

```
%load_ext autoreload
%autoreload 2
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import json
import utils.functions as f
```

WARNING:tensorflow:From d:\Anaconda\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

WARNING:root:No module named 'fairlearn': ExponentiatedGradientReduction will be unavailable. To install, run:
pip install 'aif360[Reductions]'
WARNING:root:No module named 'fairlearn': GridSearchReduction will be unavailable. To install, run:
pip install 'aif360[Reductions]'
WARNING:root:No module named 'inFairness': SenSeI and SenSR will be unavailable. To install, run:
pip install 'aif360[inFairness]'
WARNING:root:No module named 'fairlearn': GridSearchReduction will be unavailable. To install, run:
pip install 'aif360[Reductions]'

```
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

Direct Dataset

```
dataset = "direct_matching_20240213"
df_processed = pd.read_csv(f"Data/{dataset}_processed.csv", sep=";")
```

Gender

```
sectors = sorted(df_processed.job_sector.unique())
protected_attribute_gender = "cand_gender"

df_mitigation = df_processed.drop(columns=["distance_km", "match_score", "match_rank"])
columns = df_mitigation.columns
columns = columns.drop(["job_id", "job_sector"])

first_language_index = 6

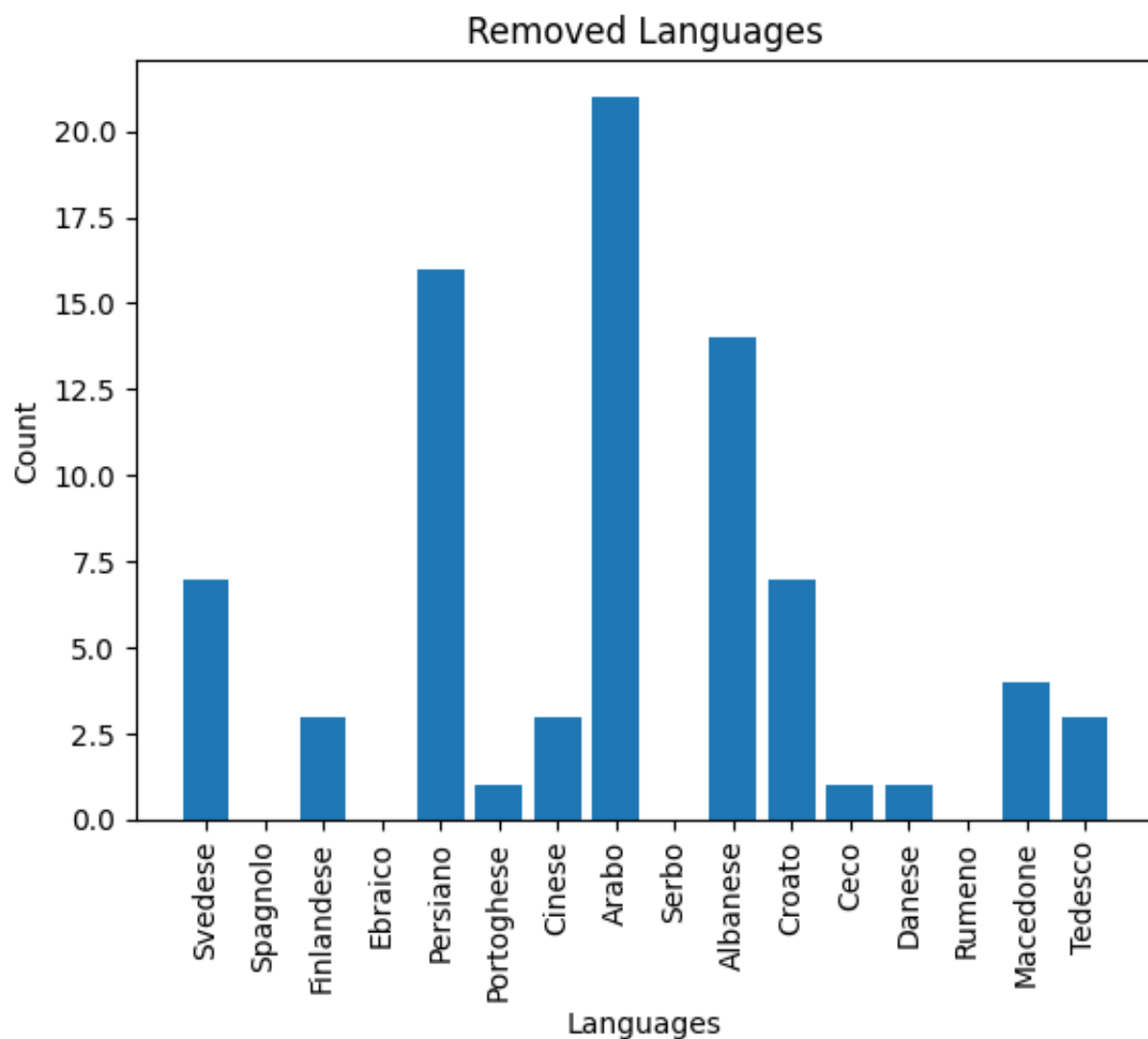
results_df_gender = f.compute_bias_differences_percentage(df_mitigation, sectors, protected_attribute_gender)
results_df_gender.to_excel('Results/Mitigation_direct/results_df_gender_direct.xlsx')
results_df_gender
```

	cand_gender	cand_age_bucket	cand_domicile_province	cand_domicile_region	job_contract
0	0.0	0.0	11.08	1.69	
1	0.0	0.0	20.00	7.00	
2	0.0	0.0	45.80	13.69	
3	0.0	0.0	21.50	16.75	
4	0.0	0.0	62.76	0.00	
5	0.0	0.0	20.09	0.00	
6	0.0	0.0	61.48	26.67	
7	0.0	0.0	37.74	44.57	
8	0.0	0.0	49.52	9.47	
9	0.0	0.0	39.26	10.74	

10 rows × 22 columns

```
%matplotlib inline
```

```
results_df_gender_total = f.compute_bias_differences_percentage(df_mitigation, sectors)
sums_gender = results_df_gender_total.iloc[:, first_language_index:].sum()
f.plot_series(sums_gender, 'Removed Languages', 'Languages')
```



```
job_df_orig_gender, job_df_repaired_gender = f.compute_repaired_df(df_mitigation,7,
```

