lab4

2024-06-04

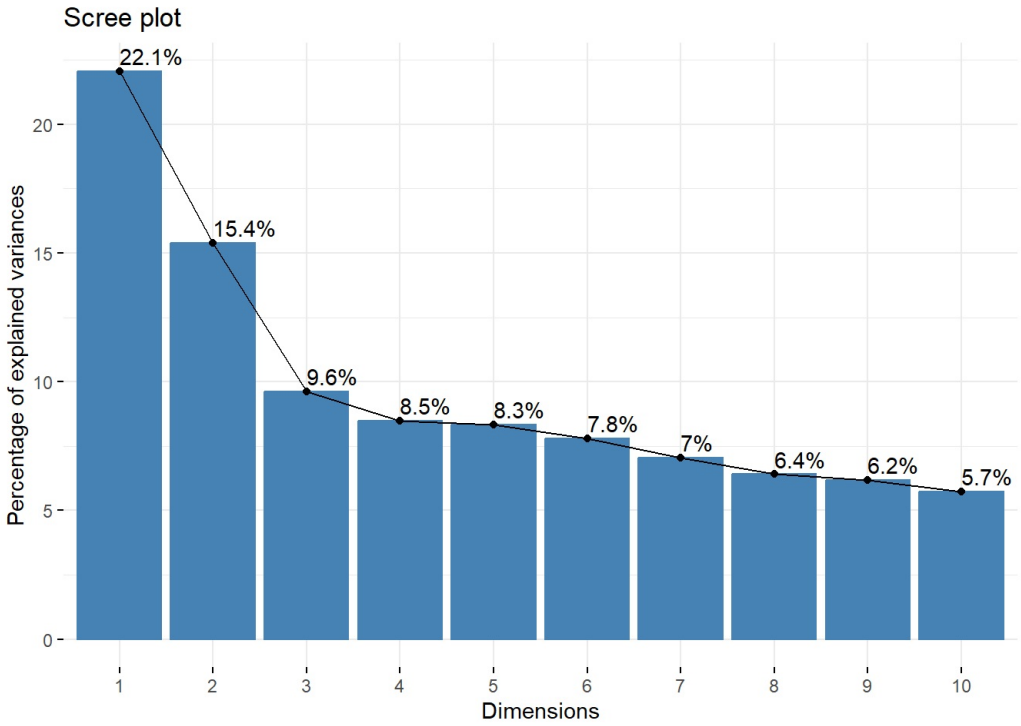
Назва команди - Команда №3

Перелік учасників колективу виконавців:

- Пономаренко Олександр (КМ-12)
- Земляний Даниїл (КМ-12)
- Борисенко Данило (КМ-11)
- Заіченко Дамир (КМ-13)
- Лук'яненко Василь (КМ-13)

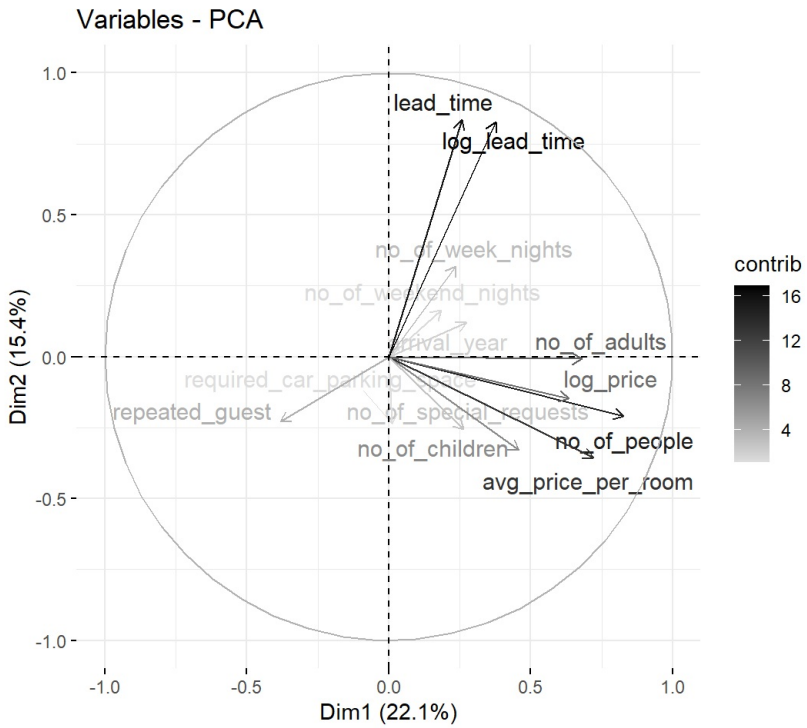
Проведемо PCA і подивимось на відповідний screeplot (графік власних чисел)

```
hotel_pca.pca <- PCA(hotel_for_pca, graph = FALSE)

fviz_screeplot(hotel_pca.pca, addlabels = TRUE)
```



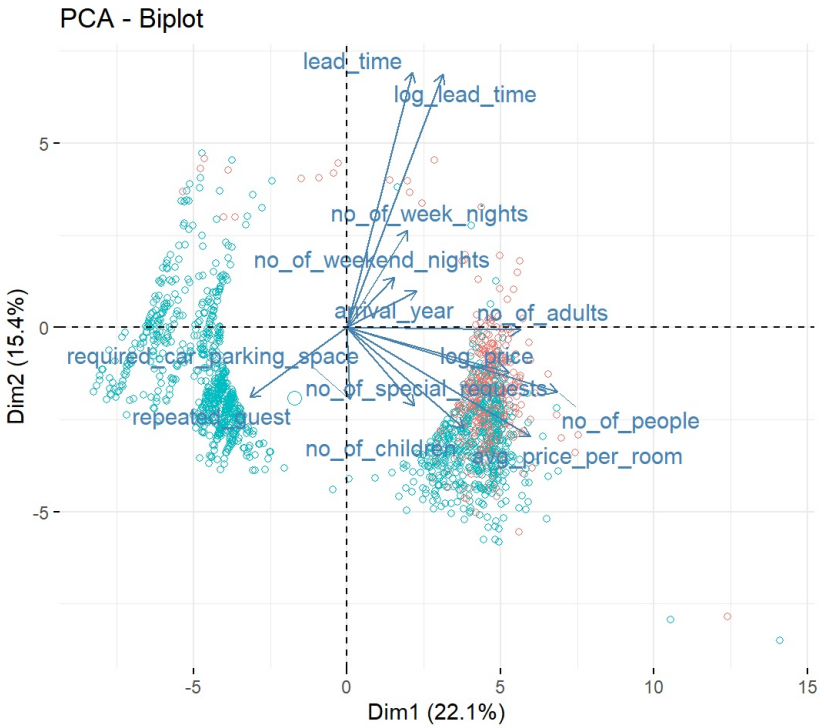
Бачимо, що як у прикладі з титнаіком у лекції 11, наші результати теж виявились не дуже хорошими. Метод головних компонент зменшив нам кількість компонент з 16 до 8, які разом нам описують 85.1% дисперсії. Це не дуже гарний результат. Спроєктуємо тепер змінні на перші дві компоненти, позначимо градієнтом кольорів внесок різних змінних до загальної дисперсії

```
fviz_pca_var(hotel_pca.pca, axes = c(1, 2), repel = TRUE, col.var="contrib",
             gradient.cols = c( "#DDDDDD","#000000"))
```



