

Smoothing - Health

Yongrong Chai

12/2/2021

```
health <- st_read("smoothing_files/PLACES_ Census Tract Data (GIS Friendly Format), 2020 release/geo_expt/
#censustract <- st_read("/Users/karidachai/Desktop/Fall 2021/MA675/tree/Census2020_Tracts/Census2020_Tracts.shp")
asthma <- health %>% dplyr::select(stateabbr, statedesc, countyname, countyfips, tractfips, geometry, t
mental <- health %>% dplyr::select(stateabbr, statedesc, countyname, countyfips, tractfips, geometry, t

asthma_ = subset(asthma, countyname == "Suffolk"& stateabbr == "MA")
mental_ = subset(mental, countyname == "Suffolk"& stateabbr == "MA")

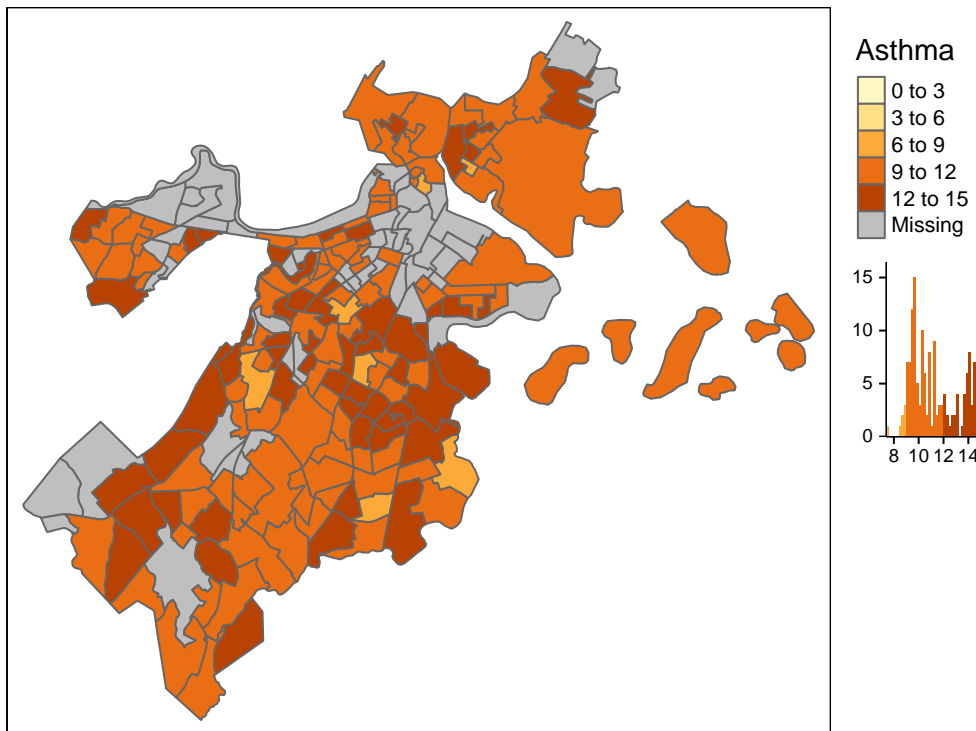
tract<- st_read("smoothing_files/Census2020_Tracts/Census2020_Tracts.shp",quiet = TRUE)

asthma_$TRACTCE10 <- substring(asthma_$tractfips,6, nchar(asthma_$tractfips))
asthma2 <- asthma_[asthma_$TRACTCE10 %in% tract$TRACTCE20,]

mental_$TRACTCE10 <- substring(mental_$tractfips,6, nchar(mental_$tractfips))
mental2 <- mental_[mental_$TRACTCE10 %in% tract$TRACTCE20,]

tract$Asthma[tract$TRACTCE20 %in% asthma2$TRACTCE10] <- asthma2$casthma_cr [asthma2$TRACTCE10 %in% trac
#tract$Asthma[is.na(tract$Asthma)] <- 0

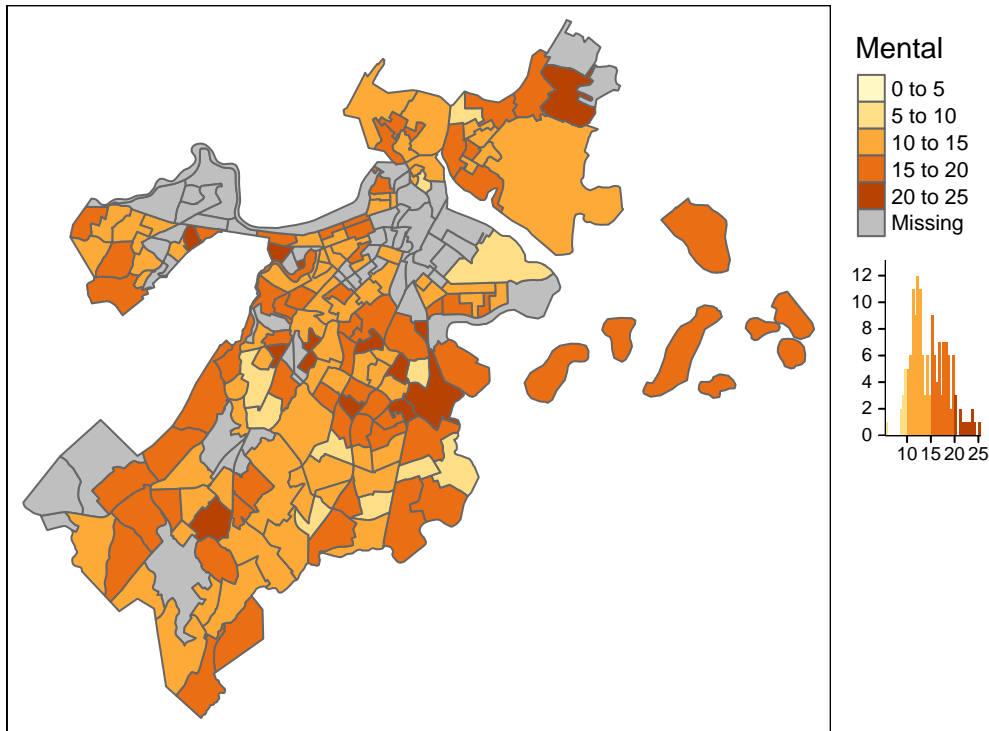
tm_shape(tract) +
  tm_polygons(col ="Asthma", style = "fixed", breaks = c(0, 3, 6, 9, 12, 15),
              legend.hist = TRUE) +
  tm_layout(legend.outside = TRUE)
```



```
tract$Mental[tract$TRACTCE20 %in% mental2$TRACTCE10] <- mental2$mhlth_crud [mental2$TRACTCE10 %in% tract$TRACTCE20]
#tract$Mental[is.na(tract$Mental)] <- 0
```

```
tm_shape(tract) +
  tm_polygons(col = "Mental", style = "fixed", breaks = c(0, 5, 10, 15, 20, 25),
    legend.hist = TRUE) +
  tm_layout(legend.outside = TRUE)
```

```
## Warning: Values have found that are higher than the highest break
```



```

#Spherical and Gaussian Variogram functions
spherical_variogram <- function (n, ps, r) function (h) {
  h <- h / r
  n + ps * ifelse(h < 1, 1.5 * h - .5 * h ^ 3, 1)
}
gaussian_variogram <- function (n, ps, r)
  function (h) n + ps * (1 - exp(-(h / r) ^ 2))
exponential_variogram <- function (n, ps, r)
  function (h) n + ps * (1 - exp(-(h / r)))
# solves `A * x = v` where `C = chol(A)` is the Cholesky factor:
chol_solve <- function (C, v) backsolve(C, backsolve(C, v, transpose = TRUE)) # the solution of triangu

#New Kriging Function
epsilon <- 1e-3
kriging_smooth_spherical <- function (formula, data, ...) {
  v <- variogram(formula, data)
  v_fit <- fit.variogram(v, vgm("Sph",...))
  v_fit$psill[1] <- max(v_fit$psill[1], epsilon)
  v_f <- spherical_variogram(v_fit$psill[1], v_fit$psill[2], v_fit$range[2])

  Sigma <- v_f(as.matrix(dist(st_coordinates(data)))) # semivariogram
  Sigma <- sum(v_fit$psill) - Sigma # prior variance
  tau2 <- v_fit$psill[1] # residual variance
  C <- chol(tau2 * diag(nrow(data)) + Sigma)
  y <- model.frame(formula, data)[, 1] # response
  x <- model.matrix(formula, data)

