Chapitre 4:

Analyse statistique et inférence

4.1. Régression Linéaire Simple

Qu'est-ce que la régression ?

- Il s'agit de modèles statistiques permettant d'explorer la relation entre une variable réponse et des variables explicatives.
- Connaissant les valeurs des variables explicatives, on peut prédire les valeurs de la variable réponse.

- Variable réponse (ou variable dépendante) :
  - La variable que l'on souhaite prédire.
- Variables explicatives (ou variables indépendantes) :
  - Les variables qui expliquent comment la variable réponse évolue.

Régression linéaire et régression logistique

- Régression linéaire
  - La variable réponse est numérique.

- Régression logistique
  - La variable réponse est logique.

- Régression linéaire/logistique simple
  - Il n'y a qu'une seule variable explicative.

Régression linéaire

- Un modèle de régression linéaire simple est de la forme:

$$Y = \beta_0 + \beta_1 . X + \epsilon$$

- Où:
  - Y est la variable dépendante (variable réponse)
  - β_0 et β_1 sont les coefficients (ordonnée a l'origine et pente)
  - X est la variable indépendante (variable explicative)
  - ϵ est une erreur aléatoire

Quelle est la variable réponse ?

- La régression permet de prédire les valeurs d'une variable réponse à partir des valeurs connues des variables explicatives. Le choix de la variable réponse dépend de la question posée, mais dans de nombreux jeux de données, une variable s'impose naturellement.
- Nous explorerons un jeu de données immobiliers taïwanais comportant quatre variables.
  - *dist_to_mrt_station_m* : Distance à la station de métro la plus proche (en mètres).
  - *n_convenience* : Nombre de supérettes de proximité accessibles à pied.
  - *house_age_years* : Âge de la maison (en années, regroupé en trois catégories).
  - *price_twd_msq* : Prix du mètre carré (en dollars taïwanais).

