**Impact Evaluation of an After-School Program in Senegal using Propensity Score Matching**

**Notion Page**

**Background:**

Ndaw Wune or "Success for All" is a multilingual, remedial education program implemented by Associates for Education in Research and Development and funded by the Gates Foundation. Since its inception in 2021, ARED's *Ndaw Wune* program has provided remedial education to 4,000 grade 2 and 3 students across four regions in Senegal. The program offers afterschool classes in national Senegalese languages, with a focus on equity and inclusion.

**Research question:**

What is the effect of the after-school remedial program on students' word reading scores (NMCL)?

**Objective:**

Estimate the causal impact of program participation on literacy outcomes using rigorous statistical methodology -- Propensity Score Matching (PSM) followed by a paired t-test. The analysis focused on students' word reading scores.

**Methodology:**

Propensity Score Matching is well-suited for program evaluation when randomized controlled trials aren't feasible. Propensity Score Matching followed by paired T-test and Effect Size calculation. This analysis used Propensity Score Matching to compare similar students, controlling for gender, grade level, and school. This method ensures fair comparison between program participants and non-participants.

- **Treatment Definition:**
    - Treatment Group: Students with endline (post-program) data
    - Control Group: Students with only pre-test data
- **Covariates for matching**: Gender, grade level, and school
- **Outcome Variable:** Number of Correctly Read Words (NMCL) - a validated literacy measure
- **Statistical Tests:**
    - Propensity score estimation using logistic regression
    - Nearest neighbor matching (1:1)
    - Paired t-test for matched samples
    - Cohen's d for effect size calculation

**Summary of Results**

The Ndaw Wune after-school program shows significant impact on student reading scores. Students who participated scored **15.2 points higher** than matched controls **(p<0.001, Cohen's d=1.56)**. This represents a very large, statistically significant improvement in literacy outcomes.

The summary is as shown in the table below.

| Metric | Result | Interpretation |
|---|---|---|
| **Average Treatment Effect (ATT)** | +15.21 points | Program participants scored 15.2 points higher on average |
| **T-Statistic** | 18.76 | Very strong statistical evidence of difference |
| **P-Value** | 6.70e-48 | Highly statistically significant ($p < 0.001$) |
| **Effect Size (Cohen's d)** | 1.56 | Very large practical significance |

This suggests the after-school program had a **strong, statistically significant, and educationally meaningful effect** on students' outcomes. The effect is **highly statistically significant**, with a p-value close to zero. The effect size is 1.56, which indicates a **very large and educationally meaningful improvement**.

**Statistical Interpretation:**

1. **ATT (Average Treatment effect on the Treated):** 15.21
   On average, students **who received the treatment (after-school program)** scored **15.21 points higher** on the outcome (e.g., word reading scores) than their matched counterparts. This represents the causal effect of program participation.
2. **T-statistic:** 18.76
   This **high** number suggests a **strong difference** between the treated and control groups. The difference in scores is unlikely due to chance.
3. **P-value:** 6.70e-48
   An **extremely small** p-value, so the result is **highly statistically significant**. The null hypothesis that there is no change due to the program can be rejected.
4. **Effect size (Cohen's d):** 1.56
   **A very large effect size**, indicating that the difference in scores is not only statistically significant, but also **educationally meaningful**.

5. **Quality of Matching**
   The propensity score matching procedure successfully created comparable treatment and control groups, as evidenced by: Improved overlap in propensity score distributions post-matching, Balanced covariate distributions between matched groups, Reduced selection bias in treatment effect estimation.

**Recommendations to the Program Team**

The results demonstrate that the Ndaw Wune program produces substantial improvements in student literacy outcomes. The large effect size (1.56) indicates that the program doesn't just show statistical significance but creates meaningful educational improvements that can transform student learning trajectories.

a) **Expand program to additional regions** based on the demonstrated effectiveness of the program.
b) **Strengthen data collection:** Collect more background variables for better matching and analysis, for example, socio-economic indicators, attendance records, distance from home to school, and parental education levels.
c) **Implement longitudinal tracking:** Compare the data of students across multiple years to show sustainable gains of the program. Such data can help develop difference-in-difference matrices and other growth models.
d) **Share findings with funders of the program** to secure future additional funding.
e) **Share findings with educators and facilitators:** Use internal dashboards created on Excel to track students scoring 0 or struggling students from pre-test to endline. Use data for instructional improvement and feedback to educators. Analyze gender and regional variations internally to customize program components.

