Untitled

Lennart Hoheisel

6/4/2021

Project 1 - Understanding Forest Damage in Germany

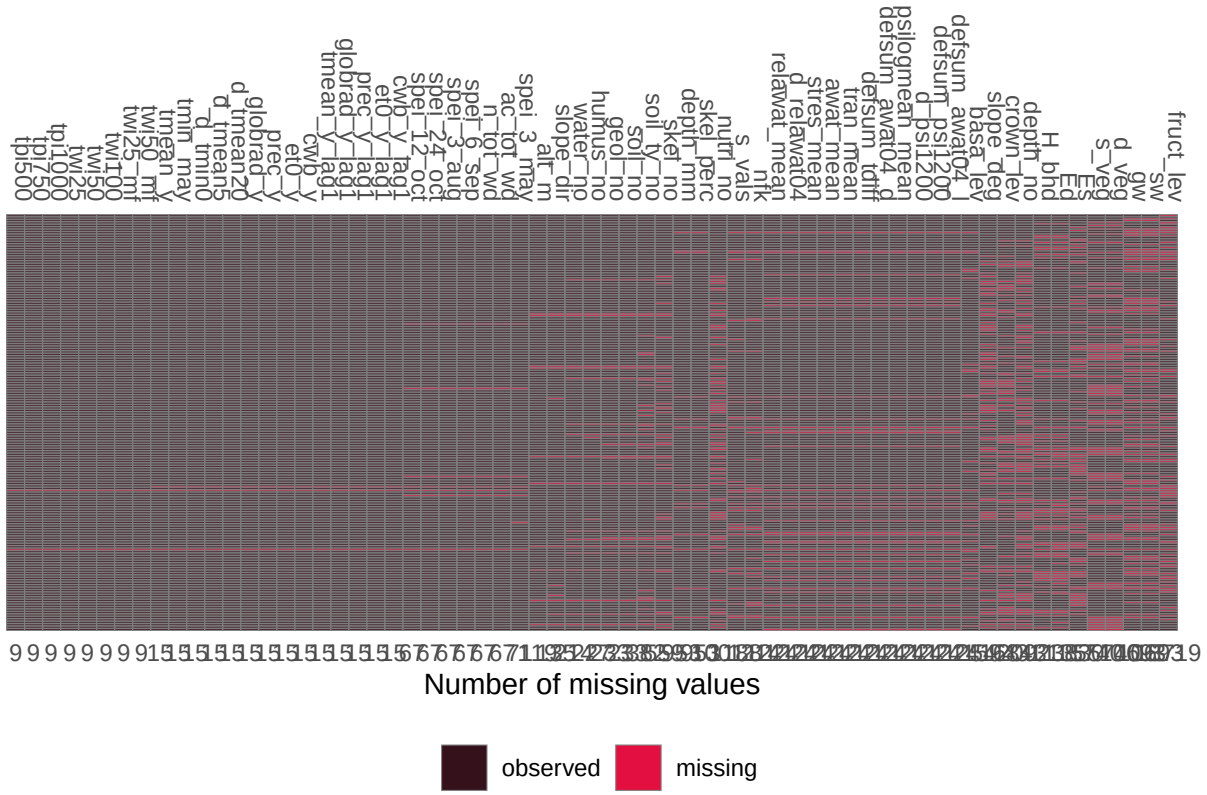
Lennart Hoheisel

```
# reading in the data
raw = read.table("foresthealth.txt", sep = ";", header = T, stringsAsFactors = T)
data = raw[,!names(raw) %in% "X" ]
```

```
# description of the data
dtype = as.data.frame(sapply(data,class))
rownames(dtype) <- 1:nrow(dtype); colnames(dtype) = c("obs_type")
dtype["index"] = colnames(data)
meaning = read.table("variablesNHA.csv", sep = ";", header = T)[, c(1, 3, 4)]
row74 = c("Es", "Evenness Index tree species", "numeric")
row75 = c("H_bhd", "Shannon's function tree diameter", "numeric")
meaning = rbind(meaning, c(row74))
meaning = rbind(meaning, c(row75))

colnames(meaning) = c("index", "description", "real_type")

desc = merge(dtype, meaning, by = "index")

# transforming each data column into the data type specified in variablesNHA.csv
for (row in 1:dim(desc)[1]) {
  dtyp_imp = desc[row, 2]
  dtyp_tru = desc[row, 4]
  if (dtyp_imp != dtyp_tru) {
    if (grepl("ordered factor", dtyp_tru)) {
      data[ , desc[row, 1]] = factor( as.vector( data[ , desc[row, 1]] ) ,
                                      ordered = is.ordered(data[ , desc[row, 1]]))
    } else if (grepl("factor", dtyp_tru)) {
      data[ , desc[row, 1]] = factor( as.vector( data[ , desc[row, 1]] ) )
    } else if (grepl("numeric", dtyp_tru)) {
      data[ , desc[row, 1]] = as.numeric( as.vector( data[ , desc[row, 1]] ) )
    }
  }
}

#checking whether dtypes align
dtype2 = as.data.frame(sapply(data,class))
rownames(dtype2) <- 1:nrow(dtype2); colnames(dtype2) = c("obs_type")
dtype2["index"] = colnames(data)
```

```
meaning2 = read.table("variablesNHA.csv", sep = ";", header = T)[, c(1, 3, 4)]
row74 = c("Es", "Evenness Index tree species", "numeric")
row75 = c("H_bhd", "Shannon's function tree diameter", "numeric")
meaning2 = rbind(meaning2, c(row74))
meaning2 = rbind(meaning2, c(row75))
colnames(meaning2) = c("index", "description", "real_type")

# final description of the data
desc2 = merge(dtype2, meaning2, by = "index")
```

```
# visualisation of missing values
dd = data
iter = 0
# dropping all columns where no NAs are present
for (col in colnames(dd) ) {
  iter = iter + 1
  if (typeof(dd[1, col]) == "str") {

  }
  else if (sum( is.na( as.vector(dd[, col]) ) ) == 0) {
    dd = dd[-iter]
    iter = iter - 1
  }
  else if (sum( is.na( as.vector(dd[, col]) ) ) > 0) {
  }
}

md_pattern(dd, pattern = FALSE, color = c('#34111b', '#e30f41'), print_yaxis = F)
```




```
sum(complete.cases(data))
```

```
## [1] 4990
```

```
# summary of data
summary(data)
```

```
##        id             x_utm            y_utm              year       source
##  624    :  127   Min.   :391728   Min.   :5269391   Min.   :1985   dbfl: 2748
##  634    :  107   1st Qu.:447949   1st Qu.:5319211   1st Qu.:1986   twi :28895
##  304    :   96   Median :487769   Median :5370287   Median :1991
##  1593   :   92   Mean   :490861   Mean   :5372134   Mean   :1995
##  786    :   92   3rd Qu.:535870   3rd Qu.:5414276   3rd Qu.:2002
##  1247   :   91   Max.   :603963   Max.   :5514225   Max.   :2016
##  (Other):31038
##  tree_sp_eu     tree_age        nbv_ratio         n_trees          gw
##  Dgl:  999   Min.   :  4.00   Min.   :0.0000   Min.   :  1.00   0   :20815
##  Gfi:11335   1st Qu.: 60.00   1st Qu.:0.1500   1st Qu.:  2.00   1   :  135
##  Gki: 3670   Median : 83.00   Median :0.2300   Median :  8.00   NA's:10693
##  Rbu: 7219   Mean   : 83.16   Mean   :0.2409   Mean   : 11.53
##  Tei: 3694   3rd Qu.:107.00   3rd Qu.:0.3100   3rd Qu.: 17.00
##  Wta: 4726   Max.   :267.00   Max.   :0.9900   Max.   :175.00
##
##     sw           geol_no        soil_no       soil_ty_no    humus_no
##  0   :19650   40     :9563   6      :8114   4      :16971   1   : 8822
##  1   : 1300   50     :5167   4      :5863   7      : 6312   2   : 6136
##  NA's:10693   60     :4767   3      :5048   11     : 2601   3   :12225
```

```
##              80     :3306   2      :3077   9      : 1531   4   : 2499
##              10     :3253   5      :2981   12     : 1375   5   : 1616
##              (Other):5354   (Other):6327   (Other): 2471   6   :  118
##              NA's   : 233   NA's   : 233   NA's   :  382   NA's:  227
##  water_no     nutri_no       slope_dir     skel_no      depth_no
##  1   :  720   1   : 3271   10     : 5997   1   :10521   1   : 5668
##  2   : 5010   2   :16366   1      : 4418   2   :18212   2   :17156
##  3   :17062   3   :11005   8      : 3688   3   : 2351   3   : 6727
##  4   : 6387   NA's: 1001   5      : 3302   NA's:  559   NA's: 2092
##  5   : 1250                2      : 3289
##  6   : 1000                (Other):10814
##  NA's:  214                NA's   :  135
##      alt_m          slope_deg           Es            H_spec
##  Min.   :  90.0   Min.   : 0.00   Min.   :0.060   Min.   :0.0000
##  1st Qu.: 390.0   1st Qu.: 5.00   1st Qu.:0.560   1st Qu.:0.3800
##  Median : 530.0   Median :13.00   Median :0.730   Median :0.7200
##  Mean   : 549.9   Mean   :19.17   Mean   :0.681   Mean   :0.7166
##  3rd Qu.: 700.0   3rd Qu.:30.00   3rd Qu.:0.840   3rd Qu.:1.0400
##  Max.   :1270.0   Max.   :99.00   Max.   :1.000   Max.   :2.0200
##  NA's   :119      NA's   :1594    NA's   :3830
##        Ed             H_bhd       crown_lev      fruct_lev        tmean_y
##  Min.   :0.3100   Min.   :0.000   1   : 4552   0      : 5812   Min.   : 3.170
##  1st Qu.:0.9600   1st Qu.:2.480   2   :12175   1      :  921   1st Qu.: 7.200
##  Median :0.9700   Median :2.630   3   :13232   0.1    :  666   Median : 8.180
##  Mean   :0.9651   Mean   :2.693   NA's: 1684   0.2    :  613   Mean   : 8.197
##  3rd Qu.:0.9800   3rd Qu.:2.790                0.5    :  543   3rd Qu.: 9.190
##  Max.   :1.0000   Max.   :4.810                (Other): 4369   Max.   :12.510
##  NA's   :3135     NA's   :3111                 NA's   :18719   NA's   :15
##     tmin_may          d_tmin0       d_tmean5       d_tmean20       globrad_y
##  Min.   :-15.390   Min.   : 23   Min.   :155.0   Min.   : 0.00   Min.   :3141
##  1st Qu.: -4.950   1st Qu.: 84   1st Qu.:218.0   1st Qu.: 9.00   1st Qu.:3827
##  Median : -3.320   Median :103   Median :234.0   Median :16.00   Median :3981
##  Mean   : -3.618   Mean   :102   Mean   :235.2   Mean   :18.35   Mean   :3971
##  3rd Qu.: -1.960   3rd Qu.:120   3rd Qu.:251.0   3rd Qu.:26.00   3rd Qu.:4129
##  Max.   :  3.780   Max.   :189   Max.   :322.0   Max.   :89.00   Max.   :4782
##  NA's   :15        NA's   :15    NA's   :15      NA's   :15      NA's   :15
##      prec_y          et0_y           cwb_y          tmean_y_lag1
##  Min.   : 356   Min.   :226.6   Min.   :-234.9   Min.   : 3.130
##  1st Qu.: 833   1st Qu.:373.6   1st Qu.: 392.7   1st Qu.: 6.990
##  Median :1010   Median :410.0   Median : 595.5   Median : 7.980
##  Mean   :1119   Mean   :413.0   Mean   : 706.2   Mean   : 8.009
##  3rd Qu.:1304   3rd Qu.:451.9   3rd Qu.: 907.9   3rd Qu.: 9.010
##  Max.   :2916   Max.   :693.2   Max.   :2649.0   Max.   :12.510
##  NA's   :15     NA's   :15      NA's   :15       NA's   :15
##  globrad_y_lag1  prec_y_lag1     et0_y_lag1      cwb_y_lag1         s_veg
##  Min.   :3141   Min.   : 429   Min.   :215.7   Min.   :-234.9   Min.   :100.0
##  1st Qu.:3815   1st Qu.: 838   1st Qu.:368.9   1st Qu.: 407.5   1st Qu.:127.0
##  Median :3969   Median : 989   Median :405.3   Median : 578.6   Median :132.0
##  Mean   :3961   Mean   :1112   Mean   :408.8   Mean   : 702.9   Mean   :132.9
##  3rd Qu.:4113   3rd Qu.:1289   3rd Qu.:447.5   3rd Qu.: 896.8   3rd Qu.:138.0
##  Max.   :4782   Max.   :2924   Max.   :693.2   Max.   :2649.0   Max.   :178.0
##  NA's   :15     NA's   :15     NA's   :15      NA's   :15       NA's   :5740
##      d_veg        spei_12_oct        spei_24_oct        spei_3_may
##  Min.   : 82.0   Min.   :-3.00000   Min.   :-3.0000   Min.   :-3.0000
```

```
##  1st Qu.:139.0   1st Qu.:-0.93000   1st Qu.:-0.7000   1st Qu.:-0.4900
##  Median :146.0   Median :-0.06000   Median : 0.0700   Median : 0.5100
##  Mean   :144.8   Mean   : 0.02697   Mean   : 0.0387   Mean   : 0.2396
##  3rd Qu.:152.0   3rd Qu.: 1.06000   3rd Qu.: 0.7700   3rd Qu.: 1.0100
##  Max.   :179.0   Max.   : 3.00000   Max.   : 3.0000   Max.   : 2.1800
##  NA's   :5740    NA's   :67         NA's   :67        NA's   :71
##    spei_3_aug        spei_6_sep       relawat_mean     d_relawat04
##  Min.   :-2.8300   Min.   :-3.0000   Min.   :0.0300   Min.   :  0.0
##  1st Qu.:-0.6200   1st Qu.:-0.5700   1st Qu.:0.2800   1st Qu.: 53.5
##  Median : 0.1200   Median : 0.1600   Median :0.4200   Median : 85.0
##  Mean   : 0.1429   Mean   : 0.2124   Mean   :0.4457   Mean   : 80.6
##  3rd Qu.: 0.8500   3rd Qu.: 1.0800   3rd Qu.:0.5800   3rd Qu.:108.0
##  Max.   : 3.0000   Max.   : 3.0000   Max.   :1.3400   Max.   :296.0
##  NA's   :67        NA's   :67        NA's   :1224     NA's   :1224
##    stres_mean       awat_mean        tran_mean      defsum_tdiff
##  Min.   :0.2000   Min.   :  2.23   Min.   :0.360   Min.   :  0.15
##  1st Qu.:0.6600   1st Qu.: 22.70   1st Qu.:1.540   1st Qu.: 52.41
##  Median :0.7800   Median : 37.92   Median :1.830   Median :113.28
##  Mean   :0.7685   Mean   : 41.08   Mean   :1.809   Mean   :123.45
##  3rd Qu.:0.9000   3rd Qu.: 54.98   3rd Qu.:2.090   3rd Qu.:180.91
##  Max.   :1.0000   Max.   :214.82   Max.   :3.240   Max.   :640.40
##  NA's   :1224     NA's   :1224     NA's   :1224    NA's   :1224
##  defsum_awat04_d  psilogmean_mean      d_psi1200      defsum_psi1200
##  Min.   :   0.0   Min.   :-19212.2   Min.   :  0.00   Min.   :-46560927
##  1st Qu.: 646.7   1st Qu.: -7207.5   1st Qu.: 58.00   1st Qu.:-17672150
##  Median :1273.7   Median : -4813.1   Median : 88.00   Median :-12298438
##  Mean   :1310.7   Mean   : -5053.8   Mean   : 84.25   Mean   :-12538221
##  3rd Qu.:1849.2   3rd Qu.: -2447.3   3rd Qu.:112.00   3rd Qu.: -6656154
##  Max.   :6333.3   Max.   :   -32.1   Max.   :189.00   Max.   :        0
##  NA's   :1224     NA's   :1224       NA's   :1224     NA's   :1224
##  defsum_awat04_l    n_tot_wd       ac_tot_wd          s_vals
##  Min.   :    0   Min.   : 45.0   Min.   : 120.0   Min.   :  0.1313
##  1st Qu.: 1165   1st Qu.: 97.0   1st Qu.: 402.0   1st Qu.:  5.6978
##  Median : 2251   Median :125.0   Median : 612.0   Median : 15.2486
##  Mean   : 2484   Mean   :126.1   Mean   : 602.3   Mean   : 32.1250
##  3rd Qu.: 3546   3rd Qu.:150.0   3rd Qu.: 770.0   3rd Qu.: 50.1770
##  Max.   :11484   Max.   :227.0   Max.   :1466.0   Max.   :283.3598
##  NA's   :1224    NA's   :67      NA's   :67       NA's   :1188
##  basa_lev        depth_mm           nfk           skel_perc
##  1   : 5054   Min.   :   0.0   Min.   : 1.264   Min.   : 0.00
##  2   : 2572   1st Qu.: 579.0   1st Qu.: 8.557   1st Qu.: 7.80
##  3   : 2621   Median : 796.0   Median :12.426   Median :19.15
##  4   : 7478   Mean   : 735.4   Mean   :11.428   Mean   :25.47
##  5   :11791   3rd Qu.: 913.8   3rd Qu.:13.989   3rd Qu.:39.50
##  6   :  873   Max.   :1586.0   Max.   :26.463   Max.   :89.80
##  NA's: 1254   NA's   :953      NA's   :1214     NA's   :953
##      tpi500            tpi750            tpi1000             twi25
##  Min.   :-82.300   Min.   :-105.600   Min.   :-147.100   Min.   : 3.400
##  1st Qu.: -5.100   1st Qu.:  -6.500   1st Qu.:  -7.800   1st Qu.: 4.600
##  Median :  3.200   Median :   4.800   Median :   6.000   Median : 5.300
##  Mean   :  4.772   Mean   :   6.626   Mean   :   7.985   Mean   : 5.515
##  3rd Qu.: 15.200   3rd Qu.:  21.400   3rd Qu.:  23.900   3rd Qu.: 6.200
##  Max.   : 93.900   Max.   : 118.500   Max.   : 153.500   Max.   :11.000
##  NA's   :9         NA's   :9          NA's   :9          NA's   :9
```

```
##      twi50            twi100          twi25_mf        twi50_mf
##  Min.   : 4.100   Min.   : 4.700   Min.   :3.600   Min.   : 4.200
##  1st Qu.: 5.200   1st Qu.: 5.900   1st Qu.:4.800   1st Qu.: 5.400
##  Median : 6.000   Median : 6.600   Median :5.400   Median : 6.000
##  Mean   : 6.123   Mean   : 6.758   Mean   :5.543   Mean   : 6.148
##  3rd Qu.: 6.800   3rd Qu.: 7.400   3rd Qu.:6.200   3rd Qu.: 6.800
##  Max.   :11.400   Max.   :11.900   Max.   :9.700   Max.   :10.400
##  NA's   :9        NA's   :9        NA's   :9       NA's   :9
```

```
# correlation plot
dd = data %>%
  select(order(colnames(data)))
dd = dd[ , -which(desc2[,2] %in% c("factor"))]

cormat <- round(cor(dd[complete.cases(dd),]),2)
head(cormat)
```

```
##            ac_tot_wd alt_m awat_mean cwb_y cwb_y_lag1 d_psi1200 d_relawat04
## ac_tot_wd       1.00  0.18      0.35  0.55       0.50     -0.48       -0.47
## alt_m           0.18  1.00      0.07  0.54       0.54     -0.45       -0.45
## awat_mean       0.35  0.07      1.00  0.43       0.28     -0.76       -0.75
## cwb_y           0.55  0.54      0.43  1.00       0.78     -0.71       -0.70
## cwb_y_lag1      0.50  0.54      0.28  0.78       1.00     -0.55       -0.54
## d_psi1200      -0.48 -0.45     -0.76 -0.71      -0.55      1.00        0.99
##            d_tmean20 d_tmean5 d_tmin0 d_veg defsum_awat04_d defsum_awat04_l
## ac_tot_wd      -0.33    -0.38    0.22 -0.34           -0.24           -0.40
## alt_m          -0.70    -0.72    0.66 -0.69           -0.45           -0.54
## awat_mean      -0.04    -0.09    0.03 -0.05           -0.02           -0.44
## cwb_y          -0.39    -0.35    0.20 -0.41           -0.40           -0.63
## cwb_y_lag1     -0.45    -0.32    0.23 -0.48           -0.39           -0.49
## d_psi1200       0.35     0.38   -0.31  0.39            0.52            0.82
##            defsum_psi1200 defsum_tdiff depth_mm    Ed    Es et0_y et0_y_lag1
## ac_tot_wd            0.46        -0.45     0.23  0.03  0.01 -0.50      -0.48
## alt_m                0.39        -0.34    -0.05 -0.03 -0.13 -0.50      -0.44
## awat_mean            0.78        -0.73     0.41  0.09 -0.01 -0.30      -0.11
## cwb_y                0.67        -0.62     0.29  0.06  0.02 -0.60      -0.42
## cwb_y_lag1           0.49        -0.45     0.30  0.03  0.02 -0.48      -0.57
## d_psi1200           -0.91         0.81    -0.24 -0.06  0.02  0.57       0.33
##            globrad_y globrad_y_lag1 H_bhd H_spec n_tot_wd n_trees nbv_ratio
## ac_tot_wd      -0.58          -0.49 -0.18  -0.14     0.95   -0.02     -0.13
## alt_m          -0.07          -0.03  0.01  -0.29     0.20    0.11      0.05
## awat_mean      -0.22           0.03  0.06  -0.03     0.34   -0.10      0.00
## cwb_y          -0.42          -0.19  0.03  -0.15     0.57    0.01      0.09
## cwb_y_lag1     -0.30          -0.37 -0.01  -0.15     0.47    0.01      0.06
## d_psi1200       0.39           0.09 -0.02   0.12    -0.49    0.05     -0.04
##              nfk prec_y prec_y_lag1 psilogmean_mean relawat_mean s_vals s_veg
## ac_tot_wd  -0.10   0.52        0.47            0.43         0.46  -0.23  0.39
## alt_m      -0.43   0.51        0.52            0.35         0.40  -0.23  0.69
## awat_mean   0.41   0.42        0.28            0.78         0.75  -0.12  0.05
## cwb_y      -0.29   0.99        0.78            0.67         0.73  -0.40  0.41
## cwb_y_lag1 -0.29   0.78        0.99            0.47         0.54  -0.42  0.43
## d_psi1200   0.14  -0.68       -0.54           -0.88        -0.95   0.26 -0.37
##            skel_perc slope_deg spei_12_oct spei_24_oct spei_3_aug spei_3_may
## ac_tot_wd       0.22      0.20        0.31        0.34       0.21       0.45
## alt_m           0.51      0.15        0.02        0.01       0.02       0.07
```

```
## awat_mean      -0.25     -0.11        0.37        0.28       0.41       0.26
## cwb_y           0.47      0.35        0.47        0.37       0.32       0.36
## cwb_y_lag1      0.47      0.36        0.06        0.29       0.10       0.08
## d_psi1200      -0.29     -0.19       -0.41       -0.31      -0.46      -0.30
##            spei_6_sep stres_mean tmean_y tmean_y_lag1 tmin_may tpi1000 tpi500
## ac_tot_wd        0.38       0.42   -0.44        -0.46    -0.26    0.04   0.05
## alt_m            0.04       0.27   -0.75        -0.74    -0.43    0.21   0.20
## awat_mean        0.46       0.78   -0.13        -0.12    -0.10    0.05   0.00
## cwb_y            0.41       0.65   -0.40        -0.43    -0.29    0.13   0.13
## cwb_y_lag1       0.07       0.44   -0.39        -0.41    -0.04    0.13   0.13
## d_psi1200       -0.52      -0.82    0.46         0.42     0.32   -0.12  -0.11
##            tpi750 tran_mean tree_age twi100 twi25 twi25_mf twi50 twi50_mf x_utm
## ac_tot_wd    0.04      0.35    -0.01  -0.15 -0.15    -0.17 -0.15    -0.17 -0.20
## alt_m        0.20      0.09     0.08  -0.15 -0.15    -0.19 -0.15    -0.18 -0.19
## awat_mean    0.02      0.74     0.06   0.11  0.10     0.11  0.11     0.11 -0.12
## cwb_y        0.13      0.48     0.09  -0.29 -0.28    -0.33 -0.28    -0.33 -0.49
## cwb_y_lag1   0.13      0.31     0.08  -0.29 -0.28    -0.33 -0.28    -0.34 -0.50
## d_psi1200   -0.11     -0.63    -0.09   0.16  0.16     0.19  0.16     0.19  0.30
##            y_utm  year
## ac_tot_wd  -0.09 -0.80
## alt_m      -0.61 -0.01
## awat_mean  -0.12 -0.10
## cwb_y      -0.35 -0.10
## cwb_y_lag1 -0.33 -0.14
## d_psi1200   0.33  0.15
```

