Exploration of Data

Matthew Perrotta

5/10/2019

Load Libraries

```r
library(tidyverse)
library(mgcv)
library(corrplot)
library(Hmisc)
library(caret)
library(PerformanceAnalytics)
library(car)
library(olsrr)
#library(MASS)
#library(raster)
```

READ: https://www.nature.com/articles/s41598-017-02560-z

Import and clean data

Two datasets are created, `swk` for Sarawak and `sbh` for Sabah. Variables that are not needed are removed.

```r
swk = read.csv('./data/data_full_sar.csv') %>%
  group_by(district, year) %>%
  mutate(district_area =
           CropRainfed.total.area +
           Herbaceous.total.area +
           TreeShrub.total.area +
           MosCrop.total.area +
           MosNatural.total.area +
           BroadEvgClop.total.area +
           Shrub.total.area +
           ShrubEvg.total.area +
           SparseVeg.total.area +
           FloodFresh.total.area +
           FloodSalt.total.area +
           Urban.total.area +
           Water.total.area +
           CropIrrigate.total.area
         ) %>%
  ungroup() %>%
  group_by(district) %>%
  mutate(CropRainfed.prop = CropRainfed.total.area/district_area,
         Herbaceous.prop = Herbaceous.total.area/district_area,
         TreeShrub.prop = TreeShrub.total.area/district_area,
         MosCrop.prop = MosCrop.total.area/district_area,
         MosNatural.prop = MosNatural.total.area/district_area,
         BroadEvgClop.prop = BroadEvgClop.total.area/district_area,
```

```
         Shrub.prop = Shrub.total.area/district_area,
         ShrubEvg.prop = ShrubEvg.total.area/district_area,
         SparseVeg.prop = SparseVeg.total.area/district_area,
         FloodFresh.prop = FloodFresh.total.area/district_area,
         FloodSalt.prop = FloodSalt.total.area/district_area,
         Urban.prop = Urban.total.area/district_area,
         Water.prop = Water.total.area/district_area,
         CropIrrigate.prop = CropIrrigate.total.area/district_area) %>%
  janitor::clean_names() %>%
  ungroup() %>%
  select(-c('x',
           'population_year',
           'cases_year',
           'expected',
           'sd')) %>%
  rename('smr' = 'sir',
         'prec_mean' = 'mean',
         'cases' = 'case_number') %>%
  na.omit()


sbh = read.csv('./data/dataFull.csv') %>%
  group_by(District, Year) %>%
  mutate(district_area =
           CropRainfed.total.area +
           Herbaceous.total.area +
           TreeShrub.total.area +
           MosCrop.total.area +
           MosNatural.total.area +
           BroadEvgClop.total.area +
           Shrub.total.area +
           ShrubEvg.total.area +
           SparseVeg.total.area +
           FloodFresh.total.area +
           FloodSalt.total.area +
           Urban.total.area +
           Water.total.area +
           CropIrrigate.total.area +
           Grass.total.area +
           MosTreeHerb.total.area
         ) %>%
  ungroup() %>%
  group_by(District) %>%
  mutate(CropRainfed.prop = CropRainfed.total.area/district_area,
         Herbaceous.prop = Herbaceous.total.area/district_area,
         TreeShrub.prop = TreeShrub.total.area/district_area,
         MosCrop.prop = MosCrop.total.area/district_area,
         MosNatural.prop = MosNatural.total.area/district_area,
         BroadEvgClop.prop = BroadEvgClop.total.area/district_area,
         Shrub.prop = Shrub.total.area/district_area,
         ShrubEvg.prop = ShrubEvg.total.area/district_area,
         SparseVeg.prop = SparseVeg.total.area/district_area,
         FloodFresh.prop = FloodFresh.total.area/district_area,
```

```r
       FloodSalt.prop = FloodSalt.total.area/district_area,
       Urban.prop = Urban.total.area/district_area,
       Water.prop = Water.total.area/district_area,
       CropIrrigate.prop = CropIrrigate.total.area/district_area,
       Grass.prop = Grass.total.area/district_area,
       MosTreeHerb.prop = MosTreeHerb.total.area/district_area) %>%
janitor::clean_names() %>%
ungroup() %>%
select(-c('x',
          'disease',
          'number_deaths',
          'mortality_rates',
          'prevalence',
          'incidence_rate')) %>%
rename('cases' = 'number_cases') %>%
na.omit()
```

Observations with `NA` values were omitted from the data set

Land Cover variable discriptions:

- `crop_rainfed` - Cropland, rainfed
- `crop_irrigate` - Cropland, irrigated or post-flooding
- `mos_crop` - Mosaic cropland (>50%) / natural vegetation (tree, shrub, herbaceous cover)(<50%)
- `mos_natural` - Mosaic natural vegetation (tree, shrub, herbaceous cover) (>50%) / cropland (<50%)
- `mos_tree_herb` - Mosaic tree and shrub (>50%) / herbaceous cover (<50%)
- `herbaceous` - Herbaceous cover
- `tree_shrub` - Tree or shrub cover

- `broad_evg_clop` - Tree cover, broadleaved, evergreen, closed to open (>15%)
- `shrub` - Shrubland
- `shrub_evg` - Shrubland evergreen

- `sparse_veg` - Sparse vegetation (tree, shrub, herbaceous cover) (<15%)
- `flood_fresh` - Tree cover, flooded, fresh or brakish water
- `flood_salt` - Tree cover, flooded, saline water

- `grass` - Grassland
- `urban` - Urban areas

- `water` - Water bodies

The Sarawak dataset does not have the following variables:

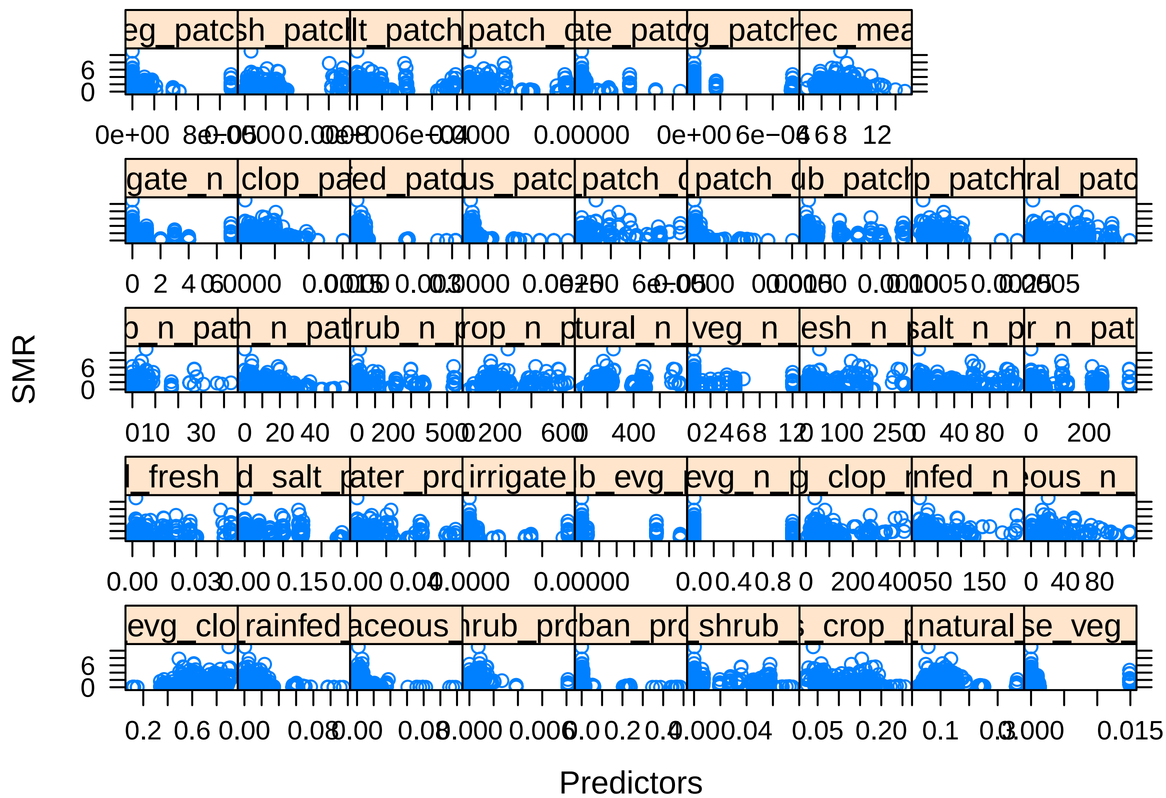
- `grass_n_patches`
- `grass_patch_density`
- `grass_total_area`
- `grass_prop`
- `mos_tree_herb_n_patches`
- `mos_tree_herb_patch_density`
- `mos_tree_herb_total_area`
- `mos_tree_herb_prop`

Creating a variable for agriculture

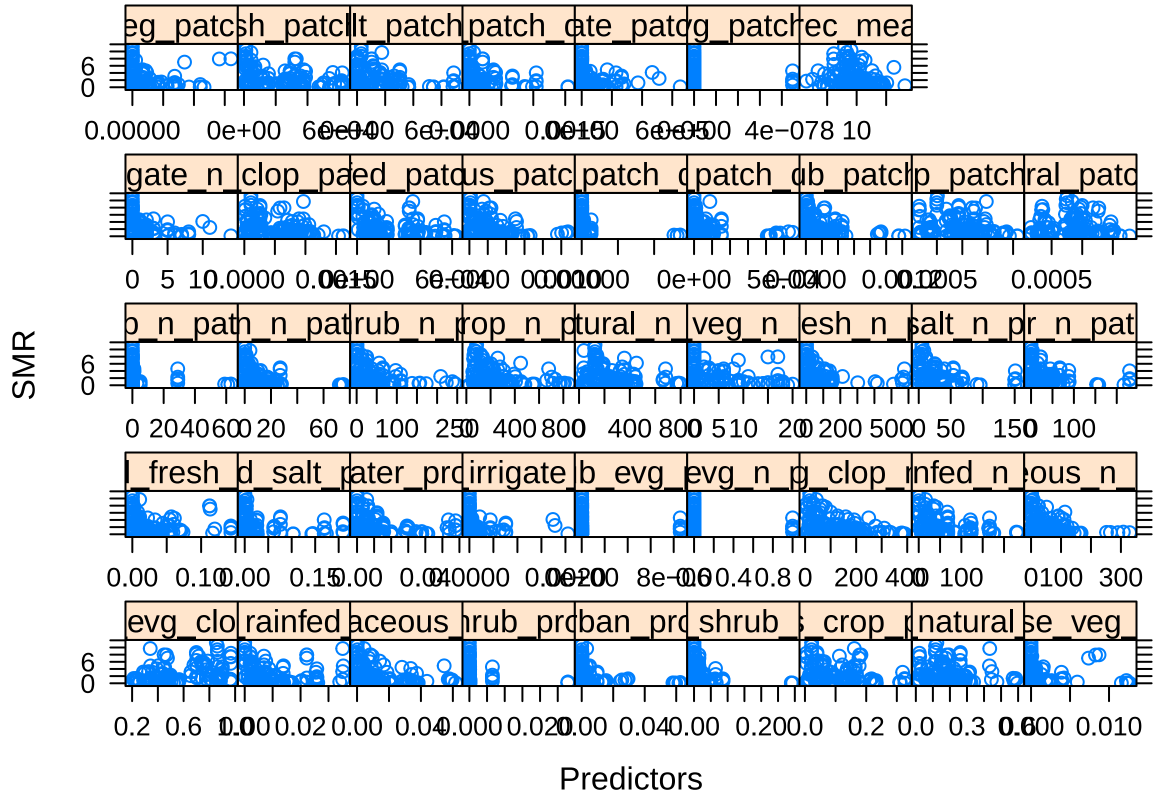
```
sbh = sbh %>%
  group_by(district, year) %>%
  mutate(agri_total_area = crop_rainfed_total_area +
           crop_irrigate_total_area +
           herbaceous_total_area +
           tree_shrub_total_area +
           mos_crop_total_area +
           mos_natural_total_area) %>%
  mutate(agri_prop = agri_total_area/district_area) %>%
  ungroup()

swk = swk %>%
  group_by(district, year) %>%
  mutate(agri_total_area = crop_rainfed_total_area +
           crop_irrigate_total_area +
           herbaceous_total_area +
           tree_shrub_total_area +
           mos_crop_total_area +
           mos_natural_total_area) %>%
  mutate(agri_prop = agri_total_area/district_area) %>%
  ungroup()
```

Exploration of the Data

```
sbh_sub = sbh %>%
  select(c("smr", "broad_evg_clop_prop", "crop_rainfed_prop", "herbaceous_prop", "shrub_prop", "urban_p

swk_sub = swk %>%
  select(c("smr", "broad_evg_clop_prop", "crop_rainfed_prop", "herbaceous_prop", "shrub_prop", "urban_p

sbh.x = model.matrix(smr~., sbh_sub, na.action = NULL)[,-1]
sbh.y = sbh_sub$smr

swk.x = model.matrix(smr~., swk_sub)[,-1]
swk.y = swk_sub$smr

featurePlot(sbh.x,
            sbh.y,
            plot = "scatter",
            labels = c("Predictors","SMR"),
            type = c('p'))
```

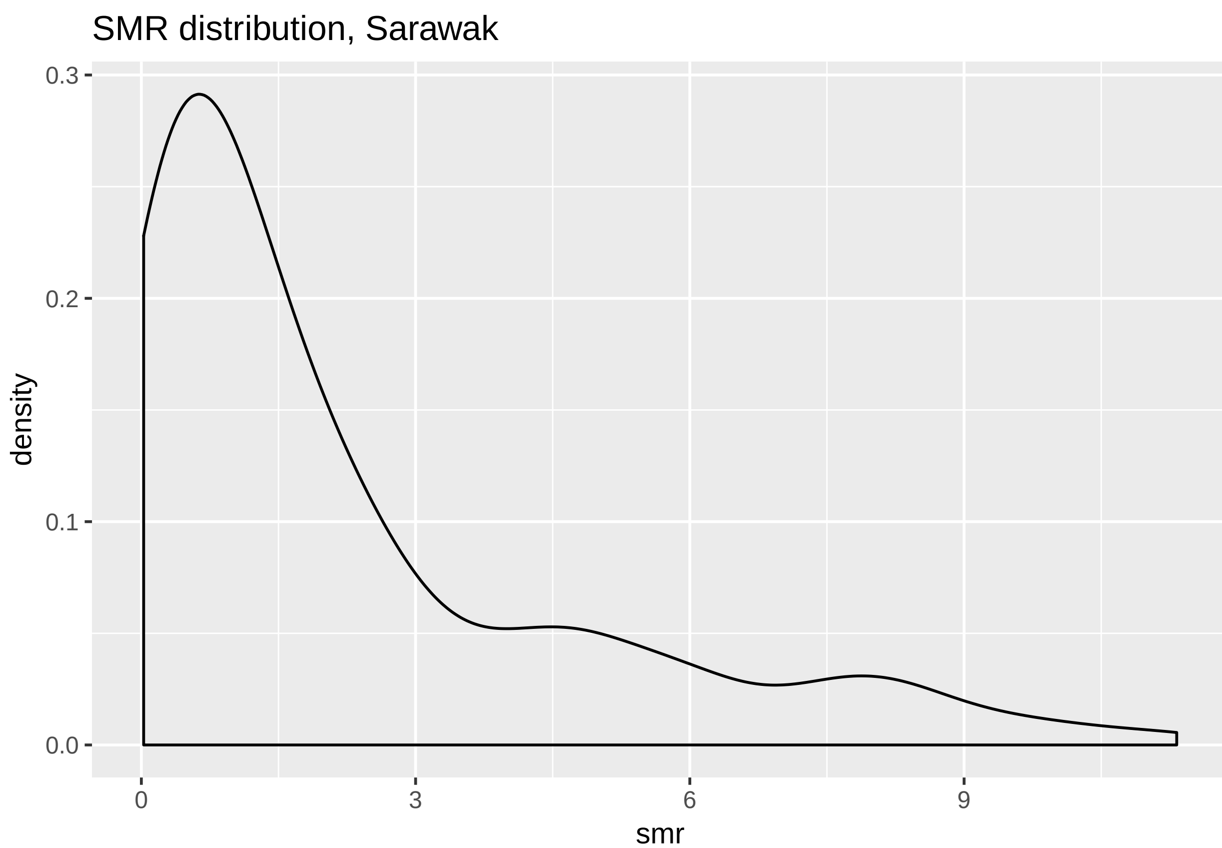



```r
featurePlot(swk.x,
            swk.y,
            plot = "scatter",
            labels = c("Predictors","SMR"),
            type = c('p'))
```

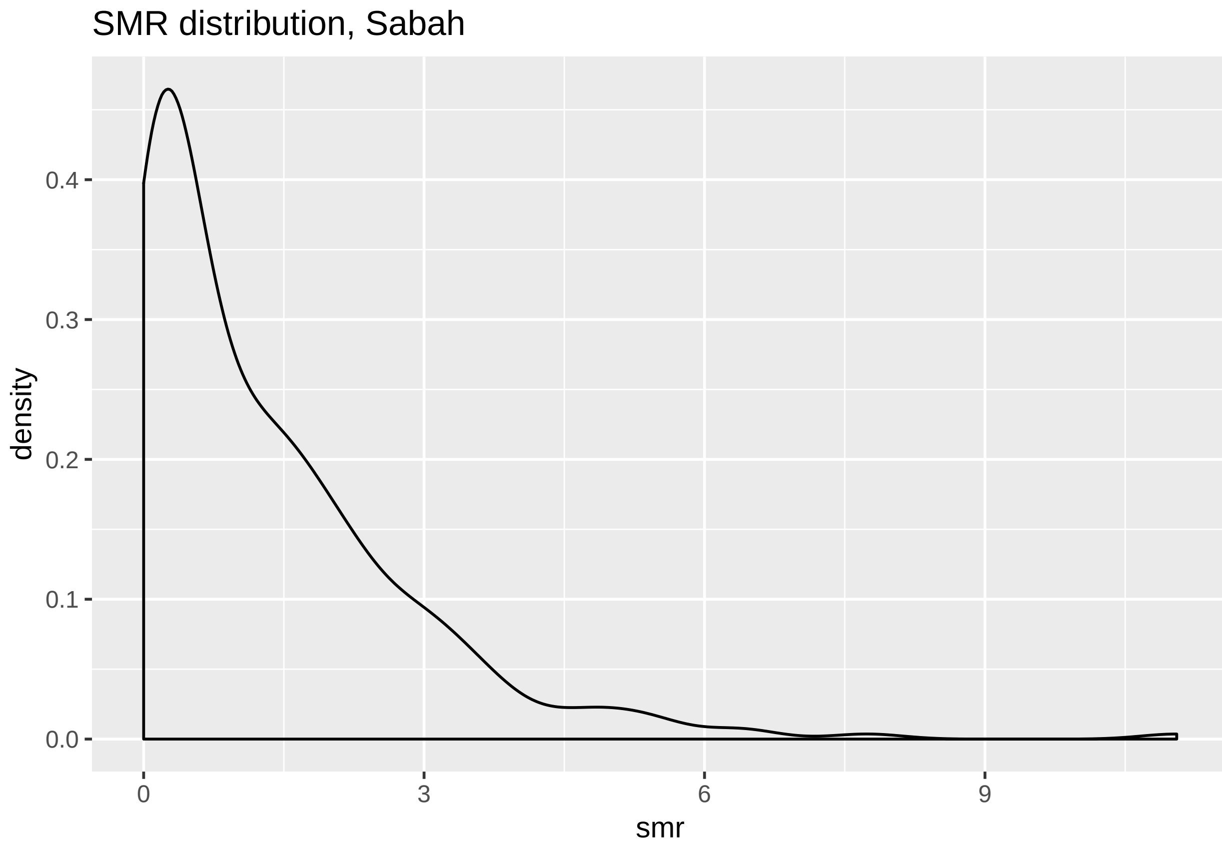


Distribution of SMR

```
ggplot(data = swk, aes(x = smr)) +
  geom_density() +
  labs(title = 'SMR distribution, Sarawak')
```

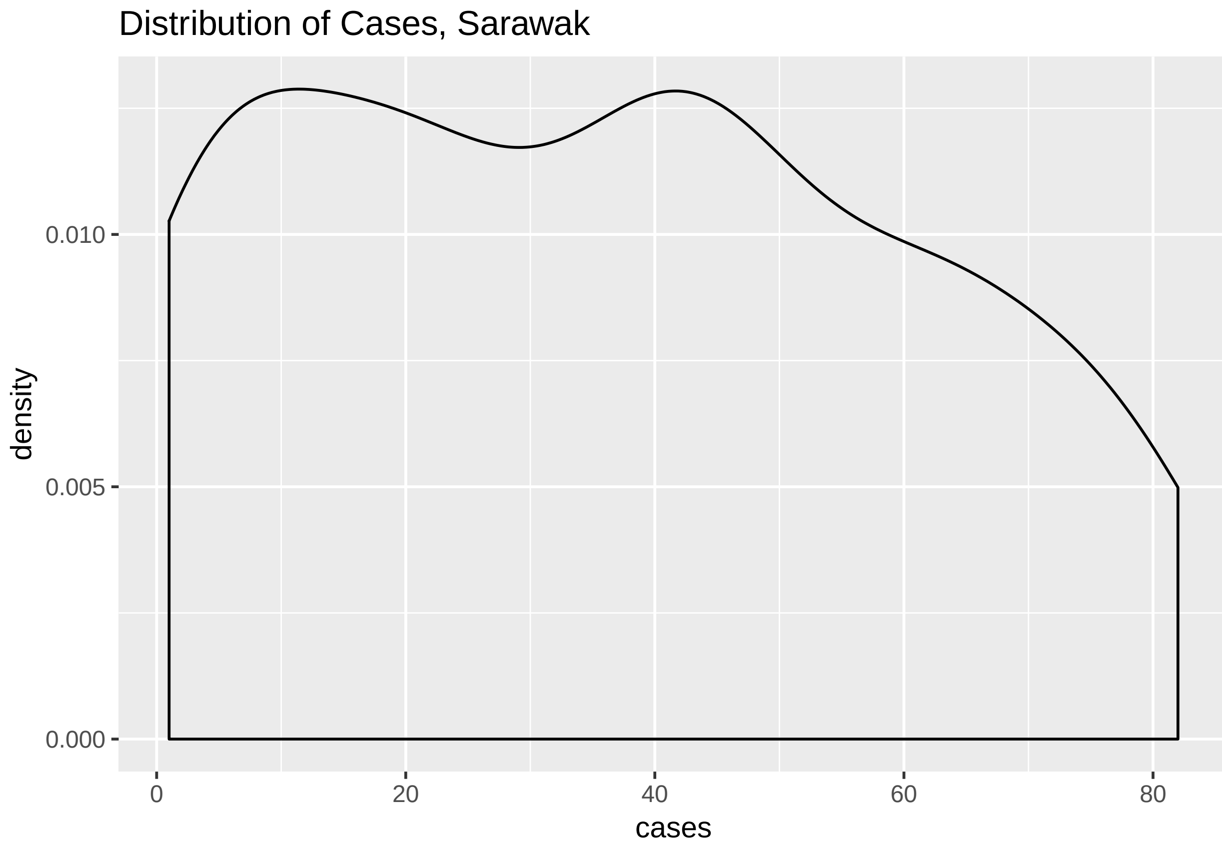



```r
ggplot(data = sbh, aes(x = smr)) +
  geom_density() +
  labs(title = 'SMR distribution, Sabah')
```