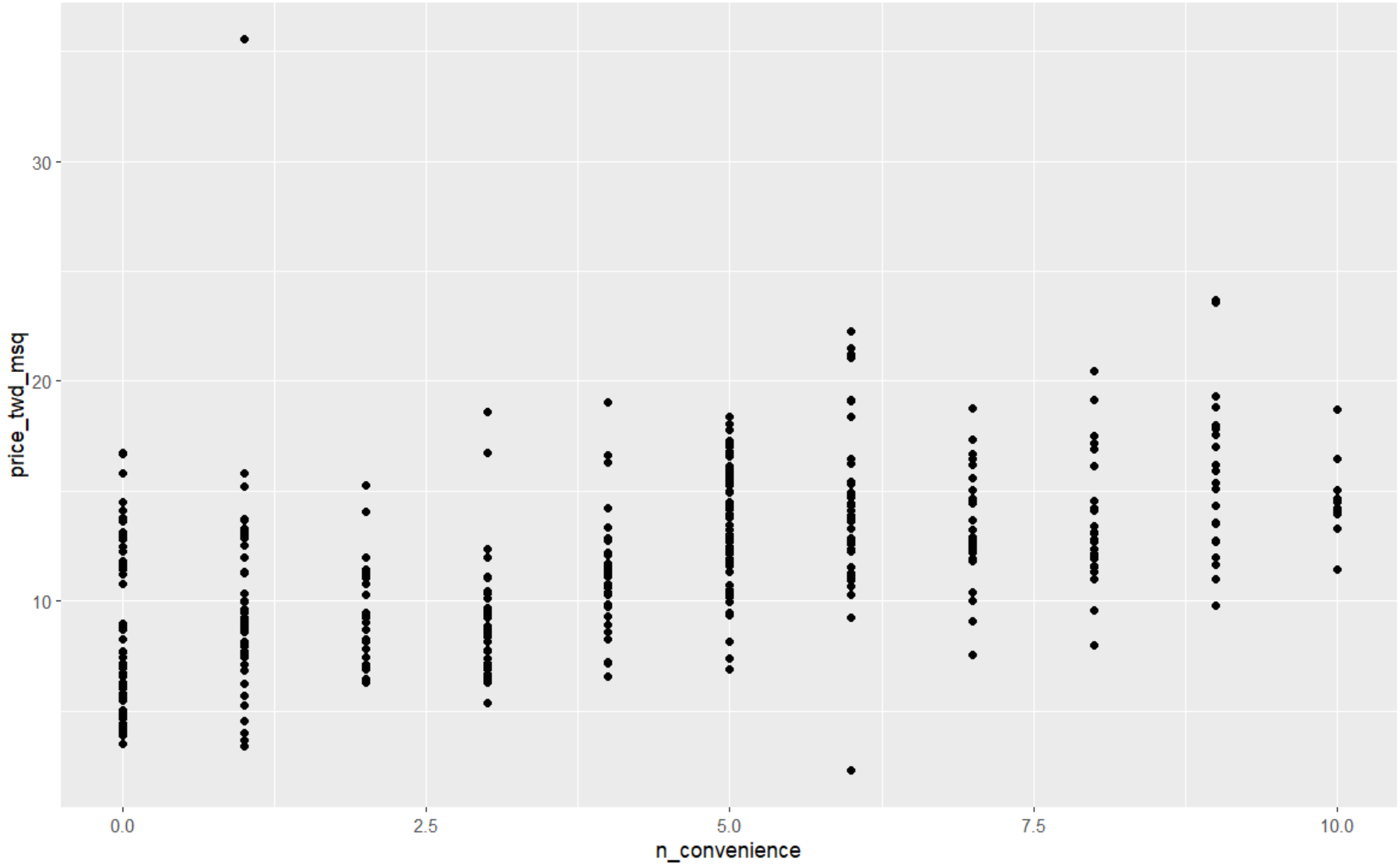
Visualisation de deux variables

- Avant d'appliquer des modèles statistiques, il est généralement conseillé de visualiser vos données. Ici, nous examinerons la relation entre le prix des maisons par unité de surface et le nombre de supérettes à proximité
- En utilisant *taiwan_real_estate*, tracez un nuage de points de price_twd_msq (axe y) en fonction de n_convenience (axe x).

```
library(ggplot2)

ggplot(taiwan_real_estate, aes(x = n_convenience, y
= price_twd_msq)) +
      geom_point()
```

Visualisation de deux variables



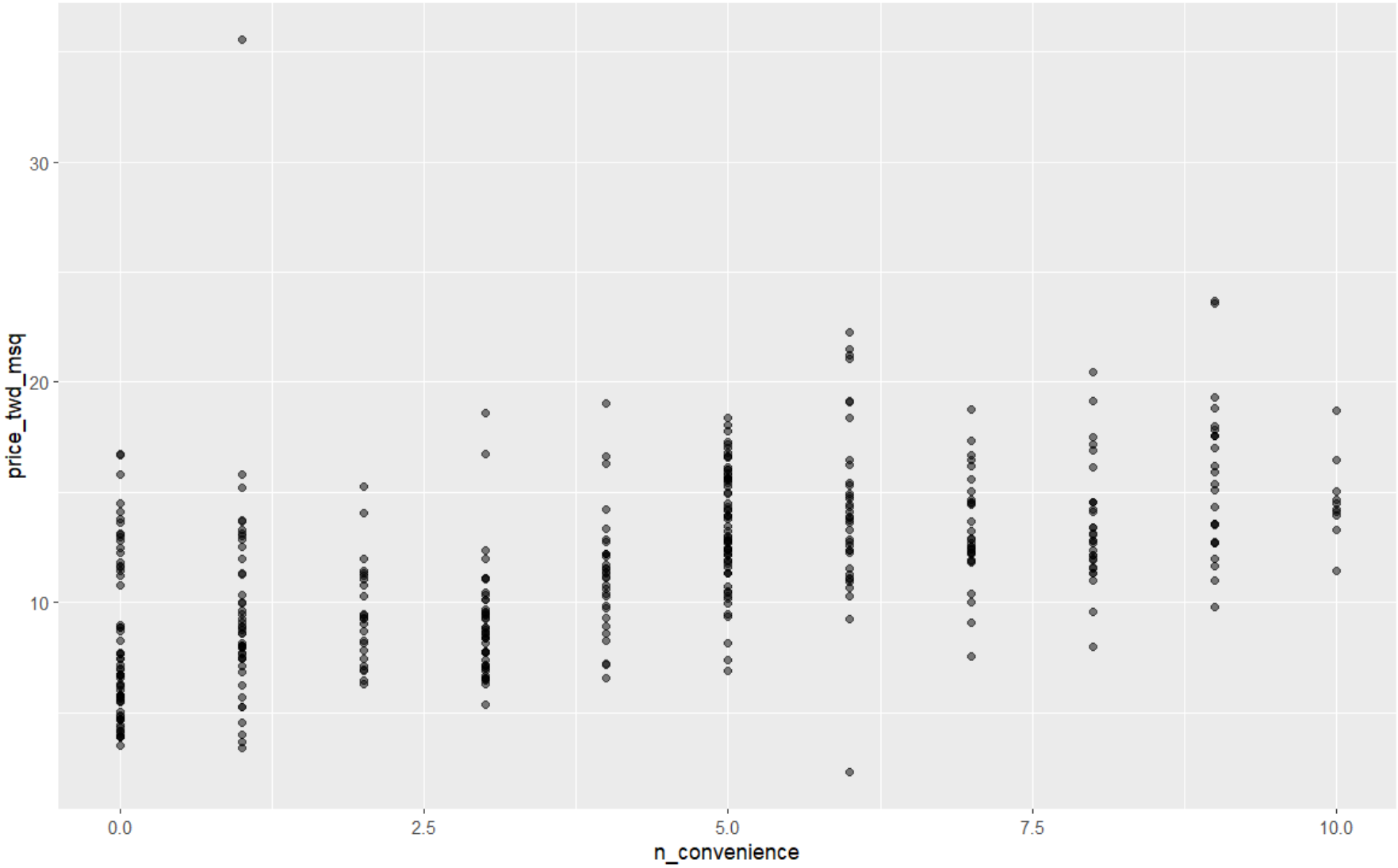
Visualisation de deux variables

- L'une des difficultés de ces données réside dans le fait que le nombre de supérettes est constitué de valeurs entières, ce qui entraîne un chevauchement des points. Pour y remédier, nous allons rendre les points transparents.

```
# Rendre les points transparents à 50%

ggplot(taiwan_real_estate, aes(x = n_convenience, y
= price_twd_msq)) +
      geom_point(alpha = 0.5)
```

