Stage_M2_NB_2_CV_Admixture

Nicolas Bettembourg

2024-04-22

Contents

Chargement des packages R

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Analyse d'ADMIXTURE - erreurs de CV

Admixture non supervisée

SeqApiPop - 629 échantillons - MAF > 0.01

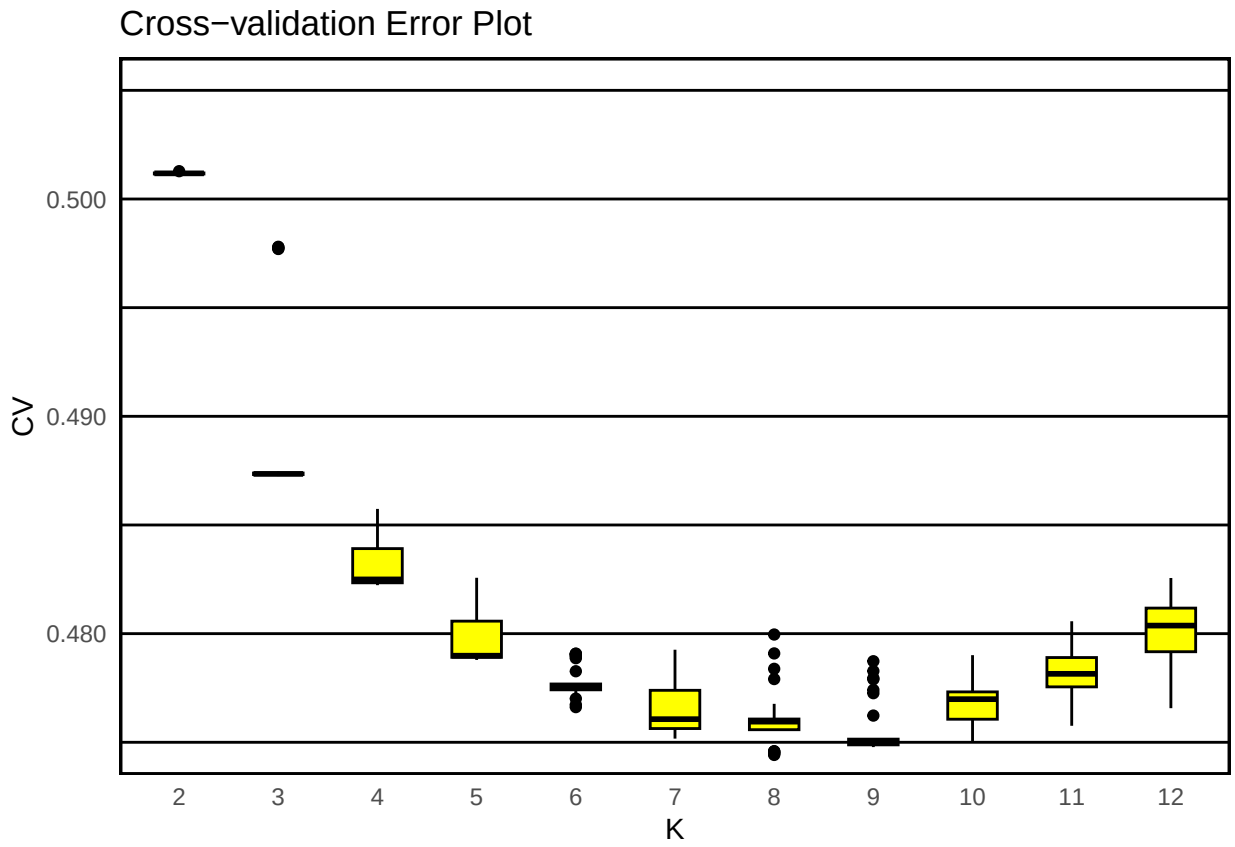
```
 #LD03 - CV Error plot
setwd("~/Documents/Stage_NB/data/maf001_LD03")

cv_error <- read.table("SeqApiPop_629_maf001_LD03_1.cv.error", header = F)

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_maf001_LD03_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

  # 26/1
merge_cv_error <- read.table("merge_cv_error", header = F)

point_min <- merge_cv_error[which.min(merge_cv_error[, 2]), ]

#box plot
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.475, 0.480, 0.485, 0.490, 0.495, 0.500, 0.505),
    color = "black",
    linetype = "solid",
    size = 0.5
```

```
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.475, 0.505))
```

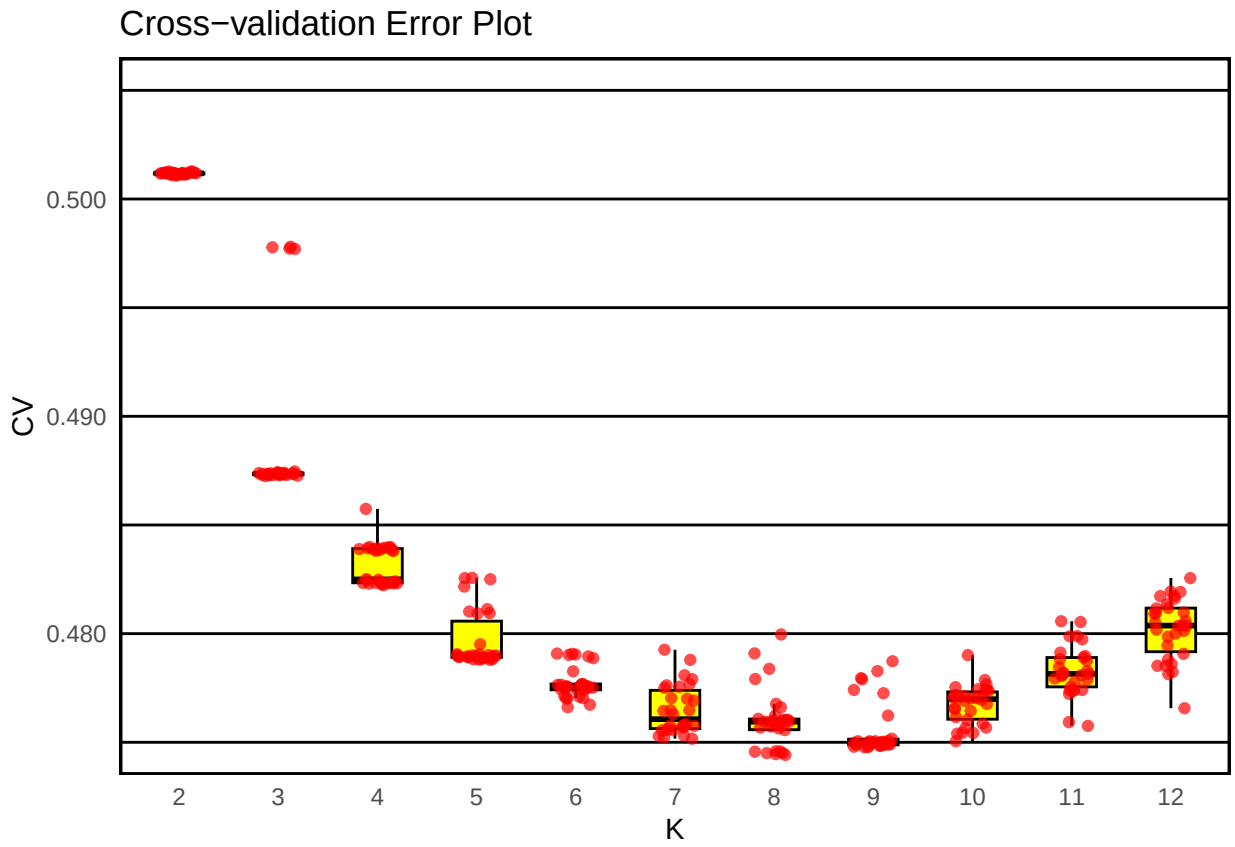
LD pruning = 0.3 (fenêtre de 1749 SNPs et pas de 175 bp)

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning: The ‘size‘ argument of ‘element_rect()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

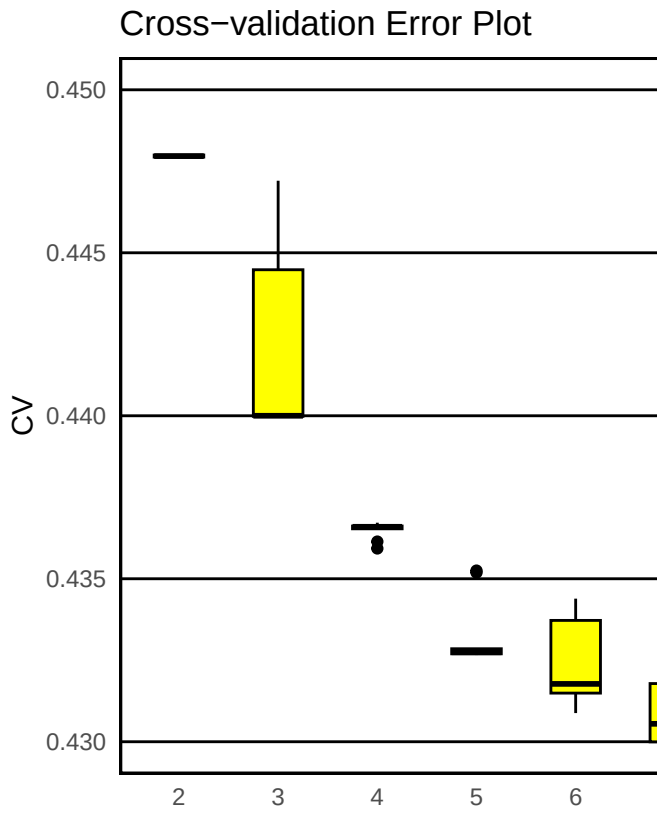



```
#jitter plot
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.475, 0.480, 0.485, 0.490, 0.495, 0.500, 0.505),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.475, 0.505))
```



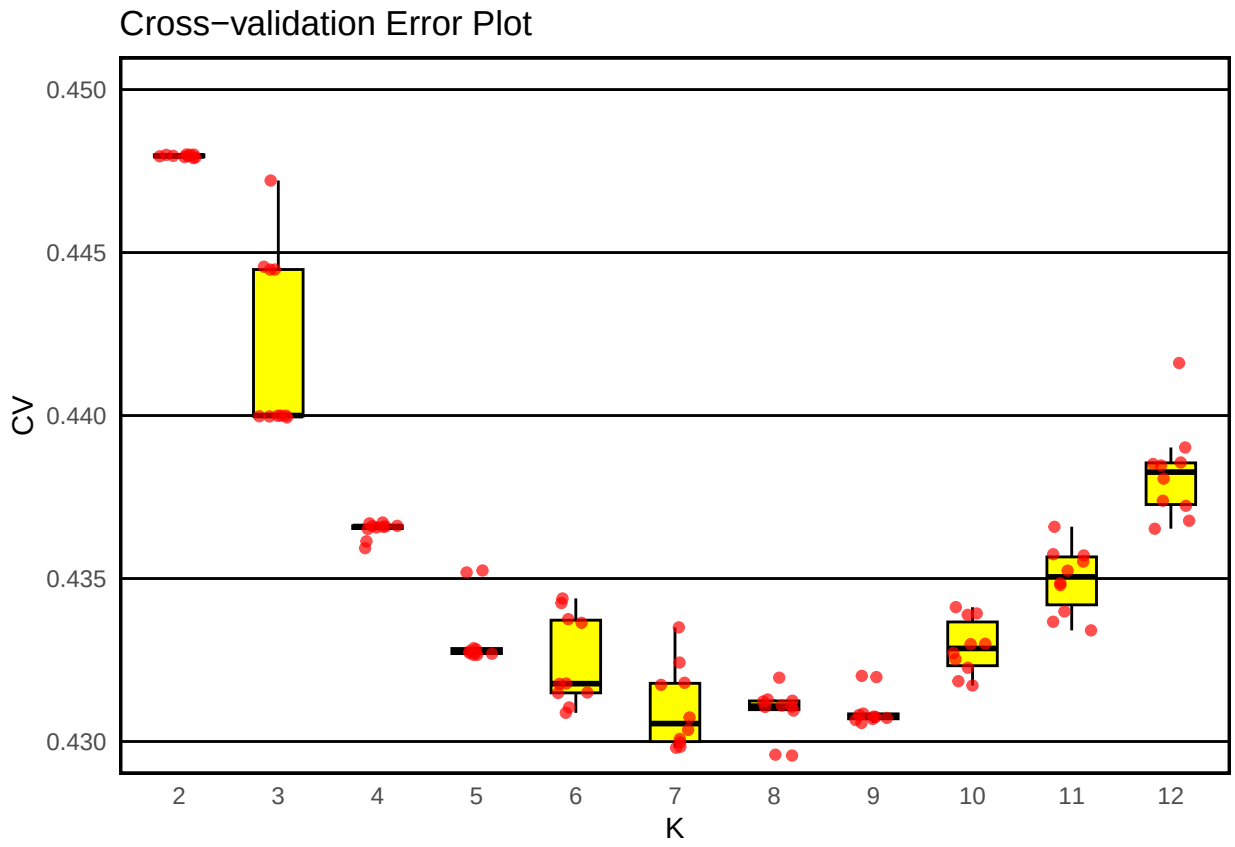

```r
#####LD02
setwd("~/Documents/Stage_NB/data/maf001_LD02")

cv_error <- read.table("SeqApiPop_629_maf001_LD02_1.cv.error", header = F)

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:10) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_maf001_LD02_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)
```

```r
merge_cv_error <- read.table("merge_cv_error", header = F)

point_min <- merge_cv_error[which.min(merge_cv_error[, 2]), ]

#box plot LD02
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.430, 0.435, 0.440, 0.445, 0.450),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.430, 0.450))
```



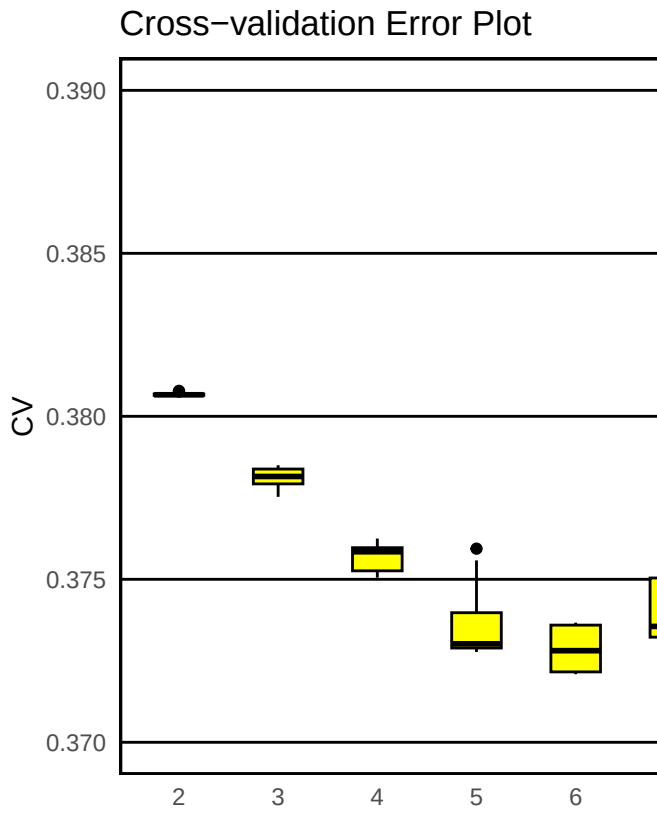
LD pruning = 0.2 (fenêtre de 1749 SNPs et pas de 175 bp)

```r
#jitter plot LD02
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.430, 0.435, 0.440, 0.445, 0.450),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.430, 0.450))
```



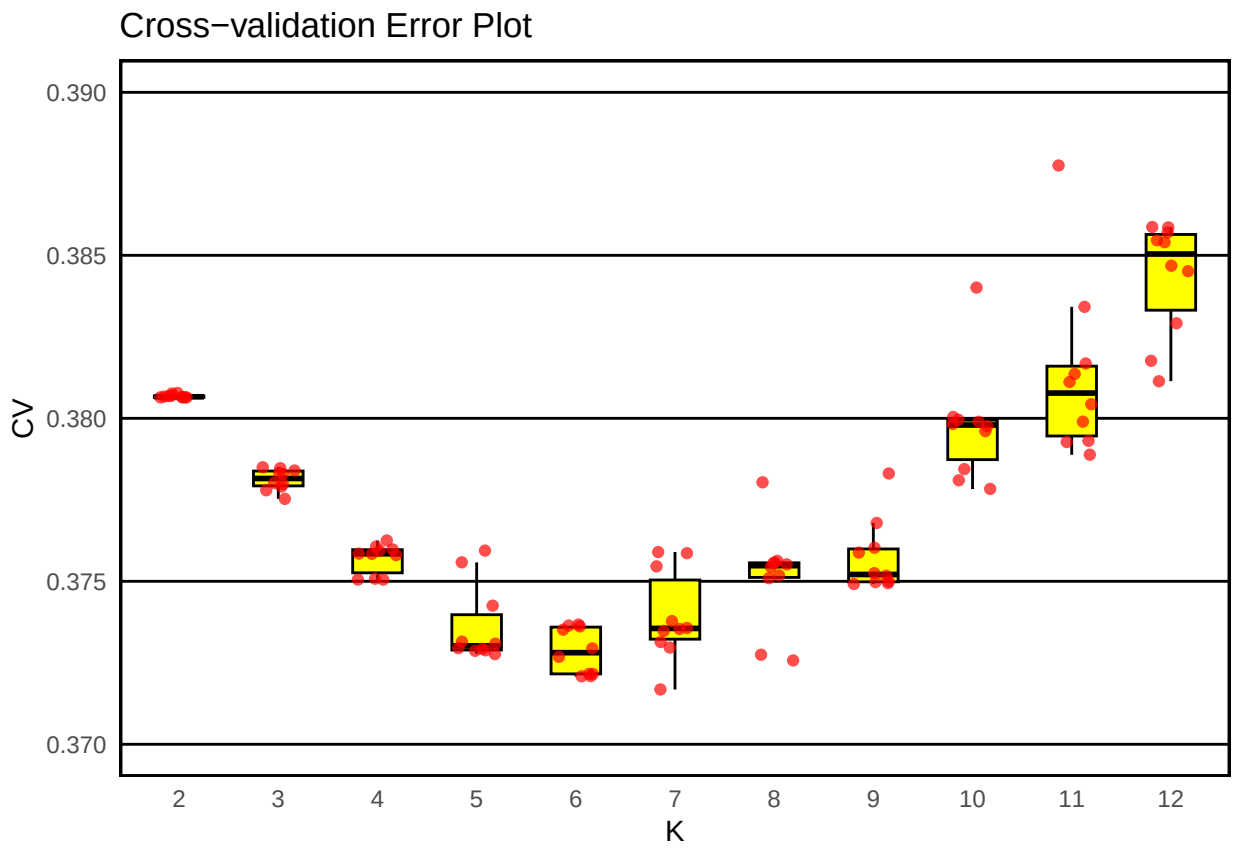

```r
#####LD01
```

```r
setwd("~/Documents/Stage_NB/data/maf001_LD01")

cv_error <- read.table("SeqApiPop_629_maf001_LD01_1.cv.error", header = F)

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:10) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_maf001_LD01_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)

point_min <- merge_cv_error[which.min(merge_cv_error[, 2]), ]

#box plot LD01
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.370, 0.375, 0.380, 0.385, 0.390),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.370, 0.390))
```



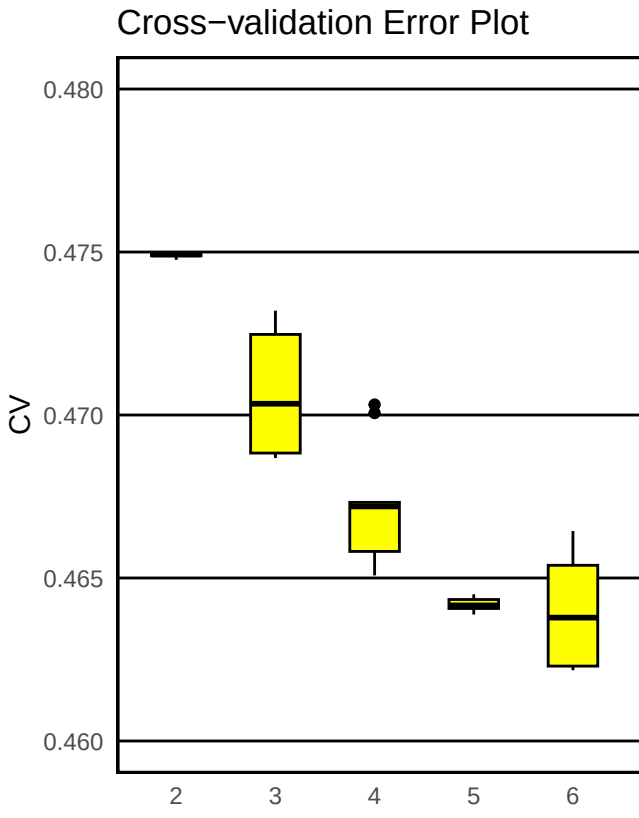
LD pruning = 0.1 (fenêtre de 1749 SNPs et pas de 175 bp)

```
#jitter plot LD01
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.370, 0.375, 0.380, 0.385, 0.390),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.370, 0.390))
```



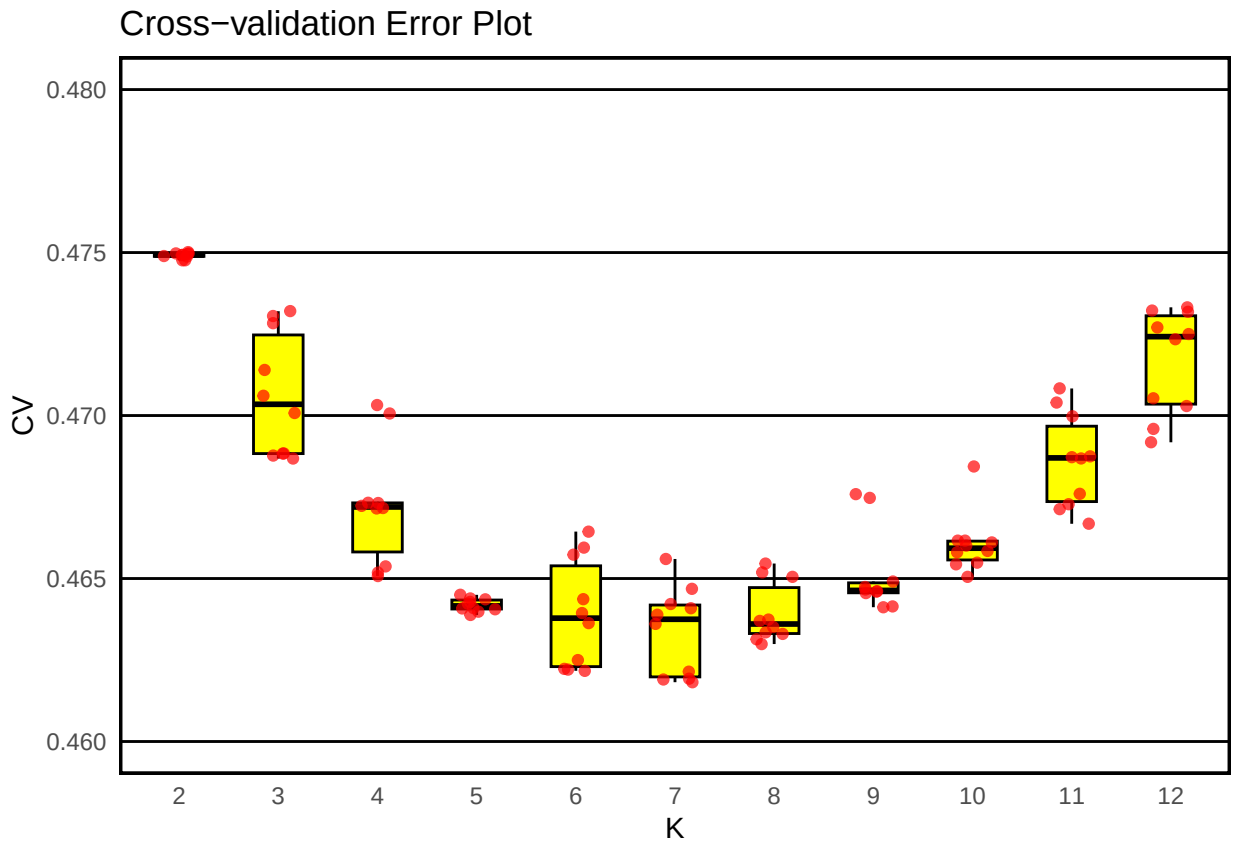

```r
#####LD005
setwd("~/Documents/Stage_NB/data/maf001_LD005")

cv_error <- read.table("SeqApiPop_629_maf001_LD005_1.cv.error", header = F)

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:10) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_maf001_LD005_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)
```

```
merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD01
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.460, 0.465, 0.470, 0.475, 0.480),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.460, 0.480))
```



