Manuel de Préparation: Types de Variables en Analyse de Données

Introduction

La compréhension des types de variables est fondamentale pour tout Data Analyst. Le choix des méthodes statistiques, des visualisations et des modèles dépend directement du type de données analysées. Ce manuel couvre exhaustivement tous les types de variables rencontrés en analyse de données bancaires.

Partie 1: Classification Générale des Variables

1.1 Vue d'Ensemble Hiérarchique

```
VARIABLES
│
├── QUALITATIVES (Catégorielles)
│   ├── Nominales
│   │   ├── Binaires (dichotomiques)
│   │   └── Polytomiques (multinomiales)
│   └── Ordinales
│
└── QUANTITATIVES (Numériques)
    ├── Discrètes
    │   ├── Comptages
    │   └── Entiers bornés
    └── Continues
        ├── Intervalles
        └── Ratios
```

1.2 Définitions Fondamentales

Type	Définition	Caractéristiques	Exemples Bancaires
Nominale	Catégories sans ordre	Égalité seulement (=, ≠)	Type de compte, Agence
Ordinale	Catégories avec ordre	Comparaison (<, >)	Rating crédit (AAA, AA, A)
Discrète	Valeurs dénombrables	Nombres entiers	Nombre de transactions
Continue	Valeurs sur intervalle	Mesures précises	Montant, Taux d'intérêt

---

Partie 2: Variables Qualitatives (Catégorielles)

2.1 Variables Nominales

Définition Variables dont les catégories n'ont pas d'ordre naturel ou hiérarchique.

Sous-types Variables Binaires (Dichotomiques):

```
Exactement 2 catégories mutuellement exclusives

Exemples bancaires:
- Client actif: Oui / Non
- Défaut de paiement: Oui / Non
- Genre: Masculin / Féminin
- Compte joint: Oui / Non
- Carte de crédit: Détenteur / Non-détenteur
```

Variables Polytomiques (Multinomiales):

```
Plus de 2 catégories sans ordre

Exemples bancaires:
- Type de compte: Épargne / Courant / Dépôt à terme
- Secteur d'activité: Commerce / Industrie / Services / Agriculture
- Canal préféré: Agence / Mobile / Internet / ATM
- État civil: Célibataire / Marié / Divorcé / Veuf
- Région: Nord / Sud / Ouest / Centre / Capitale
```

Opérations Permises

- Comparaison d'égalité (= ou ≠)
- Comptage et fréquences
- Mode (mesure de tendance centrale)

Traitement Python

```python
import pandas as pd

# Encodage one-hot (dummy variables)
df_encoded = pd.get_dummies(df, columns=['type_compte', 'region'])

# Encodage label (pour arbres de décision)
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['type_compte_encoded'] = le.fit_transform(df['type_compte'])

# Fréquences
df['type_compte'].value_counts()
df['type_compte'].value_counts(normalize=True)  # Proportions
```

Visualisations Appropriées

- Bar chart (diagramme en barres)
- Pie chart (< 6 catégories)
- Treemap

2.2 Variables Ordinales

Définition Variables catégorielles avec un ordre naturel entre les catégories.

Caractéristiques

- Ordre établi entre les catégories
- Intervalles entre catégories non nécessairement égaux
- Opérations arithmétiques non recommandées

Exemples Bancaires Détaillés Rating de Crédit:

```
AAA > AA > A > BBB > BB > B > CCC > CC > C > D

Investment Grade: AAA à BBB
Speculative Grade: BB et en dessous
```

Score de Satisfaction:

```
1: Très insatisfait
2: Insatisfait
3: Neutre
4: Satisfait
5: Très satisfait
```

Niveau de Risque:

```
Faible < Modéré < Élevé < Très élevé < Critique
```

Ancienneté Client:

```
Nouveau (< 1 an) < Junior (1-3 ans) < Confirmé (3-7 ans) < Senior (> 7 ans)
```

Niveau d'Éducation:

```
Primaire < Secondaire < Universitaire < Post-universitaire
```

Opérations Permises

- Comparaison d'égalité et d'ordre
- Médiane (mesure de tendance centrale)
- Mode
- Percentiles et quartiles

Traitement Python

```python
# Encodage ordinal
from sklearn.preprocessing import OrdinalEncoder

categories = [['Faible', 'Modéré', 'Élevé', 'Très élevé', 'Critique']]
oe = OrdinalEncoder(categories=categories)
df['risque_encoded'] = oe.fit_transform(df[['niveau_risque']])

# Définir l'ordre pour pandas
risk_order = pd.CategoricalDtype(
    categories=['Faible', 'Modéré', 'Élevé', 'Très élevé', 'Critique'],
    ordered=True
)
df['niveau_risque'] = df['niveau_risque'].astype(risk_order)

# Comparaisons possibles
df[df['niveau_risque'] > 'Modéré']
```

Visualisations Appropriées

- Bar chart ordonné
- Likert scale chart
- Stacked bar chart

Partie 3: Variables Quantitatives (Numériques)

3.1 Variables Discrètes

Définition Variables numériques qui ne peuvent prendre que des valeurs distinctes et dénombrables.

