Faux Excel (fxl) Charting in R

Introduction and Rationale

Single-case research uses the visualization of empirical data to infer the effects of environmental manipulations on observed behavior. Rather than perform quantitative tests, data are presented visually using a variety of shared conventions. For instance, data are typically grouped qualitatively by phase or condition and these are separated by visual partitions (i.e., phase change lines).

The construction of phase change lines in spreadsheet software remains tedious and subsequent modifications often require human interaction to edit and adjust. Various scripts and macros exist to streamline these efforts, but the available options are at best incomplete or insecure (most users disable macros by default). It is for these reason that the *fxl* package was designed.

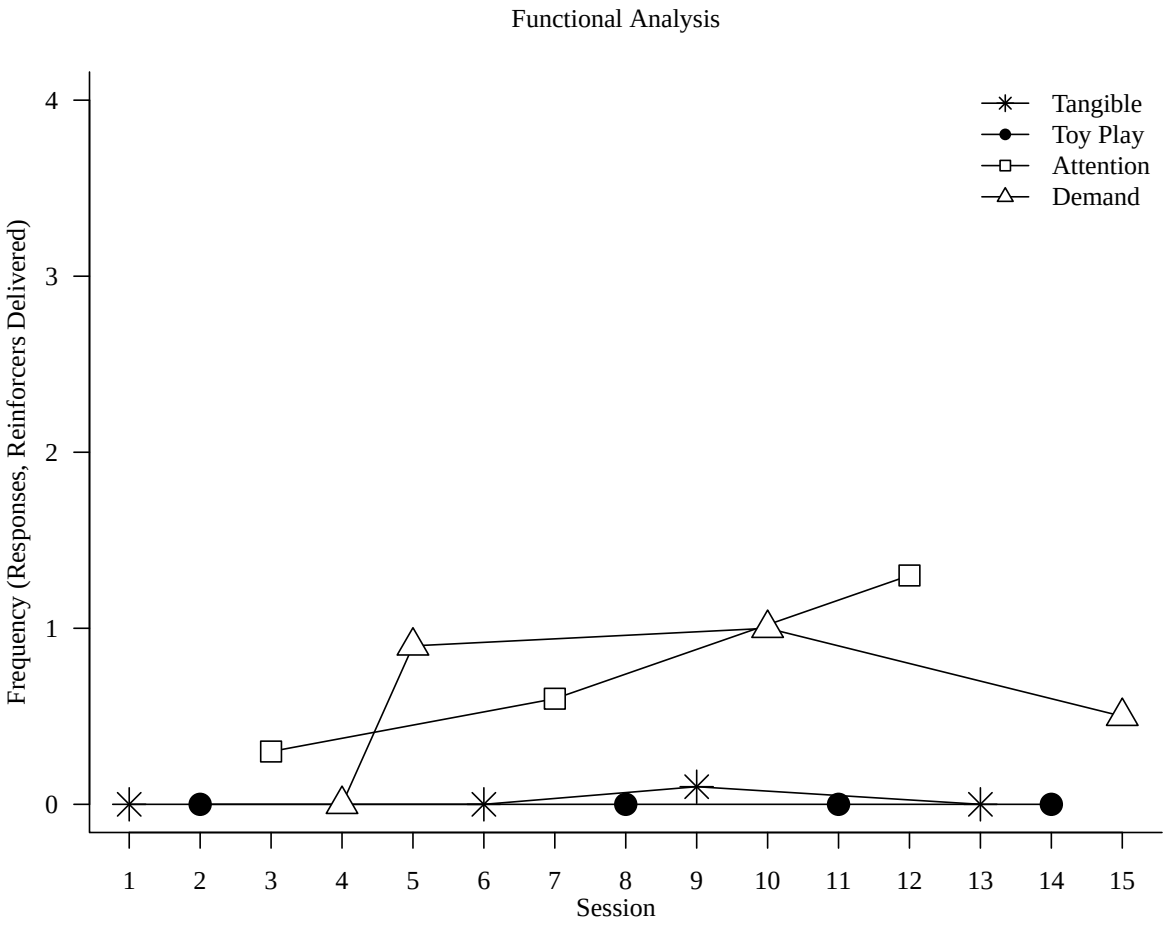
The *fxl* package was designed to support transparent, replicable, and efficently-drawn figure in single-case research design. In addition to be easily archived, effortlessly drawn, and automated, the R ecosystem supports the drawing of publication-quality figures that exceed traditional tools (e.g., Microsoft Excel). For instance, the figures output by *fxl* can be lossless by default using a variety of vector-based drawing formats.

