

lab2

2024-05-13

```
original_hotel <- read.csv("Hotel Reservations.csv")

hotel_corr <- original_hotel

hotel <- original_hotel %>% filter((avg_price_per_room > 9),
                                         (no_of_children < 8))

hotel <- hotel %>% mutate(repeated_guest = as.factor(repeated_guest),
                           required_car_parking_space = as.factor(required_car_parking_space),
                           booking_status = as.factor(booking_status),
                           room_type_reserved = as.factor(room_type_reserved),
                           no_of_special_requests = ifelse(no_of_special_requests == 0, 0, 1),
                           arrival_year_and_month = paste(arrival_year, arrival_month, arrival_date, sep = "-"),
                           no_of_people = no_of_adults + no_of_children,
                           no_of_nights = no_of_weekend_nights + no_of_week_nights,
                           no_of_people = no_of_adults + no_of_children)

hotel_reverse <- hotel
hotel_reverse$booking_status_binary <- ifelse(hotel$booking_status == "Canceled", 1, 0)

room_type_vector <- c("Room_Type 1" = "#ff746c", "Room_Type 2" = "#c89c04", "Room_Type 3" = "#58b404", "Room_Type 4" = "#08c494", "Room_Type 5" = "#08b4ec", "Room_Type 6" = "#a88cf", "Room_Type 7" = "#ff64d4")
room_label_vector <- c("Тип кімнати №1", "Тип кімнати №2", "Тип кімнати №3", "Тип кімнати №4", "Тип кімнати №5",
                      "Тип кімнати №6", "Тип кімнати №7")

meal_type_vector <- c("Meal Plan 1" = "#ff746c", "Meal Plan 2" = "#80ac04", "Meal Plan 3" = "#08bcc4", "Not Selected" = "#c87cf")
meal_label_vector <- c("BB - Bed and Breakfast", "HB - Half Board", "FB - Full Board", "Not Selected")
```

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)

У лабораторній роботі №1 було проведено розвідковий аналіз даних в ході якого було встановлено деякі цікаві закономірності в даних. Зокрема було обчислено низку характеристик відповідних вибірок. У поточній лабораторній роботі стоїть задача підсилити ці знахідки, виконавши перевірку їхньої статистичної значущості.

Нагадаємо перелік дослідницьких питань сформованих у першій лабораторній роботі.

- Як змінюється кількість дорослих та дітей в залежності від типу номеру та плану харчування?
- Яким чином розподіляється кількість особливих побажань в залежності від типу номеру?
- Як впливає кількість дорослих і дітей на скасування бронювання?
- Що впливає на кількість проведених вихідних та робочих ночей у готелі?
- Як різні типи кімнат впливають на кількість попередніх скасувань/бронювань?
- Чи є різниця в кількості попередніх скасувань для клієнтів, які вимагають паркувальне місце і тих, хто його не потребує?

В додаток до наведеного списку в ході лабораторної роботи було розглянуто ще деякі проміжні питання, які заслуговують уваги своєю цікавістю. Деякі з них віднесені до конкретних дослідницьких питань, через що досл. питання для відповідності були незначним чином переформульовані, а деякі - виділилися окремо від раніше розглянутих.

Розгляньмо детальніше про що йде мова

- Окремий інтерес становлять типи кімнат готелю, адже нашій команді так і не вдалося з'ясувати яким типам кімнат в дійсності відповідають назви Room_Type 1, 2, За інформацією з опису датасету типи зарезервованих кімнат були закодовані безпосередньо готелями (were encoded by INN Hotels)
- Для того, щоб дізнатися, в яких межах можуть лежати справжні середні значення цін кожного типу кімнати, побудуємо довірчі інтервали відповідно для кожного типу. Оскільки середнє вибіркове має асимптотично нормальну розподіл і для неї можна легко порахувати дисперсію (та оцінити її), то за правилом двох сигм дістаємо наступні довірчі інтервали:

```

cis <- list()

for (i in 1:7) {
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(mean = mean(avg_price_per_room),
              sd = sd(avg_price_per_room),
              n = n(),
              a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n())),
              b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))

  cis[[current_room_type]] <- ci
}

ci_tibble <- bind_rows(cis, .id = "Room_Type")

ci_tibble

```

```

##      Room_Type     mean       sd     n      a      b
## 1 Room_Type 1  97.37477 25.58845 27707  97.07347 97.67607
## 2 Room_Type 2  92.72696 27.87711   654  90.59044 94.86348
## 3 Room_Type 3 103.15000 29.35111     5  77.42309 128.87691
## 4 Room_Type 4 126.56520 31.08103  5995 125.77842 127.35197
## 5 Room_Type 5 132.72636 38.12698   247 127.97156 137.48115
## 6 Room_Type 6 185.47060 33.34539   949 183.34906 187.59214
## 7 Room_Type 7 209.57547 45.58719   117 201.31513 217.83581

```

Для наглядності розглянемо отриманий результат графічно:

```

cis <- list()

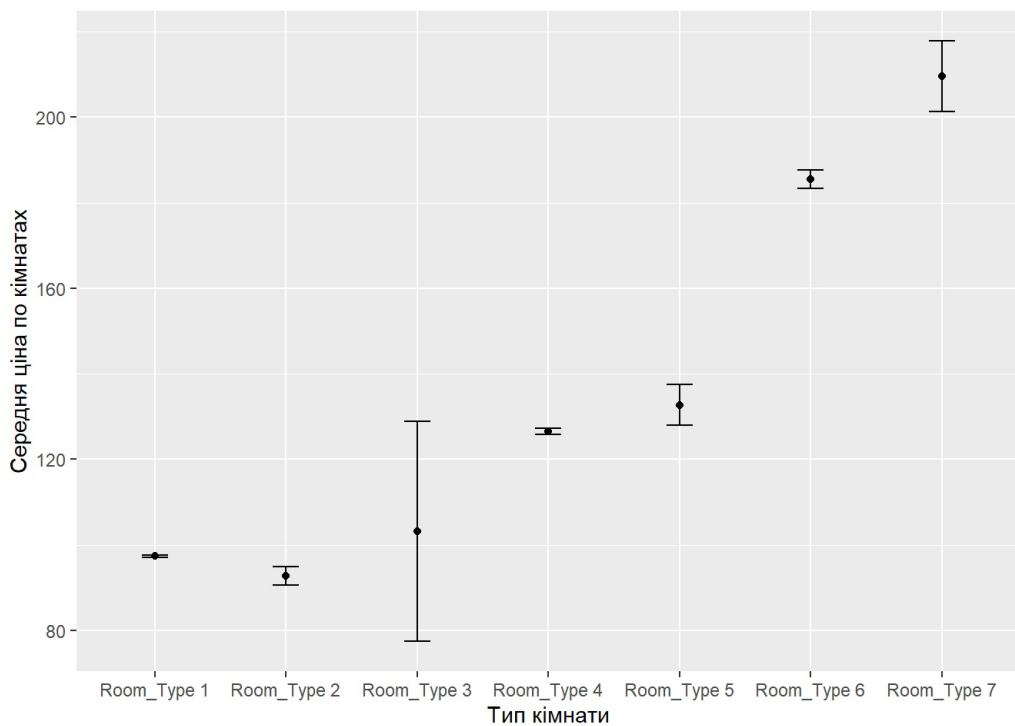
for (i in 1:7) {
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(mean = mean(avg_price_per_room),
              sd = sd(avg_price_per_room),
              n = n(),
              a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n())),
              b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))

  cis[[current_room_type]] <- ci
}

ci_df <- bind_rows(cis, .id = "Room_Type")

# Plot confidence intervals
ggplot(ci_df, aes(x = Room_Type, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип кімнати", y = "Середня ціна по кімнатах")

```



Як можемо спостерігати,

ціна за кожен наступний тип кімнати В ЦІЛОМУ має тенденцію збільшуватись.

Зокрема у даній лабораторній роботі у нас є можливість побудувати довірчі інтервали для середніх значень цін по кожному типу бронювання. Таким чином ми можемо приблизно з'ясувати якому ціновому сегменту відповідає кожен тип бронювання:

```
hotel_segment <- hotel %>% mutate(market_segment_type = as.factor(market_segment_type))

pairwise.t.test(hotel$avg_price_per_room, hotel_segment$market_segment_type,
p.adjust.method = "BH", pool.sd = FALSE)
```

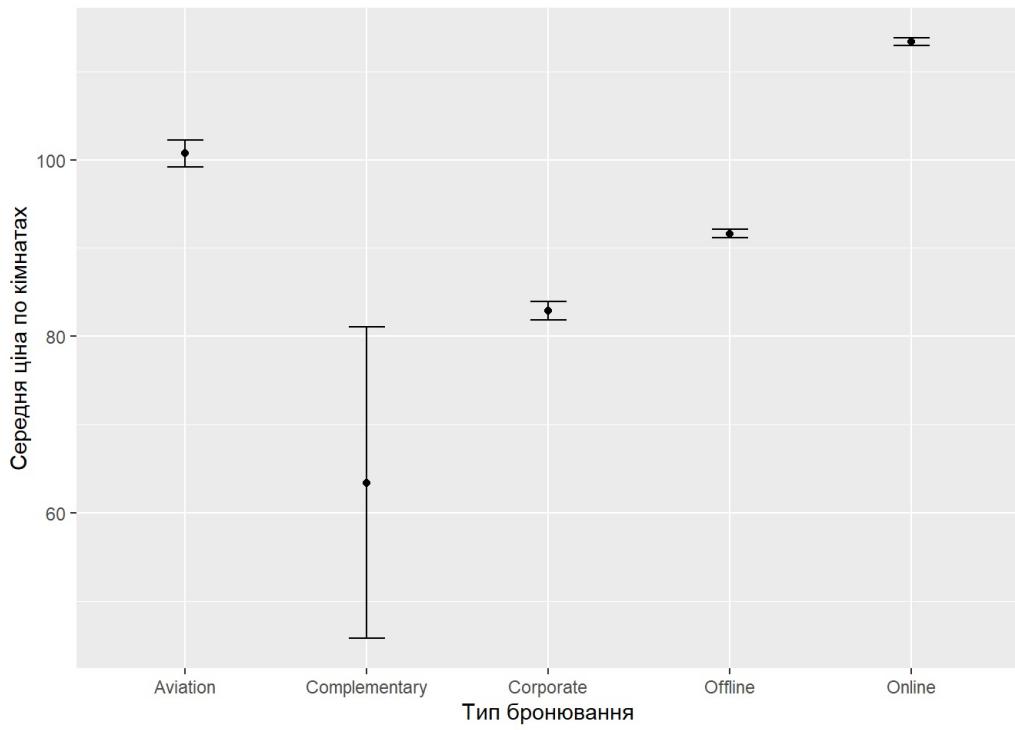
```
## 
## Pairwise comparisons using t tests with non-pooled SD
##
## data: hotel$avg_price_per_room and hotel_segment$market_segment_type
##
##          Aviation Complementary Corporate Offline
## Complementary 0.00086   -      -      -
## Corporate     < 2e-16  0.04521   -      -
## Offline        < 2e-16  0.00673  < 2e-16  -
## Online         < 2e-16  5.1e-05  < 2e-16 < 2e-16
##
## P value adjustment method: BH
```

```
cis <- list()

for (i in c("Aviation", "Complementary", "Corporate", "Offline", "Online")) {
  current_market <- i
  ci <- hotel_segment %>%
    filter(market_segment_type == current_market) %>%
    summarize(
      mean = mean(avg_price_per_room),
      sd = sd(avg_price_per_room),
      n = n(),
      a = mean(avg_price_per_room) - qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()),
      b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n())
    )
  
  cis[[current_market]] <- ci
}

ci_df <- bind_rows(cis, .id = "market_segment_type")

ggplot(ci_df, aes(x = market_segment_type, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип бронювання", y = "Середня ціна по кімнатах")
```



Варто зауважити, що враховуються лише ті записи, для яких $\text{avg_price_per_room} \geq 9$, адже в датасеті міститься досить багато записів з цінами ~0 без вказання конкретної причини безоплатного проживання протягом того чи іншого терміну (про це детальніше розказувалося в презентації до лабораторної роботи №1)

Оновлений перелік дослідницьких питань наведений нижче:

- 1. Що впливає на вибір типу номеру чи плану харчування?
- 2. Чи є істотною наявність особливих побажань?
- 3. Які характерні риси скасованих записів?
- 4. Що впливає на кількість проведених ночей у готелі?
- 5. Що впливає на кількість попередніх скасувань/нескасувань?
- 6. Які характерні риси бронювань з потребою у паркувальному місці?
- 7. Які нові цікаві відомості про повторних гостей?

Перейдемо безпосередньо до дослідницьких питань.

1. Що впливає на вибір типу номеру чи плану харчування?

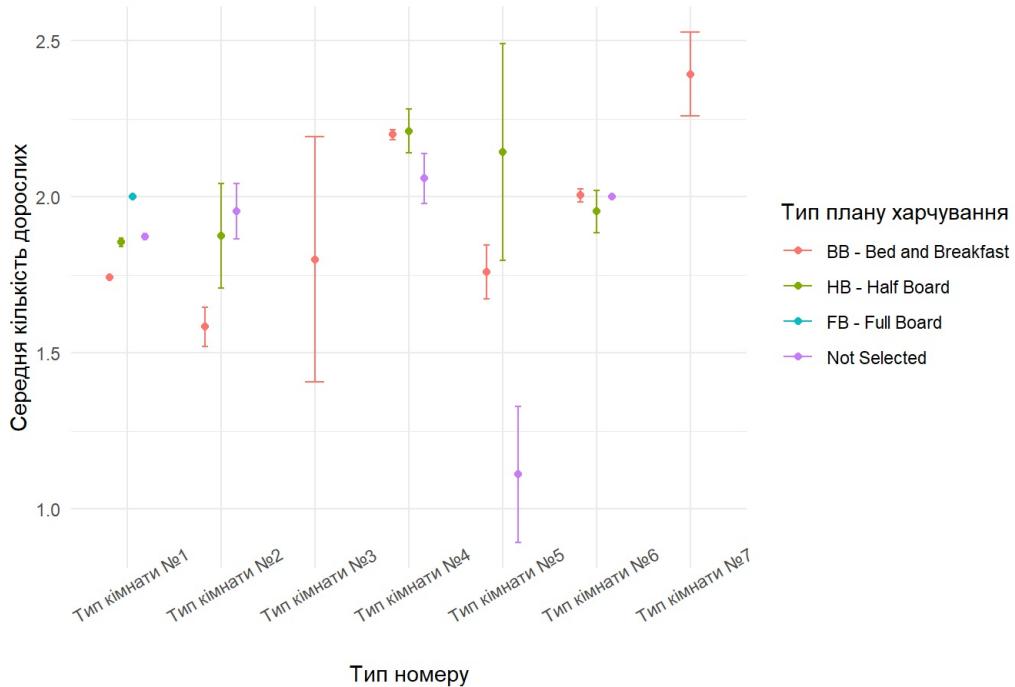
Метою цього дослідження є визначення, як змінюється кількість дорослих і дітей, що бронюють номер в готелі, в залежності від типу номеру та плану харчування. Для цього ми використовуємо групування даних за типом номеру і планом харчування, обчислюємо середню кількість дорослих і дітей, а також їх стандартні похибки. На основі цих даних будуємо довірчі інтервали для кожної групи.

```
ci_adults_children <- hotel %>%
  group_by(room_type_reserved, type_of_meal_plan) %>%
  summarize(mean_adults = mean(no_of_adults),
           sd_adults = sd(no_of_adults),
           n_rows = n(),
           ci_low_adults = mean(no_of_adults) - qnorm(0.975) * sd(no_of_adults) / sqrt(n_rows),
           ci_high_adults = mean(no_of_adults) + qnorm(0.975) * sd(no_of_adults) / sqrt(n_rows),
           mean_children = mean(no_of_children),
           sd_children = sd(no_of_children),
           ci_low_children = mean(no_of_children) - qnorm(0.975) * sd(no_of_children) / sqrt(n_rows),
           ci_high_children = mean(no_of_children) + qnorm(0.975) * sd(no_of_children) / sqrt(n_rows),
           .groups = 'drop')

# print(ci_adults_children)

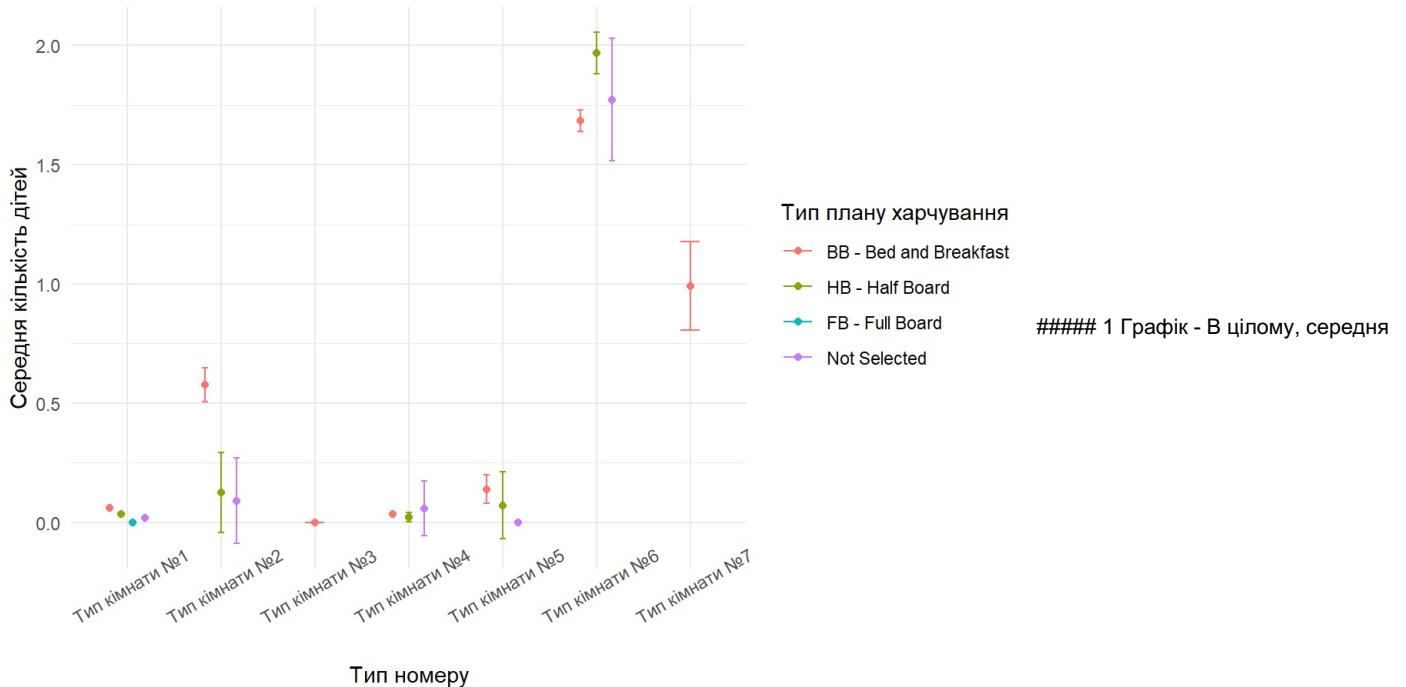
ggplot(ci_adults_children, aes(x = room_type_reserved, y = mean_adults, color = type_of_meal_plan)) +
  geom_point(position = position_dodge(width = 0.5)) +
  geom_errorbar(aes(ymin = ci_low_adults, ymax = ci_high_adults), width = 0.2, position = position_dodge(width = 0.5)) +
  labs(title = "Кількість дорослих в залежності від типу номеру та плану харчування", x = "Тип номеру", y = "Середня кількість дорослих") +
  theme_minimal() +
  scale_color_manual(name = "Тип плану харчування",
                     values = meal_type_vector,
                     labels = meal_label_vector) +
  scale_x_discrete(labels = room_label_vector) +
  theme(axis.text.x = element_text(angle = 30, hjust = 0.5), plot.title = element_text(hjust = 0.5))
```

Кількість дорослих в залежності від типу номеру та плану харчування



```
ggplot(ci_adults_children, aes(x = room_type_reserved, y = mean_children, color = type_of_meal_plan)) +
  geom_point(position = position_dodge(width = 0.5)) +
  geom_errorbar(aes(ymin = ci_low_children, ymax = ci_high_children), width = 0.2, position = position_dodge(width = 0.5)) +
  labs(title = "Кількість дітей в залежності від типу номеру та плану харчування",
       x = "Тип номеру",
       y = "Середня кількість дітей",
       color = "Тип плану харчування") +
  theme_minimal() +
  scale_color_manual(name = "Тип плану харчування",
                     values = meal_type_vector,
                     labels = meal_label_vector) +
  scale_x_discrete(labels = room_label_vector) +
  theme(axis.text.x = element_text(angle = 30, hjust = 0.5), plot.title = element_text(hjust = 0.5))
```

Кількість дітей в залежності від типу номеру та плану харчування



кількість дорослих варіється від 1.5 до 2.5 для різних типів номерів та планів харчування. Кількість дорослих значно варіється в залежності від типу номеру та плану харчування. Деякі типи номерів можуть приваблювати більше дорослих через специфічні умови або зручності, що надаються. Наприклад, номери типу Room_Type 7 з планом харчування Meal Plan 1 мають найвищу середню кількість дорослих, що може свідчити про те, що ці номери і план харчування задовольняють потреби більших груп дорослих. ##### 2 Графік - Залежно від типу номеру та вибору плану харчування спостерігається значна варіативність у кількості дітей лише для останніх типів (6 і 7). Деякі категорії номерів приваблюють більше сімей з дітьми завдяки спеціальним пропозиціям чи додатковим зручностям. Наприклад,

кімнати класу Room_Type 7 з планом харчування Meal Plan 2 виявляються найбільш популярними серед родин з дітьми, що може вказувати на їхню відповідність потребам таких відвідувачів. Тобто дивлячись на обидва графіки можна побачити що для дітей не буде нічого цікавого так як схоже що всі веселоши у тих хто бездітний, в нашому випадку наглядні результати лише у дорослих.

Розглянемо як змінюється кількість дорослих, що бронюють номер в готелі, в залежності від типу номеру. Для цього ми використовуємо групування даних за типом номеру, обчислюємо середню кількість дорослих, а також їх стандартні відхилення. На основі цих даних будуємо довірчі інтервали для кожного типу номеру, щоб оцінити варіацію кількості дорослих у різних типах номерів.

```

cis <- list()
for (i in 1:7) {
  # Filter data for the current room type
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(mean = mean(no_of_adults),
             sd = sd(no_of_adults),
             n = n(),
             a = mean(no_of_adults) + qnorm(0.025) * sd(no_of_adults) / sqrt(n()),
             b = mean(no_of_adults) + qnorm(0.975) * sd(no_of_adults) / sqrt(n()))

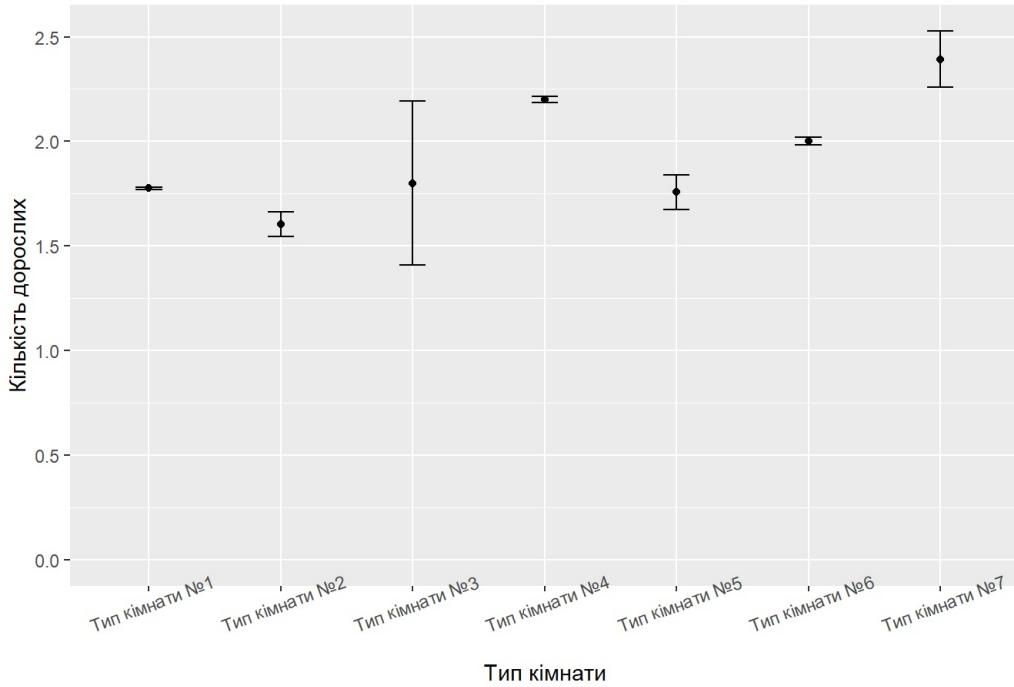
  # Store the confidence interval in the list
  cis[[current_room_type]] <- ci
}

ci_df <- bind_rows(cis, .id = "Room_Type")

# Plot confidence intervals
ggplot(ci_df, aes(x = Room_Type, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип кімнати", y = "Кількість дорослих", title = "В кімнатах 4, 6, 7 дещо більше дорослих ніж в інших") +
  ylim(0, NA) +
  scale_x_discrete(labels = room_label_vector) +
  theme(axis.text.x = element_text(angle = 20, hjust = 0.5), plot.title = element_text(hjust = 0.5))

```

В кімнатах 4, 6, 7 дещо більше дорослих ніж в інших



Можемо побачити, що кількість дорослих, що бронюють номер, значно залежить від типу номеру. Наприклад, Room_Type 7 приваблює більші групи дорослих, можливо через більші розміри або кращі умови в цих номерах. В той же час, Room_Type 2, можливо, є меншими або менш зручними для великих груп.

