Test Statistiques Inférentielles - Test 2

Sujet: Enquêtes et Tests d'Hypothèses
Niveau: Intermédiaire
Nombre de questions: 25

Questions et Réponses

Q1. Quels sont les principaux types d'échantillonnage et leurs caractéristiques?

R1. | Type | Description | Usage | |——|————-|——-| | **Aléatoire simple** | Chaque individu a même probabilité | Population homogène | | **Stratifié** | Division en strates, puis aléatoire | Groupes distincts | | **Par grappes** | Sélection de groupes entiers | Coûts logistiques | | **Systématique** | 1 sur k (ex: 1 sur 10) | Listes ordonnées |

Q2. Pourquoi l'échantillonnage stratifié est-il préférable pour une banque?

R2. Avantages: - **Représentation garantie** de tous les segments (PME, particuliers, corporates) - **Précision accrue** pour les analyses par segment - **Efficacité:** Moins d'échantillons nécessaires pour même précision

Exemple:

```
from sklearn.model_selection import train_test_split

# Stratifié sur segment client
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, stratify=df['segment']
)
```

Q3. Qu'est-ce que l'erreur d'échantillonnage et comment la réduire?

R3. Définition: Différence entre la statistique de l'échantillon et le paramètre de la population.

Erreur standard (SE):

```
SE = s / √n
```

Réduction: 1. Augmenter n (inverse de √n) 2. Utiliser un échantillonnage plus efficace (stratifié) 3. Réduire la variabilité (meilleur contrôle)

Q4. Comment formuler correctement les hypothèses pour un test?

R4. Règles: 1. H_0 contient toujours le signe **=** 2. H_1 contient <, >, ou ≠ 3. Le paramètre testé doit être le même

Exemples:

```
Test bilatéral:
H :   = 50,000    H :    50,000

Test unilatéral (supérieur):
H :    50,000    H :   > 50,000
```

```
Test unilatéral (inférieur):
H :    50,000    H :   < 50,000
```

Q5. Quelle est la relation entre niveau de confiance, marge d'erreur et taille d'échantillon?

R5.

```
Marge d'erreur = z × (s/√n)
```

Relations: - ↑ Confiance → ↑ z → ↑ Marge d'erreur (ou ↑ n) - ↓ Marge d'erreur → ↑ n nécessaire - ↑ n → ↓ Marge d'erreur

Trade-off: Pour plus de précision avec même confiance, il faut plus d'observations.

Q6. Comment interpréter un résultat "non significatif"?

R6. Ce que ça signifie: - Les données ne fournissent pas assez d'évidence pour rejeter H_0 - On ne peut pas conclure qu'il y a une différence

Ce que ça NE signifie PAS: - H_0 est vraie - Il n'y a pas de différence

Causes possibles: - Effet trop faible - Échantillon trop petit - Variabilité trop grande

---

Q7. Qu'est-ce qu'un test post-hoc et quand l'utiliser?

R7. Définition: Tests de comparaisons multiples après une ANOVA significative.

Objectif: Identifier quels groupes diffèrent.

```python
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Test de Tukey
tukey = pairwise_tukeyhsd(df['montant'], df['agence'], alpha=0.05)
print(tukey.summary())
```

Types: - **Tukey HSD:** Comparaisons toutes paires - **Bonferroni:** Plus conservateur - **Scheffé:** Plus flexible

---

Q8. Comment tester si deux distributions sont identiques?

R8. Test de Kolmogorov-Smirnov:

```python
from scipy.stats import ks_2samp

stat, p_value = ks_2samp(distribution1, distribution2)

# H : Les deux distributions sont identiques
if p_value < 0.05:
    print("Distributions différentes")
```

Utilisation: Détecter le data drift entre périodes.

Q9. Comment réaliser un test de corrélation?

R9.

```python
from scipy.stats import pearsonr, spearmanr

# Pearson (linéaire)
r, p_value = pearsonr(x, y)

# Spearman (monotone, robuste)
rho, p_value = spearmanr(x, y)

# H :   = 0 (pas de corrélation)
if p_value < 0.05:
    print(f"Corrélation significative: r = {r:.3f}")
```

Q10. Qu'est-ce que le bootstrap et comment l'utiliser?

