Manuel de Préparation: Statistiques Descriptives

Introduction

Les statistiques descriptives sont l'ensemble des méthodes permettant de résumer, organiser et présenter des données de manière significative. C'est la base de toute analyse de données et une compétence essentielle pour un Data Analyst en milieu bancaire.

Partie 1: Concepts Fondamentaux

1.1 Types de Données

Classification des Variables

```
Variables
  Qualitatives (Catégorielles)
     Nominales: Pas d'ordre (ex: type de compte)
     Ordinales: Ordre naturel (ex: rating de crédit)

  Quantitatives (Numériques)
      Discrètes: Valeurs entières (ex: nombre de transactions)
      Continues: Valeurs mesurables (ex: montant, taux)
```

Exemples Bancaires

Variable	Type	Exemples de valeurs
Numéro de compte	Nominale	ACC-001, ACC-002
Type de client	Nominale	Particulier, Entreprise
Score de crédit	Ordinale	AAA, AA, A, BBB, BB
Satisfaction	Ordinale	1, 2, 3, 4, 5
Nombre de produits	Discrète	1, 2, 3, 4
Solde du compte	Continue	15,234.56 HTG
Taux d'intérêt	Continue	8.5%

1.2 Niveaux de Mesure

Niveau	Description	Opérations permises	Exemple
Nominal	Catégories sans ordre	Égalité (=, ≠)	Agence: PAP, CAP
Ordinal	Catégories ordonnées	Égalité + Comparaison (<, >)	Rating: A > B > C
Intervalle	Écarts significatifs, zéro arbitraire	+ Soustraction	Température
Ratio	Zéro absolu	+ Multiplication, Division	Montant, Âge

Importance: Le niveau de mesure détermine les statistiques applicables.

Partie 2: Mesures de Tendance Centrale

2.1 La Moyenne (Mean)

Définition Somme des valeurs divisée par le nombre d'observations.

Formule

$\bar{x} = (\Sigma x) / n = (x_1 + x_2 + ... + x_n) / n$

Calcul Python

```python
import numpy as np
import pandas as pd

data = [1000, 1500, 2000, 2500, 10000]

# Plusieurs méthodes
moyenne = sum(data) / len(data)          # 3400
moyenne = np.mean(data)                   # 3400
moyenne = pd.Series(data).mean()          # 3400
```

Caractéristiques

- **Avantage:** Utilise toutes les données
- **Inconvénient:** Sensible aux valeurs extrêmes (outliers)
- **Usage:** Données symétriques sans outliers

Contexte Bancaire

```
Exemple: Solde moyen des comptes = 15,000 HTG
Attention: Un gros client peut fausser cette moyenne!
```

Types de Moyennes Moyenne pondérée:

$\bar{x}_w = (\Sigma w_i x_i) / \Sigma w_i$

```
Exemple: Taux moyen pondéré par encours
Prêt A: 100M @ 10%
Prêt B: 50M @ 12%
Taux moyen = (100×10 + 50×12) / 150 = 10.67%
```

Moyenne géométrique:

$G = \sqrt[n]{(x_1 \times x_2 \times ... \times x_n)}$

```
Usage: Taux de croissance composé
```

Moyenne harmonique:

$H = n / (\Sigma(1/x_i))$

```
Usage: Moyennes de ratios
```

2.2 La Médiane (Median)

Définition Valeur qui sépare les données ordonnées en deux parties égales.

Calcul

```
Si n est impair: médiane = valeur centrale
Si n est pair: médiane = moyenne des deux valeurs centrales

Données: [1000, 1500, 2000, 2500, 10000]
Ordonnées: [1000, 1500, 2000, 2500, 10000]
Médiane = 2000 (valeur centrale, n=5)
```

Calcul Python

```python
mediane = np.median(data)           # 2000
mediane = pd.Series(data).median()  # 2000
```

Caractéristiques

- **Avantage:** Robuste aux outliers
- **Inconvénient:** N'utilise pas toutes les valeurs
- **Usage:** Données asymétriques ou avec outliers

Contexte Bancaire

```
Pour les revenus clients: Préférer la médiane
Car les très hauts revenus faussent la moyenne

Exemple:
5 clients: 20K, 25K, 30K, 35K, 500K HTG
Moyenne = 122K HTG (faussée par le client riche)
Médiane = 30K HTG (plus représentative)
```

2.3 Le Mode (Mode)

Définition Valeur la plus fréquente dans un ensemble de données.