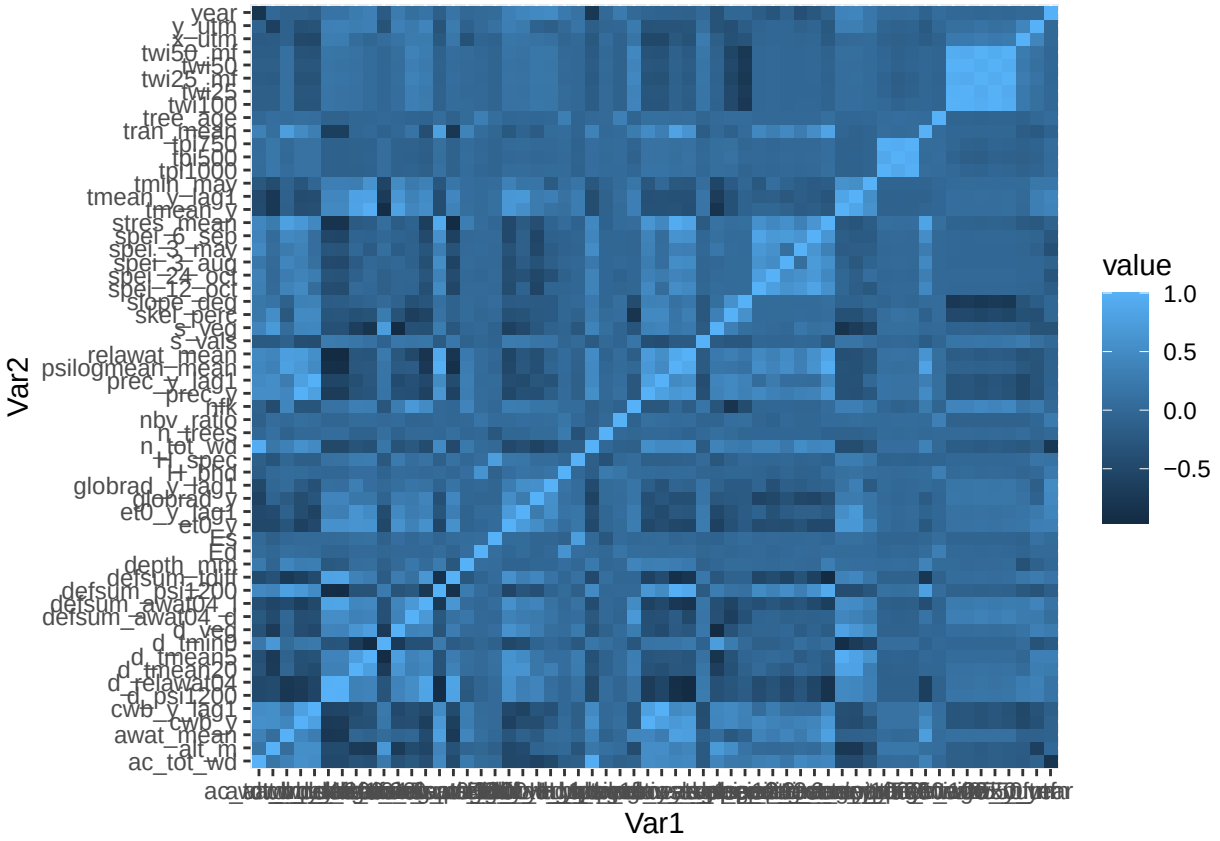
```
melted_cormat <- melt(cormat)
```

```
## Warning in melt(cormat): The melt generic in data.table has been passed a matrix
## and will attempt to redirect to the relevant reshape2 method; please note that
## reshape2 is deprecated, and this redirection is now deprecated as well. To
## continue using melt methods from reshape2 while both libraries are attached,
## e.g. melt.list, you can prepend the namespace like reshape2::melt(cormat). In
## the next version, this warning will become an error.
```

```
head(melted_cormat)
```

```
##         Var1      Var2 value
## 1  ac_tot_wd ac_tot_wd  1.00
## 2      alt_m ac_tot_wd  0.18
## 3  awat_mean ac_tot_wd  0.35
## 4      cwb_y ac_tot_wd  0.55
## 5 cwb_y_lag1 ac_tot_wd  0.50
## 6  d_psi1200 ac_tot_wd -0.48
```

```
# plotting the correlation as a first indicator
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile()
```



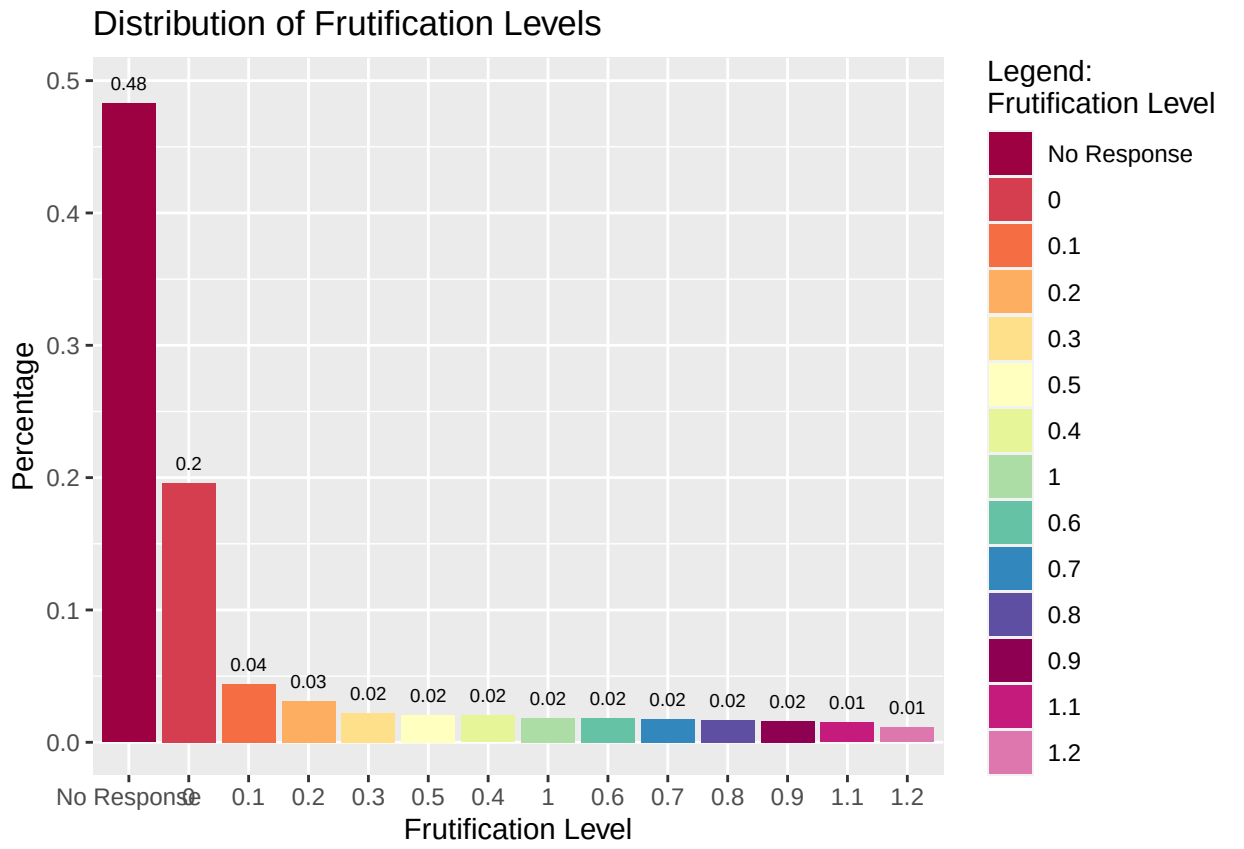

```
# function to plot categorical variables by plot (NOT ADJUSTED FOR NUMBER OF TREES IN THE PLOT)
figures = function(data, var, colours =  c("#FF5733", " #9625FA"), xlab, ylab, title) {


  total = as.numeric( length( dim( data )[1] ) )

  # finding all categories and their respective density limiting to categories with density beyond 1%
  data <- data %>%
    select(c(n_trees, {{var}}, spat, year)) %>%
    mutate( lev = str( get(var) ) ) %>%
    group_by( {{var}} ) %>%
    summarise(tot = sum(n_trees)) %>%
    mutate(dens =  tot / sum(data$n_trees)) %>%

    filter( dens>=0.01 ) %>%

    # renaming NAs
    mutate( lev = fct_explicit_na( get(var), "No Response" ) ) %>%
    arrange( desc(dens) )

  m = ggplot(data, aes(x=reorder(lev, -n), y=dens, fill=reorder(lev, -n))) +
        # make bar plot according to density found above
        geom_bar(stat = "identity") +
        scale_fill_manual( values = c(brewer.pal(n=11, "Spectral"),
                                      brewer.pal(n=10, "PiYG") ),
                           name = paste0("Legend: \n", xlab ) ) +
```

```
        # add density labels
        geom_text( aes( label=round(dens,2), y = dens + 0.01), vjust=0, color="black",
              position = position_dodge(0.9), size=2.5) +
        theme(legend.position = "right") +
        xlab(xlab) +
        ylab(ylab) +
        ggtitle(title)
  m
}
```

```
# figures for categorical variables adjusted for number of trees in the plot
# the figure code follows the same structure as the function above and is not explained for each figure

# figure 1: Distribution of Frutification Levels
data$spat <- factor(paste0(data$x_utm, data$y_utm))
dat_fruct <- data %>%
  # finding the frutification level by number of trees with this level
  select(c(n_trees, fruct_lev, spat, year)) %>%
  mutate( lev = str( fruct_lev ) ) %>%
  group_by(fruct_lev) %>%
  summarise(tot = sum(n_trees)) %>%
  mutate(dens =  tot / sum(data$n_trees)) %>%
  filter( dens>=0.01 ) %>%

  mutate( lev = fct_explicit_na( fruct_lev , "No Response" ) ) %>%
  arrange( desc(dens) )
```

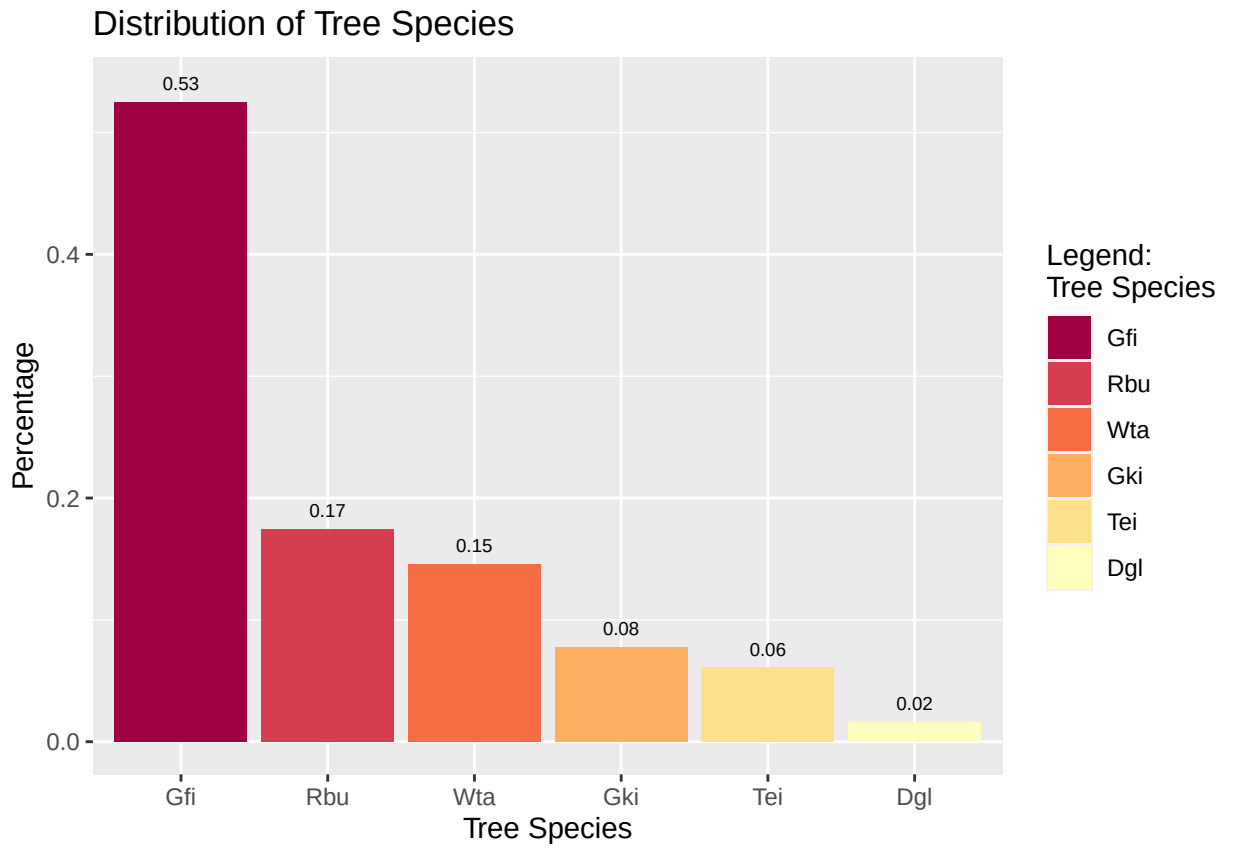
```
##  Factor w/ 31 levels "0","0.1","0.2",..: NA NA NA NA NA NA NA 1 1 1 ...
```

```
m = ggplot(dat_fruct, aes(x=reorder(lev, -tot), y=dens, fill=reorder(lev, -tot))) +

      geom_bar(stat = "identity") +
      scale_fill_manual( values = c(brewer.pal(n=11, "Spectral"),
                                    brewer.pal(n=10, "PiYG") ),
                         name = paste0("Legend: \n", "Frutification Level" ) ) +
      geom_text( aes( label=round(dens,2), y = dens + 0.01), vjust=0, color="black",
            position = position_dodge(0.9), size=2.5) +

      theme(legend.position = "right") +
      xlab("Frutification Level") +
      ylab("Percentage") +
      ggtitle("Distribution of Frutification Levels")
m
```




```r
# figure 2: Distribution of Tree Species in Europe
data$spat <- factor(paste0(data$x_utm, data$y_utm))
dat_n_trees <- data %>%
  select(c(n_trees, tree_sp_eu, spat, year)) %>%
  group_by(tree_sp_eu) %>%
  summarise(tot_spec = sum(n_trees))%>%
  mutate(tree_sp_eu, dens =  tot_spec / sum(data$n_trees)) %>%

  filter( dens>=0.01 ) %>%
  mutate( lev = fct_explicit_na( tree_sp_eu, "No Response" ) ) %>%
  arrange( desc(dens) )

xlab="Tree Species"; ylab="Percentage"; title="Distribution of Tree Species"
m = ggplot(dat_n_trees, aes(x=reorder(lev, -tot_spec), y=dens, fill=reorder(lev, -tot_spec))) +

      geom_bar(stat = "identity") +
      scale_fill_manual( values = c(brewer.pal(n=11, "Spectral"),
                                    brewer.pal(n=10, "PiYG") ),
                         name = paste0("Legend: \n", xlab ) ) +
      geom_text( aes( label=round(dens,2), y = dens + 0.01), vjust=0, color="black",
            position = position_dodge(0.9), size=2.5) +

      theme(legend.position = "right") +
      xlab(xlab) +
      ylab(ylab) +
      ggtitle(title)
```

```
m
```

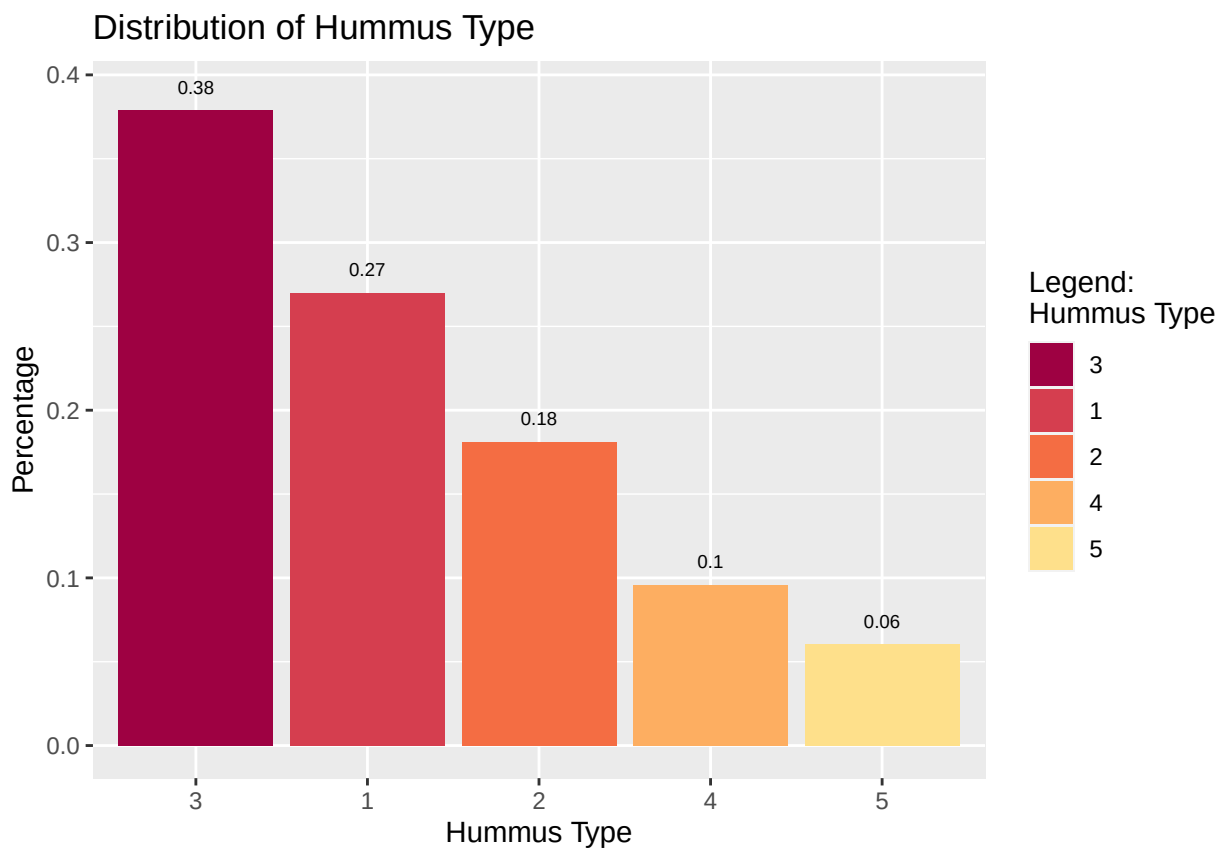



```
# figure 3: Distribution of Hummus Type
dat_hummus <- data %>%
  select(c(n_trees, humus_no, spat, year)) %>%
  mutate( lev = str( humus_no ) ) %>%
  group_by(humus_no) %>%
  summarise(tot = sum(n_trees)) %>%
  mutate(dens =  tot / sum(data$n_trees)) %>%

  filter( dens>=0.01 ) %>%

  mutate( lev = fct_explicit_na( humus_no , "No Response" ) ) %>%
  arrange( desc(dens) )
```

```
##  Factor w/ 6 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
m = ggplot(dat_hummus, aes(x=reorder(lev, -tot), y=dens, fill=reorder(lev, -tot))) +

      geom_bar(stat = "identity") +
      scale_fill_manual( values = c(brewer.pal(n=11, "Spectral"),
                                    brewer.pal(n=10, "PiYG") ),
                         name = paste0("Legend: \n", "Hummus Type" ) ) +
      geom_text( aes( label=round(dens,2), y = dens + 0.01), vjust=0, color="black",
            position = position_dodge(0.9), size=2.5) +

      theme(legend.position = "right") +
```