```

```

# generalized least squares:
xt <- backsolve(C, x, transpose = TRUE)
beta <- coef(lm.fit(xt, backsolve(C, y, transpose = TRUE))) # prior mean
names(beta) <- colnames(x)
beta_se <- sqrt(diag(chol2inv(chol(crossprod(xt)))))

chol_sigma <- chol(Sigma)
Sigma_inv <- chol2inv(chol_sigma)
C <- chol(Sigma_inv + diag(nrow(data)) / tau2)
# posterior mean (smoother):
mu <- drop(chol_solve(C, y / tau2 + Sigma_inv %*% x %*% beta))

sinv_mu <- chol_solve(chol_sigma, mu - x %*% beta)
krige <- function(new_data) { # prediction function
  D <- apply(coordinates(data), 1,
    function(coord) apply(st_coordinates(new_data), 1, dist2, coord))
  V <- sum(v_fit$psill) - v_f(D)

  t <- delete.response(terms(formula))
  xp <- model.matrix(t, model.frame(t, new_data))
  drop(xp %*% beta + V %*% sinv_mu)
}

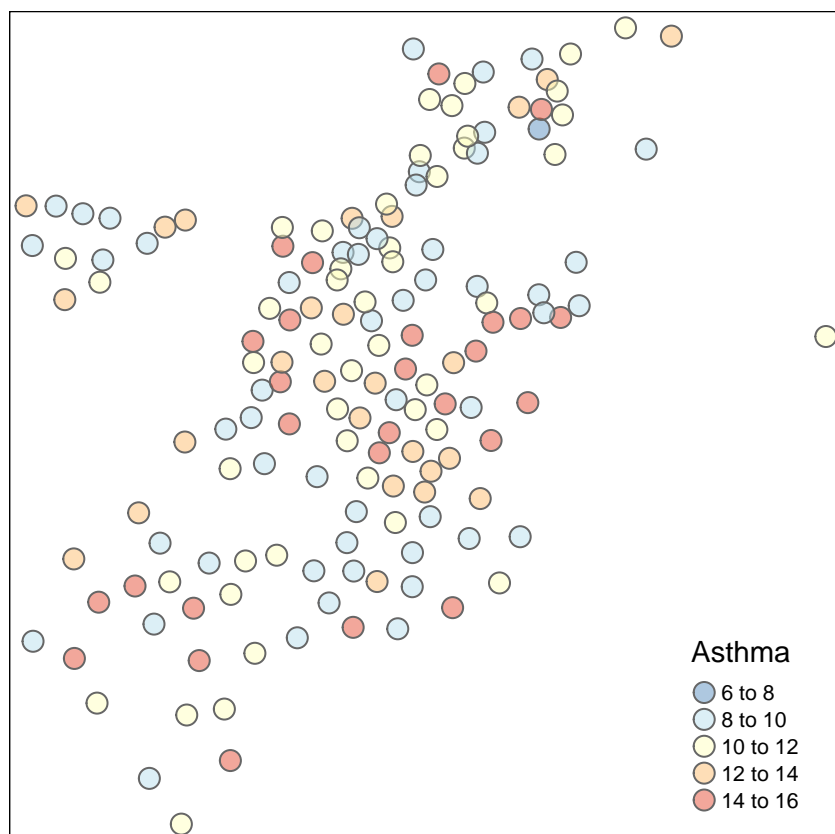
list(smooth = mu, prior_coef = beta, prior_coef_se = beta_se,
     variogram = v_fit, krige = krige)
}

tract2 <- st_centroid(tract) #Center the polygon

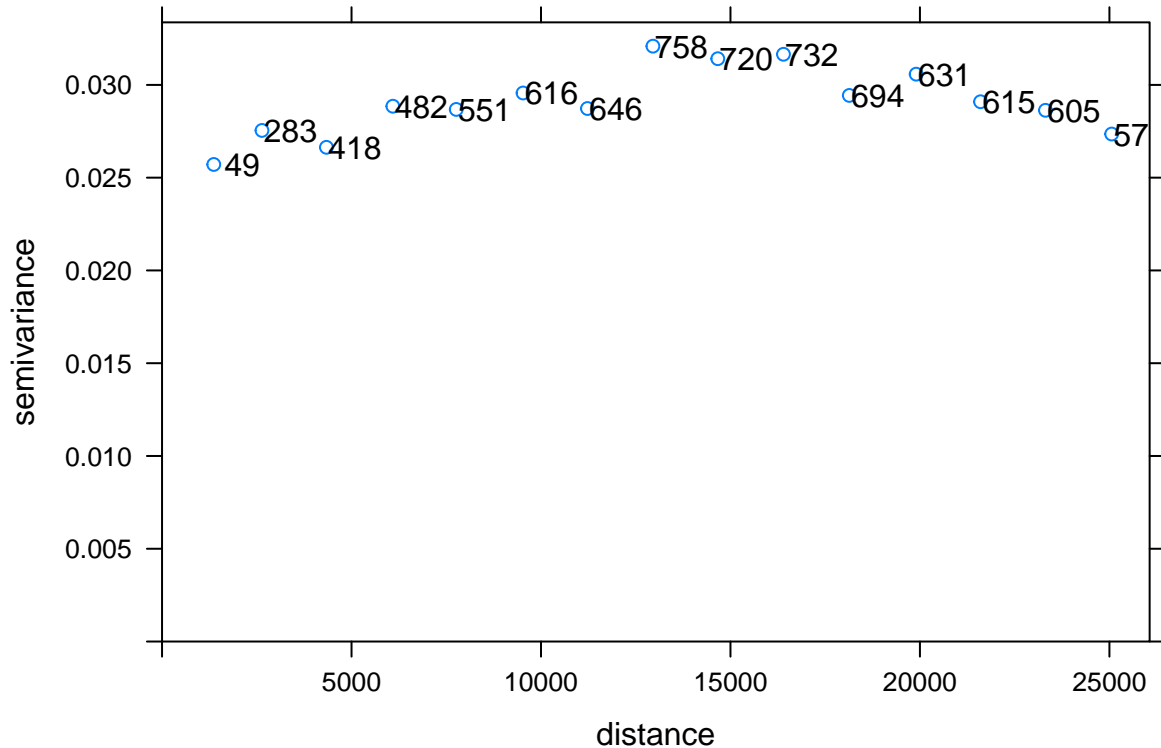
## Warning in st_centroid.sf(tract): st_centroid assumes attributes are constant
## over geometries of x

tract2 = na.omit(tract2)
#tract2$a = exp(tract2$Asthma)
tract2$a = log(tract2$Asthma)
tm_shape(tract2) +
  tm_bubbles(col = "Asthma", palette = "-RdYlBu", size = .3, alpha = .5)

