Basic area-level model

Small introduction

The `fh` function allows various estimation methods for the variance of the random effects and the mean squared error (MSE).

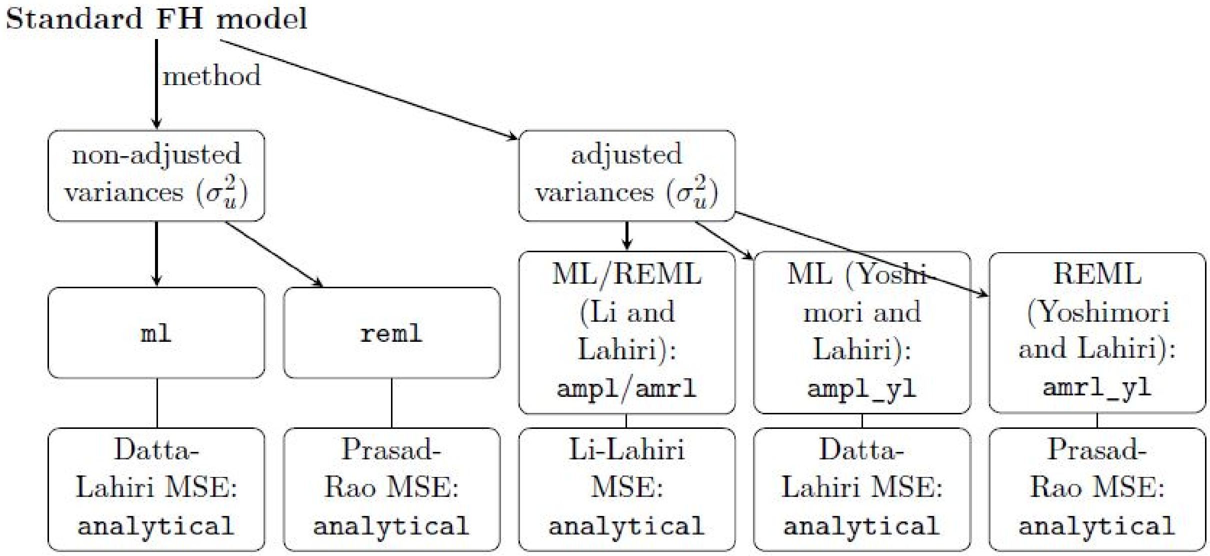

Standard FH model
method
non-adjusted variances (σ_u^2)
adjusted variances (σ_u^2)
ml
reml
ML/REML (Li and Lahiri): ampl/amrl
ML (Yoshimori and Lahiri): ampl_yl
REML (Yoshimori and Lahiri): amrl_yl
Datta-Lahiri MSE: analytical
Prasad-Rao MSE: analytical
Li-Lahiri MSE: analytical
Datta-Lahiri MSE: analytical
Prasad-Rao MSE: analytical

Figure 1: Estimation options for the basic area-level model

Load package and data

```r
library("emdi")
```

```
##
## Attaching package: 'emdi'

## The following object is masked from 'package:stats':
##
##     step
```

```r
data("eusilcA_popAgg")
data("eusilcA_smpAgg")
```

Combine inout data

```r
## Combine input data
combined_data <- combine_data(pop_data = eusilcA_popAgg,
                              pop_domains = "Domain",
                              smp_data = eusilcA_smpAgg,
                              smp_domains = "Domain")
```

Identify spatial structures

```
load_shapeaustria()
shape_austria_dis <- shape_austria_dis[order(shape_austria_dis$PB),]
austria_shape <- sp::merge(shape_austria_dis, eusilcA_smpAgg, by.x = "PB",
                       by.y = "Domain", all.x = F)
rel <- spdep::poly2nb(austria_shape, row.names = austria_shape$PB)
eusilcA_prox <- spdep::nb2mat(rel, style = "W", zero.policy = TRUE)

# Spatial correlation tests
spatialcor.tests(direct = combined_data$Mean,
                 corMatrix = eusilcA_prox)
```

```
##   Statistics     Value      p.value
## 1  Moran's I 0.2453677 5.607958e-05
## 2  Geary's C 0.6238681 2.473294e-03
```

