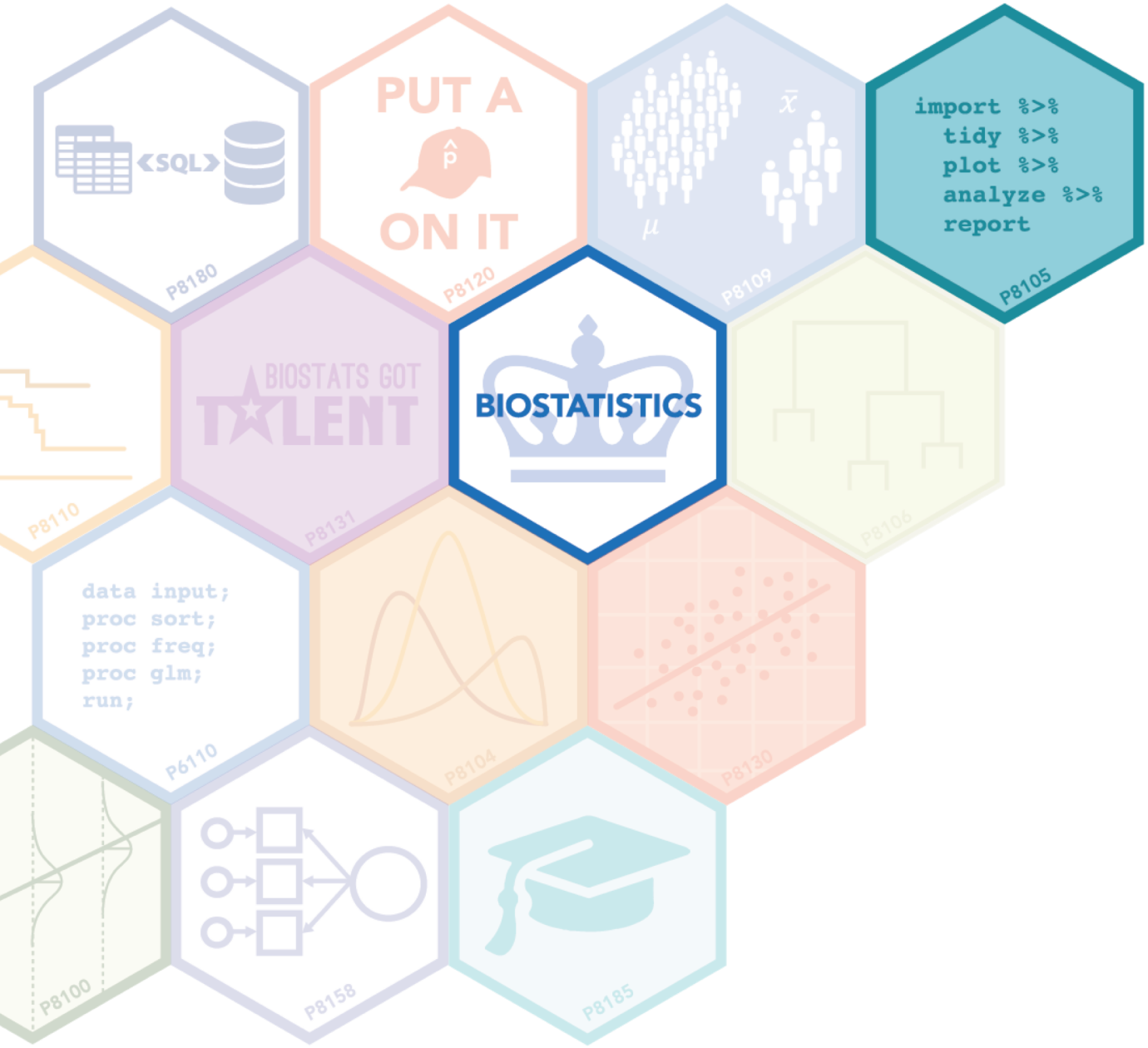