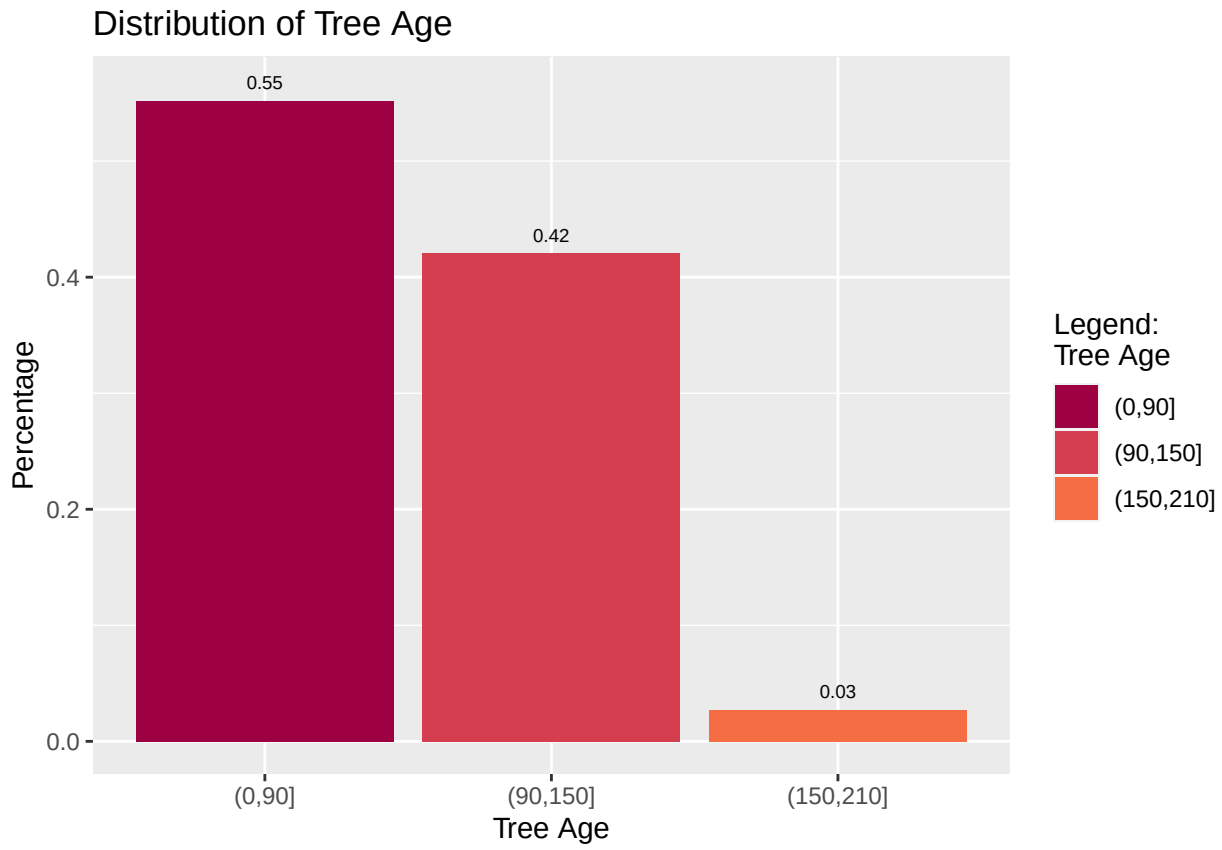
```
      xlab("Hummus Type") +
      ylab("Percentage") +
      ggtitle("Distribution of Hummus Type")
m
```




```
# tree age intervals
tree_ints <- data %>%
  select(tree_age) %>%
  mutate(ints = cut(tree_age, breaks = c(0, 90, 150, 210, 270) ))

# figure 4: Distribution of Tree Age
dat_age <- data %>%
  select(c(n_trees, tree_age, spat, year)) %>%
  mutate(ints = cut(tree_age, breaks = c(0, 90, 150, 210, 270) )) %>%
  mutate( lev = str( ints ) ) %>%
  group_by(ints) %>%
  summarise(tot = sum(n_trees)) %>%
  mutate(dens =  tot / sum(data$n_trees)) %>%

  filter( dens>=0.01 ) %>%

  mutate( lev = fct_explicit_na( ints , "No Response" ) ) %>%
  arrange( desc(dens) )
```

```
##  Factor w/ 4 levels "(0,90]","(90,150]",..: 2 2 2 2 1 1 1 2 2 2 ...
```

```
m = ggplot(dat_age, aes(x=reorder(lev, -tot), y=dens, fill=reorder(lev, -tot))) +

      geom_bar(stat = "identity") +
      scale_fill_manual( values = c(brewer.pal(n=11, "Spectral"),
                                    brewer.pal(n=10, "PiYG") ),
                         name = paste0("Legend: \n", "Tree Age" ) ) +
      geom_text( aes( label=round(dens,2), y = dens + 0.01), vjust=0, color="black",
            position = position_dodge(0.9), size=2.5) +

      theme(legend.position = "right") +
      xlab("Tree Age") +
      ylab("Percentage") +
      ggtitle("Distribution of Tree Age")
m
```

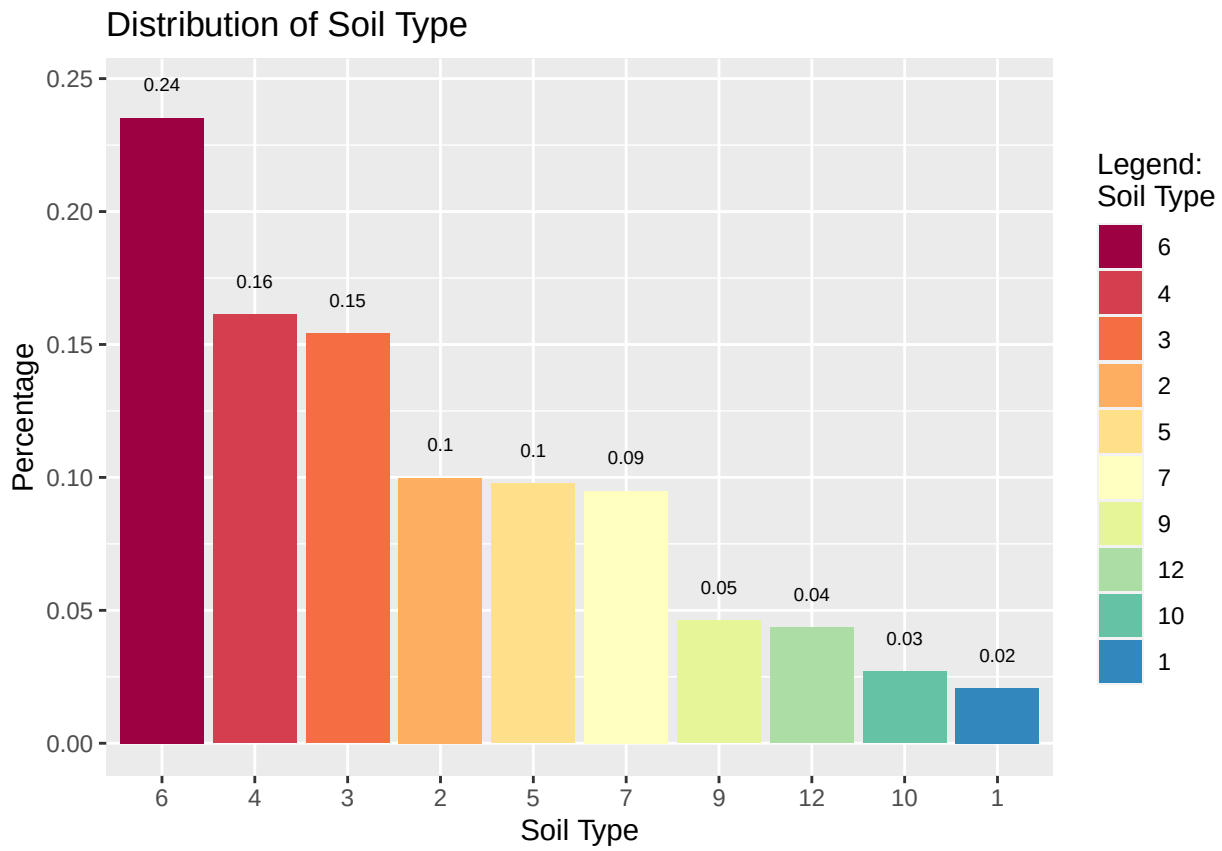



```
# figure 5: Distribution of Soil Type
dat_soil <- data %>%
  select(c(n_trees, soil_no, spat, year)) %>%
  mutate( lev = str( soil_no ) ) %>%
  group_by(soil_no) %>%
  summarise(tot = sum(n_trees)) %>%
  mutate(dens =  tot / sum(data$n_trees)) %>%

  filter( dens>=0.01 ) %>%

  mutate( lev = fct_explicit_na( soil_no , "No Response" ) ) %>%
```

```
  arrange( desc(dens) )
```

```
##  Factor w/ 13 levels "1","2","3","4",..: 7 7 7 7 6 6 6 6 6 6 ...
```

```
m = ggplot(dat_soil, aes(x=reorder(lev, -tot), y=dens, fill=reorder(lev, -tot))) +

      geom_bar(stat = "identity") +
      scale_fill_manual( values = c(brewer.pal(n=11, "Spectral"),
                                    brewer.pal(n=10, "PiYG") ),
                         name = paste0("Legend: \n", "Soil Type" ) ) +
      geom_text( aes( label=round(dens,2), y = dens + 0.01), vjust=0, color="black",
            position = position_dodge(0.9), size=2.5) +

      theme(legend.position = "right") +
      xlab("Soil Type") +
      ylab("Percentage") +
      ggtitle("Distribution of Soil Type")
m
```



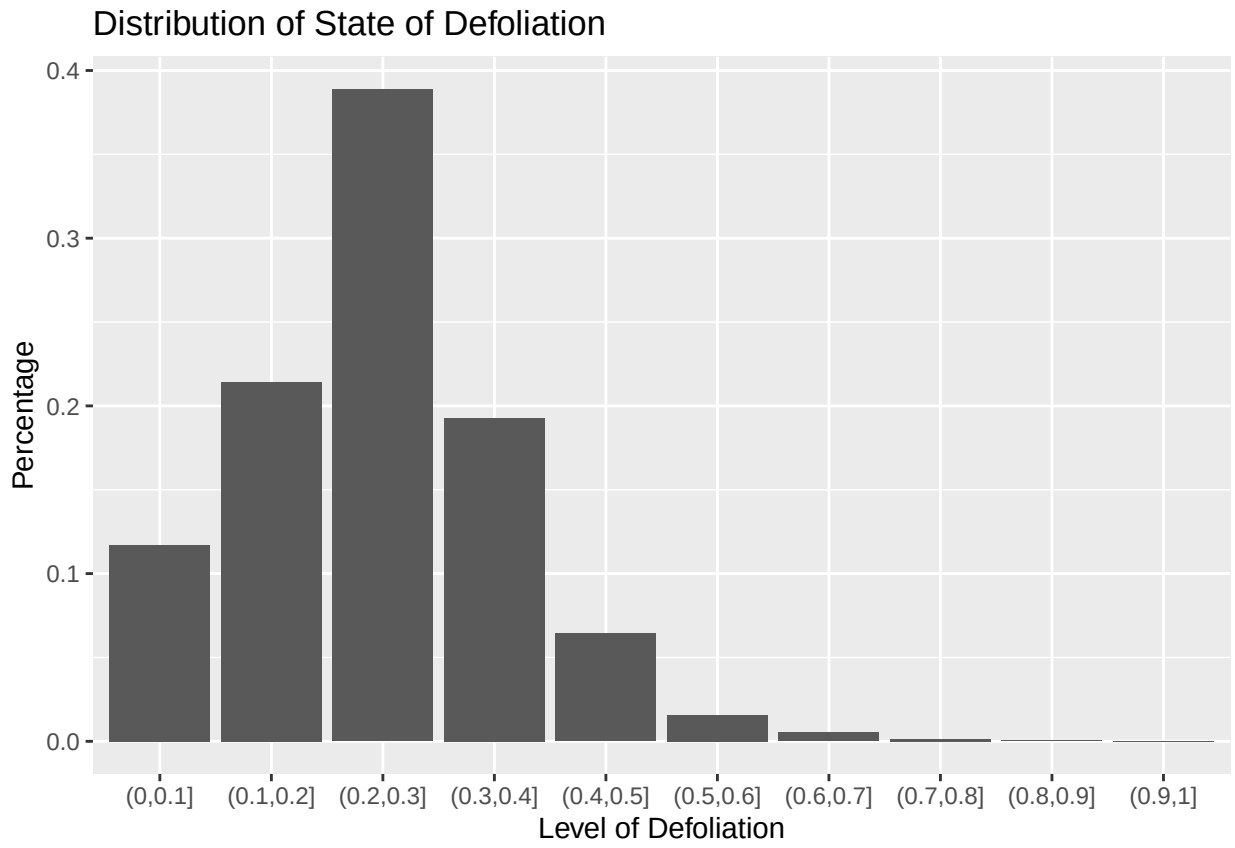

```
# figure5: Density of NBV
df <- data %>%
  select(nbv_ratio, n_trees) %>%
  mutate( ints = cut(nbv_ratio, breaks = seq(0, 1, 0.1) )) %>%
  mutate( lev = str( ints ) ) %>%
  group_by(ints) %>%
  summarise(tot = sum(n_trees)) %>%
  mutate(dens =  tot / sum(data$n_trees))
```

```
##  Factor w/ 10 levels "(0,0.1]","(0.1,0.2]",..: 1 2 2 2 1 1 1 1 1 3 ...
```

```
sum(df$dens[6:10])
```

```
## [1] 0.02305205
```

```
m <- ggplot(df) +
  geom_bar(aes(x=ints, y=dens), stat="identity")+
  theme(legend.position = "right") +
  xlab("Level of Defoliation") +
  ylab("Percentage") +
  ggtitle("Distribution of State of Defoliation")

m
```




```
# split into train and test
train_test_split <- function(data, train_size=0.75) {


  # splitting data into groups
  tree_species_prep <- data %>%
    select(-c(id, x_utm, y_utm, sw, gw, fruct_lev, spat, s_veg, d_veg)) %>%
    na.omit()

  tree_species_train <- tree_species_prep %>%
    sample_frac(train_size)

  Y_train <- tree_species_train %>%
```

```
    select(c(nbv_ratio, tree_sp_eu)) %>%
    group_split(tree_sp_eu)

  X_train <- tree_species_train %>%
    select(-c(nbv_ratio)) %>%
    group_split(tree_sp_eu)

  tree_species_test <- tree_species_prep %>%
    anti_join(tree_species_train)

  Y_test <- tree_species_test %>%
    select(c(nbv_ratio, tree_sp_eu)) %>%
    group_split(tree_sp_eu)

  X_test <- tree_species_test %>%
    select(-c(nbv_ratio)) %>%
    group_split(tree_sp_eu)

  return(list(X_train, Y_train, X_test, Y_test))

}
```

```
# modelling the random FOrest for each species
# splitting by tree species and accordinf to a random split for training and testing the model
splits = train_test_split(data, train_size=0.75)
```

```
## Joining, by = c("year", "source", "tree_sp_eu", "tree_age", "nbv_ratio", "n_trees", "geol_no", "soil
```

```
# model function
model_rf <- function(spec_data_X, spec_data_y) {
  # Build the model on the basis of training data as described above
  #
  # Inputs:
  # - spec_data_X (tibble)   <- training dataset, containing the feature information by species
  # - spec_data_y (tibble)   <- training dataset, containing the response information by species
  #
  # Outputs:
  #   - models (list)       <- list containing the model for each species
  models = list()
  for (id in 1:length(spec_data_X)) {
    # size of the data for each species usable by randomForest
    print( dim( as.data.frame( spec_data_X[[id]] ) ) )
    models[[id]] = randomForest( as.data.frame( spec_data_X[[id]] ),  spec_data_y[[id]] %>% pull(nbv_ra
  }
  return(models)
}

models <- model_rf(splits[[1]], splits[[2]])
```

```
## [1] 529  66
## [1] 4928   66
## [1] 2023   66
## [1] 4227   66
## [1] 2013   66
## [1] 2669   66
```

```r
# evaluate the model
evaluate <- function(models, X_test, y_test) {
  # Evaluate the accuracy, precision and recall
  #
  # Inputs:
  # - model(prediction model)   <- the model summary statistics are to be found for
  # - X_test (feature matrix)   <- test dataset, containing the feature information
  # - Y_test (response vector)  <- test dataset, containing the response information
  #
  # Outputs:
  #   - scores (data frame)       <- data frame containing accuracy, precision, recall scores
  scores <- as.data.frame(matrix(NA, 6, 4))
  colnames(scores) <- c("Accuracy",  "Precision",  "Recall", "R-Squared")
  names <- data %>%
    distinct(tree_sp_eu) %>%
    arrange(tree_sp_eu)
  names[] <- lapply(names, as.character)

  row.names(scores) <- names %>% pull(tree_sp_eu)

  for (mod in 1:length(models)) {

    # make predictions and cast them in intervals of low, medium and high defoliation
    y_hat <-  as.data.frame( list( as.vector( unname( predict(models[[mod]],
                                                              newdata=X_test[[mod]]) ) ) ),
                             col.names = "per") %>%
                mutate(ints = (per > 0.25) +  (per > 0.45)) %>%
                pull(ints)
    # cast the actually observed defoliation values in the same classes
    y_true <- y_test[[mod]] %>%
                mutate(ints = (nbv_ratio > 0.25) + (nbv_ratio > 0.45)) %>%
                pull(ints)

    # confusion matrix
    con_M <- confusionMatrix(as.factor(y_hat), as.factor(y_true))$table

    # find accuracy, precision and recall [this code is taken from StackOverflow XX]
    scores[mod,] <- c(sum(diag(con_M)) / sum(con_M),
                      diag(con_M) / rowSums(con_M),
                      diag(con_M) / colSums(con_M),
                      models[[mod]]$rsq)
  }
  return(scores)
}

evaluate(models, X_test=splits[[3]], y_test=splits[[4]])
```

```
## Warning in levels(reference) != levels(data): longer object length is not a
## multiple of shorter object length
```

```
## Warning in confusionMatrix.default(as.factor(y_hat), as.factor(y_true)): Levels
## are not in the same order for reference and data. Refactoring data to match.
```

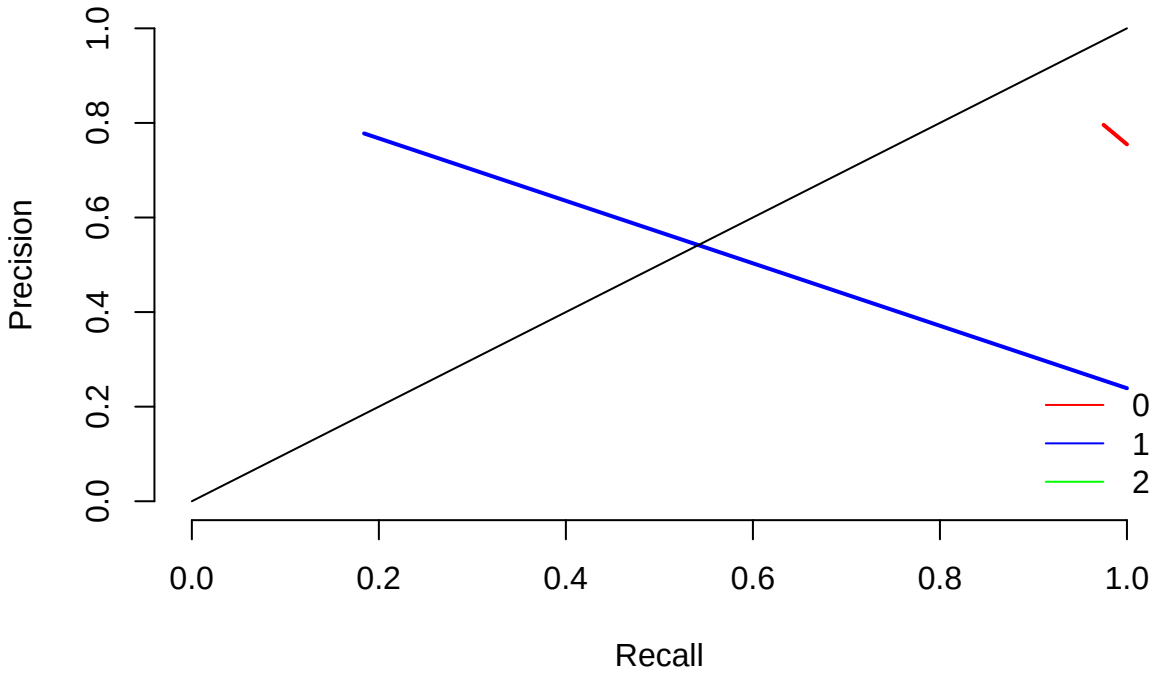
```
## Warning in matrix(value, n, p): data length [507] is not a sub-multiple or
## multiple of the number of columns [4]
```

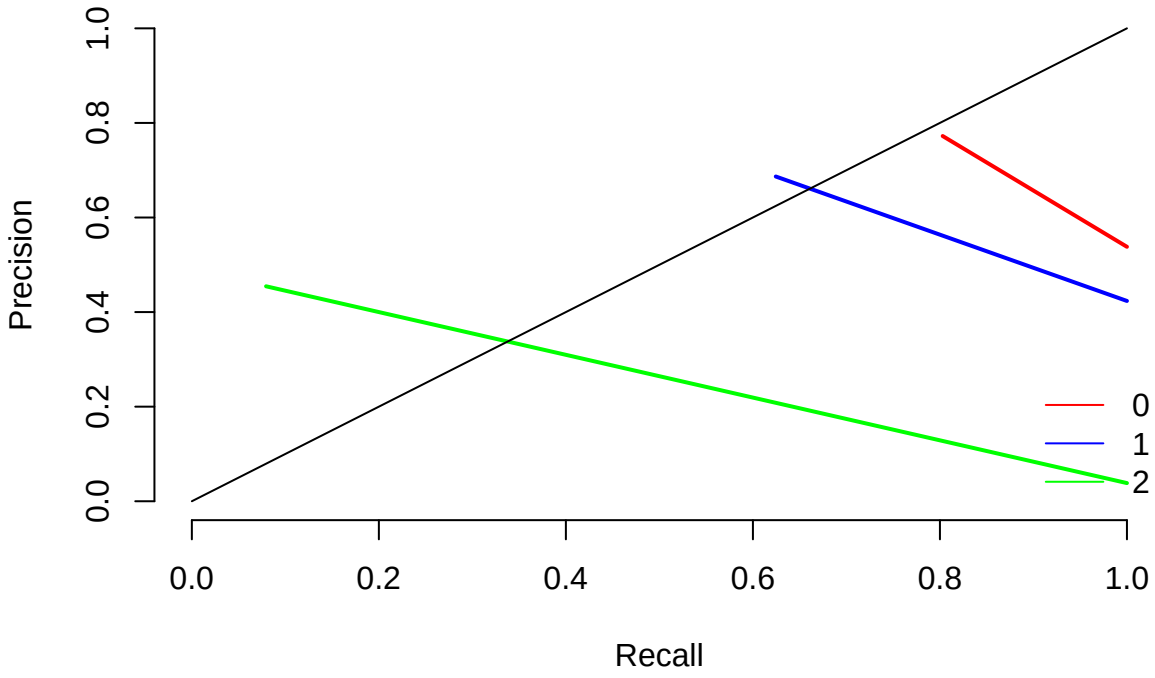
```
## Warning in matrix(value, n, p): data length [507] is not a sub-multiple or
## multiple of the number of columns [4]

## Warning in matrix(value, n, p): data length [507] is not a sub-multiple or
## multiple of the number of columns [4]

## Warning in matrix(value, n, p): data length [507] is not a sub-multiple or
## multiple of the number of columns [4]

## Warning in matrix(value, n, p): data length [507] is not a sub-multiple or
## multiple of the number of columns [4]

## Warning in matrix(value, n, p): data length [507] is not a sub-multiple or
## multiple of the number of columns [4]
```

