         Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## DATASET 512.37  512.37     1    11  7.1424 0.0217 *
## TASK    116.06  116.06     1    11  1.6179 0.2296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
paste("Confidence Interval on Parameter Estimates")
```

```
## [1] "Confidence Interval on Parameter Estimates"
```

```
confint(mm1)
```

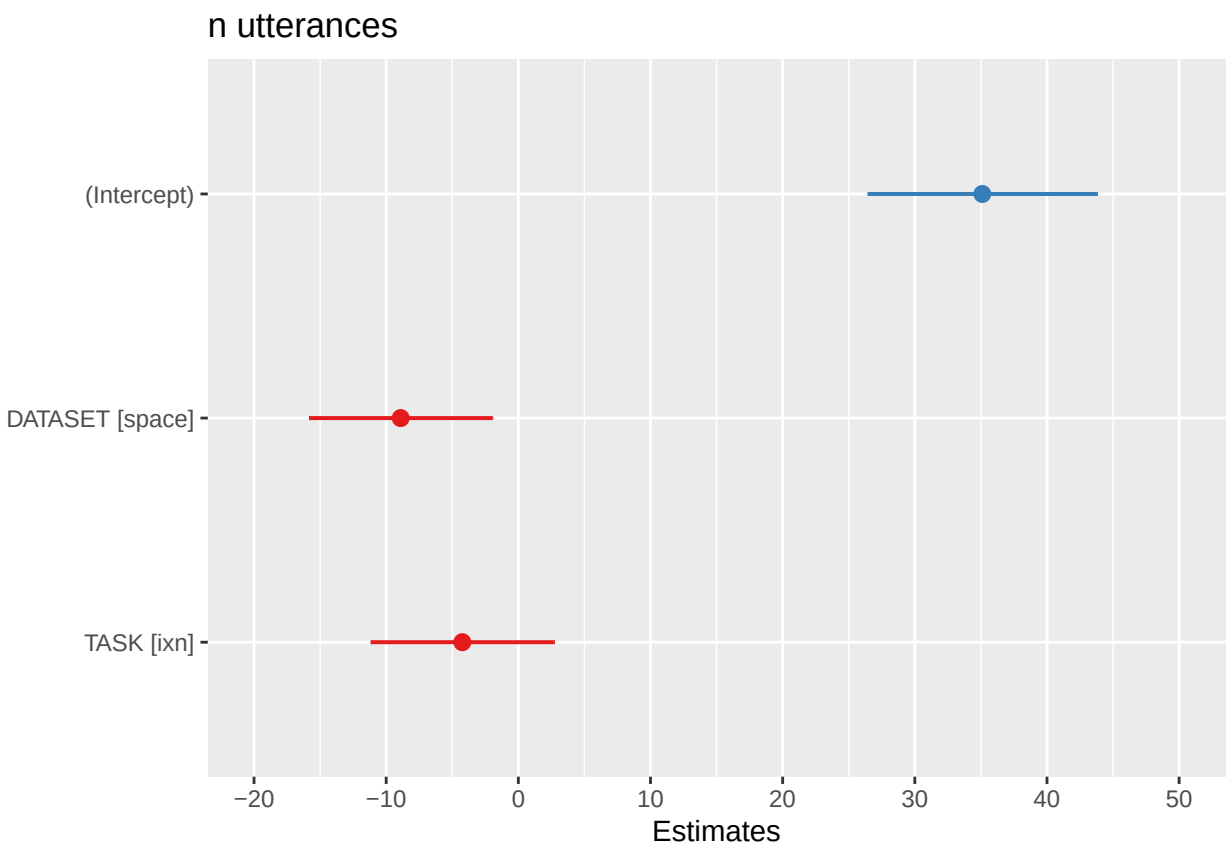
```
## Computing profile confidence intervals ...
```