```

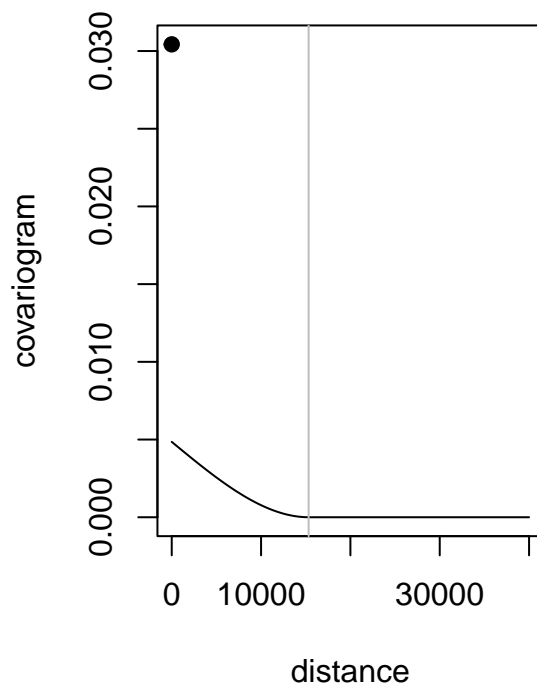
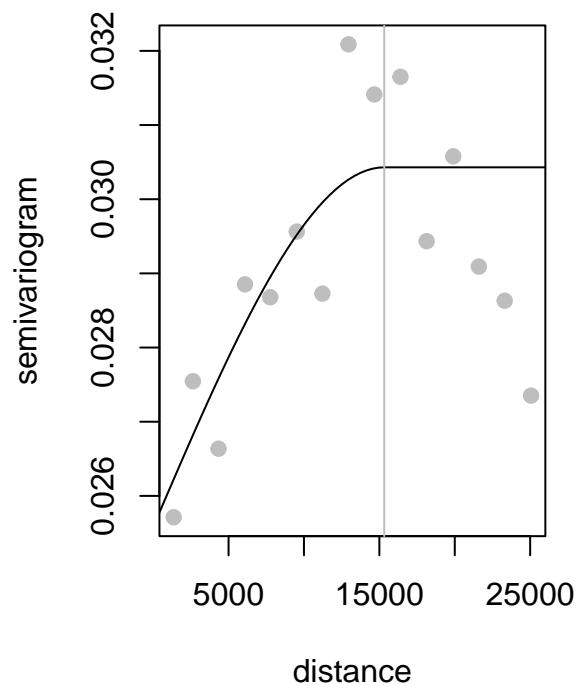


```
v <- variogram(a ~ 1, tract2)
plot(v, plot.numbers = T)
```



```
v_fit <- fit.variogram(v, vgm("Sph")) #'vgm':generate a variogram model or add to an existing model.
v_f <- spherical_variogram(v_fit$psill[1], v_fit$psill[2], v_fit$range[2])

# check variogram and covariance
op <- par(mfrow = c(1, 2))
h <- seq(0, 40000, length = 100)
plot(v$dist, v$gamma, pch = 19, col = "gray",
      xlab = "distance", ylab = "semivariogram")
lines(h, v_f(h))
abline(v = v_fit$range[2], col = "gray")
plot(h, sum(v_fit$psill) - v_f(h), type = "l",
      xlab = "distance", ylab = "covariogram",
      ylim = c(0, sum(v_fit$psill)))
points(0, sum(v_fit$psill), pch = 19)
abline(v = v_fit$range[2], col = "gray")
```

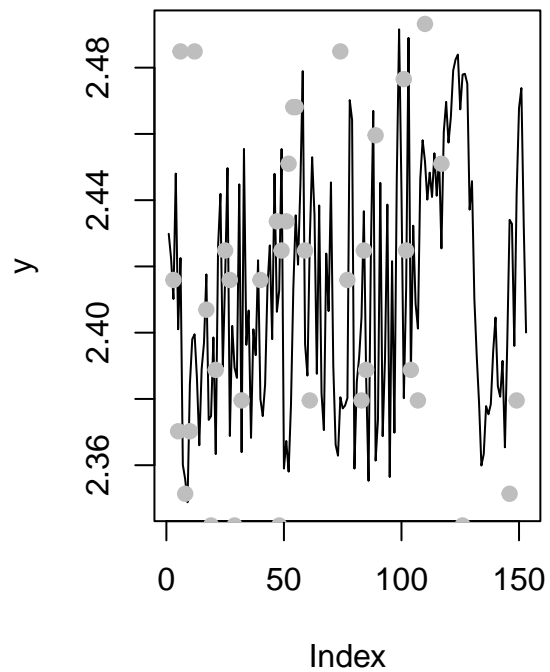
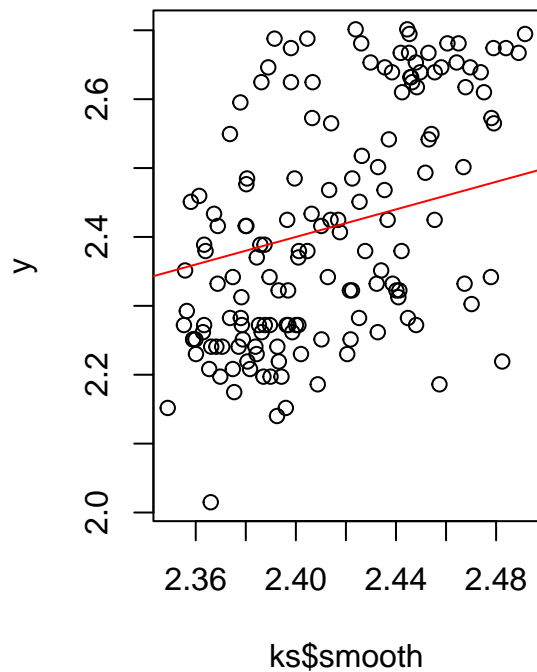


```
par(op)

#coordinates- x & y
#tract3 <- tract2 %>%
#   mutate(x = unlist(map(tract2$geometry,1)),
#          y = unlist(map(tract2$geometry,2)))
#tract3

#tract4 <- tract3 %>% st_sf() %>% as_Spatial()

# check smoothing
ks <- kriging_smooth_spherical(a ~ 1, tract2)
y <- tract2$a
op <- par(mfrow = c(1, 2))
plot(ks$smooth, y); abline(0, 1, col = "red")
plot(ks$smooth, type = "l", ylab = "y")
points(y, pch = 19, col = "gray")
abline(h = ks$prior_mean)
```