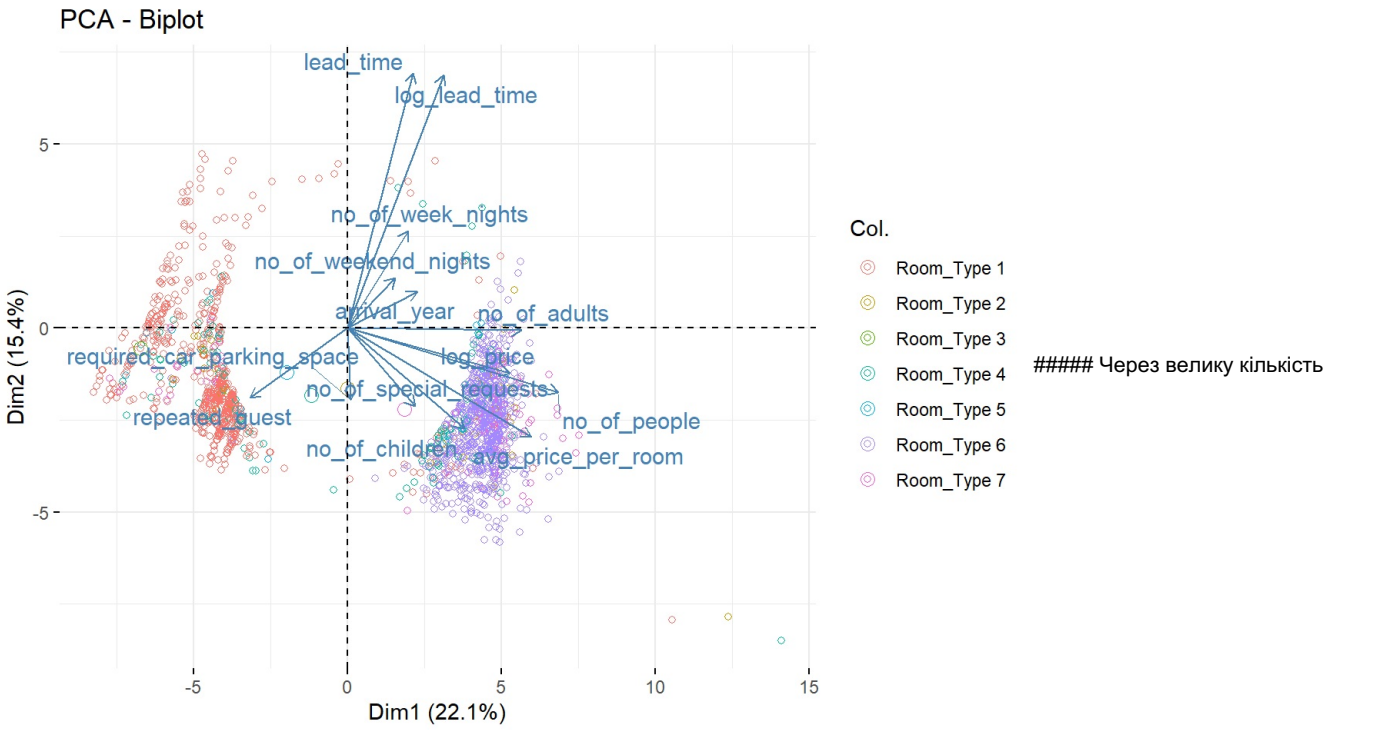
Через те, що у нас назви змінних довгі, графіки сприймаються складно, але все ж можна розгледіти певні залежності. Можна сказати, що першій компоненті із додатним знаком переважно відповідають люди, які платять більше (а також частково: з більшою кількістю людей, на більшу кількість ночей, заселяються раніше; більш загально: приїжджають на відпочинок), із від'ємним знаком - ті хто платять менше, що є однією з основних ознак повторного гостя, як ми визначили за результатами минулих лабораторних робіт. ##### Розглянемо це більше детально, за допомогою різних біграфіків (biplot). Побудуємо біграфік по цим компонентам, вказавши критерій відбору дослідження - booking_status (статус скасування замовлення), а також вказавши, що ми наносимо тільки перші 2000 "найвпливовіших" спостережнь.

```
fviz_pca_biplot(hotel_pca.pca, axes = c(1, 2), geom = "point",
                select.ind = list(contrib = 2000), pointshape = 1,
                col.ind = factor(hotel_corr$booking_status), repel = TRUE)
```



За цим графіком можна побачити, що значна частина людей з додатним знаком по першій компоненті відмінила запис до готелю, в той час як люди з від'ємної сторони цієї компоненти - майже не відміняли взагалі. Це досить цікавий результат. Подивимось тепер такий самий графік, але позначивши замість статусу скасування - тип кімнати.

