Spoken Language Diversity Across Washington Counties

Arona Cho (aronacho@uw.edu), Arpan Kapoor (akap1204@uw.edu)

CSE 163 - Intermediate Data Programming | Data Science Fair Project Report & Code | University of Washington | Github Repository

---

Summary

Research Question 1: How does language diversity in each Washington county affect their educational attainment outcomes?

Language diversity was not found to have as large of an effect on educational attainment across Washington counties as we thought it might have. In fact, for certain non-English language speaking groups, it can be argued that there is an inverse relationship, where counties with more language speakers see lower rates of educational attainment. It seems that the results for each language speaking group is either the previously mentioned case, or a very slight correlation between increasing language diversity and educational attainment.

Research Question 2: How do school district budgets and spending affect spoken language diversity in Washington counties?

From our analysis, school district budgets and spending do not seem to have a strong effect on spoken language diversity in WA counties, nor the inverse.

Research Question 3: To what extent do Washington residents continue to practice their ancestral culture through language?

Generally, we have found that Washington residents with a diverse ancestral background do not have a constant relationship with the amount of ancestral language that is spoken. Rather, in counties with more ancestrally diverse populations, the rate of residents that speak only English increases as well.

Motivation

Spoken language diversity in our communities directly affects the extent to which various cultures can communicate and interact with each other. Identifying the underlying reasons behind why certain Washington counties have larger ranges of spoken languages can allow for the expansion of language diversity in other counties lacking in this aspect. This helps foster closer, more personal, communities. We look at two vital areas in Washington state's primary communities: the academia space and family households. Local governments can use these conclusions to increase efforts and redirect resources in order to increase spoken language diversity in their respective territories.

Data Setting

For this project, we retrieved the majority of our datasets from the United States Census, focusing on data from all 39 Washington counties of the most recent year that was on record, with the majority of our data coming from 2022. We used a total of five datasets for this research project, four of which came from the US Census. The one we did not retrieve from the US Census was the dataset containing per pupil expenditure (PPE) values for all school districts in Washington, which was retrieved from the Official Washington State Open Data Portal. For this single dataset, we used the data.wa.gov website's built in aggregation tools to filter out unnecessary data points that were not relevant for our project. This was a solution to the initial issue of the dataset's size being too large, and also made it easier for us to work with the data once it had been imported. After aggregation, we were left with data for school district names and their PPEs. We used the datasets from the US Census to merge on matching school districts in order to find which counties each district is in, allowing us to find the average PPE value for each Washington county. These datasets allowed us to examine collections of county data in a more granular manner as we were able to merge them with each other in order to create plots showing trends between the shared data values.

As the majority of this data is taken from the US Census Bureau, the population which is encapsulated is all United States residents, or all Washington State residents in our case, at the time of collection. Much of the data is respondent, meaning that the data is prone to response bias. Overall, the census data is vulnerable to response biases because of the participants succumbing to social desirability which may skew the data to support more socially acceptable answers. Despite this, we believe that some solid takeaways can be formed using the resulting graphs and data plots, and could be further supported or investigated by future research and analysis in this area.

The titles, links, and sources for each dataset used in this research project are listed below:

- School Enrollment (S1401) - US Census: https://data.census.gov/table/ACSST5Y2022.S1401?q=s1401&g=050XX00US53001
- Language Spoken at Home (S1601) - US Census: https://data.census.gov/table/ACSST5Y2022.S1601?q=s1601&g=050XX00US53001
- Selected Social Characteristics in the United States (DP02) - US Census: https://data.census.gov/table/ACSDP5Y2022.DP02?q=Dp02&g=050XX00US53001
- School Districts and Associated Counties - US Census: https://www.census.gov/programs-surveys/saipe/guidance-geographies/districts-counties.html
- Per Pupil Expenditures All Years - Washington State Open Data Portal: https://data.wa.gov/education/Per-Pupil-Expenditure_AllYears/vnm3-j8pe/about_data

---

Method

---

1. Retrieve the four datasets listed in the Data Setting section containing values for all 39 Washington counties from the US Census website.
2. Retrieve the dataset containing the per pupil expenditure information for all school districts in Washington from the data.wa.gov website.
3. Use a github repository to upload all collected datasets in order to store them in an organized, shareable manner.
4. Use VSCode, utilizing the Live Share, Github Codespaces, and Jupyter extensions in order to collaboratively work on the code and datasets for this project.
5. Import the datasets from the repository using Github Codespaces and create dataframes for each of them.
6. Clean each dataset, removing any unneeded data values and reformatting any irregular indexes, columns/column names, and data values.
7. Merge datasets to have the necessary information within the same dataframe, ready to be plotted.
8. Use Plotly to create visuals in order to identify trends between the accumulated dataframes.
9. If a significant correlation is suspected from the initial visualizations, perform a regression analysis to verify or disprove the suspected hypothesis.
10. Document trends/correlations found from Plotly visualizations and any other conclusions found.

Results and Code

Importing and Cleaning

Before delving into the code that produced the results for this research project, we must highlight the code used to set up and clean our datasets. This was an important part of being able to utilize and comprehend the data we found and took up a significant portion of the time spent coding for this project.

The following packages were imported:

In [ ]:
```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import statsmodels.api as sm
from plotly.subplots import make_subplots
import doctest
# !pip install Pyarrow
```

The following cleaning and reformatting functions were used:

