Nutrients variation in milk powder recipe and its modified recipe

Kelly Chin

Table of Contents

# Introduction

Dairy raw materials (RMs) are good source of proteins and nutrients such as vitamins and minerals. The amount of these nutrients in dairy RMs differs among seasons and regions, because of different farm management and nutritional values in the diet consumed by cows. Nutritional values fluctuate in dairy RMs result in nature variation. These natural variations of nutrients influence recipe calculation and quality and food safety. The more variability it is in RMs, the more difficult it is to have stable amount of nutrients in our products.

According to the current recipe calculation mehod, the variation of nutrients is based on the 6-sigma approach, allowing 99% of the values of each nutrient in recipe design for nutrition calculation. This is a very safe and conservative way to estimate the nutrient levels. Highly processed dairy RMs and additional single dose vitamin and minerals were therefore, added in the recipe to reduce the variation and fortify the nutrient levels.

However, the nutrients in RMs might be more stable and consistent in certain pattern. Nutrients which are contributed largely by dairy RMs would have seasonal changes and are more various in certain regions due to the different farming management. Therefore, the more we understand the nutrient levels in the RMs, the more efficient we can make good use of the natural variation and reduce the unnecessary overestimated variation in our recipe design.

The objective of this internship is to understand the variation of nutrients. Based on the comprehension of the natural variation of nutrients in RMs, an advanced way of recipe design principle will be developed. Also, a modified recipe is developed to further evaluate if we can use more natural RMs in the recipe design and to understand if nutrient levels will be in compliance with the original recipe.

# Materials and Methods

## Data

**BestMix** provided the weight of RMs and nutritional values.

**Raw Material Monitoring Program (RMMP)** provided nutirent levels of dairy RMs.

**Factory measurements**to understand the nutrient levels in the blended milk powder. Data between April 2019 to September 2019 are used in the analysis.

**Quality and Food Safety (QFS) Standard** for Specialized Nutrition (SN) has set nutritional standards (ownership of R&I Developmental Physiology and Nutrition Platform) for nutritional composition of all products to assure and promote consumers’ health. The ranges of nutrients are regulated as +/- percentages around the target value on the label and are used to judge whether the actual nutritional composition of a product is in line with the labelled values at any point during the shelf life.

## R version

Data editing and analysing was performed with RStudop (Version 4.0.2, URL: <https://cran.rstudio.com>)

## Computational Statistics

**Beta PERT distribution** is used to assume the distribution of stable RMs, which we merely have minimum, most likely (target), and maximum values of nutrient levels. The mean of the distribution is therefore defined as the weighted average of the minimum, most likely and maximum values that the variable may take, with four times the weight applied to the most likely value. The three parameters defining the distribution are intuitive the estimator.

**Non-parameteric - Bootstrap** is a compuational resampling technique of inference about a population of interest using sample data, with the only input being the procedure for calculating the estimate (or estimator) of interest on a sample of data. It works by treating the observed sample as if it were the population, and we can repeatedly take R number of samples (of the same size as the original sample), *with replacement*.

The BS process in as follows:

* For each dairy RM that has RMMP data, sample one value, with replacement.
* For each RM that has only min, target, and max values, make a beta PERT distribution and sample from the beta distribution with replacement.
* Sum these sampling values
* Repeat 100000 times
* Calculate the mean and 2.5% and 97.5% percentiles

# Results

### Sodium

Sodium in original recipe is contributed by 14 RMs (6 dairy RMs + 8 stable RMs):

* SMC
* DWC
* WP3A
* WP3S
* WP8A
* WP8S
* *SL*
* *VITM*
* *L-C*
* *PC*
* *SH*
* *TPCE*
* *TSC*
* *GS*

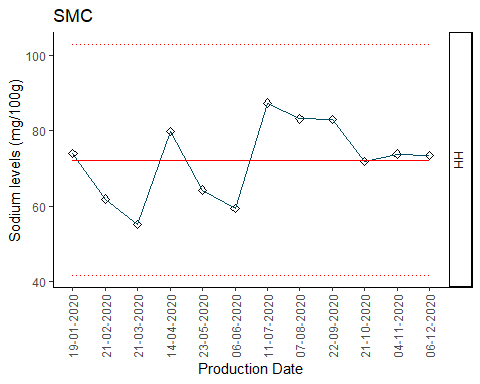
Sodium in modified recipe is contributed by 12 RMs (3 dairy RMs + 9 stable RMs):