```
with open('utils/encodings/direct_matching_20240213_cand_domicile_region_encoding.j
    region_mapping = json.load(file)
with open('utils/encodings/direct_matching_20240213_provinces_encoding.json', 'r')
    province_mapping = json.load(file)
```

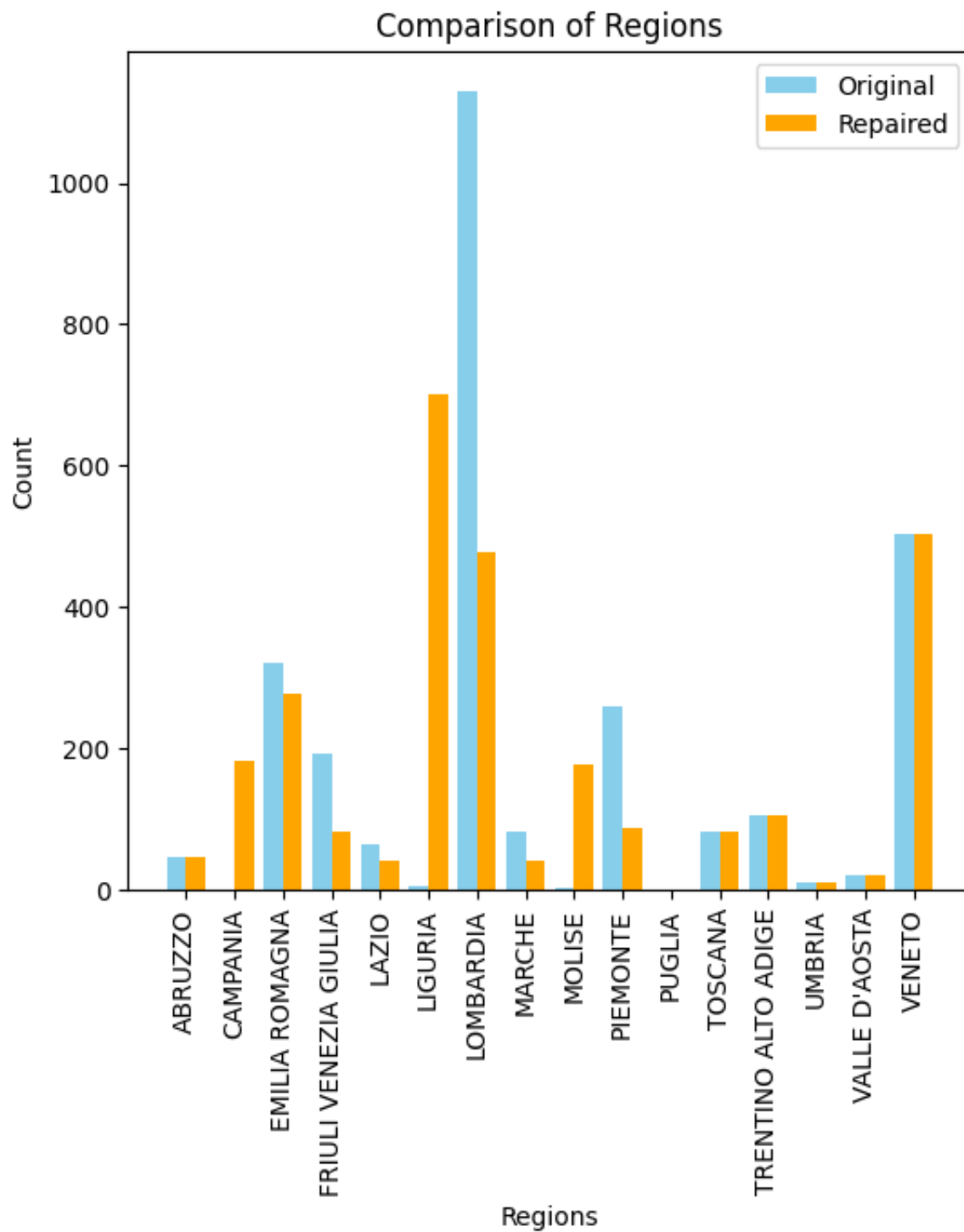
```
reversed_region_mapping = {v: k for k, v in region_mapping.items()}
reversed_province_mapping = {v: k for k, v in province_mapping.items()}
```

```
job_df_orig_gender['cand_domicile_region_str'] = job_df_orig_gender['cand_domicile_
job_df_repaired_gender['cand_domicile_region_str'] = job_df_repaired_gender['cand_c
```

```
job_df_orig_gender['cand_domicile_province_str'] = job_df_orig_gender['cand_domicil
job_df_repaired_gender['cand_domicile_province_str'] = job_df_repaired_gender['cand
```

```
orig_counts = job_df_orig_gender['cand_domicile_region_str'].value_counts()
repaired_counts = job_df_repaired_gender['cand_domicile_region_str'].value_counts()
```

```
orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repa
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Regions', 'Regio
```



Age

```
protected_attribute_age = "cand_age_bucket"

results_df_age = f.compute_bias_differences_percentage(df_mitigation, sectors, protected_attribute_age)
results_df_age.to_excel('Results/Mitigation_direct/results_df_age_direct.xlsx', index=False)
results_df_age
```

	cand_gender	cand_age_bucket	cand_domicile_province	cand_domicile_region	job_contra
0	0.0	0.0	69.08	7.69	
1	0.0	0.0	76.00	26.00	
2	0.0	0.0	75.91	33.76	
3	0.0	0.0	82.75	66.75	
4	0.0	0.0	57.76	56.71	
5	0.0	0.0	47.64	18.49	
6	0.0	0.0	52.96	31.48	
7	0.0	0.0	57.27	12.10	
8	0.0	0.0	61.56	14.21	
9	0.0	0.0	41.48	58.89	

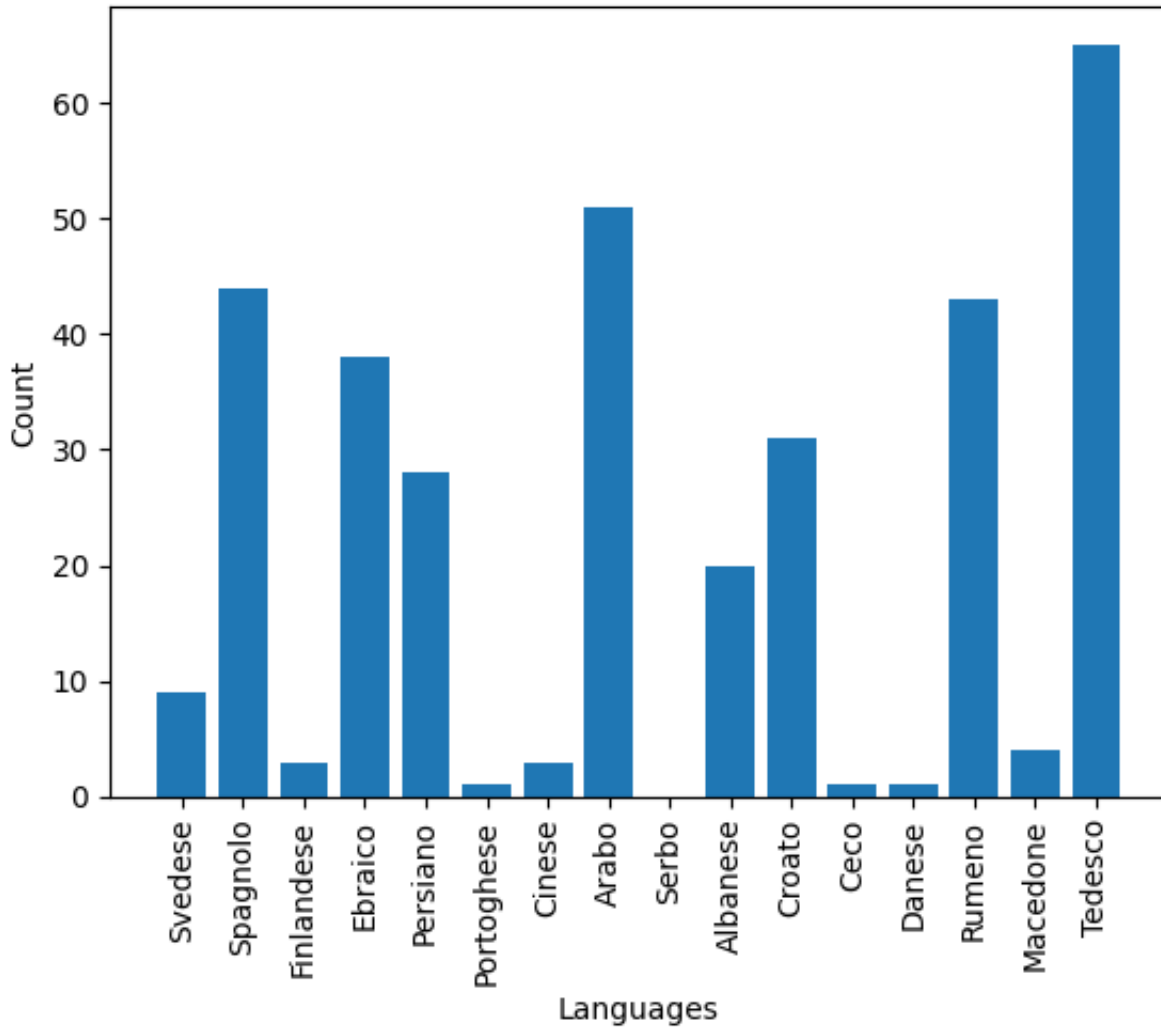
10 rows × 22 columns

