CSSS 554: Assignment 3

Walker Azam
Due: February 22nd 2023
University of Washington

Assignment 3 Overview:

In this Assignment I will carry out SAE for smoking prevalence in health reporting areas (HRAs) in King County, using the BRFSS data. The smoking variable is labeled `smoker1`.

We can load some of the libraries will be using for this assignment, such as `SUMMER`:

```
library(SUMMER)
if (!isTRUE(requireNamespace("INLA", quietly = TRUE))) {
  install.packages("INLA",
                   repos = c(getOption("repos"),
                             INLA="https://inla.r-inla-download.org/R/stable"),
                   dep=TRUE)
}
library(sf) # Load sf for spatial analysis
library(prioritizr) # Allows us to create an adjacency matrix
library(survey)
library(ggplot2)
```

Reading in the Data:

`BRFSS` contains the full BRFSS dataset with 16,283 observations and `smoker1` will be a binary variable indicating whether an observation is a smoker or not. `strata` is the strata indicator and `rwt_llcp` is the final weight.

For cleaning, I will remove any rows with missing HRA codes or Smoker status prior to any analysis too.

```
data(BRFSS)
# Dropping missing smoker1 rows or missing HRA codes
BRFSS <- subset(BRFSS, !is.na(BRFSS$smoker1))
BRFSS <- subset(BRFSS, !is.na(BRFSS$hracode))
```

After cleaning, we have 16005 observations to work with. Throughout this assignment, for the Bayesian analyses, I will use the default INLA hyperpriors. I will let Y_i and n_i denote the number of smokers and the number sampled respectively, and define p_i to be the proportion of smokers in HRA $i, i = 1, ..., n$.

Question 1: Naive Estimation

As a naive first analysis, consider the binomial model:

$$y_i|p_i \quad \sim \quad iid \text{ Binomial}(n_i, p_i)$$

Calculate the sample proportions $\hat{p}_i = y_i/n_i$, and the standard errors $\sqrt{(\hat{p}_i(1-\hat{p}_i)/n_i)}$. Map the estimates and standard errors.

We have to first create a neighbor matrix:

```
data(KingCounty)
# cast as spatial dataframe
KingCounty <- st_as_sf(KingCounty)
# compute adjacency matrix
mat <- adjacency_matrix(KingCounty)
# Setting row and col name to HRA names in our neighbor
# matrix
colnames(mat) <- rownames(mat) <- KingCounty$HRA2010v2_
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
```

Now we can find Direct Naive Estimation by not specifying any of the survey weights information and using the smoothSurvey function from SUMMER:

```
# Naive Estimation (Amat = Null is used for IID)
smoothed <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = NULL,
    responseType = "binary", responseVar = "smoker1", strataVar = NULL,
    weightVar = NULL, regionVar = "hracode", clusterVar = NULL,
    CI = 0.95)
```

By specifying the `Amat` argument as NULL, I will be using a IID model (non-spatial). This can give us access to the naive estimates which can be mapped to regions from King County:

```
# assigning a dataframe for mapping purposes:
naive_df <- smoothed$HT
# find standard error by taking the square root of the
# variance
naive_df$naive_se <- sqrt(smoothed$HT$HT.var)

# mapping mean posterior estimates for the Naive Direct
# Estimates
mapPlot(data = naive_df, geo = KingCounty, variables = c("HT.est"),
    labels = c("Naive Direct Estimates"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Estimate Value")
```

Mapping the Standard Errors:

```
# mapping standard errors
mapPlot(data = naive_df, geo = KingCounty, variables = c("naive_se"),
    labels = c("Naive Standard Errors"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Standard Error")
```

Observations for Q1 Plots:

- We can see that there is a slight North-South trend in terms of Estimates. We see lower Estimates at the areas towards the north, and higher Estimates towards areas in the South.

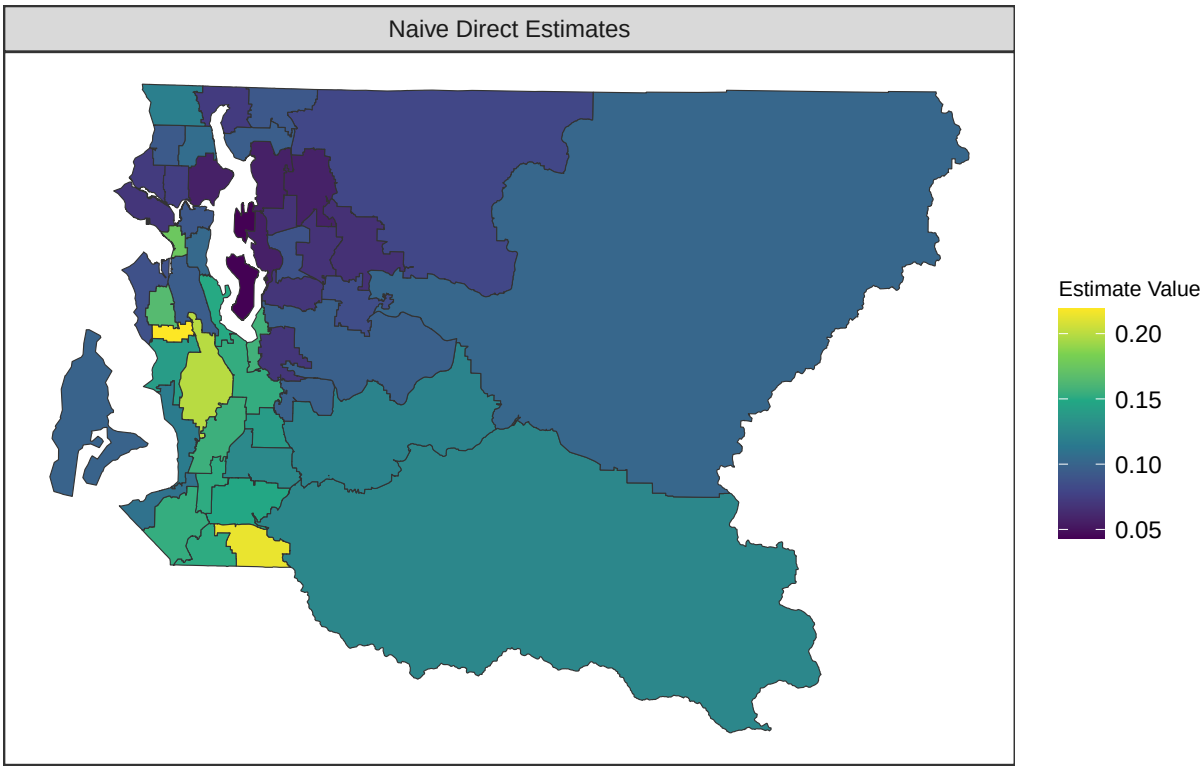

Figure 1: Map of Naive Estimates for Smokers in King County

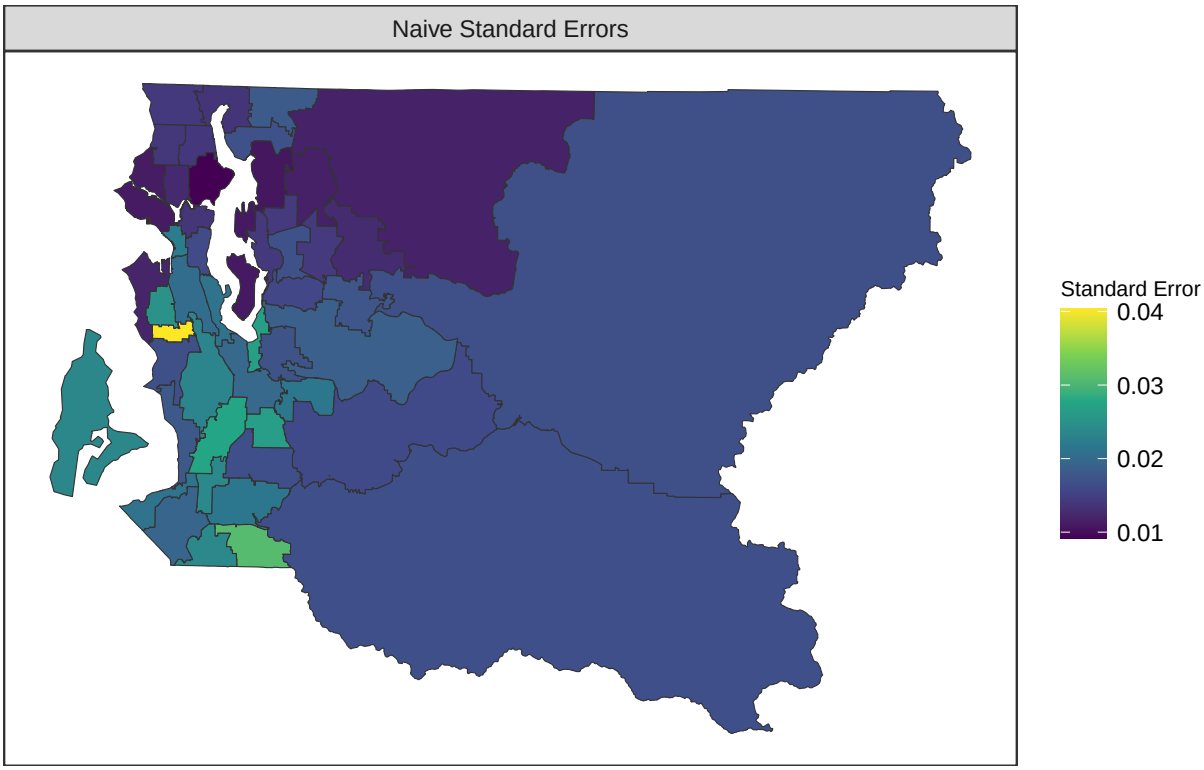

Figure 2: Map of Naive Estimates' Standard Errors for Smokers in King County

- Generally areas in the south-west had the highest estimates overall, with areas to the West having generally larger estimates than those towards the west.
- For the Standard Errors, we see that larger rural areas, specially towards the East, all have small errors.
- From both plots, the North Highline area seems like a 'hot-spot', since it has both a high estimate and error associated with the area. Although South-Auburn also has a high estimate, its standard error is noticeably smaller than North Highline.

Question 2: Weighted Estimation

Create a svydesign object and calculate weighted estimates, with associated standard errors and map each of these.