Calcul Python

```python
from scipy import stats

data = [1, 2, 2, 3, 3, 3, 4, 5]
mode = stats.mode(data, keepdims=True)  # 3 (apparaît 3 fois)

# Avec pandas
mode = pd.Series(data).mode()  # Peut retourner plusieurs modes
```

Caractéristiques

- Peut ne pas exister (données uniformes)
- Peut être multiple (bimodal, multimodal)
- Seule mesure applicable aux données nominales

Contexte Bancaire

```
Exemple: Type de transaction le plus fréquent
Transactions: [Retrait, Retrait, Dépôt, Virement, Retrait]
Mode = Retrait
```

2.4 Comparaison et Choix

Critère	Moyenne	Médiane	Mode
Données nominales	✗	✗	✓
Données ordinales	⚠	✓	✓
Données numériques	✓	✓	✓
Robuste aux outliers	✗	✓	✓
Utilise toutes données	✓	✗	✗
Distribution symétrique	✓	✓	✓
Distribution asymétrique	✗	✓	✓

Relation avec l'Asymétrie

```
Distribution symétrique: Moyenne  Médiane  Mode

Distribution asymétrique droite (positive):
Mode < Médiane < Moyenne
(Queue vers la droite tire la moyenne)

Distribution asymétrique gauche (négative):
Moyenne < Médiane < Mode
(Queue vers la gauche tire la moyenne)
```

Partie 3: Mesures de Dispersion

3.1 L'Étendue (Range)

Définition Différence entre la valeur maximale et minimale.

```
Étendue = Max - Min

Exemple: Soldes [1000, 5000, 3000, 8000, 2000]
Étendue = 8000 - 1000 = 7000 HTG
```

Limitation Très sensible aux outliers, ne considère que 2 valeurs.

3.2 La Variance (Variance)

Définition Moyenne des carrés des écarts à la moyenne.

Formules

```
Variance population ( ²):
 ² = Σ(x  -  )² / N

Variance échantillon (s²):
s² = Σ(x  - x̄)² / (n-1)    ← Correction de Bessel
```

Calcul Python

```python
# Variance échantillon (par défaut)
variance = np.var(data, ddof=1)
variance = pd.Series(data).var()

# Variance population
variance_pop = np.var(data, ddof=0)
```

Interprétation

- Variance élevée = Données dispersées
- Variance faible = Données concentrées
- **Unité = carré de l'unité originale** (problématique pour interprétation)

3.3 L'Écart-Type (Standard Deviation)

Définition Racine carrée de la variance. Mesure la dispersion moyenne autour de la moyenne.

Formule

```
  = √( ²)     ou     s = √(s²)
```

Calcul Python

```python
ecart_type = np.std(data, ddof=1)
ecart_type = pd.Series(data).std()
```

Règle Empirique (Distribution Normale)

```
68% des données:   ± 1
95% des données:   ± 2
99.7% des données:   ± 3
```

Contexte Bancaire

```
Exemple: Montants de transactions
Moyenne = 5,000 HTG, Écart-type = 1,500 HTG

Interprétation:
- ~68% des transactions entre 3,500 et 6,500 HTG
- ~95% des transactions entre 2,000 et 8,000 HTG
- Transaction de 12,000 HTG = outlier potentiel (>3 )
```

3.4 Le Coefficient de Variation (CV)

Définition Écart-type relatif à la moyenne, exprimé en pourcentage.

Formule

```
CV = (s / x̄) × 100
```

Usage Comparer la dispersion de distributions avec des moyennes différentes.

Contexte Bancaire

```
Agence A: Moyenne 50,000 HTG,   = 10,000 → CV = 20%
Agence B: Moyenne 200,000 HTG,   = 30,000 → CV = 15%

Malgré un écart-type plus élevé, l'agence B est plus homogène.
```

3.5 Interquartile Range (IQR)

Définition Différence entre le 3ème et le 1er quartile.

```
IQR = Q3 - Q1

Où:
Q1 = 25ème percentile
Q3 = 75ème percentile
```

Calcul Python

```python
Q1 = np.percentile(data, 25)
Q3 = np.percentile(data, 75)
IQR = Q3 - Q1

# Ou directement
IQR = stats.iqr(data)
```

Usage pour Outliers

```
Outlier si:
x < Q1 - 1.5 × IQR  (outlier bas)
x > Q3 + 1.5 × IQR  (outlier haut)
```

Partie 4: Mesures de Position

4.1 Quartiles

Définition Divisent les données ordonnées en 4 parties égales.

```
Q1 (25%): 25% des données sont inférieures
Q2 (50%): = Médiane
Q3 (75%): 75% des données sont inférieures
```

Calcul Python

```
quartiles = np.percentile(data, [25, 50, 75])
# ou
Q1 = df['colonne'].quantile(0.25)
Q2 = df['colonne'].quantile(0.50)
Q3 = df['colonne'].quantile(0.75)
```

4.2 Percentiles (Centiles)

Définition Divisent les données en 100 parties égales.

```
P10: 10% des données sont inférieures
P90: 90% des données sont inférieures
```