```
##                    2.5 %    97.5 %
## .sig01          5.839640 18.047250
## .sigma          5.540684 12.104462
## (Intercept)    26.986050 43.233730
## DATASETspace -15.384322 -2.425202
## TASKixn      -10.717656  2.241465
```

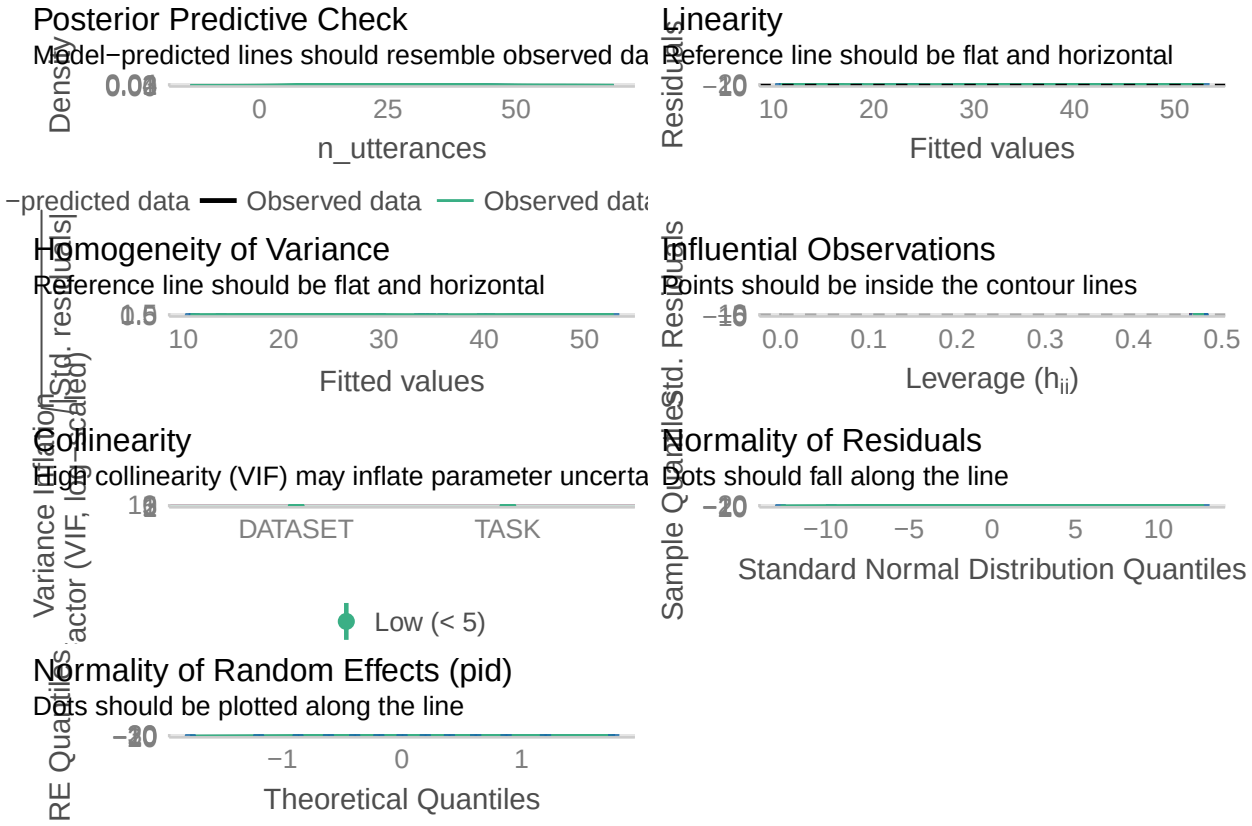
```
report(mm1) #sanity check
```

