Hierarchical GAMs

Fitting Models to Data, Not Data to Models
Model Fitting Series - With Applications in R

Jesse Ghashti
Source code by Stefano Mezzini

November 4, 2025

Centre for Scholarly Communication
The University of British Columbia | Okanagan Campus | Syilx Okanagan Nation Territory

Workshop Series Overview

Session	Topic	Date/Time
1	Simple Linear Regression	Oct 7, 9:00 AM
2	Fitting Linear Models in R	Oct 8, 10:30 AM
3	Multiple Linear Regression in R	Oct 16, 4:00 PM
4	Interaction Terms & Hierarchical Linear Models	Oct 21, 11:00 AM
5	Generalized Linear Models	Oct 23, 4:00 PM
6	Generalized Additive Models (GAMs)	Oct 28, 11:00 AM
7	Interpreting & Predicting from GAMs	Oct 29, 10:30 AM
8	Hierarchical GAMs	Nov 4, 12:00 PM
9	Penalized Models	Nov 18, 11:00 AM
10	Survival Models	Nov 25, 11:00 AM
11	Nonparametric Models	Dec 2, 11:00 AM

New Here?

<——————————-

New to R? Check out the Fundamentals of R series!

GitHub code and slides for today's workshop (and previous workshops)

——————————>

Alternatively, code/slides available at the bottom of `https://csc-ubc-okanagan.github.io/workshops/`

Last Time (Workshop 7) — Quick Recap

Key Concepts

- Interpreting and predicting from GAMs (link vs response scales)
- Population vs individual predictions (`exclude` random effects vs include)
- Tweedie vs Gamma families; model comparison with AIC/BIC/RMSE
- `discrete = TRUE/FALSE` trade-offs for `bam()` and `predict()`
- Always check diagnostics (`appraise()`, `gam.check()`, `summary()`)

Today: Interaction Terms & Hierarchical GAMs in R

Today we will...

- Review one- and multi-dimensional smooth types in `mgcv`
- Understand factor smooths (`fs`, `sz`) and random effects (`re`, `mrf`)
- Build interactions via `s()`, `te()`, and `ti()`
- Apply to real data: seasonal (`doy`) $\times$ long-term trend (`year`)

Required packages

```r
library('dplyr')     # data wrangling
library('mgcv')      # modeling
library('ggplot2')   # plotting
library('gratia')    # ggplot-based model graphics
library('lubridate') # dates
theme_set(theme_bw(base_size = 15) + theme(panel.grid = element_blank()))
```

Smooth Types in `mgcv`

```r
?mgcv::smooth.terms    # overview of available smoothers
install.packages('mgcv') # update mgcv if needed
```

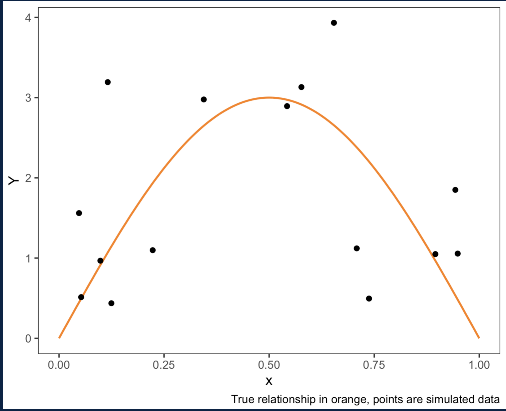
Overview:

- `tp`/`ts`: thin-plate (shrinkage) — flexible, optimal for given rank
- `cr`/`cs`: cubic (shrinkage) — faster, good default for many 1D uses
- `cc`: cyclic cubic — periodic endpoints (e.g., day-of-year)
- `ad`: adaptive — wiggliness varies over x (use sparingly)
- `ds`/`sos`/`so`: 2D+ (Duchon, sphere, soap film)