Functional Analysis; Gilroy et al. (2019)

The *fxl* package includes tools specific to the charting of functional analyses. Briefly, a variety of conventions exist regarding the markers and colors associated with specific environmental manipulations. Methods included in the *plotMultiElementFA* function detect conditions and apply such styles in an automated manner.

For convenience, an analog function analysis published Gilroy et al. (2019) is illustrated below (doi: https://doi.org/10.1080/17518423.2019.1646342):

```
library(fxl)

plotMultiElementFA(Gilroyetal2019,
 grouping = 'Condition',
 session  = 'Session',
 response = 'CTB',
 title    = 'Functional Analysis',
 xlab     = 'Session',
 ylab     = 'Frequency (Responses, Reinforcers Delivered)',
 ylims    = c(0, 4),
 legend.position = 'topright')
```



Annotated Reversal Design; Gilroy et al. (2019)

The *fxl* package supports annotating figures (i.e., drawing labels, text) when conventions such as legends are inefficient in terms of communicating ecological modification. In these situations, a variety of behavior analytic conventions are applied by directly drawing upon a panel. This approach, an annotated reversal figure, is supported with the *plotAnnotatedReversal* function.

For convenience, an annotated reversal (treatment evaluation) published in Gilroy et al. (2019) is illustrated below (doi: https://doi.org/10.1080/17518423.2019.1646342):

```
conditionLabel <- data.frame(
  Panel = rep("Attention", 11),
  X     = c(17.5,
            23.5,
            37.5,
            72.5,
            85,
            39, 57.5, 74,
            9.5,
            22,
            33),
  Y     = c(3.3,
            3,
            3,
```

