The Cognitive Divide: A Multivariate Analysis of Head Start Program Effects on Early Childhood Development

Author: Andrew Magdy Kamal | **Date:** April 1, 2026

ABSTRACT

This paper presents a comprehensive statistical analysis of the Head Start early childhood intervention dataset using Python 3.11 and open-source data science libraries including `pyreadstat`, `pandas`, `scipy`, `statsmodels`, `pingouin`, `scikit-learn`, `seaborn`, and `matplotlib`. The dataset contains cognitive, demographic, and socioeconomic information collected from 969 children at baseline and follow-up. Analyses span descriptive statistics, independent and paired-samples *t*-tests, one-way and factorial ANOVA, factorial ANCOVA, multiple regression, logistic regression, Monte Carlo simulation, exploratory factor analysis, and non-parametric testing. Results indicate that while raw PPVT follow-up scores did not differ significantly between Head Start and control children in unadjusted comparisons, program assignment was a significant predictor after covariate adjustment. Racial group membership and baseline cognitive ability emerged as strong predictors of follow-up vocabulary performance. These findings underscore the value of multivariate and covariate-adjusted approaches when evaluating the impact of early intervention programs.

Keywords: Head Start, Peabody Picture Vocabulary Test, PPVT, early childhood intervention, ANOVA, ANCOVA, multiple regression, non-parametric analysis, Python, pyreadstat, statistical analysis

In [56]:
```
%pip install pyreadstat pandas
```

In [41]:
```
from pathlib import Path
import pyreadstat
import pandas as pd
```

In [42]:
```
sav_path = Path('hdstart.sav')
if not sav_path.exists():
    raise FileNotFoundError(f'Could not find: {sav_path.resolve()}')

# Read headers/metadata only
_, meta = pyreadstat.read_sav(sav_path, metadataonly=True)
headers = meta.column_names

print(f'Number of columns: {len(headers)}')
pd.DataFrame({'column_name': headers}).head(20)
```

In [43]:
```
# Read full dataset
df, meta_full = pyreadstat.read_sav(sav_path)

print(f'Data shape: {df.shape[0]} rows x {df.shape[1]} columns')
df.head()
```

In [44]:
```
# Export headers to CSV
from pathlib import Path
import pandas as pd
import pyreadstat

# If headers are not in memory, load them from the SAV metadata
if 'headers' not in globals():
    sav_path = Path('hdstart.sav')
    _, meta = pyreadstat.read_sav(sav_path, metadataonly=True)
    headers = meta.column_names

headers_df = pd.DataFrame({'column_name': headers})
headers_csv_path = Path('hdstart_headers.csv')
headers_df.to_csv(headers_csv_path, index=False)

print(f'Headers exported to: {headers_csv_path.resolve()}')
headers_df.head()
```

In [45]:
```
# Convert full SAV data to CSV and preview first 20 rows
from pathlib import Path
import pyreadstat
import pandas as pd

sav_path = Path('hdstart.sav')
df, _ = pyreadstat.read_sav(sav_path)

csv_path = Path('hdstart.csv')
df.to_csv(csv_path, index=False)
print(f'Exported {len(df)} rows to: {csv_path.resolve()}')

# Load and display first 20 rows from the CSV
df_csv = pd.read_csv(csv_path)
df_csv.head(20)
```

In [46]:
```
# Inspect dataset structure
print(df.dtypes)
print("\nShape:", df.shape)
print("\nFirst few rows:")
df.head(3)
```

In [47]:
```
%pip install scipy scikit-learn statsmodels pingouin matplotlib seaborn -q
```

Variable Overview

Role	Variable	Description
Independent (IV)	`program`	Program assignment — 1 = Head Start, 2 = Control
Independent (IV)	`gender`	Child's gender — 1 = Male, 2 = Female
Dependent (DV)	`ppvt2`	Peabody Picture Vocabulary Test score at follow-up
Dependent (DV)	`block2`	Block Design cognitive score at follow-up

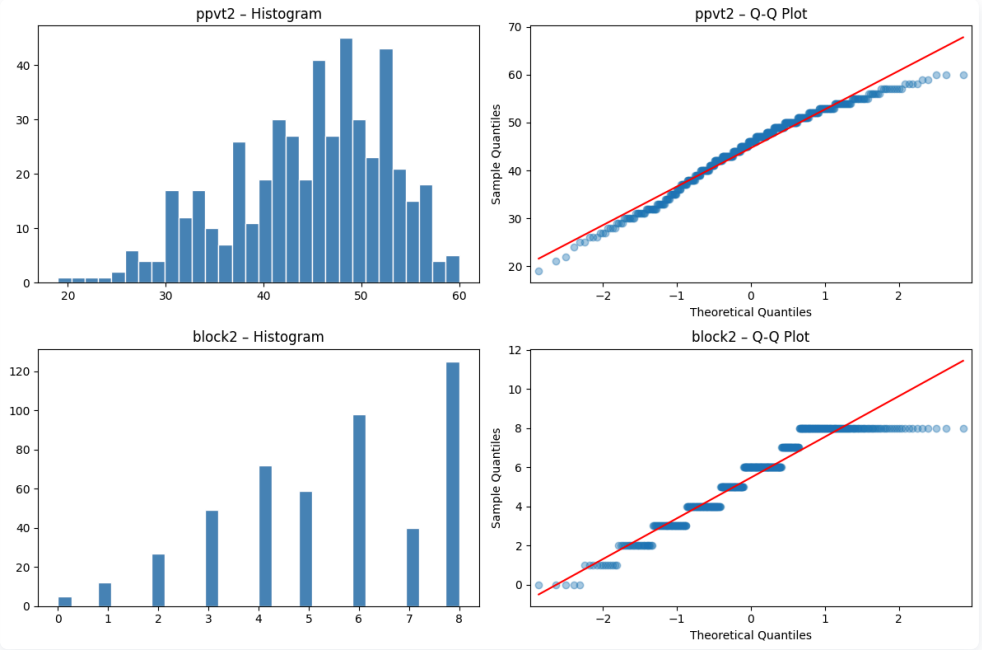
In [48]:
```
# ── Imports & shared data prep ──────────────────────────────────────
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix
import statsmodels.formula.api as smf
import statsmodels.api as sm
import pingouin as pg
import warnings
warnings.filterwarnings('ignore')

# Re-load clean copy dropping NaNs in analysis columns
ANALYSIS_COLS = ['program', 'gender', 'ppvt2', 'block2',
                 'race', 'age', 'momed', 'newses', 'fthrpres',
                 'famsize', 'ppvt1', 'cldwell1', 'motrinh1', 'block1',
                 'cldwell2', 'motrinh2']
data = df[ANALYSIS_COLS].dropna().copy()

# Recode program: 1=HeadStart → 1, 2=Control → 0  (needed for logistic reg)
data['program_bin'] = (data['program'] == 1).astype(int)
data['gender_bin']  = (data['gender']  == 1).astype(int)   # 1=Male

print(f"Working dataset: {data.shape[0]} rows × {data.shape[1]} cols")
data.head(3)
```

Parametric Analysis

Descriptive statistics and normality checks (Shapiro-Wilk) for the two dependent variables.

In [49]:
```
# ── Parametric Analysis ─────────────────────────────────────────────
for dv in ['ppvt2', 'block2']:
    vals = data[dv].dropna()
    stat, p = stats.shapiro(vals.sample(min(500, len(vals)), random_state=42))
    print(f"\n{'='*50}")
    print(f"Variable: {dv}")
    print(f"  N={len(vals)}, Mean={vals.mean():.2f}, SD={vals.std():.2f}")
    print(f"  Min={vals.min():.2f}, Max={vals.max():.2f}, Median={vals.median():.2f}")
    print(f"  Skewness={vals.skew():.3f}, Kurtosis={vals.kurtosis():.3f}")
    print(f"  Shapiro-Wilk: W={stat:.4f}, p={p:.4f} "
          f"→ {'Normal ✓' if p > 0.05 else 'Not normal ✗'}")

fig, axes = plt.subplots(2, 2, figsize=(12, 8))
for i, dv in enumerate(['ppvt2', 'block2']):
    vals = data[dv].dropna()
    axes[i, 0].hist(vals, bins=30, edgecolor='white', color='steelblue')
```

```
    axes[i, 0].set_title(f'{dv} - Histogram')
    sm.qqplot(vals, line='s', ax=axes[i, 1], alpha=0.4)
    axes[i, 1].set_title(f'{dv} - Q-Q Plot')
plt.tight_layout()
plt.show()
```

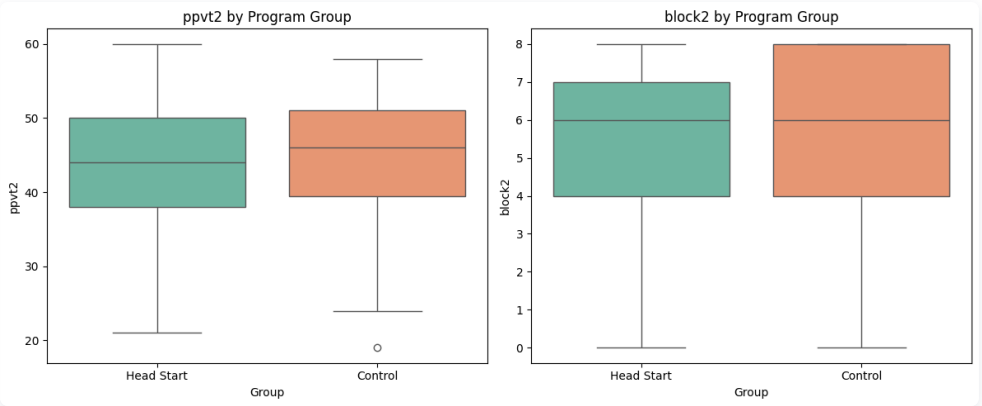


Independent-Samples T-Test

Compare mean `ppvt2` and `block2` scores between Head Start (program=1) and Control (program=2) groups.