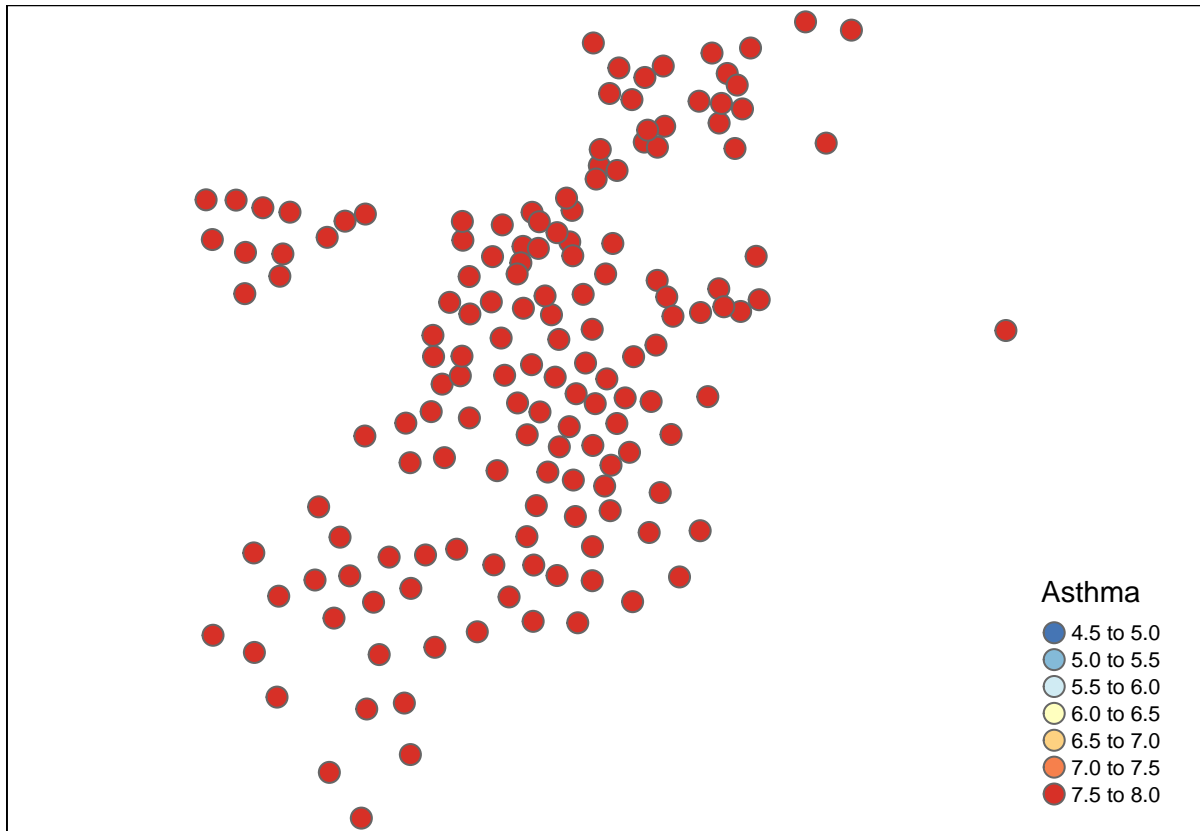
```
par(op)
tract2$smootha <- ks$smooth
breaks <- seq(4.5, 8, by = .5)
tmap_mode("plot")
```

```
## tmap mode set to plotting
```

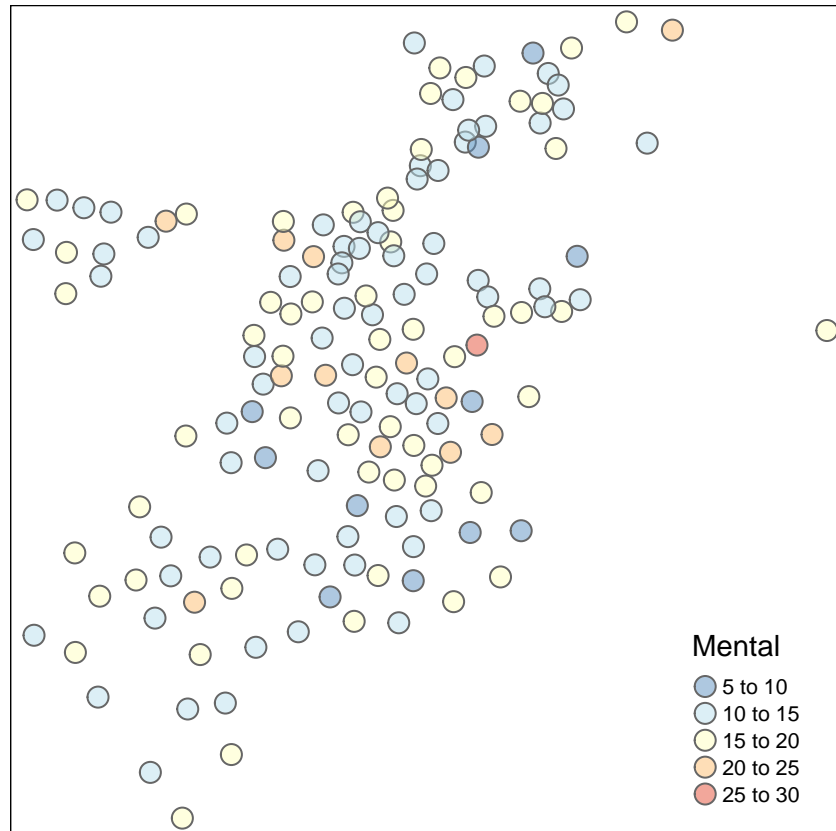
```
tmap_arrange(
  tm_shape(tract2) +
    tm_bubbles(col = "Asthma", palette = "-RdYlBu", size = .3, breaks = breaks)
  #tm_shape(tract2) +
    #tm_bubbles(col = "smootha", palette = "-RdYlBu", size = .3, breaks = breaks)
)
```

```
## Warning: Values have found that are higher than the highest break
```

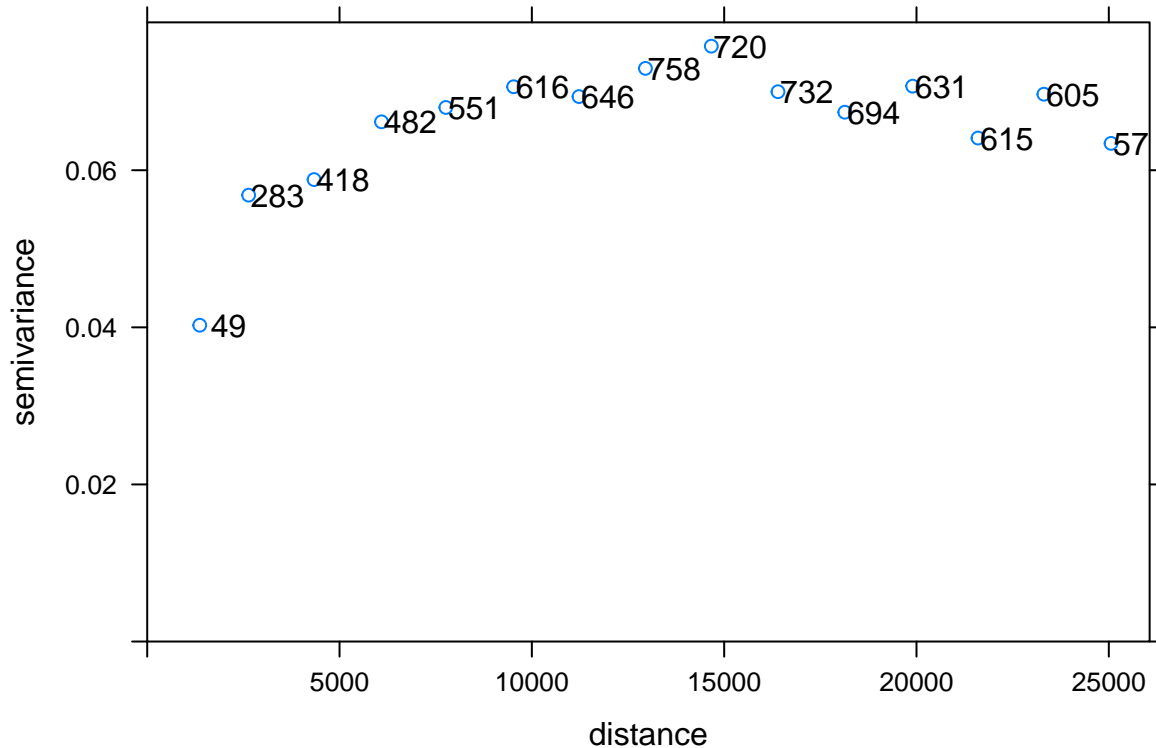
```
## Warning: Values have found that are higher than the highest break
```

```
tract2$b = log(tract2$Mental)
tm_shape(tract2) +
  tm_bubbles(col = "Mental", palette = "-RdYlBu", size = .3, alpha = .5)
```