Warm-up: Simulated 1D Signal

```r
set.seed(-10)
true_values <- tibble(x = seq(0, 1, by = 1e-3),
                      mu = 3 * sinpi(x))
d <- tibble(x = runif(15),
            mu = 3 * sinpi(x),
            y = rnorm(n = length(mu),
             mean = mu, sd = 1))
ggplot() +
  geom_line(aes(x, mu), true_values,
  color = 'darkorange1', lwd = 1) +
  geom_point(aes(x, y), d) +
  labs(y = 'Y',
       caption = 'True relationship in orange')
```

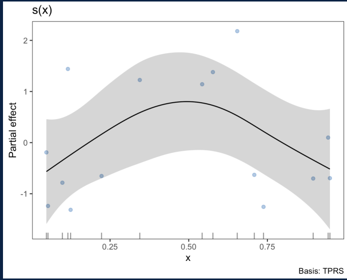
Small n (15 pts) $\Rightarrow$ good for illustrating regularization and smoother choices.

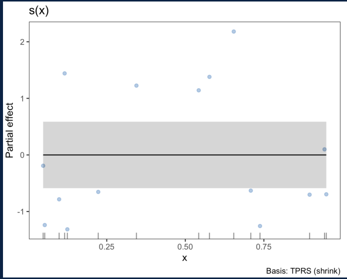


Thin-Plate vs Shrinkage Thin-Plate

```r
m <- gam(y ~ s(x, k = 10), data = d,
         family = gaussian(), method = 'REML')
draw(m, residuals = TRUE)  # bs = 'tp' (default)

m <- gam(y ~ s(x, bs = 'ts'), data = d,
         family = gaussian(), method = 'REML')
draw(m, residuals = TRUE)  # shrink-to-null possible
```

`ts` adds an extra penalty enabling the smooth to vanish if unsupported by data.

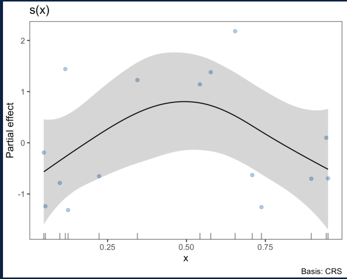


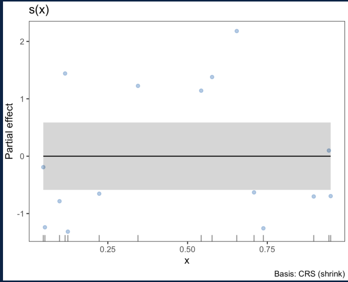


Cubic Regression Splines (with Shrinkage)

```r
m <- gam(y ~ s(x, bs = 'cr'), data = d,
         family = gaussian(), method = 'REML')
draw(m, residuals = TRUE)

m <- gam(y ~ s(x, bs = 'cs'), data = d,
         family = gaussian(), method = 'REML')
draw(m, residuals = TRUE)
```

`cr` often faster than TPRS; `cs` allows shrinkage to (near) zero.

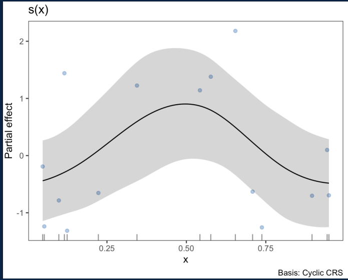


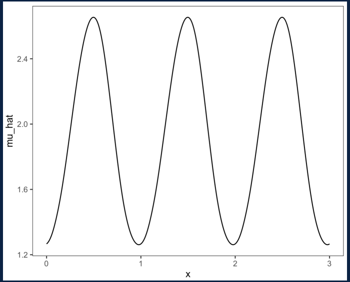


Cyclic Smoothers for Periodic Effects

```
m <- gam(y ~ s(x, bs = 'cc'),
         data = d, family = gaussian(), method = 'REML',
         knots = list(x = c(0, 1)))
draw(m, residuals = TRUE)

# show periodic extrapolation
tibble(x = seq(0, 3, by = 0.01),
       mu_hat = predict(m, newdata = tibble(x))) |>
  ggplot(aes(x, mu_hat)) + geom_line()
```

Use cases include day-of-year, hour-of-day, compass direction, etc.

