`GGally::ggduo`
plot matrix for two grouped data

Barret Schloerke
Statistics PhD Candidate
Purdue University

Google Summer of Code

About myself

- **Purdue University**
  - 4th Year PhD Candidate in Statistics
  - Research in large data visualization using R - http://deltarho.org
    - Dr. William Cleveland and Dr. Ryan Hafen
- Metamarkets.com - 1.5 years
  - Front end engineer - coffee script / node.js
- **Iowa State University**
  - B.S. in Computer Engineering
  - Research in statistical data visualization with R
    - Dr. Di Cook, Dr. Hadley Wickham, and Dr. Heike Hofmann

GGally::ggpairs

- Emerson, Green, Schloerke, Crowley, Cook, Hofmann, Wickham. "The Generalized Pairs Plot." *JCGS*, vol. 22, no. 1, pp. 79-91, 2012.
- Complete pairwise plot matrix
  - A, B, C vs. A, B, C
- Three "matrix" sections:
  - `upper`, `lower`, `diag`
- Three main section types:
  - continuous, combo, discrete
- Produces a `ggmatrix` object


```
pm <- ggpairs(
  tips, c(1,2,6),
  mapping = aes(color = sex)
); pm
```

`GGally::ggmatrix` structure

- Generic plot matrix with fine tune control of
  - Bottom and left axis labels on outer layer of plots
  - Overall X and Y axis titles and plot matrix title
  - May have a variable number of rows (n) and columns (m)
  - Contains a list made of
    - (custom) ggplot2 objects
    - Functions that will evaluate with the supplied data
  - Allows for many plots (n*m) with large data