```
## We fitted a linear mixed model (estimated using REML and nloptwrap optimizer)
## to predict n_utterances with DATASET and TASK (formula: n_utterances ~ DATASET
## + TASK). The model included pid as random effect (formula: ~1 | pid). The
## model's total explanatory power is substantial (conditional R2 = 0.68) and the
## part related to the fixed effects alone (marginal R2) is of 0.12. The model's
## intercept, corresponding to DATASET = happiness and TASK = static, is at 35.11
## (95% CI [26.44, 43.78], t(21) = 8.42, p < .001). Within this model:
##
##   - The effect of DATASET [space] is statistically significant and negative (beta
## = -8.90, 95% CI [-15.83, -1.98], t(21) = -2.67, p = 0.014; Std. beta = -0.61,
## 95% CI [-1.09, -0.14])
##   - The effect of TASK [ixn] is statistically non-significant and negative (beta
## = -4.24, 95% CI [-11.17, 2.69], t(21) = -1.27, p = 0.217; Std. beta = -0.29,
## 95% CI [-0.77, 0.19])
##
## Standardized parameters were obtained by fitting the model on a standardized
## version of the dataset. 95% Confidence Intervals (CIs) and p-values were
## computed using a Wald t-distribution approximation.
```

```
plot_model(mm1,  show.intercept = TRUE)
```




```
check_model(mm1)
```




```
#NUMBER UTTERANCES predicted by DATASET * TASK  | participatnt--> MIXED LINEAR REGRESSION
print("LMER, UTTERANCES ~ DATASET X TASK")
```

```
## [1] "LMER, UTTERANCES ~ DATASET X TASK"
```

```
mm2 <- lmer(n_utterances ~ DATASET * TASK + (1|pid), data = df)
paste("Model")
```

```
## [1] "Model"
```

```
summ(mm2)
```

Observations	26
Dependent variable	n_utterances
Type	Mixed effects linear regression

AIC	192.74
BIC	200.29
Pseudo-R^2 (fixed effects)	0.14
Pseudo-R^2 (total)	0.70

Fixed Effects					
	Est.	S.E.	t val.	d.f.	p
(Intercept)	37.57	5.37	7.00	15.55	0.00
DATASETspace	-14.24	7.90	-1.80	15.55	0.09
TASKixn	-9.57	7.90	-1.21	15.55	0.24
DATASETspace:TASKixn	10.67	14.32	0.74	11.00	0.47

p values calculated using Satterthwaite d.f.

Random Effects		
Group	Parameter	Std. Dev.
pid	(Intercept)	11.39
Residual		8.47

Grouping Variables		
Group	# groups	ICC
pid	13	0.64

```
paste("Partition Variance")
```

```
## [1] "Partition Variance"
```

```
anova(mm2)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##              Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
## DATASET      512.37  512.37     1    11  7.1424 0.0217 *
## TASK         116.06  116.06     1    11  1.6179 0.2296
## DATASET:TASK  39.80   39.80     1    11  0.5549 0.4720
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
paste("Confidence Interval on Parameter Estimates")
```