I can calculate the direct (weighted) estimates by using a `svydesign` object, which returns both the weighted estimates and standard errors. These estimates are also known as the Horvitz-Thompson estimates.

```
# Design object using strata and weights from BRFSS
# (rwt_llcp is final weights)
design <- svydesign(ids = ~1, weights = ~rwt_llcp, strata = ~strata,
    data = BRFSS)
# getting direct estimates
direct <- svyby(~smoker1, ~hracode, design, svymean)
```

I can map these estimates, and their corresponding standard errors:

```
data(KingCounty)
# mapping weighted estimates
mapPlot(data = direct, geo = KingCounty, variables = c("smoker1"),
    labels = c("Weighted Estimates"), by.data = "hracode", by.geo = "HRA2010v2_",
    legend.label = "Weighted Estimates")
```

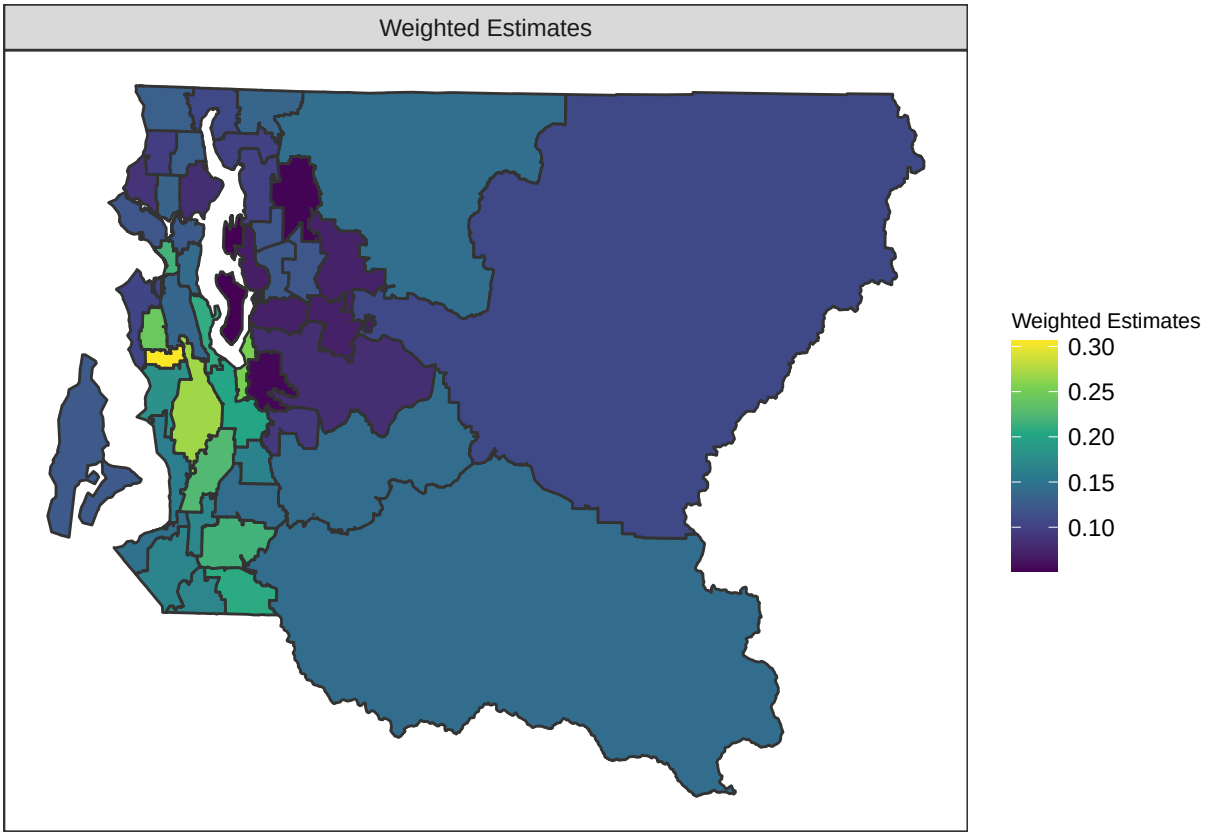

Figure 3: Map of Weighted Estimates for Smokers in King County

```
# mapping the standard errors
mapPlot(data = direct, geo = KingCounty, variables = c("se"),
    labels = c("Weighted Standard Errors"), by.data = "hracode",
    by.geo = "HRA2010v2_", legend.label = "Standard Errors")
```

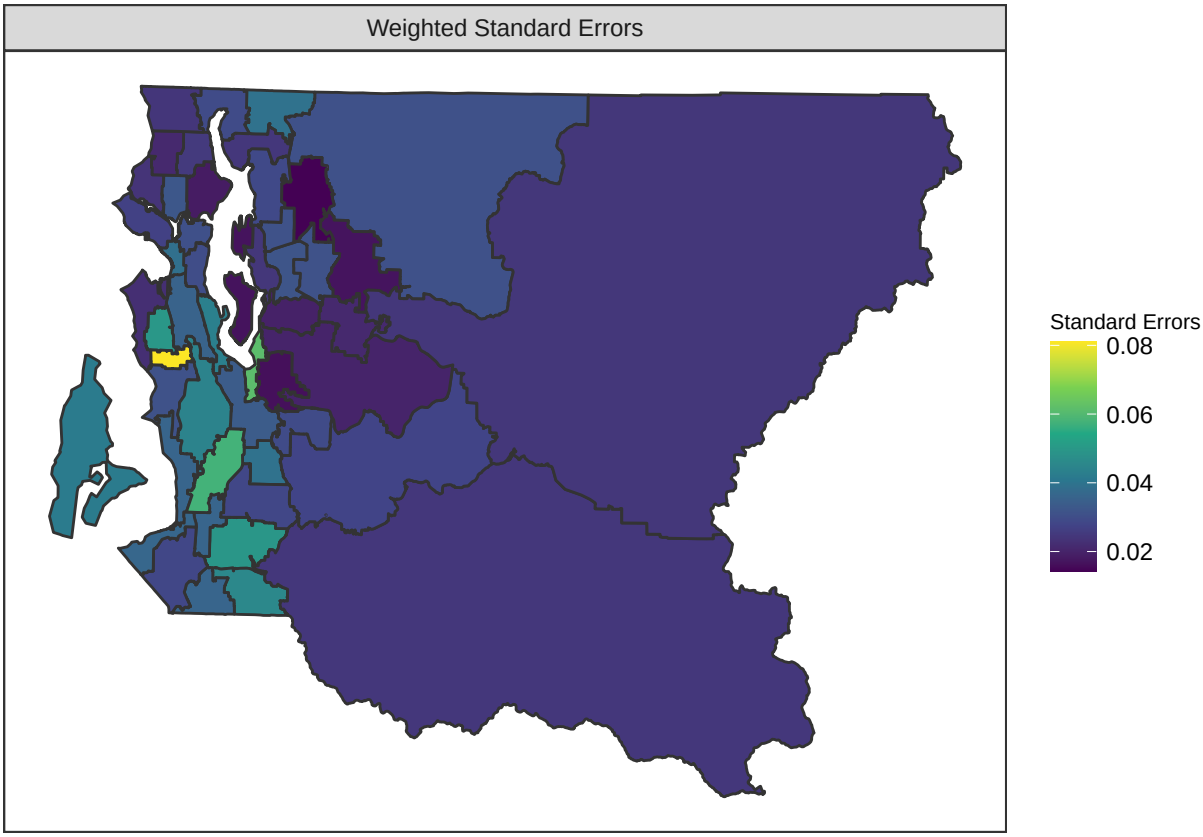

Figure 4: Map of Weighted Estimates' Standard Errors for Smokers

Observations for Q2 Plots:

- Using the survey weights, but not smoothing results, we can see that central and eastern areas of King County have lower estimates.
- The large areas in towards the center and east also have very small errors in addition.
- In general, compared to Q1, we see higher errors, and also a higher range for estimate values.
- North Highline has the largest prevalence and associated error too. Areas near Mercer Island/Bellevue have generally the lowest estimates and errors.

Question 3: Comparing Naive and Weighted Estimates

Plot the naive and weighted estimates of p_i against each other and comment. Plot the naive and weighted standard errors of p_i against each other and comment.

We can compare the two constructed models we have so far and see how the model's compare in terms of the direct estimates, and also their associated standard errors. First I'll take a look at the Estimates themselves:

```
# merging the naive and weighted dataframes for plotting
merged <- merge(naive_df, direct, by.x = c("region"), by.y = c("hracode"))

# plotting our results
ggplot(merged, aes(x = HT.est, y = smoker1)) + geom_point() +
    geom_abline(slope = 1, intercept = 0, color = "red") + ggtitle("Naive vs. Weighted Estimates") +
    xlab("Naive Estimates") + ylab("Weighted Estimates") + theme(plot.title = element_text(hjust = 0.5))
    xlim(0, 0.3) + ylim(0, 0.3)
```

Observations (Fig 5):

- It appears that the weighted estimates in general are all higher than the naive estimates (evident since most points fall above the red line). There are only a handful of small estimate regions below 0.1 where the naive estimate is higher than the weighted estimate.
- We also can see a greater range for the weighted estimates than for the naive estimates. The weighted estimates range from 0.05 - 0.31, whereas the naive range is 0.04 - 0.22.