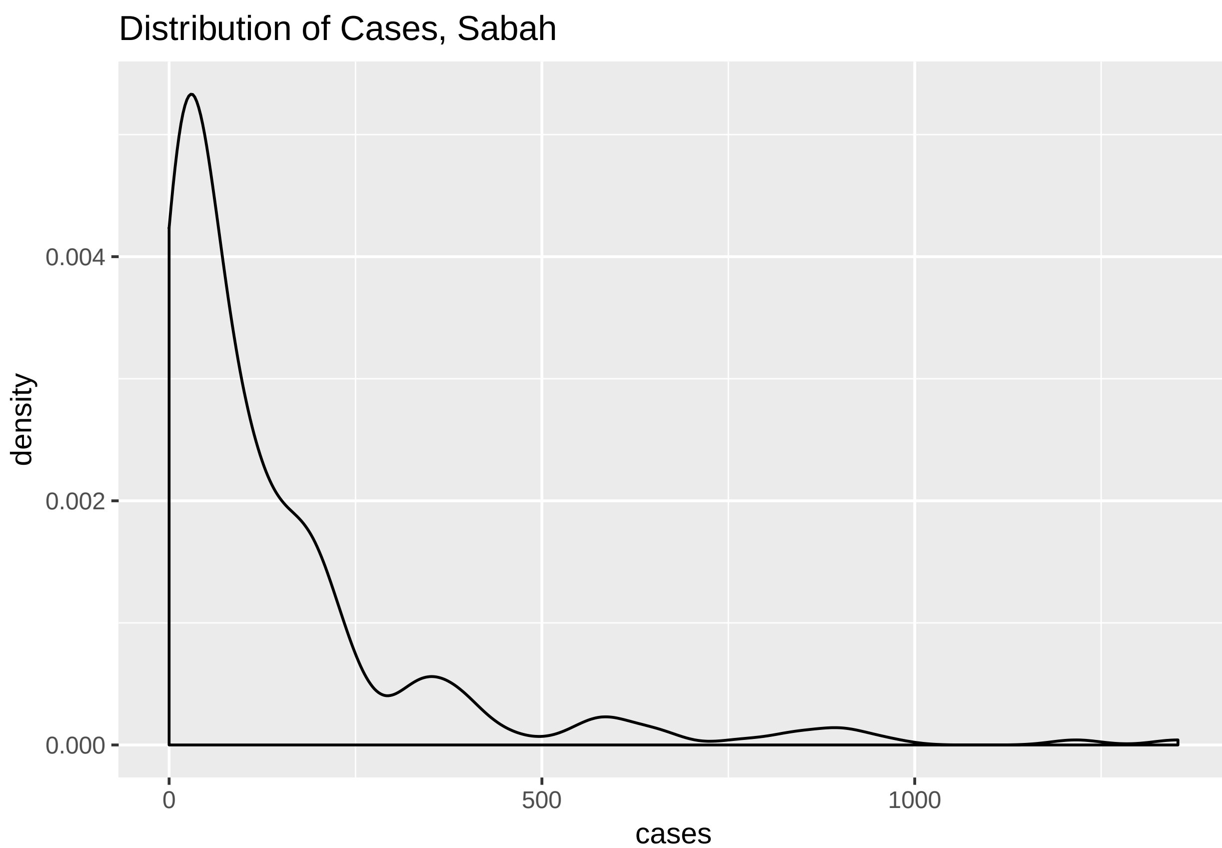


Distribution of cases

```
ggplot(data = swk, aes(x = cases)) +
  geom_density() +
  labs(title = 'Distribution of Cases, Sarawak')
```

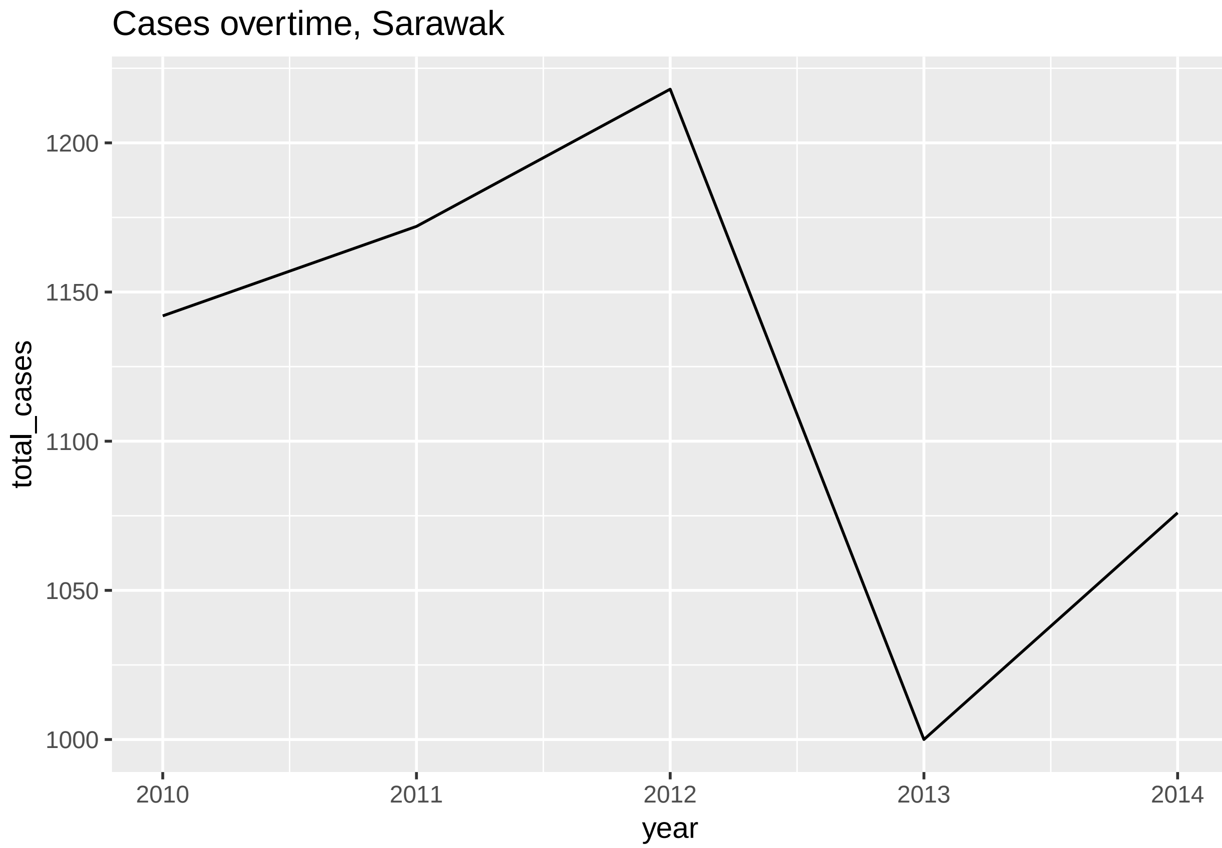



```r
ggplot(data = sbh, aes(x = cases)) +
  geom_density() +
  labs(title = 'Distribution of Cases, Sabah')
```

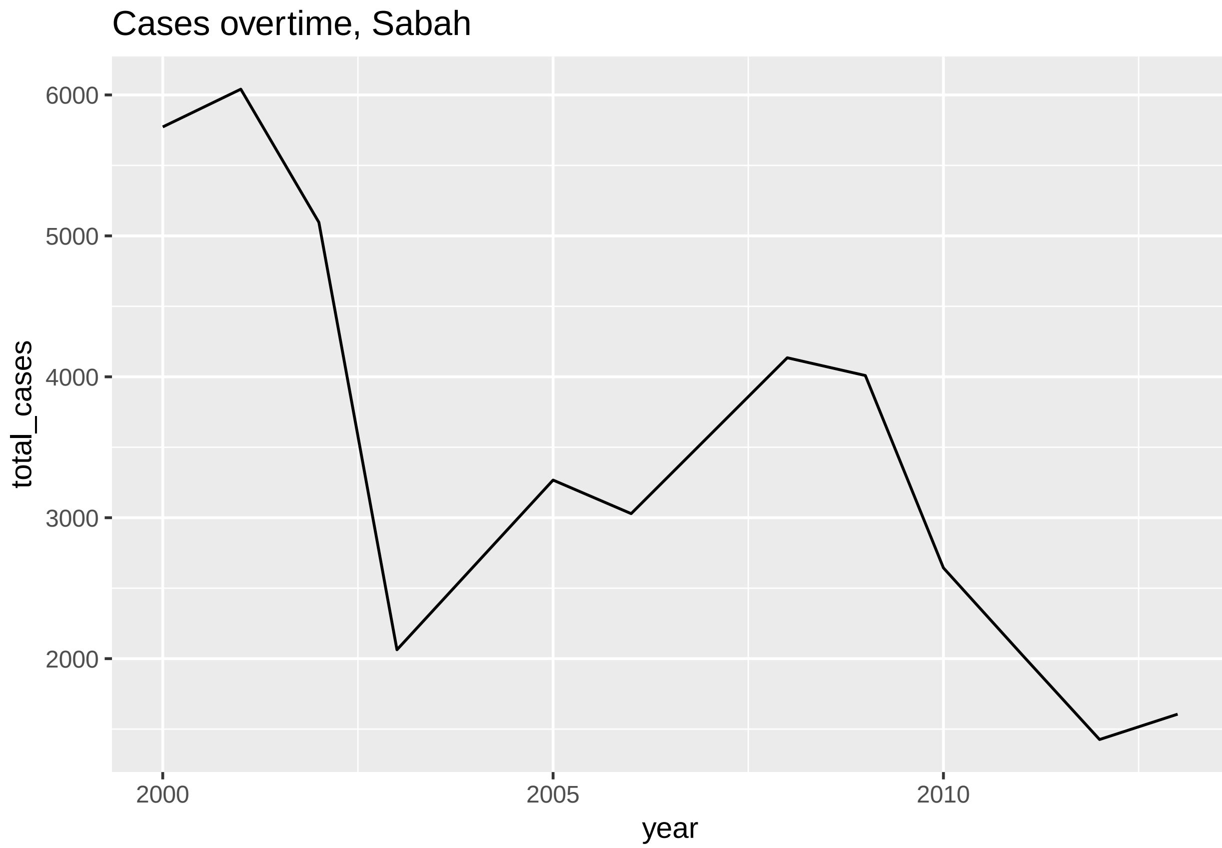


Cases overtime

```
swk %>%
  group_by(year) %>%
  mutate(total_cases = sum(cases)) %>%
  ggplot(aes(y = total_cases, x = year)) +
  geom_line() +
  labs(title = 'Cases overtime, Sarawak')
```



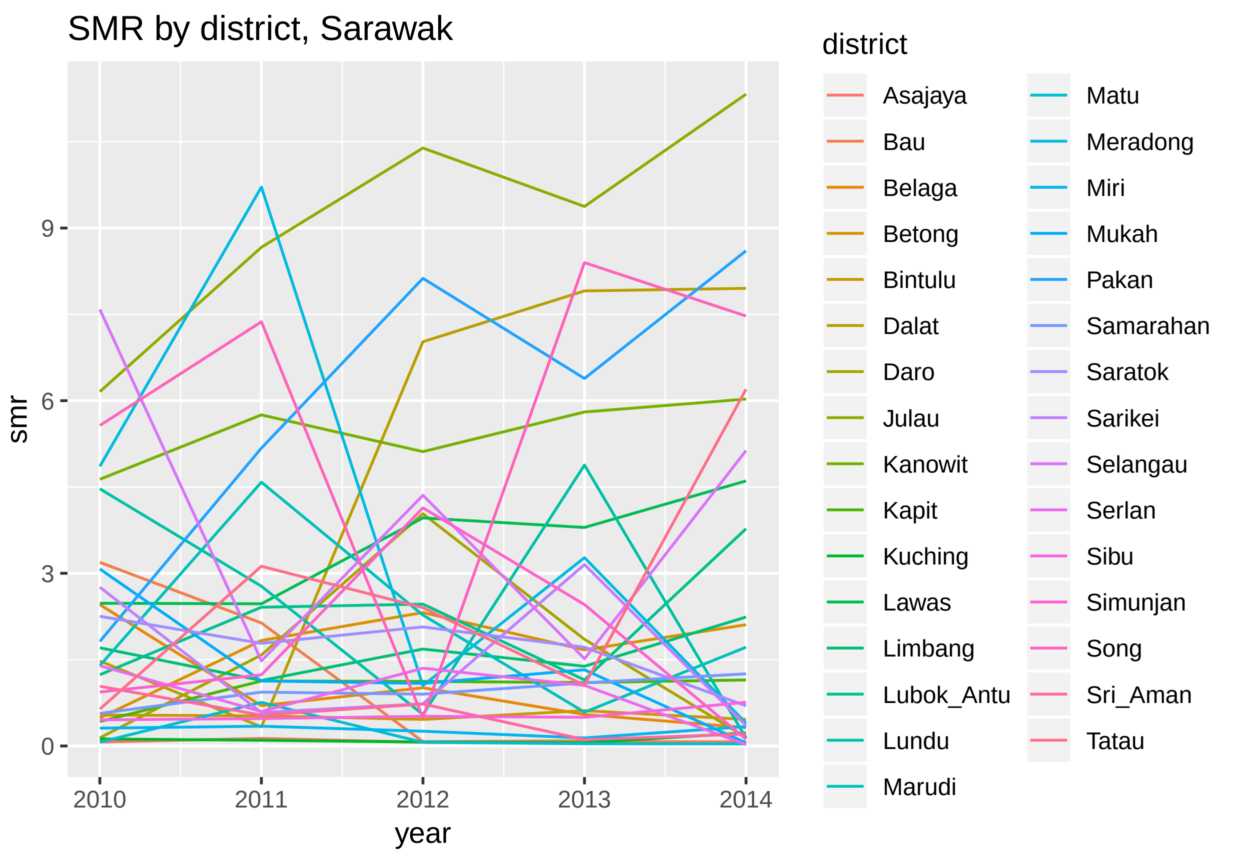

```r
sbh %>%
  group_by(year) %>%
  mutate(total_cases = sum(cases)) %>%
  ggplot(aes(y = total_cases, x = year)) +
  geom_line() +
  labs(title = 'Cases overtime, Sabah')
```



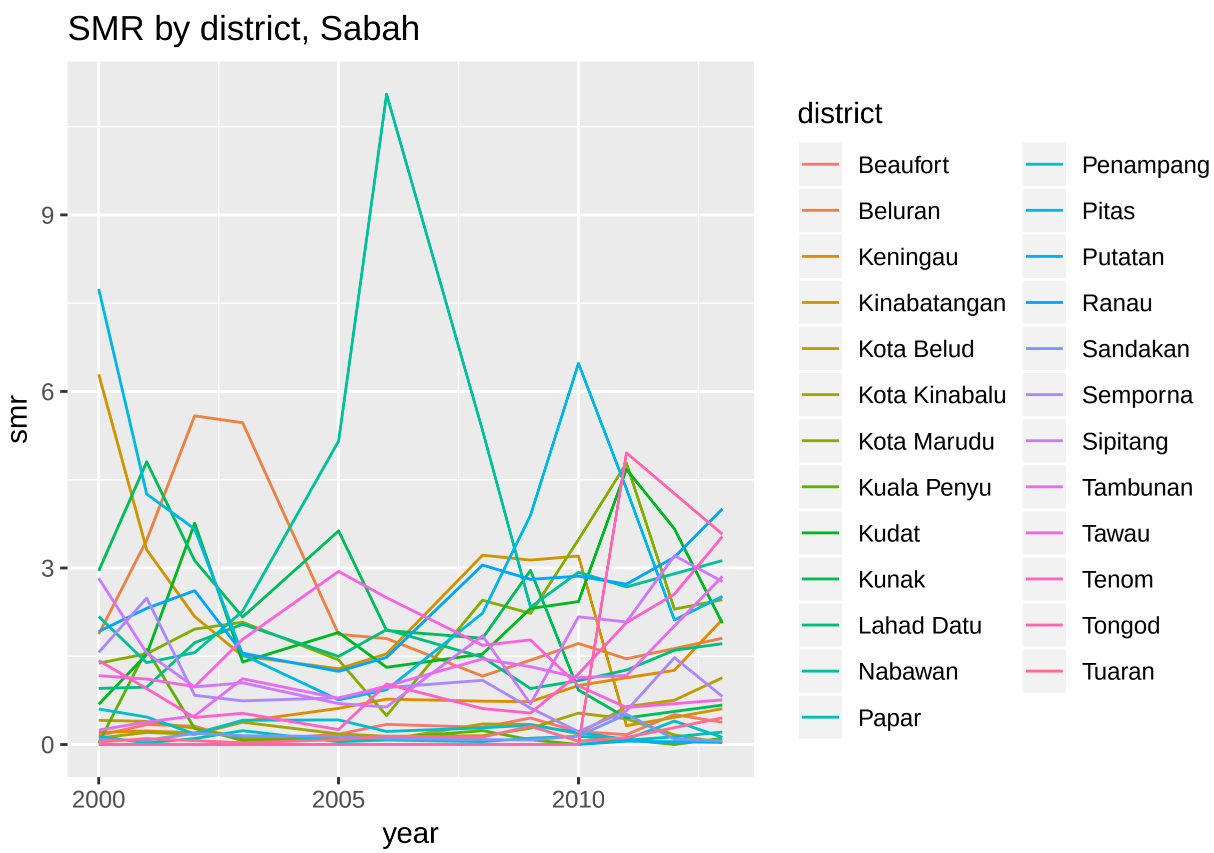
Spaghetti Plots

SMR by District

```
swk %>%
  ggplot(aes(y = smr, x = year, color = district)) +
  geom_line() +
  labs(title = 'SMR by district, Sarawak')
```

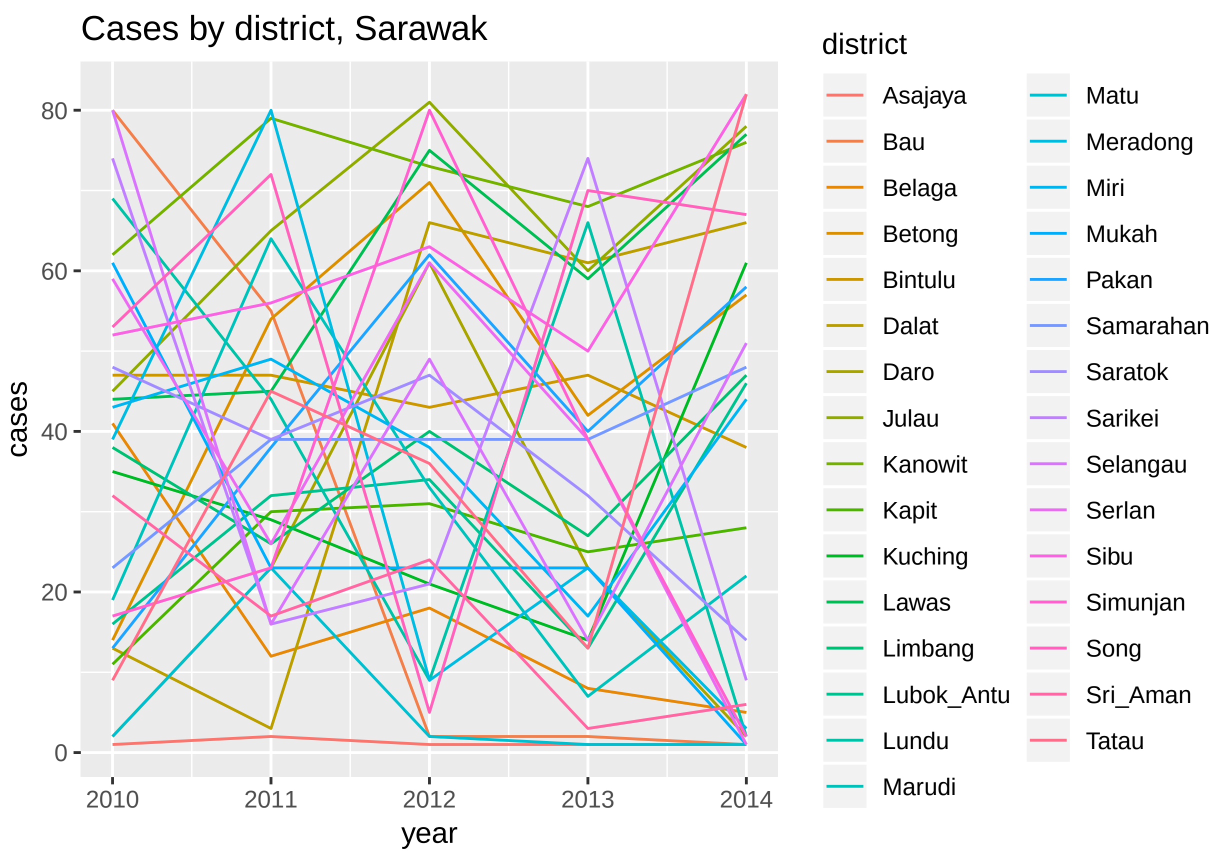



```
sbh %>%
  ggplot(aes(y = smr, x = year, color = district)) +
  geom_line() +
  labs(title = 'SMR by district, Sabah')
```

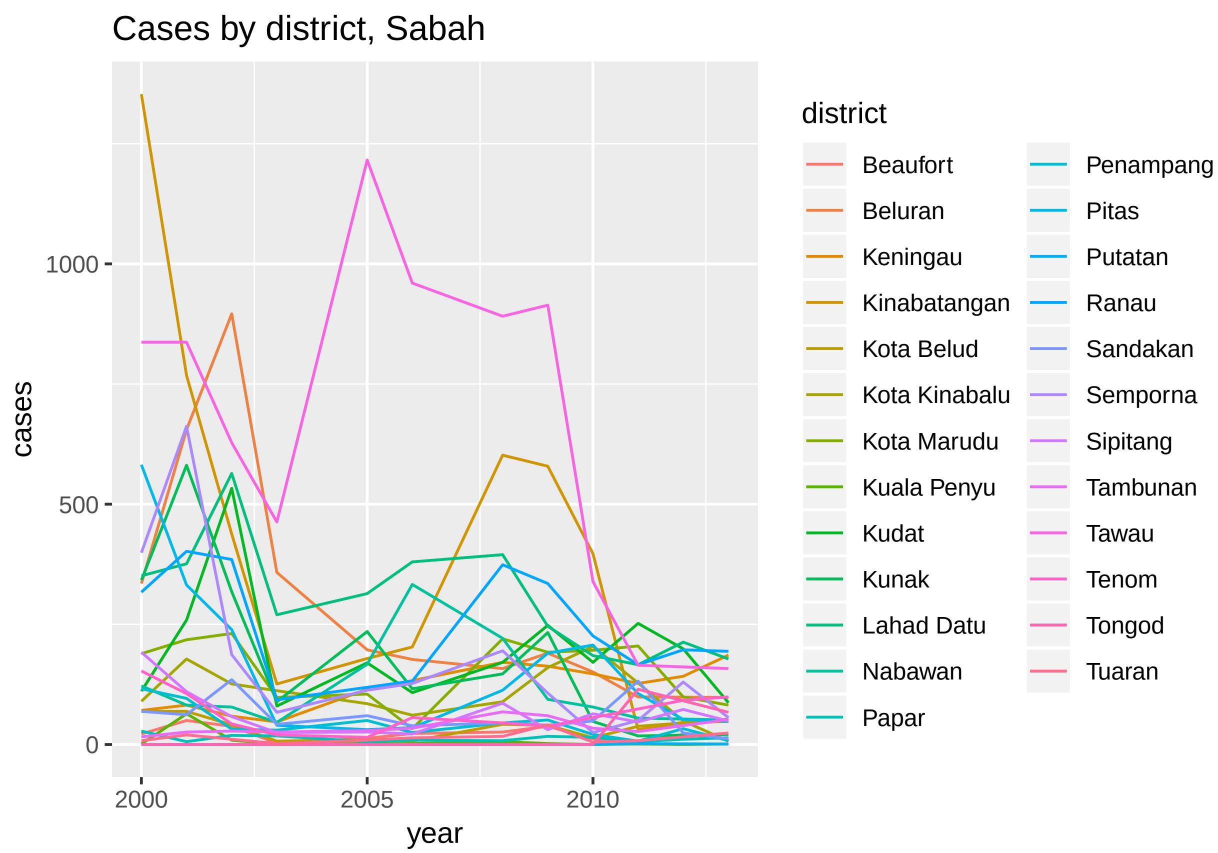


Cases by district

```
swk %>%
  ggplot(aes(y = cases, x = year, color = district)) +
  geom_line() +
  labs(title = 'Cases by district, Sarawak')
```