Visualisation de deux variables



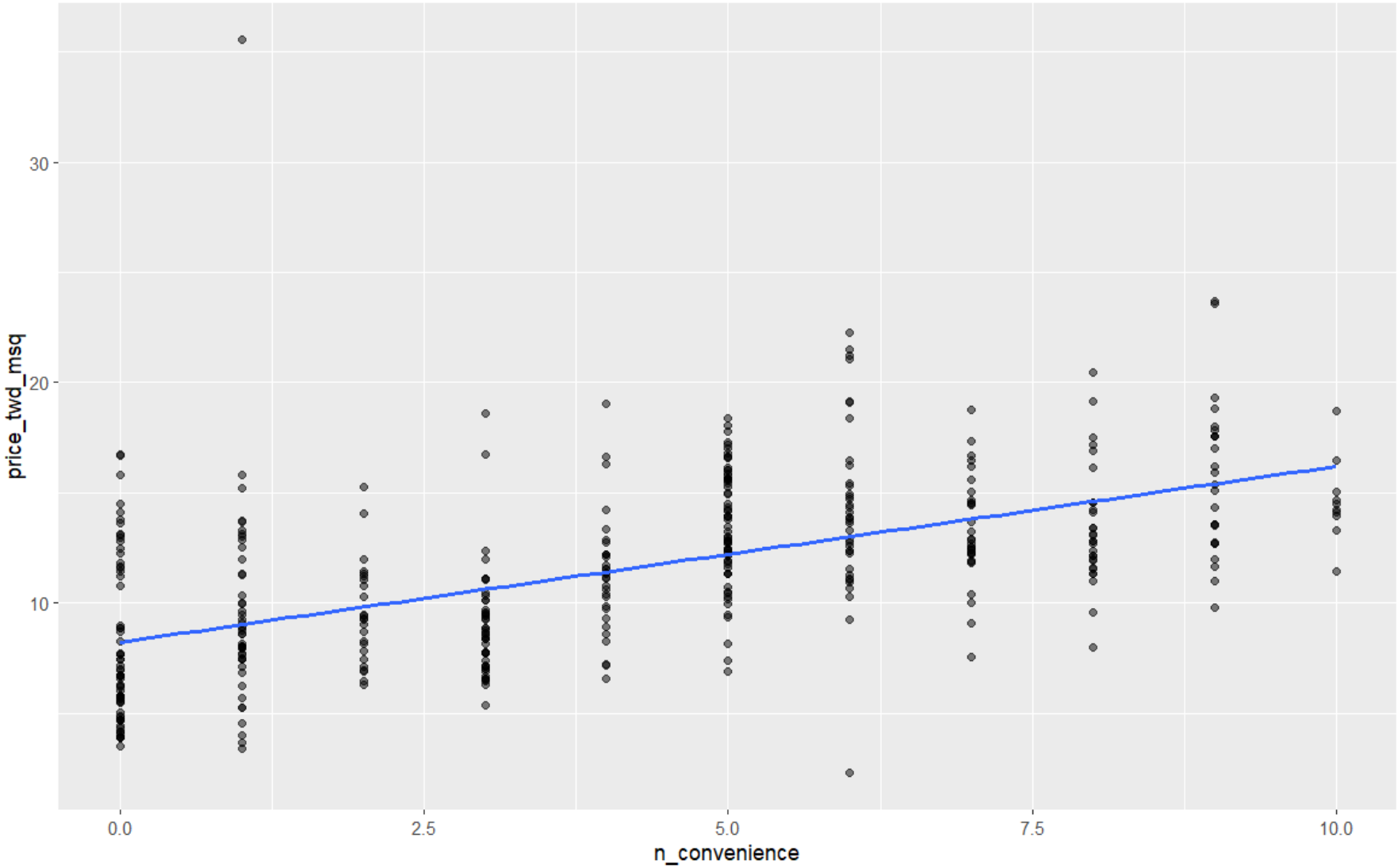
Visualisation de deux variables

- On peut mettre à jour le graphique en ajoutant une courbe de tendance, calculée à l'aide d'une régression linéaire. On peut omettre le ruban de confiance.

```
# Ajouter une courbe de tendance linéaire sans
ruban de confiance

ggplot(taiwan_real_estate, aes(x = n_convenience, y
= price_twd_msq)) +
      geom_point(alpha = 0.5) +
      geom_smooth(
            method = "lm",
            se = FALSE
            )
```

Visualisation de deux variables



Ajustement d'une régression linéaire

- La ligne droite est définie par deux éléments :
  - l'ordonnée à l'origine (*intercept*): la valeur de y lorsque x s'annule.
  - la pente (*slope*): la valeur d' augmentation de y lorsque x augmente de 1.
- Equation :

