Manuel de Révision Complet - Data Analyst UniBank Haiti

Plan de Révision (Dernière Journée)

Heure	Activité	Durée
08:00	Statistiques et Probabilités	1h30
09:30	Pause	15 min
09:45	SQL et Machine Learning	1h30
11:15	Pause	15 min
11:30	BI Bancaire et KPIs	1h
12:30	Déjeuner	45 min
13:15	Python et Visualisation	1h
14:15	Pause	15 min
14:30	Études de Cas	1h30
16:00	Fiches de Synthèse	30 min

---

1. STATISTIQUES ESSENTIELLES

1.1 Statistiques Descriptives - À Retenir

Mesures de tendance centrale: - **Moyenne:** Sensible aux outliers, utiliser quand données symétriques - **Médiane:** Robuste, préférer quand outliers ou asymétrie - **Mode:** Pour variables catégorielles

Mesures de dispersion: - **Écart-type:** Interprétable dans les mêmes unités - **Variance:** Carré de l'écart-type - **IQR:** Q3 - Q1, robuste pour détecter outliers - **CV:** Permet de comparer des dispersions différentes

Règle 68-95-99.7 (Distribution normale):

```
68% dans μ ± 1σ
95% dans μ ± 2σ
99.7% dans μ ± 3σ
```

1.2 Tests d'Hypothèses - Procédure

```
1. Formuler H₀ et H₁
2. Choisir α (généralement 0.05)
3. Calculer la statistique de test
4. Calculer la p-value
5. Si p-value < α → Rejeter H₀
6. Interpréter dans le contexte
```

Choix du test: | Comparer | Test Paramétrique | Non-Paramétrique | |———-|———————-|———————| | 1 moyenne vs valeur | t-test 1 sample | Wilcoxon signed-rank | | 2 moyennes indép. | t-test indépendant | Mann-Whitney U | | 2 moyennes appariées | t-test apparié | Wilcoxon signed-rank | | 3+ moyennes | ANOVA | Kruskal-Wallis | | 2 proportions | z-test | Chi-carré | | Indépendance | - | Chi-carré |

1.3 Corrélation - Points Clés

```
Pearson (r): Relation linéaire, données continues, normales
```

```
Spearman (ρ): Relation monotone, ordinales ou non-linéaires

Interprétation |r|:
< 0.3: Faible
0.3-0.7: Modérée
> 0.7: Forte

ATTENTION: Corrélation ≠ Causalité
```

1.4 Probabilités - Formules

```
Bayes: P(A|B) = [P(B|A) × P(A)] / P(B)

Distributions:
- Binomiale: P(X=k) = C(n,k) × p^k × (1-p)^(n-k)
- Poisson: P(X=k) = (λ^k × e^-λ) / k!
- Normale: Z = (X - μ) / σ
```

---

2. SQL POUR DATA ANALYST

2.1 Window Functions - Synthèse

```sql
-- Classement
ROW_NUMBER() -- Numéro unique
RANK()       -- Saute si égalité (1,2,2,4)
DENSE_RANK() -- Ne saute pas (1,2,2,3)
NTILE(n)     -- Divise en n groupes

-- Navigation
LAG(col, n)  -- n lignes avant
LEAD(col, n) -- n lignes après
FIRST_VALUE(col) -- Première valeur
LAST_VALUE(col)  -- Dernière valeur

-- Agrégation
SUM() OVER (ORDER BY date) -- Cumul
AVG() OVER (ROWS BETWEEN 6 PRECEDING AND CURRENT ROW) -- MA 7
```

2.2 CTE et Sous-requêtes

```sql
-- CTE simple
WITH stats AS (
    SELECT agence, SUM(montant) as total
    FROM transactions
    GROUP BY agence
)
SELECT * FROM stats WHERE total > 100000;

-- CTE multiple
WITH
cte1 AS (SELECT ...),
```

```sql
cte2 AS (SELECT ... FROM cte1 ...)
SELECT ... FROM cte2;
```

2.3 Patterns Utiles

```sql
-- Top N par groupe
WITH ranked AS (
    SELECT *, ROW_NUMBER() OVER (
        PARTITION BY agence ORDER BY solde DESC
    ) as rn
    FROM clients
)
SELECT * FROM ranked WHERE rn <= 5;

-- YoY Comparison
SELECT
    mois,
    total,
    LAG(total, 12) OVER (ORDER BY mois) as total_n1,
    (total - LAG(total, 12) OVER (ORDER BY mois)) /
        NULLIF(LAG(total, 12) OVER (ORDER BY mois), 0) * 100 as var_pct
FROM monthly_sales;

-- Cumul running
SELECT
    date,
    montant,
    SUM(montant) OVER (ORDER BY date) as cumul
FROM transactions;
```

