



HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

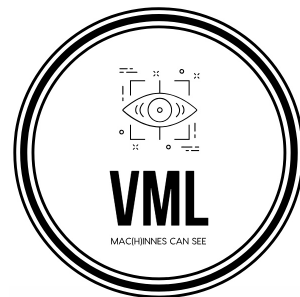
Non-linear IOR analysis with generalized additive mixed models

Tatiana Malevich

VML – Vision Modelling Lab, HSE

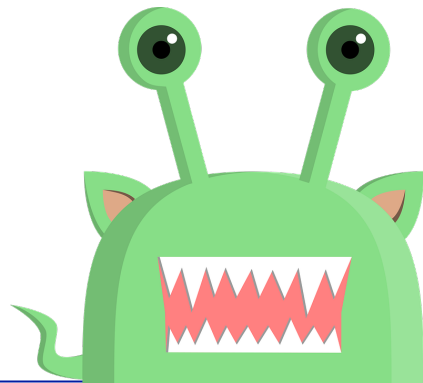
13th Research Seminar

February 22, 2019



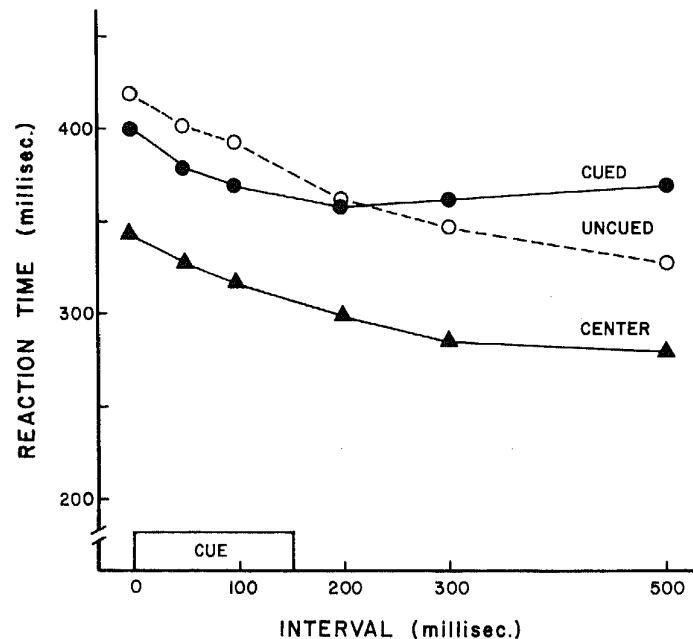
Coming soon

Malevich, T., Rybina, E., Ivtushok, E., Ardasheva, L., & W.J. MacInnes. (under prep). Inhibition of return depends not on the coding coordinates but on the hemifield of cue-target presentation



Inhibition of return (IOR)

- a delay in responding to a previously inspected location
- crucial for visual search
- must be coded in **spatiotopic**, environmental, reference frame (RF)