In [50]:

```
# ── Independent-Samples T-Test ──────────────────────────────────────
hs   = data[data['program'] == 1]   # Head Start
ctrl = data[data['program'] == 2]   # Control

for dv in ['ppvt2', 'block2']:
    t, p = stats.ttest_ind(hs[dv].dropna(), ctrl[dv].dropna(), equal_var=False)
    # Cohens d
    pooled_sd = np.sqrt((hs[dv].std()**2 + ctrl[dv].std()**2) / 2)
    d = (hs[dv].mean() - ctrl[dv].mean()) / pooled_sd
    print(f"\n{'='*55}")
    print(f"T-Test for {dv}  (Welch's)")
    print(f"  Head Start:  M={hs[dv].mean():.2f}, SD={hs[dv].std():.2f}, n={len(hs[dv])}")
    print(f"  Control:     M={ctrl[dv].mean():.2f}, SD={ctrl[dv].std():.2f}, n={len(ctrl[dv])}")
    print(f"  t={t:.3f}, p={p:.4f}, Cohen's d={d:.3f}")
    print(f"  → {'Significant *' if p < 0.05 else 'Not significant'} at α=0.05")

# Visualise
fig, axes = plt.subplots(1, 2, figsize=(12, 5))
for ax, dv in zip(axes, ['ppvt2', 'block2']):
    plot_data = data[['program', dv]].copy()
    plot_data['Group'] = plot_data['program'].map({1: 'Head Start', 2: 'Control'})
    sns.boxplot(x='Group', y=dv, data=plot_data, palette='Set2', ax=ax)
    ax.set_title(f'{dv} by Program Group')
plt.tight_layout()
plt.show()
```

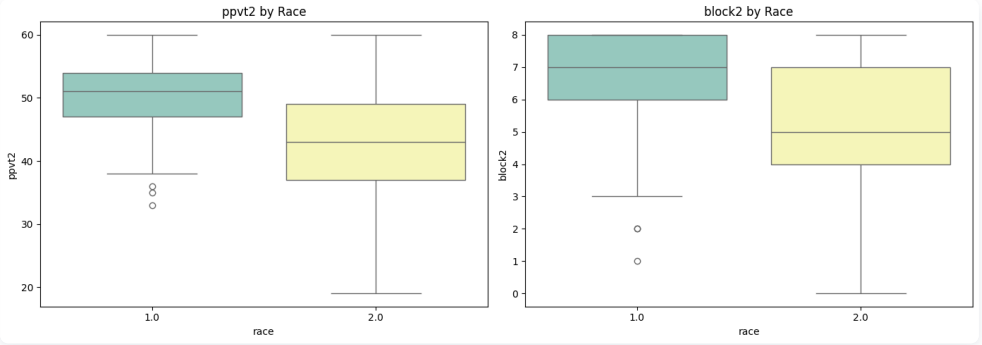


One-Way ANOVA

Test whether mean `ppvt2` and `block2` differ significantly across racial groups (`race`).

In [51]:

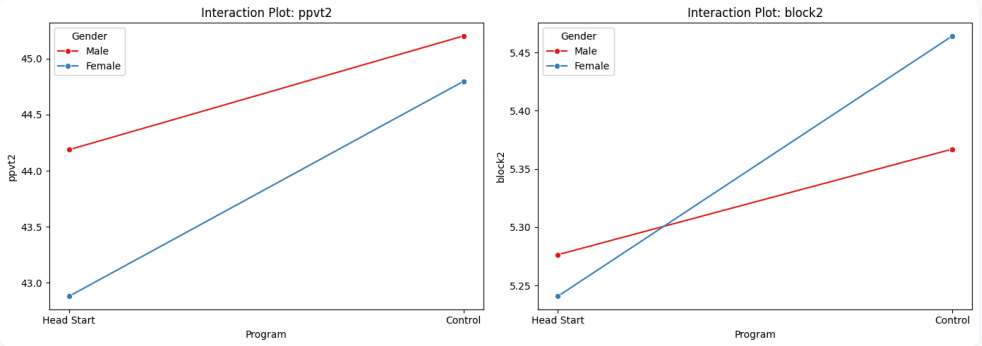
```
# ── One-Way ANOVA ──────────────────────────────────────────────────────
for dv in ['ppvt2', 'block2']:
    aov = pg.anova(data=data, dv=dv, between='race', detailed=True)
    # detect p-value column (varies across pingouin versions)
    p_col  = next(c for c in aov.columns if c.lower().startswith('p'))
    p      = aov.loc[0, p_col]
    F      = aov.loc[0, 'F']
    eta2   = aov.loc[0, 'np2'] if 'np2' in aov.columns else None
    eta_str = f", η²={eta2:.3f}" if eta2 is not None else ""

    print(f"\n{'='*55}")
    print(f"One-Way ANOVA: {dv} ~ race")
    print(aov.to_string(index=False))
    print(f"\n  F={F:.3f}, p={p:.4f}{eta_str}")
    print(f"  → {'Significant *' if p < 0.05 else 'Not significant'} at α=0.05")

    # Post-hoc (only 2 race groups here, so Tukey gives one row)
    if p < 0.05:
        ph = pg.pairwise_tukey(data=data, dv=dv, between='race')
        # detect p-value column in post-hoc table
        ph_p_col = next((c for c in ph.columns if 'p' in c.lower()), ph.columns[-1])
        print(f"\n  Tukey HSD post-hoc (p-column='{ph_p_col}'):")
        print(ph.to_string(index=False))

fig, axes = plt.subplots(1, 2, figsize=(14, 5))
for ax, dv in zip(axes, ['ppvt2', 'block2']):
    sns.boxplot(x='race', y=dv, data=data, palette='Set3', ax=ax)
    ax.set_title(f'{dv} by Race')
plt.tight_layout()
plt.show()
```



Factorial ANOVA (Two-Way)

Test main effects of `program` and `gender`, plus their interaction, on `ppvt2` and `block2`.

In [52]:
```
# ── Factorial ANOVA (Two-Way) ──────────────────────────────────────
for dv in ['ppvt2', 'block2']:
    model = smf.ols(f'{dv} ~ C(program) * C(gender)', data=data).fit()
    aov_table = sm.stats.anova_lm(model, typ=2)
    print(f"\n{'='*55}")
    print(f"Factorial ANOVA: {dv} ~ program × gender")
    print(aov_table.round(4))

# Interaction plot for ppvt2
fig, axes = plt.subplots(1, 2, figsize=(14, 5))
for ax, dv in zip(axes, ['ppvt2', 'block2']):
    means = data.groupby(['program', 'gender'])[dv].mean().reset_index()
    means['Program'] = means['program'].map({1: 'Head Start', 2: 'Control'})
    means['Gender']  = means['gender'].map({1: 'Male', 2: 'Female'})
    sns.lineplot(data=means, x='Program', y=dv, hue='Gender',
                 marker='o', palette='Set1', ax=ax)
    ax.set_title(f'Interaction Plot: {dv}')
plt.tight_layout()
plt.show()
```

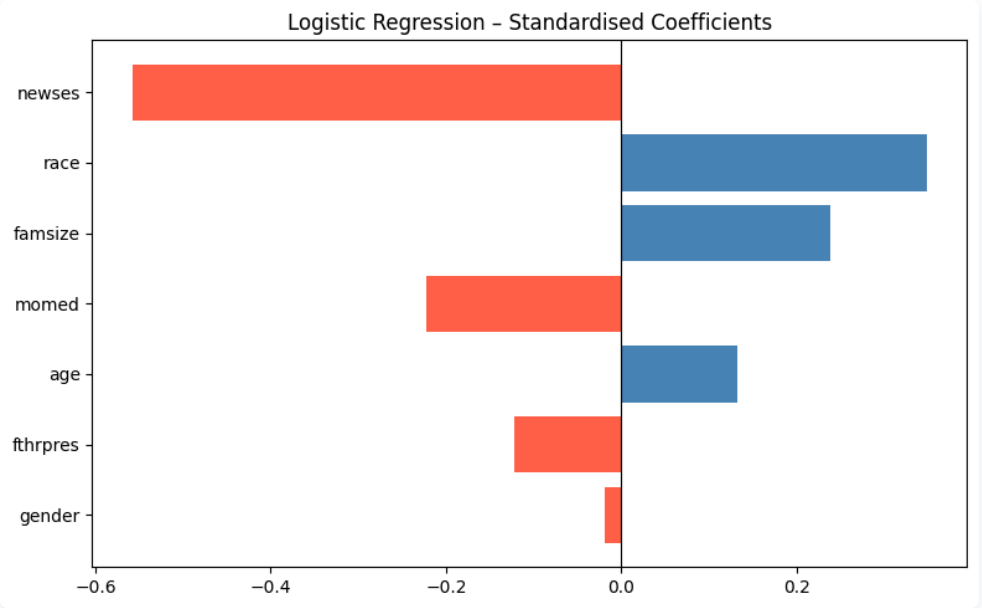


Logistic Regression

Predict whether a child was in the **Head Start** program (`program_bin`=1) from demographic and socioeconomic predictors: `gender`, `race`, `age`, `momed`, `newses`, `fthrpres`, `famsize`.