Perform model selection

```
# Generate initial object of class "fh", "emdi"
fh_std <- fh(fixed = Mean ~ cash + self_empl + unempl_ben,
             vardir = "Var_Mean", combined_data = combined_data,
             domains = "Domain", method = "ml", B = c(0,50))
# Perform stepwise variable selection
step(fh_std, criteria = "KICb2")
```

```
## Start: KICb2 = 1709.42
## Mean ~ cash + self_empl + unempl_ben
```

```
##               df  KICb2
## - unempl_ben  1 1708.3
## <none>          1709.4
## - self_empl   1 1763.0
## - cash        1 1808.6
```

```
##
## Step: KICb2 = 1708.33
## Mean ~ cash + self_empl
```

```
##              df  KICb2
## <none>         1708.3
## - self_empl  1 1765.3
## - cash       1 1816.1
```

```
##
## Call:
##  fh(fixed = Mean ~ cash + self_empl, vardir = "Var_Mean", combined_data = combined_data,
##     domains = "Domain", method = "ml", B = c(0, 50))
##
```

```
## Coefficients:
##               coefficients  std.error t.value   p.value    
## (Intercept)   3070.51231  635.94290  4.8283 1.377e-06 ***
## cash             1.05939    0.07049 15.0288 < 2.2e-16 ***
## self_empl        1.74564    0.22017  7.9284 2.219e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Estimate EBLUPs and MSEs

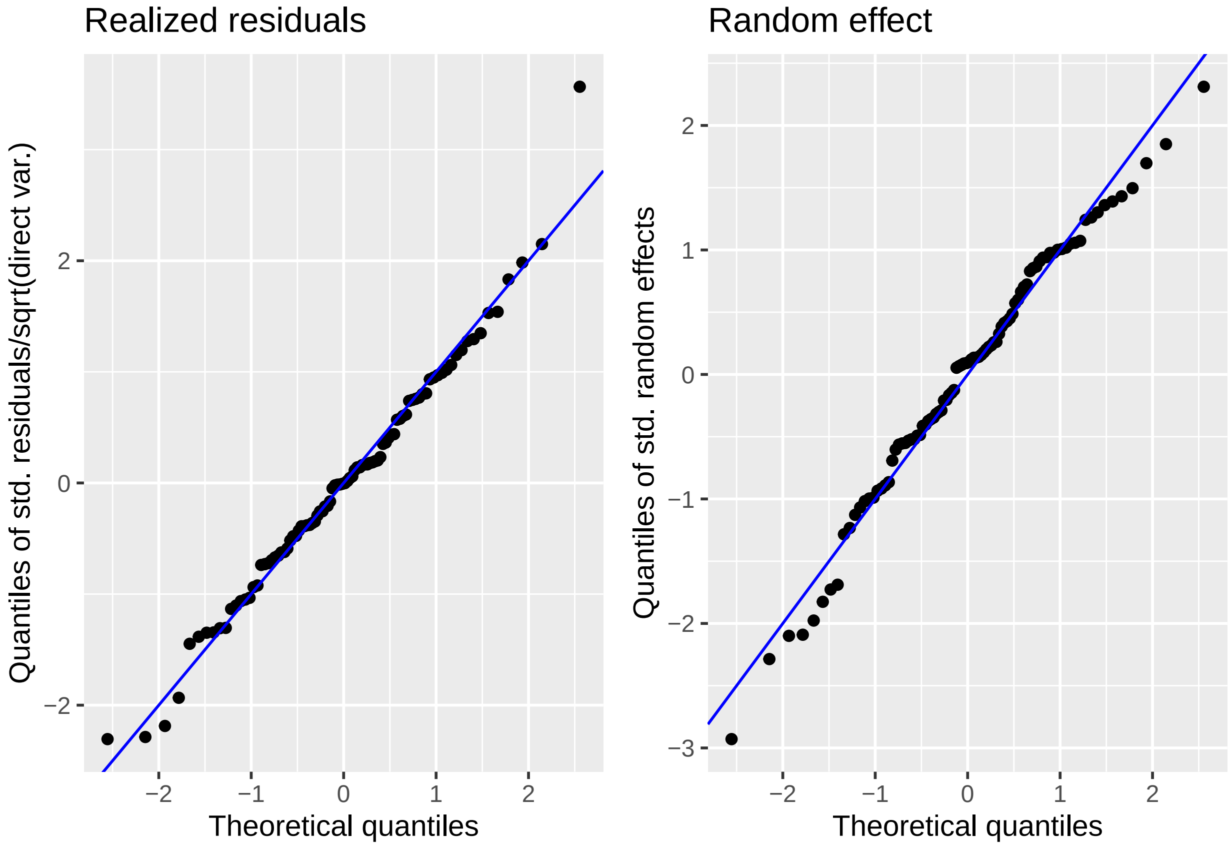
```
# Generate object of class "fh","emdi" with MSE estimation
fh_std <- fh(fixed = Mean ~ cash + self_empl, vardir = "Var_Mean",
             combined_data = combined_data, domains = "Domain",
             method = "ml", MSE = TRUE, B = c(0,50))
```

Assess the estimated model

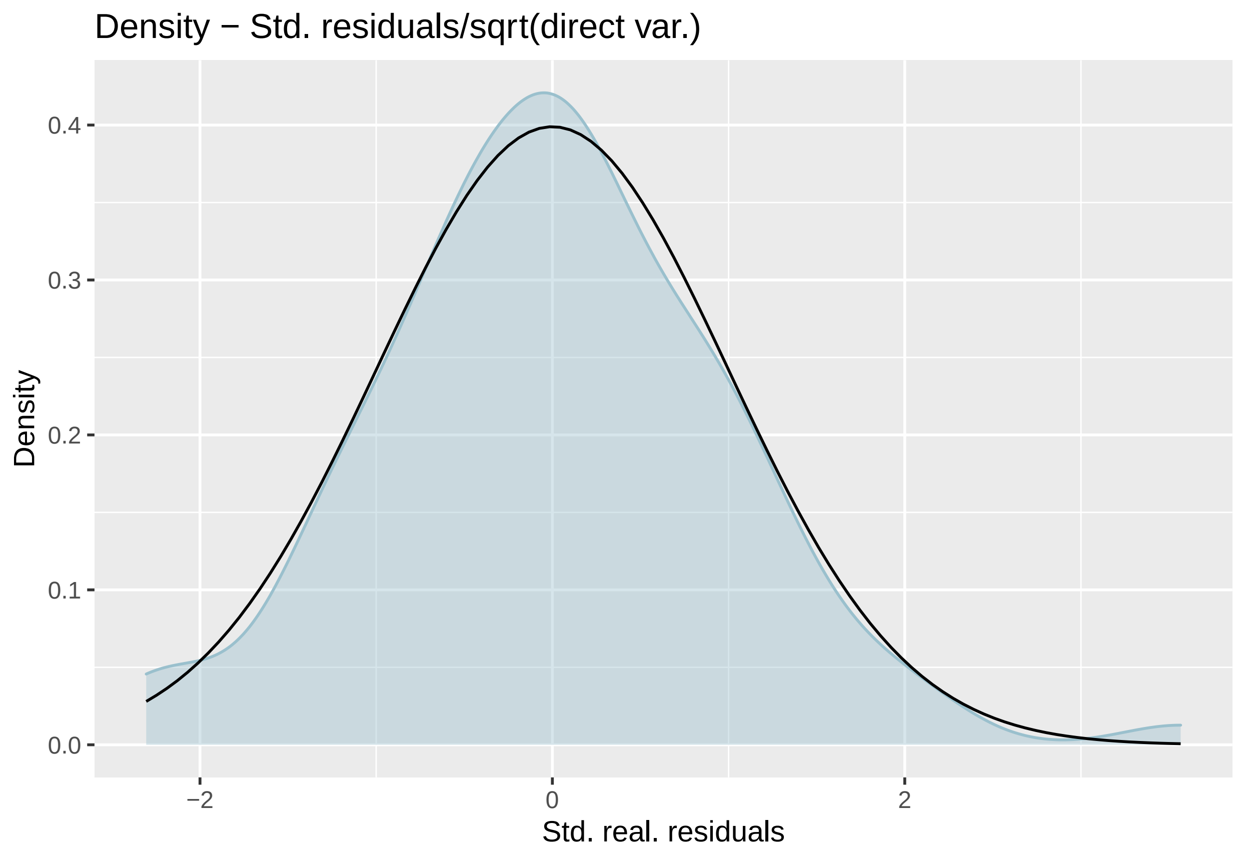
```
summary(fh_std)
```