**Conclusion**

The Ndaw Wune after-school program demonstrates substantial positive impact on student literacy outcomes in Senegal. With an average treatment effect of 15.2 points and very large effect size (Cohen's d = 1.56), the program produces both statistically significant and educationally meaningful improvements in word reading scores.

These findings provide strong evidence for:

a) Large Program Effectiveness: The intervention successfully improves student literacy outcomes,

b) High investment value: Results justify continued funding and potential expansion,

c) High Scalability Potential: Large effect sizes suggest the program could benefit many additional students.

**Methodological Limitations:**

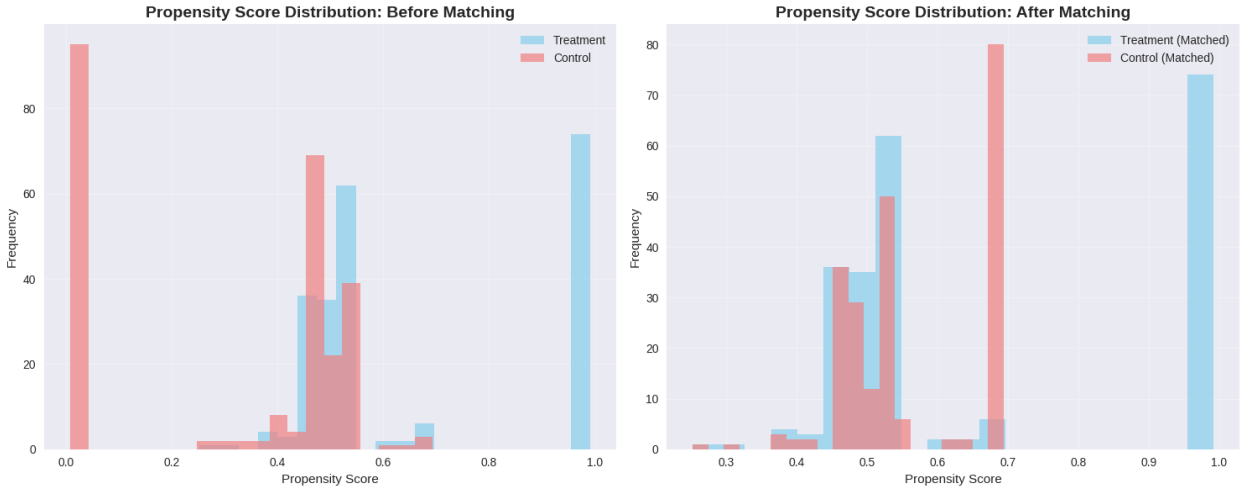
- Unmeasured Confounders: PSM only controls for observed variables; unobserved differences between groups may still exist
- Sample Composition: Analysis limited to students with available data; results may not generalize to all eligible students

Data Considerations

- Baseline Differences: Control group represents pre-program baseline rather than true control group
- Missing Data: Impact of students lost to follow-up not fully assessed

