

# Analyse des résultats de PSE

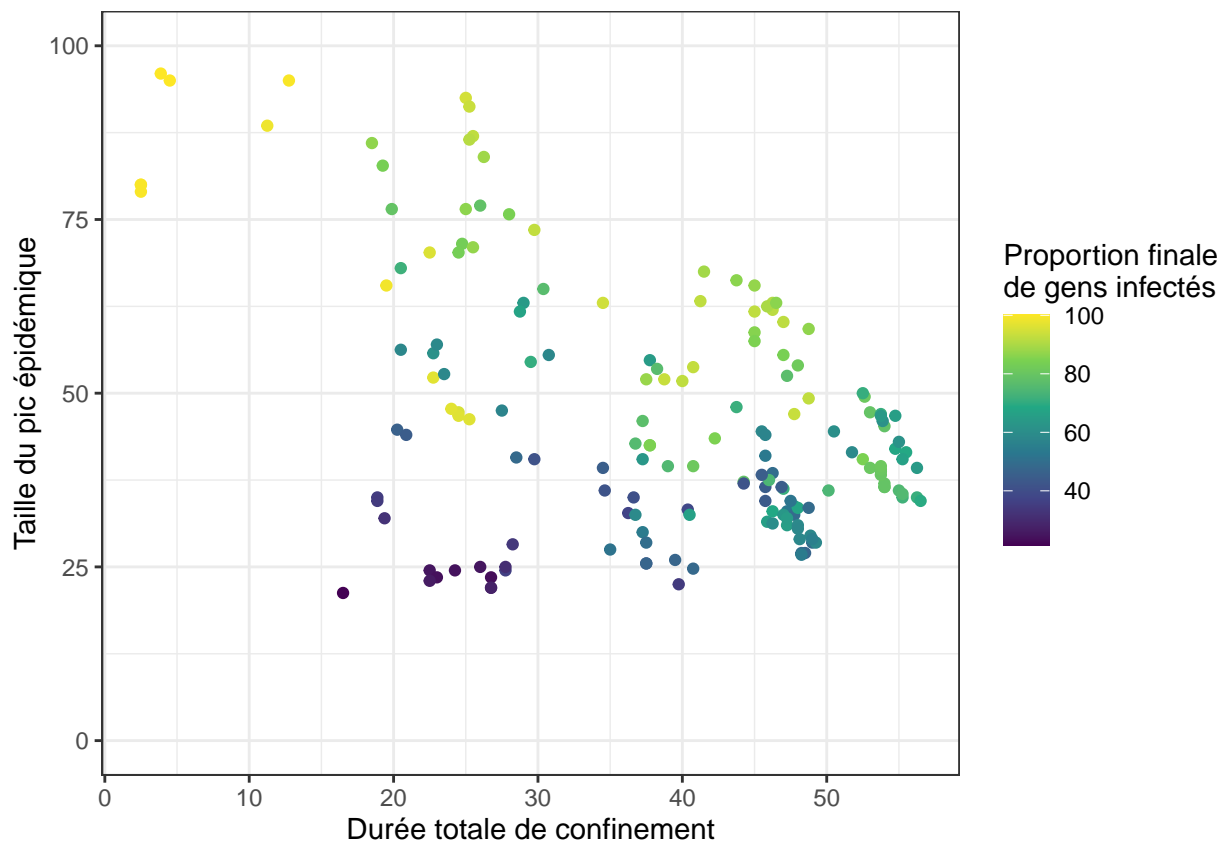
Hélène Arduin & Etienne Delay

20/06/2020

## Données globales de PSE

3 graphes pour les 3 possibilités de représentation (les points sont les médianes) :