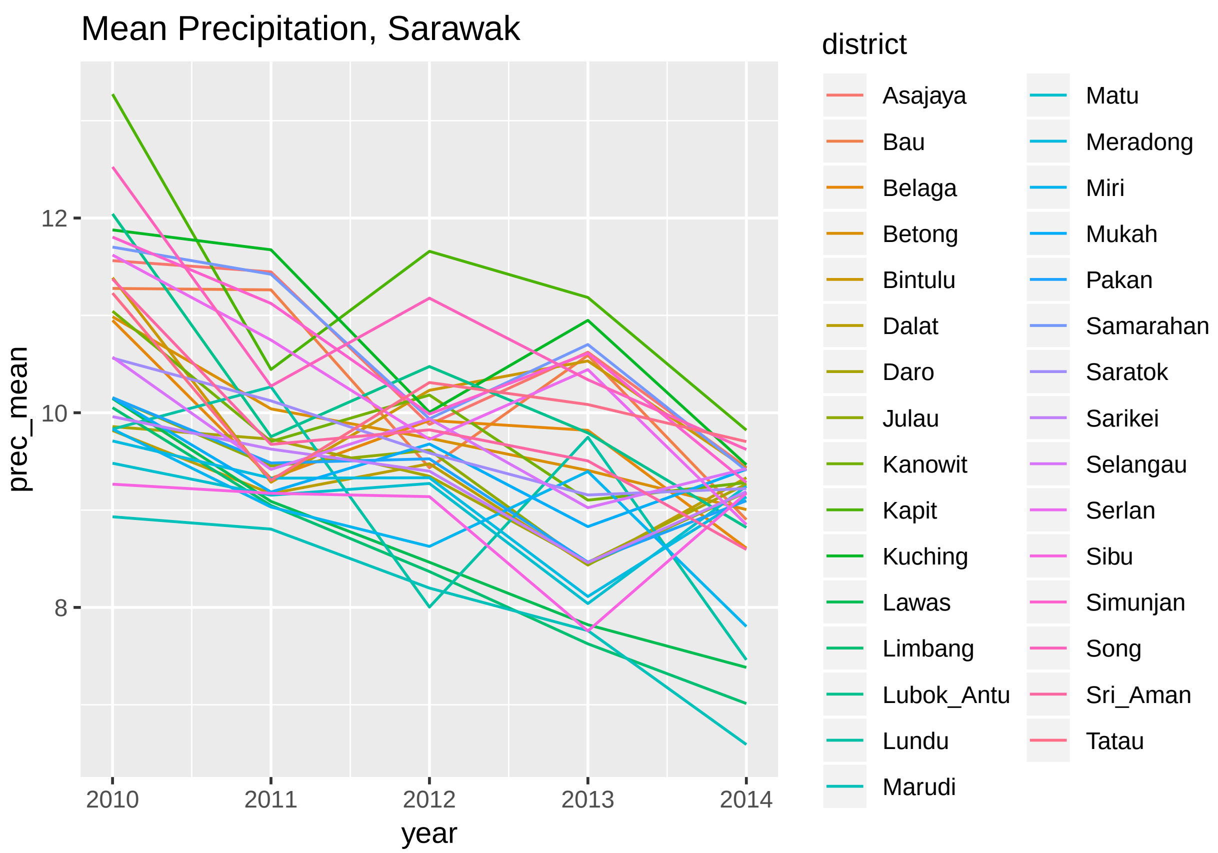



```r
sbh %>%
  ggplot(aes(y = cases, x = year, color = district)) +
  geom_line() +
  labs(title = 'Cases by district, Sabah')
```

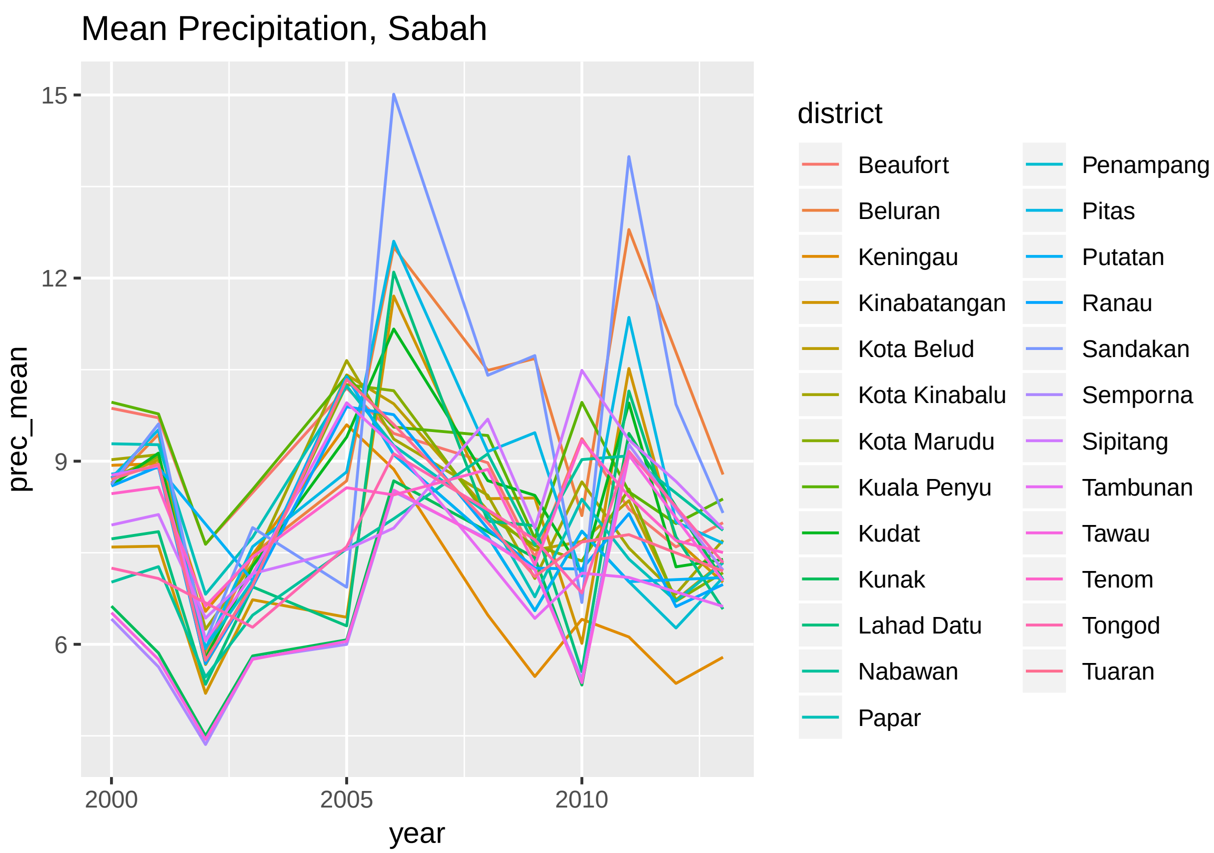


Mean precipitation across districts of Sabah and Sarawak

```r
ggplot(data = swk, aes(x = year, y = prec_mean, color = district)) +
  geom_line() +
  labs(title = 'Mean Precipitation, Sarawak')
```

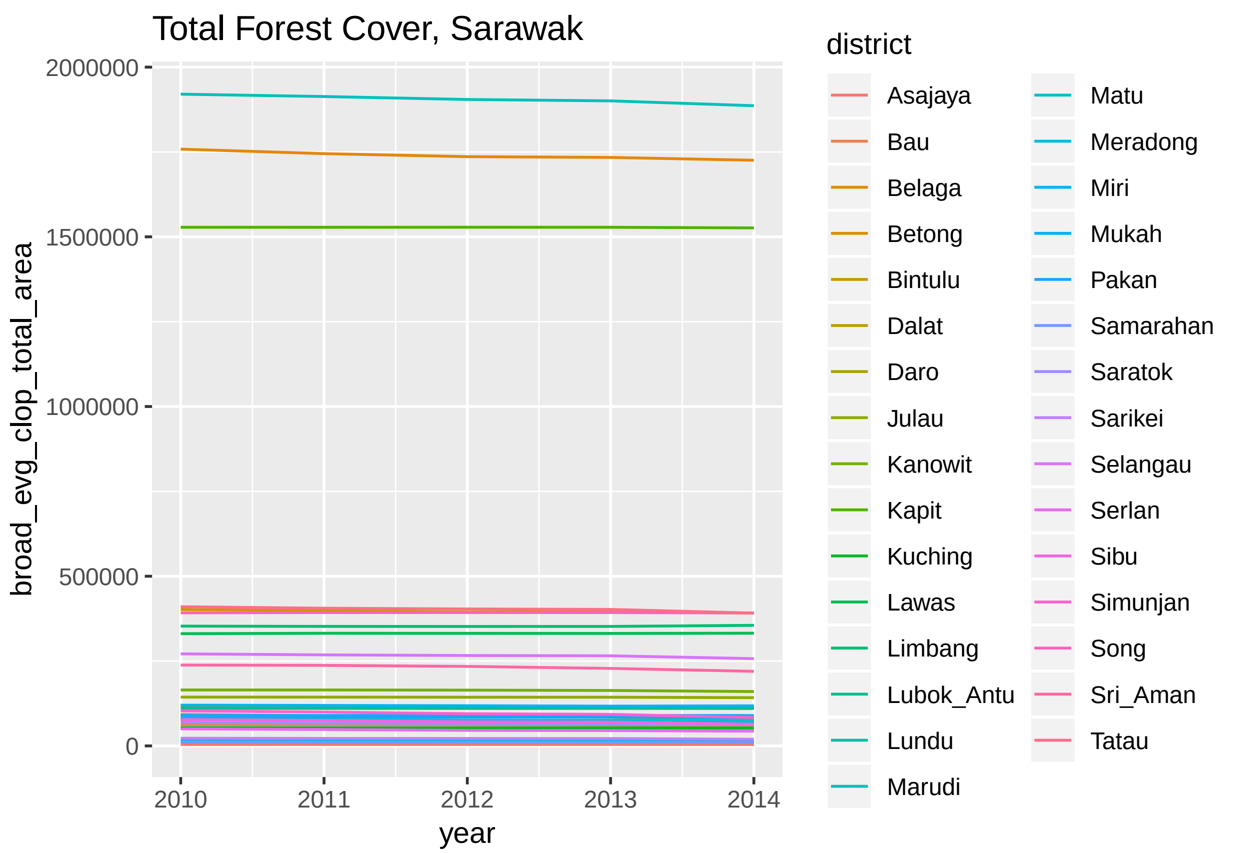



```r
ggplot(data = sbh, aes(x = year, y = prec_mean, color = district)) +
  geom_line() +
  labs(title = 'Mean Precipitation, Sabah')
```

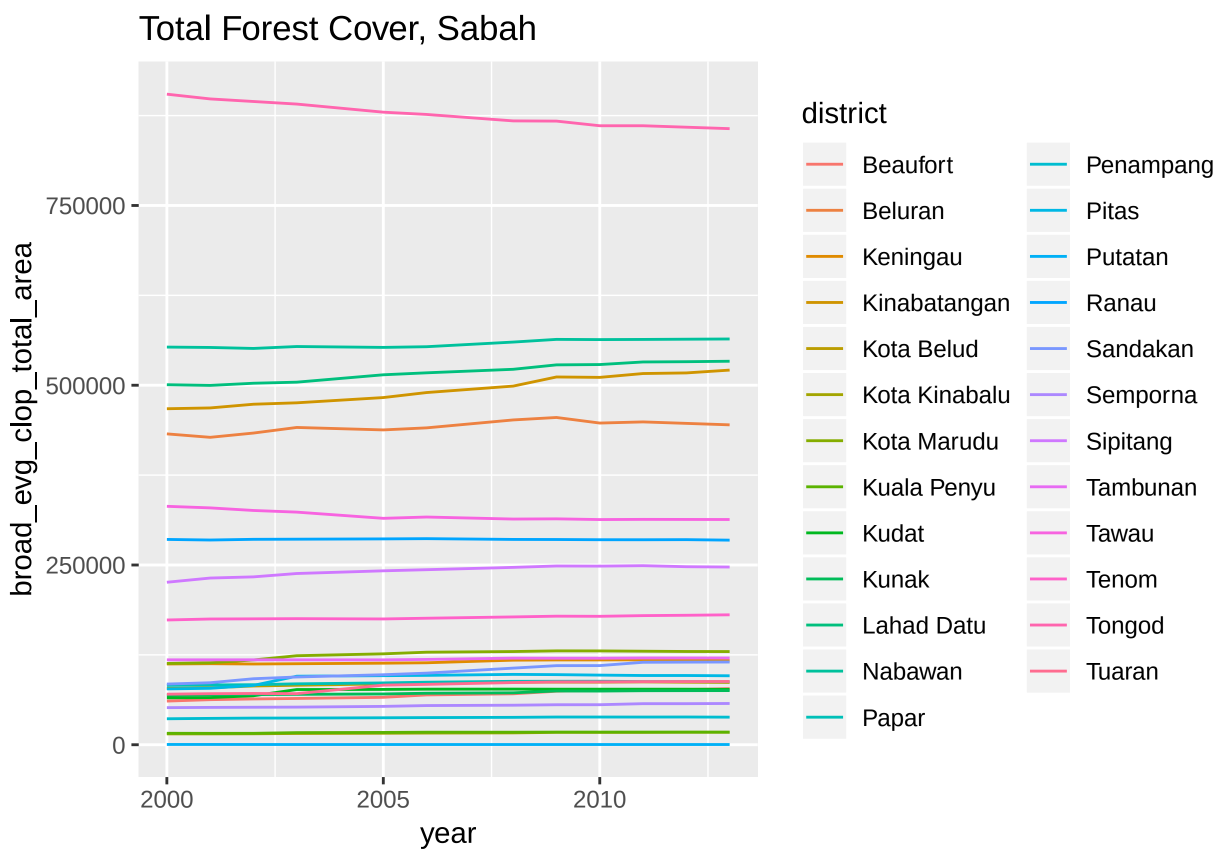


Forest cover across districts, Sarawak and Sabah

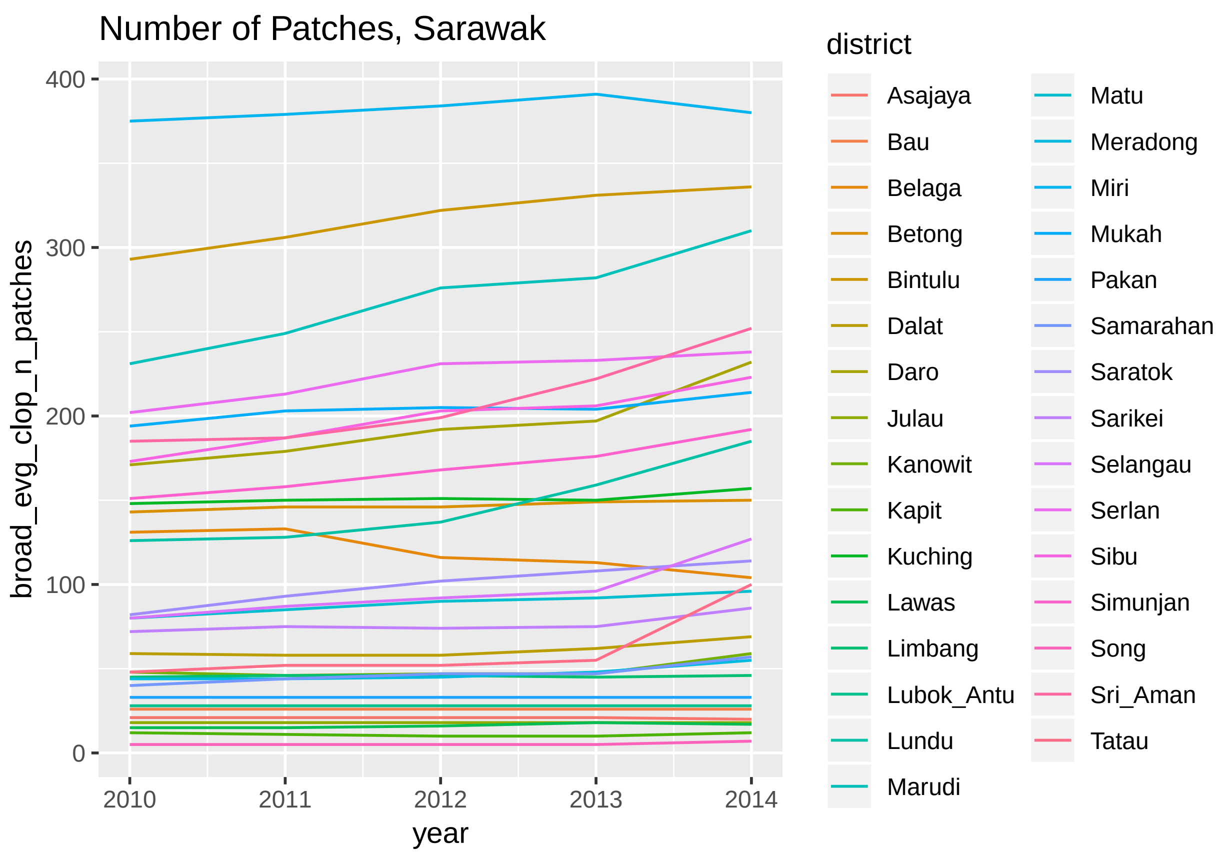
```
#Total area
ggplot(data = swk, aes(x = year, y = broad_evg_clop_total_area, color = district)) +
  geom_line() +
  labs(title = 'Total Forest Cover, Sarawak')
```



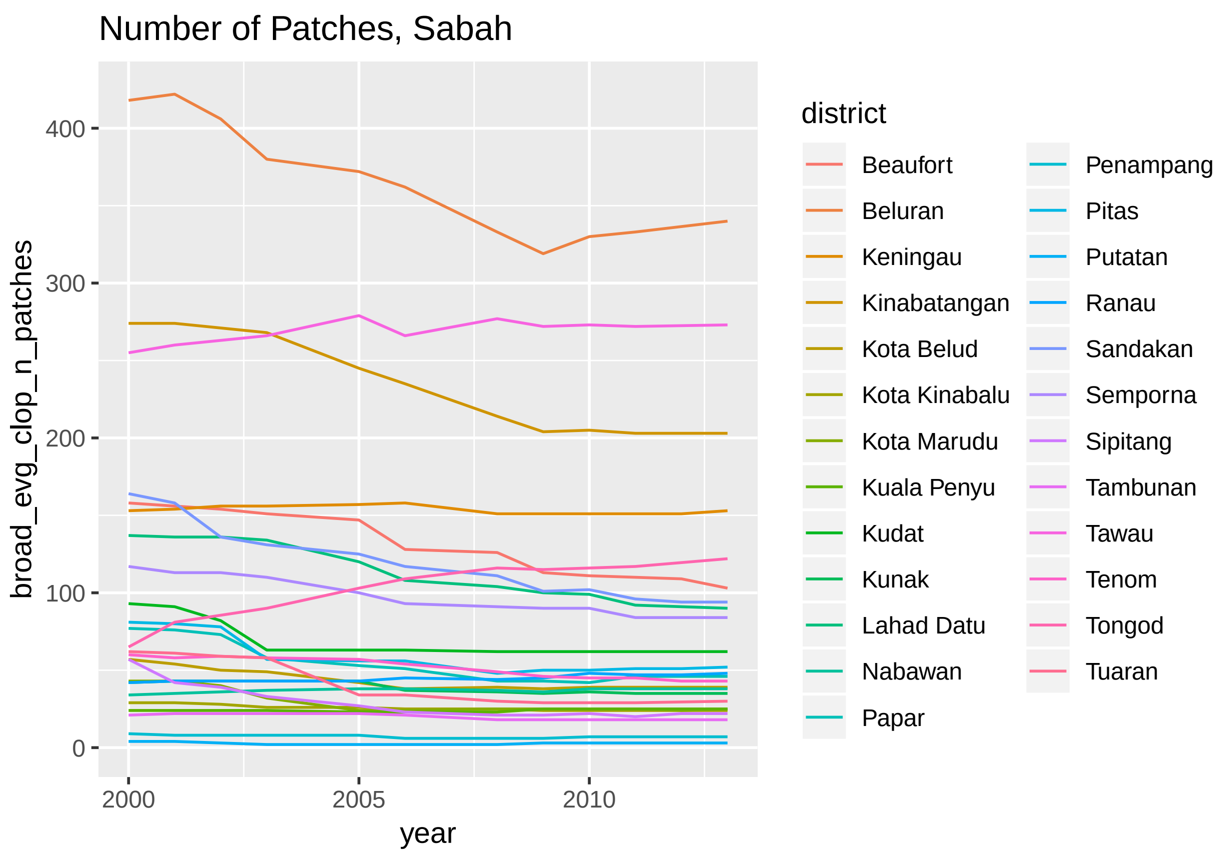

```
ggplot(data = sbh, aes(x = year, y = broad_evg_clop_total_area, color = district)) +
  geom_line() +
  labs(title = 'Total Forest Cover, Sabah')
```



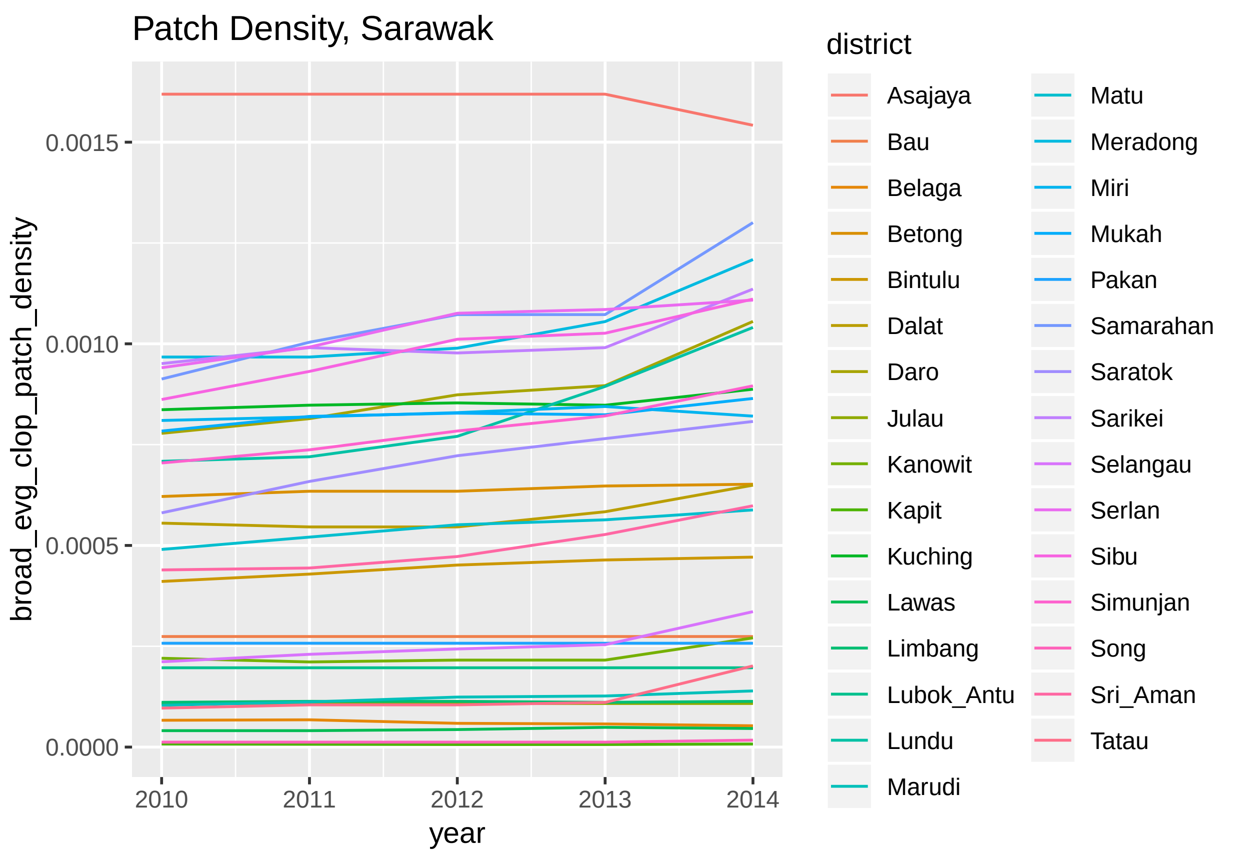

```
#Number of patches
ggplot(data = swk, aes(x = year, y = broad_evg_clop_n_patches, color = district)) +
  geom_line() +
  labs(title = 'Number of Patches, Sarawak')
```