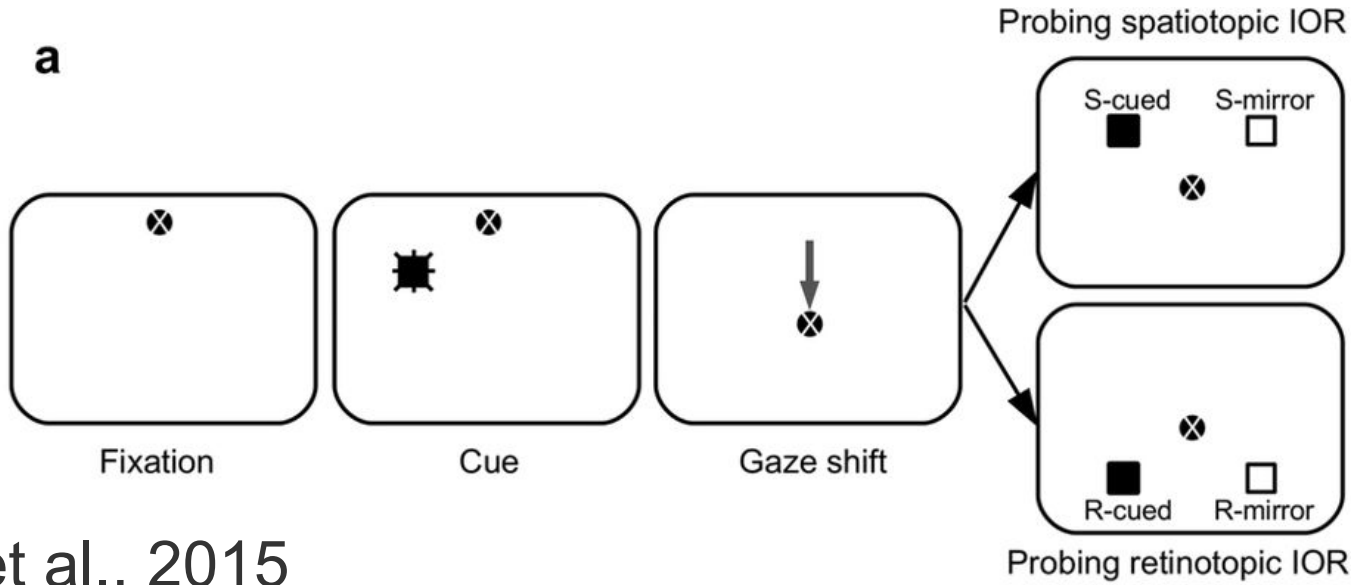


Posner & Cohen, 1984

But...

- most recent studies show that IOR is coded in both spatiotopic and retinotopic (eye-centered) RF
 - Sapir et al., 2004; Pertzov et al., 2010; Hilchey et al., 2012; Mathôt & Theeuwes, 2010; Krüger & Hunt, 2013; He et al., 2015
-

Typical paradigm (Maylor & Hockey, 1985)



He et al., 2015

Problems

- early vs late, manual vs saccadic
 - when the transition happens
 - remapping, corollary discharge signal, etc.
 - retinotopic IOR must be useless
 - IOR has temporal and spatial gradients
-

Our research question



Is there an independent source of retinotopic IOR or
is it a part of the spatiotopic gradient?

Our paradigm

- Four experiments
 - Fixed location vs gradient
 - Spatiotopic vs retinotopic vs neutral
 - Valid vs invalid
 - Manual vs saccadic RT
-

The easy (fixed) part

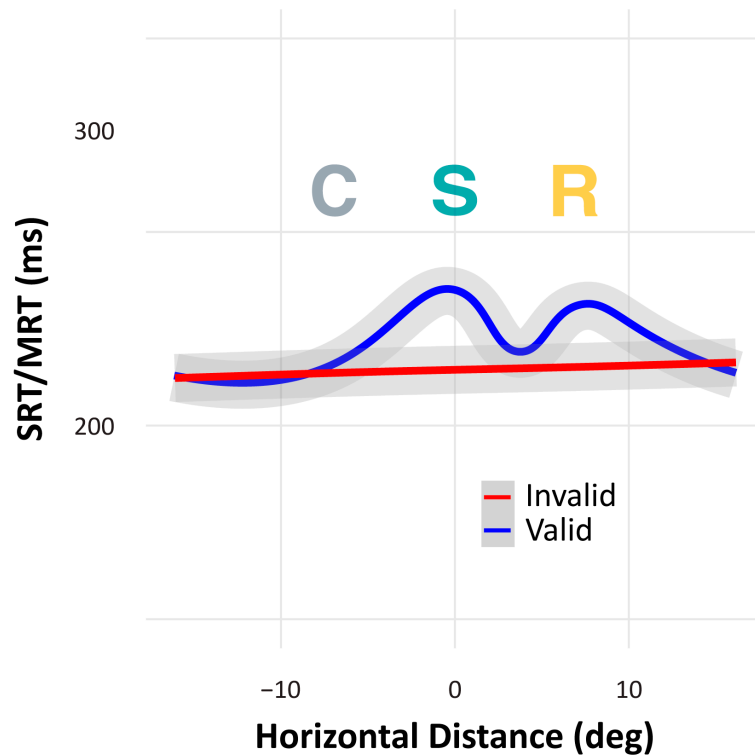
- E1: MRT
- Linear mixed effect models (LMMs)
- No effect of RF nor Validity by RF interaction