$$y = intercept + slope \times x$$

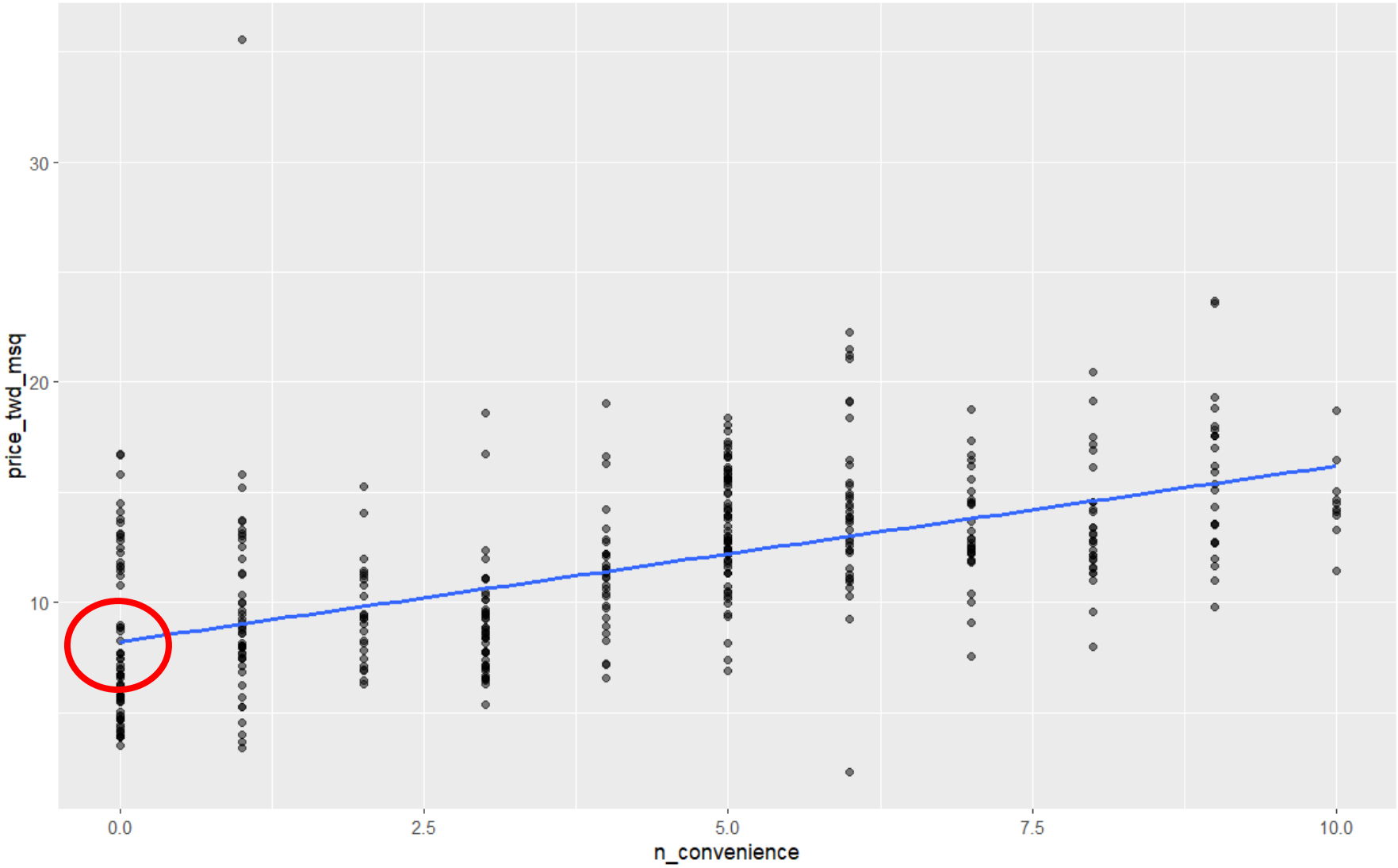
Estimation de l'ordonnée à l'origine (*intercept*)

- Les modèles de régression linéaire ajustent toujours une droite aux données. Les droites sont définies par deux propriétés : leur ordonnée à l'origine et leur pente.

- On observe ici un nuage de points représentant le prix des maisons par unité de surface en fonction du nombre de supérettes de proximité, à partir des données immobilières de Taïwan.

- Estimer l'ordonnée à l'origine de la droite de régression linéaire.

Estimation de l'ordonnée à l'origine (*intercept*)

