TD 3 : Le salaire moyen

Nettoyer avant d'interpréter

Anderson VIL | L3 Économie | 90 minutes
Analyse des données | 2026–2027

Ouverture et rappels

TD 3 : Le salaire moyen

Ligne, colonne, poste et personne

Ligne	Un poste occupé pendant l'année.
Colonne	Une variable, par exemple `SALAIRE_NET`.
Personne	Identifiée par `IDENT`, avec zéros initiaux.
Poste	Repéré par `NOPOSTE` pour une même personne.
Clé	Le couple `IDENT + NOPOSTE`.

8 012 lignes ne signifient pas 8 012 personnes distinctes. Après retrait des 12 doublons techniques, le fichier contient 8 000 postes pour 7 380 personnes.

Mini-exemple 1 : reconnaître la clé

IDENT	NOPOSTE	Secteur	Salaire	Lecture
00003257	1	AZ	999 999,00	premier poste
00003257	2	OQ	2 967,79	second poste
00011540	1	GZ	2 100,00	autre personne

Pause recherche : Combien de lignes, de personnes et de postes ? Quelle colonne ne suffit pas comme clé ?

Correction du mini-exemple 1

3	2	3
lignes	personnes	postes

`IDENT` répété ne prouve pas un doublon. Deux postes d'une même personne sont deux observations légitimes. Un doublon technique répète la clé `IDENT + NOPOSTE`, et ici toute la ligne.

Mini-exemple 2 : texte ou nombre ?

`"2 450,50"`
chaîne de caractères

`2450.50`
nombre décimal

Pause recherche : Quelles transformations rendent la moyenne calculable sans perdre les décimales ?

Correction du mini-exemple 2

```
s = pd.Series(["2 450,50", "1 800,00"])
n = pd.to_numeric(
    s.str.replace(" ", "", regex=False)
     .str.replace(",", ".", regex=False)
)
print(n.tolist(), n.mean())
```

SORTIE RÉELLE résultat

```
[2450.5, 1800.0] 2125.25
```

Les identifiants restent du texte. Convertir `IDENT` en entier supprimerait les zéros initiaux.

Mini-exemple 3 : la moyenne et son dénominateur

Salaires : 2 000, 3 000, `NaN`, 0

Pause recherche : Quelle moyenne calcule pandas ? Que mesure-t-elle si 0 signifie « non-salarié » ?

Indice : pandas ignore `NaN`, mais conserve zéro. Le dénominateur vaut donc 3, pas 4 ni 2.

Correction du mini-exemple 3

$$\bar{x}_{\text{avec zéro}} = \frac{2000 + 3000 + 0}{3} = 1\,666{,}67$$

$$\bar{x}_{\text{salariés}} = \frac{2000 + 3000}{2} = 2\,500$$

Les deux calculs sont arithmétiquement corrects. Le premier mélange un salaire avec une situation hors champ. Le second répond à la question « salaire moyen des personnes qui perçoivent un salaire ».

Variation : +833,33 euros ; +50 % relativement ; facteur multiplicatif 1,5. Ces trois écritures ne sont pas synonymes.

Quatre absences d'information différentes

`NaN`	valeur manquante pour pandas	peut représenter une non-réponse ou un hors champ ; le dictionnaire tranche
Code 999 999	nombre valide pour Python	non-réponse documentée de `SALAIRE_NET`
Zéro	nombre observé	ici, sans objet pour les non-salariés
Case vide	absence de texte	pour `PRIME`, signifie « aucune prime perçue », pas « non-renseigné »

Une même apparence peut porter des sens différents. La règle vient de la documentation, variable par variable.

Quantile, plafond et censure

Quantile 99 %
Valeur au-dessous de laquelle se trouvent au moins 99 % des salaires observés.

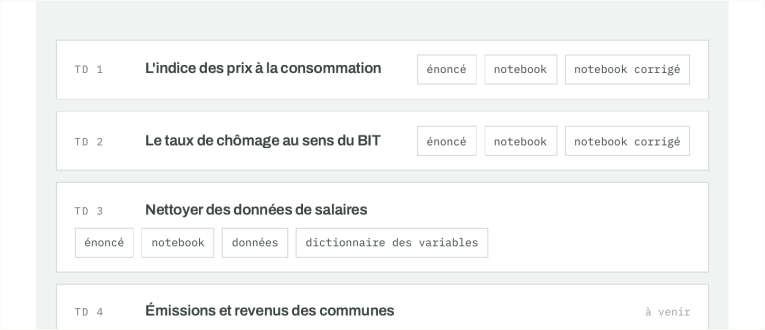
Plafond à 10 000 euros
Toute valeur supérieure est enregistrée à 10 000. La queue supérieure précise disparaît.