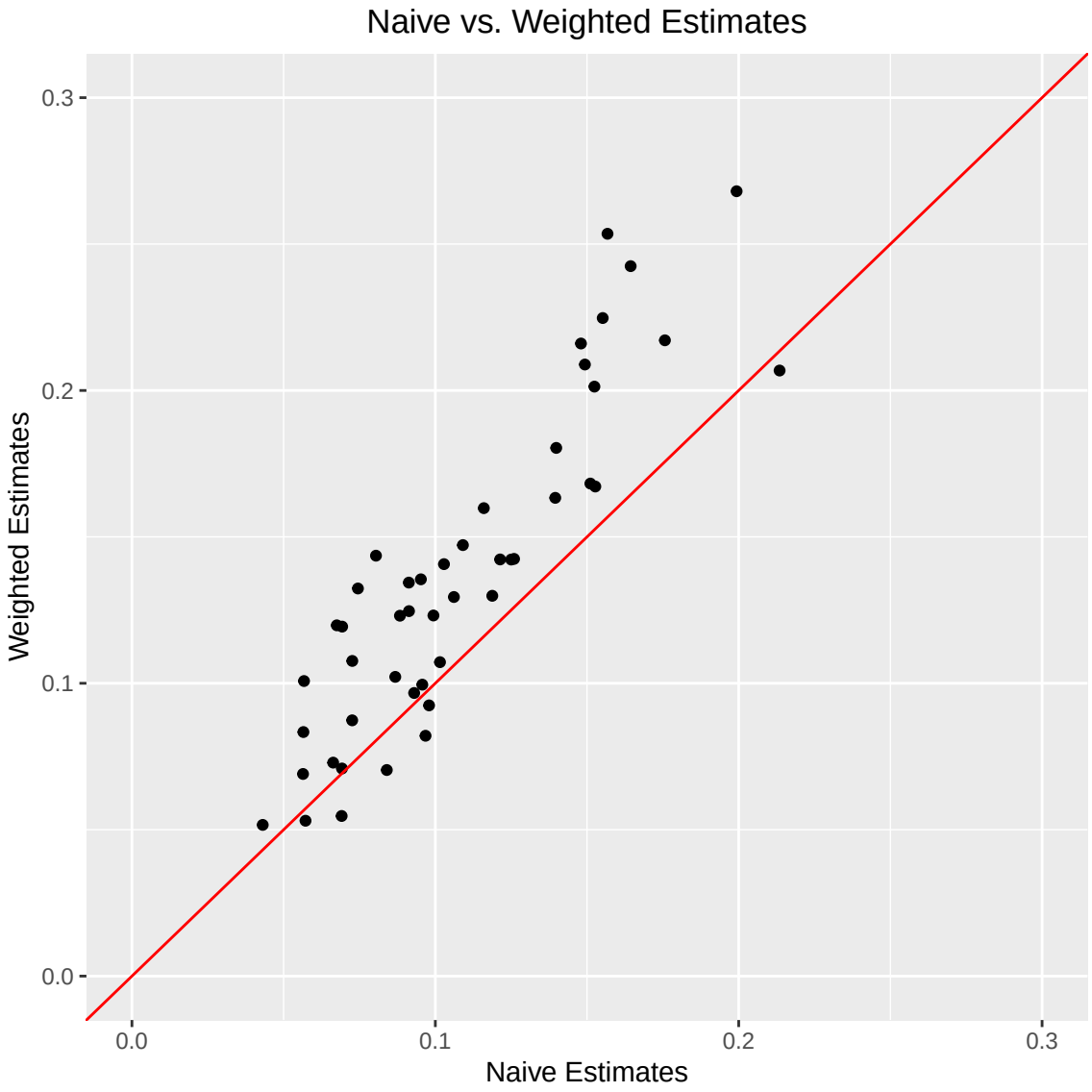

Figure 5: Naive vs. Weighted Estimates

We can also see how the standard errors compare:

```
ggplot(merged, aes(x = naive_se, y = se)) + geom_point() + geom_abline(slope = 1,
    intercept = 0, color = "red") + ggtitle("Naive vs. Weighted Standard Errors") +
    xlab("Naive SE") + ylab("Weighted SE") + theme(plot.title = element_text(hjust = 0.5)) +
    xlim(0, 0.1) + ylim(0, 0.1)
```

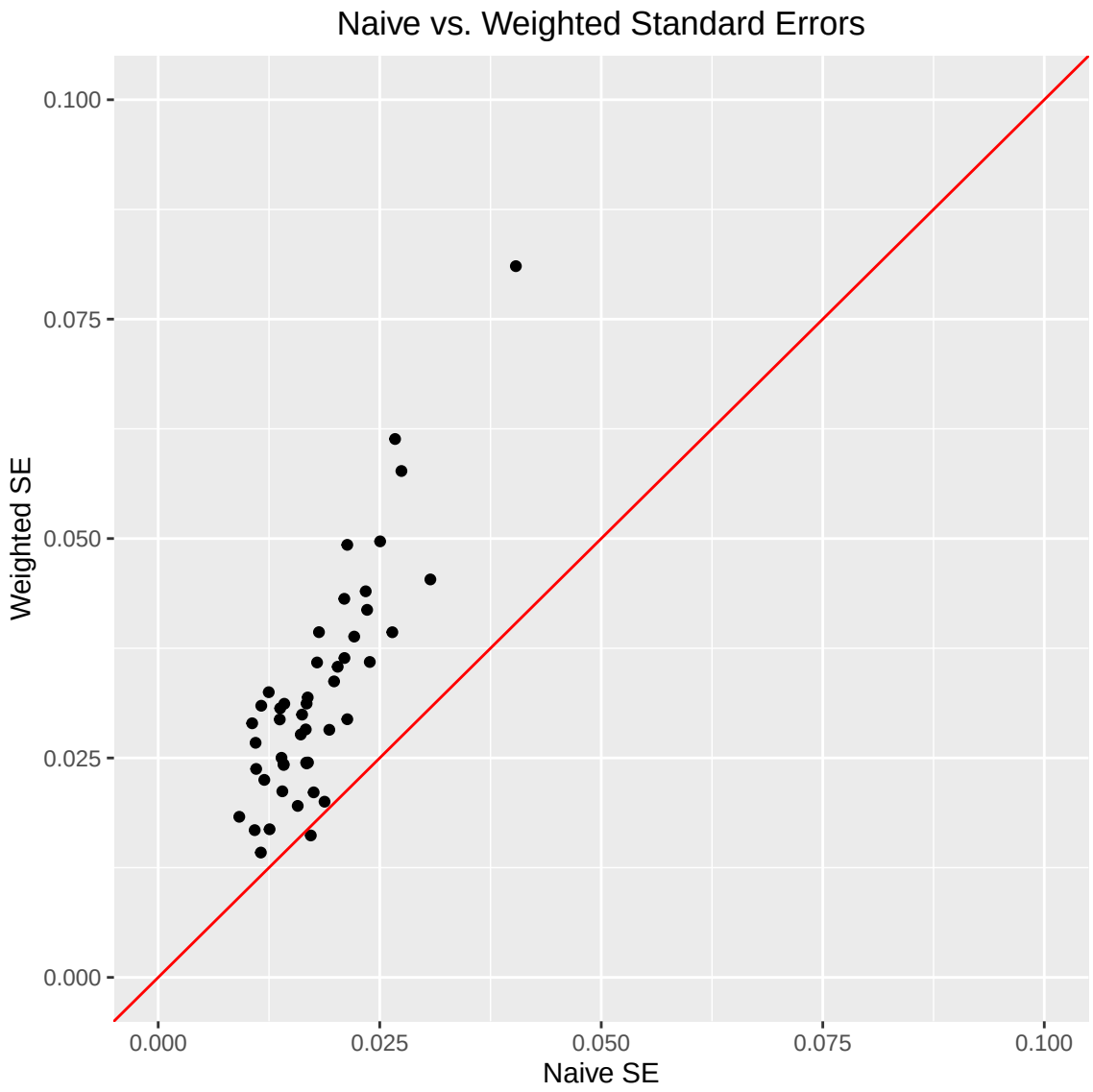

Figure 6: Naive vs. Weighted SE

Observations (Fig 6):

- The direct naive errors are all much smaller than the weighted errors.
- For the naive standard errors the range is also very similar across all regions, from 0.001 - 0.04 (in contrast the weighted standard errors range from 0.01 - 0.08). This indicates that the weighted error terms take more into account specific area qualities, which the naive direct estimates are not doing. We should expect different error terms based on the differing area sizes and populations.

Are the naive or the weighted the most appropriate summaries? Why?

Comparing the naive and weighted models, it appears that the naive model creates smaller standard errors. However, knowing that we are using responses from a, presumably complex, survey, I would reason that the weighted model's estimates are probably a little more accurate, despite the higher standard errors. This is because, we have to take into account the underlying survey design and weights appropriately when working with survey results. The Naive Direct Estimates are simple binomial probabilities, which can lead to unreliable results when we know that areas have a wide range in size and population. I don't think either these models suffice in portraying a reliable summary, but the Weighted Estimates seem more appropriate than the simple Binomial Naive Estimates.

Question 4: Smoothed Naive Estimation

Now we consider a simple binomial smoothing model utilizing BYM2 random effects. Fit this model using INLA and extract posterior medians and posterior standard deviations of p_i. Map these quantities.

I will fit the model using SUMMER's `smoothSurvey` function, which uses INLA smoothing with BYM2 if the `Amat` argument is passed as an adjacency matrix. We can specify NULL for survey weights to get smoothed naive estimates:

```
# Specifying Amat = mat to allow for BYM2 random effects
smoothed_bym2 <- smoothSurvey(data = BRFSS, geo = KingCounty,
    Amat = mat, responseType = "binary", responseVar = "smoker1",
    strataVar = NULL, weightVar = NULL, regionVar = "hracode",
    clusterVar = NULL, CI = 0.95)
```

I will map the posterior medians after extracting them from our smoothed model:

```
# extracting posterior medians from
# smoothed_bym2$smooth$median
mapPlot(data = smoothed_bym2$smooth, geo = KingCounty, variables = c("median"),
    labels = c("Smoothed Posterior Medians"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Posterior Medians")
```

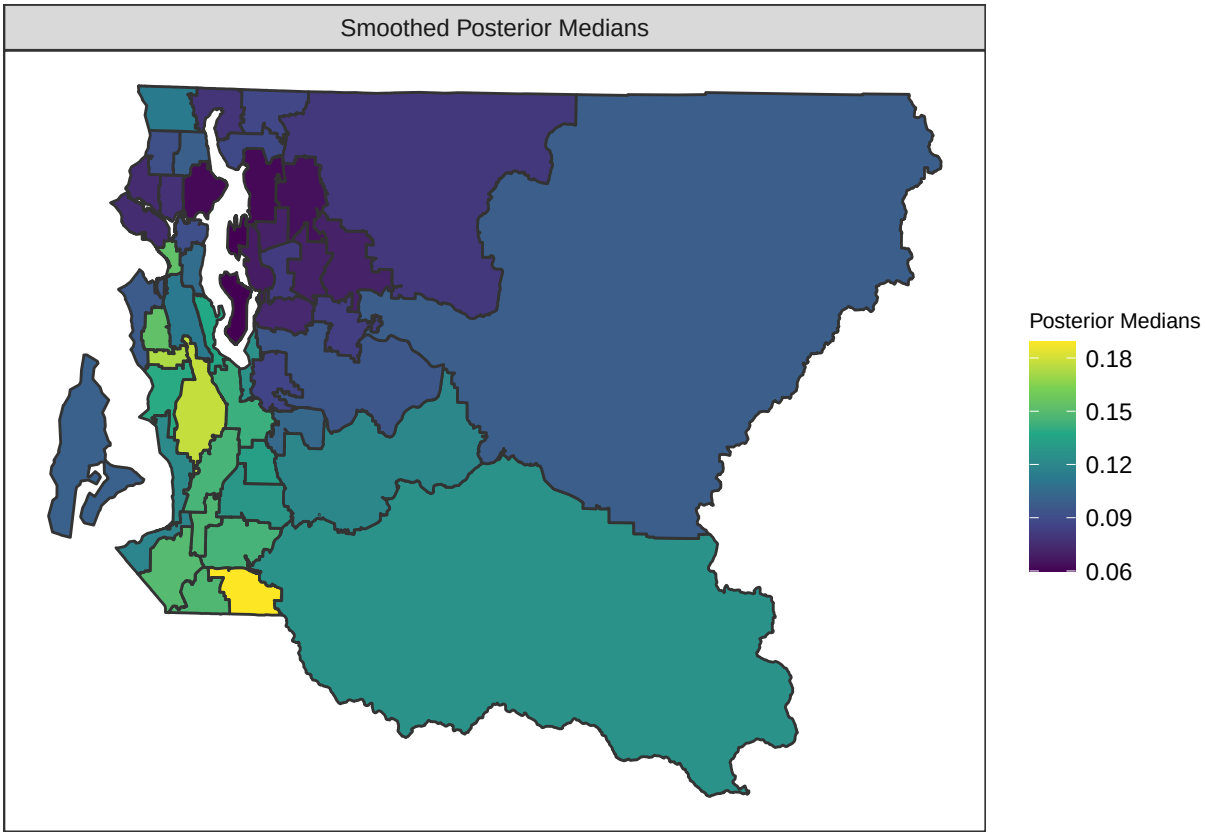

Figure 7: Posterior Medians using BYM2 Smoothing

Below I also map the posterior standard deviations from the INLA model using BYM2 smoothing:

```
# taking the square root of the variance
smoothed_bym2$smooth$std <- sqrt(smoothed_bym2$smooth$var)
# plotting
mapPlot(data = smoothed_bym2$smooth, geo = KingCounty, variables = c("std"),
    labels = c("Smoothed Posterior Standard Deviations"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Standard Deviation")
```