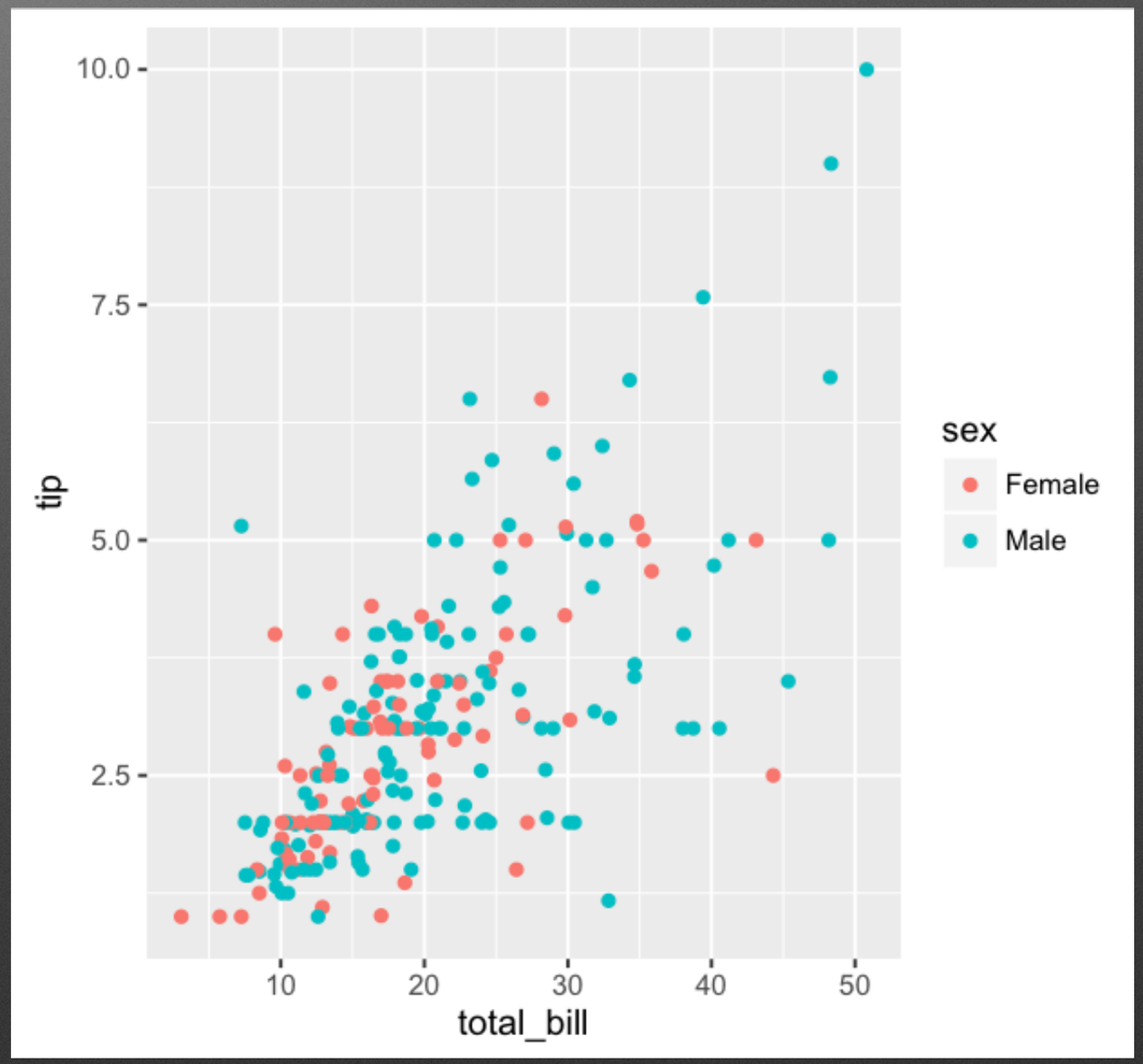
GGally::ggmatrix

- Retrieve individual plot

ggplot2_obj <- pm[3,1]