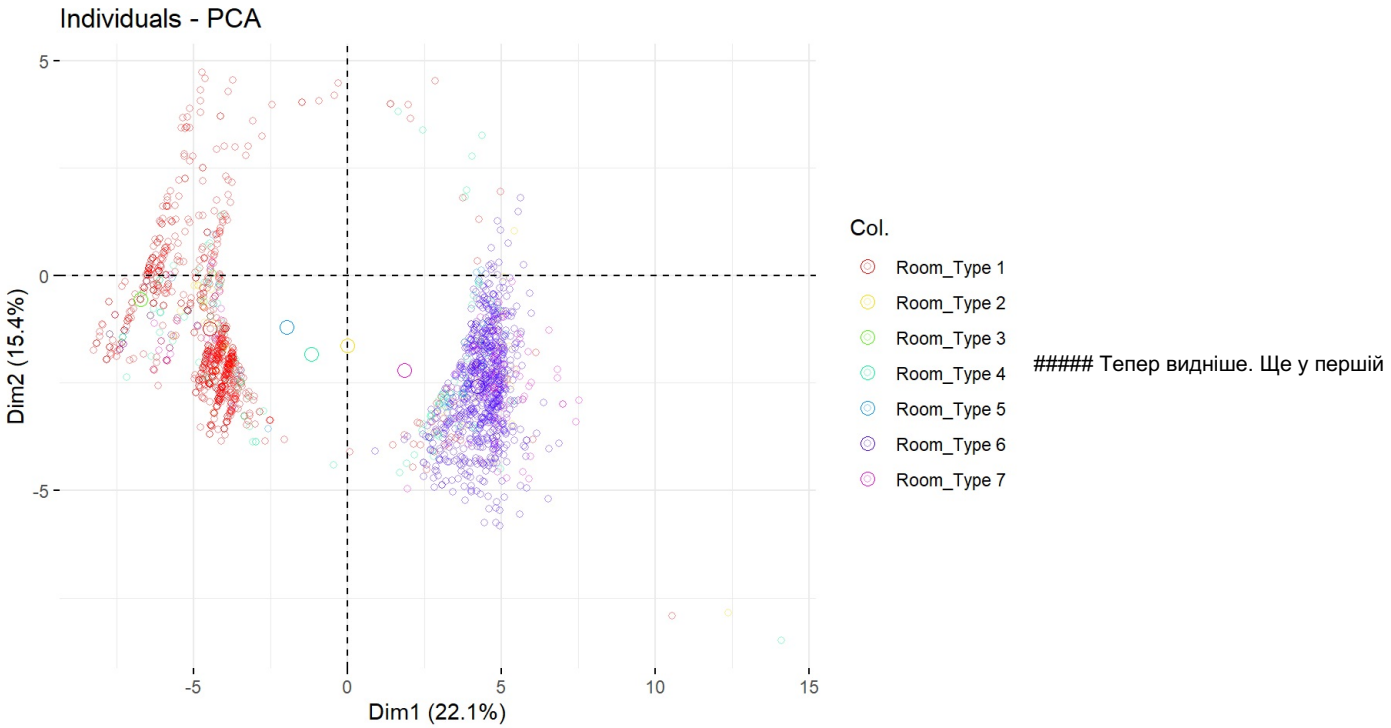
```
fviz_pca_biplot(hotel_pca.pca, axes = c(1, 2), geom = "point",
                select.ind = list(contrib = 2000), pointshape = 1,
                col.ind = factor(hotel_corr$room_type_reserved), repel = TRUE)
```



Через велику кількість атрибут, графік вийшов трохи захарщений, щоб побачити більш “чисту” картинку, побудуємо цей самий графік, але без “стрілочок”, також змінимо палітру кольорів на більш яскраву, але зі збільшеною прозорістю.

```
fviz_pca_ind(hotel_pca.pca, axes = c(1, 2), geom = c("point"),
             select.ind = list(contrib = 2000), alpha.ind = 0.4,
             col.ind = factor(hotel_corr$room_type_reserved),
             pointshape = 1, palette =  c("Room_Type 1" = "#FF0000", "Room_Type 2" = "#FFDB00", "Room_Type 3" = "
#49FF00", "Room_Type 4" = "#00FF92", "Room_Type 5" =  "#0092FF", "Room_Type 6" =  "#4900FF", "Room_Type 7" = "#FF
00DB"))
```

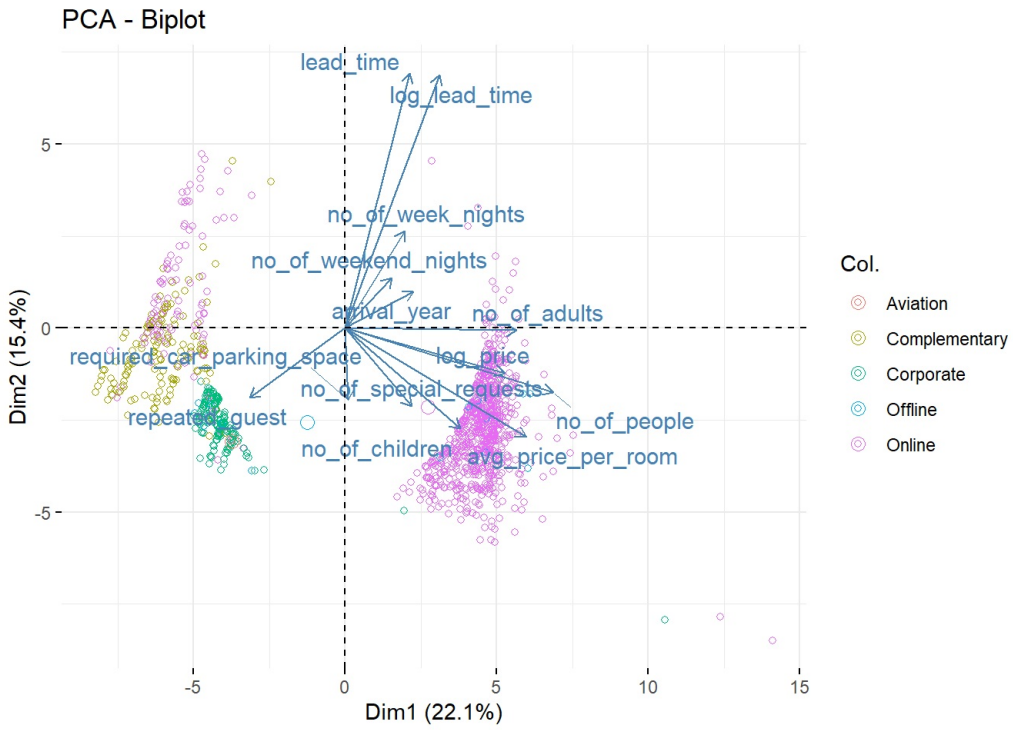
```
## Warning: No shared levels found between `names(values)` of the manual scale and the
## data's fill values.
```