Sous-types Variables de Comptage:

```
Valeurs entières non négatives (0, 1, 2, 3, ...)

Exemples bancaires:
- Nombre de transactions par mois
- Nombre de produits détenus
- Nombre de jours de retard
- Nombre de réclamations
- Nombre de visites en agence
- Nombre d'appels au service client
```

Variables Entières Bornées:

```
Valeurs entières dans une plage définie

Exemples:
- Score FICO (300-850)
- Note de risque interne (1-10)
- Nombre d'échéances (1-360 pour prêts)
```

Distributions Associées

- **Poisson:** Événements rares (fraudes, défauts)
- **Binomiale:** Succès/échecs sur n essais
- **Binomiale négative:** Temps jusqu'au r-ème succès

Traitement Python

```python
# Statistiques descriptives
df['nb_transactions'].describe()

# Distribution
df['nb_transactions'].value_counts().sort_index()

# Histogramme avec bins appropriés
import matplotlib.pyplot as plt
df['nb_transactions'].plot(kind='hist', bins=range(0, 50, 1))

# Test si Poisson
from scipy import stats
```

```python
lambda_hat = df['nb_transactions'].mean()
expected = stats.poisson.pmf(range(20), lambda_hat) * len(df)
```

Visualisations Appropriées

- Histogramme (bins entiers)
- Bar chart (si peu de valeurs distinctes)
- Dot plot

3.2 Variables Continues

Définition Variables numériques pouvant prendre théoriquement n'importe quelle valeur dans un intervalle.

Niveaux de Mesure Variables d'Intervalle:

```
Définition: Zéro arbitraire, intervalles égaux significatifs
Opérations: Addition, soustraction

Exemples (rares en banque):
- Température (°C)
- Dates (année)
- Score standardisé (z-score)
```

Variables de Ratio:

```
Définition: Zéro absolu, toutes opérations permises
Opérations: Addition, soustraction, multiplication, division

Exemples bancaires:
- Montant du prêt (0 = aucun prêt)
- Solde du compte
- Taux d'intérêt
- Revenu
- Âge
- Durée de la relation (en mois)
- Montant des frais
```

Exemples Bancaires Détaillés

Variable	Unité	Plage Typique	Distribution Attendue
Solde compte	HTG	0 - 10M+	Log-normale
Montant prêt	HTG	10K - 50M	Log-normale
Taux d'intérêt	%	0 - 25%	Normale/Uniforme
Âge client	Années	18 - 90	Normale
Revenu mensuel	HTG	10K - 5M+	Log-normale
Durée relation	Mois	1 - 360+	Exponentielle
Ratio dette/revenu	%	0 - 200%+	Bimodale

Traitement Python

```python
import numpy as np
from scipy import stats

# Statistiques descriptives complètes
def analyse_continue(series):
    return {
        'count': len(series),
        'mean': series.mean(),
        'median': series.median(),
        'std': series.std(),
        'skewness': series.skew(),
        'kurtosis': series.kurtosis(),
        'min': series.min(),
        'max': series.max(),
        'Q1': series.quantile(0.25),
        'Q3': series.quantile(0.75)
    }

# Transformation log pour distributions asymétriques
df['log_solde'] = np.log1p(df['solde'])  # log(1+x) pour gérer les 0

# Test de normalité
stat, p_value = stats.shapiro(df['solde'].sample(min(5000, len(df))))
print(f"Test Shapiro-Wilk: p-value = {p_value:.4f}")
```

Visualisations Appropriées

- Histogramme
- Density plot (KDE)
- Box plot
- Violin plot
- Q-Q plot (pour normalité)

Partie 4: Types de Variables Spéciaux

4.1 Variables Temporelles (Dates et Temps)

Types Date/Datetime:

```
Exemples:
- Date d'ouverture de compte
- Date de dernière transaction
- Timestamp de connexion
```