- Store individual plot

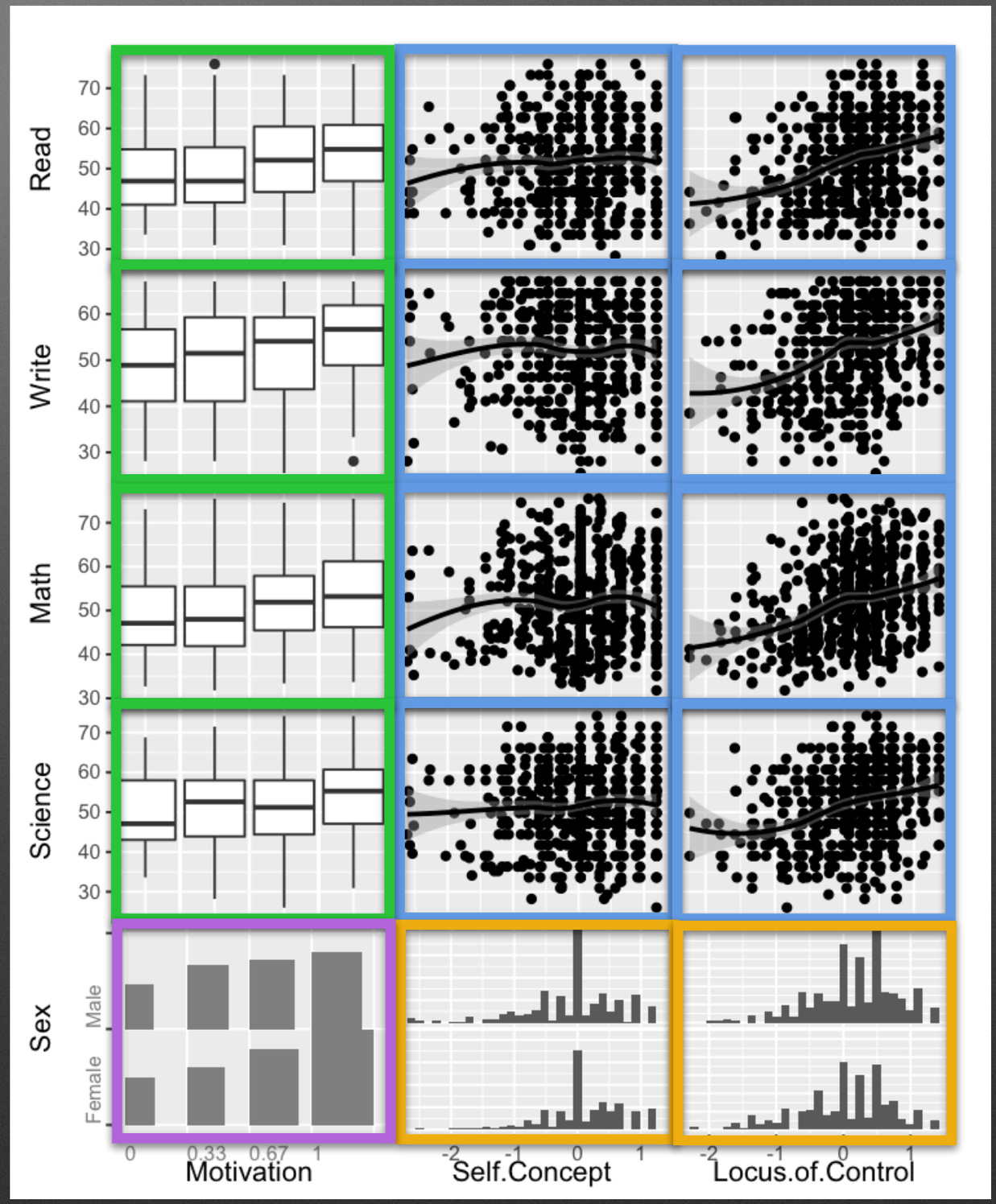
pm[row_pos, col_pos] <-
other_ggplot2_obj