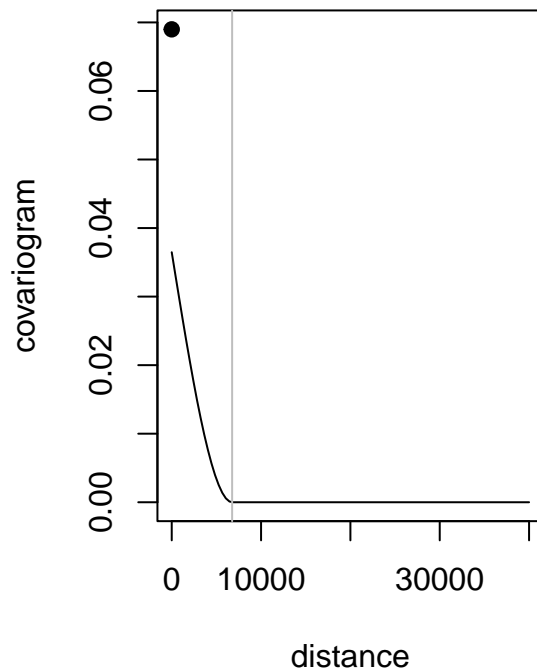
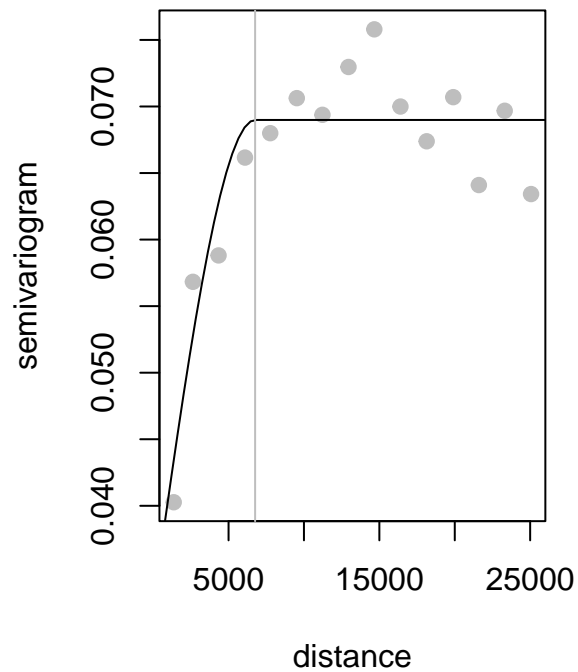


```
v <- variogram(b ~ 1, tract2)
plot(v, plot.numbers = T)
```



```
v_fit <- fit.variogram(v, vgm("Sph")) #'vgm': generate a variogram model or add to an existing model.
v_f <- spherical_variogram(v_fit$psill[1], v_fit$psill[2], v_fit$range[2])

# check variogram and covariance
op <- par(mfrow = c(1, 2))
h <- seq(0, 40000, length = 100)
plot(v$dist, v$gamma, pch = 19, col = "gray",
     xlab = "distance", ylab = "semivariogram")
lines(h, v_f(h))
abline(v = v_fit$range[2], col = "gray")
plot(h, sum(v_fit$psill) - v_f(h), type = "l",
     xlab = "distance", ylab = "covariogram",
     ylim = c(0, sum(v_fit$psill)))
points(0, sum(v_fit$psill), pch = 19)
abline(v = v_fit$range[2], col = "gray")
```

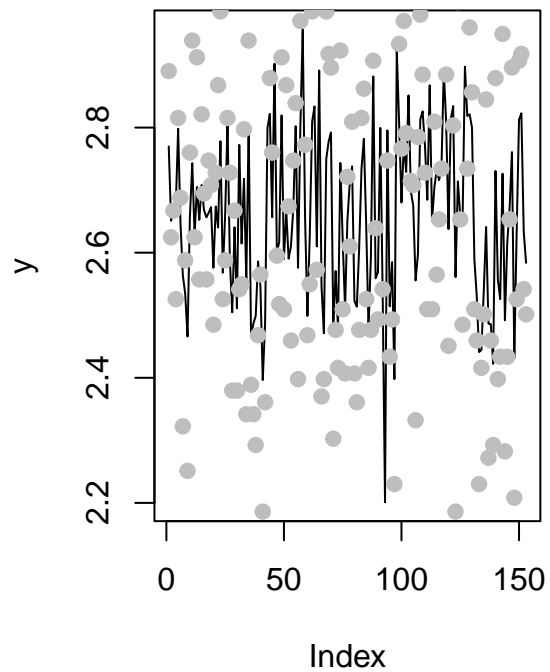
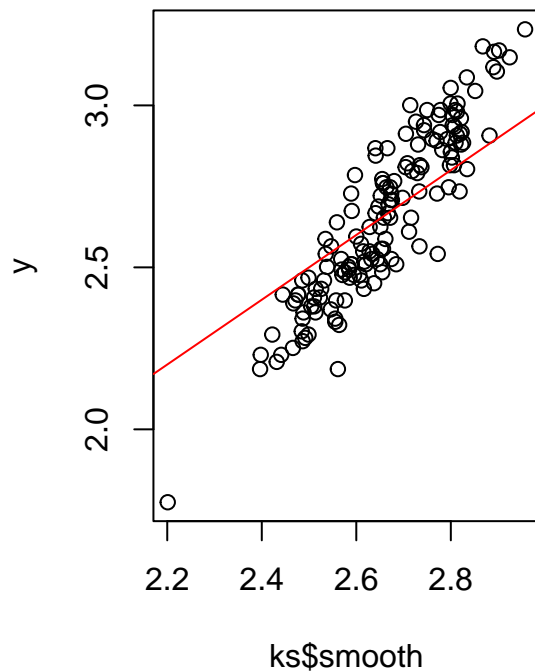


```
par(op)

#coordinates- x & y
#tract3 <- tract2 %>%
#   mutate(x = unlist(map(tract2$geometry,1)),
#          y = unlist(map(tract2$geometry,2)))
#tract3

#tract4 <- tract3 %>% st_sf() %>% as_Spatial()

# check smoothing
ks <- kriging_smooth_spherical(b ~ 1, tract2)
y <- tract2$b
op <- par(mfrow = c(1, 2))
plot(ks$smooth, y); abline(0, 1, col = "red")
plot(ks$smooth, type = "l", ylab = "y")
points(y, pch = 19, col = "gray")
abline(h = ks$prior_mean)
```