```
## Call:
##  fh(fixed = Mean ~ cash + self_empl, vardir = "Var_Mean", combined_data = combined_data,
##     domains = "Domain", method = "ml", MSE = TRUE, B = c(0, 50))
##
## Out-of-sample domains:  0
## In-sample domains:  94
##
## Variance and MSE estimation:
## Variance estimation method:  ml
## Estimated variance component(s):  1371195
## MSE method:  datta-lahiri
##
## Coefficients:
##              coefficients  std.error t.value   p.value    
## (Intercept)   3070.51231  635.94290  4.8283 1.377e-06 ***
## cash             1.05939    0.07049 15.0288 < 2.2e-16 ***
## self_empl        1.74564    0.22017  7.9284 2.219e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Explanatory measures:
##     loglike      AIC    AICc    AICb1    AICb2      BIC      KIC     KICc
## 1 -847.8303 1703.661 1703.91 1715.758 1703.461 1713.834 1707.661 1708.783
##      KICb1    KICb2        R2     AdjR2
## 1 1720.632 1708.335 0.9212817 0.9482498
##
## Residual diagnostics:
##                          Skewness Kurtosis Shapiro_W Shapiro_p
## Standardized_Residuals  0.3004662 3.971216 0.9840810 0.3119346
## Random_effects         -0.4113238 3.086048 0.9839858 0.3072834
##
## Transformation: No transformation
```

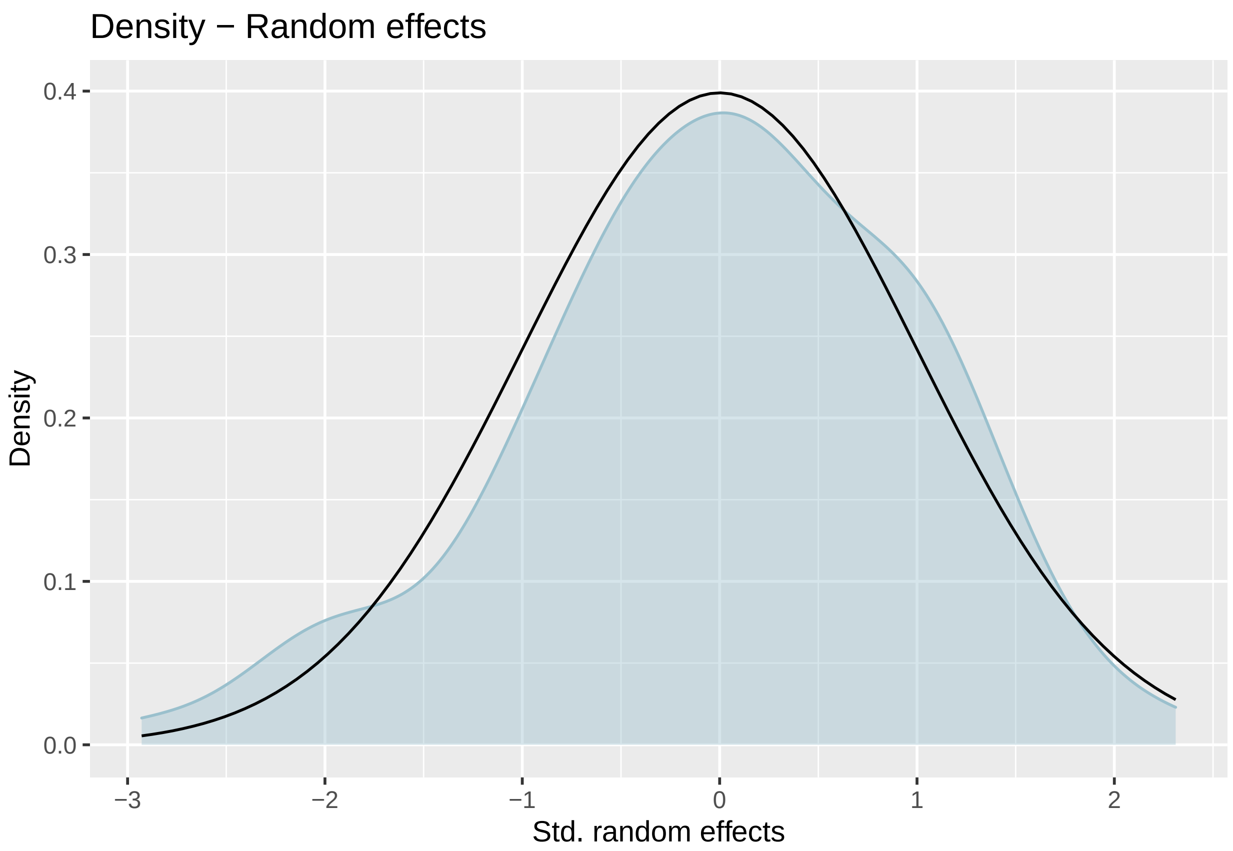
```
plot(fh_std)
```



