A geostatistical investigation of the ammonia-livestock relationship in the Po Valley, Italy

Extended analysis for SIS 2023 annual meeting, Ancona (Italy) 21-23 June 2023

Paolo Maranzano* Kelly McConville† Philipp Otto‡ Felicetta Carillo§

27 aprile 2023

Abstract

Extended analysis presented at SIS 2023 annual meeting, Ancona (Italy) 21-23 June 2023.

Data management

Load data

```
# Use all the farms (F) or only farms with animals (T)
only_animals <- F
# Buffer length for land use variables
buffer_lng <- 200

##### Load agrarian subregions covariates (census information)
load("../Municipal_data/Data_final/X_Areal_AgrSubReg_SubregAgrAggr.RData")
X_Areal <- X_Areal_AgrSubReg %>%
  mutate(across(contains("BDN"), ~ ifelse(is.na(.x),0,.x)))
rm(X_Areal_AgrSubReg)

##### Load unit-level data from CREA (sample information) to be aggregate
load(paste0("../UnitLevel_data/Data_final/CREA_UnitLevelData_SubregAgrAggr_",buffer_lng,".RData"))

## Filtering (only animal farms)
if (only_animals == T) {
  CREA <- CREA %>%
    filter(!is.na(UBA_Total))
}
```

Small area aggregation

```
## Variable to aggregate
Var_input <- c("Manure_Processed_Tot","Manure_Prod_Tot",
               "Manure_Prod_Bovines","Manure_Prod_Swines","NH3_AgricLive",
               "Anim_Tot_Heads","Anim_Bovines_Heads","Anim_Swines_Heads")
Var_transf <- c('function(x) x*100/(1000*1000)', # Quintals to ths of tonnes
```

*University of Milano-Bicocca, Italy \& Fondazione Eni Enrico Mattei (FEEM), Milano, Italy
†Harvard University, USA
‡Leibniz University of Hannover, Germany
§CREA - Politiche di Bioeconomia

```
                'function(x) x*100/(1000*1000)', # Quintals to ths of tonnes
                FALSE,FALSE,FALSE,
                FALSE,FALSE,FALSE)
Var_weight <- "Weight_Ita_Reg"
Var_domains <- c("Year","Agrarian_SubRegion")
## Aggregation
for (v in 1:length(Var_input)) {
  var_aggr <- SAE_Aggregation(Data = CREA,
                              Var_input = Var_input[v],
                              Var_weight = Var_weight,
                              Var_domains = Var_domains,
                              Transf = Var_transf[v])
  if (v == 1) {
    CREA_SAE <- var_aggr
  } else {
    CREA_SAE_add <- var_aggr %>%
      dplyr::select(Year,Agrarian_SubRegion,contains(c("Tot_","VarTot_","Mean_","VarMean_")))
    CREA_SAE <- left_join(x = CREA_SAE, y = CREA_SAE_add, by = c("Year","Agrarian_SubRegion"))
  }
}
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(c(Var_input), .fns = ~FUN(.x))`.
## Caused by warning:
## ! Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(Var_input)
##
##   # Now:
##   data %>% select(all_of(Var_input))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

```
rm(Var_weight,Var_domains,var_aggr,v,CREA_SAE_add)
```

Join CREA and census information

```
##### Join dataset
SAE_data <- inner_join(x = CREA_SAE, y = X_Areal, by = c("Agrarian_SubRegion","Year"))
```

Data preparation

```
##### Data filtering
Data_full <- SAE_data %>%
  dplyr::filter(Agrarian_SubRegion != "Lario") %>%
  mutate(BDN_Farm = BDN_Bov_Farms_Num + BDN_Swines_Farms_Num,
         Nd = BDN_Farm,
         BDN_Anim = BDN_Bov_Num + BDN_Swines_Num,
         BDN_Anim_avg = BDN_Anim / BDN_Farm,
         BDN_Bov_pc = BDN_Bov_Num/BDN_Bov_Farms_Num,
         BDN_Swines_pc = BDN_Swines_Num/BDN_Swines_Farms_Num,
         # Cells to hectars (hm2): https://en.wikipedia.org/wiki/Square_metre
```

```
        Land_Agric = Land_Agric/2500*100,
        ValAgr_High = ValAgr_High/2500*100,
        ValAgr_High_Ratio = ValAgr_High/Land_Agric,
        # NH3 (average across the year) conversion: from mg/m2 to kg/hm2
        # 1 mg/m2 = 10^(-6)kg / 10^(-4)hm2 = 10^(-2) kg/hm2
        # 1 mg/m2 = 10^(-6)kg / 10^(-6)km2 = 1 kg/km2
        NH3_AgricLive_avg = NH3_AgricLive * 100,
        # Mean HT
        MeanHT_Manure_Processed_Tot = TotHT_Manure_Processed_Tot/Nd,
        VarMeanHT_Manure_Processed_Tot = VarTotHT_Manure_Processed_Tot/(Nd^2)
        )
```

Small area modeling

Equation:

$$\hat{\bar{y}}_d = X_d\boldsymbol{\beta} + u_d + e_d$$

where * $y_d = $ Average manure processed (spread) or produced on the sub-agrarian region d * $X_d = $ is a design matrix of covariates (area-level) for sub-agrarian region d

Model 1: Fay-Herriot

- $X_d = $ includes: + Total number of animals (BDN) + Total agricultural land (SIARL) + Total land with highly-rated agricultural value (SIARL) + Average DEM of the sub-agrarian region

```
## Model formula
# model_formula <- Mean_Manure_Processed_Tot ~ Mean_Anim_Tot_Heads + NH3_AgricLive_avg
model_formula <- MeanHJ_Manure_Processed_Tot ~  BDN_Anim + Land_Agric + ValAgr_High + DEM_avg

Years <- 2016:2020
Output_FH_list <- vector(mode = "list", length = length(Years))
Models_FH_list <- vector(mode = "list", length = length(Years))

for (yr in 1:length(Years)) {
  ##### Data preparation
  ## Filtering
  Data <- Data_full %>%
    dplyr::filter(Year == Years[[yr]])

  ## Data.frame
  Data <- data.frame(Data)

  ## Data in matrix form
  Yname <- colnames(model.frame(formula = model_formula, data = Data))[1]
  Xnames <- colnames(model.frame(formula = model_formula, data = Data)[,-1])
  cor(cbind(Data[,Xnames],Data[,Yname]))

  ## Data in matrix form
  Data <- cbind(Year = Data$Year,
                Agrarian_SubRegion = Data$Agrarian_SubRegion,
                Nd = Data$Nd,
                Nd_hat = Data$Nd_hat,
                nd_hat = Data$nd_hat,
                Y = Data[[Yname]],
```

```r
                VarDir_Y = Data[[paste0("Var",Yname)]],
                Y_cv = 100*sqrt(Data[[paste0("Var",Yname)]]) / Data[[Yname]],
                model.frame(formula = model_formula, data = Data))

  ## Model estimation
  m_FH <- mseFH(formula = model_formula,
                method = "REML",
                vardir = VarDir_Y,
                MAXITER = 1000, PRECISION = 10^(-6),
                data = Data)

  # Regression (FE) coefficients
  FH_coefs <- m_FH$est$fit$estcoef
  # EBLUP estimates
  FH_eblup <- m_FH$est$eblup
  # EBLUP MSE
  FH_mse <- m_FH$mse
  # CV with respect to direct estimator
  FH_eblup_cv <- 100*sqrt(FH_mse) / Data[[Yname]]

  # Random effect variance
  sigma2_u <- m_FH$est$fit$refvar
  # Direct estimate variances
  sigma2_e <- Data$VarDir_Y
  # Shrinkage factor
  gamma_i <- sigma2_u / (sigma2_u + sigma2_e)

  ## Check EBLUP estimates computation
  # muhat = cbind(1,X)%*%m_FH$est$fit$estcoef[,1] + gamma_i*(Y - cbind(1,X)%*%m_FH$est$fit$estcoef[,1])
  # muhat - FH_eblup

  # Residuals
  FH_res <- Data[[Yname]] - FH_eblup

  # Output
  Data$FH_eblup <- as.vector(FH_eblup)
  Data$FH_eblup_cv <- as.vector(FH_eblup_cv)
  Data$FH_mse <- as.vector(FH_mse)

  # Put in the list
  Output_FH_list[[yr]] <- Data
  Models_FH_list[[yr]] <- m_FH
}

Output_FH <- bind_rows(Output_FH_list)
```

```r
round(FH_coefs,3) %>%
  knitr::kable()
```

	beta	std.error	tvalue	pvalue
X(Intercept)	0.117	0.044	2.684	0.007
XBDN_Anim	0.000	0.000	2.188	0.029
XLand_Agric	0.000	0.000	0.176	0.861

	beta	std.error	tvalue	pvalue
XValAgr_High	0.000	0.000	-0.656	0.512
XDEM_avg	0.000	0.000	-0.543	0.587

Model 2: Spatial Fay-Herriot

Equation:

$$\hat{\bar{y}}_d = X_d\boldsymbol{\beta} + u_d + e_d$$

where the RE u_d follow a an simultaneously autoregressive (SAR) process with unknown autoregression parameter $-1 \leq \rho \leq +1$ with W is the proximity matrix

$$u_d = \rho W u_d + \varepsilon_d$$

Define contiguity matrix (Queen)

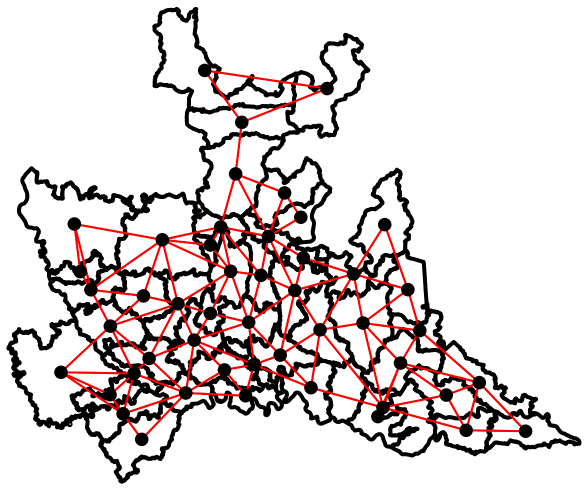
```
library(spdep)
```