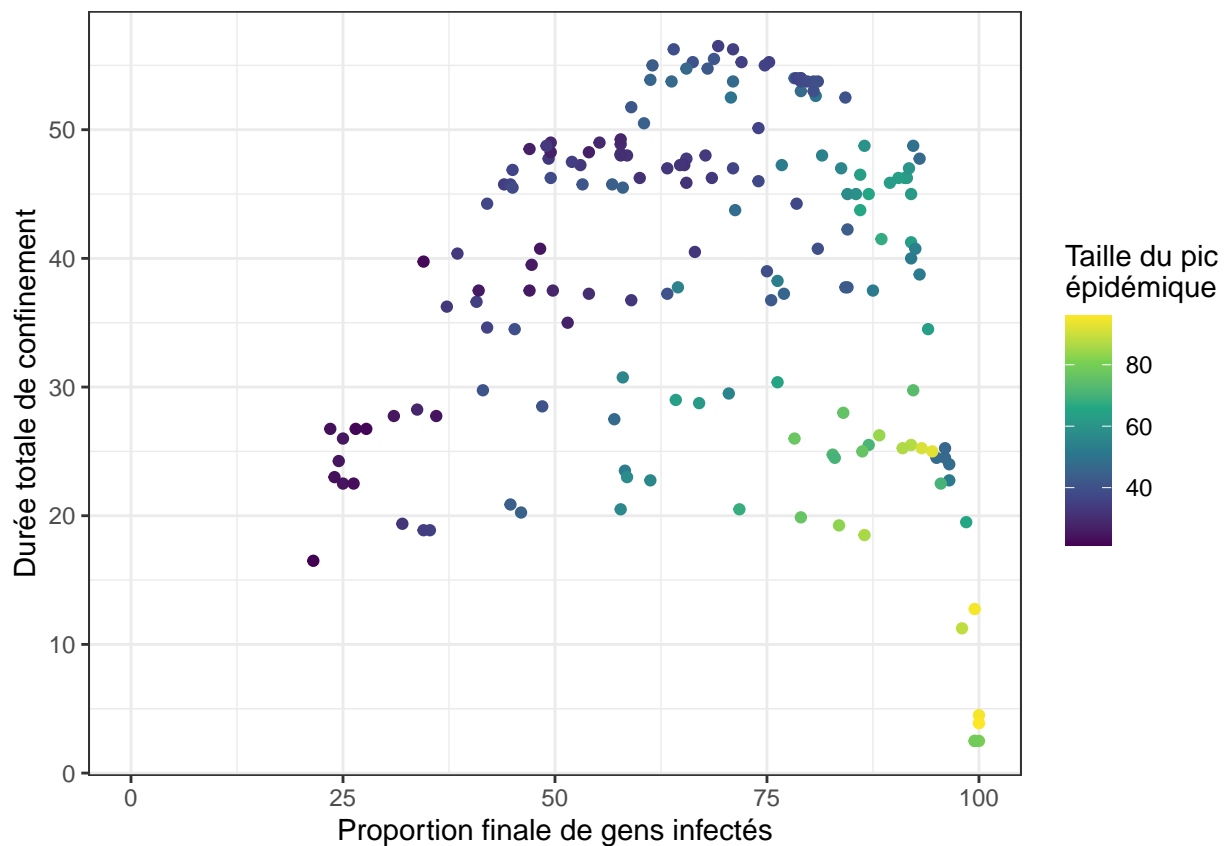
```
plot_pse1 <- data %>%  
  group_by(index) %>%  
  summarise_at(vars(propR:totalLockdownDuration), list(median)) %>%  
  ggplot(aes(x = totalLockdownDuration, y = peakHeight, color = propR)) +  
  geom_point() +  
  scale_y_continuous(breaks = seq(0, 100, 25), limits = c(0, 100)) +  
  scale_color_viridis_c(name = "Proportion finale\nde gens infectés") +  
  labs(x = "Durée totale de confinement",  
       y = "Taille du pic épidémique") +  
  theme_bw()  
plot_pse1
```



```

plot_pse2 <- data %>%
  group_by(index) %>%
  summarise_at(vars(propR:totalLockdownDuration), list(median)) %>%
  ggplot(aes(x = propR, y = totalLockdownDuration, color = peakHeight)) +
  geom_point() +
  scale_x_continuous(breaks = seq(0, 100, 25), limits = c(0, 100)) +