* SMC
* DWC
* WP3A
* *CCD*
* *LELV2*
* *L-C*
* *PC*
* *SH*
* *SL*
* *TPCE*
* *TSC*
* *GS*

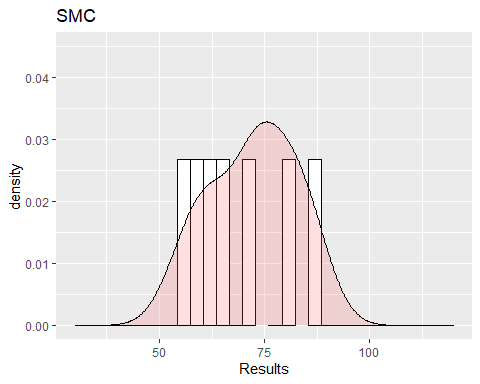
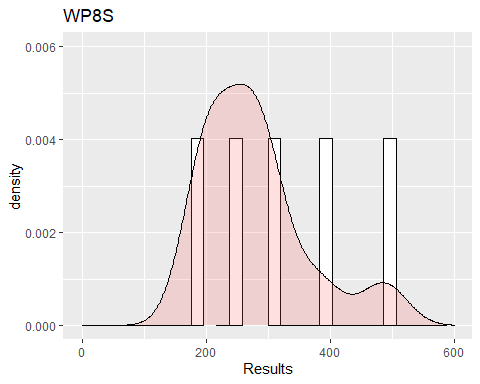
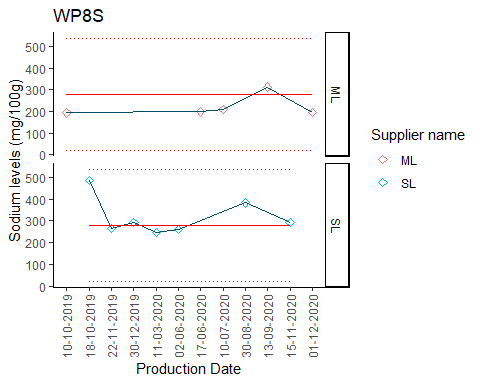
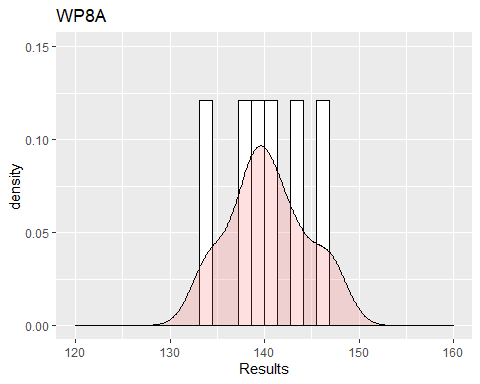
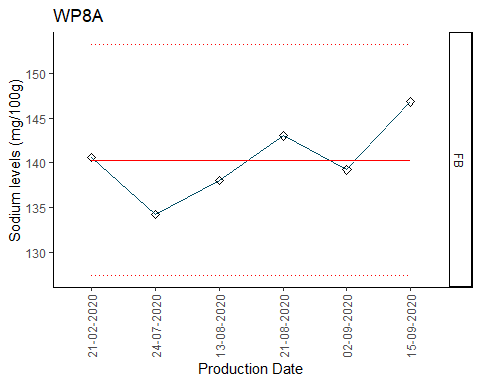
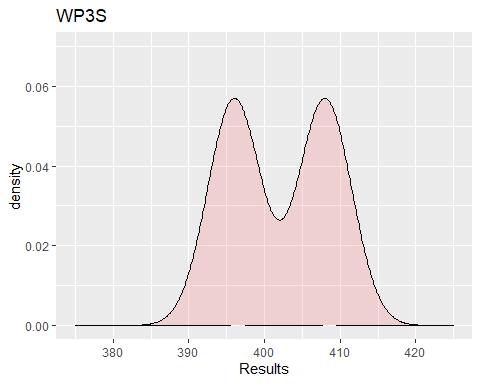
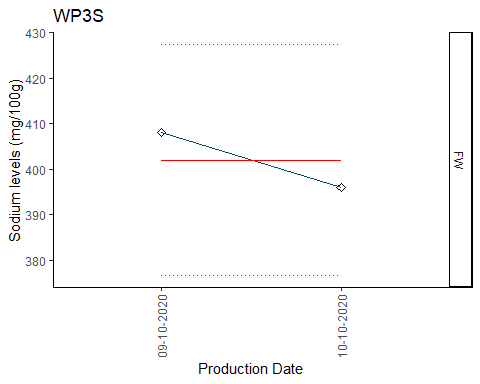
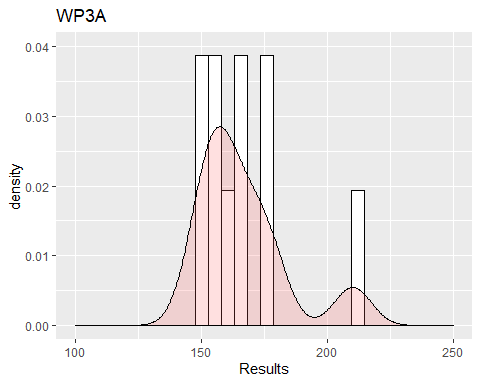