Répéter un quantile ne prouve pas une censure. Un amas de valeurs peut aussi produire cette répétition. Ici, le dictionnaire documente explicitement le plafond.

Ressources et prise en main

TD 3 : Le salaire moyen

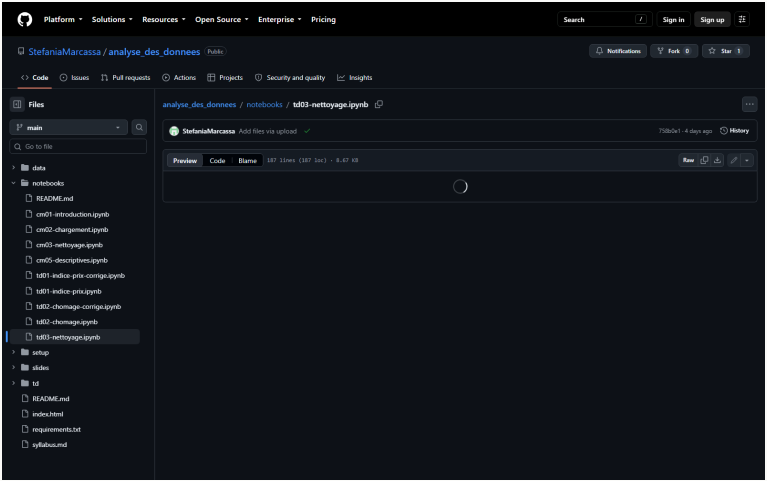
La page du cours : le point de départ



TD 3
Énoncé
Notebook étudiant
Données
Dictionnaire

Source : Capture authentique : site du cours, consultation 23/09/2026.

GitHub : localiser le notebook exact



Source : Capture authentique : dépôt GitHub, consultation 23/09/2026.

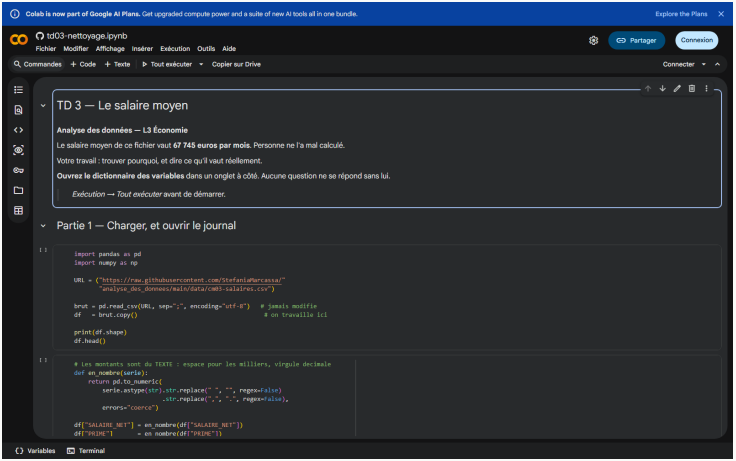
Chemin

`notebooks/`

`td03-nettoyage.ipynb`

Preview ne signifie pas exécution. GitHub affiche le fichier ; Colab exécute ses cellules.

Colab : ouvrir, copier, puis tester



Ouverture directe
Le lien GitHub peut s'ouvrir dans Colab.

Copie personnelle
Cliquer sur **Copier sur Drive** après connexion.

Source : Capture authentique : Google Colab ouvrant le notebook TD3 public, 23/09/2026.

Deux chemins vers Colab

Depuis l'ordinateur
1. Télécharger le fichier `.ipynb`.
2. Ouvrir Colab.
3. **Fichier** puis **Importer le notebook**.
4. Choisir le fichier.

Depuis GitHub
1. Ouvrir le notebook du dépôt.
2. Choisir **Open in Colab** ou utiliser l'URL Colab GitHub.
3. Enregistrer une copie dans Drive.

Exécution puis Tout exécuter. Tester chaque cellule pendant le travail. Utiliser **Exécution** → **Tout exécuter** seulement comme contrôle final depuis un état propre.

Lire le CSV : la méthode du dictionnaire

```
brut = pd.read_csv(
    "cm03-salaires.csv",
    sep=";", encoding="latin-1",
    decimal=",", thousands=" ",
    dtype={"IDENT": str,
           "SECTEUR": str, "REGION": str}
)
df = brut.copy()
```

Résultat
Les montants deviennent numériques dès la lecture. Les identifiants gardent leurs zéros.

État propre
`brut` reste inchangé ; les décisions s'appliquent à `df`.