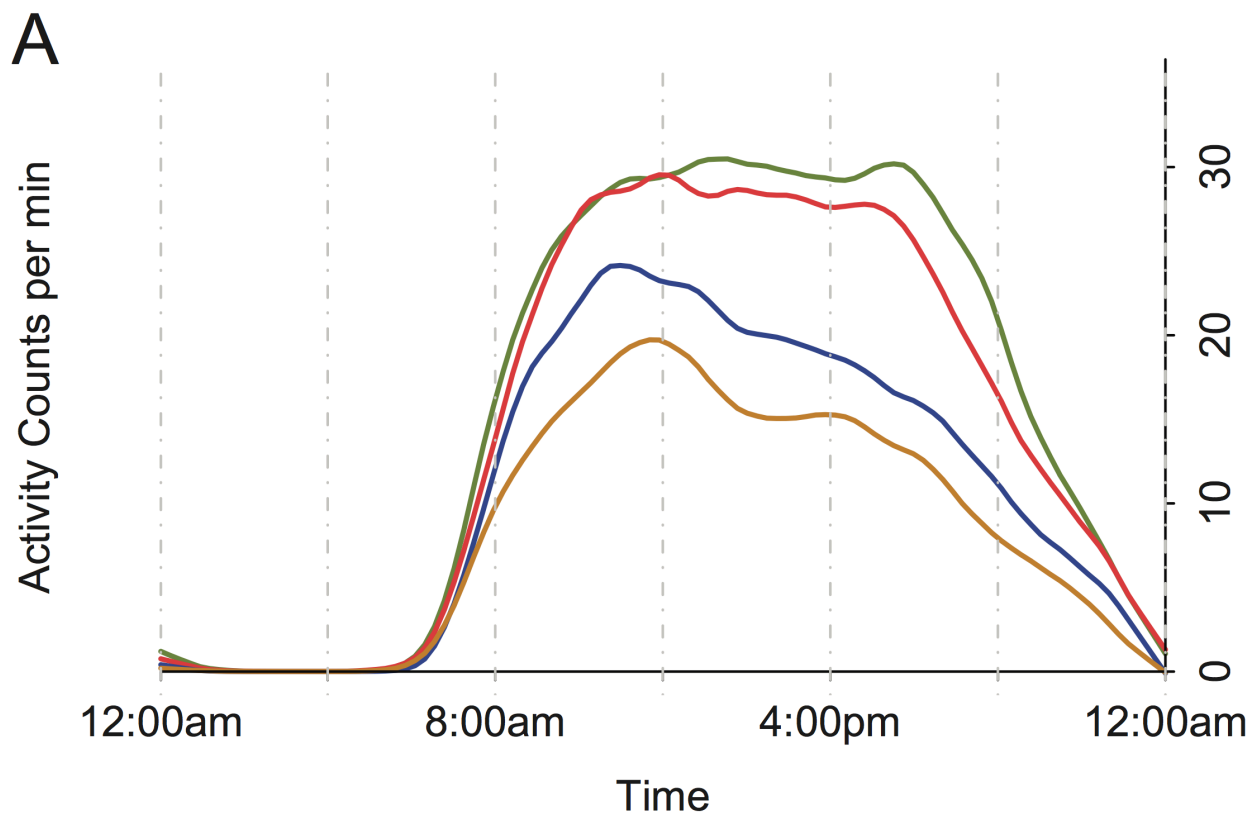


