

exercice_fr

July 24, 2026

1 Le pouvoir d'achat des ouvriers anglais du XVIe au XIXe siècle

Ce document vise à reproduire et améliorer le graphique de William Playfair sur le prix du blé et le salaire hebdomadaire des ouvriers entre 1565 et 1821.

1.1 Environnement

Ce notebook a été exécuté avec Python 3 et les libraries et modules utilisés sont listés ci-après avec leurs versions associées.

```
[1]: %matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
from pathlib import Path

def print_imported_modules():
    import sys
    for name, val in sorted(sys.modules.items()):
        if(hasattr(val, '__version__')):
            print(val.__name__, val.__version__)

print_imported_modules()
```

```
IPython 7.12.0
IPython.core.release 7.12.0
PIL 7.0.0
PIL.Image 7.0.0
PIL._version 7.0.0
_csv 1.0
_ctypes 1.1.0
_curses b'2.2'
decimal 1.70
argparse 1.1
backcall 0.1.0
cffi 1.13.2
csv 1.0
ctypes 1.1.0
```

cycler 0.10.0
dateutil 2.8.1
decimal 1.70
decorator 4.4.1
distutils 3.6.4
ipaddress 1.0
ipykernel 5.1.4
ipykernel._version 5.1.4
ipython_genutils 0.2.0
ipython_genutils._version 0.2.0
jedi 0.16.0
json 2.0.9
jupyter_client 6.0.0
jupyter_client._version 6.0.0
jupyter_core 4.6.3
jupyter_core.version 4.6.3
kiwisolver 1.1.0
logging 0.5.1.2
matplotlib 2.2.3
matplotlib.backends.backend_agg 2.2.3
numpy 1.15.2
numpy.core 1.15.2
numpy.core.multiarray 3.1
numpy.lib 1.15.2
numpy.linalg._umath_linalg b'0.1.5'
optparse 1.5.3
pandas 0.22.0
_libjson 1.33
parso 0.6.0
pexpect 4.8.0
pickleshare 0.7.5
platform 1.0.8
prompt_toolkit 3.0.3
ptyprocess 0.6.0
pygments 2.5.2
pyparsing 2.4.6
pytz 2019.3
re 2.2.1
six 1.14.0
traitlets 4.3.3
traitlets._version 4.3.3
urllib.request 3.6
zlib 1.0
zmq 17.1.2
zmq.sugar 17.1.2
zmq.sugar.version 17.1.2

1.2 Les données

Les données de l'évolution du prix du blé et du salaire moyen entre 1565 et 1821 sont disponibles sur le [Github de Vincent-Arel-Bundock's](#). Nous récupérerons les données sous la forme d'un fichier CSV où chaque ligne correspond à une année pour laquelle nous connaissons le prix du blé en shilling/quarter, et le salaire hebdomadaire en shilling.

```
[2]: DATA_FILE_PATH = Path("./Wheat_Data.csv")

if not DATA_FILE_PATH.exists():
    data_url = "https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/
    ↪master/csv/HistData/Wheat.csv"
    raw_data = pd.read_csv(data_url)
    raw_data.to_csv(DATA_FILE_PATH, index=False)
```

Voici l'explication des colonnes données sur la [page descriptive du jeu de données](#) associée :

Nom de la colonne	Libellé de colonne	Type de la colonne
Year	Année, par intervalle de 5 de 1565 à 1821	Vecteur numérique
Wheat	Prix du blé (shillings/quart de boisseau)	Vecteur numérique
Wages	Salaire hebdomadaire (shillings)	Vecteur numérique

Les données possèdent également une colonne additionnelle `rownames` que nous définissons ici comme index

```
[3]: raw_data = pd.read_csv(DATA_FILE_PATH, index_col="rownames")
raw_data
```

```
[3]:
```

	Year	Wheat	Wages
rownames			
1	1565	41.0	5.00
2	1570	45.0	5.05
3	1575	42.0	5.08
4	1580	49.0	5.12
5	1585	41.5	5.15
6	1590	47.0	5.25
7	1595	64.0	5.54
8	1600	27.0	5.61
9	1605	33.0	5.69
10	1610	32.0	5.78
11	1615	33.0	5.94
12	1620	35.0	6.01
13	1625	33.0	6.12
14	1630	45.0	6.22
15	1635	33.0	6.30
16	1640	39.0	6.37
17	1645	53.0	6.45