R10. Définition: Méthode de rééchantillonnage avec remise pour estimer la distribution d'une statistique.

```python
import numpy as np

def bootstrap_ci(data, n_bootstrap=10000, ci=95):
    means = []
    for _ in range(n_bootstrap):
        sample = np.random.choice(data, size=len(data), replace=True)
        means.append(np.mean(sample))

    lower = np.percentile(means, (100-ci)/2)
    upper = np.percentile(means, 100-(100-ci)/2)
    return lower, upper

ci_low, ci_high = bootstrap_ci(df['montant'])
```

Avantages: Pas d'hypothèse sur la distribution.

Q11. Comment calculer et interpréter l'odds ratio?

R11. Formule:

```
OR = (a/b) / (c/d) = (a×d) / (b×c)
```

Pour un tableau 2×2: | | Défaut | Non-défaut | |———-|——-|————| | Exposé | a | b | | Non-exposé| c | d |

Interprétation: - OR = 1: Pas d'association - OR > 1: Exposition augmente le risque - OR < 1: Exposition réduit le risque

---

Q12. Comment tester l'homogénéité des variances?

R12. Test de Levene (robuste):

```python
from scipy.stats import levene

stat, p_value = levene(group1, group2, group3)

# H : Les variances sont égales
```

```python
if p_value < 0.05:
    print("Variances inégales - utiliser test de Welch")
```

Impact: Si variances inégales, utiliser t-test de Welch ou tests non-paramétriques.

Q13. Comment calculer la puissance a posteriori?

R13.

```python
from statsmodels.stats.power import TTestIndPower

analysis = TTestIndPower()

# Puissance observée
power = analysis.solve_power(
    effect_size=0.5,     # Cohen's d observé
    nobs1=100,           # Taille groupe 1
    ratio=1,             # Ratio des tailles
    alpha=0.05
)
print(f"Puissance: {power:.1%}")
```

Si puissance < 80%: Le test peut ne pas avoir détecté un vrai effet.

Q14. Comment réaliser un test de McNemar pour données appariées?

R14. Usage: Comparer deux proportions sur les mêmes sujets (avant/après).

```python
from statsmodels.stats.contingency_tables import mcnemar

# Tableau de changement
#            Après +   Après -
# Avant +      a         b
# Avant -      c         d
table = [[a, b], [c, d]]

result = mcnemar(table, exact=True)
print(f"p-value: {result.pvalue}")
```

Q15. Comment interpréter un test de Kruskal-Wallis?

R15. Alternative non-paramétrique à l'ANOVA.

```python
from scipy.stats import kruskal

stat, p_value = kruskal(group1, group2, group3)

# H : Les distributions sont identiques
# H : Au moins une distribution diffère
```

Si p < 0.05: Utiliser tests post-hoc (Dunn's test) pour identifier les différences.

Q16. Qu'est-ce que le test de Fisher exact?

R16. Alternative au Chi-carré quand les effectifs attendus sont < 5.

```python
from scipy.stats import fisher_exact

# Tableau 2x2
table = [[a, b], [c, d]]
odds_ratio, p_value = fisher_exact(table)
```

Avantage: Calcul exact de la p-value (pas d'approximation).

Q17. Comment réaliser une analyse de variance à deux facteurs?

R17. ANOVA à 2 facteurs: Teste l'effet de deux variables catégorielles et leur interaction.

```python
import statsmodels.api as sm
from statsmodels.formula.api import ols

# Modèle avec interaction
model = ols('montant ~ C(region) + C(produit) + C(region):C(produit)', data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)
print(anova_table)
```

Interprétation: p-values pour chaque facteur et l'interaction.

Q18. Comment gérer les données manquantes dans un test statistique?

R18. Options: 1. **Suppression:** Si peu de manquants (< 5%) 2. **Imputation:** Moyenne, médiane, ou multiple 3. **Méthodes robustes:** Utiliser des tests qui ignorent les NA

```python
# Test t en ignorant les NA
from scipy.stats import ttest_ind

group1_clean = group1.dropna()
group2_clean = group2.dropna()
t, p = ttest_ind(group1_clean, group2_clean)
```

Important: Documenter le traitement des manquants.

Q19. Comment réaliser un test de tendance?