  scale_color_viridis_c(name = "Taille du pic\népidémique") +
  labs(x = "Proportion finale de gens infectés",
       y = "Durée totale de confinement") +
  theme_bw()
plot_pse2

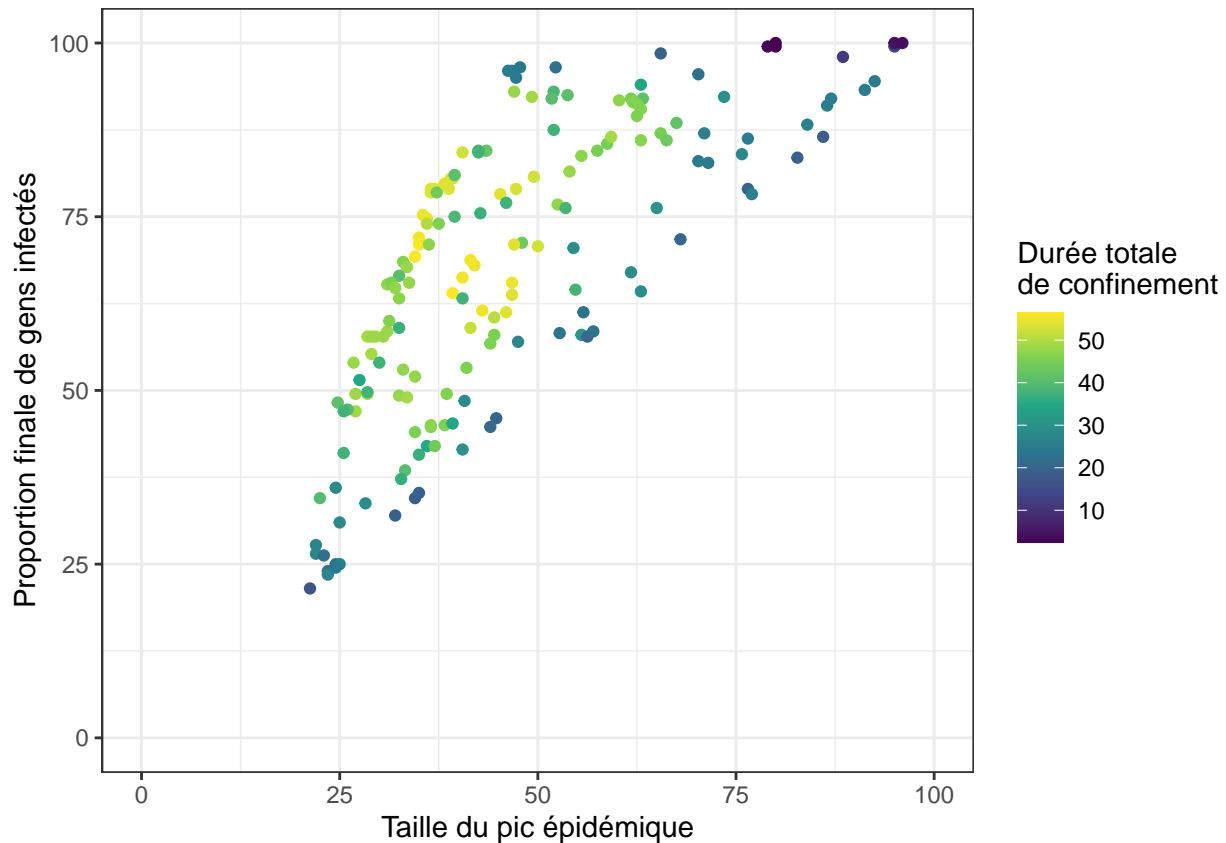
```



```

plot_pse3 <- data %>%
  group_by(index) %>%
  summarise_at(vars(propR:totalLockdownDuration), list(median)) %>%
  ggplot(aes(x = peakHeight, y = propR, color = totalLockdownDuration)) +
  geom_point() +
  scale_x_continuous(breaks = seq(0, 100, 25), limits = c(0, 100)) +
  scale_y_continuous(breaks = seq(0, 100, 25), limits = c(0, 100)) +
  scale_color_viridis_c(name = "Durée totale\nde confinement") +
  labs(x = "Taille du pic épidémique",
       y = "Proportion finale de gens infectés") +
  theme_bw()
plot_pse3