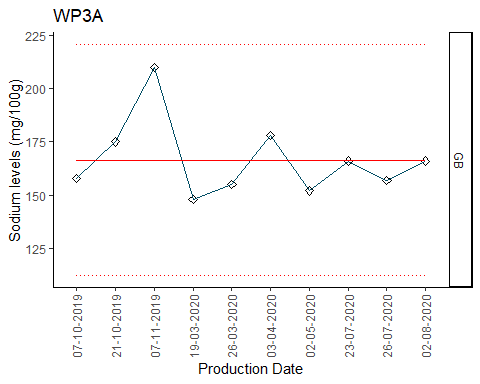
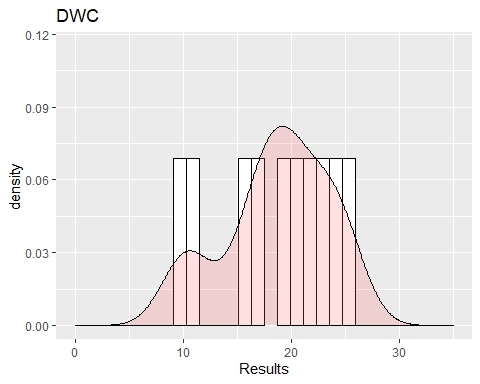
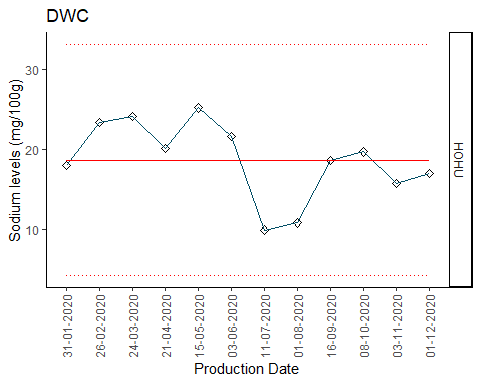
library(readxl)  
library(dplyr)  
sodium <- read\_excel("../data/raw/Examples.xlsx", sheet = "AllNutrients",   
 col\_types = c("text", "text", "numeric",   
 "text", "date", "text", "numeric",   
 "numeric", "text", "text", "numeric",   
 "text")) %>%  
 filter(Parameter == "Na")  
sodium