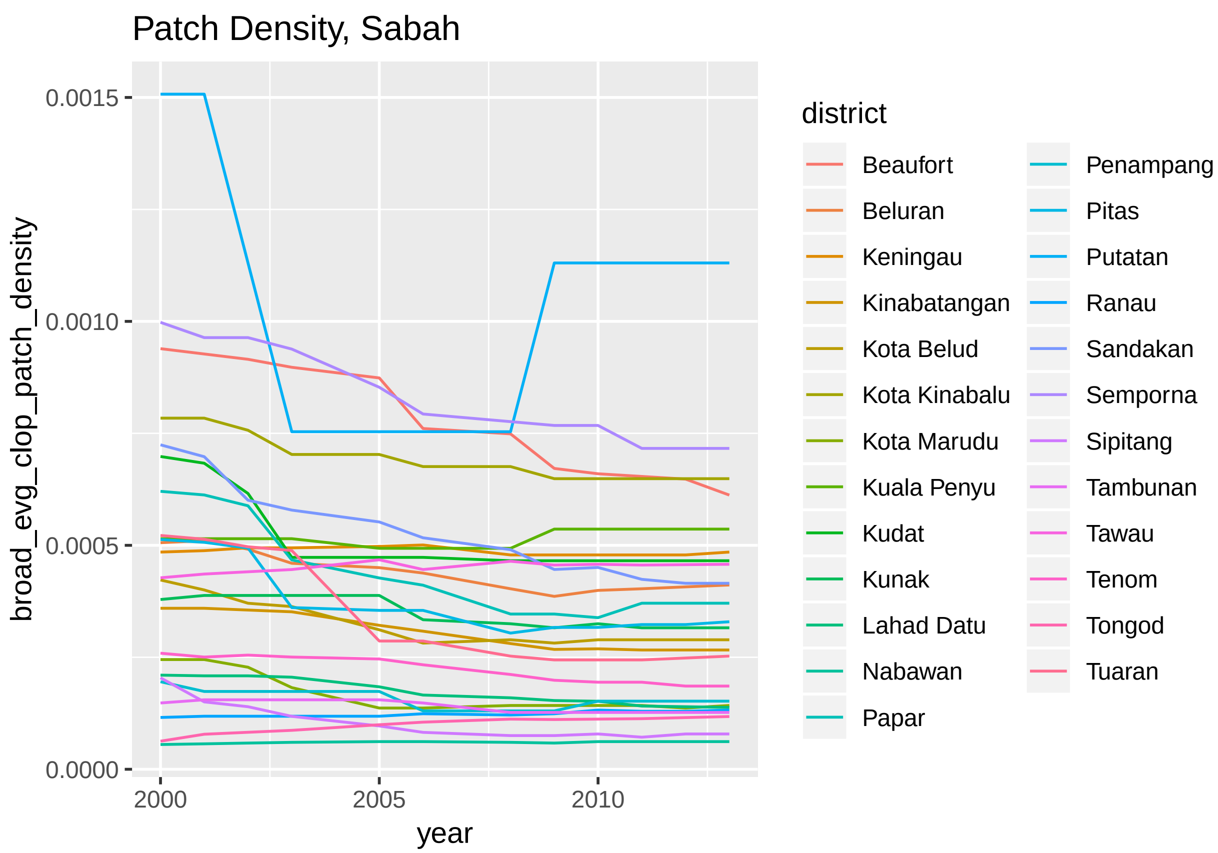

```r
ggplot(data = sbh, aes(x = year, y = broad_evg_clop_n_patches, color = district)) +
  geom_line() +
  labs(title = 'Number of Patches, Sabah')
```



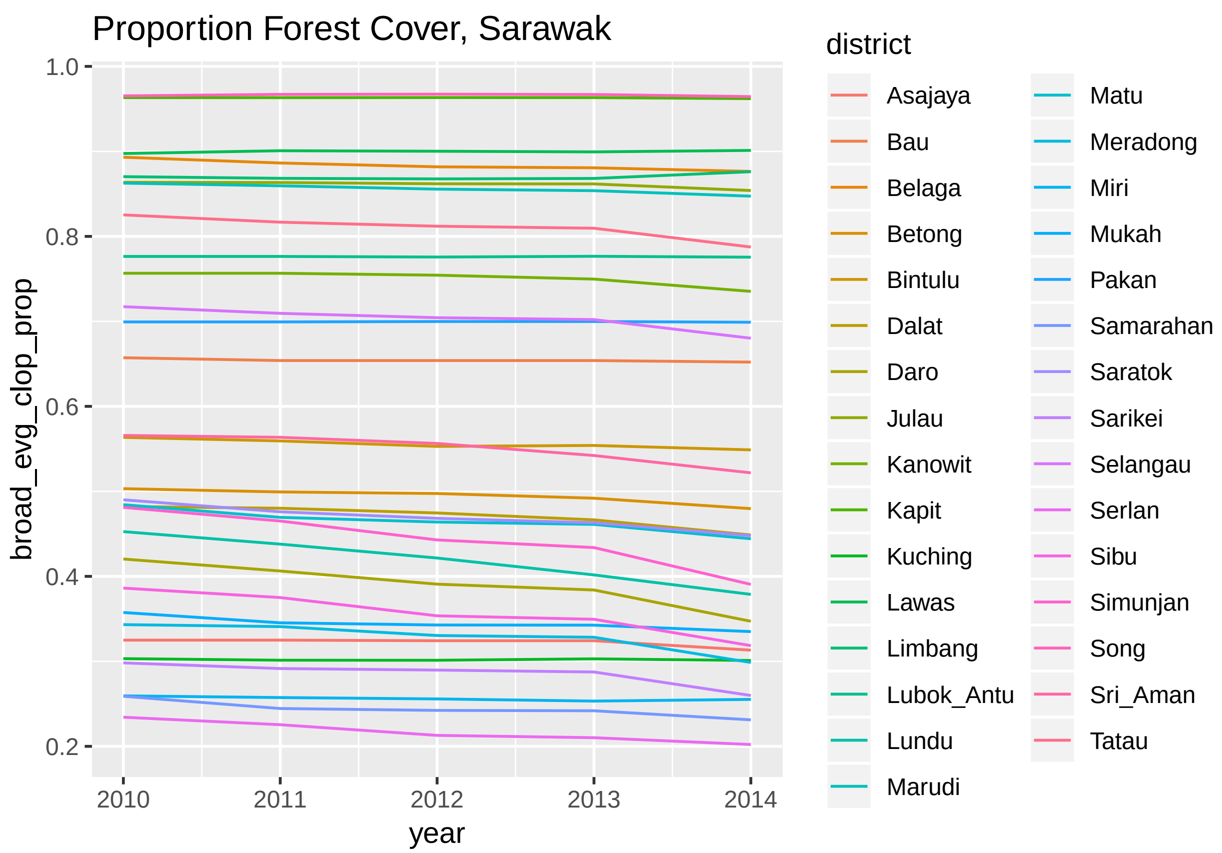

```r
#Patch density
ggplot(data = swk, aes(x = year, y = broad_evg_clop_patch_density, color = district)) +
  geom_line() +
  labs(title = 'Patch Density, Sarawak')
```



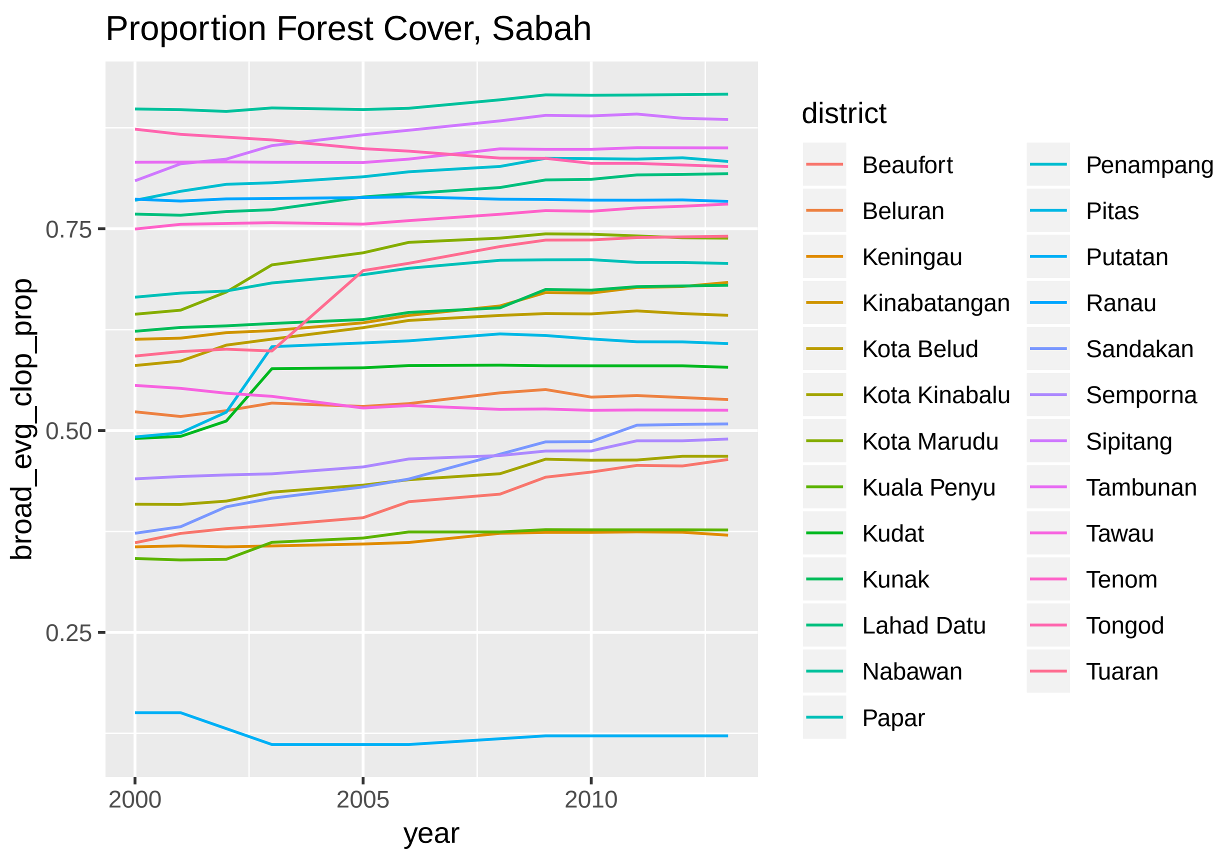

```r
ggplot(data = sbh, aes(x = year, y = broad_evg_clop_patch_density, color = district)) +
  geom_line() +
  labs(title = 'Patch Density, Sabah')
```




```r
#Patch density
ggplot(data = swk, aes(x = year, y = broad_evg_clop_prop, color = district)) +
  geom_line() +
  labs(title = 'Proportion Forest Cover, Sarawak')
```

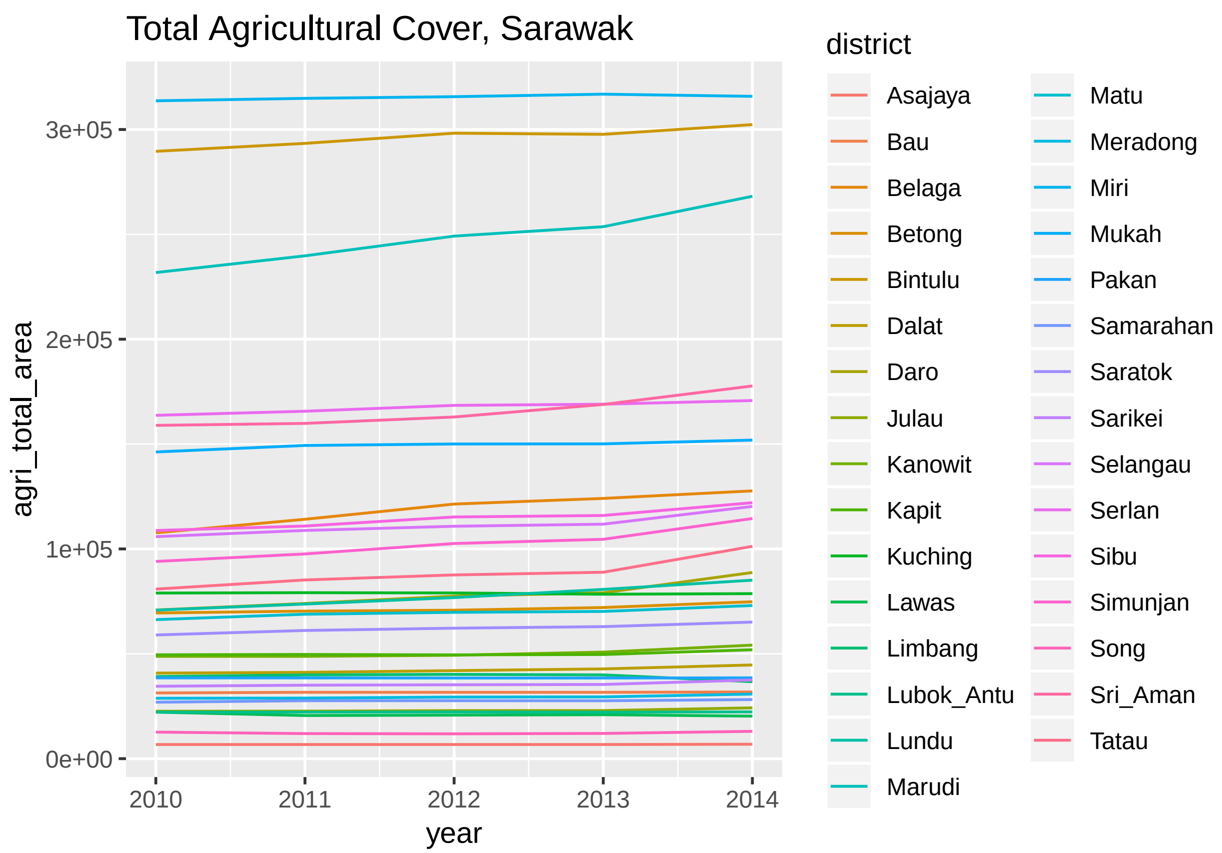



```r
ggplot(data = sbh, aes(x = year, y = broad_evg_clop_prop, color = district)) +
  geom_line() +
  labs(title = 'Proportion Forest Cover, Sabah')
```

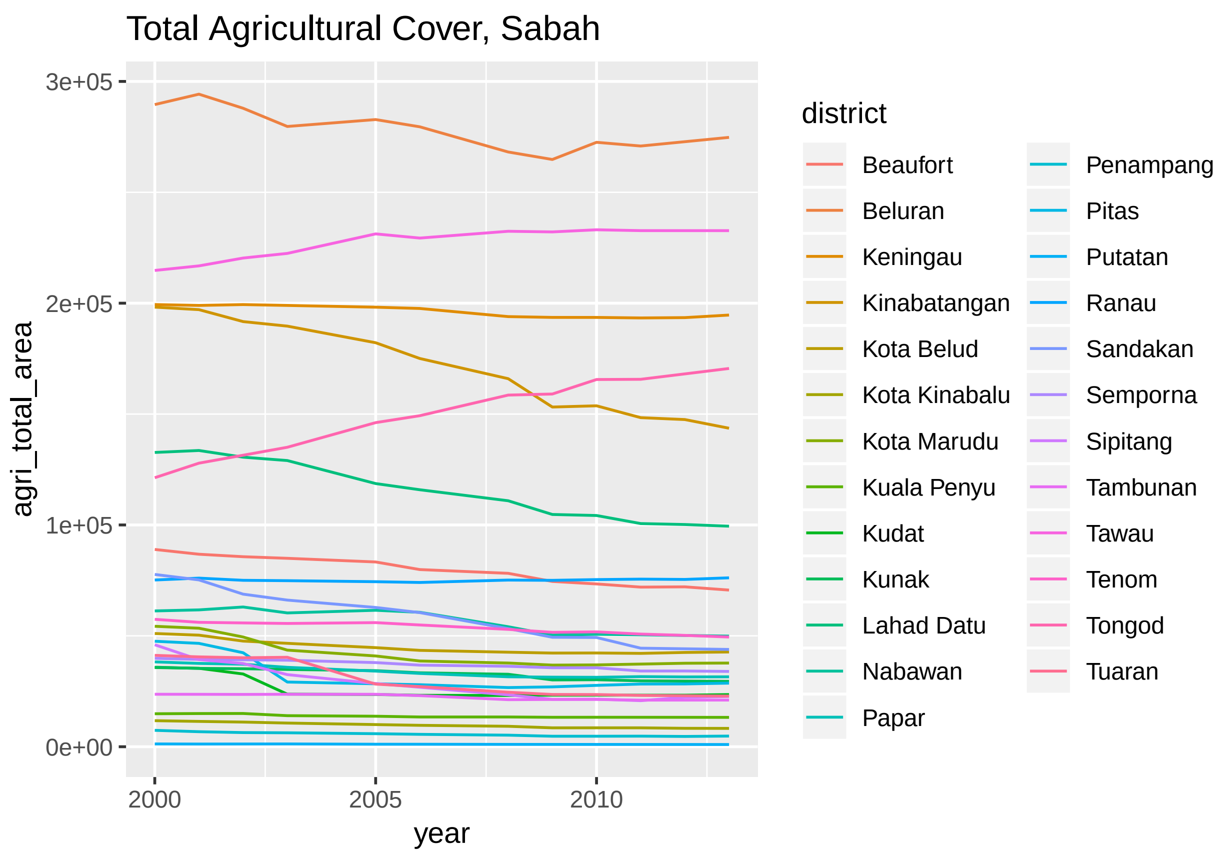


Agricultural cover across districts, Sarawak and Sabah

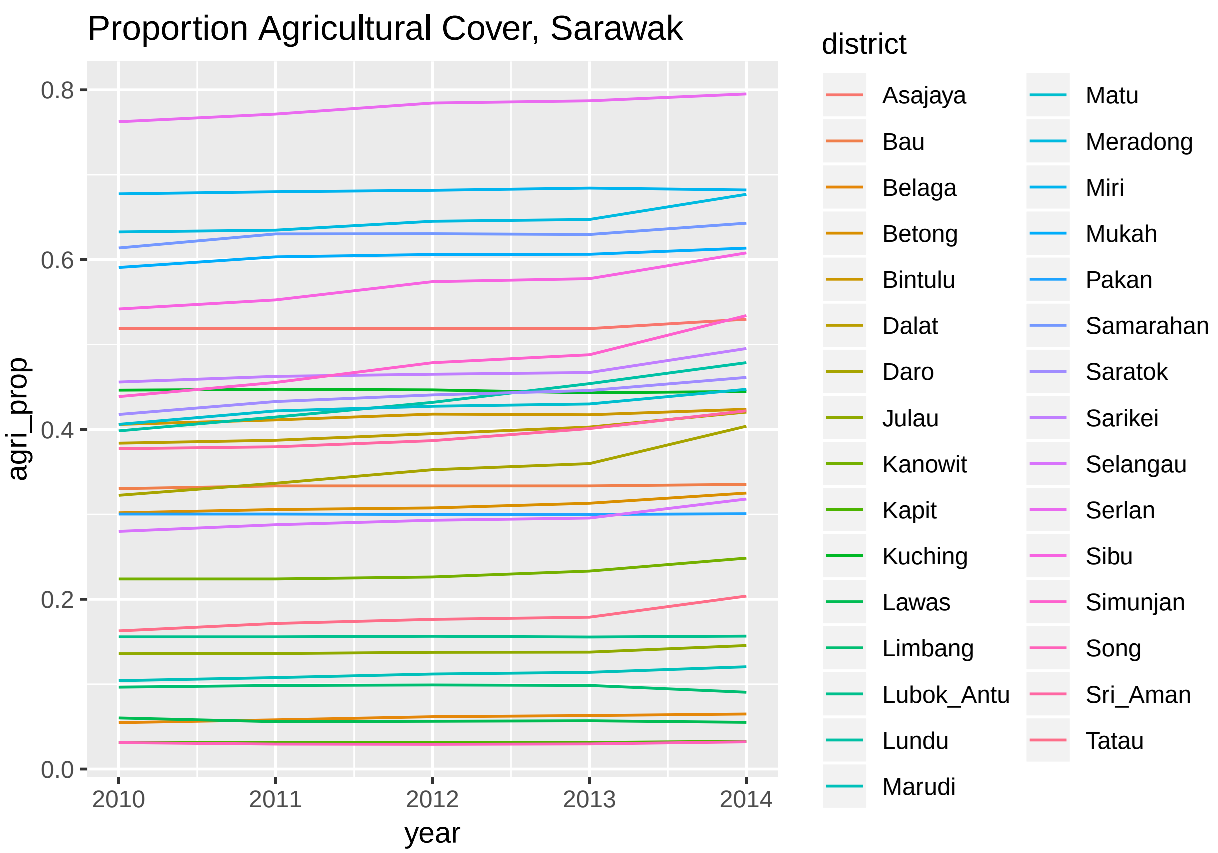
```r
#Total area
ggplot(data = swk, aes(x = year, y = agri_total_area, color = district)) +
  geom_line() +
  labs(title = 'Total Agricultural Cover, Sarawak')
```



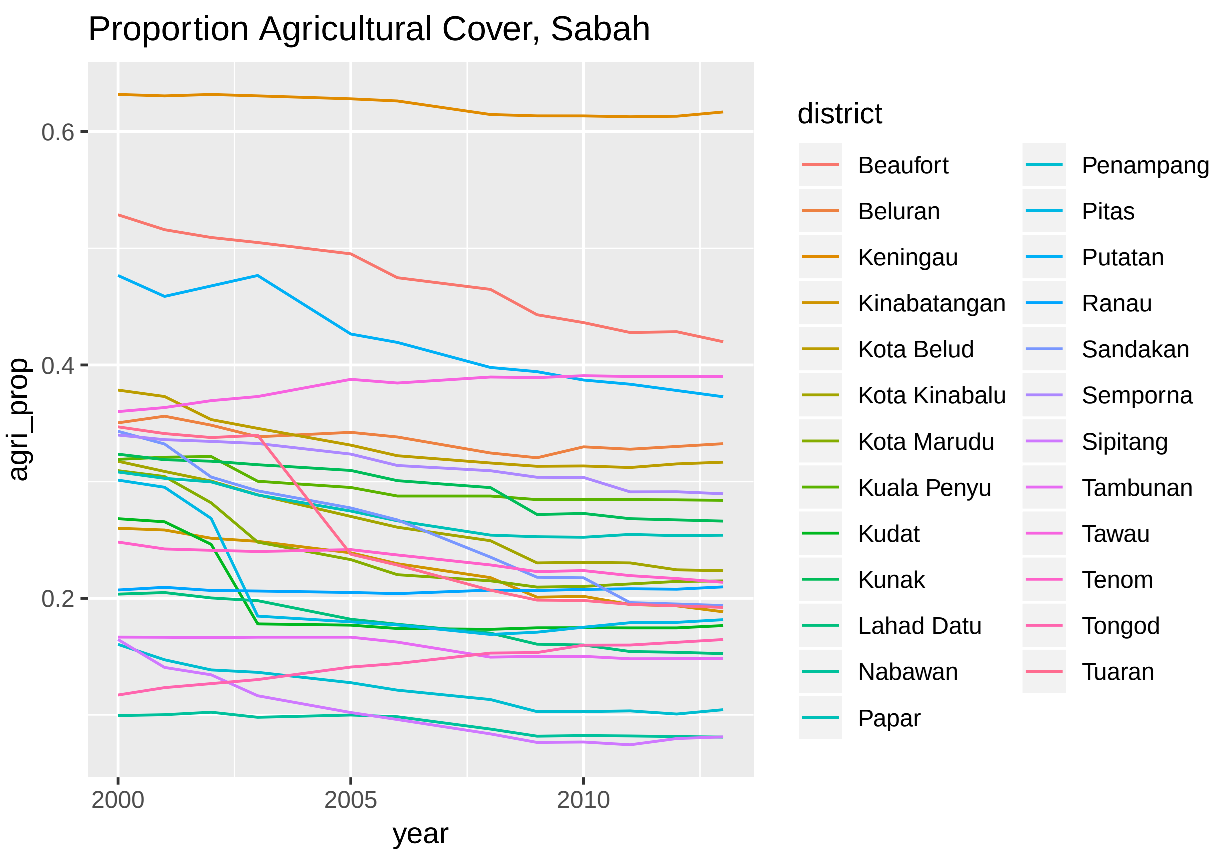

```
ggplot(data = sbh, aes(x = year, y = agri_total_area, color = district)) +
  geom_line() +
  labs(title = 'Total Agricultural Cover, Sabah')
```




```
#Proportion
ggplot(data = swk, aes(x = year, y = agri_prop, color = district)) +
  geom_line() +
  labs(title = 'Proportion Agricultural Cover, Sarawak')
```