R19. Test de Cochran-Armitage: Tendance dans les proportions.

```python
from scipy.stats import spearmanr

# Alternative: corrélation de Spearman avec variable ordinale
rho, p = spearmanr(df['niveau_risque_ordinal'], df['defaut'])

# Ou régression logistique pour tendance
import statsmodels.api as sm
model = sm.Logit(df['defaut'], sm.add_constant(df['score'])).fit()
print(model.summary())
```

Q20. Comment interpréter les résultats d'une régression logistique?

R20.

```python
import statsmodels.api as sm

X = sm.add_constant(df[['score', 'revenu']])
model = sm.Logit(df['defaut'], X).fit()
print(model.summary())
```

Interprétation: - **Coef:** Log-odds ratio - **Exp(coef):** Odds ratio - **p-value:** Significativité du prédicteur - **Pseudo R²:** Ajustement (différent de R² classique)

Q21. Qu'est-ce que l'ajustement pour facteurs de confusion?

R21. Facteur de confusion: Variable associée à la fois à l'exposition et au résultat.

Méthodes d'ajustement: 1. **Stratification:** Analyse séparée par strate 2. **Régression:** Inclure le facteur comme covariable 3. **Appariement:** Design de l'étude

Exemple:

```python
# L'âge pourrait confondre la relation score-défaut
model = sm.Logit(df['defaut'], sm.add_constant(df[['score', 'age']])).fit()
```

Q22. Comment calculer l'intervalle de confiance pour une proportion?

R22. Méthode de Wald (approximation normale):

```
IC = p̂ ± z × √(p̂(1-p̂)/n)
```

```python
from statsmodels.stats.proportion import proportion_confint

# Méthode Wilson (recommandée)
ci_low, ci_high = proportion_confint(successes, n, method='wilson')

# Exemple: 48 défauts sur 1000
ci = proportion_confint(48, 1000, method='wilson')
# → [0.036, 0.063]
```

Q23. Comment tester la normalité multivariée?

R23.

```python
from scipy.stats import shapiro
import pingouin as pg

# Mardia's test pour normalité multivariée
# (si pingouin installé)
result = pg.multivariate_normality(df[['var1', 'var2', 'var3']])
print(result)

# Alternative: vérifier chaque variable
for col in ['var1', 'var2', 'var3']:
    stat, p = shapiro(df[col])
    print(f"{col}: p = {p:.4f}")
```

Q24. Comment présenter les résultats d'un test dans un rapport?

R24. Format standard:

> "Un test t indépendant a été réalisé pour comparer le score de crédit entre les clients en défaut et non-défaut. Il existe une différence significative entre les clients en défaut (M = 520, SD = 85) et non-défaut (M = 680, SD = 95); t(998) = -8.45, p < 0.001, d = 1.2. Le score de crédit des clients en défaut est significativement plus bas, avec une taille d'effet large."

Éléments: - Test utilisé - Statistiques descriptives par groupe - Statistique du test et degrés de liberté - p-value - Taille d'effet - Conclusion en langage clair

---

Q25. Analysez ce scénario et proposez le test approprié:

"UniBank veut savoir si le taux de satisfaction (1-5) diffère selon le canal (Agence, Mobile, Web) et le segment client (Particulier, PME). N = 500."

R25. Analyse: - 2 facteurs catégoriels (canal, segment) - 1 variable ordinale (satisfaction 1-5) - Possible intéraction

Test recommandé: ANOVA à 2 facteurs ou équivalent non-paramétrique.

```python
# Option 1: ANOVA à 2 facteurs (si normalité acceptable)
from statsmodels.formula.api import ols
import statsmodels.api as sm

model = ols('satisfaction ~ C(canal) * C(segment)', data=df).fit()
anova_table = sm.stats.anova_lm(model, typ=2)

# Option 2: Si non-normalité, utiliser des tests non-paramétriques
# Kruskal-Wallis par facteur + analyse de l'interaction

# Hypothèses testées:
# H : Pas d'effet canal
# H : Pas d'effet segment
# H : Pas d'interaction canal×segment
```

Suivi si significatif: Tests post-hoc pour identifier les différences spécifiques.

Scoring

Score	Niveau
0-10	À améliorer
11-17	Intermédiaire
18-22	Avancé
23-25	Expert