```
## Caricamento del pacchetto richiesto: spData

## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
# Load shapefile
load("H:/Il mio Drive/Agrimonia_CREA_SmallArea/Data/Municipal_data/Agrarian_SubRegions/CREA_AgrSubregion
# Filter: only available subregions
Shape <- AgrSubregions_aggr %>%
  filter(Agrarian_SubRegion %in% Data$Agrarian_SubRegion)
# List of neighbours
contnb_q <- poly2nb(Shape, queen = TRUE, row.names = Shape$Agrarian_SubRegion)
# Contiguity matrix (row-standardized): other distance measures could be used: https://rpubs.com/erikaa
contmat_q <- nb2mat(contnb_q)

### Mapping
coords <- st_coordinates(st_centroid(Shape))
```

```
## Warning: st_centroid assumes attributes are constant over geometries
```

```
colnames(coords) <- c("Longitude","Latitude")
Shape_coord <- cbind(Shape,coords)
plot(x = st_geometry(Shape), border = 'black', col="white", lwd=2)
plot(contnb_q, coords = coords, pch = 19, cex = 0.8, col = 'red',add = T)
```



Model estimation

```
model_formula_sp <- as.formula(paste0(paste(deparse(model_formula), collapse = ""), " - 1"))

Output_SFH_list <- vector(mode = "list", length = length(Years))
Models_SFH_list <- vector(mode = "list", length = length(Years))

for (yr in 1:length(Years)) {

    ##### Data preparation
  ## Filtering
  Data <- Data_full %>%
    dplyr::filter(Year == Years[[yr]])


  ## Data.frame
  Data <- data.frame(Data)

  ## Data in matrix form
  Yname <- colnames(model.frame(formula = model_formula, data = Data))[1]
  Xnames <- colnames(model.frame(formula = model_formula, data = Data)[,-1])
  cor(cbind(Data[,Xnames],Data[,Yname]))

  ## Data in matrix form
  Data <- cbind(Year = Data$Year,
```

```
                Agrarian_SubRegion = Data$Agrarian_SubRegion,
                Nd = Data$Nd,
                Nd_hat = Data$Nd_hat,
                nd_hat = Data$nd_hat,
                Y = Data[[Yname]],
                VarDir_Y = Data[[paste0("Var",Yname)]],
                Y_cv = 100*sqrt(Data[[paste0("Var",Yname)]]) / Data[[Yname]],
                model.frame(formula = model_formula, data = Data))


  ## Model estimation
  m_SFH <- mseSFH(formula = model_formula_sp,
                  method = "REML",
                  vardir = VarDir_Y,
                  proxmat = contmat_q,
                  MAXITER = 1000, PRECISION = 10^(-6),
                  data = Data)

  # Regression (FE) coefficients
  SFH_coefs <- m_SFH$est$fit$estcoef
  # EBLUP estimates
  SFH_eblup <- m_SFH$est$eblup
  SFH_eblup[SFH_eblup < 10^{-6}] <- 0
  # EBLUP MSE
  SFH_mse <- m_SFH$mse
  SFH_mse[SFH_mse < 10^{-6}] <- 0
  SFH_eblup_cv <- 100*sqrt(SFH_mse)/ Data[[Yname]]


  # Random effect variance
  SFH_sigma2_u <- m_SFH$est$fit$refvar
  # Direct estimate variances
  sigma2_e <- Data$VarDir_Y
  # Shrinkage factor
  SFH_gamma_i <- SFH_sigma2_u / (SFH_sigma2_u + sigma2_e)

  # Residuals
  SFH_res <- Data[[Yname]] - SFH_eblup

  # Output
  Data$SFH_eblup <- as.vector(SFH_eblup)
  Data$SFH_eblup_cv <- as.vector(SFH_eblup_cv)
  Data$SFH_mse <- as.vector(SFH_mse)
  # Data$SFH_res <- SFH_res

  # Put in the list
  Output_SFH_list[[yr]] <- Data
  Models_SFH_list[[yr]] <- m_SFH
}

Output_SFH <- bind_rows(Output_SFH_list)
```

Model 3: Spatio-temporal Fay-Herriot with independent domain-time random effects.

Equation:

$$\hat{\bar{y}}_{dt} = X_d\boldsymbol{\beta} + u_{1d} + u_{2dt} + e_{dt}$$

where

$$e_{dt} \sim MVN(0, \sigma^2_{dt} I_M)$$

where the RE u_d follow a an simultaneously autoregressive (SAR) process with unknown autoregression parameter $-1 \leq \rho \leq +1$ with W is the proximity matrix

$$u_{1d} = \rho_1 W u_{1d} + \varepsilon_d$$

with temporal dynamics with independent (I.ID.) effects across time

$$u_{2dt} \sim MVN(0, \sigma^2_2 I_M)$$

(with $M = DT$)

```
##### Data.frame
Data_ST <- data.frame(Data_full)

## Data in matrix form
Data_ST <- cbind(Year = Data_ST$Year,
                 Agrarian_SubRegion = Data_ST$Agrarian_SubRegion,
                 VarDir_Y = Data_ST[[paste0("Var",Yname)]],
                 model.frame(formula = model_formula, data = Data_ST))

## Number of domains
Data_ST %>%
  group_by(Year) %>%
  summarise(n())
```

```
## # A tibble: 5 x 2
##    Year `n()`
##   <dbl> <int>
## 1  2016    48
## 2  2017    48
## 3  2018    48
## 4  2019    48
## 5  2020    48
```

```
nD <- length(unique(Data_ST$Agrarian_SubRegion))

## Number of time stamps
Data_ST %>%
  group_by(Agrarian_SubRegion) %>%
  summarise(n())
```

```
## # A tibble: 48 x 2
##    Agrarian_SubRegion                   `n()`
##    <chr>                                <int>
##  1 Basso Pavese                             5
##  2 Brianza                                  5
##  3 Colline Del Medio Chiese                 5
##  4 Colline Di Bergamo                       5
```

```
##   5 Colline Di Brescia                           5
##   6 Colline Oltrepo Pavese                       5
##   7 Lomellina                                    5
##   8 Morenica Del Lago Diseo                      5
##   9 Morenica Meridionale Del Benaco              5
##  10 Morenica Nord Occidentale Del Benaco        5
## # i 38 more rows
```

```
nT <- length(unique(Data_ST$Year))

## Model estimation
m_STFH <- pbmseSTFH(formula = model_formula_sp,B = 50,
                    D = nD, T = nT,
                    vardir = VarDir_Y,
                    proxmat = contmat_q,
                    model = "S",
                    theta_iter=TRUE, MAXITER = 1000, PRECISION = 10^(-6),
                    data = Data_ST)
```

```
##
## Bootstrap procedure with B = 50 iterations starts.
## b = 1
## b = 2
## b = 3
## b = 4
## b = 5
## b = 6
## b = 7
## b = 8
## b = 9
## b = 10
## b = 11
## b = 12
## b = 13
## b = 14
## b = 15
## b = 16
## b = 17
## b = 18
## b = 19
## b = 20
## b = 21
## b = 22
## b = 23
## b = 24
## b = 25
## b = 26
## b = 27
## b = 28
## b = 29
## b = 30
## b = 31
## b = 32
## b = 33
## b = 34
```

```
## b = 35
## b = 36
## b = 37
## b = 38
## b = 39
## b = 40
## b = 41
## b = 42
## b = 43
## b = 44
## b = 45
## b = 46
## b = 47
## b = 48
## b = 49
## b = 50
```

```
# m_STFH$fit$convergence
# m_STFH$fit$estvarcomp_iterations
# m_STFH$fit$goodness
# m_FH$est$fit$goodness
# m_SFH$est$fit$goodness


# Regression (FE) coefficients
STFH_coefs <- m_STFH$est$fit$estcoef
# EBLUP estimates
STFH_eblup <- m_STFH$est$eblup
STFH_eblup[STFH_eblup < 10^{-6}] <- 0
# EBLUP MSE
STFH_mse <- m_STFH$mse
STFH_mse[STFH_mse < 10^{-6}] <- 0
STFH_eblup_cv <- 100*sqrt(STFH_mse)/ Data_ST[[Yname]]


# Output
Data_ST$STFH_eblup <- as.vector(STFH_eblup)
Data_ST$STFH_eblup_cv <- as.vector(STFH_eblup_cv)
Data_ST$STFH_mse <- as.vector(STFH_mse)
```

Model 4: Spatio-temporal Fay-Herriot with autoregressive domain-time random effects.

Equation:

$$\hat{\bar{y}}_{dt} = X_d\boldsymbol{\beta} + u_{1d} + u_{2dt} + e_{dt}$$

where

$$e_{dt} \sim MVN(0, \sigma^2_{dt} I_M)$$

where the RE u_d follow a an simultaneously autoregressive (SAR) process with unknown autoregression parameter $-1 \leq \rho \leq +1$ with W is the proximity matrix

$$u_{1d} = \rho_1 W u_{1d} + \varepsilon_d$$

with temporal dynamics with independent (I.ID.) effects across time

$$u_{2dt} = \rho_2 u_{2dt-1} + \nu_{dt}$$

(with $\nu \sim WN(0, \Sigma_{2d})$)

```
m_STFH_ar1 <- pbmseSTFH(formula = model_formula_sp,B = 50,
                        D = nD, T = nT,
                        vardir = VarDir_Y,
                        proxmat = contmat_q,
                        model = "ST",
                        theta_iter=TRUE, MAXITER = 1000, PRECISION = 10^(-6),
                        data = Data_ST)
```

```
##
## Bootstrap procedure with B = 50 iterations starts.
## b = 1
## b = 2
## b = 3
## b = 4
## b = 5
## b = 6
## b = 7
## b = 8
## b = 9
## b = 10
## b = 11
## b = 12
## b = 13
## b = 14
## b = 15
## b = 16
## b = 17
## b = 18
## b = 19
## b = 20
## b = 21
## b = 22
## b = 23
## b = 24
## b = 25
## b = 26
## b = 27
## b = 28
## b = 29
## b = 30
## b = 31
## b = 32
## b = 33
## b = 34
## b = 35
## b = 36
## b = 37
## b = 38
## b = 39
## b = 40
```

```
## b = 41
## b = 42
## b = 43
## b = 44
## b = 45
## b = 46
## b = 47
## b = 48
## b = 49
## b = 50
```