18	1650	42.0	6.50
19	1655	40.5	6.60
20	1660	46.5	6.75
21	1665	32.0	6.80
22	1670	37.0	6.90
23	1675	43.0	7.00
24	1680	35.0	7.30
25	1685	27.0	7.60
26	1690	40.0	8.00
27	1695	50.0	8.50
28	1700	30.0	9.00
29	1705	32.0	10.00
30	1710	44.0	11.00
31	1715	33.0	11.75
32	1720	29.0	12.50
33	1725	39.0	13.00
34	1730	26.0	13.30
35	1735	32.0	13.60
36	1740	27.0	14.00
37	1745	27.5	14.50
38	1750	31.0	15.00
39	1755	35.5	15.70
40	1760	31.0	16.50
41	1765	43.0	17.60
42	1770	47.0	18.50
43	1775	44.0	19.50
44	1780	46.0	21.00
45	1785	42.0	23.00
46	1790	47.5	25.50
47	1795	76.0	27.50
48	1800	79.0	28.50
49	1805	81.0	29.50
50	1810	99.0	30.00
51	1815	78.0	NaN
52	1820	54.0	NaN
53	1821	54.0	NaN

Regardons si le jeu de données contient des données manquantes ? Oui, nous ne disposons pas des salaires hebdomadaires pour les années 1815, 1820 et 1821. Cependant, le sujet nous donne comme objectif de réaliser des représentations graphiques des données. Supprimer les lignes contenant les données manquantes serait ici une erreur. Nous les gardons donc. Toutefois, nous sauvegardons la dernière année pour laquelle nous avons les informations sur le salaire des employés ce qui nous sera utile pour certains graphiques.

```
[4]: raw_data[raw_data.isnull().any(axis=1)]
```

```
[4]:
```

	Year	Wheat	Wages
rownames			
51	1815	78.0	NaN
52	1820	54.0	NaN
53	1821	54.0	NaN

```
[5]: LAST_YEAR_WAGES = 1810
```

Vérifions le type des données de chaque colonne. Il s'agit bien de valeurs numériques, l'année est bien un entier (int64), tandis que le prix du blé et les salaires sont des nombres à virgules flottantes (float64)

```
[6]: raw_data.dtypes
```

```
[6]: Year      int64
Wheat    float64
Wages    float64
dtype: object
```

```
[7]: data = raw_data.copy()
```

1.3 1. Reproduction du graphe de Playfair

Nous cherchons ici à reproduire le [graphe de Playfair](#) représentant sur le même graphique l'évolution du prix du blé, ainsi que du salaire moyen entre 1565 et 1821

```
[8]: fig, ax = plt.subplots(figsize=(16, 7))

# Draw wheat price bar plot
ax.bar(data["Year"], data["Wheat"], width=5, color="#333333")

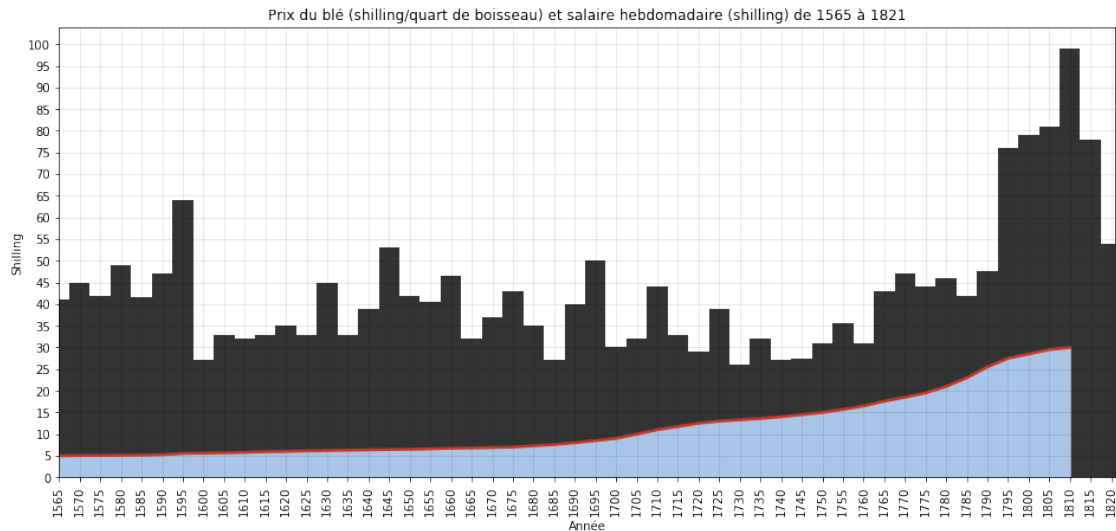
# Draw wages line plot
ax.plot(data["Year"], data["Wages"], color="#c0392b", linewidth=2.5)
ax.fill_between(data["Year"], 0, data["Wages"], color="#a9c6e8", zorder=1.5) #
    ↳ Draw blue area under wages curve

# Labels
ax.set_xlabel("Année")
ax.set_ylabel("Shilling")
ax.set_title("Prix du blé (shilling/quart de boisseau) et salaire hebdomadaire
    ↳ (shilling) de 1565 à 1821")

# Additional parameters
ax.set_xticks(range(1565, 1822, 5))
ax.set_yticks(range(0, 101, 5))
plt.xticks(rotation=90)
```

```
ax.set_xlim(1565, 1821)
ax.grid(True, alpha=0.1, zorder=10, color="black")

plt.show()
```