```



## Sélection des “meilleurs” résultats

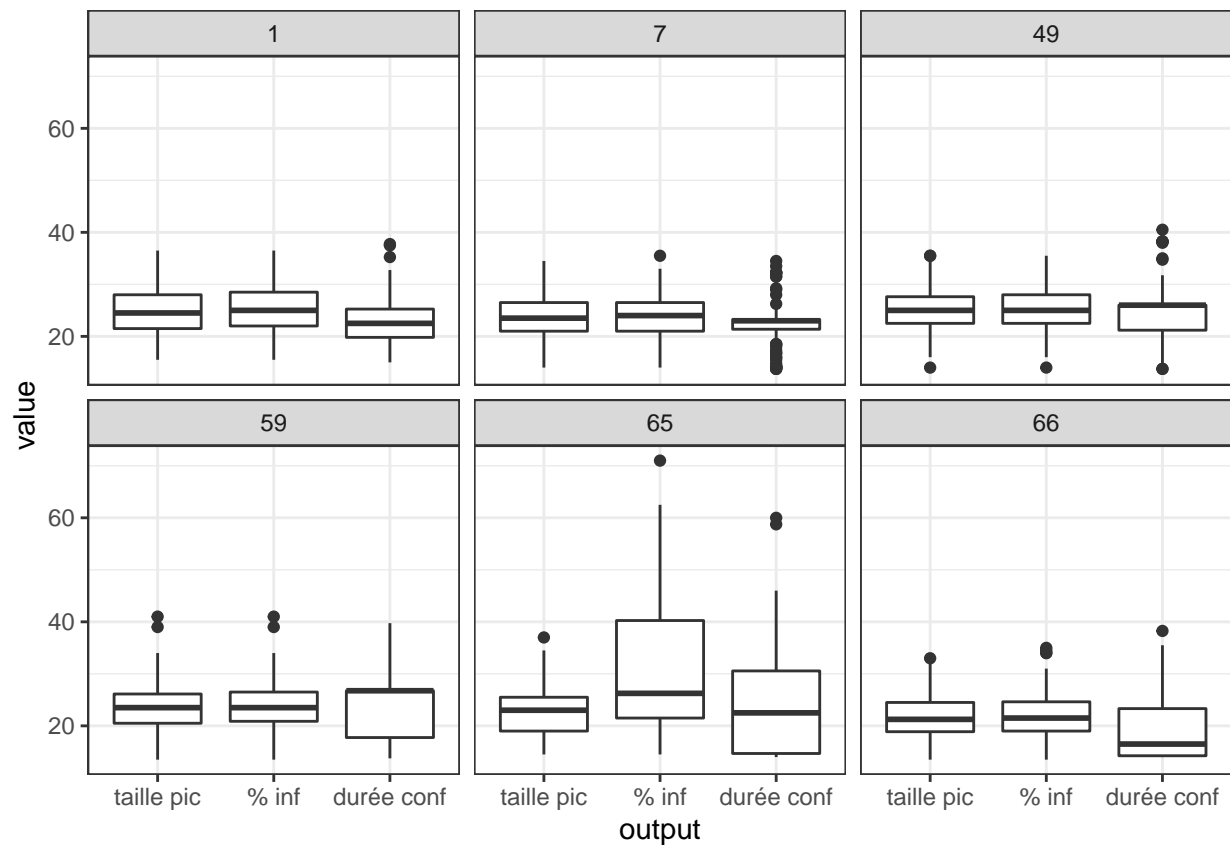
Taille du pic < 25% ; Proportion de gens guéris < 40% ; Durée totale du confinement < 25 jours

Il y a 6 points sélectionnés via les médianes, on va dire 6 scénarios, là je reprends les distributions complètes correspondant à chaque scénario.

```
## Joining, by = "index"
```