```
## Press [enter] to continue
```

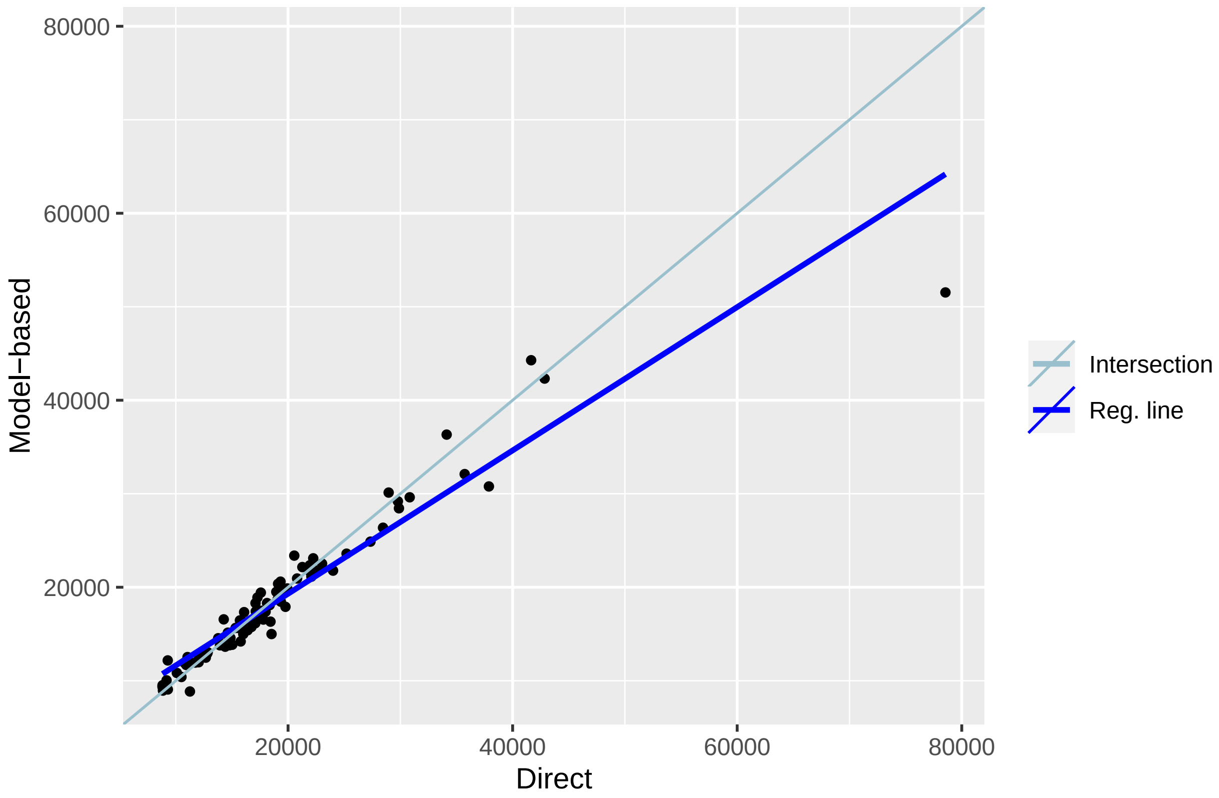


```
## Press [enter] to continue
```

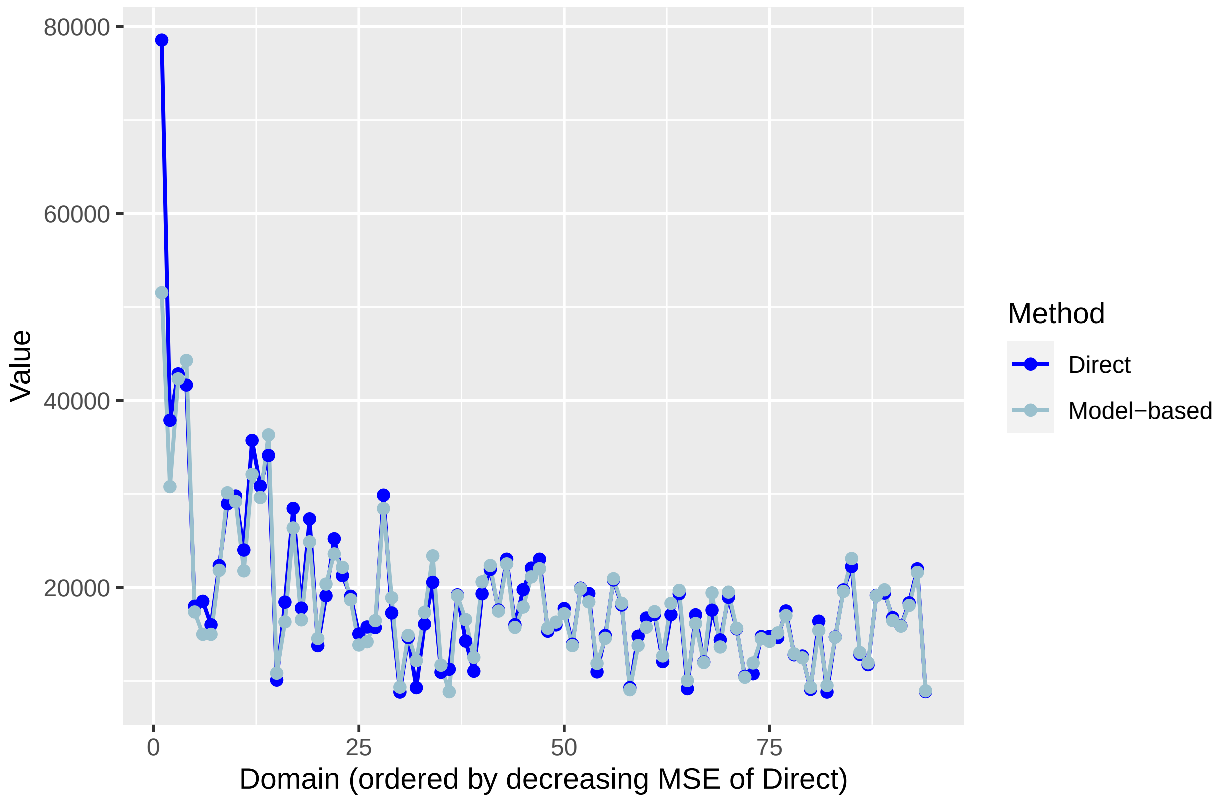


Compare results with direct estimates