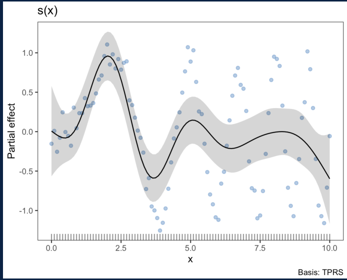


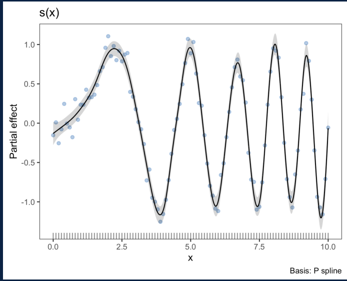


Adaptive Smooths (`bs = 'ad'`)

```r
d_ad <- tibble(x = seq(0, 10, by = 0.1),
               y = sinpi(0.1 * x^2) + rnorm(length(x), sd = 0.1))

m <- gam(y ~ s(x, bs = 'tp'), data = d_ad,
         family = gaussian(), method = 'REML')
draw(m, residuals = TRUE)

m <- gam(y ~ s(x, bs = 'ad'), data = d_ad,
         family = gaussian(), method = 'REML')
draw(m, residuals = TRUE, n = 1e4)
```

Only use ad when non-constant wiggliness is necessary; it's more complex and can overfit.

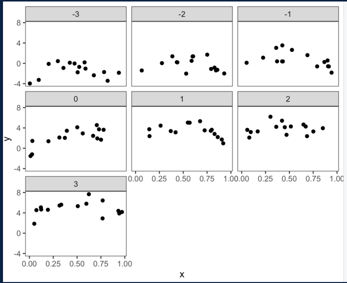


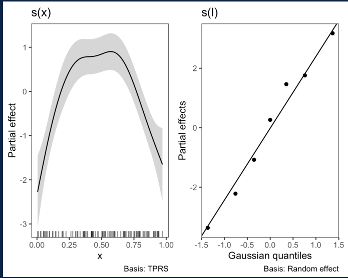


Random Effects in `mgcv` (`bs = 're'`)

```r
set.seed(0)
d <- tibble(l = rep(-3:3, 15),
            x = runif(length(l)),
            mu = 3 * sinpi(x) + l,
            y = rnorm(length(mu), mu, 1)) |>
  mutate(l = factor(l))

m <- gam(y ~ s(x) + s(l, bs = 're'),
         data = d, family = gaussian(), method = 'REML')
draw(m)
```

Factor ⇒ random intercept per level; include alongside smooth fixed effects.





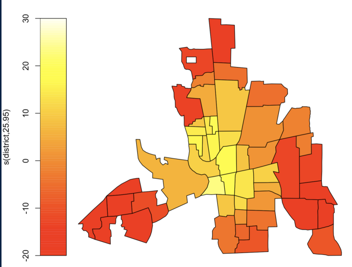
Spatial Random Effects: Markov Random Fields (`bs = 'mrf'`)

```r
# The Columbus Ohio Crime Data
data(columb); data(columb.polys)
xt <- list(polys = columb.polys)

b <- gam(crime ~ s(district, bs = "mrf", xt = xt),
         data = columb, method = "REML")
plot(b, scheme = 0) # 1 plot on next slide

b <- gam(crime ~ s(district, bs = "mrf", k = 20, xt = xt),
         data = columb, method = "REML")
plot(b, scheme = 0) # 2 plot on next slide

b <- gam(crime ~ s(district, bs = "mrf", k = 20, xt = xt) + s(income),
         data = columb, method = "REML")
plot(b, scheme = c(0, 1)) # 3 plot on next slide
```

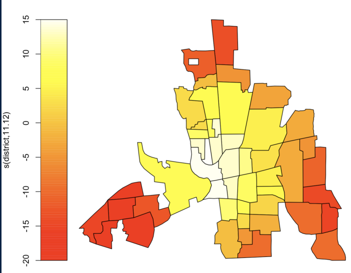
Good for adjacent areal units; supply neighborhood structure via `xt`.

