Test Statistiques Descriptives - Test 2

Sujet: Statistiques Descriptives
Niveau: Intermédiaire
Nombre de questions: 25

Questions et Réponses

Q1. Quelle est la différence entre population et échantillon?

R1. | | Population | Échantillon | |-|———-|———-| | **Définition** | Ensemble complet | Sous-ensemble | | **Notation moyenne** | μ (mu) | $\bar{x}$ (x-bar) | | **Notation écart-type** | σ (sigma) | s | | **Diviseur variance** | N | n-1 | | **Exemple** | Tous les clients | 1000 clients sélectionnés |

Q2. Comment calculer la moyenne géométrique et quand l'utiliser?

R2. Formule:

```
G = √(x  × x  × ... × x ) = ( x )^(1/n)
```

En Python:

```
from scipy.stats import gmean
geo_mean = gmean(data)
# ou
geo_mean = np.exp(np.mean(np.log(data)))
```

Utilisation: - Taux de croissance (returns) - Ratios - Données multiplicatives - Indices

---

Q3. Qu'est-ce que la moyenne harmonique et quand l'utiliser?

R3. Formule:

```
H = n / Σ(1/x )
```

En Python:

```
from scipy.stats import hmean
harm_mean = hmean(data)
```

Utilisation: - Moyennes de ratios (vitesses, taux) - Prix moyens pour quantités égales - F-score en ML

Propriété: $G^2 = A \times H$ (Moyenne arithmétique × harmonique)

Q4. Ordonnez: moyenne arithmétique, géométrique, harmonique pour des données positives.

R4. Pour des données positives non égales:

```
Harmonique   Géométrique   Arithmétique
H   G   A
```

Égalité uniquement si toutes les valeurs sont identiques.

Q5. Calculez la moyenne, médiane et mode pour une distribution groupée:

Classe	Fréquence
0-20	5
20-40	12
40-60	18
60-80	10
80-100	5

R5. Moyenne (avec milieux de classe):

```
x̄ = Σ(f  × m ) / Σf
x̄ = (5×10 + 12×30 + 18×50 + 10×70 + 5×90) / 50
x̄ = (50 + 360 + 900 + 700 + 450) / 50
x̄ = 2460 / 50 = 49.2
```

Mode: Classe modale = 40-60 (fréquence max = 18)

Médiane: Position = 50/2 = 25ème valeur → dans classe 40-60

Q6. Comment calculer l'écart-type pour des données groupées?

R6. Formule:

```
s = √[Σf (m  - x̄)² / (n-1)]
```

En Python:

```python
# Avec les milieux de classe
midpoints = [10, 30, 50, 70, 90]
frequencies = [5, 12, 18, 10, 5]

# Reconstruire les données
data = np.repeat(midpoints, frequencies)
std = np.std(data, ddof=1)
```

Q7. Qu'est-ce que le Z-score et comment l'interpréter?

R7. Formule:

```
z = (x -  ) /    ou  z = (x - x̄) / s
```

Interprétation: - z = 0: Valeur égale à la moyenne - z = 1: 1 écart-type au-dessus de la moyenne - z = -2: 2 écarts-types en dessous

Applications: - Comparer des scores de différentes échelles - Détecter les outliers (|z| > 3) - Standardisation pour ML

Q8. Comment les percentiles sont-ils utilisés dans l'analyse de risque bancaire?

R8. Applications: 1. **Scoring crédit:** Segmentation par percentiles de score 2. **VaR (Value at Risk):** P5 ou P1 des pertes 3. **Concentration:** Top 10% des expositions 4. **Benchmark:** Comparer un client à sa distribution

Exemple:

```
# VaR 95%
var_95 = np.percentile(losses, 95)

# Seuil top 10% clients
top_10_threshold = df['revenu'].quantile(0.90)
```

Q9. Quelle est la différence entre variance et écart-type en termes d'interprétation?

R9. | Variance (s^2) | Écart-type (s) | |————|————-| | Unités au carré | Mêmes unités que les données | | Interprétation difficile | Interprétation directe | | Utile pour calculs | Utile pour communication | | Additif pour variables indép. | Non additif |

Exemple: Si montant en HTG → variance en HTG^2, écart-type en HTG.

Q10. Comment calculer la médiane pour un nombre pair et impair d'observations?