2.4 Optimisation

```
 Bonnes pratiques:
- SELECT colonnes spécifiques
- Index sur colonnes WHERE, JOIN, ORDER BY
- EXISTS plutôt que IN pour sous-requêtes
- LIMIT pour limiter les résultats

 À éviter:
- SELECT *
- Fonctions sur colonnes indexées dans WHERE
- OR sur colonnes différentes
- Sous-requêtes corrélées si possible
```

---

3. MACHINE LEARNING - RÉSUMÉ

3.1 Types d'Apprentissage

```
SUPERVISÉ (avec labels)
├── Classification: prédire une catégorie
│   └── Défaut (oui/non), Fraude, Segment
└── Régression: prédire une valeur continue
```

```
    └── Montant, Score, LTV

NON SUPERVISÉ (sans labels)
├── Clustering: groupes naturels
│   └── Segmentation clients
└── Détection d'anomalies
    └── Fraude, Outliers
```

3.2 Algorithmes de Classification

```python
# Régression Logistique - Scoring crédit
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
proba = model.predict_proba(X_test)[:, 1]

# Interprétation: Odds Ratio = exp(coefficient)

# Random Forest - Fraude
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators=100)
rf.fit(X_train, y_train)

# Gradient Boosting - Performance
import xgboost as xgb
xgb_model = xgb.XGBClassifier(n_estimators=100)
```

3.3 Métriques d'Évaluation

```
CLASSIFICATION:
- Accuracy = (TP + TN) / Total
- Precision = TP / (TP + FP)  → Fiabilité prédictions +
- Recall = TP / (TP + FN)     → Couverture vrais +
- F1 = 2 × (P × R) / (P + R)
- AUC-ROC = Aire sous la courbe
- Gini = 2 × AUC - 1

RÉGRESSION:
- MAE = Moyenne |erreur|
- RMSE = √(Moyenne erreur²)
- R² = Variance expliquée
```

3.4 Patterns Courants Bancaires

```python
# Scoring crédit
def scoring_credit(df):
    features = ['revenu', 'dette_ratio', 'anciennete', 'nb_retards']
    X = df[features]
    y = df['defaut']

    model = LogisticRegression()
    model.fit(X_train, y_train)
    df['PD'] = model.predict_proba(X)[:, 1]
```

```python
    # Score = 600 - 20 × log2(odds)
    df['Score'] = 600 - 20 * np.log2(df['PD'] / (1 - df['PD']))
    return df

# Détection fraude
from sklearn.ensemble import IsolationForest
iso = IsolationForest(contamination=0.01)
df['anomalie'] = iso.fit_predict(X)  # -1 = anomalie

# Segmentation
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=5)
df['segment'] = kmeans.fit_predict(X_scaled)
```

---

4. TYPES DE VARIABLES - RAPPEL ESSENTIEL

4.1 Classification Hiérarchique

```
VARIABLES
├── QUALITATIVES (Catégorielles)
│   ├── Nominales: pas d'ordre (type compte, région, genre)
│   │   ├── Binaires: 2 catégories (oui/non, défaut/non-défaut)
│   │   └── Polytomiques: 3+ catégories (secteur, canal)
│   └── Ordinales: ordre naturel (rating AAA>AA>A, satisfaction 1-5)
│
└── QUANTITATIVES (Numériques)
    ├── Discrètes: valeurs entières (nb transactions, nb produits)
    └── Continues: valeurs décimales (montant, taux, âge)
```

4.2 Niveaux de Mesure

Niveau	Opérations	Exemple Bancaire
Nominal	= ≠	Type de compte, Agence
Ordinal	= ≠ < >	Rating crédit, Satisfaction
Intervalle	+ -	Score standardisé
Ratio	× ÷	Montant, Revenu, Âge

4.3 Implications pour l'Analyse

Type Variable	Tendance Centrale	Test Statistique	Encodage ML
Nominale	Mode	Chi-carré	One-Hot
Ordinale	Médiane	Mann-Whitney, Spearman	Label ordered
Quantitative	Moyenne, Médiane	t-test, Pearson	StandardScaler

4.4 Erreurs à Éviter

```
 Traiter un ID comme numérique (numéro de compte)
 Calculer la moyenne d'une variable ordinale (satisfaction)
 One-Hot encoder une variable ordinale (perd l'ordre)
 Oublier de normaliser avant K-Means
```