In [4]:
```
def clean(series):
    '''
    A function to clean columns in a dataframe to all lowercase strings.
    Returns a cleaned list of the given column.

    >>> clean(pd.Series(["UPPERCASE", "low", "UPandLow"]))
    ['uppercase', 'low', 'upandlow']
    >>> clean(small_test["Label (Grouping)"].head(2))
    ['population 5 years and over', '\xa0\xa0\xa0\xa0speak only english']
    '''
    cleaned = []
    for row in series:
        cleaned.append(str(row).lower())
    return cleaned

def clean_df(df):
    '''
    A function to clean column_name in a dataframe to all lowercase strings,
    cleans whitespace, and replaces within-name whitespace with underscores.
```

```
    Returns a dataframe.

    >>> clean_df(reformat_census_df(small_test, 3)).columns.tolist()[1:4]
    ['speak_only_english', 'speak_a_language_other_than_english', 'speak_a_language_other_than_english']
    >>> clean_df(small_test).columns.tolist()[2]
    'washington!!percent!!estimate'
    '''
    df = df.copy(deep=True)
    df.columns = [name.strip().lower().replace(' ','_') for name in df.columns]
    return df

def reformat_census_df(df, num_levels):
    '''
    Returns a reformatted US census dataframe, with multi-indexes.
    This function only takes in US census csv data, and must meet these requirements.
    - The faux-index is named "Label (Grouping)" or "label_(grouping)"
    - Faux-index includes one string of a multi-index, with '!!' as the delimiter
    - Has a County and Ratio equivalent within the multi-index levels
        - Ratio label type example: Estimate, percent

    >>> reformat_census_df(clean_df(small_test), 3).index[1]
    ('washington', 'percent')
    >>> reformat_census_df(small_test, 3).index.get_level_values(1).tolist()
    ['Total', 'Percent']
    '''
    df = df.copy(deep=True)
    if "label_(grouping)" in df.columns:
        df.set_index("label_(grouping)", inplace=True)
    else:
        df.set_index("Label (Grouping)", inplace=True)
    df = df.transpose()
    df['temp_index'] = df.index
    regex_pattern = r"^[\w,\s]+"
    for i in range(num_levels - 1):
        regex_pattern += "\!\![\w,\s]+"
    regex_pattern += "$"
    df = df.loc[df['temp_index'].str.contains(regex_pattern)]
    df.insert(0, "County", "")
    df.insert(1, "Ratio", "")

    for row in df['temp_index']:
        terms = row.split("!!")
        terms[0] = terms[0].split(",")[0]
        df.loc[row, ['County']] = terms[0]
        df.loc[row, ['Ratio']] = terms[1]

    df.rename_axis(None, axis=1, inplace=True)
    df.set_index(['County', 'Ratio'], inplace=True)
    df.drop(columns=['temp_index'], inplace=True, axis=1)
    return df

doctest.testmod()
```

Out[4]:
```
TestResults(failed=0, attempted=6)
```

The following code was used to import and set up the dataframes:

In [6]:
```
# RQ1: reading in csv containing languages spoken at home
language_demographic = pd.read_csv("data/S1601LanguagesSpokenAtHome.csv")
language_demographic = reformat_census_df(language_demographic, 3)
language_demographic = clean_df(language_demographic)
language_demographic = language_demographic.loc[
    (language_demographic.index.get_level_values("County"), "Percent"), :]

# RQ1: reading in csv containing school enrollment data
enrollment = pd.read_csv("data/S1401SchoolEnrollment.csv")
enrollment = reformat_census_df(enrollment, 3)
enrollment = clean_df(enrollment)
enrollment = enrollment.loc[(enrollment.index.get_level_values("County"), "Percent"), :]

# RQ1: cleaning the languages and enrollment percentages to decimal floats
sub_language = language_demographic.iloc[:, [1, 2, 4, 8, 12, 16]].apply(
    lambda x: x.str.replace("%","").astype(float) / 100)

sub_enrollment = enrollment.iloc[:, [7, 8]].apply(
    lambda x: x.str.replace("%","").astype(float) / 100)

# RQ2: reading in csv containing school districts and their PPE
ppe = pd.read_csv("data/SchoolDistrictPPEAverages.csv")
ppe = clean_df(ppe)
ppe["districtname"] = clean(ppe["districtname"])

# RQ2: reading in csv containing counties and their school districts
counties = pd.read_csv("data/Counties.csv", header=2)
counties = counties.loc[counties["State Postal Code"] == "WA"]
counties = clean_df(counties)
counties["school_district_name"] = clean(counties["school_district_name"])

# RQ2: merging ppe and counties by district name
```

```
counties_ppe = ppe.merge(counties, right_on="school_district_name", left_on="districtname")
counties_ppe.set_index("school_district_name", inplace=True)
counties_ppe.drop(columns=["districtname"], inplace=True)

# RQ2: merging ppe and languages spoken by county
ppe_languages = counties_ppe[["ppe", "county_names"]].groupby("county_names").mean()
ppe_languages["County"] = ppe_languages.index
ppe_languages = ppe_languages.merge(sub_language, left_on="county_names", right_on="County")
ppe_languages.set_index("County", inplace=True)

# RQ3: reading in csv containing ancestry languages spoken at home
households = pd.read_csv("data/DP02AncestryLanguagesSpokenAtHome.csv")
households = reformat_census_df(households, 2)
households = clean_df(households)

# RQ3: getting the total diversity for non-American ancestry
ancestry_backgrounds = households.iloc[:, -32:-4].copy().loc[
    (households.index.get_level_values("County"), "Estimate"), :]
ancestry_backgrounds = ancestry_backgrounds.apply(
    lambda x: x.str.replace(",","").astype(float))

ancestry_backgrounds.columns = [name.strip() for name in ancestry_backgrounds.columns]

# RQ3: getting the ancestry backgrounds in each county
diverse_count = []
full_count = [x.sum() for _, x in ancestry_backgrounds.iterrows()]
for _, row in ancestry_backgrounds.iloc[:, 1:].iterrows():
    diverse_count.append(row.sum())

ancestry_backgrounds["diversity_ratio"] = [x / y for x, y in zip(diverse_count, full_count)]

# RQ3: getting the languages being used in each household and merging with ancestral backgrounds
lang_usages = households.iloc[:, -45:-33].copy().loc[
    (households.index.get_level_values("County"), "Percent"), :]

ancestry_backgrounds.index = ancestry_backgrounds.index.droplevel(1)
lang_usages.index = lang_usages.index.droplevel(1)

diverse_lang = pd.DataFrame(ancestry_backgrounds["diversity_ratio"]).merge(
    lang_usages, right_index=True, left_index=True)

diverse_lang.iloc[:, 2:] = diverse_lang.iloc[:, 2:].apply(
    lambda x: x.str.replace("%","").astype(float) / 100)

diverse_lang = diverse_lang.iloc[1:, :]
```

As seen in the previous code cell, the following lambda function was used in order to convert string percentage values (in the form of __._%) found in our datasets into float versions:

In [ ]:
```
lambda x: x.str.replace("%","").astype(float) / 100
```

RQ1: How does language diversity in each Washington county affect their educational attainment outcomes?