The easy (fixed) part

- E2: SRT
- Linear mixed effect models (LMMs)
- No effect of RF nor Validity by RF interaction

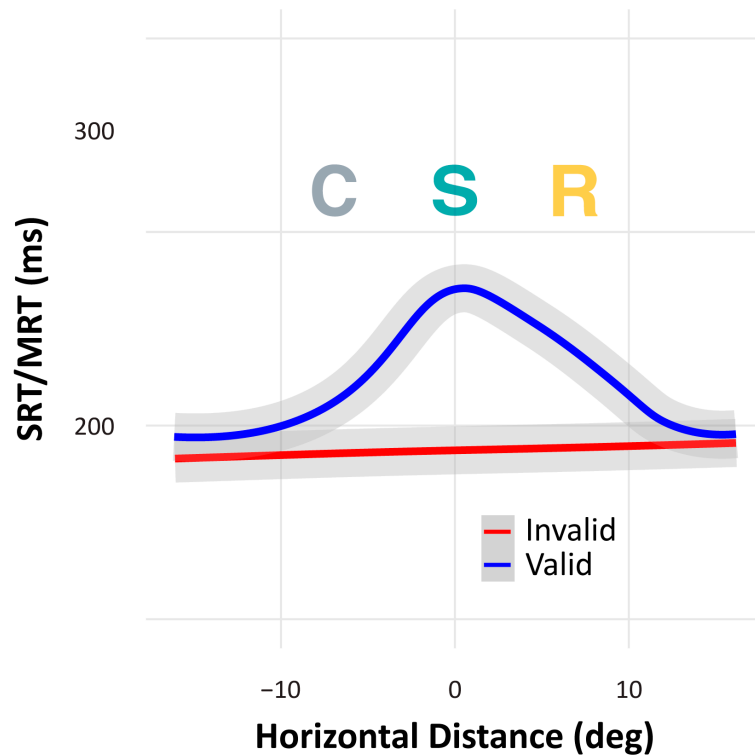
The tough (gradient) part

Independent retino source?



The tough (gradient) part

A part of spatio gradient?



The tough (gradient) part

- The relationship might be non-linear

GAMMs: Generalized Additive (Mixed) Models

- Non-linear modelling, incl. nonlinear interactions
 - **Generalized**: variety of distributions (like GLM: normal, binomial, Gamma, Student t (“scat”) etc.)
 - **Additive**: terms add together
 - **Mixed**: can handle random effects
-

GAMMs in R

- `mgcv` package (Simon Wood)
 - `gamm4`: `mgcv` + `lme4`
 - `itsadug`: for plotting
 - `mgcVIZ`: for plotting with `ggplot` objects
-

library(mgcv)

- `gam_mod <- gam(y ~ x1 + s(x2), data = mydata)`
 - can handle more distributions
 - `bam_mod <- bam(y ~ x1 + s(x2), data = mydata)`
 - can handle autocorrelations (AR1)
 - faster, for bigger data
-

library(mgcv)

gam(y ~ x1 – linear term, could be multiple, could be
interactions (x1*x3)

