

# R-tools

Ran at 2014-03-05 04:11:54

```

> wiki_username <- "Jouni"
> library(OpasnetUtils)
> library(ggplot2)
> library(rgdal)
> library(maptools)
> library(RColorBrewer)
> library(classInt)
> library(OpasnetUtilsExt)
> library(RgoogleMaps)
> objects.latest("Op_en6007", code_name = "answer") # [[OpasnetUtils/Drafts]] findrest
> objects.latest("Op_en6289", code_name = "initiate") # [[Building model]] Generic building model.
> objects.latest("Op_en5417", code_name = "initiate") # [[Population of Kuopio]]
> objects.latest("Op_en5932", code_name = "initiate") # [[Baseline building stock]] Building ovariables
> objects.latest("Op_en5932", code_name = "disperse") # [[Baseline building stock]] Summarised Piltti matrix
> # Default run: http://en.opasnet.org/en-opwiki/index.php?title=Special:RTools&id=aO1R3Xdcg2rASbKH
> colnames(If@output)[colnames(If@output) == "City.area"] <- "Emission.site"
> colnames(If@output)[colnames(If@output) == "Emission.height"] <- "Emission.height"
> districts <- tidy(opbase.data("Op_en5484.kuopio_city_districts"), widecol = "Location")
> districts <- Ovariable("districts", data = data.frame(districts, Result = 1))
> decisions <- opbase.data("Op_en5932", subset = 'Decisions')
> DecisionTableParser(decisions)
> # Remove previous decisions, if any.
> rm(
+ "buildings",
+ "buildingStock",
+ "buildingTypes",
+ "construction",
+ "efficiencies",
+ "efficienciesNew",
+ "energyUse",
+ "heatingShares",
+ "heatingSharesNew",
+ "renovation",
+ "renovationShares",
+ "fuelTypes",
+ "year",
+ envir = openv
+ )
> fuelTypes <- CheckDecisions(EvalOutput(fuelTypes))
Processing fuelTypes decisions ... done!

> ##### Manage the data before calculating
> # The building stock is measured as m^2 floor area.
> openv.setN(5) # Set the number of iterations to 1.
> obsyear <- (192:205) * 10
> ##### Remove columns and rows that are not needed from data. These must be done before EvalOutput.
> population <- EvalOutput(population)
> temp <- oapply(population, cols = "City area", FUN = "sum") # Sum across city areas.
> temp <- unkeep(temp * 1, prevresults = TRUE) # Remove populationResult column because it would cause trouble in merge.
> population <- population / temp
> buildingStock@data <- buildingStock@data[
+ buildingStock@data$Observation == "AreaHR",
+ colnames(buildingStock@data) != "Observation"
+ ]
> buildingStock <- buildingStock * population
> construction@data <- construction@data[
+ construction@data$Observation == "Area",
+ colnames(construction@data) != "Observation"
+ ]
> construction <- construction * constructionAreas / 100 / 3 # Statistics are for three years (2010-2012)
> energyUse@data <- energyUse@data[
+ energyUse@data[["Energy use"]] == "Heat",
+ colnames(energyUse@data) != "Energy use"
+ ]
> savingPotential@data <- savingPotential@data[
+ savingPotential@data$Observation == "Relative",
+ colnames(savingPotential@data) != "Observation"
+ ]

```

```

> savingPotential <- 1 - savingPotential / 100
> # Note: If you add decisions to savingPotential, this formula must be updated.
> # The data are not yet specific to construction year, so remove index:
> heatingSharesNew@data <- heatingSharesNew@data[colnames(heatingSharesNew@data) != "Constructed"]
> # Fill in Heating types and convert from % to fraction.
> heatingSharesNew <- findrest(heatingSharesNew, cols = "Heating", total = 100)
> result(heatingSharesNew) <- result(heatingSharesNew) / 100
> heatingShares <- EvalOutput(heatingShares)
> result(heatingShares) <- result(heatingShares) / 100 # From % to fraction
> # Fill in the rest of the data (the last emission class was omitted because it is determined by the total).
> efficienciesNew <- findrest(efficienciesNew, cols = "Efficiency", total = 100)
> result(efficienciesNew) <- result(efficienciesNew) / 100
> # When renovations are done, which type are they?
> renovationShares <- findrest(renovationShares, cols = "Renovation", total = 100)
> result(renovationShares) <- result(renovationShares) / 100
> # What fraction of buildings is renovated per year?