For this research question, we merged the educational enrollment and languages spoken at home datasets. This allowed us to plot the attainment values for college undergraduate and postgraduate students in all Washington counties in comparison to the rates of various different language speaking groups.

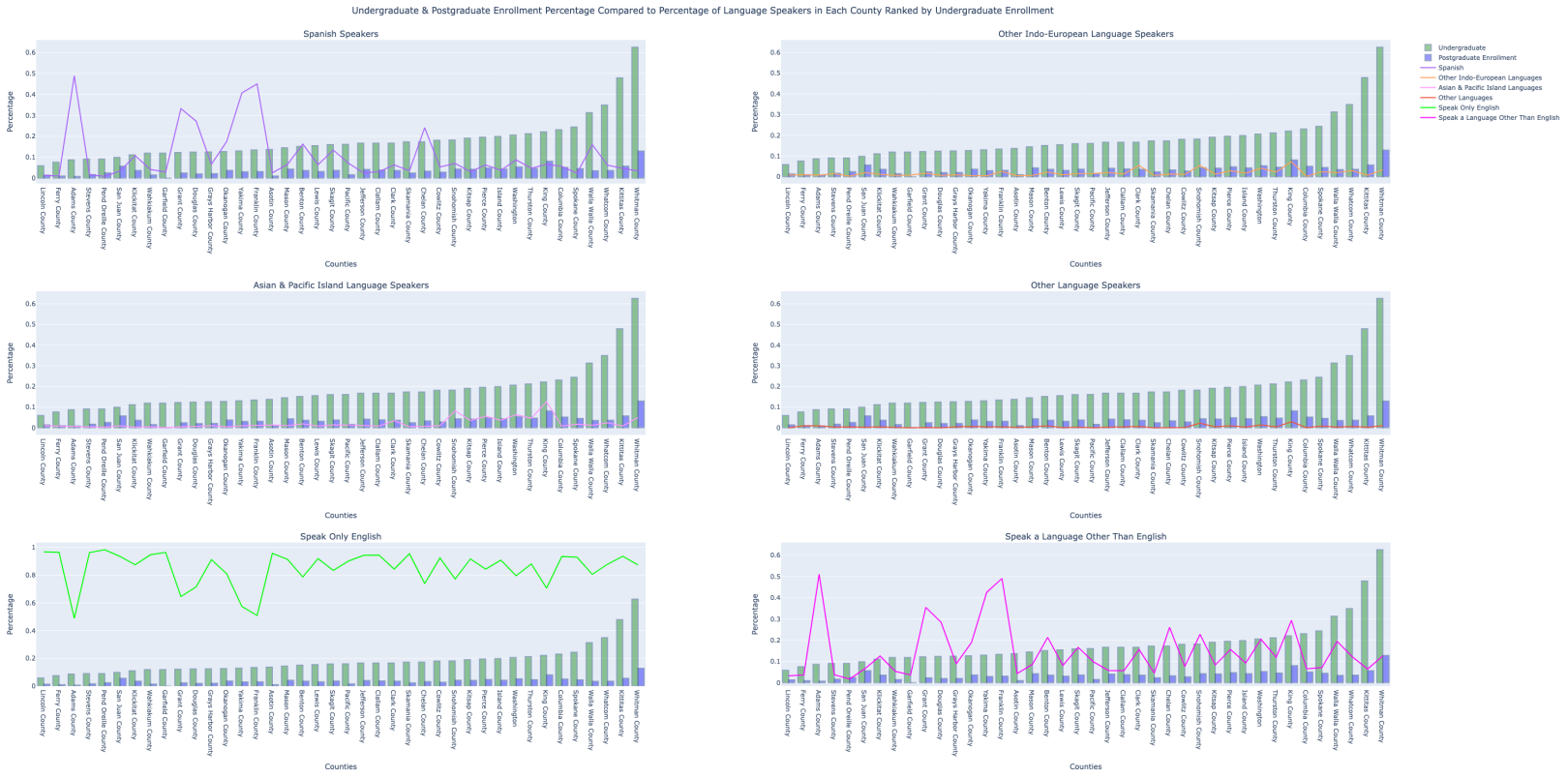
In [ ]:
```
enrollment_language = pd.merge(sub_enrollment, sub_language, on=["County"])
enrollment_language = enrollment_language.sort_values(by="college,_undergraduate")

all_lang_colors = ["#AB63FA", "#FFA15A", "#FF97FF", "#EF553B", "lime", "magenta"]

fig1 = make_subplots(3,2, subplot_titles=("Spanish Speakers",
                                          "Other Indo-European Language Speakers",
                                          "Asian & Pacific Island Language Speakers",
                                          "Other Language Speakers",
                                          "Speak Only English", "Speak a Language Other Than English"))

names = {"spanish": "Spanish",
         "other_indo-european_languages": "Other Indo-European Languages",
         "asian_and_pacific_island_languages": "Asian & Pacific Island Languages",
         "other_languages": "Other Languages", "speak_only_english": "Speak Only English",
         "speak_a_language_other_than_english": "Speak a Language Other Than English"}

for (i, j), type, color, bool in zip([(1, 1), (1, 2), (2, 1), (2, 2), (3, 1), (3, 2)],
                                     ["spanish", "other_indo-european_languages",
                                      "asian_and_pacific_island_languages",
                                      "other_languages", "speak_only_english",
                                      "speak_a_language_other_than_english"],
                                     all_lang_colors,
                                     [True, False, False, False, False, False]):
      fig1.add_trace(go.Bar(x=enrollment_language.index,
                            y=enrollment_language["college,_undergraduate"],
                            name="Undergraduate",
                            showlegend=bool,
```

```
                              marker_color = "green",
                              opacity=0.4,
                              marker_line_color="rgb(8,48,107)",
                              marker_line_width=2),
                              row = i, col = j)
       fig1.add_trace(go.Bar(x=enrollment_language.index,
                              y=enrollment_language["graduate,_professional_school"],
                              name="Postgraduate Enrollment",
                              showlegend=bool,
                              marker_color = "blue",
                              opacity=0.4,
                              marker_line_color="rgb(8,48,107)",
                              marker_line_width=2),
                              row = i, col = j)
       fig1.add_trace(go.Scatter(x=enrollment_language.index, y=enrollment_language[type],
                                 line=dict(color=color), name=names[type]), row = i, col = j)

fig1.update_layout(autosize=False, width=3000,
                   height=1500,
                   title_text="Undergraduate & Postgraduate Enrollment Percentage Compared to Percentage o
                            + " Language Speakers in Each County Ranked by Undergraduate Enrollment",
                  title_x=0.45, xaxis_title="Counties", yaxis_title="Percentage")

for i in range(1, 7):
       fig1["layout"][f"xaxis{i}"]["title"]="Counties"
       fig1["layout"][f"yaxis{i}"]["title"]="Percentage"
```