## # A tibble: 54 x 12  
## Product Parameter Results Unit Production\_date `Supplier name` `TS Code`  
## <chr> <chr> <dbl> <chr> <dttm> <chr> <dbl>  
## 1 SMC Na 87.3 mg/1~ 2020-07-11 00:00:00 HH 123456789  
## 2 SMC Na 83.1 mg/1~ 2020-08-07 00:00:00 HH 123456789  
## 3 SMC Na 82.9 mg/1~ 2020-09-22 00:00:00 HH 123456789  
## 4 SMC Na 71.8 mg/1~ 2020-10-21 00:00:00 HH 123456789  
## 5 SMC Na 73.8 mg/1~ 2020-11-04 00:00:00 HH 123456789  
## 6 SMC Na 73.4 mg/1~ 2020-12-06 00:00:00 HH 123456789  
## 7 SMC Na 79.8 mg/1~ 2020-04-14 00:00:00 HH 123456789  
## 8 SMC Na 61.9 mg/1~ 2020-02-21 00:00:00 HH 123456789  
## 9 SMC Na 73.9 mg/1~ 2020-01-19 00:00:00 HH 123456789  
## 10 SMC Na 55.1 mg/1~ 2020-03-21 00:00:00 HH 123456789  
## # ... with 44 more rows, and 5 more variables: `IS Code` <dbl>, Batch <chr>,  
## # `LOQ/2` <chr>, `LOQ/2 sum` <dbl>, `Result as MxNS` <chr>

library(ggplot2)  
library(dplyr)  
###########  
# SMC #  
###########  
SMC <- sodium %>%  
 filter(Product == "SMC")  
SMC$Results <- as.numeric(SMC$Results)  
SMC<- SMC[order(as.Date(SMC$Production\_date, format = "%d/%m/%Y")),]  
SMC$Production\_date <- as.character(SMC$Production\_date, format = "%d-%m-%Y")  
  
p <- function(A){  
 ggplot(A, aes(x=Production\_date, y=Results, group=Product))+   
 geom\_point(size=2, shape=23) + geom\_line(color="#014d64") +  
 geom\_line(aes(y = mean(Results, na.rm = TRUE)), color = "red", size=0.7,   
 linetype = "solid") +  
 geom\_line(aes(y = mean(Results, na.rm = TRUE) - 3\*sd(Results, na.rm = TRUE)), color = "red", size=0.7,   
 linetype = "dotted") +  
 geom\_line(aes(y = mean(Results, na.rm = TRUE) + 3\*sd(Results, na.rm = TRUE)), color = "red", size=0.7,   
 linetype = "dotted") +  
 #geom\_line(aes(y = mean(Results)+2\*sd(Results)), color = "blue", size=0.7, linetype = "solid") +  
 facet\_grid(`Supplier name` ~.)+   
 scale\_x\_discrete(limits=A$Production\_date)+  
 theme\_classic() + xlab("Production Date") + ylab("Sodium levels (mg/100g)") +  
 theme(panel.border = element\_blank(), axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1), panel.grid.major = element\_blank(), panel.grid.minor = element\_blank())  
}  
p(SMC) + labs(title = "SMC")



df <- data.frame(SMC)  
ggplot(df, aes(x=Results)) +   
 geom\_histogram(aes(y=..density..), colour="black", fill="white", bins = 30)+  
 geom\_density(alpha=.2, fill="#FF6666") + xlim(30, 120) + ylim(0, 0.045) +  
 labs(title = "SMC")

###############  
## Bootstrap ##  
###############  
library(mc2d)

## Loading required package: mvtnorm

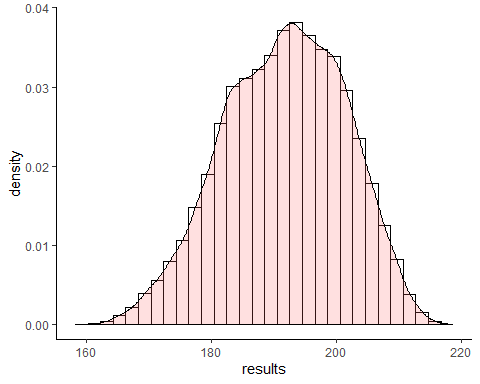
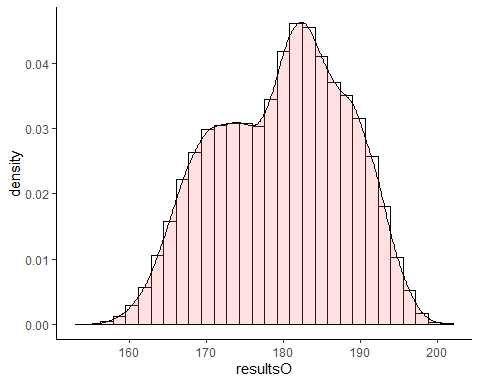
##   
## Attaching package: 'mc2d'