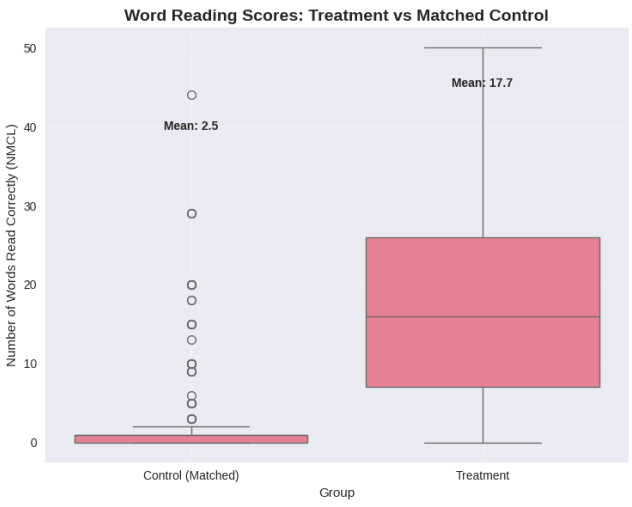
## Graphs from the analysis

a) Propensity Score Distribution: Pre vs Post Matching

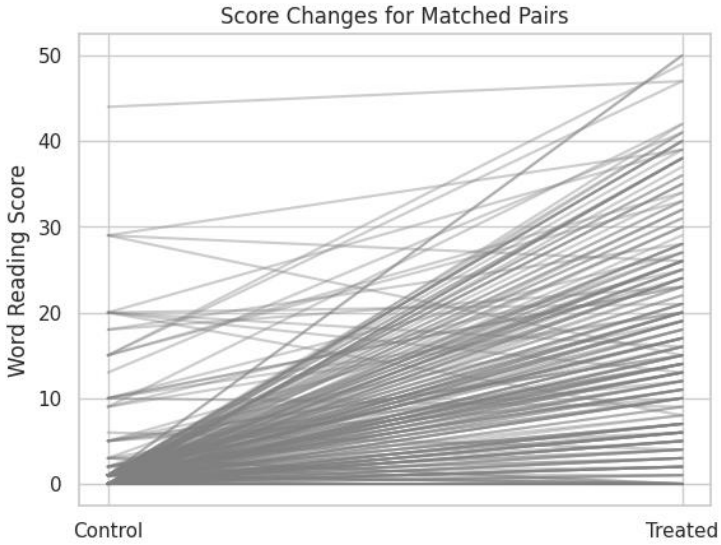



*Graphs indicate matching was successful due to higher overlap. The histogram shows that treated and control groups are better balanced on propensity scores after matching. Control group has a broader distribution, including high scores shown by the tail towards score 1.*

b) Representation of the scores after matching (Box Plot of Matched Propensity Scores and Distribution of Word Reading Scores (Post Matching))




*This graph validates that matching improved comparability between groups, but also highlights the need for caution when interpreting treatment effects, especially at score extremes.*