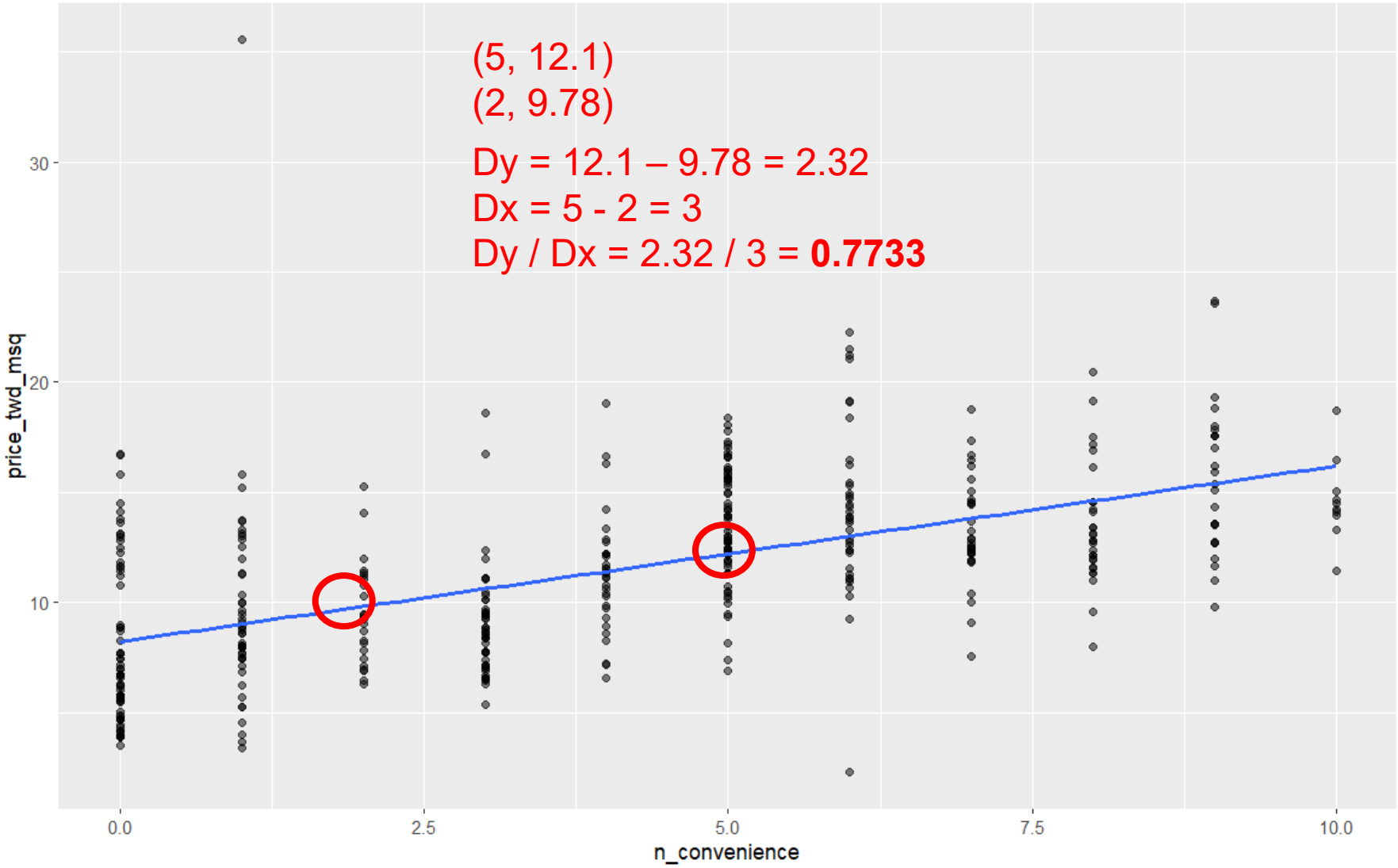


Estimation de la pente (*slope*)

- Voici le même nuage de points représentant le prix des maisons par unité de surface en fonction du nombre de commerces de proximité, à partir des données immobilières taïwanaises.

- Cette fois-ci, estimer la pente de la droite de tendance. Cliquez une fois sur la droite, puis double-cliquez à différents endroits pour relever la valeur de la pente.

Estimation de la pente (*slope*)



Régression linéaire avec `lm()`

- Bien que **`ggplot`** puisse afficher une droite de tendance de régression linéaire à l'aide de **`geom_smooth()`**, il ne permet pas d'accéder à l'ordonnée à l'origine et à la pente en tant que variables, ni de manipuler les résultats du modèle comme des variables. Il est donc parfois nécessaire d'effectuer soi-même une régression linéaire.

- On effectue une régression linéaire avec **`price_twd_msq`** comme variable de réponse, **`n_convenience`** comme variable explicative et **`taiwan_real_estate`** comme ensemble ou jeu de données.

- On utilise la fonction **`lm()`**.

Régression linéaire avec `lm()`

```
# Effectuer une régression linéaire de price_twd_msq
en fonction de n_convenience

lm(formula = price_twd_msq ~ n_convenience, data =
taiwan_real_estate)


Call:
lm(formula = price_twd_msq ~ n_convenience, data =
taiwan_real_estate)

Coefficients:
  (Intercept)  n_convenience
       8.2242         0.7981
```

Equation:

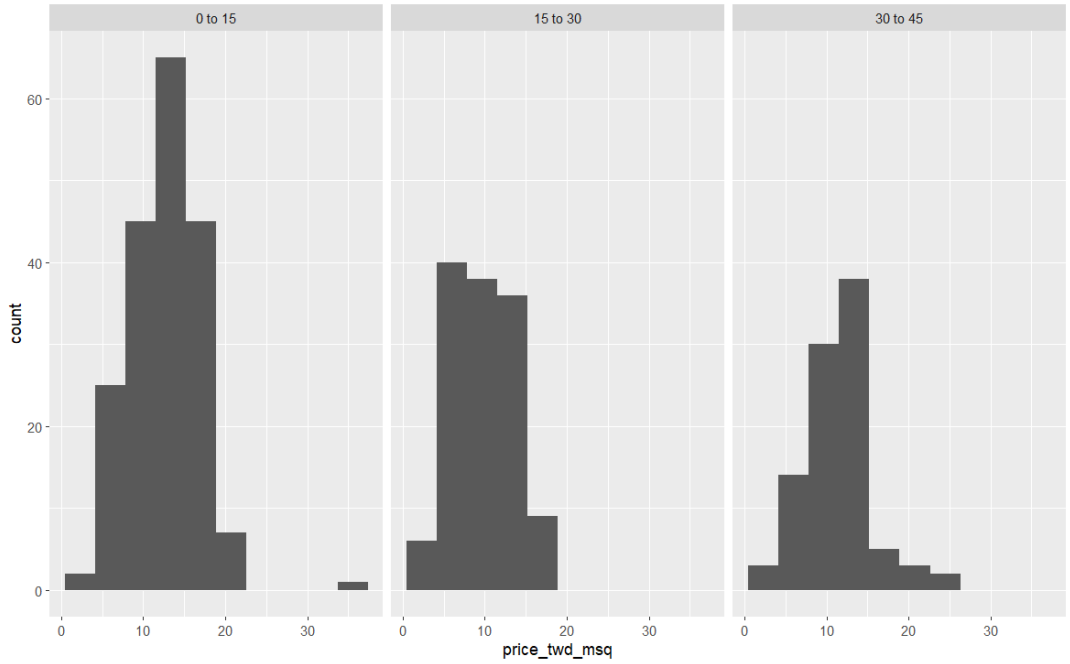
```
price_twd_msq = 8.2242 + 0.7981 * n_convenience
```

