

Laboratoire 10

Analyse coûts-bénéfices (ACB) pour les projets de transport

Séance pratique Excel & Python

Module : Transport durable et analyse des politiques publiques

Durée : 180–200 minutes

Contents

Objective	2
1 Fichiers & aperçu du scénario	3
2 Activité 1 – Explorer l’outil Excel USDOT	4
3 Activité 2 – Modèle ACB Python de base	5
4 Activité 3 – Modèle ACB Python étendu	8
5 Activité 4 – Devoir : Reconcevoir pour la viabilité	10

Objectif

Dans ce TP, vous allez :

- Explorer un **modèle Excel ACB USDOT** professionnel et identifier les entrées, les sorties et les indicateurs clés (VAN, RBC).
- Implémenter un **modèle ACB Python de base** à partir de données d'itinéraires, de trajets et de service.
- Étendre le modèle Python pour inclure les **coûts d'investissement, les coûts d'exploitation et des bénéfices supplémentaires** (émissions, sécurité, congestion, développement économique).
- Tester la **viabilité du projet** sous différentes hypothèses via une analyse de sensibilité.
- Reconcevoir un scénario de projet (en modifiant bénéfices, coûts ou subventions) afin de le rendre économiquement viable et interpréter les résultats VAN/RBC.

Ce manuel se concentre sur des instructions claires, étape par étape, qui correspondent aux diapositives du TP 10 et aux notebooks pratiques.:contentReference[oaicite:0]index=0

1 Fichiers & aperçu du scénario

1.1 Fichiers utilisés dans ce TP

- **Modèle Excel**
 - USDOT BCA Spreadsheet Template 11.18.24.xlsx
- **Notebooks Python**
 - `Transport_CBA_Practical_with_less_componenets.ipynb` (modèle de base)
 - `CBA_Practical_Notebook_with_extra_components.ipynb` (modèle étendu)
- **Données CSV**
 - `Routes_Table.csv`
 - `Trip_Table.csv`
 - `Service_Table.csv`

1.2 Scénario ACB transport (partie Python)

Les notebooks Python mettent en œuvre une ACB pour un projet de transport public en utilisant des données réalistes :

- Chaque ligne du jeu de données fusionné représente **un trajet** ou un service.
- Les variables clés par trajet incluent :
 - distance parcourue,
 - temps de trajet planifié et temps réel (retard),
 - coût par km,
 - bénéfice par trajet (p. ex. gains de temps).
- À partir de ces éléments, les notebooks calculent :
 - le coût de transport,
 - le coût du retard,
 - le coût total,
 - les bénéfices,
 - la Valeur Actualisée Nette (VAN) et le Ratio Bénéfices-Coûts (RBC) sur plusieurs années.

Dans le notebook étendu, des coûts et bénéfices supplémentaires au *niveau du projet* sont ajoutés (investissement, exploitation, émissions, sécurité, etc.), reflétant les composantes incluses vs manquantes discutées dans le support.

Réflexion & points clés

Considérez le fichier Excel comme une boîte noire professionnelle, et les notebooks Python comme votre manière d'ouvrir cette boîte et de reconstituer la logique pas à pas.

2 Activité 1 – Explorer l'outil Excel USDOT

Tâche 1 : Ouvrir et explorer le modèle

Objectif : Comprendre la structure d'un tableur ACB professionnel.

Étapes :

1. Ouvrez `USDOT BCA Spreadsheet Template 11.18.24.xlsx` dans Excel (ou Google Sheets si nécessaire).
2. Identifiez et parcourez les feuilles principales. En général, on trouve :
 - des feuilles d'entrée (description du projet, coûts, bénéfices),
 - des feuilles de calcul (actualisation, résultats intermédiaires),
 - des feuilles de sortie/synthèse (VAN, RBC, TRI, décision).
3. Sur la feuille des *entrées de coûts*, repérez :
 - les coûts d'investissement (construction, acquisition de terrain),
 - les coûts annuels d'exploitation & de maintenance (personnel, énergie, réparations).
4. Sur la feuille des *bénéfices*, localisez :
 - les gains de temps de déplacement,
 - les économies de coûts d'exploitation des véhicules,
 - les bénéfices liés à la sécurité, aux émissions, à la congestion ou au développement économique (si inclus).
5. Sur la feuille *résumé/sortie*, notez :
 - la Valeur Actualisée Nette (VAN),
 - le Ratio Bénéfices-Coûts (RBC),
 - tout indicateur feu tricolore ou champ de recommandation (p. ex. Économiquement viable si $RBC > 1$).