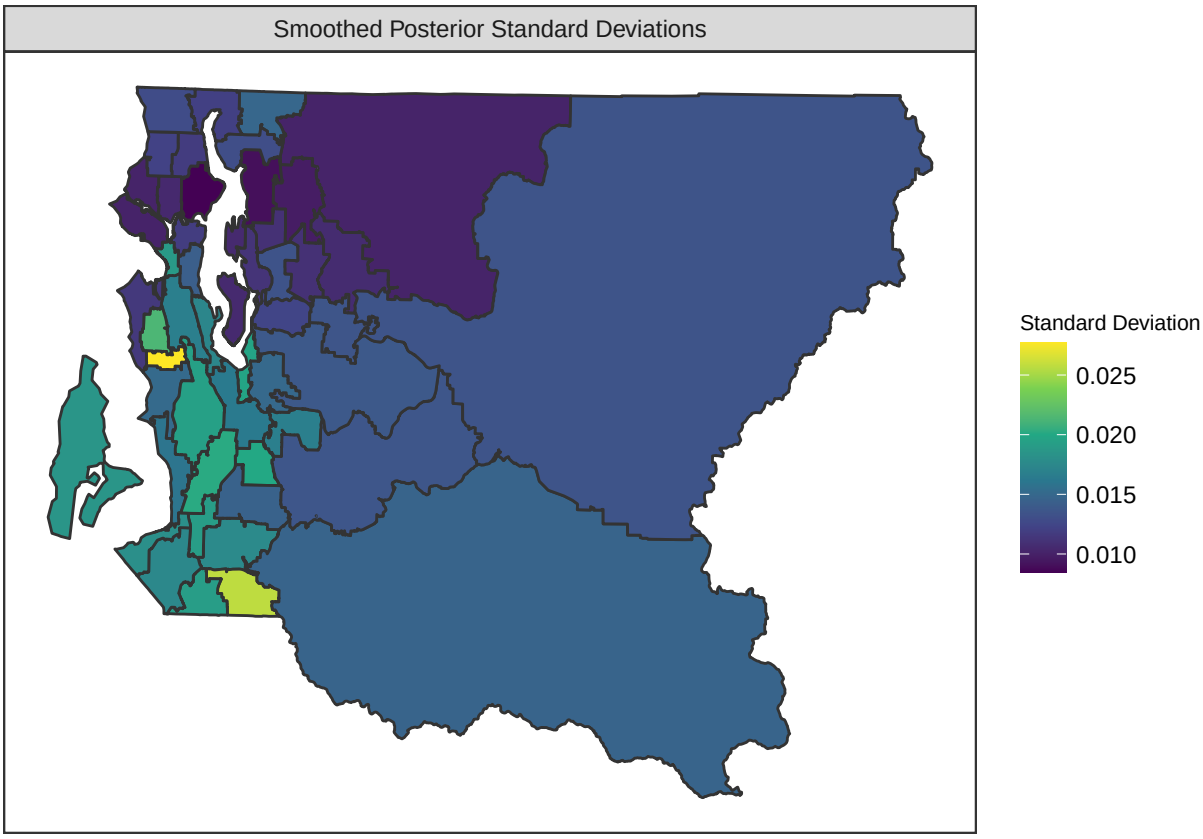

Figure 8: Posterior Standard Deviations using BYM2 Smoothing

Observations for Q4:

- This model uses spatial smoothing, but does not take into account survey weights. As a result, we do see adjacent areas being more similar to each other in terms of estimates. For example, the North Highline region's estimate is much lower in this plot compared to our Naive Direct Estimates.
- In general neighboring areas have lower contrast between them, and we can observe a north-south trend in terms of increasing estimates for areas.
- The posterior standard deviations have more error for some of more smoothed areas, for example North Highline and Vashon Island. The errors still remain lowest of the areas in the north, and towards the central/east.

Question 5: Comparing Naive and Smoothed Binomial Estimates

Plot the naive and smoothed binomial estimates of p_i against each other and comment. Plot the naive and smoothed binomial standard errors of p_i against each other and comment.

```
# merging naive and smoothed results
merged2 <-  merge(naive_df, smoothed_bym2$smooth, by = 'region')
# plotting...
ggplot(merged2, aes(x = HT.est, y = median)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  ggtitle("Naive vs. Smoothed Estimates") +
  xlab("Naive Estimates") +
  ylab("Smoothed Estimates") +
  theme(plot.title = element_text(hjust = 0.5)) + xlim(0, 0.25) + ylim(0, 0.25)
```

Observations (Fig 9):

- We can see a lot of alignment between the Naive estimates and the smoothed estimates!
- Towards the high naive estimates (>=0.15), we can see shrinkage due to smoothing, since the estimates become smaller for the Smoothed model.
- At the lower extreme, we can also see shrinkage, with a handful of estimates around 0.06 increase in value for the smoothed model compared to the naive output.

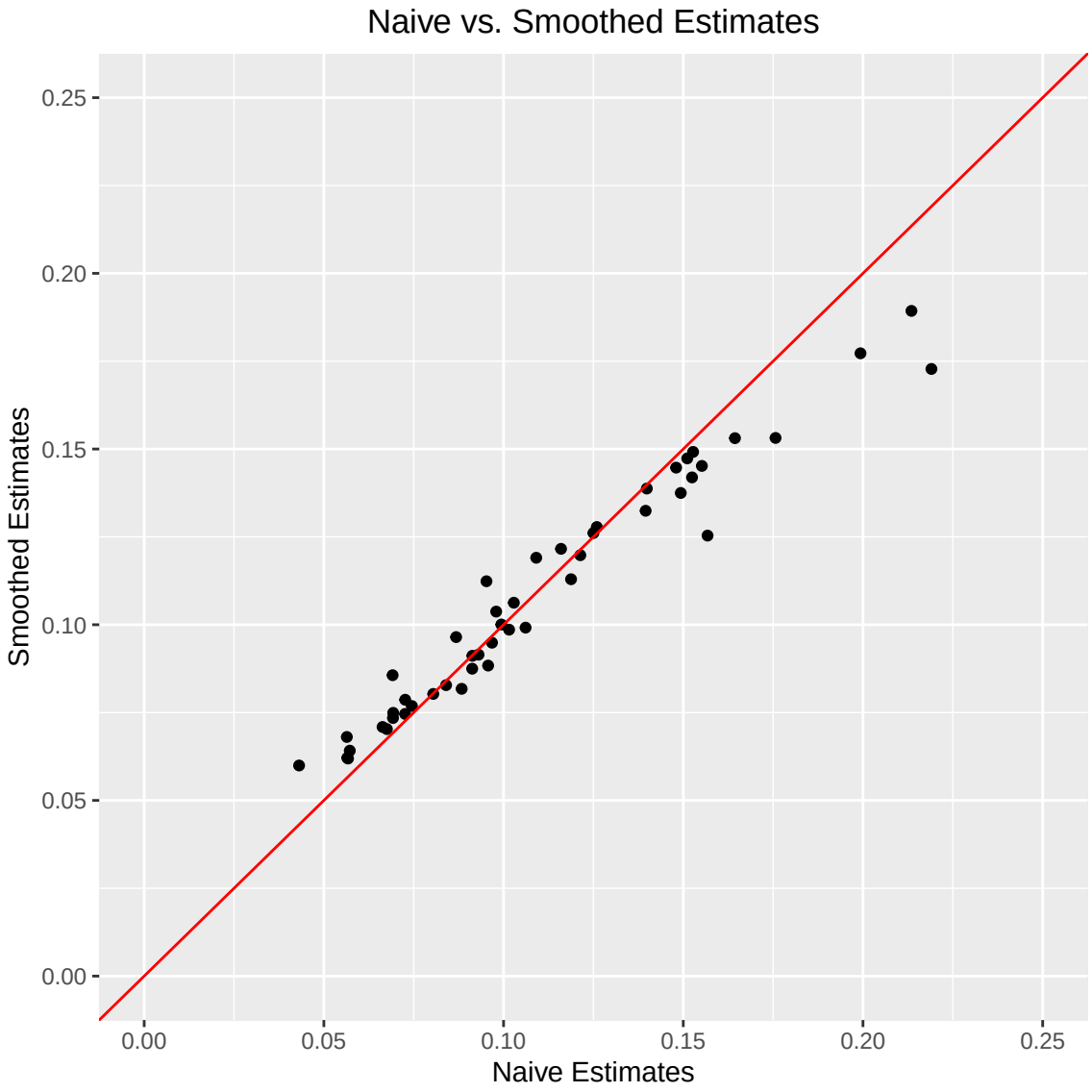

Figure 9: Naive vs. Smoothed Estimates

```r
# using square root of variance
merged2$smoothed_se <- sqrt(merged2$var)
# plotting
ggplot(merged2, aes(x = naive_se, y = smoothed_se)) + geom_point() +
    geom_abline(slope = 1, intercept = 0, color = "red") + ggtitle("Naive vs. Smoothed Standard Errors")
    xlab("Naive SE") + ylab("Smoothed SE") + theme(plot.title = element_text(hjust = 0.5)) +
    xlim(0, 0.04) + ylim(0, 0.04)
```