довірчі інтервали для дисперсії по кількості людей відносно типів кімнат

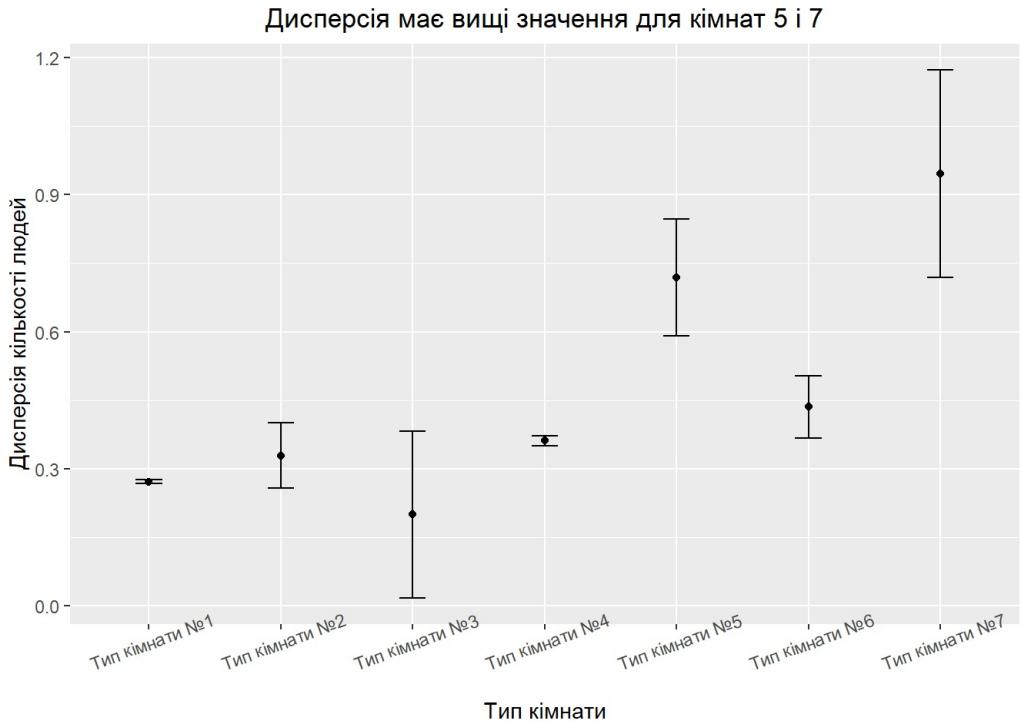
```

cis <- list()
for (i in 1:7) {
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(
      n = n(),
      mean = mean(no_of_people),
      var = var(no_of_people),
      fourth_moment = mean((no_of_people - mean)^4),
      sd_var = sqrt((fourth_moment - var^2) / n),
      a = var - qnorm(0.975) * sd_var,
      b = var + qnorm(0.975) * sd_var
    )
  cis[[current_room_type]] <- ci
}

ci_df <- bind_rows(cis, .id = "Room_Type")

ggplot(ci_df, aes(x = Room_Type, y = var)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип кімнати", y = "Дисперсія кількості людей", title = "Дисперсія має вищі значення для кімнат 5 і 7") +
  scale_x_discrete(labels = room_label_vector) +
  theme(axis.text.x = element_text(angle = 20, hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



Бачимо що дисперсія кількості людей значно відрізняється між різними типами номерів. Найменша дисперсія спостерігається в інтервалі Room_Type 3, але це зобумовлено тим що у ньому спостерігаються найменша кількість записів. Найвища дисперсія в Room_Type 7 може вказувати на більшу варіативність у розмірах груп, що бронюють цей тип номеру.

Тепер розглянемо як змінюється кількість дітей, що бронюють номер в готелі, в залежності від типу номеру. Для цього ми використовуємо групування даних за типом номеру, обчислюємо середню кількість дітей, а також їх стандартні відхилення. На основі цих даних будуємо довірчі інтервали для кожного типу номеру, щоб оцінити варіацію кількості дітей у різних типах номерів.

```

cis <- list()
for (i in 1:7) {
  # Filter data for the current room type
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(mean = mean(no_of_children),
              sd = sd(no_of_children),
              n = n(),
              a = mean(no_of_children) + qnorm(0.025) * sd(no_of_children) / sqrt(n()),
              b = mean(no_of_children) + qnorm(0.975) * sd(no_of_children) / sqrt(n()))

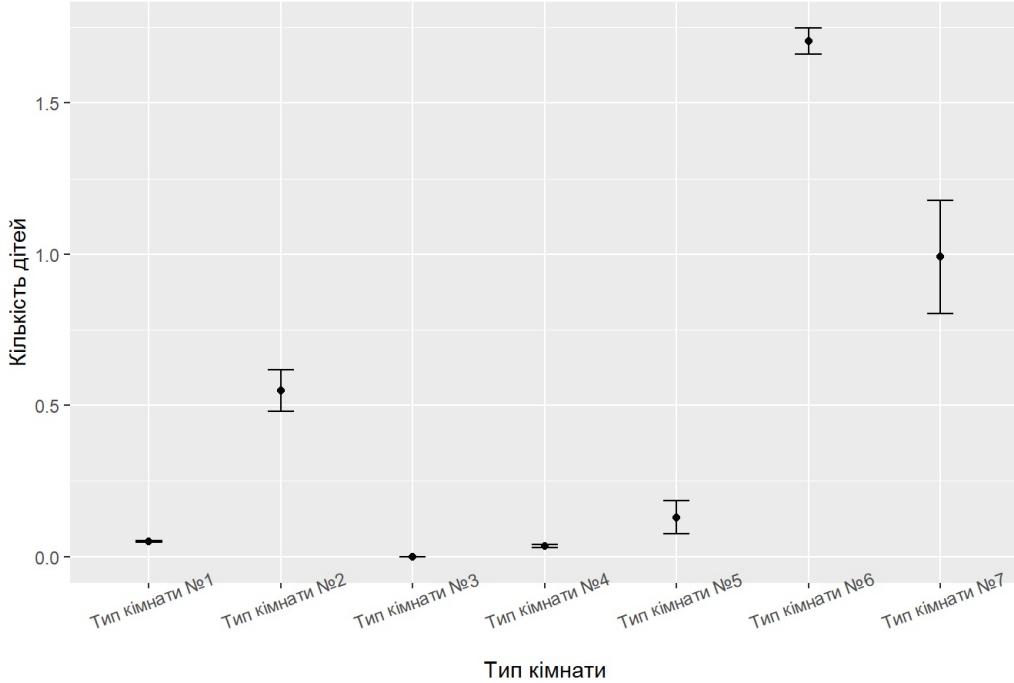
  # Store the confidence interval in the list
  cis[[current_room_type]] <- ci
}

ci_df <- bind_rows(cis, .id = "Room_Type")

# Plot confidence intervals
ggplot(ci_df, aes(x = Room_Type, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип кімнати", y = "Кількість дітей", title = "В 2 і 7 більше дітей ніж в інших, в 6 - значно більше") +
  ylim(0, NA) +
  scale_x_discrete(labels = room_label_vector) +
  theme(axis.text.x = element_text(angle = 20, hjust = 0.5), plot.title = element_text(hjust = 0.5))

```

В 2 і 7 більше дітей ніж в інших, в 6 - значно більше



Бачимо що кількість дітей, які бронюють номер, значно залежить від типу номеру. Наприклад, Room_Type 7 приваблює більше дітей, можливо через більші розміри або кращі умови для сімей. В той же час, Room_Type 1 і Room_Type 3 мають найменшу середню кількість дітей, що може свідчити про те, що ці номери менш зручні або менш популярні серед сімей з дітьми. І взагалі Room_Type 3

мабуть предназначений для певного кола осіб так як має малу кількість записів та взагалі без пар з дітьми

Дослідимо як змінюється кількість людей, що бронюють номер в готелі, в залежності від типу номеру. Для цього ми використовуємо групування даних за типом номеру, обчислюємо середню кількість людей, а також їх стандартні відхилення. На основі цих даних будуємо довірчі інтервали для кожного типу номеру, щоб оцінити варіацію кількості людей у різних типах номерів.

```

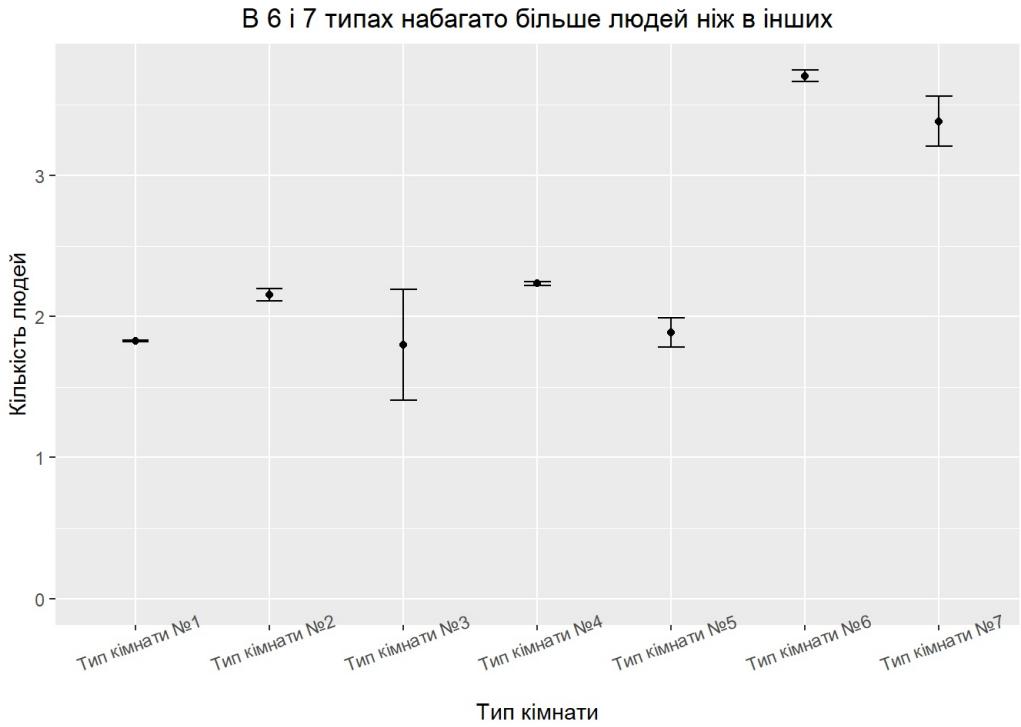
for (i in 1:7) {
  # Filter data for the current room type
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(mean = mean(no_of_people),
             sd = sd(no_of_people),
             n = n(),
             a = mean(no_of_people) + qnorm(0.025) * sd(no_of_people) / sqrt(n()),
             b = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))

  # Store the confidence interval in the list
  cis[[current_room_type]] <- ci
}

ci_df <- bind_rows(cis, .id = "Room_Type")

ggplot(ci_df, aes(x = Room_Type, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип кімнати", y = "Кількість людей", title = "В 6 і 7 типах набагато більше людей ніж в інших") +
  ylim(0, NA) +
  scale_x_discrete(labels = room_label_vector) +
  theme(axis.text.x = element_text(angle = 20, hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



Кількість людей, які бронюють номер, майже для всіх значень тримається на ожному рівні (приблизно 2) але деякі типи відрізнились. Наприклад, Room_Type 7 приваблює більше людей, можливо через більші розміри або кращі умови для великих груп. В той же час, Room_Type 3 має найширший довірчий інтервал, що свідчить про високу невизначеність у кількості людей, які бронюють цей тип номеру.

Наступним з'ясуємо як змінюється кількість дорослих, що бронюють номер у готелі, в залежності від типу плану харчування. Для цього ми використовуємо групування даних за типом плану харчування, обчислюємо середню кількість дорослих та їх стандартні відхилення. На основі цих даних будуємо довірчі інтервали для кожної групи.

```

cis <- list()

for (i in 1:3) {
  current_meal_plan <- paste0("Meal Plan ", i)
  ci <- hotel %>%
    filter(type_of_meal_plan == current_meal_plan) %>%
    summarize(mean = mean(no_of_adults),
              sd = sd(no_of_adults),
              n = n(),
              a = mean(no_of_adults) + qnorm(0.025) * sd(no_of_adults) / sqrt(n()),
              b = mean(no_of_adults) + qnorm(0.975) * sd(no_of_adults) / sqrt(n()))

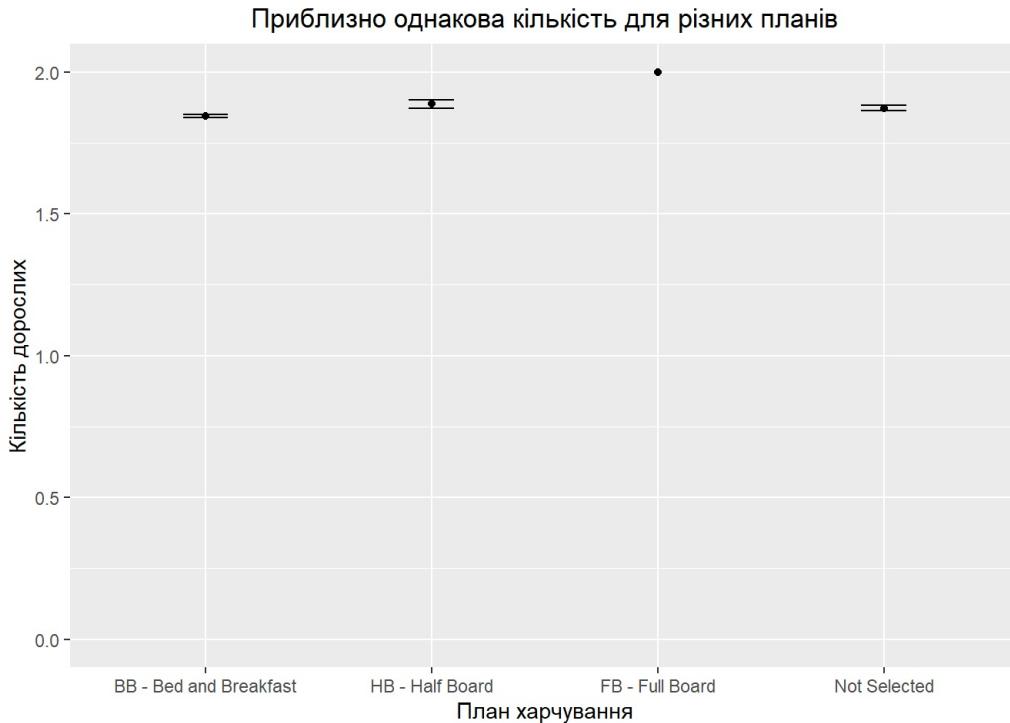
  cis[[current_meal_plan]] <- ci
}
ci <- hotel %>%
  filter(type_of_meal_plan == "Not Selected") %>%
  summarize(mean = mean(no_of_adults),
            sd = sd(no_of_adults),
            n = n(),
            a = mean(no_of_adults) + qnorm(0.025) * sd(no_of_adults) / sqrt(n()),
            b = mean(no_of_adults) + qnorm(0.975) * sd(no_of_adults) / sqrt(n()))

cis[["Not Selected"]] <- ci

ci_df <- bind_rows(cis, .id = "Meal_plan")

# Plot confidence intervals
ggplot(ci_df, aes(x = Meal_plan, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "План харчування", y = "Кількість дорослих", title = "Приблизно однацова кількість для різних планів") +
  ylim(0, NA) +
  scale_x_discrete(labels = meal_label_vector) +
  theme(axis.text.x = element_text(angle = , hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



На основі графіка можна зробити висновок, що середня кількість дорослих, що бронюють номери в готелі, незначно варіється в залежності від типу плану харчування. Усі типи планів харчування, включаючи опцію "Not Selected", мають середню кількість дорослих близько 2. Це свідчить про те, що вибір плану харчування не має суттєвого впливу на кількість дорослих у бронюванні.

Побудуємо довірчі інтервали для середньої кількості дітей по кожному типу плану харчування.

Використовуємо групування даних за типом плану харчування, обчислюємо середню кількість дітей, а також їх стандартні похибки. На основі цих даних будуємо довірчі інтервали для кожної групи.

```

cis <- list()

for (i in 1:3) {
  current_meal_plan <- paste0("Meal Plan ", i)
  ci <- hotel %>%
    filter(type_of_meal_plan == current_meal_plan) %>%
    summarize(mean = mean(no_of_children),
              sd = sd(no_of_children),
              n = n(),
              a = mean(no_of_children) + qnorm(0.025) * sd(no_of_children) / sqrt(n()),
              b = mean(no_of_children) + qnorm(0.975) * sd(no_of_children) / sqrt(n()))

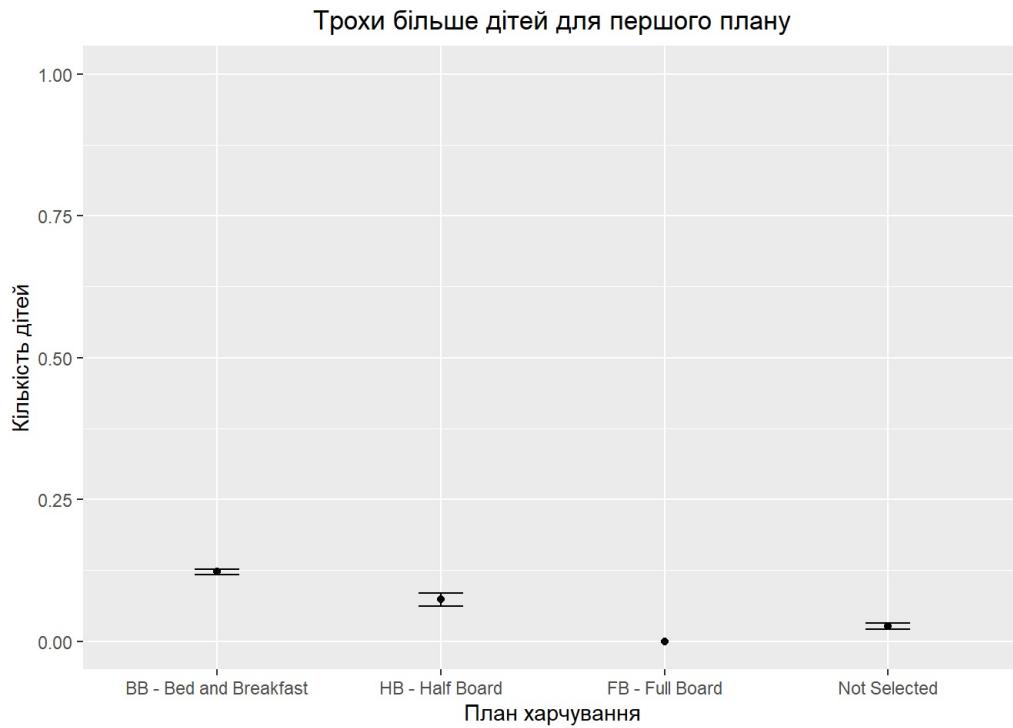
  cis[[current_meal_plan]] <- ci
}
ci <- hotel %>%
  filter(type_of_meal_plan == "Not Selected") %>%
  summarize(mean = mean(no_of_children),
            sd = sd(no_of_children),
            n = n(),
            a = mean(no_of_children) + qnorm(0.025) * sd(no_of_children) / sqrt(n()),
            b = mean(no_of_children) + qnorm(0.975) * sd(no_of_children) / sqrt(n()))

cis[["Not Selected"]] <- ci

ci_df <- bind_rows(cis, .id = "Meal_plan")

# Plot confidence intervals
ggplot(ci_df, aes(x = Meal_plan, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "План харчування", y = "Кількість дітей", title = "Трохи більше дітей для первого плану") +
  ylim(0, 1) +
  scale_x_discrete(labels = meal_label_vector) +
  theme(axis.text.x = element_text(angle = , hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



Тип плану харчування суттєво не впливає на середню кількість дітей, що бронюють номери у готелі. Середні значення залишаються низькими для всіх типів планів, діти можуть і поголодувати в принципі, а довірчі інтервали підтверджують стабільність цих даних.

Повторимо все те ж саме що і для попередніх, але тут і для дорослих і для дітей

```

cis <- list()

for (i in 1:3) {
  current_meal_plan <- paste0("Meal Plan ", i)
  ci <- hotel %>%
    filter(type_of_meal_plan == current_meal_plan) %>%
    summarize(mean = mean(no_of_people),
              sd = sd(no_of_people),
              n = n(),
              a = mean(no_of_people) + qnorm(0.025) * sd(no_of_people) / sqrt(n()),
              b = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))

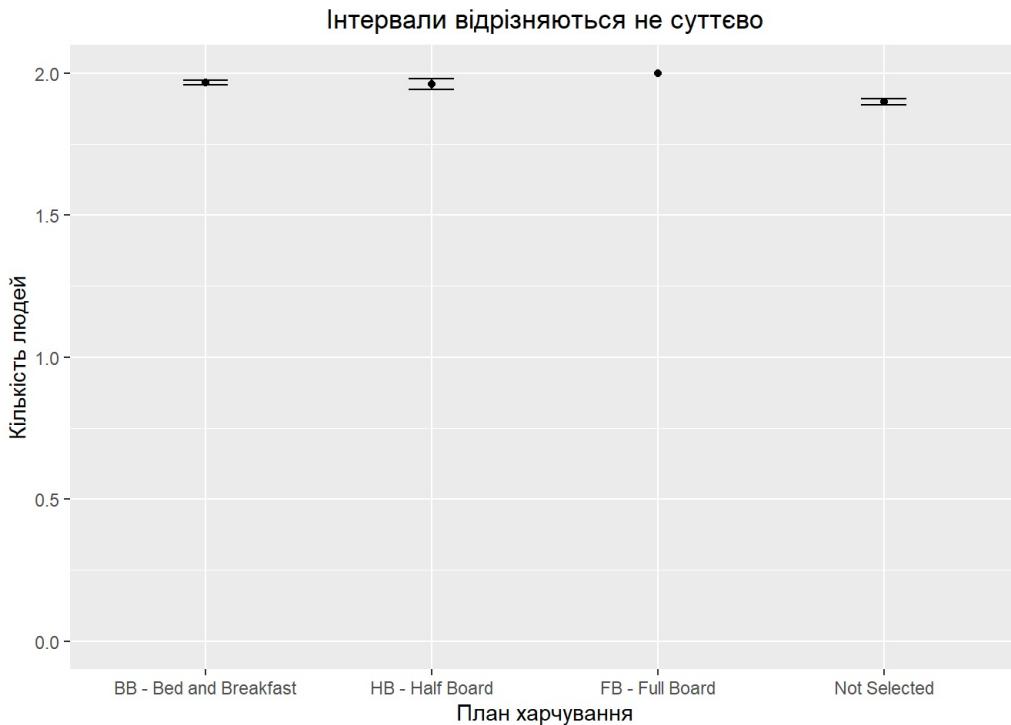
  cis[[current_meal_plan]] <- ci
}
ci <- hotel %>%
  filter(type_of_meal_plan == "Not Selected") %>%
  summarize(mean = mean(no_of_people),
            sd = sd(no_of_people),
            n = n(),
            a = mean(no_of_people) + qnorm(0.025) * sd(no_of_people) / sqrt(n()),
            b = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))

cis[["Not Selected"]] <- ci

ci_df <- bind_rows(cis, .id = "Meal_plan")

# Plot confidence intervals
ggplot(ci_df, aes(x = Meal_plan, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "План харчування", y = "Кількість людей", title = "Інтервали відрізняються не суттєво") +
  ylim(0, NA) +
  scale_x_discrete(labels = meal_label_vector) +
  theme(axis.text.x = element_text(angle = , hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



В цілому нічого нового спостерігати ми не можемо, дані результати є аналогом графіку для дорослих лише трохи більші показники за рахунок невеликих додаткових значень від дітей, незмінним залишається тільки Meal Plan 3 за рахунок того що там 0 інтервал для дітей

Тип плану харчування суттєво не впливає на середню кількість дорослих або дітей, що бронюють номери у готелі. Це вказує на те, що при виборі типу плану харчування клієнти готелю орієнтується на інші фактори, а не на кількість осіб у своїй групі.

Мноожинне тестування

Код виконує мноожинне парне порівняння кількості дорослих між різними типами номерів за допомогою t-тесту з непуленованим стандартним відхиленням. Кожне парне порівняння проводиться між двома групами (типами номерів). Використовується корекція Бенджаміні-Хохберга для p-значень, щоб контролювати рівень помилок типу I при мноожинному тестуванні.

Умова

Цей код проводить парні t-тести для різних змінних у залежності від типу заброньованого номера. Цей t-тест перевіряє, чи є статистично значуща різниця у кількості дорослих між різними типами номерів.