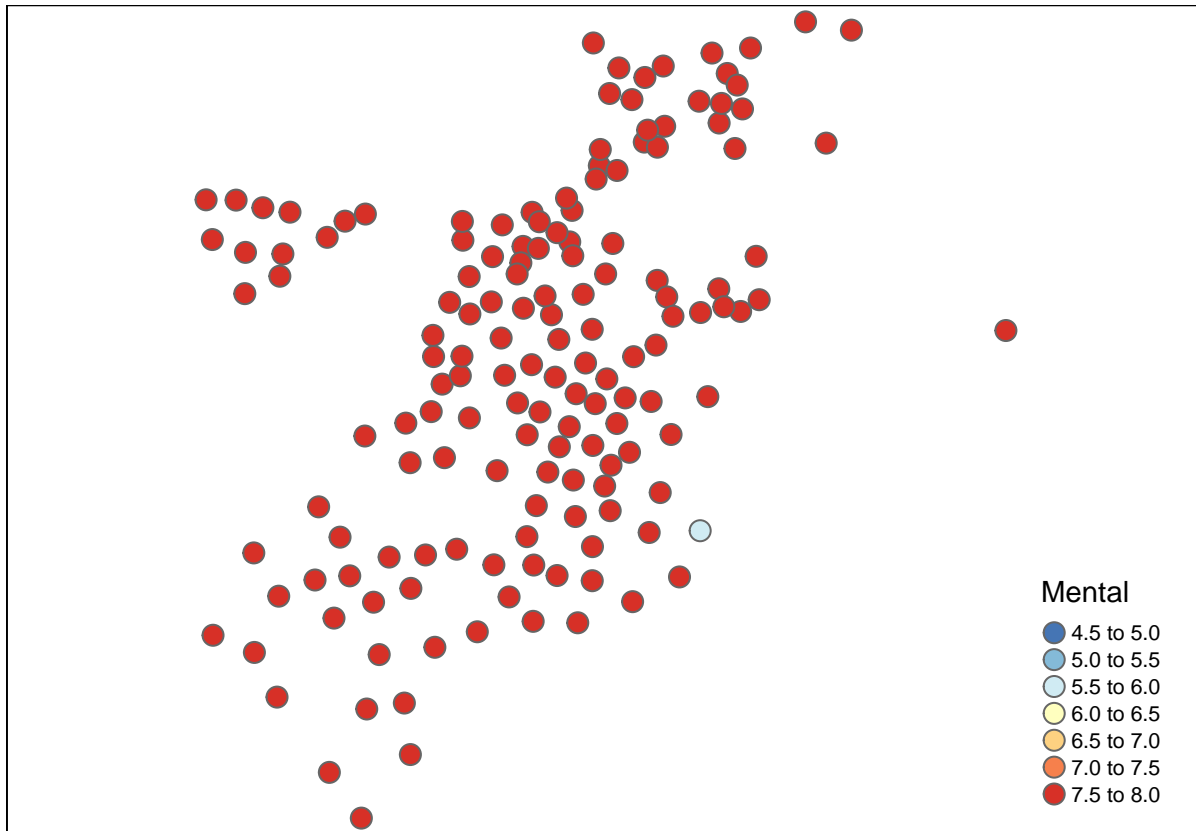
```
par(op)
tract2$smoothb <- ks$smooth
breaks <- seq(4.5, 8, by = .5)
tmap_mode("plot")
```

```
## tmap mode set to plotting
```

```
tmap_arrange(
  tm_shape(tract2) +
    tm_bubbles(col = "Mental", palette = "-RdYlBu", size = .3, breaks = breaks)
  #tm_shape(tract2) +
    #tm_bubbles(col = "smooth", palette = "-RdYlBu", size = .3, breaks = breaks)
)
```

```
## Warning: Values have found that are higher than the highest break
```

```
## Warning: Values have found that are higher than the highest break
```



Merge data with enviornment group

```
environ <- st_read("smoothing_files/Tree_dem_env_1129/Tree_dem_env.shp", quiet = TRUE)
names(environ)
```

```
## [1] "GEOID20" "men_tmp" "mn_pm25" "OBJECTI" "STATEFP" "COUNTYF"
## [7] "TRACTCE" "NAME20" "NAMELSA" "MTFCC20" "FUNCSTA" "ALAND20"
## [13] "AWATER2" "INTPTLA" "INTPTLO" "Sh_STA_" "Sh_STL_" "Tre_Cnp"
## [19] "TotalPp" "MinrPrc" "PpDnsty" "MedGrTx" "Mdn_Inc" "MnrPrc_t"
## [25] "MnrPrc_s" "MnrPrc_r" "MdGrTx_t" "MdGrTx_s" "MdGrTx_r" "Mdn_Inc_t"
## [31] "Mdn_Inc_s" "Mdn_Inc_r" "mn_pm25_s" "mn_pm25_r" "mn_tmp_s" "mn_tmp_r"
## [37] "geometry"
```

```
names(tract2)
```

```
## [1] "OBJECTID" "STATEFP20" "COUNTYFP20" "TRACTCE20" "GEOID20"
## [6] "NAME20" "NAMELSAD20" "MTFCC20" "FUNCSTAT20" "ALAND20"
## [11] "AWATER20" "INTPTLAT20" "INTPTLON20" "Shape_STAr" "Shape_STLe"
## [16] "geometry" "Asthma" "Mental" "a" "smootha"
## [21] "b" "smoothb"
```

```
library(tidyverse)
newhealth <- environ %>% inner_join(as.data.frame(tract2)[,-16], by = 'GEOID20')
newhealth <- newhealth %>% st_as_sf(sf_column_name = "geometry")
```

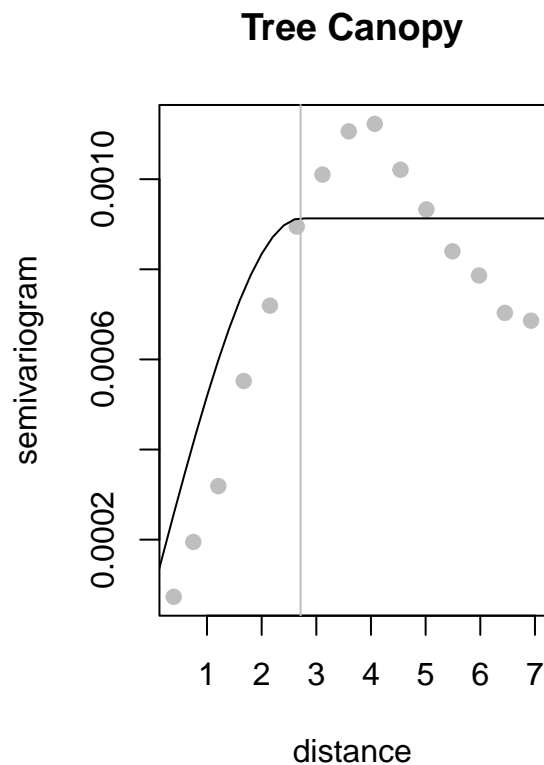
```

smoothing_formula <- smootha ~ MnrPrc_s + MnrPrc_r + MdGrTx_s + MdGrTx_r + Mdn_Inc_s + Mdn_Inc_r + mn_p
v <- variogram(smoothing_formula, newhealth)

v_fit <- fit.variogram(v, vgm("Wav"))
v_f <- spherical_variogram(v_fit$psill[1], v_fit$psill[2], v_fit$range[2])
op <- par(mfrow = c(1, 2))
h <- seq(0, 200, length = 1000)
plot(v$dist, v$gamma, pch = 19, col = "gray",
     xlab = "distance", ylab = "semivariogram", main = "Tree Canopy")
lines(h, v_f(h))
abline(v = v_fit$range[2], col = "gray")

# check smoothing:
ks_as <- kriging_smooth_spherical(smoothing_formula, newhealth)
y <- newhealth$smootha
op <- par(mfrow = c(1, 2))

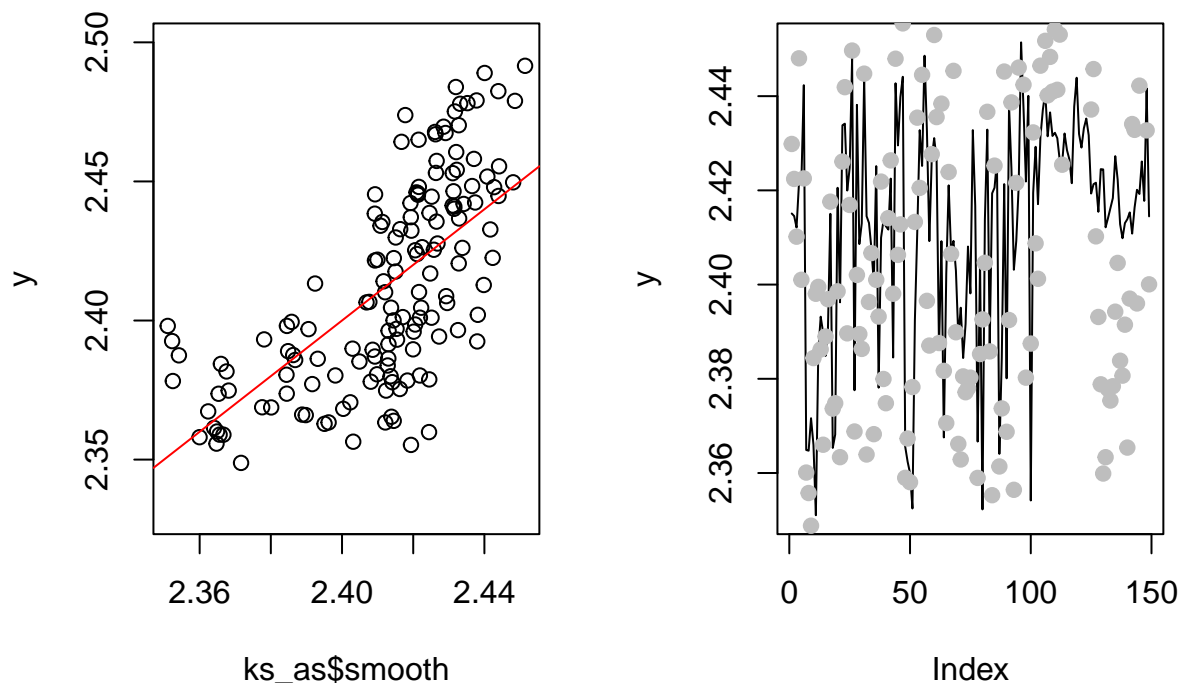
```



```

plot(ks_as$smooth, y, ylim = c(2.33, 2.5)); abline(0, 1, col = "red")
plot(ks_as$smooth, type = "l", ylab = "y")
points(y, pch = 19, col = "gray")
abline(h = ks_as$prior_coef)