These six graphs are the ouput of the code above. Each of the six graphs has plotted bars in the background representing the enrollment percentage values of undergraduate students in green and postgraduate students in blue for each county. Additionaly for each graph, the percentage of residents who speak the highlighted language or language group is represented by the colored line plot, with one language group per graph. Each graph is sorted by undergraduate enrollment percentages by county, meaning the county with the highest undergraduate enrollment rate is on the right. Let's look at the top four graphs first, looking at speaking rates of each of the four non-English only groups. Starting with Spanish speakers, we can clearly see that there is no correlation between Spanish speakers and college enrollment in Washington counties, as there are 3-4 large spikes in speaking rates towards the lower end of the college enrollment percentages. This is the only graph out of the first four that has such a strong negative correlation. From this, a conclusion could be made opposite to what we initially hypothesized; that larger Spanish speaking and hispanic communities actually have lower college enrollment rates than English only speakers due to these communities historically recieving less economic and societal support. Although, it could be argued that this negative correlation is not relevant as there may be external factors that affect college enrollment rates within communities containing Spanish speakers. This graph was the one that suprised us the most, and also became the most thought provoking out of the ones created for the first research question.

Looking at the remaining three graphs, there is a very slight correlation between increasing language speaking rates and college enrollment rates, with the largest being seen in the Asian & Pacific Island Language Speakers graph and the smallest being seen in the Other Language Speakers graph. Although the correlation is minimal, it does help indicate that speaking a language other than English may help contribute towards higher college enrollment rates in Washington counties. However, it is most likely a minimal one of many factors that contribute towards this.

Finally, let's examine the bottom two graphs that contain information on those who speak only English, and those who speak a language other than English (a combination of the top four graphs). The majority of conclusions between these two graphs can be shared as they are essentially inverses of each other. One key detail noticed about these graphs is that there is a much higher amplitude in the speaking rates towards the lower end of college enrollment for counties, showing that counties with higher rates of English only speakers tend to have higher enrollment rates as well. Initially, this was suprising, but as we thought about it more, it became a little clearer. We can conclude that communities with higher rates of English only speakers tend to not house as many minority populations as those with higher language diversity, thus not representing many minority communities. These minority communities are often not supported as well as their white counterparts in society, having harder times pursuing higher education after high school due to many reasons such as financial situations, housing, societal restrictions. Although these graphs did not help show that investing in increased language diversity would lead to higher college enrollment rates, it did help further illuminate the issue of struggling minority communities in relation to higher education. As well as their presence in Washingtonian and American society as communities who consistently face increased hardships in comparison to other communities who have historically held more socioeconomic strength.



RQ2: How do school district budgets and spending affect spoken language diversity in WA counties?

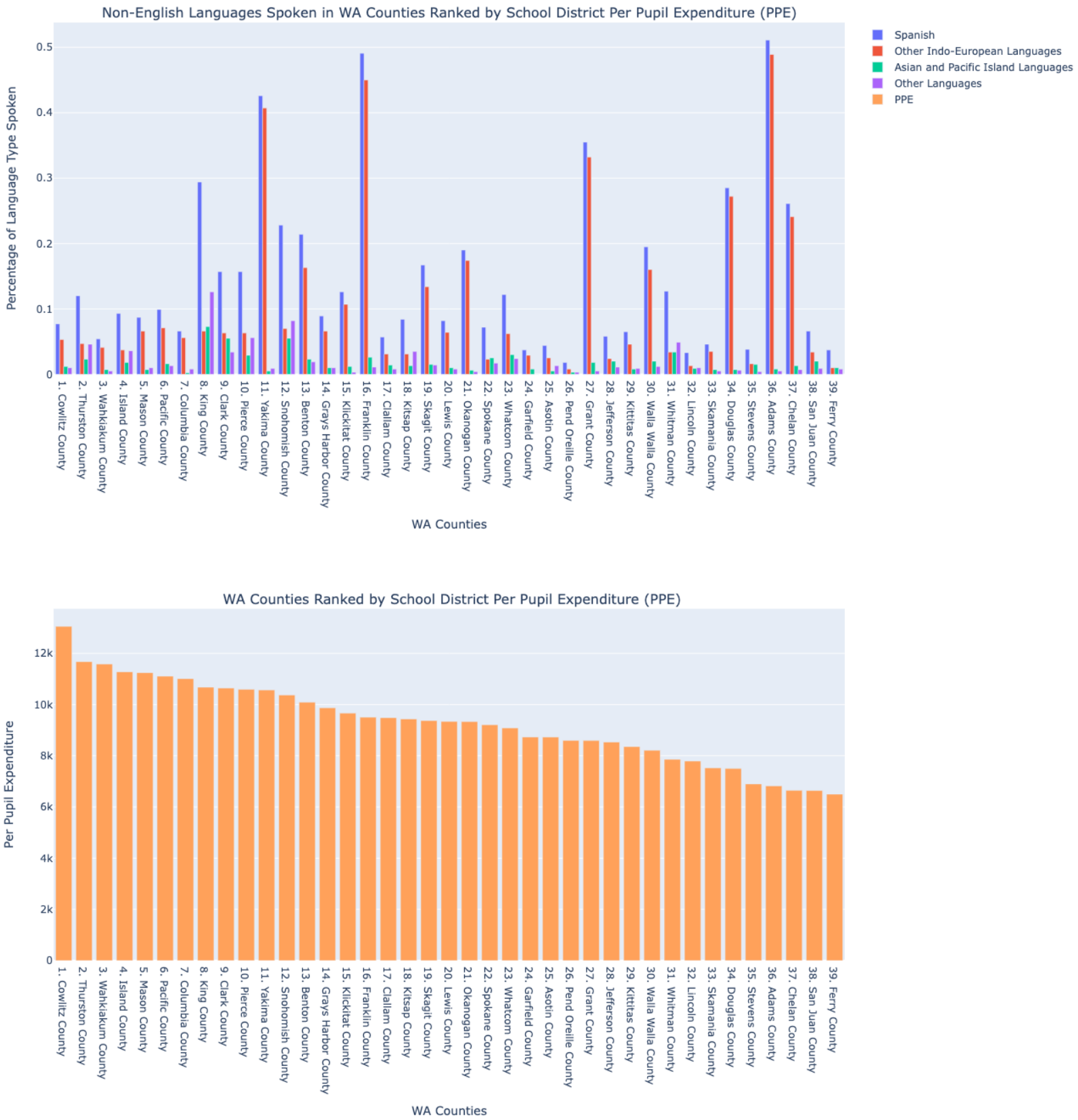
For this question, we used the merged dataset with per pupil expenditures and languages spoken at home to get a dataframe that included both variables by county. Using that dataframe, we were able to plot two subplots, with one showing the rates of the language type being spoken per county excluding English and the other showing the per pupil expenditure. These two plots use the same x-axis of counties sorted in order of the most to least per pupil expediture.