> renovation <- EvalOutput(renovation)
> result(renovation) <- result(renovation) / 100 # From % to fraction
> colnames(If@output)[colnames(If@output) == "City.area"] <- "Emission.site"
> colnames(If@output)[colnames(If@output) == "Emission.height"] <- "Emission.height"
> ##### Actual model
> buildings <- EvalOutput(buildings)
> ##### Sort factors
> buildings@output <- buildings@output[
+ buildings@output$EfficiencyPolicy == "BAU",
+ colnames(buildings@output) != "EfficiencyPolicy"
+ ]
> buildings@output$Renovation <- factor(
+ buildings@output$Renovation,
+ levels = c("None", "General", "Windows", "Technical systems", "Sheath reform"),
+ ordered = TRUE
+ )
> buildings@output$Efficiency <- factor(
+ buildings@output$Efficiency,
+ levels = c("Old", "New", "Low-energy", "Passive"),
+ ordered = TRUE
+ )
> buildings@output$RenovationPolicy <- factor(
+ buildings@output$RenovationPolicy,
+ levels = c("BAU", "Active renovation"),
+ ordered = TRUE
+ )
> heatingEnergy <- EvalOutput(heatingEnergy)
> heatingEnergy <- unkeep(heatingEnergy, cols = "Building2", sources = TRUE, prevresults = TRUE)