FUNCTION-ON-SCALAR MODELS

Jeff Goldsmith, PhD
Department of Biostatistics

Motivation



J. A. Schrack, V. Zipunnikov, J. Goldsmith, J. Bai, E. M. Simonsick, C. M. Crainiceanu, L. Ferrucci (2014). Assessing the "Physical Cliff": Detailed Quantification of Aging and Physical Activity. Journal of Gerontology: Medical Sciences, 69 973-979.

Questions to address

- Association?
- Confounding?
- Significance?

Functions as outcomes

- Activity may depend on covariates
- Associations may be different at different times of day
- Shift from functions as predictors to functions as outcomes
- “Function-on-scalar” regression

Simple linear regression

- The function-on-scalar model that is analogous to simple linear regression is

$$y_i(t) = \beta_0(t) + \beta_1(t)x_i + \epsilon_i(t)$$

- Functional response y_i
- Scalar predictor x_i
- Functional covariate is of interest
- Linear model
  - Most common approach

Linear FoSR

- The MLR equivalent is

$$y_i(t) = \beta_0(t) + \sum_{l=1}^{p} x_{il}\beta_l(t) + \epsilon_i(t)$$

- Functional response y_i
- Scalar predictor x_i
- Functional covariates are of interest
- Linear model
  - Most common approach