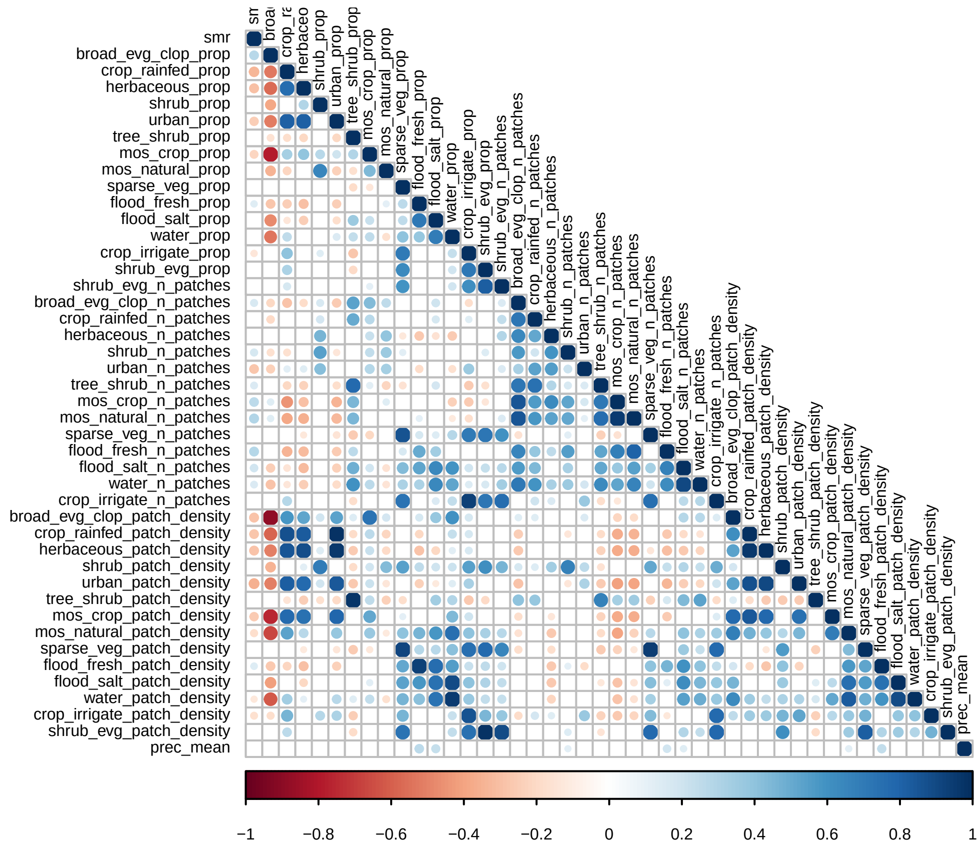



```
ggplot(data = sbh, aes(x = year, y = agri_prop, color = district)) +
  geom_line() +
  labs(title = 'Proportion Agricultural Cover, Sabah')
```

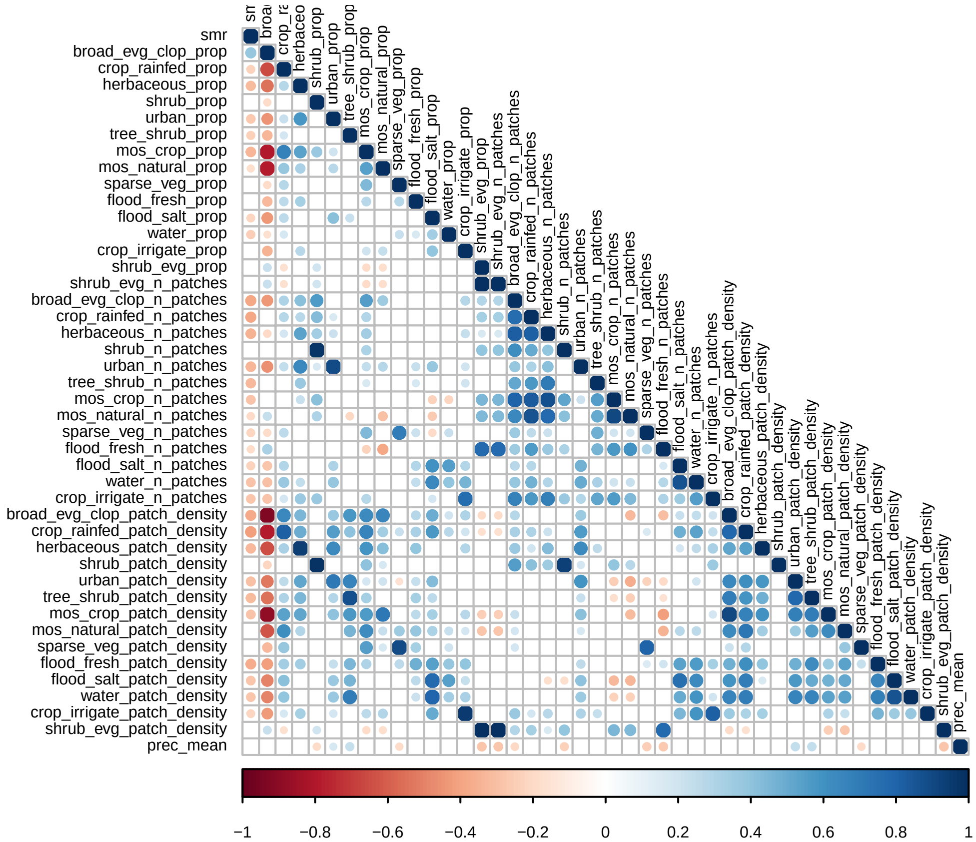


Correlations

```
res_sub <- rcorr(as.matrix(sbh_sub))

corrplot(res_sub$r, type = "lower",
         p.mat = res_sub$P, sig.level = 0.05, insig = "blank", tl.col = "black", tl.cex = 0.5, cl.cex = 0
```

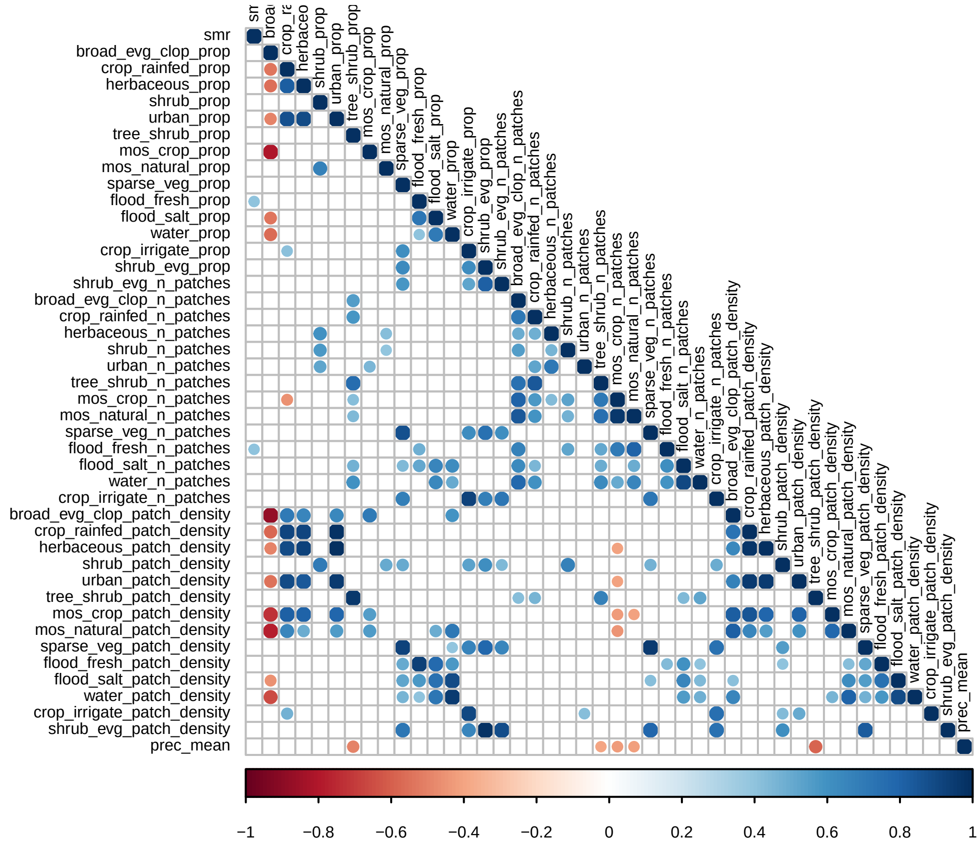



```r
res_sub2 <- rcorr(as.matrix(swk_sub))

corrplot(res_sub2$r, type = "lower",
        p.mat = res_sub2$P, sig.level = 0.05, insig = "blank", tl.col = "black", tl.cex = 0.5, cl.cex=0
```

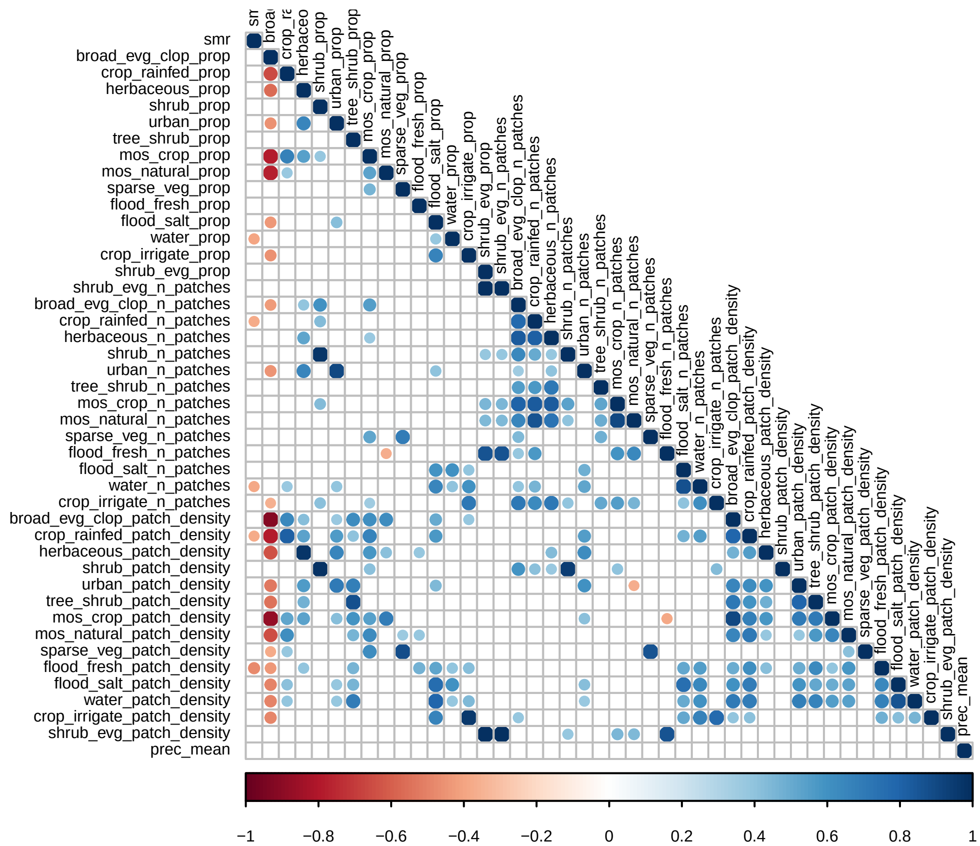


Take into account year

```
sbh_sub_time = sbh %>%
  filter(year == 2000) %>%
  select(c("smr", "broad_evg_clop_prop", "crop_rainfed_prop", "herbaceous_prop", "shrub_prop", "urban_p

res_sub_time = rcorr(as.matrix(sbh_sub_time))

corrplot(res_sub_time$r, type = "lower",
         p.mat = res_sub_time$P, sig.level = 0.05, insig = "blank", tl.col = "black", tl.cex = 0.5, cl.c
```




```
swk_sub_time = swk %>%
  filter(year == 2010) %>%
  select(c("smr", "broad_evg_clop_prop", "crop_rainfed_prop", "herbaceous_prop", "shrub_prop", "urban_p

res_sub_time2 = rcorr(as.matrix(swk_sub_time))

corrplot(res_sub_time2$r, type = "lower",
        p.mat = res_sub_time2$P, sig.level = 0.05, insig = "blank", tl.col = "black", tl.cex = 0.5, cl.
```



Models

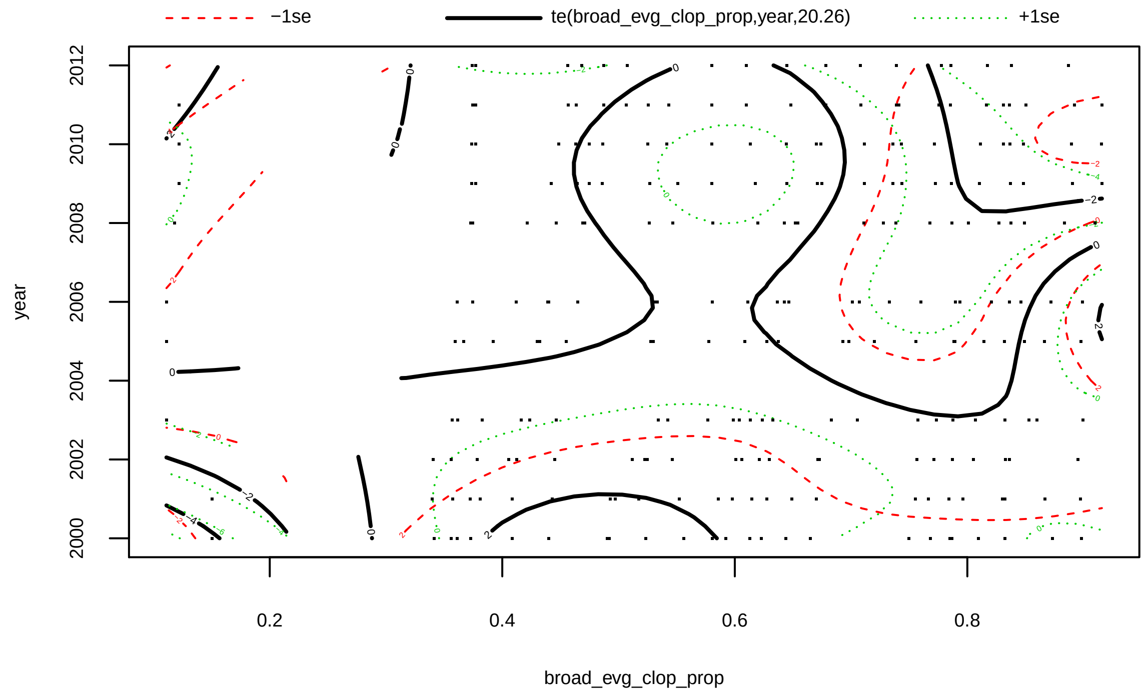
Split data into training and testing sets

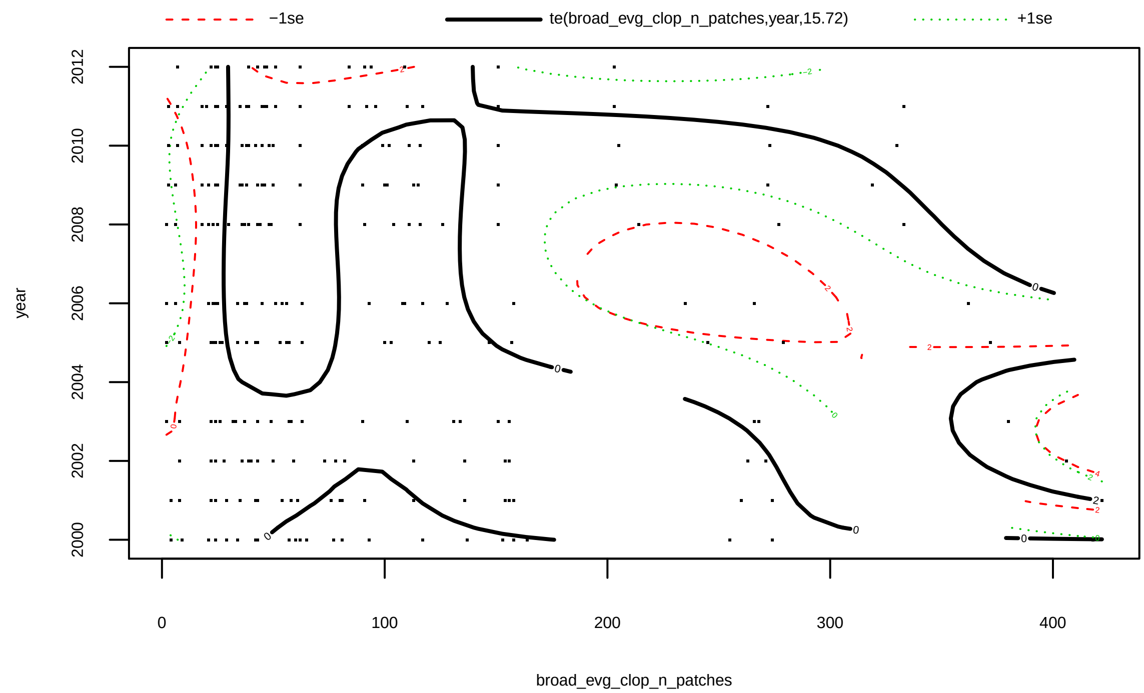
```r
train_sbh = sbh %>%
  filter(year %in% c(2000, 2001, 2002, 2003, 2005, 2006, 2008, 2009, 2010, 2011, 2012))

test_sbh = sbh %>%
  filter(year %in% c(2013))

train_swk = swk %>%
  filter(year %in% c(2010, 2011, 2012, 2013))

test_swk = swk %>%
  filter(year %in% c(2014))
```

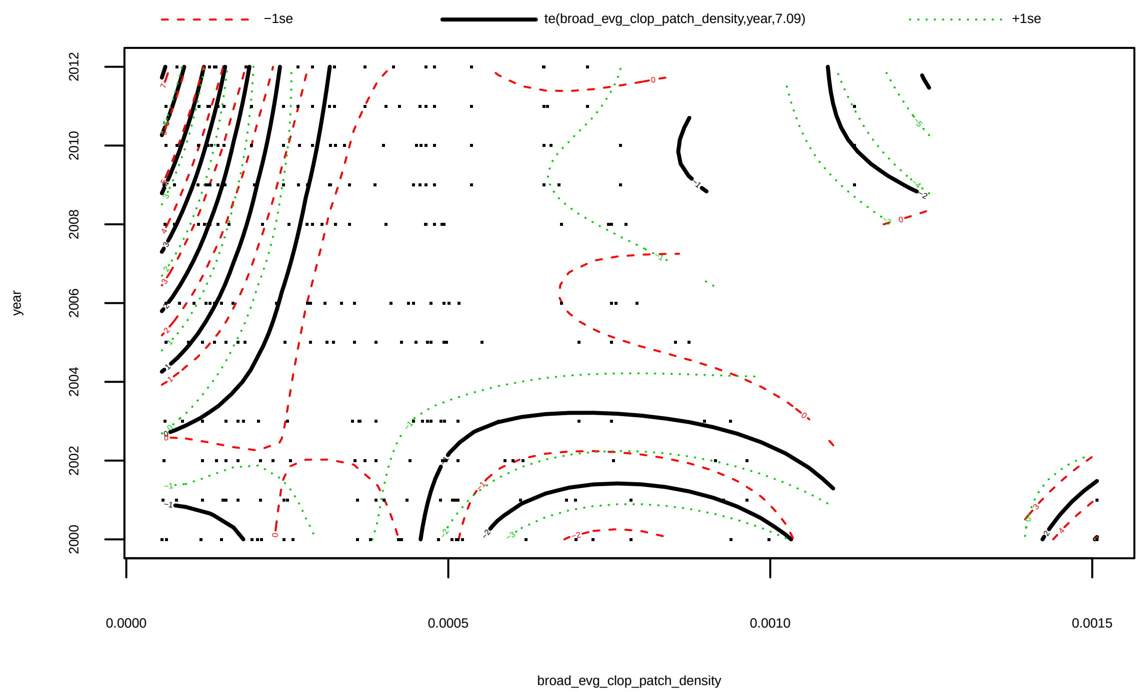
Sabah

```r
sbh_forest.fit = gam(smr ~ te(broad_evg_clop_prop, year) +
                       te(broad_evg_clop_n_patches, year) +
                       te(broad_evg_clop_patch_density, year),
                     data = train_sbh, method = 'GCV.Cp'
                     #, family = poisson(link = log)
                     )
```

```r
plot(sbh_forest.fit)
```



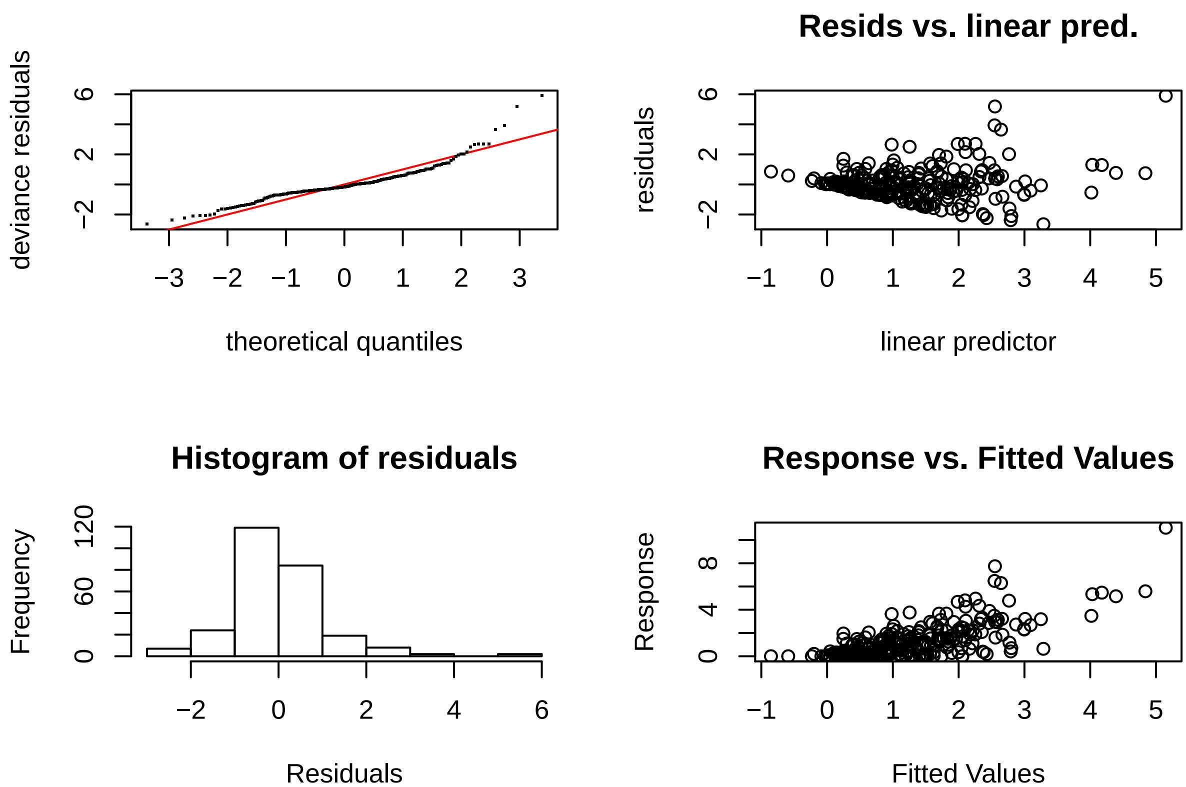

−1se
te(broad_evg_clop_n_patches,year,15.72)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0
100
200
300
400
broad_evg_clop_n_patches




```r
summary(sbh_forest.fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## smr ~ te(broad_evg_clop_prop, year) + te(broad_evg_clop_n_patches,
##     year) + te(broad_evg_clop_patch_density, year)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.28200    0.07164    17.9   <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
## te(broad_evg_clop_prop,year)          20.262   22.2 2.375 0.000601 ***
## te(broad_evg_clop_n_patches,year)     15.725   20.0 2.085 0.000171 ***
## te(broad_evg_clop_patch_density,year)  7.093   20.0 1.995 1.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.395   Deviance explained = 49.4%
## GCV = 1.6313  Scale est. = 1.3599    n = 265
```