pm[3,1]

GGally::ggduo

- Pairwise plot matrix for two grouped data
  - A, B, C vs. D, E, F, G
- Four main types:
  - continuous
  - comboVertical
  - comboHorizontal
  - discrete




```
ggduo(psych, 1:3, 4:8, showStrips = FALSE)
```

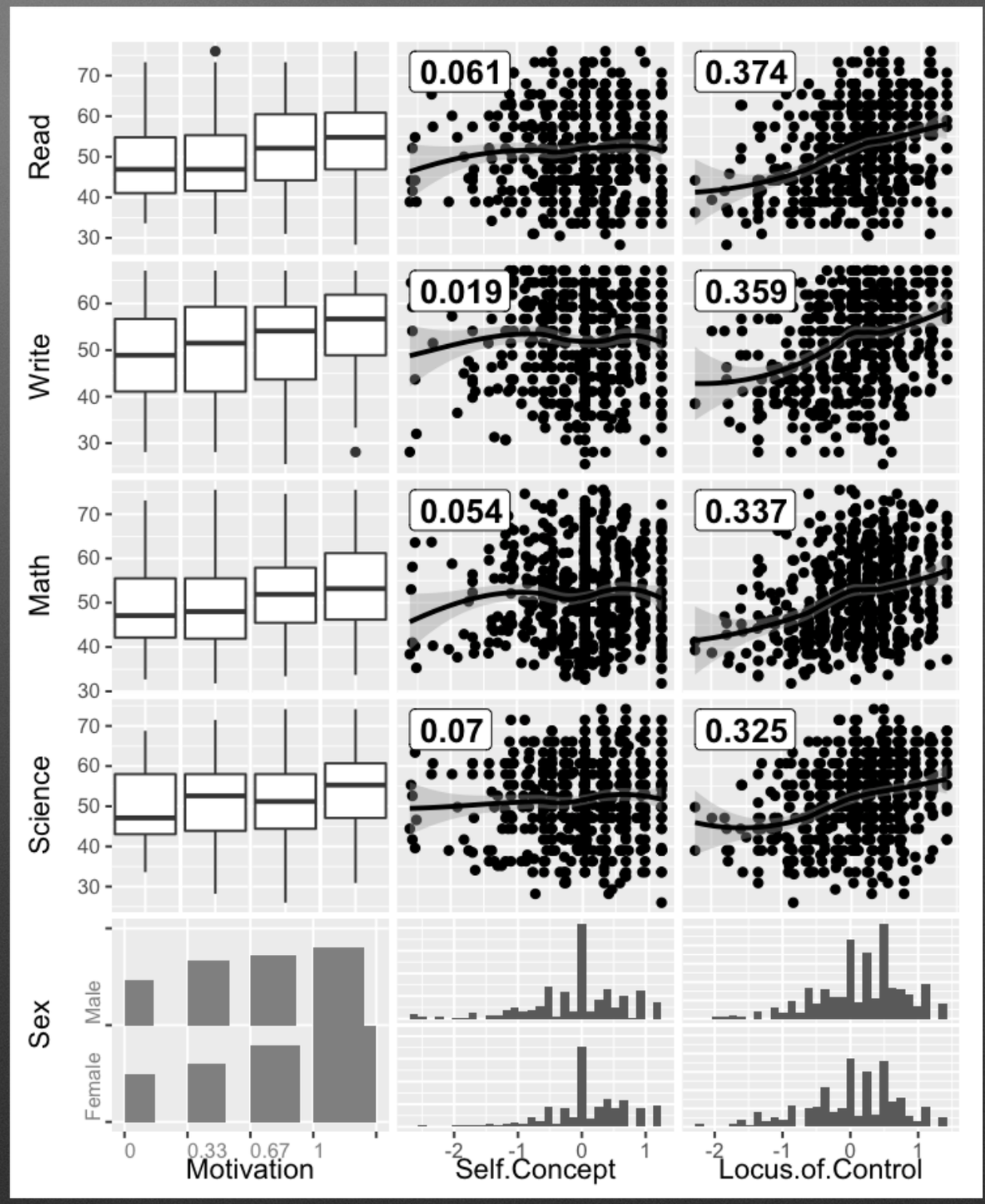
Application

- Can directly be used in
  - Canonical correlation analysis
  - Multiple time series analysis
  - Regression analysis

Canonical correlation analysis (CCA)

- `GGally::ggpairs` can be used for "within" correlation
- `GGally::ggduo` is useful for "between" correlations
- Can supply custom functions
  - Just like in `GGally::ggpairs`

```
ggduo(
  dt, 1:3, 4:8,
  types = list(
    continuous =
loess_with_cor
  ),
  showStrips = FALSE
)
```