5. KPIs BANCAIRES

5.1 Rentabilité

KPI	Formule	Benchmark
ROE	Résultat Net / Capitaux Propres	12-18%
ROA	Résultat Net / Total Actifs	1-2%
NIM	(Rev. Int. - Ch. Int.) / Actifs Prod.	3-5%
CIR	Charges Exploit. / PNB	45-55%

5.2 Qualité des Actifs

KPI	Formule	Benchmark
NPL Ratio	Prêts > 90j / Total Prêts	< 5%
Coverage	Provisions / NPL	> 100%
Cost of Risk	Dotations Prov. / Encours	1-3%

4.3 Solvabilité et Liquidité

KPI	Formule	Exigence
CAR	Fonds Propres / RWA	≥ 12% (BRH)
LDR	Prêts / Dépôts	80-90%
LCR	HQLA / Sorties 30j	≥ 100%

5.4 Commercial

KPI	Formule	Usage
Cross-sell	Nb Produits / Nb Clients	Engagement
Churn	Clients Perdus / Clients Début	Rétention
CAC	Coûts Acquisition / Nouveaux Clients	Efficacité marketing
LTV	Revenu × Durée × Marge	Valeur client
NPS	% Promoteurs - % Détracteurs	Satisfaction

6. PYTHON - RAPPELS

6.1 Pandas Essentiels

```
# Chargement et exploration
df = pd.read_csv('file.csv')
df.head(), df.info(), df.describe()
df.shape, df.columns, df.dtypes

# Valeurs manquantes
```

```
df.isnull().sum()
df.fillna(df['col'].median())
df.dropna(subset=['col'])

# Filtrage
df[df['col'] > 100]
df[(cond1) & (cond2)]
df.query('col > 100 and type == "A"')

# Agrégation
df.groupby('cat')['val'].agg(['sum', 'mean', 'count'])
df.pivot_table(values='val', index='row', columns='col', aggfunc='sum')

# Transformation
df['new'] = df['a'] / df['b']
df['cat'] = pd.cut(df['val'], bins=[0, 100, 500, 1000])
df['date'] = pd.to_datetime(df['date'])
df['year'] = df['date'].dt.year
```

6.2 Visualisation (Matplotlib/Seaborn)

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

# Histogramme
plt.hist(df['col'], bins=30)

# Box plot
df.boxplot(column='val', by='cat')

# Scatter
plt.scatter(df['x'], df['y'])

# Heatmap corrélation
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
```

---

7. EDA - MÉTHODOLOGIE

7.1 Framework

```
1. COMPRENDRE le contexte business
2. CHARGER et examiner la structure
3. PROFILER chaque variable (univarié)
4. EXPLORER les relations (bivarié)
5. IDENTIFIER problèmes de qualité
6. NETTOYER et transformer
7. DOCUMENTER les insights
```

7.2 Checklist Qualité Données

```
□ Valeurs manquantes (isnull)
□ Doublons (duplicated)
```

```
□ Types de données corrects
□ Valeurs aberrantes (outliers)
□ Cohérence (cross-validation)
□ Distributions attendues
□ Cardinalité des catégories
```

7.3 Traitement des Outliers

```python
# Méthode IQR
Q1, Q3 = df['col'].quantile([0.25, 0.75])
IQR = Q3 - Q1
lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR
outliers = df[(df['col'] < lower) | (df['col'] > upper)]

# Méthode Z-score
from scipy import stats
z = np.abs(stats.zscore(df['col']))
outliers = df[z > 3]
```

---

8. SEGMENTATION CLIENT

8.1 RFM

```
R (Recency): Jours depuis dernière activité
F (Frequency): Nombre de transactions
M (Monetary): Montant total

Score 1-5 par quintile, inversé pour R
```

8.2 Segments Types

Segment	Profil	Action
Champions	RFM élevé	Fidéliser, récompenser
Fidèles	F+M élevé	Maintenir, cross-sell
Nouveaux	R élevé, F bas	Activer, onboarding
À risque	R bas, F élevé	Réactiver
Perdus	Tout bas	Win-back sélectif

---

9. FORMULES IMPORTANTES

Statistiques

```
Moyenne: x̄ = Σxᵢ / n
Variance: s² = Σ(xᵢ - x̄)² / (n-1)
IC 95%: x̄ ± 1.96 × (s/√n)
```

Finance

```
ROE = Résultat / Capitaux Propres
Expected Loss = PD × LGD × EAD
CAGR = (Vf/Vi)^(1/n) - 1
```

Variation