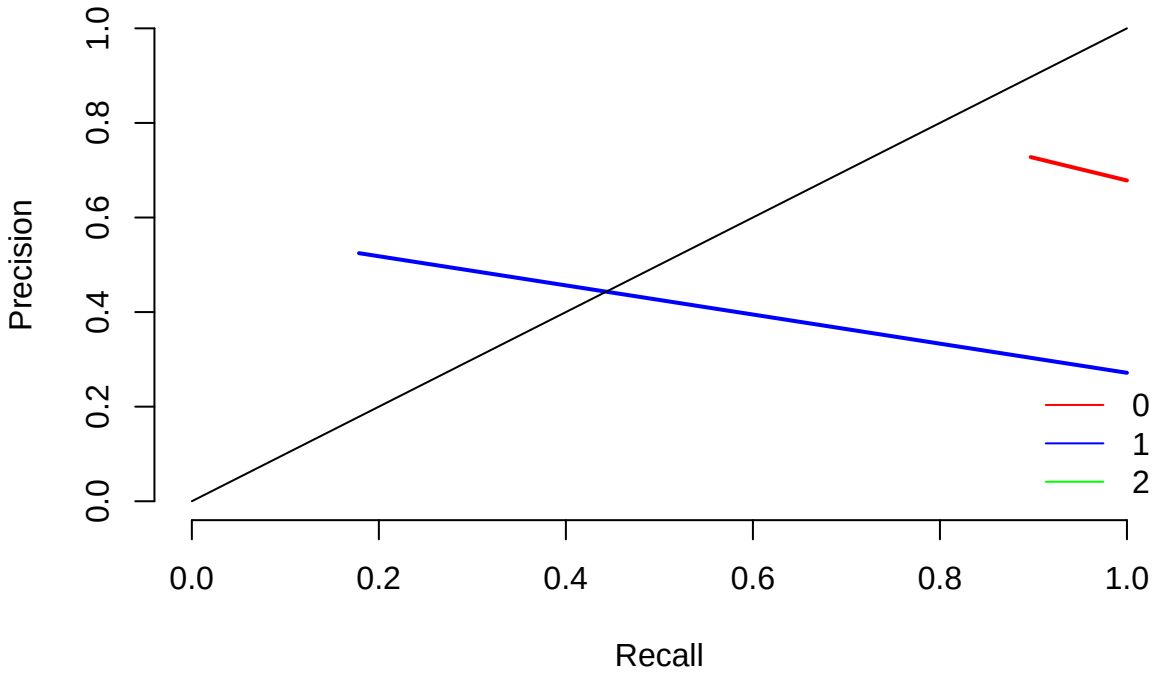
```
##      Accuracy Precision    Recall R-Squared
## Dgl 0.8176101 0.8108108 0.9090909       NaN
## Gfi 0.7294333 0.8182990 0.6495327 0.6666667
## Gki 0.6342944 0.7677725 0.3896104 0.6666667
## Rbu 0.7500000 0.8329741 0.5943775 0.8333333
## Tei 0.6662031 0.7705882 0.5718085 0.6666667
## Wta 0.6940211 0.8432203 0.6293823 0.8888889
```

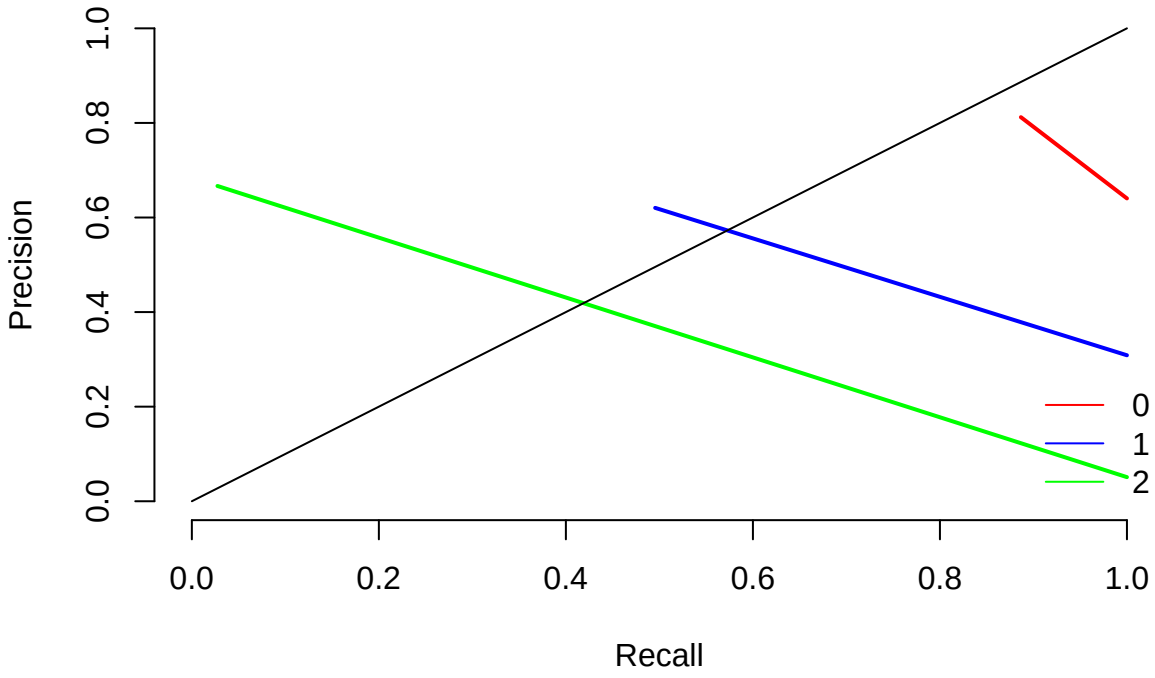
```
roc_auc <- function(splits = splits, test = 2) {
  # Evaluate the AUC scores and plot auc curve according to these sources [XX]
  #
  # Inputs:
  # - splits(data frame)   <- data used for prediction
  # - test (integer)   <- values [0, 2], indicating whether test AUC is estimated
  #
  # Outputs:
  #   - aucs (data frame)       <- data frame containing AUCS


  # store AUCs
  aucs <- matrix(NA, 6, 3)
  colnames(aucs) <- c("0 - 25 %",  "25 - 45 %",  "45 - 100 %")
  names <- data %>%
    distinct(tree_sp_eu) %>%
    arrange(tree_sp_eu)
  names[] <- lapply(names, as.character)

  row.names(aucs) <- names %>% pull(tree_sp_eu)

  for (spec in 1:6) {
    # casting both train and test response
    d_y_train = splits[[2]][[spec]] %>%

      mutate(ints = (nbv_ratio > 0.25) + (nbv_ratio > 0.45)) %>%
      pull(ints)

    d_y_test = splits[[4]][[spec]] %>%

      mutate(ints = (nbv_ratio > 0.25) + (nbv_ratio > 0.45)) %>%
```

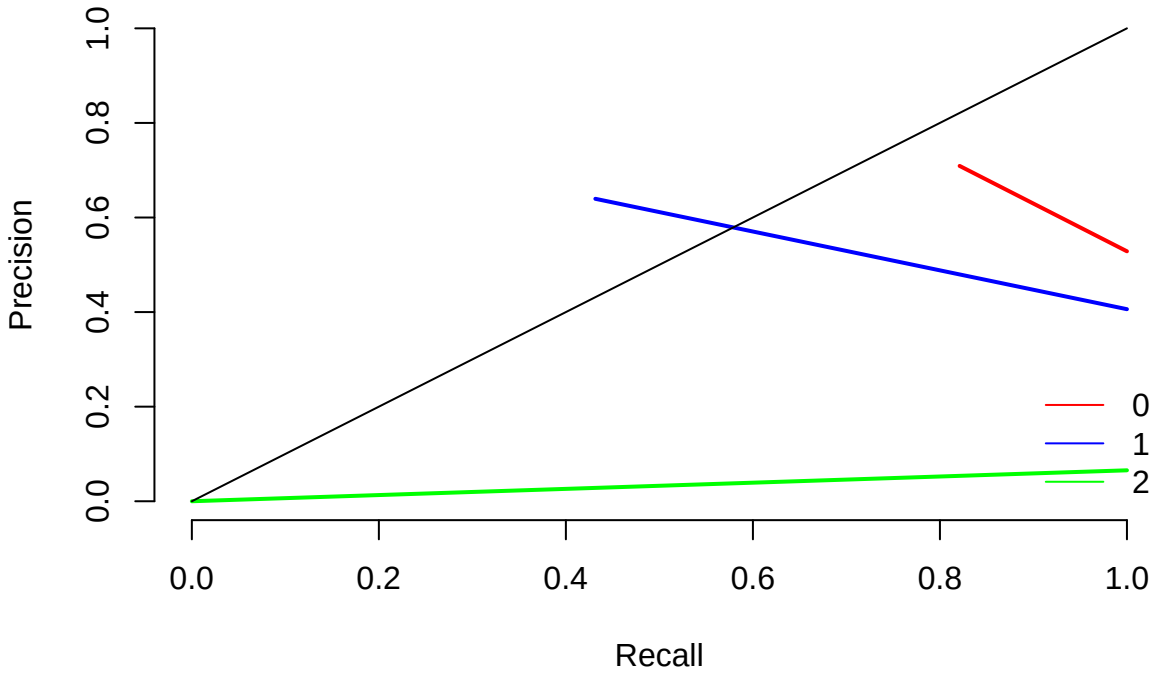
```
      pull(ints)

    # creating the plot for AUCs
    plot(x=NA, y=NA, xlim=c(0,1), ylim=c(0,1),
         ylab="Precision",
         xlab="Recall",
         bty='n')

    colors <- c("red", "blue", "green")
    y_true = as.factor(d_y_train)

    # this part of the code is taken from one of my sources (Eric Plog's Medium.com Article:
    # {https://medium.com/@plog397/auc-roc-curve-scoring-function-for-multi-class-classification-982287
    for (i in seq_along(levels(y_true))) {
      cur.class <- levels(y_true)[i]
      # binarizing classifier response
      binary.labels <- as.factor(d_y_train == cur.class)
      # fitting a new model
      model <- randomForest( as.data.frame( splits[[1]][[spec]] ), binary.labels )
      pred <- predict(model, splits[[1+test]][[spec]], type='response')
      score <- as.numeric( as.logical(pred) )# posterior for  positive class
      if (test == 2) {
        test.labels <- d_y_test == cur.class
      } else {
        test.labels <- d_y_train == cur.class
      }
      pred <- prediction(score, test.labels)
      perf <- performance(pred, "prec", "rec")
      roc.x <- unlist(perf@x.values)
      roc.y <- unlist(perf@y.values)
      lines(roc.y ~ roc.x, col = colors[i], lwd = 2)
      # store AUC
      auc <- performance(pred, "auc")
      auc <- unlist(slot(auc, "y.values"))
      aucs[spec, i] <- auc
    }
    lines(x=c(0,1), c(0,1))
    legend("bottomright", levels(y_true), lty=1,
           bty="n", col = colors)
  }
  return(aucs)
}

roc_auc(splits)
```

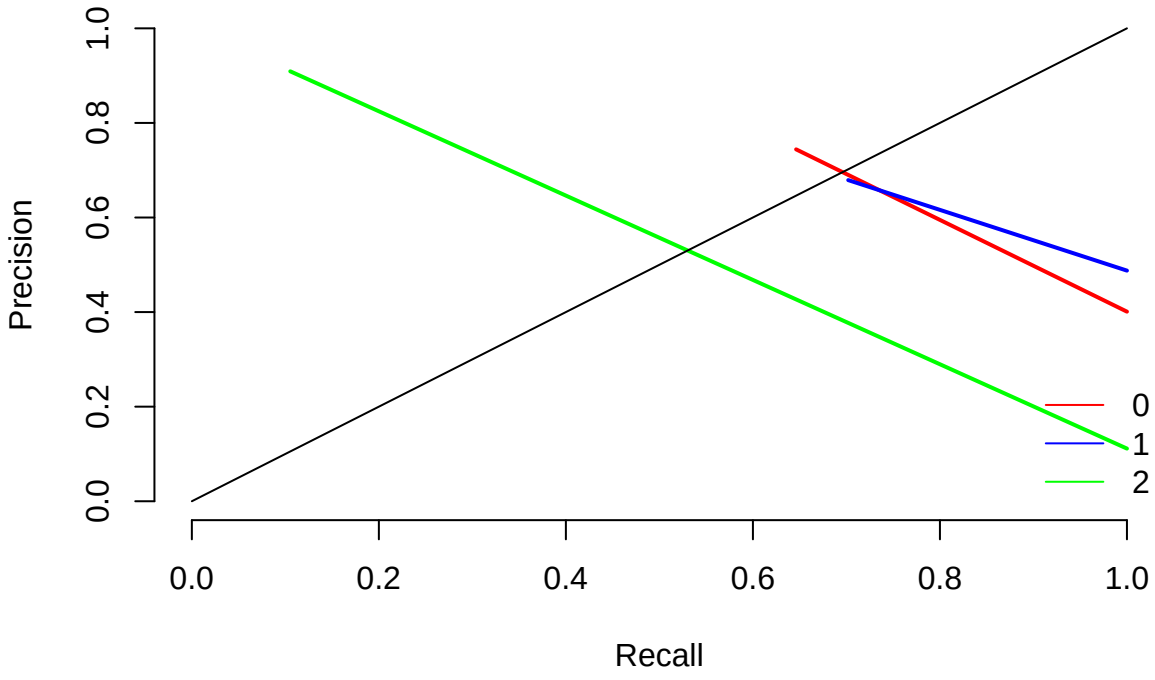

Precision
0.0
0.2
0.4
0.6
0.8
1.0
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2


Precision
0.0
0.2
0.4
0.6
0.8
1.0
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2


Precision
0.0
0.2
0.4
0.6
0.8
1.0
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2


Precision
0.0
0.2
0.4
0.6
0.8
1.0
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2


Precision
0.0
0.2
0.4
0.6
0.8
1.0
Recall
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2




```
##      0 - 25 % 25 - 45 % 45 - 100 %
## Dgl 0.6028846 0.5838408  0.5000000
## Gfi 0.7636095 0.7075791  0.5377814
## Gki 0.5947723 0.5591771  0.5000000
## Rbu 0.7607691 0.6800608  0.5133307
## Tei 0.7217358 0.6326153  0.4977679
## Wta 0.7487354 0.6930668  0.5519719
```

```r
# feature importance function
feat_imp <- function(mods = models) {
  # Feature Importance is returned beyond some threshold from passed models
  #
  # Inputs:
  # - models (list)   <- data used for prediction
  #
  # Outputs:
  #   - importance (data frame)       <- data frame containing the feature importance of each model (by

  impi = as.data.frame( round(importance(mods[[1]]), 2) )
  imp_df <- as.data.frame(impi)
  # create a data frame object of the feature importance
  for (mod in 2:6) {
    impi = as.data.frame( round(importance(mods[[mod]]), 2) )
    imp_df[,mod] <- impi
  }
  imp_df = as.data.frame(imp_df)
```

```
  #adding the name of tree species used
  names <- data %>%
    distinct(tree_sp_eu) %>%
    arrange(tree_sp_eu)
  names[] <- lapply(names, as.character)
  # adding the column names of variables
  colnames(imp_df) <- names %>% pull(tree_sp_eu)
  imp_df <- imp_df %>%
    arrange(desc(Gfi))
  return(imp_df)
}

importance = feat_imp()
```

```
# data preperation for gam
tree_species_mean <- function(data, feat_imp = importance) {
  # Data preparation for the gam model to select the most important features and remove the problematic
  #
  # Inputs:
  # - data (tibble)   <- data intended to transform
  # - feat_imp        <- feature importance from randomForest model
  # Outputs:
  #   - tree_data (tibble)       <- transformed data (by tree species)
  # temporal-spatio model
  data$spat <- factor(paste0(data$x_utm, data$y_utm))

  # splitting data into groups
  tree_species_list <- data %>%
    group_split(tree_sp_eu)

  # number of species
  tree_species = data %>%
    distinct(tree_sp_eu)

  # feature importance prep
  feat_imp <- feat_imp %>%
    rownames_to_column("feature")

  # features extracted

  # seperate the data into tree species and find unique values (averaging)
  tree_data = list(tree_species_list, tree_species)
  for (spec in 1:dim(tree_species)[1] ) {
    # getting the feature importance for the current species and extract all features with relative imp
    feat_imp_spec <- as.data.frame( feat_imp[ , c(1, spec + 1)] ) %>%
      mutate(perc = feat_imp[,spec + 1] / sum (feat_imp[,spec + 1]) ) %>%
      filter(perc > 0.01) %>%
      arrange(desc(perc))

    # inlcude the most important features
    tree_data[[spec]] <- tree_species_list[[spec]][,c("x_utm", "y_utm", "year", "spat", "nbv_ratio", fea
    # remove duplicated columns and transform problematic columns
    tree_data[[spec]] <- tree_data[[spec]][, !duplicated(colnames(tree_data[[spec]]))] %>%
      group_by(year, spat, x_utm, y_utm) %>%
```

```
      mutate(across(where(is.double), mean)) %>%
      distinct()
  }
  return(tree_data)
}
```

```
# getting the necessary data
tree_data = tree_species_mean(data)
```

```
# model is only documented for the first tree species, all other models follow the same idea
# the various tested models are excluded from this code, only the final model code is given
# tests across models (e.g. AIC, BIC etc.) are excluded from the string for the reason that only final
```

```
# assigning the model data
dat_dgl <- data.frame(tree_data[[1]])

# train - test split
train_dgl <- dat_dgl %>%
  sample_frac(0.8)

# transforming the test feature matrix
test_dgl_X <- dat_dgl %>%
  anti_join(train_dgl) %>%
  select(-c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "slope_dir", "spei_3_aug"
```

```
# transforming test response vector
test_dgl_Y <- dat_dgl %>%
  anti_join(train_dgl) %>%
  select(c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "slope_dir", "spei_3_aug"
```

```
# building gam model with defoliation untransformed as response
mod_dgl_fin <- gam(nbv_ratio ~ te(y_utm, x_utm, year, bs = c("tp","tp"), d = c(2,1),
                                  k = c(25,20))
                   # cubic regresion smoothers for covariates
                   + s(tree_age, bs="cr", k=30) + s(H_bhd, bs="cr", k=10) + s(H_spec, bs="cr", k=10)
                   + slope_dir + soil_no,
                   data=train_dgl,
                   # correlation structure of the resiudals in accordance with EIckenscheidt MOdel
                   correlation = corARMA(form =~ year | spat, p=1, q=1),
                   # assuming normality and inducing logit-link function
                   family = gaussian(link="logit"),
                   # weighting by number of trees per location
                   weights = train_dgl$n_trees,
                   # method used is REML as suggested in Wood [2007]
                   method="REML")
```

```
## Warning in newton(lsp = lsp, X = G$X, y = G$y, Eb = G$Eb, UrS = G$UrS, L =
## G$L, : Fitting terminated with step failure - check results carefully
```

```
# predicting on test data with final model
y_pred <- predict(mod_dgl_fin, newdata=test_dgl_X, type = "response")
test_dgl_Y <- test_dgl_Y$nbv_ratio[!is.na(y_pred)]; y_pred <- y_pred[!is.na(y_pred)]
```

```r
# exmaining mean-variance relationship
e <- residuals(mod_dgl_fin); fv <- fitted(mod_dgl_fin)
lm(log(e^2) ~ log(fv))
```

```
##
## Call:
## lm(formula = log(e^2) ~ log(fv))
##
## Coefficients:
## (Intercept)      log(fv)
##     -3.8058       0.9943
```

```r
# examining the fitted mean, according to Wood [XX]
mean(dat_dgl$nbv_ratio); mean(fitted(mod_dgl_fin)^(1/0.65))
```

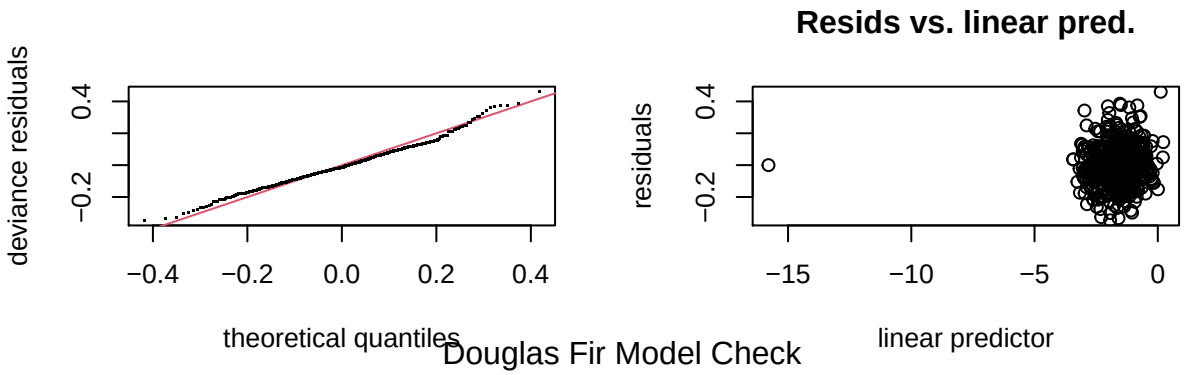
```
## [1] 0.1630531
```

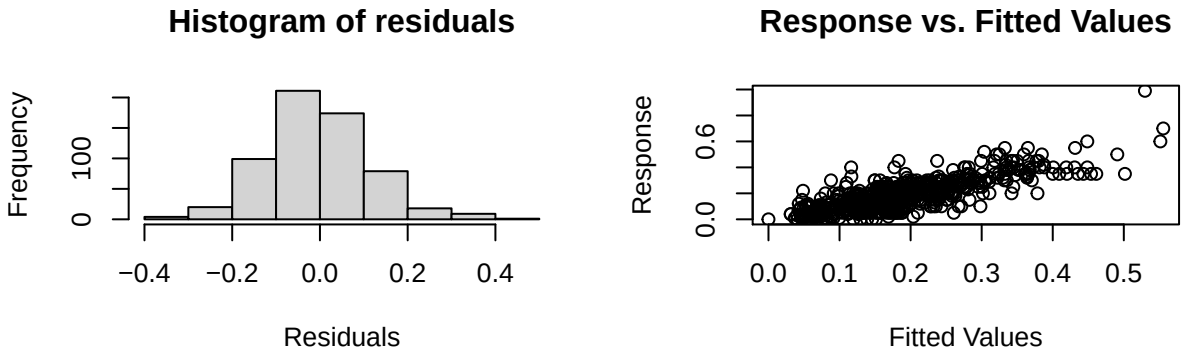
```
## [1] 0.08339602
```

```r
# Checking the gam model for residual patterns
par(mfrow = c(2, 2))
gam.check(mod_dgl_fin)
```