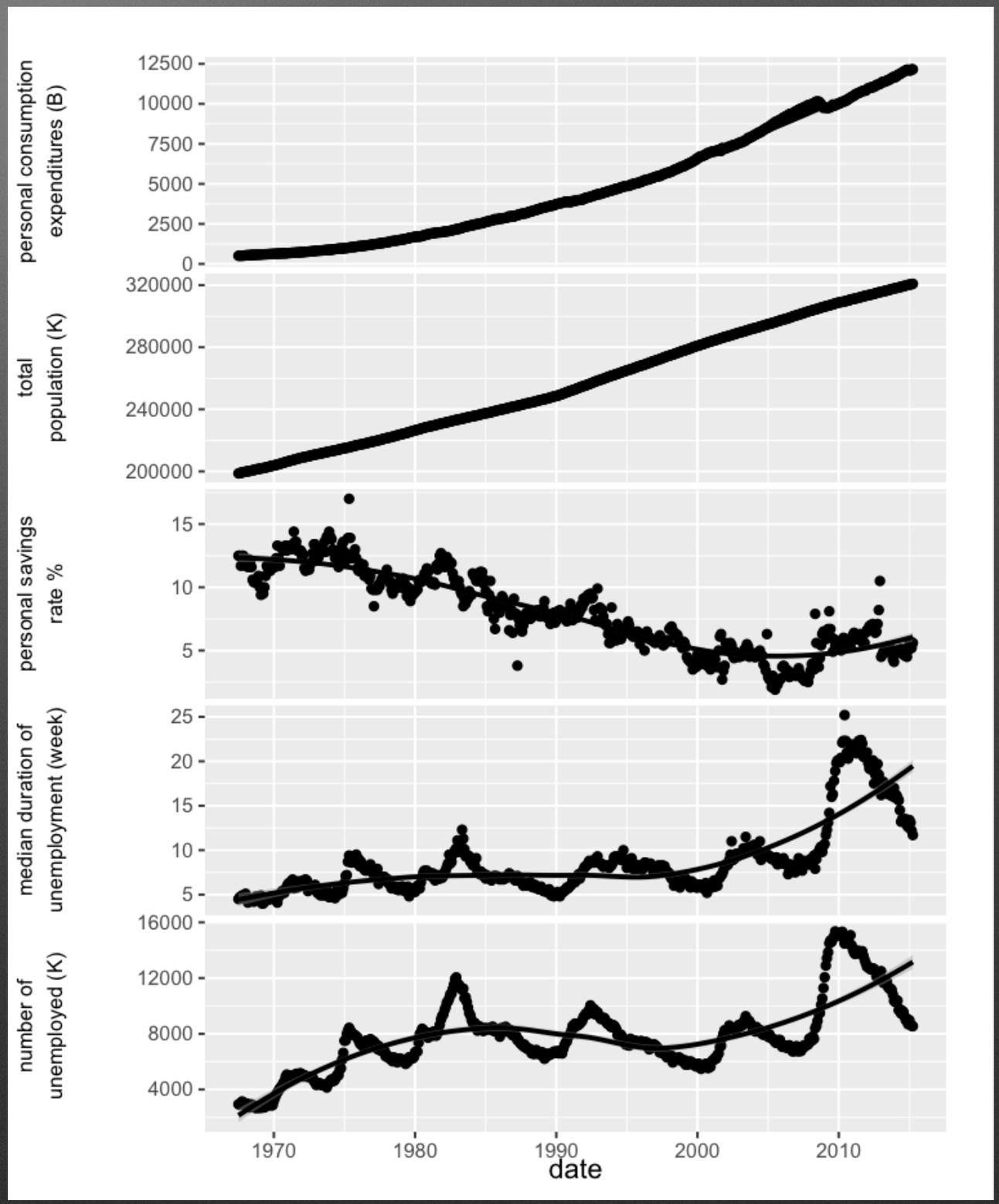


`ggduo(psych, 1:3, 4:8, showStrips = FALSE)`

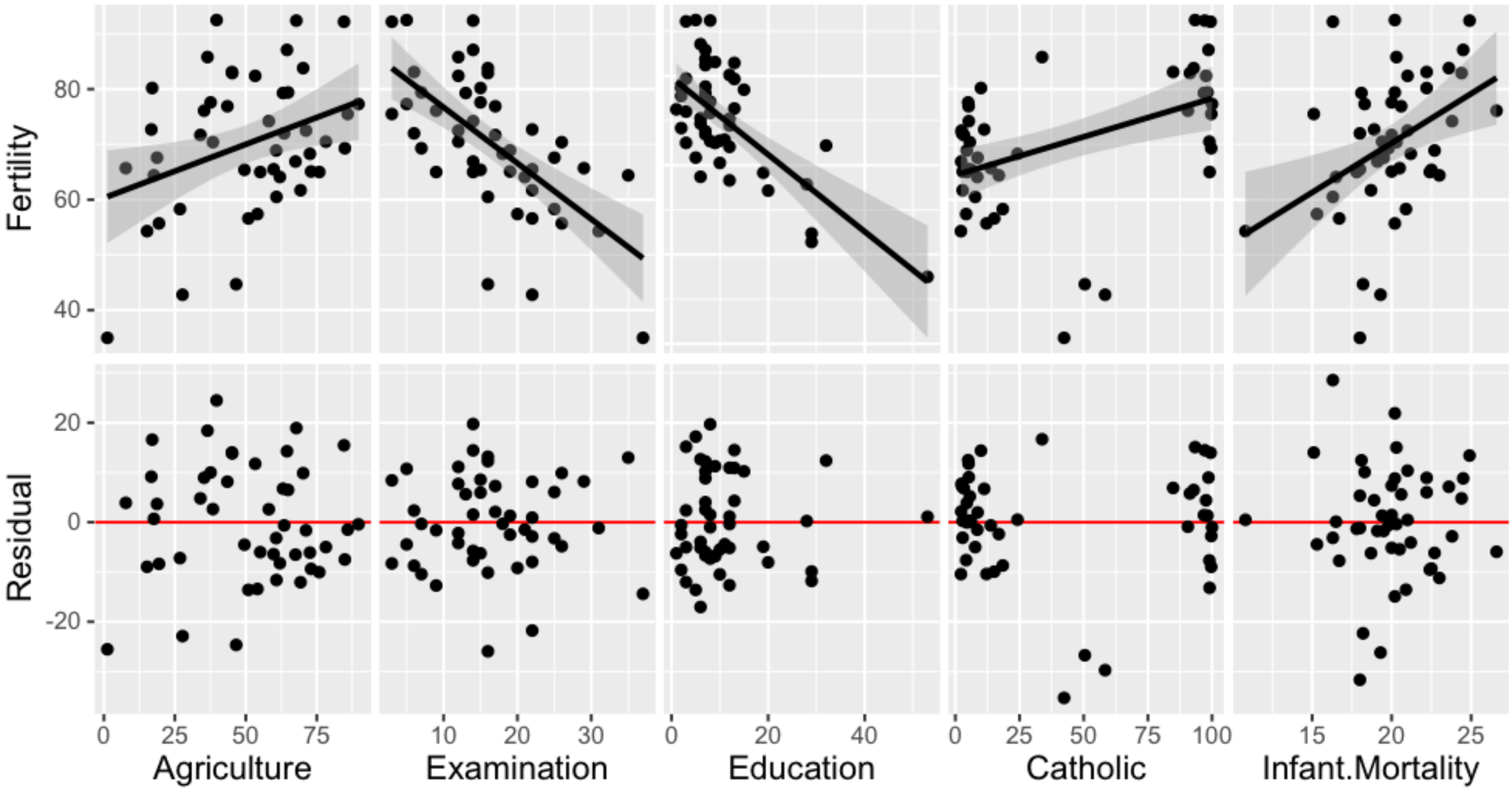
Multiple time series

- Multiple Y variables over time

```
ggduo(
  economics, 1, 2:6,
  columnLabelsY = y_labels
) +
  theme(
    axis.title.y =
      element_text(size = 9)
  )
```



Regression analysis



Regression analysis

Known information:

```
swiss$Residual <- seq_len(nrow(swiss))

residuals <- lapply(data[2:6], function(x) {
  summary(lm(Fertility ~ x, data = data))$residuals
})

y_range <- range(unlist(residuals))
```