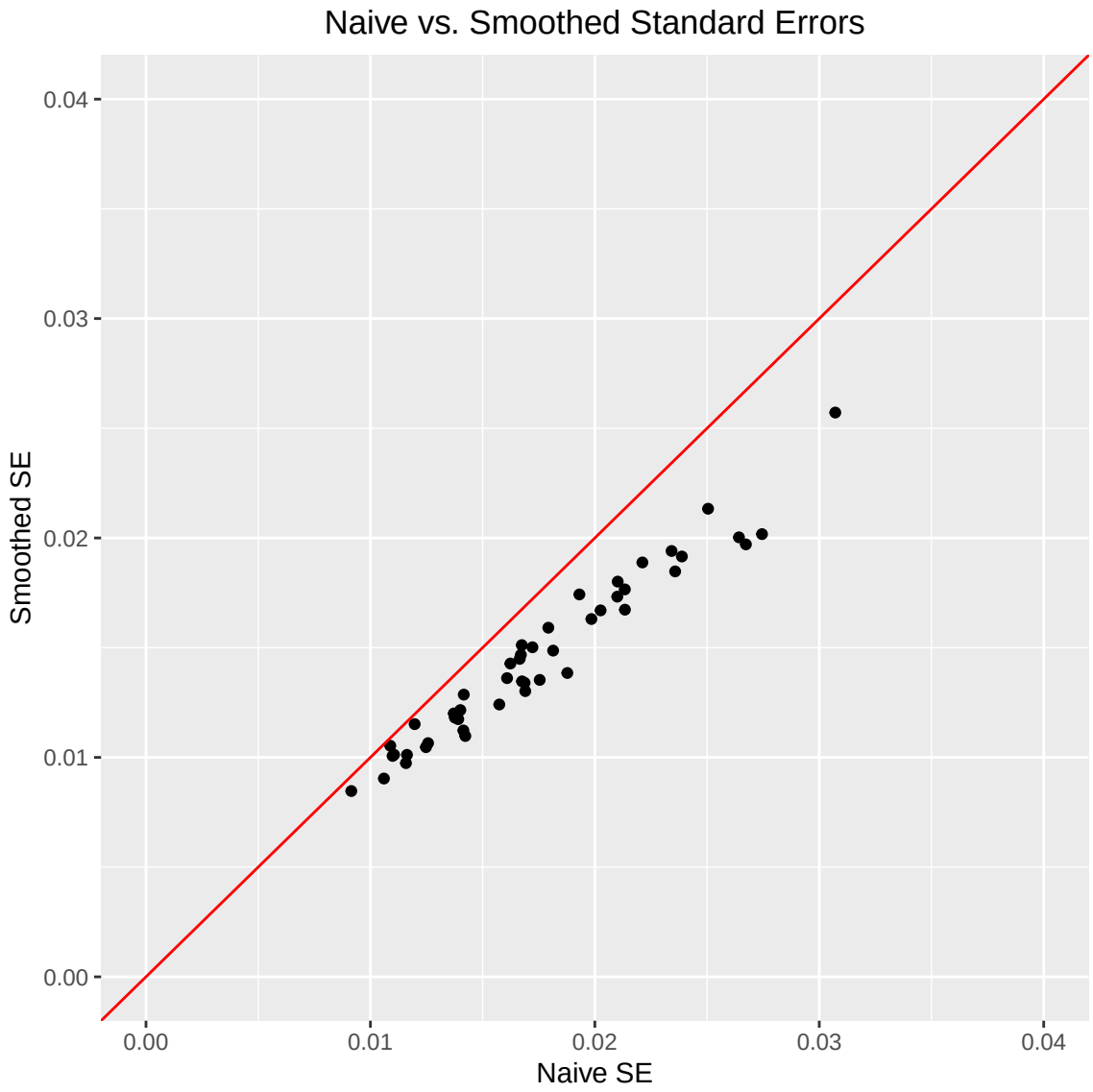

Figure 10: Naive vs. Smoothed SE

Observations (Fig 10):

- Clearly the smoothed model has much lower errors for all the regions compared to the naive model. We can see a greater reduction in error as the errors get higher in value for the naive model.
- Again, this can be attributed to the spatial BYM2 smoothing employed by the model from Q4.

- Considering that the estimates are pretty aligned between the two models, yet the smoothed model has lower errors, I would certainly suggest using the latter model over the former.

Question 6: Smoothed Weighted Estimations

Fit a model using the SUMMER package for Smoothed Weighted Estimation with BYM2 random effects. Extract posterior medians and posterior standard deviations of p_i. Map these quantities.

We can use BYM2 smoothing and also pass in Survey Weight information this time:

```
# Specifying Amat = mat to allow for BYM2 random effects
FHmodel <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = mat,
    responseType = "binary", responseVar = "smoker1", strataVar = "strata",
    weightVar = "rwt_llcp", regionVar = "hracode", clusterVar = "~1",
    CI = 0.95)
```

Below I map the posterior medians for this model:

```
# extracting posterior medians: FHmodel$smooth$median
mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = c("median"),
    labels = c("Smoothed Weighted Posterior Medians"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Posterior Medians")
```

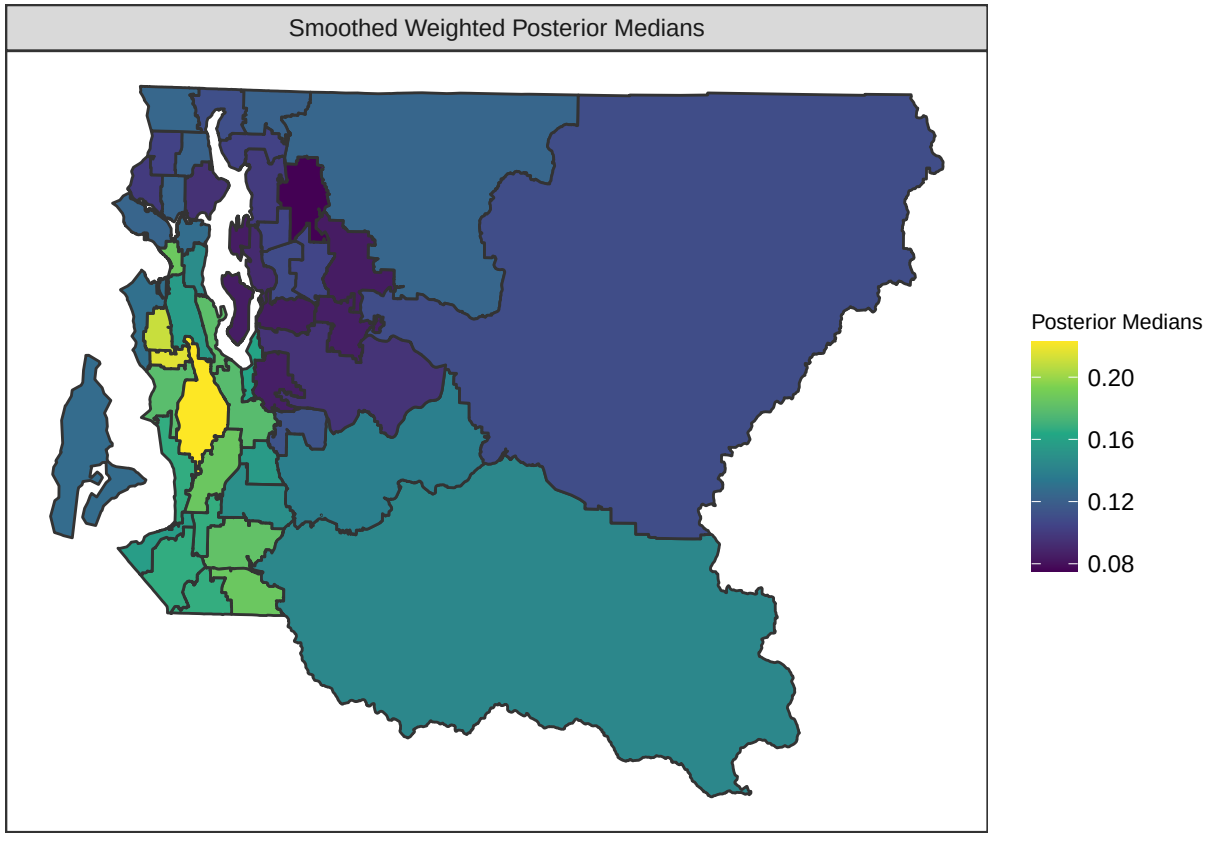

Figure 11: Smoothed Weighted Estimations

And we can also see the posterior standard deviations:

```
# getting standard deviation by taking the sqrt of variance
FHmodel$smooth$std <- sqrt(FHmodel$smooth$var)
# plotting...
mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = c("std"),
    labels = c("Smoothed Weighted Standard Deviations"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Standard Deviation")
```

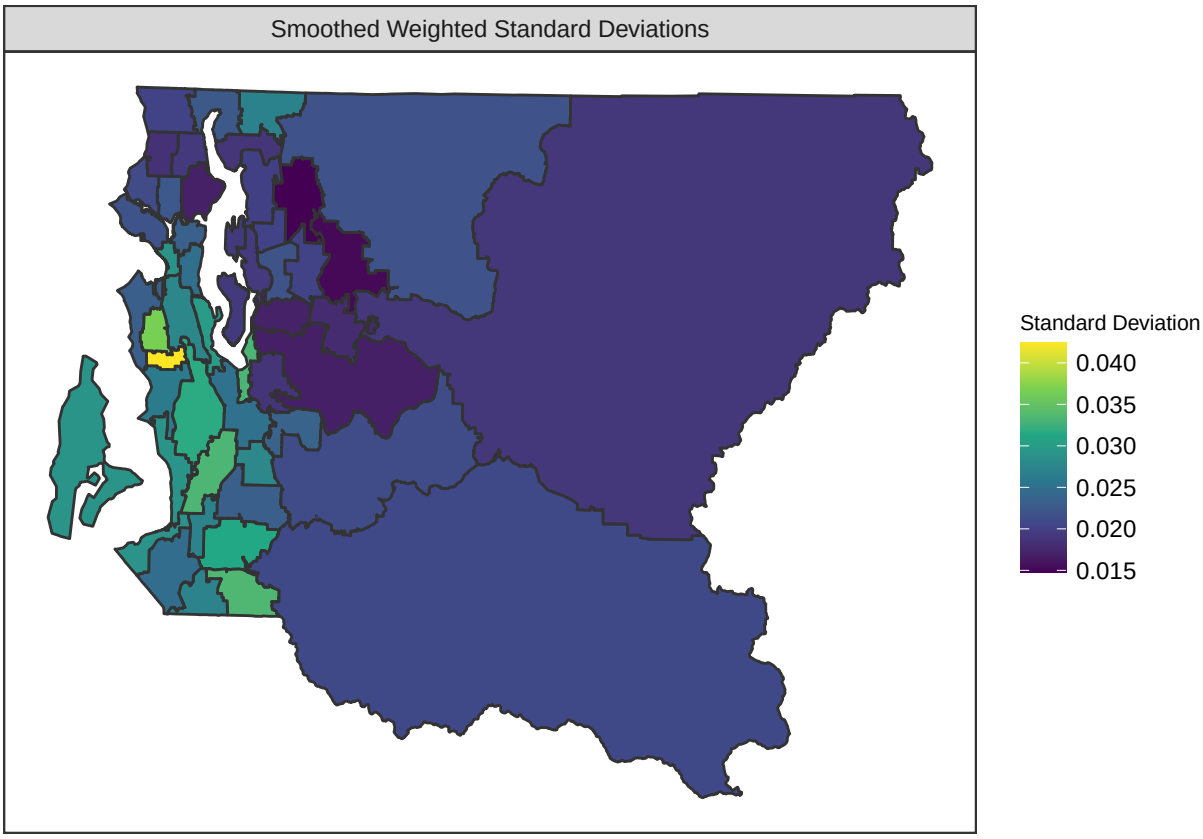

Figure 12: Fay-Herriot Standard Deviations

Observations for Q6 plots:

