

# 04-correlation-analysis

February 16, 2026

## 0.1 04-correlation-analysis.ipynb

Runs the **core Pearson correlation tests** between the main constructs reported in Chapter 4.

**What it does:** - Loads the survey dataset (`data.csv`) and maps Likert responses to numeric values (1-5). - Applies **reverse-coding** for negatively worded items. - Computes construct scores (Fairness, Transparency, Trust, Willingness) as mean composites. - Calculates **Pearson r and p-values** for key relationships (e.g., Fairness–Trust, Trust–Willingness, Transparency–Trust). - Visualizes the results in a small heatmap-style summary plot with r and p displayed.

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from scipy.stats import pearsonr

df = pd.read_csv("data.csv") #reads data from the same repository
df.shape
```

[1]: (84, 28)

```
[2]: aware_col = "Before this survey, were you aware that some companies use_
    ↳AI-based systems in recruitment and selection (e.g., automated CV screening,
    ↳AI ranking of candidates, AI video interview analysis)?"

pref_col = "If you had two similar job opportunities, and the only difference_
    ↳was the recruitment process, which would you prefer?"

screen_col = "For the initial screening of applications (deciding who is_
    ↳invited to the first interview), which option would you feel most_
    ↳comfortable with?"

likelihood_col = "If a company clearly states that it uses AI tools as part of_
    ↳its recruitment process, how likely would you be to apply for a job or_
    ↳internship there?"

# Likert items for constructs
```

```

same_criteria = "Please indicate how strongly you agree or disagree with the
    ↳following statements about AI-based recruitment systems.\n(AI-based
    ↳recruitment includes tools such as automated CV screening, AI ranking of
    ↳candidates, AI video interview analysis, etc.) [AI-based recruitment treats
    ↳all candidates according to the same criteria.]"

reduce_bias = "Please indicate how strongly you agree or disagree with the
    ↳following statements about AI-based recruitment systems.\n(AI-based
    ↳recruitment includes tools such as automated CV screening, AI ranking of
    ↳candidates, AI video interview analysis, etc.) [Using AI in recruitment can
    ↳help reduce human bias in hiring decisions.]"

unfair_rev = "Please indicate how strongly you agree or disagree with the
    ↳following statements about AI-based recruitment systems.\n(AI-based
    ↳recruitment includes tools such as automated CV screening, AI ranking of
    ↳candidates, AI video interview analysis, etc.) [AI-based recruitment systems
    ↳are inherently unfair. (reverse-coded)]"

understand = "Please indicate how strongly you agree or disagree with the
    ↳following statements about AI-based recruitment systems.\n(AI-based
    ↳recruitment includes tools such as automated CV screening, AI ranking of
    ↳candidates, AI video interview analysis, etc.) [I understand, at least in
    ↳general terms, how AI recruitment systems evaluate candidates.]"

explain = "Please indicate how strongly you agree or disagree with the
    ↳following statements about AI-based recruitment systems.\n(AI-based
    ↳recruitment includes tools such as automated CV screening, AI ranking of
    ↳candidates, AI video interview analysis, etc.) [Companies clearly explain
    ↳when and how they use AI in their recruitment process.]"

blackbox_rev = "Please indicate how strongly you agree or disagree with the
    ↳following statements about AI-based recruitment systems.\n(AI-based
    ↳recruitment includes tools such as automated CV screening, AI ranking of
    ↳candidates, AI video interview analysis, etc.) [AI-based recruitment systems
    ↳feel like a "black box" to me. (reverse-coded)]"

trust_item = "Please indicate how strongly you agree or disagree with the
    ↳following statements about AI-based recruitment systems.\n(AI-based
    ↳recruitment includes tools such as automated CV screening, AI ranking of
    ↳candidates, AI video interview analysis, etc.) [I trust AI systems to
    ↳evaluate candidates fairly.]"

```