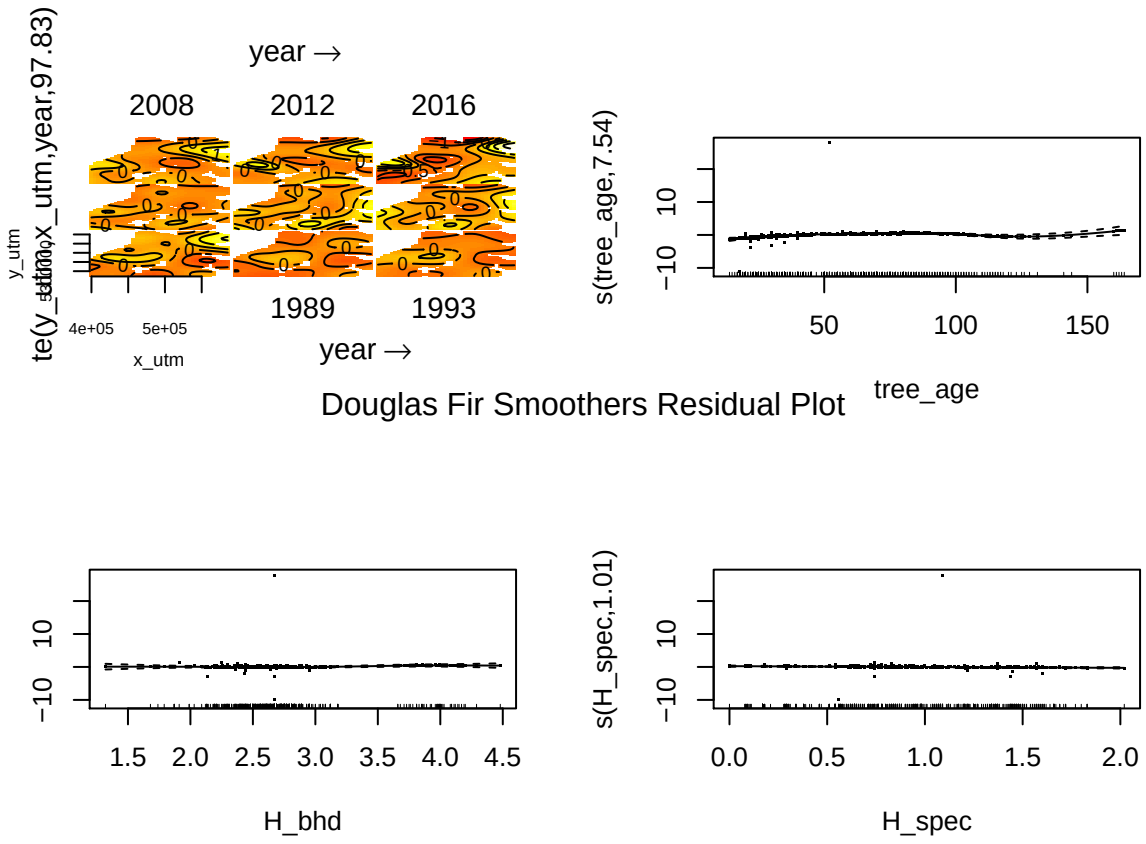
```
##
## Method: REML   Optimizer: outer newton
## step failed after 12 iterations.
## Gradient range [-5.643246,1.087005]
## (score -592.3298 & scale 0.01758183).
## Hessian positive definite, eigenvalue range [0.006308838,303.1794].
## Model rank =  566 / 566
##
## Basis dimension (k) checking results. 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(y_utm,x_utm,year) 499.00  97.83    1.03    0.80
## s(tree_age)           29.00   7.54    1.00    0.46
## s(H_bhd)               9.00   3.33    0.99    0.41
## s(H_spec)              9.00   1.01    1.02    0.74
```

```r
mtext("Douglas Fir Model Check", side = 3, line = -13, outer = TRUE)
```

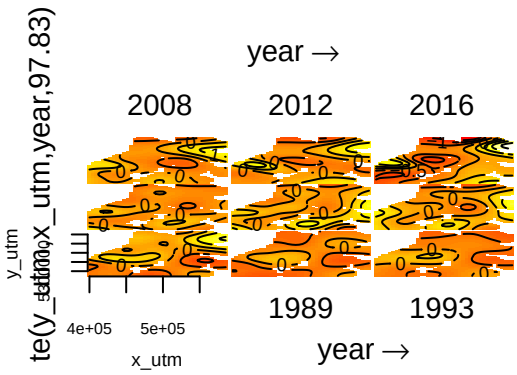





```
# plotting the models location smoothers
par(mfrow=c(2, 2))
plot(mod_dgl_fin, residuals = T)
mtext("Douglas Fir Smoothers Residual Plot", side = 3, line = -13, outer = TRUE)
```




```
# plotting the map
plot(mod_dgl_fin, select = 1)
mtext("Spruce Spatio-Temporal Smoother", side = 3, line = -22, outer = TRUE)

par(mfrow=c(2, 2))
```



Spruce Spatio−Temporal Smoother

```
plot(mod_dgl_fin, select = 2)
plot(mod_dgl_fin, select = 3)
plot(mod_dgl_fin, select = 4)
mtext("Douglas fir Smoothers", side = 3, line = -13, outer = TRUE)
# summary of the model
summary(mod_dgl_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(25, 20)) + s(tree_age, bs = "cr", k = 30) + s(H_bhd,
##     bs = "cr", k = 10) + s(H_spec, bs = "cr", k = 10) + slope_dir +
##     soil_no
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.246e-01  4.279e-01   0.525 0.599961
## slope_dir1  -1.582e+00  3.858e-01  -4.100 4.84e-05 ***
## slope_dir2  -1.896e+00  3.947e-01  -4.803 2.09e-06 ***
## slope_dir3  -1.772e+00  3.900e-01  -4.544 6.96e-06 ***
## slope_dir4  -1.735e+00  3.755e-01  -4.620 4.93e-06 ***
## slope_dir5  -1.499e+00  4.101e-01  -3.656 0.000284 ***
## slope_dir6  -1.424e+00  3.887e-01  -3.664 0.000276 ***
```

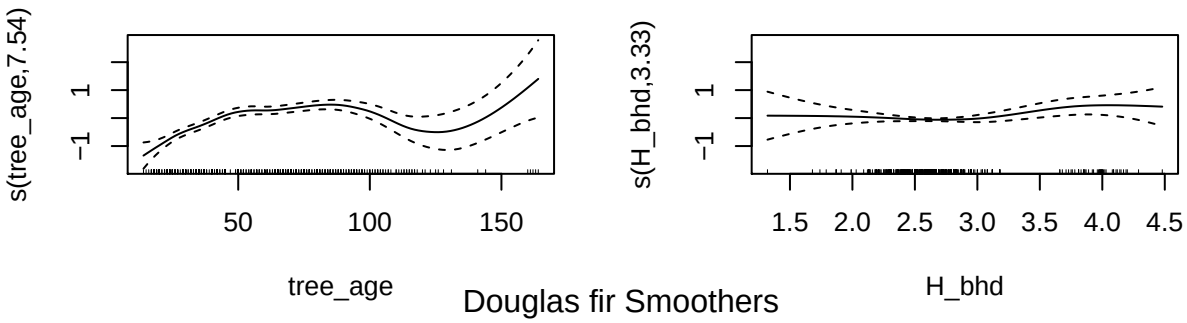
```
## slope_dir7  -1.552e+00  3.966e-01  -3.914 0.000104 ***
## slope_dir8  -1.733e+00  3.901e-01  -4.443 1.10e-05 ***
## slope_dir10 -1.772e+00  4.019e-01  -4.408 1.29e-05 ***
## soil_no2    -2.704e-02  1.913e-01  -0.141 0.887684
## soil_no3    -1.032e-02  1.877e-01  -0.055 0.956182
## soil_no4    -3.508e-02  1.608e-01  -0.218 0.827386
## soil_no5    -5.589e-01  1.837e-01  -3.042 0.002480 **
## soil_no6    -1.813e-01  1.697e-01  -1.068 0.286017
## soil_no7    -3.072e-01  2.196e-01  -1.399 0.162480
## soil_no8    -1.285e-01  5.266e-01  -0.244 0.807301
## soil_no9    -4.129e-02  2.534e-01  -0.163 0.870615
## soil_no10   -1.380e+01  5.822e+05   0.000 0.999981
## soil_no12   -1.384e+00  1.355e+00  -1.021 0.307549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf  Ref.df      F p-value
## te(y_utm,x_utm,year) 97.831 124.984  4.644 < 2e-16 ***
## s(tree_age)           7.536   9.170 14.416 < 2e-16 ***
## s(H_bhd)              3.329   4.120  2.305 0.06129 .
## s(H_spec)             1.009   1.017  7.851 0.00503 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.681   Deviance explained =   75%
## -REML = -592.33  Scale est. = 0.017582  n = 615
```

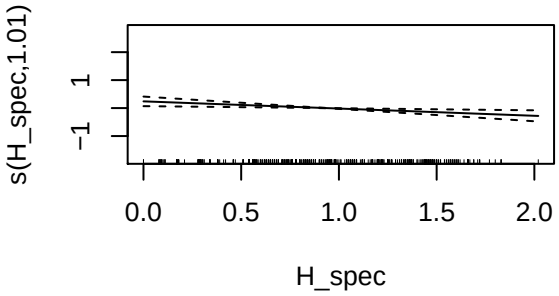
```
anova(mod_dgl_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(25, 20)) + s(tree_age, bs = "cr", k = 30) + s(H_bhd,
##     bs = "cr", k = 10) + s(H_spec, bs = "cr", k = 10) + slope_dir +
##     soil_no
##
## Parametric Terms:
##           df     F  p-value
## slope_dir  9 4.673 5.86e-06
## soil_no    1 0.000        1
##
## Approximate significance of smooth terms:
##                           edf  Ref.df      F p-value
## te(y_utm,x_utm,year)  97.831 124.984  4.644 < 2e-16
## s(tree_age)            7.536   9.170 14.416 < 2e-16
## s(H_bhd)               3.329   4.120  2.305 0.06129
## s(H_spec)              1.009   1.017  7.851 0.00503
```

```
# computing residual mean squared error from the data above
rmse(test_dgl_Y, y_pred)
```

```
## [1] 0.08843152
```






```
dat_gfi <- data.frame(tree_data[[2]])


train_gfi <- dat_gfi %>%
  sample_frac(0.8)

test_gfi_X <- dat_gfi %>%
  anti_join(train_gfi) %>%
  select(-c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "slope_dir", "H_bhd", "geo
```

```
test_gfi_Y <- dat_gfi %>%
  anti_join(train_gfi) %>%
  select(c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "slope_dir", "H_bhd", "geo
```

```
mod_gfi_fin <- gam(nbv_ratio ~ te(y_utm, x_utm, year, bs = c("tp","tp"), d = c(2,1),
                                  k = c(45,25))
                   + s(tree_age, bs="cr", k=20) + s(H_bhd, bs="cr", k=20)
                   + s(ac_tot_wd, bs="cr", k=20)
                   + slope_dir + geol_no

                   ,
                   data=train_gfi,
```

```
                    correlation = corARMA(form =~ year | spat, p=1, q=1),
                    family = gaussian(link="logit"),
                    weights = train_gfi$n_trees,
                    method="REML")

y_pred <- predict(mod_gfi_fin, newdata=test_gfi_X, type = "response")
test_gfi_Y <- test_gfi_Y$nbv_ratio[!is.na(y_pred)]; y_pred <- y_pred[!is.na(y_pred)]

e <- residuals(mod_gfi_fin); fv <- fitted(mod_gfi_fin)
lm(log(e^2) ~ log(fv))
```

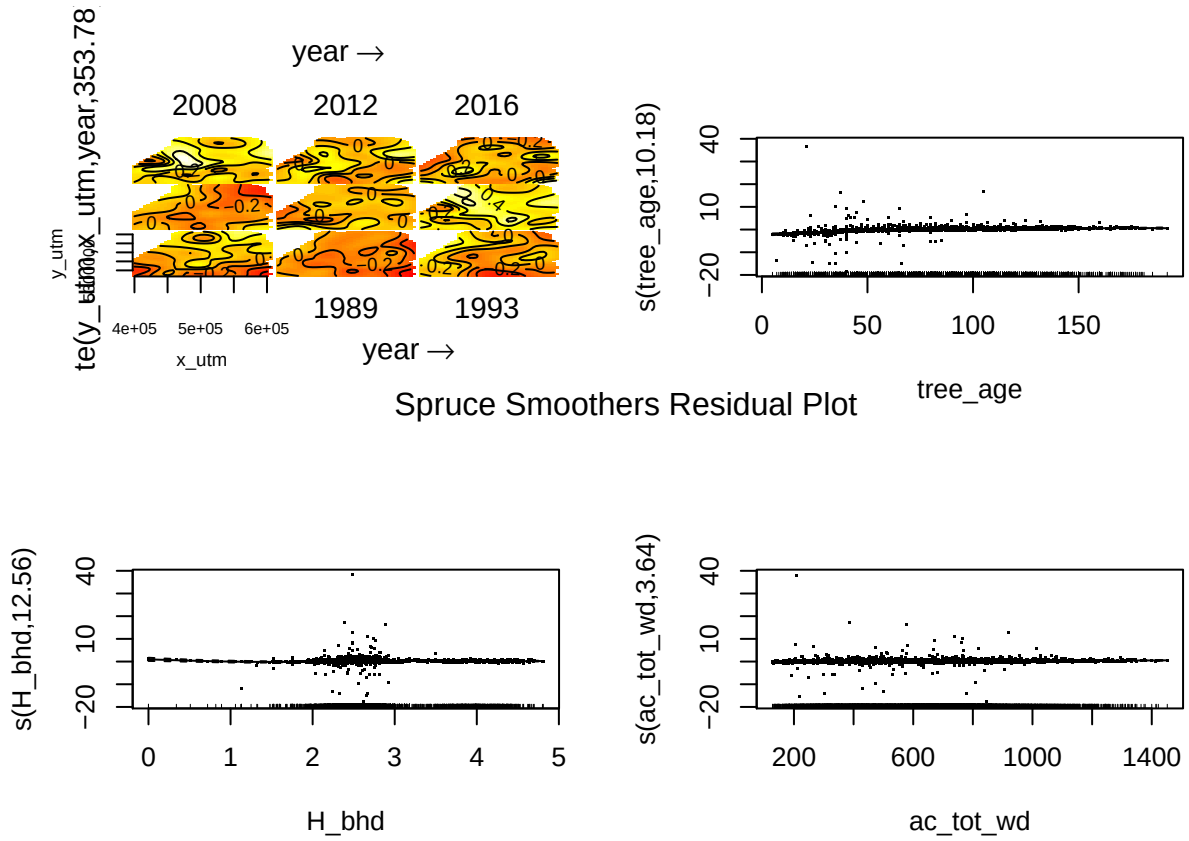
```
##
## Call:
## lm(formula = log(e^2) ~ log(fv))
##
## Coefficients:
## (Intercept)      log(fv)
##     -3.7402       0.3201
```

```
mean(dat_gfi$nbv_ratio); mean(fitted(mod_gfi_fin)^(1/0.65))
```

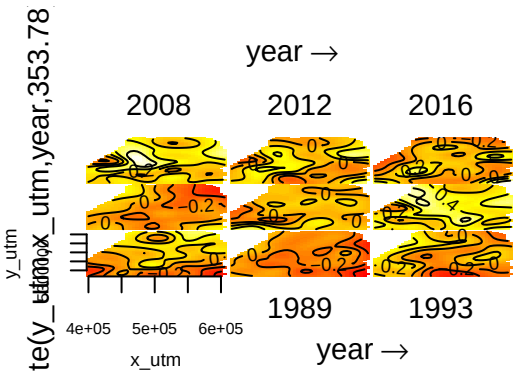
```
## [1] 0.2337024
```

```
## [1] 0.122577
```

```
par(mfrow=c(2, 2))
plot(mod_gfi_fin, residuals = T)
mtext("Spruce Smoothers Residual Plot", side = 3, line = -13, outer = TRUE)
```

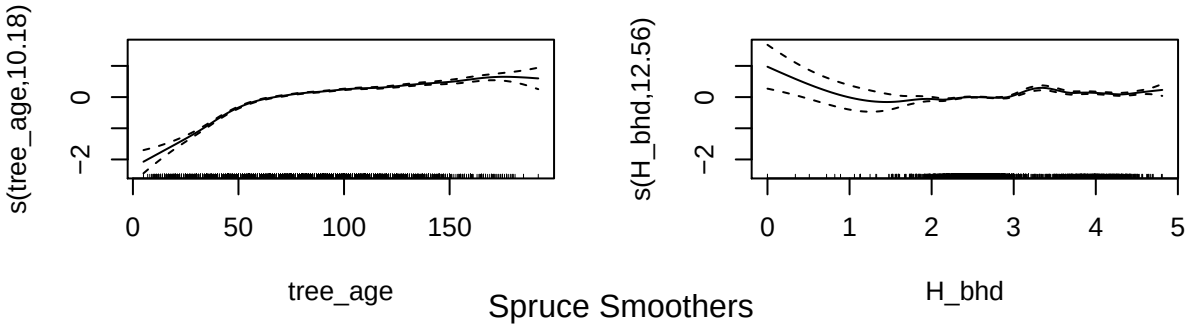


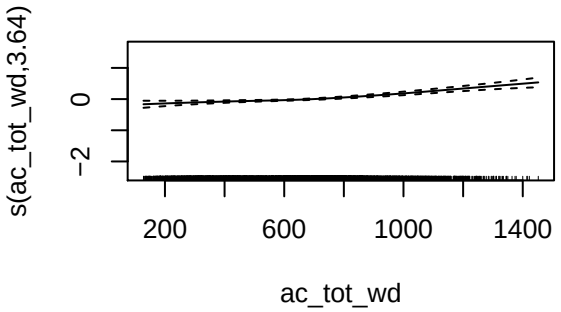

```
plot(mod_gfi_fin, select = 1)
mtext("Spruce Spatio-Temporal Smoother", side = 3, line = -22, outer = TRUE)

par(mfrow=c(2, 2))
```



Spruce Spatio−Temporal Smoother

```
plot(mod_gfi_fin, select = 2)
plot(mod_gfi_fin, select = 3)
plot(mod_gfi_fin, select = 4)
mtext("Spruce Smoothers", side = 3, line = -13, outer = TRUE)

par(mfrow = c(2, 2))
```