Source : Dictionnaire `cm03-salaires.csv`, p. 1.

Lire le CSV : la méthode du notebook

```
brut = pd.read_csv("cm03-salaires.csv",
                   sep=";", encoding="utf-8")

def en_nombre(serie):
    texte = serie.astype(str)
    texte = texte.str.replace(" ", "", regex=False)
    texte = texte.str.replace(",", ".", regex=False)
    return pd.to_numeric(texte, errors="coerce")

df = brut.copy()
df["SALAIRE_NET"] = en_nombre(df["SALAIRE_NET"])
df["PRIME"] = en_nombre(df["PRIME"])
```

Contrôle du fichier fourni : aucun octet non ASCII. UTF-8 et latin-1 donnent ici les mêmes montants. Pour un autre CSV, suivre le dictionnaire. Lire les identifiants comme du texte.

Inspecter avant de calculer

```
print(df.shape)
display(df.head())
print(df.dtypes)
```

`shape` : lignes, colonnes
`head()` : aperçu
`dtypes` : types stockés

SORTIE RÉELLE sorties vérifiées

```
(8012, 15)
```

IDENT	NOPOSTE	SALAIRE_NET	REGION
00000262	1	999999.00	76
00001293	1	999999.00	27
00001555	1	4609.14	28
00003257	1	999999.00	44
00003257	2	2967.79	27

`IDENT`, `SECTEUR` et `REGION` doivent rester du texte.

Ouvrir le journal avant le nettoyage

```
journal = []

def noter(etape, decision, justification, data):
    journal.append({
        "etape": etape,
        "decision": decision,
        "lignes": len(data),
        "salaires_valides": data["SALAIRE_NET"].notna().sum(),
        "salaire_moyen": round(data["SALAIRE_NET"].mean(), 2),
        "justification": justification
    })
    return pd.DataFrame(journal)

noter("Chargement", "Aucune", "Fichier brut", df)
```

Le journal se tient pendant le travail. Ajouter le dénominateur `salaires_valides` évite de comparer deux moyennes sans savoir sur combien de salaires elles portent.

Questions 1 et 2 : ordre de grandeur

TD 3 : Le salaire moyen

Q1. Taille et moyenne de départ

Pause recherche : Combien de lignes le fichier contient-il ? Quel est le salaire moyen avant toute correction ?

À annoncer avant le calcul
Une ligne représente un poste. La moyenne utilise toutes les 8 012 lignes, car aucun salaire n'est encore manquant pour pandas.

Q1. Sortie et réponse

```
print(df.shape)
print(round(
    df["SALAIRE_NET"].mean(), 2
))
```

SORTIE RÉELLE sortie réelle

```
(8012, 15)
67745.36
```

Réponse Q1. Le fichier compte 8 012 postes et 15 variables. La moyenne brute vaut 67 745,36 euros mensuels, avec 8 012 valeurs prises au dénominateur.

Q2. Une référence externe bien définie



LES SALAIRES DANS LE SECTEUR PRIVÉ EN 2024

Insee
Mesurer pour comprendre

Avec le reflux de l'inflation, le pouvoir d'achat des salaires progresse à nouveau

Insee Première • n° 2079 • Octobre 2025

En 2024, un salarié du secteur privé gagne en moyenne 2 733 euros net par mois en équivalent temps plein (EQTP). Avec la baisse de l'inflation en 2024 (+2,0 %, après +4,9 % en 2023), le salaire net moyen en euros constants se redresse : +0,8 %, après -1,0 % en 2023.

La moitié des salariés du secteur privé perçoit moins de 2 190 euros net par mois en EQTP. Un salarié sur dix gagne moins de 1 492 euros, tandis qu'un sur dix perçoit plus de 4 334 euros. En 2024, les salaires progressent de manière homogène sur l'ensemble de l'échelle salariale, à la différence des deux années précédentes au cours desquelles les revalorisations du Smic ont préservé le pouvoir d'achat du bas de la distribution quand l'inflation érodait celui des salaires plus élevés. Dans ce contexte, les disparités salariales se stabilisent en 2024, après trois années consécutives de recul.

RÉFÉRENCE

Année : 2024
Secteur : privé
Unité : euros nets par mois
Standardisation : EQTP
Moyenne : **2 733 euros**

Source : Insee Première n° 2079, « Les salaires dans le secteur privé en 2024 », paru le 23/10/2025.

Q2. Comparaison d'ordre de grandeur

$$\frac{67\,745{,}36}{2\,733} = \mathbf{24{,}79}$$

Réponse Q2. La moyenne brute du fichier vaut environ 24,8 fois la moyenne Insee 2024 retenue. Cet écart signale immédiatement un problème de données ou de champ.