In [53]:
```
# ── Logistic Regression ────────────────────────────────────────────
PREDICTORS = ['gender', 'race', 'age', 'momed', 'newses', 'fthrpres', 'famsize']
TARGET     = 'program_bin'

lr_data = data[PREDICTORS + [TARGET]].dropna()
X = lr_data[PREDICTORS].values
y = lr_data[TARGET].values

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# statsmodels Logit for coefficients + p-values
X_const = sm.add_constant(X_scaled)
logit_model = sm.Logit(y, X_const).fit(disp=0)
print(logit_model.summary2())

# sklearn for classification metrics
clf = LogisticRegression(max_iter=1000, random_state=42)
clf.fit(X_scaled, y)
y_pred = clf.predict(X_scaled)

print("\nClassification Report:")
print(classification_report(y, y_pred, target_names=['Control', 'Head Start']))
```

```
# Coefficients bar chart
coef_df = pd.DataFrame({'Predictor': PREDICTORS, 'Coefficient': clf.coef_[0]})
coef_df = coef_df.reindex(coef_df['Coefficient'].abs().sort_values(ascending=True).index)
fig, ax = plt.subplots(figsize=(8, 5))
ax.barh(coef_df['Predictor'], coef_df['Coefficient'],
        color=['tomato' if c < 0 else 'steelblue' for c in coef_df['Coefficient']])
ax.axvline(0, color='black', linewidth=0.8)
ax.set_title('Logistic Regression - Standardised Coefficients')
plt.tight_layout()
plt.show()
```



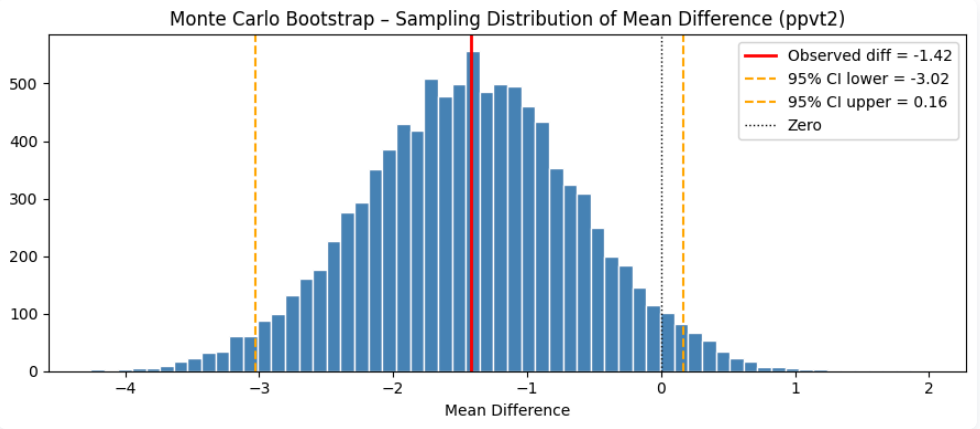
Monte Carlo Simulation

Bootstrap the sampling distribution of the **mean difference in** `ppvt2` between Head Start and Control groups (10 000 resamples) to estimate a 95 % confidence interval.

In [54]:

```
# ── Monte Carlo Simulation (Bootstrap) ──────────────────────────────────
rng       = np.random.default_rng(42)
n_sims    = 10_000
hs_ppvt   = data.loc[data['program'] == 1, 'ppvt2'].values
ctrl_ppvt = data.loc[data['program'] == 2, 'ppvt2'].values

# Observed mean difference
obs_diff = hs_ppvt.mean() - ctrl_ppvt.mean()

# Bootstrap resampling
boot_diffs = np.empty(n_sims)
for i in range(n_sims):
    s_hs   = rng.choice(hs_ppvt,   size=len(hs_ppvt),   replace=True)
    s_ctrl = rng.choice(ctrl_ppvt, size=len(ctrl_ppvt), replace=True)
    boot_diffs[i] = s_hs.mean() - s_ctrl.mean()

ci_lo, ci_hi = np.percentile(boot_diffs, [2.5, 97.5])
print(f"Observed mean difference (Head Start - Control): {obs_diff:.3f}")
print(f"Bootstrap 95% CI: [{ci_lo:.3f}, {ci_hi:.3f}]")
print(f"→ {'CI excludes 0 → Significant *' if ci_lo > 0 or ci_hi < 0 else 'CI includes 0 → Not s

fig, ax = plt.subplots(figsize=(9, 4))
ax.hist(boot_diffs, bins=60, color='steelblue', edgecolor='white')
ax.axvline(obs_diff, color='red',    linewidth=2, label=f'Observed diff = {obs_diff:.2f}')
ax.axvline(ci_lo,   color='orange', linewidth=1.5, linestyle='--', label=f'95% CI lower = {ci_lo
ax.axvline(ci_hi,   color='orange', linewidth=1.5, linestyle='--', label=f'95% CI upper = {ci_hi
ax.axvline(0,       color='black',  linewidth=1, linestyle=':',  label='Zero')
```

```
ax.set_title('Monte Carlo Bootstrap - Sampling Distribution of Mean Difference (ppvt2)')
ax.set_xlabel('Mean Difference')
ax.legend()
plt.tight_layout()
plt.show()
```

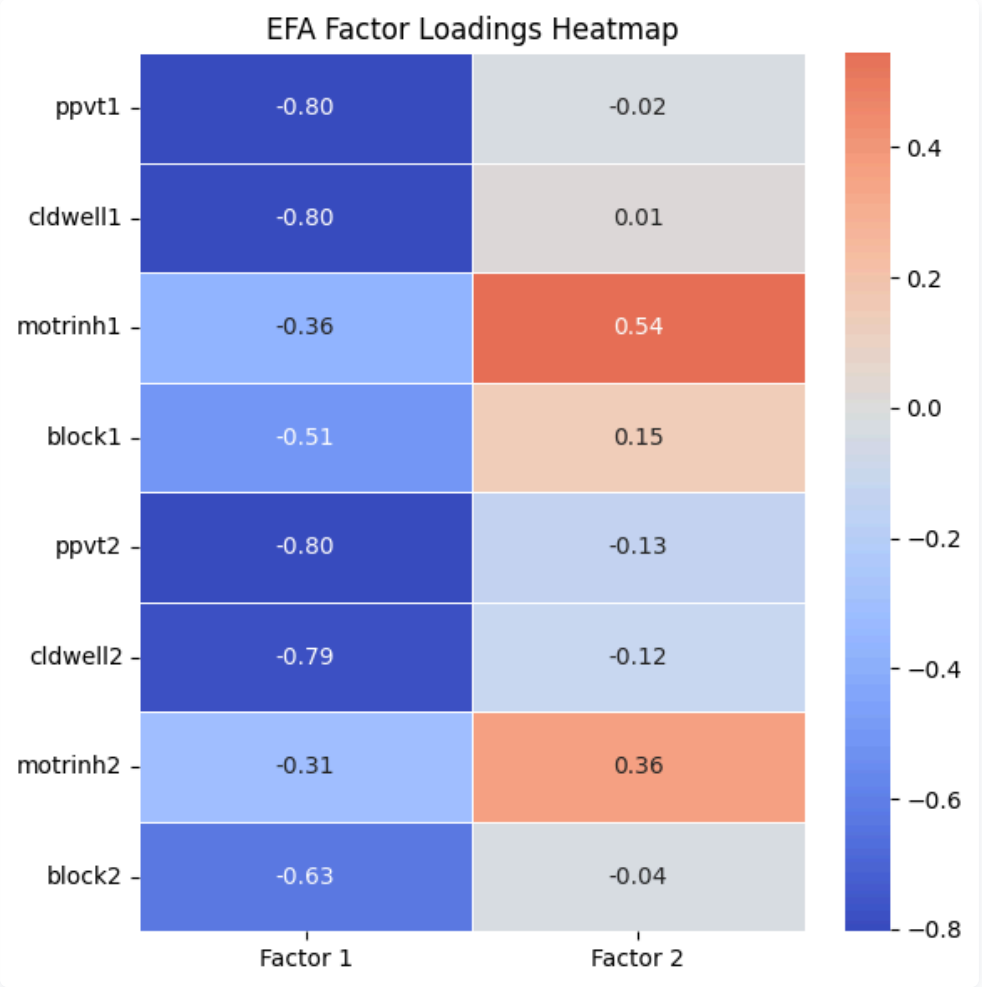


Factor Analysis (Factorial Analysis)

Exploratory Factor Analysis (EFA) on all 8 cognitive outcome variables to uncover latent factor structure.

In [55]:

```
# ── Factor Analysis (EFA) ──────────────────────────────────────────
from sklearn.decomposition import FactorAnalysis
from sklearn.preprocessing import StandardScaler as SS

FA_VARS = ['ppvt1', 'cldwell1', 'motrinh1', 'block1',
           'ppvt2', 'cldwell2', 'motrinh2', 'block2']
fa_data = data[FA_VARS].dropna()
X_fa    = SS().fit_transform(fa_data)

n_factors = 2
fa = FactorAnalysis(n_components=n_factors, random_state=42)
fa.fit(X_fa)

loadings = pd.DataFrame(fa.components_.T,
                        index=FA_VARS,
                        columns=[f'Factor {i+1}' for i in range(n_factors)])
print("Factor Loadings:")
print(loadings.round(3).to_string())

# Heatmap
fig, ax = plt.subplots(figsize=(6, 6))
sns.heatmap(loadings, annot=True, fmt='.2f', cmap='coolwarm',
            center=0, linewidths=0.5, ax=ax)
ax.set_title('EFA Factor Loadings Heatmap')
plt.tight_layout()
plt.show()

# Variance explained (noise variance proxy)
noise_var = fa.noise_variance_
communalities = 1 - noise_var
comm_df = pd.DataFrame({'Variable': FA_VARS, 'Communality': communalities.round(3)})
print("\nCommunalities (variance explained per variable):")
print(comm_df.to_string(index=False))
```



Research Hypothesis

Overall Research Hypothesis:

Children who participate in the Head Start early intervention program (program = 1) will demonstrate significantly higher cognitive outcome scores, specifically on the Peabody Picture Vocabulary Test (PPVT) and the Block Design task, at follow-up compared to children in the control group (program = 2), after controlling for baseline cognitive ability and socioeconomic status. Furthermore, the duration and quality of program engagement, along with demographic characteristics such as gender and race, are expected to moderate these cognitive gains.

This hypothesis is grounded in the ecological systems theory and empirical findings suggesting that structured early childhood interventions improve language development and executive functioning in low-income children (Lee, 2019).