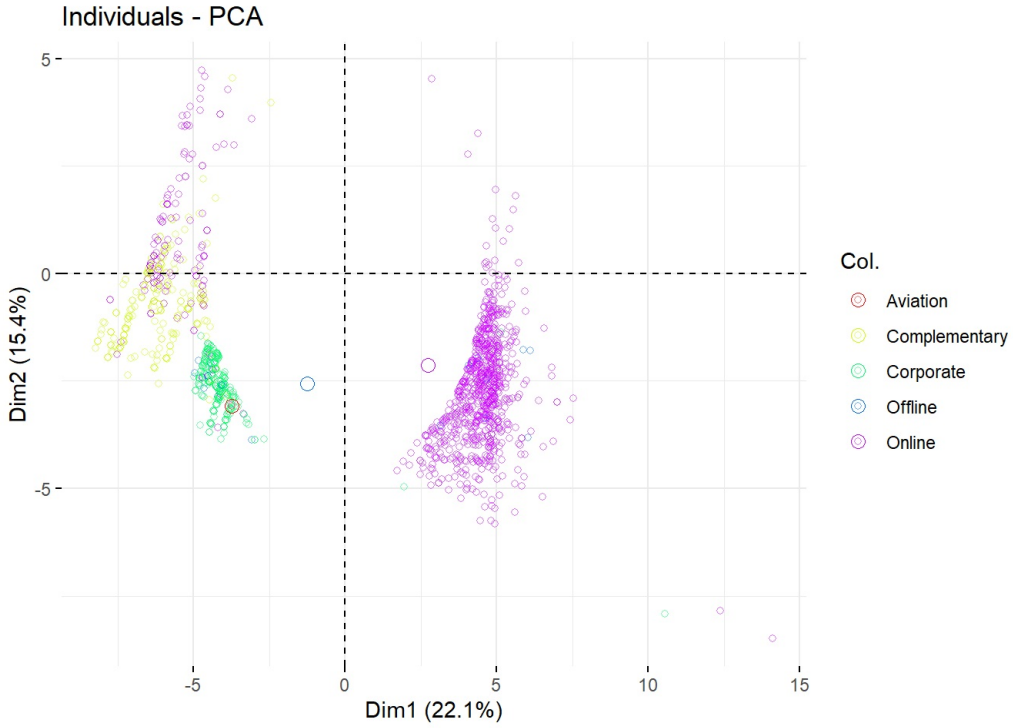
Тепер видніше. Ще у першій лабораторній роботі ми помітили, що скоріше за все, 6 і 7 типи кімнат дорожчі ніж перший (найбільш популярний). За цим графіком бачимо, що дійсно ті, хто платили менше заслелялись в переважній більшості до першого типу кімнати, в той час як ті хто платили менше і з більшою кількістю людей - у 6 і 7 типи.

Побудуємо аналогічні графіки, але для market_segment_type (ринкового сигментування записів)

```
fviz_pca_biplot(hotel_pca.pca, axes = c(1, 2), geom = "point",
                select.ind = list(contrib = 1500), pointshape = 1,
                col.ind = factor(hotel_corr$market_segment_type), repel = TRUE)
```




```
fviz_pca_ind(hotel_pca.pca, axes = c(1, 2), geom = c("point"),
             select.ind = list(contrib = 1500), alpha.ind = 0.5,
             col.ind = factor(hotel_corr$market_segment_type),
             pointshape = 1, palette = c("#FF0000", "#CCFF00" ,"#00FF66", "#0066FF", "#CC00FF"))
```



Бачимо ще більш цікаву картину. Виявляється що переважна більшість людей з мінусовим знаком першої компоненти (ті хто платять менше, і часто повторні гості) належать до особливого сегменту ринку! Тобто люди зліва - ті, що приїжджають майже безкоштовно (Complementary), або на відрядження (Corporate чи Aviation). Також можна помітити, що була частина людей, які як і всі брали квитки онлайн, але при цьому платили менше. Це можна пояснити тим, що гість вже знає за що можна платити і не платити.

Створимо непараметричні регресії залежності середньої ціни за кімнату від кількості людей. Відповідно матимемо 2 моделі: одна з оцінкою Надараї-Вотсона, інша - з локально лінійною.

```
x_grid <- seq(min(hotels$no_of_people), max(hotels$no_of_people), by = 1)

h <- npregbw(avg_price_per_room ~ no_of_people, data = hotels, regtype = 'll')
```

```
## Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 /Multistart 1 of 1 |Multistart 1 of
1 |
```

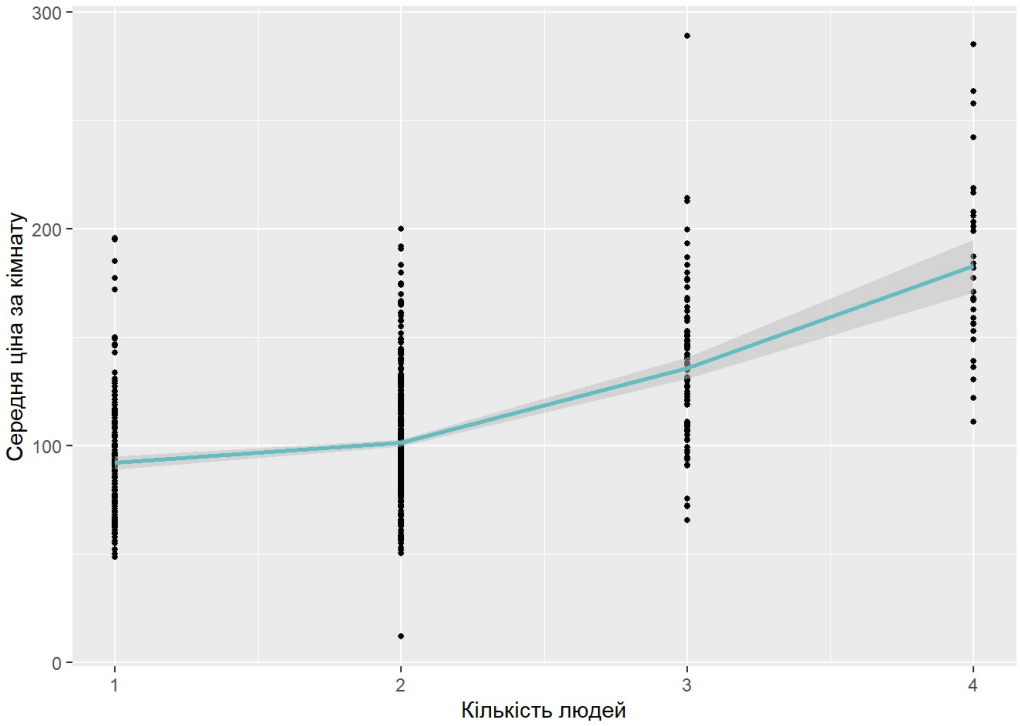
```
ll_LOOCV <- npreg(h, newdata = data.frame(no_of_people = x_grid))
ll_df <- tibble(x = x_grid,
                y_hat = ll_LOOCV$mean,
                h = round(h$bw, 3))


h <- npregbw(avg_price_per_room ~ no_of_people, data = hotels, regtype = 'lc')
```

```
## Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 /Multistart 1 of 1 |Multistart 1 of
1 |
```

```
nw_LOOCV <- npreg(h, newdata = data.frame(no_of_people = x_grid))
nw_df <- tibble(x = x_grid,
                y_hat = nw_LOOCV$mean,
                h = round(h$bw, 3))
```