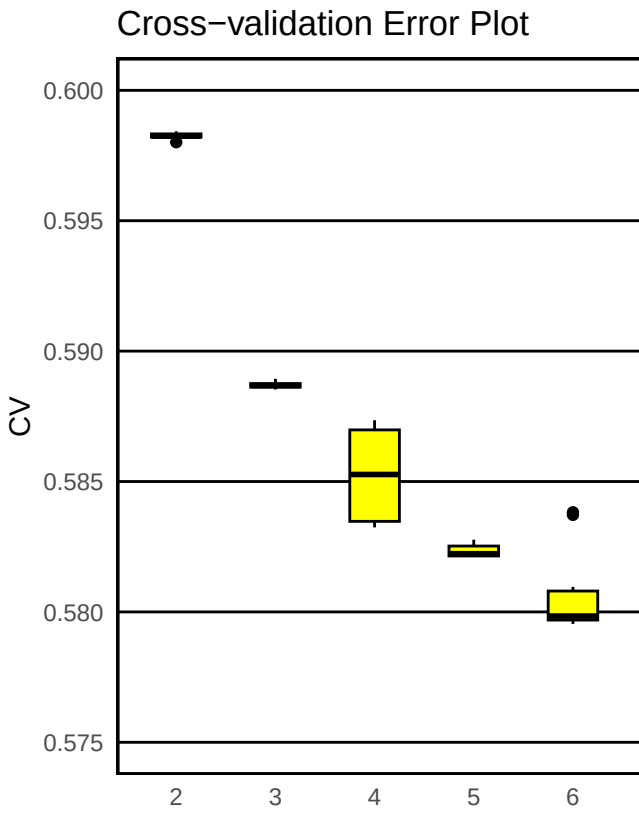
LD pruning = 0.05 (fenêtre de 1749 SNPs et pas de 175 bp)

```r
#jitter plot LD01
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.460, 0.465, 0.470, 0.475, 0.480),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.460, 0.480))
```



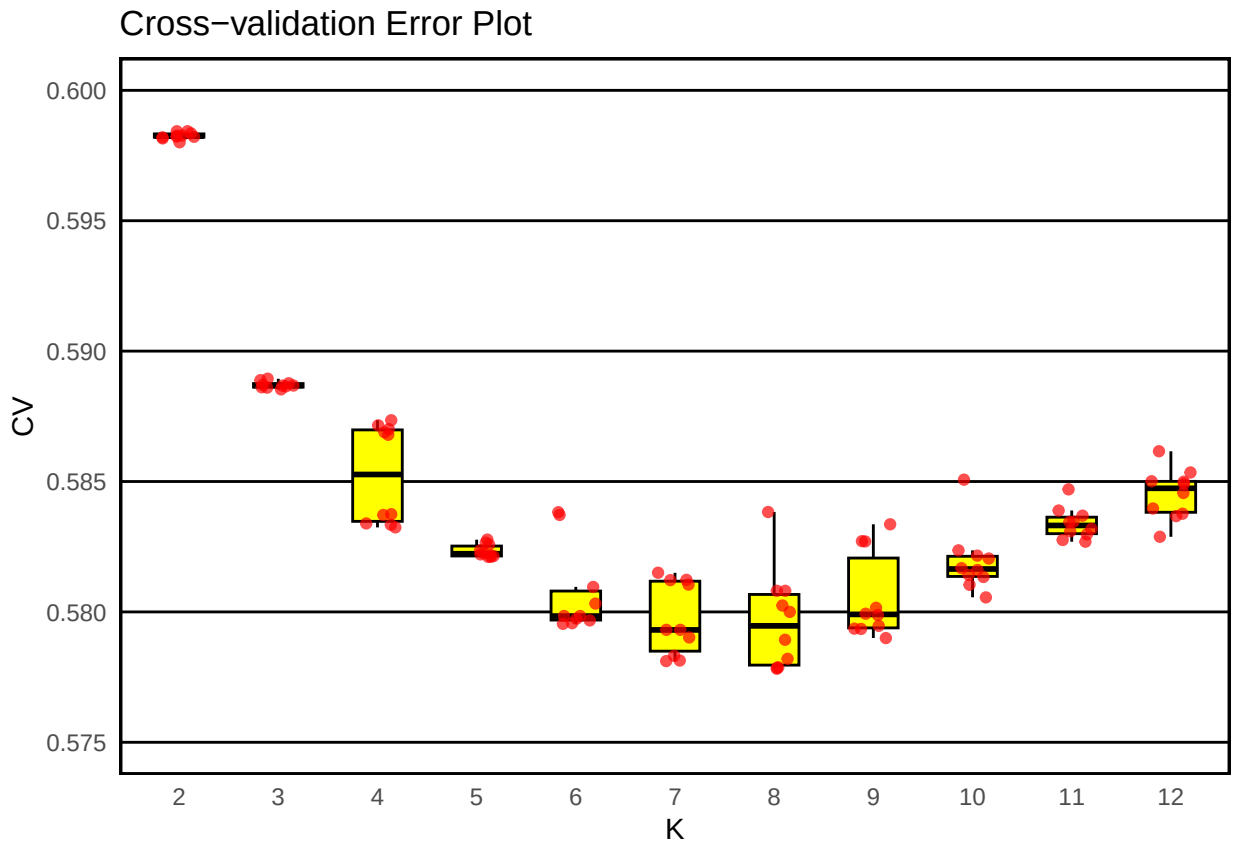

```r
#####LD004
```

```r
setwd("~/Documents/Stage_NB/data/maf001_LD004")

cv_error <- read.table("SeqApiPop_629_maf001_LD004_1.cv.error", header = F)

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:10) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_maf001_LD004_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD004
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.575, 0.58, 0.585, 0.59, 0.595, 0.6),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.575, 0.6))
```



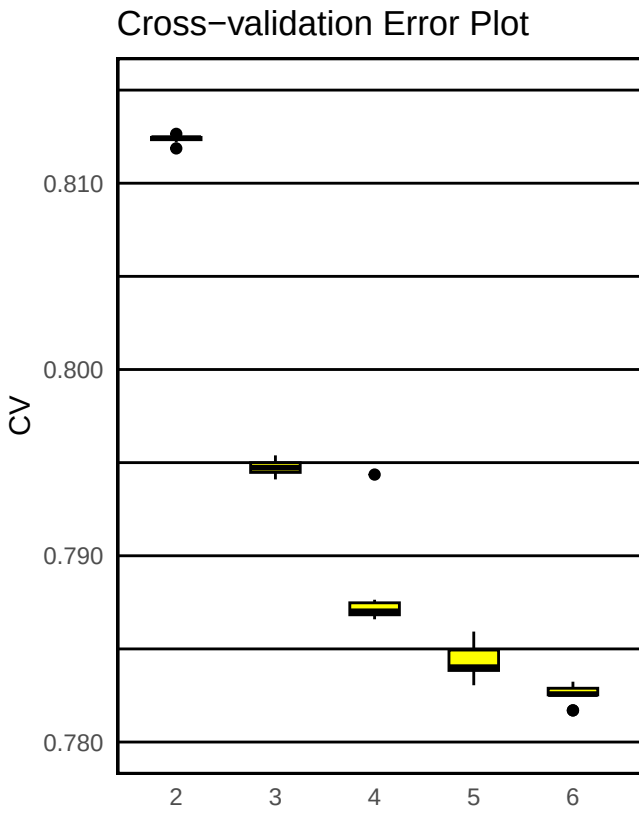
LD pruning = 0.04 (fenêtre de 1749 SNPs et pas de 175 bp)

```
#jitter plot LD004
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.575, 0.58, 0.585, 0.59, 0.595, 0.6),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.575, 0.6))
```




```
#####LD003
setwd("~/Documents/Stage_NB/data/maf001_LD003")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:10) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_maf001_LD003_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)
```

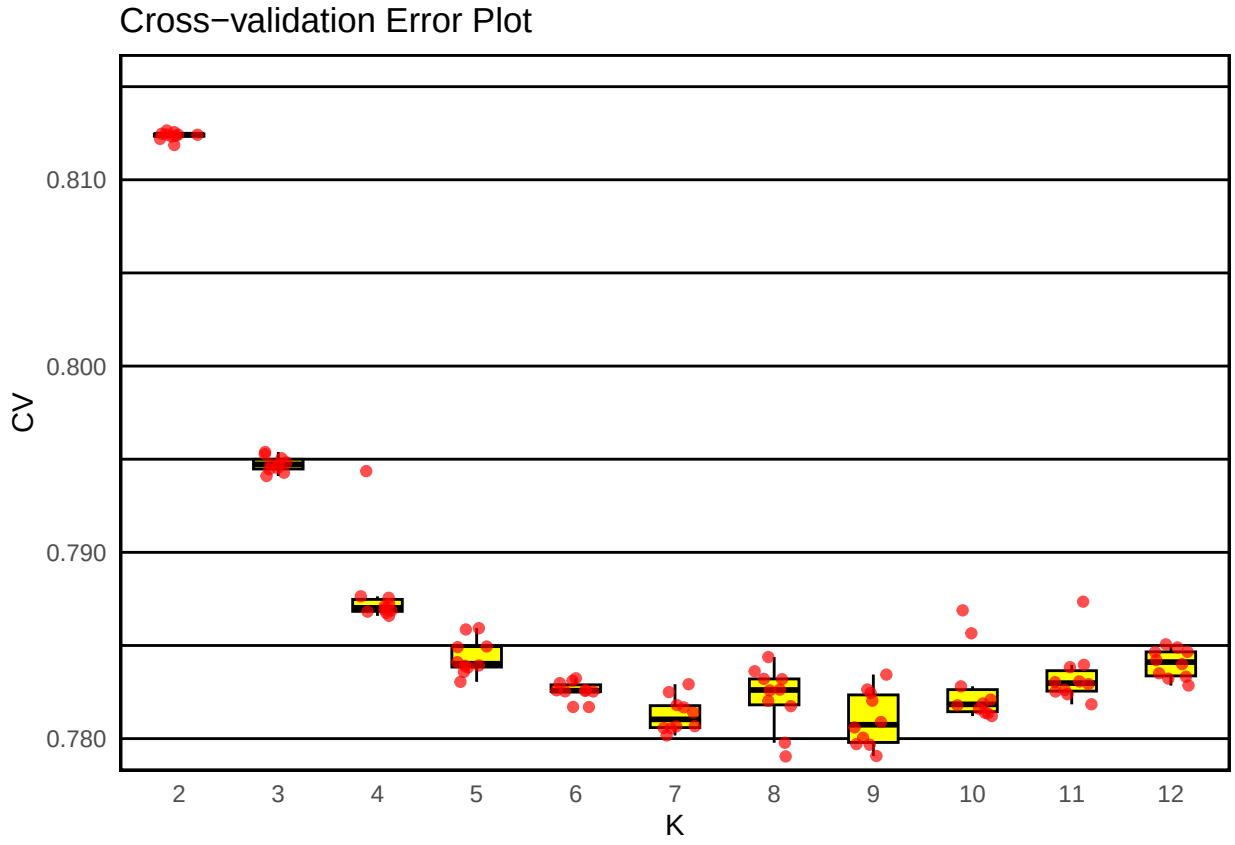
```r
#box plot LD01
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.78, 0.785, 0.79, 0.795, 0.8, 0.805, 0.81, 0.815),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.78, 0.815))
```



LD pruning = 0.03 (fenêtre de 1749 SNPs et pas de 175 bp)

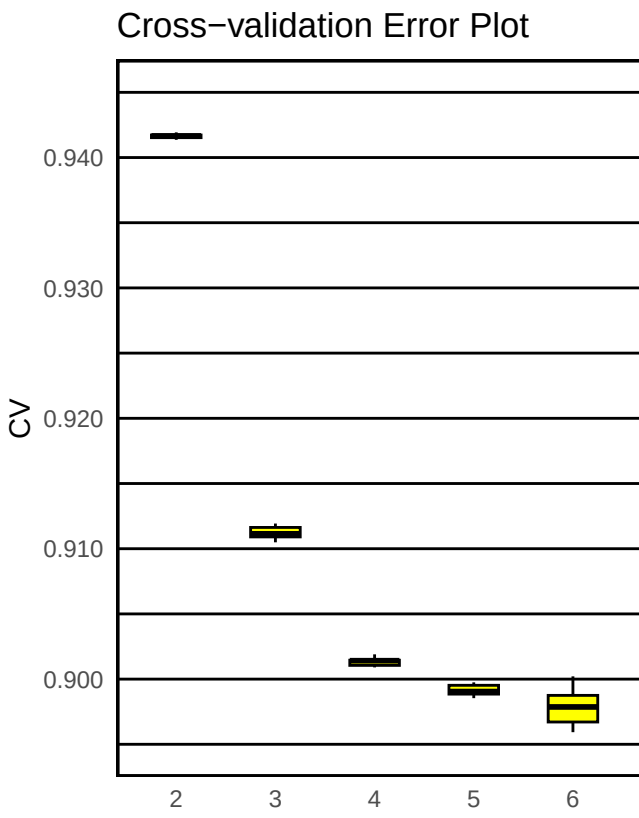
```r
#jitter plot LD01
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.78, 0.785, 0.79, 0.795, 0.8, 0.805, 0.81, 0.815),
```

```
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.78, 0.815))
```



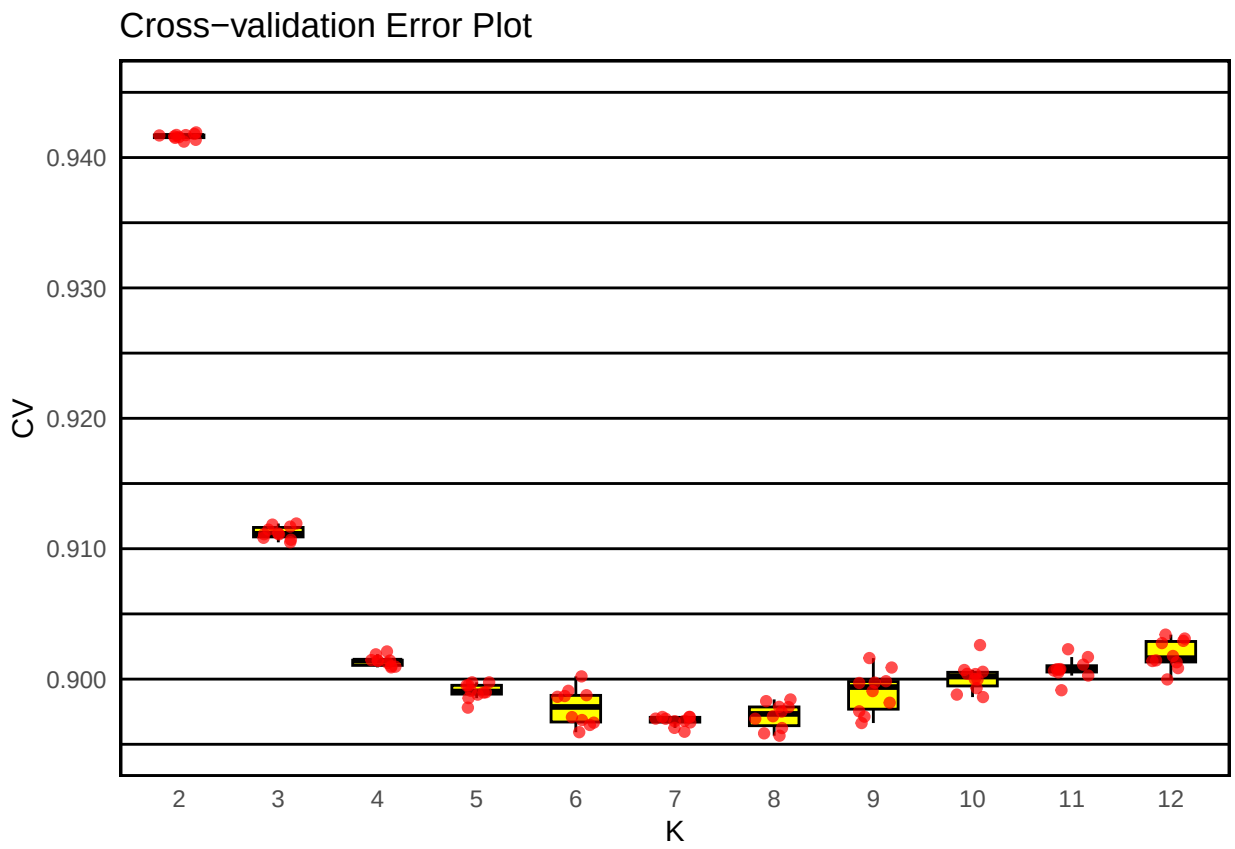

```
#####LD001
setwd("~/Documents/Stage_NB/data/maf001_LD001")

# Étape 1: Créer une liste vide pour stocker les données
```

```r
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:10) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_maf001_LD001_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD01
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.895, 0.9, 0.905, 0.91, 0.915, 0.92, 0.925, 0.93, 0.935, 0.94, 0.945),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.895, 0.945))
```



LD pruning = 0.01 (fenêtre de 1749 SNPs et pas de 175 bp)

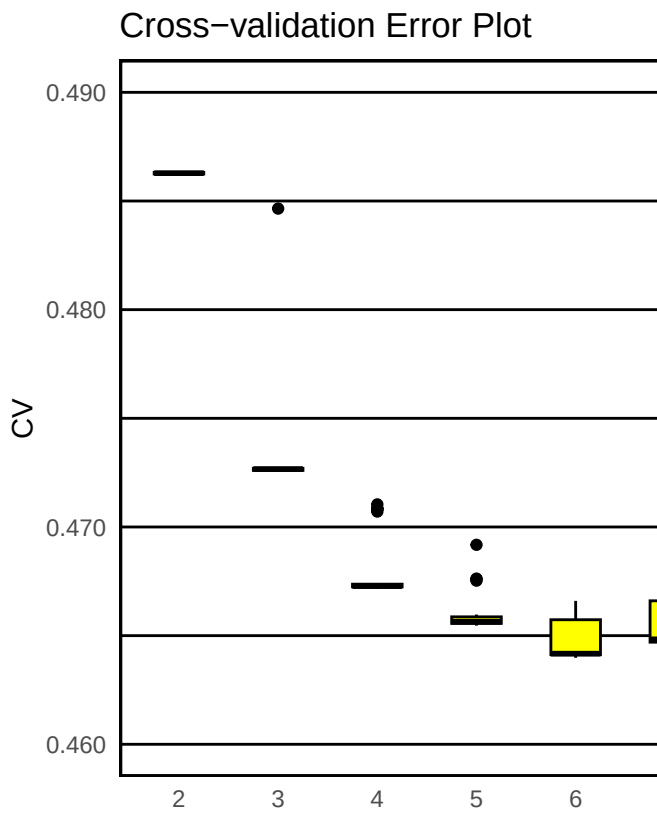
```
#jitter plot LD01
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.895, 0.9, 0.905, 0.91, 0.915, 0.92, 0.925, 0.93, 0.935, 0.94, 0.945),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.895, 0.945))
```



SeqApiPop - 561 échantillons - MAF > 0.01

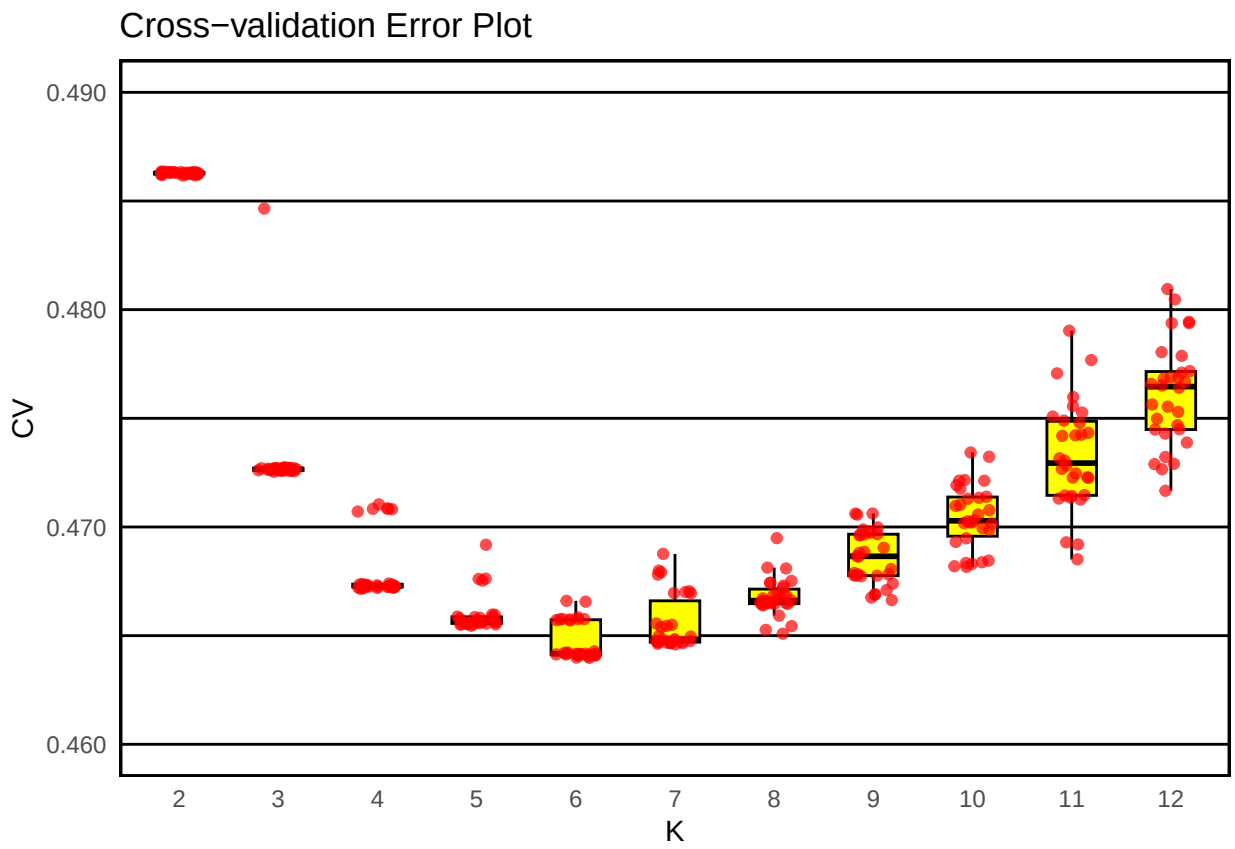
```
#LD03 - CV Error plot
setwd("~/Documents/Stage_NB/data/SeqApiPop_561_maf001_LD03")

cv_error <- read.table("SeqApiPop_561_maf001_LD03_1.cv.error", header = F)

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_561_maf001_LD03_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)
```

```r
# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

  # 26/1
merge_cv_error <- read.table("merge_cv_error", header = F)

point_min <- merge_cv_error[which.min(merge_cv_error[, 2]), ]

#box plot
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.46, 0.465, 0.47,0.475, 0.480, 0.485, 0.490),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.46, 0.49))
```