Questions 3 à 7 : le code 999 999

TD 3 : Le salaire moyen

Q3 à Q5. Ce que pandas voit, puis ce qu'il manque

Pause recherche : Combien de manquants `isna()` signale-t-il ? Quelle valeur des cinq plus fréquentes ne peut pas être un salaire ? Que dit le dictionnaire ?

Ne pas corriger immédiatement. Afficher d'abord les deux diagnostics et laisser apparaître la contradiction.

Q3 et Q4. Deux diagnostics complémentaires

```
print(df["SALAIRE_NET"].isna().sum())
print(df["SALAIRE_NET"]
      .value_counts().head(5))
```

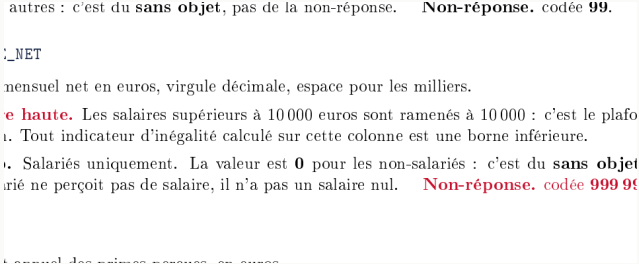
SORTIE RÉELLE sortie réelle

0

0.00	675
999999.00	525
10000.00	72
1859.27	3
1822.74	2

Q3 : zéro manquant pour pandas. **Q4 :** 525 valeurs égales à 999 999 euros.

Q5. La preuve vient du dictionnaire



autres : c'est du **sans objet**, pas de la non-réponse. **Non-réponse.** codée **99**.

:_NET

nensuel net en euros, virgule décimale, espace pour les milliers.

e haute. Les salaires supérieurs à 10 000 euros sont ramenés à 10 000 : c'est le plafo
ı. Tout indicateur d'inégalité calculé sur cette colonne est une borne inférieure.

. Salariés uniquement. La valeur est **0** pour les non-salariés : c'est du **sans objet**
arié ne perçoit pas de salaire, il n'a pas un salaire nul. **Non-réponse.** codée **999 99**

t annuel des primes perçues, en euros

Source : Dictionnaire `cm03-salaires.csv`, p. 3.

`SALAIRE_NET`
Non-réponse codée **999 999**.

Python ne peut pas deviner ce sens. La valeur est numérique, donc elle entre sans erreur dans la moyenne.

Q6. Une fonction dont le code est un argument

```
def retirer_non_reponse(serie, code):
    """Remplace le code documente par NaN."""
    return serie.replace(code, np.nan)

for colonne in ["SALAIRE_NET", "PRIME"]:
    df[colonne] = retirer_non_reponse(
        df[colonne], 999999.0
    )
```

Pourquoi une fonction ?
La décision devient visible, testable et réutilisable.

Pourquoi un argument ?
Chaque variable peut employer un autre code de non-réponse.

Q6. Mesurer l'effet et noter la décision

```
avant = 67745.36160883676
apres = df["SALAIRE_NET"].mean()

print(round(apres, 2))
print(round(avant - apres, 2))
print(round(avant / apres, 2))

noter("Code 999999", "Conversion en NaN",
      "Dictionnaire : non-reponse", df)
```

SORTIE RÉELLE sortie réelle

Moyenne : **2 374,30 euros** sur **7 487 salaires observés**.
Écart : **65 371,06 euros**. Facteur : **28,53**.

Q7. Correct pour Python, incorrect pour l'économie

Avant
`mean()` reçoit 8 012 nombres. Il calcule exactement leur moyenne.

Après
`mean()` ignore 525 `NaN`. Il calcule exactement la moyenne des 7 487 valeurs restantes.

Réponse Q7. Aucun message d'erreur ne signale une convention statistique mal interprétée. Les contrôles de plausibilité, le dictionnaire et le journal détectent ce type d'erreur silencieuse.

Questions 8 et 9 : le piège du code 9

TD 3 : Le salaire moyen

Q8. Deux codes qui se ressemblent

Pause recherche : Afficher la distribution de `MOTIF_ABS`. Lequel des codes 9 et 99 est une non-réponse ? Que signifie l'autre ?

Méthode
Utiliser `value_counts(dropna=False)` pour conserver les cases vides, puis `sort_index()` pour lire les codes dans l'ordre.

Q8. Distribution complète de MOTIF_ABS

```
df["MOTIF_ABS"].value_counts(
    dropna=False
).sort_index()
```

SORTIE RÉELLE sortie réelle

1	458
2	175
3	34
4	54
5	45
9	61
99	32
NaN	7 153

Le tableau seul ne donne pas le sens des codes. Il montre seulement leurs effectifs.