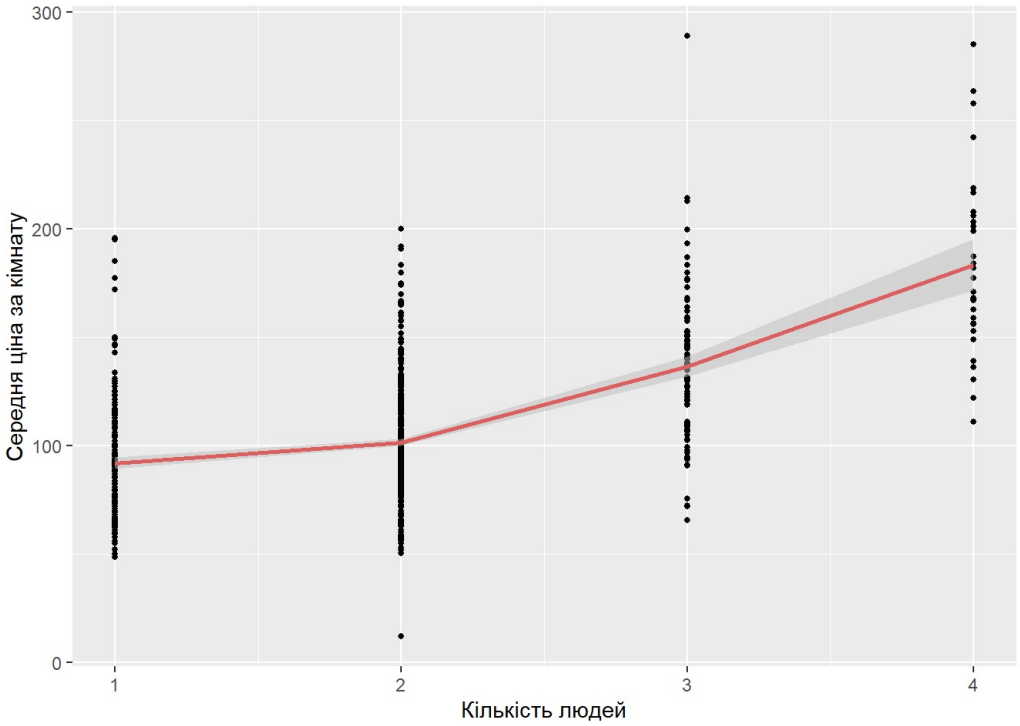
Відповідно побудуємо графік для оцінки Надараї-Вотсона, відмічаючи сірим кольором поточкові довірчі інтервали. Маємо помітно широкі довірчі інтервали для 5-ьох людей, адже це пов'язано з тим, що подібних записів у датасеті досить мало.

```
ggplot(data = data.frame(x = x_grid,
                         y_hat = nw_LOOCV$mean,
                         lower = nw_LOOCV$mean - qnorm(0.975)*nw_LOOCV$merr,
                         upper = nw_LOOCV$mean + qnorm(0.975)*nw_LOOCV$merr),
       aes(x = x, y = y_hat)) +
  geom_point(data = hotels, aes(x = no_of_people, y = avg_price_per_room), size = 1) +
  geom_line(linewidth = 1, color = "#08bcc4") +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.5, fill = "gray") +
  labs(x = "Кількість людей", y = "Середня ціна за кімнату", color = "Тип", linetype = "Тип")
```



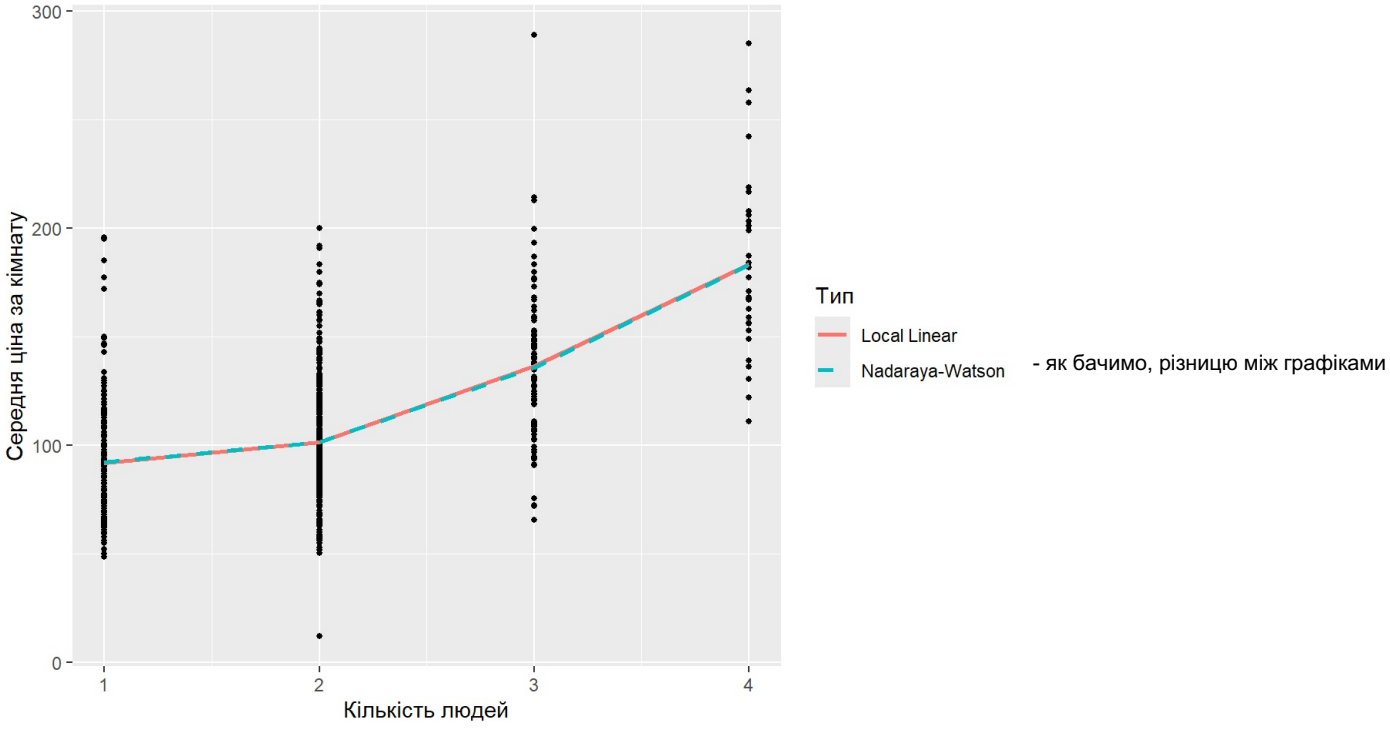
Аналогічно побудуємо довірчий інтервал для локальної лінійної регресії. Можемо спостерігати практично ідентичний результат до попереднього.

```
ggplot(data = data.frame(x = x_grid,
                         y_hat = ll_LOOCV$mean,
                         lower = ll_LOOCV$mean - qnorm(0.975)*ll_LOOCV$merr,
                         upper = ll_LOOCV$mean + qnorm(0.975)*ll_LOOCV$merr),
       aes(x = x, y = y_hat)) +
  geom_point(data = hotels, aes(x = no_of_people, y = avg_price_per_room), size = 1) +
  geom_line(linewidth = 1, color = "red") +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.5, fill = "gray") +
  labs(x = "Кількість людей", y = "Середня ціна за кімнату", color = "Тип", linetype = "Тип")
```