> temp <- buildings * districts
> temp@output <- temp@output[temp@output$Year == 2040, ]
> temp <- unkeep(temp, sources = TRUE, prevresults = TRUE)
> temp@output <- dropall(temp@output)
> temp <- oapply(temp, cols = c("Building", "Heating", "Efficiency", "Renovation"), FUN = "sum", na.rm = TRUE)
> MyRmap(
+ ova2spat(temp, coord = c("E", "N"), proj4string = "+init=epsg:3067"), # National Land Survey uses EPSG:3067 (ETRS-TM35FIN)
+ plotvar = "Result", legend_title = "Floor area", numbins = 8, pch = 19, cex = 2
+ )
[1] "http://maps.google.com/maps/api/staticmap?
center=62.8896559930947,27.6599798008151&zoom=11&size=640x480&mapttype=mobile&format=png32&sensor=true"

> ##### Calculate emissions
> emis <- heatingEnergy
> emis@output <- emis@output[emis@output$Year >= 1980, ]
> emis <- oapply(emis, cols = c("Efficiency", "Renovation", "Building"), FUN = "sum")
> emis@output$Year <- as.numeric(levels(emis@output$Year)[emis@output$Year])
> emis <- emis * fuelTypes * emissionFactors * 3.6 * 1E-9 # convert from kWh/a to MJ/a and mg to ton