Visualisation de données numériques et catégoriques

- Si la variable explicative est catégorielle, le nuage de points utilisé précédemment pour visualiser les données n'est pas approprié. Il est préférable de tracer un histogramme pour chaque catégorie.
- L'ensemble de données sur l'immobilier taïwanais comporte une variable catégorielle : l'âge de chaque maison. Les âges sont répartis en trois groupes : 0 à 15 ans, 15 à 30 ans et 30 à 45 ans.

Visualisation de données numériques et catégoriques

```
# Représenter le price_twd_msq sous forme d'histogramme
ggplot(taiwan_real_estate, aes(x = price_twd_msq)) +
# Créer un histogramme à 10 classes.
geom_histogram(bins = 10) +
# Organiser le graphique de sorte que chaque groupe d'âge ait son propre panneau.
facet_wrap(vars(house_age_years))
```



Calcul des moyennes par catégorie

- Une bonne méthode pour explorer les variables catégoriques consiste à calculer des statistiques descriptives telles que la moyenne pour chaque catégorie. Ici, nous examinerons les moyennes groupées des prix des maisons.

```
taiwan_real_estate %>%
  # Grouper par house_age_years
  group_by(house_age_years) %>%
  # Résumer pour calculer le prix moyen des maisons
par unité de surface
  summarize(mean_price_by_group = mean(price_twd_msq))
house_age_years       mean_price_by_group
  <ord>                             <dbl>
1 0 to 15                            12.6
2 15 to 30                           9.88
3 30 to 45                           11.4
```

La fonction `lm()` avec une variable explicative catégorique

- Les régressions linéaires fonctionnent également avec des variables explicatives catégoriques. Dans ce cas, le code permettant d'exécuter le modèle est identique, mais les coefficients renvoyés diffèrent.

```
# Effectuer une régression linéaire de price_twd_msq
vs. house_age_years
lm(formula = price_twd_msq ~ house_age_years, data =
taiwan_real_estate)
Call:
lm(formula = price_twd_msq ~ house_age_years, data =
taiwan_real_estate)

Coefficients:
      (Intercept)  house_age_years.L  house_age_years.Q
          11.3025            -0.8798             1.7462
```

La fonction `lm()` avec une variable explicative catégorique

- On remarque que:
  - Il n'y a pas de coefficient pour la première catégorie.
  - Le coefficient de chaque catégorie est calculé par rapport à l'ordonnée à l'origine.
- Heureusement, on peut y remédier en modifiant légèrement la formule pour ajouter « + 0 ». On précise ainsi que tous les coefficients doivent être donnés par rapport à 0.
- Les coefficients ne sont que la moyenne pour chaque catégorie.

```
lm(formula = price_twd_msq ~ house_age_years + 0,
data = taiwan_real_estate)
Call:
lm(formula = price_twd_msq ~ house_age_years + 0, data = taiwan_real_estate)

Coefficients:
 house_age_years0 to 15  house_age_years15 to 30  house_age_years30 to 45
                 12.637                    9.877                   11.393
```

Faire des prédictions

- Le principal avantage de l'utilisation de modèles plutôt que du simple calcul de statistiques descriptives est que les modèles permettent d'effectuer des prédictions.

```
mdl_price_vs_conv <- lm(formula = price_twd_msq ~
n_convenience, data = taiwan_real_estate)

new_data <- data.frame(n_convenience = 0:10)

predict(mdl_price_vs_conv, new_data)
```

```
 1         2         3         4         5         6         7
8.224237  9.022317  9.820397 10.618477 11.416556 12.214636 13.012716

8         9         10        11
13.810795 14.608875 15.406955 16.205035
```