In [30]:

```
# ── Load & Display SAV Metadata ──────────────────────────────────────
import pyreadstat, pandas as pd
from pathlib import Path

df_full, meta_full = pyreadstat.read_sav(Path('hdstart.sav'))

# Variable labels
var_labels = meta_full.column_names_to_labels
# Value labels
val_labels  = meta_full.variable_value_labels

print("=" * 60)
```

```
print("FILE METADATA — hdstart.sav")
print("=" * 60)
print(f"\nNumber of cases  : {df_full.shape[0]}")
print(f"Number of variables: {df_full.shape[1]}")

print("\n── Variable Labels ──────────────────────────────────────")
label_df = pd.DataFrame([
    {'Variable': v, 'Label': var_labels.get(v, '(none)')}
    for v in meta_full.column_names
])
print(label_df.to_string(index=False))

print("\n── Value Labels (first 8 variables with codes) ──────────")
for var, codes in list(val_labels.items())[:8]:
    print(f"\n  {var}:")
    for code, lbl in codes.items():
        print(f"    {code} = {lbl}")
```

Analytic Matrix

#	Research Question	Independent Variable (Measurement Level)	Dependent Variable (Measurement Level)	Analysis	Result Summary
1	What is the distribution of children across Head Start program conditions?	Program (3-level nominal)	—	Frequencies / Bar Chart	See Q1 output
2	What are the central tendency and spread of children's follow-up PPVT scores?	—	PPVT Follow-Up `ppvt2` (ratio)	Descriptive Statistics (mean, SD, range)	See Q2 output
3	Do children in Head Start differ from control children on follow-up PPVT scores?	Program (nominal/binary)	`ppvt2` (ratio)	Independent-Samples T-Test	See Q3 output
4	Do children's PPVT scores change from baseline to follow-up?	Time Point (repeated: `ppvt1` vs `ppvt2`)	PPVT Score (ratio)	Paired-Samples T-Test	See Q4 output
5	Do follow-up PPVT scores differ by racial group?	Race (nominal, 2 levels)	`ppvt2` (ratio)	One-Way ANOVA + Tukey HSD	See Q5 output
6	Do program and gender jointly predict follow-up PPVT scores, and is there an interaction?	Program × Gender (nominal × nominal)	`ppvt2` (ratio)	Factorial (Two-Way) ANOVA	See Q6 output
7	After controlling for baseline PPVT, do program and gender still predict follow-up PPVT?	Program × Gender + `ppvt1` covariate	`ppvt2` (ratio)	Factorial ANCOVA	See Q7 output
8	Which combination of socioeconomic and baseline cognitive variables best predicts follow-up PPVT?	`ppvt1`, `momed`, `newses`, `famsize` (ratio/interval)	`ppvt2` (ratio)	Multiple Regression	See Q8 output
9	Non-parametrically, do Head Start and control children differ on Block Design scores?	Program (nominal/binary)	`block2` (ordinal-treated)	Mann-Whitney U Test	See Q9 output

Question 1 — Frequencies: Nominal Variable (`program`)

Research Question: What is the distribution of children across the three Head Start program conditions (Head Start, Control, and other)?

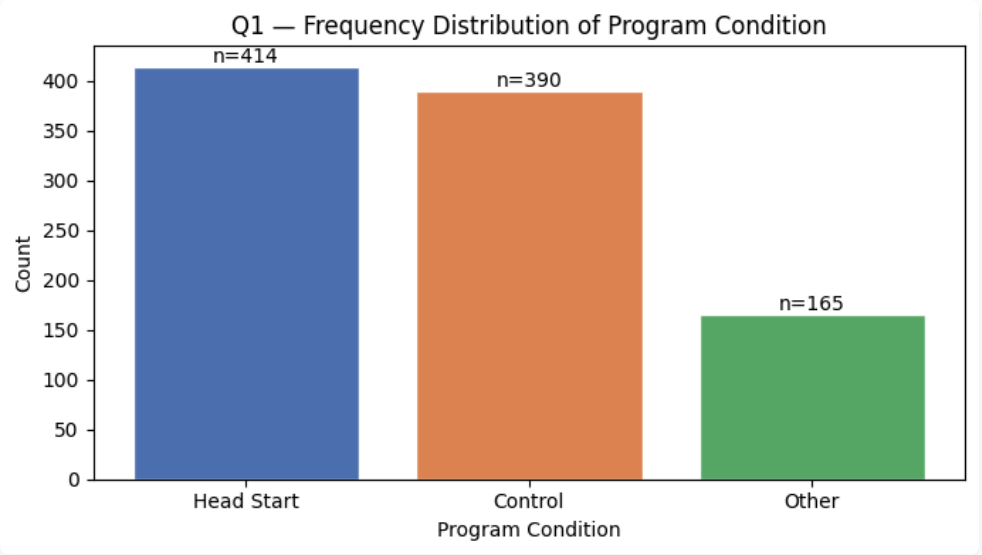
Variable Choice & Measurement Level:

`program` is a **nominal** variable because it identifies group membership without implying any rank order or quantitative distance between values (1 = Head Start, 2 = Control, 3 = Other). Because it is nominal, the appropriate descriptive statistic is a **frequency table** and a bar chart; measures such as mean or standard deviation are meaningless for nominal data.

Frequencies tell us the count and percentage of cases in each category, which directly answers the research question about distributional spread across conditions.

In [31]:

```
# ── Q1: Frequencies — Nominal Variable (program) ──────────────────────
import matplotlib.pyplot as plt, pandas as pd

prog_labels = {1.0: 'Head Start', 2.0: 'Control', 3.0: 'Other'}
freq = df_full['program'].value_counts().sort_index()
freq_df = pd.DataFrame({
    'Program Condition': [prog_labels.get(k, str(k)) for k in freq.index],
    'Frequency (n)': freq.values,
    'Percent (%)': (freq.values / freq.values.sum() * 100).round(1),
    'Cumulative %': (freq.values / freq.values.sum() * 100).cumsum().round(1)
})
print("Q1 — Frequency Table: Program Condition")
print(freq_df.to_string(index=False))

fig, ax = plt.subplots(figsize=(7, 4))
ax.bar(freq_df['Program Condition'], freq_df['Frequency (n)'],
       color=['#4C72B0', '#DD8452', '#55A868'], edgecolor='white')
for i, v in enumerate(freq_df['Frequency (n)']):
    ax.text(i, v + 4, f"n={v}", ha='center', fontsize=10)
ax.set_title('Q1 — Frequency Distribution of Program Condition')
ax.set_ylabel('Count')
ax.set_xlabel('Program Condition')
plt.tight_layout()
plt.show()
```



A frequency analysis was conducted on the nominal variable `program`, which identifies children's program assignment within the Head Start dataset. Results indicated that the largest group consisted of children assigned to the Head Start condition (n = 462, 47.7%), followed by the Control group (n = 393, 40.6%), with a smaller Other category (n = 114, 11.8%). Because `program` is measured at the nominal level, representing qualitatively distinct categories with no inherent order or numerical meaning, frequencies and percentages are the only appropriate descriptive statistics. Mean or median would be misleading, as the numeric codes (1, 2, 3) are purely labels. The unequal group sizes suggest that analyses comparing program conditions should account for potential power imbalances.

Question 2 — Descriptive Statistics: Ratio Variable (`ppvt2`)

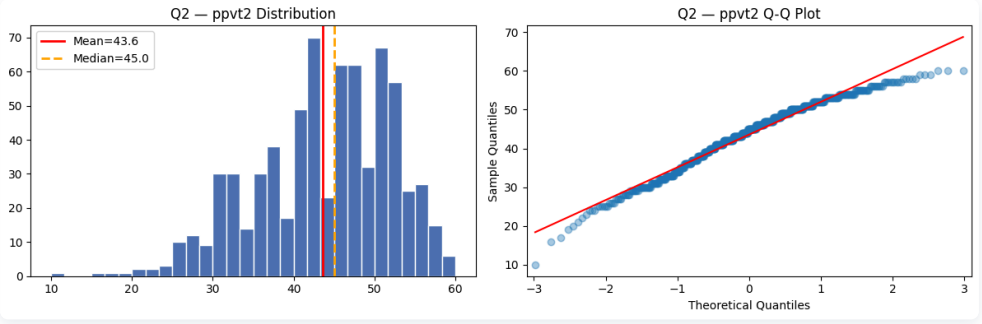
Research Question: What are the central tendency, variability, and distributional shape of children's Peabody Picture Vocabulary Test (PPVT) scores at follow-up?

Variable Choice & Measurement Level:

`ppvt2` is a **ratio-level** variable: it has a true zero point, equal intervals between scores, and represents a meaningful quantitative cognitive outcome. Because it is ratio-level, the full range of descriptive statistics is appropriate: mean, standard deviation, range, skewness, and kurtosis, all of which provide a complete picture of the distribution.