LD pruning = 0.3 (fenêtre de 1749 SNPs et pas de 175 bp)

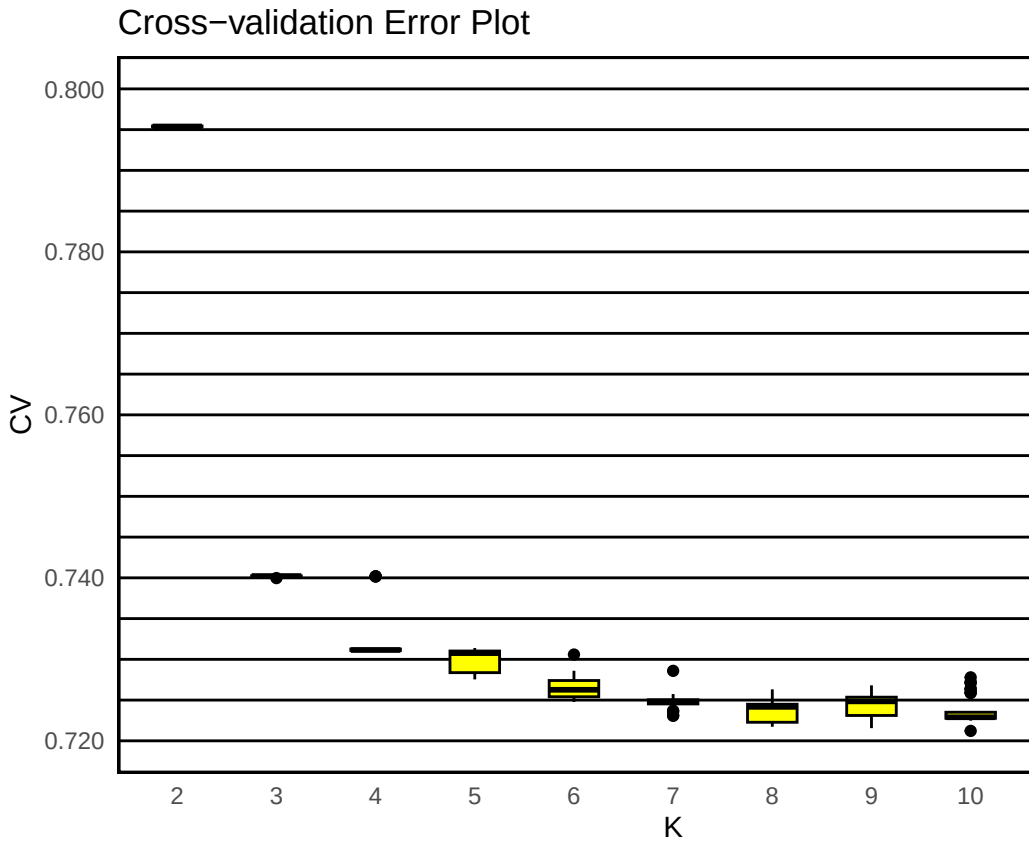
```
#jitter plot
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.46, 0.465, 0.47,0.475, 0.480, 0.485, 0.490),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.46, 0.49))
```



SeqApiPop - 629 échantillons - SNPsBeeMuSe filtered

```
setwd("~/Documents/Stage_NB/data/SeqApiPop_629_SNPsBeeMuSe")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_SNPsBeeMuSe_filtered_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)
```

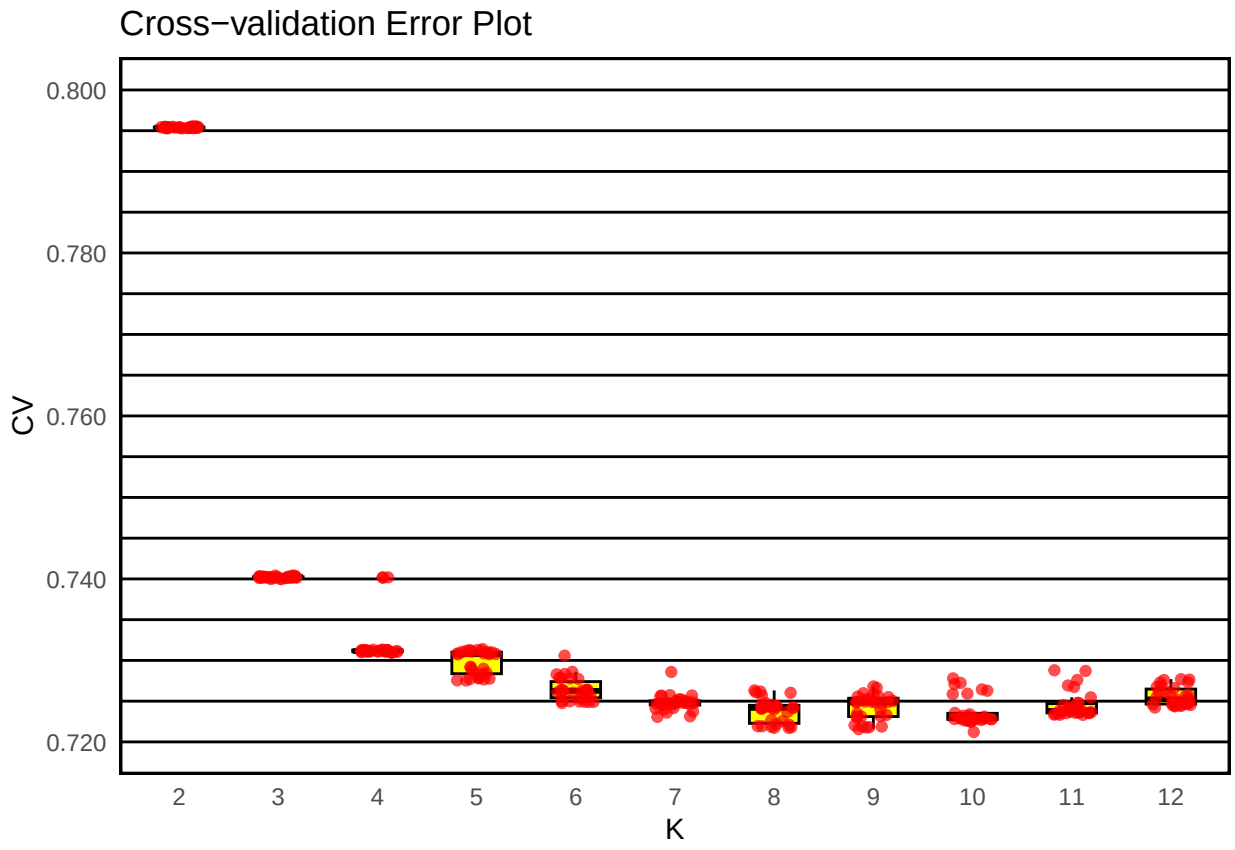
```
merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.72,0.725,0.73,0.735,0.74,0.745,0.75,0.755,0.76,0.765,0.77,0.775,0.78,0.785,0.79,0.
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.72, 0.8))
```



No LD pruning - 10030 SNPs

```
#jitter plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
```

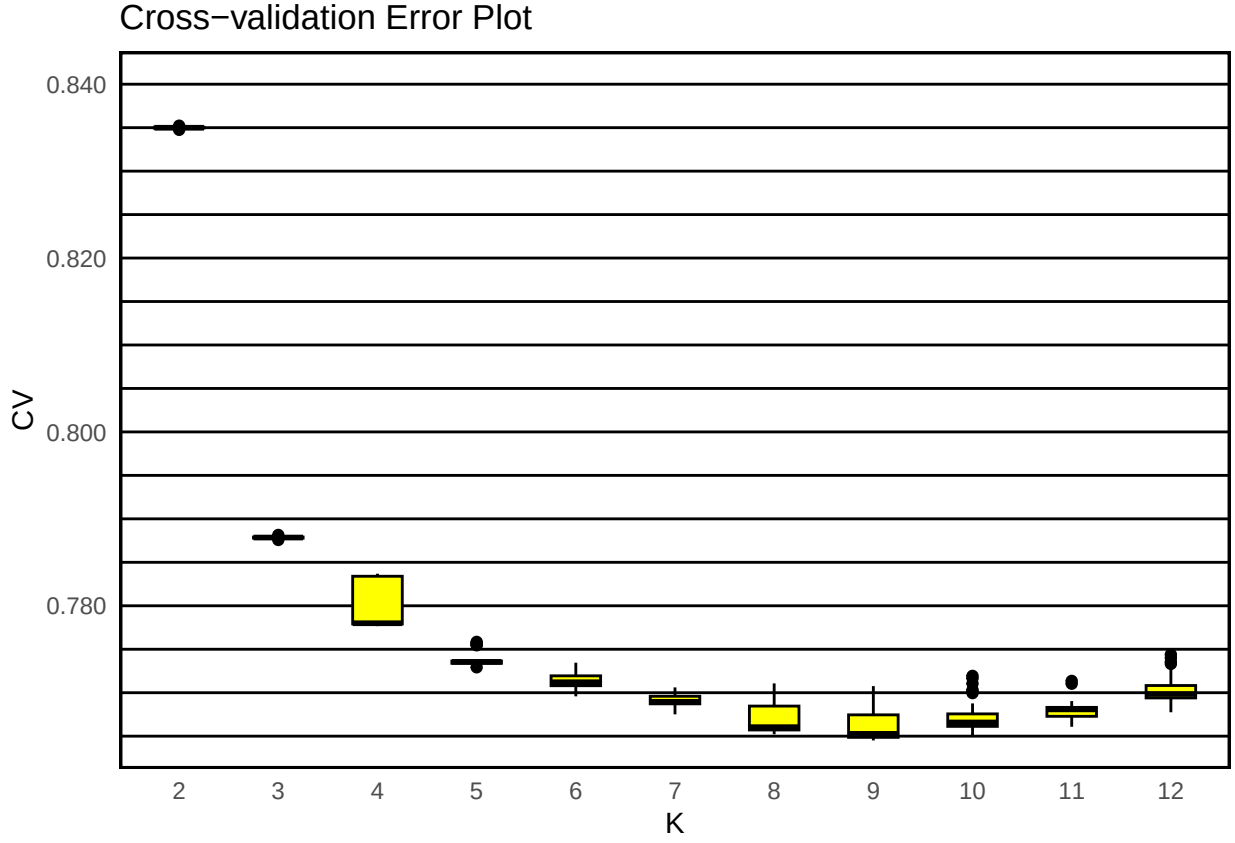
```
  geom_hline(
    yintercept = c(0.72,0.725,0.73,0.735,0.74,0.745,0.75,0.755,0.76,0.765,0.77,0.775,0.78,0.785,0.79,0.7
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.72, 0.8))
```



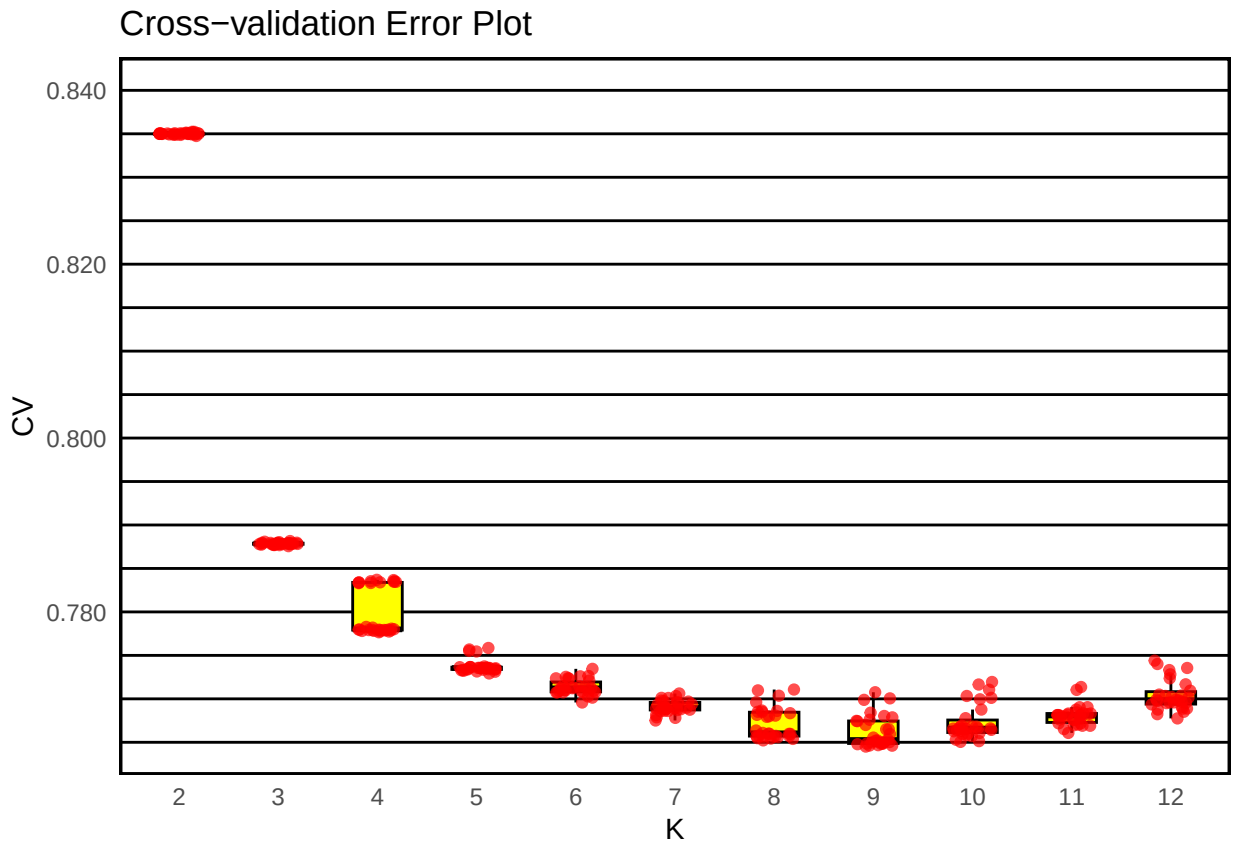

```
setwd("~/Documents/Stage_NB/data/SeqApiPop_629_SNPsBeeMuSe")
```

```r
# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_SNPsBeeMuSe_filtered_maf001_LD03_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.765, 0.77, 0.775, 0.78, 0.785, 0.79, 0.795, 0.8, 0.805, 0.81, 0.815,0.82,0.825,0.83
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.765, 0.84))
```

MAF > 0.01 - LD pruning = 0.3 (fenêtre de 1749 SNPs et pas de 175 bp) - 3848 SNPs



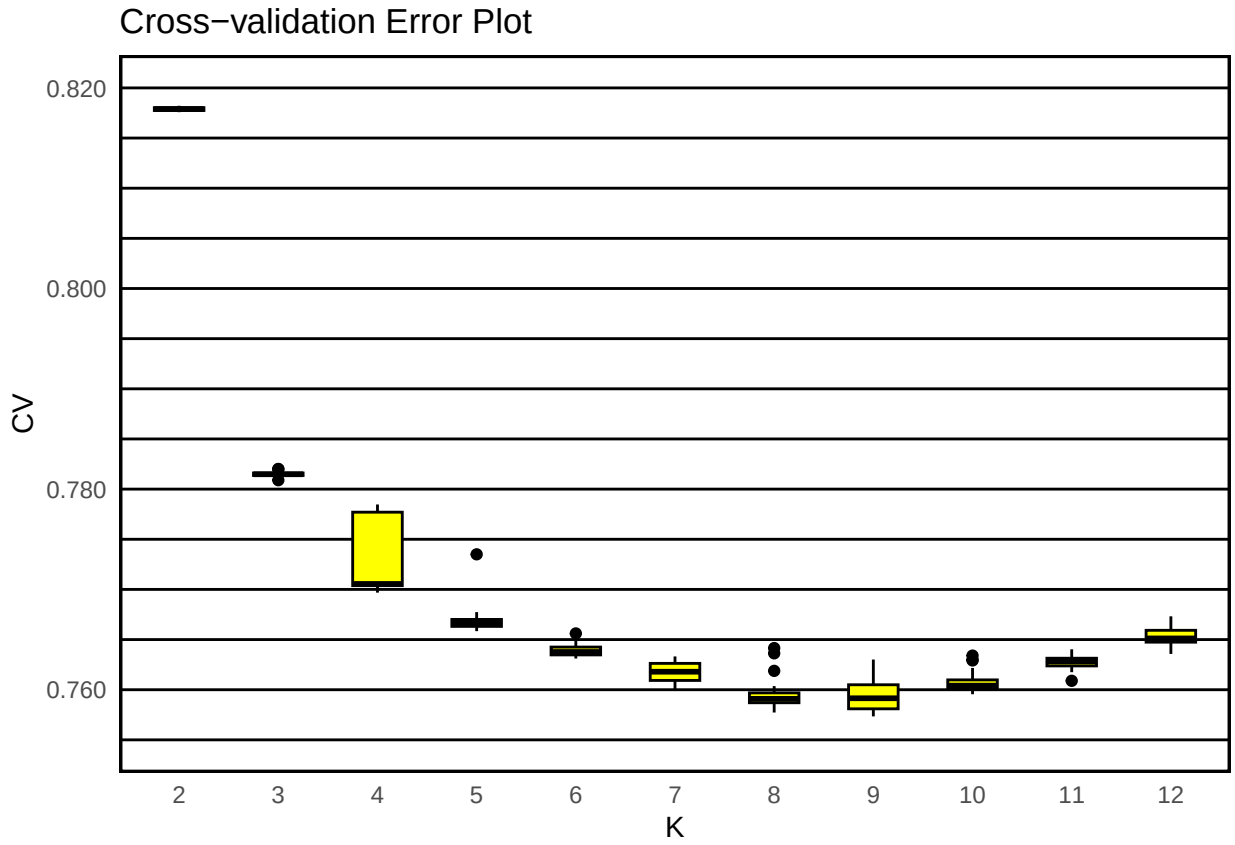

```
#jitter plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.765, 0.77, 0.775, 0.78, 0.785, 0.79, 0.795, 0.8, 0.805, 0.81, 0.815,0.82,0.825,0.83
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.765, 0.84))
```




```
setwd("~/Documents/Stage_NB/data/SeqApiPop_629_SNPsBeeMuSe")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_629_SNPsBeeMuSe_filtered_maf001_LD03_default_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)
```

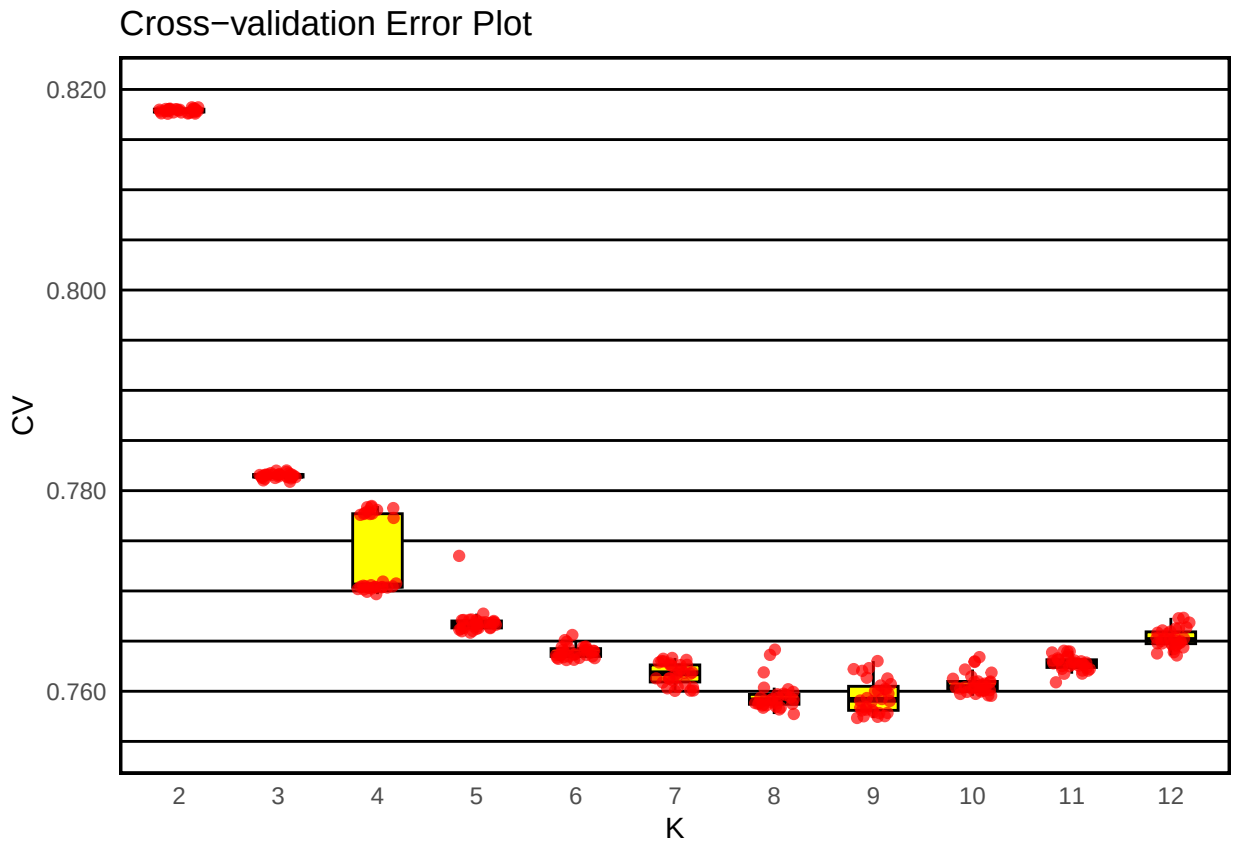
```r
#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.755, 0.76, 0.765, 0.77, 0.775, 0.78, 0.785, 0.79, 0.795, 0.8, 0.805, 0.81, 0.815,0
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.755, 0.82))
```

MAF > 0.01 - LD pruning = 0.1 (fenêtre de 50 SNPs et pas de 10 bp) - 1055 SNPs




```r
#jitter plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
```

```
  yintercept = c(0.755, 0.76, 0.765, 0.77, 0.775, 0.78, 0.785, 0.79, 0.795, 0.8, 0.805, 0.81, 0.815,0
  color = "black",
  linetype = "solid",
  size = 0.5
) +
geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
labs(title = "Cross-validation Error Plot",
     x = "K",
     y = "CV") +
scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
theme_minimal() +
theme(
  panel.border = element_rect(color = "black", fill = NA, size = 1),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()
) +
coord_cartesian(ylim = c(0.755, 0.82))
```

