

03-cronbach-alpha

February 16, 2026

0.1 03-cronbach-alpha.ipynb

Calculates **Internal consistency reliability using the Cronbach's alpha** for the main multi-item constructs in the survey.

What it does: - Loads the raw Google Forms responses (`data.csv`). - Converts Likert responses to numeric values (1-5) and applies **reverse-coding** for negatively worded items. - Computes Cronbach's alpha for each construct (e.g., fairness, transparency, trust, willingness-to-apply). - Outputs small **summary table** of alpha values.

```
[1]: import pandas as pd
import numpy as np

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)]"

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

def cronbach_alpha(df_items):
    # df_items: columns are items, rows are respondents
    df_items = df_items.dropna()
    k = df_items.shape[1]
    if k < 2:
        return np.nan
    item_var = df_items.var(ddof=1).sum()
    total_var = df_items.sum(axis=1).var(ddof=1)
    return (k / (k - 1)) * (1 - item_var / total_var) if total_var != 0 else np.
    nan
    ↳nan

# 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
})

```

[3]:

```

alpha_fair = cronbach_alpha(pd.DataFrame({"same": same_n, "bias": bias_n, "unfair": unfair_n}))
alpha_trans = cronbach_alpha(pd.DataFrame({"understand": under_n, "explain": explain_n, "blackbox": blackbox_n}))
alpha_trust = cronbach_alpha(pd.DataFrame({"trust": trust_n, "discrim": discrim_n}))
alpha_will = cronbach_alpha(pd.DataFrame({"apply": apply_n, "discourage": discourage_n, "likelihood": df[likelihood_col]}))

# Create summary table
alpha_table = pd.DataFrame({
    "Construct": ["Fairness", "Transparency", "Trust", "Willingness"],
    "Cronbach_Alpha": [
        round(alpha_fair, 3),
        round(alpha_trans, 3),
        round(alpha_trust, 3),
        round(alpha_will, 3)
    ]
})
alpha_table

```

[3]:

	Construct	Cronbach_Alpha
0	Fairness	0.186
1	Transparency	0.151
2	Trust	0.514

3 Willingness 0.491