## Distributions des 3 variables de sortie pour chaque scénario

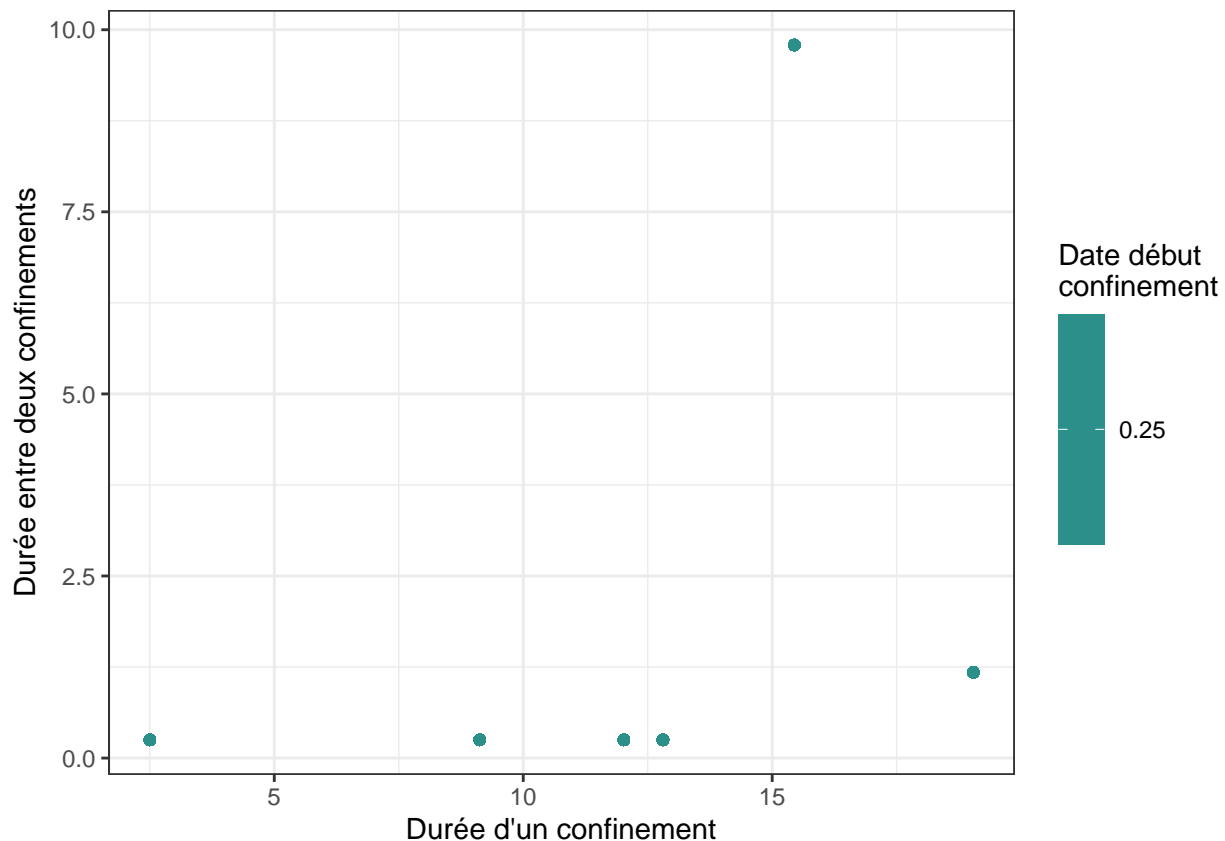
```
plot_distrib <- best_area %>%
  pivot_longer(cols = propR:totalLockdownDuration, names_to = "output", values_to = "value") %>%
  # ggplot(aes(x = output, y = value, color = lockdownPauseDuration)) +
  ggplot(aes(x = output, y = value)) +
  geom_boxplot(aes(group = output)) +
  scale_x_discrete(labels = c("taille pic", "% inf", "durée conf")) +
  # facet_grid(lockdownStartingDate ~ lockdownDuration) +
  facet_wrap(~ index) +
  theme_bw()
plot_distrib
```



Dans la plupart des cas les distributions sont relativement ramassées, sauf pour la figure en bas au milieu.

### Paramètres d'entrée qui correspondent aux scénarios sélectionnés

```
plot_input_space <- best_area %>%
  ggplot(aes(x = lockdownDuration, y = lockdownPauseDuration, color = lockdownStartingDate)) +
  geom_point() +
  scale_color_viridis_c(name = "Date début\nconfinement") +
  labs(x = "Durée d'un confinement",
       y = "Durée entre deux confinements") +
  theme_bw()
plot_input_space
```



Ici il y a une seule date de début du confinement sélectionnée (dès le deuxième pas de temps). Tous les points sauf un ont des confinements qui s'enchaînent très vite (peut-être nécessaire de regarder le nombre de confinements associés à ces points-là).