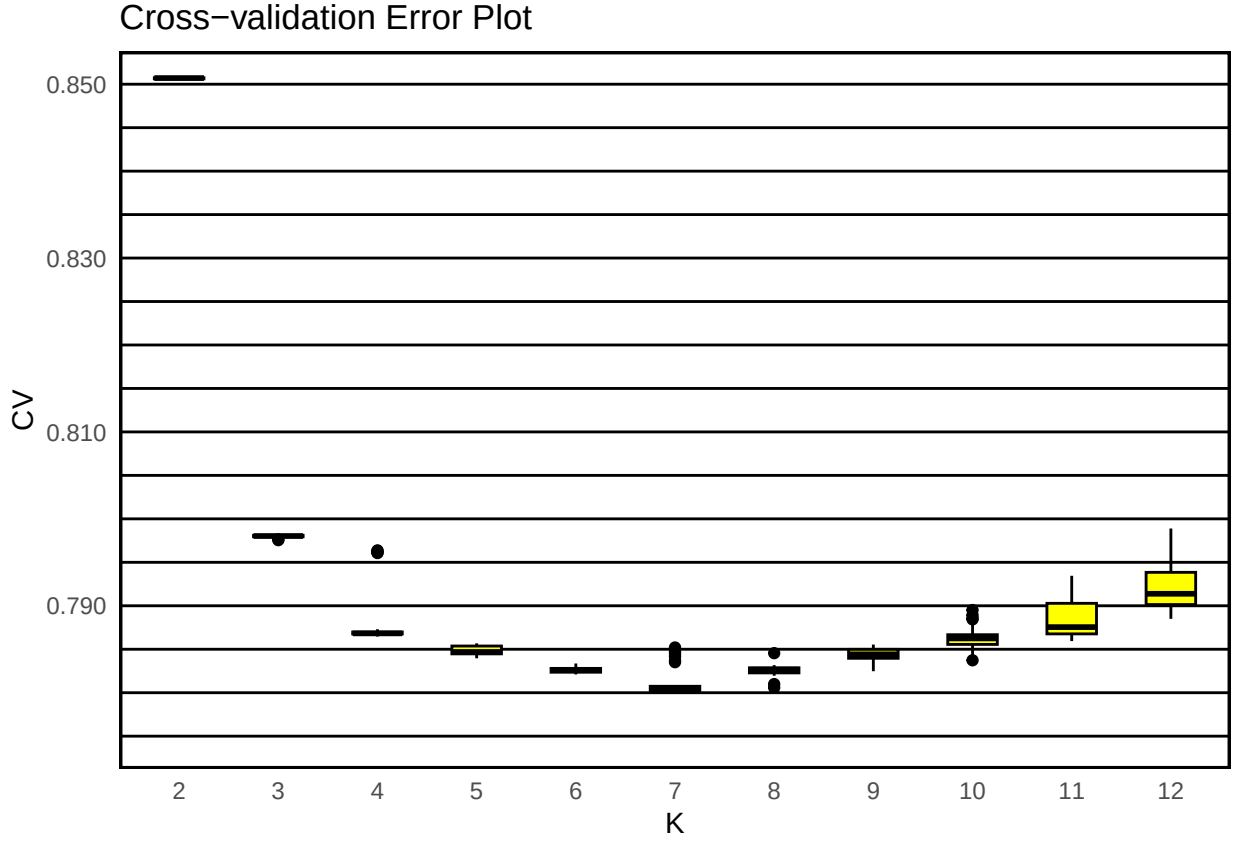


SeqApiPop - 561 échantillons - SNPsBeeMuSe filtered

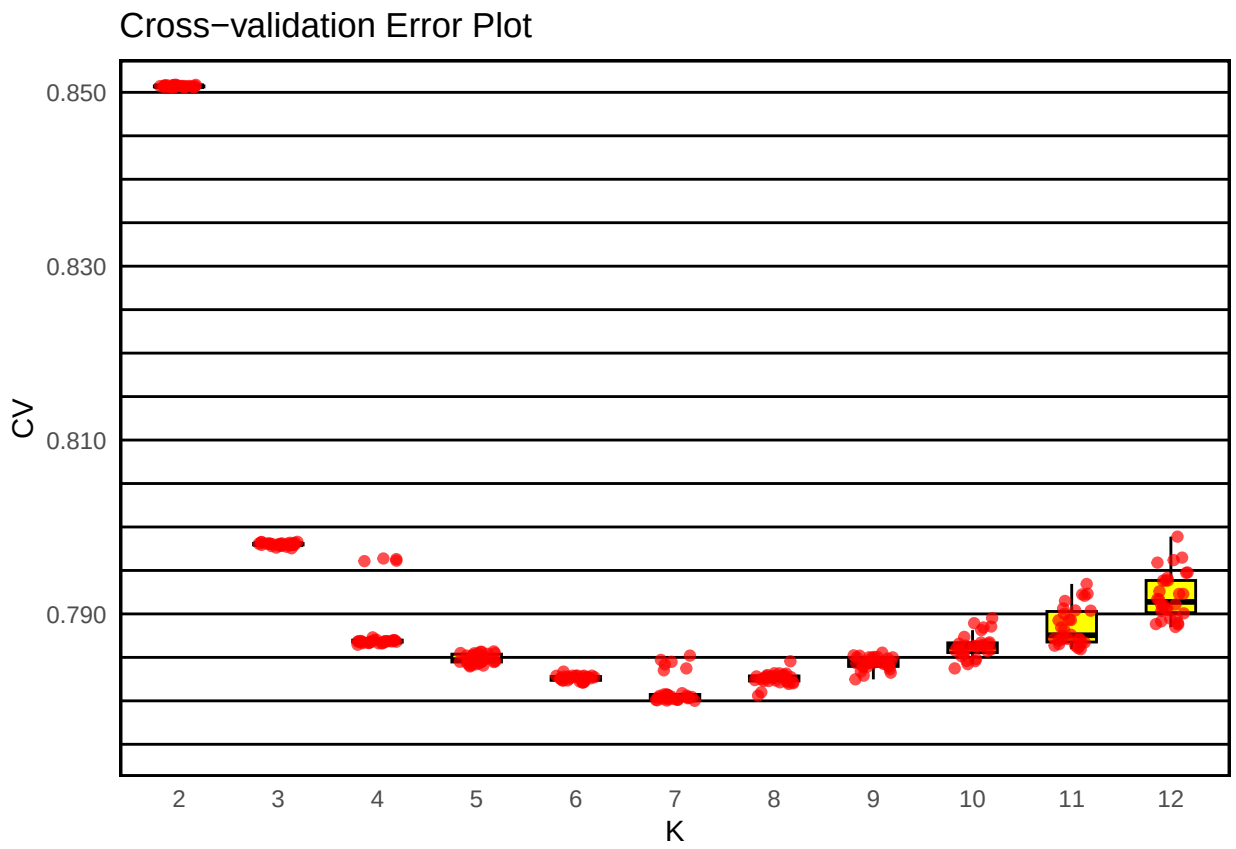
```
setwd("~/Documents/Stage_NB/data/SeqApiPop_561_SNPsBeeMuse_LD03")
```

```r
liste_de_donnees <- list()

for (i in 1:30) {
  merge_cv_error <- paste0('SeqApiPop_561_SNPsBeeMuSe_filtered_maf001_LD03_pruned_', i, '.cv.error')
  donnees <- read.table(merge_cv_error, header = FALSE)
  liste_de_donnees[[i]] <- donnees
}

donnees_combinees <- do.call(rbind, liste_de_donnees)
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)
merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.775, 0.78, 0.785, 0.79,0.795,0.8,0.805,0.81,0.815,0.82,0.825,0.83,0.835,0.84,0.845
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.775, 0.85))
```

MAF > 0.01 - LD pruning = 0.3 (fenêtre de 1749 SNPS et pas de 175 bp) - 3848 SNPs



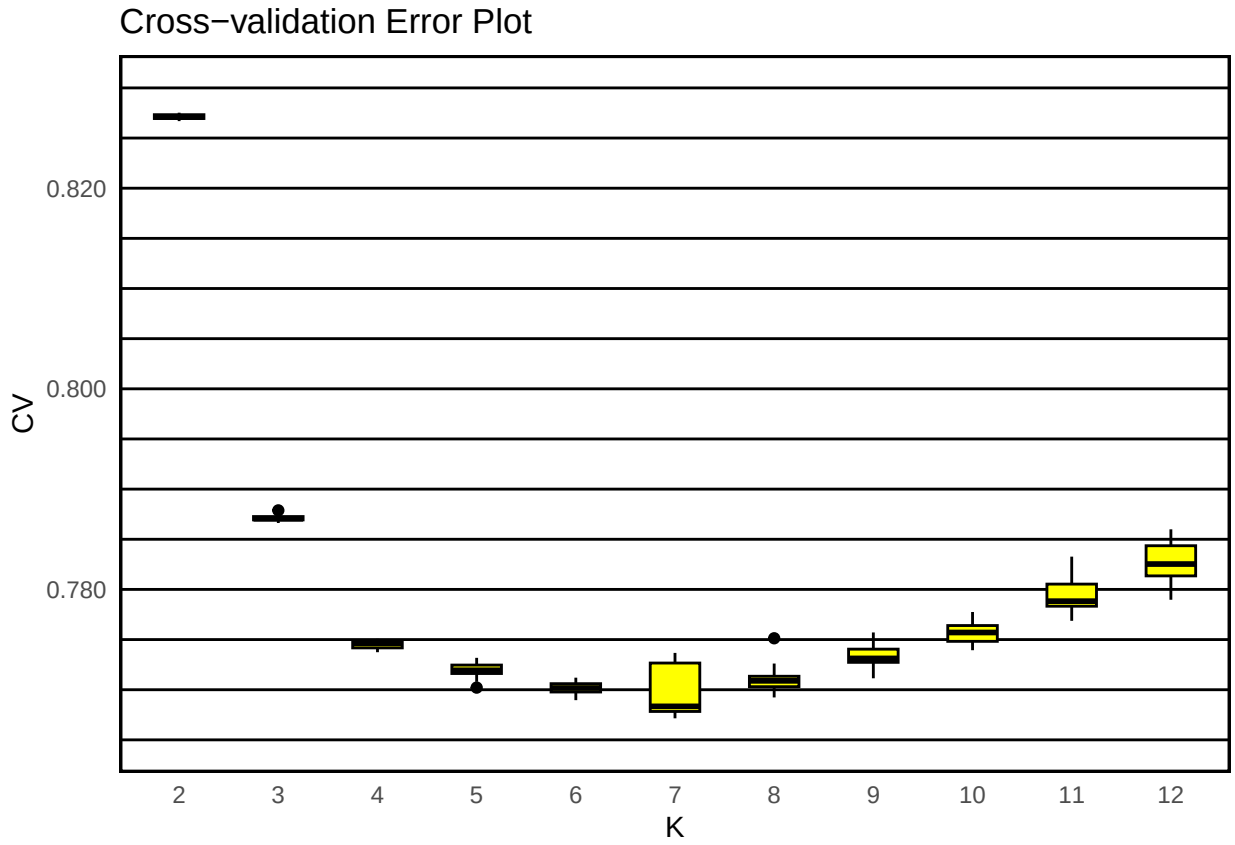

```
#jitter plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.775, 0.78, 0.785, 0.79,0.795,0.8,0.805,0.81,0.815,0.82,0.825,0.83,0.835,0.84,0.845
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.775, 0.85))
```




```
setwd("~/Documents/Stage_NB/data/SeqApiPop_561_SNPsBeeMuSe_LD_default")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('SeqApiPop_561_SNPsBeeMuSe_filtered_maf001_LD03_default_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)
```

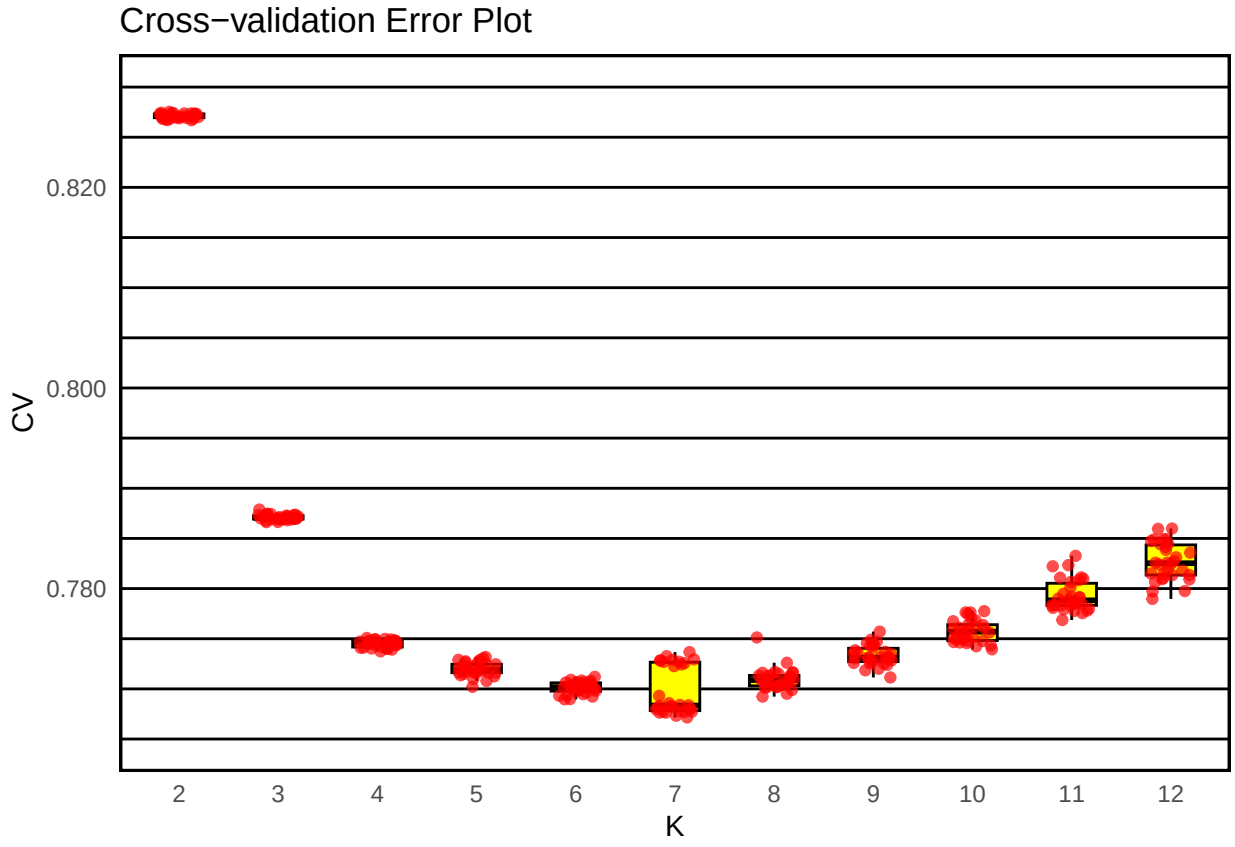
```r
#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.765,0.77, 0.775, 0.78, 0.785, 0.79,0.795,0.8,0.805,0.81,0.815,0.82,0.825,0.83),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.765, 0.83))
```

MAF > 0.01 - LD pruning = 0.1 (fenêtre de 50 SNPS et pas de 10 bp) - 1055 SNPs




```r
#jitter plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
```

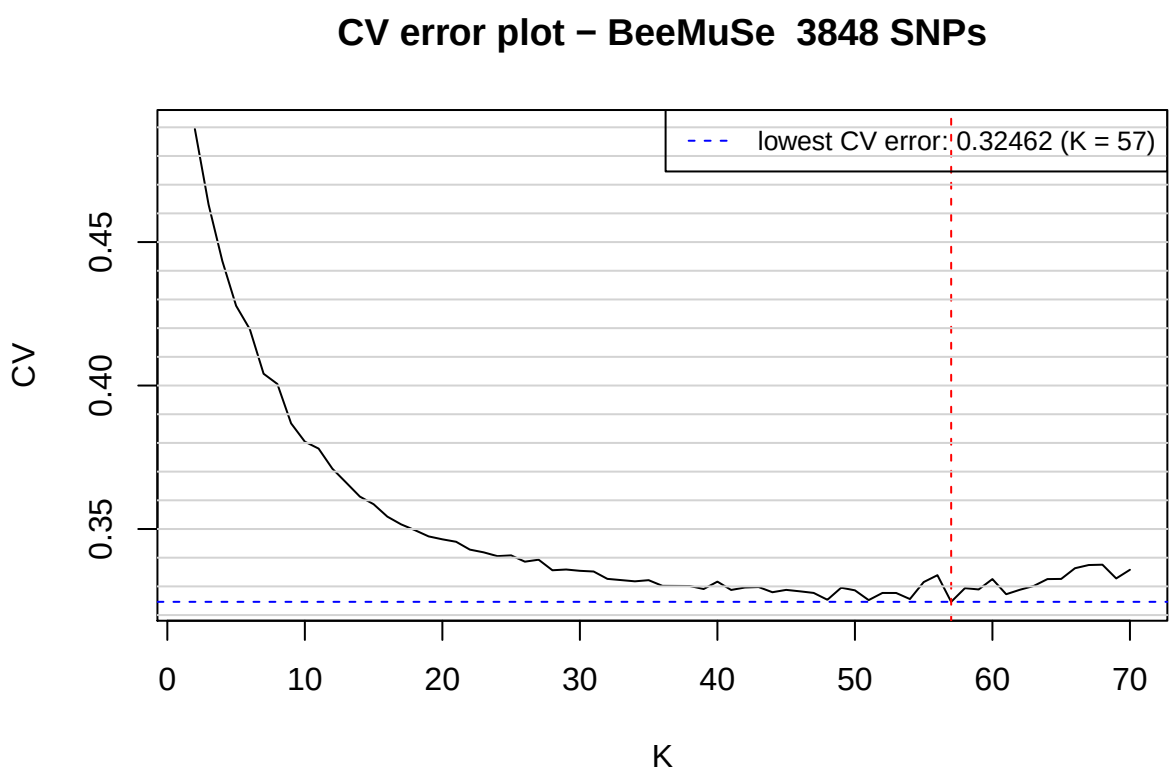
```
    yintercept = c(0.765,0.77, 0.775, 0.78, 0.785, 0.79,0.795,0.8,0.805,0.81,0.815,0.82,0.825,0.83),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.765, 0.83))
```



BeeMuSe - 12000 SNPs

```
K_new <- c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 
CV_new <- c(0.48943, 0.46307, 0.44338, 0.42781, 0.41962, 0.40411, 0.40055, 0.38682, 0.38044, 0.37802, 0
```

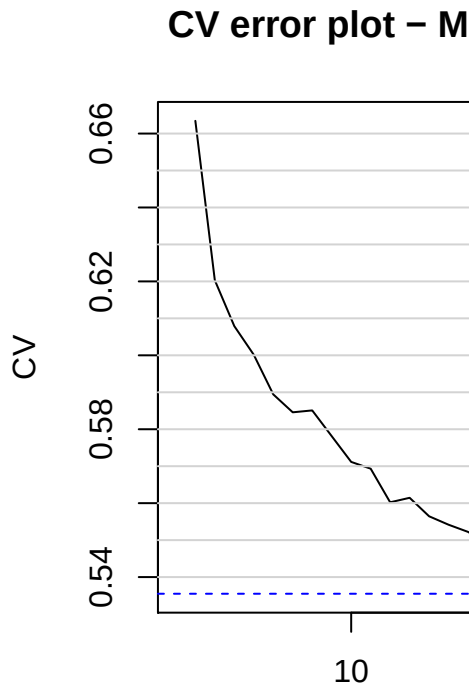
```r
# Trouver l'indice de la valeur la plus basse de CV
indice_min_new <- which.min(CV_new)

# Créer le graphique
plot(K_new, CV_new, type="l", col="black", xlab="K", ylab="CV", main="CV error plot - BeeMuSe  3848 SNPs

# Ajouter la ligne avec le trait hachuré bleu pour la valeur la plus basse
abline(h=CV_new[indice_min_new], col="blue", lty=2)

# Ajouter la droite verticale rouge pour la valeur la plus basse de CV
abline(v=K_new[indice_min_new], col="red", lty=2)

# Ajouter les lignes de grille horizontales à intervalles de 0.01
abline(h=seq(0, 1, by=0.01), col="lightgray")

# Ajouter la légende avec la valeur de K correspondant à la plus basse erreur CV
legend("topright", legend=sprintf("lowest CV error: %.5f (K = %d)", CV_new[indice_min_new], K_new[indic
```



Merged Data - BeeMuSe SeqApiPop

```r
# Valeurs CV - Admixture non supervisée
K <- c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 
```

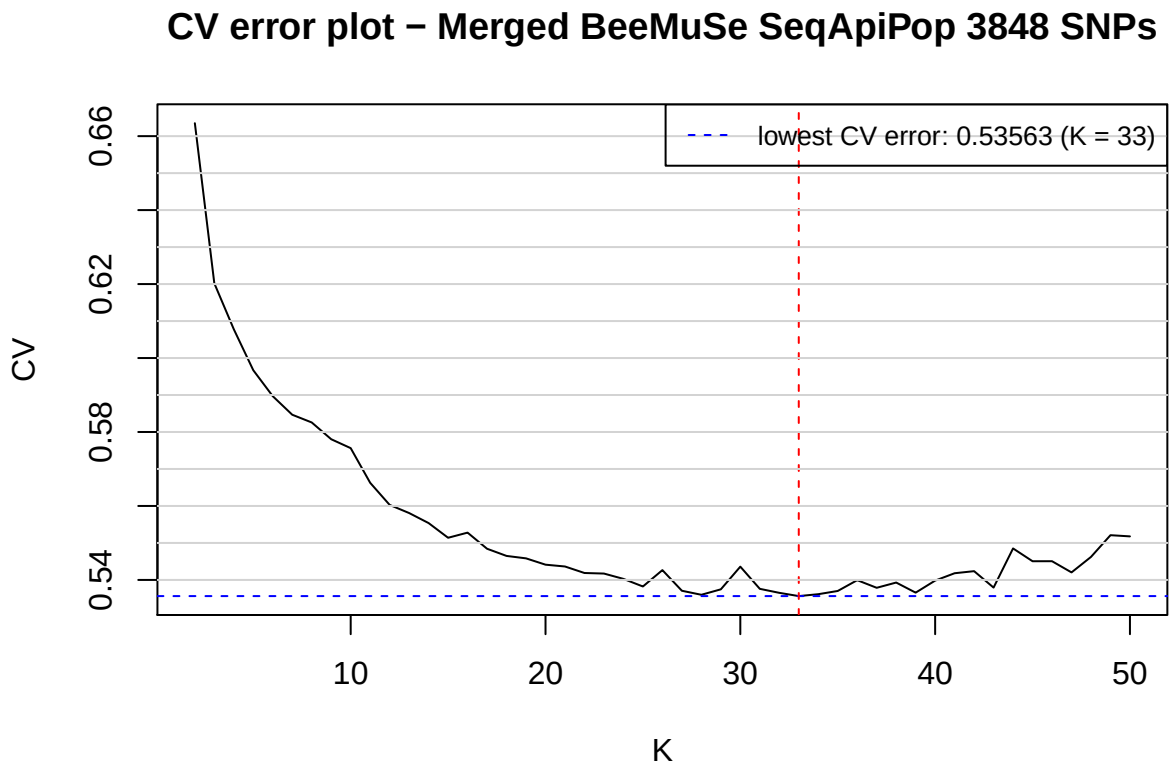
```
CV <- c(0.66344, 0.62032, 0.60790, 0.60015, 0.58943, 0.58458, 0.58509, 0.57812, 0.57116, 0.56933, 0.560
)

# Trouver l'indice de la valeur la plus basse de CV
indice_min <- which.min(CV)

# Créer le graphique
plot(K, CV, type="l", col="black", xlab="K", ylab="CV", main="CV error plot - Merged BeeMuSe SeqApiPop 

# Ajouter la ligne avec le trait hachuré bleu pour la valeur la plus basse
abline(h=CV[indice_min], col="blue", lty=2)

# Ajouter la droite verticale rouge pour la valeur la plus basse de CV
abline(v=K[indice_min], col="red", lty=2)

# Ajouter les lignes de grille horizontales à intervalles de 0.01
abline(h=seq(0, 1, by=0.01), col="lightgray")

# Ajouter la légende
legend("topright", legend=sprintf("lowest CV error: %.5f (K = %d)", CV[indice_min], K[indice_min]), col=
```