- Using both the survey weights and smoothing, the SeaTac/Tukwila region has the highest estimate value. We still see the central areas with the lowest estimates, but the regions in the south (and southwest) are a lot more similar to each other, with fewer areas of high contrast.
- The north-south trend is less evident in this model, but areas to the west do generally have a higher estimate than to the east.
- As for the standard deviations, the large areas towards to center and east all have low standard deviations.

Question 7: Comparing Weighted and Smoothed Weighted Estimates

Plot the weighted and smoothed weighted estimates of p_i against each other and comment. Plot the weighted and smoothed weighted standard errors of p_i against each other and comment.

```
merged3 <-  merge(direct, FHmodel$smooth, by.x = c('hracode'), by.y = c('region') )

ggplot(merged3, aes(x = smoker1, y = median)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  ggtitle("Weighted vs. Smoothed Weighted Estimates") +
  xlab("Weighted Estimates") +
  ylab("Smoothed Weighted Estimates") +
  theme(plot.title = element_text(hjust = 0.5)) + xlim(0, 0.35) + ylim(0, 0.35)
```

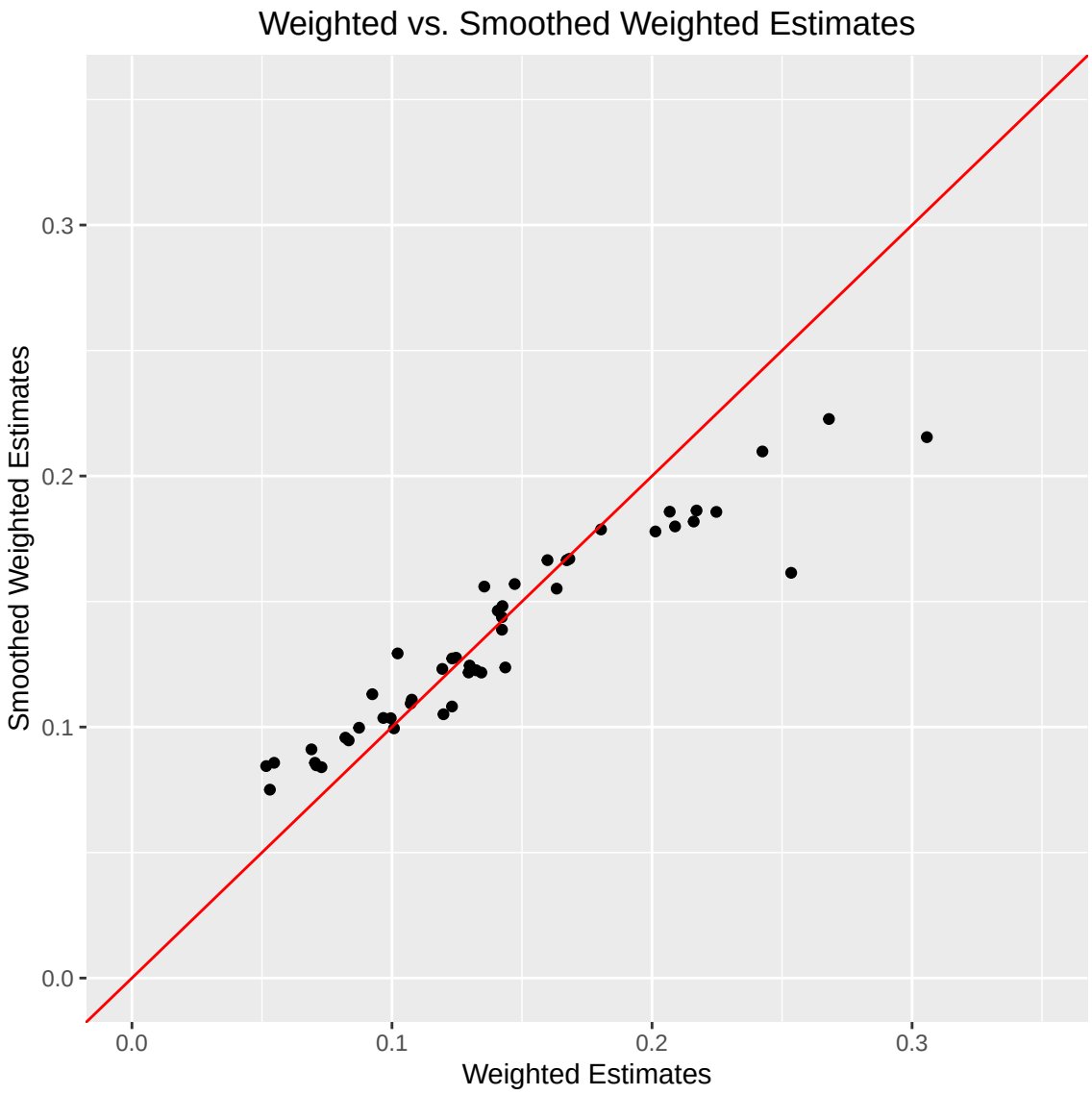

Figure 13: Weighted vs. Smoothed Weighted Estimates

Observations (Fig 13):

- Compared to the weighted estimates, the smoothed weighted estimates see shrinkage due to the BYM2 smoothing. For example, points above 0.2 are lower in the smoothed weighted model (fall below the red line).
- We see this to a lesser extent in the other extreme, where smaller estimates are higher in the smoothed model.
- The range for the smoothed weighted estimates is also smaller than that of the weighted estimates with no smoothing.
- A lot of areas are in alignment, specially between 0.1 and 0.2 values.

```
merged3$std <- sqrt(FHmodel$smooth$var)

ggplot(merged3, aes(x = se, y = std)) + geom_point() + geom_abline(slope = 1,
    intercept = 0, color = "red") + ggtitle("Weighted vs. Smoothed Weighted Standard Errors") +
    xlab("Weighted SE") + ylab("Smoothed Weighted SE") + theme(plot.title = element_text(hjust = 0.5)) +
    xlim(0, 0.09) + ylim(0, 0.09)
```

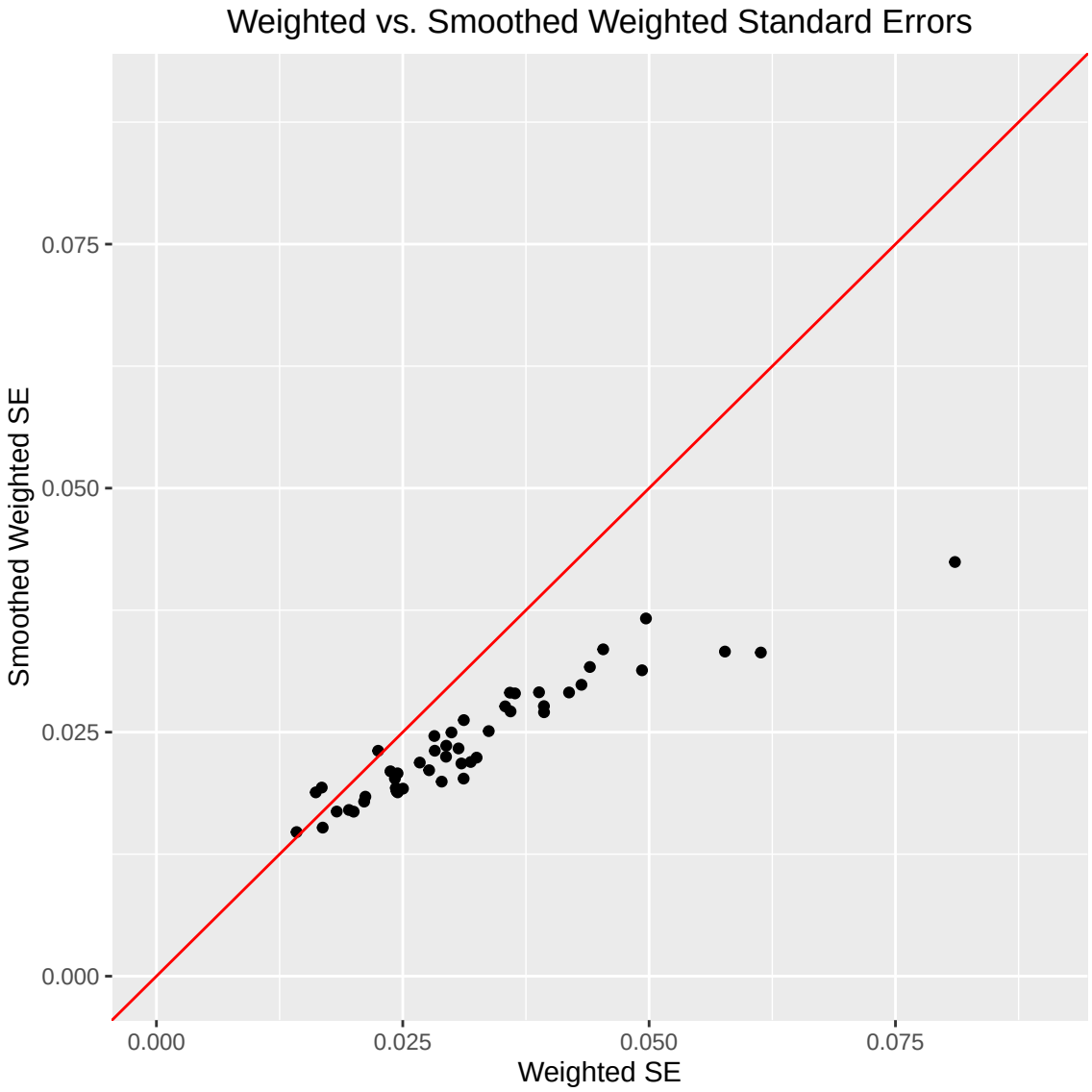

Figure 14: Naive vs. Smoothed SE

Observations (Fig 14):

- We can see that for almost all areas, the error is lowered using the smoothing over using weighted alone.
- There is a much wider range for errors in the weighted model (between 0.01 and 0.08) than in the smooth weighted model (0.01 to 0.04). Here smoothing does a lot to reduce the errors.
- The reduction is most noticeable for the high extremes on the weighted SE axis.