```
# Regression (FE) coefficients
STFH_ar1_coefs <- m_STFH_ar1$est$fit$estcoef
# EBLUP estimates
STFH_ar1_eblup <- m_STFH_ar1$est$eblup
STFH_ar1_eblup[STFH_ar1_eblup < 10^{-6}] <- 0
# EBLUP MSE
STFH_ar1_mse <- m_STFH_ar1$mse
STFH_ar1_mse[STFH_ar1_mse < 10^{-6}] <- 0
STFH_ar1_eblup_cv <- 100*sqrt(STFH_ar1_mse)/ Data_ST[[Yname]]


# Output
Data_ST$STFH_ar1_eblup <- as.vector(STFH_ar1_eblup)
Data_ST$STFH_ar1_eblup_cv <- as.vector(STFH_ar1_eblup_cv)
Data_ST$STFH_ar1_mse <- as.vector(STFH_ar1_mse)
```

Models comparison and selection using likelihood criteria

```
LikCrits <- matrix(data = NA, nrow = 4, ncol = 3)
colnames(LikCrits) <- c("LogLike","AIC","BIC")
rownames(LikCrits) <- c("FH","SFH","STFH","STFH_ar1")
LikCrits[1,] <- m_FH$est$fit$goodness[1:3]
LikCrits[2,] <- m_SFH$est$fit$goodness[1:3]
LikCrits[3,] <- m_STFH$est$fit$goodness[1:3]
LikCrits[4,] <- m_STFH_ar1$est$fit$goodness[1:3]

round(LikCrits,3) %>%
  knitr::kable()
```

	LogLike	AIC	BIC
FH	9.918	-7.836	3.391
SFH	10.408	-8.815	2.412
STFH	77.170	-140.340	-115.975
STFH_ar1	77.421	-138.842	-110.997

Graphical analysis of estimated values

Extract estimates

```
Output_SFH <- Output_SFH %>%
  dplyr::select(Year,Agrarian_SubRegion,contains("SFH"))
```

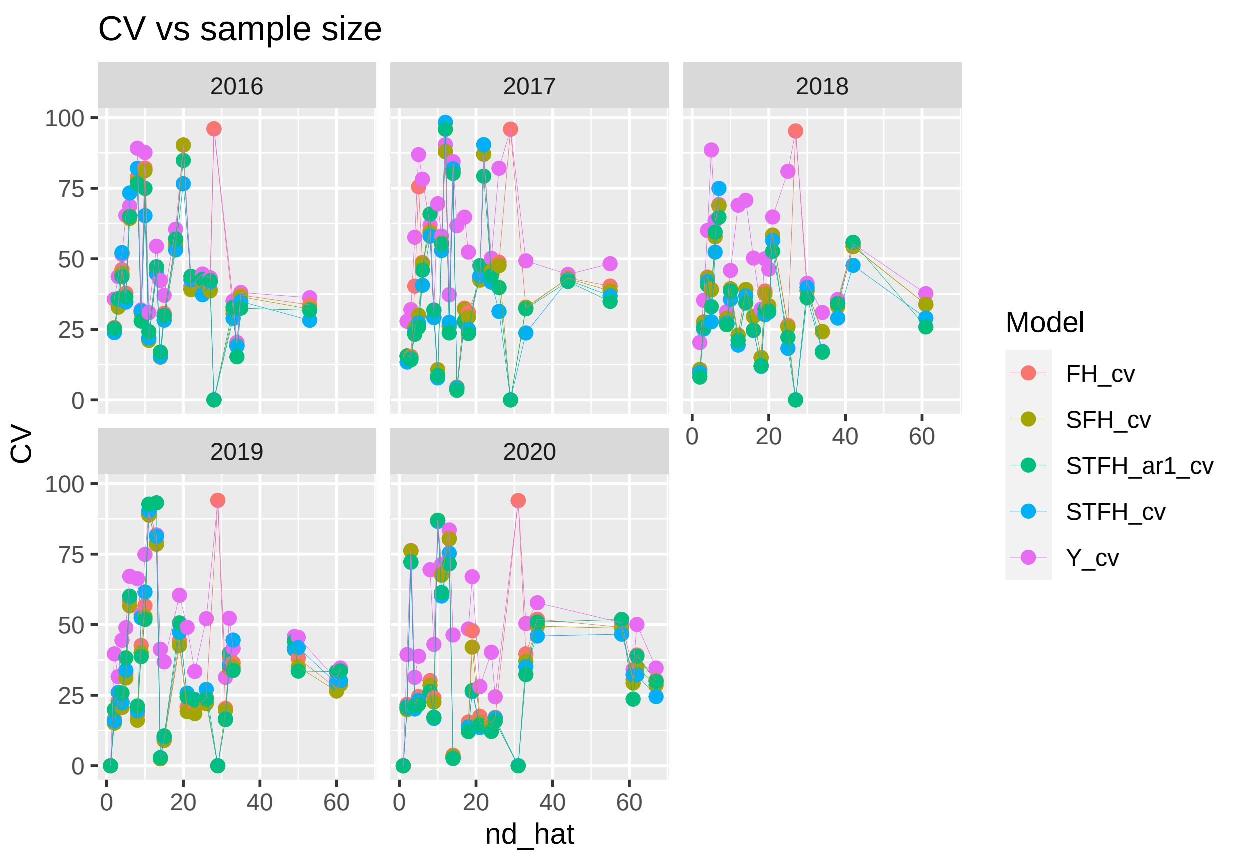
```r
Output_ST <- Data_ST %>%
  dplyr::select(Year,Agrarian_SubRegion,contains("STFH"))
Output <- left_join(x = Output_FH, y = Output_SFH, by = c("Agrarian_SubRegion","Year"))
Output <- left_join(x = Output, y = Output_ST, by = c("Agrarian_SubRegion","Year"))

# Transforming to total manure processed (thousands of tonns)
Output <- Output %>%
  mutate(Y = Y * Nd_hat,
         FH_eblup = FH_eblup * Nd_hat,
         SFH_eblup = SFH_eblup * Nd_hat,
         STFH_eblup = STFH_eblup * Nd_hat,
         STFH_ar1_eblup = STFH_ar1_eblup * Nd_hat)

# Prediction dataset building
Data_pred <- Data_full %>%
  dplyr::select(Agrarian_SubRegion,Year,NH3_AgricLive,PM10_mean,PM2.5_mean)
Data_pred <- left_join(x = Data_pred, y = Output, by = c("Agrarian_SubRegion","Year"))
Data_pred <- left_join(x = Data_pred, y = Shape_coord, by = c("Agrarian_SubRegion"))
Data_pred <- Data_pred %>%
  mutate(mean_eblup = (FH_eblup+SFH_eblup+STFH_eblup+STFH_ar1_eblup)/4,
         manure_anim = mean_eblup / BDN_Anim)
Data_pred <- Data_pred %>%
  st_as_sf() %>%
  st_transform(crs = 4326)
```

Variability of the estimates

Coefficient of variability as a function of sample size

```r
Output %>%
  group_by(nd_hat,Year) %>%
  dplyr::summarise(Y_cv = mean(Y_cv,na.rm=T),
                   FH_cv = mean(FH_eblup_cv,na.rm=T),
                   SFH_cv = mean(SFH_eblup_cv,na.rm=T),
                   STFH_cv = mean(STFH_eblup_cv,na.rm=T),
                   STFH_ar1_cv = mean(STFH_ar1_eblup_cv,na.rm=T)) %>%
  pivot_longer(cols = c(Y_cv,FH_cv,SFH_cv,STFH_cv,STFH_ar1_cv),
               names_to = "Model", values_to = "CV") %>%
  ggplot(mapping = aes(x = nd_hat)) +
  geom_point(mapping = aes(y = CV, col = Model), size = 2) +
  geom_line(mapping = aes(y = CV, col = Model), size = 0.05) +
  facet_wrap(~ Year) +
  labs(title = "CV vs sample size")
```

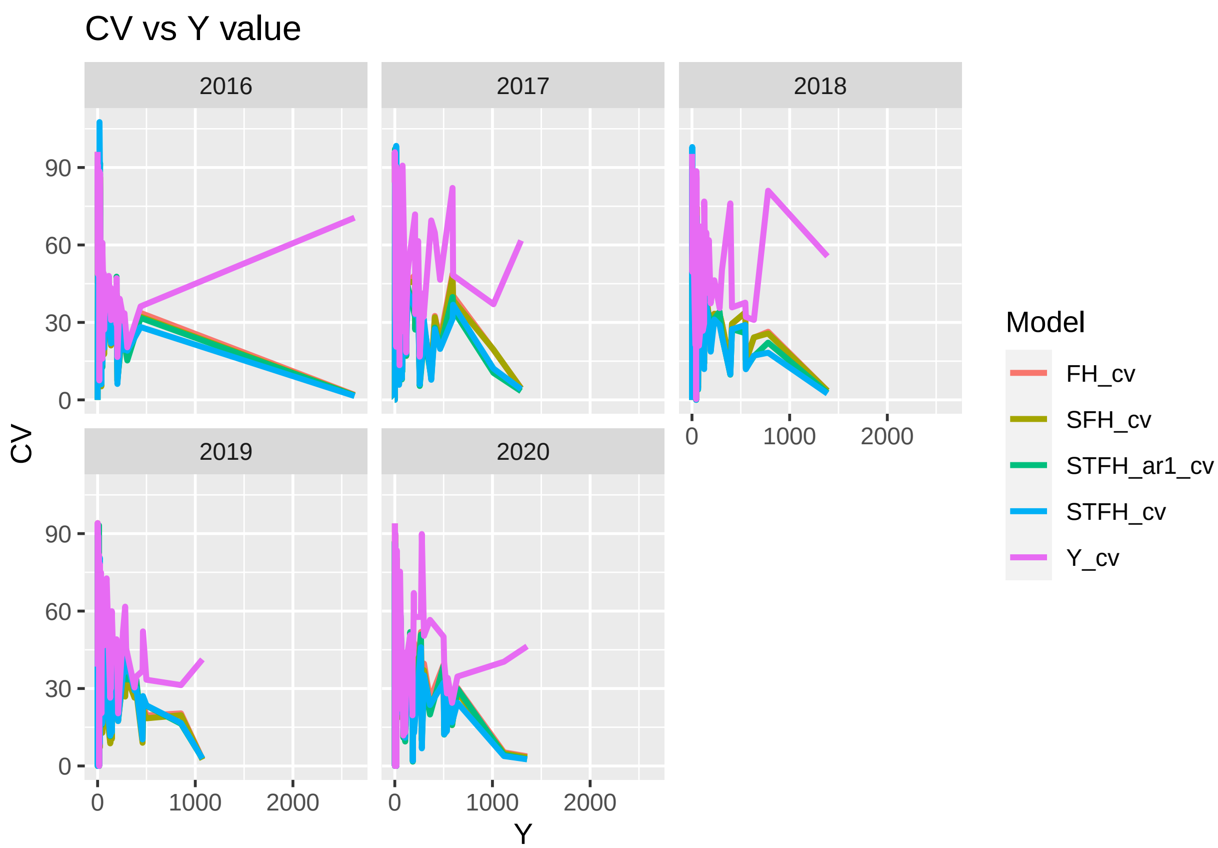
```
## `summarise()` has grouped output by 'nd_hat'. You can override using the
## `.groups` argument.