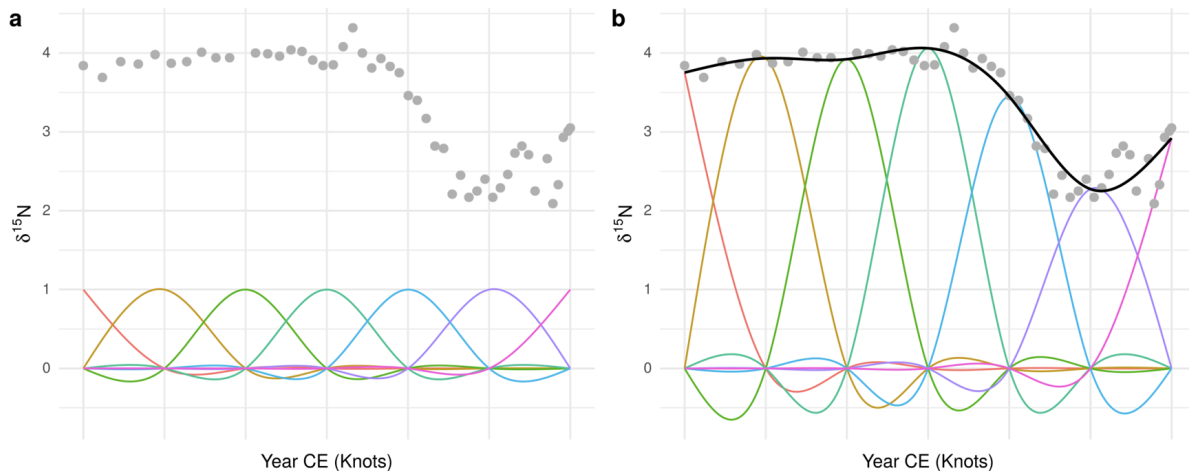
+ s(x2), – non-linear smooth

data = mydata)

GAMMs

- Use basis functions to model nonlinear smooths

from Noam Ross'
presentation



Basis functions

- Can have 1 and more dimensions
- Default: thin plate regression (“tp”)

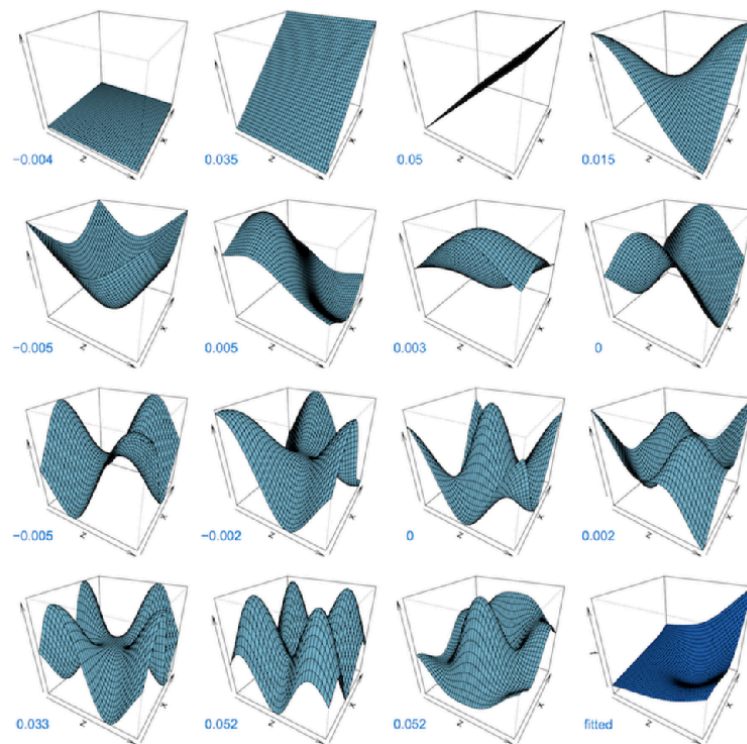
`gam(y ~ s(x, bs = “tp”, k = 20), data = mydata)`

- Other (ca. 17): cubic (“cr”) - simple, cyclic (“cc”) – seasonal, angular, etc.
 - ?smooth.terms
-