```

results_df_age_total = f.compute_bias_differences_percentage(df_mitigation,sectors,
sums_age = results_df_age_total.iloc[:, first_language_index:].sum()
f.plot_series(sums_age,'Removed Languages','Languages')

```

Removed Languages

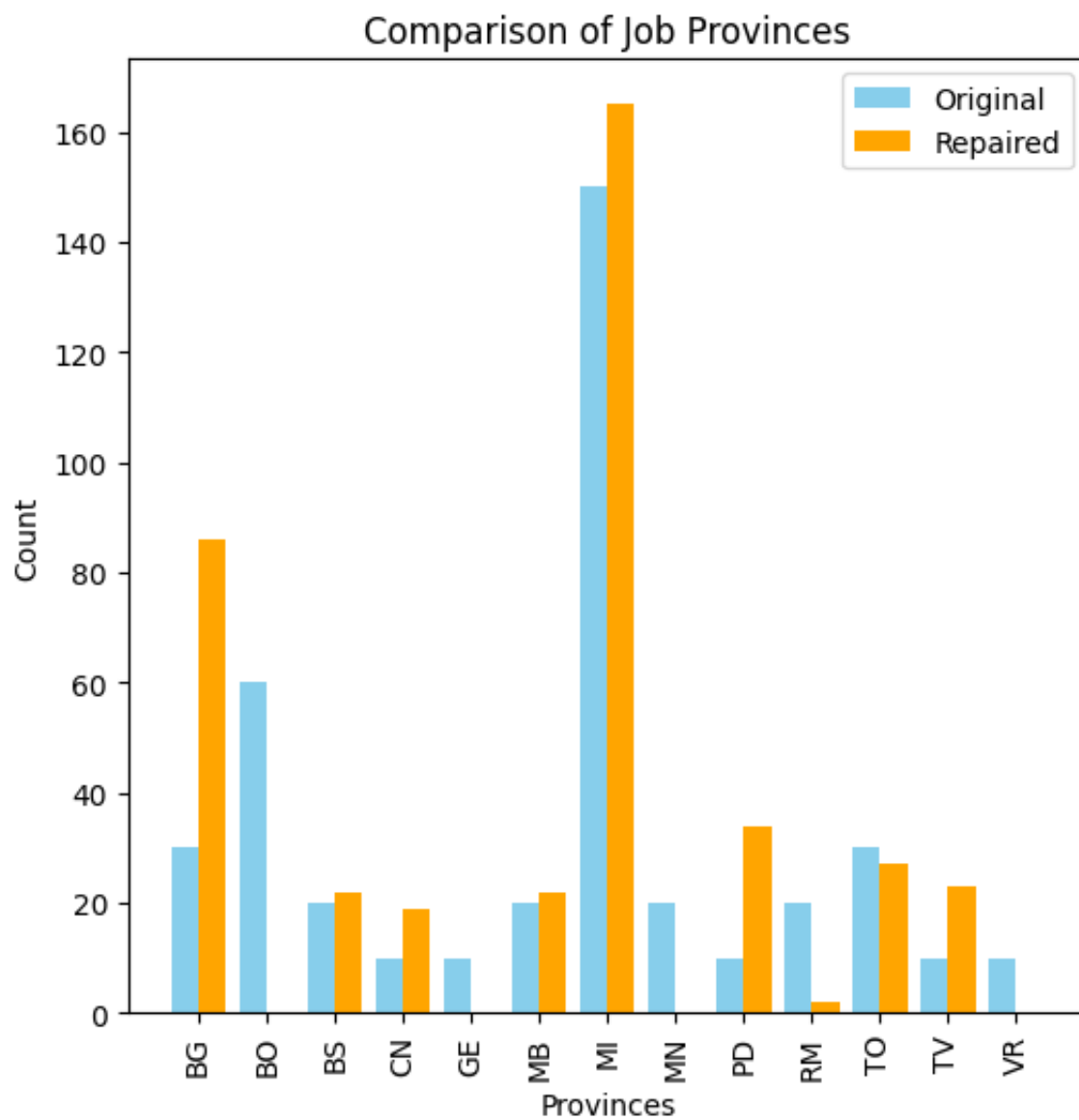


```
job_df_orig_age, job_df_repaired_age = f.compute_repaired_df(df_mitigation,3,protected_languages)
job_df_orig_age['job_work_province_str'] = job_df_orig_age['job_work_province'].map(lambda x: x.replace(' ', '_'))
job_df_repaired_age['job_work_province_str'] = job_df_repaired_age['job_work_province'].map(lambda x: x.replace(' ', '_'))