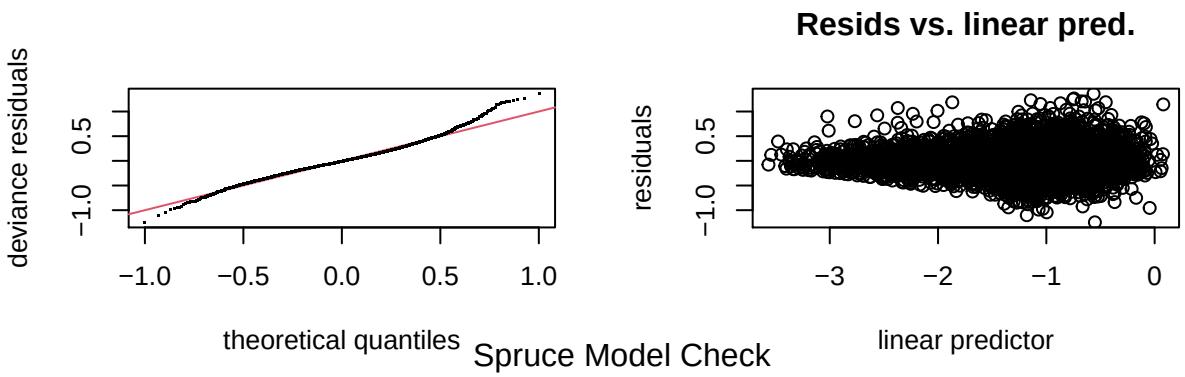


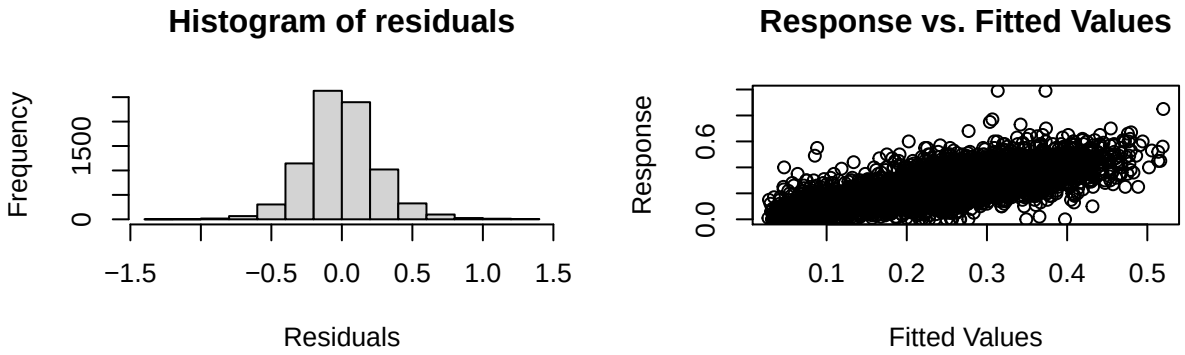



```
gam.check(mod_gfi_fin)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-0.0002098543,3.431686e-05]
## (score -8783.671 & scale 0.06821472).
## Hessian positive definite, eigenvalue range [0.9579816,4014.818].
## Model rank =  1199 / 1199
##
## 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(y_utm,x_utm,year) 1124.00  353.78    0.95  <2e-16 ***
## s(tree_age)            19.00   10.18    0.98   0.045 *
## s(H_bhd)               19.00   12.56    1.00   0.580
## s(ac_tot_wd)           19.00    3.64    1.03   0.985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mtext("Spruce Model Check", side = 3, line = -13, outer = TRUE)
```






```
summary(mod_gfi_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(45, 25)) + s(tree_age, bs = "cr", k = 20) + s(H_bhd,
##     bs = "cr", k = 20) + s(ac_tot_wd, bs = "cr", k = 20) + slope_dir +
##     geol_no
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.147309   0.031137 -36.847  < 2e-16 ***
## slope_dir1   0.033935   0.024924   1.362  0.17338
## slope_dir2   0.038624   0.025858   1.494  0.13529
## slope_dir3  -0.040204   0.029166  -1.378  0.16810
## slope_dir4  -0.022046   0.027553  -0.800  0.42365
## slope_dir5  -0.008842   0.028010  -0.316  0.75227
## slope_dir6   0.150092   0.027898   5.380 7.67e-08 ***
## slope_dir7   0.102495   0.025796   3.973 7.16e-05 ***
## slope_dir8   0.063506   0.024622   2.579  0.00992 **
## slope_dir10  0.043906   0.025104   1.749  0.08034 .
## geol_no20    0.040075   0.020759   1.930  0.05359 .
## geol_no30   -0.130604   0.058949  -2.216  0.02675 *
```

```
## geol_no40   -0.028903   0.024303  -1.189  0.23438
## geol_no50   -0.095703   0.035818  -2.672  0.00756 **
## geol_no60   -0.221940   0.039589  -5.606 2.14e-08 ***
## geol_no70    0.041504   0.047369   0.876  0.38096
## geol_no80   -0.061934   0.043390  -1.427  0.15352
## geol_no90   -0.159881   0.040162  -3.981 6.93e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf  Ref.df       F  p-value
## te(y_utm,x_utm,year) 353.780 463.534   5.935  < 2e-16 ***
## s(tree_age)           10.175  12.177 211.811  < 2e-16 ***
## s(H_bhd)              12.563  14.715  10.083  < 2e-16 ***
## s(ac_tot_wd)           3.636   4.687  15.043 1.06e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.613   Deviance explained = 63.2%
## -REML = -8783.7  Scale est. = 0.068215  n = 8048
```

```
anova(mod_gfi_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(45, 25)) + s(tree_age, bs = "cr", k = 20) + s(H_bhd,
##     bs = "cr", k = 20) + s(ac_tot_wd, bs = "cr", k = 20) + slope_dir +
##     geol_no
##
## Parametric Terms:
##           df     F p-value
## slope_dir  9 12.78  <2e-16
## geol_no    8 13.59  <2e-16
##
## Approximate significance of smooth terms:
##                           edf  Ref.df       F  p-value
## te(y_utm,x_utm,year) 353.780 463.534   5.935  < 2e-16
## s(tree_age)           10.175  12.177 211.811  < 2e-16
## s(H_bhd)              12.563  14.715  10.083  < 2e-16
## s(ac_tot_wd)           3.636   4.687  15.043 1.06e-13
```

```
rmse(test_gfi_Y, y_pred)
```

```
## [1] 0.08381654
```

```
dat_gki <- data.frame(tree_data[[3]])

train_gki <- dat_gki %>%
  sample_frac(0.8)

test_gki_X <- dat_gki %>%
  anti_join(train_gki) %>%
```

```
  select(-c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "slope_dir", "depth_mm",
```

```
test_gki_Y <- dat_gki %>%
  anti_join(train_gki) %>%
  select(c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "slope_dir", "depth_mm",
```

```
mod_gki_fin <- gam(nbv_ratio^0.65 ~ te(y_utm, x_utm, year, bs = c("tp","tp"), d = c(2,1),
                                       k = c(45,25))
                   + s(tree_age, bs="cr", k=40) + s(globrad_y_lag1, bs="cr", k=40)+ s(s_vals, bs="cr",
                   + s(ac_tot_wd, bs="cr", k=40)
                   + s(tpi750, bs="cr", k=100)
                   + slope_dir + soil_ty_no + depth_mm,
                   data=train_gki,
                   correlation = corARMA(form =~ year | spat, p=1, q=1),
                   family = gaussian(link="logit"),
                   weights = train_gki$n_trees,
                   method="REML")

y_pred <- predict(mod_gki_fin, newdata=test_gki_X, type = "response")
test_gki_Y <- test_gki_Y$nbv_ratio[!is.na(y_pred)]; y_pred <- y_pred[!is.na(y_pred)]^(1/0.65)

e <- residuals(mod_gki_fin); fv <- fitted(mod_gki_fin)
lm(log(e^2) ~ log(fv))
```

```
##
## Call:
## lm(formula = log(e^2) ~ log(fv))
##
## Coefficients:
## (Intercept)      log(fv)
##     -4.4985       0.2961
```

```
mean(dat_gki$nbv_ratio); mean(fitted(mod_gki_fin)^(1/0.65))
```

```
## [1] 0.2355777
```

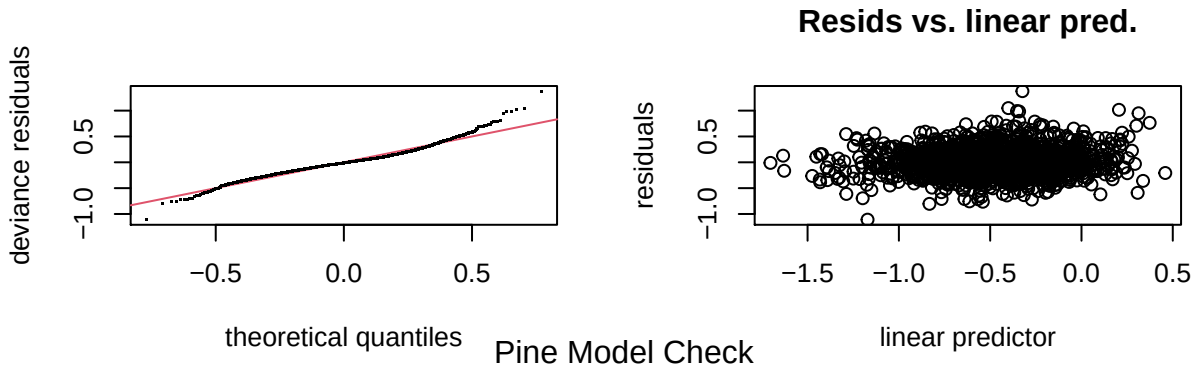
```
## [1] 0.2315507
```

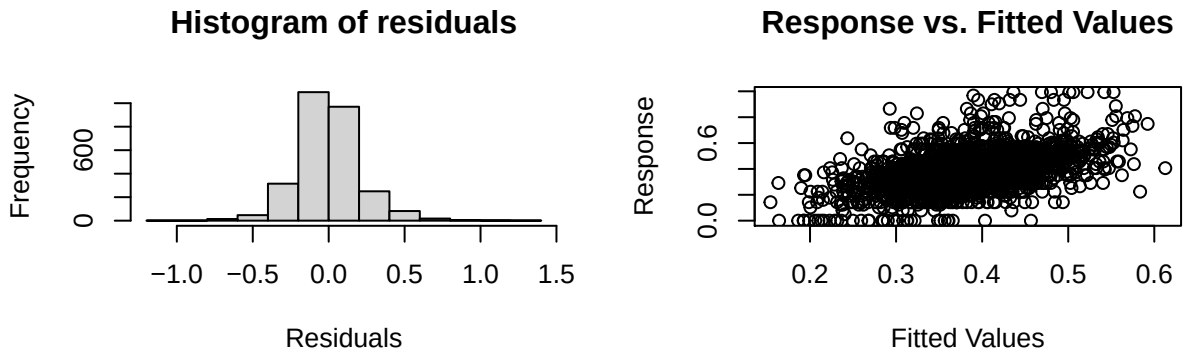
```
par(mfrow = c(2, 2))
gam.check(mod_gki_fin)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 14 iterations.
## Gradient range [-0.0004113996,6.347977e-05]
## (score -1847.847 & scale 0.04655541).
## Hessian positive definite, eigenvalue range [0.0004111729,1386.306].
## Model rank =  1440 / 1440
##
## 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(y_utm,x_utm,year) 1124.00  162.50    0.98   0.065 .
## s(tree_age)            39.00    7.94    1.02   0.910
## s(globrad_y_lag1)      39.00    1.52    0.97   0.035 *
## s(s_vals)              79.00    1.00    0.85  <2e-16 ***
## s(ac_tot_wd)           39.00    5.41    0.97   0.045 *
## s(tpi750)              99.00   16.50    0.91  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mtext("Pine Model Check", side = 3, line = -13, outer = TRUE)
```






```
summary(mod_gki_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio^0.65 ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(45, 25)) + s(tree_age, bs = "cr", k = 40) + s(globrad_y_lag1,
##     bs = "cr", k = 40) + s(s_vals, bs = "cr", k = 80) + s(ac_tot_wd,
##     bs = "cr", k = 40) + s(tpi750, bs = "cr", k = 100) + slope_dir +
##     soil_ty_no + depth_mm
##
## Parametric coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.996e-02  9.028e-02  -0.664 0.506674
```

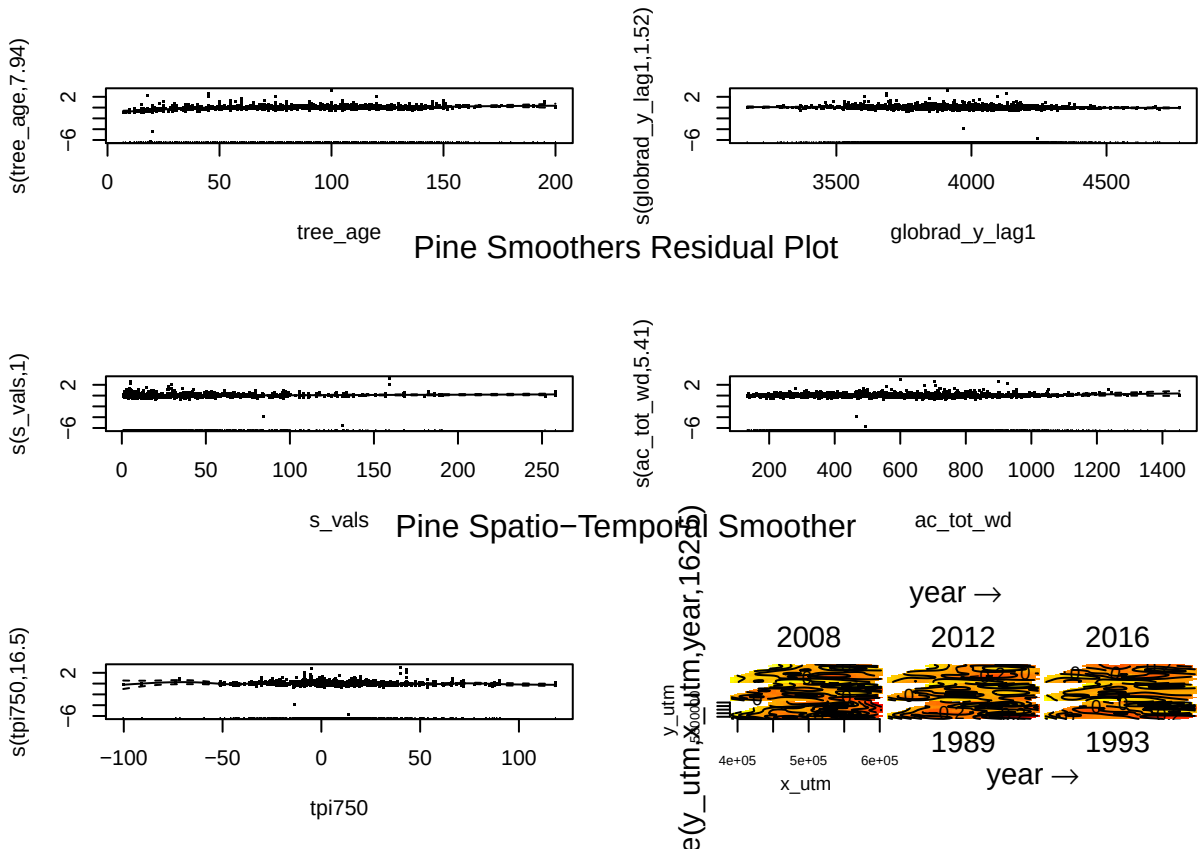
```
## slope_dir1   -2.843e-01  4.146e-02  -6.858 8.70e-12 ***
## slope_dir2   -2.477e-01  5.417e-02  -4.573 5.04e-06 ***
## slope_dir3   -1.598e-01  6.366e-02  -2.510 0.012140 *
## slope_dir4   -1.714e-01  4.753e-02  -3.607 0.000315 ***
## slope_dir5   -2.190e-01  4.332e-02  -5.055 4.61e-07 ***
## slope_dir6   -2.116e-01  4.137e-02  -5.114 3.38e-07 ***
## slope_dir7   -1.770e-01  4.498e-02  -3.936 8.51e-05 ***
## slope_dir8   -1.516e-01  4.280e-02  -3.542 0.000404 ***
## slope_dir10  -2.480e-01  3.468e-02  -7.150 1.12e-12 ***
## soil_ty_no3  -3.918e-01  8.853e-02  -4.426 1.00e-05 ***
## soil_ty_no4  -1.123e-01  6.815e-02  -1.647 0.099634 .
## soil_ty_no5  -1.791e-01  1.136e-01  -1.576 0.115143
## soil_ty_no6  -7.328e-02  1.189e-01  -0.616 0.537774
## soil_ty_no7  -9.108e-02  7.465e-02  -1.220 0.222555
## soil_ty_no8  -6.809e-02  7.902e-02  -0.862 0.388943
## soil_ty_no9  -5.130e-02  7.794e-02  -0.658 0.510478
## soil_ty_no10  5.526e-02  9.272e-02   0.596 0.551222
## soil_ty_no11 -4.277e-02  7.781e-02  -0.550 0.582544
## soil_ty_no12 -7.587e-02  8.799e-02  -0.862 0.388605
## depth_mm     -1.798e-04  5.292e-05  -3.398 0.000690 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf  Ref.df      F  p-value
## te(y_utm,x_utm,year) 162.504 213.076  3.816  < 2e-16 ***
## s(tree_age)            7.938   9.849 16.908  < 2e-16 ***
## s(globrad_y_lag1)      1.521   1.851  2.265  0.12552
## s(s_vals)              1.001   1.002  8.980  0.00274 **
## s(ac_tot_wd)           5.408   6.894  2.481  0.01551 *
## s(tpi750)             16.505  19.716  3.163 3.22e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.396   Deviance explained = 44.5%
## -REML = -1847.8  Scale est. = 0.046555  n = 2799
```

```
anova(mod_gki_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio^0.65 ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(45, 25)) + s(tree_age, bs = "cr", k = 40) + s(globrad_y_lag1,
##     bs = "cr", k = 40) + s(s_vals, bs = "cr", k = 80) + s(ac_tot_wd,
##     bs = "cr", k = 40) + s(tpi750, bs = "cr", k = 100) + slope_dir +
##     soil_ty_no + depth_mm
##
## Parametric Terms:
##            df      F  p-value
## slope_dir   9  7.969 1.04e-11
## soil_ty_no 10  3.218 0.000395
## depth_mm    1 11.545 0.000690
```

```
##
## Approximate significance of smooth terms:
##                            edf  Ref.df      F  p-value
## te(y_utm,x_utm,year) 162.504 213.076  3.816  < 2e-16
## s(tree_age)            7.938   9.849 16.908  < 2e-16
## s(globrad_y_lag1)      1.521   1.851  2.265  0.12552
## s(s_vals)              1.001   1.002  8.980  0.00274
## s(ac_tot_wd)           5.408   6.894  2.481  0.01551
## s(tpi750)             16.505  19.716  3.163 3.22e-06
```

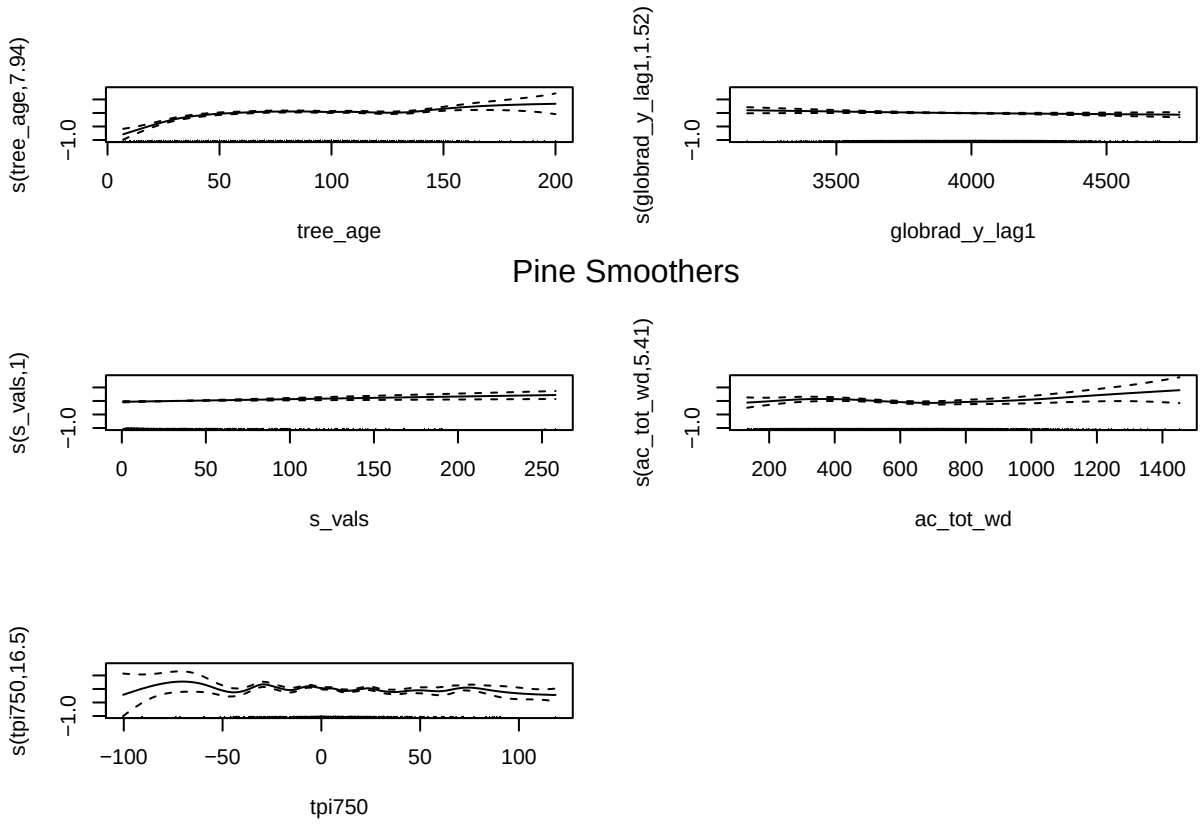
```r
par(mfrow=c(3, 2))
plot(mod_gki_fin, select = 2, residuals = T)
plot(mod_gki_fin, select = 3, residuals = T)
plot(mod_gki_fin, select = 4, residuals = T)
plot(mod_gki_fin, select = 5, residuals = T)
plot(mod_gki_fin, select = 6, residuals = T)
mtext("Pine Smoothers Residual Plot", side = 3, line = -11, outer = TRUE)

plot(mod_gki_fin, select = 1)
mtext("Pine Spatio-Temporal Smoother", side = 3, line = -22, outer = TRUE)
```




```r
par(mfrow=c(3, 2))
plot(mod_gki_fin, select = 2)
plot(mod_gki_fin, select = 3)
plot(mod_gki_fin, select = 4)
plot(mod_gki_fin, select = 5)
plot(mod_gki_fin, select = 6)
```

```
mtext("Pine Smoothers", side = 3, line = -12, outer = TRUE)

rmse(test_gki_Y, y_pred)
```

```
## [1] 0.1078339
```




```
dat_rbu <- data.frame(tree_data[[4]])


train_rbu <- dat_rbu %>%
  sample_frac(0.8)

test_rbu_X <- dat_rbu %>%
  anti_join(train_rbu) %>%
  select(-c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "ac_tot_wd", "n_tot_wd",
```

```
test_rbu_Y <- dat_rbu %>%
  anti_join(train_rbu) %>%
  select(c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "ac_tot_wd", "n_tot_wd",
```

```
mod_rbu_fin <- gam(nbv_ratio^0.75 ~ te(y_utm, x_utm, year, bs = c("tp","tp"), d = c(2,1),
                                       k = c(40,25))
                   + s(tree_age, bs="cr", k=20)
                   + s(ac_tot_wd, k = 10) + s(n_tot_wd, k = 10)
```

```
                    + H_bhd + s(globrad_y_lag1, bs="cr", k=10) + H_spec + prec_y
                    + prec_y_lag1
                    + slope_dir + soil_no,
                    data=train_rbu,
                    correlation = corARMA(form =~ year | spat, p=1, q=1),
                    family = gaussian(link="logit"),
                    weights = train_rbu$n_trees,
                    method="REML")

y_pred <- predict(mod_rbu_fin, newdata=test_rbu_X, type = "response")
test_rbu_Y <- test_rbu_Y$nbv_ratio[!is.na(y_pred)]; y_pred <- y_pred[!is.na(y_pred)]^(1/0.75)

e <- residuals(mod_rbu_fin); fv <- fitted(mod_rbu_fin)
lm(log(e^2) ~ log(fv))
```

```
##
## Call:
## lm(formula = log(e^2) ~ log(fv))
##
## Coefficients:
## (Intercept)      log(fv)
##     -3.8550       0.7269
```

```
mean(dat_rbu$nbv_ratio); mean(fitted(mod_rbu_fin)^(1/0.75))
```