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: Removed 35 rows containing missing values (`geom_point()`).
```




```
Output %>%
  group_by(Agrarian_SubRegion,Year) %>%
  dplyr::summarise(Y = mean(Y, na.rm=T),
                   Y_cv = mean(Y_cv,na.rm=T),
                   FH_cv = mean(FH_eblup_cv,na.rm=T),
                   SFH_cv = mean(SFH_eblup_cv,na.rm=T),
                   STFH_cv = mean(STFH_eblup_cv,na.rm=T),
                   STFH_ar1_cv = mean(STFH_ar1_eblup_cv,na.rm=T)) %>%
  pivot_longer(cols = c(Y_cv,FH_cv,SFH_cv,STFH_cv,STFH_ar1_cv),
               names_to = "Model", values_to = "CV") %>%
  ggplot(mapping = aes(x = Y)) +
  geom_line(mapping = aes(y = CV, col = Model), size = 1.05) +
  facet_wrap(~ Year) +
  labs(title = "CV vs Y value")
```

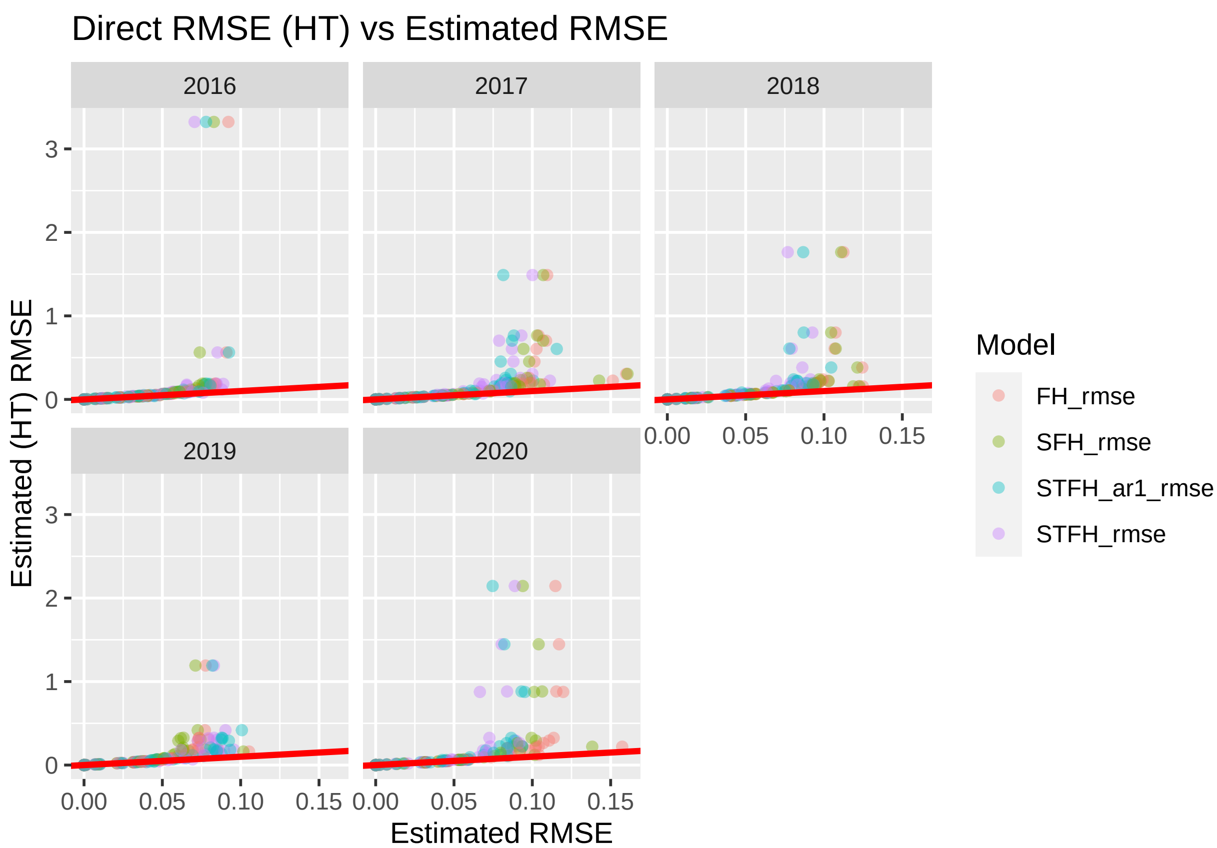
```
## `summarise()` has grouped output by 'Agrarian_SubRegion'. You can override
## using the `.groups` argument.
```

```
## Warning: Removed 20 rows containing missing values (`geom_line()`).
```




```
Output %>%
  group_by(Agrarian_SubRegion,Year) %>%
  dplyr::summarise(Y_rmse = mean(sqrt(VarDir_Y),na.rm=T),
            FH_rmse = mean(sqrt(FH_mse),na.rm=T),
            SFH_rmse = mean(sqrt(SFH_mse),na.rm=T),
            STFH_rmse = mean(sqrt(STFH_mse),na.rm=T),
            STFH_ar1_rmse = mean(sqrt(STFH_ar1_mse),na.rm=T)) %>%
  pivot_longer(cols = c(FH_rmse,SFH_rmse,STFH_rmse,STFH_ar1_rmse),
               names_to = "Model", values_to = "RMSE") %>%
  ggplot(mapping = aes(y = Y_rmse)) +
  geom_point(mapping = aes(x = RMSE, col = Model), alpha = 0.4) +
  geom_abline(slope = 1, intercept = 0, col = "red", size = 1.1) +
  facet_wrap(~ Year) +
  labs(y = "Estimated (HT) RMSE", x = "Estimated RMSE",
       title = "Direct RMSE (HT) vs Estimated RMSE")
```

```
## `summarise()` has grouped output by 'Agrarian_SubRegion'. You can override
## using the `.groups` argument.
```

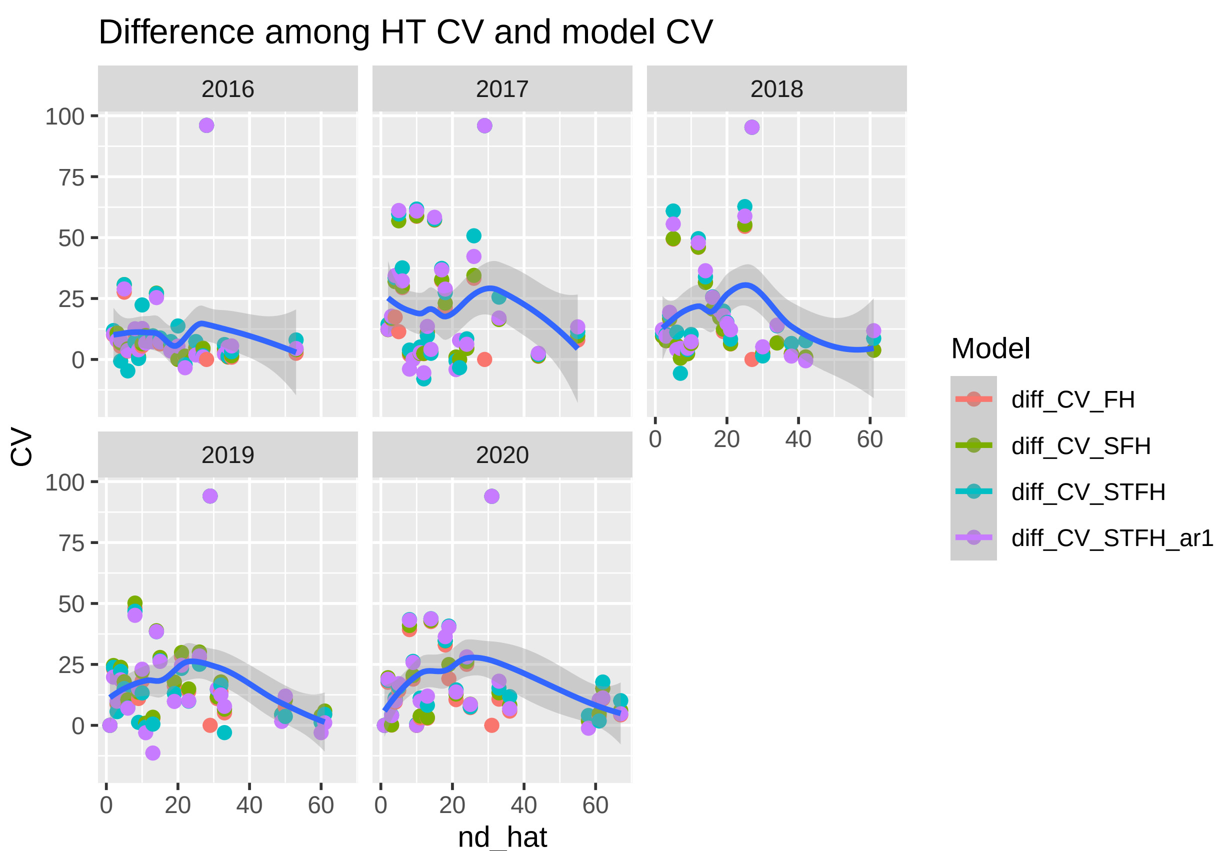