Spatial Random Effects Visualizations

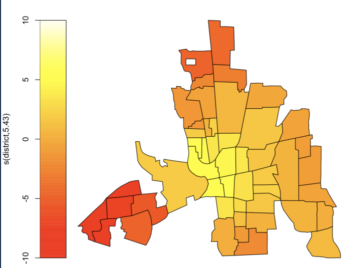
Plot 1



Plot 2



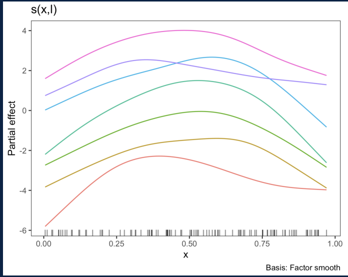
Plot 3

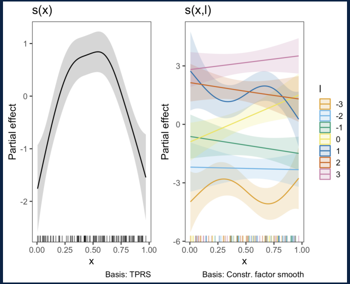


Factor Smooths: `fs` vs `sz`

```r
# unconstrained factor smooths (common smoothness + REs)
m <- gam(y ~ s(x, l, bs = 'fs'),
         data = d, family = gaussian(), method = 'REML')
draw(m, residuals = TRUE)  # residuals won't plot here

# sum-to-zero factor smooths (deviations from average)
m <- gam(y ~ s(x) + s(x, l, bs = 'sz'),
         data = d, family = gaussian(), method = 'REML')
draw(m, pages = 1)
```

`sz` shows each level's deviation from the common trend; `fs` shows level-specific trends with shared smoothness.

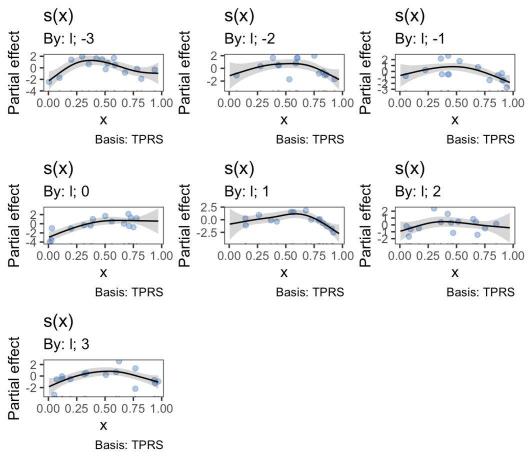




by-Smooths and the Important Intercepts

```r
# different smoothness per factor level; need fixed intercepts!
m <- gam(y ~ l + s(x, by = l),
         data = d, family = gaussian(), method = 'REML')
draw(m, residuals = TRUE)
```

by smooths do not include the factor's fixed effects—add them explicitly (e.g., l).

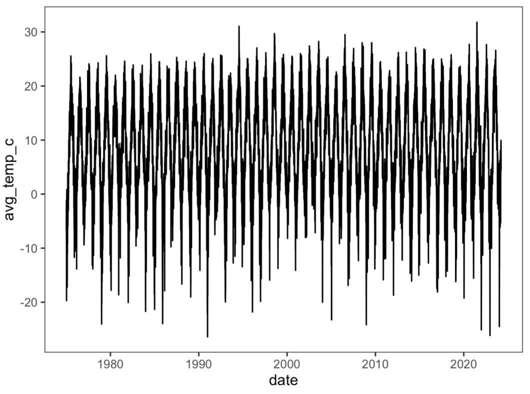


Kelowna Daily Temperatures: Preparing Data

```r
weather <- read.csv('data/weatherstats_kelowna_daily.csv') |>
  transmute(avg_temp_c = avg_temperature,
            date = as.POSIXct(date),
            year = year(date),
            doy  = yday(date)) |>
  filter(year >= 1975)
  head(weather)

ggplot(weather, aes(x = date, y = avg_temp_c)) +
  geom_line()
```