```
## [1] "Confidence Interval on Parameter Estimates"
```

```
confint(mm2)
```

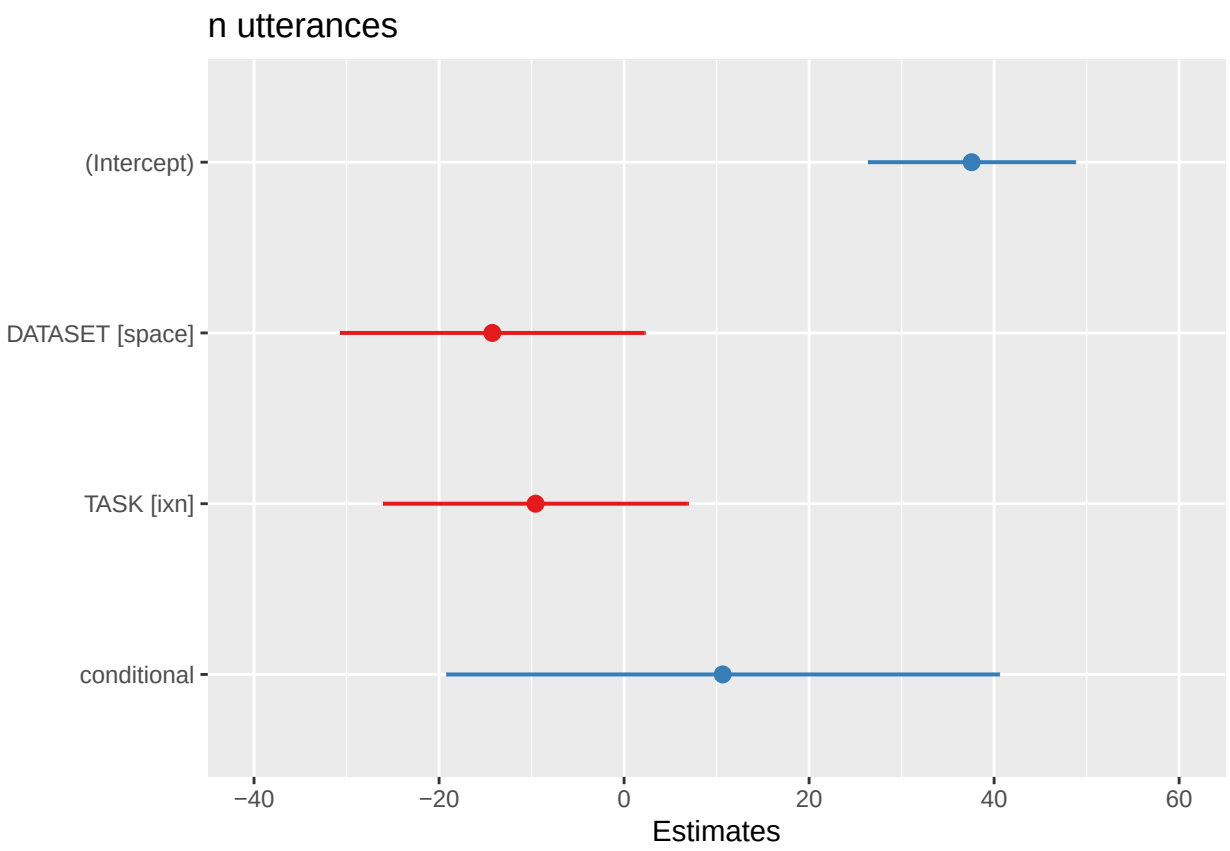
```
## Computing profile confidence intervals ...
```