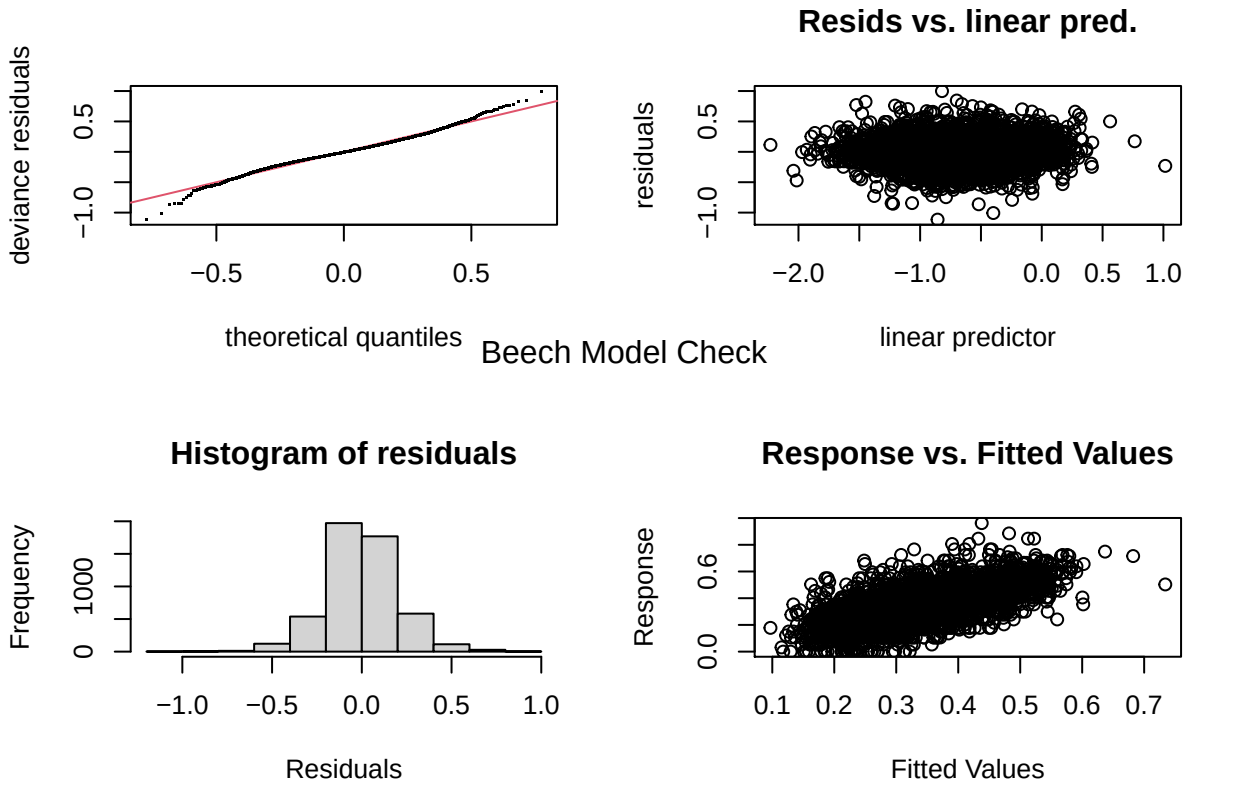
```
## [1] 0.2254479
```

```
## [1] 0.2317904
```

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

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-6.75899e-05,2.197634e-05]
## (score -4403.959 & scale 0.04316361).
## Hessian positive definite, eigenvalue range [6.758318e-05,2561.207].
## Model rank =  1070 / 1070
##
## 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(y_utm,x_utm,year) 999.00 344.33    0.98    0.05 *
## s(tree_age)           19.00   8.48    0.98    0.14
## s(ac_tot_wd)           9.00   4.31    1.01    0.75
## s(n_tot_wd)            9.00   1.00    1.00    0.46
## s(globrad_y_lag1)      9.00   5.62    0.99    0.32
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mtext("Beech Model Check", side = 3, line = -13, outer = TRUE)
```




```
summary(mod_rbu_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio^0.75 ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(40, 25)) + s(tree_age, bs = "cr", k = 20) + s(ac_tot_wd,
##     k = 10) + s(n_tot_wd, k = 10) + H_bhd + s(globrad_y_lag1,
##     bs = "cr", k = 10) + H_spec + prec_y + prec_y_lag1 + slope_dir +
##     soil_no
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.087e+00  9.548e-02 -11.384  < 2e-16 ***
## H_bhd        1.357e-01  1.548e-02   8.770  < 2e-16 ***
## H_spec      -4.151e-02  1.469e-02  -2.825 0.004747 **
## prec_y       2.298e-04  4.758e-05   4.830 1.41e-06 ***
## prec_y_lag1 -2.624e-05  3.538e-05  -0.742 0.458390
## slope_dir1   4.480e-02  4.610e-02   0.972 0.331224
## slope_dir2   9.808e-03  4.506e-02   0.218 0.827678
## slope_dir3  -2.364e-02  4.716e-02  -0.501 0.616129
## slope_dir4  -3.546e-02  4.672e-02  -0.759 0.447896
```

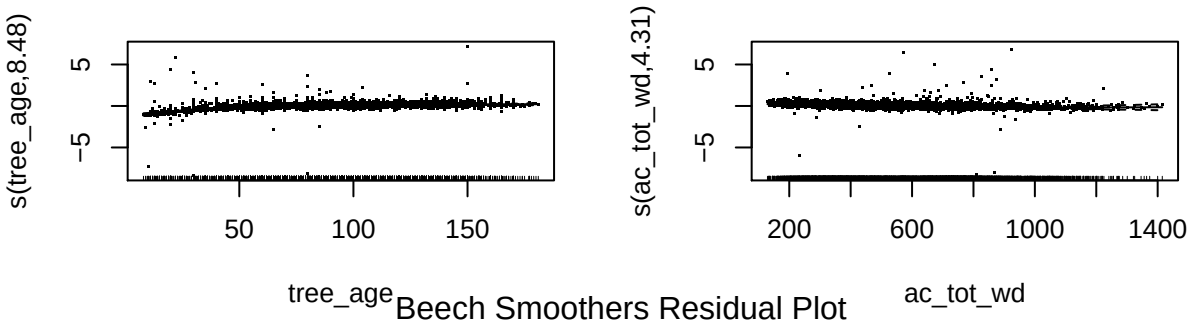
```
## slope_dir5  -5.691e-02  4.476e-02  -1.272 0.203602
## slope_dir6   4.912e-02  4.521e-02   1.086 0.277323
## slope_dir7   2.351e-02  4.860e-02   0.484 0.628612
## slope_dir8   8.013e-02  4.643e-02   1.726 0.084447 .
## slope_dir10  1.546e-03  4.677e-02   0.033 0.973637
## soil_no2    -1.939e-01  4.796e-02  -4.043 5.37e-05 ***
## soil_no3    -1.574e-01  4.500e-02  -3.497 0.000475 ***
## soil_no4    -1.917e-01  4.324e-02  -4.434 9.48e-06 ***
## soil_no5    -1.426e-01  4.488e-02  -3.177 0.001499 **
## soil_no6    -2.319e-01  4.385e-02  -5.288 1.29e-07 ***
## soil_no7    -2.479e-01  4.469e-02  -5.545 3.09e-08 ***
## soil_no8     2.732e-01  1.105e-01   2.473 0.013417 *
## soil_no9    -3.039e-01  4.829e-02  -6.294 3.38e-10 ***
## soil_no10   -2.259e-01  5.826e-02  -3.878 0.000107 ***
## soil_no11   -5.611e-02  6.185e-02  -0.907 0.364326
## soil_no12   -1.921e-03  6.671e-02  -0.029 0.977034
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf  Ref.df      F  p-value
## te(y_utm,x_utm,year) 344.326 457.268  4.319  < 2e-16 ***
## s(tree_age)            8.481  10.323 75.205  < 2e-16 ***
## s(ac_tot_wd)           4.306   5.327  5.266 5.72e-05 ***
## s(n_tot_wd)            1.000   1.000  6.516   0.0107 *
## s(globrad_y_lag1)      5.618   6.705  8.382 7.29e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.576   Deviance explained = 60.9%
## -REML =  -4404  Scale est. = 0.043164  n = 5145
```

```
anova(mod_rbu_fin)
```

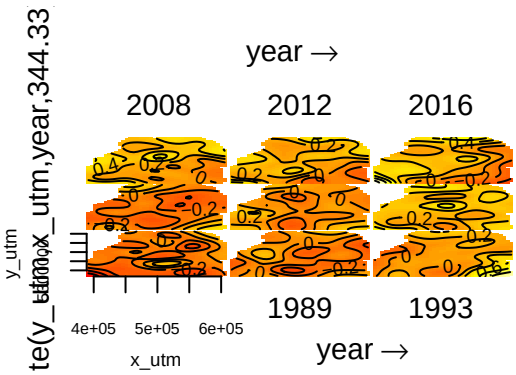
```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio^0.75 ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(40, 25)) + s(tree_age, bs = "cr", k = 20) + s(ac_tot_wd,
##     k = 10) + s(n_tot_wd, k = 10) + H_bhd + s(globrad_y_lag1,
##     bs = "cr", k = 10) + H_spec + prec_y + prec_y_lag1 + slope_dir +
##     soil_no
##
## Parametric Terms:
##             df      F  p-value
## H_bhd        1 76.916  < 2e-16
## H_spec       1  7.981  0.00475
## prec_y       1 23.328 1.41e-06
## prec_y_lag1  1  0.550  0.45839
## slope_dir    9  5.767 5.24e-08
## soil_no     11 10.759  < 2e-16
##
## Approximate significance of smooth terms:
```

```
##                           edf  Ref.df      F  p-value
## te(y_utm,x_utm,year) 344.326 457.268  4.319  < 2e-16
## s(tree_age)            8.481  10.323 75.205  < 2e-16
## s(ac_tot_wd)           4.306   5.327  5.266 5.72e-05
## s(n_tot_wd)            1.000   1.000  6.516   0.0107
## s(globrad_y_lag1)      5.618   6.705  8.382 7.29e-10
```

```r
par(mfrow=c(2, 2))
plot(mod_rbu_fin, select = 2, residuals = T)
plot(mod_rbu_fin, select = 3, residuals = T)
plot(mod_rbu_fin, select = 4, residuals = T)
plot(mod_rbu_fin, select = 5, residuals = T)
mtext("Beech Smoothers Residual Plot", side = 3, line = -13, outer = TRUE)
```

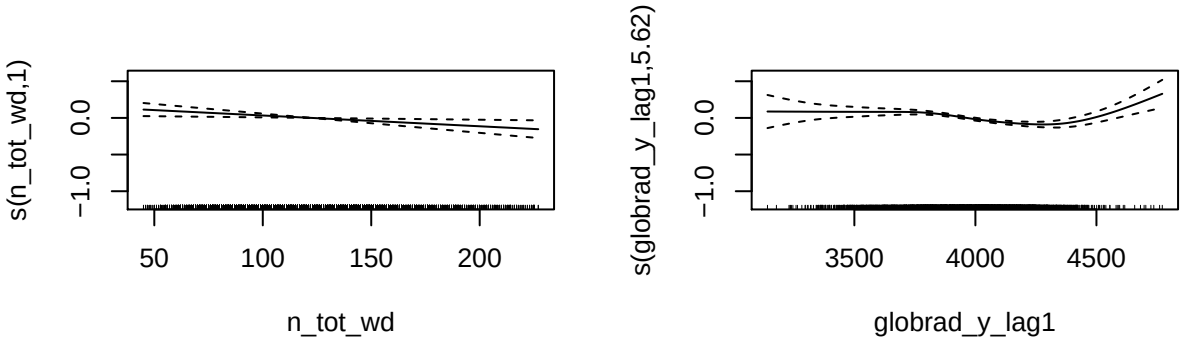



```r
plot(mod_rbu_fin, select = 1)
mtext("Beech Spatio-Temporal Smoother", side = 3, line = -22, outer = TRUE)

par(mfrow=c(2, 2))
```



Beech Spatio−Temporal Smoother

```
plot(mod_rbu_fin, select = 2)
plot(mod_rbu_fin, select = 3)
plot(mod_rbu_fin, select = 4)
plot(mod_rbu_fin, select = 5)
mtext("Beech Smoothers", side = 3, line = -13, outer = TRUE)
```




```
rmse(test_rbu_Y, y_pred)
```

```
## [1] 0.08621973
```

```
dat_tei <- data.frame(tree_data[[5]])
#dat_tei <- one_hot(as.data.table(dat_tei), cols = c("soil_no", "slope_dir"))
#dat_tei <- dat_tei[,-c("soil_no_8", "slope_dir_7")]

train_tei <- dat_tei %>%
  sample_frac(0.8)

test_tei_X <- dat_tei %>%
  anti_join(train_tei) %>%
  select(-c(nbv_ratio))
```

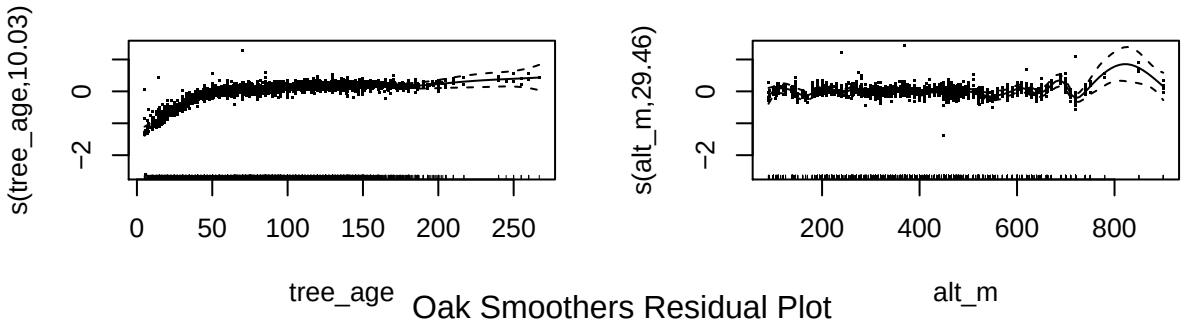
```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "ac_tot_wd", "slope_dir",
```

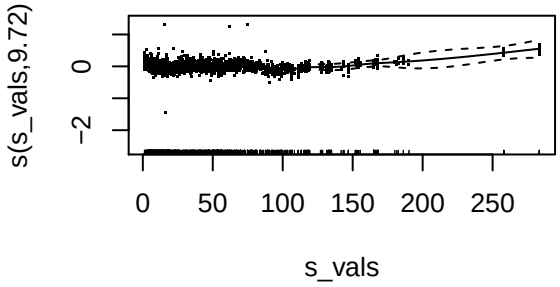
```
test_tei_Y <- dat_tei %>%
  anti_join(train_tei) %>%
  select(c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "ac_tot_wd", "slope_dir",
```

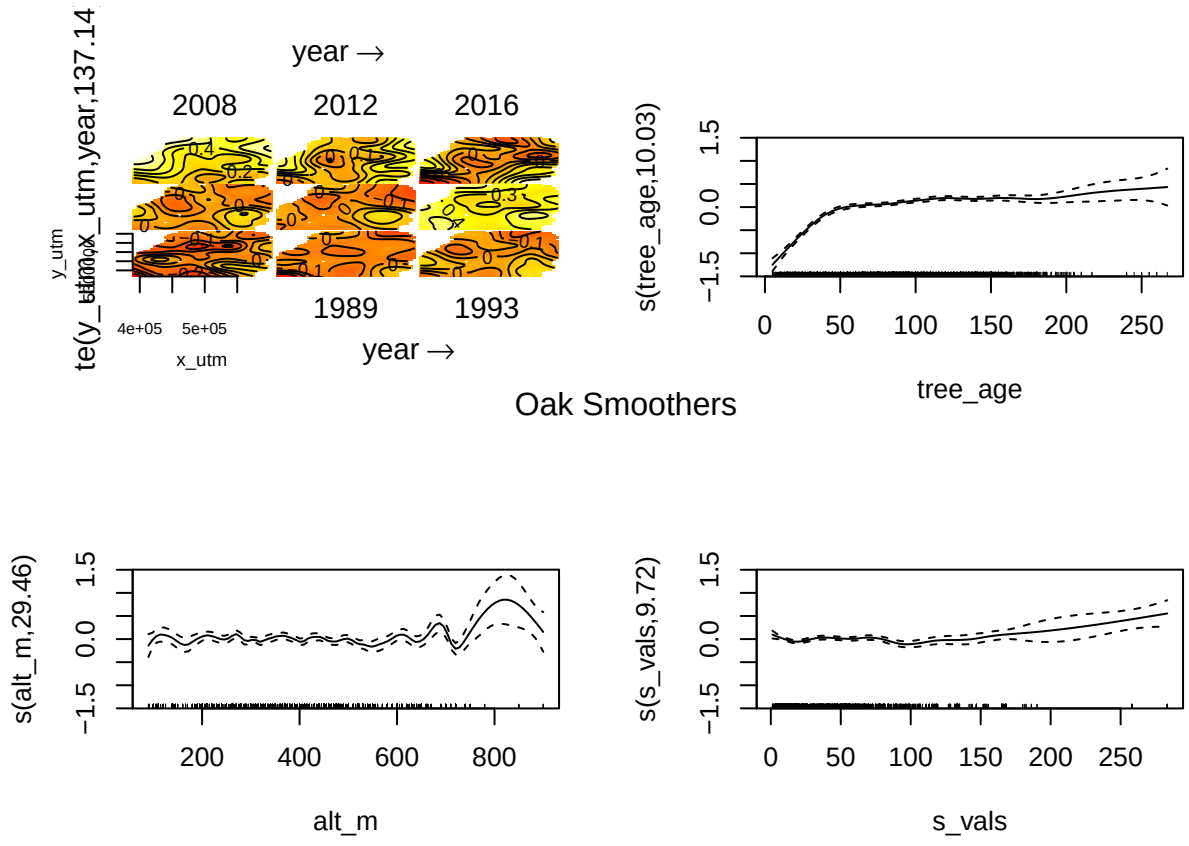
```
mod_tei_fin <- gam(nbv_ratio^0.5 ~ te(y_utm, x_utm, year, bs = c("tp","tp"), d = c(2,1),
                                      k = c(40,25))
                   + s(tree_age, bs="cr", k=120) + s(alt_m, bs="cr", k=100) + s(s_vals, bs="cr", k=100)
                   + slope_dir + soil_no
                   + n_tot_wd,
```

```
                    data=dat_tei,
                    correlation = corARMA(form =~ year | spat, p=1, q=1),
                    family = gaussian(link="logit"),
                    weights = dat_tei$n_trees,
                    method="REML")

y_pred <- predict(mod_tei_fin, newdata=test_tei_X, type = "response")
test_tei_Y <- test_tei_Y$nbv_ratio[!is.na(y_pred)]; y_pred <- y_pred[!is.na(y_pred)]^2

par(mfrow=c(2, 2))
plot(mod_tei_fin, select = 2, residuals = T)
plot(mod_tei_fin, select = 3, residuals = T)
plot(mod_tei_fin, select = 4, residuals = T)
mtext("Oak Smoothers Residual Plot", side = 3, line = -13, outer = TRUE)

par(mfrow=c(2, 2))
```

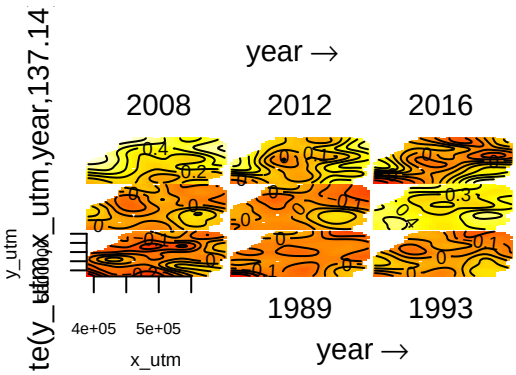





```
plot(mod_tei_fin)
mtext("Oak Smoothers", side = 3, line = -13, outer = TRUE)
```




```
plot(mod_tei_fin, select = 1)
mtext("Oak Spatio-Temporal Smoother", side = 3, line = -22, outer = TRUE)

par(mfrow=c(2, 2))
```



Oak Spatio−Temporal Smoother

```
plot(mod_tei_fin, select = 2)
plot(mod_tei_fin, select = 3)
plot(mod_tei_fin, select = 4)
mtext("Oak Smoothers", side = 3, line = -13, outer = TRUE)

e <- residuals(mod_tei_fin); fv <- fitted(mod_tei_fin)
lm(log(e^2) ~ log(fv))
```

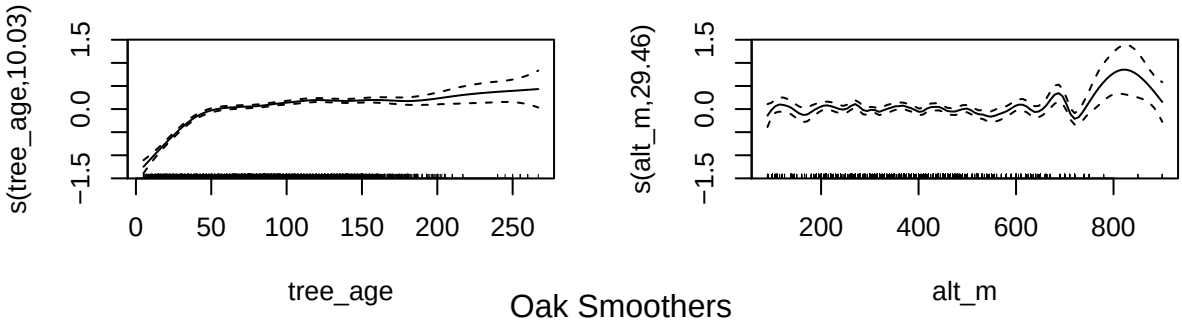
```
##
## Call:
## lm(formula = log(e^2) ~ log(fv))
##
## Coefficients:
## (Intercept)      log(fv)
##      -7.192       -1.434
```

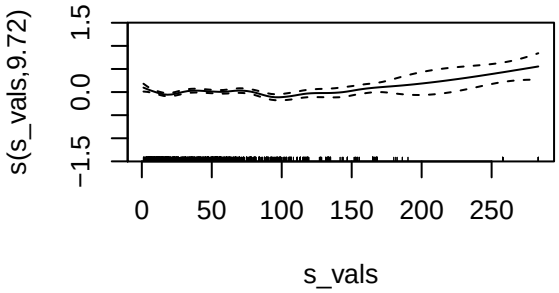
```
mean(dat_tei$nbv_ratio); mean(fitted(mod_tei_fin)^2)
```