Which of the weighted or the smoothed weighted would you recommend using? Why?

I would recommend using the smoothed weighted model over the weighted model alone. When comparing the estimates for the areas between the two models, we see relatively similar estimates, with the smoothed output having a little shrinkage. However, when looking at the standard errors, we can see that the smoothed weighted model has lower errors than the weighted model. I would be more confident in the predictive abilities of the smoothed weighted model, as a result.

Question 8: Summarize the HRA variation in smoking prevalence across King County

We can see how each model's estimates compare to each other across HRA's in King County:

```
library(gridExtra)
p1 <- mapPlot(data = naive_df, geo = KingCounty, variables = c("HT.est"),
    labels = c("Naive Direct Estimates"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Estimate Value")
p2 <- mapPlot(data = direct, geo = KingCounty, variables = c("smoker1"),
    labels = c("Weighted Estimates"), by.data = "hracode", by.geo = "HRA2010v2_",
    legend.label = "Weighted Estimates")
p3 <- mapPlot(data = smoothed_bym2$smooth, geo = KingCounty,
    variables = c("median"), labels = c("Smoothed Posterior Medians"),
    by.data = "region", by.geo = "HRA2010v2_", legend.label = "Posterior Medians")
p4 <- mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = c("median"),
    labels = c("Smoothed Weighted Posterior Medians"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Posterior Medians")
grid.arrange(grobs = list(p1, p2, p3, p4), ncol = 2)
```

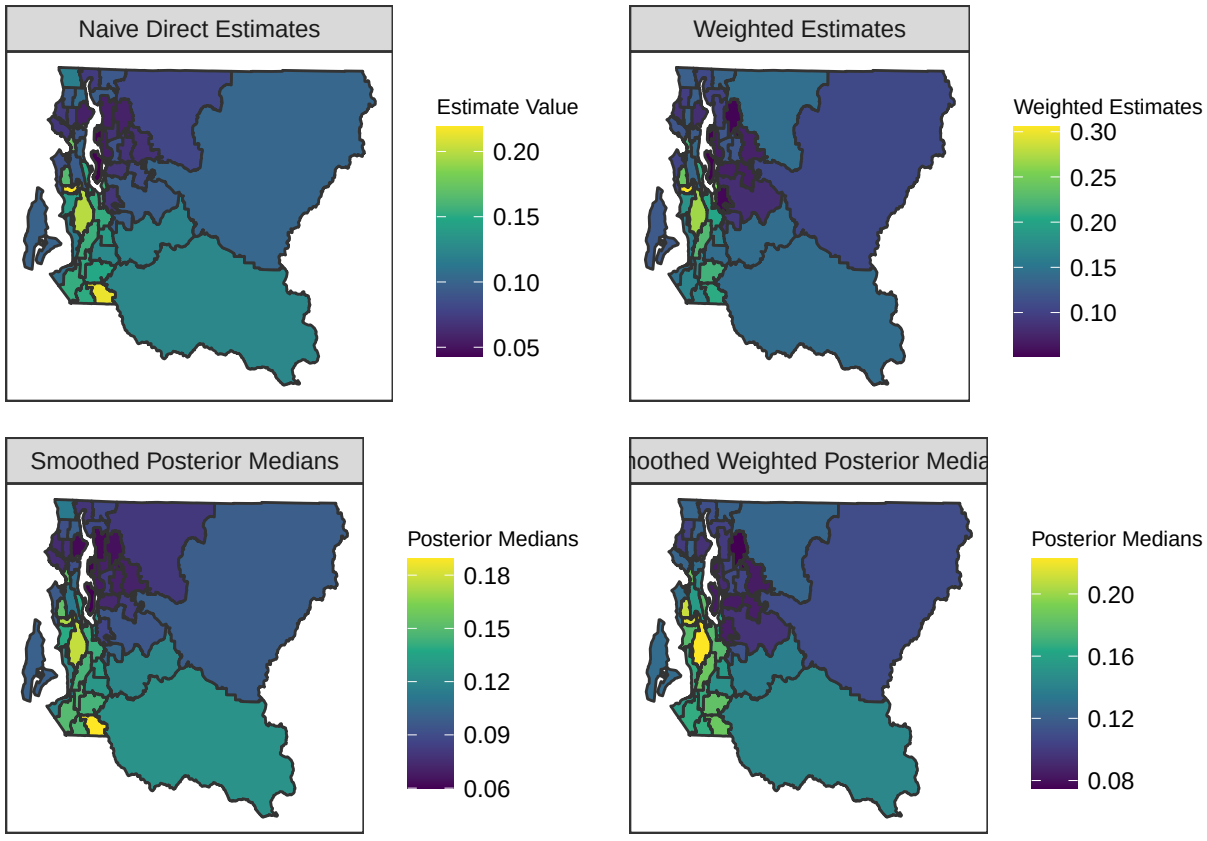

Figure 15: Summary of All Model Estimates

From the maps above, we can see that each model has some large, and some subtle, changes in estimates. Across all models we see estimates are lowest in the central/center-north areas. The smaller areas towards the south/south-west have higher estimates for smoking. Comparing the smoothed variations of the models

to those direct estimates, we see a smaller range of estimates. In the Naive Direct Estimate and Smoothed Unweighted models, the South-Auburn area is relatively a hot spot. However, in the weighted and weighted smooth models, this region stands out a lot less. In all models, the North Highline region remains one of the area's with the highest estimates.

```
library(gridExtra)
e1 <- mapPlot(data = naive_df, geo = KingCounty, variables = c("naive_se"),
    labels = c("Naive Standard Errors"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Error Value")
e2 <- mapPlot(data = direct, geo = KingCounty, variables = c("se"),
    labels = c("Weighted Standard Errors"), by.data = "hracode",
    by.geo = "HRA2010v2_", legend.label = "Error Value")
e3 <- mapPlot(data = smoothed_bym2$smooth, geo = KingCounty,
    variables = c("std"), labels = c("Smoothed Posterior Standard Deviation"),
    by.data = "region", by.geo = "HRA2010v2_", legend.label = "Error Value")
e4 <- mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = c("std"),
    labels = c("Smoothed Weighted Posterior Standard Deviations"),
    by.data = "region", by.geo = "HRA2010v2_", legend.label = "Error Value")
grid.arrange(grobs = list(e1, e2, e3, e4), ncol = 2)
```

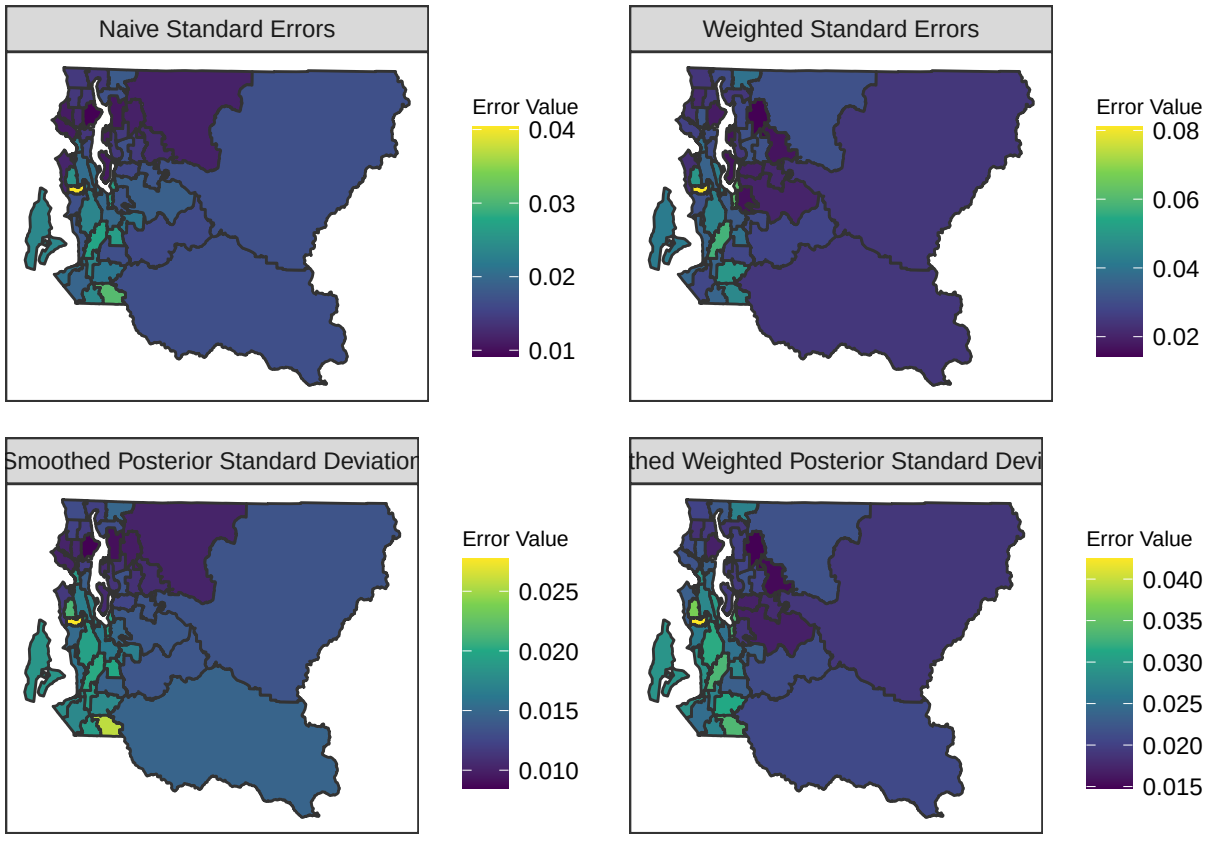

Figure 16: Summary of All Model Errors

Seeing the errors for each associated map, we can see that the smoothed models all have much lower error ranges than the direct models. Across all maps, the larger regions towards the East all had the lowest errors. Its also worth noting that the weighted model variants both had higher error values that their direct

counterpart models. However as previously discussed, since these results stem from a survey, the weighted models are more appropriate since it uses survey weights. The smoothed direct model has the lowest error values overall, but since it doesn't account for the survey design, or the underlying sampling method and biases, the results may not be as accurate as the weighted smoothed model.