Basis expansion

- The functional coefficients are usually expanded in terms of a basis:

$$\beta_l(t) \approx \sum_{k=1}^{K} \phi_k(t)\beta_{kl}$$

- Several basis options are possible
  - FPC
  - Splines (my preference)
  - Wavelets
  - Fourier

Basis expansion

- For response data on a common finite grid, the model can be expressed

$$Y = XB\Phi^T + E$$

- Y is the matrix of row-stacked responses
- X is the usual design matrix
- Φ is the matrix of basis functions evaluated over the common grid
- B is the matrix of basis coefficients
- E is the matrix of row-stacked errors

Recast model

- By vectorizing the response and the linear predictor, we obtain the equivalent model formulation

$$vec(Y^T) = (X \otimes \Phi)vec(B^T) + vec(E^T)$$

- vec() concatenates the columns of the matrix argument
- $\otimes$ is the kronecker product
- This reformulates function-on-scalar regression as a usual least-squares problem
- Goal is to estimate the columns of B or, equivalently, the elements of vec(B)

Correlated errors

- Errors $\epsilon_i(t)$ are correlated within a subject, but variable selection methods assume independent errors
- Three approaches:
  - Ignore this issue
  - Use GLS in place of OLS by "pre-whitening" the left and right side of the matrix formulation of the model:
    - **I.e. define $Y^* = Y(L^{-1})^T$ where $\Sigma = LL^T$ is the error covariance matrix, and similarly modify the RHS**
  - Jointly model the coefficient vector and the residual covariance
    - **Easiest in a Bayesian setting**

Smoothness constraints

- The preceding does not include smoothness constraints on estimated coefficients
- Such constraints often take the form of a penalty

$$\lambda_l \int [\beta_l(t)'']^2 dt$$

- Can be expressed in terms of a penalty on the basis coefficients
- Alternatively, one can be careful about the dimension of the spline basis

Testing

- "Global" test: under the null, there is no association between the predictor and outcome at any time

$$H_0 : \beta_k(t) = 0 \text{ for all } t$$

- "Local" test: under the null, there is no association between the predictor and outcome at time t

$$H_0 : \beta_k(t) = 0 \text{ for a specific } t$$

- Both can be implemented by focusing on basis coefficients
- The former avoids some multiple comparisons issues
- Analogous to F and t tests in MLR

Connection to MLR

- Integrating over t for each subject gives the average activity for a subject

$$\int y_i(t)\,dt = \bar{y}_i$$

- Can similarly average over coefficients in FoSR to approximate estimates in MLR

$$\int \beta_0(t)\,dt = \bar{\beta}_0 \text{ and } \int \beta_1(t)\,dt = \bar{\beta}_1$$

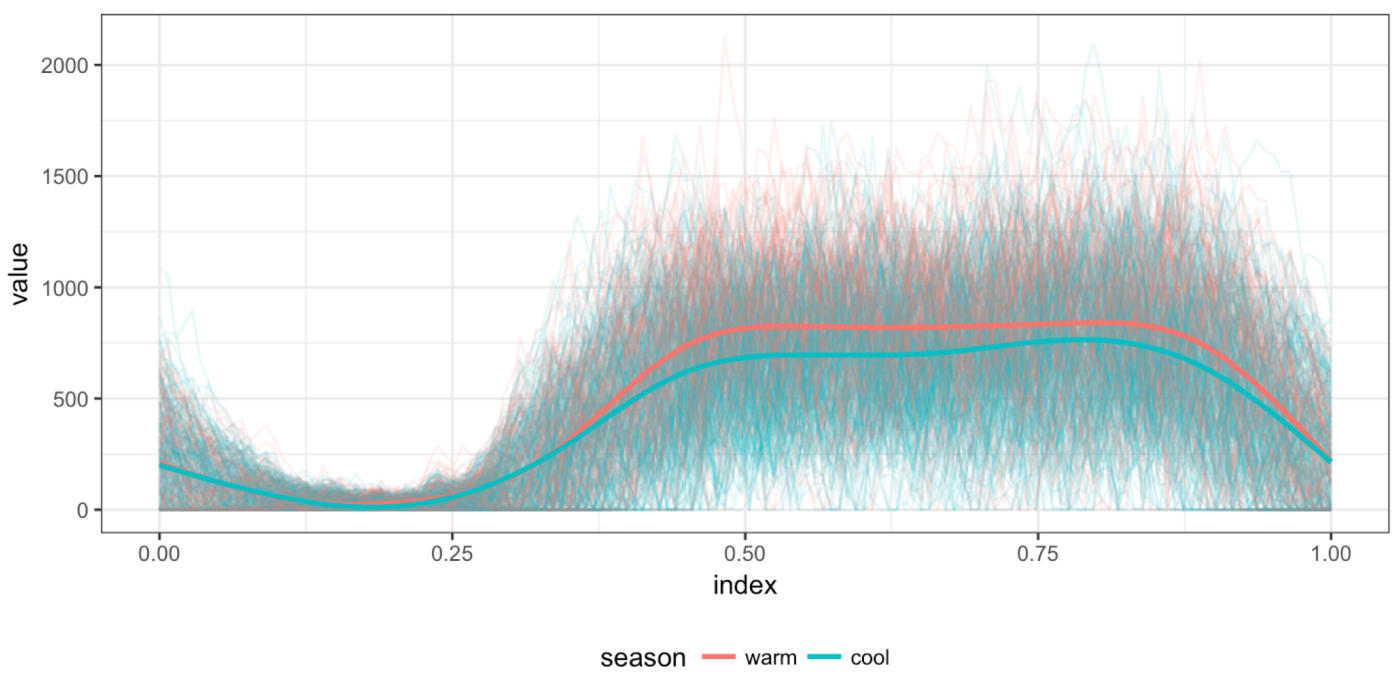
- Will differ slightly from an MLR fit to average activity counts
- Illustrates the connection (and difference) between approaches