Накладемо обидва графіки один на одного:

```
a_df <- tibble(x = rep(x_grid, 2),
               y_hat = c(ll_LOOCV$mean,
                         nw_LOOCV$mean),
               type = c(rep('Local Linear', length(x_grid)),
         rep('Nadaraya-Watson', length(x_grid))))

ggplot(hotels, aes(x = no_of_people, y = avg_price_per_room)) +
  geom_point(size = 1) +
  geom_line(data = a_df, aes(x = x, y = y_hat, color = type, linetype = type),
            linewidth = 1) +
  scale_linetype_manual(values = c("solid", "dashed")) +
  labs(x = "Кількість людей", y = "Середня ціна за кімнату", color = "Тип", linetype = "Тип")
```



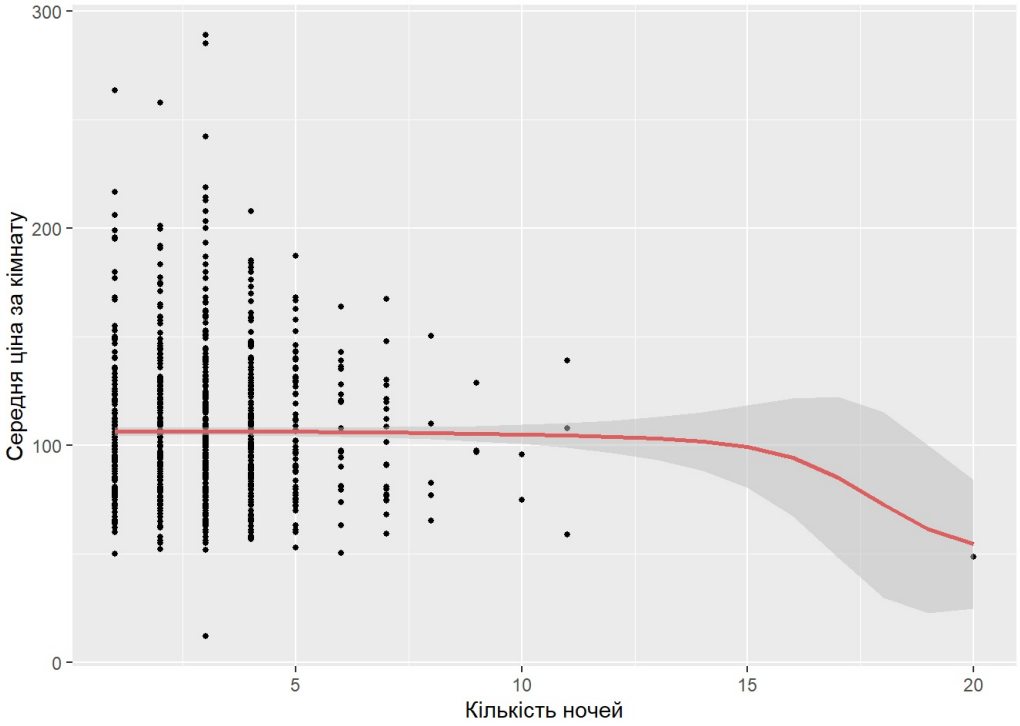
- як бачимо, різницю між графіками для оцінок складно побачити неозброєним оком

Побудуємо непараметричну регресію для залежності середньої ціни за кімнату від кількості проведених ночей у готелі використовуючи оцінку Надараї-Вотсона.

```
x_grid_nights <- seq(min(hotels$no_of_nights), max(hotels$no_of_nights), by = 1)

h <- npregbw(avg_price_per_room ~ no_of_nights, data = hotels, regtype = 'lc')
```

```
## Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 |Multistart 1 of 1 /Multistart 1 of 1 |Multistart 1 of
1 |
```

```
ni_LOOCV <- npreg(h, newdata = data.frame(no_of_nights = x_grid_nights))


ggplot(data = data.frame(x = x_grid_nights,
                         y_hat = ni_LOOCV$mean,
                         lower = ni_LOOCV$mean - qnorm(0.975)*ni_LOOCV$merr,
                         upper = ni_LOOCV$mean + qnorm(0.975)*ni_LOOCV$merr),
       aes(x = x, y = y_hat)) +
  geom_point(data = hotels, aes(x = no_of_nights, y = avg_price_per_room), size = 1) +
  geom_line(linewidth = 1, color = "red") +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.5, fill = "gray") +
  labs(x = "Кількість ночей", y = "Середня ціна за кімнату", color = "Тип", linetype = "Тип")
```



- загалом картина така, що важко помітити хоч якісь закономірності. Сладно сказати, що кількість ночей має певний вплив на ціну за номер.

Повернемось до однієї з регресійних моделей, розглянутих у попередній лабораторній роботі, а саме “як впливає повторність гостя та потреба у паркувальному місці на ціну”.

```
x_grid <- seq(0, 1, by = 1)

h <- npregbw(log_price ~ factor(required_car_parking_space) + factor(repeated_guest), data = hotels, regtype = 'l
l')
```

```
## Multistart 1 of 2 |Multistart 1 of 2 |Multistart 1 of 2 |Multistart 1 of 2 /Multistart 1 of 2 -Multistart 1 of
2 |Multistart 1 of 2 |Multistart 2 of 2 |Multistart 2 of 2 |Multistart 2 of 2 /Multistart 2 of 2 -Multistart 2 of
2 |Multistart 2 of 2 |
```