thin plate

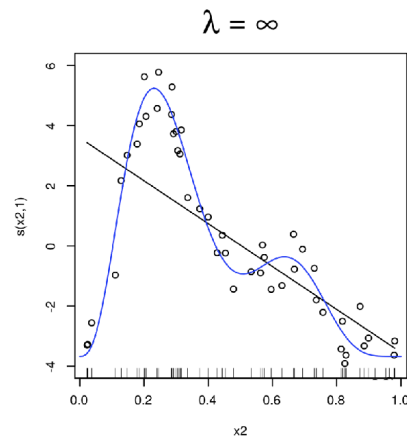
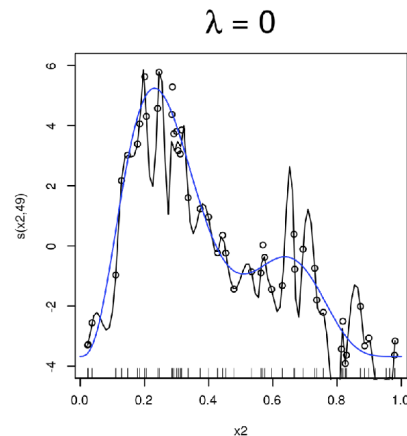
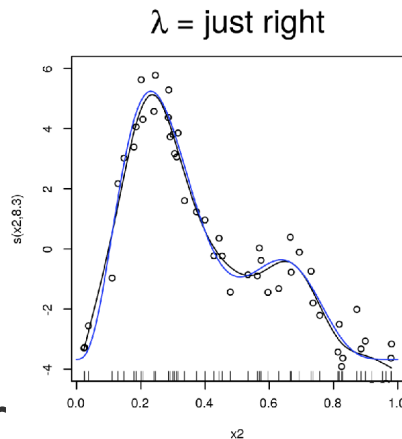
- multidimensional
- uses all data, not “knots”

from Noam Ross'
presentation



Wiggleness

Smoothing parameter



from Noam Ross'
presentation

$$\text{Log}(L) - \lambda W$$

Likelihood Wiggleness

Smoothing parameter

- `gam_mod$sp`
- `gam(y ~ s(x, bs = "tp", k = 20), data = mydata, sp = 0.1)`

mgcv will do it for you and better than you



Smoothing parameter

```
gam_mod <- gam(y ~ x1 + s(x2), data = mydata,  
               family = "gaussian", method = "REML")
```

- REML: best but hard to compare if different parametric terms
 - ML
 - GCV (default): overfitting
-

Rule of thumb

Start from default settings unless you have a really good reason

But keep in mind exceptions

What you can model

- $s(x_1) + s(x_3)$ – additive
 - $s(x_1, x_3)$ – interaction (isotropic: both in deg)
 - $te(x_1, x_3)$ – tensor (not isotropic: deg and sec)
 - $ti(x_1, x_3)$ – only marginal
 - $s(x_1, by = x_2)$ – factor-smooth interactions
 - more complex interactions, etc
-

Random effects

- Random intercept:
 - `gam(y ~ s(x1, bs = "re"), data=mydata)`
 - Random intercept & slope:
 - `gam(y ~ s(x1, x2, bs = "re"), data=mydata)`
 - Random smooth (incl. slope & intercept)
 - `gam(y ~ s(x2, x1, bs = "re", m=1), data=mydata)`
-

Random effects

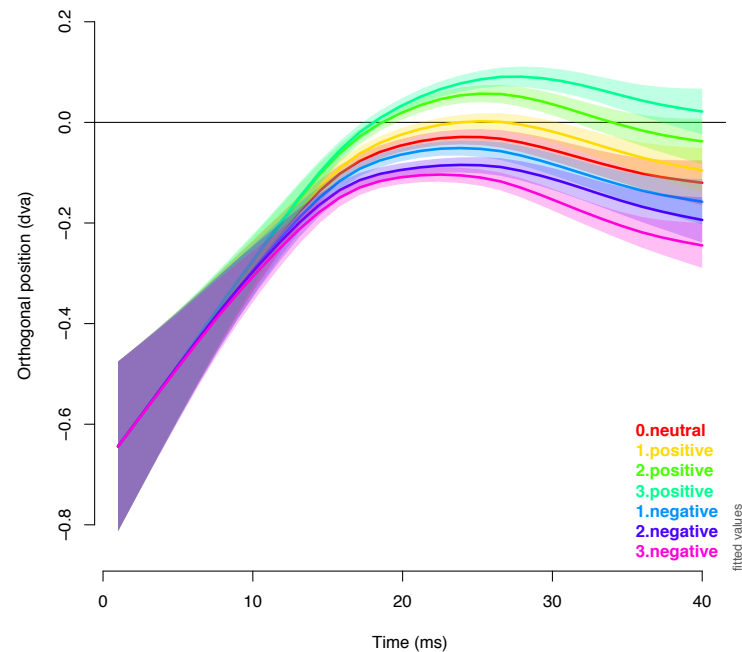
- Complex structure doesn't mean the best-fit
 - Always check
 - Mind the order
 - If you have plenty, use gamm4 package
-