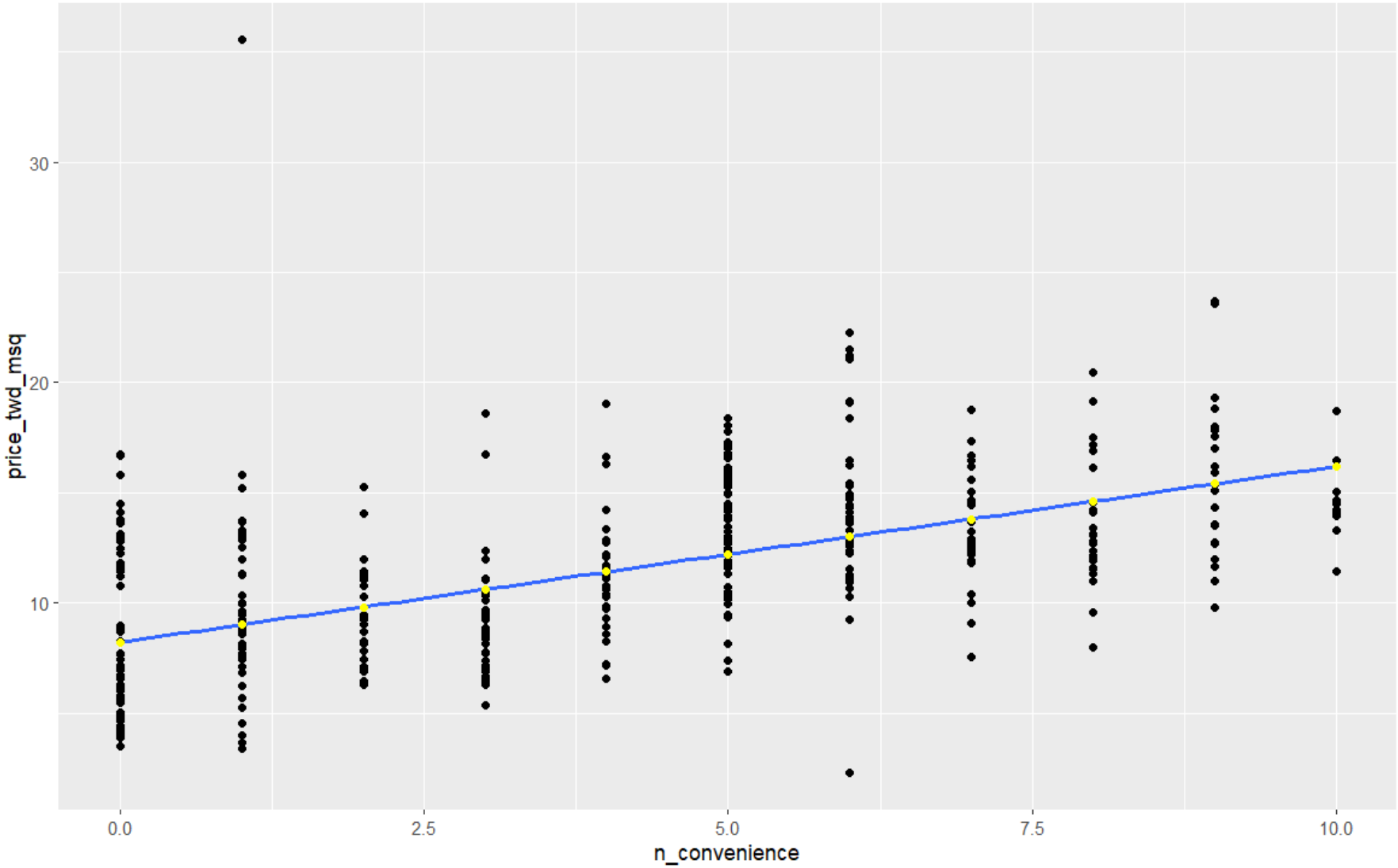
Visualisation des prédictions

```
prediction_data <- new_data

prediction_data <- prediction_data %>%
 mutate(
  price_twd_msq = predict(mdl_price_vs_conv, new_data)
 )
```

- Les données de prédiction que nous avons calculées comportent une colonne de valeurs de variables explicatives et une colonne de valeurs de variables de réponse. Nous pouvons donc les représenter sur le même nuage de points que les valeurs de la variable de réponse en fonction des variables explicatives.

Visualisation des prédictions



Les limites de la prédiction

- Dans le dernier exemple, nous avons effectué des prédictions sur des situations plausibles, susceptibles de se produire dans la réalité. Autrement dit, les cas où le nombre de supérettes à proximité était compris entre zéro et dix. Pour tester les limites de la capacité de prédiction du modèle, essayer des situations impossibles.

```
explanatory_data_2 <- data.frame(n_convenience = c(-1, 2.5))

predict(mdl_price_vs_conv, explanatory_data_2)

       1        2
 7.426158 10.219437
```

4.2. Régression Linéaire Multiple

Régression linéaire mutiple

- Un modèle de régression linéaire multiple est de la forme:

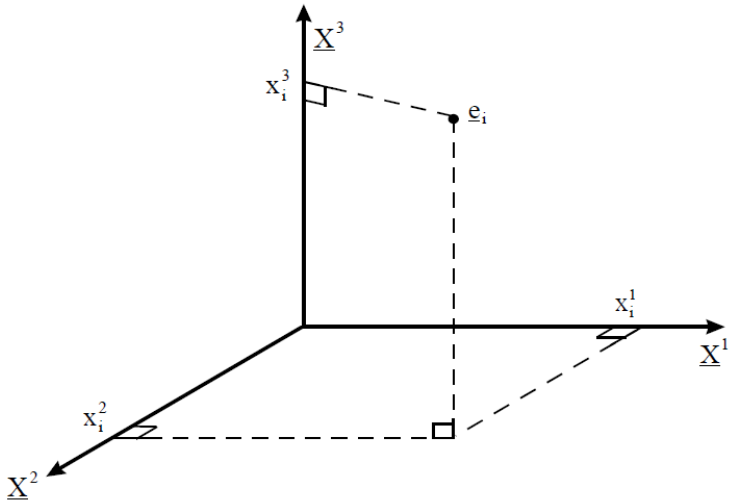
$$Y = \beta_0 + \beta_1 . X_1 + \beta_2 . X_2 + \ \ldots + + \beta_p . X_p + \epsilon$$

- Où:
  - Y est la variable dépendante (variable réponse)
  - $\beta_0, \beta_1 \ldots$ et β_p sont les coefficients
  - $X_1, X_2 \ \ldots et \ X_p$ sont les variables indépendantes (variables explicatives)
  - ϵ est une erreur aléatoire

4.3. Analyse en composantes principales (ACP)

Analyse en composantes principales (ACP)

- L'A.C.P. permet d'explorer les liaisons entre variables et les ressemblances entre individus.
- **On cherche à représenter le nuage des individus.**
  - A chaque individu noté $\boldsymbol{e_i}$, on peut associer un point dans $\boldsymbol{R^p}$ = espace des individus.
  - A chaque variable du tableau X est associé un axe de $\boldsymbol{R^p}$.