```



```
par(op)
rbind(ks_as$prior_coef, ks_as$prior_coef_se) #to see coef values

##      (Intercept) MnrPrc_s MnrPrc_r MdGrTx_s MdGrTx_r Mdn_Inc_s Mdn_Inc_r
## [1,]          8.2  -0.0062  -0.0024  -0.124   0.0028  -0.166  -0.0175
## [2,]          2.0   0.0036   0.0076   0.024   0.0018   0.048   0.0089
##      mn_pm25_s mn_pm25_r mn_tmp_s  mn_tmp_r Tre_Cnp
## [1,]    -0.010    0.014 -0.000063 0.00000074  0.017
## [2,]     0.019     0.024  0.000040 0.00000451  0.042

library(flextable)
coef1 <- as.data.frame(t(rbind(ks_as$prior_coef, ks_as$prior_coef_se)))
coef1 <- cbind(row.names(coef1), coef1)
flextable(coef1) %>% theme_booktabs()

## Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
## engine is used and not `xelatex` or `lualatex`. You can avoid this warning
## by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
## compatible engine by defining `latex_engine: xelatex` in the YAML header of the
## R Markdown document.
```

row.names(coef1)	V1	V2
(Intercept)	8.24169533	1.9697229
MnrPrc_s	-0.00616590	0.0036344

row.names(coef1)	V1	V2
MnrPrc_r	- 0.00236685	0.0075536
MdGrTx_s	- 0.12446441	0.0237732
MdGrTx_r	0.00284747	0.0018065
Mdn_Inc_s	- 0.16641901	0.0479271
Mdn_Inc_r	- 0.01746738	0.0089063
mn_pm25_s	- 0.01049538	0.0185436
mn_pm25_r	0.01356145	0.0244551
mn_tmp_s	- 0.00006318	0.0000397
mn_tmp_r	0.00000074	0.0000045
Tre_Cnp	0.01673274	0.0424968

```

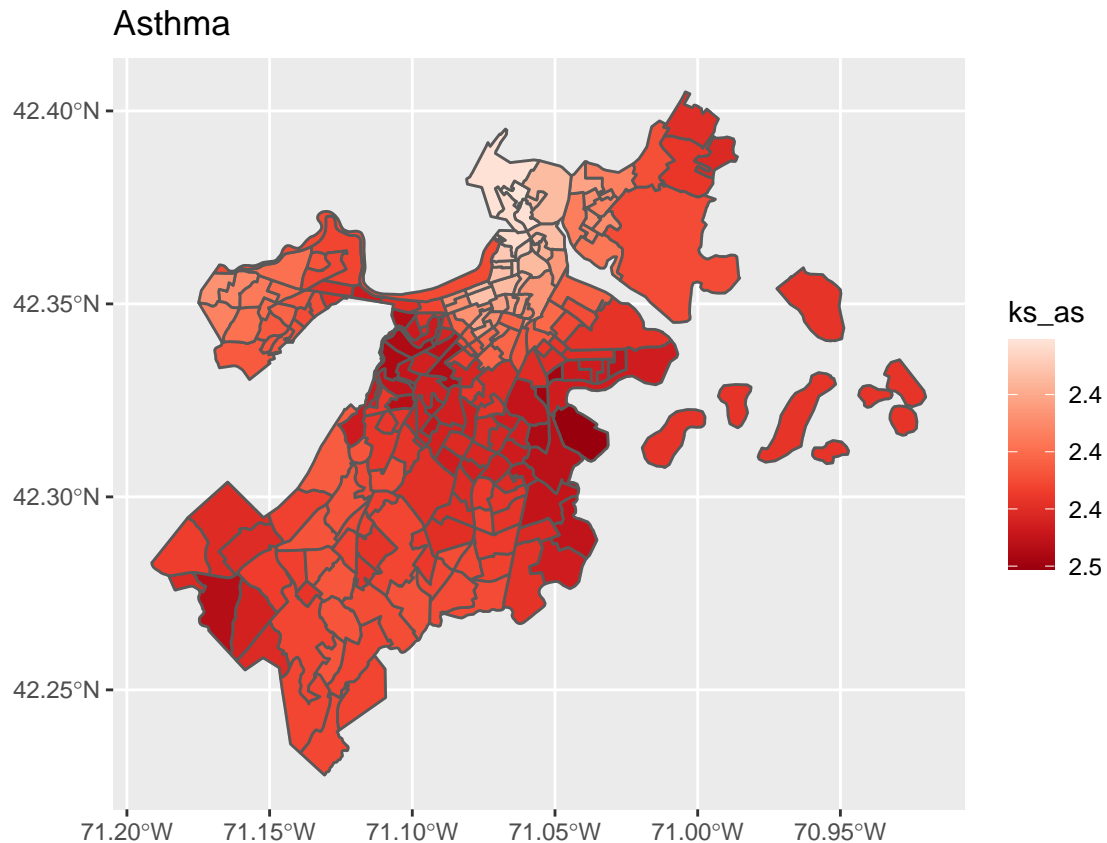
tract7 <- st_read("smoothing_files/matches/matches.shp", quiet = T)
newhealth$ks_as <- ks_as$smooth
tract7$ks_as[tract7$GEOID20 %in% newhealth$GEOID20] <- newhealth$ks_as[newhealth$GEOID20 %in% tract7$GEOID20]
grid <- st_centroid(tract7[is.na(tract7$ks_as),])

## Warning in st_centroid.sf(tract7[is.na(tract7$ks_as), ]): st_centroid assumes
## attributes are constant over geometries of x

pred_AS <- krige(ks_as ~1, newhealth, grid$geometry, model = v_fit)

## [using ordinary kriging]
tract7$ks_as[is.na(tract7$ks_as)] <- pred_AS$var1.pred
ggplot(data = tract7) + geom_sf(color = 'grey') + geom_sf(aes(fill = ks_as)) +
  scale_fill_distiller(palette="Reds", trans = "reverse") + labs(title = 'Asthma')