Q8. Le dictionnaire tranche

heures ou plus ». Toute moyenne calculée sur cette colonne est une borne inférieure de la durée réelle.

Champ. Toutes les lignes. **Non-réponse.** codée **99**.

`MOTIF_ABS`

Motif de l'absence au cours de la semaine de référence. Numérique.

1 Congés
2 Maladie
3 Congé de maternité ou de paternité
4 Formation
5 Raisons familiales
9 **Chômage partiel**
99 Non-réponse

Attention. Sur cette variable, 9 est une situation réelle et non une non-réponse. La non-réponse est codée 99. Un `replace(9, NaN)` appliqué par habitude supprime une catégorie entière.

Champ. Personnes absentes de leur poste au cours de la semaine de référence. La case est **vide** pour les autres : c'est du **sans objet**, pas de la non-réponse. **Non-réponse.** codée **99**.

`SALAIRE_NET`

Source : Dictionnaire `cm03-salaires.csv`, p. 3.

`MOTIF_ABS=9`
Chômage partiel : modalité réelle.

`MOTIF_ABS=99`
Non-réponse : 32 observations.

`NaN`
Sans objet pour les personnes non absentes.

Q9. Ce qu'un remplacement automatique détruirait

Réponse Q9. `replace(9, np.nan)` appliqué à `MOTIF_ABS` transformerait **61 situations de chômage partiel** en valeurs manquantes.

Fonction sûre
`retirer_non_reponse(serie, code)` reçoit le code documenté.

Usage sûr
La fonction ne protège que si l'utilisateur lui passe **99** pour cette variable. Le dictionnaire reste indispensable.

Questions 10 et 11 : zéro et champ

TD 3 : Le salaire moyen

Q10. Qui porte un salaire égal à zéro ?

Pause recherche : Compter les zéros, puis croiser `STATUT` et `CONTRAT` uniquement pour ces lignes. Le résultat est-il ambigu ?

Ne pas convertir les zéros en manquants avant le croisement. Leur structure permet de comprendre leur sens.

Q10. Filtre et tableau croisé

```
z = df["SALAIRE_NET"].eq(0)
print(z.sum())

pd.crosstab(
    df.loc[z, "STATUT"],
    df.loc[z, "CONTRAT"]
)
```

SORTIE RÉELLE sortie réelle

675 salaires nuls

	CONTRAT=8
STATUT=3	675

Une seule case est remplie. Tous les zéros appartiennent au même couple de modalités.

Q10. Preuve documentaire du hors champ

RE_NET

e mensuel net en euros, virgule décimale, espace pour les milliers.

ure haute. Les salaires supérieurs à 10 000 euros sont ramenés à 10 000 : c'est le plafond d
on. Tout indicateur d'inégalité calculé sur cette colonne est une borne inférieure.

ıp. Salariés uniquement. La valeur est **0** pour les non-salariés : c'est du **sans objet**. U
alarié ne perçoit pas de salaire, il n'a pas un salaire nul. **Non-réponse.** codée **999 999**.

ınt annuel des primes perçues, en euros.

ıtion. La case **vide** signifie « **aucune prime perçue** », et non « non renseigné ». L
ction change la moyenne d'un facteur deux selon qu'on la traite comme un zéro ou comme u
uant. Dites laquelle des deux vous avez retenue.

`STATUT=3`
Non-salarié.

`CONTRAT=8`
Sans objet : non-salarié.

`SALAIRE_NET=0`
Sans objet pour les non-salariés, pas un revenu d'activité nul.

Source : Dictionnaire `cm03-salaires.csv`, pp. 2–3.

Q11. Deux moyennes, deux dénominateurs

```
m_avec = df["SALAIRE_NET"].mean()
m_sans = df.loc[
    df["SALAIRE_NET"] > 0,
    "SALAIRE_NET"
].mean()
print("Avec les zeros :", round(m_avec, 2)),
print("Sans les zeros :", round(m_sans, 2)),
print("Ecart :", round(m_avec - m_sans, 2), "euros"
     )
```

SORTIE RÉELLE sortie réelle

Avec zéros : **2 374,30**
Sans zéros : **2 609,57**
Écart : **235,27 euros**
Hausse relative : **9,91 %**

Q11. La décision porte sur le champ

Réponse retenue. Pour calculer un salaire moyen, exclure les 675 lignes hors champ et conserver les 525 non-réponses comme `NaN`. La moyenne vaut 2 609,57 euros sur **6 812 salaires observés**.