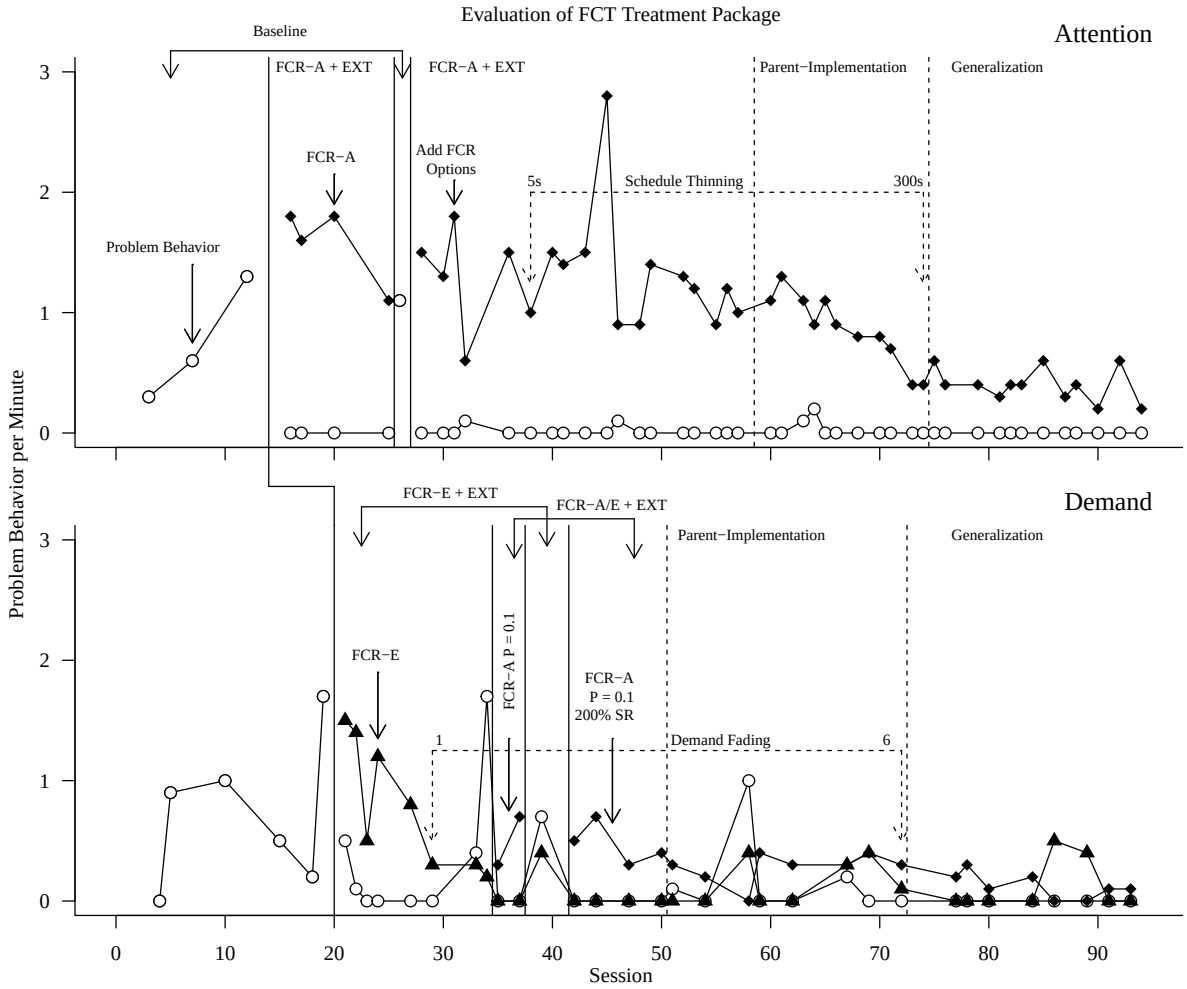
```
            3,
            3,
            2.05,2.05,2.05,
            1.5,
            2.25,
            2.15),
  Cex   = rep(0.75, 11),
  Text  = c("Baseline",
            "FCR-A + EXT",
            "FCR-A + EXT",
            "Parent-Implementation",
            "Generalization",
            "5s",
            "Schedule Thinning",
            "300s",
            "Problem Behavior",
            "FCR-A",
            "Add FCR\r\nOptions"),
  Srt   = rep(0, 11)
)

conditionLabel2 <- data.frame(
  Panel = rep("Demand", 10),
  X     = c(35,
            50.5,
            65,
            85,
            30, 60, 71,
            26,
            36.5,
            47.5),
  Y     = c(3.35,
            3.25,
            3,
            3,
            1.3, 1.3, 1.3,
            2,
            2.4,
            1.5),
  Cex   = rep(0.75, 10),
  Text  = c("FCR-E + EXT",
            "FCR-A/E + EXT",
            "Parent-Implementation",
            "Generalization",
            "1",
            "Demand Fading",
            "6",
            "FCR-E",
            "FCR-A P = 0.1",
            "FCR-A\r\nP = 0.1\r\n200% SR"),
  Srt   = c(rep(0, 8),
            90,
            0)
```

```
)

conditionLabel <- rbind(
  conditionLabel,
  conditionLabel2
)

panelLabel <- data.frame(
  Panel = c("Attention",
            "Demand"),
  X     = rep(95, 2),
  Y     = rep(3.25, 2),
  Cex   = rep(1.25, 2),
  Text  = c("Attention",
            "Demand")
)

annotatePhaseLines <- data.frame(
  Panel = c(rep("Attention", 2),
            rep("Demand", 5)),
  X     = c(58.5, 74.5,
            34.5, 37.5, 41.5, 50.5, 72.5),
  Lty   = c(2, 2,
            1, 1, 1, 2, 2)
)

annotateBrackets <- data.frame(
  Panel = c("Attention",
            "Demand",
            "Demand",
            "Attention",
            "Demand",
            "Attention",
            "Attention",
            "Attention",
            "Demand",
            "Demand",
            "Demand"),
  X1    = c(5,
            22.5,
            36.5,
            38,
            29,
            7,
            20,
            31,