MAF > 0.01 - LD pruning = 0.3 (fenêtre de 1749 SNPS et pas de 175 bp)

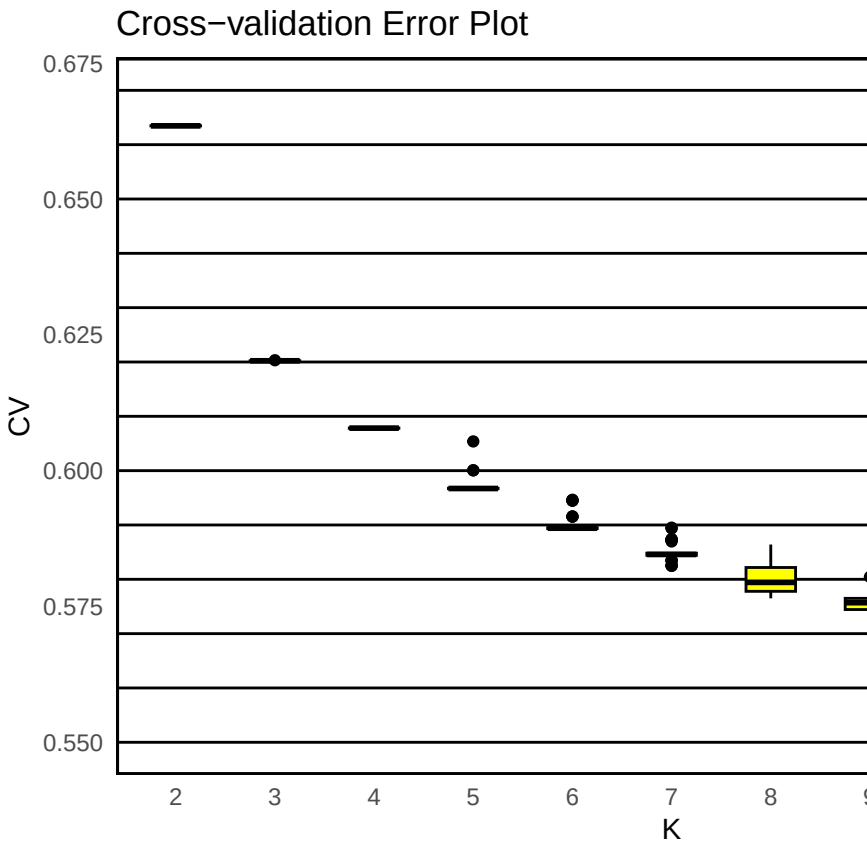
```
# Nouvelles données
K <- c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 
CV <- c(0.66350, 0.62020, 0.60778, 0.59670, 0.58968, 0.58467, 0.58258, 0.57804, 0.57563, 0.56626, 0.560
```

```r
# Trouver l'indice de la valeur la plus basse de CV
indice_min <- which.min(CV)

# Créer le graphique
plot(K, CV, type="l", col="black", xlab="K", ylab="CV", main="CV error plot - Merged BeeMuSe SeqApiPop 3

# Ajouter la ligne avec le trait hachuré bleu pour la valeur la plus basse
abline(h=CV[indice_min], col="blue", lty=2)

# Ajouter la droite verticale rouge pour la valeur la plus basse de CV
abline(v=K[indice_min], col="red", lty=2)

# Ajouter les lignes de grille horizontales à intervalles de 0.01
abline(h=seq(0, 1, by=0.01), col="lightgray")

# Ajouter la légende
legend("topright", legend=sprintf("lowest CV error: %.5f (K = %d)", CV[indice_min], K[indice_min]), col=
```



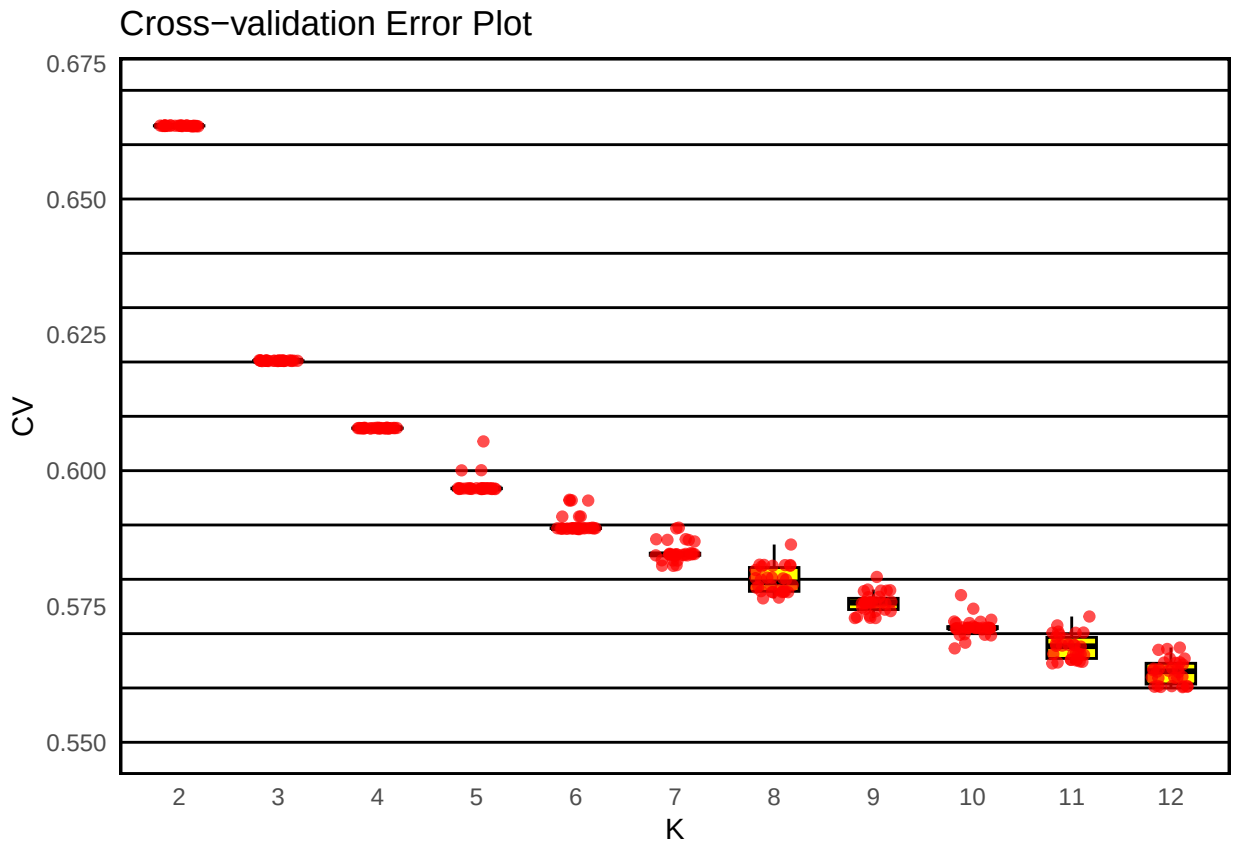

```r
setwd("~/Documents/Stage_NB/data/merged_BeeMuSe_SeqApiPop_629_filtered_maf001_LD03")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()
```

```r
# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_629_filtered_maf001_LD03_', i, '.cv.error')
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.55, 0.67))
```



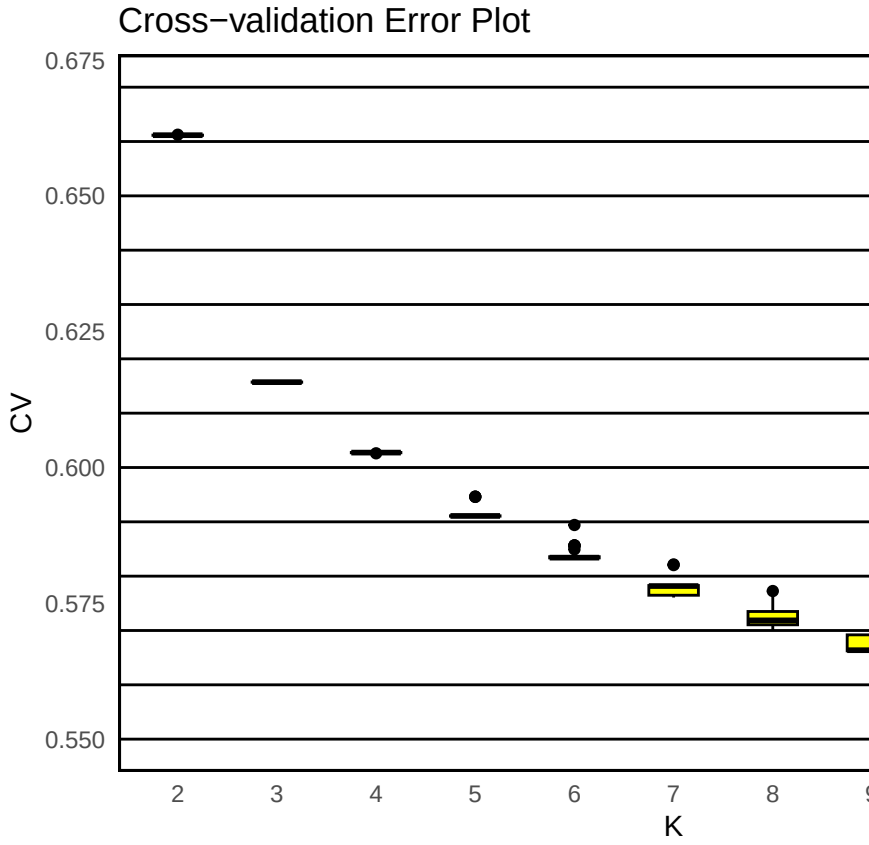
629 échantillons - K2 à K9 - 30 exécutions

```
#jitter plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62, 0.63, 0.64, 0.65, 0.66, 0.67),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.55, 0.67))
```




```
setwd("~/Documents/Stage_NB/data/merged_data_3848_561_not_supervised")

liste_de_donnees <- list()
for (i in 1:30) {
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_561_filtered_maf001_LD03_', i, '.cv.error')
  donnees <- read.table(merge_cv_error, header = FALSE)
  liste_de_donnees[[i]] <- donnees
}

donnees_combinees <- do.call(rbind, liste_de_donnees)
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)
merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.54,0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62, 0.63, 0.64, 0.65,0.66,0.67),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
```

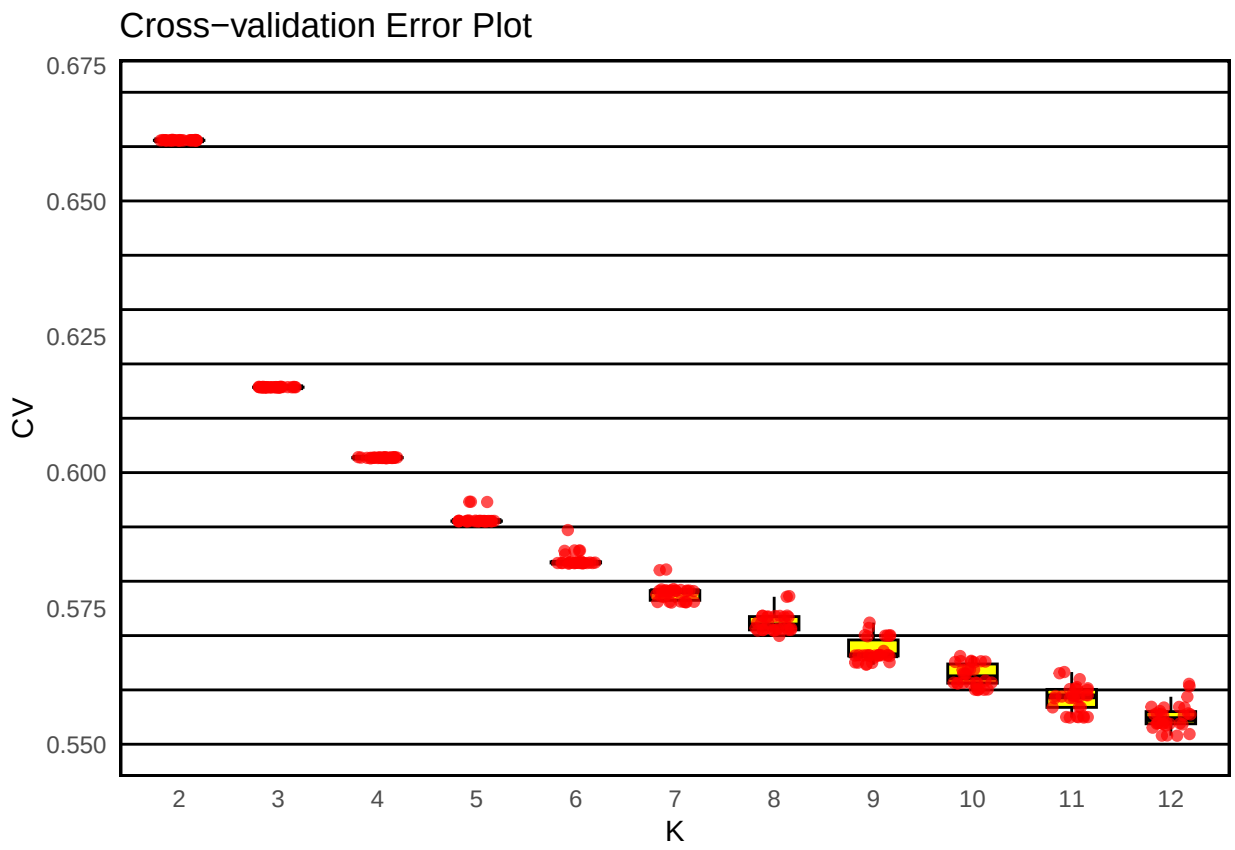
```
      x = "K",
      y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.55, 0.67))
```



561 échantillons - K2 à K9 - 30 exécutions

```
#jitter plot filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.54,0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62, 0.63, 0.64, 0.65,0.66,0.67),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
```

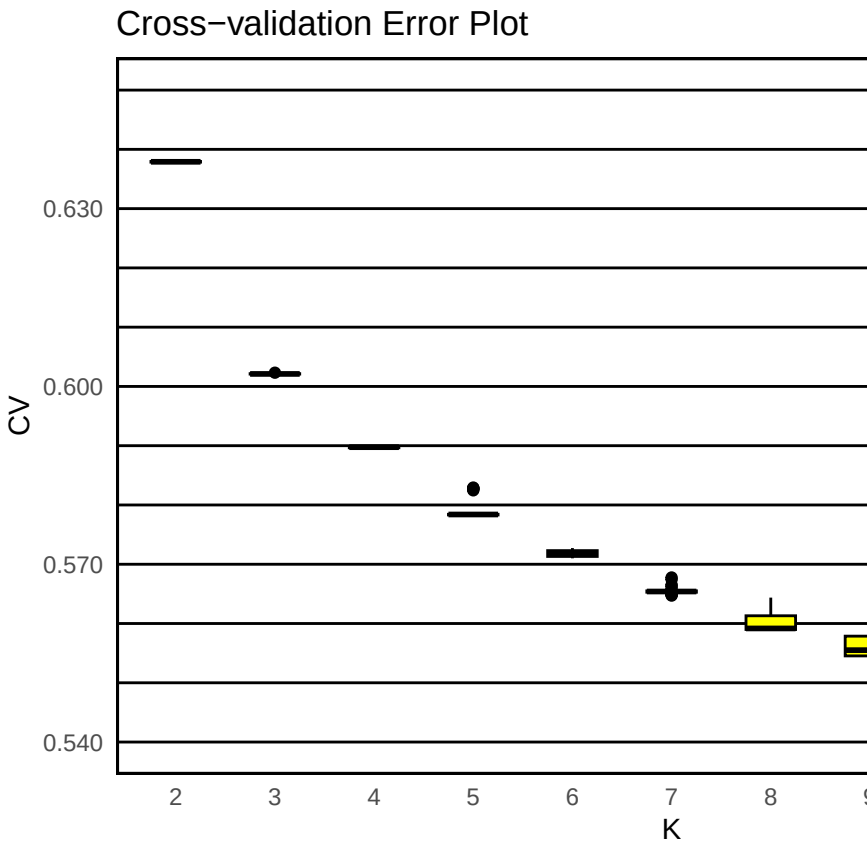
```
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.55, 0.67))
```



MAF > 0.01 - LD pruning = 0.1 (fenêtre de 50 SNPS et pas de 10 bp)

```
setwd("~/Documents/Stage_NB/data/merged_data_1055_561_not_supervised")

liste_de_donnees <- list()
for (i in 1:30) {
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_561_filtered_MAF001_LD_default_', i, '.cv.error')
  donnees <- read.table(merge_cv_error, header = FALSE)
  liste_de_donnees[[i]] <- donnees
}

donnees_combinees <- do.call(rbind, liste_de_donnees)
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)
merge_cv_error <- read.table("merge_cv_error", header = F)
```

```r
#box plot filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.54,0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62, 0.63, 0.64, 0.65),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.54, 0.65))
```



561 échantillons - K2 à K9 - 30 exécutions

```r
#jitter plot filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.54,0.55, 0.56, 0.57, 0.58, 0.59, 0.6, 0.61, 0.62, 0.63, 0.64, 0.65),
```

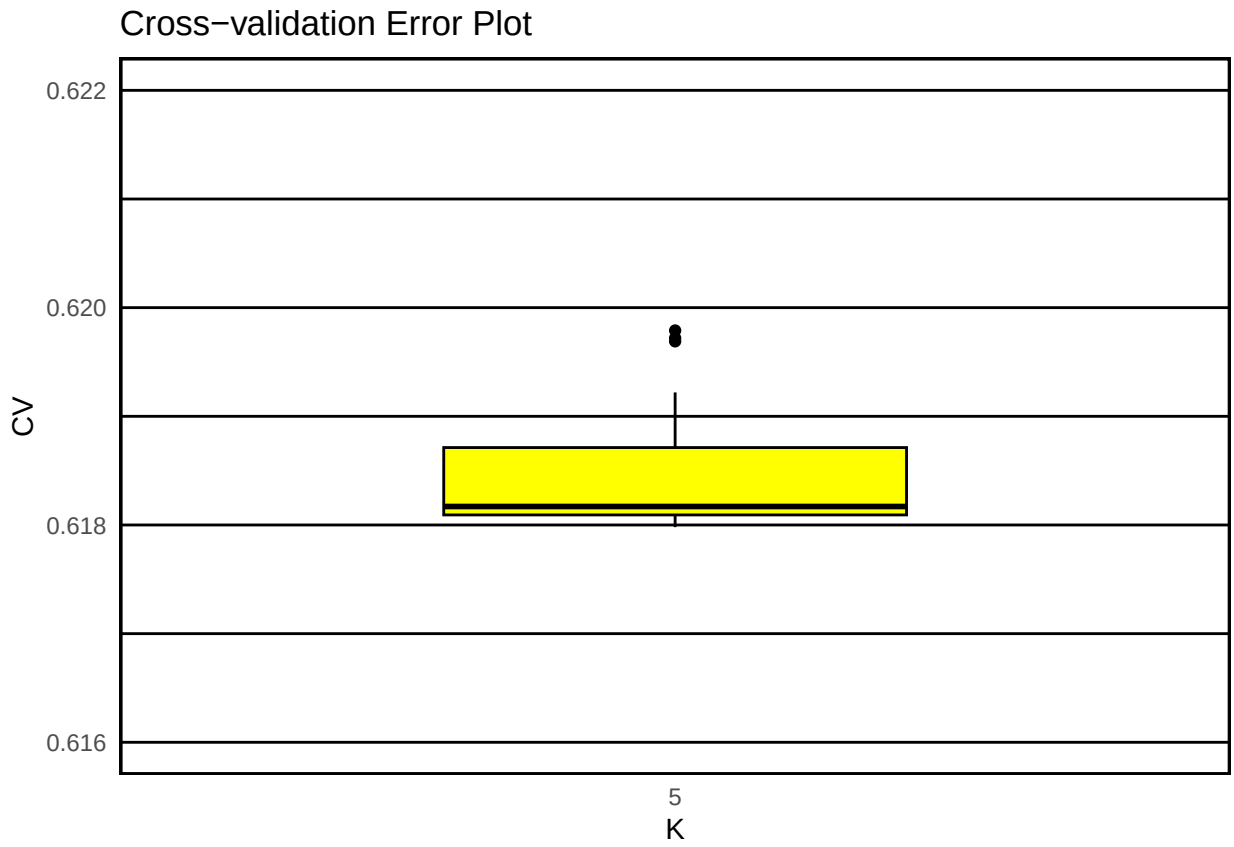
```
  color = "black",
  linetype = "solid",
  size = 0.5
) +
geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
labs(title = "Cross-validation Error Plot",
     x = "K",
     y = "CV") +
scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
theme_minimal() +
theme(
  panel.border = element_rect(color = "black", fill = NA, size = 1),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()
) +
coord_cartesian(ylim = c(0.54, 0.65))
```

Admixture supervisée

Merged Data - BeeMuSe SeqApiPop

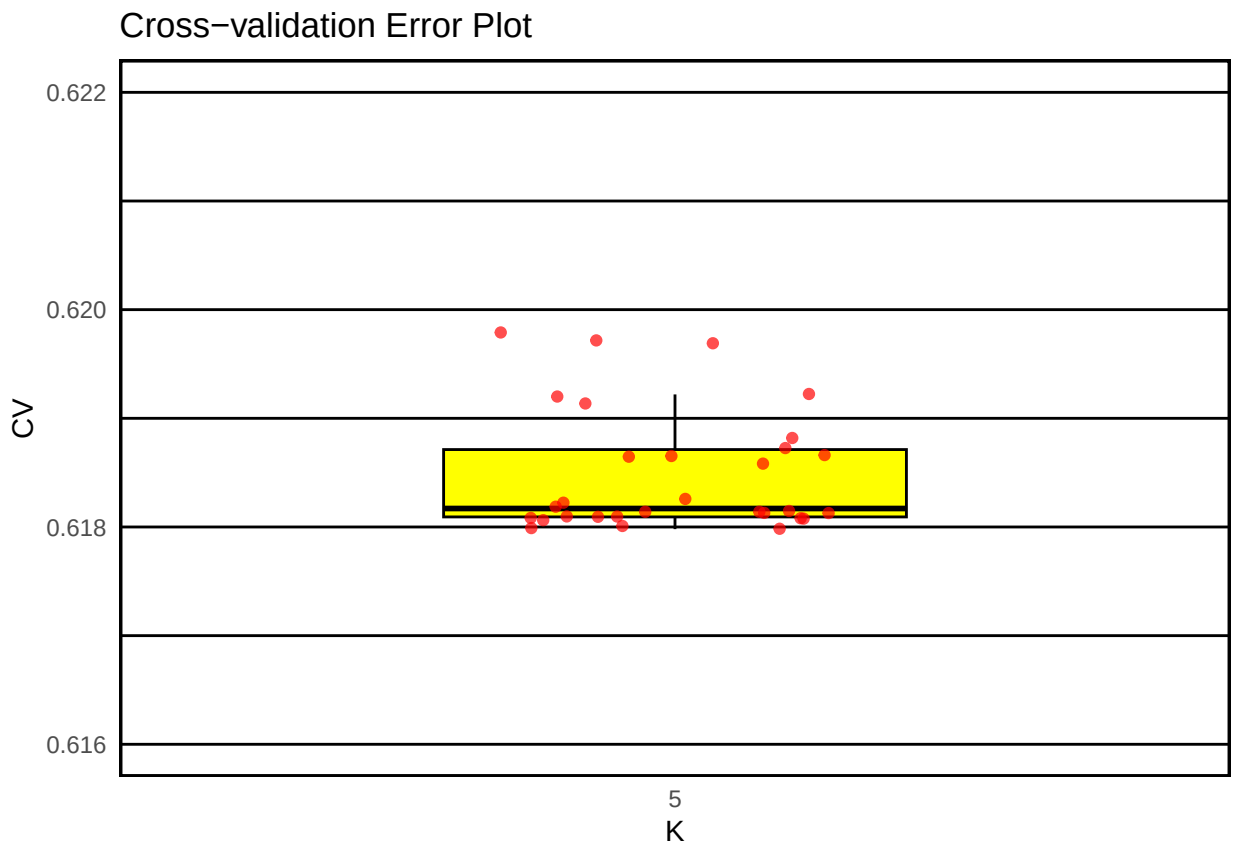
629 échantillons - LD pruning = 0.1 (fenêtre de 50 et pas de 10 bp) - 1055 SNPs

```r
setwd("~/Documents/Stage_NB/data/merged_data_3848_629_supervised")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_629_filtered_MAF001_LD_default_K5_95_supervised_',
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.616,0.617,0.618,0.619,0.62,0.621,0.622),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.616, 0.622))
```