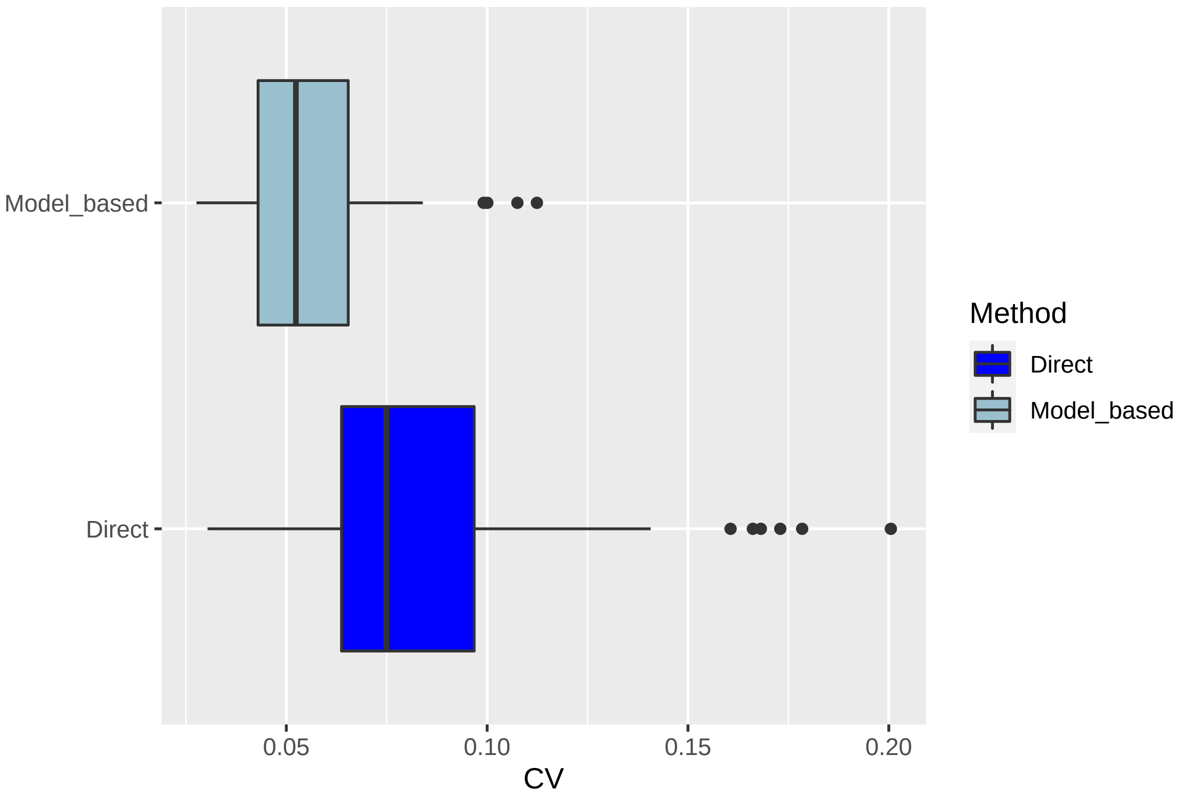
```
compare_plot(fh_std, CV = TRUE, label = "no_title")
```



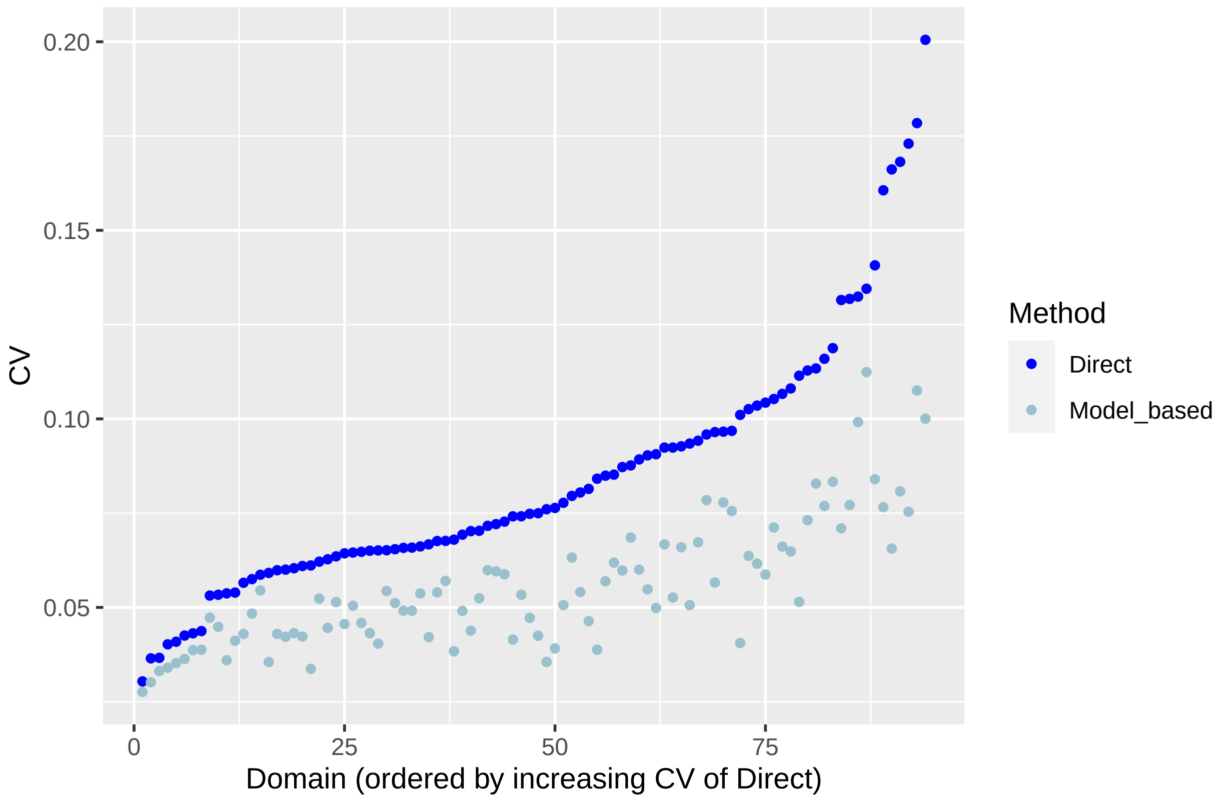
```
## Press [enter] to continue
```