Calcul Python

```
P10 = np.percentile(data, 10)
P90 = np.percentile(data, 90)

# Multiples percentiles
percentiles = np.percentile(data, [10, 25, 50, 75, 90])
```

Contexte Bancaire

```
Distribution des soldes clients:
P10 = 5,000 HTG (10% ont moins de 5K)
P50 = 25,000 HTG (médiane)
P90 = 150,000 HTG (10% ont plus de 150K)
P99 = 1,000,000 HTG (1% sont millionnaires)
```

4.3 Déciles

Définition Divisent les données en 10 parties égales.

```
D1 = P10, D2 = P20, ..., D9 = P90
```

Partie 5: Mesures de Forme

5.1 Asymétrie (Skewness)

Définition Mesure le degré d'asymétrie de la distribution par rapport à la moyenne.

Formule de Fisher

$$\text{Skewness} = \left[\frac{n}{(n-1)(n-2)}\right] \times \Sigma\left[\frac{x - \bar{x}}{s}\right]^3$$

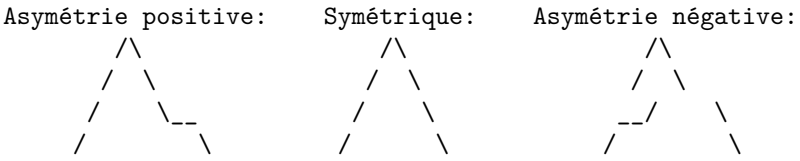
Interprétation

```
Skewness = 0: Distribution symétrique
Skewness > 0: Asymétrie positive (queue à droite)
Skewness < 0: Asymétrie négative (queue à gauche)

|Skewness| < 0.5: Approximativement symétrique
0.5   |Skewness| < 1: Modérément asymétrique
|Skewness|   1: Fortement asymétrique
```

Calcul Python

```python
from scipy.stats import skew

asymetrie = skew(data)
# ou
asymetrie = df['colonne'].skew()
```

Visualisation



Contexte Bancaire

```
Les revenus sont typiquement asymétriques à droite:
- Beaucoup de revenus moyens
- Quelques très hauts revenus
→ Skewness positif
```

5.2 Aplatissement (Kurtosis)

Définition Mesure l'épaisseur des queues de la distribution.

Formule

```
Kurtosis excess = Kurtosis - 3

(3 = kurtosis d'une distribution normale)
```

Interprétation

```
Kurtosis excess = 0: Mésokurtique (comme normale)
Kurtosis excess > 0: Leptokurtique (queues épaisses, pic aigu)
Kurtosis excess < 0: Platykurtique (queues fines, pic aplati)
```

Calcul Python

```python
from scipy.stats import kurtosis

# Excess kurtosis (Fisher's definition)
kurt = kurtosis(data, fisher=True)
# ou
kurt = df['colonne'].kurtosis()
```

Contexte Bancaire

```
Rendements financiers: Souvent leptokurtiques
→ Événements extrêmes plus fréquents que prévu par normale
→ Important pour la gestion des risques
```

Partie 6: Analyse Descriptive en Python

6.1 Résumé Statistique Complet

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

def descriptive_stats(series):
    """Calcule toutes les statistiques descriptives"""

    return {
        # Tendance centrale
        'count': len(series),
        'mean': series.mean(),
        'median': series.median(),
        'mode': series.mode().iloc[0] if not series.mode().empty else None,

        # Dispersion
        'std': series.std(),
        'var': series.var(),
        'range': series.max() - series.min(),
        'iqr': series.quantile(0.75) - series.quantile(0.25),
        'cv': (series.std() / series.mean()) * 100,

        # Position
        'min': series.min(),
        'Q1': series.quantile(0.25),
        'Q3': series.quantile(0.75),
        'max': series.max(),

        # Forme
        'skewness': series.skew(),
        'kurtosis': series.kurtosis()
    }

# Utilisation
df['solde'].pipe(descriptive_stats)
```

6.2 Avec Pandas describe()

```
# Statistiques de base
df.describe()

# Tous les percentiles
df.describe(percentiles=[.1, .25, .5, .75, .9, .95, .99])

# Pour variables catégorielles
df.describe(include=['object'])

# Tout inclus
df.describe(include='all')
```

6.3 Analyse par Groupe

```
# Statistiques par segment
df.groupby('segment')['solde'].describe()

# Statistiques personnalisées par groupe
df.groupby('segment')['solde'].agg(['mean', 'median', 'std', 'count'])

# Plusieurs colonnes, plusieurs stats
df.groupby('segment').agg({
    'solde': ['mean', 'std'],
    'nb_transactions': ['sum', 'mean'],
    'ancienneté': 'median'
})
```

Partie 7: Tableaux de Fréquences

7.1 Variables Qualitatives