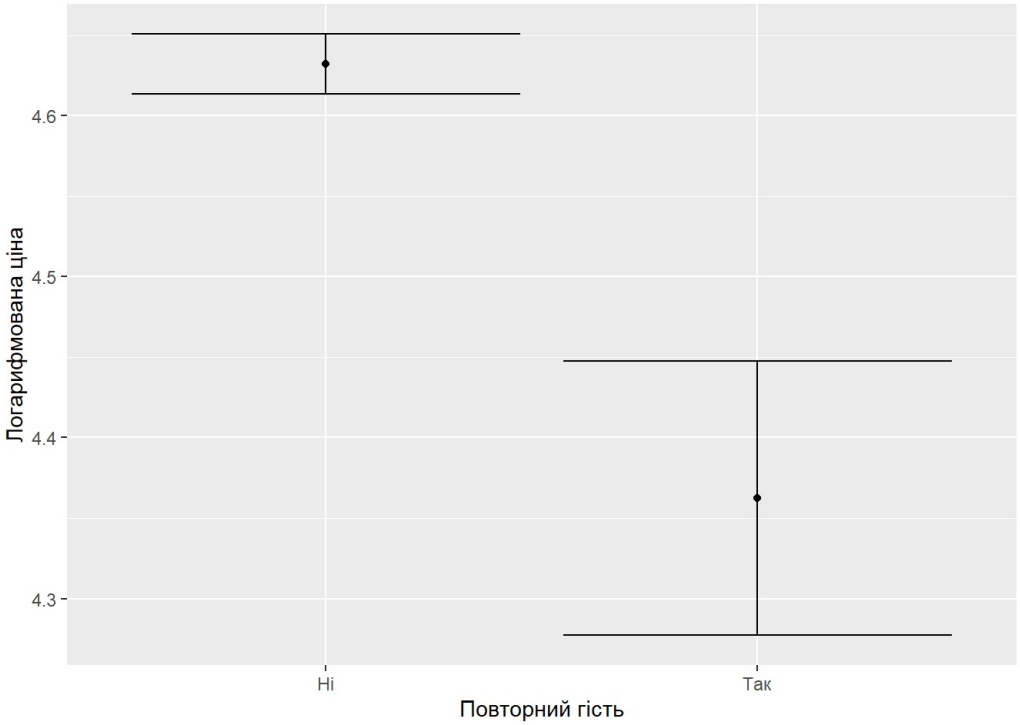
```
ll_LOOCV_gst <- npreg(h, newdata = data.frame(
  repeated_guest = x_grid,
  required_car_parking_space = 0))


h <- npregbw(log_price ~ factor(required_car_parking_space) + factor(repeated_guest), data = hotels, regtype = 'l
l')
```

```
## Multistart 1 of 2 |Multistart 1 of 2 |Multistart 1 of 2 |Multistart 1 of 2 /Multistart 1 of 2 -Multistart 1 of
2 |Multistart 1 of 2 |Multistart 2 of 2 |Multistart 2 of 2 |Multistart 2 of 2 /Multistart 2 of 2 -Multistart 2 of
2 |Multistart 2 of 2 |
```

```
ll_LOOCV_car <- npreg(h, newdata = data.frame(
  repeated_guest = x_grid,
  required_car_parking_space = 1))
```

Побудуємо довірчі інтервали для логарифмованої ціни в залежності від того чи є гість повторним, чи ні. При цьому зафіксуємо необхідність у паркувальному місці на медіанному рівні (тобто 0)

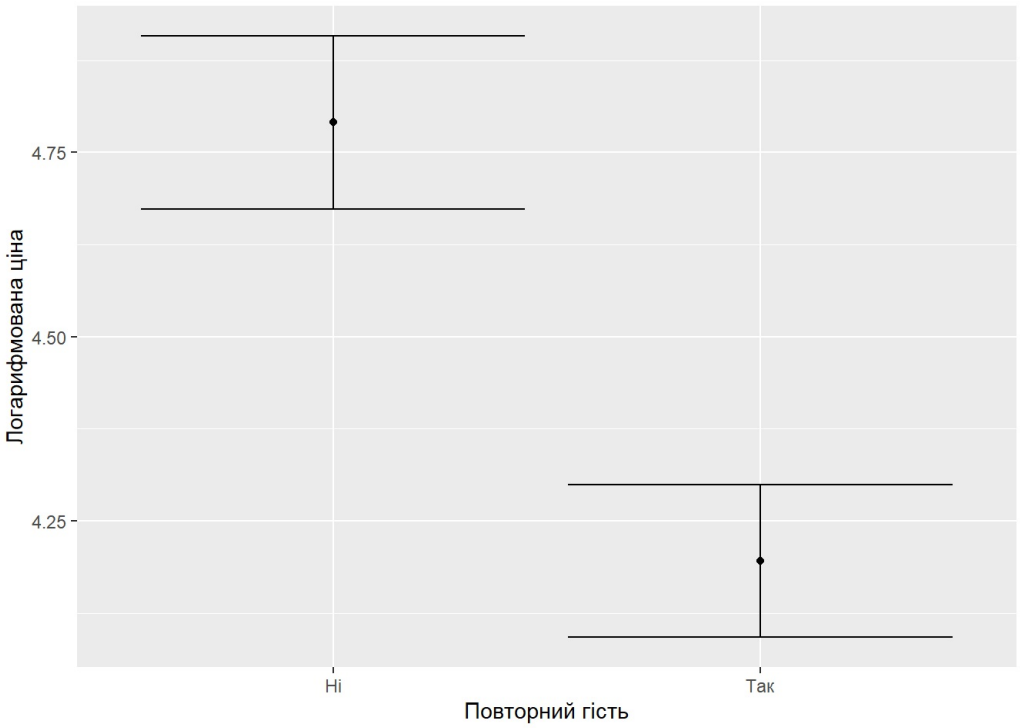
```
ggplot(data = data.frame(x = c("Ні", "Так"),
                         y = ll_LOOCV_gst$mean,
                         se = ll_LOOCV_gst$merr),
       aes(x = x, y = y)) +
  geom_point() +
  geom_errorbar(aes(ymin = y - qnorm(0.975)*se,
                    ymax = y + qnorm(0.975)*se)) + labs(x = "Повторний гість", y = "Логарифмована ціна")
```



- спостерігаємо результат, що є дуже подібним до того, що ми бачили у лінійній регресійній моделі, тобто повторні гості платять приблизно на ~22% менше

Аналогічно побудуємо довірчі інтервали для логарифмованої ціни в залежності від того чи є гість повторним, чи ні. Тільки в цьому випадку зафіксуємо необхідність у паркувальному місці на одиниці.

```
ggplot(data = data.frame(x = c("Ні", "Так"),
                         y = ll_LOOCV_car$mean,
                         se = ll_LOOCV_car$merr),
       aes(x = x, y = y)) +
  geom_point() +
  geom_errorbar(aes(ymin = y - qnorm(0.975)*se,
                    ymax = y + qnorm(0.975)*se)) + labs(x = "Повторний гість", y = "Логарифмована ціна")
```



- можемо бачити різницю приблизно у 30%, як і у моделі лінійної регресії з попередньої лабораторної роботи

Для повнішого порівняння з попередньою лабораторною роботою необхідно побудувати частково лінійну