```
pairwise.t.test(hotel$no_of_adults, factor(hotel$room_type_reserved),  
p.adjust.method = "BH", pool.sd = FALSE)  
  
##  
## Pairwise comparisons using t tests with non-pooled SD  
##  
## data: hotel$no_of_adults and factor(hotel$room_type_reserved)  
##  
##          Room_Type 1 Room_Type 2 Room_Type 3 Room_Type 4 Room_Type 5  
## Room_Type 2 5.1e-08   -      -      -      -  
## Room_Type 3 0.9127    0.4493   -      -      -  
## Room_Type 4 < 2e-16   < 2e-16  0.1536   -      -  
## Room_Type 5 0.7136    0.0057  0.8853  < 2e-16   -  
## Room_Type 6 < 2e-16   < 2e-16  0.4493  < 2e-16  9.8e-08  
## Room_Type 7 1.5e-14   < 2e-16  0.0529  0.0085  4.2e-13  
##          Room_Type 6  
## Room_Type 2  -  
## Room_Type 3  -  
## Room_Type 4  -  
## Room_Type 5  -  
## Room_Type 6  -  
## Room_Type 7 1.8e-07  
##  
## P value adjustment method: BH
```

Результати

- R 1 і R 2: p-значення 5.1e-08, що значно менше 0.05, вказує на статистично значущу різницю в кількості дорослих між цими типами номерів.
- R 1 і R 3: p-значення 0.9127, що значно більше 0.05, вказує на відсутність статистично значущої різниці.
- R 1 і R 4: p-значення < 2e-16, що вказує на значущу різницю.
- R 1 і R 6: p-значення < 2e-16, що вказує на значущу різницю.

Наступний t-тест перевіряє, чи є статистично значуща різниця у кількості дітей між різними типами номерів.

```
pairwise.t.test(hotel$no_of_children, factor(hotel$room_type_reserved),  
p.adjust.method = "BH", pool.sd = FALSE)
```

```
##  
## Pairwise comparisons using t tests with non-pooled SD  
##  
## data: hotel$no_of_children and factor(hotel$room_type_reserved)  
##  
##          Room_Type 1 Room_Type 2 Room_Type 3 Room_Type 4 Room_Type 5  
## Room_Type 2 < 2e-16   -      -      -      -  
## Room_Type 3 < 2e-16   < 2e-16  -      -      -  
## Room_Type 4 3.3e-06   < 2e-16  < 2e-16  -      -  
## Room_Type 5 0.0047   < 2e-16  7.3e-06  0.0011  -  
## Room_Type 6 < 2e-16   < 2e-16  < 2e-16  < 2e-16  < 2e-16  
## Room_Type 7 < 2e-16  2.6e-05  < 2e-16  < 2e-16  1.4e-14  
##          Room_Type 6  
## Room_Type 2  -  
## Room_Type 3  -  
## Room_Type 4  -  
## Room_Type 5  -  
## Room_Type 6  -  
## Room_Type 7 3.4e-11  
##  
## P value adjustment method: BH
```

Аналогічно, t-тест для кількості дітей покаже, чи є статистично значущі відмінності між типами номерів щодо кількості дітей, які бронюють номери.

Цей t-тест перевіряє, чи є статистично значуща різниця у загальній кількості людей (дорослих і дітей) між різними типами номерів.

```
pairwise.t.test(hotel$no_of_people, factor(hotel$room_type_reserved),  
p.adjust.method = "BH", pool.sd = FALSE)
```

```

## 
##  Pairwise comparisons using t tests with non-pooled SD
##
## data: hotel$no_of_people and factor(hotel$room_type_reserved)
##
##          Room_Type 1 Room_Type 2 Room_Type 3 Room_Type 4 Room_Type 5
## Room_Type 2 < 2e-16   -      -      -      -
## Room_Type 3 0.90231  0.17804  -      -      -
## Room_Type 4 < 2e-16  0.00087  0.11797  -      -
## Room_Type 5 0.29180  1.3e-05  0.72928  1.5e-09  -
## Room_Type 6 < 2e-16  < 2e-16  0.00087  < 2e-16  < 2e-16
## Room_Type 7 < 2e-16  < 2e-16  0.00069  < 2e-16  < 2e-16
##          Room_Type 6
## Room_Type 2 -
## Room_Type 3 -
## Room_Type 4 -
## Room_Type 5 -
## Room_Type 6 -
## Room_Type 7 0.00091
##
## P value adjustment method: BH

```

Результати t-тесту для загальної кількості людей покажуть, чи є значущі відмінності у кількості людей (включаючи дорослих і дітей), які бронюють різні типи номерів.

Цей t-тест перевіряє, чи є статистично значуща різниця у середній ціні за номер між різними типами номерів.

```

pairwise.t.test(hotel$avg_price_per_room, factor(hotel$room_type_reserved),
p.adjust.method = "BH", pool.sd = FALSE)

```

```

## 
##  Pairwise comparisons using t tests with non-pooled SD
##
## data: hotel$avg_price_per_room and factor(hotel$room_type_reserved)
##
##          Room_Type 1 Room_Type 2 Room_Type 3 Room_Type 4 Room_Type 5
## Room_Type 2 4.1e-05   -      -      -      -
## Room_Type 3 0.68270  0.49608  -      -      -
## Room_Type 4 < 2e-16  < 2e-16  0.16471  -      -
## Room_Type 5 < 2e-16  < 2e-16  0.10110  0.01586  -
## Room_Type 6 < 2e-16  < 2e-16  0.00419  < 2e-16  < 2e-16
## Room_Type 7 < 2e-16  < 2e-16  0.00092  < 2e-16  < 2e-16
##          Room_Type 6
## Room_Type 2 -
## Room_Type 3 -
## Room_Type 4 -
## Room_Type 5 -
## Room_Type 6 -
## Room_Type 7 2.6e-07
##
## P value adjustment method: BH

```

T-тест для середньої ціни за номер виявить, чи є значущі відмінності у цінах між різними типами номерів. Це може вказувати на те, які типи номерів є дорожчими або дешевшими у порівнянні з іншими.

- Rt 1 і Rt 2: p-значення 2.4e-05, що значно менше 0.05, вказує на статистично значущу різницю в середній ціні за номер між цими типами номерів.
- Rt 1 і Rt 3: p-значення 0.68270, що значно більше 0.05, вказує на відсутність статистично значущої різниці.

Існують статистично значущі відмінності в середній ціні за номер між багатьма типами номерів.

Деякі типи номерів не мають значущих відмінностей у середній ціні за номер, що може вказувати на схожість в їхній ціновій політиці.

2. Чи є істотною наявність особливих побажань?

Побудуємо довірчі інтервали для середнього числа особливих побажань для кожного типу номеру. Слід зауважити, що навідміну від першої лабораторної роботи, ми рахуємо не кількість особливих бажань для кожного запису, а лише чи особливі бажання "були" (тобто 1) або "не були" (тобто 0).

```

ci_special_requests <- hotel %>%
  group_by(room_type_reserved) %>%
  summarize(mean_requests = mean(no_of_special_requests),
            sd_requests = sd(no_of_special_requests),
            n_requests = n(),
            ci_low_requests = mean(no_of_special_requests) + qnorm(0.025) * sd(no_of_special_requests) / sqrt(n()),
            ci_high_requests = mean(no_of_special_requests) + qnorm(0.975) * sd(no_of_special_requests) / sqrt(n())
  ),
  print(ci_special_requests)

```

```

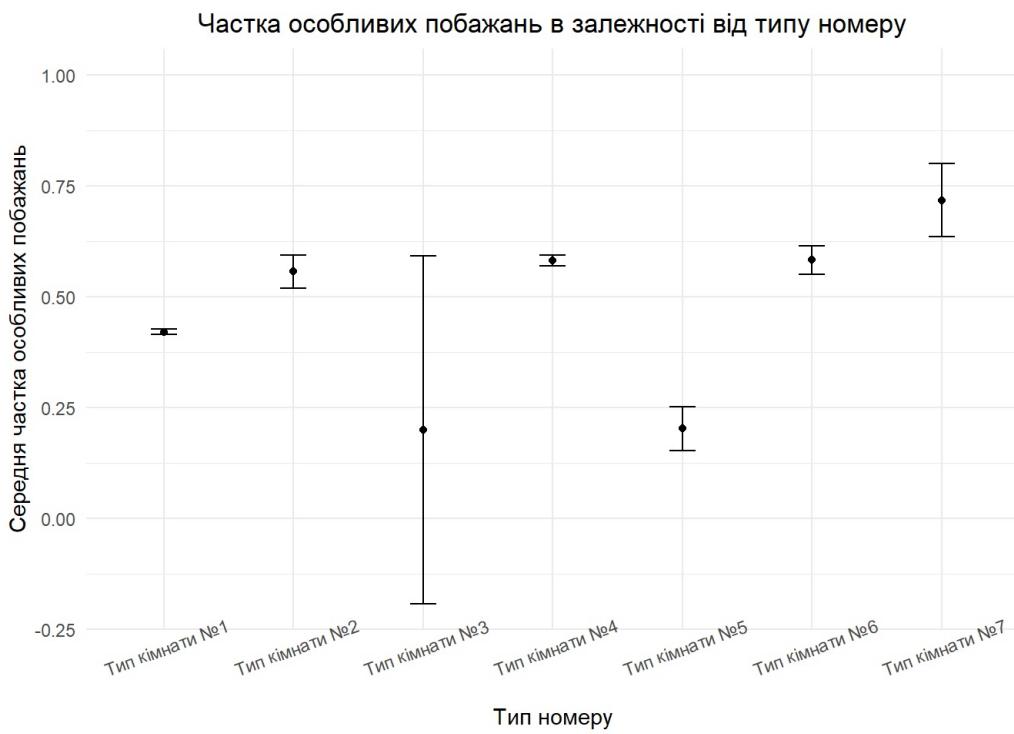
## # A tibble: 7 × 6
##   room_type_reserved mean_requests sd_requests n_requests ci_low_requests
##   <fct>                <dbl>        <dbl>      <int>        <dbl>
## 1 Room_Type 1         0.421        0.494     27707       0.415
## 2 Room_Type 2         0.557        0.497      654        0.518
## 3 Room_Type 3         0.2          0.447       5        -0.192
## 4 Room_Type 4         0.582        0.493     5995        0.569
## 5 Room_Type 5         0.202        0.403     247        0.152
## 6 Room_Type 6         0.583        0.493     949        0.551
## 7 Room_Type 7         0.718        0.452     117        0.636
## # i 1 more variable: ci_high_requests <dbl>

```

```

ggplot(ci_special_requests, aes(x = room_type_reserved, y = mean_requests)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_requests, ymax = ci_high_requests), width = 0.2) +
  labs(title = "Частка особливих побажань в залежності від типу номеру", x = "Тип номеру", y = "Середня частка особливих побажань") +
  theme_minimal() + ylim(NaN, 1) + scale_x_discrete(label = room_label_vector) +
  theme(axis.text.x = element_text(angle = 20, hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



Отже, можна сказати, що насправді довірчі інтервали ми будуємо для частки особливих бажань для кожного типу кімнати. Через це, наші середні значення можуть бути лише в інтервалі [0;1]. Таким чином, якщо середнє значення, наприклад, 0.7 це означає що в середньостатистичному записі для цього типу кімнат люди мали якісь особливі побажання. На основі отриманих інтервалів побудуємо графік, на вісі абсцис позначимо типи кімнат, а на вісі ординат середнє значення статусу особливих бажань (частку). За графіком можемо побачити, що для найпопулярнішого типу кімнат (першого) приблизно 42% (меншість) записів мали додаткові бажання. Також можемо помітити, що для типів кімнат 2, 4 і 6, більшість людей брали особливі бажання, навіть враховуючи довірчі інтервали (для кожного з цих типів початок інтервалу не опускається менше 0.518). Типи кімнат 3, 5 і 7, мали дуже малу кількість записів і відповідно їх довірчі інтервали досить великі, для типів 5 і 7 великий розмір інтервалів не впливає сильно впливає на тенденцію (для 5 типу бажання мала меншість а для 7 більшість), а для типу 3 довірчий інтервал сильно впливає на тенденцію, але враховуючи що для третього типу було всього 5 записів, цим типом кімнат можна знехтувати.

інтервал дисперсія

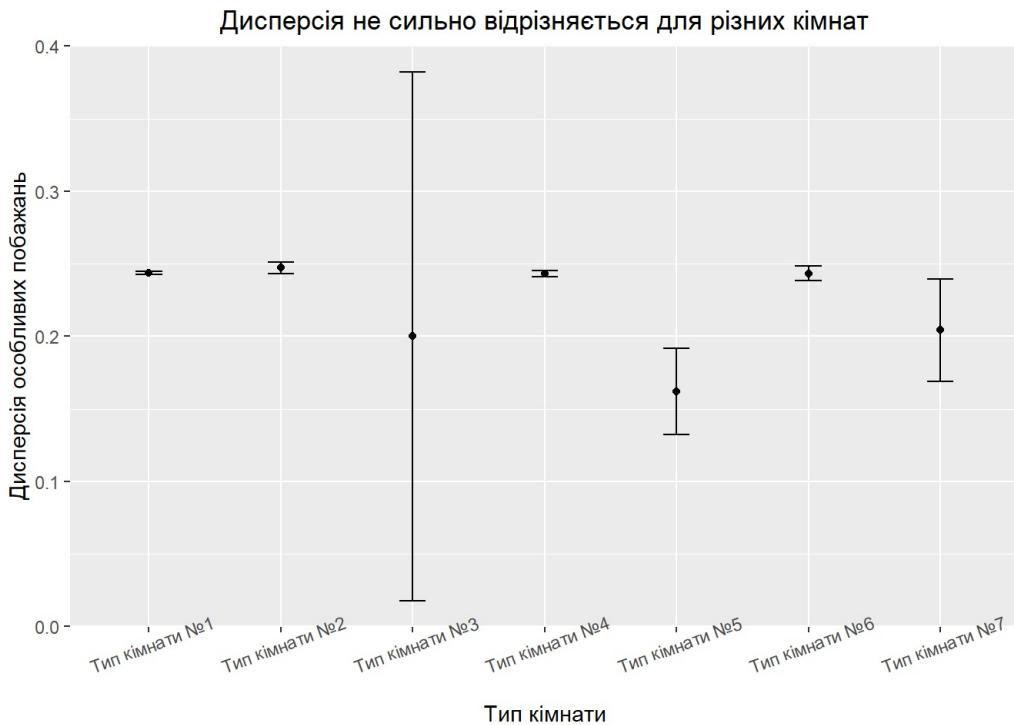
```

cis <- list()
for (i in 1:7) {
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(
      n = n(),
      mean = mean(no_of_special_requests),
      var = var(no_of_special_requests),
      fourth_moment = mean((no_of_special_requests - mean)^4),
      sd_var = sqrt((fourth_moment - var^2) / n),
      a = var - qnorm(0.975) * sd_var,
      b = var + qnorm(0.975) * sd_var
    )
  cis[[current_room_type]] <- ci
}

ci_df <- bind_rows(cis, .id = "Room_Type")

ggplot(ci_df, aes(x = Room_Type, y = var)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип кімнати", y = "Дисперсія особливих побажань", title = "Дисперсія не сильно відрізняється для різних кімнат") + scale_x_discrete(label = room_label_vector) +
  theme(axis.text.x = element_text(angle = 20, hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



Побудуємо довірчі інтервали для середнього значення часу до прибуття, в залежності від наявності особливих бажань. Нанесемо їх на графік, позначимо по іксам групи людей по особливим бажанням, а по ігрикам середні значення часу до прибуття (в годинах) і навколо них отримані довірчі інтервали.

```

ci_special_requests <- hotel %>%
  group_by(no_of_special_requests) %>%
  summarize(mean_lead_time = mean(lead_time),
            sd_lead_time = sd(lead_time),
            n_requests = n(),
            ci_low_requests = mean(lead_time) + qnorm(0.025) * sd(lead_time) / sqrt(n()),
            ci_high_requests = mean(lead_time) + qnorm(0.975) * sd(lead_time) / sqrt(n()))

print(ci_special_requests)

```

```

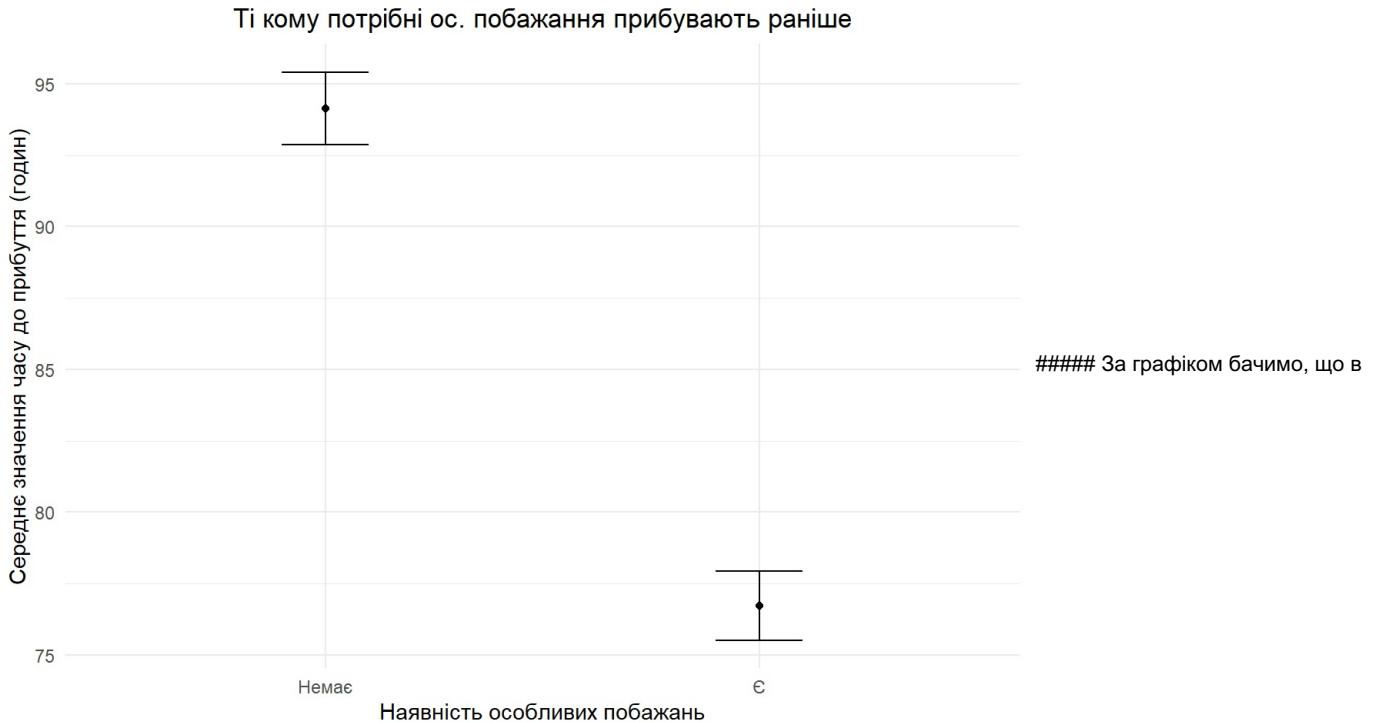
## # A tibble: 2 × 6
##   no_of_special_requests  mean_lead_time  sd_lead_time  n_requests  ci_low_requests  ci_high_requests
##   <dbl>              <dbl>          <dbl>       <int>        <dbl>
## 1 0                  94.1           91.0     19469        92.9
## 2 1                  76.7           78.4     16205        75.5
## # i 1 more variable: ci_high_requests <dbl>

```

```

ggplot(ci_special_requests, aes(x = as.factor(no_of_special_requests), y = mean_lead_time)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_requests, ymax = ci_high_requests), width = 0.2) +
  labs(title = "Ті кому потрібні ос. побажання прибувають раніше", x = "Наявність особливих побажань", y = "Середнє значення часу до прибуття (годин)" ) +
  theme_minimal() + scale_x_discrete(label = c("0" = "Немає", "1" = "Є")) + theme(axis.text.x = element_text(angle = , hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



середньому люди у яких були особливі побажання прибували до готелю раніше ніж люди без побажань, на приблизно 18 годин. У таблиці спостерігаємо, що стандартне відхилення досить велике, але за допомогою довірчих інтервалів стає зрозуміло, що в дійсності більшість записів були досить близько до середніх значень часу прибуття.

Побудуємо тепер довірчі інтервали для середньої кількості людей у кімнаті, в залежності від наявності особливих побажань. Нанесемо це на графік.

```

ci_special_requests <- hotel %>%
  group_by(no_of_special_requests) %>%
  summarize(mean_no_of_people = mean(no_of_people),
            sd_no_of_people = sd(no_of_people),
            n_requests = n(),
            ci_low_requests = mean(no_of_people) + qnorm(0.025) * sd(no_of_people) / sqrt(n()),
            ci_high_requests = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))

print(ci_special_requests)

```

```

## # A tibble: 2 × 6
##   no_of_special_requests mean_no_of_people sd_no_of_people n_requests
##             <dbl>                <dbl>           <dbl>      <int>
## 1                 0                  1.84          0.627     19469
## 2                 1                  2.10          0.631     16205
## # i 2 more variables: ci_low_requests <dbl>, ci_high_requests <dbl>

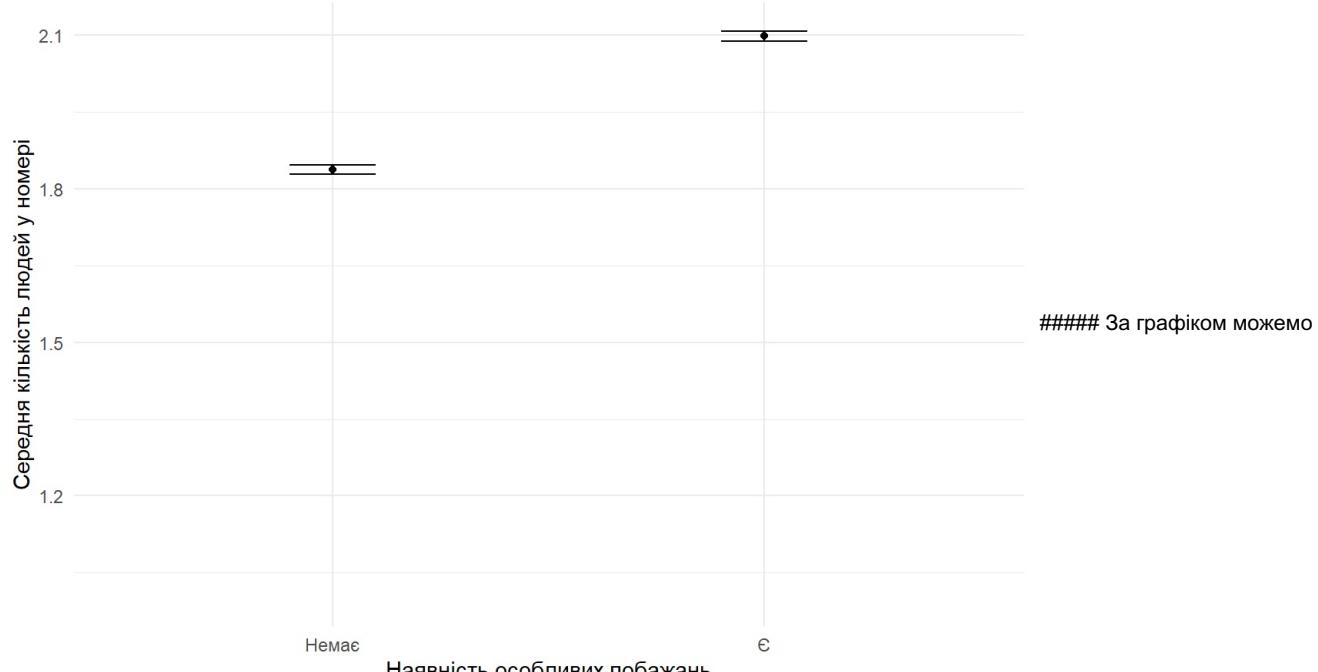
```

```

ggplot(ci_special_requests, aes(x = as.factor(no_of_special_requests), y = mean_no_of_people)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_requests, ymax = ci_high_requests), width = 0.2) +
  labs(title = "Дешо більше людей там де є ос. побажання", x = "Наявність особливих побажань", y = "Середня кількість людей у номері") +
  theme_minimal() + ylim(1, NaN)+ scale_x_discrete(label = c("0" = "Немає", "1" = "Є")) + theme(axis.text.x = element_text(angle = , hjust = 0.5), plot.title = element_text(hjust = 0.5))

```

Дещо більше людей там де є ос. побажання



припустити, що якщо особливі бажання були, то в середньому заселялось більше людей, приблизно на 0.3 людини. Перевіримо це за допомогою відповідної гіпотези.