            24,
            36,
            45.5),
  X2    = c(26.25,
            39.5,
            47.5,
            74,
```

```
            72,
            7,
            20,
            31,
            24,
            36,
            45.5),
  Y1    = c(3.175,
            3.275,
            3.175,
            2,
            1.25,
            1.4,
            2.15,
            2.1,
            1.9,
            1.35,
            1.35),
  Y2    = c(2.95,
            2.95,
            2.85,
            1.25,
            0.5,
            0.75,
            1.9,
            1.9,
            1.35,
            0.75,
            0.65),
  Lty   = c(1,1,1,
            2,2,
            1,1,1,
            1,1,1)
)

plotAnnotatedReversal(data     = Gilroyetal2019Tx,
                      grouping = 'Participant',
                      session  = 'Session',
                      response = 'CTB',
                      response2= 'FCR',
                      response3= 'FCR2',
                      condCol  = 'Condition',
                      pnum     = 'PhaseNum',
                      poff     = 'LineOff',
                      title    = 'Evaluation of FCT Treatment Package',
                      xlab     = 'Session',
                      ylab     = 'Problem Behavior per Minute',
                      deltaX   = 10,
                      ymins    = list("Attention"= 0,
                                      "Demand"   = 0),
                      ymaxs    = list("Attention"= 3,
                                      "Demand"   = 3),
                      clabs    = conditionLabel,
```

```
                    plabs    = panelLabel,
                    alines   = annotatePhaseLines,
                    abracks  = annotateBrackets)
```

```
## Warning in text.default(conds[c, "X"], conds[c, "Y"], conds[c, "Text"], : font
## width unknown for character 0xd

## Warning in text.default(conds[c, "X"], conds[c, "Y"], conds[c, "Text"], : font
## width unknown for character 0xd

## Warning in text.default(conds[c, "X"], conds[c, "Y"], conds[c, "Text"], : font
## width unknown for character 0xd
```