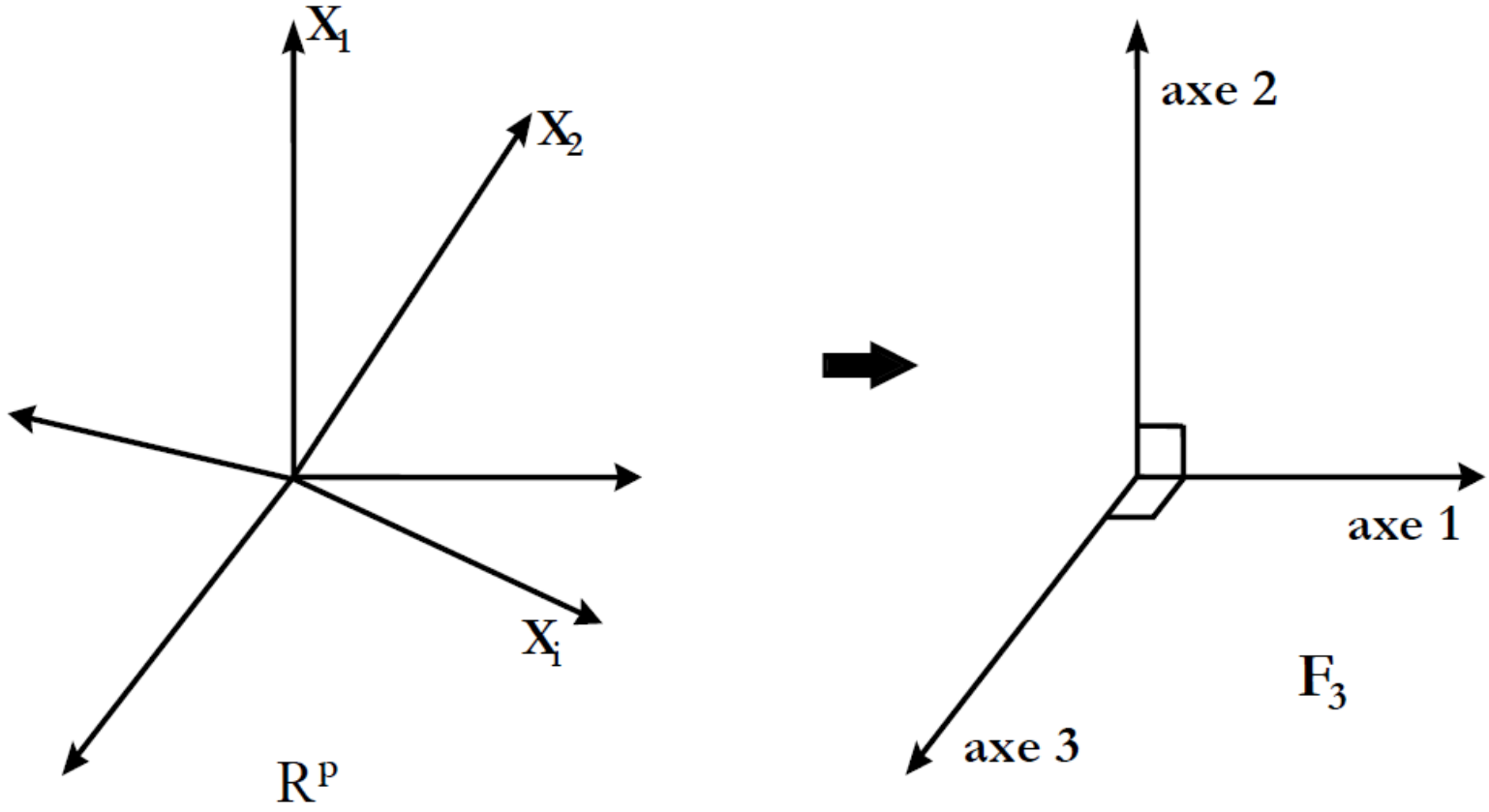


Principe de l'ACP

- On cherche une représentation des n individus , dans un sous-espace F_k de R^p de dimension k (k est petit 2, 3 …; par exemple un plan)
- Autrement dit, on cherche à définir k **nouvelles variables** obtenues comme des combinaisons linéaires des p variables initiales qui feront perdre le moins d'information possible.

- Ces variables seront appelées ***composantes principales***
- les axes qu'elles déterminent : ***axes principaux***

Axes principaux



ACP sous R

```
library(ggplot2)
library(dplyr)
# ACP
res <- prcomp(iris[,1:4], scale = TRUE)
# Coordonnées des individus
ind <- as.data.frame(res$x) %>%
  mutate(Species = iris$Species)

ggplot(ind, aes(PC1, PC2, color = Species)) +
  geom_point(size = 3, alpha = 0.8) +
  labs(title = "ACP sur iris") +
  theme_minimal()
```