Durée/Intervalle:

```
Exemples:
- Ancienneté en mois
- Durée du prêt
- Temps depuis dernière activité
```

Traitement Python

```python
import pandas as pd

# Conversion
df['date_ouverture'] = pd.to_datetime(df['date_ouverture'])

# Extraction de composantes
df['annee'] = df['date_ouverture'].dt.year
df['mois'] = df['date_ouverture'].dt.month
df['jour_semaine'] = df['date_ouverture'].dt.dayofweek
df['trimestre'] = df['date_ouverture'].dt.quarter

# Calcul de durées
df['ancienneté_jours'] = (pd.Timestamp.now() - df['date_ouverture']).dt.days

# Variables cycliques (mois, jour de semaine)
df['mois_sin'] = np.sin(2 * np.pi * df['mois'] / 12)
df['mois_cos'] = np.cos(2 * np.pi * df['mois'] / 12)
```

4.2 Variables Géographiques

Types Coordonnées:

- Latitude, Longitude
- Code postal
- Adresse

Régions/Zones:

Exemples bancaires:
- Département: Ouest, Artibonite, Nord, Sud, etc.
- Zone: Urbaine / Rurale
- Agence de rattachement

Traitement Python

```python
# Agrégation géographique
df.groupby('departement')['solde'].agg(['mean', 'count'])

# Distance au siège
from geopy.distance import geodesic
siege = (18.5392, -72.3362)  # Port-au-Prince
df['distance_siege'] = df.apply(
    lambda x: geodesic((x['lat'], x['lon']), siege).km, axis=1
)
```

4.3 Variables Textuelles

Types Texte Libre:

- Commentaires clients
- Notes des agents

- Motifs de réclamation

Identifiants:

- Numéro de compte (à ne pas traiter comme numérique!)
- Code client
- Numéro de carte

Traitement Python

```python
# NE JAMAIS traiter les identifiants comme numériques
df['numero_compte'] = df['numero_compte'].astype(str)

# Analyse de texte simple
df['longueur_commentaire'] = df['commentaire'].str.len()
df['nb_mots'] = df['commentaire'].str.split().str.len()

# Détection de mots-clés
df['mentionne_fraude'] = df['commentaire'].str.contains('fraude|suspicious', case=False)
```

4.4 Variables Monétaires

Caractéristiques Spéciales

- Toujours positives ou zéro (ratio)
- Souvent asymétriques (log-normale)
- Sensibles à l'inflation
- Multi-devises possible (HTG, USD)

Exemples Bancaires

- Solde du compte
- Montant de transaction
- Montant du prêt
- Intérêts courus
- Frais et commissions
- Provision pour pertes

Traitement Python

```python
# Gestion des devises
df['montant_htg'] = df.apply(
    lambda x: x['montant'] * taux_change[x['devise']], axis=1
)

# Catégorisation en tranches
bins = [0, 10000, 50000, 100000, 500000, float('inf')]
labels = ['Micro', 'Petit', 'Moyen', 'Grand', 'Très grand']
df['tranche_montant'] = pd.cut(df['montant'], bins=bins, labels=labels)

# Transformation pour modélisation
df['log_montant'] = np.log1p(df['montant'])
```

4.5 Variables de Ratio/Pourcentage

Caractéristiques

- Bornées (généralement 0-100% ou 0-1)
- Peuvent être > 100% (ex: ratio dette/revenu)
- Distribution Beta souvent appropriée

Exemples Bancaires

```
- Taux d'intérêt
- NPL ratio
- LTV (Loan-to-Value)
- Taux de défaut
- Part de marché
- Taux de provision
```