Multiple Baseline; Gilroy et al. (2015)

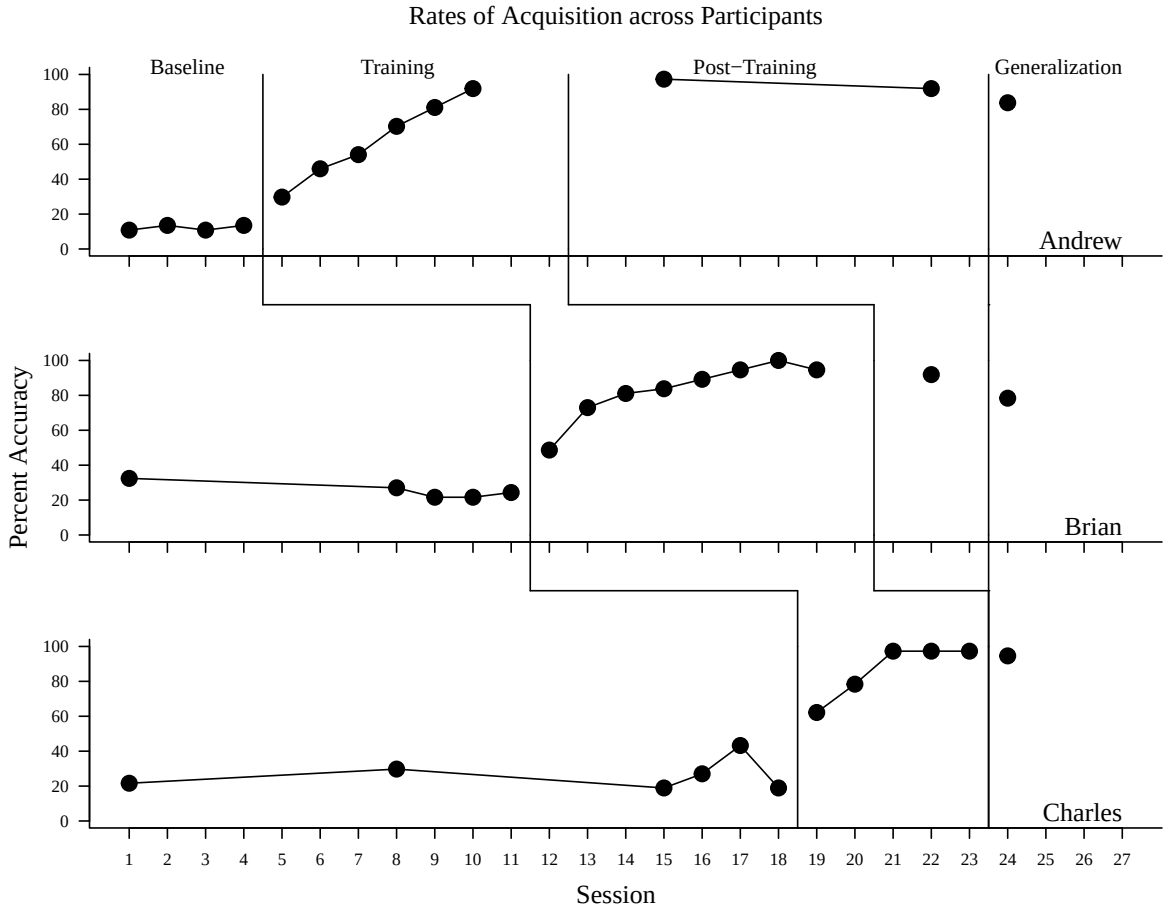
One of the most irritating aspects of single-case design is the plotting of condition changes. Specifically, across-panel drawing is an rare practice and rarely any software has been designed to support such practices. The *fxl* package includes methods that enable pixel-perfect drawing of phase/condition changes using straightforward conventions. Briefly, data are loaded into a data frame by panel (e.g., name), session number, condition (e.g., baseline), and phase number (e.g., 1, 2). Constructed in this way, R can look to see who changes in phases can be drawn and the linked together.

For convenience, a multiple baseline design published in Gilroy et al. (2015) is illustrated below (doi:

http://dx.doi.org/10.1016/j.rasd.2015.04.004):

```
# Specify location, text, and aesthetics of condition labels
conditionLabels <- data.frame(
  Panel = rep("Andrew", 4),          # Only on top row (for andrew)
  X     = c(3.5,                     # Positioned between phase change lines
            9,
            19,
            27),
  Y     = rep(100, 4),               # Positioned near top of panel
  Cex   = rep(1.25, 4),              # Slightly oversized for clarity
  Text  = c("Baseline",              # Content of phase labels
            "Training",
            "Post-Training",
            "Generalization")
)

# Specify location, text, and aesthetics of panel labels
panelLabels <- data.frame(
  Panel = c("Andrew",                # Specify the panels to draw upon
            "Brian",
            "Charles"),
  X     = rep(27, 3),                # Positioned near right of panel edge
  Y     = c(0,                       # Positioned near the abscissa
            0,
            0),
  Cex   = rep(1.5, 3),               # Slightly oversized for clarity
  Text  = c("Andrew",                # Specific text to write
            "Brian",
            "Charles")
)

# Adding in 'fake' phase data, since not all have identical exposure
dataFrameAdds <- data.frame(
  Participant   = c("Andrew",          # Specify where data is associated
                    "Brian",
                    "Charles"),
  Session       = c(23,                # Add in an 'edge' at session 23 (pre-generalization probe)
                    23,
                    23),
  Condition   = c("Generalization", # Specify condition (just for clarity sake)
                    "Maintenance",
                    "Maintenance"),
  Responding    = c(NA,                # Leave bx as NA, so nothing is plotted
                    NA,
                    NA),
  PhaseNum    = c(3,                 # Note phase index (important)
                    3,
                    4),
  LineOff     = c(0,                 # Specify line offset (in case of line overlap)
                    0,
                    0)
)
```

```
# Add in the faux data, so R can guess where phase changes look best
Gilroyetal2015 <- rbind(Gilroyetal2015,
                        dataFrameAdds)

plotMultipleBaseline(data     = Gilroyetal2015,
                     grouping = 'Participant',
                     session  = 'Session',
                     response = 'Responding',
                     pnum     = 'PhaseNum',
                     poff     = 'LineOff',
                     ymins    = list("Andrew"  = 0,    # Set lower limits (specific to panel)
                                     "Brian"   = 0,
                                     "Charles" = 0),
                     ymaxs    = list("Andrew"  = 100,  # Set upper limits (specific to panel)
                                     "Brian"   = 100,
                                     "Charles" = 100),
                     title    = 'Rates of Acquisition across Participants',
                     xlab     = 'Session',
                     xmax     = 27,                    # Extend out max session, for panel tags
                     ylab     = 'Percent Accuracy',
                     clabs    = conditionLabels,
                     plabs    = panelLabels)
```