À relever :

- Quelle(s) feuille(s) affichent les coûts d'investissement ?
- Quelle(s) feuille(s) affichent les gains de temps ?
- Où la VAN et le RBC sont-ils affichés exactement (nom de la feuille, référence de cellule) ?

Tâche 2 : Relier les composantes Excel au modèle Python

Objectif : Faire le lien entre ce que vous voyez dans Excel et ce que vous allez implémenter en Python.

Étapes :

1. Faites une liste à deux colonnes dans vos notes :
 - Colonne 1 : composantes visibles dans Excel (p. ex. Coûts d'investissement , Gains de temps , Bénéfices d'émissions).
 - Colonne 2 : indiquez si ces composantes sont :
 - déjà incluses dans le notebook Python *de base*,
 - incluses uniquement dans le notebook Python *étendu*,
 - ou pas incluses du tout (pour l'instant).
2. Utilisez le support ACB inclus vs manquant comme guide : les coûts de transport et de retard sont inclus ; les coûts d'investissement, d'exploitation, ainsi que les éléments environnementaux et de sécurité ne le sont pas au départ.:contentReference[oaicite:2]index=2

Résultat attendu : Vous devriez maintenant avoir une idée claire des composantes ACB que vous allez modéliser en Python et de celles qui manquent dans la version de base.

3 Activité 2 – Modèle ACB Python de base

Cette partie utilise `Transport_CBA_Practical_with_less_componenets.ipynb` pour construire une ACB simplifiée basée principalement sur les coûts de transport et de retard, et sur un bénéfice forfaitaire par trajet.:contentReference[oaicite:3]index=3

Tâche 3 : Charger les données et inspecter les tables

Objectif : Charger les jeux de données principaux et comprendre ce que contient chaque table.

Étapes :

1. Ouvrez `Transport_CBA_Practical_with_less_componenets.ipynb` dans Google Colab.
2. Exécutez les premières cellules qui importent les bibliothèques (`pandas`, etc.).
3. Repérez la cellule qui charge les fichiers CSV, généralement quelque chose comme :

```
routes_df = pd.read_csv("Routes_Table.csv")
trips_df  = pd.read_csv("Trip_Table.csv")
service_df = pd.read_csv("Service_Table.csv")
```

4. Exécutez cette cellule et affichez le `head()` de chaque dataframe.
5. Pour chaque table, notez :
 - les colonnes clés (IDs de ligne, distance, retard, etc.),
 - l'identifiant clé utilisé pour fusionner les tables (p. ex. ID de ligne ou de trajet).

Point de contrôle : Vous devez pouvoir expliquer en une phrase ce que représente chacune des trois tables.

Tâche 4 : Fusionner les données et calculer les coûts

Objectif : Construire une table fusionnée et calculer, par trajet, les coûts de transport et de retard.

Logique attendue (les noms exacts des colonnes peuvent varier) :

- **Coût de transport** par trajet :

$$\text{Coût de transport} = \text{distance_km} \times \text{cost_per_km}$$

- **Coût du retard** par trajet :

$$\text{Coût du retard} = \max(0, \text{delay_minutes}) \times 0.5$$

(en supposant 0,5 \$ par minute de retard)

- **Coût total** par trajet :

$$\text{Coût total} = \text{Coût de transport} + \text{Coût du retard}$$

Étapes :

1. Exécutez la cellule de fusion (cherchez quelque chose comme `merged_df = ...` qui joint les trois tables).
2. Vérifiez que le dataframe fusionné contient les colonnes nécessaires : distance, retard et coût par km.
3. Exécutez la cellule qui calcule :
 - le coût de transport,
 - le coût du retard,
 - le coût total.
4. Affichez quelques lignes pour vérifier que les calculs sont cohérents (pas de retards négatifs, etc.).

À relever :

- La formule utilisée pour le coût du retard (p. ex. la pénalité par minute appliquée).
- Un exemple de ligne montrant distance, retard et coût total.

Tâche 5 : Ajouter les bénéfices par trajet et calculer VAN/RBC

Objectif : Ajouter une hypothèse simple de bénéfice et calculer la VAN et le RBC sur plusieurs années.

Étapes typiques dans le notebook :

1. Ajouter une colonne de bénéfice par trajet, par exemple :

```
merged_df["Benefit"] = 10.0    # USD per trip (baseline)
```

2. Agréger les totaux annuels :

- coûts_annuels,
- bénéfices_annuels,
- choisir la période d'analyse (p. ex. 10 ans),
- choisir le taux d'actualisation (p. ex. 5%).