## The following objects are masked from 'package:base':  
##   
## pmax, pmin

library(ggplot2)  
set.seed(123)  
resultsO <- replicate(100000, {  
 smc <- sample(SMC$Results, 1)   
 SMC\_dist <- (79.359\*smc)/100000  
 DWC <- sample(DWC$Results, 1)  
 DWC\_dist <- (54.605\*DWC)/100000  
 wpc35a <- sample(WPC35A$Results, 1)  
 wpc35a\_dist <- (2.857\*wpc35a)/100000  
 wpc35s <- sample(WPC35$Results, 1)  
 wpc35s\_dist <- (2.856\*wpc35s)/100000  
 wpc80l <- sample(WPC80L$Results, 1)  
 wpc80l\_dist <- (6.236\*wpc80l)/100000  
 wpc80s <- sample(WPC80$Results, 1)  
 wpc80s\_dist <- (1.282\*wpc80s)/100000  
  
 soy\_beta <- rpert(n = 1, min = 40\*0.9, max = 40\*1.1, mode = 40, shape = 4)  
 soy\_dist <- (0.049673\*soy\_beta)/100000  
 vit\_beta <- rpert(n = 1, min = 1\*0.9, max = 1\*1.1, mode = 1, shape = 4)  
 vit\_dist <- (0.120433\*vit\_beta)/100000  
 gos\_beta <- rpert(n = 1, min =2.25\*0.9, max=11.25, mode= 2.25, shape =4)  
 gos\_dist <- (14.372\*gos\_beta)/100000  
 sod\_beta <- rpert(n = 1, min = 56000, max = 58600, mode=57500, shape = 4)  
 sod\_dist <- (0.045555\*sod\_beta)/100000  
 tri\_p\_beta <- rpert(n = 1, min = 0, max = 300\*1.1, mode = 0, shape = 4)  
 tri\_p\_dist <- (0.1\*tri\_p\_beta)/100000  
 tri\_beta <- rpert(n = 1, min = 21000, max = 25000, mode=23500, shape = 4)  
 tri\_dist <- (0.242591\*tri\_beta)/100000   
 l\_car <- rpert(n = 1, min = 0, max = 100\*1.1, mode = 0, shape = 4)  
 l\_car\_dist <- (0.014\*l\_car)/100000  
 kcl\_beta <- rpert(n = 1, min = 0, max = 500\*1.1, mode = 0, shape = 4)  
 kcl\_dist <- (0.328983\*kcl\_beta)/100000  
   
 targetO <- (SMC\_dist + DWC\_dist + wpc35a\_dist + wpc35s\_dist +  
 wpc80l\_dist + wpc80s\_dist +   
 soy\_dist + vit\_dist + gos\_dist + sod\_dist +  
 tri\_p\_dist + tri\_dist + l\_car\_dist + kcl\_dist)\*1000  
})  
  
resultsO <- as.data.frame(resultsO)  
  
  
################  
# modified #  
################  
library(mc2d)  
set.seed(123)  
results <- replicate(100000, {  
 smc <- sample(SMC$Results, 1)   
 SMC\_dist <- (79.398\*smc)/100000  
   