1.4 2. Amélioration de la présentation des données

Playfair a combiné les deux quantités dans un même graphique en simplifiant les unités “shilling par quart de boisseau de blé” et “shilling par semaine” en “shilling” sur une seule ordonnée. Nous créons donc deux ordonnées différentes avec pour chacune l’unité associée.

```
[9]: fig, ax1 = plt.subplots(figsize=(16, 7))

# Draw wheat price bar plot
ax1.bar(data["Year"], data["Wheat"], width=5, color="#333333")

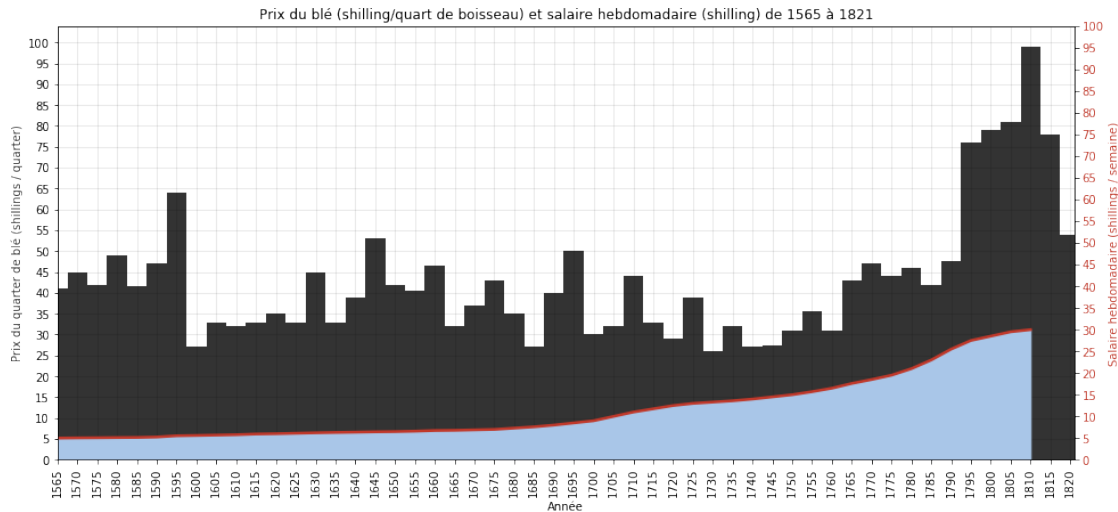
# Draw wages line plot (on another axis)
ax2 = ax1.twinx()
ax2.plot(data["Year"], data["Wages"], color="#c0392b", linewidth=2.5)
ax2.fill_between(data["Year"], 0, data["Wages"], color="#a9c6e8", zorder=1.5) #
    ↳ Draw blue area under wages curve

# Labels
ax1.set_xlabel("Année")
ax1.set_ylabel("Prix du quarter de blé (shillings / quarter)", color="#333333")
ax1.set_title("Prix du blé (shilling/quart de boisseau) et salaire hebdomadaire
    ↳ (shilling) de 1565 à 1821")
ax2.set_ylabel("Salaire hebdomadaire (shillings / semaine)", color="#c0392b")
```

```
# Additional parameters
ax1.set_xticks(range(1565, 1822, 5))
ax1.set_yticks(range(0, 101, 5))
ax1.set_xlim(1565, 1821)
ax1.grid(True, alpha=0.1, zorder=10, color="black")
ax1.tick_params(axis='x', rotation=90)

ax2.set_ylim(bottom=0) # Force ax2 to start at the very bottom, otherwise 0 is
↳ slightly above the bottom
ax2.tick_params(axis="y", labelcolor="#c0392b")
ax2.set_yticks(range(0, 101, 5))

plt.show()
```