R10. Nombre impair (n):

```
Médiane = valeur à la position (n+1)/2
```

Nombre pair (n):

```
Médiane = (valeur à n/2 + valeur à n/2+1) / 2
```

Exemple: - [3, 5, 7, 9, 11] → n=5 → position 3 → médiane = 7 - [3, 5, 7, 9] → n=4 → moyenne de positions 2 et 3 → (5+7)/2 = 6

Q11. Qu'est-ce que la mesure de dispersion MAD (Median Absolute Deviation)?

R11. Formule:

```
MAD = median(|x  - median(x)|)
```

En Python:

```
from scipy.stats import median_abs_deviation
mad = median_abs_deviation(data)
```

Avantages: - Très robuste aux outliers - Point de rupture = 50% (vs 0% pour écart-type) - Relation avec σ: $\sigma \approx 1.4826 \times MAD$ (pour distribution normale)

Q12. Comment interpréter un coefficient de variation de 150%?

R12. Un CV de 150% signifie: - L'écart-type est **1.5 fois la moyenne** - **Variabilité extrême** des données - Probablement présence d'**outliers** ou distribution très asymétrique - La moyenne n'est **pas représentative**

Action recommandée: - Utiliser la médiane et l'IQR - Investiguer les valeurs extrêmes - Considérer une transformation ou segmentation

---

Q13. Comment calculer et interpréter le coefficient de Gini pour mesurer l'inégalité?

R13. Formule (simplifiée):

```
Gini = Σ|x  - x | / (2n² )
```

Interprétation: - 0 = Égalité parfaite - 1 = Inégalité maximale

Application bancaire: Concentration du portefeuille

```
def gini_coefficient(x):
    x = np.sort(x)
    n = len(x)
    index = np.arange(1, n+1)
    return (2 * np.sum(index * x) - (n+1) * np.sum(x)) / (n * np.sum(x))
```

Q14. Comment les moments statistiques sont-ils liés aux mesures descriptives?

R14. | Moment | Ordre | Mesure | |——-|——-|——-| | **1er moment centré** | 1 | Toujours 0 | | **2ème moment centré** | 2 | Variance | | **3ème moment standardisé** | 3 | Skewness | | **4ème moment standardisé** | 4 | Kurtosis |

Formule générale du moment centré:

```
m  = Σ(x  - x̄)  / n
```

Q15. Qu'est-ce que la moyenne tronquée (trimmed mean)?

R15. La **moyenne tronquée** ignore un pourcentage des valeurs extrêmes.

```
from scipy.stats import trim_mean

# Tronquer 10% de chaque côté
trimmed = trim_mean(data, proportiontocut=0.1)
```

Avantages: - Robuste aux outliers - Conserve plus d'information que la médiane - Bon compromis entre moyenne et médiane

---

Q16. Comment calculer les déciles et les utiliser pour la segmentation client?

R16.

```
# Déciles
deciles = np.percentile(df['montant'], np.arange(10, 100, 10))

# Segmentation en déciles
df['decile'] = pd.qcut(df['montant'], q=10, labels=range(1, 11))

# Analyse par décile
df.groupby('decile').agg({
    'montant': ['mean', 'sum'],
    'client_id': 'count'
})
```

Application: Analyse Pareto - "Les 10% des clients les plus importants génèrent X% du volume"

Q17. Comment mesurer la symétrie d'une distribution avec le coefficient de Pearson?

R17. Coefficient de skewness de Pearson:

```
Sk  = 3(Moyenne - Médiane) / Écart-type
```

Interprétation: - $Sk_p \approx 0$: Symétrique - $Sk_p > 0$: Asymétrie positive - $Sk_p < 0$: Asymétrie négative

Avantage: Facile à calculer avec les statistiques de base.

Q18. Comment analyser une distribution bimodale?

R18. Une distribution bimodale a deux pics distincts.

Analyse: 1. **Identifier la cause:** Mélange de deux populations? 2. **Segmenter:** Analyser chaque groupe séparément 3. **Statistiques par segment:**

```
# Clustering pour identifier les groupes
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=2)
df['cluster'] = kmeans.fit_predict(df[['variable']])

# Stats par cluster
df.groupby('cluster')['variable'].describe()
```

Q19. Comment créer un tableau croisé avec des pourcentages en ligne, colonne et total?