```
Var % = (Nouveau - Ancien) / Ancien × 100
YoY = (Année N - Année N-1) / Année N-1 × 100
```

---

10. CONSEILS POUR L'ENTRETIEN

Questions Techniques

1. Toujours donner un exemple concret (bancaire si possible)
2. Expliquer le "pourquoi" pas juste le "quoi"
3. Mentionner les limites et alternatives

Études de Cas

1. CLARIFIER le problème et les données
2. STRUCTURER l'approche avant de commencer
3. EXPLIQUER les choix méthodologiques
4. INTERPRÉTER dans le contexte business
5. PROPOSER des next steps

Communication

- Vulgariser pour non-techniques
- Utiliser des analogies
- Admettre ce qu'on ne sait pas

---

11. TERMES À CONNAÎTRE

Terme	Définition Rapide
ACID	Atomicity, Consistency, Isolation, Durability
ETL	Extract, Transform, Load
OLAP	Online Analytical Processing (analyse)
OLTP	Online Transaction Processing (opérationnel)
Data Warehouse	Entrepôt de données historiques
Data Lake	Stockage données brutes
Feature Engineering	Création de variables
Overfitting	Modèle trop ajusté aux données d'entraînement
Cross-validation	Validation croisée
p-value	Probabilité d'observer le résultat si H_0 vraie

---

10. RÉGRESSION LINÉAIRE - ESSENTIEL

10.1 Modèle et Interprétation

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \varepsilon$

β_1 = Changement de Y pour 1 unité de X_1 (toutes choses égales)
R^2 = Variance expliquée par le modèle
R^2 ajusté = R^2 pénalisant les variables inutiles

10.2 Hypothèses LINE

L - Linéarité: Relation linéaire entre X et Y
I - Indépendance: Observations indépendantes
N - Normalité: Résidus normaux
E - Égalité variances: Homoscédasticité

10.3 Diagnostics Clés

VIF > 10: Multicolinéarité → supprimer variable
Durbin-Watson:
  - DW ≈ 2: OK
  - DW < 1.5: Autocorrélation positive
  - DW > 2.5: Autocorrélation négative

Tests:
- Shapiro-Wilk: Normalité résidus
- Breusch-Pagan: Hétéroscédasticité

10.4 Interprétation Coefficients

```
# Si p-value < 0.05 → Coefficient significatif
# Si IC ne contient pas 0 → Significatif

# Régression logistique: Odds Ratio = exp(β)
# OR = 1.5 → 50% plus de chances
```

---

11. SÉRIES TEMPORELLES - ESSENTIEL

11.1 Composantes (TSCI)

T - Tendance: Direction long terme
S - Saisonnalité: Pattern répétitif (période fixe)
C - Cycle: Fluctuations économiques long terme
I - Irrégulier: Bruit aléatoire

11.2 Stationnarité

Série stationnaire: Moyenne et variance constantes

Test ADF:
- p < 0.05 → Stationnaire ✓

```
- p > 0.05 → Non stationnaire → Différencier

Transformation: d = diff(Y) pour retirer tendance
```

11.3 Modèles

```
ARIMA(p, d, q):
- p = Ordre autorégressif (lags Y)
- d = Ordre différenciation
- q = Ordre moyenne mobile (lags erreur)

SARIMA: ARIMA + saisonnalité
Holt-Winters: Niveau + Tendance + Saisonnalité
Prophet: Jours fériés + patterns complexes
```

11.4 Métriques

```
MAPE = Erreur % moyenne
  < 10%: Excellent
  10-20%: Bon
  > 20%: À améliorer

AIC/BIC: Plus bas = Meilleur modèle
```

---

12. TESTS NON-PARAMÉTRIQUES - ESSENTIEL

12.1 Quand Utiliser?

```
✓ Distribution non normale
✓ Petit échantillon (n < 30)
✓ Données ordinales
✓ Outliers importants
```

12.2 Correspondance Tests

```
Paramétrique → Non-Paramétrique

t-test indépendant → Mann-Whitney U
t-test apparié → Wilcoxon signé
ANOVA → Kruskal-Wallis
Pearson → Spearman
```

12.3 Corrélations

```
Pearson: Relation linéaire, données normales
Spearman: Relation monotone, basé sur rangs
Kendall: Alternative robuste petits échantillons

Interprétation identique: |r| > 0.7 = forte
```

---

13. A/B TESTING - ESSENTIEL

13.1 Terminologie

```
Baseline: Performance actuelle (groupe contrôle)
MDE: Effet Minimal Détectable (plus petite amélioration intéressante)
Lift: (Traitement - Contrôle) / Contrôle × 100%
Puissance: P(détecter un vrai effet) = 80% standard
α: P(faux positif) = 5% standard
```

13.2 Étapes d'un A/B Test