Нульова гіпотеза: "У записах в яких не потрібні і потрібні особливі бажання, в середньому заселялась однакова кількість людей"

```
estimates <- hotel %>%
  group_by(no_of_special_requests) %>%
  summarise(mean_hat = mean(no_of_people),
            var_hat = var(no_of_people) / n())

mean_hat_req <- estimates %>% filter(no_of_special_requests == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(no_of_special_requests == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(no_of_special_requests == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(no_of_special_requests == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- 2 * pnorm(abs(T), lower.tail = FALSE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Requests):", mean_hat_req, "\n")

## Mean (Requests): 2.098858

cat("Mean (No Requests):", mean_hat_no_req, "\n")

## Mean (No Requests): 1.837639

cat("T-statistic:", T, "\n")

## T-statistic: -39.02298

cat("P-value:", p_value, "\n")

## P-value: 0

cat("95% Confidence Interval:", conf.int, "\n")

## 95% Confidence Interval: -0.274339 -0.2480991
```

```
t.test(no_of_people ~ no_of_special_requests, data = hotel, alternative = "two.sided")
```

```
##  
## Welch Two Sample t-test  
##  
## data: no_of_people by no_of_special_requests  
## t = -39.023, df = 34425, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0  
## 95 percent confidence interval:  
## -0.2743395 -0.2480986  
## sample estimates:  
## mean in group 0 mean in group 1  
## 1.837639 2.098858
```

Бачимо, що p_value вийшло дуже малим, отже ми не маємо підстав не відхилити гіпотезу H_0 , і маємо підстави не відхилити H_1 , яка каже про те, що в залежності від наявності особливих бажань, кількість людей у номері відрізняється. Справді бачимо, що відрізняється приблизно на 0.3

Протестуємо гіпотезу про те, що люди, яким не потрібні і потрібні спец запити ночують в середньому однакову кількість ночей.

```
estimates <- hotel %>%  
  group_by(no_of_special_requests) %>%  
  summarise(mean_hat = mean(no_of_nights),  
           var_hat = var(no_of_nights) / n())  
  
mean_hat_req <- estimates %>% filter(no_of_special_requests == 1) %>% pull(mean_hat)  
mean_hat_no_req <- estimates %>% filter(no_of_special_requests == 0) %>% pull(mean_hat)  
var_hat_req <- estimates %>% filter(no_of_special_requests == 1) %>% pull(var_hat)  
var_hat_no_req <- estimates %>% filter(no_of_special_requests == 0) %>% pull(var_hat)  
  
se <- sqrt(var_hat_req + var_hat_no_req)  
  
T <- (mean_hat_no_req - mean_hat_req) / se  
  
p_value <- 2 * pnorm(abs(T), lower.tail = FALSE)  
  
conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)  
  
cat("Mean (Requests):", mean_hat_req, "\n")
```

```
## Mean (Requests): 3.166739
```

```
cat("Mean (No Requests):", mean_hat_no_req, "\n")
```

```
## Mean (No Requests): 2.934203
```

```
cat("T-statistic:", T, "\n")
```

```
## T-statistic: -12.25289
```

```
cat("P-value:", p_value, "\n")
```

```
## P-value: 1.621207e-34
```

```
cat("95% Confidence Interval:", conf.int, "\n")
```

```
## 95% Confidence Interval: -0.2697318 -0.1953393
```

```
#Люди, яким не потрібні і потрібні спец запити ночують в середньому однакову кількість ночей  
t.test(no_of_nights ~ no_of_special_requests, data = hotel, alternative = "two.sided")
```

```

## Welch Two Sample t-test
##
## data: no_of_nights by no_of_special_requests
## t = -12.253, df = 34006, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -0.2697331 -0.1953380
## sample estimates:
## mean in group 0 mean in group 1
## 2.934203      3.166739

```

```

ci <- hotel %>%
  filter(no_of_special_requests == 0) %>%
  summarise(mean = mean(no_of_nights),
            sd = sd(no_of_nights),
            n = n(),
            a = mean(no_of_nights) + qnorm(0.025) * sd(no_of_nights) / sqrt(n()),
            b = mean(no_of_nights) + qnorm(0.975) * sd(no_of_nights) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 2.934203 1.74762 19469 2.909655 2.958752

```

```

ci <- hotel %>%
  filter(no_of_special_requests == 1) %>%
  summarise(mean = mean(no_of_nights),
            sd = sd(no_of_nights),
            n = n(),
            a = mean(no_of_nights) + qnorm(0.025) * sd(no_of_nights) / sqrt(n()),
            b = mean(no_of_nights) + qnorm(0.975) * sd(no_of_nights) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 3.166739 1.815031 16205 3.138793 3.194684

```

Бачимо, що p-value вийшло дуже малим, отже ми відхиляємо гіпотезу H_0 , і приймаємо альтернативну гіпотезу H_1 , тобто таку, яка каже, що люди з і без побажань ночують в середньому різну кількість ночей. По отриманим середнім значенням дійсно бачимо, що група людей у яких є особливі побажання в середньому ночують на приблизно 0.2 ночі більше, ніж група людей без побажань.

Тепер протестуємо гіпотезу про те, що люди, яким НЕ потрібні спеціальні запити, в середньому платять більше.

```

estimates <- hotel %>%
  group_by(no_of_special_requests) %>%
  summarise(mean_hat = mean(avg_price_per_room),
            var_hat = var(avg_price_per_room) / n())

mean_hat_req <- estimates %>% filter(no_of_special_requests == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(no_of_special_requests == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(no_of_special_requests == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(no_of_special_requests == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- pnorm(T, lower.tail = TRUE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Requests):", mean_hat_req, "\n")

## Mean (Requests): 111.3596

cat("Mean (No Requests):", mean_hat_no_req, "\n")

## Mean (No Requests): 99.98524

cat("T-statistic:", T, "\n")

```

```
## T-statistic: -32.9086
```

```
cat("P-value:", p_value, "\n")
```

```
## P-value: 8.278696e-238
```

```
cat("95% Confidence Interval:", conf.int, "\n")
```

```
## 95% Confidence Interval: -12.05183 -10.69696
```

```
# Люди, яким не потрібні спеціальні запити платять більше  
t.test(avg_price_per_room ~ no_of_special_requests, data = hotel, alternative = "less")
```

```
##  
## Welch Two Sample t-test  
##  
## data: avg_price_per_room by no_of_special_requests  
## t = -32.909, df = 33178, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 0 and group 1 is less than 0  
## 95 percent confidence interval:  
##      -Inf -10.80586  
## sample estimates:  
## mean in group 0 mean in group 1  
##      99.98524     111.35964
```

```
ci <- hotel %>%  
  filter(no_of_special_requests == 0) %>%  
  summarize(mean = mean(avg_price_per_room),  
           sd = sd(avg_price_per_room),  
           n = n(),  
           a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n()),  
           b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))  
ci
```

```
##      mean      sd      n      a      b  
## 1 99.98524 30.85209 19469 99.55187 100.4186
```

```
ci <- hotel %>%  
  filter(no_of_special_requests == 1) %>%  
  summarize(mean = mean(avg_price_per_room),  
           sd = sd(avg_price_per_room),  
           n = n(),  
           a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n()),  
           b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))  
ci
```

```
##      mean      sd      n      a      b  
## 1 111.3596 33.81783 16205 110.839 111.8803
```

Бачимо, що знову p-value вийшло дуже малим, що означає, що ми не маємо підстав не відхилити нульову гіпотезу і відповідно можемо припускати, що люди, яким потрібні спеціальні запити, в середньому платять більше. За середніми значеннями бачимо, що група людей яким потрібні спеціальні запити в середньому платять на 11\$ більше ніж люди яким спеціальні запити не потрібні.

3. Які характерні риси скасованих записів?

Порахуємо частку скасованих записів, що приходиться на кожну кількість дорослих, вказаних у бронюванні. Отриманий результат зображеній на графіку нижче:

— Довірчі інтервали для часток скасувань залежно від кількості дорослих:

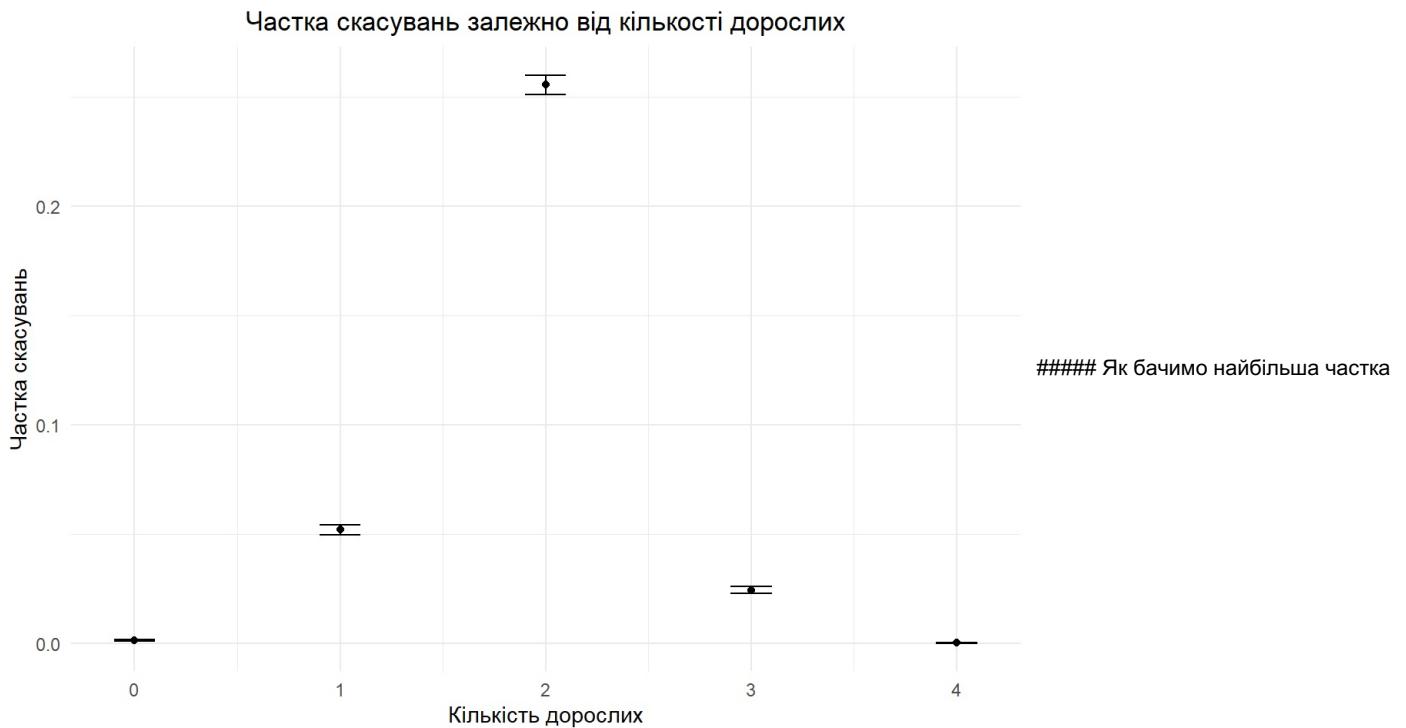
```

ci_adults <- hotel %>%
  group_by(no_of_adults, booking_status) %>%
  summarize(count = n(), .groups = 'drop') %>%
  mutate(total = sum(count)) %>%
  mutate(proportion = count / total) %>%
  filter(booking_status == "Canceled") %>%
  mutate(ci_low = proportion - qnorm(0.975) * sqrt(proportion * (1 - proportion) / total),
         ci_high = proportion + qnorm(0.975) * sqrt(proportion * (1 - proportion) / total))

# print(ci_adults)

ggplot(ci_adults, aes(x = no_of_adults, y = proportion)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low, ymax = ci_high), width = 0.2) +
  labs(title = "Частка скасувань залежно від кількості дорослих",
       x = "Кількість дорослих", y = "Частка скасувань") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

```



приходиться на бронювання з вказаною кількістю у 2 дорослих. Загалом картина досить очікувана, адже було б природньо сподіватися, що кількість дорослих, що бронюють готель, буде мати розподіл, що віддалено нагадує нормальній.

Як відомо, у датасеті загалом ~36k записів. Переглянемо яка кількість записів відповідає значенню no_of_adults = 2:

```
nrow(hotel %>% filter(no_of_adults == 2))
```

```
## [1] 25857
```

Спостережувана кількість лише підтверджує попередні міркування.

дисперсія по кількості дітей відносно статусу бронювання

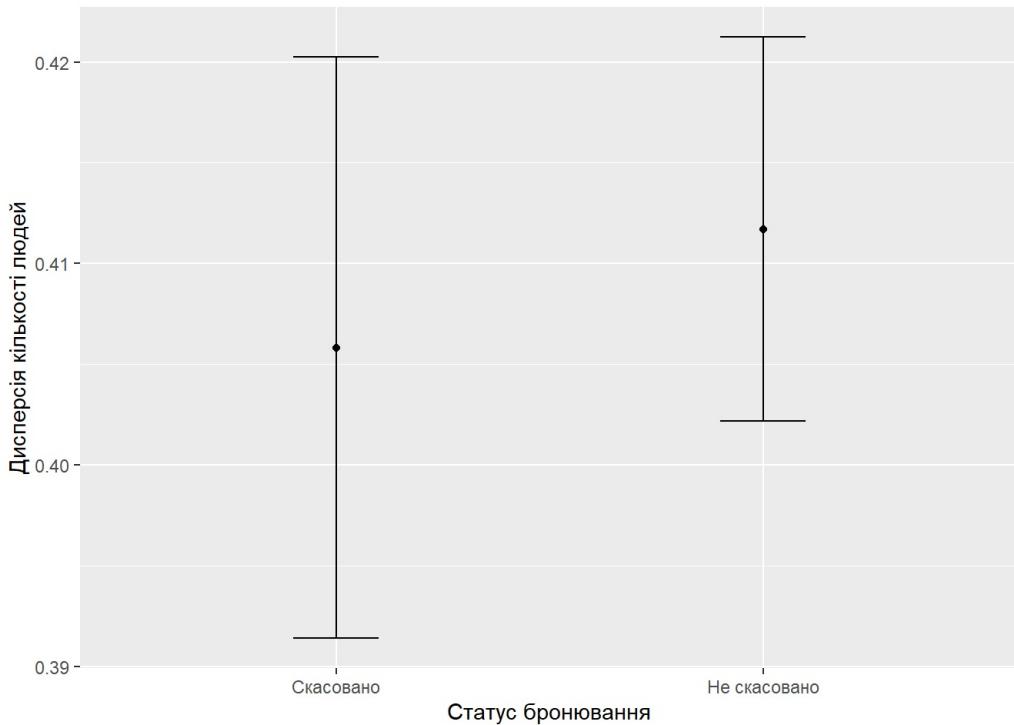
```

cis <- list()
for (current in c("Canceled", "Not_Canceled")) {
  ci <- hotel %>%
    filter(booking_status == current) %>%
    summarize(
      n = n(),
      mean = mean(no_of_people),
      var = var(no_of_people),
      fourth_moment = mean((no_of_people - mean)^4),
      sd_var = sqrt((fourth_moment - var^2) / n),
      a = var - qnorm(0.975) * sd_var,
      b = var + qnorm(0.975) * sd_var
    )
  cis[[current]] <- ci
}

ci_df <- bind_rows(cis, .id = "Booking_Status")

ggplot(ci_df, aes(x = Booking_Status, y = var)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Статус бронювання", y = "Дисперсія кількості людей") +
  scale_x_discrete(labels = c("Canceled" = "Скасовано", "Not_Canceled" = "Не скасовано"))+
  theme(axis.text.x = element_text(angle = , hjust = 0.5), plot.title = element_text(hjust = 0.5))

```



Перейдемо до часток скасувань, що приходяться безпосередньо на дітей:

```

ci_children <- hotel %>%
  group_by(no_of_children, booking_status) %>%
  summarize(count = n(), .groups = 'drop') %>%
  mutate(total = sum(count)) %>%
  mutate(proportion = count / total) %>%
  filter(booking_status == "Canceled") %>%
  mutate(ci_low = proportion - qnorm(0.975) * sqrt(proportion * (1 - proportion) / total),
         ci_high = proportion + qnorm(0.975) * sqrt(proportion * (1 - proportion) / total))

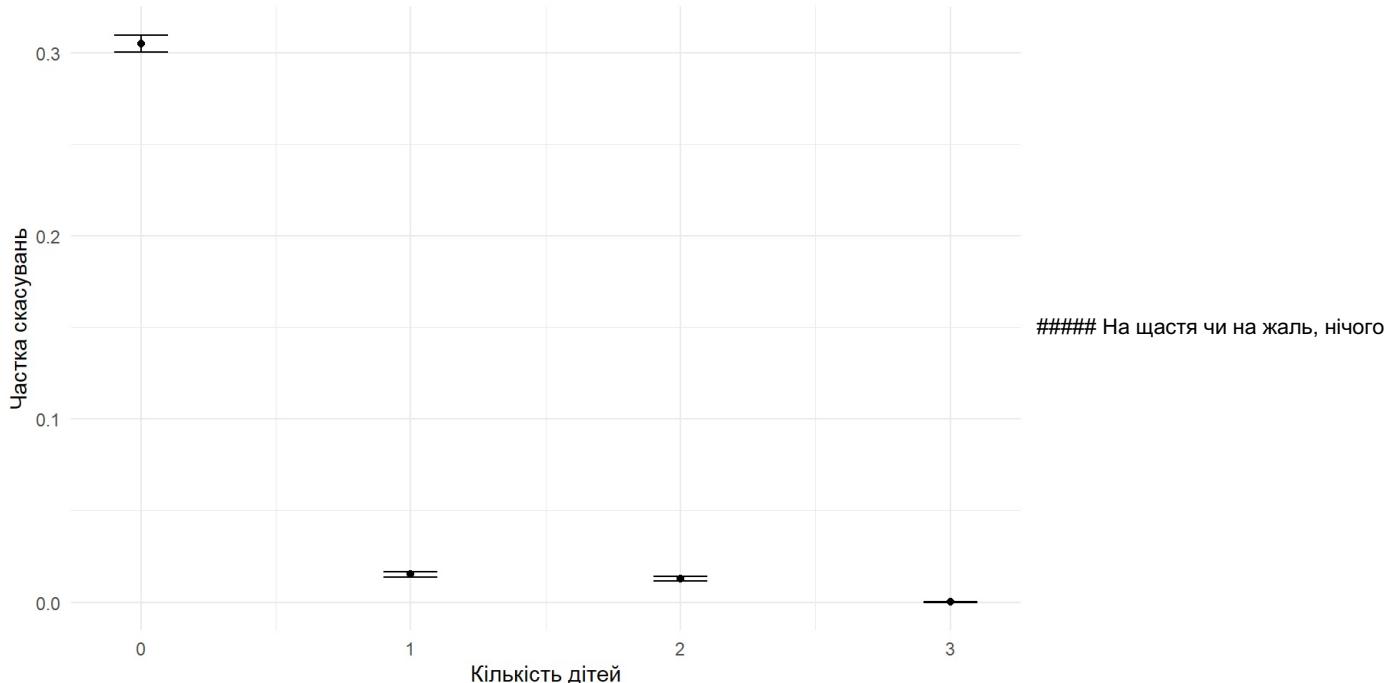
# print(ci_children)

ggplot(ci_children, aes(x = no_of_children, y = proportion)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low, ymax = ci_high), width = 0.2) +
  labs(title = "Більше скасувань без дітей - бо більше записів",
       x = "Кількість дітей", y = "Частка скасувань") +
  theme_minimal()

theme(plot.title = element_text(hjust = 0.5))

```

Більше скасувань без дітей - бо більше записів



цікавого помітти не вдається. Те, що записів без дітей значно більше ніж записів з конкретними кількостями дітей є природним і очікуваним.

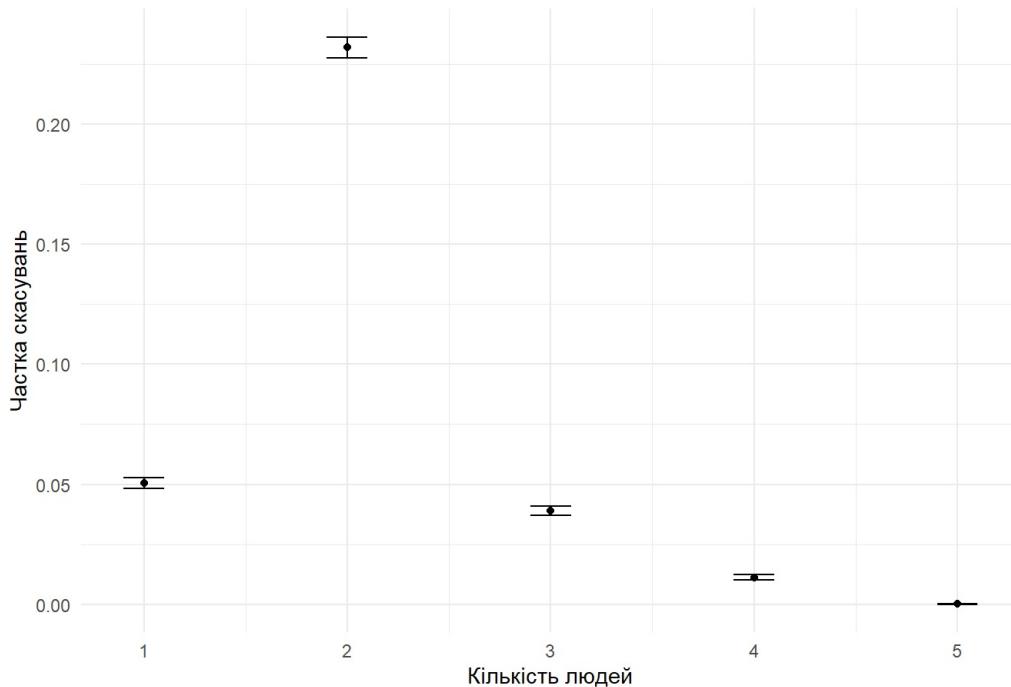
Підsumовуючи два отримані вище результати можемо побачити досить очікувану картину для частки скасувань по загальній кількості людей:

```
ci_people <- hotel %>%
  group_by(no_of_people, booking_status) %>%
  summarize(count = n(), .groups = 'drop') %>%
  mutate(total = sum(count)) %>%
  mutate(proportion = count / total) %>%
  filter(booking_status == "Canceled") %>%
  mutate(ci_low = proportion + qnorm(0.025) * sqrt(proportion * (1 - proportion) / total),
         ci_high = proportion + qnorm(0.975) * sqrt(proportion * (1 - proportion) / total))

# print(ci_children)

ggplot(ci_people, aes(x = no_of_people, y = proportion)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low, ymax = ci_high), width = 0.2) +
  labs(title = "Найбільше для 2 людей - бо найбільше записів",
       x = "Кількість людей", y = "Частка скасувань") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Найбільше для 2 людей - бо найбільше записів



Долові цікаво було би порівняти середню кількість людей у записах зі скасованими і нескасованими бронюваннями. Протестуємо відповідну гіпотезу про те, що не відмінені бронювання в середньому мають більше людей, ніж відмінені:

```
estimates <- hotel_reverse %>%
  group_by(booking_status_binary) %>%
  summarise(mean_hat = mean(no_of_people),
            var_hat = var(no_of_people) / n())

mean_hat_req <- estimates %>% filter(booking_status_binary == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(booking_status_binary == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(booking_status_binary == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(booking_status_binary == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- pnorm(T, lower.tail = TRUE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Requests):", mean_hat_req, "\n")

## Mean (Requests): 2.033594

cat("Mean (No Requests):", mean_hat_no_req, "\n")

## Mean (No Requests): 1.917721

cat("T-statistic:", T, "\n")

## T-statistic: -16.15134

cat("P-value:", p_value, "\n")

## P-value: 5.556227e-59

cat("95% Confidence Interval:", conf.int, "\n")

## 95% Confidence Interval: -0.1299349 -0.1018124

t.test(no_of_people ~ booking_status_binary, data = hotel_reverse, alternative = "less")

##
## Welch Two Sample t-test
##
## data: no_of_people by booking_status_binary
## t = -16.151, df = 23890, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is less than 0
## 95 percent confidence interval:
##       -Inf -0.1040726
## sample estimates:
## mean in group 0 mean in group 1
##      1.917721      2.033594

ci <- hotel_reverse %>%
  filter(booking_status == "Not_Canceled") %>%
  summarize(mean = mean(no_of_people),
            sd = sd(no_of_people),
            n = n(),
            a = mean(no_of_people) + qnorm(0.025) * sd(no_of_people) / sqrt(n()),
            b = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))

ci

##      mean        sd        n        a        b
## 1 1.917721 0.6416413 23797 1.909568 1.925873
```

```

ci <- hotel_reverse %>%
  filter(booking_status == "Canceled") %>%
  summarize(mean = mean(no_of_people),
            sd = sd(no_of_people),
            n = n(),
            a = mean(no_of_people) + qnorm(0.025) * sd(no_of_people) / sqrt(n()),
            b = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 2.033594 0.6370452 11877 2.022137 2.045051

```

Як бачимо p-value надзвичайно мале, тож припустити що нескасовані бронювання мають в середньому меншу кількість людей ніж скасовані бронювання, хоч ця різниця, як видно по довірчих інтервалах, є відносно незначною.

Протестуємо досить примітивну гіпотезу про те, що в середньому на нескасованих бронюваннях час до прибуття є більшим, ніж у скасованих (це може бути цікаво з тієї точки зору, що інтуїтивно можливо у тих, хто відміняє бронювання, відбуваються неочікувані зміни планів, а зазвичай бронювання роблять приблизно за 5-7 днів до прибуття як спланований візит на визначений термін):

```

estimates <- hotel_reverse %>%
  group_by(booking_status_binary) %>%
  summarise(mean_hat = mean(lead_time),
            var_hat = var(lead_time) / n())

mean_hat_req <- estimates %>% filter(booking_status_binary == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(booking_status_binary == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(booking_status_binary == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(booking_status_binary == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- pnorm(T, lower.tail = TRUE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Requests):", mean_hat_req, "\n")

```

```

## Mean (Requests): 139.1415

```

```

cat("Mean (No Requests):", mean_hat_no_req, "\n")

```

```

## Mean (No Requests): 59.81611

```

```

cat("T-statistic:", T, "\n")

```

```

## T-statistic: -79.48999

```

```

cat("P-value:", p_value, "\n")

```

```

## P-value: 0

```

```

cat("95% Confidence Interval:", conf.int, "\n")

```

```

## 95% Confidence Interval: -81.28133 -77.36952

```

```

#ті хто не відмінили прибувають пізніше ніж ті що відмінили
t.test(lead_time ~ booking_status_binary, data = hotel_reverse, alternative = "less")

```

```

## Welch Two Sample t-test
##
## data: lead_time by booking_status_binary
## t = -79.49, df = 16992, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is less than 0
## 95 percent confidence interval:
##      -Inf -77.68388
## sample estimates:
## mean in group 0 mean in group 1
##      59.81611     139.14153

```

```

ci <- hotel %>%
  filter(booking_status == "Not_Canceled") %>%
  summarize(mean = mean(lead_time),
            sd = sd(lead_time),
            n = n(),
            a = mean(lead_time) + qnorm(0.025) * sd(lead_time) / sqrt(n()),
            b = mean(lead_time) + qnorm(0.975) * sd(lead_time) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 59.81611 64.02325 23797 59.00267 60.62955

```

```

ci <- hotel %>%
  filter(booking_status == "Canceled") %>%
  summarize(mean = mean(lead_time),
            sd = sd(lead_time),
            n = n(),
            a = mean(lead_time) + qnorm(0.025) * sd(lead_time) / sqrt(n()),
            b = mean(lead_time) + qnorm(0.975) * sd(lead_time) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 139.1415 98.90445 11877 137.3628 140.9203

```

Виходячи зі значення p_value немає підстав не відхилити нульову гіпотезу (тобто, можемо сподіватися, що середній час до прибуття для нескасованих резервацій є меншим за середній час “до прибуття” для скасованих резервацій)

Розглянемо отриманий результат графічно:

Побудуємо довірчі інтервали для середньої кількості годин у lead_time (кількість годин, що пройшла від моменту бронювання до прибуття / відміни бронювання) в залежності від статусу бронювання.