R19.

```
# Tableau de base
ct = pd.crosstab(df['region'], df['produit'])

# Pourcentages en ligne
ct_row = pd.crosstab(df['region'], df['produit'], normalize='index') * 100

# Pourcentages en colonne
ct_col = pd.crosstab(df['region'], df['produit'], normalize='columns') * 100

# Pourcentages du total
ct_all = pd.crosstab(df['region'], df['produit'], normalize='all') * 100

# Avec marges
ct_margins = pd.crosstab(df['region'], df['produit'], margins=True)
```

Q20. Comment calculer les statistiques descriptives pour plusieurs variables simultanément?

R20.

```
# describe() pour toutes les variables
df.describe()  # numériques par défaut
df.describe(include='all')  # toutes

# Statistiques personnalisées
df.agg(['count', 'mean', 'std', 'min', 'max', 'skew'])

# Fonction personnalisée
def custom_stats(series):
    return pd.Series({
        'n': len(series),
        'mean': series.mean(),
```

```
        'median': series.median(),
        'std': series.std(),
        'cv': series.std() / series.mean() * 100,
        'skew': series.skew(),
        'kurt': series.kurtosis()
    })

df.select_dtypes(include=[np.number]).apply(custom_stats)
```

Q21. Comment comparer les statistiques entre deux groupes?

R21.

```
# Comparaison côte à côte
comparison = df.groupby('groupe').agg({
    'montant': ['mean', 'median', 'std'],
    'age': ['mean', 'median', 'std']
})

# Différence
stats_g1 = df[df['groupe'] == 'A']['montant'].describe()
stats_g2 = df[df['groupe'] == 'B']['montant'].describe()
diff = stats_g1 - stats_g2

# Ratio
ratio = stats_g1 / stats_g2
```

Q22. Comment interpréter un box plot avec des outliers?

R22.

```
    *           ← Outlier (> Q3 + 1.5×IQR)
    |
              ← Maximum (sans outliers)


               ← Q3 (75ème percentile)
              ← Médiane (Q2)
               ← Q1 (25ème percentile)


              ← Minimum (sans outliers)
    |
    *           ← Outlier (< Q1 - 1.5×IQR)
```

La boîte contient 50% des données (IQR)

Q23. Quelle est la différence entre corrélation de Pearson et de Spearman?

R23. | Pearson | Spearman | |———|———-| | Relations **linéaires** | Relations **monotones** | | Sensible aux outliers | Robuste aux outliers | | Données continues | Données ordinales acceptées | | Basé sur valeurs | Basé sur **rangs** |

```python
# Pearson
pearson_corr = df['x'].corr(df['y'], method='pearson')

# Spearman
spearman_corr = df['x'].corr(df['y'], method='spearman')
```

Q24. Comment résumer les statistiques pour un rapport exécutif?

R24. Format recommandé:

Métrique	Valeur	Interprétation
N	10,000	Taille de l'échantillon
Moyenne	85,000 HTG	Tendance centrale
Médiane	52,000 HTG	Valeur typique
Écart-type	95,000 HTG	Dispersion
Min - Max	1K - 2M HTG	Étendue
P25 - P75	20K - 100K HTG	Plage centrale

Message clé: "Le montant médian est de 52K HTG avec une forte variabilité (CV=112%), suggérant des segments clients distincts."

Q25. Analysez et interprétez le tableau suivant pour deux agences:

Statistique	Agence A	Agence B
N clients	5,000	2,000
Montant moyen	100K	120K
Montant médian	95K	60K
Écart-type	40K	150K
Skewness	0.3	2.5

R25. Analyse:

Agence A: - Distribution quasi-symétrique (skew=0.3, moyenne≈médiane) - Variabilité modérée (CV=40%) - Clientèle homogène

Agence B: - Distribution très asymétrique (skew=2.5, moyenne » médiane) - Variabilité extrême (CV=125%) - Présence de quelques très gros clients - Médiane plus représentative (60K vs 120K)

Recommandations: - Agence A: Utiliser la moyenne pour le reporting - Agence B: Utiliser la médiane, investiguer les gros clients - Segmenter Agence B pour comprendre les deux populations

---

Scoring

Score	Niveau
0-10	À améliorer

Score	Niveau
11-17	Intermédiaire
18-22	Avancé
23-25	Expert