> emis <- unkeep(emis, sources = TRUE, prevresults = TRUE)
> emis <- emis * emissionLocations
> colnames(emis@output)[colnames(emis@output) == "Emission site"] <- "Emission.site"
> colnames(emis@output)[colnames(emis@output) == "Emission height"] <- "Emission.height"
> emis@output$Emission.site <- ifelse(
+ emis@output$Emission.site == "At site of consumption",
+ levels(emis@output$City.area)[emis@output$City.area],
+ levels(emis@output$Emission.site)[emis@output$Emission.site]
+ )
> ##### Calculate health impacts
> bg.mort <- 45182 / 5203826 # same values as used in PILTTI unit: deaths per person-year
> erf <- Ovariable("erf", data = data.frame(Pollutant = "PM2.5", Result = 0.0097))

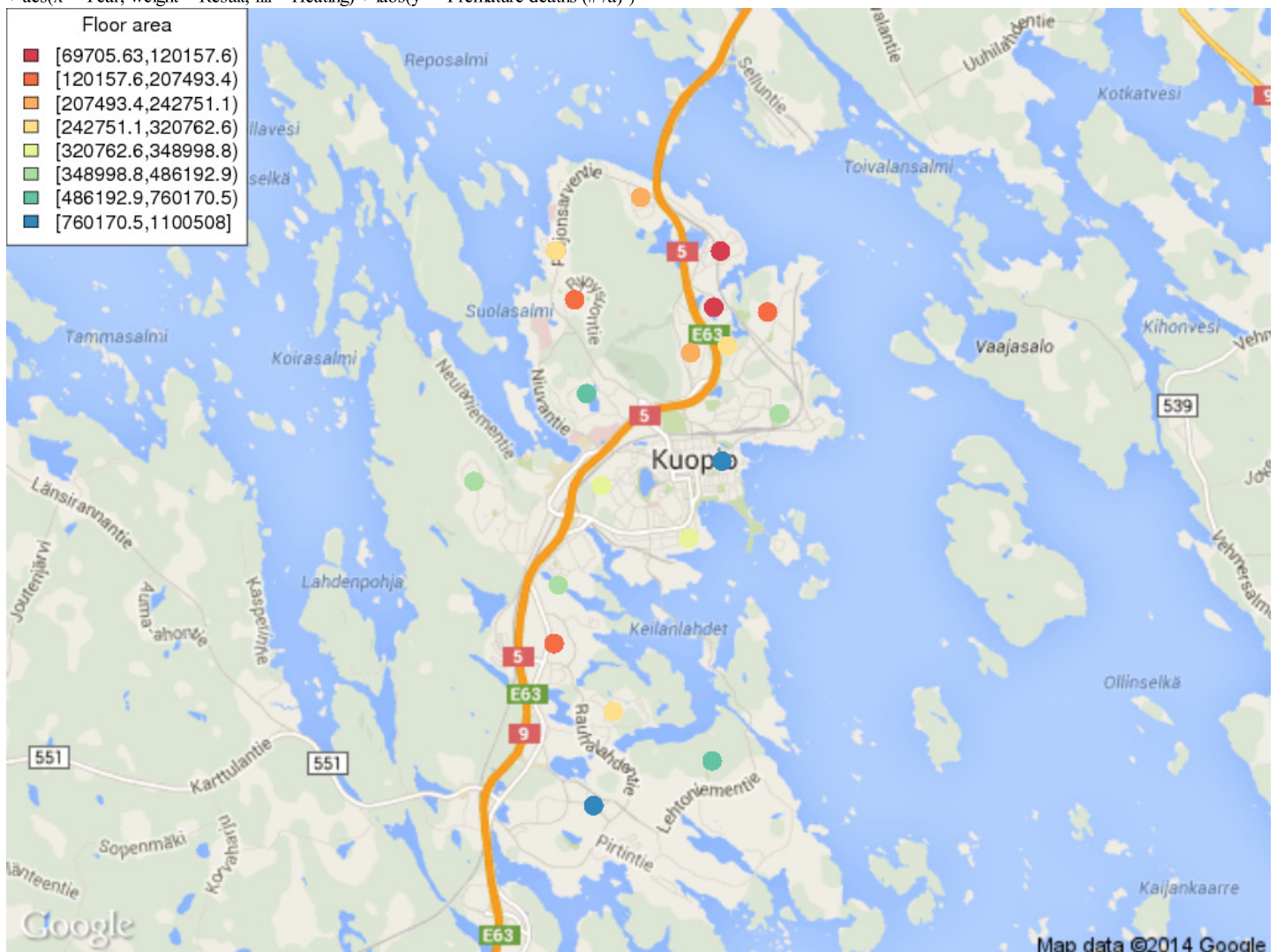
```

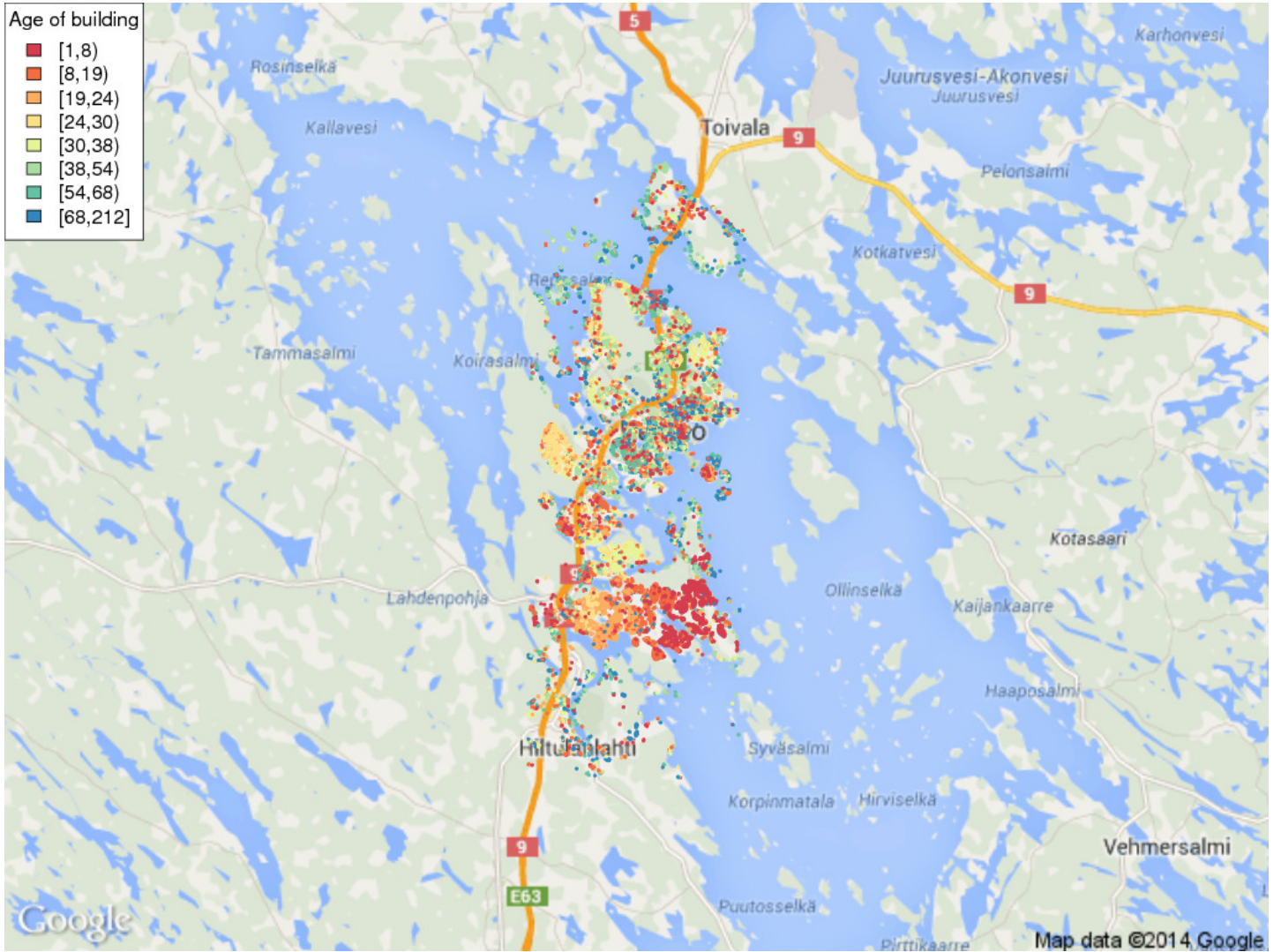
```

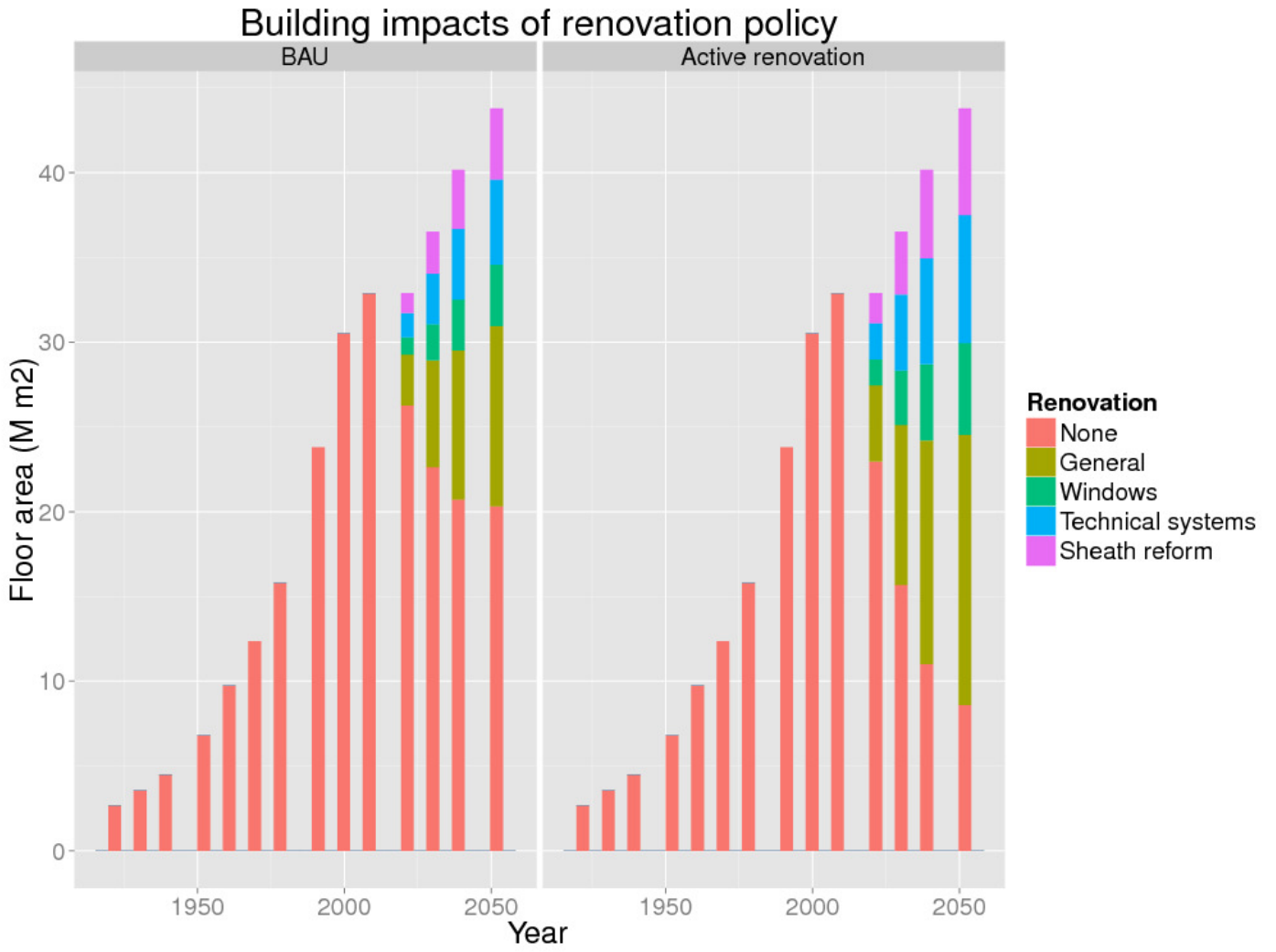
> # J. T. Tuomisto, A. Wilson, et al. Uncertainty in mortality response to airborne fine particulate matter... 2008
> # unit: m^3 /ug
> health <- (emis * 1E+6 /365 / 24 / 3600) * iF / (20 / 24 / 3600) * (erf * 1E+6) * bg.mort
> # emis is in ton /a
> # iF = conc (g /m3) * pop (#) * BR (m3 /s) / emis (g /s) => conc = iF * emis / pop / BR
> # HIA = conc * erf * pop * backgroundrisk = emis (g /s) * iF / BR (m3 /s) * erf(m3 /g) * backgroundrisk
> ##### Plot graphs
> ##### Buildings in Kuopio on map
> server <- TRUE
> if(server) # The following code only works from Opasnet server.
+ {
+ shp <- readOGR('PG:host=localhost user=postgres dbname=spatial_db','kuopio_house')
+ proj4string(shp) <- ("+init=epsg:3067") # The map projection of NLS Finland.
+
+ epsg4326String <- CRS("+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs")
+ shp2 <- spTransform(shp,epsg4326String) # Convert to longitude-latitude projection.
+
+ MyRmap(shp2, plotvar = "ika", legend_title = "Age of building", numbins = 8, pch = 19, cex = 0.3) # Draw the map.
+ }
OGR data source with driver: PostgreSQL Source: "PG:host=localhost user=postgres dbname=spatial_db", layer: "kuopio_house" with 16444 features and 9 fields
Feature type: wkbPoint with 2 dimensions [1] "http://maps.google.com/maps/api/staticmap?