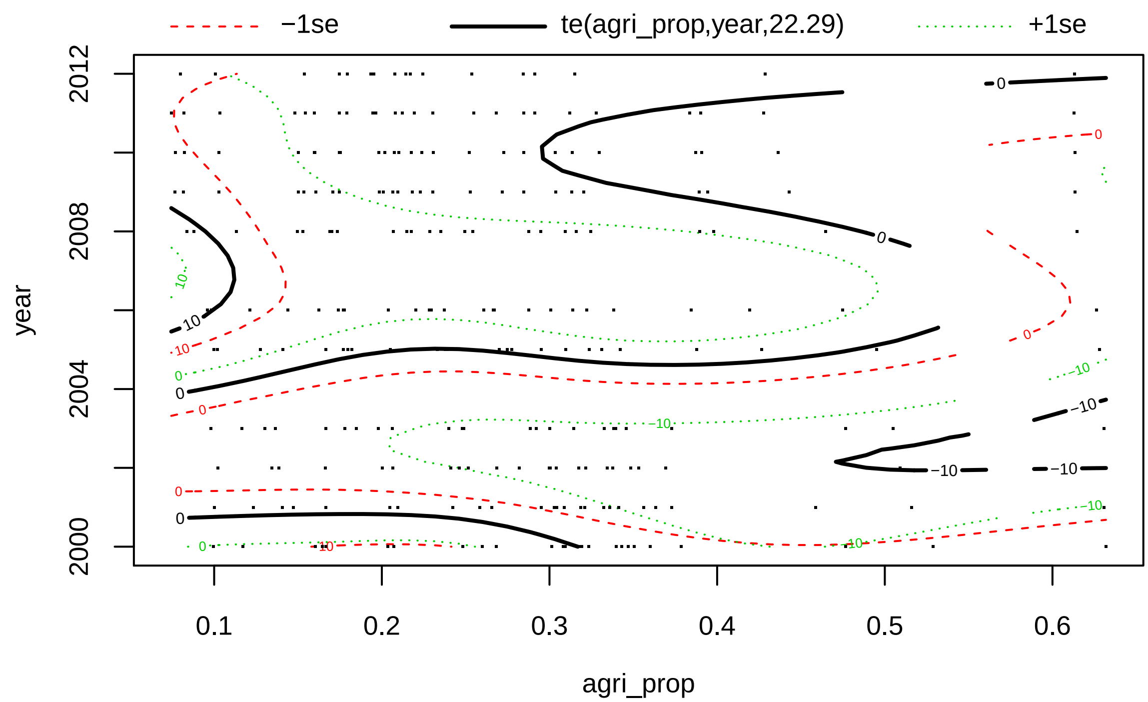
```r
gam.check(sbh_forest.fit)
```

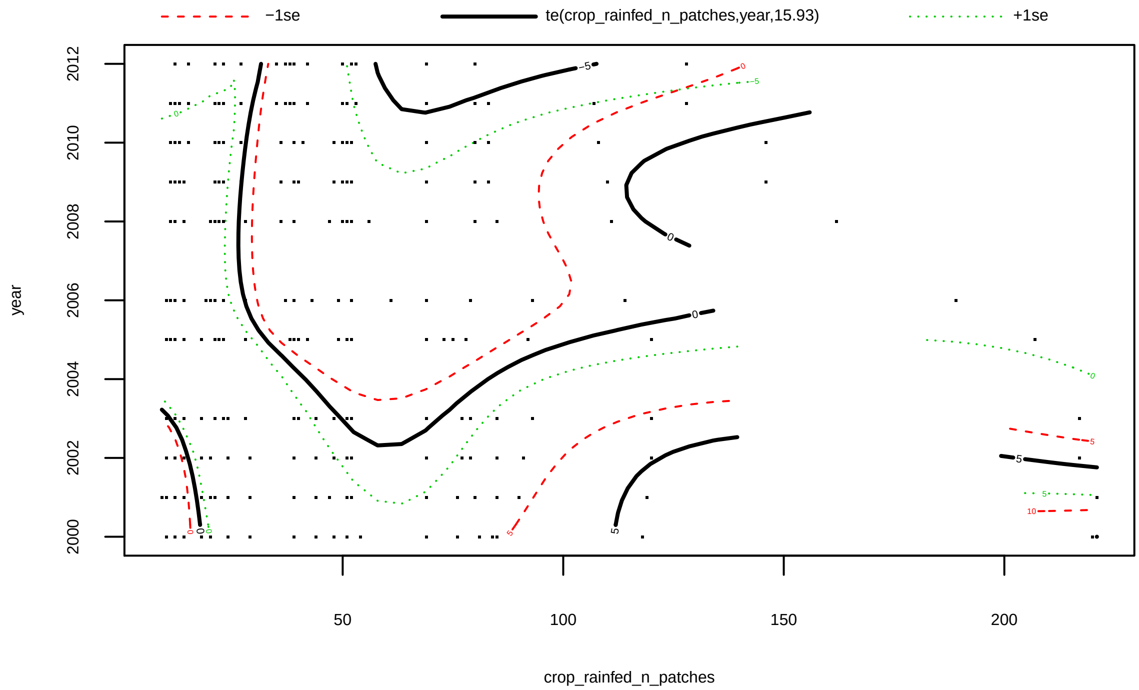



```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 8 iterations.
## The RMS GCV score gradient at convergence was 1.615406e-07 .
## The Hessian was positive definite.
## Model rank =  65 / 65
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                                          k'   edf k-index p-value
## te(broad_evg_clop_prop,year)          24.00 20.26    0.68  <2e-16 ***
## te(broad_evg_clop_n_patches,year)     20.00 15.72    1.05   0.815
## te(broad_evg_clop_patch_density,year) 20.00  7.09    0.84   0.005 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

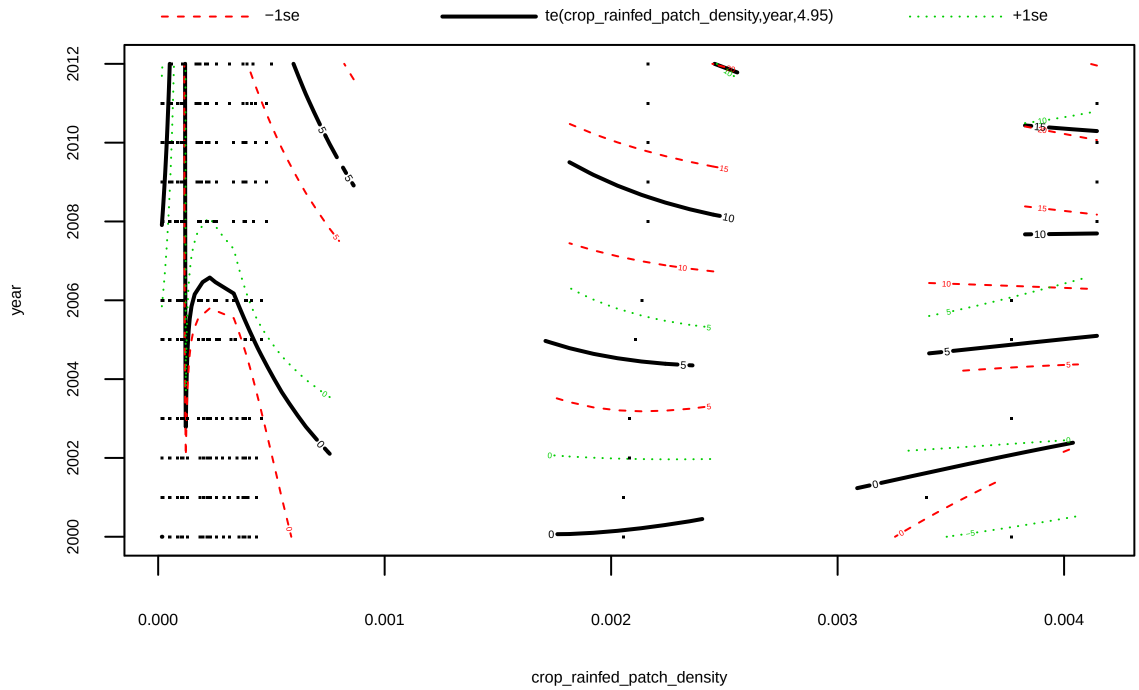
```r
sbh_agri.fit = gam(smr ~ te(agri_prop, year) +
                       te(crop_rainfed_n_patches, year) +
                       te(crop_rainfed_patch_density, year) +
                       te(crop_irrigate_n_patches, year) +
                       te(crop_irrigate_patch_density, year) +
                       te(mos_crop_n_patches, year) +
```

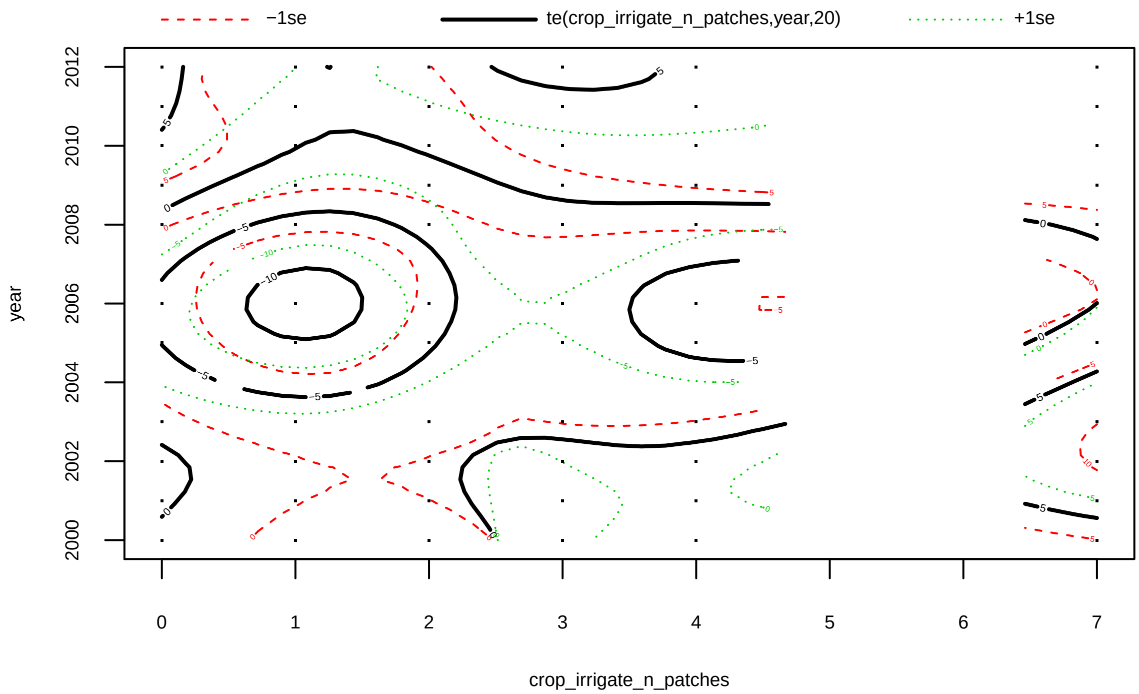
```
                    te(mos_crop_patch_density, year) +
                    te(mos_natural_n_patches, year) +
                    te(mos_natural_patch_density, year),
                  data = train_sbh, method = 'GCV.Cp'
                #, family = poisson(link = log)
                )

plot(sbh_agri.fit)
```



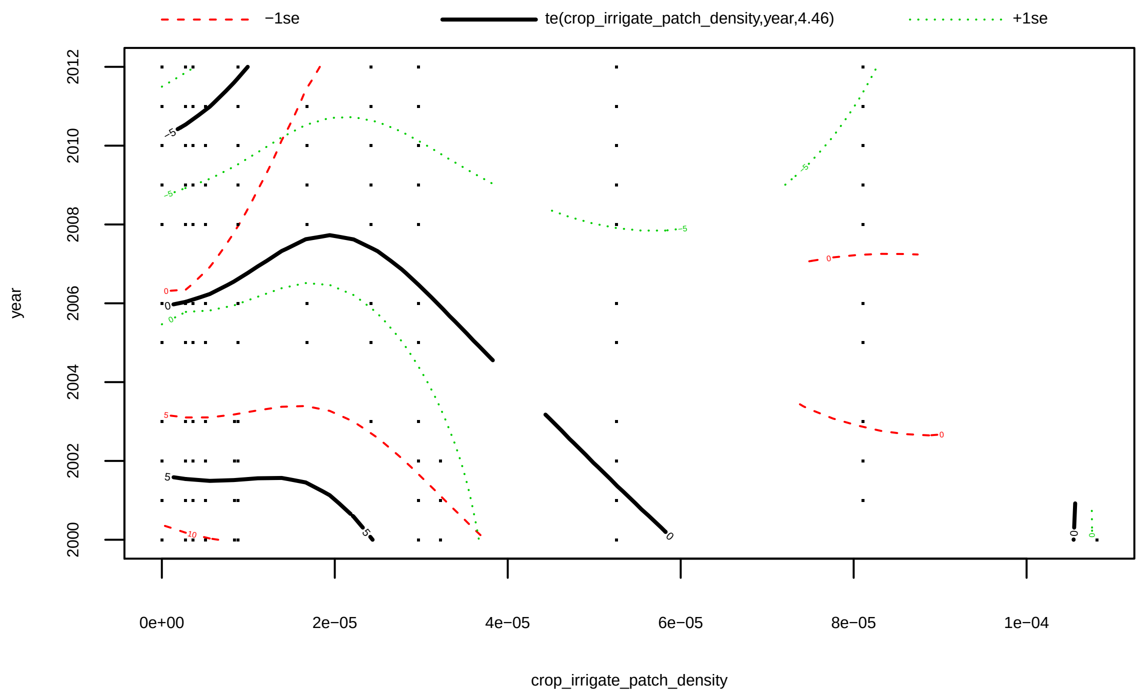


−1se
te(crop_rainfed_n_patches,year,15.93)
+1se
year
2000
2002
2004
2006
2008
2010
2012
50
100
150
200
crop_rainfed_n_patches

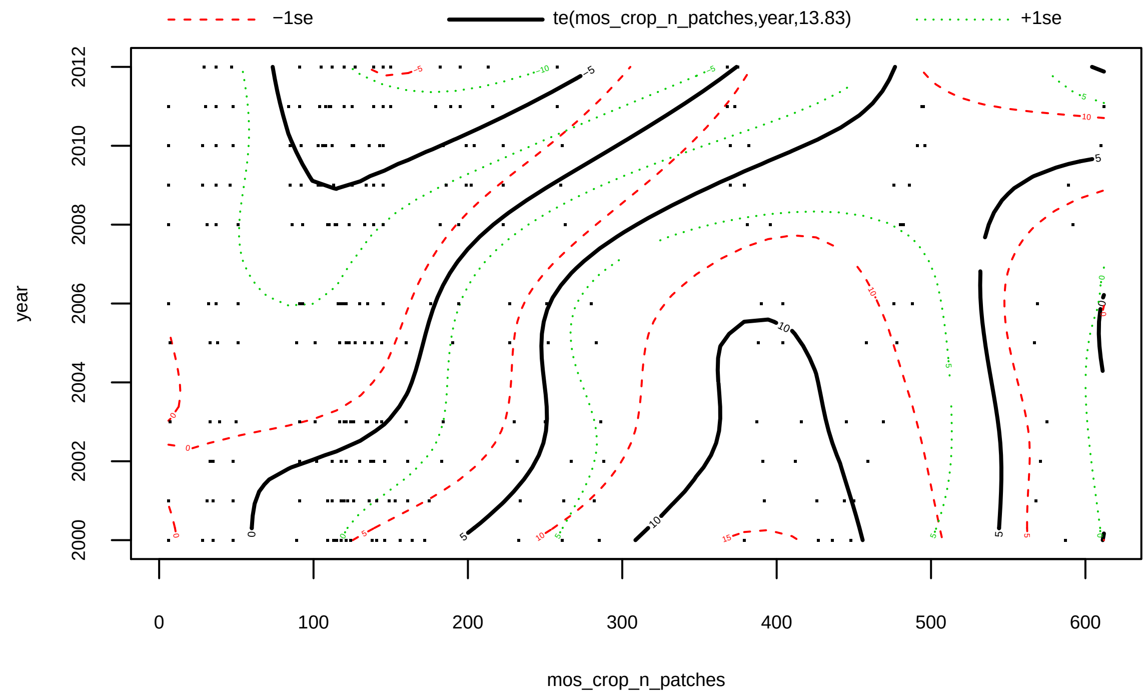

−1se
te(crop_rainfed_patch_density,year,4.95)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0.000
0.001
0.002
0.003
0.004
crop_rainfed_patch_density

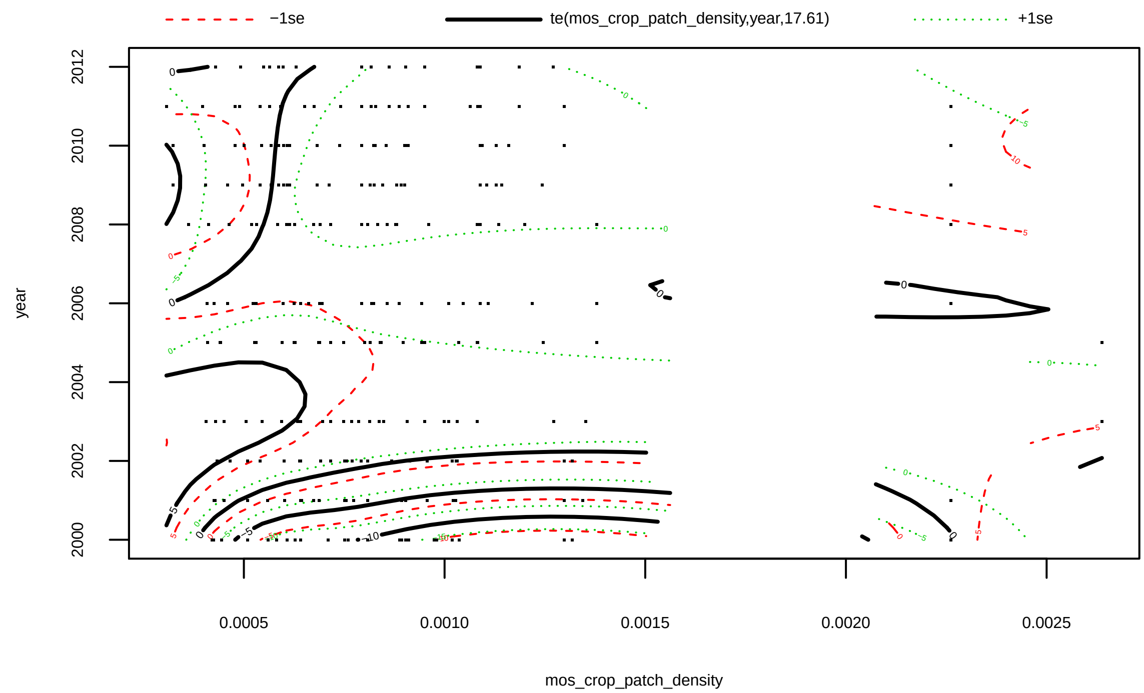


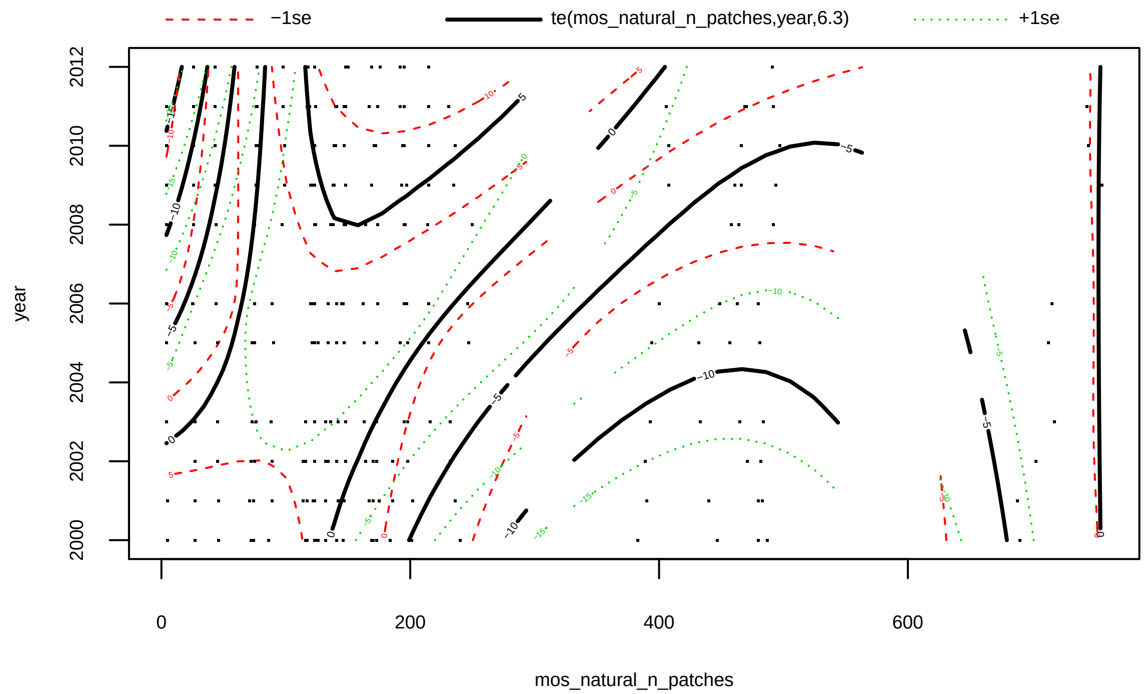
−1se
te(crop_irrigate_n_patches,year,20)
+1se
year
crop_irrigate_n_patches
0
1
2
3
4
5
6
7
2000
2002
2004
2006
2008
2010
2012



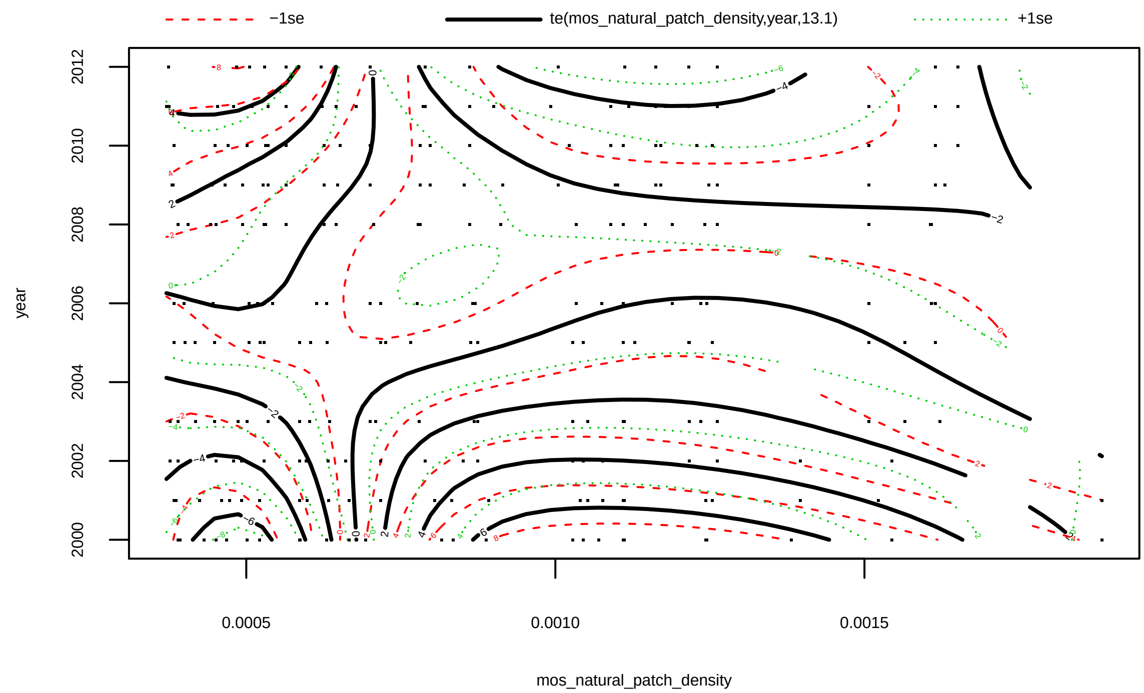
−1se
te(crop_irrigate_patch_density,year,4.46)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0e+00
2e−05
4e−05
6e−05
8e−05
1e−04
crop_irrigate_patch_density


−1se
te(mos_crop_n_patches,year,13.83)
+1se
year
mos_crop_n_patches
2000
2002
2004
2006
2008
2010
2012
0
100
200
300
400
500
600


−1se
te(mos_crop_patch_density,year,17.61)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0.0005
0.0010
0.0015
0.0020
0.0025
mos_crop_patch_density



−1se
te(mos_natural_n_patches,year,6.3)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0
200
400
600
mos_natural_n_patches

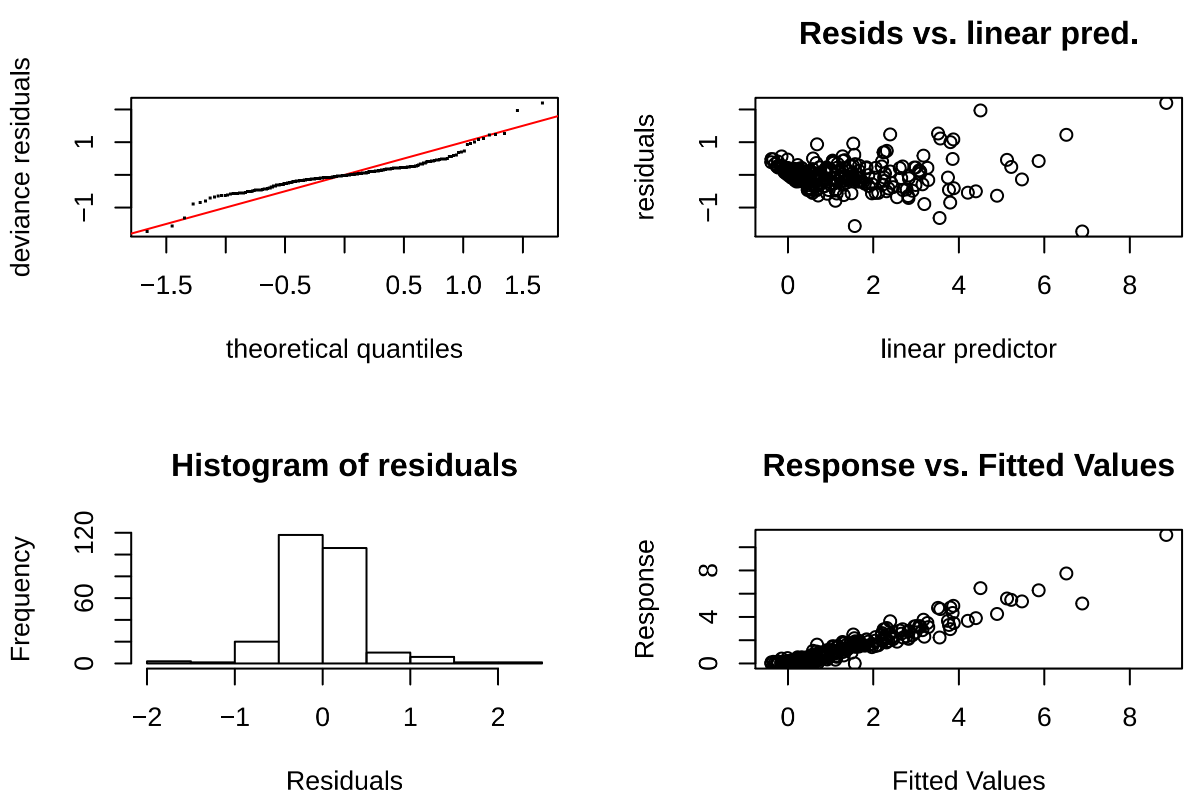