```

cis <- list()

for (status in c("Canceled", "Not_Canceled")) {
  ci <- hotel %>%
    filter(booking_status == status) %>%
    summarize(mean = mean(lead_time),
              sd = sd(lead_time),
              n = n(),
              a = mean(lead_time) - qnorm(0.975) * sd(lead_time) / sqrt(n()),
              b = mean(lead_time) + qnorm(0.975) * sd(lead_time) / sqrt(n()))

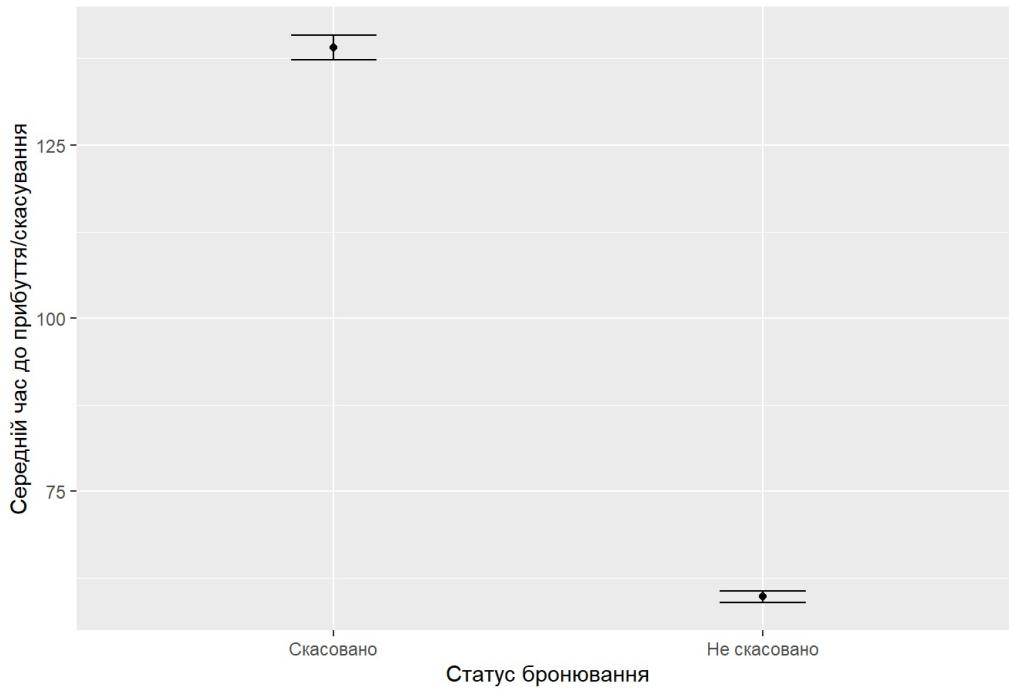
  cis[[status]] <- ci
}

ci_df <- bind_rows(cis, .id = "Booking_Status")

# Plot confidence intervals
ggplot(ci_df, aes(x = Booking_Status, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Статус бронювання", y = "Середній час до прибуття/скасування", title = "Скасовані мають більший час д о прибуття") +
  scale_x_discrete(labels = c("Canceled" = "Скасовано", "Not_Canceled" = "Не скасовано"))+
  theme(axis.text.x = element_text(angle = , hjust = 0.5), plot.title = element_text(hjust = 0.5))

```

Скасовані мають більший час до прибуття



Як бачимо, зазвичай люди, що скасовують резервації, роблять це через ~5-6 діб після бронювання. У тих, хто бронювання не відмінив, середня кількість годин до прибуття помітна нижча, що протирічить раніше висунутим інтуїтивним очікуванням.

4. Що впливає на кількість проведених ночей у готелі?

Було б досить цікаво подивитися на те як залежить ціна від кількості проведених ночей (загалом) у готелі. Чи дійсно відвідувачі не обирають дорогі кімнати, лишаючись на відносно тривалий час? Якою буде ціна для перших кількох проведених ночей? Чи проявляються бронювання, у яких відвідувачі захотіли "переночувати" чи "недешево відпочити"? Побудуємо довірчі інтервали для середніх значень цін за кімнати по кожному значенню проведених ночей у готелі:

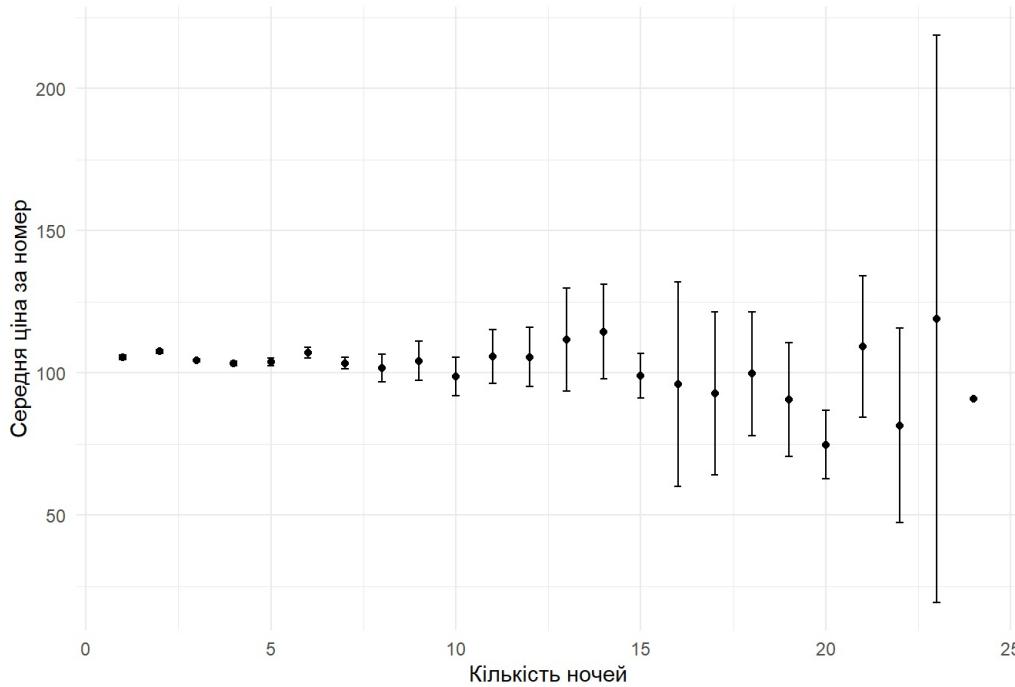
```
hotel <- hotel %>%
  mutate(no_of_nights = no_of_week_nights + no_of_weekend_nights)

ci_special_requests <- hotel %>%
  group_by(no_of_nights) %>%
  summarize(mean_price = mean(avg_price_per_room),
            sd_price = sd(avg_price_per_room),
            n_price = n(),
            ci_low_price = mean(avg_price_per_room) - qnorm(0.975) * sd(avg_price_per_room) / sqrt(n_price),
            ci_high_price = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n_price),
            .groups = 'drop') # This will suppress the message

# print(ci_special_requests)

ggplot(ci_special_requests, aes(x = no_of_nights, y = mean_price)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_price, ymax = ci_high_price), width = 0.2) +
  labs(title = "Дуже маленька тенденція на зменшення ціни", x = "Кількість ночей", y = "Середня ціна за номер") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Дуже маленька тенденція на зменшення ціни



Як видно з малюнку, в очі нічого не кидається. Свого роду аномальність для останніх двох значень ночей пояснюється тим, що 23 проведеним ночам відповідає всього лише 2 записи, а 24 ночам - 1 запис. Загалом отримані значення досить непогано апроксимувалися б горизонтальною прямою, тож явного додатнього/від'ємного зв'язку між ціною і кількістю ночей не спостерігається.

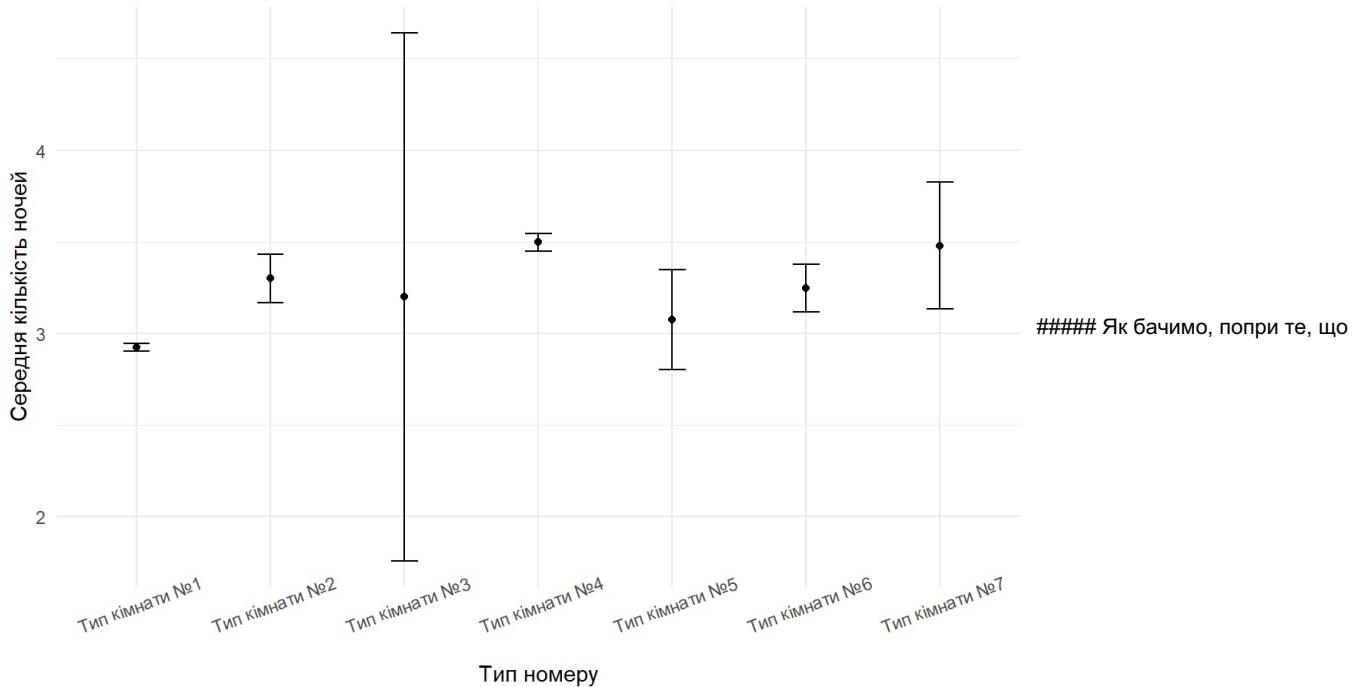
Побудуємо довірчі інтервали для середніх значень кількості ночей по кожному типу кімнати:

```
ci_nights_room <- hotel %>%
  group_by(room_type_reserved) %>%
  summarize(mean_nights = mean(no_of_nights),
            sd_nights = sd(no_of_nights),
            n_nights = n(),
            ci_low_nights = mean(no_of_nights) + qnorm(0.025) * sd(no_of_nights) / sqrt(n_nights),
            ci_high_nights = mean(no_of_nights) + qnorm(0.975) * sd(no_of_nights) / sqrt(n_nights))

# print(ci_nights_room)

ggplot(ci_nights_room, aes(x = room_type_reserved, y = mean_nights)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_nights, ymax = ci_high_nights), width = 0.2) +
  labs(title = "Немає значних тенденцій", x = "Тип номеру", y = "Середня кількість ночей") +
  theme_minimal() + scale_x_discrete(label = room_label_vector) + theme(axis.text.x = element_text(angle = 20, hjust = 0.5), plot.title = element_text(hjust = 0.5))
```

Немає значних тенденцій



попарно інтервали знаходяться не зовсім на одному рівні, в середньому різниця у проведених ночах навіть у добу не те щоб є значною. Для 3 типу кімнати замало записів, тому не представляється можливим робити по ньому будь-які судження.

Для цікавості можемо розглянути довірчий інтервал для дисперсії по кількості проведених ночей відносно кожного типу кімнати.

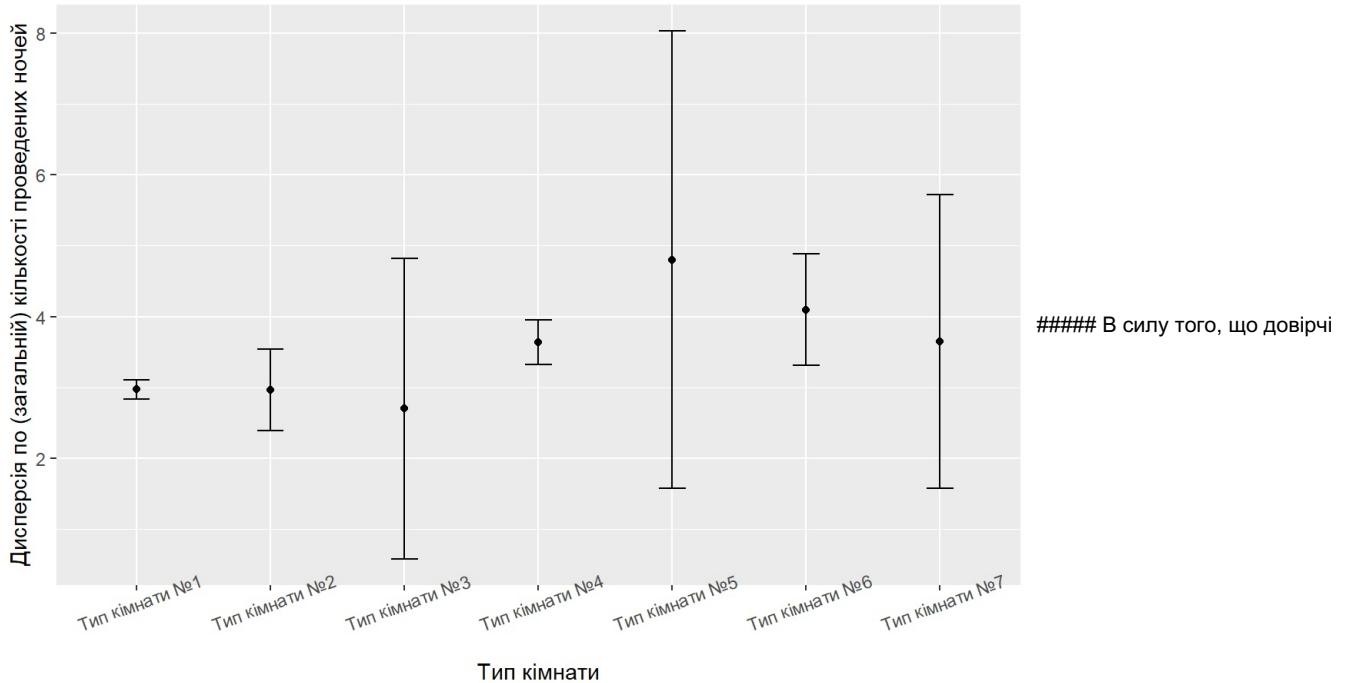
```

cis <- list()
for (i in 1:7) {
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(
      n = n(),
      mean = mean(no_of_nights),
      var = var(no_of_nights),
      fourth_moment = mean((no_of_nights - mean)^4),
      sd_var = sqrt((fourth_moment - var^2) / n),
      a = var - qnorm(0.975) * sd_var,
      b = var + qnorm(0.975) * sd_var
    )
  cis[[current_room_type]] <- ci
}

ci_df <- bind_rows(cis, .id = "Room_Type")

ggplot(ci_df, aes(x = Room_Type, y = var)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип кімнати", y = "Дисперсія по (загальній) кількості проведених ночей", title = "Дисперсія більше та мінімальна відповідно до кількості записів") +
  scale_x_discrete(label = room_label_vector) +
  theme(axis.text.x = element_text(angle = 20, hjust = 0.5), plot.title = element_text(hjust = 0.5))
  
```

Дисперсія більше там де менше записів



інтервали для 3, 5 і 7 типів кімнат є досить широкими, то оцінити картину загалом не представляється можливим. Що можна помітити, так це відносно незначну різницю у значеннях довірчих інтервалів по типам кімнат 1, 2, 4 і 6, тож ніяким типам кімнат не надаватимемо перевагу у зручності/комфортності, які могли б вплинути на кількість проведених ночей.

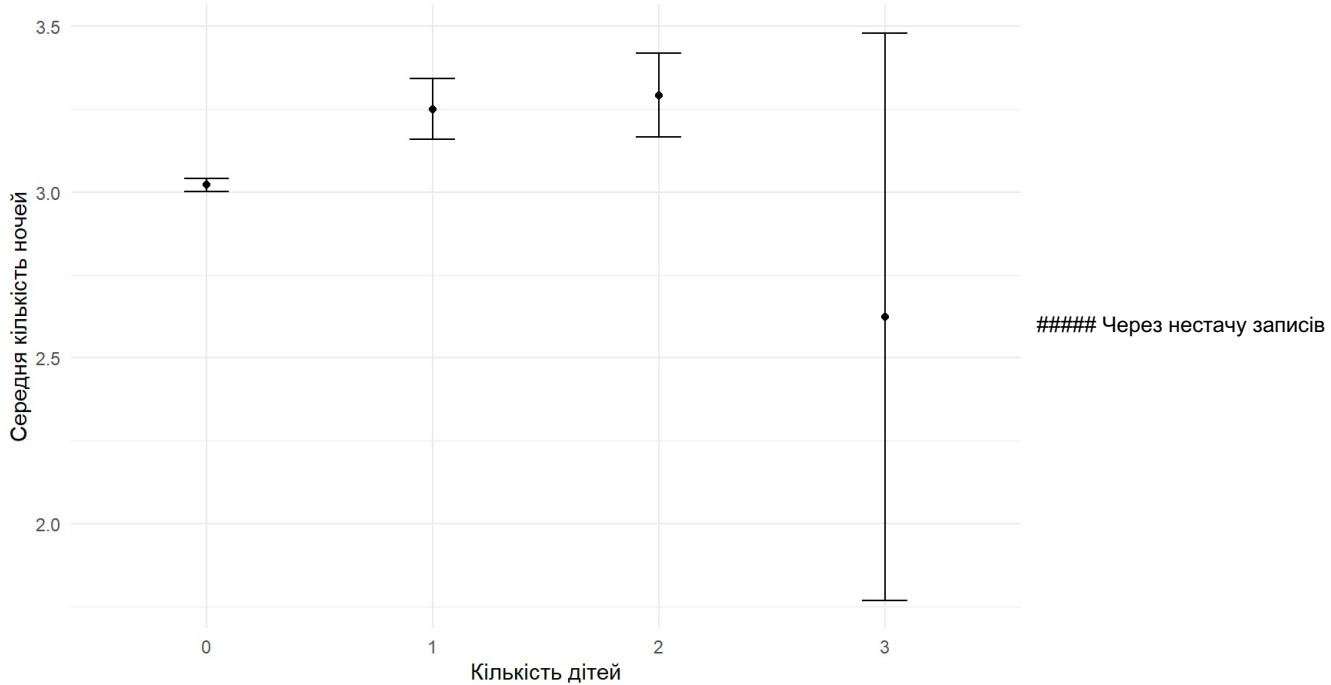
Заради цікавості розглянемо довірчі інтервали для середньої кількості нічей по кожному значенню кількості дітей. Можливо сім'ї з дітьми планують заселятися на довший проміжок часу (щось типу сімейної подорожі на відпочинок)

```
ci_nights_children <- hotel %>%
  group_by(no_of_children) %>%
  summarize(mean_nights = mean(no_of_nights),
            sd_nights = sd(no_of_nights),
            n_nights = n(),
            ci_low_nights = mean(no_of_nights) + qnorm(0.025) * sd(no_of_nights) / sqrt(n_nights),
            ci_high_nights = mean(no_of_nights) + qnorm(0.975) * sd(no_of_nights) / sqrt(n_nights))

# print(ci_nights_children)

ggplot(ci_nights_children, aes(x = as.factor(no_of_children), y = mean_nights)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_nights, ymax = ci_high_nights), width = 0.2) +
  labs(title = "Кількість нічей дещо зростає від кількості дітей", x = "Кількість дітей", y = "Середня кількість нічей") +
  theme_minimal() + theme(plot.title = element_text(hjust = 0.5))
```

Кількість ночей дещо зростає від кількості дітей



бронювань з трьома дітьми довірчий інтервал очікувано виходить зашироким. Для кількостей 0-2 ситуація приблизно однакова (значущої різниці у середній кількості ночей не спостерігається)

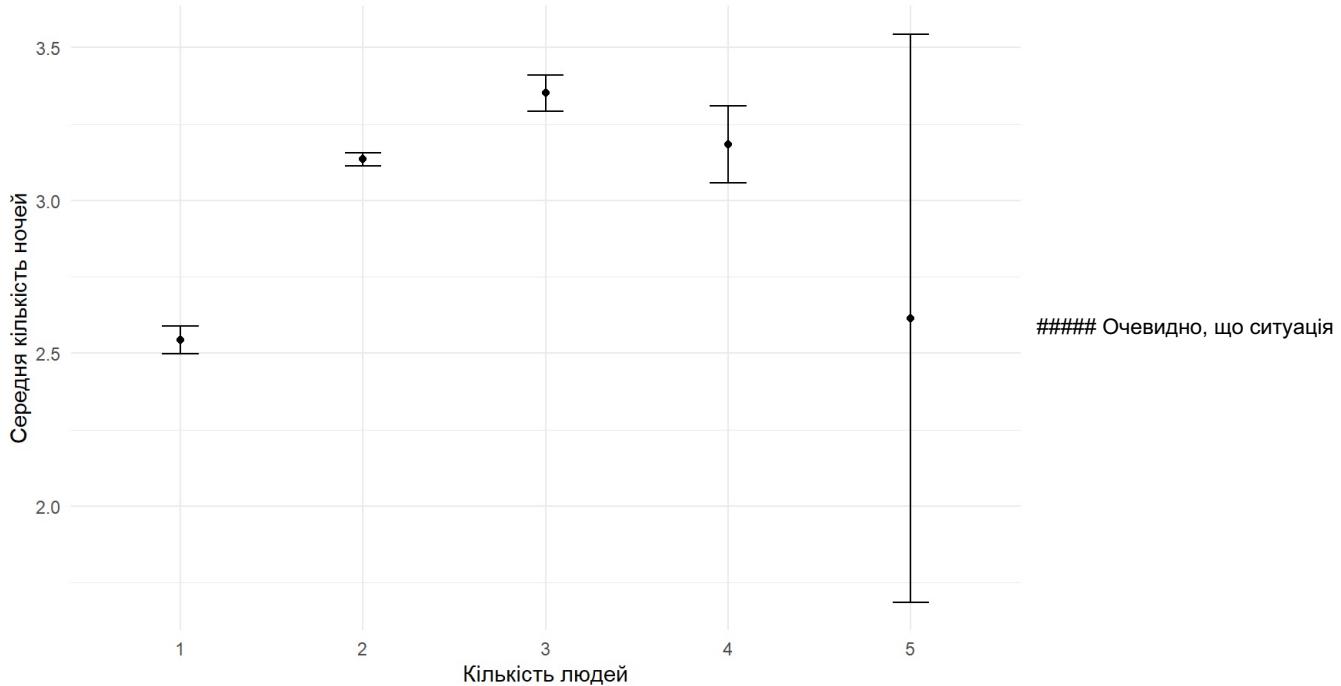
Подивимось на аналогічний графік відносно загальної кількості людей:

```
ci_nights_people <- hotel %>%
  group_by(no_of_people) %>%
  summarize(mean_nights = mean(no_of_nights),
            sd_nights = sd(no_of_nights),
            n_nights = n(),
            ci_low_nights = mean(no_of_nights) + qnorm(0.025) * sd(no_of_nights) / sqrt(n_nights),
            ci_high_nights = mean(no_of_nights) + qnorm(0.975) * sd(no_of_nights) / sqrt(n_nights))

# print(ci_nights_children)

ggplot(ci_nights_people, aes(x = as.factor(no_of_people), y = mean_nights)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_nights, ymax = ci_high_nights), width = 0.2) +
  labs(title = "Дещо зростає кількість ночей в залежності від кількості людей", x = "Кількість людей", y = "Середня кількість ночей") +
  theme_minimal() + theme(plot.title = element_text(hjust = 0.5))
```