```
Output %>%
  group_by(nd_hat,Year) %>%
  dplyr::summarise(diff_CV_FH = mean(Y_cv - FH_eblup_cv,na.rm=T),
            diff_CV_SFH = mean(Y_cv - SFH_eblup_cv,na.rm=T),
            diff_CV_STFH = mean(Y_cv - STFH_eblup_cv,na.rm=T),
            diff_CV_STFH_ar1 = mean(Y_cv - STFH_ar1_eblup_cv,na.rm=T)) %>%
  pivot_longer(cols = c(diff_CV_FH,diff_CV_SFH,diff_CV_STFH,diff_CV_STFH_ar1),
               names_to = "Model", values_to = "CV") %>%
  ggplot(mapping = aes(x = nd_hat, group = 1)) +
  geom_point(mapping = aes(y = CV, col = Model), size = 2) +
  geom_smooth(mapping = aes(y = CV, col = Model)) +
  facet_wrap(~ Year) +
  labs(title = "Difference among HT CV and model CV")
```

```
## `summarise()` has grouped output by 'nd_hat'. You can override using the
## `.groups` argument.
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

## Warning: Removed 28 rows containing non-finite values (`stat_smooth()`).

## Warning: The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
```

```
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: colour
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

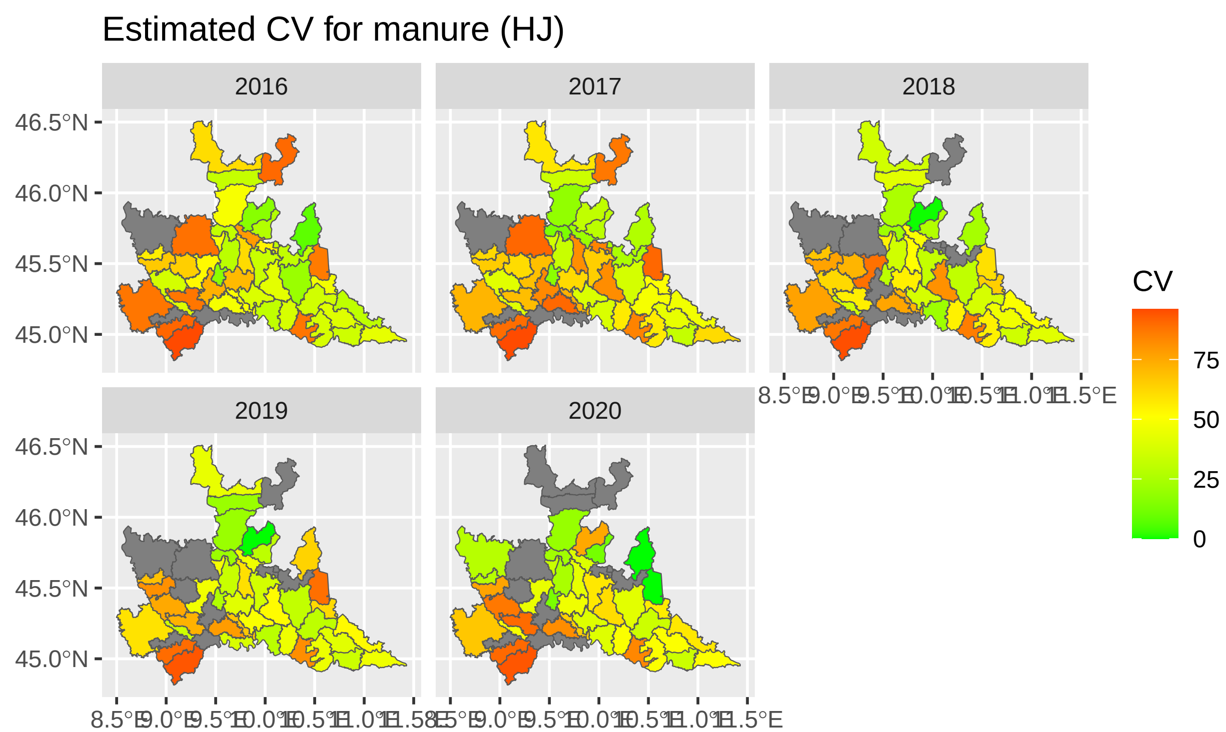
```
## Warning: Removed 28 rows containing missing values (`geom_point()`).
```



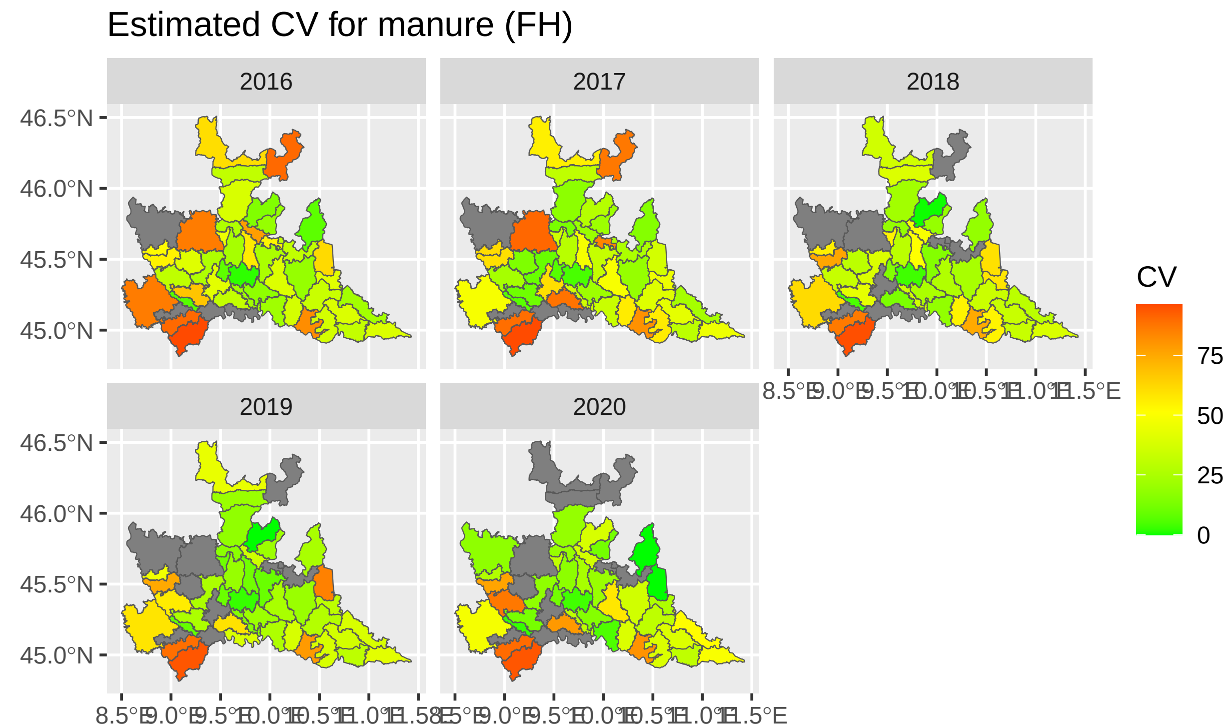
Coefficient of variability across space

```
##### CV spatial plot
# Direct prediction (Hajek estimator)
Data_pred %>%
```

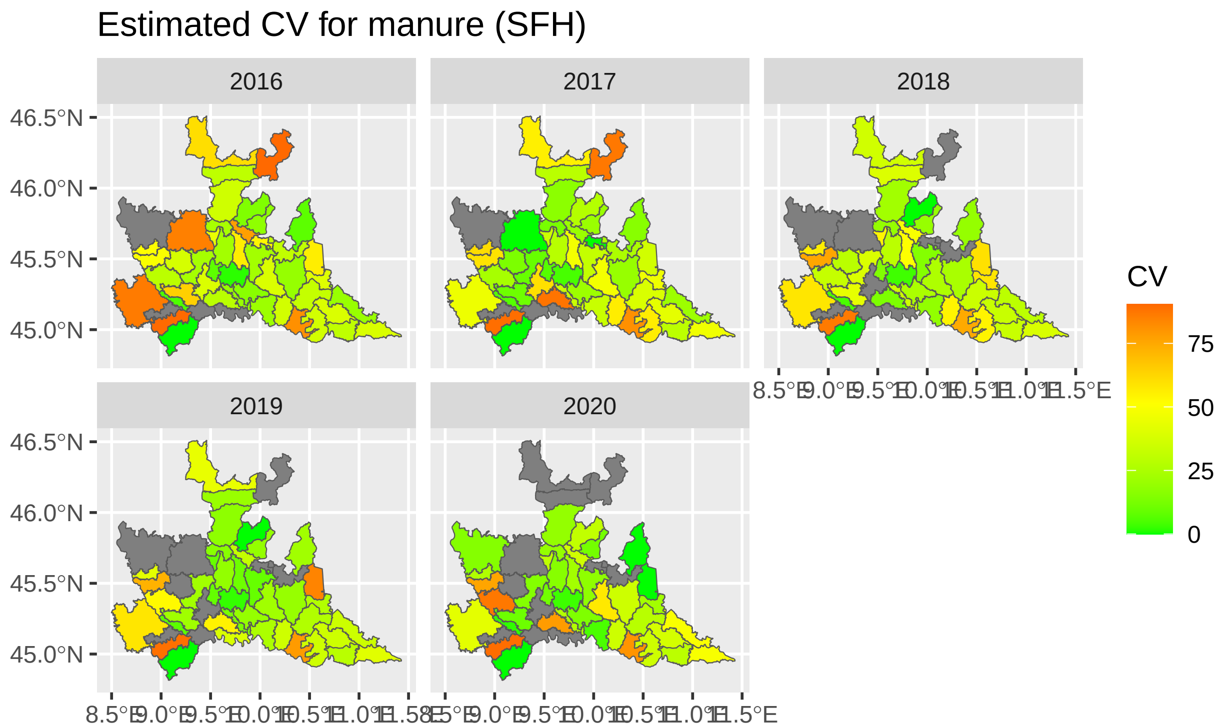
```
  ggplot() +
  geom_sf(mapping = aes(fill = Y_cv)) +
  facet_wrap(~ Year) +
  scale_fill_gradient2(name = "CV",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y_cv,na.rm=T)
  labs(title = "Estimated CV for manure (HJ)")
```