Cyclic seasonality (doy) + long-term changes (year) $\Rightarrow$ interaction candidates.

	avg_temp_c	date	year	doy
1	4.94	2024-03-25	2024	85
2	4.55	2024-03-24	2024	84
3	7.40	2024-03-23	2024	83
4	5.85	2024-03-22	2024	82

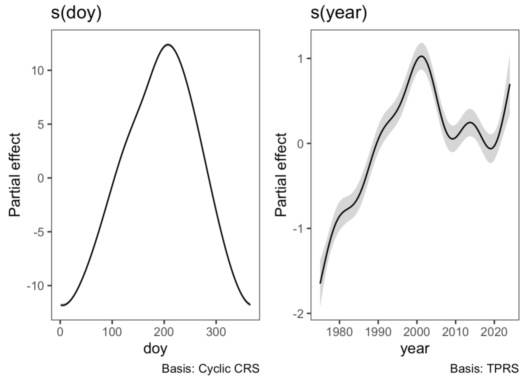


Additive Components Only: `s(doy) + s(year)`

```r
m <- bam(avg_temp_c ~ s(doy, bs = 'cc') + s(year, bs = 'tp', k = 10),
         family = gaussian(), data = weather, method = 'REML',
         knots = list(doy = c(0.5, 366.5)))
draw(m, rug = FALSE, n = 200)
summary(m)
```

Captures seasonality and trend but *no* interaction (seasonal shape fixed across years).

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.47811    0.02781   304.9   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
          edf Ref.df        F p-value
s(doy)  7.944  8.000 10828.52  <2e-16 ***
s(year) 8.369  8.889    60.16  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.829   Deviance explained = 82.9%
-REML =  49123  Scale est. = 13.871    n = 17952
```

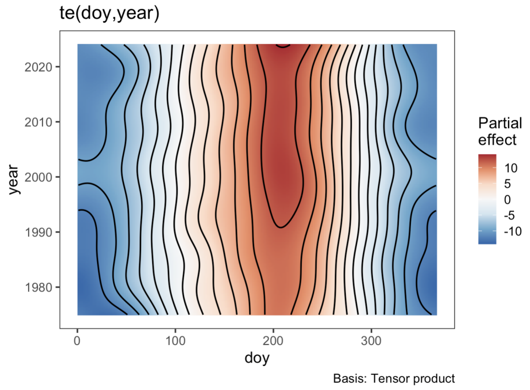


Tensor Product: `te(doy, year)`

```
m <- bam(avg_temp_c ~
           te(doy, year, k = c(10, 10), bs = c('cc', 'tp')),
         family = gaussian(), data = weather, method = 'REML',
         knots = list(doy = c(0.5, 366.5)))
draw(m, rug = FALSE, n = 200)
summary(m)
```

A single 2D surface; main effects not separated—significance harder to parse.

```
avg_temp_c ~ te(doy, year, k = c(10, 10), bs = c("cc", "tp"))

Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.47900    0.02761   307.1   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
               edf Ref.df    F p-value
te(doy,year) 75.41  84.46 1054  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

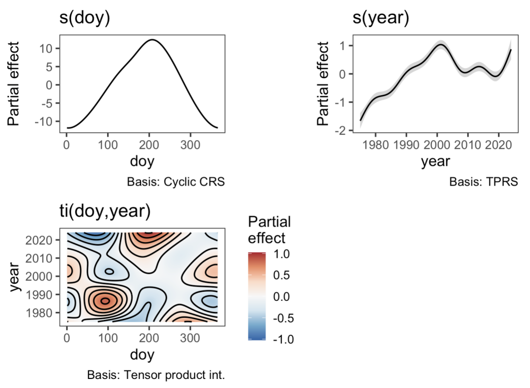


Separated Effects with Interaction: `ti(doy, year)`

```r
m <- bam(avg_temp_c ~
           s(doy,  k = 10, bs = 'cc') +
           s(year, k = 10, bs = 'tp') +
           ti(doy, year, k = c(5, 5), bs = c('cc', 'cr')),
         family = gaussian(), data = weather, method = 'REML',
         knots = list(doy = c(0.5, 366.5)))
draw(m, rug = FALSE, scales = 'free', n = 200)
summary(m)    # main + interaction terms reported separately
```

`ti` decomposes into main effects + identifiable interaction—clearer inference than `te`.