Дещо зростає кількість ночей в залежності від кількості людей



загалом практично нічим не відрізняється

Дослідимо довірчі інтервали для середньої кількості ночей в залежності від потреби у паркувальному місці:

```

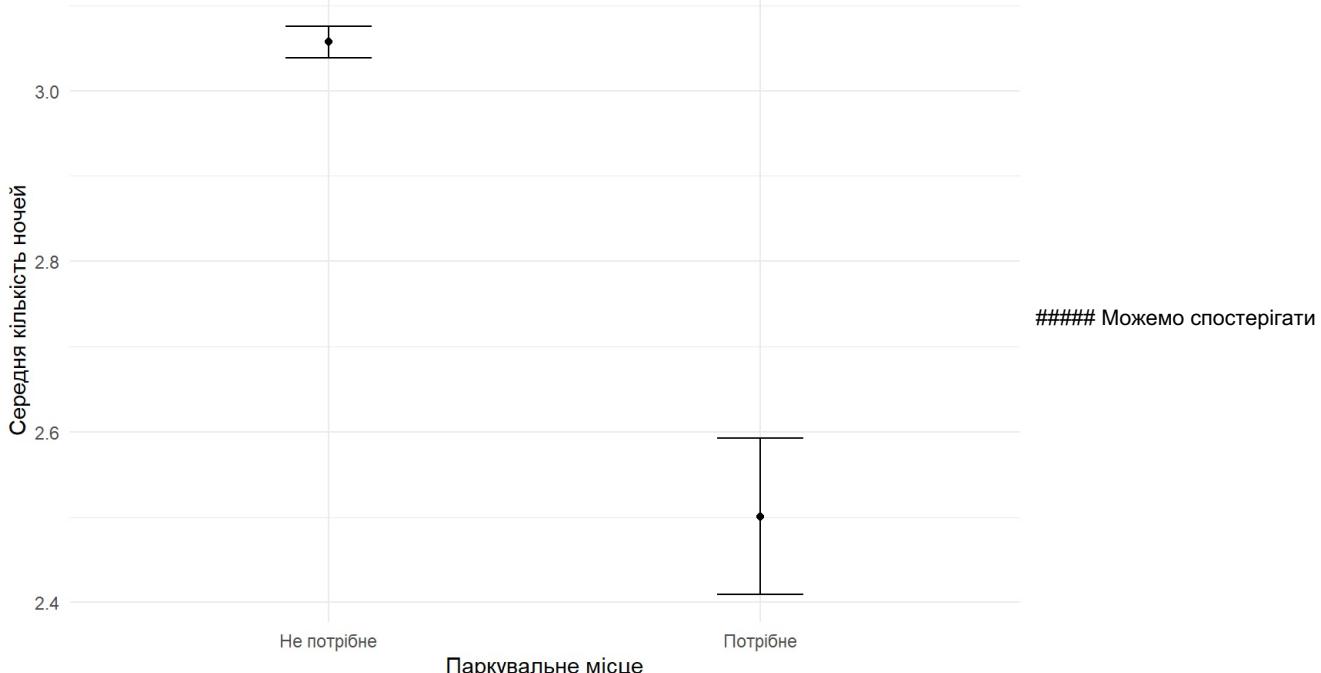
ci_nights_parking <- hotel %>%
  group_by(required_car_parking_space) %>%
  summarize(mean_nights = mean(no_of_nights),
            sd_nights = sd(no_of_nights),
            n_nights = n(),
            ci_low_nights = mean(no_of_nights) - qnorm(0.975) * sd(no_of_nights) / sqrt(n_nights),
            ci_high_nights = mean(no_of_nights) + qnorm(0.975) * sd(no_of_nights) / sqrt(n_nights),
            .groups = 'drop') # This will suppress the message

# print(ci_nights_parking)

# Plotting the mean nights with confidence intervals for parking space requirement
ggplot(ci_nights_parking, aes(x = required_car_parking_space, y = mean_nights)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_nights, ymax = ci_high_nights), width = 0.2) +
  labs(title = "Ті, кому потрібне паркувальне місце нощують менше",
       x = "Паркувальне місце", y = "Середня кількість ночей") +
  theme_minimal() + theme(plot.title = element_text(hjust = 0.5)) + scale_x_discrete(label = c("0" = "Не потрібне",
  ", "1" = "Потрібне"))

```

Ті, кому потрібне паркувальне місце нощують менше



наступну картину: в середньому люди, що добиралися на власному автомобілі, лишались на менший проміжок часу ніж інші. Це досить цікавий результат, адже можна було б очікувати, що сімейні подорожі автомобілем переважатимуть по середній кількості ночей інші бронювання. Проте

— Кількість ночей залежно від часу до прибууття ##### APPROVED (треба доробити) #####

```

hotel <- hotel %>%
  mutate(lead_time_category = cut(lead_time, breaks = 10))

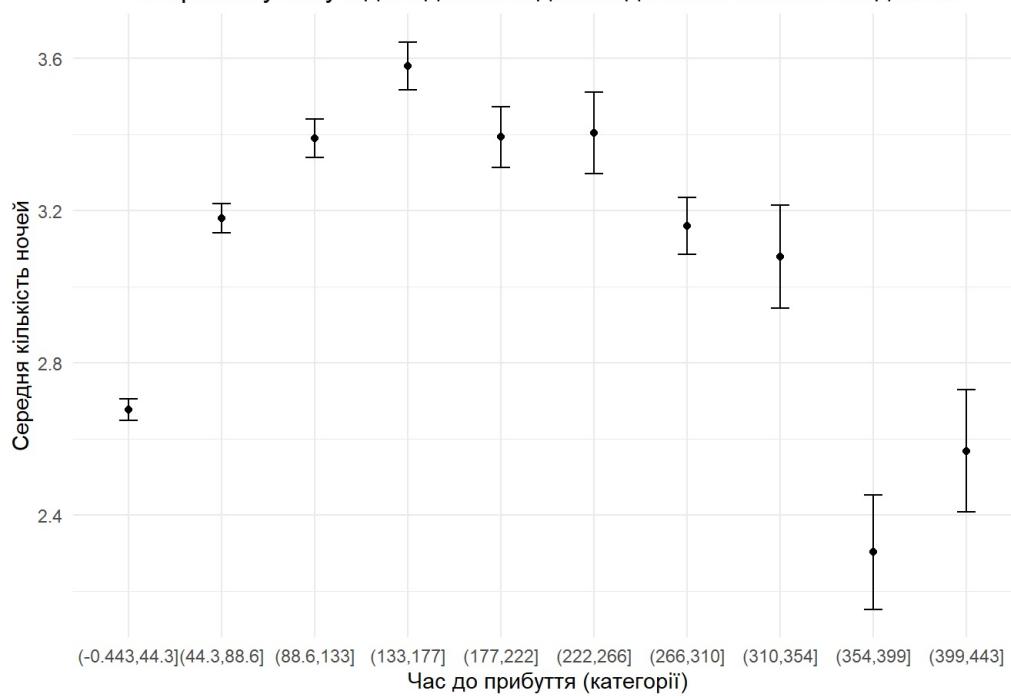
# Grouping by the new lead_time_category and summarizing
ci_nights_lead_time <- hotel %>%
  group_by(lead_time_category) %>%
  summarize(mean_nights = mean(no_of_nights, na.rm = TRUE),
            sd_nights = sd(no_of_nights, na.rm = TRUE),
            n_nights = n(),
            ci_low_nights = mean(no_of_nights, na.rm = TRUE) - qnorm(0.975) * sd(no_of_nights, na.rm = TRUE) / sqrt(n_nights),
            ci_high_nights = mean(no_of_nights, na.rm = TRUE) + qnorm(0.975) * sd(no_of_nights, na.rm = TRUE) / sqrt(n_nights),
            .groups = 'drop')

# print(ci_nights_lead_time)

ggplot(ci_nights_lead_time, aes(x = lead_time_category, y = mean_nights)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_nights, ymax = ci_high_nights), width = 0.2) +
  labs(title = "В проміжку часу від 88 до 266 годин люди залишаються на довше",
       x = "Час до прибууття (категорії)", y = "Середня кількість ночей") +
  theme_minimal() + theme(plot.title = element_text(hjust = 0.5))

```

В проміжку часу від 88 до 266 годин люди залишаються на довше



5. Що впливає на кількість попередніх скасувань/нескасувань?

- Можемо побудувати довірчі інтервали для середньої кількості попередніх скасувань для кожного типу кімнати.

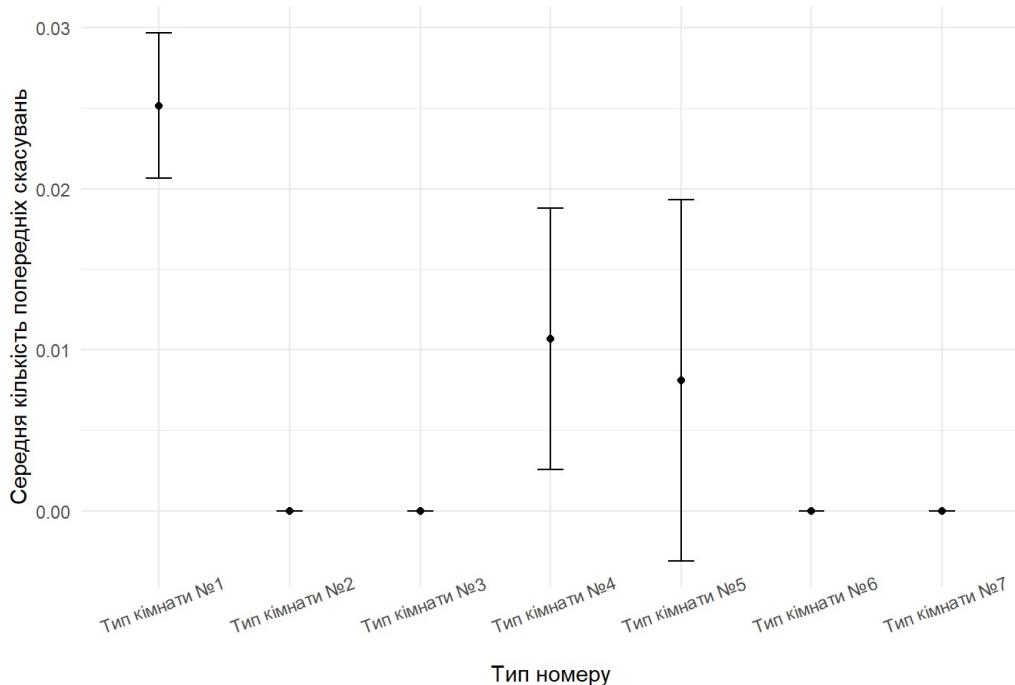
Кількість попередніх скасувань дуже мала, незалежно від типу кімнати - всього 1-3 попередніх скасування на 100 записів. Для деяких кімнат, таких як 2,3,6,7 - інтервали лежать в околі 0, що свідчить про майже відсутність попередніх скасувань в цих номерах. Відрізняється лише інтервал для першого типу кімнати, але, як було зазначено вище, ця кількість є дуже малою

```
ci_previous_cancellations <- hotel %>%
  group_by(room_type_reserved) %>%
  summarize(mean_cancellations = mean(no_of_previous_cancellations),
            sd_cancellations = sd(no_of_previous_cancellations),
            n_cancellations = n(),
            ci_low_cancellations = mean(no_of_previous_cancellations) + qnorm(0.025) * sd(no_of_previous_cancellations) / sqrt(n()),
            ci_high_cancellations = mean(no_of_previous_cancellations) + qnorm(0.975) * sd(no_of_previous_cancellations) / sqrt(n()))

# print(ci_previous_cancellations)

ggplot(ci_previous_cancellations, aes(x = room_type_reserved, y = mean_cancellations)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_cancellations, ymax = ci_high_cancellations), width = 0.2) +
  labs(title = "Кількість попередніх скасувань для різних типів кімнат", x = "Тип номеру", y = "Середня кількість попередніх скасувань") +
  theme_minimal() + scale_x_discrete(label = room_label_vector) + theme(axis.text.x = element_text(angle = 20), plot.title = element_text(hjust = 0.5))
```

Кількість попередніх скасувань для різних типів кімнат



Побудуємо довірчі інтервали для дисперсії попередніх скасувань за типами кімнат. Бачимо дуже схожу ситуацію, для деяких типів кімнат дисперсія взагалі відсутня, тобто немає попередніх скасувань. Інтервали існують лише для 1, 4 та 5 типів кімнат, але вони не несуть ніякої ясності в досліджувану залежність.

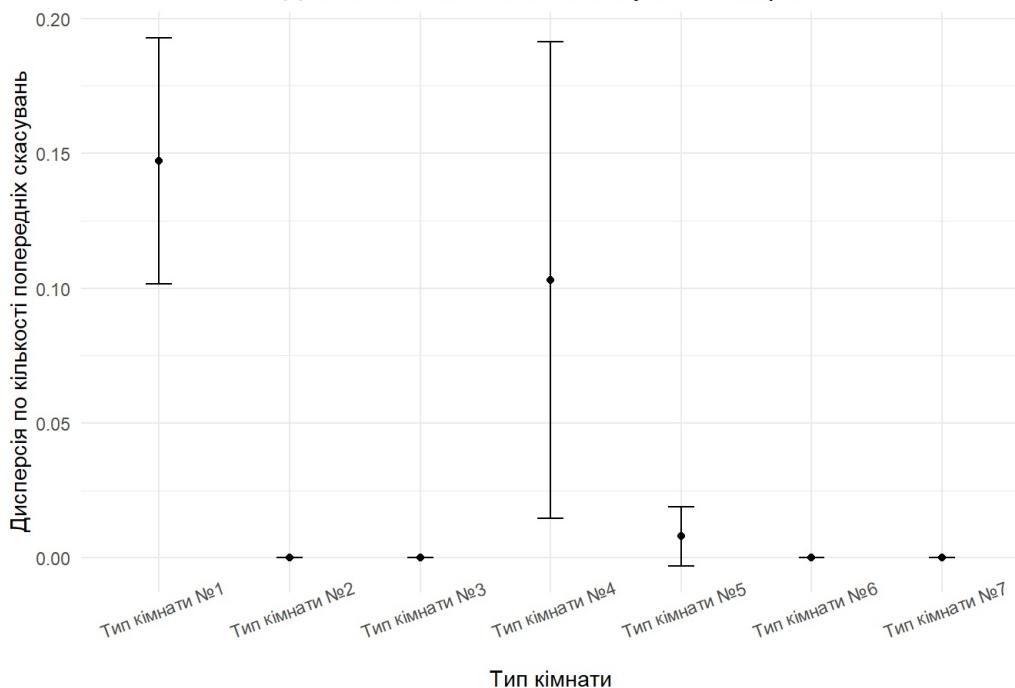
```

cis <- list()
for (i in 1:7) {
  current_room_type <- paste0("Room_Type ", i)
  ci <- hotel %>%
    filter(room_type_reserved == current_room_type) %>%
    summarize(
      n = n(),
      mean = mean(no_of_previous_cancellations),
      var = var(no_of_previous_cancellations),
      fourth_moment = mean((no_of_previous_cancellations - mean)^4),
      sd_var = sqrt((fourth_moment - var^2) / n),
      a = var - qnorm(0.975) * sd_var,
      b = var + qnorm(0.975) * sd_var
    )
  cis[[current_room_type]] <- ci
}

ci_df <- bind_rows(cis, .id = "Room_Type")

ggplot(ci_df, aes(x = Room_Type, y = var)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Тип кімнати", y = "Дисперсія по кількості попередніх скасувань", title = "Для деяких типів кімнат відсутня дисперсія") +
  theme_minimal() + scale_x_discrete(label = room_label_vector) + theme(axis.text.x = element_text(angle = 20), plot.title = element_text(hjust = 0.5))
  
```

Для деяких типів кімнат відсутня дисперсія



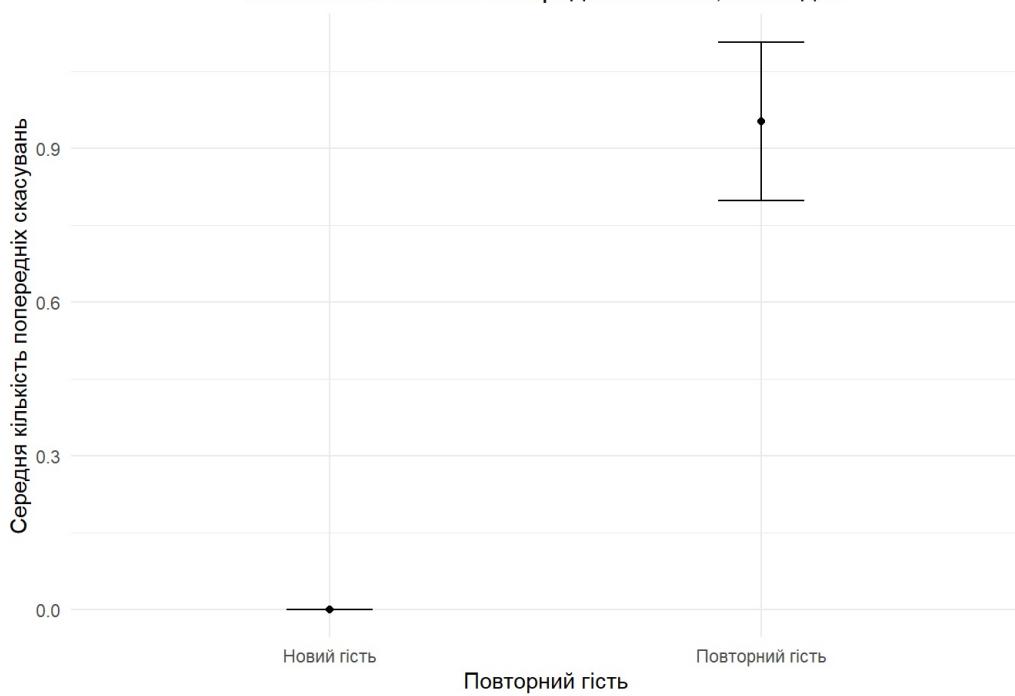
Спробуємо побудувати схожі довірчі інтервали для кількості попередніх скасувань, але згрупуємо дані за фактом повторного гостя. Тут вже бачимо набагато більш змістовну картину - виявляється, що у повторного гостя кількість попередніх скасувань знаходиться в околі 1. Те що у гостя, який приїхав вперше немає попередніх скасувань не викликає ніяких запитань.

```
ci_guest <- hotel %>%
  group_by(repeated_guest) %>%
  summarize(mean_cancellations = mean(no_of_previous_cancellations),
            sd_cancellations = sd(no_of_previous_cancellations),
            n_cancellations = n(),
            ci_low_cancellations = mean(no_of_previous_cancellations) + qnorm(0.025) * sd(no_of_previous_cancellations) / sqrt(n()),
            ci_high_cancellations = mean(no_of_previous_cancellations) + qnorm(0.975) * sd(no_of_previous_cancellations) / sqrt(n()))

# print(ci_previous_cancellations)

ggplot(ci_guest, aes(x = repeated_guest, y = mean_cancellations)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_cancellations, ymax = ci_high_cancellations), width = 0.2) +
  labs(title = "Новий гість не має попередніх записів, очевидно", x = "Повторний гість", y = "Середня кількість попередніх скасувань") +
  theme_minimal() + scale_x_discrete(label = c("0" = "Новий гість", "1" = "Повторний гість"))+ theme(plot.title = element_text(hjust = 0.5))
```

Новий гість не має попередніх записів, очевидно



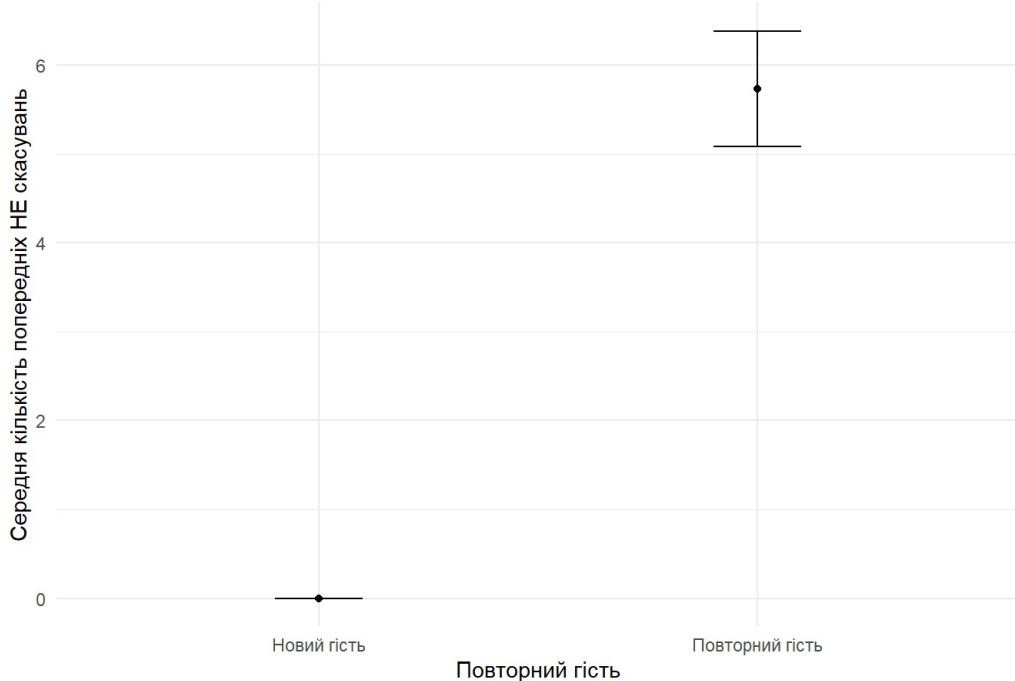
Побудуємо аналогічний інтервал для не відмінених записів, знову така сама картина - повторні гості мають в середньому приблизно 6 попередніх нескасувань, тобто приїхали в готель вже в сьомий раз, що є досить цікавим відкриттям - якщо гості так багато разів повертаються значить, напевно, готель є хорошим. І знову нічого цікавого що для гостей, які приїхали вперше немає попередніх нескасованих записів.

```
ci_guest_2 <- hotel %>%
  group_by(repeated_guest) %>%
  summarize(mean_cancellations = mean(no_of_previous_bookings_not_canceled),
            sd_cancellations = sd(no_of_previous_bookings_not_canceled),
            n_cancellations = n(),
            ci_low_cancellations = mean(no_of_previous_bookings_not_canceled) + qnorm(0.025) * sd(no_of_previous_
bookings_not_canceled) / sqrt(n()),
            ci_high_cancellations = mean(no_of_previous_bookings_not_canceled) + qnorm(0.975) * sd(no_of_previous
_bookings_not_canceled) / sqrt(n()))

# print(ci_previous_cancellations)

ggplot(ci_guest_2, aes(x = repeated_guest, y = mean_cancellations)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_cancellations, ymax = ci_high_cancellations), width = 0.2) +
  labs(title = "Новий гість не має попередніх записів, очевидно", x = "Повторний гість", y = "Середня кількість п
опередніх НЕ скасувань") +
  theme_minimal() + scale_x_discrete(label = c("0" = "Новий гість", "1" = "Повторний гість")) + theme(plot.title =
element_text(hjust = 0.5))
```

Новий гість не має попередніх записів, очевидно



Виходячи з питання про вплив попередніх скасувань бронювань сформуємо таку гіпотезу:

Відмінені і невідмінені бронювання мають однакову кількість попередніх скасувань бронювань

Протестувавши дану гіпотезу бачимо, що нульову гіпотезу можемо відхилити, тобто кількість попередніх скасувань бронювань має статистично значущу різницю

```
# ті що відмінили візит і ті що не відмінили мають однакову кількість попередніх скасувань бронювань
t.test(no_of_previous_cancellations ~ booking_status_binary, data = hotel_reverse, alternative = "two.sided")
```

```
## 
## Welch Two Sample t-test
##
## data: no_of_previous_cancellations by booking_status_binary
## t = 6.853, df = 34714, p-value = 7.354e-12
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  0.01694468 0.03052022
## sample estimates:
## mean in group 0 mean in group 1
##      0.029289406     0.005556959
```

```
ci <- hotel %>%
  filter(booking_status == "Not_Canceled") %>%
  summarize(mean = mean(no_of_previous_cancellations),
            sd = sd(no_of_previous_cancellations),
            n = n(),
            a = mean(no_of_previous_cancellations) + qnorm(0.025) * sd(no_of_previous_cancellations) / sqrt(n())
  ),
            b = mean(no_of_previous_cancellations) + qnorm(0.975) * sd(no_of_previous_cancellations) / sqrt(n())
)
ci
```

```
##       mean        sd       n        a        b
## 1 0.02928941 0.409922 23797 0.0240812 0.03449761
```

```
ci <- hotel %>%
  filter(booking_status == "Canceled") %>%
  summarize(mean = mean(no_of_previous_cancellations),
            sd = sd(no_of_previous_cancellations),
            n = n(),
            a = mean(no_of_previous_cancellations) + qnorm(0.025) * sd(no_of_previous_cancellations) / sqrt(n())
  ),
            b = mean(no_of_previous_cancellations) + qnorm(0.975) * sd(no_of_previous_cancellations) / sqrt(n())
)
ci
```

```
##      mean      sd      n      a      b
## 1 0.005556959 0.2420221 11877 0.001204349 0.009909569
```

```
estimates <- hotel_reverse %>%
  group_by(booking_status_binary) %>%
  summarise(mean_hat = mean(no_of_previous_cancellations),
            var_hat = var(no_of_previous_cancellations) / n())

mean_hat_req <- estimates %>% filter(booking_status_binary == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(booking_status_binary == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(booking_status_binary == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(booking_status_binary == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- 2 * pnorm(abs(T), lower.tail = FALSE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Canceled):", mean_hat_req, "\n")
```

```
## Mean (Canceled): 0.005556959
```

```
cat("Mean (Not Canceled):", mean_hat_no_req, "\n")
```

```
## Mean (Not Canceled): 0.02928941
```

```
cat("T-statistic:", T, "\n")
```

```
## T-statistic: 6.852965
```

```
cat("P-value:", p_value, "\n")
```

```
## P-value: 7.233456e-12
```

```
cat("95% Confidence Interval:", conf.int, "\n")
```

```
## 95% Confidence Interval: 0.01694491 0.03051998
```