модель. Для цього непараметрично оцінимо залежність ціни від необхідності у паркувальному місці та фактора повторності гостя, а параметрично - вплив ринкового сегменту, кількості людей, кількосіт ночей і наявності особливих побажань.

```
bw <- npplregbw(log_price ~ required_car_parking_space + repeated_guest  | factor(Online) + factor(Corporate) + f
actor(Complementary) + factor(Aviation) + no_of_people + no_of_nights + no_of_special_requests, data = hotels, re
gtype = "ll")
```

```
## Multistart 1 of 5 |Multistart 1 of 5 |Multistart 1 of 5 |Multistart 1 of 5 /Multistart 1 of 5 -Multistart 1 of
5 \Multistart 1 of 5 |Multistart 1 of 5 |Multistart 2 of 5 |Multistart 2 of 5 |Multistart 2 of 5 /Multistart 2 of
5 -Multistart 2 of 5 |Multistart 2 of 5 |Multistart 3 of 5 |Multistart 3 of 5 |Multistart 3 of 5 /Multistart 3 of
5 -Multistart 3 of 5 \Multistart 3 of 5 |Multistart 3 of 5 |Multistart 4 of 5 |Multistart 4 of 5 |Multistart 4 of
5 /Multistart 4 of 5 -Multistart 4 of 5 \Multistart 4 of 5 |Multistart 4 of 5 |Multistart 4 of 5 |Multistart 5 of
5 |Multistart 5 of 5 |Multistart 5 of 5 /Multistart 5 of 5 -Multistart 5 of 5 \Multistart 5 of 5 |Multistart 5 of
5 /Multistart 5 of 5 -Multistart 5 of 5 \Multistart 5 of 5 |Multistart 5 of 5 |Multistart 5 of 5 |Multistart 5 of
5 /Multistart 5 of 5 -Multistart 5 of 5 \Multistart 5 of 5 |Multistart 5 of 5 /Multistart 5 of 5 -
Multistart 1 of 5 |Multistart 1 of 5 |Multistart 1 of 5 |Multistart 1 of 5 /Multistart 1 of 5 -Multistart 1 of 5
\Multistart 1 of 5 |Multistart 1 of 5 |Multistart 2 of 5 |Multistart 2 of 5 |Multistart 2 of 5 /Multistart 2 of 5
-Multistart 2 of 5 \Multistart 2 of 5 |Multistart 2 of 5 |Multistart 3 of 5 |Multistart 3 of 5 |Multistart 3 of 5
/Multistart 3 of 5 -Multistart 3 of 5 \Multistart 3 of 5 |Multistart 3 of 5 |Multistart 4 of 5 |Multistart 4 of 5
|Multistart 4 of 5 /Multistart 4 of 5 -Multistart 4 of 5 |Multistart 4 of 5 |Multistart 5 of 5 |Multistart 5 of 5
|Multistart 5 of 5 /Multistart 5 of 5 -Multistart 5 of 5 \Multistart 5 of 5 |Multistart 5 of 5 |Multistart 5 of 5
|                   Multistart 1 of 5 |Multistart 1 of 5 |Multistart 1 of 5 |Multistart 1 of 5 /Multistart 1 of 5
-Multistart 1 of 5 \Multistart 1 of 5 |Multistart 1 of 5 |Multistart 1 of 5 |Multistart 1 of 5 /Multistart 1 of 5
-Multistart 1 of 5 \Multistart 2 of 5 |Multistart 2 of 5 |Multistart 2 of 5 /Multistart 2 of 5 -Multistart 2 of 5
\Multistart 2 of 5 |Multistart 2 of 5 /Multistart 2 of 5 |Multistart 2 of 5 |Multistart 3 of 5 |Multistart 3 of 5
|Multistart 3 of 5 /Multistart 3 of 5 -Multistart 3 of 5 \Multistart 3 of 5 |Multistart 3 of 5 |Multistart 4 of 5
|Multistart 4 of 5 |Multistart 4 of 5 /Multistart 4 of 5 -Multistart 4 of 5 \Multistart 4 of 5 |Multistart 4 of 5
|Multistart 5 of 5 |Multistart 5 of 5 |Multistart 5 of 5 /Multistart 5 of 5 -Multistart 5 of 5 \Multistart 5 of 5
|Multistart 5 of 5 |Multistart 5 of 5 |Multistart 5 of 5 /
```

```
model_nppl <- npplreg(bw)

summary(model_nppl)
```

```
##
## Partially Linear Model
## Regression data: 1000 training points, in 9 variable(s)
## With 2 linear parametric regressor(s), 7 nonparametric regressor(s)
##
##                      y(z)
## Bandwidth(s): 0.03675903 0.08551121 0 0.4999993 0.702902 15.91005 135095.4
##
##                      x(z)
## Bandwidth(s): 0.03152686 0.13456906 0 0.4999999 1.754398e+05  856142.6
##               0.17914519 0.08022809 0 0.4052250 1.260874e-01 5621470.4
##
## Bandwidth(s): 9.477155e+05
##               1.231491e-01
##
##                  required_car_parking_space repeated_guest
## Coefficient(s):                 0.06891901    -0.09528177
##
## Kernel Regression Estimator: Local-Linear
## Bandwidth Type: Fixed
##
## Residual standard error: 0.2416388
## R-squared: 0.3355571
##
## Continuous Kernel Type: Second-Order Gaussian
## No. Continuous Explanatory Vars.: 3
##
## Unordered Categorical Kernel Type: Aitchison and Aitken
## No. Unordered Categorical Explanatory Vars.: 4
```

Розглянемо регресійну модель з попередньої лабораторної роботи. Як бачимо, коефіцієнти лишаються статистично значущими і мають приблизно ті самі значення, тож непараметрична регресія дала подібний результат.

```
model_nppl_ti <- tibble(term = names(model_nppl$xcoef),
                        estimate = model_nppl$xcoef,
                        std.error = model_nppl$xcoeferr,
                        p.value = 2*pnorm(-abs(estimate/std.error)))
model_nppl_gl <- data.frame(Num.Obs. = model_nppl$nobs)
mod_nppl <- list(tidy = model_nppl_ti, glance = model_nppl_gl)
class(mod_nppl) <- "modelsummary_list"

modelsummary(list(model_car_ext, mod_nppl),
             stars = TRUE, gof_omit = "^(?!Num.Obs.)")
```

	(1)	(2)
(Intercept)	4.283***	
	(0.007)	
required_car_parking_space1	0.070***	
	(0.007)	
no_of_special_requests	0.006+	
	(0.003)	
no_of_people	0.163***	
	(0.002)	
no_of_nights	-0.017***	
	(0.001)	
log(lead_time)	-0.011***	
	(0.001)	
repeated_guest1	-0.159***	
	(0.011)	
OnlineTRUE	0.150***	
	(0.004)	
CorporateTRUE	-0.008	
	(0.008)	
ComplementaryTRUE	-0.491***	
	(0.146)	
AviationTRUE	0.258***	
	(0.013)	
required_car_parking_space		0.069
		(0.048)
repeated_guest		-0.095
		(0.066)
Num.Obs.	35674	1000

+ p < 0.1, * p < 0.05, ** p < 0.01, *** p < 0.001