## Sélection de quelques focus

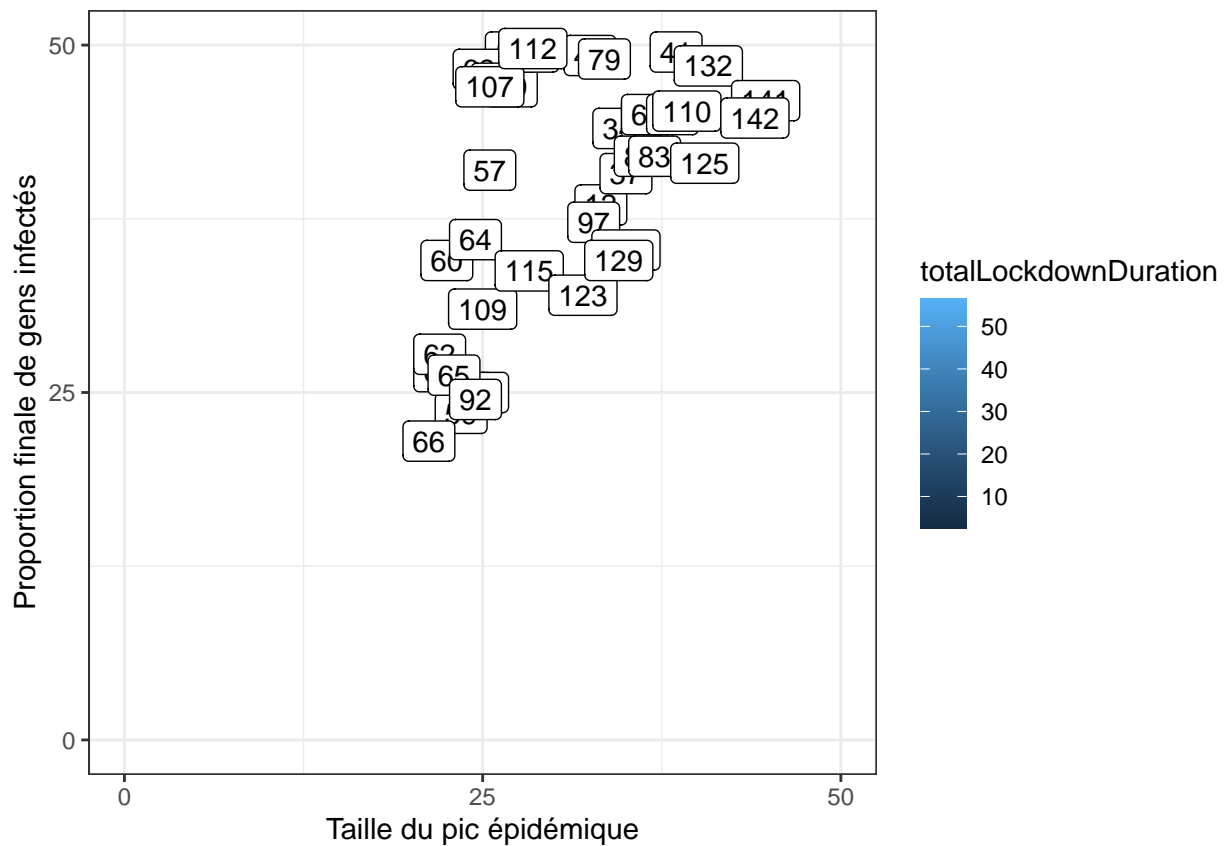
On va se focaliser sur le points 66 qui a un nombre de jours de confinement de 16.5 et minimise les deux autres paramètres

```
all <- data %>%
  group_by(index) %>%
  summarise_at(vars(propR:totalLockdownDuration), list(median))

plot_foc1 <- ggplot() +
  geom_point(data = all, aes(x = peakHeight, y = propR, color = totalLockdownDuration)) +
  geom_label(data = all, aes(x = peakHeight, y = propR, label = index)) +
  # geom_density_2d(data = best_area, aes(x = peakHeight, y = propR), size=0.25, bins = 100) +
  # geom_point(data = best_area, aes(x = peakHeight, y = propR), alpha = 0.2, shape=3) +
  scale_x_continuous(breaks = seq(0, 100, 25), limits = c(0, 50)) +
  scale_y_continuous(breaks = seq(0, 100, 25), limits = c(0, 50)) +
  # scale_color_viridis_c(name = "Durée totale\nde confinement") +
  labs(x = "Taille du pic épidémique",
       y = "Proportion finale de gens infectés") +
  theme_bw()
plot_foc1
```

## Warning: Removed 132 rows containing missing values (geom\_point).

```
## Warning: Removed 132 rows containing missing values (geom_label).
```



```
sel <- best_area$index == 66
plot_foc2 <- ggplot() +
  geom_point(data = all, aes(x = peakHeight, y = propR, color = totalLockdownDuration)) +
  geom_density_2d(data = best_area[sel,], aes(x = peakHeight, y = propR), size=0.25, bins = 10) +
  geom_point(data = best_area[sel,], aes(x = peakHeight, y = propR), alpha = 0.2, shape=3) +
  scale_x_continuous(breaks = seq(0, 100, 25), limits = c(0, 50)) +
  scale_y_continuous(breaks = seq(0, 100, 25), limits = c(0, 50)) +
  scale_color_viridis_c(name = "Durée totale\nde confinement") +
  labs(x = "Taille du pic épidémique",
       y = "Proportion finale de gens infectés") +
  theme_bw()
plot_foc2
```

```
## Warning: Removed 132 rows containing missing values (geom_point).
```