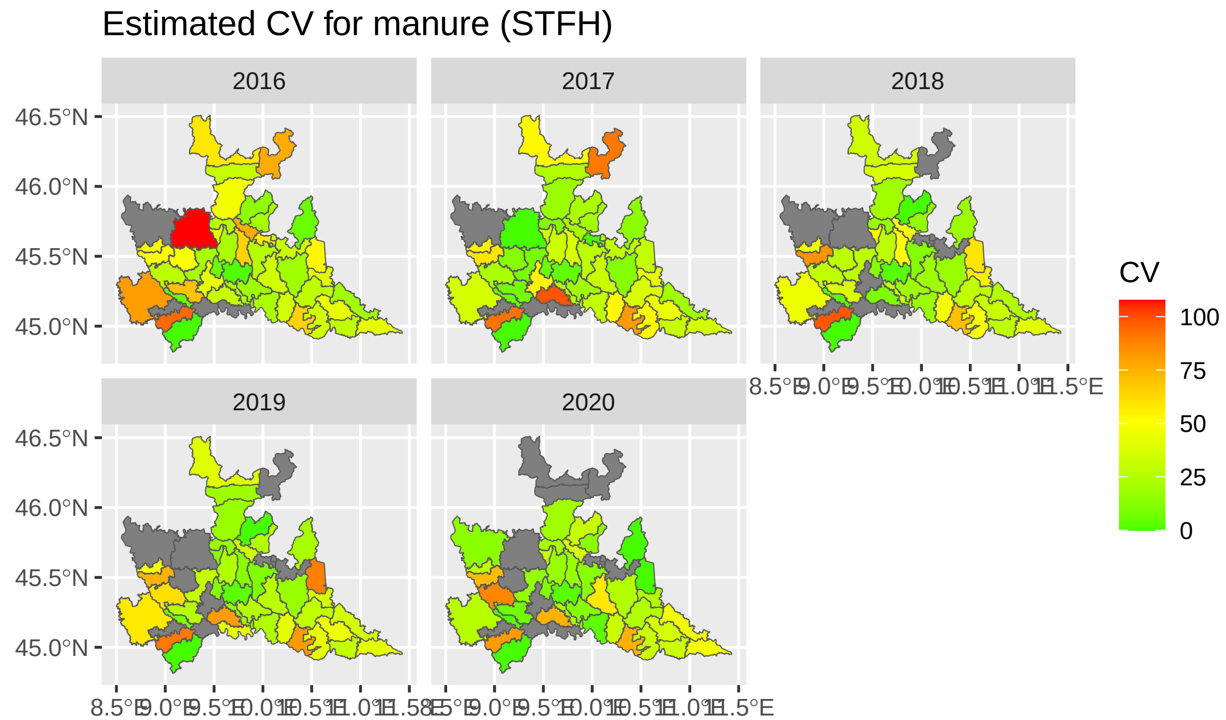

```
# Modelled prediction: Fay-Herriott estimator
Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = FH_eblup_cv)) +
  facet_wrap(~ Year) +
  scale_fill_gradient2(name = "CV",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y_cv,na.rm=T)
  labs(title = "Estimated CV for manure (FH)")
```



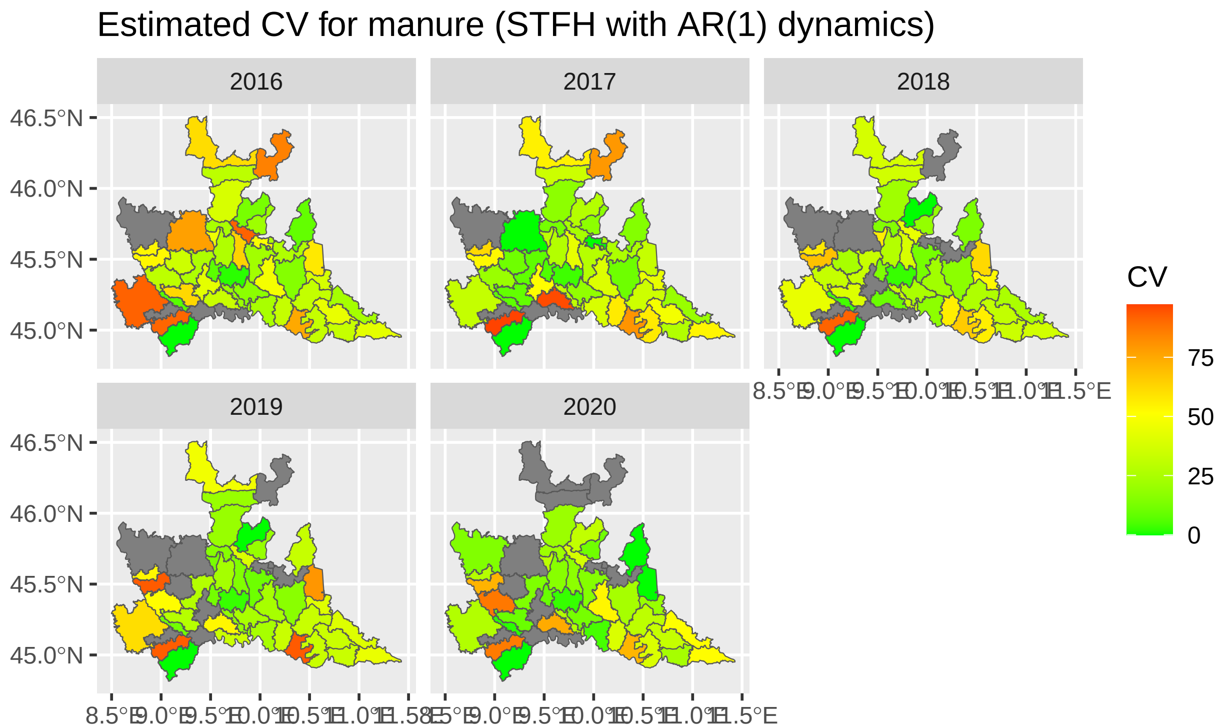

```r
# Modelled prediction: spatial Fay-Herriott estimator
Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = SFH_eblup_cv)) +
  facet_wrap(~ Year) +
  scale_fill_gradient2(name = "CV",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y_cv,na.rm=T)
  labs(title = "Estimated CV for manure (SFH)")
```




```
# Modelled prediction: spatio-temporal Fay-Herriott estimator
Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = STFH_eblup_cv)) +
  facet_wrap(~ Year) +
  scale_fill_gradient2(name = "CV",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y_cv,na.rm=T)
  labs(title = "Estimated CV for manure (STFH)")
```




```
# Modelled prediction: spatio-temporal Fay-Herriott estimator with AR(1) dynamics
Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = STFH_ar1_eblup_cv)) +
  facet_wrap(~ Year) +
  scale_fill_gradient2(name = "CV",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y_cv,na.rm=T)
  labs(title = "Estimated CV for manure (STFH with AR(1) dynamics)")
```

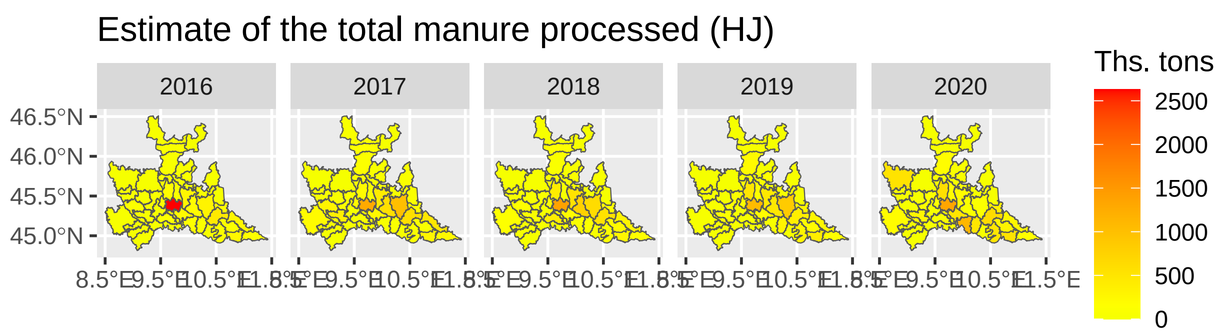


Point estimates (prediction spatial plot)

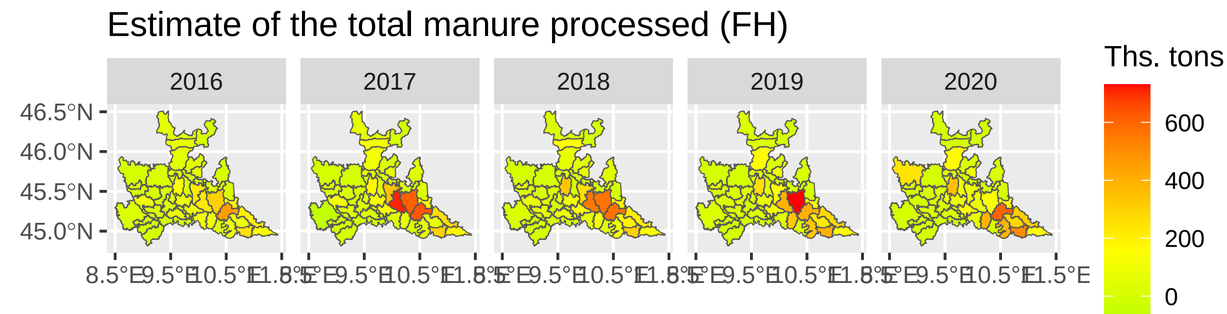
```
# Non-modelled prediction (Hajek estimator)
p0 <- Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = Y)) +
  scale_x_continuous(breaks = c(8.5,9.5,10.5,11.5), labels = paste0(c(8.5,9.5,10.5,11.5),'°E')) +
  facet_wrap(~ Year, ncol = 5) +
  scale_fill_gradient2(name = "Ths. tons",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y,na.rm=T)) +
  labs(title = "Estimate of the total manure processed (HJ)")
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank(),
        plot.margin=unit(c(1,1,-0.5,1), "cm"))
```

```
## List of 3
##  $ axis.text.x : list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ axis.ticks.x: list()
##   ..- attr(*, "class")= chr [1:2] "element_blank" "element"
##  $ plot.margin : 'simpleUnit' num [1:4] 1cm 1cm -0.5cm 1cm
##   ..- attr(*, "unit")= int 1
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

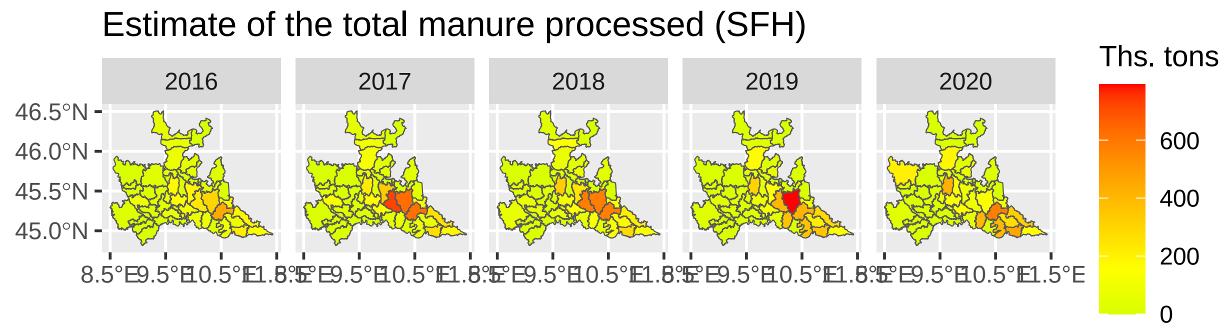
```
p0
```