```



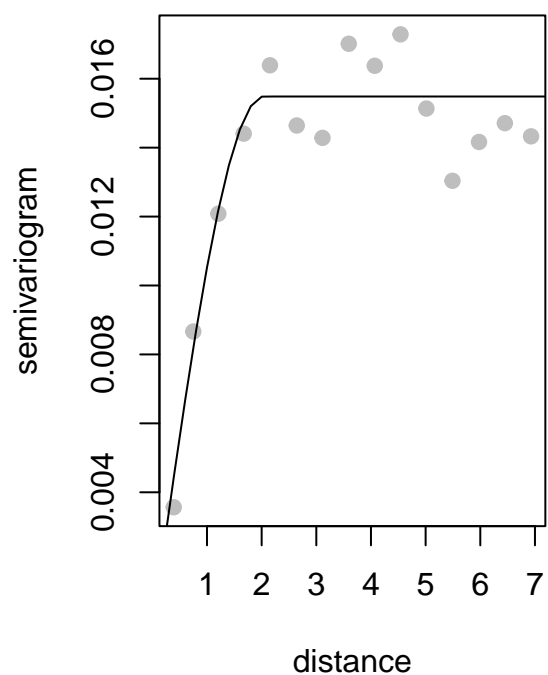
```
smoothing_formula1 <- smoothb ~ MnrPrc_s + MnrPrc_r + MdGrTx_s + MdGrTx_r + Mdn_Inc_s + Mdn_Inc_r + mn_
v2 <- variogram(smoothing_formula1, newhealth)
```

```
v_fit <- fit.variogram(v2, vgm("Sph"))
v_f <- spherical_variogram(v_fit$psill[1], v_fit$psill[2], v_fit$range[2])
op <- par(mfrow = c(1, 2))
h <- seq(0, 200, length = 1000)
plot(v2$dist, v2$gamma, pch = 19, col = "gray",
      xlab = "distance", ylab = "semivariogram", main = "Tree Canopy")
lines(h, v_f(h))
abline(v2 = v_fit$range[2], col = "gray")
```

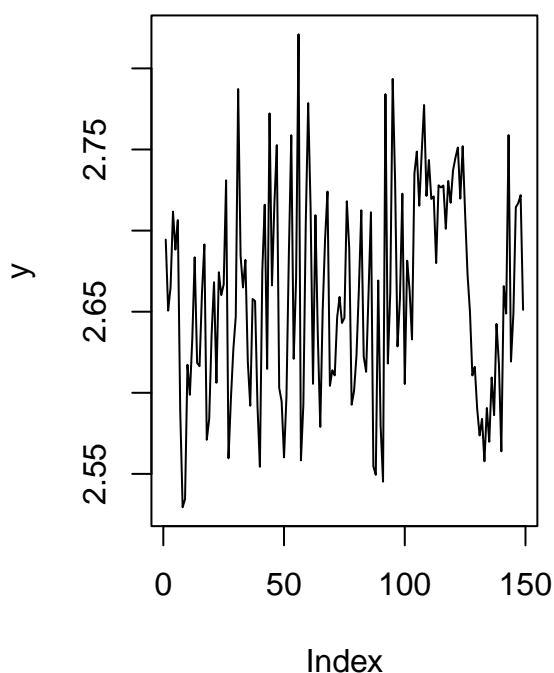
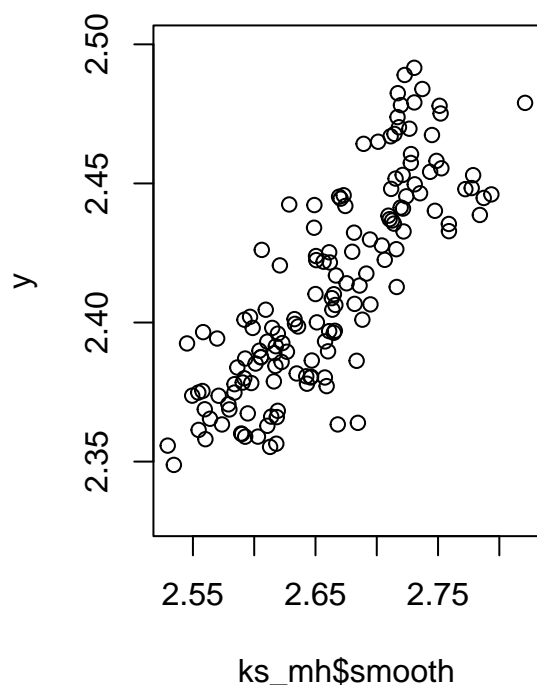
```
## Warning in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...): "v2" is not
## a graphical parameter
```

```
# check smoothing:
ks_mh <- kriging_smooth_spherical(smoothing_formula1, newhealth)
y <- newhealth$smootha
op <- par(mfrow = c(1, 2))
```

Tree Canopy



```
plot(ks_mh$smooth, y, ylim = c(2.33, 2.5)); abline(0, 1, col = "red")
plot(ks_mh$smooth, type = "l", ylab = "y")
points(y, pch = 19, col = "gray")
abline(h = ks_mh$prior_coef)
```



```
par(op)
```

```
coef2 <- as.data.frame(t(rbind(ks_mh$prior_coef, ks_mh$prior_coef_se)))
coef2 <- cbind(row.names(coef2),coef2)
flextable(coef2) %>% theme_booktabs()
```

```
## Warning: Warning: fonts used in `flextable` are ignored because the `pdflatex`
## engine is used and not `xelatex` or `lualatex`. You can avoid this warning
## by using the `set_flextable_defaults(fonts_ignore=TRUE)` command or use a
## compatible engine by defining `latex_engine: xelatex` in the YAML header of the
## R Markdown document.
```

row.names(coef2)	V1	V2
(Intercept)	3.810052	4.955669
MnrPrc_s	0.014590	0.005963
MnrPrc_r	-0.023110	0.008192
MdGrTx_s	-0.354464	0.044445
MdGrTx_r	0.004320	0.002019
Mdn_Inc_s	-0.068609	0.113720
Mdn_Inc_r	-0.046186	0.010228
mn_pm25_s	0.002598	0.024897

row.names(coef2)	V1	V2
mn_pm25_r	0.020363	0.027121
mn_tmp_s	0.000050	0.000096
mn_tmp_r	-0.000017	0.000005
Tre_Cnp	0.325314	0.058076

```
newhealth$ks_mh <- ks_mh$smooth
tract7$ks_mh[tract7$GEOID20 %in% newhealth$GEOID20] <- newhealth$ks_mh[newhealth$GEOID20 %in% tract7$GEOID20]
grid <- st_centroid(tract7[is.na(tract7$ks_mh),])
```

```
## Warning in st_centroid.sf(tract7[is.na(tract7$ks_mh), ]): st_centroid assumes
## attributes are constant over geometries of x
```

```
pred_MH <- krige(ks_mh ~1, newhealth, grid$geometry, model = v_fit)
```

```
## [using ordinary kriging]
```

```
tract7$ks_mh[is.na(tract7$ks_mh)] <- pred_MH$var1.pred
ggplot(data = tract7) + geom_sf(color = 'grey') + geom_sf(aes(fill = ks_mh)) +
  scale_fill_distiller(palette="Blues", trans = "reverse") + labs(title = 'Mental Health')
```