```
## Press [enter] to continue
```



```
## Press [enter] to continue
```



```
compare(fh_std)
```

```
## Brown test
##
## Null hypothesis: EBLUP estimates do not differ significantly from the
##       direct estimates
##
##   W.value Df   p.value
##  46.97181 94 0.9999874
##
## Correlation between synthetic part and direct estimator:  0.94
```

Benchmarking for consistent estimates

```
data("eusilcA_smp")
mean(eusilcA_smp$eqIncome)
```

```
## [1] 20140.09
```

```
fh_bench <- benchmark(fh_std, benchmark = 20140.09,
                      share = eusilcA_popAgg$ratio_n, type = "ratio")
# Showing the first 6 rows of the results with head command
head(fh_bench)
```

```
##                Domain   Direct       FH FH_Bench Out
## 1           Amstetten 14768.57 14242.04 14480.61   0
## 2              Baden 21995.72 21616.40 21978.49   0
## 3            Bludenz 12069.59 12680.38 12892.79   0
## 4     Braunau am Inn 10770.53 11925.82 12125.59   0
## 5            Bregenz 35731.20 32101.69 32639.43   0
## 6 Bruck-Mürzzuschlag 23027.37 22523.50 22900.79   0
```

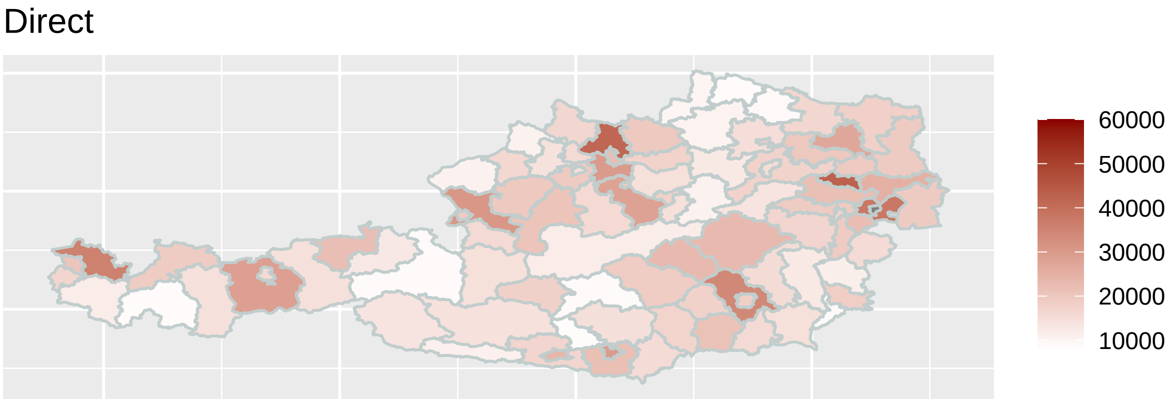
Extract and visualize the results