job_df_orig_age['cand_domicile_region_str'] = job_df_orig_age['cand_domicile_region'].map(lambda x: x.replace(' ', '_'))
job_df_repaired_age['cand_domicile_region_str'] = job_df_repaired_age['cand_domicile_region'].map(lambda x: x.replace(' ', '_'))
```

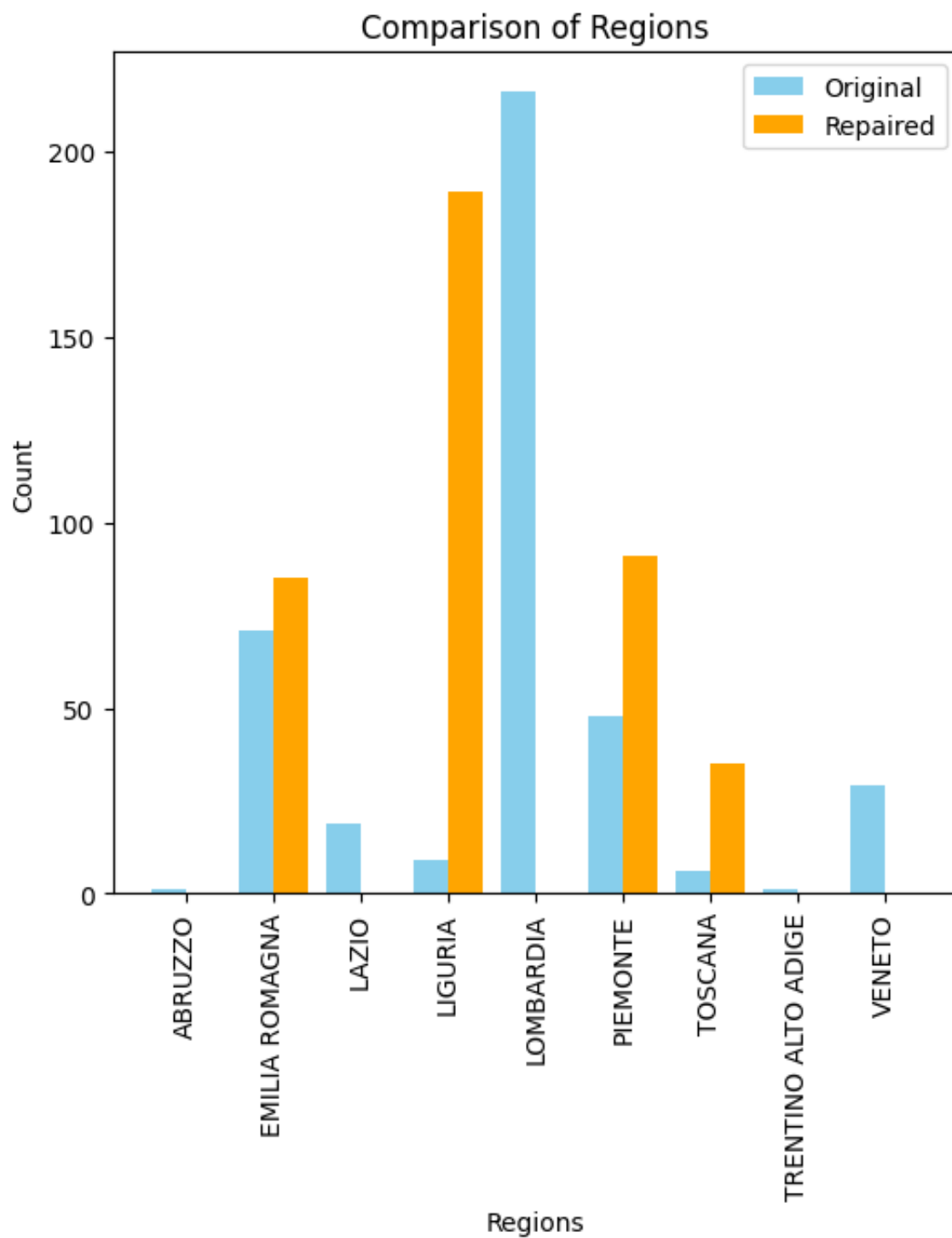
```
orig_counts = job_df_orig_age['job_work_province_str'].value_counts()
repaired_counts = job_df_repaired_age['job_work_province_str'].value_counts()

orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repaired_counts)
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Job Provinces', 'Comparison of Job Provinces')
```



```
orig_counts = job_df_orig_age['cand_domicile_region_str'].value_counts()
repaired_counts = job_df_repaired_age['cand_domicile_region_str'].value_counts()

orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repaired_counts, all_index)
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Regions', 'Region')
```



Location

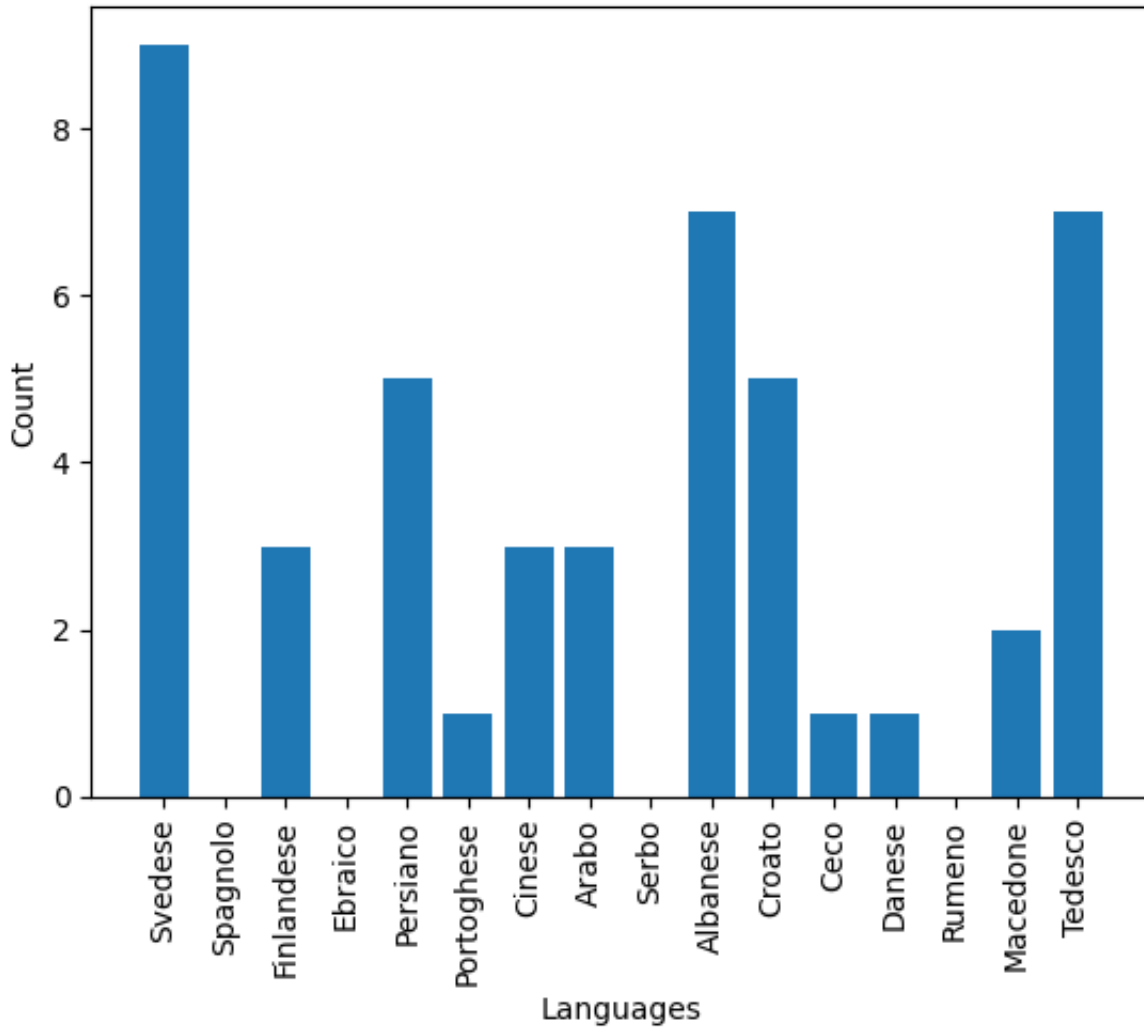
```
protected_attribute_location = "same_location"  
results_df_location = f.compute_bias_differences_percentage(df_mitigation,sectors,p  
results_df_location.to_excel('Results/Mitigation_direct/results_df_location_direct.  
results_df_location
```

	cand_gender	cand_age_bucket	cand_domicile_province	cand_domicile_region	job_contra
0	0.0	0.0	63.38	32.77	
1	0.0	0.0	64.00	57.00	
2	0.0	0.0	65.15	6.20	
3	0.0	0.0	79.00	50.25	
4	0.0	0.0	54.87	64.87	
5	0.0	0.0	72.39	2.10	
6	0.0	0.0	66.30	44.81	
7	0.0	0.0	53.94	46.16	
8	0.0	0.0	73.43	12.36	
9	0.0	0.0	57.04	47.04	

10 rows × 23 columns