Factor-smooth interactions

- Non-ordered factor: smooth for each, intercept at 0
- Ordered: intercept = ref level; each smooth = difference smooth

Post-hoc tests

- I have 7 levels in a factor



Post-hoc tests

- Well... just give up



Post-hoc tests

- You can use ordered factors (up to 3 levels; relative to the reference)
 - You can set contrasts (no such thing as `emmeans::contrast` for `lme`) but you can try manually
 - You can rely on plots
-

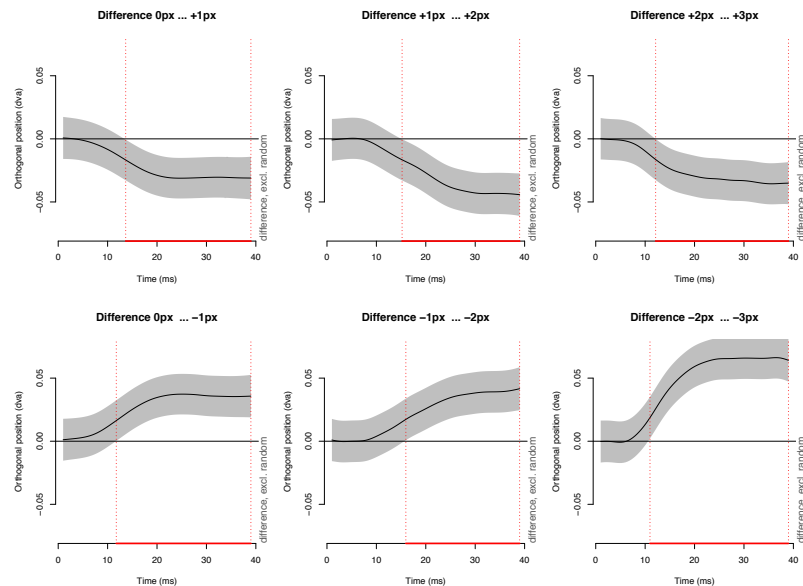
Post-hoc tests

```
> coef(contrast(emfinAHOOffsetDeg, "consec", adjust = "fdr"))
  offsetDir degrees c.1 c.2 c.3 c.4 c.5 c.6 c.7 c.8 c.9 c.10 c.11 c.12
1 3.positive   Five -1  0  0  0  0  0  0  0  0  0  0  0
2 2.positive   Five  1 -1  0  0  0  0  0  0  0  0  0  0
3 1.positive   Five  0  1 -1  0  0  0  0  0  0  0  0  0
4 0.neutral    Five  0  0  1 -1  0  0  0  0  0  0  0  0
5 1.negative   Five  0  0  0  1 -1  0  0  0  0  0  0  0
6 2.negative   Five  0  0  0  0  1 -1  0  0  0  0  0  0
7 3.negative   Five  0  0  0  0  0  1  0  0  0  0  0  0
8 3.positive   Ten  0  0  0  0  0  0 -1  0  0  0  0  0
9 2.positive   Ten  0  0  0  0  0  0  1 -1  0  0  0  0
10 1.positive   Ten  0  0  0  0  0  0  0  1 -1  0  0  0
11 0.neutral    Ten  0  0  0  0  0  0  0  0  1 -1  0  0
12 1.negative   Ten  0  0  0  0  0  0  0  0  0  1 -1  0
13 2.negative   Ten  0  0  0  0  0  0  0  0  0  0  1 -1
14 3.negative   Ten  0  0  0  0  0  0  0  0  0  0  0  1
```

Post-hoc tests

```
plot_diff(gam_mod, view="x1",  
comp=list(fac=c("level 1","level 5")))
```

significant difference doesn't mean
the result is significant



Checking & comparing

- `gam.check(model)`
 - `summary(model)`
 - `Itsadug: CompareML(model1, model2)`
 - `anova(model1, model2, test="Chisq")`: must be nested
-