center=62.8782737800078,27.6603987763551&zoom=10&size=640x480&maptype=mobile&format=png32&sensor=true"

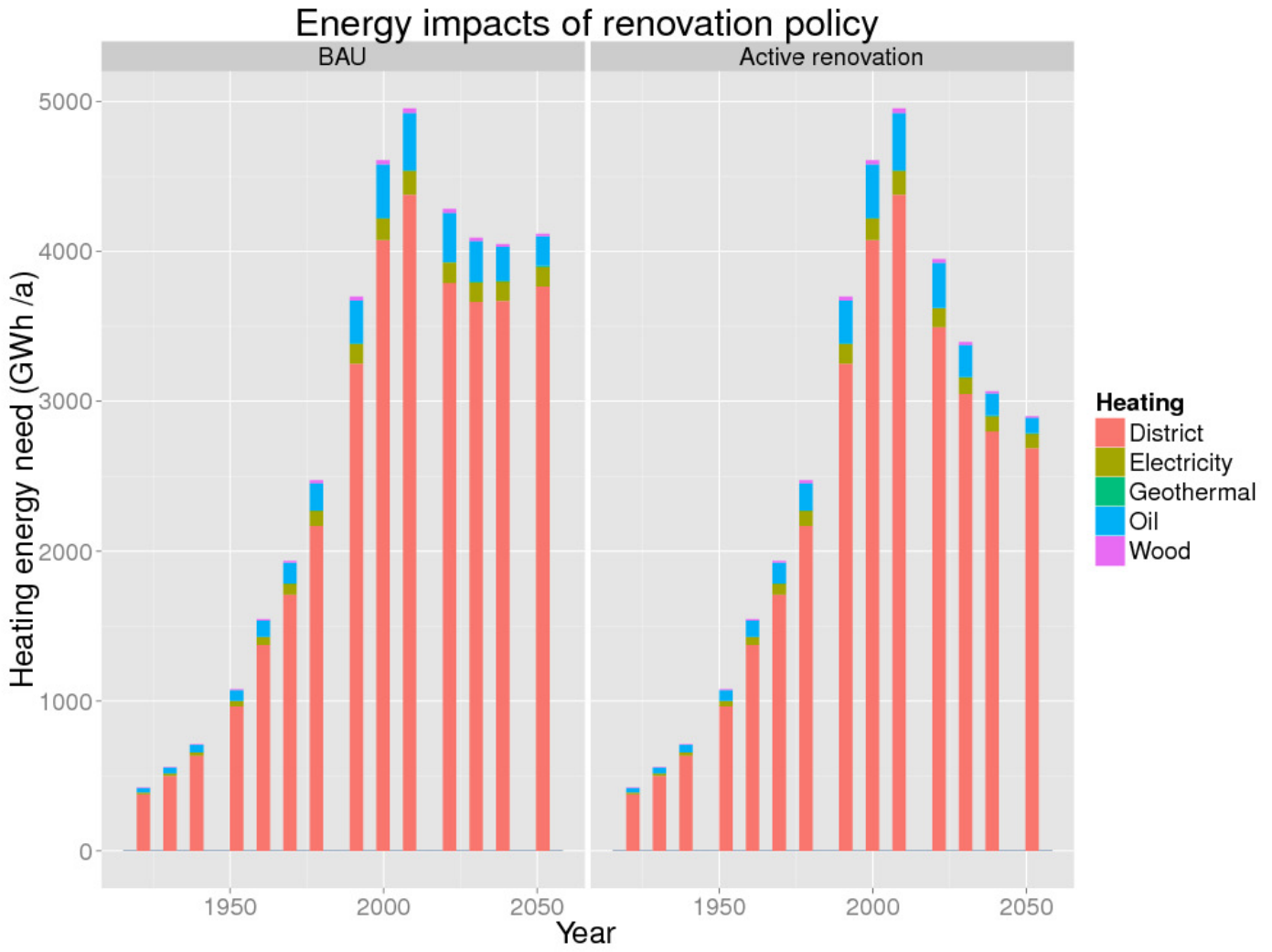
> ### Impact graphs
> BS <- 24
> ggplot(buildings@output) + geom_bar() + facet_grid(. ~ RenovationPolicy) + theme_gray(base_size = BS) +
+ aes(x = Year, weight = buildingsResult/1000000, fill = Renovation) + labs(y = "Floor area (M m2)", title = "Building impacts of renovation policy")
> ggplot(heatingEnergy@output) + geom_bar() + facet_grid(. ~ RenovationPolicy) + theme_gray(base_size = BS) + labs(title = "Energy impacts of renovation policy") +
+ aes(x = Year, weight = heatingEnergyResult/1E+6, fill = Heating) + labs(y = "Heating energy need (GWh /a)")
> emis@output <- emis@output[emis@output$Renovation == "BAU", ]
> ggplot(emis@output) + geom_bar() + facet_grid(Pollutant ~ FuelPolicy, scales = "free_y") + theme_gray(base_size = BS) + labs(title = "Emission impacts of biofuel
policy") +
+ aes(x = Year, weight = Result, fill = Fuel) + labs(y = "Emissions to air (ton /a)")
> ggplot(health@output) + geom_bar() + facet_grid(FuelPolicy ~ RenovationPolicy) + theme_gray(base_size = BS) + labs(title = "Health impacts of fuel and renovation
policy") +
+ aes(x = Year, weight = Result, fill = Heating) + labs(y = "Premature deaths (# /a)")

```









### Emission impacts of biofuel policy