Overall, looking at the most appropriate model for this problem set – the Smoothed Weighted Map – we can make some regional observations. Using the posterior medians, we can summarise the HRA Variation in smoking where the regions towards the east have lower estimates that those in the west. Overall prevalence is highest in the western areas. Moreover, regions in the north tend to have lower estimates than towards the south. Area with the highest prevalence is SeaTac/Tukwila, North Highline and Delridge, which as are close to each other near the west. The central areas directly east of, and including, Mercer Island all have the lowest estimates, with associated low errors, such as Bellevue etc... The area with the higher estimates, also have higher errors, whereas the more rural areas have low errors and estimates. Areas north of Lake Union also have very low estimates and errors.

Appendix: R-Code Provided

Below I am providing the R code used in this assignment as an Appendix:

```
library(knitr)
# Setting up RMarkdown
opts_chunk$set(collapse = TRUE, fig.align = "center", tidy = TRUE,
    tidy.opts = list(blank = TRUE, width.cutoff = 60, strip.white = TRUE),
    warning = FALSE, message = FALSE, cache = FALSE)
library(SUMMER)
if (!isTRUE(requireNamespace("INLA", quietly = TRUE))) {
    install.packages("INLA", repos = c(getOption("repos"), INLA = "https://inla.r-inla-download.org/R/s
        dep = TRUE)
}
library(sf)  # Load sf for spatial analysis
library(prioritizr)  # Allows us to create an adjacency matrix
library(survey)
library(ggplot2)
data(BRFSS)
# Dropping missing smoker1 rows or missing HRA codes
BRFSS <- subset(BRFSS, !is.na(BRFSS$smoker1))
BRFSS <- subset(BRFSS, !is.na(BRFSS$hracode))
data(KingCounty)
# cast as spatial dataframe
KingCounty <- st_as_sf(KingCounty)
# compute adjacency matrix
mat <- adjacency_matrix(KingCounty)
# Setting row and col name to HRA names in our neighbor
# matrix
colnames(mat) <- rownames(mat) <- KingCounty$HRA2010v2_
mat <- as.matrix(mat[1:dim(mat)[1], 1:dim(mat)[1]])
# Naive Estimation (Amat = Null is used for IID)
smoothed <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = NULL,
    responseType = "binary", responseVar = "smoker1", strataVar = NULL,
    weightVar = NULL, regionVar = "hracode", clusterVar = NULL,
    CI = 0.95)
# assigning a dataframe for mapping purposes:
naive_df <- smoothed$HT
# find standard error by taking the square root of the
# variance
naive_df$naive_se <- sqrt(smoothed$HT$HT.var)

# mapping mean posterior estimates for the Naive Direct
# Estimates
mapPlot(data = naive_df, geo = KingCounty, variables = c("HT.est"),
    labels = c("Naive Direct Estimates"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Estimate Value")
# mapping standard errors
mapPlot(data = naive_df, geo = KingCounty, variables = c("naive_se"),
    labels = c("Naive Standard Errors"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Standard Error")
# Design object using strata and weights from BRFSS
# (rwt_llcp is final weights)
design <- svydesign(ids = ~1, weights = ~rwt_llcp, strata = ~strata,
```

```
    data = BRFSS)
# getting direct estimates
direct <- svyby(~smoker1, ~hracode, design, svymean)
data(KingCounty)
# mapping weighted estimates
mapPlot(data = direct, geo = KingCounty, variables = c("smoker1"),
    labels = c("Weighted Estimates"), by.data = "hracode", by.geo = "HRA2010v2_",
    legend.label = "Weighted Estimates")
# mapping the standard errors
mapPlot(data = direct, geo = KingCounty, variables = c("se"),
    labels = c("Weighted Standard Errors"), by.data = "hracode",
    by.geo = "HRA2010v2_", legend.label = "Standard Errors")
# merging the naive and weighted dataframes for plotting
merged <- merge(naive_df, direct, by.x = c("region"), by.y = c("hracode"))

# plotting our results
ggplot(merged, aes(x = HT.est, y = smoker1)) + geom_point() +
    geom_abline(slope = 1, intercept = 0, color = "red") + ggtitle("Naive vs. Weighted Estimates") +
    xlab("Naive Estimates") + ylab("Weighted Estimates") + theme(plot.title = element_text(hjust = 0.5))
    xlim(0, 0.3) + ylim(0, 0.3)
ggplot(merged, aes(x = naive_se, y = se)) + geom_point() + geom_abline(slope = 1,
    intercept = 0, color = "red") + ggtitle("Naive vs. Weighted Standard Errors") +
    xlab("Naive SE") + ylab("Weighted SE") + theme(plot.title = element_text(hjust = 0.5)) +
    xlim(0, 0.1) + ylim(0, 0.1)
# Specifying Amat = mat to allow for BYM2 random effects
smoothed_bym2 <- smoothSurvey(data = BRFSS, geo = KingCounty,
    Amat = mat, responseType = "binary", responseVar = "smoker1",
    strataVar = NULL, weightVar = NULL, regionVar = "hracode",
    clusterVar = NULL, CI = 0.95)
# extracting posterior medians from
# smoothed_bym2$smooth$median
mapPlot(data = smoothed_bym2$smooth, geo = KingCounty, variables = c("median"),
    labels = c("Smoothed Posterior Medians"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Posterior Medians")
# taking the square root of the variance
smoothed_bym2$smooth$std <- sqrt(smoothed_bym2$smooth$var)
# plotting
mapPlot(data = smoothed_bym2$smooth, geo = KingCounty, variables = c("std"),
    labels = c("Smoothed Posterior Standard Deviations"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Standard Deviation")
# merging naive and smoothed results
merged2 <- merge(naive_df, smoothed_bym2$smooth, by = "region")
# plotting...
ggplot(merged2, aes(x = HT.est, y = median)) + geom_point() +
    geom_abline(slope = 1, intercept = 0, color = "red") + ggtitle("Naive vs. Smoothed Estimates") +
    xlab("Naive Estimates") + ylab("Smoothed Estimates") + theme(plot.title = element_text(hjust = 0.5))
    xlim(0, 0.25) + ylim(0, 0.25)
# using square root of variance
merged2$smoothed_se <- sqrt(merged2$var)
# plotting
ggplot(merged2, aes(x = naive_se, y = smoothed_se)) + geom_point() +
    geom_abline(slope = 1, intercept = 0, color = "red") + ggtitle("Naive vs. Smoothed Standard Errors")
    xlab("Naive SE") + ylab("Smoothed SE") + theme(plot.title = element_text(hjust = 0.5)) +
```

```
    xlim(0, 0.04) + ylim(0, 0.04)
# Specifying Amat = mat to allow for BYM2 random effects
FHmodel <- smoothSurvey(data = BRFSS, geo = KingCounty, Amat = mat,
    responseType = "binary", responseVar = "smoker1", strataVar = "strata",
    weightVar = "rwt_llcp", regionVar = "hracode", clusterVar = "~1",
    CI = 0.95)
# extracting posterior medians: FHmodel$smooth$median
mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = c("median"),
    labels = c("Smoothed Weighted Posterior Medians"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Posterior Medians")
# getting standard deviation by taking the sqrt of variance
FHmodel$smooth$std <- sqrt(FHmodel$smooth$var)
# plotting...
mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = c("std"),
    labels = c("Smoothed Weighted Standard Deviations"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Standard Deviation")
merged3 <- merge(direct, FHmodel$smooth, by.x = c("hracode"),
    by.y = c("region"))

ggplot(merged3, aes(x = smoker1, y = median)) + geom_point() +
    geom_abline(slope = 1, intercept = 0, color = "red") + ggtitle("Weighted vs. Smoothed Weighted Esti
    xlab("Weighted Estimates") + ylab("Smoothed Weighted Estimates") +
    theme(plot.title = element_text(hjust = 0.5)) + xlim(0, 0.35) +
    ylim(0, 0.35)
merged3$std <- sqrt(FHmodel$smooth$var)

ggplot(merged3, aes(x = se, y = std)) + geom_point() + geom_abline(slope = 1,
    intercept = 0, color = "red") + ggtitle("Weighted vs. Smoothed Weighted Standard Errors") +
    xlab("Weighted SE") + ylab("Smoothed Weighted SE") + theme(plot.title = element_text(hjust = 0.5)) +
    xlim(0, 0.09) + ylim(0, 0.09)
library(gridExtra)
p1 <- mapPlot(data = naive_df, geo = KingCounty, variables = c("HT.est"),
    labels = c("Naive Direct Estimates"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Estimate Value")
p2 <- mapPlot(data = direct, geo = KingCounty, variables = c("smoker1"),
    labels = c("Weighted Estimates"), by.data = "hracode", by.geo = "HRA2010v2_",
    legend.label = "Weighted Estimates")
p3 <- mapPlot(data = smoothed_bym2$smooth, geo = KingCounty,
    variables = c("median"), labels = c("Smoothed Posterior Medians"),
    by.data = "region", by.geo = "HRA2010v2_", legend.label = "Posterior Medians")
p4 <- mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = c("median"),
    labels = c("Smoothed Weighted Posterior Medians"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Posterior Medians")
grid.arrange(grobs = list(p1, p2, p3, p4), ncol = 2)
library(gridExtra)
e1 <- mapPlot(data = naive_df, geo = KingCounty, variables = c("naive_se"),
    labels = c("Naive Standard Errors"), by.data = "region",
    by.geo = "HRA2010v2_", legend.label = "Error Value")
e2 <- mapPlot(data = direct, geo = KingCounty, variables = c("se"),
    labels = c("Weighted Standard Errors"), by.data = "hracode",
    by.geo = "HRA2010v2_", legend.label = "Error Value")
e3 <- mapPlot(data = smoothed_bym2$smooth, geo = KingCounty,
    variables = c("std"), labels = c("Smoothed Posterior Standard Deviation"),
```

```
    by.data = "region", by.geo = "HRA2010v2_", legend.label = "Error Value")
e4 <- mapPlot(data = FHmodel$smooth, geo = KingCounty, variables = c("std"),
    labels = c("Smoothed Weighted Posterior Standard Deviations"),
    by.data = "region", by.geo = "HRA2010v2_", legend.label = "Error Value")
grid.arrange(grobs = list(e1, e2, e3, e4), ncol = 2)
```