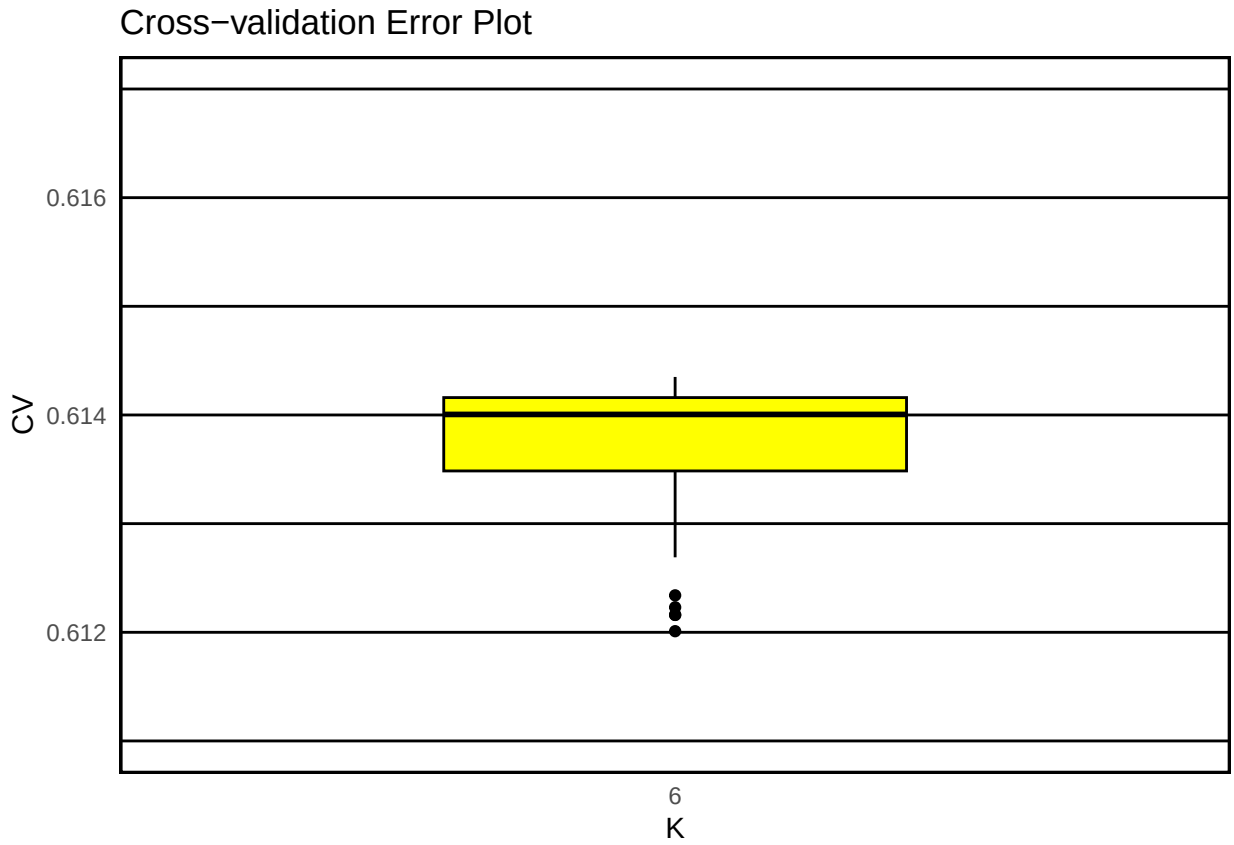
$\mathbf{K = 5}$

```
#jitter plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.616,0.617,0.618,0.619,0.62,0.621,0.622),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.616, 0.622))
```




```r
setwd("~/Documents/Stage_NB/data/merged_data_3848_629_supervised")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_629_filtered_MAF001_LD_default_K6_95_supervised_',
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)
```

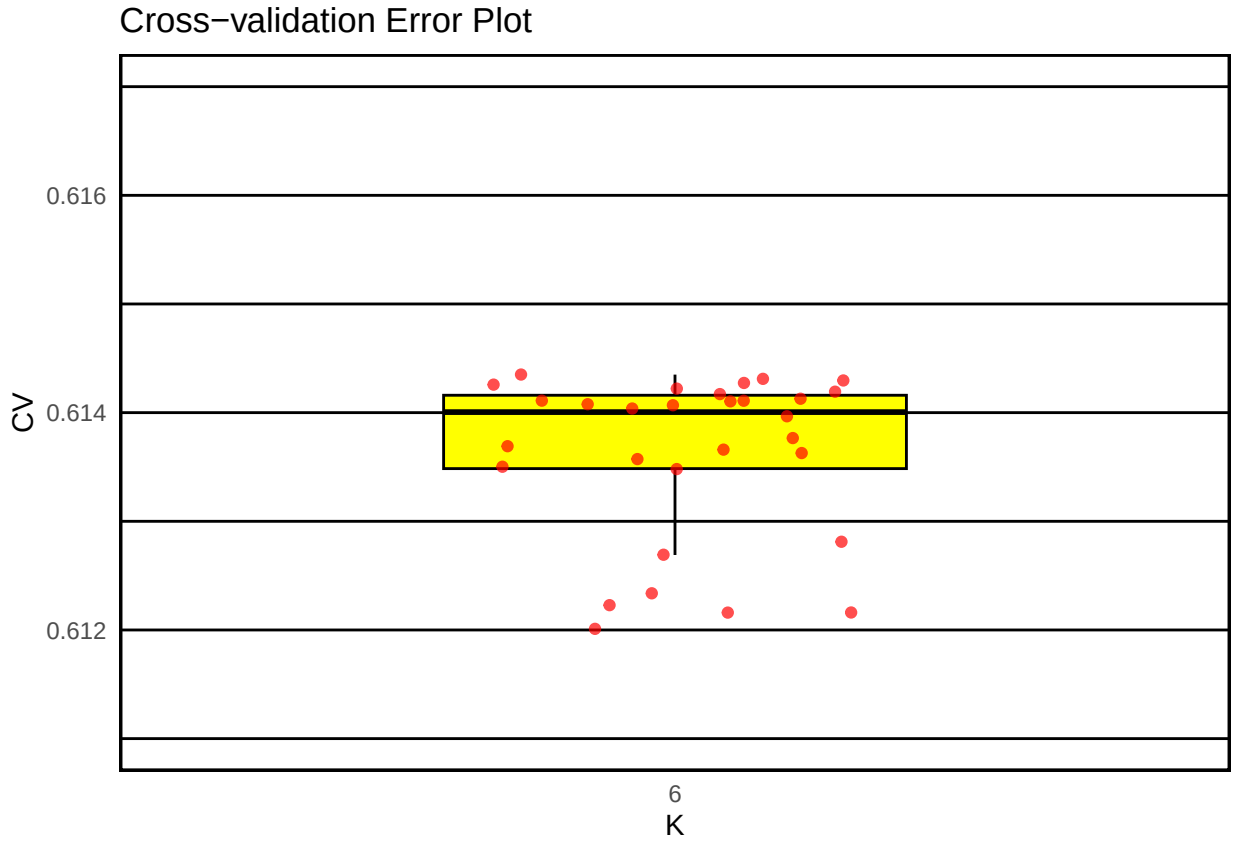
```r
#box plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.611,0.612,0.613,0.614,0.615,0.616,0.617),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.611, 0.617))
```



K = 6

```r
#jitter plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.611,0.612,0.613,0.614,0.615,0.616,0.617),
```

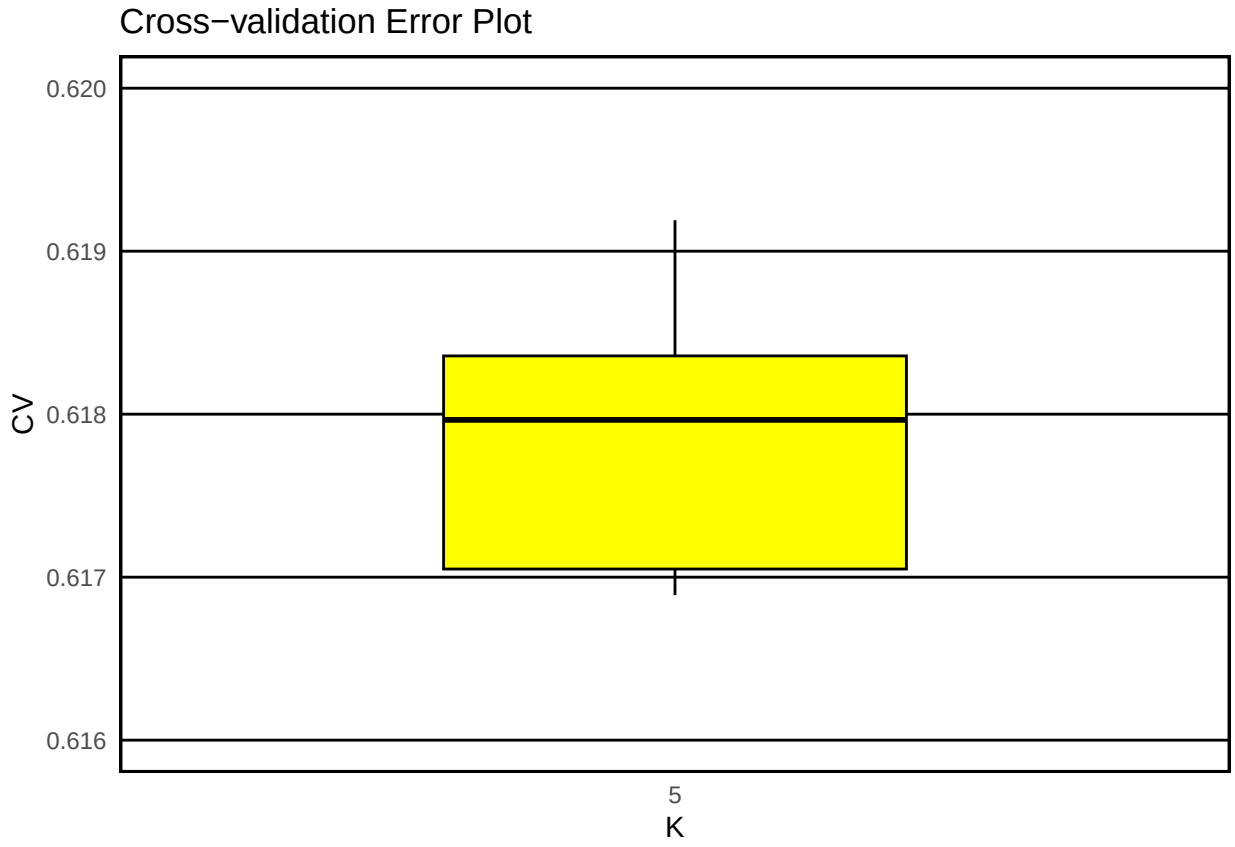
```
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.611, 0.617))
```



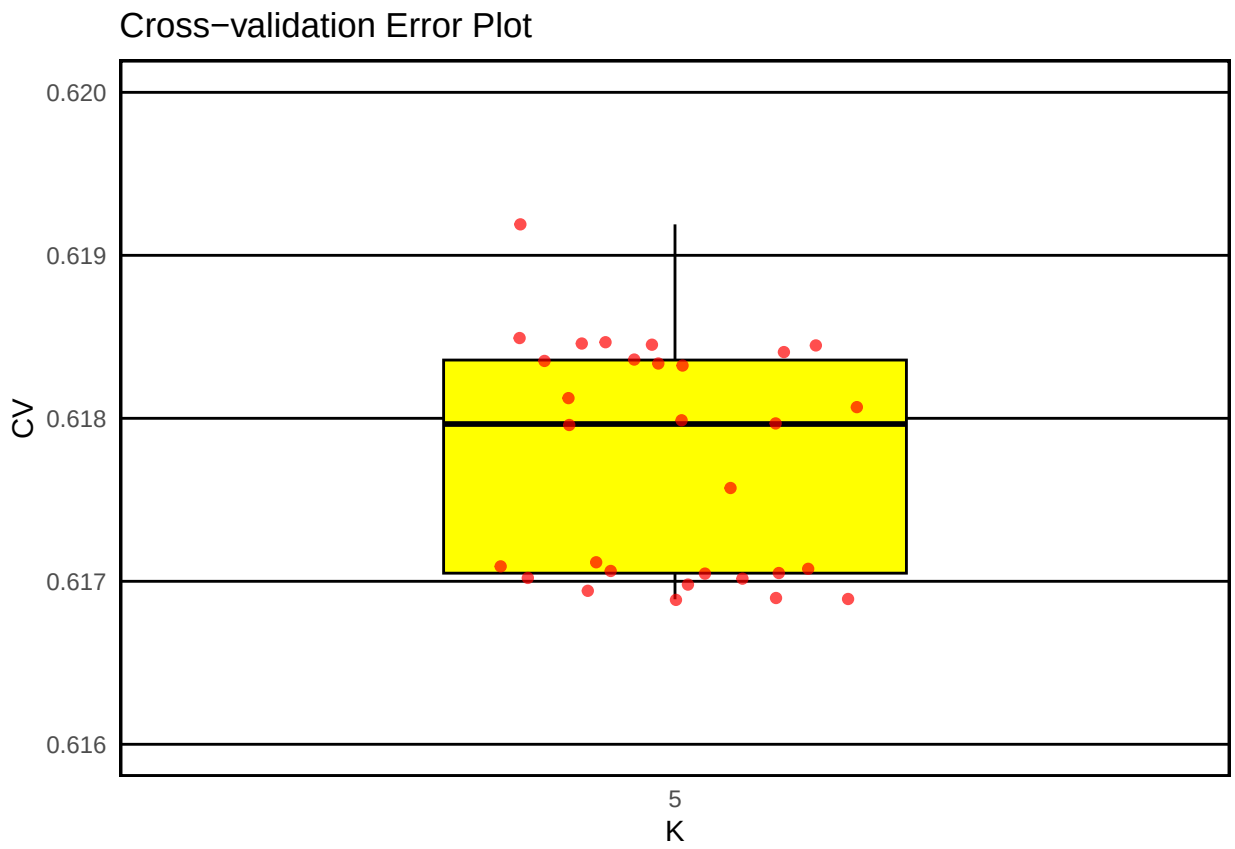

```
setwd("~/Documents/Stage_NB/data/merged_data_3848_561_supervised")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()
```

```r
# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_561_filtered_MAF001_LD_default_K5_95_supervised_',
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.616,0.617,0.618,0.619,0.62),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.616, 0.62))
```

561 échantillons - LD pruning = 0.3 (fenêtre de 1749 et pas de 175 bp) K = 5 - 3848 SNPs

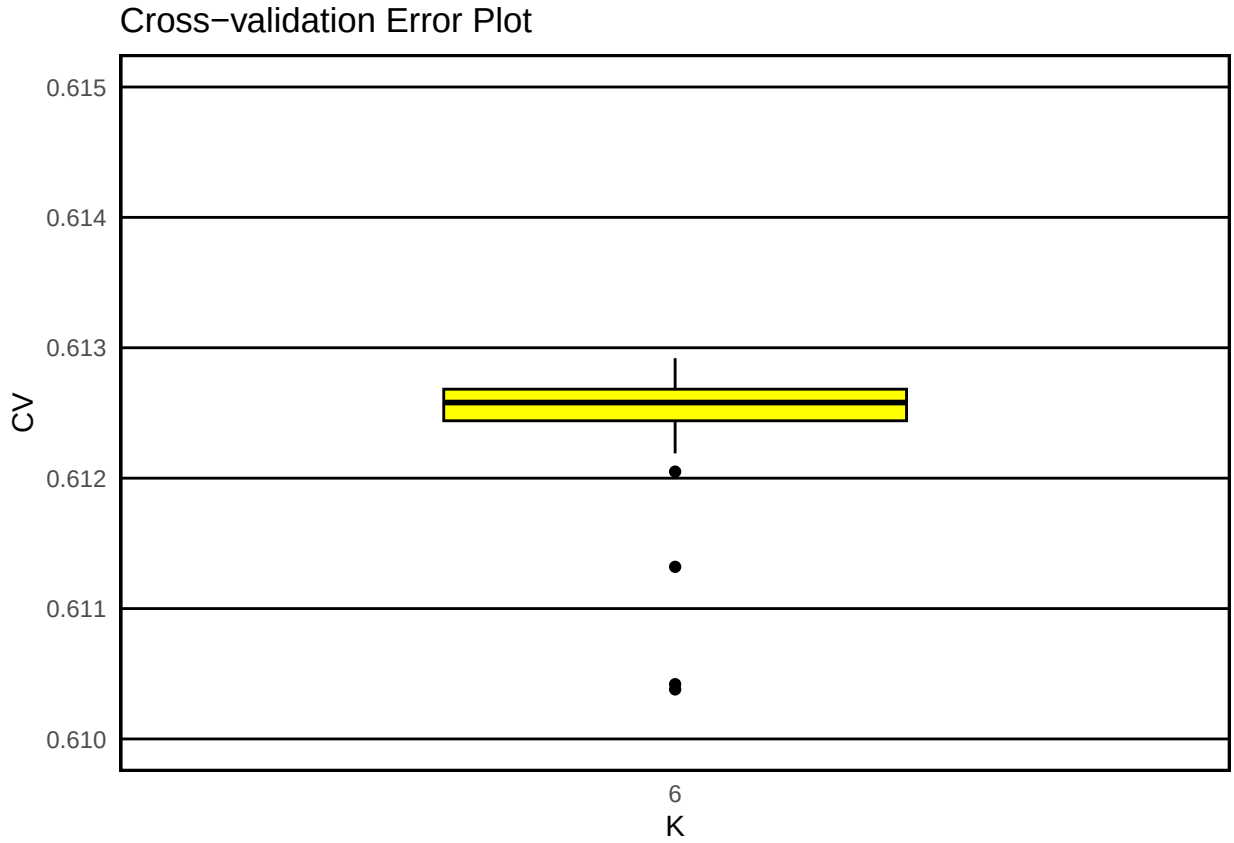



```
#jitter plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.616,0.617,0.618,0.619,0.62),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.616, 0.62))
```




```
setwd("~/Documents/Stage_NB/data/merged_data_3848_561_supervised")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_561_filtered_MAF001_LD_default_K6_95_supervised_',
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)
```

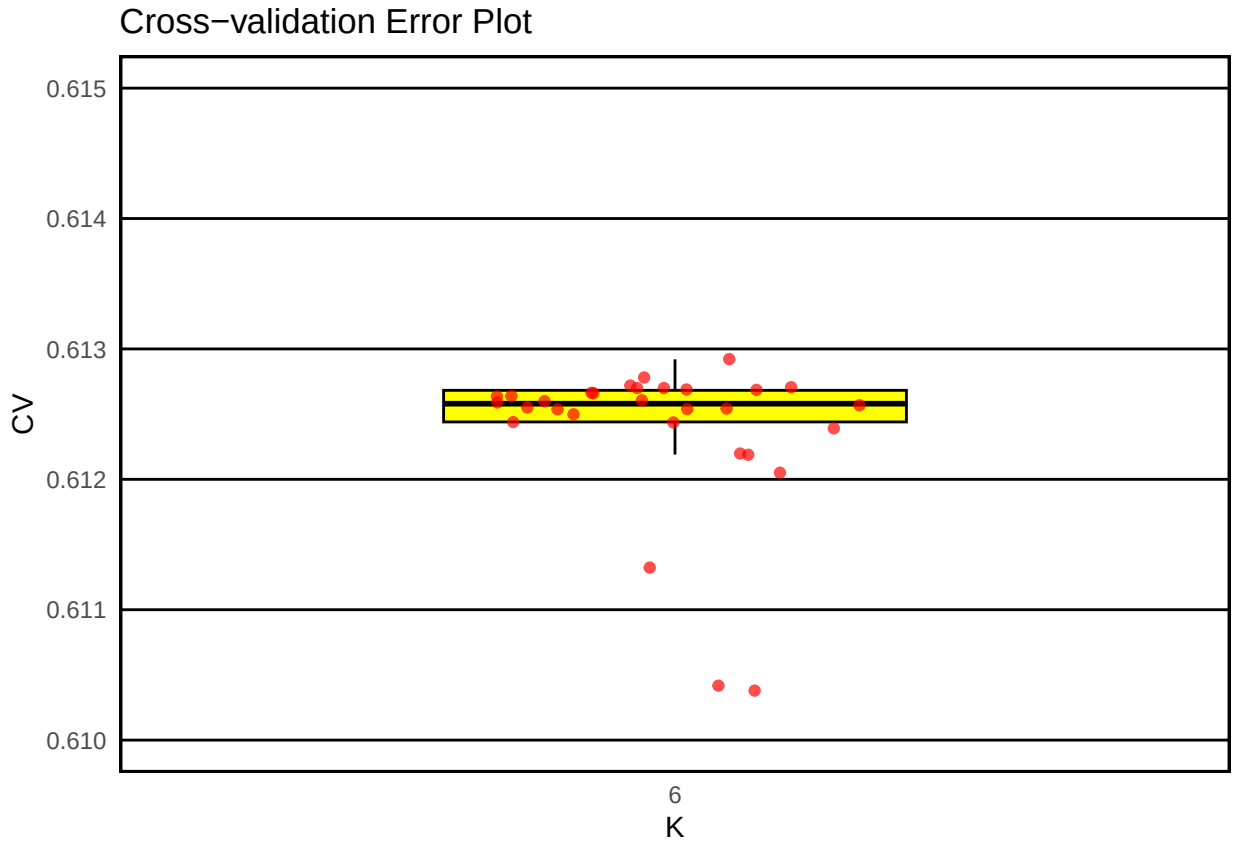
```r
#box plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.61,0.611,0.612,0.613,0.614,0.615),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.61, 0.615))
```



K = 6

```r
#jitter plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.61,0.611,0.612,0.613,0.614,0.615),
```