In [ ]:
```
ppe_languages = ppe_languages.sort_values(by="ppe", ascending=False)
all_lang_colors = ["#636EFA", "#EF553B", "#00CC96", "#AB63FA", "#FFA15A"]
all_counties = []

for i in range(len(ppe_languages.index)):
    all_counties.append(str(i + 1) + ". " + ppe_languages.index[i])
all_lang_types = ppe_languages.iloc[:, 2:].columns
all_lang_labels = ["Spanish", "Other Indo-European Languages",
                   "Asian and Pacific Island Languages", "Other Languages"]
fig2 = make_subplots(2, 1, subplot_titles=(
    "Non-English Languages Spoken in WA Counties Ranked by School District Per Pupil Expenditure (PPE)",
    "WA Counties Ranked by School District Per Pupil Expenditure (PPE)"))

for i in range(len(all_lang_labels)):
    fig2.add_trace(go.Bar(x=all_counties,
                    y=ppe_languages[all_lang_types[i]],
                    name=all_lang_labels[i],
                    marker_color=all_lang_colors[i]
                    ), row = 1, col = 1)

fig2.add_trace(go.Bar(x=all_counties,
                y=ppe_languages["ppe"],
                name="PPE",
                marker_color=all_lang_colors[4]
                ), row = 2, col = 1)

fig2.update_layout(autosize=False, width=1300, height=1400)
fig2["layout"]["xaxis"]["title"]="WA Counties"
fig2["layout"]["xaxis2"]["title"]="WA Counties"
fig2["layout"]["yaxis"]["title"]="Percentage of Language Type Spoken"
fig2["layout"]["yaxis2"]["title"]="Per Pupil Expenditure"
```

From this code above, we were able to generate this visualization below. To restate our motive for choosing to use two subplots was to search for any correlations between the amount of PPEs and the language spoken in a county. The PPE for each county was determined by taking the mean PPE of all K-12 school districts in the county. The subplots share the same x-axis of a PPE-ranked list of Washington state counties. To express the various language categories spoken in each county, we separated them by the non-English categories that were reported from our initial dataset: Spanish, Other Indo-European, Asian and Pacific Island, and Other languages. All of this resulted in the first visualization. When we order the counties by PPE, no clear trendline is shown from the tops of the bars. We chose to include the second subplot of the general average PPE by county for transparency and to also examine the slope in the directionality of the PPE averages.

We can see that from the highest PPE in Cowlitz County with around \$12,000 to the lower PPE in Ferry County with around \$6,500, showing is a gradual linear decrease in PPE. Comparatively, the plot above with the languages includes various spikes of Spanish and Other Indo-European languages. Overall there is basic no correlation between these two variables. This poses the question: should language diversity be a factor in PPE levels? From our results, it certainly seems as though they do not have any influence over the expenditure levels. However, because we can infer that language diversity implies racial diversity to an extent, and drawing from knowledge about the intentional survey of people of Hispanic or Latino Origin for civil rights purposes in censuses, it does not seem far fetched for language diversity to be a factor in expenditure amounts. As

fluency in one language does not guarantee fluency in another, and while we also would like to look at a larger sample and English literacy rates by language, we believe that perhaps investing PPE by language diversity could benefit early to adolescent literacy in students in a direction towards higher equity in learning resources.



RQ3: To what extent do Washington residents continue to practice their ancestral culture through language?