```
## [1] 0.2508663
```

```
## [1] 0.2415573
```

```
par(mfrow = c(2, 2))
```

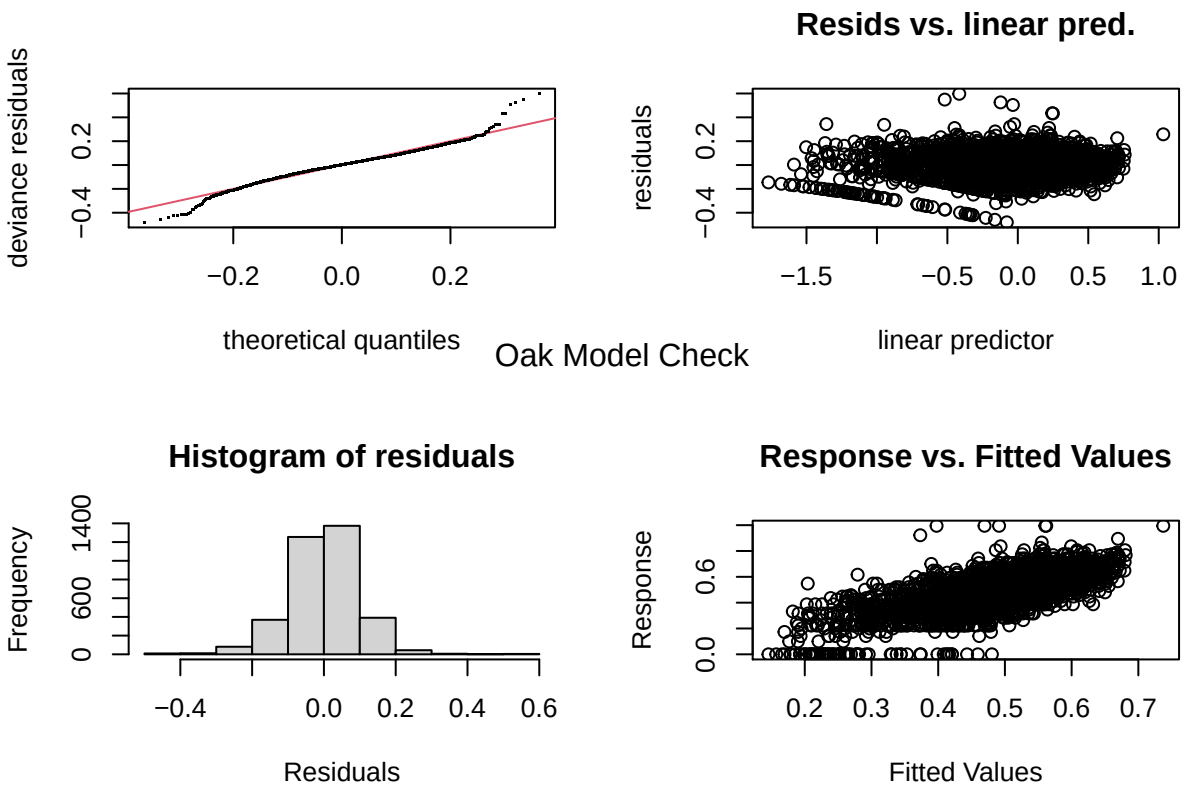





```
gam.check(mod_tei_fin)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 9 iterations.
## Gradient range [-1.332886e-05,1.749447e-06]
## (score -2855.396 & scale 0.01002534).
## Hessian positive definite, eigenvalue range [1.040238,1759.396].
## Model rank =  1338 / 1338
##
## 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(y_utm,x_utm,year) 999.00 137.14    0.95  <2e-16 ***
## s(tree_age)          119.00  10.03    0.94  <2e-16 ***
## s(alt_m)              99.00  29.46    0.84  <2e-16 ***
## s(s_vals)             99.00   9.72    0.85  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mtext("Oak Model Check", side = 3, line = -13, outer = TRUE)
```




```
summary(mod_tei_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio^0.5 ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(40, 25)) + s(tree_age, bs = "cr", k = 120) + s(alt_m,
##     bs = "cr", k = 100) + s(s_vals, bs = "cr", k = 100) + slope_dir +
##     soil_no + n_tot_wd
##
## Parametric coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3494412  0.1235758   2.828 0.004716 **
## slope_dir1  -0.1443757  0.0796329  -1.813 0.069919 .
## slope_dir2  -0.0781901  0.0816625  -0.957 0.338395
## slope_dir3  -0.1569397  0.0834810  -1.880 0.060203 .
## slope_dir4  -0.0919690  0.0820485  -1.121 0.262407
## slope_dir5  -0.0742161  0.0805868  -0.921 0.357145
## slope_dir6  -0.0396020  0.0810608  -0.489 0.625195
## slope_dir7  -0.0514480  0.0833710  -0.617 0.537213
## slope_dir8  -0.0779347  0.0817191  -0.954 0.340310
## slope_dir10 -0.1082779  0.0763883  -1.417 0.156439
## soil_no2    -0.2530935  0.0659966  -3.835 0.000128 ***
## soil_no3    -0.3266785  0.0632623  -5.164 2.56e-07 ***
```

```
## soil_no4    -0.2418125  0.0592836  -4.079 4.63e-05 ***
## soil_no5    -0.1733851  0.0690471  -2.511 0.012082 *
## soil_no6    -0.1869123  0.0571528  -3.270 0.001085 **
## soil_no7    -0.1231538  0.0629416  -1.957 0.050474 .
## soil_no8     0.1044867  0.1481397   0.705 0.480657
## soil_no9    -0.1688051  0.0641787  -2.630 0.008572 **
## soil_no10   -0.1187965  0.0781408  -1.520 0.128533
## soil_no11   -0.0217709  0.1469396  -0.148 0.882224
## soil_no12   -0.3084908  0.0734518  -4.200 2.74e-05 ***
## n_tot_wd    -0.0011259  0.0006686  -1.684 0.092269 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf Ref.df      F  p-value
## te(y_utm,x_utm,year) 137.142 192.76  3.005  < 2e-16 ***
## s(tree_age)           10.029  12.49 80.557  < 2e-16 ***
## s(alt_m)              29.462  35.78  2.636 4.82e-07 ***
## s(s_vals)              9.715  11.76  4.410 5.94e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.457   Deviance explained = 48.9%
## -REML = -2855.4  Scale est. = 0.010025  n = 3546
```

```
anova(mod_tei_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio^0.5 ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(40, 25)) + s(tree_age, bs = "cr", k = 120) + s(alt_m,
##     bs = "cr", k = 100) + s(s_vals, bs = "cr", k = 100) + slope_dir +
##     soil_no + n_tot_wd
##
## Parametric Terms:
##           df     F  p-value
## slope_dir  9 2.103   0.0261
## soil_no   11 5.194 3.73e-08
## n_tot_wd   1 2.836   0.0923
##
## Approximate significance of smooth terms:
##                           edf  Ref.df      F  p-value
## te(y_utm,x_utm,year) 137.142 192.760  3.005  < 2e-16
## s(tree_age)           10.029  12.485 80.557  < 2e-16
## s(alt_m)              29.462  35.784  2.636 4.82e-07
## s(s_vals)              9.715  11.758  4.410 5.94e-07
```

```
rmse(test_tei_Y, y_pred)
```

```
## [1] 0.08645169
```

```
dat_wta <- data.frame(tree_data[[6]])
#dat_wta <- one_hot(as.data.table(dat_wta), cols = c("soil_no", "slope_dir"))
```

```r
#dat_wta <- dat_wta[,-c("soil_no_1", "slope_dir_7")]

train_wta <- dat_wta %>%
  sample_frac(0.8)

test_wta_X <- dat_wta %>%
  anti_join(train_wta) %>%
  select(-c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "ac_tot_wd", "n_trees", "
```

```r
test_wta_Y <- dat_wta %>%
  anti_join(train_wta) %>%
  select(c(nbv_ratio))
```

```
## Joining, by = c("x_utm", "y_utm", "year", "spat", "nbv_ratio", "tree_age", "ac_tot_wd", "n_trees", "
```

```r
mod_wta_fin <- gam(nbv_ratio^0.8 ~ te(y_utm, x_utm, year, bs = c("tp","tp"), d = c(2,1),
                                      k = c(25,20))
                   + s(tree_age, bs="cr", k=20) + s(tpi750, bs="cr", k=80)
                   + ac_tot_wd + n_tot_wd + n_trees + alt_m + globrad_y
                   + slope_dir + soil_no,
                   data=train_wta,
                   correlation = corARMA(form =~ year | spat, p=1, q=1),
                   family = gaussian(link="logit"),
                   weights = train_wta$n_trees,
                   method="REML")

e <- residuals(mod_wta_fin); fv <- fitted(mod_wta_fin)
lm(log(e^2) ~ log(fv))
```

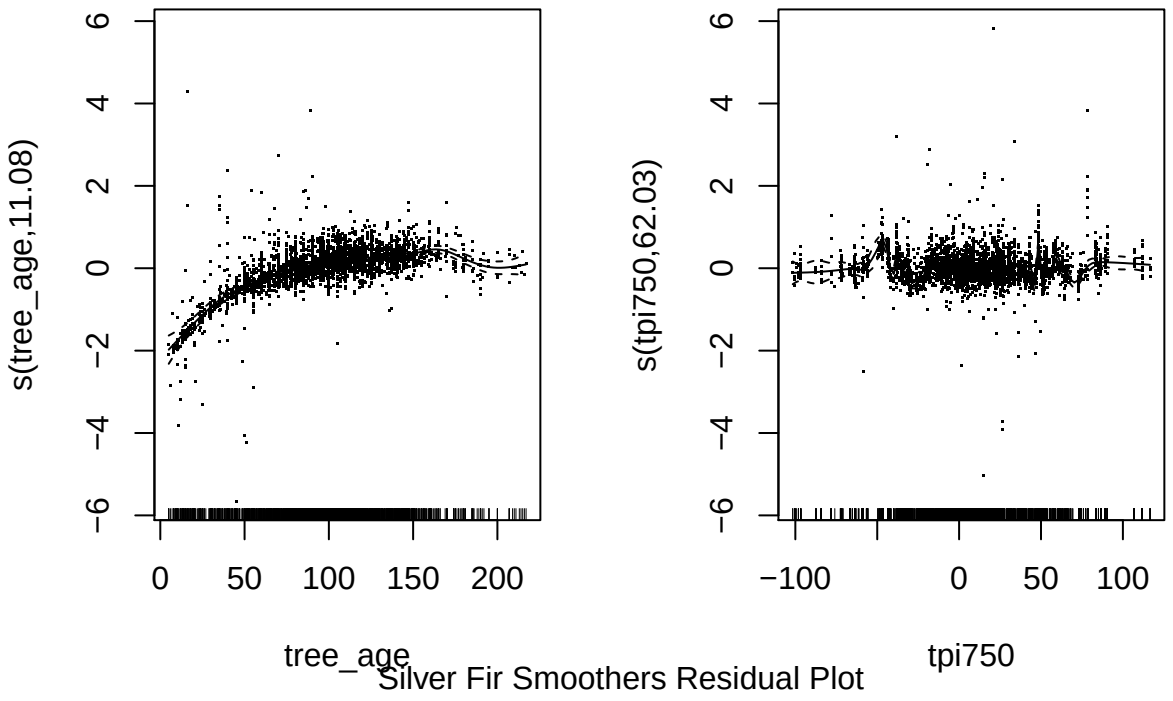
```
##
## Call:
## lm(formula = log(e^2) ~ log(fv))
##
## Coefficients:
## (Intercept)      log(fv)
##     -3.5494       0.6975
```

```r
mean(dat_wta$nbv_ratio); mean(fitted(mod_wta_fin))^1.25
```

```
## [1] 0.2946043
```

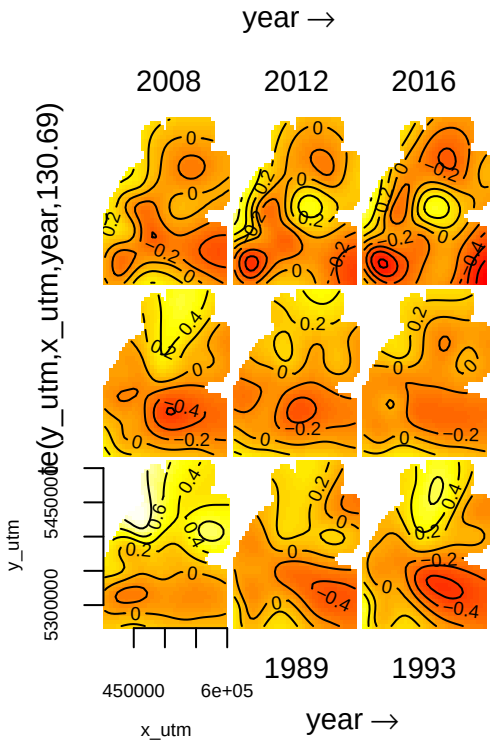
```
## [1] 0.2912323
```

```r
y_pred <- predict(mod_wta_fin, newdata=test_wta_X, type = "response")
test_wta_Y <- test_wta_Y$nbv_ratio[!is.na(y_pred)]; y_pred <- y_pred[!is.na(y_pred)]^1.25

par(mfrow=c(1, 2))
plot(mod_wta_fin, select = 2, residuals = TRUE)
plot(mod_wta_fin, select = 3, residuals = TRUE)
mtext("Silver Fir Smoothers Residual Plot", side = 3, line = -22, outer = TRUE)
```



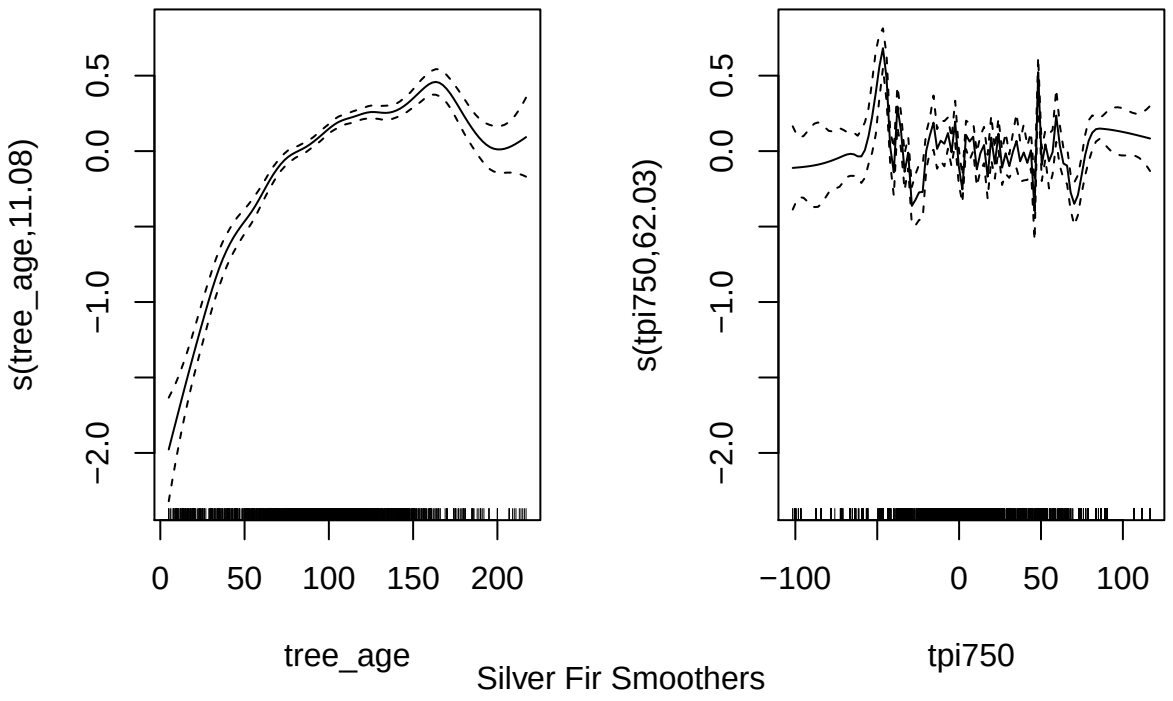
Silver Fir Smoothers Residual Plot

```
plot(mod_wta_fin, select = 1)
mtext("Silver Fir Spatio-Temporal Smoother", side = 3, line = -22, outer = TRUE)

par(mfrow=c(1, 2))
```



Silver Fir Spatio−Temporal Smoother

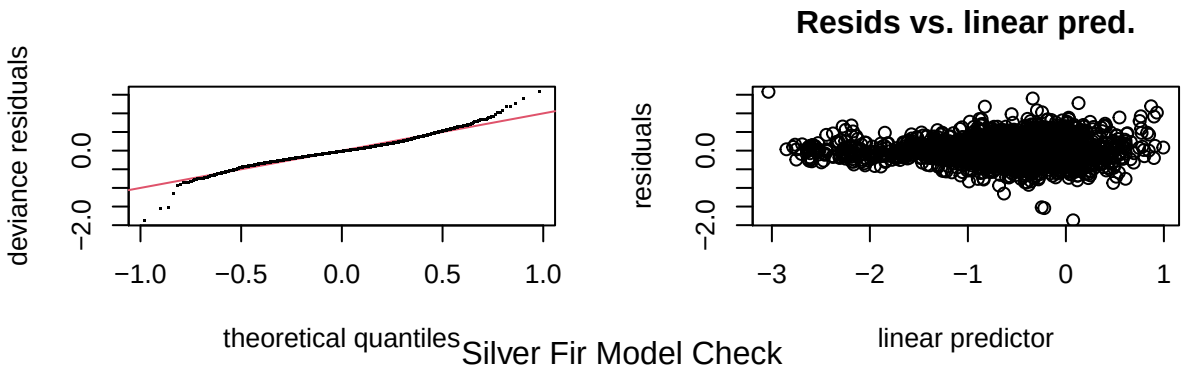
```
plot(mod_wta_fin, select = 2)
plot(mod_wta_fin, select = 3)
mtext("Silver Fir Smoothers", side = 3, line = -22, outer = TRUE)
```



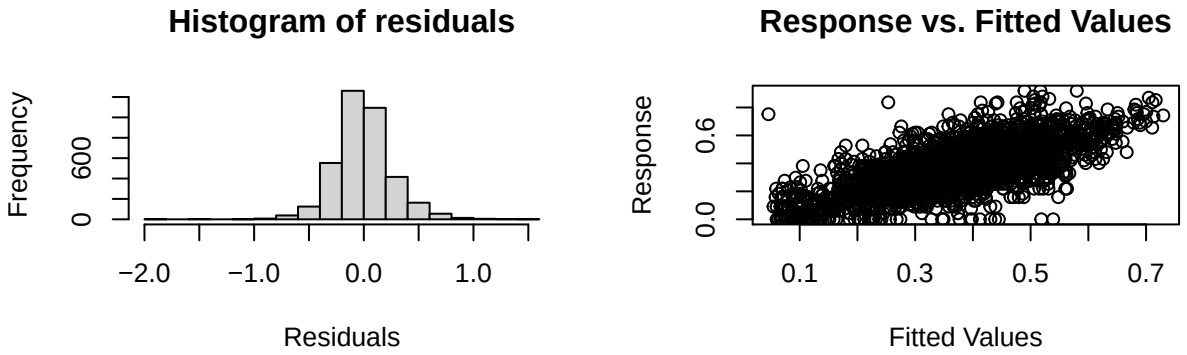

```
par(mfrow = c(2, 2))
gam.check(mod_wta_fin)
```

```
##
## Method: REML   Optimizer: outer newton
## full convergence after 5 iterations.
## Gradient range [-6.419213e-06,7.625966e-07]
## (score -2596.059 & scale 0.07231958).
## Hessian positive definite, eigenvalue range [2.172054,1854.084].
## Model rank =  623 / 623
##
## 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(y_utm,x_utm,year) 499.0 130.7    0.98    0.12
## s(tree_age)           19.0  11.1    1.00    0.48
## s(tpi750)             79.0  62.0    0.82  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mtext("Silver Fir Model Check", side = 3, line = -13, outer = TRUE)
```



Silver Fir Model Check




```
summary(mod_wta_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio^0.8 ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(25, 20)) + s(tree_age, bs = "cr", k = 20) + s(tpi750,
##     bs = "cr", k = 80) + ac_tot_wd + n_tot_wd + n_trees + alt_m +
##     globrad_y + slope_dir + soil_no
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.251e+00  2.326e-01  -5.376 8.09e-08 ***
## ac_tot_wd    6.856e-04  1.492e-04   4.596 4.47e-06 ***
## n_tot_wd    -1.281e-03  6.771e-04  -1.891  0.05864 .
## n_trees      6.938e-03  4.427e-04  15.672  < 2e-16 ***
## alt_m        1.955e-04  8.548e-05   2.287  0.02226 *
## globrad_y    4.906e-05  4.965e-05   0.988  0.32321
## slope_dir1   1.271e-01  4.080e-02   3.116  0.00185 **
## slope_dir2   9.738e-02  4.492e-02   2.168  0.03024 *
## slope_dir3  -4.407e-02  5.626e-02  -0.783  0.43345
## slope_dir4   7.124e-02  4.669e-02   1.526  0.12715
## slope_dir5  -8.431e-02  4.877e-02  -1.729  0.08393 .
## slope_dir6   7.759e-02  4.873e-02   1.592  0.11142
```

```
## slope_dir7   2.766e-02  4.080e-02   0.678  0.49782
## slope_dir8   7.641e-02  4.101e-02   1.863  0.06252 .
## slope_dir10  1.378e-02  4.173e-02   0.330  0.74124
## soil_no2     6.457e-02  7.007e-02   0.921  0.35687
## soil_no3    -8.933e-02  6.908e-02  -1.293  0.19600
## soil_no4    -2.132e-02  6.676e-02  -0.319  0.74945
## soil_no5    -2.392e-02  6.643e-02  -0.360  0.71883
## soil_no6     1.895e-02  7.043e-02   0.269  0.78791
## soil_no7     1.125e-01  8.024e-02   1.402  0.16102
## soil_no8     1.070e+00  4.174e-01   2.564  0.01039 *
## soil_no9     5.656e-02  1.011e-01   0.560  0.57582
## soil_no10   -3.629e-02  8.197e-02  -0.443  0.65802
## soil_no11   -5.478e-01  2.100e-01  -2.608  0.00914 **
## soil_no12   -4.194e-01  8.607e-02  -4.873 1.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                          edf Ref.df      F p-value
## te(y_utm,x_utm,year) 130.69 168.15  5.151  <2e-16 ***
## s(tree_age)           11.08  13.20 59.686  <2e-16 ***
## s(tpi750)             62.03  68.62 10.648  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.663   Deviance explained = 68.7%
## -REML = -2596.1  Scale est. = 0.07232   n = 3738
```

```
anova(mod_wta_fin)
```

```
##
## Family: gaussian
## Link function: logit
##
## Formula:
## nbv_ratio^0.8 ~ te(y_utm, x_utm, year, bs = c("tp", "tp"), d = c(2,
##     1), k = c(25, 20)) + s(tree_age, bs = "cr", k = 20) + s(tpi750,
##     bs = "cr", k = 80) + ac_tot_wd + n_tot_wd + n_trees + alt_m +
##     globrad_y + slope_dir + soil_no
##
## Parametric Terms:
##           df       F  p-value
## ac_tot_wd  1  21.119 4.47e-06
## n_tot_wd   1   3.578   0.0586
## n_trees    1 245.610  < 2e-16
## alt_m      1   5.230   0.0223
## globrad_y  1   0.976   0.3232
## slope_dir  9   5.250 3.96e-07
## soil_no   11   9.396  < 2e-16
##
## Approximate significance of smooth terms:
##                          edf Ref.df      F p-value
## te(y_utm,x_utm,year) 130.69 168.15  5.151  <2e-16
## s(tree_age)           11.08  13.20 59.686  <2e-16
## s(tpi750)             62.03  68.62 10.648  <2e-16
```

```r
rmse(test_wta_Y, y_pred)
```

```
## [1] 0.1089584
```