

Étude de Cas - Data Visualization

Niveau Intermédiaire - UniBank Haiti

Contexte

UniBank Haiti souhaite créer un tableau de bord exécutif pour le suivi mensuel de la performance. En tant que Data Analyst, vous devez concevoir les visualisations appropriées et justifier vos choix.

Données Disponibles

Tables

CLIENTS (50,000 lignes)

- client_id, nom, segment (Mass/Affluent/Premium), agence_id
- date_inscription, age, revenu_annuel, nb_produits

TRANSACTIONS (2M lignes)

- transaction_id, client_id, date, type (dépôt/retrait/virement)
- montant, canal (agence/mobile/web)

PRETS (15,000 lignes)

- pret_id, client_id, montant, taux, duree_mois
- date_octroi, date_echeance, solde_restant, jours_retard
- secteur (Agriculture/Commerce/Services/Industrie)

AGENCES (25 lignes)

- agence_id, nom, region, date_ouverture
-

Problème 1: Choix de Graphiques

Question 1.1

Le directeur demande de visualiser “comment les transactions sont réparties entre les canaux (agence, mobile, web)”.

Quel graphique recommandez-vous et pourquoi?

Solution 1.1

Recommandation: Diagramme en barres horizontales (bar chart)

Justification: - Seulement 3 catégories: simple à lire - Permet la comparaison précise des volumes - Les labels sont lisibles horizontalement

Alternative acceptable: Pie chart (mais limité pour comparaison précise)

Canal | Transactions
-----+-----

Mobile		45%
Agence		35%
Web		20%

Question 1.2

Le responsable risque veut voir “la distribution des montants de prêts”.

Quel graphique recommandez-vous?

Solution 1.2

Recommandation: Histogramme avec courbe de densité (KDE)

Justification: - Variable continue (montant) - Montre la forme de la distribution - Révèle l’asymétrie et les outliers - Box plot en complément pour les statistiques sommaires

```
import seaborn as sns
import matplotlib.pyplot as plt

fig, axes = plt.subplots(1, 2, figsize=(14, 5))

# Histogramme
sns.histplot(prets['montant'], kde=True, ax=axes[0])
axes[0].set_title('Distribution des Montants de Prêts')

# Box plot
axes[1].boxplot(prets['montant'])
axes[1].set_title('Box Plot des Montants')
```

Question 1.3

La direction veut suivre “l’évolution du NPL ratio sur les 24 derniers mois”.

Quel graphique et quels éléments ajouter?

Solution 1.3

Recommandation: Line chart avec éléments contextuels

Éléments à inclure: 1. Ligne du NPL ratio 2. Ligne de seuil d’alerte (5%) 3. Zone de danger colorée 4. Annotations pour événements importants 5. Moyenne mobile (optionnel)

```
plt.figure(figsize=(12, 6))
plt.plot(df['mois'], df['npl_ratio'], 'b-', linewidth=2, label='NPL Ratio')
plt.axhline(y=5, color='r', linestyle='--', label='Seuil (5%)')
plt.fill_between(df['mois'], 5, 10, alpha=0.2, color='red', label='Zone danger')
plt.xlabel('Mois')
plt.ylabel('NPL Ratio (%)')
plt.title('Évolution du NPL Ratio - 24 mois')
plt.legend()
```

Problème 2: Dashboard Design

Question 2.1

Concevez la structure d'un dashboard exécutif mensuel. Listez les 5-7 visualisations clés et leur disposition.

Solution 2.1

TABLEAU DE BORD EXÉCUTIF - UniBank Haiti
[Mois Sélectionné]

[KPI CARDS - Ligne supérieure]

PNB	ROE	NPL	CAR	Clients
+5%	14.2%	4.1%	15.2%	+3%

[ÉVOLUTION - Zone centrale]

Line Chart: PNB et Résultat Net (12 mois)
[Avec comparaison N-1 en pointillé]

[DÉTAIL - Zone inférieure]

PIE: Structure PNB	BAR: Performance par Agence
- NII: 65%	(Top 10 par volume)
- Fees: 25%	
- Other: 10%	

TABLE: KPIs par Segment (Mass/Affluent/Premium)
[Avec sparklines de tendance]

Justifications: 1. **KPI Cards en haut:** Vue d'ensemble immédiate 2. **Line chart central:** Évolution = message principal 3. **Pie chart:** Composition simple (3 éléments) 4. **Bar chart agences:** Comparaison de performance 5. **Table avec sparklines:** Détail par segment + tendance

Question 2.2

Quels filtres interactifs proposeriez-vous?

Solution 2.2

Filtres recommandés: 1. **Période:** Sélecteur de mois/trimestre/année 2. **Agence/Région:** Dropdown ou slicer 3. **Segment client:** Multi-select (Mass, Affluent, Premium) 4. **Comparison:** Toggle pour afficher/masquer N-1

Disposition: - En haut du dashboard - Visibles mais compacts - Valeurs par défaut = mois courant, toutes agences

Problème 3: Analyse Visuelle

Question 3.1

Voici les données du portefeuille de prêts par secteur:

Secteur	Encours (M)	NPL (M)	NPL %
Agriculture	800	72	9.0%
Commerce	1,200	48	4.0%
Services	600	18	3.0%
Industrie	400	12	3.0%

Créez une visualisation qui montre à la fois le volume et le risque par secteur.