```
# Modelled prediction: Fay-Herriott estimator
Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = FH_eblup)) +
  scale_x_continuous(breaks = c(8.5,9.5,10.5,11.5), labels = paste0(c(8.5,9.5,10.5,11.5),'°E')) +
  facet_wrap(~ Year, ncol = 5) +
  scale_fill_gradient2(name = "Ths. tons",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y,na.rm=T)) +
  labs(title = "Estimate of the total manure processed (FH)")
```



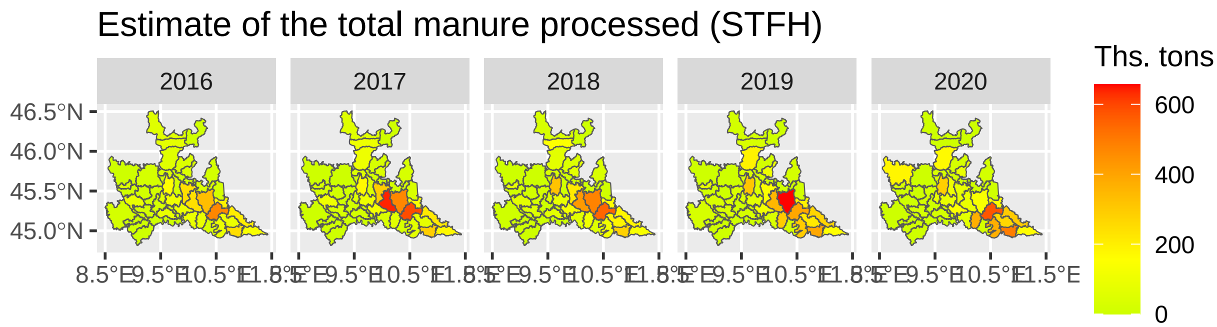

```r
# Modelled prediction: spatial Fay-Herriott estimator
Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = SFH_eblup)) +
  scale_x_continuous(breaks = c(8.5,9.5,10.5,11.5), labels = paste0(c(8.5,9.5,10.5,11.5),'°E')) +
  facet_wrap(~ Year,ncol = 5) +
  scale_fill_gradient2(name = "Ths. tons",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y,na.rm=T)) +
  labs(title = "Estimate of the total manure processed (SFH)")
```



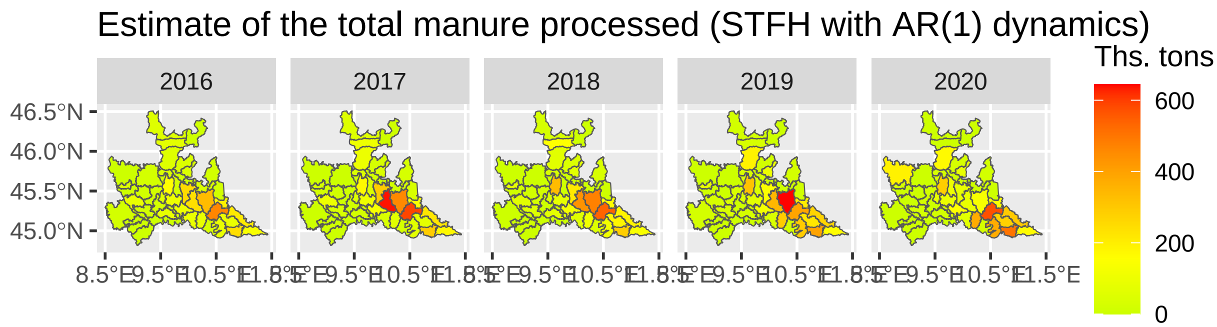

```r
# Modelled prediction: spatio-temporal Fay-Herriott estimator
p1 <- Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = STFH_eblup)) +
  scale_x_continuous(breaks = c(8.5,9.5,10.5,11.5), labels = paste0(c(8.5,9.5,10.5,11.5),'°E')) +
  facet_wrap(~ Year, ncol = 5) +
  scale_fill_gradient2(name = "Ths. tons",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y,na.rm=T)) +
  labs(title = "Estimate of the total manure processed (STFH)")
  theme(plot.margin=unit(c(-0.5,1,1,1), "cm"))
```

```
## List of 1
##  $ plot.margin: 'simpleUnit' num [1:4] -0.5cm 1cm 1cm 1cm
##   ..- attr(*, "unit")= int 1
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

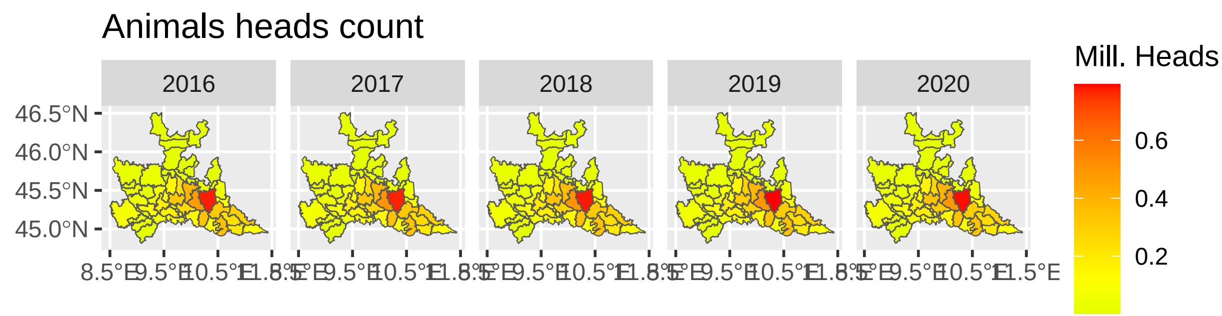
```r
p1
```




```r
# Modelled prediction: spatio-temporal Fay-Herriott estimator with AR(1) dynamics
Data_pred %>%
  ggplot() +
  geom_sf(mapping = aes(fill = STFH_ar1_eblup)) +
  scale_x_continuous(breaks = c(8.5,9.5,10.5,11.5), labels = paste0(c(8.5,9.5,10.5,11.5),'°E')) +
  facet_wrap(~ Year,ncol = 5) +
  scale_fill_gradient2(name = "Ths. tons",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$Y,na.rm=T)) +
  labs(title = "Estimate of the total manure processed (STFH with AR(1) dynamics)")
```




```r
# Animals
Data_pred %>%
  mutate(BDN_Anim = BDN_Anim/1000000) %>%
  ggplot() +
  geom_sf(mapping = aes(fill = BDN_Anim)) +
  scale_x_continuous(breaks = c(8.5,9.5,10.5,11.5), labels = paste0(c(8.5,9.5,10.5,11.5),'°E')) +
  facet_wrap(~ Year, ncol = 5) +
  scale_fill_gradient2(name = "Mill. Heads",
                       low="green", high="red", mid = "yellow", midpoint = mean(Data_pred$BDN_Anim/10000
  labs(title = "Animals heads count")
```




```
p_preds <- ggpubr::ggarrange(p0,p1, ncol = 1) +
  theme(plot.margin=margin(1,0.1,0.1,0.1,"cm"))
ggpubr::ggexport(p_preds,filename = "EstimHJ_STFH.png", width = 2400, height = 1000, res = 160,ncol = 1
```

```
## file saved to EstimHJ_STFH.png
```

```
p2 <- Data_pred %>%
  dplyr::select(Agrarian_SubRegion,Year,NH3_AgricLive,EBLUP = STFH_eblup) %>%
  mutate(Year = as.factor(Year)) %>%
  ggplot(mapping = aes(y = NH3_AgricLive)) +
  geom_point(mapping = aes(x = EBLUP, col = Year)) +
  geom_smooth(mapping = aes(x = EBLUP, col = as.factor(Year)), se = F) +
  labs(title = "NH3 from manure management vs EBLUP from STFH model",
       y = "NH3 emissions (kg/hectar)", x = "EBLUP of manure processed (Ths. Tons)")
ggpubr::ggexport(p2,filename = "EDANH3.png", width = 2400, height = 1000, res = 160)
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## file saved to EDANH3.png
```