```
# Fréquences absolues
freq_abs = df['type_compte'].value_counts()

# Fréquences relatives
freq_rel = df['type_compte'].value_counts(normalize=True)

# Tableau complet
freq_table = pd.DataFrame({
    'Effectif': freq_abs,
    'Fréquence': freq_rel,
    'Pourcentage': freq_rel * 100,
    'Cumul': freq_rel.cumsum() * 100
})
```

7.2 Variables Quantitatives (Classes)

```
# Créer des classes
bins = [0, 10000, 50000, 100000, 500000, float('inf')]
labels = ['<10K', '10K-50K', '50K-100K', '100K-500K', '>500K']
```

```python
df['tranche_solde'] = pd.cut(df['solde'], bins=bins, labels=labels)

# Tableau de fréquences
freq_table = df['tranche_solde'].value_counts().sort_index()
```

7.3 Tableaux Croisés (Contingence)

```python
# Table de contingence
contingence = pd.crosstab(df['segment'], df['produit'])

# Avec marges
contingence = pd.crosstab(df['segment'], df['produit'], margins=True)

# Avec pourcentages (par ligne)
contingence_pct = pd.crosstab(df['segment'], df['produit'], normalize='index')
```

Partie 8: Applications Bancaires

8.1 Analyse du Portefeuille de Prêts

```python
def analyze_loan_portfolio(df):
    """Analyse descriptive d'un portefeuille de prêts"""

    report = {}

    # Distribution des montants
    report['montant_stats'] = df['montant'].describe()
    report['montant_skew'] = df['montant'].skew()

    # Répartition par type
    report['type_distribution'] = df['type_pret'].value_counts(normalize=True)

    # Répartition par rating
    report['rating_distribution'] = df['rating'].value_counts().sort_index()

    # Concentration
    report['top10_exposure'] = df.nlargest(10, 'montant')['montant'].sum() / df['montant'].sum()

    # NPL par segment
    report['npl_by_segment'] = df.groupby('segment').apply(
        lambda x: (x['statut'] == 'NPL').sum() / len(x) * 100
    )

    return report
```

8.2 Analyse de la Clientèle

```python
def customer_analysis(df):
    """Statistiques descriptives de la base clients"""

    analysis = {}
```

```python
    # Démographie
    analysis['age_stats'] = df['age'].describe()
    analysis['age_median'] = df['age'].median()

    # Ancienneté
    analysis['tenure_stats'] = df['anciennete_mois'].describe()

    # Produits détenus
    analysis['products_per_customer'] = df['nb_produits'].describe()

    # Soldes
    analysis['balance_percentiles'] = df['solde_total'].quantile([.25, .5, .75, .9, .95])

    # Répartition géographique
    analysis['geo_distribution'] = df['region'].value_counts(normalize=True)

    return analysis
```

Partie 9: Résumé des Formules

Tendance Centrale

```
Moyenne:     x̄ = Σx  / n
Médiane:     Valeur centrale des données ordonnées
Mode:        Valeur la plus fréquente
```

Dispersion

```
Étendue:     R = Max - Min
Variance:    s² = Σ(x  - x̄)² / (n-1)
Écart-type:  s = √s²
CV:          CV = (s / x̄) × 100
IQR:         IQR = Q3 - Q1
```

Position

```
Quartiles:   Q1 (25%), Q2 (50%), Q3 (75%)
Percentiles: P_k = valeur telle que k% sont inférieurs
```

Forme

```
Skewness:    Mesure d'asymétrie (0 = symétrique)
Kurtosis:    Mesure d'aplatissement (0 = mésokurtique)
```

Questions d'Entretien Fréquentes

1. **Quand utiliser la moyenne vs la médiane?** → Médiane si outliers ou distribution asymétrique; moyenne si symétrique
2. **Qu'indique un CV élevé?** → Grande dispersion relative; données hétérogènes

3. **Comment détecter des outliers avec les statistiques descriptives?** → IQR method (< Q1-1.5×IQR ou > Q3+1.5×IQR) ou Z-score > 3
4. **Que signifie un skewness positif?** → Distribution avec queue à droite; moyenne > médiane
5. **Comment résumer une variable catégorielle?** → Tableau de fréquences, mode, proportions

---

Checklist Analyse Descriptive

```
Identifier le type de chaque variable
Calculer les mesures de tendance centrale
Calculer les mesures de dispersion
Examiner la forme de la distribution
Créer des tableaux de fréquences
Identifier les outliers potentiels
Segmenter l'analyse si pertinent
Visualiser les résultats
Interpréter dans le contexte business
```

Rappel final: Les statistiques descriptives sont la première étape de toute analyse. Elles permettent de comprendre les données AVANT d'appliquer des méthodes plus avancées.