```
##                             2.5 %    97.5 %
## .sig01                   5.468708 17.569089
## .sigma                   5.540710 12.104731
## (Intercept)             27.350579 47.792279
## DATASETspace           -29.282779  0.806589
## TASKixn                -24.616112  5.473256
## DATASETspace:TASKixn   -17.180459 38.513793
```

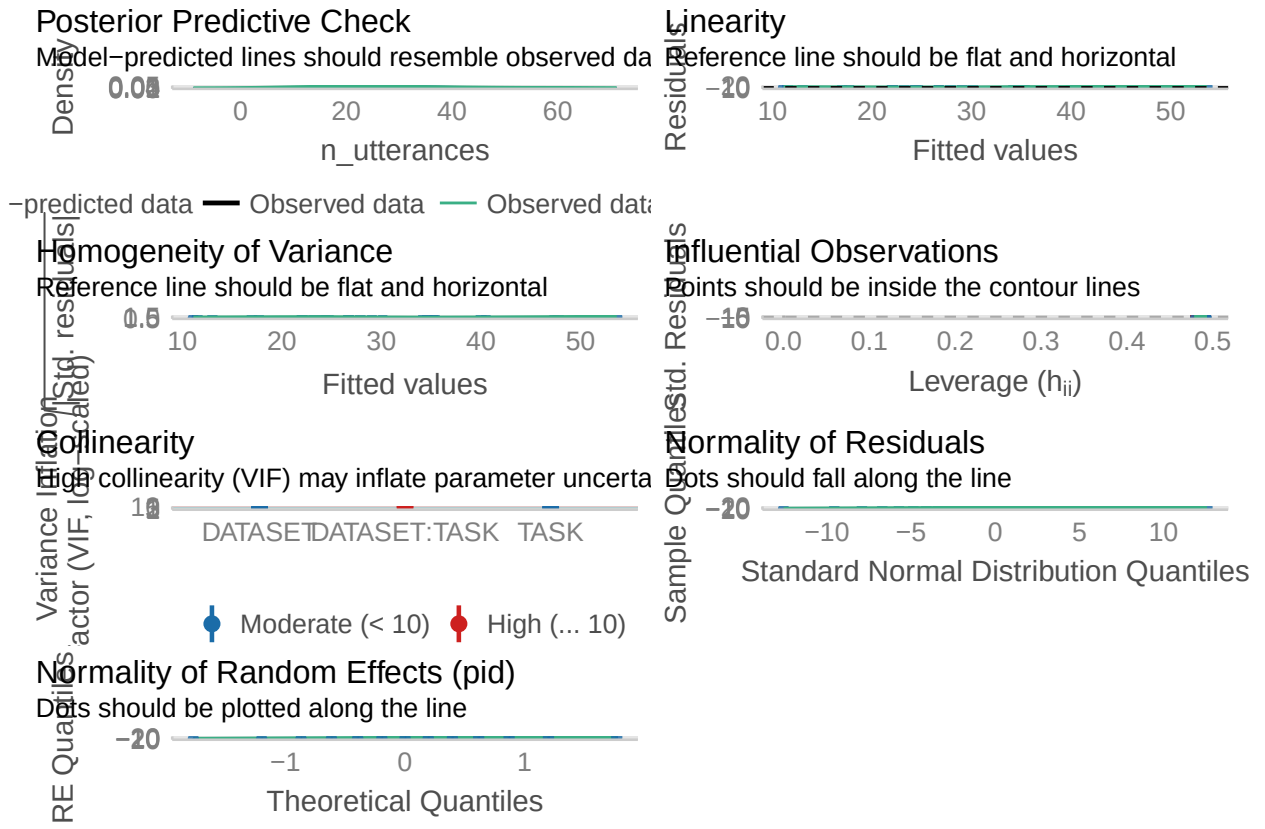
```
report(mm2) #sanity check
```

```
## We fitted a linear mixed model (estimated using REML and nloptwrap optimizer)
## to predict n_utterances with DATASET and TASK (formula: n_utterances ~ DATASET
## * TASK). The model included pid as random effect (formula: ~1 | pid). The
## model's total explanatory power is substantial (conditional R2 = 0.70) and the
## part related to the fixed effects alone (marginal R2) is of 0.14. The model's
## intercept, corresponding to DATASET = happiness and TASK = static, is at 37.57
## (95% CI [26.38, 48.76], t(20) = 7.00, p < .001). Within this model:
##
##   - The effect of DATASET [space] is statistically non-significant and negative
## (beta = -14.24, 95% CI [-30.71, 2.24], t(20) = -1.80, p = 0.086; Std. beta =
## -0.98, 95% CI [-2.11, 0.15])
##   - The effect of TASK [ixn] is statistically non-significant and negative (beta
## = -9.57, 95% CI [-26.04, 6.90], t(20) = -1.21, p = 0.240; Std. beta = -0.66,
## 95% CI [-1.79, 0.47])
##   - The effect of DATASET [space] × TASK [ixn] is statistically non-significant
## and positive (beta = 10.67, 95% CI [-19.20, 40.54], t(20) = 0.74, p = 0.465;
## Std. beta = 0.73, 95% CI [-1.32, 2.79])
##
## Standardized parameters were obtained by fitting the model on a standardized
## version of the dataset. 95% Confidence Intervals (CIs) and p-values were
## computed using a Wald t-distribution approximation.
```