 DWC <- sample(DWC$Results, 1)  
 DWC\_dist <- (112.371\*DWC)/100000  
   
 wpc35a <- sample(WPC35A$Results, 1)  
 wpc35a\_dist <- (5.733\*wpc35a)/100000  
   
 cacl2.2h2o <- rpert(n = 1, min = 0, max = 20\*1.1, mode = 0, shape = 4)  
 cacl2.2h20\_dist <- (0.115333\*cacl2.2h2o)/100000  
   
 soy\_beta <- rpert(n = 1, min = 40\*0.9, max = 40\*1.1, mode = 40, shape = 4)  
 soy\_dist <- (0.049673\*soy\_beta)/100000  
   
 lact\_beta <- rpert(n = 1, min = 0, max = 46, mode = 14.5, shape = 4)  
 lact\_dist <- (7.802\*lact\_beta)/100000  
   
 l\_car <- rpert(n = 1, min = 0, max = 100\*1.1, mode = 0, shape = 4)  
 l\_car\_dist <- (0.008204\*l\_car)/100000  
   
 kcl\_beta <- rpert(n = 1, min = 0, max = 500\*1.1, mode = 0, shape = 4)  
 kcl\_dist <- (0.340722\*kcl\_beta)/100000  
   
 gos\_beta <- rpert(n = 1, min =2.25\*0.9, max=11.25, mode= 2.25, shape =4)  
 gos\_dist <- (14.372\*gos\_beta)/100000  
   
 sod\_beta <- rpert(n = 1, min = 56000, max = 58600, mode=57500, shape = 4)  
 sod\_dist <- (0.045555\*sod\_beta)/100000  
   
 tri\_p\_beta <- rpert(n = 1, min = 0, max = 300\*1.1, mode = 0, shape = 4)  
 tri\_p\_dist <- (0.1\*tri\_p\_beta)/100000  
   
 tri\_beta <- rpert(n = 1, min = 21000, max = 25000, mode=23500, shape = 4)  
 tri\_dist <- (0.322713\*tri\_beta)/100000  
   
 target <- (SMC\_dist + DWC\_dist + wpc35a\_dist +   
 cacl2.2h20\_dist + soy\_dist + lact\_dist +   
 l\_car\_dist + kcl\_dist + gos\_dist +   
 sod\_dist + tri\_p\_dist + tri\_dist)\*1000  
})  
  
results <- as.data.frame(results)

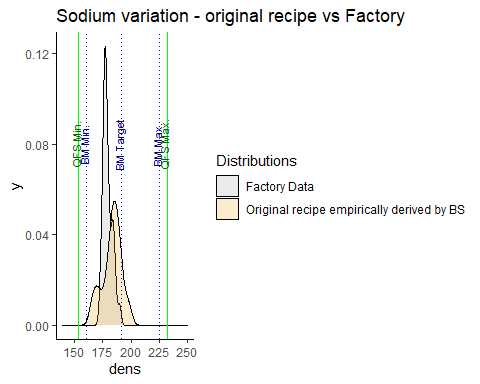
Bootstrap

# original recipe bs result #  
ggplot(resultsO, aes(x=resultsO)) +   
 geom\_histogram(aes(y=..density..), colour="black",   
 fill="white", bins=30)+  
 geom\_density(alpha=.2, fill="#FF6666") +  
 theme\_classic()   
# modified recipe bs result #  
ggplot(results, aes(x=results)) +   
 geom\_histogram(aes(y=..density..), colour="black", fill="white", bins=30)+  
 geom\_density(alpha=.2, fill="#FF6666") +  
 theme\_classic()



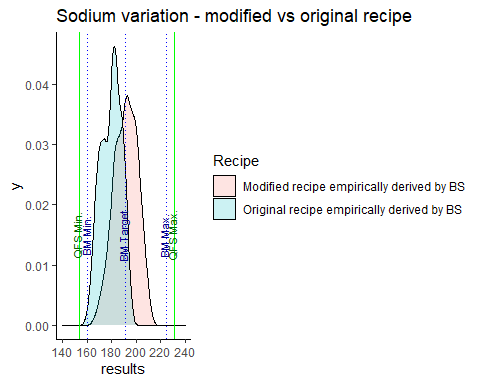
Sodium variation

# FACTORY'S RESULTS FROM ORIGINAL RECIPE  
Mittelwert <- c(176, 176, 175.5, 174, 172.9, 177.4, 177.2, 178.2, 180.1, 177.3, 177.3, 179, 174.8, 177.7, 176, 184, 177, 184, 184, 186, 182, 184, 178, 190, 180.6, 177.1, 186.2, 180.6, 177.7, 178.9, 180.3)   
  