```
results_df_location_total = f.compute_bias_differences_percentage(df_mitigation, sec
sums_location = results_df_location_total.iloc[:, first_language_index:-1].sum() #-
f.plot_series(sums_location, 'Removed Languages', 'Languages')
```

Removed Languages



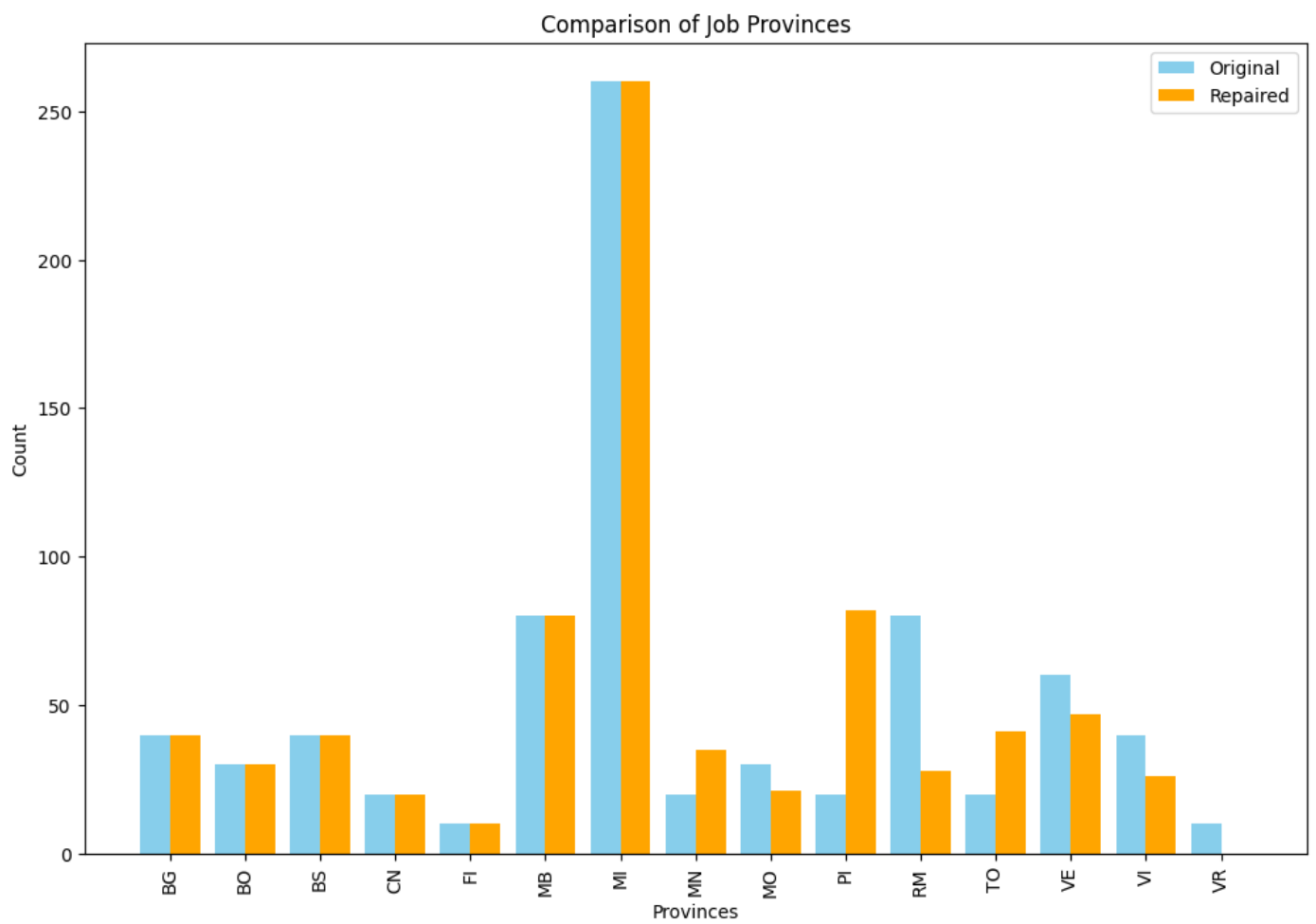
```
job_df_orig_location, job_df_repaired_location = f.compute_repaired_df(df_mitigation)
job_df_orig_location['job_work_province_str'] = job_df_orig_location['job_work_province_str']
job_df_repaired_location['job_work_province_str'] = job_df_repaired_location['job_work_province_str']
```

```
job_df_orig_location['cand_domicile_province_str'] = job_df_orig_location['cand_domicile_province_str']
job_df_repaired_location['cand_domicile_province_str'] = job_df_repaired_location['cand_domicile_province_str']
```

```
job_df_orig_location['cand_domicile_region_str'] = job_df_orig_location['cand_domicile_region_str']
job_df_repaired_location['cand_domicile_region_str'] = job_df_repaired_location['cand_domicile_region_str']
```

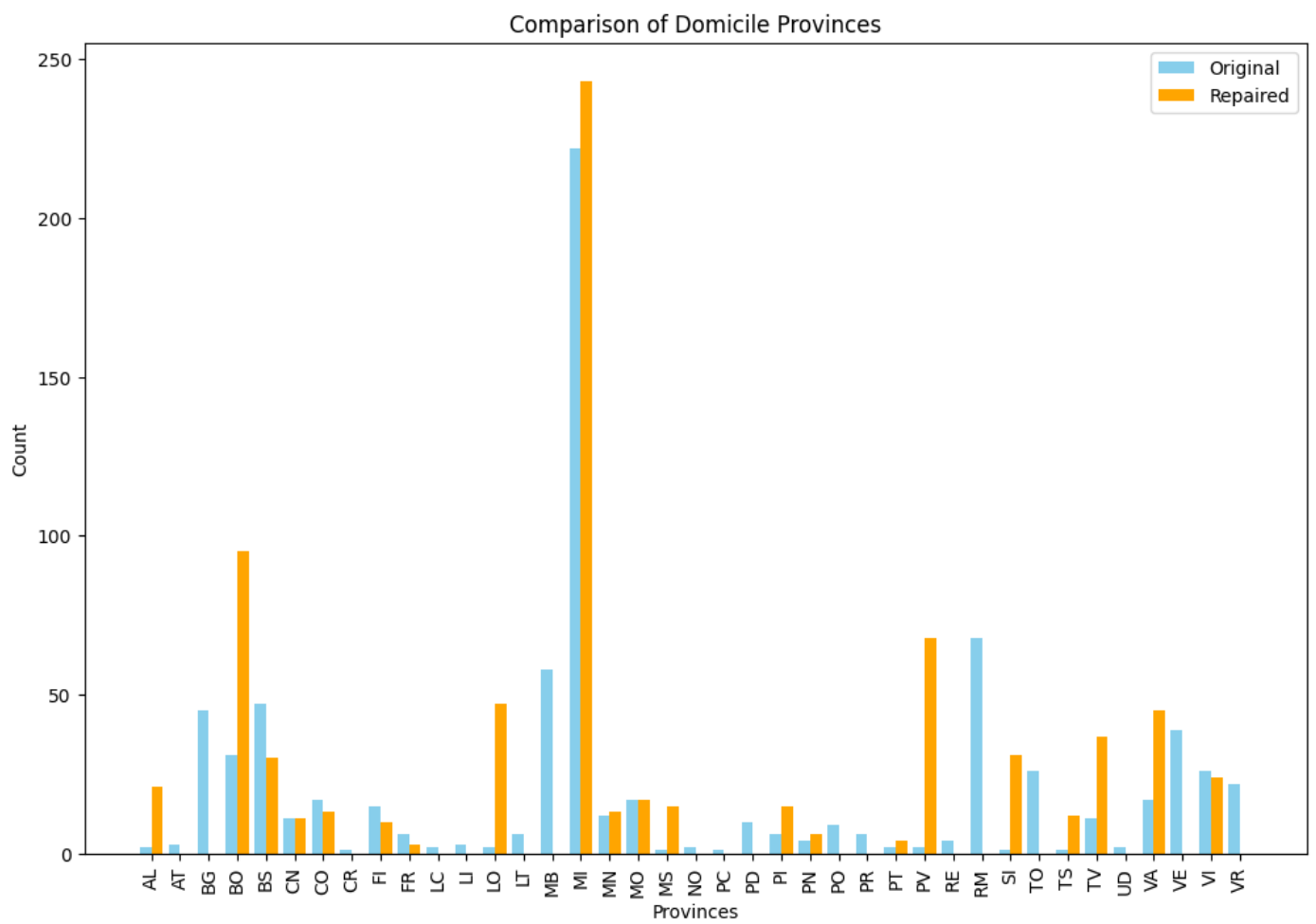
```
orig_counts = job_df_orig_location['job_work_province_str'].value_counts()
repaired_counts = job_df_repaired_location['job_work_province_str'].value_counts()
```