Switch to code

Code

```
load("./DataCode/HeadStart.RDA")

as_refundObj(accel) %>%
    left_join(dplyr::select(covariate_data, id, season)) %>%
    ggplot(aes(x = index, y = value, group = id, color = season)) + geom_path(alpha = .1) +
    geom_smooth(aes(group = season), se = FALSE)
## Joining, by = "id"
## `geom_smooth()` using method = 'gam'
```



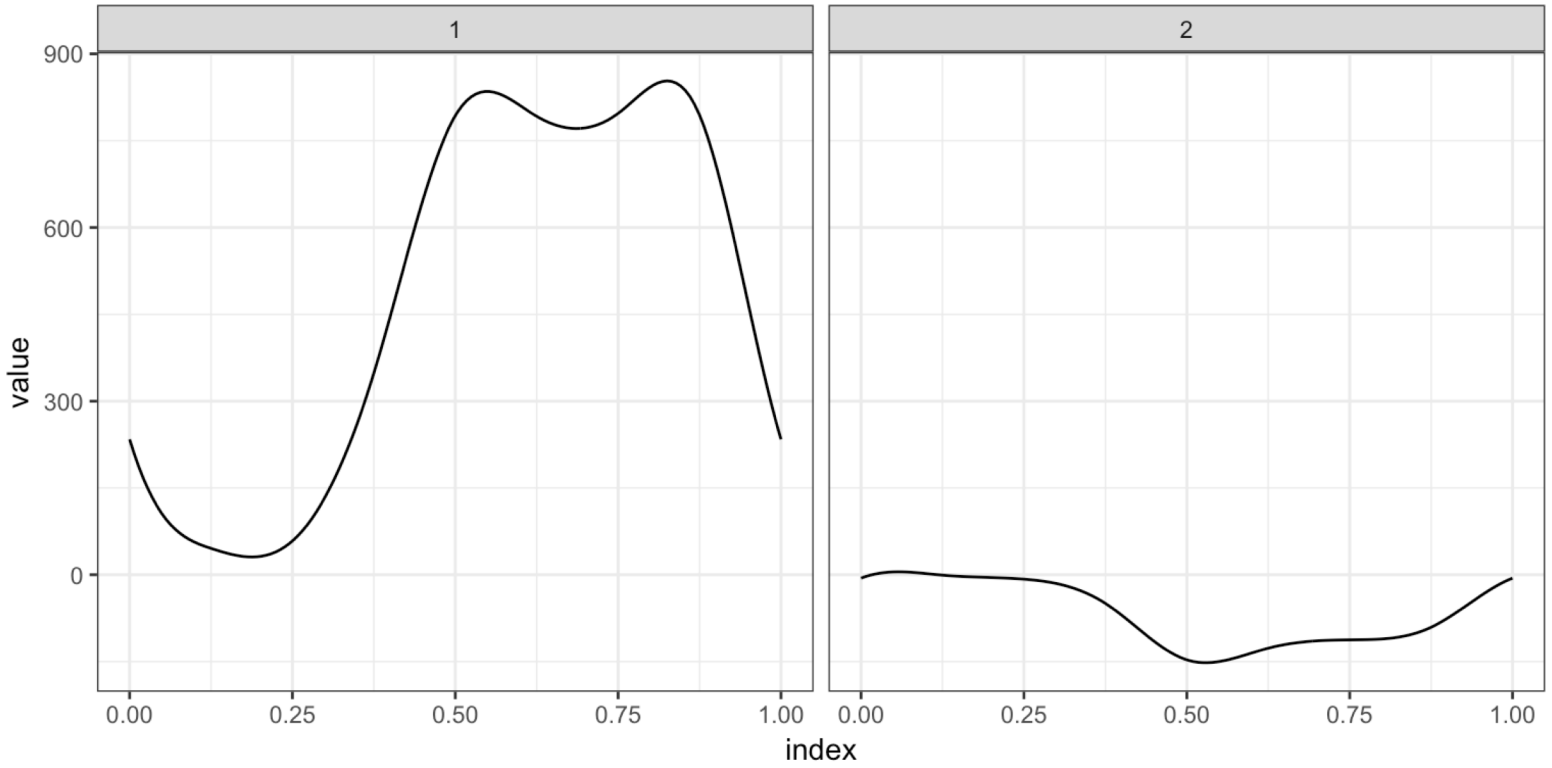
Code

```
temp_df = covariate_data
temp_df$accel = accel

fosr_slr = bayes_fosr(accel ~ season, data = temp_df,
                                          est.method = "GLS", Kt = 8, basis = "pbs")
## Using OLS to estimate residual covariance
## GLS

as_refundObj(fosr_slr$beta.hat) %>%
    ggplot(aes(x = index, y = value)) + facet_grid(~id) + geom_path()
```