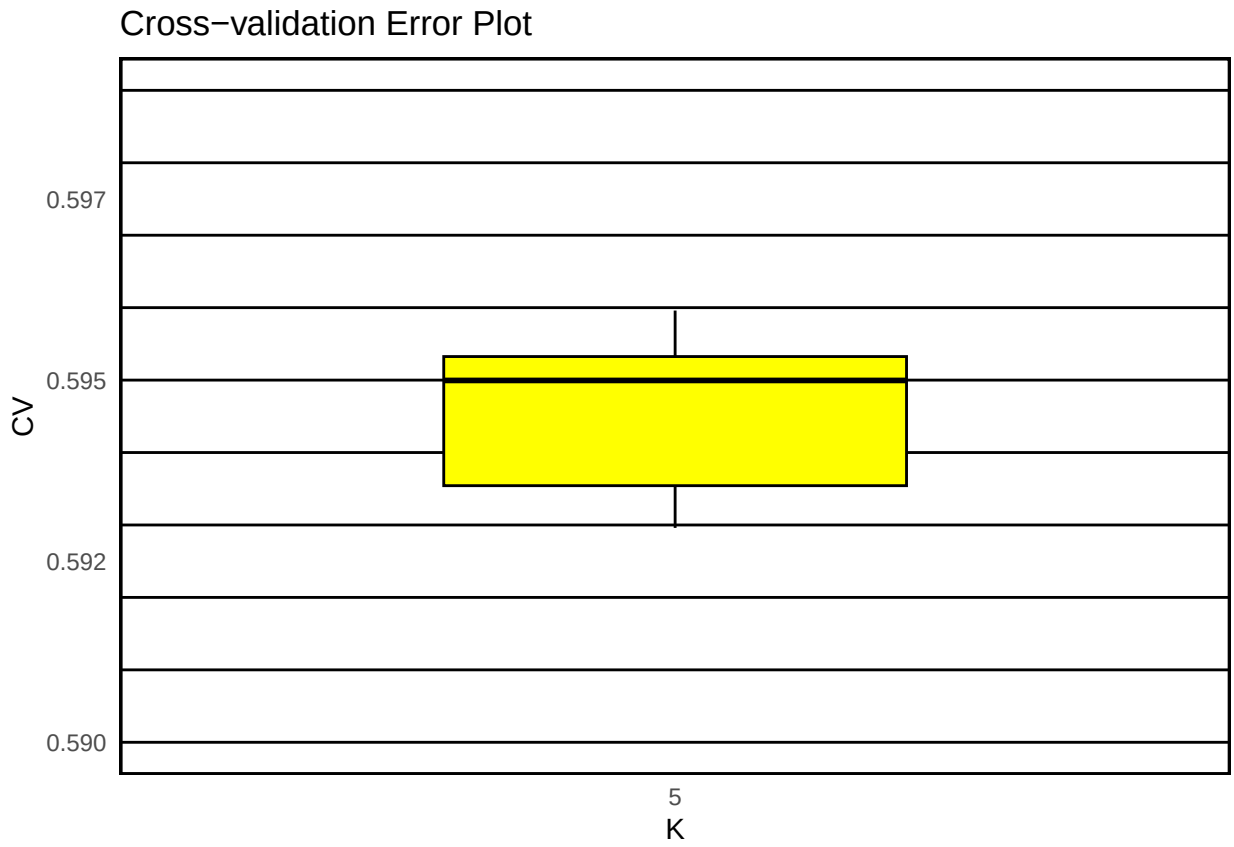
```
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.61, 0.615))
```



561 échantillons - LD pruning = 0.1 (fenêtre de 50 et pas de 10 bp) - 1055 SNPs

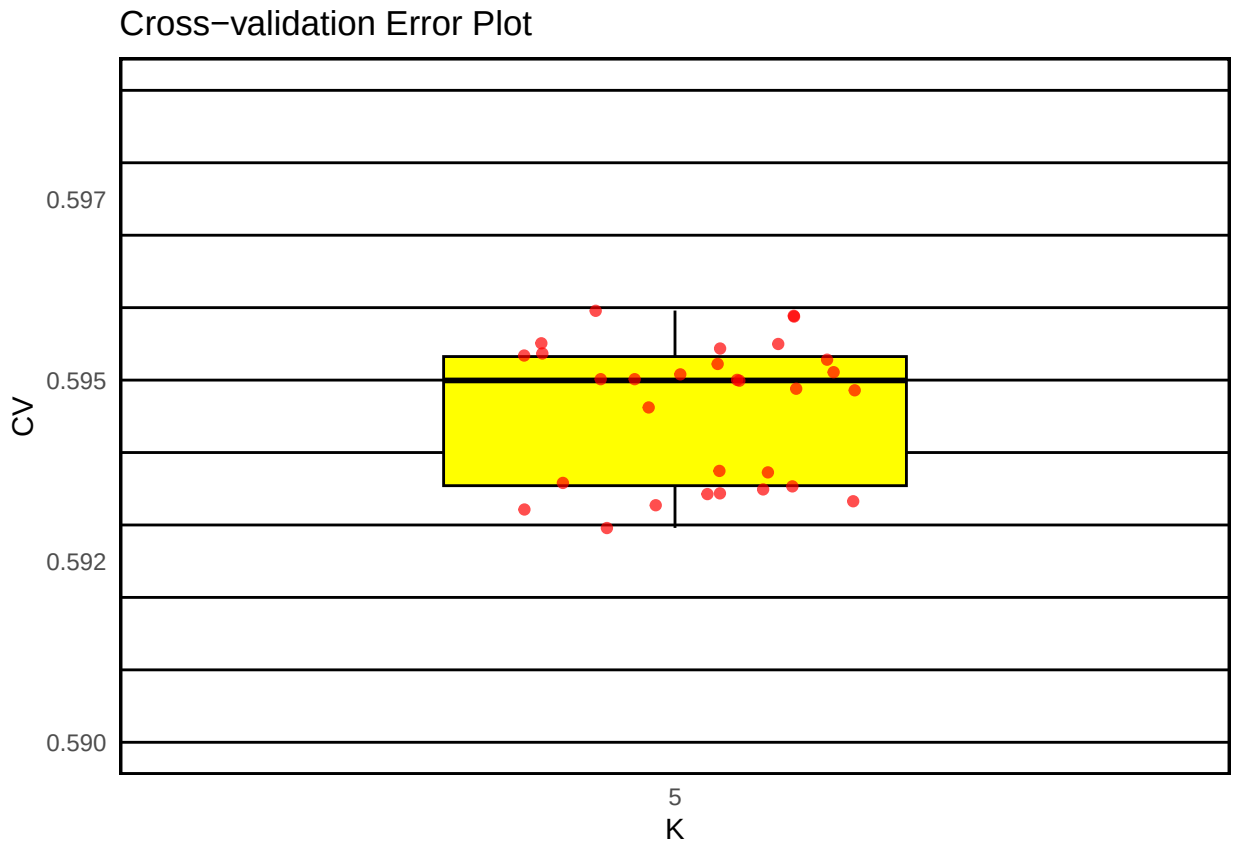
```
setwd("~/Documents/Stage_NB/data/merged_data_1055_561_supervised")
```

```r
# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_561_filtered_MAF001_LD_default_K5_90_supervised_',
  #merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_561_filtered_MAF001_LD_default_K5_95_supervised_'
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)

#box plot LD03 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.59, 0.591, 0.592, 0.593,0.594,0.595,0.596,0.597,0.598,0.599),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.59, 0.599))
```



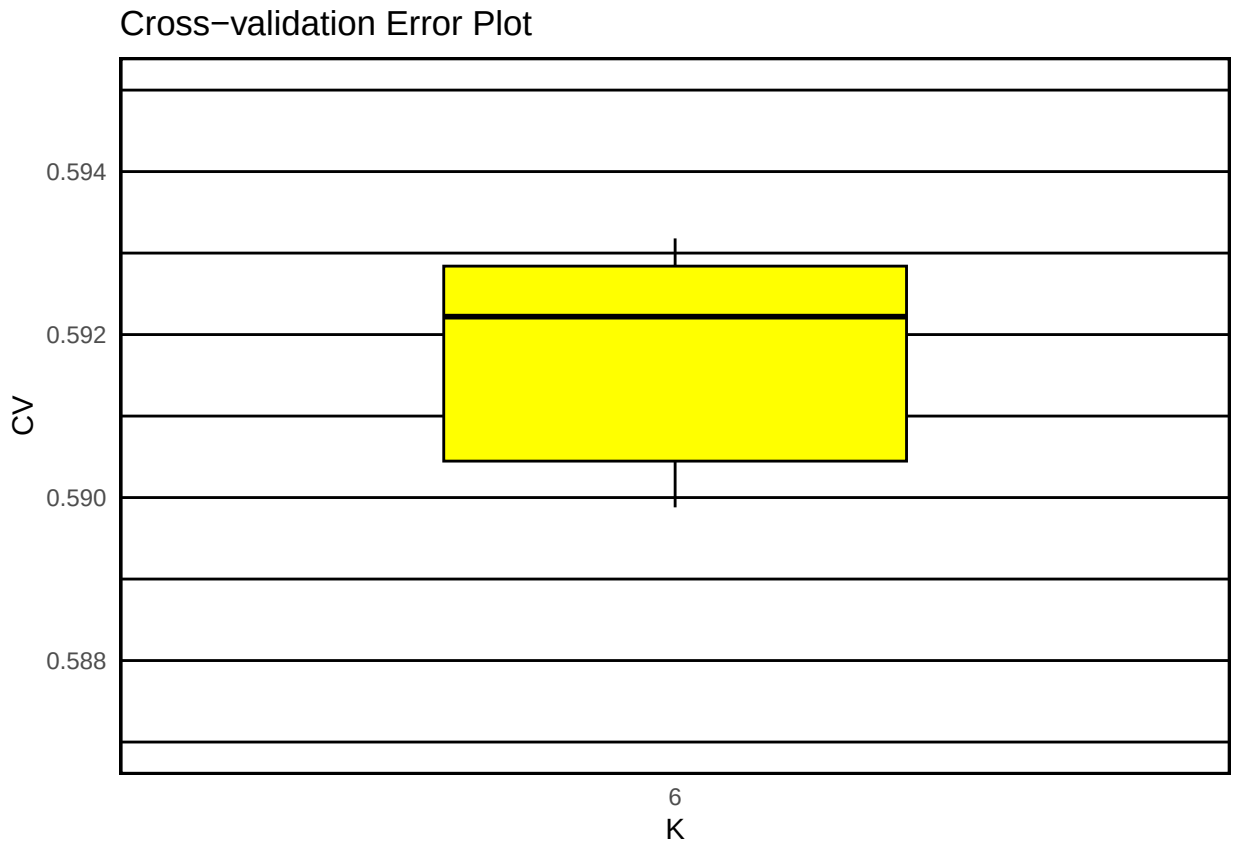
$\mathbf{K = 5}$

```
#jitter plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.59, 0.591, 0.592, 0.593,0.594,0.595,0.596,0.597,0.598,0.599),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
  geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.59, 0.599))
```




```r
setwd("~/Documents/Stage_NB/data/merged_data_1055_561_supervised")

# Étape 1: Créer une liste vide pour stocker les données
liste_de_donnees <- list()

# Étape 2: Parcourir les fichiers
for (i in 1:30) {
  # Générer le nom du fichier
  merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_561_filtered_MAF001_LD_default_K6_90_supervised_',
  #merge_cv_error <- paste0('merged_BeeMuSe_SeqApiPop_561_filtered_MAF001_LD_default_K5_95_supervised_'
  # Lire les données du fichier
  donnees <- read.table(merge_cv_error, header = FALSE)
  # Ajouter les données à la liste
  liste_de_donnees[[i]] <- donnees
}

# Étape 3: Combinez les données en une seule structure (par exemple, data frame)
donnees_combinees <- do.call(rbind, liste_de_donnees)

# Étape 4: Enregistrez ou affichez le résultat final sans numéro de lignes
write.table(donnees_combinees, "merge_cv_error", sep = "\t", col.names = FALSE, row.names = FALSE)

merge_cv_error <- read.table("merge_cv_error", header = F)
```

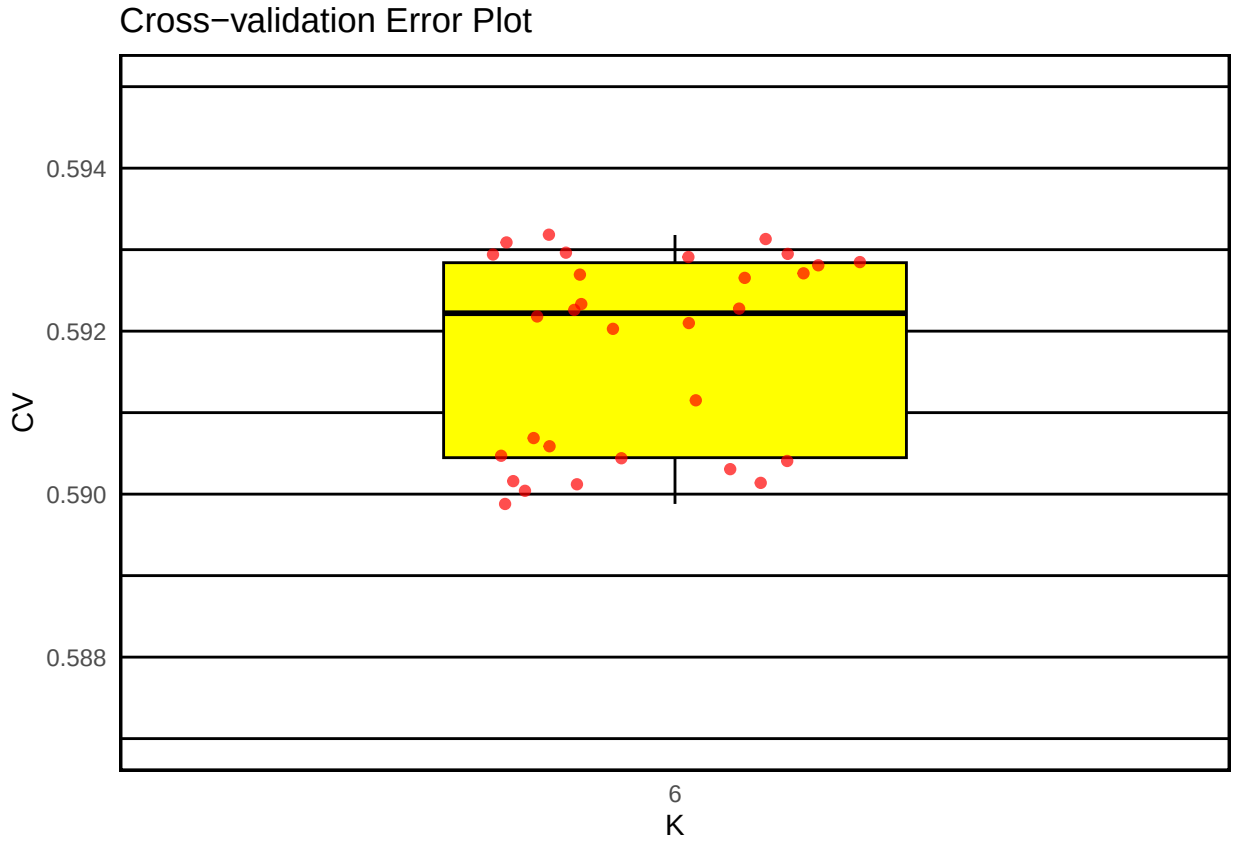
```r
#box plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.586,0.587,0.588,0.589,0.59, 0.591, 0.592, 0.593,0.594,0.595),
    color = "black",
    linetype = "solid",
    size = 0.5
  ) +
  geom_boxplot(width = 0.5, fill = "yellow", color = "black") +
  labs(title = "Cross-validation Error Plot",
       x = "K",
       y = "CV") +
  scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
  theme_minimal() +
  theme(
    panel.border = element_rect(color = "black", fill = NA, size = 1),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank()
  ) +
  coord_cartesian(ylim = c(0.587, 0.595))
```



K = 6

```r
#jitter plot LDO3 filtered
ggplot(merge_cv_error, aes(x = factor(merge_cv_error[,1]), y = merge_cv_error[,2])) +
  geom_hline(
    yintercept = c(0.587,0.588,0.589,0.59, 0.591, 0.592, 0.593,0.594,0.595),
```

```
  color = "black",
  linetype = "solid",
  size = 0.5
) +
geom_boxplot(width = 0.5, fill = "yellow", color = "black", outlier.shape = NA) +
geom_jitter(width = 0.2, alpha = 0.7, color = "red") +
labs(title = "Cross-validation Error Plot",
     x = "K",
     y = "CV") +
scale_y_continuous(labels = scales::number_format(accuracy = 0.001)) +
theme_minimal() +
theme(
  panel.border = element_rect(color = "black", fill = NA, size = 1),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank()
) +
coord_cartesian(ylim = c(0.587, 0.595))
```



Admixture supervisé - Création du fichier liste individu / population

561 échantillons - MAF > 0.01 - LD pruning = 0.3 (fenêtre de 1749 SNPs et pas de 149 bp) - 3848 SNPs

```r
setwd("~/Documents/Stage_NB/data/Qfiles/SeqApiPop_561_maf001_LD03")

Q_3_561 <- read.table("SeqApiPop_561_maf001_LD03_pruned.3.r10.Q", header = FALSE)

colnames(Q_3_561)[colnames(Q_3_561) == "V1"] <- "Vert"
colnames(Q_3_561)[colnames(Q_3_561) == "V2"] <- "Noir"
colnames(Q_3_561)[colnames(Q_3_561) == "V3"] <- "Orange"

# Create an empty vector to store the category for each row
categories <- character(nrow(Q_3_561))

# Initialisation du vecteur de catégories
categories <- rep("-", nrow(Q_3_561))

# Itérer à travers chaque ligne
for (i in 1:nrow(Q_3_561)) {
  # Vérifier si aucune valeur dans la ligne ne dépasse 0.9
  if (all(Q_3_561[i,] <= 0.9)) {
    categories[i] <- "-"
  } else {
    # Vérifier quelle colonne a la valeur supérieure à 0.9
    if (Q_3_561[i,1] > 0.9) {
      categories[i] <- "Vert"
    } else if (Q_3_561[i,2] > 0.9) {
      categories[i] <- "Noir"
    } else if (Q_3_561[i,3] > 0.9) {
      categories[i] <- "Orange"
    }
  }
}

# Write the categories to a single list file
write(categories, file = "output_list_K3_561_90.txt")
```

```r
setwd("~/Documents/Stage_NB/data/Qfiles")

output_list_K3_561_95_merged <- readLines("output_list_K3_561_95_merged.txt")
output_list_K3_561_95_merged <- gsub("Beemuse", "-", output_list_K3_561_95_merged)
writeLines(output_list_K3_561_95_merged, "merged_data_K3_561_95.pop")

output_list_K3_561_90_merged <- readLines("output_list_K3_561_90_merged.txt")
output_list_K3_561_90_merged <- gsub("Beemuse", "-", output_list_K3_561_90_merged)
writeLines(output_list_K3_561_90_merged, "merged_data_K3_561_90.pop")
```

```r
texte_complet <- paste(output_list_K3_561_95_merged, collapse = " ")
K3_95 <- unlist(strsplit(texte_complet, "\\s+"))
nombre_apparitions <- table(K3_95)
print(nombre_apparitions)
```

K = 3

```
## K3_95
##      -   Noir Orange   Vert
##   1081     68    143     17
```

```
texte_complet <- paste(output_list_K3_561_90_merged, collapse = " ")
K3_90 <- unlist(strsplit(texte_complet, "\\s+"))
nombre_apparitions <- table(K3_90)
print(nombre_apparitions)
```

```
## K3_90
##      -   Noir Orange   Vert
##   1033     84    175     17
```

```
setwd("~/Documents/Stage_NB/data/SeqApiPop_561_maf001_LD03")

labels <- read.csv('~/Documents/Stage_NB/data/SeqApiPop_labels.csv')
samples_561 <- read.table("SeqApiPop_561_maf001_LD03_pruned.fam", header = FALSE)

samples_561 <- samples_561[, 1:2] # Keep only the first two columns
colnames(samples_561)[colnames(samples_561) == "V1"] <- "name"

merged_labels_samples_561 <- merge(labels, samples_561,  by = 'name')

Label_samples_561 <- subset(merged_labels_samples_561, select = "Label")

writeLines(as.character(Label_samples_561$Label), "Label_samples_561.txt")
```

```
setwd("~/Documents/Stage_NB/data/Qfiles/SeqApiPop_561_maf001_LD03")

Q_5_561 <- read.table("SeqApiPop_561_maf001_LD03_pruned.5.r10.Q", header = FALSE)

colnames(Q_5_561)[colnames(Q_5_561) == "V1"] <- "Bleu"
colnames(Q_5_561)[colnames(Q_5_561) == "V2"] <- "Jaune"
colnames(Q_5_561)[colnames(Q_5_561) == "V3"] <- "Vert"
colnames(Q_5_561)[colnames(Q_5_561) == "V4"] <- "Orange"
colnames(Q_5_561)[colnames(Q_5_561) == "V5"] <- "Noir"

# Create an empty vector to store the category for each row
categories <- character(nrow(Q_5_561))

# Initialisation du vecteur de catégories
categories <- rep("-", nrow(Q_5_561))

# Itérer à travers chaque ligne
for (i in 1:nrow(Q_5_561)) {
  # Vérifier si aucune valeur dans la ligne ne dépasse 0.95
```

```
  if (all(Q_5_561[i,] <= 0.95)) {
    categories[i] <- "-"
  } else {
    # Vérifier quelle colonne a la valeur supérieure à 0.95
    if (Q_5_561[i,1] > 0.95) {
      categories[i] <- "Bleu"
    } else if (Q_5_561[i,2] > 0.95) {
      categories[i] <- "Jaune"
    } else if (Q_5_561[i,3] > 0.95) {
      categories[i] <- "Vert"
    } else if (Q_5_561[i,4] > 0.95) {
      categories[i] <- "Orange"
    } else if (Q_5_561[i,5] > 0.95) {
      categories[i] <- "Noir"
    }
  }
}

# Write the categories to a single list file
write(categories, file = "output_list_K5_561.txt")
```

```
setwd("~/Documents/Stage_NB/data/Qfiles")

# Read the file content
output_list_K5_561_merged <- readLines("output_list_K5_561_merged.txt")

# Replace "Beemuse" with "-"
output_list_K5_561_merged <- gsub("Beemuse", "-", output_list_K5_561_merged)

# Write the modified content back to the file
writeLines(output_list_K5_561_merged, "merged_data_K5_561.pop")
```

K = 5

```
setwd("~/Documents/Stage_NB/data/Qfiles/SeqApiPop_561_maf001_LD03")

Q_6_561 <- read.table("SeqApiPop_561_maf001_LD03_pruned.6.r13.Q", header = FALSE)

colnames(Q_6_561)[colnames(Q_6_561) == "V1"] <- "Rouge"
colnames(Q_6_561)[colnames(Q_6_561) == "V2"] <- "Bleu"
colnames(Q_6_561)[colnames(Q_6_561) == "V3"] <- "Orange"
colnames(Q_6_561)[colnames(Q_6_561) == "V4"] <- "Vert"
colnames(Q_6_561)[colnames(Q_6_561) == "V5"] <- "Jaune"
colnames(Q_6_561)[colnames(Q_6_561) == "V6"] <- "Noir"

# Create an empty vector to store the category for each row
categories <- character(nrow(Q_6_561))
```

```r
# Initialisation du vecteur de catégories
categories <- rep("-", nrow(Q_6_561))

# Itérer à travers chaque ligne
for (i in 1:nrow(Q_6_561)) {
  # Vérifier si aucune valeur dans la ligne ne dépasse 0.95
  if (all(Q_6_561[i,] <= 0.95)) {
    categories[i] <- "-"
  } else {
    # Vérifier quelle colonne a la valeur supérieure à 0.95
    if (Q_6_561[i,1] > 0.95) {
      categories[i] <- "Rouge"
    } else if (Q_6_561[i,2] > 0.95) {
      categories[i] <- "Bleu"
    } else if (Q_6_561[i,3] > 0.95) {
      categories[i] <- "Orange"
    } else if (Q_6_561[i,4] > 0.95) {
      categories[i] <- "Vert"
    } else if (Q_6_561[i,5] > 0.95) {
      categories[i] <- "Jaune"
    } else if (Q_6_561[i,6] > 0.95) {
      categories[i] <- "Noir"
    }
  }
}

# Write the categories to a single list file
write(categories, file = "output_list_K6_561.txt")
```

```r
setwd("~/Documents/Stage_NB/data/Qfiles")

# Read the file content
output_list_K6_561_merged <- readLines("output_list_K6_561_merged.txt")

# Replace "Beemuse" with "-"
output_list_K6_561_merged <- gsub("Beemuse", "-", output_list_K6_561_merged)

# Write the modified content back to the file
writeLines(output_list_K6_561_merged, "merged_data_K6_561.pop")
```