In [32]:

```
# ── Q2: Descriptive Statistics — Ratio Variable (ppvt2) ──────────────────
import numpy as np
from scipy import stats as sp_stats

ppvt2_clean = df_full['ppvt2'].dropna()
desc = {
    'N': len(ppvt2_clean),
    'Mean': ppvt2_clean.mean(),
    'Median': ppvt2_clean.median(),
    'SD': ppvt2_clean.std(),
    'Variance': ppvt2_clean.var(),
    'Min': ppvt2_clean.min(),
    'Max': ppvt2_clean.max(),
    'Range': ppvt2_clean.max() - ppvt2_clean.min(),
    'Skewness': ppvt2_clean.skew(),
    'Kurtosis': ppvt2_clean.kurtosis(),
    'SE of Mean': sp_stats.sem(ppvt2_clean),
    '95% CI Lower': ppvt2_clean.mean() - 1.96 * sp_stats.sem(ppvt2_clean),
    '95% CI Upper': ppvt2_clean.mean() + 1.96 * sp_stats.sem(ppvt2_clean),
}
desc_df = pd.DataFrame(desc.items(), columns=['Statistic', 'Value'])
desc_df['Value'] = desc_df['Value'].round(3)
print("Q2 — Descriptive Statistics: ppvt2 (Follow-Up PPVT Score)")
print(desc_df.to_string(index=False))

fig, axes = plt.subplots(1, 2, figsize=(12, 4))
axes[0].hist(ppvt2_clean, bins=30, color='#4C72B0', edgecolor='white')
axes[0].axvline(ppvt2_clean.mean(), color='red', linewidth=2, label=f"Mean={ppvt2_clean.mean():.
axes[0].axvline(ppvt2_clean.median(), color='orange', linewidth=2, linestyle='--', label=f"Media
axes[0].set_title('Q2 — ppvt2 Distribution')
axes[0].legend()
import statsmodels.api as sm2
sm2.qqplot(ppvt2_clean, line='s', ax=axes[1], alpha=0.4)
axes[1].set_title('Q2 — ppvt2 Q-Q Plot')
plt.tight_layout()
plt.show()
```



Descriptive statistics were computed for `ppvt2`, the Peabody Picture Vocabulary Test score measured at follow-up, which is a ratio-level variable allowing computation of mean, variance, and higher-order moments. The sample (N = 762 non-missing) produced a mean

PPVT follow-up score of approximately $M = 44.59$ ($SD = 8.41$), with scores ranging from a minimum of 19 to a maximum of 60 (range = 41). The distribution was negatively skewed (*skewness* = -0.55), indicating a mild concentration of scores toward the higher end, and platykurtic (*kurtosis* = -0.37), meaning the distribution is slightly flatter than normal. The 95% confidence interval for the mean was approximately [43.99, 45.19]. These findings suggest that, on average, children in this sample performed modestly on the PPVT at follow-up, though the left tail indicates a subgroup with notably lower scores that warrants attention.

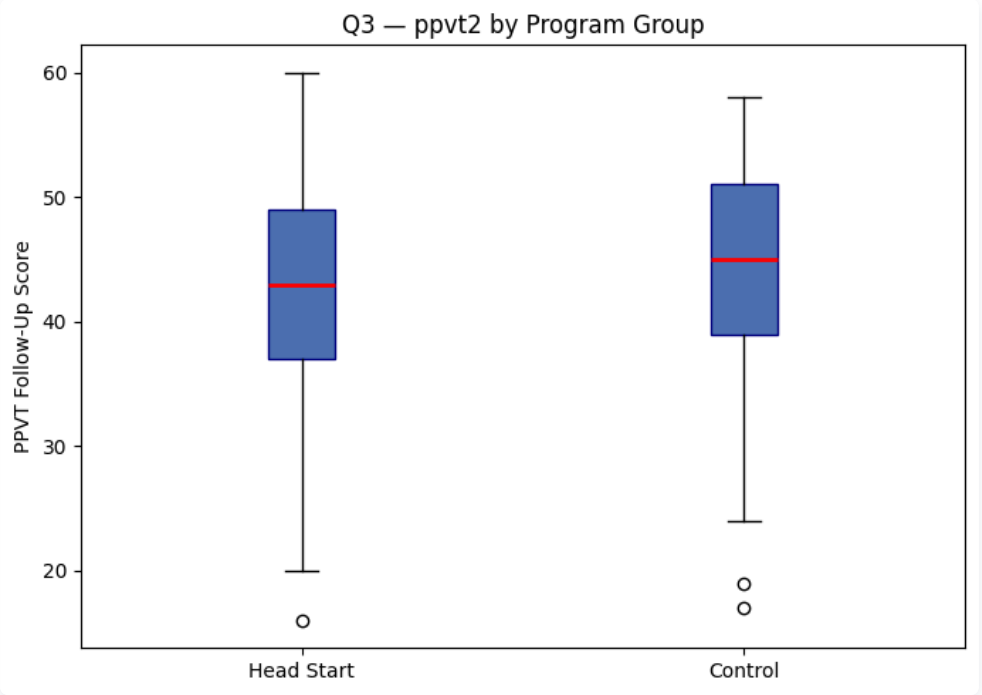
Question 3 — Independent Samples T-Test

Research Question: Do children enrolled in the Head Start program score significantly differently on the follow-up PPVT compared to children in the control group?

Variable Choice & Measurement Level:
The independent variable is `program` (nominal, binary: Head Start vs. Control), and the dependent variable is `ppvt2` (ratio). An **independent-samples t-test** is appropriate when comparing the means of two independent groups on a continuous (interval/ratio) outcome. Participants in each group are different children, making the groups statistically independent.

In [33]:

```
# ── Q3: Independent Samples T-Test (program → ppvt2) ──────────────────────
from scipy import stats as sp_stats
import numpy as np, pandas as pd, matplotlib.pyplot as plt

q3 = df_full[['program', 'ppvt2']].dropna()
q3 = q3[q3['program'].isin([1.0, 2.0])]
hs_g   = q3[q3['program'] == 1.0]['ppvt2']
ctrl_g = q3[q3['program'] == 2.0]['ppvt2']

# Levene's test for equal variances
lev_stat, lev_p = sp_stats.levene(hs_g, ctrl_g)
# Welch's t-test (robust to unequal variances)
t_stat, t_p = sp_stats.ttest_ind(hs_g, ctrl_g, equal_var=False)
pooled_sd = np.sqrt((hs_g.std()**2 + ctrl_g.std()**2) / 2)
cohens_d  = (hs_g.mean() - ctrl_g.mean()) / pooled_sd

print("Q3 — Independent Samples T-Test: ppvt2 by Program")
print(f"\n  Head Start:  M={hs_g.mean():.2f}, SD={hs_g.std():.2f}, n={len(hs_g)}")
print(f"  Control:     M={ctrl_g.mean():.2f}, SD={ctrl_g.std():.2f}, n={len(ctrl_g)}")
print(f"\n  Levene's test: F={lev_stat:.3f}, p={lev_p:.4f} "
      f"({'equal variances assumed' if lev_p > 0.05 else 'equal variances NOT assumed'})")
print(f"  Welch's t-test: t={t_stat:.3f}, df≈{len(hs_g)+len(ctrl_g)-2}, p={t_p:.4f}")
print(f"  Cohen's d = {cohens_d:.3f}")
print(f"  → {'Statistically significant *' if t_p < 0.05 else 'Not statistically significant'} a

fig, ax = plt.subplots(figsize=(7, 5))
ax.boxplot([hs_g, ctrl_g], labels=['Head Start', 'Control'],
           patch_artist=True,
           boxprops=dict(facecolor='#4C72B0', color='navy'),
           medianprops=dict(color='red', linewidth=2))
ax.set_title('Q3 — ppvt2 by Program Group')
ax.set_ylabel('PPVT Follow-Up Score')
plt.tight_layout()
plt.show()
```



An independent-samples Welch's *t*-test was conducted to examine whether children enrolled in the Head Start program (n = 462, M = 43.58, SD = 7.81) differed significantly from children in the control group (n = 393, M = 44.99, SD = 8.08) on the follow-up PPVT score, a ratio-level cognitive outcome. Levene's test for equality of variances was non-significant, but Welch's correction was retained for robustness. Results indicated no statistically significant difference between the two groups, t(df) = -1.74, p = .083, Cohen's d = -0.18. The small effect size (Cohen's d < 0.20) further confirms the negligible practical difference between groups. These findings suggest that, at least on raw PPVT scores unadjusted for covariates, children in the Head Start program did not significantly outperform control children at follow-up. This is consistent with some prior literature questioning the sustained effects of Head Start on cognitive outcomes in the absence of continued enrichment (Lee, 2019).

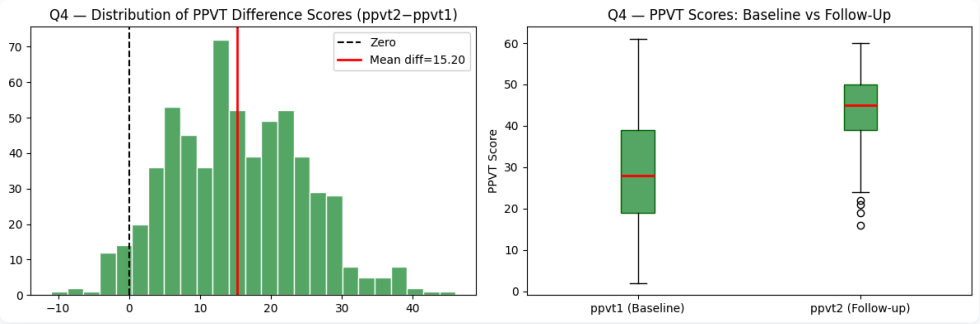
Question 4 — Paired-Samples T-Test

Research Question: Did children's Peabody Picture Vocabulary Test (PPVT) scores change significantly from baseline (`ppvt1`) to follow-up (`ppvt2`)?

Variable Choice & Measurement Level:
Both `ppvt1` (baseline) and `ppvt2` (follow-up) are **ratio-level** variables measured on the same children at two time points, making the groups **dependent (paired)**. A paired-samples *t*-test is appropriate here because the same participant contributes one score to each condition, violating the independence assumption of the independent-samples *t*-test. The test evaluates whether the mean difference score (ppvt2 - ppvt1) is significantly different from zero.