3. Calculer les valeurs actualisées puis :

$$\text{VAN} = \sum_t \frac{\text{Bénéfices}_t - \text{Coûts}_t}{(1+r)^t}$$

$$\text{RBC} = \frac{\sum_t \frac{\text{Bénéfices}_t}{(1+r)^t}}{\sum_t \frac{\text{Coûts}_t}{(1+r)^t}}$$

À faire :

1. Exécutez la cellule qui définit la colonne de bénéfice (vérifiez la valeur par défaut, p. ex. 10 \$ par trajet).
2. Exécutez les cellules qui calculent les totaux annuels et les actualisent.
3. Repérez et exécutez la cellule qui affiche la **VAN** et le **RBC** de base du projet.

À relever (modèle de base) :

- VAN (modèle de base, avec le bénéfice actuel par trajet),
- RBC (modèle de base),
- si le projet est considéré comme viable ($\text{RBC} > 1$) ou non.

Tâche 6 : Analyse de sensibilité – Augmenter la pénalité de retard

Objectif : Tester la sensibilité de la VAN/RBC à des coûts de retard plus élevés.

Étapes :

1. Repérez la ligne où la pénalité par minute de retard est définie (p. ex. 0,5 US-D/min).
2. Dupliquez cette cellule plus bas et augmentez la pénalité, par exemple :

```
delay_penalty = 1.0    # instead of 0.5, for high-delay scenario
```

(ou ajustez l'expression équivalente utilisée dans votre notebook).

3. Réexécutez les cellules qui recalculent le coût du retard, le coût total, les valeurs annuelles et la VAN/RBC.
4. Notez la nouvelle VAN et le nouveau RBC pour le scénario *retard élevé*.

Comparer :

- La VAN diminue-t-elle ? De combien ?
- Le RBC passe-t-il sous 1, rendant le projet non viable ?

Réflexion & points clés

L'analyse de sensibilité sert précisément à cela : Si la réalité est pire que prévu, le projet tient-il toujours ?

4 Activité 3 – Modèle ACB Python étendu

Cette partie utilise `CBA_Practical_Notebook_with_extra_components.ipynb` pour ajouter les coûts et bénéfices manquants : coûts d'investissement, coûts d'exploitation, émissions, sécurité, congestion et développement économique.

Tâche 7 : Exécuter le notebook étendu avec les valeurs par défaut

Objectif : Obtenir la VAN et le RBC de référence pour le modèle étendu.

Étapes :

1. Ouvrez `CBA_Practical_Notebook_with_extra_components.ipynb` dans Colab.
2. Utilisez **Runtime** → **Restart and run all** pour exécuter le notebook avec les hypothèses par défaut.
3. Laissez le notebook se terminer. Faites défiler jusqu'à la fin et repérez :
 - les valeurs affichées de la VAN,
 - la valeur affichée du RBC,
 - un message final du type *Projet économiquement viable* ou *Non viable*.

À relever (référence étendue) :

- VAN (modèle étendu, valeurs par défaut),
- RBC (modèle étendu),
- recommandation de viabilité (Viable / Non viable).

Tâche 8 : Identifier les coûts et bénéfices supplémentaires

Objectif : Voir comment le modèle étendu complète les composantes manquantes .

Variables typiques dans le notebook étendu :

- **Coûts au niveau du projet**
 - `capital_cost` (ponctuel, p. ex. \$500,000),
 - `annual_operating_cost` ou similaire (p. ex. \$100,000 par an).
- **Bénéfices annuels supplémentaires**, souvent stockés dans un dictionnaire comme :

```
additional_annual_benefits = {
    "emission_reduction": 200000,
    "safety": 100000,
    "congestion": 50000,
    "economic_dev": 300000,
}
```

- Ceux-ci s'ajoutent aux bénéfices par trajet lors du calcul des bénéfices totaux actualisés : `contentReference[oaicite:6]index=6`

Étapes :

1. Repérez où `capital_cost` et les coûts annuels d'exploitation sont définis.
2. Repérez où `additional_annual_benefits` (ou équivalent) est défini.
3. Notez brièvement quels composants (émissions, sécurité, etc.) sont maintenant inclus alors qu'ils manquaient auparavant.

Tâche 9 : Test de sensibilité dans le modèle étendu

Objectif : Comparer le scénario de base et le scénario retard élevé dans le modèle étendu.