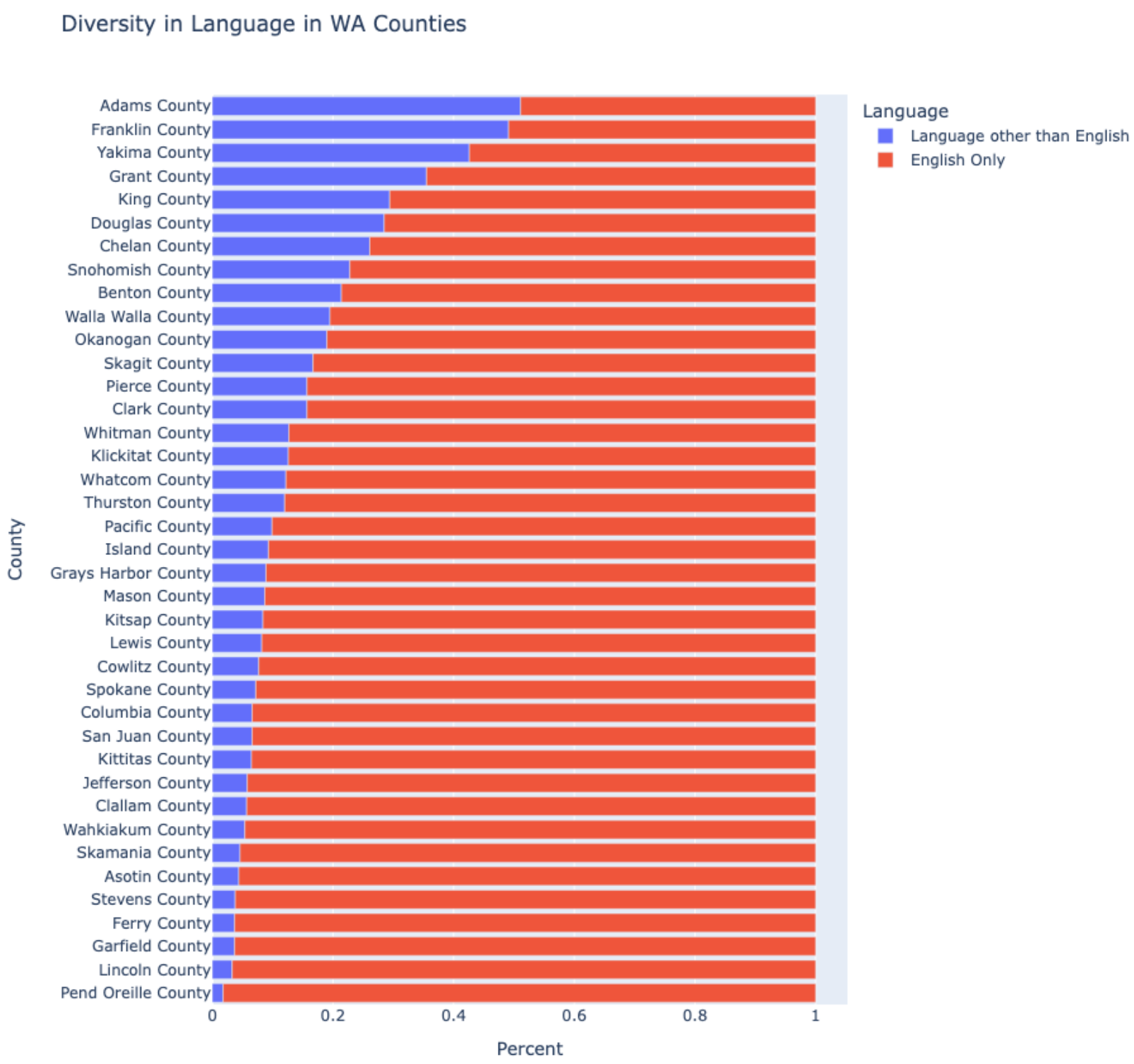
Using the dataframe created above that included ancestral background and language diversity, we first plotted the language diversity by county to visualize the overall breakdown of language by county. The following plot was created by plotting the percentages of households that speak English only by a various range of language diversity that can be seen in a Washington county with a linear regression line overlaying it.

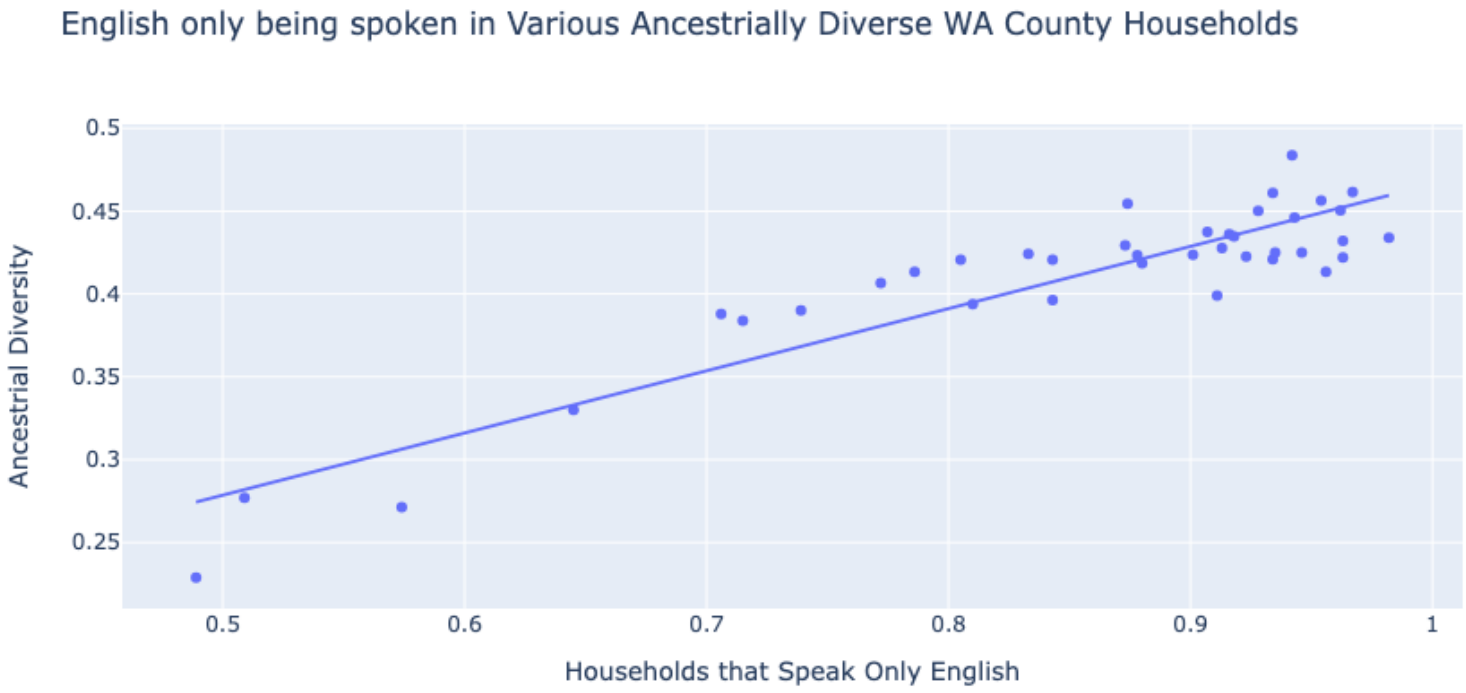
In [ ]:
```
lang_counties = []
lang_counts = []
lang_kind = []

for idx, row in diverse_lang.iterrows():
    lang_counties.append(idx)
    lang_counts.append(row["english_only"])
    lang_kind.append("English Only")
    lang_counties.append(idx)
    lang_counts.append(row["language_other_than_english"])
    lang_kind.append("Language other than English")

lang_data = {}
lang_data["County"] = lang_counties
lang_data["Percent"] = lang_counts
lang_data["Language"] = lang_kind
lang_data = pd.DataFrame(lang_data).sort_values(by="Percent")
fig3 = px.bar(lang_data, x="Percent", y="County", color="Language",
              title="Diversity in Language in WA Counties", height=900)

# Let's analyze the strength of the relationships of the two variables:
```

```
# ancestral diversity in each county and English only being spoken in households
fig3 = px.scatter(diverse_lang, y='diversity_ratio', x='english_only', trendline='ols',
                  title='English only being spoken in Various Ancestrally Diverse WA County Households',
                  labels=dict(diversity_ratio="Ancestral Diversity",
                              english_only="Households that Speak Only English"))
```

First, to gauge the diversity of languages at a smaller scale, we chose to group all non-English speaking households together against the exclusively English speaking households. We were intrigued by the variability of the households that spoke English only and thought that the English Only households provided a range that was wide enough that could potentially show us a relationship to the ancestral background when utilized as a predicted variable.