```
orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repaired_counts)
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Job Provinces',
```



```
orig_counts = job_df_orig_location['cand_domicile_province_str'].value_counts()
repaired_counts = job_df_repaired_location['cand_domicile_province_str'].value_counts()

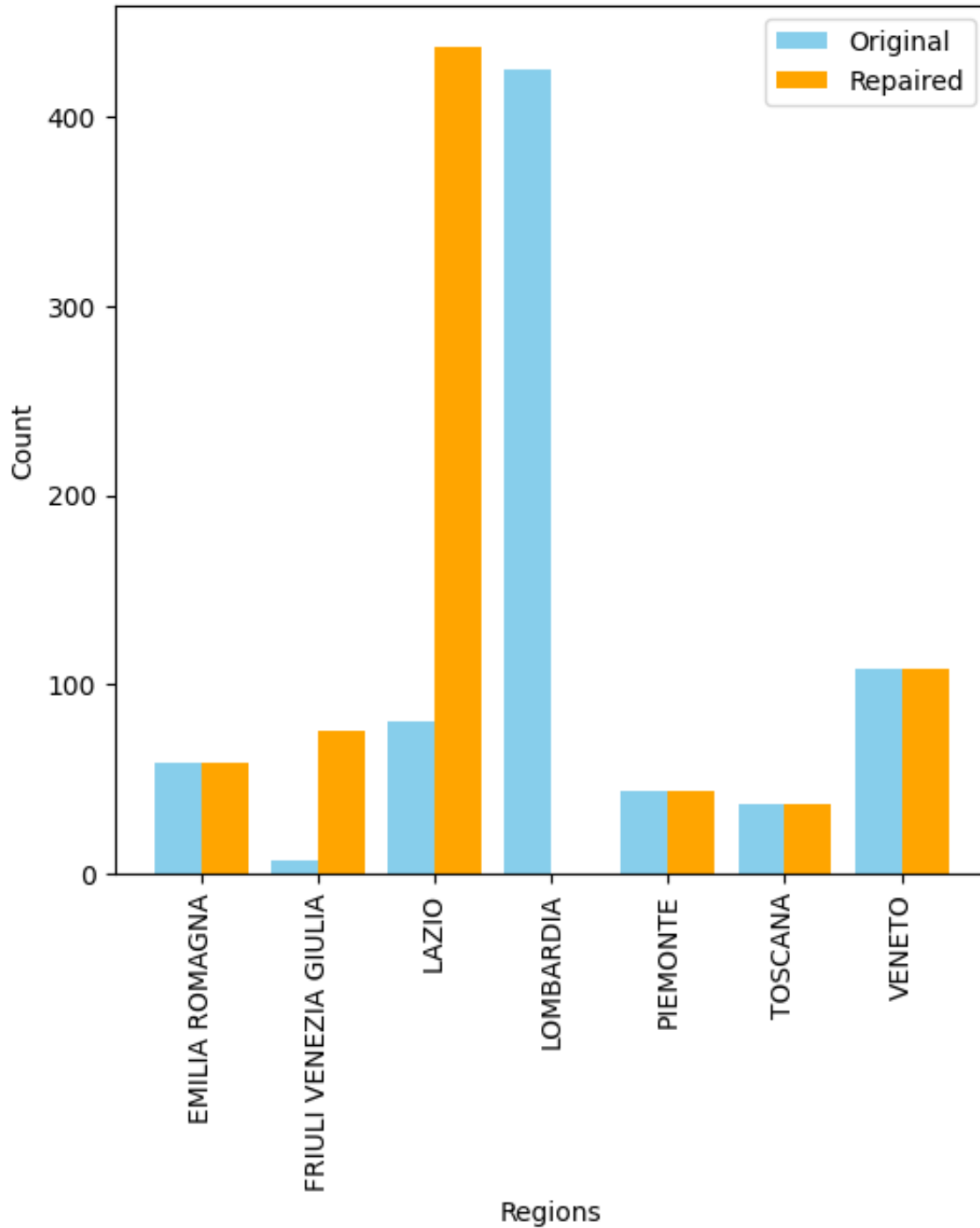
orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repaired_counts, all_index)
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Domicile Provinces')
```



```
orig_counts = job_df_orig_location['cand_domicile_region_str'].value_counts()
repaired_counts = job_df_repaired_location['cand_domicile_region_str'].value_counts()

orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repaired_counts, all_index)
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Regions', 'Region')
```

Comparison of Regions



Reverse Dataset