```
head(estimators(fh_std, MSE = TRUE))
```

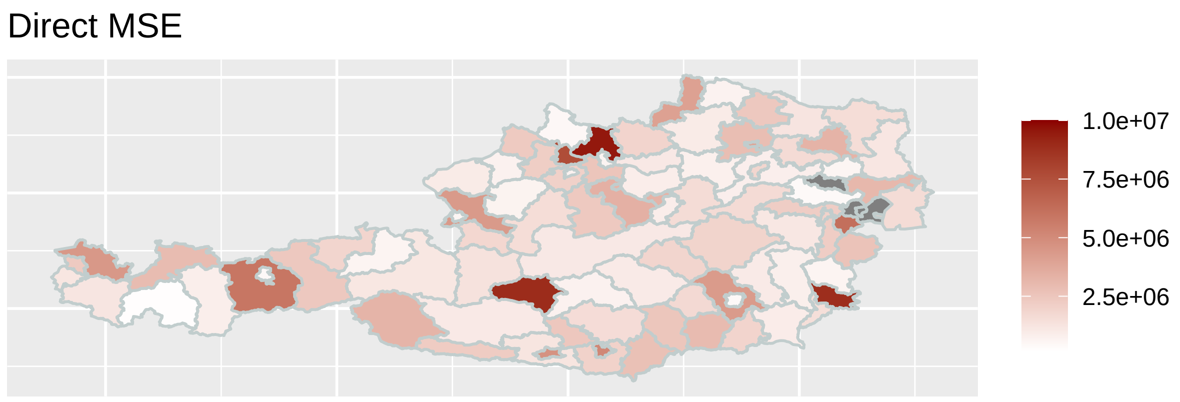
```
##                Domain   Direct Direct_MSE       FH    FH_MSE
## 1           Amstetten 14768.57   926167.4 14242.04  599010.6
## 2               Baden 21995.72   446534.3 21616.40  356586.1
## 3             Bludenz 12069.59  1243265.0 12680.38  716040.1
## 4      Braunau am Inn 10770.53  1029502.4 11925.82  643500.2
## 5             Bregenz 35731.20  4467316.4 32101.69 1302156.0
## 6  Bruck-Mürzzuschlag 23027.37  1971664.0 22523.50  906339.2
```

Visualization of results on maps

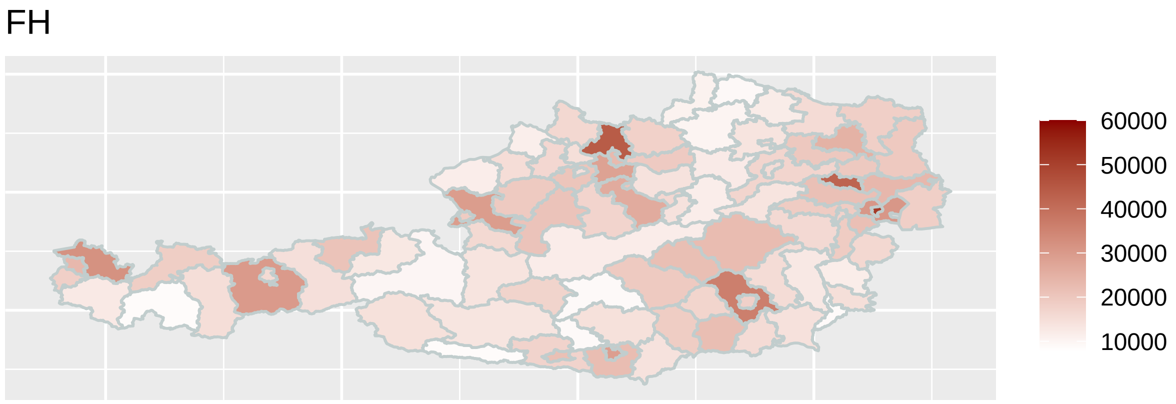
```
# Load shape file
load_shapeaustria()

# Maps with adjusted scales
map_plot(object = fh_std, MSE = TRUE,
         map_obj = shape_austria_dis, map_dom_id = "PB",
         scale_points = list(Direct = list(ind = c(8000, 60000),
                                           MSE = c(200000, 10000000)),
                             FH = list(ind = c(8000, 60000),
                                       MSE = c(200000, 10000000))))
```



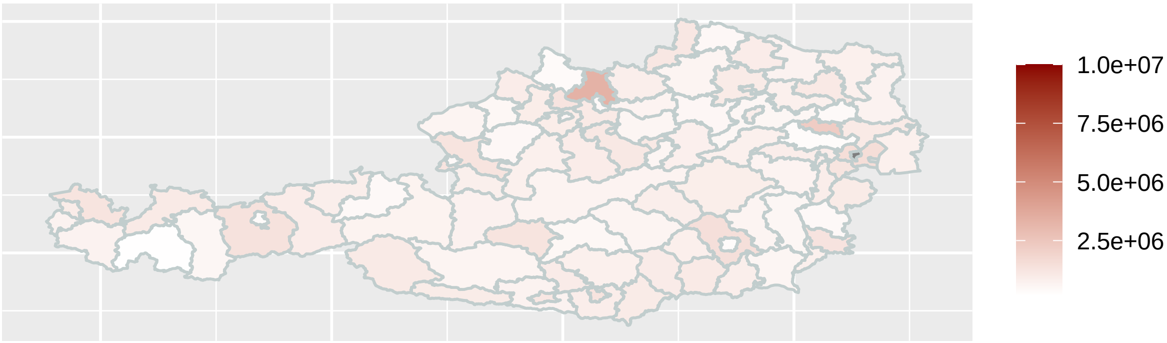
```
## Press [enter] to continue
```



```
## Press [enter] to continue
```



```
## Press [enter] to continue
```



FH MSE