K = 6

561 échantillons - MAF > 0.01 - LD pruning = 0.1 (fenêtre de 50 SNPs et pas de 10 bp) - 1055 SNPs

```r
setwd("~/Documents/Stage_NB/data/Qfiles/SeqApiPop_561_LD03_default_1055")

Q_3_561_default <- read.table("SeqApiPop_561_SNPsBeeMuSe_filtered_maf001_LD03_default_pruned.3.r0.Q", h
```

```r
colnames(Q_3_561_default)[colnames(Q_3_561_default) == "V1"] <- "Noir"
colnames(Q_3_561_default)[colnames(Q_3_561_default) == "V2"] <- "Vert"
colnames(Q_3_561_default)[colnames(Q_3_561_default) == "V3"] <- "Orange"

# Create an empty vector to store the category for each row
categories <- character(nrow(Q_3_561_default))

# Initialisation du vecteur de catégories
categories <- rep("-", nrow(Q_3_561_default))

# Itérer à travers chaque ligne
for (i in 1:nrow(Q_3_561_default)) {
  # Vérifier si aucune valeur dans la ligne ne dépasse 0.9
  if (all(Q_3_561_default[i,] <= 0.9)) {
    categories[i] <- "-"
  } else {
    # Vérifier quelle colonne a la valeur supérieure à 0.9
    if (Q_3_561_default[i,1] > 0.9) {
      categories[i] <- "Noir"
    } else if (Q_3_561_default[i,2] > 0.9) {
      categories[i] <- "Vert"
    } else if (Q_3_561_default[i,3] > 0.9) {
      categories[i] <- "Orange"
    }
  }
}

# Write the categories to a single list file
write(categories, file = "output_list_K3_561_LD_default_90.txt")
```

```r
setwd("~/Documents/Stage_NB/data/Qfiles")

output_list_K3_561_default_95_merged <- readLines("output_list_K3_561_LD_default_95_merged.txt")
output_list_K3_561_default_95_merged <- gsub("Beemuse", "-", output_list_K3_561_default_95_merged)
writeLines(output_list_K3_561_default_95_merged, "merged_data_K3_561_LD_default_95.pop")

output_list_K3_561_default_90_merged <- readLines("output_list_K3_561_LD_default_90_merged.txt")
output_list_K3_561_default_90_merged <- gsub("Beemuse", "-", output_list_K3_561_default_90_merged)
writeLines(output_list_K3_561_default_90_merged, "merged_data_K3_561_LD_default_90.pop")
```

```r
texte_complet <- paste(output_list_K3_561_default_95_merged, collapse = " ")
K3_95 <- unlist(strsplit(texte_complet, "\\s+"))
nombre_apparitions <- table(K3_95)
print(nombre_apparitions)
```

K = 3

```
## K3_95
##      -   Noir Orange   Vert
##   1176     55     63     15
```

```r
texte_complet <- paste(output_list_K3_561_default_90_merged, collapse = " ")
K3_90 <- unlist(strsplit(texte_complet, "\\s+"))
nombre_apparitions <- table(K3_90)
print(nombre_apparitions)
```

```
## K3_90
##       -   Noir Orange   Vert
##    1096     77    119     17
```

```r
setwd("~/Documents/Stage_NB/data/Qfiles/SeqApiPop_561_LD03_default_1055")

Q_5_561_default <- read.table("SeqApiPop_561_SNPsBeeMuSe_filtered_maf001_LD03_default_pruned.5.r22.Q", h

colnames(Q_5_561_default)[colnames(Q_5_561_default) == "V1"] <- "Rouge"
colnames(Q_5_561_default)[colnames(Q_5_561_default) == "V2"] <- "Vert"
colnames(Q_5_561_default)[colnames(Q_5_561_default) == "V3"] <- "Noir"
colnames(Q_5_561_default)[colnames(Q_5_561_default) == "V4"] <- "Orange"
colnames(Q_5_561_default)[colnames(Q_5_561_default) == "V5"] <- "Jaune"

categories <- character(nrow(Q_5_561_default))
categories <- rep("-", nrow(Q_5_561_default))

for (i in 1:nrow(Q_5_561_default)) {
  # Vérifier si aucune valeur dans la ligne ne dépasse 0.95
  if (all(Q_5_561_default[i,] <= 0.95)) {
    categories[i] <- "-"
  } else {
    # Vérifier quelle colonne a la valeur supérieure à 0.95
    if (Q_5_561_default[i,1] > 0.95) {
      categories[i] <- "Rouge"
    } else if (Q_5_561_default[i,2] > 0.95) {
      categories[i] <- "Vert"
    } else if (Q_5_561_default[i,3] > 0.95) {
      categories[i] <- "Noir"
    } else if (Q_5_561_default[i,4] > 0.95) {
      categories[i] <- "Orange"
    } else if (Q_5_561_default[i,5] > 0.95) {
      categories[i] <- "Jaune"
    }
  }
}

write(categories, file = "output_list_K5_561_LD_default_95.txt")
```

```r
setwd("~/Documents/Stage_NB/data/Qfiles")

output_list_K5_561_default_95_merged <- readLines("output_list_K5_561_LD_default_95_merged.txt")
output_list_K5_561_default_95_merged <- gsub("Beemuse", "-", output_list_K5_561_default_95_merged)
```

```
writeLines(output_list_K5_561_default_95_merged, "merged_data_K5_561_LD_default_95.pop")

output_list_K5_561_default_90_merged <- readLines("output_list_K5_561_LD_default_90_merged.txt")
output_list_K5_561_default_90_merged <- gsub("Beemuse", "-", output_list_K5_561_default_90_merged)
writeLines(output_list_K5_561_default_90_merged, "merged_data_K5_561_LD_default_90.pop")
```

```
texte_complet <- paste(output_list_K5_561_default_95_merged, collapse = " ")
K5_95 <- unlist(strsplit(texte_complet, "\\s+"))
nombre_apparitions <- table(K5_95)
print(nombre_apparitions)
```

$\mathbf{K = 5}$

```
## K5_95
##      -  Jaune   Noir Orange  Rouge   Vert
##   1178     21     49     24     22     15
```

```
texte_complet <- paste(output_list_K5_561_default_90_merged, collapse = " ")
K5_90 <- unlist(strsplit(texte_complet, "\\s+"))
nombre_apparitions <- table(K5_90)
print(nombre_apparitions)
```

```
## K5_90
##      -  Jaune   Noir Orange  Rouge   Vert
##   1121     23     74     47     27     17
```

```
setwd("~/Documents/Stage_NB/data/Qfiles/SeqApiPop_561_LD03_default_1055")

Q_6_561_default <- read.table("SeqApiPop_561_SNPsBeeMuSe_filtered_maf001_LD03_default_pruned.6.r23.Q", h

colnames(Q_6_561_default)[colnames(Q_6_561_default) == "V1"] <- "Bleu"
colnames(Q_6_561_default)[colnames(Q_6_561_default) == "V2"] <- "Rouge"
colnames(Q_6_561_default)[colnames(Q_6_561_default) == "V3"] <- "Noir"
colnames(Q_6_561_default)[colnames(Q_6_561_default) == "V4"] <- "Jaune"
colnames(Q_6_561_default)[colnames(Q_6_561_default) == "V5"] <- "Orange"
colnames(Q_6_561_default)[colnames(Q_6_561_default) == "V6"] <- "Vert"

categories <- character(nrow(Q_6_561_default))
categories <- rep("-", nrow(Q_6_561_default))

for (i in 1:nrow(Q_6_561_default)) {
  # Vérifier si aucune valeur dans la ligne ne dépasse 0.9
  if (all(Q_6_561_default[i,] <= 0.9)) {
    categories[i] <- "-"
  } else {
    # Vérifier quelle colonne a la valeur supérieure à 0.9
    if (Q_6_561_default[i,1] > 0.9) {
      categories[i] <- "Bleu"
```

```
    } else if (Q_6_561_default[i,2] > 0.9) {
      categories[i] <- "Rouge"
    } else if (Q_6_561_default[i,3] > 0.9) {
      categories[i] <- "Noir"
    } else if (Q_6_561_default[i,4] > 0.9) {
      categories[i] <- "Jaune"
    } else if (Q_6_561_default[i,5] > 0.9) {
      categories[i] <- "Orange"
    } else if (Q_6_561_default[i,6] > 0.9) {
      categories[i] <- "Vert"
    }
  }
}

write(categories, file = "output_list_K6_561_LD_default_90.txt")
```

```
setwd("~/Documents/Stage_NB/data/Qfiles")

output_list_K6_561_default_95_merged <- readLines("output_list_K6_561_LD_default_95_merged.txt")
output_list_K6_561_default_95_merged <- gsub("Beemuse", "-", output_list_K6_561_default_95_merged)
writeLines(output_list_K6_561_default_95_merged, "merged_data_K6_561_LD_default_95.pop")

output_list_K6_561_default_90_merged <- readLines("output_list_K6_561_LD_default_90_merged.txt")
output_list_K6_561_default_90_merged <- gsub("Beemuse", "-", output_list_K6_561_default_90_merged)
writeLines(output_list_K6_561_default_90_merged, "merged_data_K6_561_LD_default_90.pop")
```

```
texte_complet <- paste(output_list_K6_561_default_95_merged, collapse = " ")
K6_95 <- unlist(strsplit(texte_complet, "\\s+"))
nombre_apparitions <- table(K6_95)
print(nombre_apparitions)
```

$\mathbf{K = 6}$

```
## K6_95
##      -   Bleu  Jaune   Noir Orange  Rouge   Vert
##   1211     12     18     11     20     22     15
```

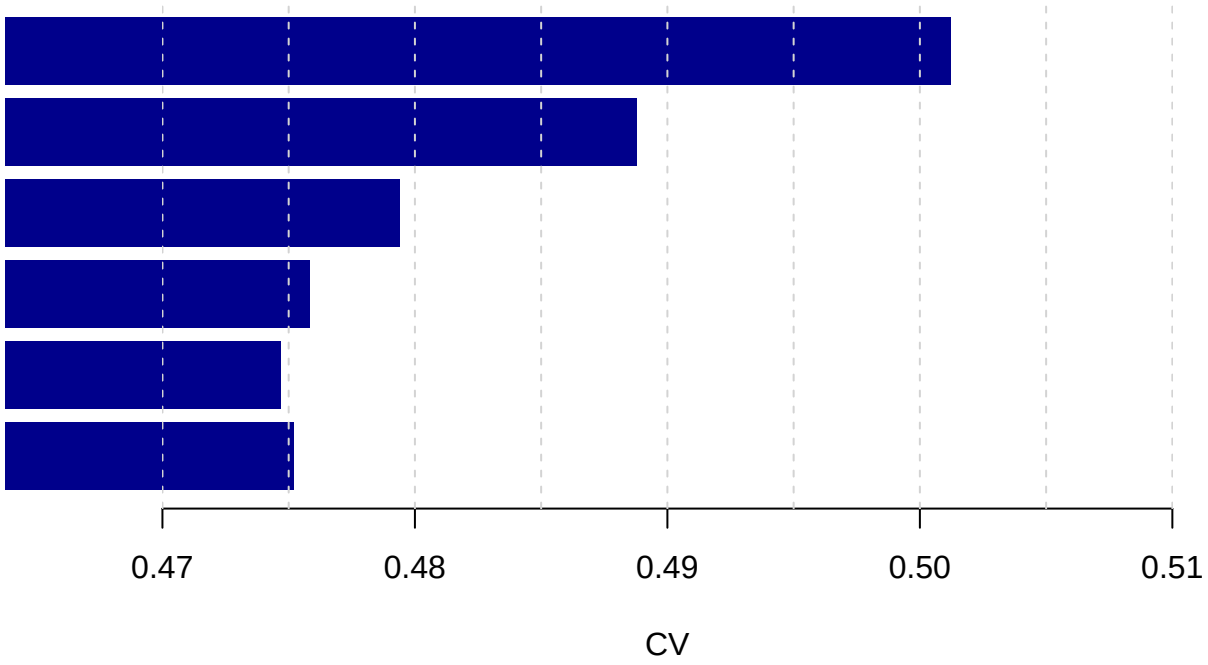
```
texte_complet <- paste(output_list_K6_561_default_90_merged, collapse = " ")
K6_90 <- unlist(strsplit(texte_complet, "\\s+"))
nombre_apparitions <- table(K6_90)
print(nombre_apparitions)
```

```
## K6_90
##      -   Bleu  Jaune   Noir Orange  Rouge   Vert
##   1166     14     21     24     42     26     16
```

Légende - CV plot error - Admixture

SeqApiPop 629 échantillons - MAF > 0.01 - LD pruning = 0.3 (fenêtre de 1749 SNPs et pas de 175 bp)

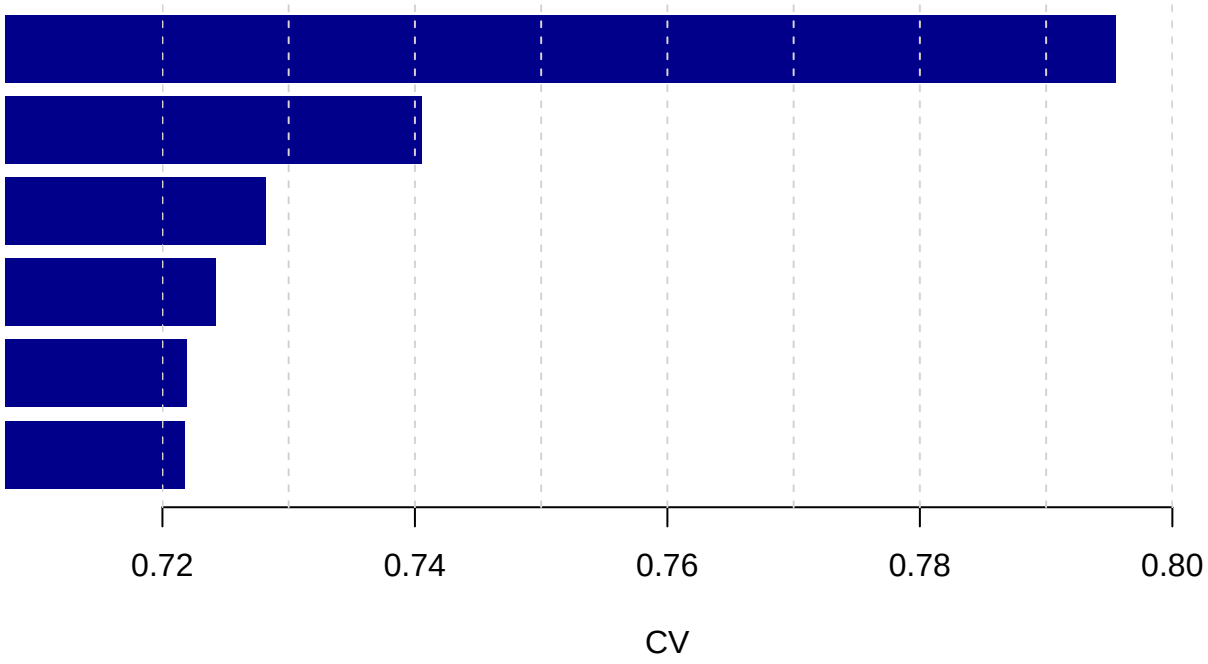
```
# Définir les valeurs et les noms des groupes
valeurs <- c(0.475215, 0.4747, 0.47582, 0.4794, 0.4888, 0.5012)

# Créer le barplot avec des barres plus espacées et moins larges
bp <- barplot(valeurs, horiz = TRUE, xlim = c(0.47, 0.51), width = 0.1,
        xlab = "CV", las = 1, col = "darkblue", border = NA)

# Ajouter un cadrillage
abline(v = seq(0.47, 0.51, by = 0.005), col = "lightgray", lty = 2)
```



SeqApiPop 629 échantillons - SNPsBeeMuSe filtered - 10030 SNPS

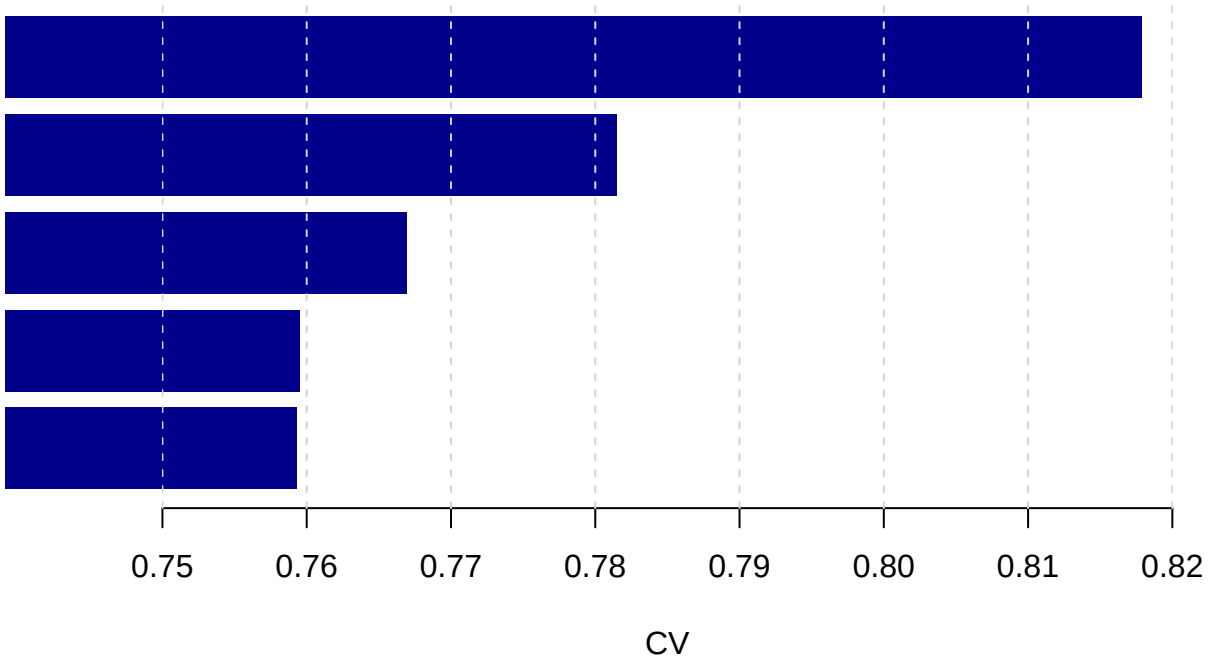
```
# Définir les valeurs et les noms des groupes
valeurs <- c(0.7218, 0.7219, 0.7242, 0.7282, 0.7405, 0.7955)

# Créer le barplot avec des barres plus espacées et moins larges
bp <- barplot(valeurs, horiz = TRUE, xlim = c(0.72, 0.8), width = 0.1,
        xlab = "CV", las = 1, col = "darkblue", border = NA)
```

```r
# Ajouter un cadrillage
abline(v = seq(0.72, 0.8, by = 0.01), col = "lightgray", lty = 2)
```



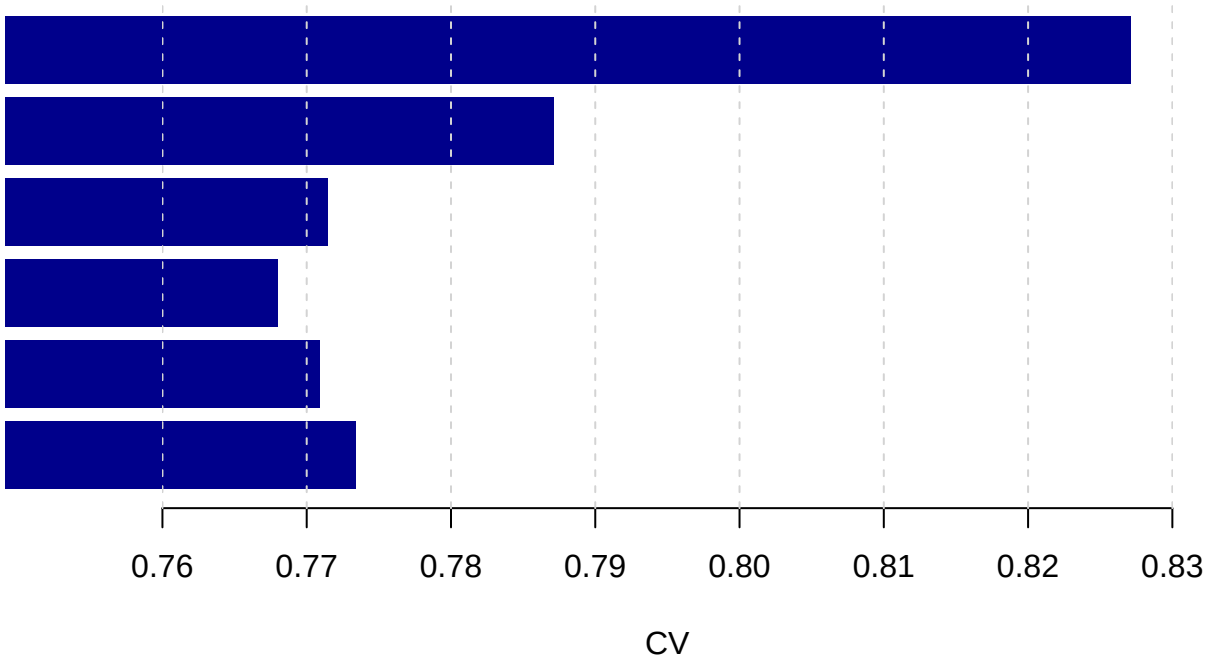
SeqApiPop 629 échantillons - SNPsBeeMuSe filtered - MAF > 0.01 - LD pruning = 0.1 (fenêtre de 50 SNPs et pas de 10 bp) - 1055 SNPs

```r
# Définir les valeurs et les noms des groupes
valeurs <- c(0.7593, 0.7595, 0.7669, 0.7815, 0.8179)

# Créer le barplot avec des barres plus espacées et moins larges
bp <- barplot(valeurs, horiz = TRUE, xlim = c(0.75, 0.82), width = 0.1,
        xlab = "CV", las = 1, col = "darkblue", border = NA)

# Ajouter un cadrillage
abline(v = seq(0.75, 0.82, by = 0.01), col = "lightgray", lty = 2)
```



SeqApiPop 561 échantillons - SNPsBeeMuSe filtered - MAF > 0.01 - LD pruning = 0.1 (fenêtre de 50 SNPs et pas de 10 bp) - 1055 SNPS

```
# Définir les valeurs et les noms des groupes
valeurs <- c(0.7734 ,0.7709, 0.7680, 0.7715, 0.7871, 0.8271)

# Créer le barplot avec des barres plus espacées et moins larges
bp <- barplot(valeurs, horiz = TRUE, xlim = c(0.76, 0.83), width = 0.1,
        xlab = "CV", las = 1, col = "darkblue", border = NA)

# Ajouter un cadrillage
abline(v = seq(0.76, 0.83, by = 0.01), col = "lightgray", lty = 2)
```


0.76
0.77
0.78
0.79
0.80
0.81
0.82
0.83
CV