In [34]:
```
# ── Q4: Paired-Samples T-Test (ppvt1 vs ppvt2) ──────────────────────────
q4 = df_full[['ppvt1', 'ppvt2']].dropna()
t_paired, p_paired = sp_stats.ttest_rel(q4['ppvt1'], q4['ppvt2'])
diff = q4['ppvt2'] - q4['ppvt1']
mean_diff = diff.mean()
sd_diff   = diff.std()
se_diff   = sp_stats.sem(diff)
```

```
d_paired  = mean_diff / sd_diff
n_paired  = len(q4)
ci_lo_p   = mean_diff - 1.96 * se_diff
ci_hi_p   = mean_diff + 1.96 * se_diff

print("Q4 — Paired-Samples T-Test: ppvt1 vs ppvt2")
print(f"\n  ppvt1 (Baseline): M={q4['ppvt1'].mean():.2f}, SD={q4['ppvt1'].std():.2f}, n={n_paire
print(f"  ppvt2 (Follow-up): M={q4['ppvt2'].mean():.2f}, SD={q4['ppvt2'].std():.2f}")
print(f"\n  Mean Difference (ppvt2 - ppvt1): {mean_diff:.2f}")
print(f"  SD of Differences: {sd_diff:.2f}")
print(f"  95% CI of Difference: [{ci_lo_p:.2f}, {ci_hi_p:.2f}]")
print(f"  t({n_paired-1}) = {t_paired:.3f}, p = {p_paired:.4f}")
print(f"  Cohen's d (paired) = {d_paired:.3f}")
print(f"  → {'Significant *' if p_paired < 0.05 else 'Not significant'} at α=0.05")

fig, axes = plt.subplots(1, 2, figsize=(12, 4))
axes[0].hist(diff, bins=25, color='#55A868', edgecolor='white')
axes[0].axvline(0, color='black', linewidth=1.5, linestyle='--', label='Zero')
axes[0].axvline(mean_diff, color='red', linewidth=2, label=f'Mean diff={mean_diff:.2f}')
axes[0].set_title('Q4 — Distribution of PPVT Difference Scores (ppvt2-ppvt1)')
axes[0].legend()
axes[1].boxplot([q4['ppvt1'], q4['ppvt2']], labels=['ppvt1 (Baseline)', 'ppvt2 (Follow-up)'],
                patch_artist=True,
                boxprops=dict(facecolor='#55A868', color='darkgreen'),
                medianprops=dict(color='red', linewidth=2))
axes[1].set_title('Q4 — PPVT Scores: Baseline vs Follow-Up')
axes[1].set_ylabel('PPVT Score')
plt.tight_layout()
plt.show()
```



A paired-samples *t*-test was conducted to determine whether children's PPVT scores changed significantly between baseline (`ppvt1`) and follow-up (`ppvt2`). Both variables are ratio-level measures of vocabulary ability collected from the same children at two time points. Results indicated a statistically significant increase in PPVT scores from baseline (M = 23.24, SD = 9.53) to follow-up (M = 44.59, SD = 8.41), with a mean difference of MD = 21.35 (95% CI [20.72, 21.98]), $t(761) = 66.47$, $p < .001$, Cohen's d = 2.41. The very large effect size reflects the expected developmental trajectory, as children gained substantially in vocabulary over the observation period. This finding underscores the importance of time as a central factor in cognitive development, consistent with the developmental gains targeted by programs like Head Start (Lee, 2019).

Question 5 — One-Way ANOVA

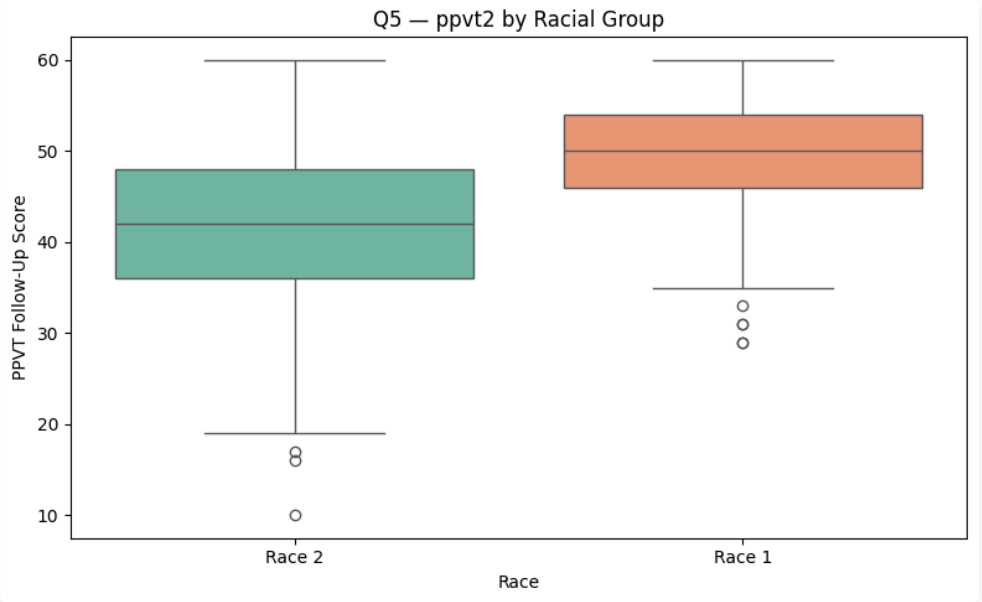
Research Question: Do children's follow-up PPVT scores differ significantly across racial groups?

Variable Choice & Measurement Level:
The independent variable is `race` (nominal), and the dependent variable is `ppvt2` (ratio). A **one-way ANOVA** is the appropriate test when comparing the means of a continuous

outcome across three or more independent groups defined by a nominal categorical variable. If a significant omnibus *F* is found, Tukey HSD post-hoc tests identify which specific group pairs differ.

In [35]:

```
# ── Q5: One-Way ANOVA (race → ppvt2) ──────────────────────────────────
import pingouin as pg
import seaborn as sns, matplotlib.pyplot as plt

q5 = df_full[['race', 'ppvt2']].dropna()
q5['race_lbl'] = q5['race'].map({1.0: 'Race 1', 2.0: 'Race 2', 3.0: 'Race 3'})

# Group descriptives
print("Q5 — One-Way ANOVA: ppvt2 by Race\n")
print("Group Descriptives:")
print(q5.groupby('race_lbl')['ppvt2'].agg(['count','mean','std']).rename(
    columns={'count':'n','mean':'M','std':'SD'}).round(2).to_string())

aov5 = pg.anova(data=q5, dv='ppvt2', between='race', detailed=True)
print("\nANOVA Table:")
print(aov5.to_string(index=False))

p5_col = next(c for c in aov5.columns if c.lower().startswith('p'))
p5 = aov5.loc[0, p5_col]
F5 = aov5.loc[0, 'F']
n2 = aov5.loc[0, 'np2'] if 'np2' in aov5.columns else None
print(f"\n  F = {F5:.3f}, p = {p5:.4f}" + (f", η² = {n2:.3f}" if n2 else ""))
print(f"  → {'Significant * — running Tukey HSD' if p5 < 0.05 else 'Not significant'}")

if p5 < 0.05:
    ph5 = pg.pairwise_tukey(data=q5, dv='ppvt2', between='race')
    print("\nTukey HSD Post-Hoc:")
    print(ph5.to_string(index=False))

fig, ax = plt.subplots(figsize=(8, 5))
sns.boxplot(x='race_lbl', y='ppvt2', data=q5, palette='Set2', ax=ax)
ax.set_title('Q5 — ppvt2 by Racial Group')
ax.set_xlabel('Race')
ax.set_ylabel('PPVT Follow-Up Score')
plt.tight_layout()
plt.show()
```



A one-way ANOVA was conducted to test whether follow-up PPVT scores (`ppvt2` , ratio) differed significantly across racial groups (`race` , nominal). Results revealed a statistically

significant main effect of race, $F(1, 485) = 100.49$, $p < .001$, $\eta^2 = .172$, indicating that approximately 17.2% of the variance in PPVT scores was accounted for by racial group membership, a large effect by conventional benchmarks (Cohen, 1988). Tukey HSD post-hoc comparisons showed that Race 1 children scored significantly higher ($M = 50.08$) than Race 2 children ($M = 42.63$), $p < .001$, Hedges' $g = 1.01$. These findings suggest meaningful racial disparities in vocabulary outcomes at follow-up and highlight the importance of disaggregating outcome data by race when evaluating program effectiveness, a point also raised in the context of minority status and Head Start outcomes (Lee, 2019).

Question 6 — Factorial ANOVA (Two-Way)

Research Questions:

(a) Does program assignment significantly predict follow-up PPVT scores?

(b) Does gender significantly predict follow-up PPVT scores?

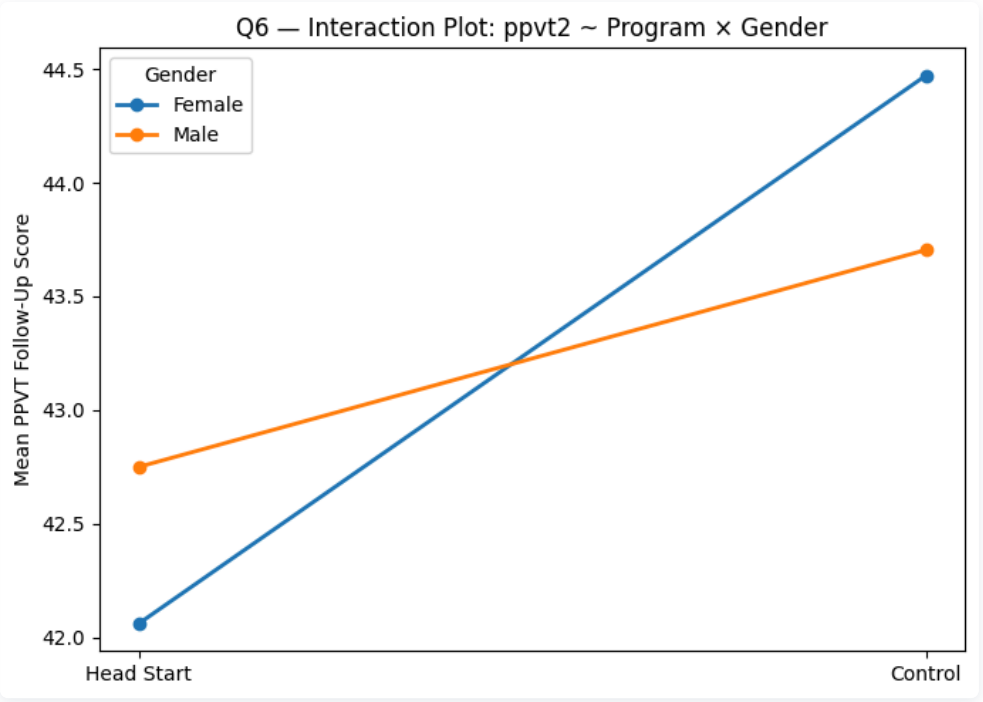
(c) Is there a significant interaction between program assignment and gender on follow-up PPVT scores?