Étapes :

1. Repérez l'endroit où la pénalité par minute de retard est définie (similaire à la Tâche 6).
2. Augmentez la pénalité (par exemple la doubler) pour créer un cas retard élevé .
3. Réexécutez les cellules concernées ou l'ensemble du notebook pour obtenir la VAN et le RBC mis à jour.
4. Remplissez un petit tableau de comparaison dans vos notes :

Scénario	VAN	RBC
Référence étendue	...	...
Retard élevé (étendu)	...	...

Questions :

- L'ajout de bénéfices supplémentaires (émissions, sécurité, etc.) rend-il le projet plus robuste en cas de retards élevés ?
- Le RBC reste-t-il au-dessus de 1 dans votre cas retard élevé , ou passe-t-il en dessous ?

Réflexion & points clés

Une ACB professionnelle devrait inclure la plupart des grandes catégories de coûts et de bénéfices. Omettre des bénéfices (p. ex. émissions, sécurité) peut sous-estimer la viabilité ; omettre des coûts (p. ex. investissement, exploitation) peut la surestimer.

5 Activité 4 – Devoir : Reconcevoir pour la viabilité

Lors de nombreuses exécutions initiales, le projet peut afficher une **VAN négative** et un **RBC** ; **1**, ce qui signifie qu'il n'est pas viable. Le devoir consiste à reconcevoir le scénario afin de rendre le projet viable en ajustant les bénéfices et/ou les coûts de manière réaliste.

Tâche 10 : Exécution de référence pour le devoir

Objectif : Établir un point de départ clair dans le modèle étendu.

Étapes :

1. Ouvrez `CBA_Practical_Notebook_with_extra_components.ipynb`.
2. Assurez-vous que toutes les variables sont réglées sur leurs valeurs par défaut (vous pouvez utiliser la version fournie par l'enseignant).
3. Exécutez toutes les cellules une fois, du début à la fin.
4. Relevez soigneusement :
 - la VAN de référence,
 - le RBC de référence,
 - la recommandation du projet (Viable / Non viable).

Tâche 11 : Reconcevoir les hypothèses du projet

Objectif : Rendre le projet économiquement viable grâce à des changements réalistes des bénéfices et/ou des coûts.

Vous devez modifier au moins 2 à 3 éléments. Exemples de leviers (selon le guide) :

- **Augmenter le bénéfice par trajet** (gains de temps) :

- Par exemple, remplacer :

```
merged_df["Benefit"] = 10
```

par

```
merged_df["Benefit"] = 30
```

comme suggéré (représentant une valeur du temps plus élevée).:contentReference[oaicite:9]index=9

- **Ajouter ou augmenter des bénéfices annuels :**

- Réduction des émissions (p. ex. 200,000/an),
- Bénéfices de sécurité (p. ex. 100,000/an),
- Réduction de la congestion (p. ex. 50,000/an),
- Développement économique (p. ex. 300,000/an).
- Ajustez `additional_annual_benefits` en conséquence.

- **Réduire les coûts de manière réaliste :**

- Supposer une meilleure exploitation (coût de retard plus faible),
- Optimiser les itinéraires (coût de transport ou coût par km plus faible),
- Réduire les coûts annuels d'exploitation.

- **Subvention ou aide publique (optionnel) :**

- Le coût d'investissement peut être entièrement ou partiellement subventionné, par exemple :

```
capital_cost = 0 # fully subsidised capital
```

Contraintes :

- Les changements doivent être *plausibles* pour un projet de transport public.
- Ne modifiez pas artificiellement le taux d'actualisation, sauf si cela est explicitement autorisé.

Tâche 12 : Recalculer & comparer les scénarios

Objectif : Quantifier l'effet de votre reconception sur la viabilité du projet.

Étapes :

1. Après avoir modifié les lignes concernées, réexécutez toutes les cellules du début à la fin.
2. Relevez la **nouvelle** VAN et le nouveau RBC.
3. Créez un tableau de comparaison simple dans votre rapport ou vos notes :

Scénario	VAN	RBC
Référence (par défaut)	...	...
Reconcevoir	...	...

4. Vérifiez si le RBC est maintenant supérieur à 1 et si la VAN est positive.

Livrables (pour le devoir) :

- Fichier .ipynb modifié avec vos changements.
- Court rapport (PDF) qui :
 - explique quelles variables vous avez modifiées et pourquoi,
 - présente la VAN et le RBC avant/après,
 - indique clairement si le projet est désormais économiquement viable,
 - justifie brièvement la reconception du point de vue des politiques publiques.

Réflexion & points clés

En pratique, l'ACB est itérative : on obtient rarement un projet viable du premier coup. Les professionnels ajustent la conception, le phasage, le financement et l'exploitation jusqu'à ce que les chiffres et les objectifs de politique publique soient cohérents.