The scatterplot shows the rates of English Only speaking households, as determined above, and ancestral diversity. We can clearly see a postiive and fairly strong linear relationship, indicating that there is some correlation between these two variables. Using the least-ordinary squares table, it shows that the R-squared value is 0.827, meaning that 82.7% of the variability in the ancestral diversity can be attributed to English Only households. Also, with the p-value being less than the 5% significance level of 0.000, we can reject the notion that English Only households do not have a correlation with ancestral diversity. Before running these analyses, we expected the results to have an inverse relationship, as we assumed that more ancestral diversity was indicative of more non-English languages being used. We were surprised to have found that this was the opposite and to conclude that it is to a low extent that Washington residents continue to practice their ancestral culture through language. When taking on a communicative lens to further understand these results, many times, immigrant families purposefully choose not to teach their children their native tongue in hopes of faster assimilation by picking up English and becoming Americanized. Another reason for this positive correlation could be that when counties facilitate people with diverse ancestral backgrounds, to communicate with one another, English may have defaulted to be the most universal and accessible language.





In [8]:
```
X = diverse_lang['english_only']
Y = diverse_lang['diversity_ratio']
X = sm.add_constant(X)
m = sm.OLS(Y.astype(float), X.astype(float))

r = m.fit()
r.summary()
```

Out[8]:

OLS Regression Results

Dep. Variable:	diversity_ratio	**R-squared:**	0.827
Model:	OLS	**Adj. R-squared:**	0.822
Method:	Least Squares	**F-statistic:**	177.1
Date:	Tue, 12 Mar 2024	**Prob (F-statistic):**	1.12e-15
Time:	22:47:11	**Log-Likelihood:**	94.625
No. Observations:	39	**AIC:**	-185.3
Df Residuals:	37	**BIC:**	-181.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.0909	0.024	3.736	0.001	0.042	0.140
english_only	0.3754	0.028	13.307	0.000	0.318	0.433

Omnibus:	1.297	**Durbin-Watson:**	1.591
Prob(Omnibus):	0.523	**Jarque-Bera (JB):**	1.044
Skew:	-0.155	**Prob(JB):**	0.593
Kurtosis:	2.261	**Cond. No.**	13.9

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Impact and Limitations

Our greatest informant and limitation was our reliance on the US Census Bureau for reliable data. As 4 out of 5 of the datasets we chose to analyze were obtained from this singular, albeit grand, source, we believe that it is natural that our analysis would extend the same biases and limitations. As mentioned above in the data setting section, generally, census data is prone to response bias, with social desirability bias arguably the most susceptible to the responders as it deals with demographic data. An implication of this can be that responders of certain cultures and demographics that are less recognized in the United States may not have felt the same inclination for declaring their ancestry and language compared to those who are in communities that are actively celebrated. Especially as we look at ancestry in Washington state, we have to recognize that there are communities that had their lineage systemically erased from archives and history, such as the African and Indigenous Americans, and therefore the responders may not be able to report their full ancestral background. We also noticed the lack of AAPI ancestry in our data. With the dataset's notes stating that "this table lists only the largest ancestry groups..." and that it only recorded the first and second ancestry, we questioned how the US Census defined "largest" and

what groups made that “largest” cutoff. While they also redirected to “more detailed tables” for more ancestral groups, we agreed that this was an example of erasure, as it initially conveyed that there were no people of AAPI ancestry in Washington state.

As for our graphs, we recognize that there is skewing of the axes for a closer look at the visualizations, which can lead to unintentional misinformed opinions and understandings. While we did not see a correlation between the per pupil expenditures and the languages spoken in each county, we can see that it is apparent that there is a high proportion of Spanish speakers in comparison to the other non-English languages. This is true for the visualizations showing educational attainment by language and, although it is a loose trend, the counties with the most amount of Spanish speaker seemed to have the least amount of educational attainment. We found trends interesting, as we first thought that language spoken did not have any influence in the ways counties are supported and uplifted, however we urge viewers to think critically about other socioeconomic factors that we failed to highlight in our project that are popular indicators of wellbeing (eg. race, income). We also acknowledge that we were only able to analyze one year’s worth of data, which decreases the reliability of our project. If we were able to invest more time into this project, we envision that we would have been able to produce much more reliable and valid results.

Challenge Goals

Multiple Datasets:

As mentioned in the data setting section, we used multiple datasets from the US Census in order to compare trends across Washington counties by connecting location based data, the Washington county, as our primary key. For example, to answer our third research question, we have taken data about the spoken language diversity of Washington counties and also data about county-specific ancestral backgrounds and have merged them together with the county being the main denominator. By using multiple datasets, we were able to achieve higher relationality and diversity in the kinds of data that were analyzed.

New Libraries:

We have mainly used Plotly Express and Plotly Graph Objects from the Plotly library to create interactive visualizations to support the communication of our analyses. We believe that including visualizations that have some form of interactivity with a viewer encourages more direct attention and interpretation. We especially chose to lean into the hover label feature of Plotly, which allows for the viewer to hover over a data point on a visualization and indicate its x and y values without having to physically reference the axis. We found that because we wanted to communicate relationships between features, stacking the plot on top of one another created a sense of relativity, with the hover labels helping to highlight the more granular information, allowing for data to be conveyed more transparently.