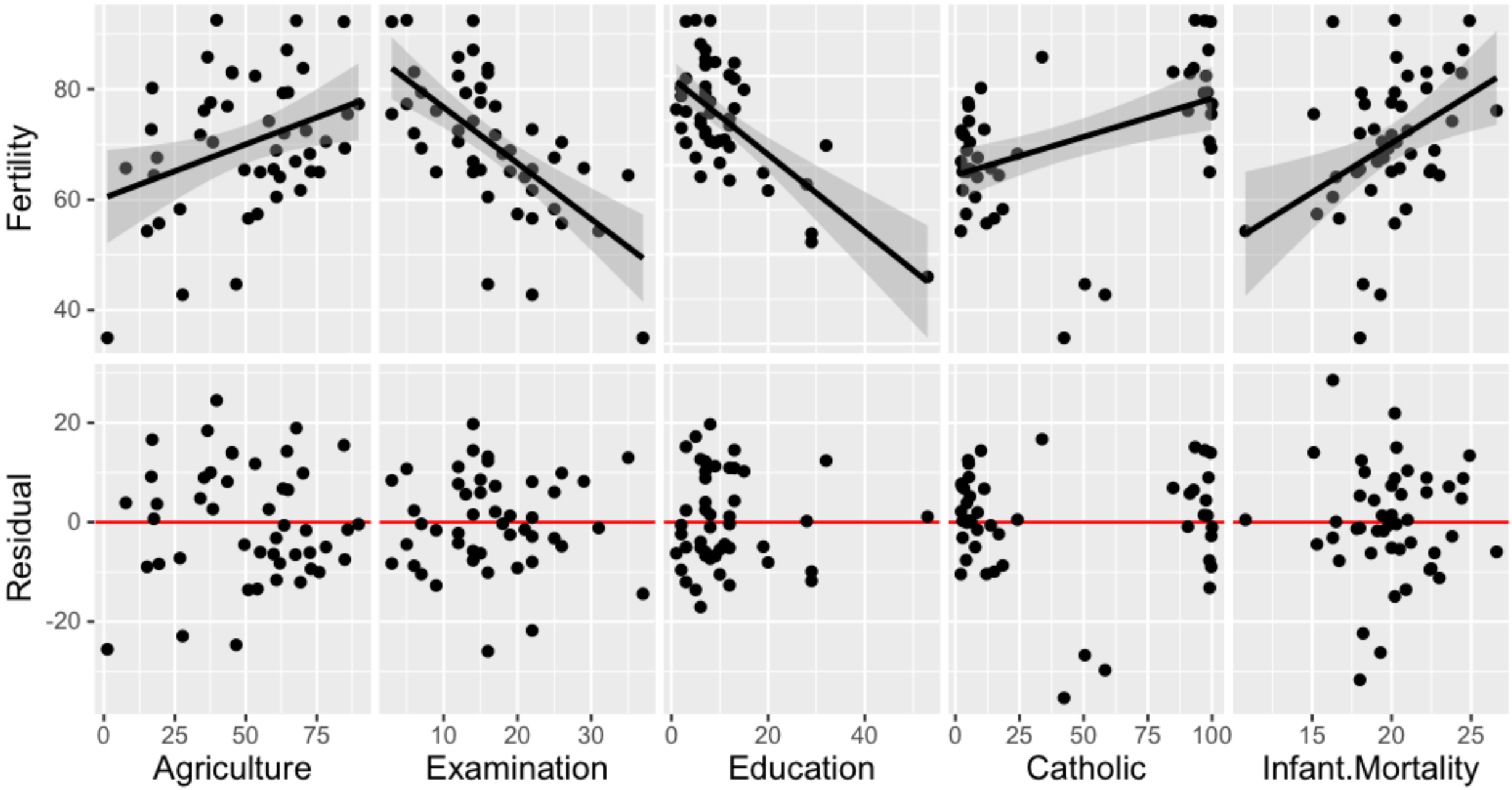
Regression analysis - manually

```
pm <- ggduo(
  swiss, 2:6, c("Fertility", "Residual"),
  types = list(continuous = ggally_smooth_lm)
)
for (j in 1:5) {
  resid_data = data.frame(
    x = swiss[[j + 1]],
    y = residuals[[j]]
  )
  # store plot
  pm[2,j] <- ggplot(data = resid_data, mapping = aes(x, y)) +
    ylim(y_range) +
    geom_hline(yintercept = 0, color = "red") +
    geom_point()
}
pm
```