Concurrent Reversal Designs; Gilroy et al. (2021)

In addition to multiple-baseline designs, single-case research may perform individual reversal designs across participants. In these cases, control is demonstrated both within- and across-participants and this warrants a slightly different charting strategy (despite keeping the same graphing conventions as the multiple baseline). However, the data supplied to *fxl* methods remains largely the same.

As an example of this, a collection of reversal designs published in Gilroy et al. (2021) are illustrated below (doi: https://doi.org/10.1002/jaba.826):

```
# Specify location, text, and aesthetics of condition labels
conditionLabel <- data.frame(
  Panel = rep("John", 6),          # Only on top row (for andrew)
  X     = c(2.5,                   # Positioned between phase change lines
            6.1,
            9,
            12.25,
            15,
            18.25),
  Y     = rep(19, 6),              # Positioned near top of (John's) panel
  Cex   = rep(1.25, 6),            # Slightly oversized for clarity
  Text  = c("Baseline",            # Content of phase labels
            "FR-Lowest",
            "Baseline",
            "FR-Inelastic",
            "FR-Elastic",
            "FR-Inelastic")
)

# Specify location, text, and aesthetics of panel labels
panelLabel <- data.frame(
  Panel = c("John",                # Specify the panels to draw upon
            "Anthony",
            "Charles"),
  X     = rep(26, 3),              # Positioned near right of panel edge
  Y     = c(5,                     # Positioned at various heights, for each case
            12,
            21),
  Cex   = rep(1.5, 3),             # Slightly oversized for clarity
  Text  = c("John",                # Specific text to write
            "Anthony",
            "Charles")
)

# Specify if/how we want to show a legend
legendParams <- data.frame(
  panel =      "John",                 # legend drawn on panel for John
  position =   "topright",             # drawn in top right corner
  legend = c("Responses Observed",     # Items to draw for legend
             "Reinforcers Produced"),
  col = c('black',                     # colors to draw lines
          'black'),
  lty = c(1, 2),                       # draw solid lines
  pch = c(19,2),                       # markers to draw
  bty = "n",                           # don't draw surrounding box
```

```
  pt.cex = 2.25,                        # oversize markers
  cex = 1.25,                           # slightly oversize text
  text.col = "black",                   # draw text in black
  horiz = F,                            # organize items vertically
  box.lty = 0                           # don't draw surrounding box
)

plotConcurrentReversals(data     = Gilroyetal2021,
                        grouping = 'Participant',
                        session  = 'Session',
                        response = 'Responding',
                        response2= 'Reinforcers',
                        pnum     = 'PhaseNum',
                        poff     = 'LineOff',
                        title    = 'Individual Evaluations of Reinforcer Efficacy and Elasticity across
                        xlab     = 'Session',
                        ylab     = 'Frequency (Responses, Reinforcers Delivered)',
                        ymins    = list("John"    = 0,
                                        "Anthony" = 0,
                                        "Charles" = 0),
                        ymaxs    = list("John"    = 20,
                                        "Anthony" = 10,
                                        "Charles" = 20),
                        clabs    = conditionLabel,
                        plabs    = panelLabel,
                        llabs    = legendParams)
```

Individual Evaluations of Reinforcer Efficacy and Elasticity across Reinforcers