Протестуємо ще одну аналогічну гіпотезу:

Відмінені і невідмінені бронювання мають однакову кількість попередніх НЕ скасувань бронювань

Протестувавши дану гіпотезу бачимо, що нульову гіпотезу можемо відхилити, тобто кількість попередніх НЕ скасувань бронювань має статистично значущу різницю

```
# ті що відмінили візит і ті що не відмінили мають однакову кількість попередніх НЕ скасувань бронювань
t.test(no_of_previous_bookings_not_canceled ~ booking_status_binary, data = hotel_reverse, alternative = "two.sided")
```

```
##
## Welch Two Sample t-test
##
## data: no_of_previous_bookings_not_canceled by booking_status_binary
## t = 14.531, df = 24193, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  0.1642063 0.2154134
## sample estimates:
## mean in group 0 mean in group 1
## 0.191998991 0.002189105
```

```

ci <- hotel %>%
  filter(booking_status == "Not_Canceled") %>%
  summarize(mean = mean(no_of_previous_bookings_not_canceled),
            sd = sd(no_of_previous_bookings_not_canceled),
            n = n(),
            a = mean(no_of_previous_bookings_not_canceled) + qnorm(0.025) * sd(no_of_previous_bookings_not_canceled) / sqrt(n()),
            b = mean(no_of_previous_bookings_not_canceled) + qnorm(0.975) * sd(no_of_previous_bookings_not_canceled) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 0.191999 2.006689 23797 0.1665033 0.2174947

```

```

ci <- hotel %>%
  filter(booking_status == "Canceled") %>%
  summarize(mean = mean(no_of_previous_bookings_not_canceled),
            sd = sd(no_of_previous_bookings_not_canceled),
            n = n(),
            a = mean(no_of_previous_bookings_not_canceled) + qnorm(0.025) * sd(no_of_previous_bookings_not_canceled) / sqrt(n()),
            b = mean(no_of_previous_bookings_not_canceled) + qnorm(0.975) * sd(no_of_previous_bookings_not_canceled) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 0.002189105 0.1297532 11877 -0.0001444219 0.004522632

```

```

estimates <- hotel_reverse %>%
  group_by(booking_status_binary) %>%
  summarise(mean_hat = mean(no_of_previous_bookings_not_canceled),
            var_hat = var(no_of_previous_bookings_not_canceled) / n())
mean_hat_req <- estimates %>% filter(booking_status_binary == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(booking_status_binary == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(booking_status_binary == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(booking_status_binary == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- 2 * pnorm(abs(T), lower.tail = FALSE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Canceled):", mean_hat_req, "\n")

```

```

## Mean (Canceled): 0.002189105

```

```

cat("Mean (Not Canceled):", mean_hat_no_req, "\n")

```

```

## Mean (Not Canceled): 0.191999

```

```

cat("T-statistic:", T, "\n")

```

```

## T-statistic: 14.53076

```

```

cat("P-value:", p_value, "\n")

```

```

## P-value: 7.735575e-48

```

```

cat("95% Confidence Interval:", conf.int, "\n")

```

```

## 95% Confidence Interval: 0.1642076 0.2154122

```

6. Чи є різниця в кількості попередніх скасувань для клієнтів, які вимагають паркувальне місце і тих, хто

його не потребує?

Побудуємо інтервал для кількості людей, згрупувавши дані за необхідністю в паркувальному місці. Видно, що в записах, в яких вказана необхідність в паркувальному місці в середньому більше людей, ніж в тих записах, в яких потреба не вказана. Складно оцінити наскільки ця різниця є значущою лише по інтервалах, бо значення відрізняються не більше ніж на 0,2. Вважаючи, що ми говоримо про кількість людей, різниця здається зовсім невеликою. Тому, існує необхідність протестувати відповідну гіпотезу про рівність цих груп. За результатом тестування гіпотези відхиляємо нульову гіпотезу, тобто дійсно різниця є значущою для даних груп.

```
cis <- list()

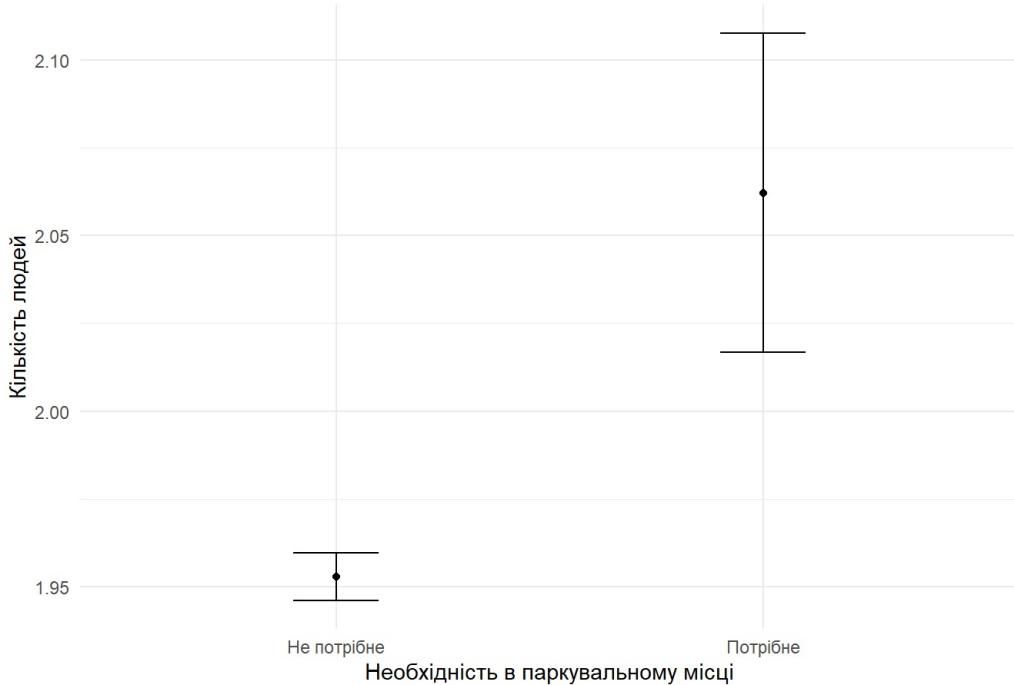
for (i in 0:1) {
  current_parking_space <- i
  ci <- hotel %>%
    filter(required_car_parking_space == current_parking_space) %>%
    summarize(mean = mean(no_of_people),
             sd = sd(no_of_people),
             n = n(),
             a = mean(no_of_people) - qnorm(0.975) * sd(no_of_people) / sqrt(n()),
             b = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))

  cis[[as.character(current_parking_space)]] <- ci
}

ci_df <- bind_rows(cis, .id = "required_car_parking_space")

# Plot confidence intervals
ggplot(ci_df, aes(x = required_car_parking_space, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Необхідність в паркувальному місці", y = "Кількість людей", title = "Там, де потрібне паркувальне місце більше людей") + theme_minimal() + scale_x_discrete(label = c("0" = "Не потрібне", "1" = "Потрібне")) + theme(plot.title = element_text(hjust = 0.5))
```

Там, де потрібне паркувальне місце більше людей



```
t.test(no_of_people ~ required_car_parking_space, data = hotel_reverse, alternative = "two.sided")
```

```
## 
## Welch Two Sample t-test
##
## data: no_of_people by required_car_parking_space
## t = -4.656, df = 1141.3, p-value = 3.603e-06
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -0.15522792 -0.06318716
## sample estimates:
## mean in group 0 mean in group 1
## 1.952950      2.062157
```

```

estimates <- hotel_reverse %>%
  group_by(required_car_parking_space) %>%
  summarise(mean_hat = mean(no_of_people),
            var_hat = var(no_of_people) / n())

mean_hat_req <- estimates %>% filter(required_car_parking_space == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(required_car_parking_space == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(required_car_parking_space == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(required_car_parking_space == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- 2 * pnorm(abs(T), lower.tail = FALSE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Needed):", mean_hat_req, "\n")

```

```
## Mean (Needed): 2.062157
```

```
cat("Mean (Not needed):", mean_hat_no_req, "\n")
```

```
## Mean (Not needed): 1.95295
```

```
cat("T-statistic:", T, "\n")
```

```
## T-statistic: -4.655982
```

```
cat("P-value:", p_value, "\n")
```

```
## P-value: 3.224399e-06
```

```
cat("95% Confidence Interval:", conf.int, "\n")
```

```
## 95% Confidence Interval: -0.1551791 -0.06323596
```

інтервал дисперсія

Побудуємо інтервал для дисперсії кількості попередніх скасувань бронювань і згрупуємо за необхідністю в паркувальному місці. Як видно, дисперсія має широкий інтервал для тих, кому необхідно паркувальне місце - це спричинено тим, що таких записів не так і багато.

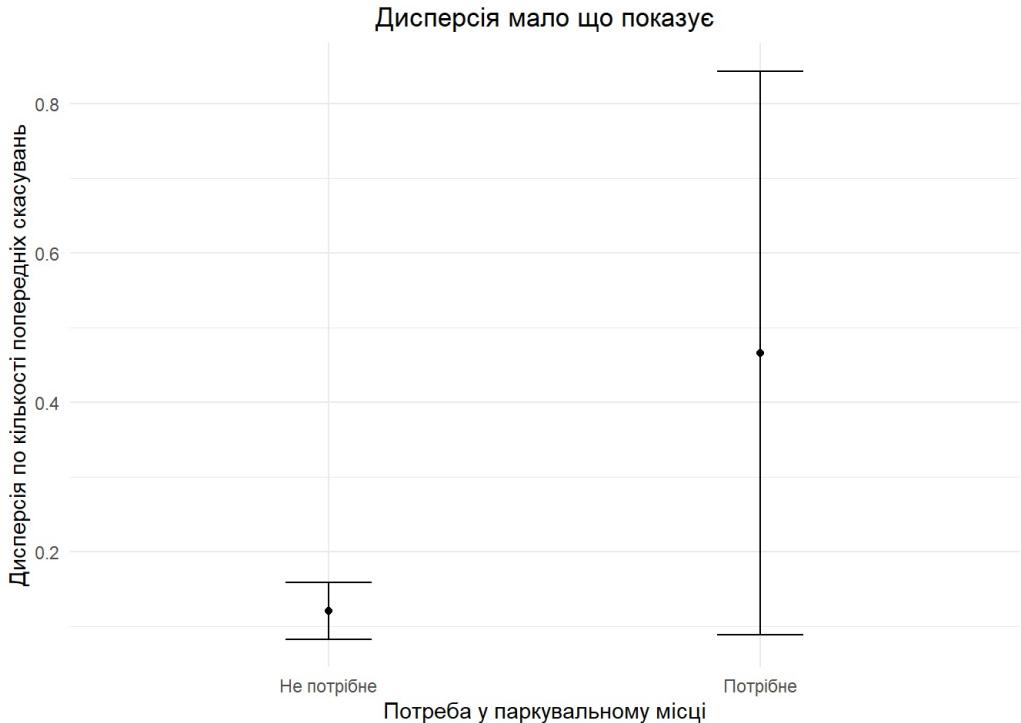
```

cis <- list()
for (current_parking_space in c(0, 1)) {
  ci <- hotel %>%
    filter(required_car_parking_space == current_parking_space) %>%
    summarize(
      n = n(),
      mean = mean(no_of_previous_cancellations),
      var = var(no_of_previous_cancellations),
      fourth_moment = mean((no_of_previous_cancellations - mean)^4),
      sd_var = sqrt((fourth_moment - var^2) / n),
      a = var - qnorm(0.975) * sd_var,
      b = var + qnorm(0.975) * sd_var
    )
  cis[[as.character(current_parking_space)]] <- ci
}

cis_df <- bind_rows(cis, .id = "required_car_parking_space")

ggplot(cis_df, aes(x = required_car_parking_space, y = var)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Потреба у паркувальному місці", y = "Дисперсія по кількості попередніх скасувань", title = "Дисперсія мало що показує") + theme_minimal() + scale_x_discrete(label = c("0" = "Не потрібне", "1" = "Потрібне")) + theme(plot.title = element_text(hjust = 0.5))

```



Тепер побудуємо інтервал для часу до прибуття, знову розділивши по групам за необхідністю в паркувальному місці. Бачимо цікавий результат: ті, кому потрібне місце - приїжджають набагато раніше, приблизно на добу. Тобто планують свою поїздку за менший час, можливо тому, що є на чому швидко доїхати.

```

cis <- list()

for (i in 0:1) {
  current_parking_space <- i
  ci <- hotel %>%
    filter(required_car_parking_space == current_parking_space) %>%
    summarize(mean = mean(lead_time),
             sd = sd(lead_time),
             n = n(),
             a = mean(lead_time) - qnorm(0.975) * sd(lead_time) / sqrt(n()),
             b = mean(lead_time) + qnorm(0.975) * sd(lead_time) / sqrt(n()))

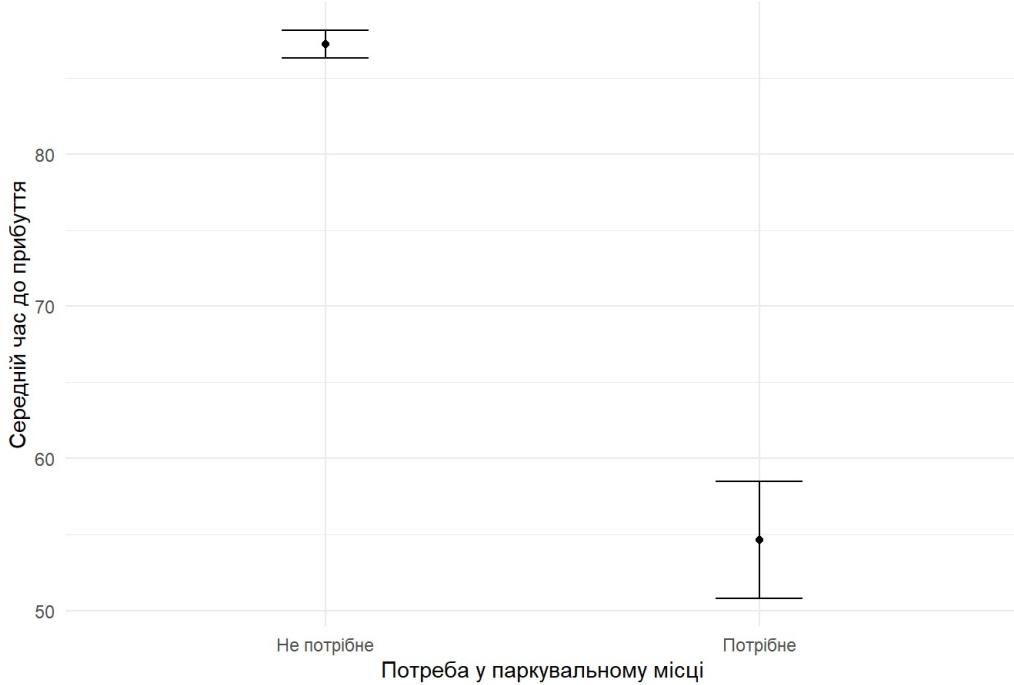
  cis[[as.character(current_parking_space)]] <- ci
}

ci_df <- bind_rows(cis, .id = "required_car_parking_space")

# Plot confidence intervals
ggplot(ci_df, aes(x = required_car_parking_space, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Потреба у паркувальному місці", y = "Середній час до прибуття", title = "Ti, кому потрібне паркувальне місце, прибувають раніше") +
  theme_minimal() + scale_x_discrete(label = c("0" = "Не потрібне", "1" = "Потрібне")) +
  theme(plot.title = element_text(hjust = 0.5))

```

Ti, кому потрібне паркувальне місце, прибувають раніше



Досліджуючи необхідність в паркувальному місці виникає гіпотеза, яку необхідно протестувати:

Люди, яким не потрібно паркувальне місце в середньому платять більше за тих, кому паркувальне місце потрібне

Перевірка даної гіпотези дає можливість відхилити нульову, тобто люди в середньому платять більше за необхідності паркувального місця, це цікавий результат - ті у кого є машина більше грошей на дорогий номер? Чи є інші фактори які на це впливають?

```
# Люди, яким не потрібно паркувальне місце в середньому платять більше за тих, кому паркувальне місце потрібне
t.test(avg_price_per_room ~ required_car_parking_space, data = hotel, alternative = "less")
```

```
##
## Welch Two Sample t-test
##
## data: avg_price_per_room by required_car_parking_space
## t = -11.344, df = 1138.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is less than 0
## 95 percent confidence interval:
##      -Inf -11.86987
## sample estimates:
## mean in group 0 mean in group 1
##          104.7263        118.6111
```

```

ci <- hotel %>%
  filter(required_car_parking_space == 0) %>%
  summarise(mean = mean(avg_price_per_room),
            sd = sd(avg_price_per_room),
            n = n(),
            a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n()),
            b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 104.7263 32.37639 34580 104.385 105.0675

```

```

ci <- hotel %>%
  filter(required_car_parking_space == 1) %>%
  summarise(mean = mean(avg_price_per_room),
            sd = sd(avg_price_per_room),
            n = n(),
            a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n()),
            b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 118.6111 40.07392 1094 116.2365 120.9858

```

```

estimates <- hotel %>%
  group_by(required_car_parking_space) %>%
  summarise(mean_hat = mean(avg_price_per_room),
            var_hat = var(avg_price_per_room) / n())
mean_hat_req <- estimates %>% filter(required_car_parking_space == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(required_car_parking_space == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(required_car_parking_space == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(required_car_parking_space == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- pnorm(T, lower.tail = TRUE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Needed):", mean_hat_req, "\n")

```

```

## Mean (Needed): 118.6111

```

```

cat("Mean (Not needed):", mean_hat_no_req, "\n")

```

```

## Mean (Not needed): 104.7263

```

```

cat("T-statistic:", T, "\n")

```

```

## T-statistic: -11.34357

```

```

cat("P-value:", p_value, "\n")

```

```

## P-value: 3.991131e-30

```

```

cat("95% Confidence Interval:", conf.int, "\n")

```

```

## 95% Confidence Interval: -16.28391 -11.4858

```

Побудуємо відповідний довірчий інтервал для середньої ціни за кімнату в залежності від необхідності в паркуванні. Бачимо значну різницю приблизно в 15-20\$, що є досить великим розривом, беручи до уваги ціни на кімнати, описані на початку дослідження.

```

cis <- list()

for (i in 0:1) {
  current_parking_space <- i
  ci <- hotel %>%
    filter(required_car_parking_space == current_parking_space) %>%
    summarize(mean = mean(avg_price_per_room),
              sd = sd(avg_price_per_room),
              n = n(),
              a = mean(avg_price_per_room) - qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()),
              b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))

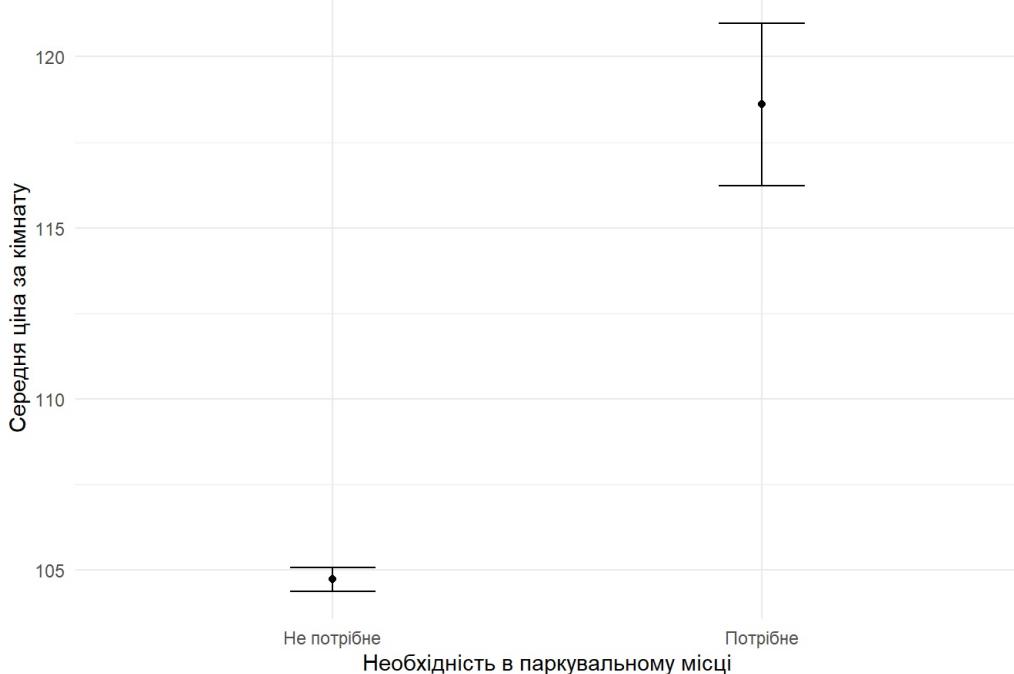
  cis[[as.character(current_parking_space)]] <- ci
}

ci_df <- bind_rows(cis, .id = "required_car_parking_space")

# Plot confidence intervals
ggplot(ci_df, aes(x = required_car_parking_space, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = a, ymax = b), width = 0.2) +
  labs(x = "Необхідність в паркувальному місці", y = "Середня ціна за кімнату", title = "Ті, кому потрібне паркувальне місце, платять більше") + theme_minimal() + scale_x_discrete(label = c("0" = "Не потрібне", "1" = "Потрібне")) + theme(plot.title = element_text(hjust = 0.5))

```

Ті, кому потрібне паркувальне місце, платять більше



7. Які нові цікаві відомості про повторних гостей?

Під час EDA повторний гість був дослідженний лише частково, через неможливість зробити статистичне виведення. Зараз же можна протестувати різноманітні гіпотези щодо цієї групи записів

І першою гіпотезою буде те, що тих гостей, які приїжджають вперше і не вперше однакова кількість.

Протестувавши гіпотезу можемо зробити висновок, що різниця в кількості гостей в таких записах є статистично значущою. А якщо подивимося на відповідні довірчі інтервали, побачимо, що ця різниця є досить великою приблизно 0.8 людини, тобто майже на одну людину менше в середньому в записах з повторними гостями.

```

# кількість повторних і неповторних гостей однакова
t.test(no_of_people ~ repeated_guest, data = hotel, alternative = "two.sided")

```

```

## 
## Welch Two Sample t-test
## 
## data: no_of_people by repeated_guest
## t = 48.099, df = 879.91, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  0.7288707 0.7908840
## sample estimates:
## mean in group 0 mean in group 1
##           1.973360          1.213483

```

```

estimates <- hotel %>%
  group_by(repeated_guest) %>%
  summarise(mean_hat = mean(no_of_people),
            var_hat = var(no_of_people) / n())

mean_hat_req <- estimates %>% filter(repeated_guest == 1) %>% pull(mean_hat)
mean_hat_no_req <- estimates %>% filter(repeated_guest == 0) %>% pull(mean_hat)
var_hat_req <- estimates %>% filter(repeated_guest == 1) %>% pull(var_hat)
var_hat_no_req <- estimates %>% filter(repeated_guest == 0) %>% pull(var_hat)

se <- sqrt(var_hat_req + var_hat_no_req)

T <- (mean_hat_no_req - mean_hat_req) / se

p_value <- 2 * pnorm(abs(T), lower.tail = FALSE)

conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)

cat("Mean (Requests):", mean_hat_req, "\n")

```

```
## Mean (Requests): 1.213483
```

```
cat("Mean (No Requests):", mean_hat_no_req, "\n")
```

```
## Mean (No Requests): 1.97336
```

```
cat("T-statistic:", T, "\n")
```

```
## T-statistic: 48.09885
```

```
cat("P-value:", p_value, "\n")
```

```
## P-value: 0
```

```
cat("95% Confidence Interval:", conf.int, "\n")
```

```
## 95% Confidence Interval: 0.7289133 0.7908413
```

```

ci <- hotel %>%
  filter(repeated_guest == 0) %>%
  summarize(mean = mean(no_of_people),
            sd = sd(no_of_people),
            n = n(),
            a = mean(no_of_people) + qnorm(0.025) * sd(no_of_people) / sqrt(n()),
            b = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))
ci

```

```
##      mean        sd       n       a       b
## 1 1.97336 0.6362855 34873 1.966682 1.980039
```

```

ci <- hotel %>%
  filter(repeated_guest == 1) %>%
  summarize(mean = mean(no_of_people),
            sd = sd(no_of_people),
            n = n(),
            a = mean(no_of_people) + qnorm(0.025) * sd(no_of_people) / sqrt(n()),
            b = mean(no_of_people) + qnorm(0.975) * sd(no_of_people) / sqrt(n()))
ci

```

```
##      mean        sd       n       a       b
## 1 1.213483 0.4365982 801 1.183248 1.243718
```

Так само протестуємо гіпотезу про рівність (в середньому) оплати для гостей, що прибули вперше і не вперше.