Traitement Python

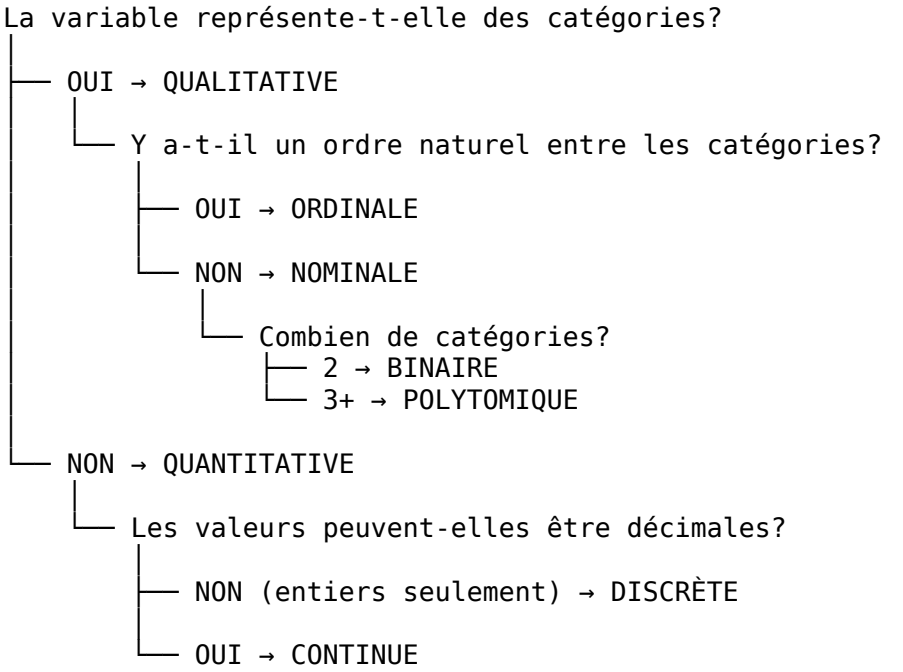
```python
# Vérification des bornes
assert df['taux_interet'].between(0, 100).all(), "Taux hors bornes"

# Transformation logit pour ratios [0,1]
def logit(p):
    return np.log(p / (1 - p))

df['logit_ratio'] = logit(df['ratio'].clip(0.001, 0.999))
```

Partie 5: Identification du Type de Variable

5.1 Arbre de Décision



```
└── Le zéro est-il significatif (absence)?
    ├── OUI → RATIO
    └── NON → INTERVALLE
```

5.2 Tests Automatisés Python

```python
def identifier_type_variable(series):
    """Identifie automatiquement le type d'une variable"""

    # Vérifier si c'est une date
    if pd.api.types.is_datetime64_any_dtype(series):
        return 'TEMPORELLE'

    # Vérifier si c'est numérique
    if pd.api.types.is_numeric_dtype(series):
        unique_ratio = series.nunique() / len(series)

        # Si peu de valeurs uniques, potentiellement ordinale/catégorielle
        if series.nunique() <= 10:
            return 'DISCRETE ou ORDINALE (à vérifier)'

        # Vérifier si entiers
        if (series == series.astype(int)).all():
            return 'DISCRETE'
        else:
            return 'CONTINUE'

    # C'est catégorielle
    n_unique = series.nunique()

    if n_unique == 2:
        return 'NOMINALE BINAIRE'
    elif n_unique <= 10:
        return 'NOMINALE ou ORDINALE (à vérifier)'
    else:
        return 'NOMINALE POLYTOMIQUE ou TEXTE'

# Application à tout le DataFrame
for col in df.columns:
    print(f"{col}: {identifier_type_variable(df[col])}")
```

Partie 6: Implications pour l'Analyse

6.1 Statistiques par Type

Type	Tendance Centrale	Dispersion	Association
Nominale	Mode	Entropie	Chi-carré, V de Cramér
Ordinale	Médiane	IQR	Spearman, Kendall
Discrète	Moyenne, Médiane	Variance, IQR	Pearson, Spearman
Continue	Moyenne, Médiane	Écart-type, IQR	Pearson, Spearman

6.2 Tests Statistiques par Type

Comparaison	Nominale	Ordinale	Numérique
2 groupes	Chi-carré	Mann-Whitney U	t-test
3+ groupes	Chi-carré	Kruskal-Wallis	ANOVA
Association	Cramér's V	Spearman ρ	Pearson r

6.3 Modèles par Type de Variable Cible

Type de Y	Modèles Appropriés
Binaire	Régression logistique, Classification
Multinomiale	Régression logistique multinomiale
Ordinale	Régression ordinale
Comptage	Régression Poisson, Binomiale négative
Continue	Régression linéaire, Arbres de régression

Partie 7: Transformations de Variables

7.1 Transformation de Type

```
# Nominale → Numérique (One-Hot)
df_encoded = pd.get_dummies(df, columns=['type_compte'])

# Continue → Ordinale (Binning)
df['tranche_age'] = pd.cut(df['age'], bins=[0, 25, 35, 50, 65, 100],
                           labels=['Jeune', 'Adulte', 'Mature', 'Senior', 'Retraité'])

# Continue → Binaire
df['haut_revenu'] = (df['revenu'] > df['revenu'].median()).astype(int)

# Ordinale → Numérique (Label Encoding)
ordre = {'Faible': 1, 'Moyen': 2, 'Élevé': 3}
df['risque_num'] = df['risque'].map(ordre)
```