```r
summary(sbh_agri.fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## smr ~ te(agri_prop, year) + te(crop_rainfed_n_patches, year) +
##     te(crop_rainfed_patch_density, year) + te(crop_irrigate_n_patches,
##     year) + te(crop_irrigate_patch_density, year) + te(mos_crop_n_patches,
##     year) + te(mos_crop_patch_density, year) + te(mos_natural_n_patches,
##     year) + te(mos_natural_patch_density, year)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.28200    0.03525   36.36   <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(agri_prop,year)                   22.290  23.12 10.462  < 2e-16 ***
## te(crop_rainfed_n_patches,year)      15.926  20.00  6.197  < 2e-16 ***
## te(crop_rainfed_patch_density,year)   4.953  16.00  1.776 2.66e-08 ***
## te(crop_irrigate_n_patches,year)     20.000  20.00  5.339  < 2e-16 ***
```

```
## te(crop_irrigate_patch_density,year)  4.458   20.00   0.731 0.000354 ***
## te(mos_crop_n_patches,year)           13.832   20.00   7.983  < 2e-16 ***
## te(mos_crop_patch_density,year)       17.608   20.00   4.012 5.95e-12 ***
## te(mos_natural_n_patches,year)         6.301   20.00   4.266  < 2e-16 ***
## te(mos_natural_patch_density,year)    13.096   20.00   4.606  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.854   Deviance explained = 91.9%
## GCV = 0.59973  Scale est. = 0.32937    n = 265
```

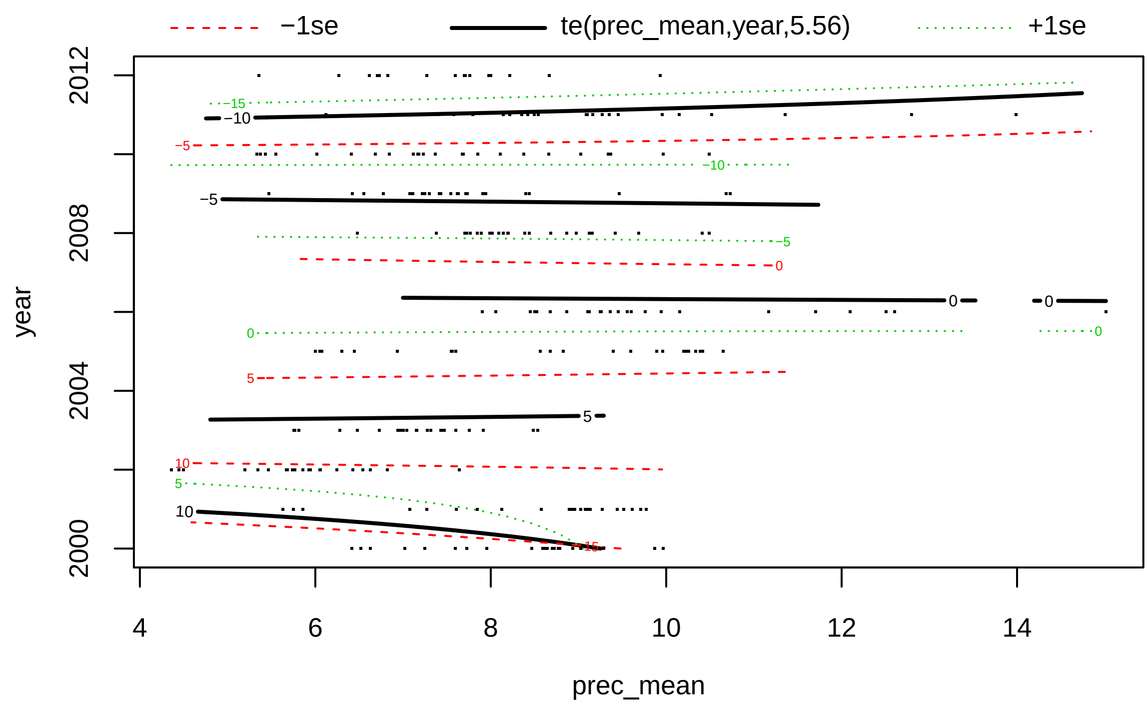
```
gam.check(sbh_agri.fit)
```

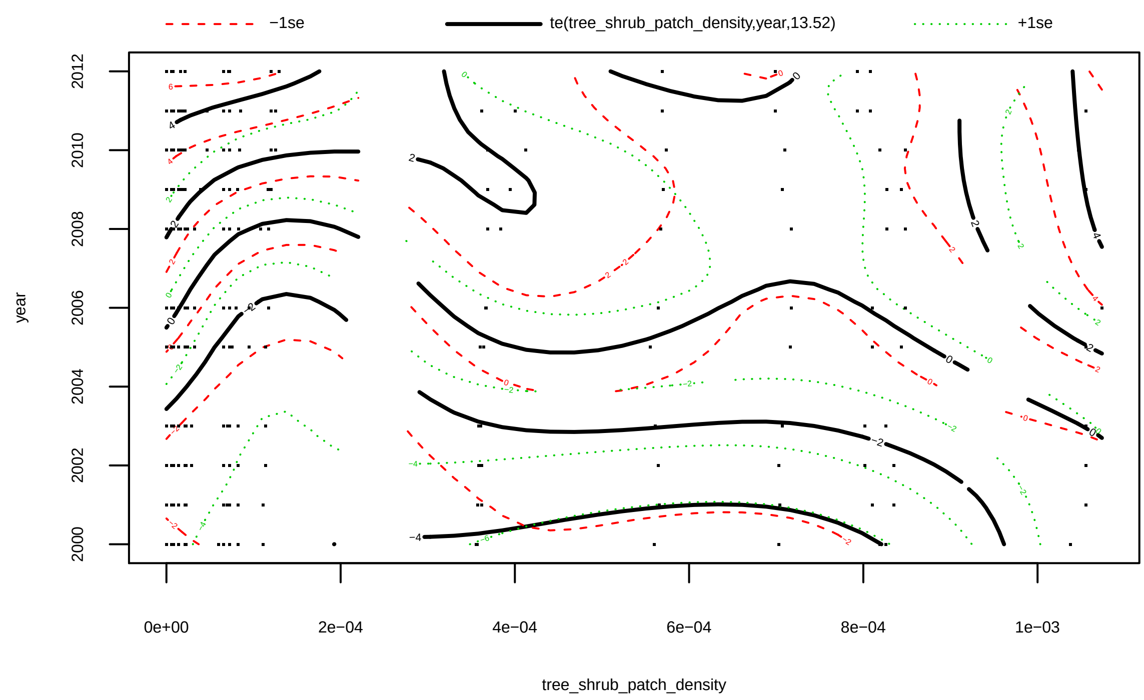


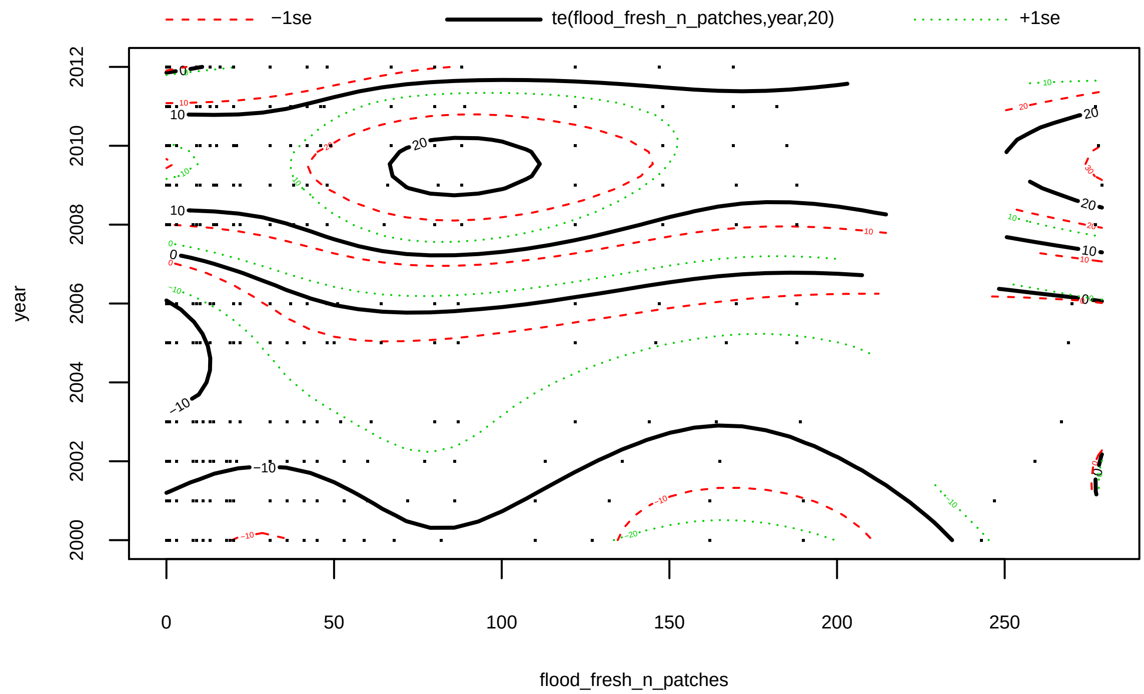

```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 44 iterations by steepest
## descent step failure.
## The RMS GCV score gradient at convergence was 5.846555e-08 .
## The Hessian was positive definite.
## Model rank =  185 / 185
##
## Basis dimension (k) checking 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
```

```
## te(agri_prop,year)                     24.00 22.29    1.21    1.00
## te(crop_rainfed_n_patches,year)        20.00 15.93    1.01    0.62
## te(crop_rainfed_patch_density,year)    20.00  4.95    1.25    1.00
## te(crop_irrigate_n_patches,year)       20.00 20.00    1.06    0.80
## te(crop_irrigate_patch_density,year)   20.00  4.46    1.06    0.83
## te(mos_crop_n_patches,year)            20.00 13.83    1.06    0.90
## te(mos_crop_patch_density,year)        20.00 17.61    1.18    0.99
## te(mos_natural_n_patches,year)         20.00  6.30    1.05    0.78
## te(mos_natural_patch_density,year)     20.00 13.10    1.13    0.97
```

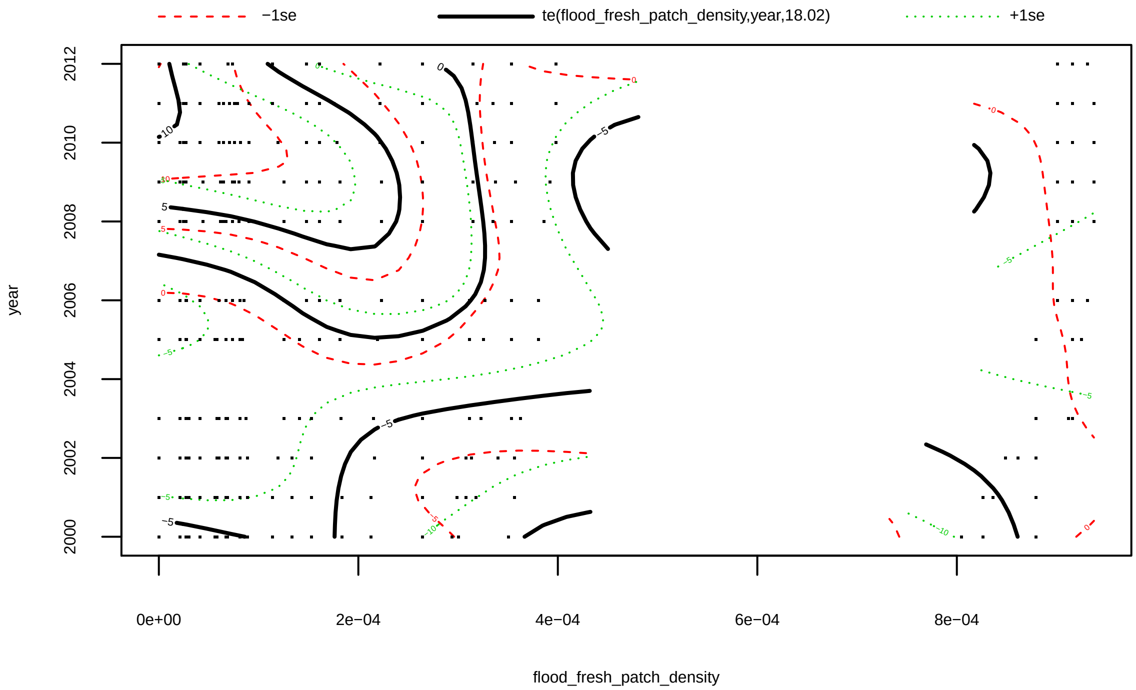
```r
sbh_water.fit = gam(smr ~ te(prec_mean, year) +
                      te(tree_shrub_patch_density, year) +
                      te(flood_fresh_n_patches, year) +
                      te(flood_fresh_patch_density, year) +
                      te(flood_fresh_prop, year) +
                      te(flood_salt_n_patches, year) +
                      te(flood_salt_patch_density, year) +
                      te(flood_salt_prop, year),
                    data = train_sbh, method = 'GCV.Cp')