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.48249    0.02772     306   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Approximate significance of smooth terms:
                edf Ref.df        F p-value
s(doy)        7.944  8.000 10901.94  <2e-16 ***
s(year)       8.467  8.919    61.16  <2e-16 ***
ti(doy,year) 10.594 12.000    10.26  <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R-sq.(adj) =   0.83   Deviance explained = 83.1%
-REML =  49078  Scale est. = 13.776    n = 17952
```

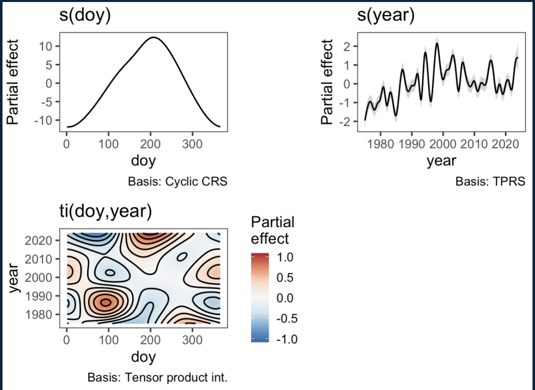


Allowing More Long-Term Fluctuation

```r
m <- bam(avg_temp_c ~
           s(doy,  k = 10, bs = 'cc') +
           s(year, k = 50, bs = 'tp') +
           ti(doy, year, k = c(5, 5), bs = c('cc', 'cr')),
         family = gaussian(), data = weather, method = 'REML',
         knots = list(doy = c(0.5, 366.5)))
draw(m, rug = FALSE, scales = 'free', n = 400)
summary(m)
```

Raising k increases flexibility; let REML penalization control complexity—still check diagnostics.

```
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.48765    0.02738     310   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
               edf Ref.df        F p-value
s(doy)       7.946   8.00 11153.95  <2e-16 ***
s(year)     44.271  48.01    21.68  <2e-16 ***
ti(doy,year) 10.656 12.00    10.78  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.835   Deviance explained = 83.5%
-REML =  48920  Scale est. = 13.412    n = 17952
```



Choosing Between `s()`, `te()`, and `ti()`

Guideline

Additive-only (`s(x) + s(z)`):

- Seasonal pattern constant over years; trend independent of season.

Tensor product `te(x, z)`:

- Flexible joint surface; inference is on the combined surface.

ANOVA-style `ti(x, z)`:

- Main effects + interaction separated—preferred for interpretability/testing.

Best Practices for Interaction & Hierarchical GAMs

Do's

- Match smooth type to structure: `cc` for cyclic, `re` for random intercepts
- Prefer `ti()` when you need interpretable main vs interaction effects
- Use shrinkage bases (`ts`, `cs`) for selection/regularization
- Validate k with `gam.check()` and residual diagnostics

Don'ts

- Don't forget fixed intercepts when using `by`-smooths
- Don't overfit with large k without checking penalties/diagnostics
- Don't ignore boundary behavior with non-cyclic smooths on periodic data

Key Takeaways

- `mgcv` offers many smoother families: choose based on structure (cyclic, adaptive, multidimensional)
- Hierarchical pieces: `re`, `fs`, `sz` complement `s()` for group-specific trends
- Interactions: `ti()` separates main effects and interactions for cleaner inference; `te()` for unified surfaces
- Always pair flexibility with diagnostics and sensible k

What's Next?

Additional Questions?
Book an Appointment!

Next Workshop:

Penalized Models
–> November 18, 11:00 AM

Thank You!

Questions?

Workshop Materials:

`https://github.com/csc-ubc-okanagan/ubco-csc-modeling-workshop`

Contact:

jesse.ghashti@ubc.ca