1.5 3. Le pouvoir d'achat des ouvriers

L'objectif de Playfair était de montrer que le pouvoir d'achat des ouvriers avait augmenté au cours du temps. Nous allons donc tenter de mieux faire ressortir cette information.

Nous définissons le pouvoir d'achat d'un ouvrier comme la quantité de blé qu'un ouvrier peut acheter avec son salaire hebdomadaire. Observons les variations au cours du temps. Ici, il sera impossible de calculer la pouvoir d'achat pour les années 1815, 1820 et 1821 puisque nous ne disposons pas des informations sur le salaire pour ces années. Notre graphique s'arrêtera donc à l'année `LAST_YEAR_WAGES=1810`.

```
[10]: # Define buying power as the wheat quantity an employee can buy with his weekly
↳ wage
data["Buying_power"] = data["Wages"] / data["Wheat"]
```

```

fig, ax = plt.subplots(figsize=(16, 7))

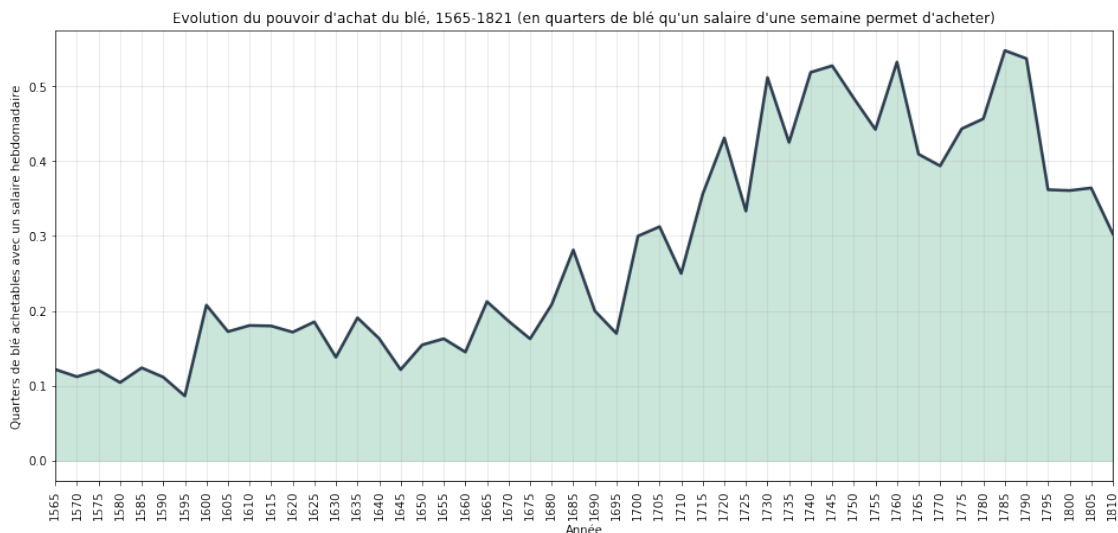
# Draw buying power
ax.plot(data["Year"], data["Buying_power"], color="#2c3e50", linewidth=2.5)
ax.fill_between(data["Year"], 0, data["Buying_power"], color="#a8d5c2", alpha=0.
↳6)

# Labels
ax.set_xlabel("Année")
ax.set_ylabel("Quarters de blé achetés avec un salaire hebdomadaire")
ax.set_title("Evolution du pouvoir d'achat du blé, 1565-1821 (en quarters de
↳blé qu'un salaire d'une semaine permet d'acheter)")

# Additional parameters
# We limit x-axis to LAST_YEAR_WAGES because we don't have wages data after
↳that year
ax.set_xticks(range(1565, LAST_YEAR_WAGES+1, 5))
plt.xticks(rotation=90)
ax.set_xlim(1565, LAST_YEAR_WAGES)
ax.grid(True, alpha=0.3)

plt.show()

```



Nous observons bien une augmentation globale du pouvoir d'achat des ouvriers, en particulier entre 1565 et 1790. Cependant, nous notons une baisse conséquente du pouvoir d'achat après 1790.