Formulation à éviter : la moyenne avec les zéros n'est pas un « revenu d'activité moyen ». Le fichier ne mesure pas le revenu d'activité des indépendants ; il code seulement leur salaire salarié comme sans objet.

Autre question, autre variable. Étudier les revenus de tous les actifs demanderait une variable de revenu couvrant aussi les non-salariés.

Q11. Appliquer la décision et journaliser

```
df = df.loc[
    df["SALAIRE_NET"].isna()
    | (df["SALAIRE_NET"] > 0)
].copy()

noter(
    "Salaires nuls",
    "Exclusion du champ salarial",
    "STATUT=3, CONTRAT=8",
    df
)
```

SORTIE RÉELLE état après décision

7 337 lignes ; 6 812 salaires observés ; 525 non-réponses ; moyenne 2 609,57 euros.

`copy()` évite une vue ambiguë après filtrage et préserve un objet de travail indépendant.

Questions 12 et 13 : censure haute

TD 3 : Le salaire moyen

Q12. Lire les quantiles avant le dictionnaire

Pause recherche : Calculer les quantiles à 90, 95, 99 et 99,9 %. Que remarquez-vous sur les deux derniers ?

Q12. Quantiles et effectif au maximum

```
q = df["SALAIRE_NET"].quantile(
    [0.90, 0.95, 0.99, 0.999]
).round(2)
print(q)
print((df["SALAIRE_NET"]
       == df["SALAIRE_NET"].max()).sum())
```

SORTIE RÉELLE sortie réelle

90 %	4 251,87
95 %	5 077,18
99 %	10 000,00
99,9 %	10 000,00

Maximum : **72 lignes**

Q12. Le dictionnaire documente le plafond

est codée 99. Un `replace(9, NaN)` appliqué par habitude supprime une catégorie entière.

Champ. Personnes absentes de leur poste au cours de la semaine de référence. La case est **vide** pour les autres : c'est du **sans objet**, pas de la non-réponse. **Non-réponse.** codée **99**.

`SALAIRE_NET`

Salaire mensuel net en euros, virgule décimale, espace pour les milliers.

Censure haute. Les salaires supérieurs à 10 000 euros sont ramenés à 10 000 : c'est le plafond de diffusion. Tout indicateur d'inégalité calculé sur cette colonne est une borne inférieure.

Champ. Salariés uniquement. La valeur est **0** pour les non-salariés : c'est du **sans objet**. Un non-salarié ne perçoit pas de salaire, il n'a pas un salaire nul. **Non-réponse.** codée **999 999**.

`PRIME`

Montant annuel des primes perçues, en euros.

Attention. La case **vide** signifie « **aucune prime perçue** », et non « non renseigné ». La distinction change la moyenne d'un facteur deux selon qu'on la traite comme un zéro ou comme un manquant. Dites laquelle des deux vous avez retenue.

Champ. Salariés uniquement. **Non-réponse.** codée **999 999**.

Plafond de diffusion
Les salaires supérieurs à 10 000 euros sont ramenés à 10 000.

Information perdue
Pour une valeur enregistrée à 10 000, le salaire initial exact n'est pas observable.

Source : Dictionnaire `cm03-salaires.csv`, p. 3.

Q12. Interpréter les 72 valeurs au plafond

Résultat vérifié. 72 valeurs sont enregistrées à 10 000 euros, soit $72/6\,812 = \mathbf{1{,}057}\,\%$ des salaires observés dans le champ retenu.

Ce que le fichier ne permet pas d'affirmer : que 72 salaires étaient strictement supérieurs à 10 000 euros avant diffusion. Une valeur initiale exactement égale au plafond serait indiscernable après codage.

Conclusion Q12. Les quantiles 99 % et 99,9 % coïncident avec le plafond. Cette répétition est compatible avec une censure que le dictionnaire confirme.

Q13. Une note méthodologique rigoureuse

Les salaires mensuels nets supérieurs à 10 000 euros sont enregistrés à ce plafond. Le fichier contient 72 valeurs au plafond, soit 1,057 % des 6 812 salaires observés dans le champ salarial retenu. La moyenne et les indicateurs sensibles à la queue supérieure ne peuvent pas reconstituer précisément les hauts salaires.

Les trois éléments attendus
Plafond : 10 000 euros. Dénominateur : 6 812 salaires observés. Limite : perte de l'information au-dessus du plafond.

Ce que la censure affecte, ou pas forcément

Moyenne, variance, Gini	Peuvent être sous-estimés lorsque les valeurs de la queue supérieure sont ramenées au plafond. Ne sont plus identifiables précisément.
Quantiles au-dessus du seuil touché	
D9/D1	Peut rester inchangé si la censure commence au-dessus du neuvième décile. Ici, D9 vaut 4 251,87 euros, sous le plafond.
Médiane et quantiles centraux	Peuvent rester inchangés si le plafond ne modifie aucune observation proche de ces seuils.