Our full model

```
full_mod <- bam(SRT ~ validity
  + s(dX)
  + s(dX, by = validity)
  + s(FTOA)
  + s(Euclidian distance)
  + s(SRT2)
  + ti(FTOA, dX, by=validity)
  + s(FTOA, Subject, bs='fs', m=1),
  data = gamdt, method = "ML", family = Gamma(link = "identity"))
```

Our full model

```
par(mfrow=c(2,2))  
gam.check(full_mod)
```

Our full model

Basis dimension (k) checking results. Low p-value (k-index<1) may indicate that k is too low, especially if edf is close to k'.

	k'	edf	k-index	p-value
s(dX)	9.00	1.00	0.99	0.240
s(dX):validityValid	9.00	1.00	0.99	0.290
s(FTOA)	9.00	5.40	0.99	0.145
s(Euclidian distance)	9.00	3.86	0.98	0.045 *
s(SRT2)	9.00	1.00	0.99	0.240
ti(FTOA,dX):validityValid	16.00	1.00	1.01	0.790
s(FTOA, Subject)	234.00	86.12	0.99	0.175

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Checking for k

```
rsd <- residuals(full_mod, type="deviance")  
gam(rsd~s(dX,k=20)-1, data=gamdt, select=TRUE)
```

```
#or  
rsd <- residuals(full_mod)  
bam(rsd ~ s(dX, k = 20), data=gamdt)
```

```
# look at AIC and other scores. Any pattern?
```

Our full model

If you want an easy summary
(but worse plotting):

```
summary(full_mod)
```

```
contrasts(gamdt$validity) <-  
  "contr.treatment"
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	249.989	4.462	56.031	< 2e-16 ***
validityValid	6.295	1.264	4.979	6.61e-07 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Our full model

summary(full_mod)

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(dX)	1.000	1.000	3.399	0.0653 .
s(dX):validityValid	1.000	1.000	0.233	0.6290
s(FTOA)	5.395	6.397	32.050	< 2e-16 ***
s(Euclidian distance)	3.858	4.748	18.828	< 2e-16 ***
s(SRT2)	1.000	1.000	49.368	2.41e-12 ***
ti(FTOA,dX):validityValid	1.001	1.002	0.000	0.9869
s(FTOA, Subject)	86.119	233.000	5.313	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) = 0.293 Deviance explained = 96.9%

-ML = -1455.5 Scale est. = 0.0307 n = 4849

Our full model

- plot – partial effects
 - plot_smooth – summed effects
 - plot_parametric – parametric
 - you can exclude random (rm.ranef=TRUE)
 - fvisgam – summed interactions
 - pvisgam – partial
-

Our fin model

```
fin_mod <- bam(SRT ~ validity
  + s(dX)
  + s(FTOA)
  + s(Euclidian distance)
  + s(SRT2)
  + s(FTOA, Subject, bs='fs', m=1),
  data = gamdt, method = "ML", family = Gamma(link = "identity"))
```

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	249.977	4.460	56.054	< 2e-16 ***
validityValid	6.303	1.264	4.987	6.36e-07 ***

Approximate significance of smooth terms:

	edf	Ref.df	F	p-value
s(dX)	1.000	1.001	5.998	0.0143 *
s(FTOA)	5.399	6.402	32.024	< 2e-16 ***
s(Euclidian distance)	3.859	4.749	18.800	< 2e-16 ***
s(SRT2)	1.000	1.000	49.204	2.62e-12 ***
s(FTOA, Subject)	86.159	233.000	5.314	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Main findings

- No independent retino IOR
- The whole valid hemifield is inhibited
- GAMMs are nice and we should use them more often



Resources

- Simon Wood – Generalised Additive Models – An introduction with R – but whatever you find
 - DataCamp course
 - https://www.youtube.com/watch?v=q4_t8jXcQgc
 - <https://www.fromthebottomoftheheap.net/tag/gam/>
-