```
dataset = "reverse_matching_20240213"
df_processed = pd.read_csv(f>Data/{dataset}_processed.csv", sep=";")
df_mitigation = df_processed.drop(columns=["distance_km", "match_score", "match_rank"])
columns = df_mitigation.columns
columns = columns.drop(["job_id", "job_sector"])
sectors = sorted(df_processed.job_sector.unique())
```

Gender

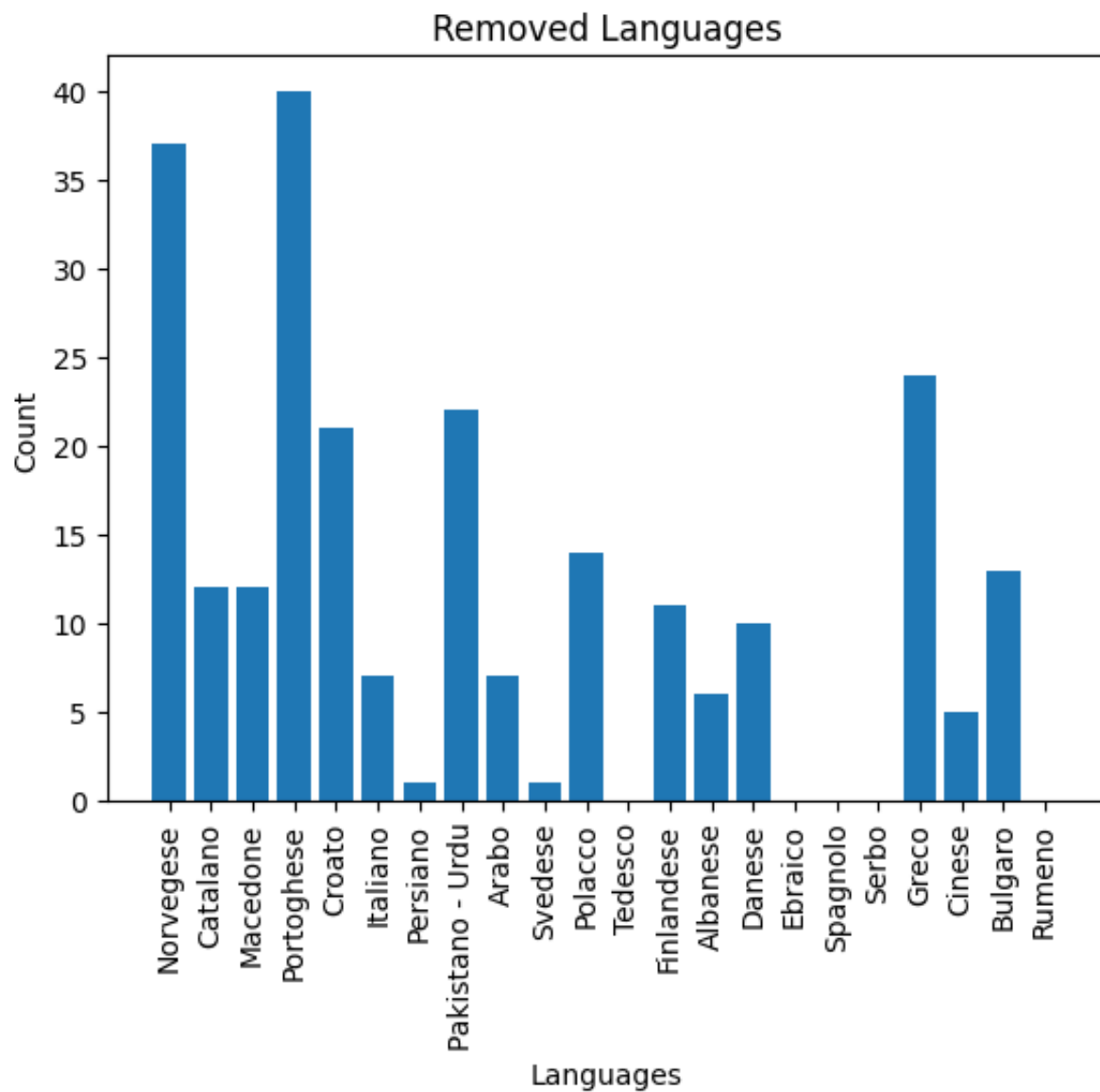
```
protected_attribute_gender = "cand_gender"
```

```
results_df_gender = f.compute_bias_differences_percentage(df_mitigation, sectors, protected_attribute_gender)  
results_df_gender.to_excel('Results/Mitigation_reverse/results_df_gender_reverse.xlsx')  
results_df_gender
```

	cand_gender	cand_age_bucket	cand_domicile_province	cand_domicile_region	job_contract
0	0.0	0.0	39.19	0.00	
1	0.0	0.0	52.09	34.20	
2	0.0	0.0	33.83	15.18	
3	0.0	0.0	28.80	0.00	
4	0.0	0.0	22.84	0.00	
5	0.0	0.0	61.22	48.83	
6	0.0	0.0	32.91	40.98	
7	0.0	0.0	28.98	0.00	
8	0.0	0.0	55.62	31.28	
9	0.0	0.0	34.77	0.00	
10	0.0	0.0	23.45	0.04	
11	0.0	0.0	31.32	0.00	
12	0.0	0.0	31.12	13.23	
13	0.0	0.0	32.58	0.00	
14	0.0	0.0	38.58	25.94	
15	0.0	0.0	37.59	39.49	

16 rows × 28 columns

```
results_df_gender_total = f.compute_bias_differences_percentage(df_mitigation, sectors, protected_attribute_gender)  
sums_gender = results_df_gender_total.iloc[:, first_language_index:].sum()  
f.plot_series(sums_gender, 'Removed Languages', 'Languages')
```



```
job_df_orig_gender, job_df_repaired_gender = f.compute_repaired_df(df_mitigation,5,
```

```
with open('utils/encodings/reverse_matching_20240213_cand_domicile_region_encoding.
    region_mapping = json.load(file)
with open('utils/encodings/reverse_matching_20240213_provinces_encoding.json', 'r')
    province_mapping = json.load(file)
```

```
reversed_region_mapping = {v: k for k, v in region_mapping.items()}
reversed_province_mapping = {v: k for k, v in province_mapping.items()}
```

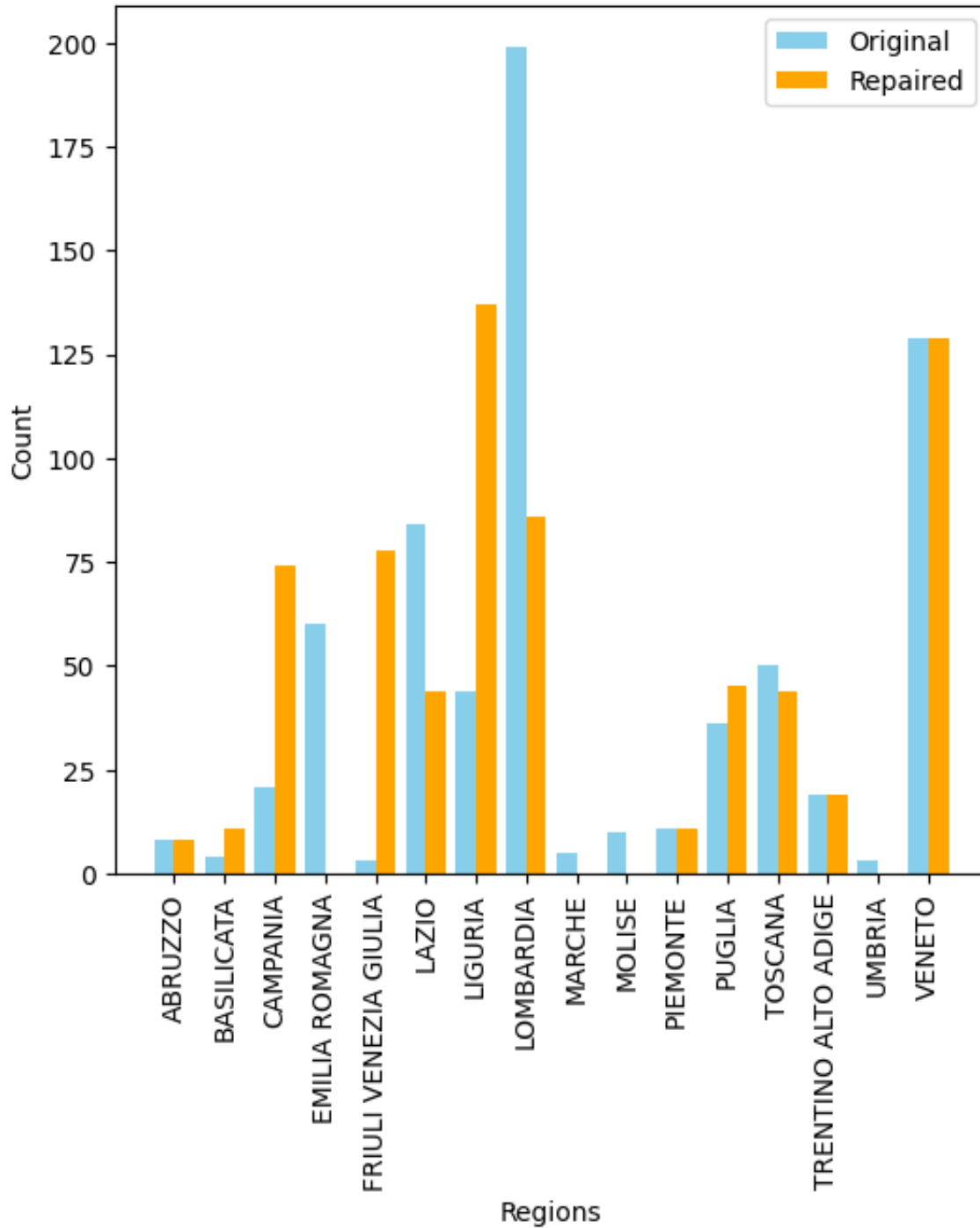
```
job_df_orig_gender['cand_domicile_region_str'] = job_df_orig_gender['cand_domicile_
job_df_repaired_gender['cand_domicile_region_str'] = job_df_repaired_gender['cand_c
```