Code

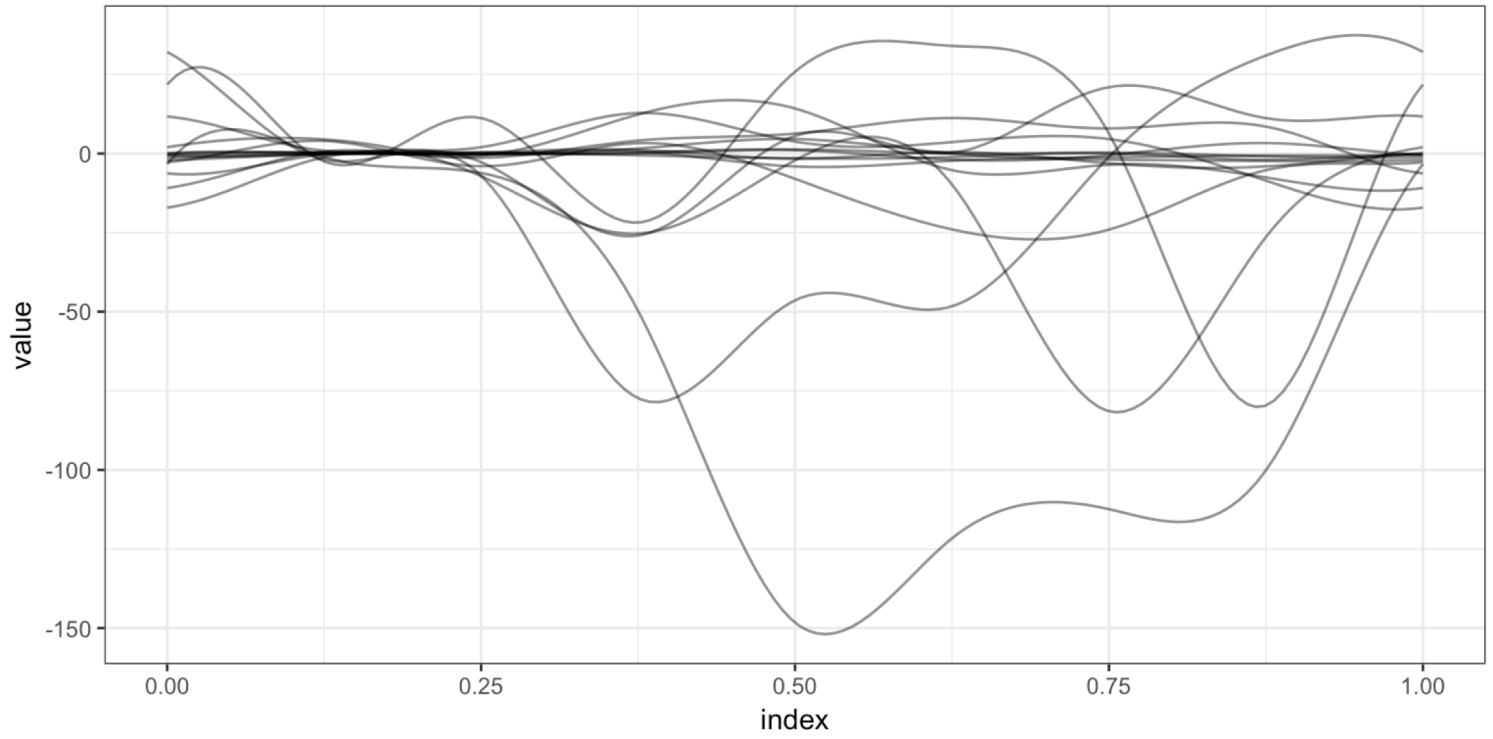


Code

```
fosr_mlr = bayes_fosr(accel ~ season + sex + BMIZ + TV +
                      videogames + mom_work + asthma + child_age +
                      mom_age + educ_mom + num_rooms + mom_born_US + tricep +
                      subscap + skinfold, data = temp_df,
                                          est.method = "GLS", Kt = 8, basis = "pbs")
## Using OLS to estimate residual covariance
## GLS

as_refundObj(fosr_mlr$beta.hat[-1,]) %>%
    ggplot(aes(x = index, y = value, group = id)) + geom_path(alpha = .5)
```

Code



Code

```
temp_df = covariate_data
temp_df$mean_accel = apply(accel, 1, mean)

mlr = lm(mean_accel ~ season + sex + BMIZ + TV +
                      videogames + mom_work + asthma + child_age +
                      mom_age + educ_mom + num_rooms + mom_born_US + tricep +
                      subscap + skinfold, data = temp_df)

summary(mlr) %>%
    broom::tidy() %>%
    knitr::kable(digits = 2)
```

Code

term	estimate	std.error	statistic	p.value
(Intercept)	585.54	53.19	11.01	0.00
seasoncool	-60.17	10.53	-5.72	0.00
sexmale	-9.38	10.14	-0.92	0.36
BMIZ	-0.10	3.56	-0.03	0.98
TV>=2h	-16.71	10.18	-1.64	0.10
videogames>=1h	-7.47	11.47	-0.65	0.52
mom_workyes	-4.44	10.15	-0.44	0.66
asthmayes	-2.99	10.73	-0.28	0.78
child_age	-0.91	0.65	-1.40	0.16
mom_age	-0.33	0.87	-0.38	0.71
educ_mom	-0.84	1.68	-0.50	0.62
num_rooms	5.17	4.79	1.08	0.28
mom_born_USyes	3.14	12.76	0.25	0.81
tricep	0.83	1.48	0.56	0.58
subscap	0.01	1.36	0.01	0.99
skinfold	-0.66	0.79	-0.83	0.41

Code

```
table.compare = data.frame(beta.mlr = coef(mlr),
                           beta.fosr = apply(fosr_mlr$beta.hat, 1, mean))

knitr::kable(table.compare, digits = 2)
```

Code

	beta.mlr	beta.fosr
(Intercept)	585.54	541.60
seasoncool	-60.17	-67.39
sexmale	-9.38	-14.54
BMIZ	-0.10	-0.17
TV>=2h	-16.71	-18.08
videogames>=1h	-7.47	-0.44
mom_workyes	-4.44	-7.57
asthmayes	-2.99	2.81
child_age	-0.91	-0.59
mom_age	-0.33	0.19
educ_mom	-0.84	-0.97
num_rooms	5.17	3.70
mom_born_USyes	3.14	-0.13
tricep	0.83	-0.10
subscap	0.01	-0.82
skinfold	-0.66	-0.07