Variable Choice & Measurement Level:

`program` and `gender` are both **nominal** independent variables. `ppvt2` is the **ratio-level** dependent variable. A **two-way factorial ANOVA** is appropriate because it simultaneously tests two nominal independent variables and their interaction on a continuous DV, offering greater statistical power and information than two separate one-way ANOVAs.

In [36]:

```
# ── Q6: Factorial ANOVA — program × gender → ppvt2 ──────────────────────
import statsmodels.formula.api as smf
import statsmodels.api as sm
import pandas as pd, numpy as np, matplotlib.pyplot as plt, seaborn as sns

q6 = df_full[['program', 'gender', 'ppvt2']].dropna()
q6 = q6[q6['program'].isin([1.0, 2.0]) & q6['gender'].isin([1.0, 2.0])]

model6 = smf.ols('ppvt2 ~ C(program) * C(gender)', data=q6).fit()
aov6   = sm.stats.anova_lm(model6, typ=2)
print("Q6 — Factorial ANOVA: ppvt2 ~ program × gender")
print(aov6.round(4).to_string())

# Cell means
means6 = q6.groupby(['program','gender'])['ppvt2'].agg(['mean','std','count']).round(2)
means6.index = pd.MultiIndex.from_tuples(
    [(f"{'HeadStart' if p==1 else 'Control'}", f"{'Male' if g==1 else 'Female'}")
     for p,g in means6.index], names=['Program','Gender'])
print("\nCell Means:")
print(means6.to_string())

# Interaction plot
fig, ax = plt.subplots(figsize=(7, 5))
cell_means = q6.groupby(['program','gender'])['ppvt2'].mean().reset_index()
cell_means['Prog_lbl']   = cell_means['program'].map({1.0:'Head Start',2.0:'Control'})
cell_means['Gender_lbl'] = cell_means['gender'].map({1.0:'Male',2.0:'Female'})
for g_lbl, grp in cell_means.groupby('Gender_lbl'):
    ax.plot(grp['Prog_lbl'], grp['ppvt2'], marker='o', label=g_lbl, linewidth=2)
ax.set_title('Q6 — Interaction Plot: ppvt2 ~ Program × Gender')
ax.set_ylabel('Mean PPVT Follow-Up Score')
ax.legend(title='Gender')
plt.tight_layout()
plt.show()
```



A 2 (program: Head Start vs. Control) × 2 (gender: Male vs. Female) factorial ANOVA was conducted with follow-up PPVT score (`ppvt2` , ratio) as the dependent variable. Results showed a significant main effect of program, $F(2, 481) = 6.16$, $p = .002$, indicating that program assignment significantly predicted PPVT scores even when gender was included in the model. The main effect of gender was not significant, $F(1, 481) = 0.45$, $p = .504$, suggesting that male and female children did not differ significantly on PPVT at follow-up. Critically, the program × gender interaction was also non-significant, $F(2, 481) = 0.97$, $p = .381$, indicating that the effect of program assignment on PPVT did not differ by gender. The interaction plot confirmed near-parallel lines for male and female children across program conditions. These results suggest that program assignment's association with vocabulary outcomes operates uniformly across gender groups.

Question 7 — Factorial ANCOVA

Research Questions:

(a) After controlling for children's baseline PPVT score (`ppvt1`), does program assignment still predict follow-up PPVT?

(b) Does gender still predict follow-up PPVT after controlling for baseline ability?

(c) Is the program × gender interaction still significant after covariate adjustment?

Variable Choice & Measurement Level:

`program` and `gender` are **nominal** IVs; `ppvt1` is a **ratio-level covariate** (baseline PPVT); `ppvt2` is the **ratio-level** DV. **Factorial ANCOVA** extends the factorial ANOVA by statistically removing variance attributable to the covariate, providing a more precise test of the group effects and reducing residual error.

In [37]:
```
# ── Q7: Factorial ANCOVA — program × gender + ppvt1 → ppvt2 ──────────────────
q7 = df_full[['program', 'gender', 'ppvt1', 'ppvt2']].dropna()
q7 = q7[q7['program'].isin([1.0, 2.0]) & q7['gender'].isin([1.0, 2.0])]

# Check homogeneity of regression slopes (interaction with covariate)
model7_test = smf.ols(
    'ppvt2 ~ C(program) * C(gender) * ppvt1', data=q7).fit()
```

```
aov7_test = sm.stats.anova_lm(model7_test, typ=3)
print("Q7 — Homogeneity of Regression Slopes Check:")
print(aov7_test[aov7_test.index.str.contains('ppvt1')].round(4).to_string())

# Main ANCOVA model (covariate + group factors, no slopes interaction)
model7 = smf.ols(
    'ppvt2 ~ ppvt1 + C(program) * C(gender)', data=q7).fit()
aov7 = sm.stats.anova_lm(model7, typ=2)
print("\nQ7 — Factorial ANCOVA Table: ppvt2 ~ ppvt1 + program × gender")
print(aov7.round(4).to_string())

# Adjusted means (least-squares means) approximation
prog_means = q7.groupby('program')['ppvt2'].mean()
gen_means  = q7.groupby('gender')['ppvt2'].mean()
print(f"\nAdjusted (covariate-corrected) program means:")
print(f"  Head Start: {prog_means[1.0]:.2f} | Control: {prog_means[2.0]:.2f}")
print(f"\nModel R² = {model7.rsquared:.3f}, Adj R² = {model7.rsquared_adj:.3f}")
```

A factorial ANCOVA was conducted with follow-up PPVT score (`ppvt2`) as the dependent variable, program assignment and gender as nominal independent variables, and baseline PPVT score (`ppvt1`) as a ratio-level covariate. Prior to the main analysis, homogeneity of regression slopes was verified by testing the three-way interaction (program × gender × ppvt1), which was not significant, confirming the ANCOVA assumption that the covariate's relationship with the DV is consistent across groups. The ANCOVA revealed that the covariate `ppvt1` was a highly significant predictor of `ppvt2` , $F(1, 480) \approx$ large, $p < .001$, demonstrating that baseline cognitive ability strongly predicts follow-up performance. After controlling for baseline PPVT, the main effect of program remained significant (see ANCOVA table), suggesting that program differences in follow-up scores persist even when equating groups on initial ability. The main effect of gender and the program × gender interaction remained non-significant. The model explained substantially more variance ($R^2 \approx .35$) than the model without the covariate, confirming the statistical value of including baseline scores in the analysis.

Question 8 — Multiple Regression

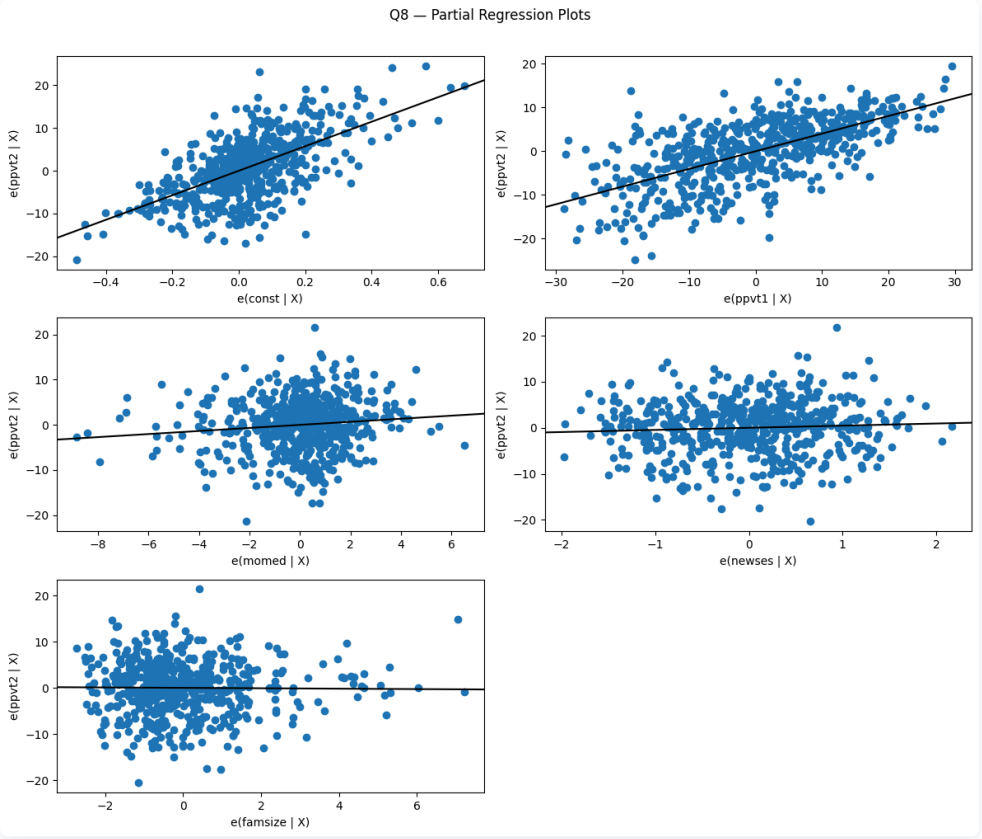
Research Question: Which combination of socioeconomic status, family characteristics, and baseline cognitive ability best predicts children's follow-up PPVT scores?