Solution 3.1

Option 1: Grouped Bar Chart

```
fig, ax1 = plt.subplots(figsize=(10, 6))

x = np.arange(len(secteurs))
width = 0.35

# Barres pour l'encours
bars1 = ax1.bar(x - width/2, encours, width, label='Encours (M)', color='steelblue')

# Axe secondaire pour NPL ratio
ax2 = ax1.twinx()
bars2 = ax2.bar(x + width/2, npl_pct, width, label='NPL %', color='coral')

ax1.set_xlabel('Secteur')
ax1.set_ylabel('Encours (M)')
ax2.set_ylabel('NPL Ratio (%)')
ax1.set_xticks(x)
ax1.set_xticklabels(secteurs)
ax2.axhline(y=5, color='red', linestyle='--', alpha=0.5, label='Seuil 5%')

fig.legend(loc='upper right')
plt.title('Portefeuille de Prêts: Volume et Risque par Secteur')
```

Option 2: Bubble Chart - X = Encours - Y = NPL ratio - Taille bulle = Montant NPL

Insight attendu: Agriculture a le plus gros problème (fort NPL + volume significatif).

Question 3.2

Comment visualiseriez-vous la segmentation client (RFM) pour identifier les segments prioritaires?

Solution 3.2

Recommandation: Heatmap ou Scatter Plot coloré

```
# Scatter plot 3D en 2D avec couleur
plt.figure(figsize=(10, 8))
scatter = plt.scatter(
    df['frequency'],
    df['monetary'],
    c=df['recency'],
    s=50,
    cmap='RdYlGn_r', # Rouge = ancien, Vert = récent
    alpha=0.6
)
plt.colorbar(scatter, label='Recency (jours)')
plt.xlabel('Frequency (nb transactions)')
plt.ylabel('Monetary (total HTG)')
plt.title('Segmentation RFM')

# Annotations des segments
plt.annotate('Champions', xy=(high_f, high_m), fontsize=12)
plt.annotate('À risque', xy=(low_f, high_m), fontsize=12, color='red')
```

Alternative: Treemap - Taille = Nombre de clients - Couleur = Valeur moyenne - Hiérarchie = Segment RFM

Problème 4: Critique et Amélioration

Question 4.1

Un collègue vous montre ce graphique. Identifiez les problèmes et proposez des améliorations.

[Pie Chart avec 12 secteurs économiques,
certaines tranches très petites (1-2%),
pas de légende claire,
couleurs similaires pour certains secteurs]

Solution 4.1

Problèmes identifiés: 1. **Trop de catégories:** 12 tranches difficiles à lire 2. **Petites tranches illisibles:** 1-2% impossible à distinguer 3. **Couleurs confuses:** Certaines trop similaires 4. **Pas de valeurs:** Difficile de lire les pourcentages exacts

Améliorations proposées: 1. Utiliser un **bar chart horizontal** à la place 2. Grouper les petits secteurs dans "Autres" (< 3%) 3. Trier par valeur décroissante 4. Ajouter les valeurs sur les barres 5. Limiter à 7-8 catégories maximum

```
# Bar chart amélioré
df_sorted = df.sort_values('pct', ascending=True)
plt.barh(df_sorted['secteur'], df_sorted['pct'])
```

```
for i, v in enumerate(df_sorted['pct']):
    plt.text(v + 0.5, i, f'{v:.1f}%')
```

Problème 5: Questions d'Entretien

Question 5.1

“Comment choisiriez-vous le nombre de bins pour un histogramme?”

Solution 5.1

Plusieurs approches: 1. **Règle de Sturges:** $k = 1 + 3.322 \times \log_{10}(n)$ 2. **Règle de Freedman-Diaconis:** Basée sur IQR et n 3. **Itération:** Tester plusieurs valeurs et choisir celle qui révèle le mieux la forme

En pratique: Commencer avec 20-30 bins et ajuster pour la clarté.

Question 5.2

“Comment assureriez-vous que vos visualisations sont honnêtes et ne trompent pas l’audience?”

Solution 5.2

Principes d'intégrité visuelle: 1. **Axe Y à zéro** pour les bar charts 2. **Échelles cohérentes** lors de comparaisons 3. **Pas de dimensions 3D** inutiles 4. **Contexte fourni:** Benchmarks, périodes de comparaison 5. **Sources et dates** mentionnées 6. **Éviter le cherry-picking** des périodes 7. **Montrer l'incertitude** quand approprié

Livrables Attendus

1. **Maquette du dashboard** (schéma)
 2. **Justification des choix** de graphiques
 3. **Code Python** pour 2-3 visualisations clés
 4. **Recommandations** d'amélioration
-

Critères d'Évaluation

Critère	Points
Choix appropriés de graphiques	25%
Justification des décisions	25%
Design et lisibilité	20%
Code fonctionnel	15%
Insights business	15%

Ressources

- Données: Fournies en CSV
- Outils: Python (Matplotlib, Seaborn) ou Power BI
- Temps alloué: 2-3 heures