Regression analysis - custom function

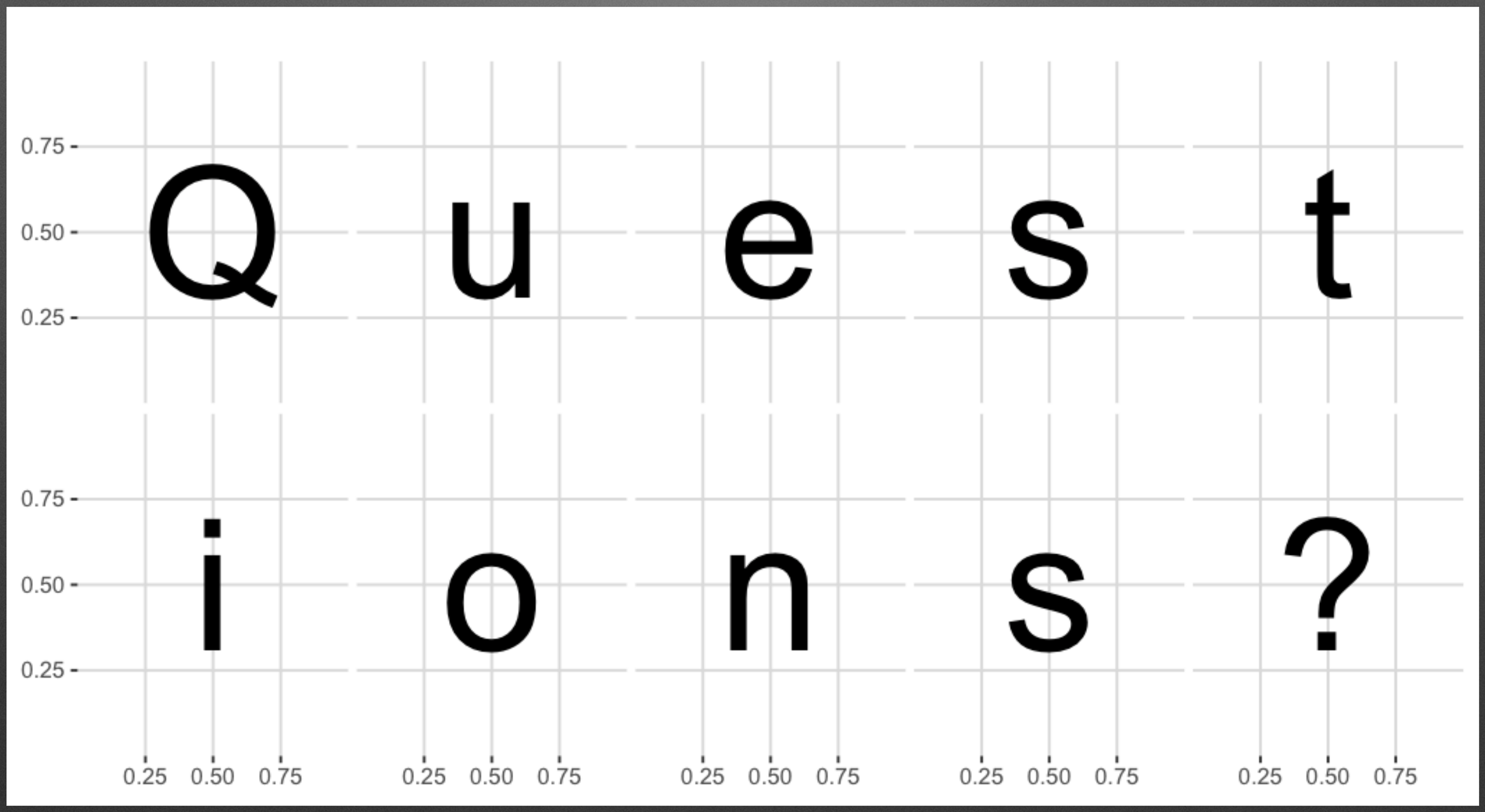
```
lm_or_res <- function(data, mapping, ..., lc = "red", ls = 1) {
  if (as.character(mapping$y) != "Residual") {
    return(ggally_smooth_lm(data, mapping, ...))
  }

  resid_data <- data.frame(
    x = data[[as.character(mapping$x)]],
    y = residuals[[as.character(mapping$x)]]
  )
  ggplot(data = resid_data, mapping = aes(x, y)) +
    ylim(y_range) +
    geom_hline(yintercept = 0, color = lc, size = ls) +
    geom_point(...)
}

ggduo(swiss, 2:6, c(1,7), types = list(continuous = lm_or_res))
```

Regression analysis



`GGally::ggduo`

- `GGally::ggduo`
  - Pairs plot matrix for two grouped data
  - Direct application
    - Canonical correlation analysis
    - Multiple time series analysis
    - Regression analysis
  - Custom functions!
- Future
  - Look into functions that take model objects directly … broom!


Quest
ions?
0.75
0.50
0.25
0.25 0.50 0.75

Data

```
tips <- reshape::tips
economics <- ggplot2::economics
swiss <- datasets::swiss

# http://www.ats.ucla.edu/stat/r/dae/canonical.htm (June 23, 2016)
psych <- read.csv("http://www.ats.ucla.edu/stat/data/mmreg.csv")
colnames(psych) <- c("Control", "Concept", "Motivation", "Read",
"Write", "Math", "Science", "Sex")
psych <- data.frame(
  Motivation = psych$Motivation,
  Self.Concept = psych$Concept,
  Locus.of.Control = psych$Control,
  Read = psych$Read,
  Write = psych$Write,
  Math = psych$Math,
  Science = psych$Science,
  Sex = c("0" = "Male", "1" = "Female")[as.character(psych$Sex)]
)
```