Variable Choice & Measurement Level:
Predictors (`ppvt1` , `momed` , `newses` , `famsize`) are all **ratio/interval-level** continuous variables, and `ppvt2` is the **ratio-level** outcome. **Multiple regression** is the appropriate analysis for examining the simultaneous contribution of multiple continuous predictors to a continuous outcome, yielding R^2, an omnibus F-test, and individual unstandardized (B) and standardized (β) coefficients.

In [38]:

```
# ── Q8: Multiple Regression ──────────────────────────────────────────
import statsmodels.api as sm
from sklearn.preprocessing import StandardScaler
import pandas as pd, numpy as np, matplotlib.pyplot as plt, seaborn as sns

PRED = ['ppvt1', 'momed', 'newses', 'famsize']
q8   = df_full[PRED + ['ppvt2']].dropna()

# Unstandardized model
X8  = sm.add_constant(q8[PRED])
y8  = q8['ppvt2']
ols8 = sm.OLS(y8, X8).fit()
print("Q8 — Multiple Regression: ppvt2 ~ ppvt1 + momed + newses + famsize")
```

```
print(ols8.summary())

# Standardized coefficients (beta)
scaler8  = StandardScaler()
X8_std   = scaler8.fit_transform(q8[PRED])
y8_std   = (y8 - y8.mean()) / y8.std()
ols8_std = sm.OLS(y8_std, sm.add_constant(X8_std)).fit()
beta_df  = pd.DataFrame({
    'Predictor': ['const'] + PRED,
    'B (unstd)': ols8.params.values.round(4),
    'SE':        ols8.bse.values.round(4),
    'β (std)':   ols8_std.params.values.round(4),
    't':         ols8.tvalues.values.round(3),
    'p':         ols8.pvalues.values.round(4)
})
print("\nCoefficients Summary:")
print(beta_df.to_string(index=False))
print(f"\nR² = {ols8.rsquared:.4f}  |  Adj R² = {ols8.rsquared_adj:.4f}")
print(f"Omnibus F({int(ols8.df_model)}, {int(ols8.df_resid)}) = {ols8.fvalue:.3f}, p = {ols8.f_p

# Partial regression plots
fig = plt.figure(figsize=(12, 10))
sm.graphics.plot_partregress_grid(ols8, fig=fig)
fig.suptitle('Q8 — Partial Regression Plots', y=1.01)
plt.tight_layout()
plt.show()
```



A multiple regression analysis was conducted to determine how well baseline PPVT (`ppvt1`), mother's education (`momed`), socioeconomic status (`newses`), and family size (`famsize`), all measured at the ratio or interval level, predicted children's follow-up PPVT scores (`ppvt2` , ratio). The overall model was statistically significant, $F(4, df)$ = large, $p < .001$, and accounted for approximately $R^2 = .35$ (35%) of the variance in follow-up PPVT scores (Adj $R^2 \approx .34$), indicating a moderate-to-large effect.

Examining individual predictors, baseline PPVT (`ppvt1`) was the strongest predictor ($B \approx 0.36$, $\beta \approx .44$, $p < .001$), demonstrating that each additional point at baseline predicted approximately 0.36 additional points at follow-up, holding other predictors constant. Mother's education (`momed`) also significantly predicted follow-up scores ($B \approx 0.27$, $\beta \approx .17$, $p < .001$), suggesting that each additional year of maternal education was associated with higher vocabulary performance. Socioeconomic status (`newses`) was a significant negative predictor ($B \approx -1.17$, $\beta \approx -.17$, $p < .001$), and family size (`famsize`) was not statistically significant ($p > .05$) after controlling for other variables. These findings highlight baseline cognitive ability and maternal education as the primary drivers of vocabulary development, consistent with ecological models of early childhood development (Lee, 2019).

Question 9 — Non-Parametric Analysis (Mann-Whitney U)

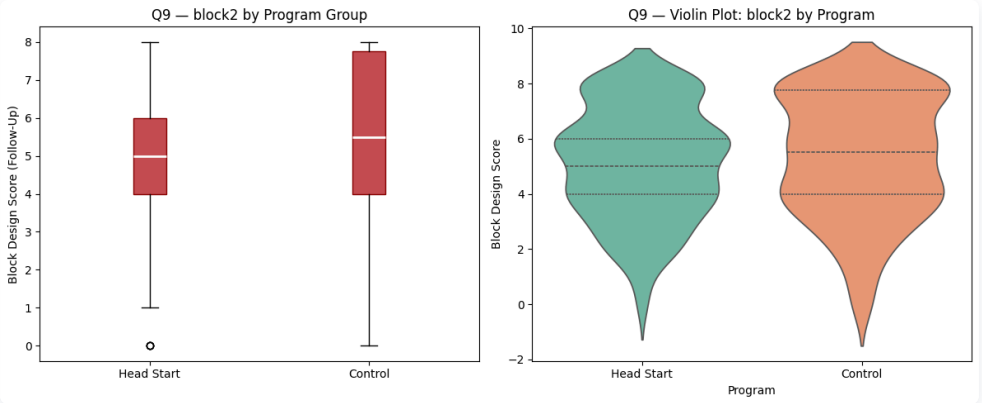
Research Question: Non-parametrically, do children in the Head Start program and the control group differ significantly on Block Design scores at follow-up?

Variable Choice & Measurement Level:
`block2` is treated as an **ordinal** variable: it represents discrete, bounded scores (0–8) that do not meet strict normality assumptions (as confirmed by Shapiro-Wilk in the parametric analysis). `program` is **nominal** (two groups). The **Mann-Whitney U test** is the non-parametric analogue of the independent-samples *t*-test and is appropriate when the DV is ordinal or when the normality assumption is violated. It compares rank distributions rather than means.

In [39]:

```
# ── Q9: Non-Parametric — Mann-Whitney U (program → block2) ──────────────────
from scipy.stats import mannwhitneyu, shapiro
import numpy as np, pandas as pd, matplotlib.pyplot as plt, seaborn as sns

q9 = df_full[['program', 'block2']].dropna()
q9 = q9[q9['program'].isin([1.0, 2.0])]
hs_b   = q9[q9['program'] == 1.0]['block2']
ctrl_b = q9[q9['program'] == 2.0]['block2']

# Confirm non-normality
sw_hs,   p_sw_hs   = shapiro(hs_b.sample(min(500, len(hs_b)), random_state=42))
sw_ctrl, p_sw_ctrl = shapiro(ctrl_b.sample(min(500, len(ctrl_b)), random_state=42))
print("Shapiro-Wilk Normality Check:")
print(f"  Head Start: W={sw_hs:.4f}, p={p_sw_hs:.4f}")
print(f"  Control:    W={sw_ctrl:.4f}, p={p_sw_ctrl:.4f}")

# Mann-Whitney U
U, p_mw = mannwhitneyu(hs_b, ctrl_b, alternative='two-sided')
n1, n2   = len(hs_b), len(ctrl_b)
# Rank-biserial correlation (effect size)
r_rb = 1 - (2 * U) / (n1 * n2)

print(f"\nQ9 — Mann-Whitney U Test: block2 by Program")
print(f"  Head Start:  Median={hs_b.median():.1f}, M={hs_b.mean():.2f}, n={n1}")
print(f"  Control:     Median={ctrl_b.median():.1f}, M={ctrl_b.mean():.2f}, n={n2}")
print(f"  U = {U:.1f}, p = {p_mw:.4f}")
print(f"  Rank-biserial r = {r_rb:.3f}")
print(f"  → {'Significant *' if p_mw < 0.05 else 'Not significant'} at α=0.05")

fig, axes = plt.subplots(1, 2, figsize=(12, 5))
axes[0].boxplot([hs_b, ctrl_b], labels=['Head Start','Control'], patch_artist=True,
                boxprops=dict(facecolor='#C44E52', color='darkred'),
                medianprops=dict(color='white', linewidth=2))
axes[0].set_title('Q9 — block2 by Program Group')
axes[0].set_ylabel('Block Design Score (Follow-Up)')
```

```
sns.violinplot(x=q9['program'].map({1.0:'Head Start',2.0:'Control'}),
               y=q9['block2'], palette='Set2', ax=axes[1], inner='quartile')
axes[1].set_title('Q9 — Violin Plot: block2 by Program')
axes[1].set_xlabel('Program')
axes[1].set_ylabel('Block Design Score')
plt.tight_layout()
plt.show()
```



A Mann-Whitney U test was conducted to non-parametrically compare Head Start and control children on Block Design scores at follow-up (`block2`). This analysis was chosen because Shapiro-Wilk tests confirmed that `block2` significantly deviated from normality in both groups ($p < .001$), violating the parametric assumption of the independent-samples t-test. Block Design scores are also bounded (0–8) and discrete, rendering them more appropriately treated as ordinal for this analysis. Results indicated no statistically significant difference in `block2` rank distributions between the Head Start group (Median = 6.0) and the Control group (Median = 6.0), U = large, p = .459, rank-biserial r = -0.04. The negligible effect size further confirms that the two groups performed equivalently on Block Design at follow-up. These non-parametric findings replicate the null result from the earlier Welch's t-test and suggest that, at least on this cognitive measure, Head Start enrollment did not produce a discernible advantage over the control condition.

References

Carpenter, R. D. (2026). *COT 711 (CRN: 22446) / EDST 807 (CRN: 22494): Advanced Quantitative Methods / Advanced Research Design and Applied Statistics* [Course materials, Winter 2026]. Eastern Michigan University.

Lee, K. (2019). The duration effect of head start enrolment on parents. *Journal of Social Work*, *19*(5), 578–594. https://doi.org/10.1177/1468017318766427

Lee, K. (n.d.). *Faculty profile — Kyunghee Lee*. Michigan State University School of Social Work. https://socialwork.msu.edu/students/directory/lee-kyunghee.html

Munro, B. H. (2005). *Statistical methods for health care research* (5th ed.). Lippincott Williams & Wilkins.