Nous proposons ci-après une autre visualisation permettant d'interpréter le pouvoir d'achat d'un ouvrier. Il s'agit du nombre de semaine de travail nécessaires à l'achat d'un quarter de blé. De la même manière, nous notons une diminution globale du nombre de semaine nécessaire de travail à

l'achat d'un quarter de blé, traduisant donc une augmentation du pouvoir d'achat.

```
[11]: # Define buying power as the wheat quantity an employee can buy with hiw weekly wage
data["Weeks_of_Wages_per_quarter_of_Wheat"] = data["Wheat"] / data["Wages"]

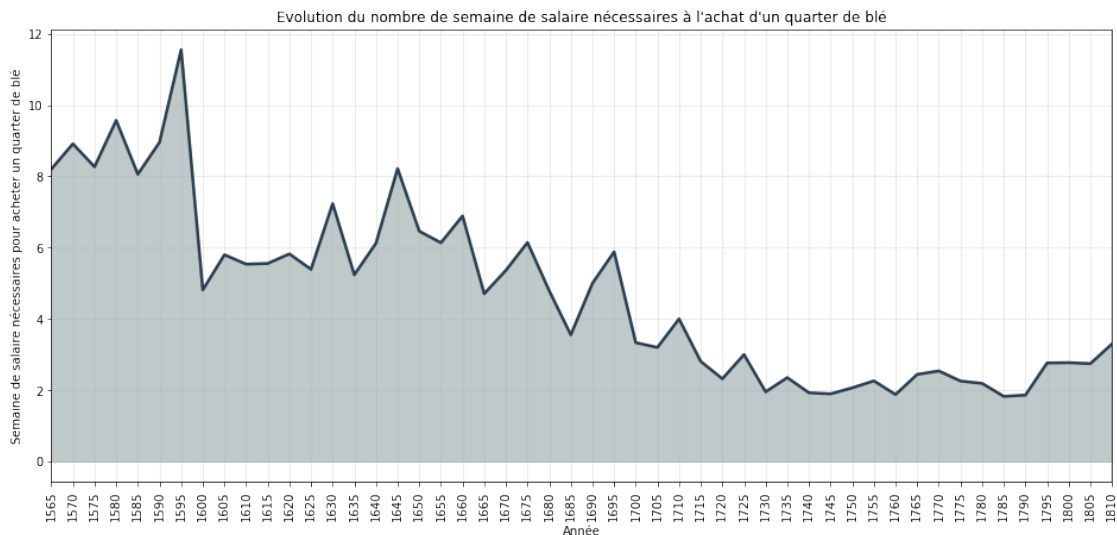
fig, ax = plt.subplots(figsize=(16, 7))

# Draw buying power
ax.plot(data["Year"], data["Weeks_of_Wages_per_quarter_of_Wheat"],
        color="#2c3e50", linewidth=2.5)
ax.fill_between(data["Year"], 0, data["Weeks_of_Wages_per_quarter_of_Wheat"],
               color="#95a5a6", alpha=0.6)

# Labels
ax.set_xlabel("Année")
ax.set_ylabel("Semaine de salaire nécessaires pour acheter un quarter de blé")
ax.set_title("Evolution du nombre de semaine de salaire nécessaires à l'achat d'un quarter de blé")

# Additional parameters
# We limit x-axis to LAST_YEAR_WAGES because we don't have wages data after that year
ax.set_xticks(range(1565, LAST_YEAR_WAGES+1, 5))
plt.xticks(rotation=90)
ax.set_xlim(1565, LAST_YEAR_WAGES)
ax.grid(True, alpha=0.3)

plt.show()
```



Pour observer si les salaires augmentent au même temps que le prix du quarter de blé, nous construisons la représentation ci-dessous. Ce graphique nous permet d'affirmer qu'entre 1565 et 1750, le prix du blé fluctue beaucoup par rapport aux salaires qui, toutefois augmente constamment puisque nous notre courbe ne "reviens pas en arrière".