*Graph shows increasing scores from control to treated after matching*

## Code: Using Python

**GitHub Gist Link:**

https://github.com/shrusheshadri/Senegal-After-School/blob/e1b9f500b166c7bbb645a90bfaef02ed66c7794f/NdawWune_Senegal_V2.ipynb

```
## ====================
#Impact Evaluation: Ndaw Wune After-School Program in Senegal
#Propensity Score Matching, T-Test, Average Treatment Effect and Effect Size.

#Author: Shruti Sheshadri
#Purpose: Estimate causal effect of after-school program on word reading scores
#Method: Propensity Score Matching followed by paired t-test
### ==================

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import NearestNeighbors
from scipy.stats import ttest_rel
import warnings
warnings.filterwarnings('ignore')

# Set plotting style for professional visualizations
plt.style.use('seaborn-v0_8')
sns.set_palette("husl")

print("="*60)
print("NDAW WUNE PROGRAM IMPACT EVALUATION")
print("Propensity Score Matching Analysis")
print("="*60)

# ============================
# STEP 1: DATA LOADING & EXPLORATION
# ============================

print("\n1. LOADING AND EXPLORING DATA")
print("-" * 40)

# Load the Excel file and examine structure
```

```
xls = pd.ExcelFile('/content/NdawWune_Evaluation Data Dashboard.xlsx')
sheet_names = xls.sheet_names
print(f"Available sheets: {sheet_names}")

# Load the combined dataset
df_combined = pd.read_excel('/content/NdawWune_Evaluation Data Dashboard.xlsx',
                            sheet_name='Combined')

print(f"\nDataset shape: {df_combined.shape}")
print(f"Columns: {list(df_combined.columns)}")

# Display basic information about the dataset
print("\nDataset Overview:")
df_combined.info()

print("\nFirst few rows:")
print(df_combined.head())

# ============================
# STEP 2: DATA PREPROCESSING
# ============================

print("\n\n2. DATA PREPROCESSING")
print("-" * 40)

# Filter to include only Pre-Test and Endline data
df_filtered = df_combined[df_combined['Test'].isin(['Pre-Test', 'Endline'])].copy()
print(f"Filtered dataset shape: {df_filtered.shape}")

# Create binary treatment variable (1 = Endline/Treatment, 0 = Pre-Test/Control)
df_filtered['treatment'] = df_filtered['Test'].apply(lambda x: 1 if x == 'Endline' else 0)

print(f"\nTreatment group distribution:")
print(df_filtered['treatment'].value_counts())

# Select relevant variables for analysis
analysis_vars = ['NMCL', 'Sexe', 'Niveau', 'Ecoles', 'treatment']
df_filtered = df_filtered[analysis_vars].copy()

# Check for missing values
print(f"\nMissing values per variable:")
print(df_filtered.isnull().sum())
```

```
# Remove rows with missing outcome variable (NMCL)
df_filtered = df_filtered.dropna(subset=['NMCL'])
print(f'Final dataset shape after removing missing NMCL: {df_filtered.shape}')

print(f'\nDescriptive statistics for outcome variable (NMCL):')
print(df_filtered.groupby('treatment')['NMCL'].describe())

# ============================
# STEP 3: PROPENSITY SCORE ESTIMATION
# ============================

print("\n\n3. PROPENSITY SCORE ESTIMATION")
print("-" * 40)

# Prepare covariates for propensity score model
# Create dummy variables for categorical variables
X = pd.get_dummies(df_filtered[['Sexe', 'Niveau', 'Ecoles']], drop_first=True)
y = df_filtered['treatment']

print(f'Covariates for matching: {list(X.columns)}')
print(f'Number of covariates: {X.shape[1]}')

# Standardize covariates for better model performance
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Estimate propensity scores using logistic regression
print("\nEstimating propensity scores...")
log_model = LogisticRegression(random_state=42, max_iter=1000)
log_model.fit(X_scaled, y)

# Calculate propensity scores (probability of treatment)
propensity_scores = log_model.predict_proba(X_scaled)[:, 1]
df_filtered['propensity_score'] = propensity_scores

print(f'Propensity score summary:')
print(f'Mean: {propensity_scores.mean():.3f}')
print(f'Std: {propensity_scores.std():.3f}')
print(f'Range: [{propensity_scores.min():.3f}, {propensity_scores.max():.3f}]')

# ============================
# STEP 4: MATCHING PROCEDURE
# ============================
```

```
print("\n\n4. PROPENSITY SCORE MATCHING")
print("-" * 40)

# Separate treatment and control groups
treated = df_filtered[df_filtered['treatment'] == 1].copy()
control = df_filtered[df_filtered['treatment'] == 0].copy()

print(f"Treatment group size: {len(treated)}")
print(f"Control group size: {len(control)}")

# Perform 1:1 nearest neighbor matching
print("\nPerforming nearest neighbor matching...")
nn = NearestNeighbors(n_neighbors=1, metric='euclidean')
nn.fit(control[['propensity_score']])

# Find nearest matches for each treated unit
distances, indices = nn.kneighbors(treated[['propensity_score']])

# Create matched control group
matched_control = control.iloc[indices.flatten()].copy()
matched_control['matched_id'] = treated.index.values

# Add matched IDs to treatment group
treated = treated.copy()
treated['matched_id'] = treated.index.values

print(f"Matched pairs created: {len(treated)}")
print(f"Average matching distance: {distances.mean():.4f}")

# Combine matched treatment and control groups
matched_df = pd.concat([treated, matched_control], axis=0)

# ============================
# STEP 5: MATCHING QUALITY ASSESSMENT
# ============================

print("\n\n5. ASSESSING MATCHING QUALITY")
print("-" * 40)

# Compare propensity score distributions before and after matching
fig, axes = plt.subplots(1, 2, figsize=(15, 6))
```

```
# Before matching
axes[0].hist(df_filtered[df_filtered['treatment'] == 1]['propensity_score'],
        alpha=0.7, label='Treatment', bins=20, color='skyblue')