7.2 Normalisation et Standardisation

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler

# Standardisation (z-score)
scaler = StandardScaler()
df['montant_std'] = scaler.fit_transform(df[['montant']])

# Normalisation (0-1)
minmax = MinMaxScaler()
df['montant_norm'] = minmax.fit_transform(df[['montant']])

# Log-transformation
df['log_montant'] = np.log1p(df['montant'])
```

Partie 8: Applications Bancaires

8.1 Exemple Complet: Analyse Client

```python
# Classification des variables d'un dataset client bancaire
client_variables = {
    'id_client': 'NOMINALE (identifiant, ne pas analyser)',
    'type_compte': 'NOMINALE POLYTOMIQUE',
    'segment': 'ORDINALE',
    'age': 'CONTINUE (RATIO)',
    'revenu_mensuel': 'CONTINUE (RATIO)',
    'nb_produits': 'DISCRETE',
    'solde_moyen': 'CONTINUE (RATIO)',
    'region': 'NOMINALE POLYTOMIQUE',
    'rating_interne': 'ORDINALE',
    'actif': 'NOMINALE BINAIRE',
    'date_ouverture': 'TEMPORELLE',
    'score_satisfaction': 'ORDINALE'
}

# Analyse adaptée à chaque type
def analyse_complete(df):
    for col, type_var in client_variables.items():
        if col not in df.columns:
            continue

        if 'NOMINALE' in type_var:
            print(f"\n{col} ({type_var}):")
            print(df[col].value_counts())

        elif 'ORDINALE' in type_var:
            print(f"\n{col} ({type_var}):")
            print(f"Médiane: {df[col].mode()[0]}")

        elif 'CONTINUE' in type_var or 'DISCRETE' in type_var:
            print(f"\n{col} ({type_var}):")
            print(df[col].describe())
```

8.2 Variables pour Scoring de Crédit

```python
# Variables typiques d'un modèle de scoring
scoring_features = {
    # Variables continues
    'revenu': 'RATIO - Log-transformer avant modélisation',
    'dette_totale': 'RATIO - Calculer ratio dette/revenu',
    'anciennete_emploi': 'RATIO - Mois',

    # Variables discrètes
    'nb_credits_actifs': 'DISCRETE - Peut être catégorisée',
    'nb_retards_12m': 'DISCRETE - Variable de risque clé',

    # Variables ordinales
    'niveau_education': 'ORDINALE - Encoder avec ordre',
    'rating_precedent': 'ORDINALE - Si client existant',
```

```
    # Variables binaires
    'proprietaire': 'BINAIRE - 1=Propriétaire, 0=Locataire',
    'employe': 'BINAIRE - 1=Employé, 0=Indépendant/Sans emploi',

    # Variables nominales
    'secteur_activite': 'NOMINALE - One-hot encoding',
    'region': 'NOMINALE - One-hot ou target encoding'
}
```

Questions d'Entretien

1. **Quelle est la différence entre variable nominale et ordinale?** → Nominale: pas d'ordre (type de compte); Ordinale: ordre naturel (rating crédit)
2. **Pourquoi ne pas traiter un numéro de compte comme numérique?** → C'est un identifiant, les opérations arithmétiques n'ont pas de sens
3. **Quelle mesure de tendance centrale pour une variable ordinale?** → La médiane (la moyenne n'est pas appropriée car intervalles inégaux)
4. **Comment traiter une variable continue très asymétrique?** → Transformation log, ou utiliser la médiane au lieu de la moyenne
5. **Quand utiliser one-hot encoding vs label encoding?** → One-hot pour nominales (pas d'ordre); Label pour ordinales (préserve l'ordre)
6. **Qu'est-ce qu'une variable de ratio vs d'intervalle?** → Ratio: zéro absolu (montant=0 signifie rien); Intervalle: zéro arbitraire (température)

Checklist Types de Variables

```
□ Identifier le type de chaque variable avant analyse
□ Ne pas traiter les identifiants comme numériques
□ Choisir les statistiques appropriées au type
□ Appliquer les visualisations adaptées
□ Encoder correctement pour la modélisation
□ Transformer si nécessaire (log, binning)
□ Vérifier les bornes et valeurs aberrantes
□ Documenter les transformations appliquées
```

Rappel final: Le type de variable détermine TOUT: statistiques, visualisations, tests, modèles. Une erreur de classification peut invalider l'ensemble de l'analyse.