Question 14 : journal et synthèse

TD 3 : Le salaire moyen

Q14. Le journal complet

Étape	Décision	Lignes	Salaires valides	Moyenne	Justification
Chargement	Aucune	8 012	8 012	67 745,36	Fichier converti en nombres
Code 999 999	Conversion en `NaN`	8 012	7 487	2 374,30	Non-réponse documentée
Salaires nuls	Exclusion du champ	7 337	6 812	2 609,57	`STATUT=3`, `CONTRAT=8`

Chaque ligne dit trois choses : combien de lignes restent, combien de salaires entrent dans la moyenne, et quelle question l'indicateur mesure désormais.

Q14. Variation totale et formulation corrigée

$$67\,745{,}36 \longrightarrow 2\,609{,}57$$

$-65\,135{,}80$ **euros**	$-96{,}15$ **%**	**25,96**
différence	variation relative	facteur brut/final

Réponse : aucune des deux décisions de nettoyage inscrites au journal ne laisse la moyenne inchangée. L'opération nécessaire mais numériquement neutre est `df = brut.copy()`, elle préserve le fichier brut et change la traçabilité, pas le nombre.

Trois décisions, trois effets distincts

Opération	Nombre de lignes	Moyenne	Définition
`copy()`	inchangé	inchangée	sépare source et objet de travail
999 999 vers `NaN`	inchangé	chute de 28,53 fois	retire les non-réponses du calcul
Exclusion des zéros hors champ	−675	augmente de 9,91 %	limite le calcul aux salariés

Une décision peut agir sur le nombre de lignes, le dénominateur, la moyenne ou le sens. Le journal doit permettre de distinguer ces effets.

Synthèse des décisions

1. Charger les montants comme nombres et préserver les identifiants en texte.
2. Conserver une copie brute avant toute transformation.
3. Convertir 999 999 en `NaN` pour `SALAIRE_NET` et `PRIME`.
4. Ne jamais appliquer un code de non-réponse sans consulter la variable.
5. Exclure les zéros uniquement du champ du salaire salarié, avec justification.
6. Documenter le plafond sans inventer la valeur des hauts salaires.

Résultat final du parcours principal : 2 609,57 euros sur 6 812 salaires observés, dans un fichier synthétique de 7 337 lignes après restriction au champ salarial.

Annexes : questions facultatives et audit

TD 3 : Le salaire moyen

Q15. Trois comptages de doublons

```
print(brut.duplicated().sum())
print(brut.duplicated(subset=["IDENT"]).sum())
print(brut.duplicated(
    subset=["IDENT", "NOPOSTE"]
).sum())
```

SORTIE RÉELLE sortie réelle

12 lignes identiques partout

632 occurrences répétées de IDENT

12 couples répétés IDENT + NOPOSTE

Q15. Doublon technique ou personne multi-postes

12 doublons techniques
Même clé et même contenu. Après retrait : 8 000 postes.

632 IDENT répétés
Occurrences au-delà de la première. Elles correspondent à des personnes avec plusieurs postes, sauf les répétitions techniques.

Réponse Q15. La clé est `IDENT + NOPOSTE`. Si l'analyse porte sur les personnes, il faut agréger les postes par `IDENT`, pas supprimer arbitrairement des lignes.

Q15. Le dictionnaire décrit l'unité et la clé

Lignes	8 012 (dont 12 doublons techniques)
Colonnes	15

Chargement correct :

```
brut = pd.read_csv("cm03-salaires.csv", encoding="latin-1", sep=";",
    decimal=",", thousands=" ",
    dtype={"IDENT": str, "SECTEUR": str, "REGION": str})
```

Unité statistique

Une ligne est un poste occupé dans l'année, pas une personne. Une personne ayant occupé deux emplois apparaît deux fois, avec le même `IDENT` et deux valeurs de `NOPOSTE`.

La clé du fichier est donc le couple `IDENT` + `NOPOSTE`. Le fichier contient 7 380 personnes distinctes pour 8 000 postes, une fois les doublons techniques retirés.

Les variables

`IDENT`

Identifiant de la personne. Texte de 8 caractères, zéros initiaux compris. À charger en `str` : lu comme un entier, il perd ses zéros et ne s'apparie plus à aucun autre fichier.

Champ. Toutes les lignes. **Non-réponse.** Aucune.

`NOPOSTE`

Numéro du poste dans l'année pour cette personne. Numérique, 1 ou 2.