Протестувавши гіпотезу бачимо, що нульову гіпотезу можна відхилити, тобто гості не платять однаково. А побудувавши відповідні інтервали бачимо, що різниця в ціні є доволі значною - приблизно 30\$, що насправді є дуже великою різницею. Можливо, гості які приїжджають не вперше мають якісь знижки, чи вибирають номери дешевше, а може є ще якісь фактори які на це впливають?

```
# Не повторні і повторні гости платять однаково  
t.test(avg_price_per_room ~ repeated_guest, data = hotel, alternative = "two.sided")
```

```
##  
## Welch Two Sample t-test  
##  
## data: avg_price_per_room by repeated_guest  
## t = 41.497, df = 904.93, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0  
## 95 percent confidence interval:  
## 28.27714 31.08463  
## sample estimates:  
## mean in group 0 mean in group 1  
## 105.81853 76.13764
```

```
estimates <- hotel %>%  
  group_by(repeated_guest) %>%  
  summarise(mean_hat = mean(avg_price_per_room),  
           var_hat = var(avg_price_per_room) / n())  
  
mean_hat_req <- estimates %>% filter(repeated_guest == 1) %>% pull(mean_hat)  
mean_hat_no_req <- estimates %>% filter(repeated_guest == 0) %>% pull(mean_hat)  
var_hat_req <- estimates %>% filter(repeated_guest == 1) %>% pull(var_hat)  
var_hat_no_req <- estimates %>% filter(repeated_guest == 0) %>% pull(var_hat)  
  
se <- sqrt(var_hat_req + var_hat_no_req)  
  
T <- (mean_hat_no_req - mean_hat_req) / se  
  
p_value <- 2 * pnorm(abs(T), lower.tail = FALSE)  
  
conf.int <- c(mean_hat_no_req - mean_hat_req - qnorm(0.975) * se, mean_hat_no_req - mean_hat_req + qnorm(0.975) * se)  
  
cat("Mean (Requests):", mean_hat_req, "\n")
```

```
## Mean (Requests): 76.13764
```

```
cat("Mean (No Requests):", mean_hat_no_req, "\n")
```

```
## Mean (No Requests): 105.8185
```

```
cat("T-statistic:", T, "\n")
```

```
## T-statistic: 41.49704
```

```
cat("P-value:", p_value, "\n")
```

```
## P-value: 0
```

```
cat("95% Confidence Interval:", conf.int, "\n")
```

```
## 95% Confidence Interval: 28.27901 31.08275
```

```
ci <- hotel %>%  
  filter(repeated_guest == 0) %>%  
  summarize(mean = mean(avg_price_per_room),  
           sd = sd(avg_price_per_room),  
           n = n(),  
           a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n()),  
           b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))  
ci
```

```
##      mean      sd      n      a      b  
## 1 105.8185 32.66498 34873 105.4757 106.1614
```

```

ci <- hotel %>%
  filter(repeated_guest == 1) %>%
  summarize(mean = mean(avg_price_per_room),
            sd = sd(avg_price_per_room),
            n = n(),
            a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n()),
            b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))
ci

```

```

##      mean      sd      n      a      b
## 1 76.13764 19.62838 801 74.77834 77.49694

```

Побудуємо довірчий інтервал для середньої кількості ночей згрупувавши дані за повторністю гостя. Як бачимо, в середньому повторний гість ночує меншу кількість ночей - приблизно на 1 добу. Чому так виходить, гості незалежно вибирають ночувати менше чи є ще якісь фактори які на це впливають? Це питання потребує подальшого дослідження

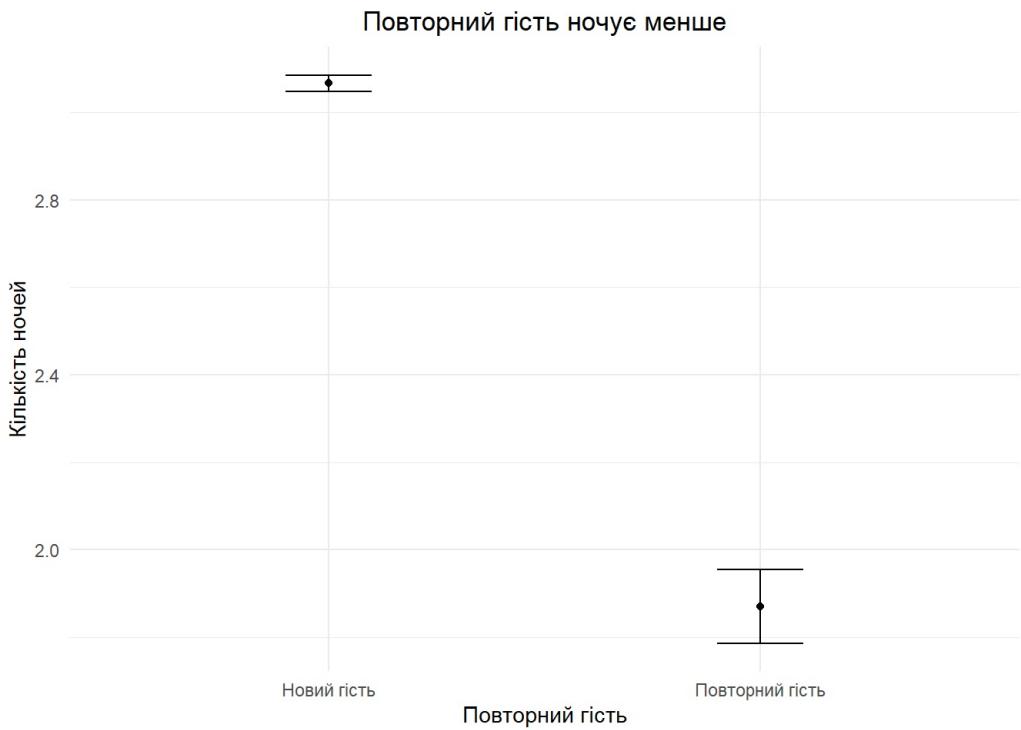
```

ci_guest <- hotel %>%
  group_by(repeated_guest) %>%
  summarize(mean_cancellations = mean(no_of_nights),
            sd_cancellations = sd(no_of_nights),
            n_cancellations = n(),
            ci_low_cancellations = mean(no_of_nights) + qnorm(0.025) * sd(no_of_nights) / sqrt(n()),
            ci_high_cancellations = mean(no_of_nights) + qnorm(0.975) * sd(no_of_nights) / sqrt(n()))

# print(ci_previous_cancellations)

ggplot(ci_guest, aes(x = repeated_guest, y = mean_cancellations)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_low_cancellations, ymax = ci_high_cancellations), width = 0.2) +
  labs(title = "Повторний гість ночує менше", x = "Повторний гість", y = "Кількість ночей") +
  theme_minimal() + scale_x_discrete(label = c("0" = "Новий гість", "1" = "Повторний гість")) + theme(plot.title =
element_text(hjust = 0.5))

```



Спробуємо оцінити незалежність записів з повторним гостем від інших величин, таких як статус бронювання і необхідність в паркувальному місці. Як бачимо, обидві гіпотези відхиляємо, тобто величини не є незалежними

```

# незалежність повторного гостя і статусу бронювання
cont_tab <- xtabs(~ repeated_guest + booking_status, data = hotel)
cont_tab

```

```

##          booking_status
## repeated_guest Canceled Not_Canceled
##             0      11861       23012
##             1        16         785

```

```

chisq.test(cont_tab, correct = FALSE)

```

```

## 
## Pearson's Chi-squared test
## 
## data: cont_tab
## X-squared = 361.36, df = 1, p-value < 2.2e-16

```

```

# незалежність повторного гостя і необхідності в паркуванні
cont_tab <- xtabs(~ repeated_guest + required_car_parking_space, data = hotel)
cont_tab

```

```

##           required_car_parking_space
## repeated_guest      0      1
##                 0 33913   960
##                 1   667   134

```

```
chisq.test(cont_tab, correct = FALSE)
```

```

## 
## Pearson's Chi-squared test
## 
## data: cont_tab
## X-squared = 514.53, df = 1, p-value < 2.2e-16

```

Частка записів з менш ніж 2 людьми не перевищує 21%.

```

observed_count <- sum(hotel$no_of_people < 2)
# частка записів, де людей < 2 > 21%
prop.test(observed_count, nrow(hotel), p = 0.21, alternative = "less")

```

```

## 
## 1-sample proportions test with continuity correction
## 
## data: observed_count out of nrow(hotel), null probability 0.21
## X-squared = 11.426, df = 1, p-value = 0.0003622
## alternative hypothesis: true p is less than 0.21
## 95 percent confidence interval:
##  0.0000000 0.2062342
## sample estimates:
##          p
## 0.2026966

```

Бутстреп

БЛОК 1

Порахуємо довірчий інтервал для медіани avg_price_per_room

```

medianFunc <- function(data, indices) {
  median(data[indices])
}

```

З лекцій відомо, що медіана має асимптотично нормальній розподіл, проте підрахунок її дисперсії є проблематичним. Тому замість того, щоб рахувати її дисперсію і брати від неї корінь, згенеруємо 200 бутстреп-вибірок (стандартна кількість для оцінювання конкретних параметрів), для кожної бутстреп-вибірки порахуємо медіану, і від отриманого списку у 200 медіан порахуємо вибікову дисперсію. Отримане значення слугуватиме апроксимацією стандартної похибки для оцінки медіани. Тож, маємо:

```

x <- hotel$avg_price_per_room

bootResult <- boot(data = x, statistic = medianFunc, R = 200)

print(bootResult)

```

```

## 
## ORDINARY NONPARAMETRIC BOOTSTRAP
## 
## 
## Call:
## boot(data = x, statistic = medianFunc, R = 200)
## 
## 
## Bootstrap Statistics :
##      original   bias   std. error
## t1*       100 -0.00645  0.04287293

```

Користуючись правилом двох сигм будуємо відповідний довірчий інтервал для оцінки медіани avg_price_per_room:

```

ci <- hotel %>%
  summarize(median = median(avg_price_per_room),
            n = n(),
            a = median(avg_price_per_room) + qnorm(0.025) * sd(bootResult$t),
            b = median(avg_price_per_room) + qnorm(0.975) * sd(bootResult$t))
ci

```

```

##   median     n      a      b
## 1    100 35674 99.91597 100.084

```

БЛОК 2

Порахуємо довірчий інтервал для медіани lead_time. Для цього виконаємо ту саму послідовність дій, що була описана вище у випадку з медіаною для avg_price_per_room:

```

x <- hotel$lead_time

bootResult2 <- boot(data = x, statistic = medianFunc, R = 200)

print(bootResult2)

```

```

## 
## ORDINARY NONPARAMETRIC BOOTSTRAP
## 
## 
## Call:
## boot(data = x, statistic = medianFunc, R = 200)
## 
## 
## Bootstrap Statistics :
##      original   bias   std. error
## t1*       59 -0.4475  0.6023353

```

Побудова довірчого інтервалу для відповідної медіани

```

ci <- hotel %>%
  summarize(median = median(lead_time),
            n = n(),
            a = median(lead_time) + qnorm(0.025) * sd(bootResult2$t),
            b = median(lead_time) + qnorm(0.975) * sd(bootResult2$t))
ci

```

```

##   median     n      a      b
## 1    59 35674 57.81944 60.18056

```

Заради власної цікавості побудуємо аналогічним чином довірчий інтервал для 90-ого персентиля lead_time:

```

x <- hotel$lead_time

quantile_value <- 0.9

bootResult_quantile <- boot(data = x, statistic = function(data, indices) {
  quantile(data[indices], probs = quantile_value)
}, R = 200)

se_quantile <- sd(bootResult_quantile$)

ci_quantile <- hotel %>%
  summarize(percentile_90 = quantile(lead_time, probs = quantile_value),
            n = n()) %>%
  mutate(lower_bound = percentile_90 - qnorm(0.975) * se_quantile,
        upper_bound = percentile_90 + qnorm(0.975) * se_quantile)

print(ci_quantile)

```

```

##   percentile_90      n lower_bound upper_bound
## 1          213 35674    210.1908   215.8092

```

БЛОК 3

Порахуємо кореляцію між змінними repeated_guest та no_of_previous_bookings_not_canceled

```

cor(as.numeric(hotel_corr$repeated_guest), as.numeric(hotel_corr$no_of_previous_bookings_not_canceled), method='spearman')

```

```

## [1] 0.9327746

```

За допомогою бутстрепу оцінимо значення оцінки кореляції, порахуємо її зміщення та стандартне відхилення

```

library("boot")
data <- read.csv("Hotel Reservations.csv")
data <- data %>% select(no_of_previous_bookings_not_canceled, repeated_guest)
colnames(data) <- c("x", "y")

data <- as.data.frame(data)
x <- data$Var1
y <- data$Var2
dat <- data.frame(x,y)

b3 <- boot(data,
  statistic = function(data, i) {
    cor(data[i, "x"], data[i, "y"], method='spearman')
  },
  R = 2000
)
b3

```

```

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = data, statistic = function(data, i) {
##   cor(data[i, "x"], data[i, "y"], method = "spearman")
## }, R = 2000)
##
##
## Bootstrap Statistics :
##       original      bias     std. error
## t1* 0.9327746 1.871415e-06 0.005771699

```

Використовуючи раніше отримані дані згенеруємо довірчі інтервали для оцінки кореляції Спірмана:

```

boot.ci(b3, type = c("norm", "basic", "perc", "bca"), L = empinf(b3, index=1L, type="jack"))

```

```

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = b3, type = c("norm", "basic", "perc", "bca"),
##          L = empinf(b3, index = 1, type = "jack"))
##
## Intervals :
## Level      Normal           Basic
## 95%   ( 0.9215,  0.9441 )  ( 0.9213,  0.9442 )
##
## Level      Percentile        BCa
## 95%   ( 0.9213,  0.9443 )  ( 0.9206,  0.9438 )
## Calculations and Intervals on Original Scale

```

Що можна сказати про ці довірчі інтервали: загальновідомо, що Normal (асимптотичний нормальний), Basic (пивотальний) та Percentile (персентильний) довірчі інтервали є інтервалами першого порядку. Це означає, що зі збільшенням n справжнє покриття інтервалів прямує до альфа зі швидкістю $O(1/n)$. Для BCa (bias-corrected) це $O(1/n^2)$, тобто він прямує значно швидше за попередні 3.

Попри цей факт, усі інтервали дуже схожі між собою. Хіба що можна виділити, що BCa має на ~ 0.0005 меншу праву границю за усі 3 інтервали першого порядку.

Збережені результати

```

print('BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS')

## [1] "BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS"

print('Based on 2000 bootstrap replicates')

## [1] "Based on 2000 bootstrap replicates"

print('')

## [1] ""

print('CALL : ')

## [1] "CALL : "

print('boot.ci(boot.out = b3, type = c("norm", "basic", "perc", "bca"), ')

## [1] "boot.ci(boot.out = b3, type = c(\"norm\", \"basic\", \"perc\", \"bca\"), "

print('    L = empinf(b3, index = 1, type = "jack"))')

## [1] "    L = empinf(b3, index = 1, type = \"jack\"))"

print(NA)

## [1] NA

print('Intervals : ')

## [1] "Intervals : "

print('Level      Normal           Basic      ')

## [1] "Level      Normal           Basic      "

print('95%   ( 0.9212,  0.9440 )  ( 0.9214,  0.9441 )  ')

## [1] "95%   ( 0.9212,  0.9440 )  ( 0.9214,  0.9441 )  "

```

```

print('')

## [1] ""

print('Level      Percentile      BCa      ')
## [1] "Level      Percentile      BCa      "

print('95%  ( 0.9214,  0.9441 )  ( 0.9211,  0.9436 )  ')
## [1] "95%  ( 0.9214,  0.9441 )  ( 0.9211,  0.9436 )  "

print('Calculations and Intervals on Original Scale')
## [1] "Calculations and Intervals on Original Scale"

```

ДОДАТКОВІ РЕЗУЛЬТАТИ

БЛОК 1

Побудуємо довірчі інтервали для середньої кількості дорослих/дітей в залежності від заброньованого типу кімнати. При цьому окремими кольорами позначатимемо інтервали в залежності від того чи вказувалася при бронюванні необхідність у паркувальному місці, адже так можна буде з'ясувати додаткові закономірності.

```

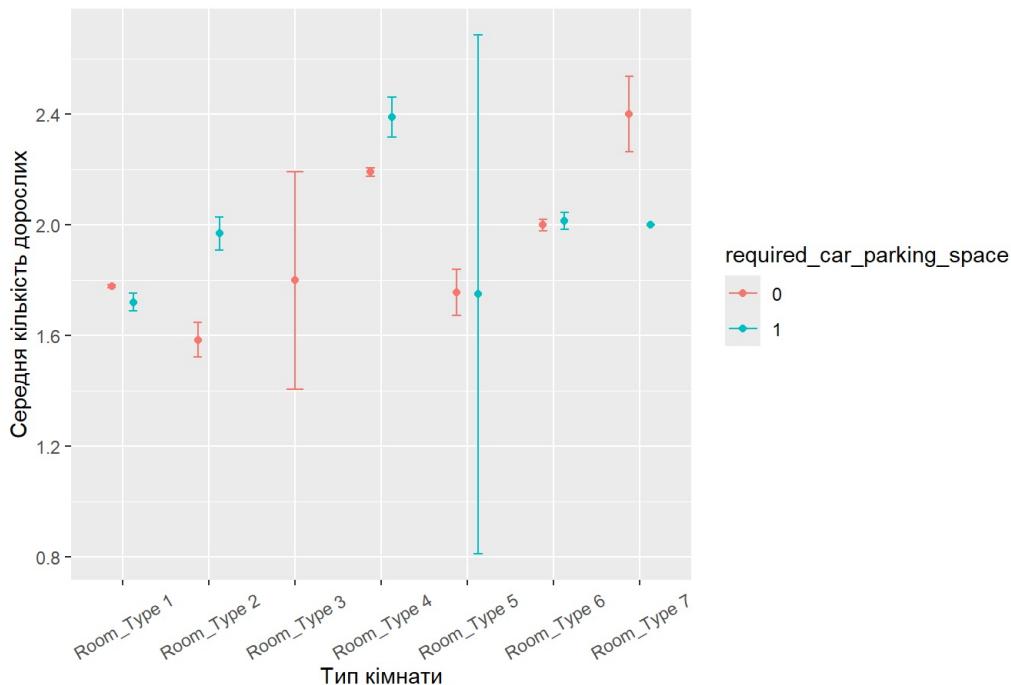
ci_adults_children_parking <- hotel %>%
  group_by(room_type_reserved, required_car_parking_space) %>%
  summarize(mean_adults = mean(no_of_adults),
           sd_adults = sd(no_of_adults),
           ci_low_adults = mean(no_of_adults) - qnorm(0.975) * sd(no_of_adults) / sqrt(n()),
           ci_high_adults = mean(no_of_adults) + qnorm(0.975) * sd(no_of_adults) / sqrt(n()),
           mean_children = mean(no_of_children),
           sd_children = sd(no_of_children),
           ci_low_children = mean(no_of_children) - qnorm(0.975) * sd(no_of_children) / sqrt(n()),
           ci_high_children = mean(no_of_children) + qnorm(0.975) * sd(no_of_children) / sqrt(n()),
           .groups = 'drop')

# print(ci_adults_children_parking)

ggplot(ci_adults_children_parking, aes(x = room_type_reserved, y = mean_adults, color = required_car_parking_space)) +
  geom_point(position = position_dodge(width = 0.5)) +
  geom_errorbar(aes(ymin = ci_low_adults, ymax = ci_high_adults), width = 0.2, position = position_dodge(width = 0.5)) +
  labs(title = "Кількість дорослих за типом кімнати та необхідністю у паркувальному місці", x = "Тип кімнати", y =
  "Середня кількість дорослих") +
  theme(axis.text.x = element_text(angle=30, hjust=0.5, vjust=0.5))

```

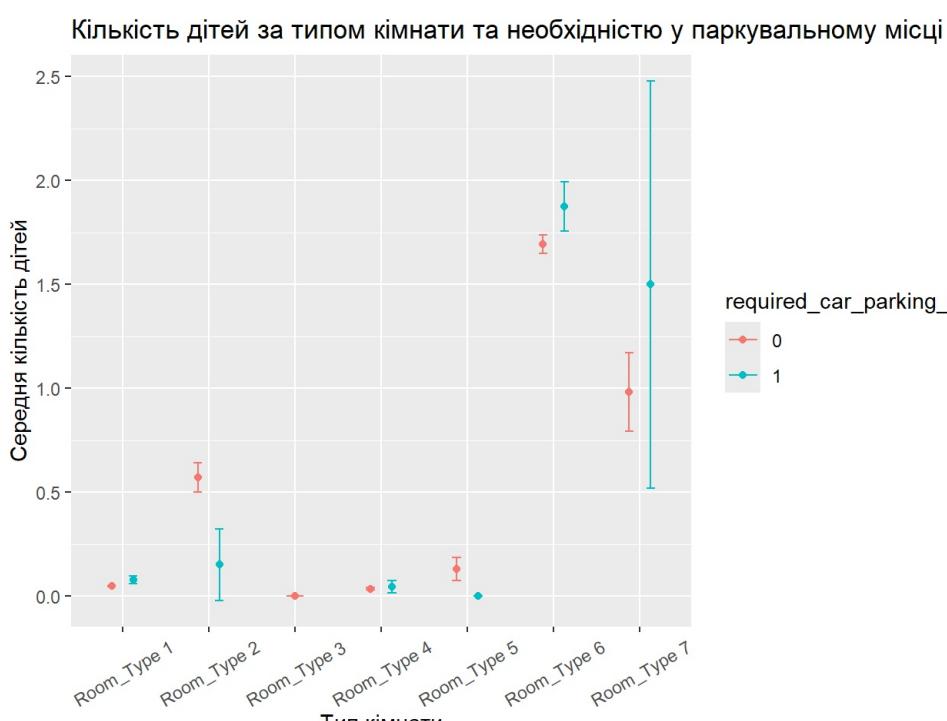
Кількість дорослих за типом кімнати та необхідністю у паркувальному місці



- Графік для середньої кількості дорослих нічим не привертає увагу. Загалом картина така, що в середньому кількість дорослих на кожен тип кімнати ніяк не залежить від потреби у паркувальному місці, що підтверджують довірчі інтервали.
- Третій тип кімнати має занадто мало записів на весь датасет, через що він не має довірчого інтервалу для середньої кількості дорослих, які потребували паркувальне місце.
- Для 5 і 7 типів кімнат ситуація пояснюється тим, що загальна кількість дорослих, що бронювала ці типи кімнат, вказуючи потребу у паркувальному місці, дорівнює відповідно 6 і 11, що замало для побудови досить вузького і точного довірчого інтервалу.

Перейдемо до розгляду графіку для середньої кількості дітей

```
ggplot(ci_adults_children_parking, aes(x = room_type_reserved, y = mean_children, color = required_car_parking_space)) +
  geom_point(position = position_dodge(width = 0.5)) +
  geom_errorbar(aes(ymin = ci_low_children, ymax = ci_high_children), width = 0.2, position = position_dodge(width = 0.5)) +
  labs(title = "Кількість дітей за типом кімнати та необхідністю у паркувальному місці", x = "Тип кімнати", y = "Середня кількість дітей") +
  theme(axis.text.x = element_text(angle=30, hjust=0.5, vjust=0.5))
```



Тут ситуація вже цікавіша.

- Попри те, що для деяких типів кімнат (6 і 7) існує досить небагато записів, що призводить до надто широких довірчих інтервалів, можна помітити те, що найчастіше сім'ї з дітьми селяться у 6-ий тип кімнати.
- Також цікаву закономірність можна помітити для 2-го і 7-го типів кімнат: в середньому сім'ї з більшою кількістю дітей вказують, що добиратимуться без власного автомобіля

БЛОК 2

Гіпотеза про те, що в середньому невідмінені кімнати коштують більше за відмінені

```
# Не відмінені коштують більше  
t.test(avg_price_per_room ~ booking_status_binary, data = hotel_reverse, alternative = "less")
```

```
##  
## Welch Two Sample t-test  
##  
## data: avg_price_per_room by booking_status_binary  
## t = -22.724, df = 24064, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 0 and group 1 is less than 0  
## 95 percent confidence interval:  
##       -Inf -7.656091  
## sample estimates:  
## mean in group 0 mean in group 1  
##      102.4042      110.6578
```

```
ci <- hotel %>%  
  filter(booking_status == "Not_Canceled") %>%  
  summarize(mean = mean(avg_price_per_room),  
           sd = sd(avg_price_per_room),  
           n = n(),  
           a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n()),  
           b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))  
ci
```

```
##      mean      sd      n      a      b  
## 1 102.4042 32.65948 23797 101.9893 102.8192
```

```
ci <- hotel %>%  
  filter(booking_status == "Canceled") %>%  
  summarize(mean = mean(avg_price_per_room),  
           sd = sd(avg_price_per_room),  
           n = n(),  
           a = mean(avg_price_per_room) + qnorm(0.025) * sd(avg_price_per_room) / sqrt(n()),  
           b = mean(avg_price_per_room) + qnorm(0.975) * sd(avg_price_per_room) / sqrt(n()))  
ci
```

```
##      mean      sd      n      a      b  
## 1 110.6578 32.16265 11877 110.0793 111.2362
```

Як видно з надзвичайно малого значення p-value, немає підстав не відхилити нульову гіпотезу. Тобто можемо припускати, що в середньому невідмінені кімнати коштують менше за відмінені