# NEW BESTMIX SPEC VS ODL BESTMIX SPEC  
results$Recipe <- NA  
results$Recipe <- "Modified recipe empirically derived by BS"  
resultsO$Recipe <- NA  
resultsO$Recipe <- "Original recipe empirically derived by BS"  
names(resultsO) <- names(results)  
dat <- rbind(results, resultsO)  
  
resBS\_Original <- sample(resultsO[,1], 31)  
  
library(ggplot2)  
set.seed(123)  
#Original recipe  
datA <- data.frame(dens= c(Mittelwert, resBS\_Original), Distributions = c(rep("Factory Data", 31), rep("Original recipe empirically derived by BS", 31)))  
  
ggplot(datA, aes(x = dens, fill = Distributions)) +   
 geom\_density(alpha = 0.2) + xlim(140, 250) +   
 scale\_fill\_manual(values = c("#999999", "#E69F00")) +  
 # bestmix min, target, and max  
 geom\_vline(xintercept = c(160.571, 191.362, 224.985), colour = "blue" ,  
 size=0.7, linetype = "dotted") +  
 annotate("text", x=c(158.9, 189.4, 223), y=0.08, angle = 90,   
 label = c("BM Min.", "BM Target", "BM Max."),  
 colour = "dark blue", size = 3) +  
 # QFS range for MILKPOWDER  
 geom\_vline(xintercept = c(154.14, 231.71), colour = "green" ,   
 size=0.7, linetype = "solid") +  
 annotate("text", x=c(152, 229), y=0.08, angle = 90,   
 label = c("QFS Min.", "QFS Max."), size = 3, colour = "dark green") +  
 ggtitle("Sodium variation - original recipe vs Factory") + theme\_classic()



Simulated total sodium in original recipe shows a wider variation than the factory’s measurements of total sodium in milkpowder Both factory’s measurements and bootstrap’s simulation of total sodium fall into the BestMix spec. and QFS levels.

# MODIFIED RECIPE VS ORIGINAL RECIPE  
ggplot(dat, aes(x = results, fill = Recipe)) +  
 geom\_density(alpha = 0.2) +   
 # bestmix min, target, and max  
 geom\_vline(xintercept = c(160.571, 191.362, 224.985), colour = "blue" ,  
 size=0.7, linetype = "dotted") +  
 annotate("text", x=c(158.9, 189.4, 223), y=0.015, angle = 90,   
 label = c("BM Min.", "BM Target", "BM Max."),  
 colour = "dark blue", size = 3) +  
 # QFS range for MILKPOWDER  
 geom\_vline(xintercept = c(154.14, 231.71), colour = "green" ,   
 size=0.7, linetype = "solid") +  
 annotate("text", x=c(152, 229), y=0.015, angle = 90,   
 label = c("QFS Min.", "QFS Max."), size = 3, colour = "dark green") +  
 ggtitle("Sodium variation - modified vs original recipe") +  
 theme\_classic() +  
 xlim(140, 240)



Sodium is a relative stable nutrient. The estimated sodium level by bootstrap is approximately overlapping the sodium level from factory’s measurements. Moreover, the sodium in modified recipe overlaps the sodium simulated in original recipe and both fall into the BestMix spec. and QFS tolerance.

# Overall findings and room for improvements

* Bootstrap works well in the estimation of nutrient in original recipe and the simulated variation is indeed smaller than the 6 sigma approach. Bootstrap can be used in future recipe calculation.
* The nutritional values in modified recipe are in compliance with the nutritional values in original recipe. We can indeed try to reduce the use of highly-processed dairy RMs.
* Small sample size and limited data points of the nutritional values of dairy RMs were available, which may lead to a baised estimation for bootstrap simulation. More annual measurements of dairy raw materials will help to develop a more precise estimation.
* Improvement of data integration from different sources is desired. Data from BestMix, RMMP, Factory, and QFS standard can be integrated. A more direct way of nutrient data integration can be developed.

# Appendix

## Contribution of RMs in MILKPOWDER original and modified recipe

Contributions of dairy RMs were calculated by median of RMMP measurements. Contributions of stable RMs were calculated by target values of BestMix spec.