```
1. HYPOTHÈSE: "Le nouvel email augmentera le taux de conversion de 2%"
2. DESIGN: Calculer taille échantillon, définir métriques, durée minimale
3. RANDOMISATION: Assigner aléatoirement clients aux groupes A/B
4. EXÉCUTION: Collecter données sans peeking (min 7 jours)
5. ANALYSE: Test statistique, IC, décision
```

13.3 Calcul Taille Échantillon

```
n ≈ 16 × p(1-p) / δ²

Où:
p = baseline (ex: 5%)
δ = MDE (ex: 1%)

Plus le MDE est petit, plus n doit être grand
Doubler MDE → n divisé par 4
```

13.4 Analyse et Décision

```
p-value < 0.05 + Lift > 0 → Déployer B
p-value < 0.05 + Lift < 0 → Garder A (B est pire!)
p-value ≥ 0.05 → Pas de conclusion, continuer ou abandonner

IC 95% sur la différence:
- Ne contient pas 0 → Significatif
- Entièrement positif → B meilleur
- Entièrement négatif → A meilleur
```

13.5 Pièges Courants

```
 Peeking: Regarder avant la fin → Inflation faux positifs
 Durée insuffisante: Min 7 jours (cycle complet)
 Multiple testing: Corriger si plusieurs variantes/métriques
 Effet nouveauté: Résultats initiaux peuvent être biaisés
 Contamination: Clients A interagissent avec clients B
```

---

14. ÉTHIQUE ET GOUVERNANCE - ESSENTIEL

14.1 Principes Éthiques (TERB)

```
T - Transparence: Expliquer comment les décisions sont prises
```

E - Équité: Traitement égal indépendamment du genre, âge, origine
R - Responsabilité: Assumer les conséquences des décisions
B - Bénéfice: L'analyse doit créer de la valeur pour le client aussi

14.2 Biais Algorithmiques

Disparate Impact (DI) = Taux_minorité / Taux_majorité

DI < 0.8 (80%) → DISCRIMINATION potentielle

Exemple:
- Taux approbation hommes: 60%
- Taux approbation femmes: 45%
- DI = 45/60 = 0.75 < 0.8 → ALERTE

14.3 Variables Proxy Dangereuses

Variables qui peuvent servir de proxy discriminatoire:
- Code postal → Corrélé à l'origine ethnique/revenu
- Prénom → Corrélé au genre/origine
- Type de téléphone → Corrélé au revenu

Solution: Vérifier corrélation avec variables sensibles

14.4 Explicabilité (XAI)

SHAP: Explication locale (par client) + globale (modèle)
Feature Importance: Impact relatif de chaque variable

Droit à l'explication:
- Obligatoire pour refus de crédit
- Le client doit comprendre pourquoi

14.5 Droits des Personnes (AREPO)

A - Accès: Voir ses données
R - Rectification: Corriger les erreurs
E - Effacement: Droit à l'oubli
P - Portabilité: Récupérer ses données
O - Opposition: Refuser certains traitements

14.6 Gouvernance des Données

Classification: Public < Interne < Confidentiel < Strictement confidentiel
Moindre privilège: Accès minimal nécessaire au rôle
Audit trail: Tracer tous les accès aux données
Rétention: 10 ans pour transactions (obligation légale AML)

CHECKLIST FINALE

□ Types de Variables: Nominale/Ordinale/Discrète/Continue

□ Statistiques descriptives et tests
□ SQL: CTEs, Window Functions, optimisation
□ ML: Classification, Régression, Clustering, métriques (AUC, Gini)
□ Types de Modèles: Descriptif/Prédictif, Supervisé/Non supervisé
□ KPIs bancaires (rentabilité, risque, liquidité)
□ Python/Pandas: manipulation, visualisation
□ EDA: méthodologie, qualité données
□ Segmentation RFM
□ RÉGRESSION: LINE, R^2, VIF, Durbin-Watson
□ SÉRIES TEMPORELLES: ARIMA, stationnarité, MAPE
□ TESTS NON-PARAM: Mann-Whitney, Kruskal-Wallis, Spearman
□ A/B TESTING: MDE, puissance 80%, randomisation, peeking
□ ÉTHIQUE: Disparate Impact ≥ 0.8, SHAP, droits AREPO
□ GOUVERNANCE: Classification données, audit trail, rétention
□ Interprétation dans contexte business

Bonne chance pour votre entretien!