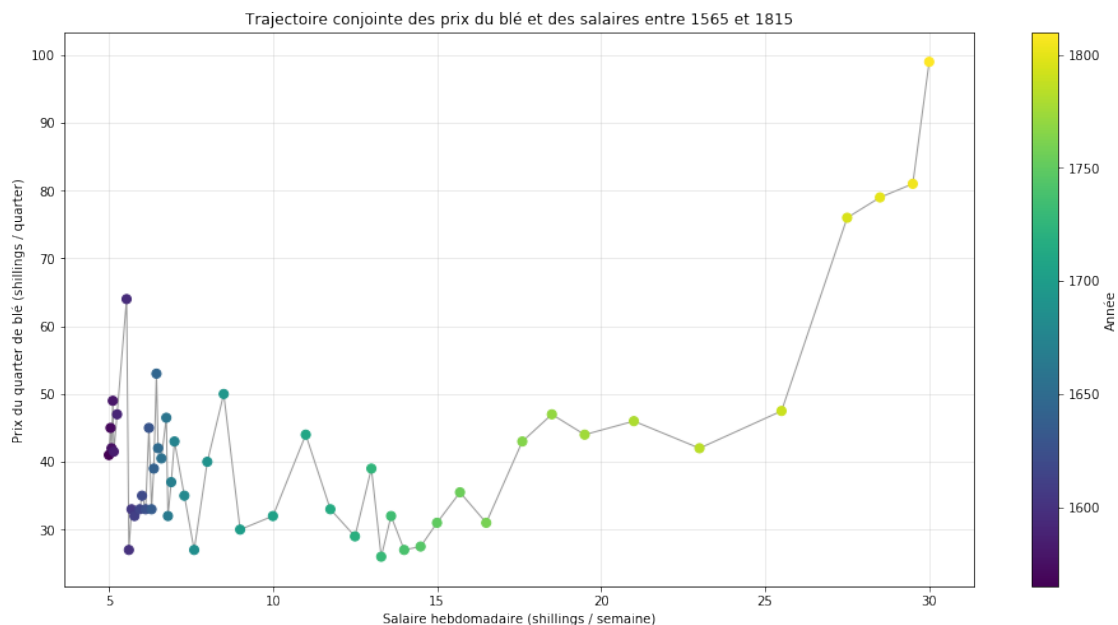
La seconde partie entre 1750 et 1815 montre une forte augmentation des prix du blé mais également une forte augmentation des salaires hebdomadaires comparé à avant 1750.

```
[12]: fig, ax = plt.subplots(figsize=(16, 8))

# Plot Wages and Wheat
ax.plot(data["Wages"], data["Wheat"], color="#999999", linewidth=1, zorder=1)
sc = ax.scatter(data["Wages"], data["Wheat"], c=data["Year"], cmap="viridis",
               ↪s=50, zorder=2)

# Labels
cbar = plt.colorbar(sc, ax=ax)
cbar.set_label("Année")
ax.set_xlabel("Salaire hebdomadaire (shillings / semaine)")
ax.set_ylabel("Prix du quarter de blé (shillings / quarter)")
ax.set_title("Trajectoire conjointe des prix du blé et des salaires entre 1565_
               ↪et 1815")
ax.grid(True, alpha=0.3)

plt.show()
```



1.6 Synthèse

Nous avons proposé 3 visualisations différentes pour tenter de représenter l'évolution du pouvoir d'achat des ouvriers.

- Quarters de blé achetable par semaine de salaire
- Nombre de semaine de salaire nécessaires pour acheter un quarter de blé
- Trajectoire conjointe des prix du blé et des salaires

La première représentation est selon moi la meilleure pour répondre à l'objectif de Playfair qui était de montrer que le pouvoir d'achat a globalement augmenté. Nous avons une grandeur facile à interpréter et à présenter. La représentation de la trajectoire conjointe permet de comprendre davantage la dynamique entre les deux variables et il est moins facile d'interprétation.