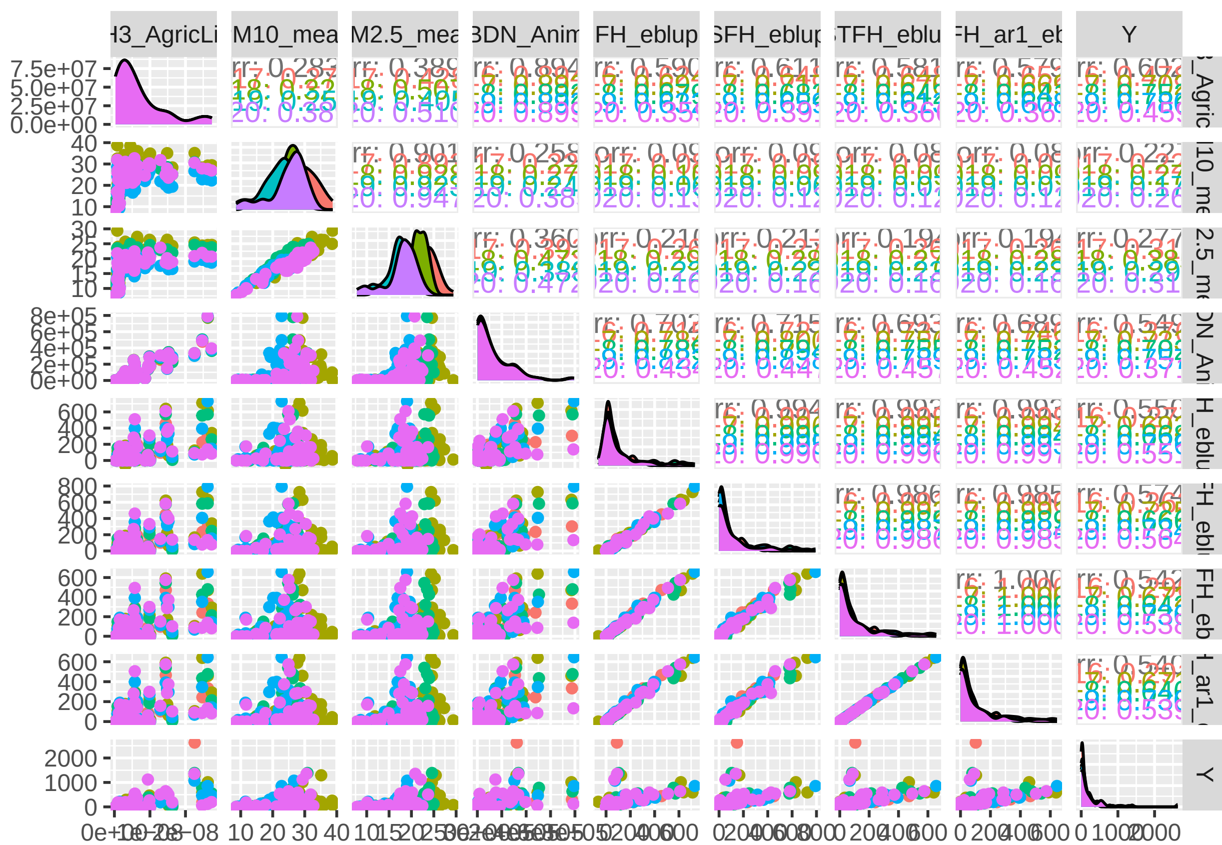
Correlation analysis

```
Data_pred %>%
  dplyr::select(Agrarian_SubRegion,Year,NH3_AgricLive,PM10_mean,PM2.5_mean,BDN_Anim,FH_eblup,SFH_eblup,S
  GGally::ggpairs(columns = 3:11, ggplot2::aes(colour=as.factor(Year)))
```

```
## Registered S3 method overwritten by 'GGally':
```

```
##   method from
##   +.gg   ggplot2

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning: Removed 48 rows containing missing values (`geom_point()`).

## Warning: Removed 48 rows containing non-finite values (`stat_density()`).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning: Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).

## Warning: Removed 48 rows containing non-finite values (`stat_density()`).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 48 rows containing missing values

## Warning: Removed 48 rows containing missing values (`geom_point()`).
```

```
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
## Removed 48 rows containing missing values (`geom_point()`).
```



Estimating the empirical relationship between manure and NH3 emissions

```
library(mgcv)
```

```
## Caricamento del pacchetto richiesto: nlme
##
## Caricamento pacchetto: 'nlme'
## Il seguente oggetto è mascherato da 'package:dplyr':
##
##     collapse
```

```
## Il seguente oggetto è mascherato da 'package:lme4':
##
##     lmList
```

```
## This is mgcv 1.8-41. For overview type 'help("mgcv-package")'.
```

```r
mod0 <- lm(NH3_AgricLive ~ STFH_ar1_eblup + Latitude + Longitude + Latitude*Longitude + Year, data = Da
mod1 <- glm(NH3_AgricLive ~ STFH_ar1_eblup +  Latitude + Longitude + Latitude*Longitude + Year, data = 
            family = Gamma(link = "log"))
mod2 <- gam(NH3_AgricLive ~ STFH_ar1_eblup +
              s(Latitude,Longitude,bs = "tp") +
              Year,
            method = "REML",
            data = Data_pred)
mod3 <- gam(NH3_AgricLive ~ STFH_ar1_eblup +
              s(Latitude,Longitude,bs = "tp") +
              Year,
            family = Gamma(link = "log"),
            method = "REML",
            data = Data_pred)
mod4 <- gam(NH3_AgricLive ~ s(STFH_ar1_eblup) +
              s(Latitude,Longitude,bs = "tp") +
              Year,
            method = "REML",
            data = Data_pred)
mod5 <- gam(NH3_AgricLive ~ s(STFH_ar1_eblup) +
              s(Latitude,Longitude,bs = "tp") +
              Year,
            family = Gamma(link = "log"),
            method = "REML",
            data = Data_pred)

performance::performance(mod0)
```

```
## # Indices of model performance
##
## AIC       |      AICc |       BIC |    R2 | R2 (adj.) |      RMSE |     Sigma
## ------------------------------------------------------------------------------
## -8425.074 | -8424.591 | -8400.709 | 0.356 |     0.342 | 5.601e-09 | 5.672e-09
```

```r
performance::performance(mod1)
```

```
## # Indices of model performance
##
## AIC       |      AICc |       BIC | Nagelkerke's R2 |      RMSE | Sigma
## ------------------------------------------------------------------------
## -8622.193 | -8621.710 | -8597.828 |           0.398 | 7.547e-09 | 0.882
```

```r
performance::performance(mod2)
```

```
## # Indices of model performance
##
## AIC       |      AICc |       BIC |    R2 |      RMSE |     Sigma
## ------------------------------------------------------------------
## -8780.807 | -8770.775 | -8670.312 | 0.864 | 2.408e-09 | 2.586e-09
```

```r
performance::performance(mod3)
```

```
## Warning in mean.default(get_data(x, verbose = FALSE)[[smooth]], na.rm = TRUE):
## l'argomento non è numerico o logico: si restituisce NA

## Warning in mean.default(get_data(x, verbose = FALSE)[[smooth]], na.rm = TRUE):
## l'argomento non è numerico o logico: si restituisce NA

## Warning in mean.default(get_data(x, verbose = FALSE)[[smooth]], na.rm = TRUE):
## l'argomento non è numerico o logico: si restituisce NA

## # Indices of model performance
##
## AIC       |      AICc |       BIC |    R2 |      RMSE | Sigma
## ---------------------------------------------------------------
## -9023.305 | -9013.558 | -8914.295 | 0.795 | 2.961e-09 | 0.386
```

```r
performance::performance(mod4)
```

```
## # Indices of model performance
##
## AIC       |      AICc |       BIC |    R2 |      RMSE |     Sigma
## -------------------------------------------------------------------
## -8806.683 | -8794.598 | -8686.101 | 0.880 | 2.253e-09 | 2.468e-09
```

```r
library(texreg)
```

```
## Version:  1.38.6
## Date:     2022-04-06
## Author:   Philip Leifeld (University of Essex)
##
## Consider submitting praise using the praise or praise_interactive functions.
## Please cite the JSS article in your publications -- see citation("texreg").

##
## Caricamento pacchetto: 'texreg'

## Il seguente oggetto è mascherato da 'package:tidyr':
##
##     extract
```

```r
texreg(l = list(mod0,mod1,mod2,mod3,mod4,mod5),digits = 3,
       custom.model.names = c("LM","GLM (gamma)","GAM (norm)","GAM (gamma)","GAM (norm)", "GAM (gamma)"
```

```
##
## \usepackage{graphicx}
##
## \begin{table}
## \begin{center}
## \scalebox{0.5}{
## \begin{tabular}{l c c c c c c}
## \hline
##  & LM & GLM (gamma) & GAM (norm) & GAM (gamma) & GAM (norm) & GAM (gamma) \\
## \hline
## (Intercept)                & $0.000^{*}$   & $-153.385$    & $0.000$       & $-7.406$      & $0.00
##                            & $(0.000)$     & $(201.350)$   & $(0.000)$     & $(31.445)$    & $(0.0
## STFH\_ar1\_eblup           & $0.000^{***}$ & $0.003^{***}$ & $0.000^{***}$ & $0.002^{***}$ &
```

```
##                              & $(0.000)$     & $(0.001)$     & $(0.000)$       & $(0.000)$       &
## Latitude                     & $-0.000^{*}$  & $2.524$       &                 &                 &
##                              & $(0.000)$     & $(3.998)$     &                 &                 &
## Longitude                    & $-0.000^{*}$  & $16.171$      &                 &                 &
##                              & $(0.000)$     & $(18.475)$    &                 &                 &
## Year                         & $-0.000$      & $0.009$       & $-0.000$        & $-0.006$        & $-0.00
##                              & $(0.000)$     & $(0.043)$     & $(0.000)$       & $(0.016)$       & $(0.00
## Latitude:Longitude           & $0.000^{*}$   & $-0.354$      &                 &                 &
##                              & $(0.000)$     & $(0.408)$     &                 &                 &
## EDF: s(Latitude,Longitude) &                 &               & $27.540^{***}$ & $27.062^{***}$ & $27.63
##                              &               &               & $(28.859)$      & $(28.756)$      & $(28.8
## EDF: s(STFH\_ar1\_eblup)    &               &               &                 &                 & $3.380
##                              &               &               &                 &                 & $(4.19
## \hline
## R$^2$                        & $0.356$       & $$            & $0.864$         & $0.795$         & $0.880
## Adj. R$^2$                   & $0.342$       & $$            & $$              & $$              & $$
## Num. obs.                    & $240$         & $240$         & $240$           & $240$           & $240$
## AIC                          & $$            & $-8622.193$   & $-8780.813$     & $-9021.406$     & $-8800
## BIC                          & $$            & $-8597.828$   & $-8670.318$     & $-8912.395$     & $-8680
## Log Likelihood               & $$            & $4318.096$    & $4422.152$      & $4542.022$      & $4438
## Deviance                     & $$            & $182.126$     & $0.000$         & $30.987$        & $0.000
## Deviance explained           & $$            & $$            & $0.881$         & $0.878$         & $0.890
## Dispersion                   & $$            & $$            & $0.000$         & $0.116$         & $0.000
## GCV score                    & $$            & $$            & $-4247.141$     & $-4466.418$     & $-4262
## Num. smooth terms            & $$            & $$            & $1$             & $1$             & $2$
## \hline
## \multicolumn{7}{l}{\scriptsize{$^{***}p<0.001$; $^{**}p<0.01$; $^{*}p<0.05$}}
## \end{tabular}
## }
## \caption{Statistical models}
## \label{table:coefficients}
## \end{center}
## \end{table}
```