Messy Data:

We initially did not believe that we would have to deal with messy data, as our first glance at our datasets seemed to have been stored neatly. We quickly realized that the data from the robust US Census’s interactive interface did not translate as well when reading them as CSVs for analysis. Multi-indexes and column sections and headings were particularly messy, as we had to manually separate them with regex patterns, pivot the tables, and rename the columns to remove non-breaking and white spaces. As we were also using a dataset that was from collectors outside of the US Census, we had to normalize the county and district names to make sure joins and merges of the separate tables were performed correctly without data loss.

Result Validity:

After looking at the results of our first visualization from our third research question, we deemed that performing a linear regression on the two variables, ancestral diversity and the number of households that spoke English exclusively, could produce a meaningful output. Although not as major as the other challenge goals, we felt that the inclusion of statistical analysis to answer this question helped strengthen our overall understanding of the relationships about language diversity within Washington state.

Plan Evaluation

Time Utilized:

Our proposed work plan time estimates were relatively accurate. Our estimates were close to reality for the most part as we were able judge based on previous experience in working with datasets and visualizations from class assignments in CSE 163. The two areas that took longer than anticipated were cleaning the data and creating the visualizations. Since we initially did not anticipate having to clean the data much, we estimated a lower amount of time spent on this step. However, as mentioned previously, the datasets ended up being quite messy and required a lengthy amount of cleaning code and time. For creating the visualizations, we took longer than expected since we had not gotten much experience using Plotly as it was a new library

from the one(s) using during class assignments. Other than these two areas, we were able to follow our proposed plan fairly accurately and spent a fair amount of time on each step of this research project.

Developing Code:

For each high-level section, the work was divided into half. Before each member went off to work independently, conversations were held ahead of time in order to discuss the scope and define what half of the workload would look like. For each research question, instead of directly using pandas and Python to wrangle the data, we seperated chunks of code to establish a basis for testing and validation later on.

Testing Code:

For each research question, as well as the cleaning functions, the member responsible for the majority of the code worked on the corresponding test cases. This allowed for the person most familiar with the code and thought process to curate the test cases for them.

Coordinating Work:

Throughout this project, all members discussed with one another before and after starting each high-level section. All members notified the other member about potential challenges or conflicts as soon as possible, and constant communication was encouraged to make sure no member felt unsupported.

Testing

We tested our code by using a variety of assertations, doctests, and statistical regression evaluations. The majority of testing was done using assert statements. We compared the data values contained in each created graph with the values held in each dataframe corresponding to its respective graph. This helped ensure that no values were lost or added during the graph creation. For the cleaning functions, we used doctests and a smaller testing data file. The doctests can be found within the docstring inside each cleaning function located in the Results and Code section of this report, and the smaller data file can be located in our project repository as 'small_census_test.csv'. We know that our code computes the expected result because our assertation tests include statements to check sorted groupings of values. This not only ensures that all values we want to include in the graph are included, but also guarantees that the values are presented in the correct order or manner. By comparing sorted values, we can make sure that the highest and lowest values are in the right place in relativity to each other. This is increasingly important as many of our visualizations rely on sorted values in order to make conclusions and takeways more clear and concise. Additionally, to check for the goodness of the linear model used in the last research question, we found the mean squared error of the model and found it to be at around 0.2, which is fairly close to 0, meaning that the linear model was a good fit for the data it was being used on. The code for all of our testing (excluding the doctests found in each cleaning function located in the Results and Code section of this report) can be found below.

Assert statements for RQ1 code:

In [ ]:
```
# Undergraduate data
assert sorted(fig1.to_dict()["data"][0]["y"]) == sorted(
    sub_enrollment[
        "college,_undergraduate"].tolist()), "Undergrad data does not match expected"
# Postgrad data
assert sorted(fig1.to_dict()["data"][1]["y"]) == sorted(
    sub_enrollment[
        "graduate,_professional_school"].tolist()), "Postgrad data does not match expected"
# Spanish data
assert sorted(fig1.to_dict()["data"][2]["y"]) == sorted(
    sub_language[
        "spanish"].tolist()), "Spanish data does not match expected"
# Indo-Euro data
assert sorted(fig1.to_dict()["data"][5]["y"]) == sorted(
    sub_language[
        "other_indo-european_languages"].tolist()), "Indo-European data does not match expected"
# Asian data
assert sorted(fig1.to_dict()["data"][8]["y"]) == sorted(
    sub_language[
        "asian_and_pacific_island_languages"].tolist()), "API data does not match expected"
# Other data
assert sorted(fig1.to_dict()["data"][11]["y"]) == sorted(
    sub_language[
        "other_languages"].tolist()), "Other language data does not match expected"
# English data
assert sorted(fig1.to_dict()["data"][14]["y"]) == sorted(
    sub_language[
        "speak_only_english"].tolist()), "English data does not match expected"
# Other than English data
assert sorted(fig1.to_dict()["data"][17]["y"]) == sorted(
    sub_language[
        "speak_a_language_other_than_english"].tolist()), "Other than English data does not match expected"
# Counties
```

```
assert sorted(fig1.to_dict()["data"][0]["x"]) == sorted(
    enrollment.index.get_level_values("County")), "Counties not accounted for"
```

Assert statements for RQ2 code:

In [ ]:
```
# Testing for all counties included and are sorted by the PPE
ppe_languages_sorted = ppe_languages.sort_values(by="ppe", ascending=False).index
for i in range(len(fig2.to_dict()["data"])):
    all_labels = fig2.to_dict()["data"][i]["x"]
    for label, idx in zip(all_labels, ppe_languages_sorted):
        assert(label.split(" ", 1)[1] == idx), "Data does not match expected"
```

Assert statements and statistical analysis evaluations for RQ3 code:

In [ ]:
```
# Testing for all data points to be on the plot
assert sorted(fig3.to_dict()['data'][0]['x']) == sorted(
    diverse_lang['english_only'].tolist()), "Data points do not match expected"
```

In [ ]:
```
# The mean squared error for checking the goodness of fit of the linear model
sm.tools.eval_measures.mse(diverse_lang['english_only'], diverse_lang['diversity_ratio'], axis=0)
```

```
0.20066900124783196
```

Collaboration

In the process of this project, we consulted the Plotly documentation for guidance on using the Plotly library, University of Washington's Professor Ott Toomet's textbooks (1, 2) on Python for Machine Learning for linear regression and general statistical analysis and interpretation, and Stack Overflow for advice on debugging code. We have not used generative AI in any way for this project.