```

discrim_rev = "Please indicate how strongly you agree or disagree with the
↳following statements about AI-based recruitment systems.\n(AI-based
↳recruitment includes tools such as automated CV screening, AI ranking of
↳candidates, AI video interview analysis, etc.) [I worry that AI recruitment
↳systems might discriminate against certain groups of applicants.
↳(reverse-coded)]"

apply_item = "Please indicate how strongly you agree or disagree with the
↳following statements about AI-based recruitment systems.\n(AI-based
↳recruitment includes tools such as automated CV screening, AI ranking of
↳candidates, AI video interview analysis, etc.) [I would apply to a company
↳that uses AI in its recruitment process.]"

discourage_rev = "Please indicate how strongly you agree or disagree with the
↳following statements about AI-based recruitment systems.\n(AI-based
↳recruitment includes tools such as automated CV screening, AI ranking of
↳candidates, AI video interview analysis, etc.) [Knowing that a company uses
↳AI to screen applications would discourage me from applying.
↳(reverse-coded)]"

```

```

[3]: likert_map = {
    "Strongly disagree": 1,
    "Disagree": 2,
    "Neither agree nor disagree": 3,
    "Agree": 4,
    "Strongly agree": 5
}

```

```

def to_num(s):
    return s.map(likert_map)

def reverse_1to5(x):
    return 6 - x

```

```

[4]: # Numeric items
same_n = to_num(df[same_criteria])
bias_n = to_num(df[reduce_bias])
unfair_n = reverse_1to5(to_num(df[unfair_rev]))           # reverse-coded

under_n = to_num(df[understand])
explain_n = to_num(df[explain])
blackbox_n = reverse_1to5(to_num(df[blackbox_rev]))      # reverse-coded

trust_n = to_num(df[trust_item])
discrim_n = reverse_1to5(to_num(df[discrim_rev]))        # reverse-coded

apply_n = to_num(df[apply_item])

```

```
discourage_n = reverse_1to5(to_num(df[discourage_rev]))    # reverse-coded

# Constructs
Fairness = (same_n + bias_n + unfair_n) / 3
Transparency = (under_n + explain_n + blackbox_n) / 3
Trust = (trust_n + discrim_n) / 2
Willingness = (apply_n + discourage_n + df[likelihood_col]) / 3

constructs = pd.DataFrame({
    "Fairness": Fairness,
    "Transparency": Transparency,
    "Trust": Trust,
    "Willingness": Willingness
})

constructs.describe().loc[["count", "mean", "std"]]
```

```
[4]:
```

	Fairness	Transparency	Trust	Willingness
count	84.000000	84.000000	84.000000	84.000000
mean	3.130952	2.857143	2.821429	3.015873
std	0.720735	0.669243	0.897261	0.713054

```
[5]: r_ft, p_ft = pearsonr(constructs["Fairness"], constructs["Trust"])
r_tw, p_tw = pearsonr(constructs["Trust"], constructs["Willingness"])
r_tt, p_tt = pearsonr(constructs["Transparency"], constructs["Trust"])

print("Pearson correlations:")
print(f"Fairness   Trust: r = {r_ft:.2f}, p = {p_ft:.3f}")
print(f"Trust     Willingness: r = {r_tw:.2f}, p = {p_tw:.3f}")
print(f"Transparency Trust: r = {r_tt:.2f}, p = {p_tt:.3f}")
```

```
Pearson correlations:
Fairness   Trust: r = 0.29, p = 0.008
Trust     Willingness: r = 0.38, p = 0.000
Transparency Trust: r = 0.17, p = 0.113
```

```
[6]: labels = ["Fairness-Trust", "Trust-Willingness", "Transparency-Trust"]
r_vals = np.array([r_ft, r_tw, r_tt])
p_vals = np.array([p_ft, p_tw, p_tt])

# Heatmap-like 1-row display
plt.figure(figsize=(9, 2.2))
im = plt.imshow([r_vals], aspect="auto") # 1 row heatmap
plt.colorbar(label="Pearson r")

plt.xticks(range(len(labels)), labels, rotation=25, ha="right")
plt.yticks([])
```

```

for i, (r, p) in enumerate(zip(r_vals, p_vals)):
    plt.text(i, 0, f"r={r:.2f}\np={p:.3f}", ha="center", va="center",
             ↪fontsize=11)

plt.title("Key Pearson Correlations (with p-values)")
plt.tight_layout()
plt.show()

```