```
plot_model(mm2,  show.intercept = TRUE)
```



```
check_model(mm2)
```




```
### POISSON Mixed Effects Models
### ARF poisson is recommended over OLS regression for count data
### BUT they are challenging to interpret (log odds) and the estimates need to be translated (logodds?)
### NOT innapropriate to use OLS instead
#
# #NUMBER UTTERANCES predicted by TASK + DATASET  | participatnt--> POISSON MIXED LINEAR REGRESSION
# print("POISSON-MER, UTTERANCES ~ DATASET + TASK")
# pmm1 <- glmer(n_utterances ~ TASK + DATASET + (1|pid), data = df, family = "poisson")
# paste("Model")
# summ(pmm1)
# paste("Partition Variance")
# anova(pmm1)
# paste("Confidence Interval on Parameter Estimates")
# confint(pmm1)
# report(pmm1) #sanity check
# plot_model(pmm1,  show.intercept = TRUE)
# check_model(pmm1)
#
# #NUMBER UTTERANCES predicted by TASK X DATASET  | participatnt--> POISSON MIXED LINEAR REGRESSION
# print("POISSON-MER, UTTERANCES ~ DATASET X TASK")
# pmm2 <- glmer(n_utterances ~ TASK * DATASET + (1|pid), data = df, family = "poisson")
# paste("Model")
# summ(pmm2)
```

```
# paste("Partition Variance")
# anova(pmm2)
# paste("Confidence Interval on Parameter Estimates")
# confint(pmm2)
# report(pmm2) #sanity check
# plot_model(pmm2,  show.intercept = TRUE)
# check_model(pmm2)
```

REPRODUCIBILITY

```
#DOC R and package versions
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur ... 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lmerTest_3.1-3       lme4_1.1-31          Matrix_1.5-3
##  [4] sjPlot_2.8.11        see_0.7.5.1          report_0.5.7.1
##  [7] parameters_0.20.2.14 performance_0.10.2.7 modelbased_0.8.6.3
## [10] insight_0.19.1.2     effectsize_0.8.3.6   datawizard_0.7.0.5
## [13] correlation_0.8.3.3  bayestestR_0.13.0.10 easystats_0.6.0.8
## [16] ggstatsplot_0.10.0   ggformula_0.10.2     ggridges_0.5.4
## [19] scales_1.2.1         ggstance_0.3.6       kableExtra_1.3.4
## [22] lubridate_1.9.2      jtools_2.2.1         summarytools_1.0.1
## [25] forcats_1.0.0        stringr_1.5.0        dplyr_1.1.0
## [28] purrr_1.0.1          readr_2.1.4          tidyr_1.3.0
## [31] tibble_3.1.8         tidyverse_1.3.2      Hmisc_4.8-0
## [34] ggplot2_3.4.1        Formula_1.2-4        survival_3.5-3
## [37] lattice_0.20-45
##
## loaded via a namespace (and not attached):
##   [1] utf8_1.2.3              tidyselect_1.2.0        htmlwidgets_1.6.1
##   [4] grid_4.2.2              munsell_0.5.0           codetools_0.2-19
##   [7] ragg_1.2.5              interp_1.1-3            withr_2.5.0
##  [10] colorspace_2.1-0        highr_0.10              knitr_1.42
##  [13] rstudioapi_0.14         robustbase_0.95-0       labeling_0.4.2
##  [16] emmeans_1.8.4-1         polyclip_1.10-4         bit64_4.0.5
##  [19] farver_2.1.1            opdisDownsampling_0.8.2 coda_0.19-4
```