plot(sbh_water.fit)
```



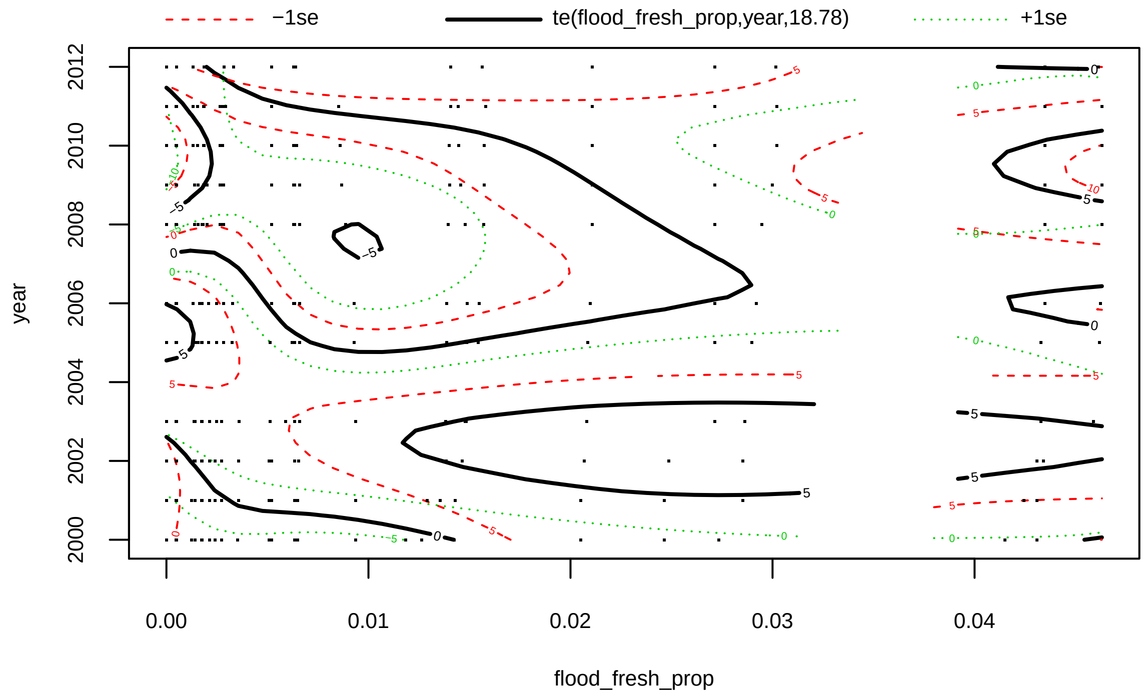

−1se
te(tree_shrub_patch_density,year,13.52)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0e+00
2e−04
4e−04
6e−04
8e−04
1e−03
tree_shrub_patch_density



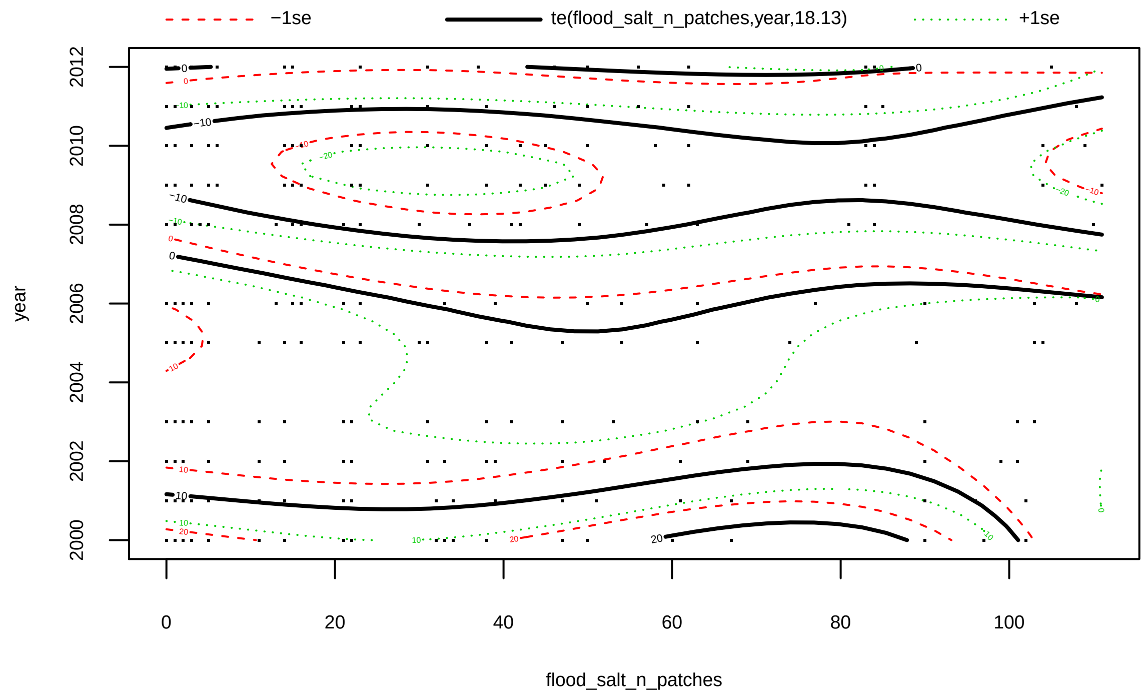
−1se
te(flood_fresh_n_patches,year,20)
+1se
year
flood_fresh_n_patches
2000
2002
2004
2006
2008
2010
2012
0
50
100
150
200
250
20
10
0
−10



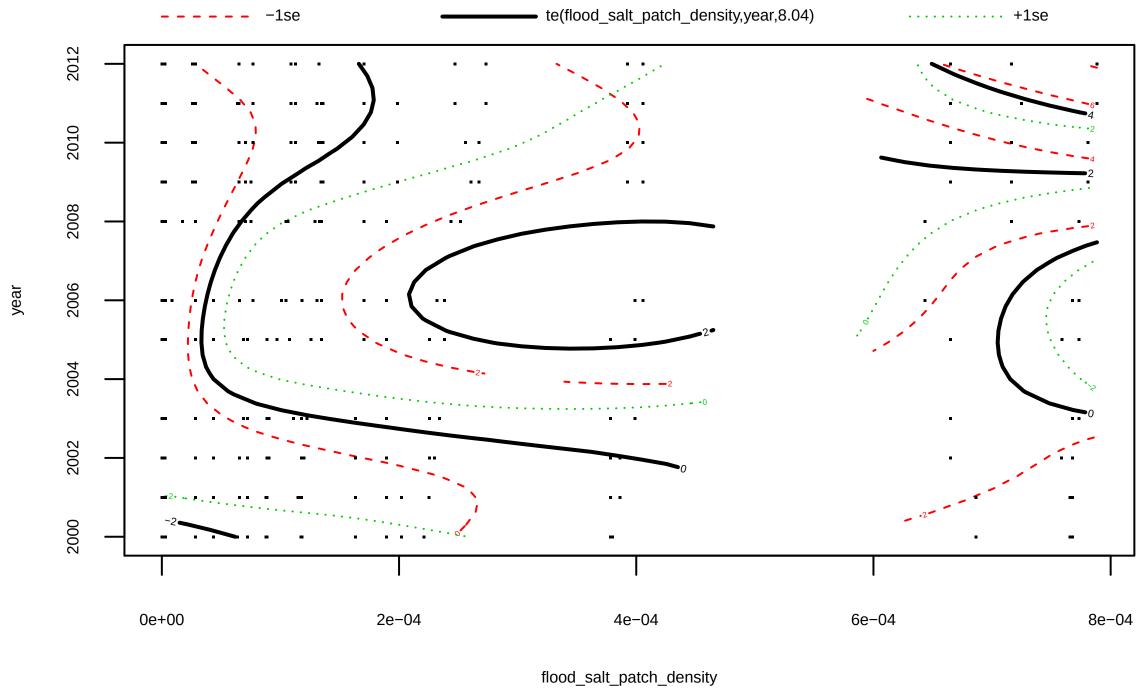
−1se
te(flood_fresh_patch_density,year,18.02)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0e+00
2e−04
4e−04
6e−04
8e−04
flood_fresh_patch_density

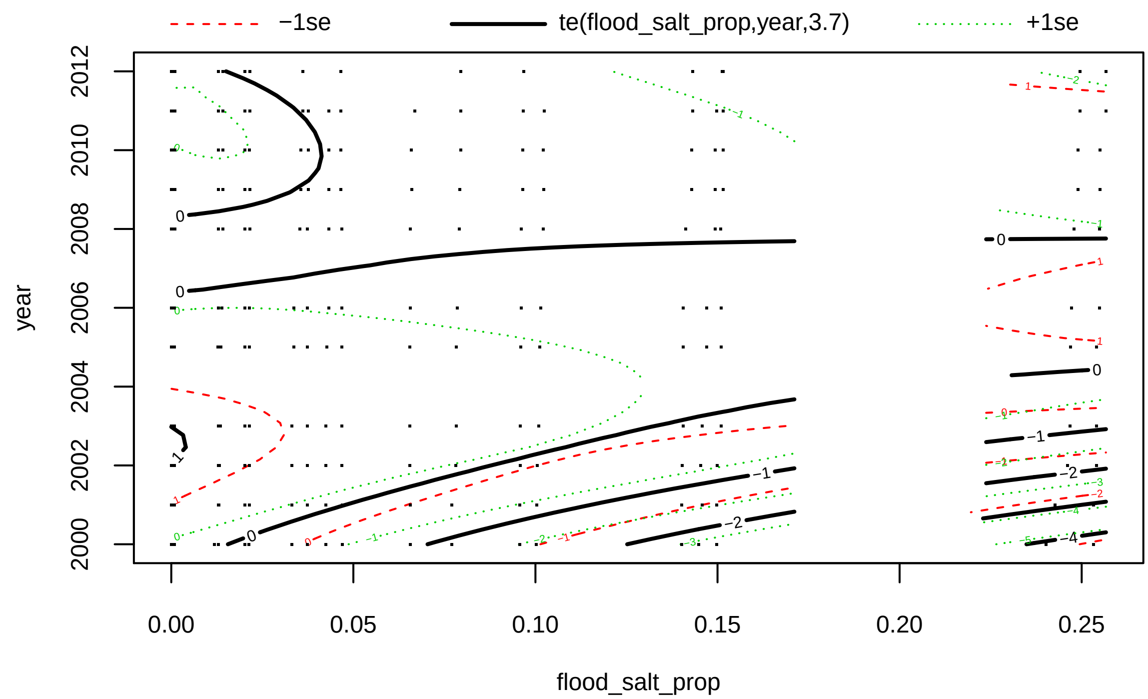


−1se
te(flood_fresh_prop,year,18.78)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0.00
0.01
0.02
0.03
0.04
flood_fresh_prop



−1se
te(flood_salt_n_patches,year,18.13)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0
20
40
60
80
100
flood_salt_n_patches


−1se
te(flood_salt_patch_density,year,8.04)
+1se
year
2000
2002
2004
2006
2008
2010
2012
0e+00
2e−04
4e−04
6e−04
8e−04
flood_salt_patch_density

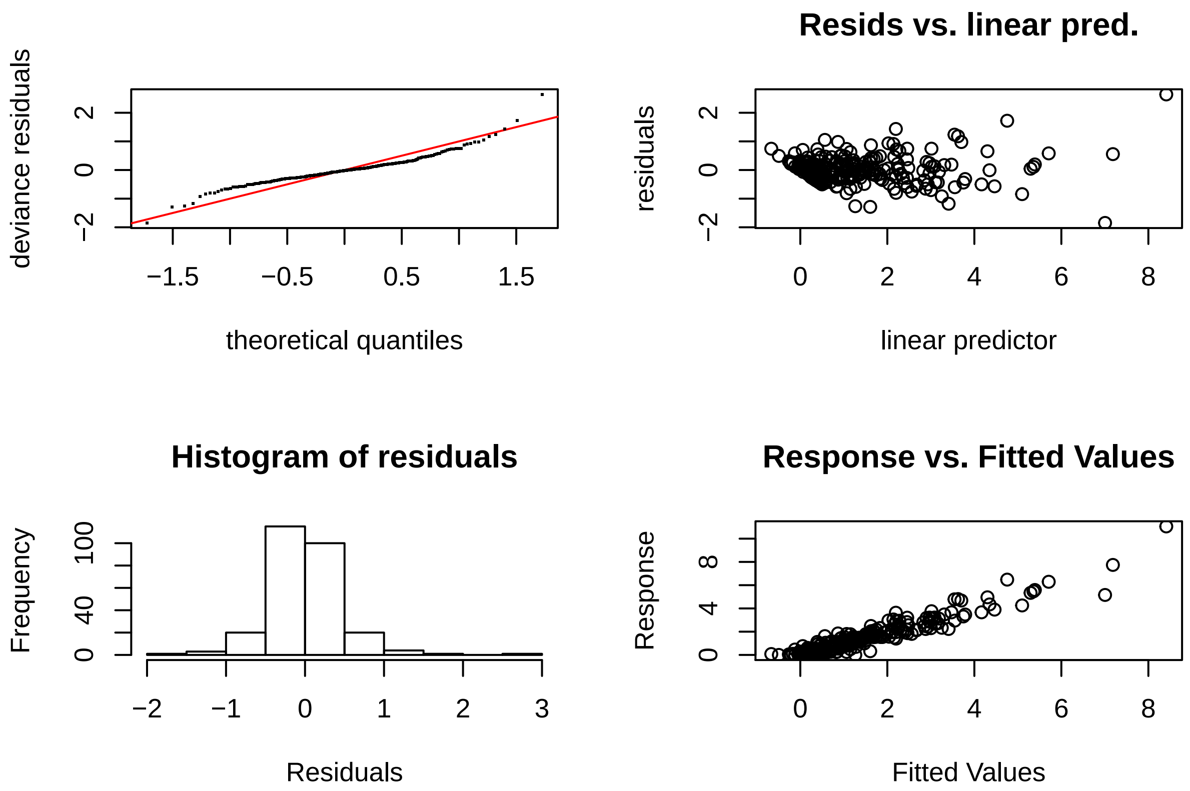



```r
summary(sbh_water.fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## smr ~ te(prec_mean, year) + te(tree_shrub_patch_density, year) +
##     te(flood_fresh_n_patches, year) + te(flood_fresh_patch_density,
##     year) + te(flood_fresh_prop, year) + te(flood_salt_n_patches,
##     year) + te(flood_salt_patch_density, year) + te(flood_salt_prop,
##     year)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.28200    0.03658   35.05   <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(prec_mean,year)                 5.562  5.999  2.458 0.028127 *
## te(tree_shrub_patch_density,year) 13.524 20.000  7.144  < 2e-16 ***
## te(flood_fresh_n_patches,year)    20.000 20.000 10.501  < 2e-16 ***
## te(flood_fresh_patch_density,year) 18.016 20.000  3.023 8.61e-09 ***
```

```
## te(flood_fresh_prop,year)          18.785 20.000  6.142  < 2e-16 ***
## te(flood_salt_n_patches,year)      18.127 20.000 10.894  < 2e-16 ***
## te(flood_salt_patch_density,year)   8.043  9.918  3.349 0.000765 ***
## te(flood_salt_prop,year)            3.699 16.000  1.107 2.92e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.842   Deviance explained = 90.5%
## GCV = 0.59387  Scale est. = 0.35463   n = 265
```

```r
gam.check(sbh_water.fit)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 84 iterations.
## The RMS GCV score gradient at convergence was 1.020103e-07 .
## The Hessian was positive definite.
## Model rank =  165 / 165
##
## 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
## te(prec_mean,year)                24.00  5.56    1.06    0.83
## te(tree_shrub_patch_density,year) 20.00 13.52    1.06    0.82
```

```
## te(flood_fresh_n_patches,year)     20.00 20.00    1.15    0.99
## te(flood_fresh_patch_density,year) 20.00 18.02    1.19    1.00
## te(flood_fresh_prop,year)          20.00 18.78    1.19    1.00
## te(flood_salt_n_patches,year)      20.00 18.13    1.05    0.80
## te(flood_salt_patch_density,year)  20.00  8.04    1.15    0.98
## te(flood_salt_prop,year)           20.00  3.70    1.26    1.00
```

```
pred_forest = predict(sbh_forest.fit, newdata = test_sbh)
pred_agri = predict(sbh_agri.fit, newdata = test_sbh)
pred_water = predict(sbh_water.fit, newdata = test_sbh)



mean(data.matrix(test_sbh[,57] - pred_forest)^2)
```

```
## [1] 1.007464
```

```
mean(data.matrix(test_sbh[,57] - pred_agri)^2)
```

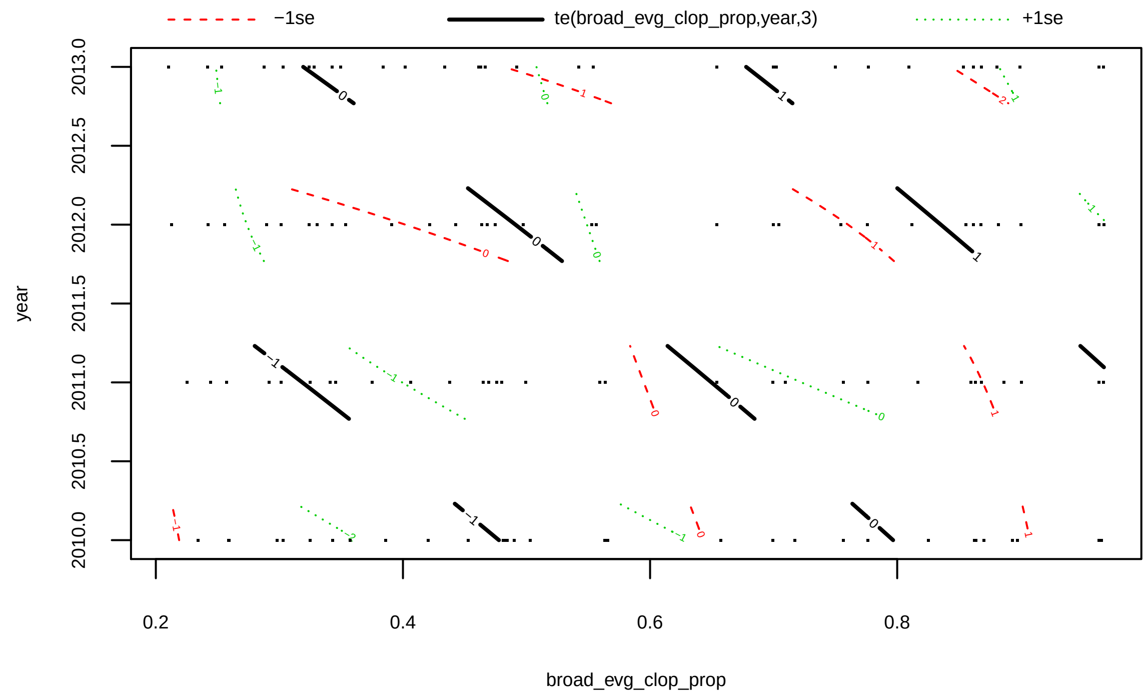
```
## [1] 4.180557
```

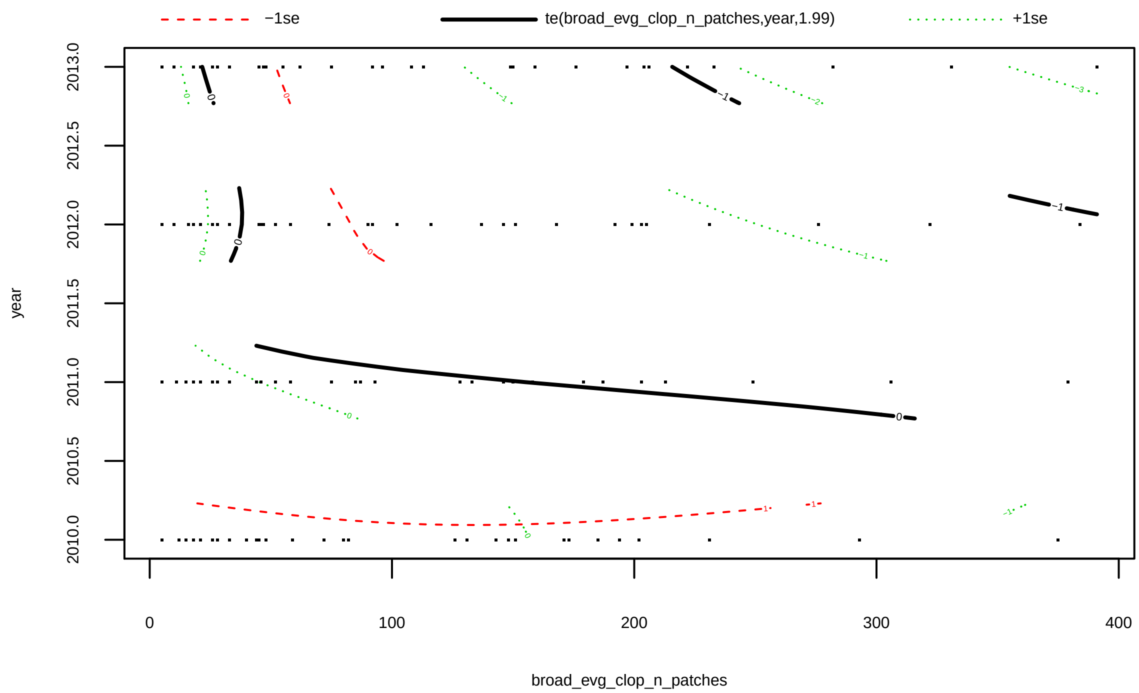
```
mean(data.matrix(test_sbh[,57] - pred_water)^2)
```

```
## [1] 4.67863
```

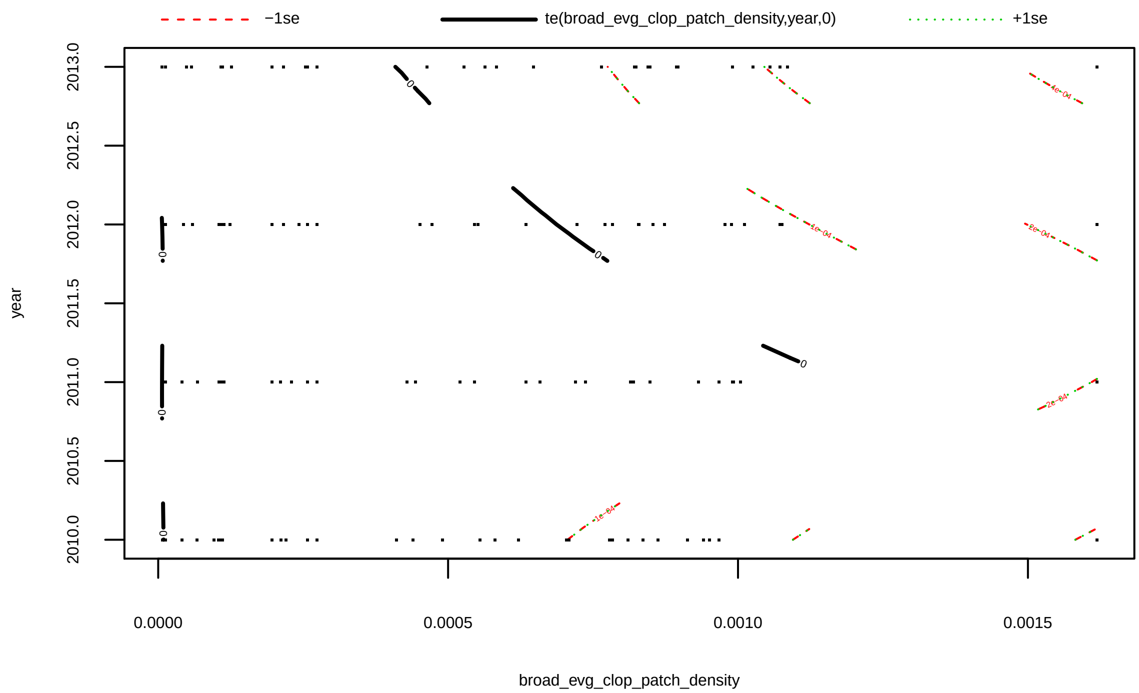
Sarawak

```
swk_forest.fit = gam(smr ~ te(broad_evg_clop_prop, year, k = c(10,4)) +
                       te(broad_evg_clop_n_patches, year, k = c(10,4)) +
                       te(broad_evg_clop_patch_density, year, k = c(10,4)),
                     data = train_swk, method = 'GCV.Cp'
                     #, family = poisson(link = log)
                     )

plot(swk_forest.fit)
```


−1se
te(broad_evg_clop_prop,year,3)
+1se
year
2013.0
2012.5
2012.0
2011.5
2011.0
2010.5
2010.0
0.2
0.4
0.6
0.8
broad_evg_clop_prop



−1se
te(broad_evg_clop_n_patches,year,1.99)
+1se
year
2010.0
2010.5
2011.0
2011.5
2012.0
2012.5
2013.0
0
100
200
300
400
broad_evg_clop_n_patches

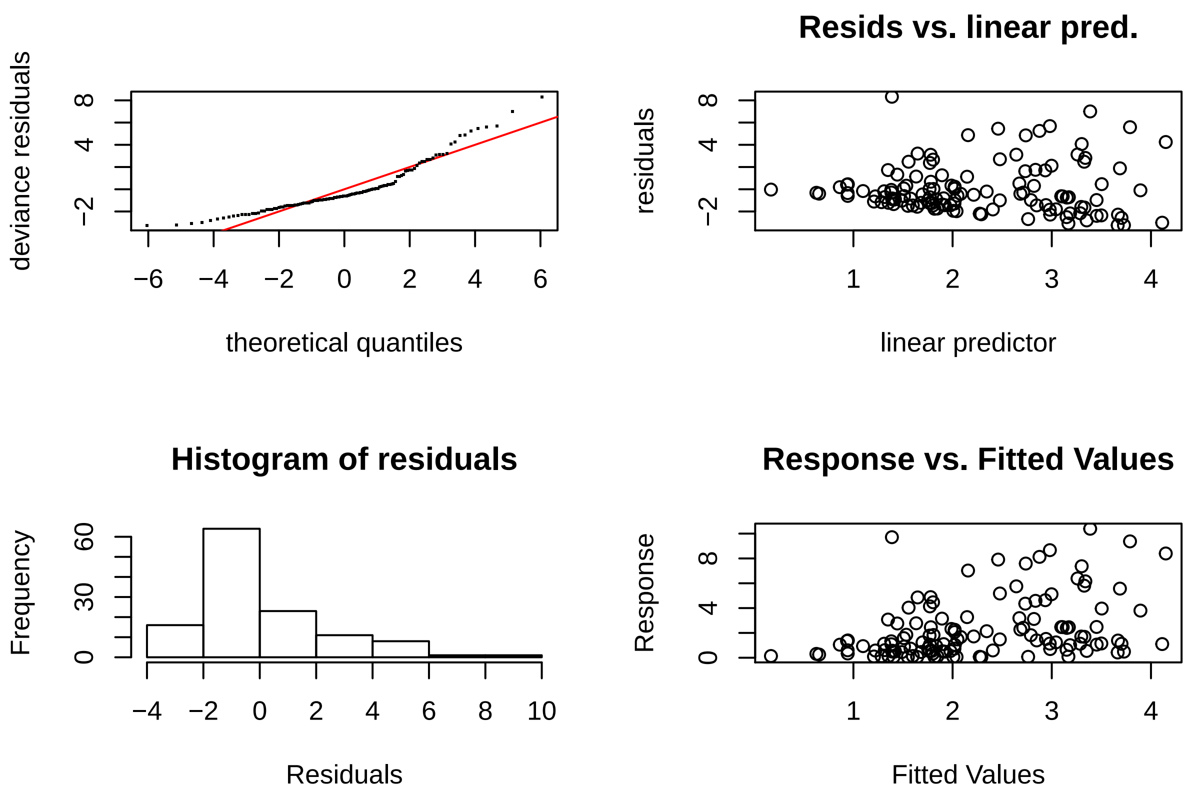



```r
summary(swk_forest.fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## smr ~ te(broad_evg_clop_prop, year, k = c(10, 4)) + te(broad_evg_clop_n_patches,
##     year, k = c(10, 4)) + te(broad_evg_clop_patch_density, year,
##     k = c(10, 4))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2566     0.2048   11.02   <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(broad_evg_clop_prop,year)          3.000e+00  3.000 4.769 0.00354 **
## te(broad_evg_clop_n_patches,year)     1.995e+00  2.511 1.418 0.26543
## te(broad_evg_clop_patch_density,year) 7.360e-08 36.000 0.000 0.48284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## R-sq.(adj) =  0.112   Deviance explained = 14.8%
## GCV = 5.4627  Scale est. = 5.1986    n = 124
```

```
gam.check(swk_forest.fit)
```




```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 24 iterations.
## The RMS GCV score gradient at convergence was 2.333913e-07 .
## The Hessian was positive definite.
## Model rank =  112 / 112
##
## 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
## te(broad_evg_clop_prop,year)          3.90e+01 3.00e+00    1.17    0.97
## te(broad_evg_clop_n_patches,year)     3.60e+01 1.99e+00    1.14    0.96
## te(broad_evg_clop_patch_density,year) 3.60e+01 7.36e-08    0.61  <2e-16
##
## te(broad_evg_clop_prop,year)
## te(broad_evg_clop_n_patches,year)
## te(broad_evg_clop_patch_density,year) ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
pred_forest2 = predict(swk_forest.fit, newdata = test_swk)

mean(data.matrix(test_swk[,5] - pred_forest2)^2)
```

```
## [1] 6.58894
```