Source : Dictionnaire `cm03-salaires.csv`, p. 1.

Après dédoublonnage
8 000 postes
7 380 personnes

Zéros initiaux
`IDENT` doit être lu comme texte.

Q16. Imputer la moyenne

```
impute = df["SALAIRE_NET"].fillna(
    df["SALAIRE_NET"].mean()
)

print(round(df["SALAIRE_NET"].mean(), 2),
      round(impute.mean(), 2))
print(round(df["SALAIRE_NET"].std(), 2),
      round(impute.std(), 2))
```

SORTIE RÉELLE sortie réelle

Moyenne : 2 609,57 avant et après.
Écart-type : 1 464,77 avant ; 1 411,38 après, soit $-3,64$ %.

Q16. Ce que l'imputation change vraiment

Dénominateurs. 525 salaires sont manquants après restriction du champ. Ils représentent 7,156 % des 7 337 lignes ; 6 812 salaires sont observés.

La baisse de l'écart-type n'est pas égale à la part imputée. Elle dépend de la définition de l'écart-type et des dénominateurs. Ici, elle vaut 3,64 %, pas 7,16 %.

Inférence. Une imputation par la moyenne traite les valeurs ajoutées comme si elles étaient connues sans incertitude. L'effet sur une erreur-type dépend ensuite de l'estimateur et du mécanisme de données manquantes, il n'est pas automatiquement identique dans tous les modèles.

Q17. Test de robustesse sur le champ

Décision	Moyenne	Interprétation
Conserver les zéros	2 374,30 euros	moyenne de la variable codée sur 7 487 valeurs, mélangeant salaires et sans objet
Exclure les zéros hors champ	2 609,57 euros	salaire moyen sur 6 812 salaires observés

Réponse Q17. Le niveau varie de 9,91 % selon la décision. La conclusion principale survit. En revanche, l'intitulé économique change avec le champ.

Annexe : reformulations apportées, partie 1

Avant	Dans ce document
« 26 fois plus élevé » à partir d'environ 2 600 euros	24,79 fois avec la référence Insee 2024 de 2 733 euros nets mensuels en EQTP ; comparaison de cohérence seulement.
Moyenne avec zéros appelée « revenu d'activité moyen »	Moyenne de la variable codée, non interprétable comme revenu des indépendants ; le CSV ne mesure pas leur revenu d'activité.
Répétition des quantiles présentée comme preuve suffisante	Signal empirique compatible avec un plafond ; le dictionnaire apporte la preuve documentaire.
72 « salaires strictement supérieurs »	72 valeurs observées au plafond ; les valeurs initialement égales à 10 000 sont indiscernables des valeurs rabattues.

Annexe : reformulations apportées, partie 2

Avant	Dans ce document
« Tous les indicateurs d'inégalité sont des bornes inférieures »	La censure affecte les statistiques sensibles à la queue ; D9/D1 peut rester inchangé si D9 reste sous le plafond.
« Le mécanisme de baisse de l'écart-type est proportionnel à la part imputée »	Résultat chiffré : 7,156 % imputés, écart-type en baisse de 3,64 % ; relation non linéaire.
« Les erreurs-types seraient trop petites »	Risque d'incertitude sous-estimée, avec effet dépendant de l'estimateur, du modèle et du mécanisme de données manquantes.
Q14 : « étape nécessaire sans changer le résultat »	Aucune décision du journal n'est neutre sur la moyenne. `copy()` est nécessaire et numériquement neutre ; la question préparée était ambiguë.

Annexe : sources et accès vérifiés

- Page du cours : https://stefaniamarcassa.github.io/analyse_des_donnees/
- Énoncé : https://stefaniamarcassa.github.io/analyse_des_donnees/td/td03-enonce.pdf
- Notebook étudiant : https://github.com/StefaniaMarcassa/analyse_des_donnees/blob/main/notebooks/td03-nettoyage.ipynb
- CSV : https://github.com/StefaniaMarcassa/analyse_des_donnees/blob/main/data/cm03-salaires.csv
- Dictionnaire : https://stefaniamarcassa.github.io/analyse_des_donnees/data/cm03-dictionnaire.pdf
- Dépôt : https://github.com/StefaniaMarcassa/analyse_des_donnees
- Colab direct : https://colab.research.google.com/github/StefaniaMarcassa/analyse_des_donnees/blob/main/notebooks/td03-nettoyage.ipynb
- Insee : https://www.insee.fr/fr/statistiques/8657156

Contrôle du 23/09/2026. Énoncé, notebook étudiant, CSV et dictionnaire accessibles. Les fichiers locaux permettent un parcours autonome.