```
##  [22] vctrs_0.5.2             generics_0.1.3          twosamples_2.0.0
##  [25] TH.data_1.1-1           xfun_0.37               timechange_0.2.0
##  [28] doParallel_1.0.17       R6_2.5.1                bitops_1.0-7
##  [31] assertthat_0.2.1        vroom_1.6.1             multcomp_1.4-22
##  [34] nnet_7.3-18             googlesheets4_1.0.1     gtable_0.3.1
##  [37] benchmarkmeData_1.0.4   sandwich_3.0-2          qqplotr_0.0.6
##  [40] rlang_1.0.6             zeallot_0.1.0           systemfonts_1.0.4
##  [43] splines_4.2.2           gargle_1.3.0            broom_1.0.3
##  [46] rapportools_1.1         mosaicCore_0.9.2.1      checkmate_2.1.0
##  [49] yaml_2.3.7              reshape2_1.4.4          modelr_0.1.10
##  [52] backports_1.4.1         tools_4.2.2             tcltk_4.2.2
##  [55] ellipsis_0.3.2          RColorBrewer_1.1-3      Rcpp_1.0.10
##  [58] plyr_1.8.8              base64enc_0.1-3         rpart_4.1.19
##  [61] deldir_1.0-6            zoo_1.8-11              haven_2.5.1
##  [64] ggrepel_0.9.3           cluster_2.1.4           fs_1.6.1
##  [67] magrittr_2.0.3          data.table_1.14.2       magick_2.7.3
##  [70] reprex_2.0.2            googledrive_2.0.0       mvtnorm_1.1-3
##  [73] sjmisc_2.8.9            matrixStats_0.63.0      hms_1.1.2
##  [76] patchwork_1.1.2         evaluate_0.20           xtable_1.8-4
##  [79] sjstats_0.18.2          jpeg_0.1-10             readxl_1.4.2
##  [82] gridExtra_2.3           ggeffects_1.1.5         compiler_4.2.2
##  [85] crayon_1.5.2            minqa_1.2.5             htmltools_0.5.4
##  [88] mgcv_1.8-41             tzdb_0.3.0              DBI_1.1.3
##  [91] tweenr_2.0.2            sjlabelled_1.2.0        dbplyr_2.3.0
##  [94] MASS_7.3-58.2           boot_1.3-28.1           cli_3.6.0
##  [97] pryr_0.1.6              benchmarkme_1.0.8       qqconf_1.3.1
## [100] parallel_4.2.2          pkgconfig_2.0.3         statsExpressions_1.4.0
## [103] numDeriv_2016.8-1.1     foreign_0.8-84          xml2_1.3.3
## [106] paletteer_1.5.0         foreach_1.5.2           memuse_4.2-3
## [109] svglite_2.1.1           webshot_0.5.4           estimability_1.4.1
## [112] rvest_1.0.3             snakecase_0.11.0        digest_0.6.31
## [115] pracma_2.4.2            rmarkdown_2.20          cellranger_1.1.0
## [118] htmlTable_2.4.1         nloptr_2.0.3            lifecycle_1.0.3
## [121] nlme_3.1-162            jsonlite_1.8.4          viridisLite_0.4.1
## [124] fansi_1.0.4             labelled_2.10.0         pillar_1.8.1
## [127] DEoptimR_1.0-11         fastmap_1.1.0           httr_1.4.4
## [130] glue_1.6.2              png_0.1-8               iterators_1.0.14
## [133] pander_0.6.5            bit_4.0.5               ggforce_0.4.1
## [136] stringi_1.7.12          rematch2_2.1.2          textshaping_0.3.6
## [139] latticeExtra_0.6-30     caTools_1.18.2
```

```
#CITE R
citation()
```

```
##
## To cite R in publications use:
##
##   R Core Team (2022). R: A language and environment for statistical
##   computing. R Foundation for Statistical Computing, Vienna, Austria.
##   URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
```

```
##     title = {R: A Language and Environment for Statistical Computing},
##     author = {{R Core Team}},
##     organization = {R Foundation for Statistical Computing},
##     address = {Vienna, Austria},
##     year = {2022},
##     url = {https://www.R-project.org/},
##   }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
## citing R packages.
```