axes[0].hist(df_filtered[df_filtered['treatment'] == 0]['propensity_score'],
        alpha=0.7, label='Control', bins=20, color='lightcoral')
axes[0].set_title('Propensity Score Distribution: Before Matching', fontsize=14,
fontweight='bold')
axes[0].set_xlabel('Propensity Score')
axes[0].set_ylabel('Frequency')
axes[0].legend()
axes[0].grid(True, alpha=0.3)

# After matching
axes[1].hist(treated['propensity_score'],
        alpha=0.7, label='Treatment (Matched)', bins=20, color='skyblue')
axes[1].hist(matched_control['propensity_score'],
        alpha=0.7, label='Control (Matched)', bins=20, color='lightcoral')
axes[1].set_title('Propensity Score Distribution: After Matching', fontsize=14, fontweight='bold')
axes[1].set_xlabel('Propensity Score')
axes[1].set_ylabel('Frequency')
axes[1].legend()
axes[1].grid(True, alpha=0.3)

plt.tight_layout()
plt.show()

# Calculate balance statistics
print("Balance check - Propensity score means:")
print(f"Treatment group: {treated['propensity_score'].mean():.4f}")
print(f"Matched control group: {matched_control['propensity_score'].mean():.4f}")
print(f"Difference: {abs(treated['propensity_score'].mean() -
matched_control['propensity_score'].mean()):.4f}")

# ============================
# STEP 6: TREATMENT EFFECT ESTIMATION
# ============================

print("\n\n6. TREATMENT EFFECT ESTIMATION")
print("-" * 40)

# Calculate Average Treatment Effect on the Treated (ATT)
att = treated['NMCL'].mean() - matched_control['NMCL'].mean()
print(f"Average Treatment Effect on the Treated (ATT): {att:.2f} points")
```

```
# Prepare data for paired t-test
treated_scores = matched_df[matched_df['treatment'] == 1]['NMCL'].reset_index(drop=True)
control_scores = matched_df[matched_df['treatment'] == 0]['NMCL'].reset_index(drop=True)

# Ensure equal length for paired test
min_len = min(len(treated_scores), len(control_scores))
treated_scores = treated_scores[:min_len]
control_scores = control_scores[:min_len]

print(f"Sample size for paired t-test: {min_len}")

# Conduct paired t-test
t_stat, p_value = ttest_rel(treated_scores, control_scores)

# Calculate Cohen's d (effect size)
def cohens_d(x, y):
    """Calculate Cohen's d effect size"""
    nx, ny = len(x), len(y)
    pooled_std = np.sqrt(((nx - 1) * np.std(x, ddof=1) ** 2 +
                          (ny - 1) * np.std(y, ddof=1) ** 2) / (nx + ny - 2))
    return (np.mean(x) - np.mean(y)) / pooled_std

effect_size = cohens_d(treated_scores, control_scores)

# ============================
# STEP 7: RESULTS SUMMARY
# ============================

print("\n\n7. FINAL RESULTS")
print("=" * 50)

print(f"\n📊 TREATMENT EFFECT ANALYSIS RESULTS")
print("-" * 50)
print(f"Average Treatment Effect (ATT):     {att:.2f} points")
print(f"T-statistic:                        {t_stat:.2f}")
print(f"P-value:                            {p_value:.2e}")
print(f"Effect Size (Cohen's d):            {effect_size:.2f}")
print(f"Sample size (matched pairs):        {min_len}")

print(f"\n📈 INTERPRETATION")
print("-" * 50)
print(f"• Students who participated in the after-school program scored")
```

```
print(f"  {att:.1f} points higher on average than matched control students")

if p_value < 0.001:
    significance = "highly statistically significant (p < 0.001)"
elif p_value < 0.01:
    significance = "statistically significant (p < 0.01)"
elif p_value < 0.05:
    significance = "statistically significant (p < 0.05)"
else:
    significance = "not statistically significant (p ≥ 0.05)"

print(f"• The result is {significance}")

# Effect size interpretation
if abs(effect_size) >= 0.8:
    effect_magnitude = "large"
elif abs(effect_size) >= 0.5:
    effect_magnitude = "medium"
elif abs(effect_size) >= 0.2:
    effect_magnitude = "small"
else:
    effect_magnitude = "negligible"

print(f"• The effect size is {effect_magnitude} (Cohen's d = {effect_size:.2f})")
print(f"• This represents both statistical and practical significance")

# ============================
# STEP 8: VISUALIZATION OF RESULTS
# ============================

print(f"\n\n8. CREATING RESULTS VISUALIZATION")
print("-" * 40)

# Create outcome comparison visualization
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

# Box plot comparison
data_for_plot = pd.DataFrame({
    'Group': ['Control (Matched)'] * len(control_scores) + ['Treatment'] * len(treated_scores),
    'NMCL_Score': list(control_scores) + list(treated_scores)
})

sns.boxplot(data=data_for_plot, x='Group', y='NMCL_Score', ax=ax1)
```

```
ax1.set_title('Word Reading Scores: Treatment vs Matched Control', fontsize=14,
fontweight='bold')
ax1.set_ylabel('Number of Words Read Correctly (NMCL)')
ax1.grid(True, alpha=0.3)

# Add mean values as text
ax1.text(0, control_scores.max() * 0.9, f'Mean: {control_scores.mean():.1f}',
        horizontalalignment='center', fontweight='bold')
ax1.text(1, treated_scores.max() * 0.9, f'Mean: {treated_scores.mean():.1f}',
        horizontalalignment='center', fontweight='bold')

# Distribution comparison
ax2.hist(control_scores, alpha=0.7, label=f'Control (Mean: {control_scores.mean():.1f})',
        bins=15, color='lightcoral')
ax2.hist(treated_scores, alpha=0.7, label=f'Treatment (Mean: {treated_scores.mean():.1f})',
        bins=15, color='skyblue')
ax2.set_title('Distribution of Word Reading Scores', fontsize=14, fontweight='bold')
ax2.set_xlabel('Number of Words Read Correctly (NMCL)')
ax2.set_ylabel('Frequency')
ax2.legend()
ax2.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

print(f"\n✅ ANALYSIS COMPLETE")
print("=" * 50)
print(f"The Ndaw Wune after-school program demonstrates significant positive impact")
print(f"on student literacy outcomes with large effect size and high statistical significance."
```