```
job_df_orig_gender['cand_domicile_province_str'] = job_df_orig_gender['cand_domicil
job_df_repaired_gender['cand_domicile_province_str'] = job_df_repaired_gender['cand
```

```
orig_counts = job_df_orig_gender['cand_domicile_region_str'].value_counts()
repaired_counts = job_df_repaired_gender['cand_domicile_region_str'].value_counts()
```

```
orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repa
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Regions', 'Regio
```

Comparison of Regions



Age

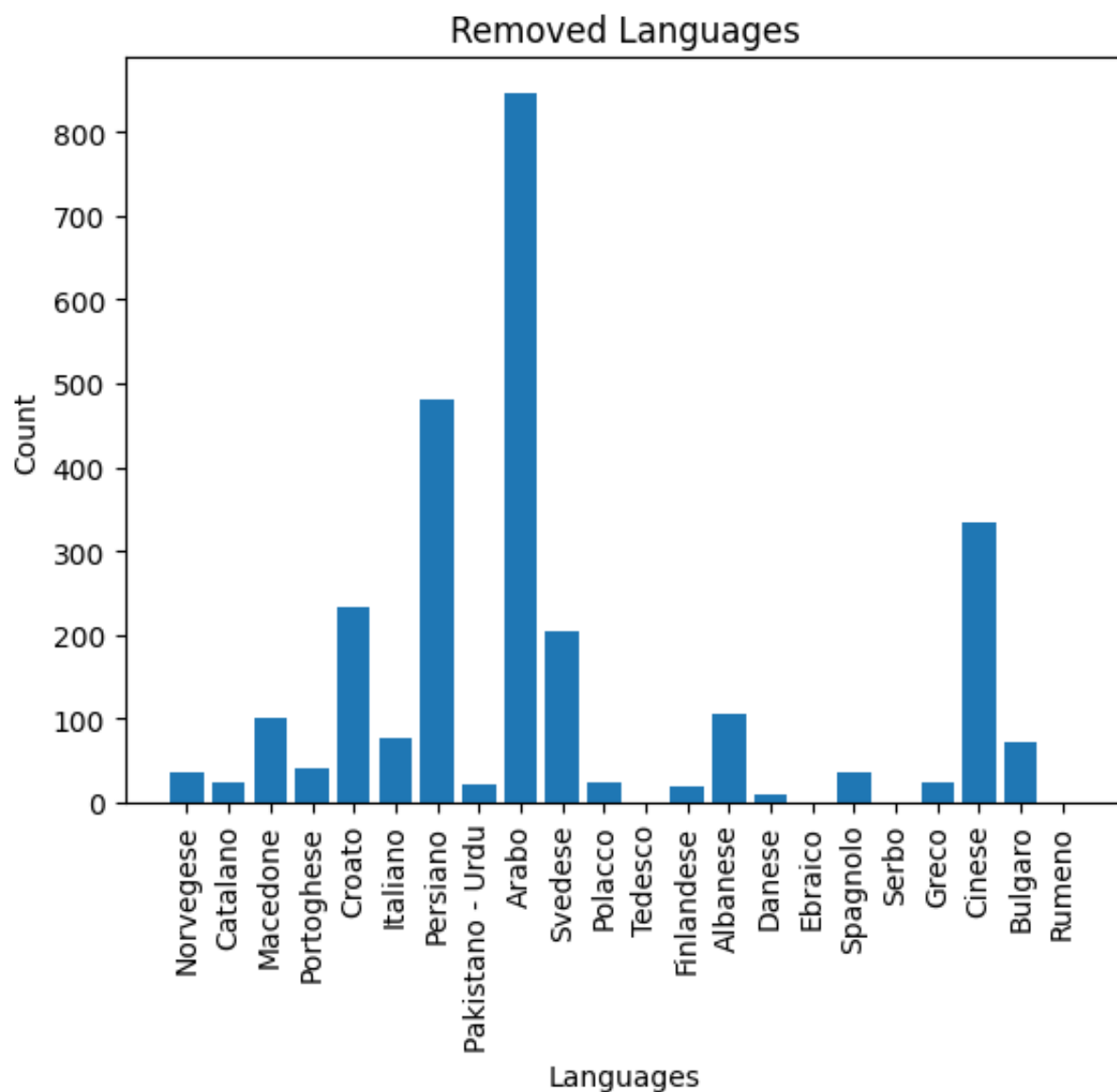
```
protected_attribute_age = "cand_age_bucket"

results_df_age = f.compute_bias_differences_percentage(df_mitigation, sectors, protected_attribute_age)
results_df_age.to_excel('Results/Mitigation_reverse/results_df_age_reverse.xlsx', index=False)
results_df_age
```

	cand_gender	cand_age_bucket	cand_domicile_province	cand_domicile_region	job_contr
0	0.0	0.0	45.80	14.61	
1	0.0	0.0	68.54	47.91	
2	0.0	0.0	59.88	32.56	
3	0.0	0.0	48.70	23.25	
4	0.0	0.0	52.10	11.77	
5	0.0	0.0	68.37	58.02	
6	0.0	0.0	63.61	29.75	
7	0.0	0.0	42.73	23.12	
8	0.0	0.0	56.27	44.09	
9	0.0	0.0	50.90	43.85	
10	0.0	0.0	65.07	56.45	
11	0.0	0.0	54.94	9.40	
12	0.0	0.0	53.91	28.75	
13	0.0	0.0	37.05	45.01	
14	0.0	0.0	52.17	66.95	
15	0.0	0.0	43.48	42.57	

16 rows × 28 columns

```
results_df_age_total = f.compute_bias_differences_percentage(df_mitigation,sectors,
sums_age = results_df_age_total.iloc[:, first_language_index:].sum()
f.plot_series(sums_age,'Removed Languages','Languages')
```



```

job_df_orig_age, job_df_repaired_age = f.compute_repaired_df(df_mitigation,14,prote
job_df_orig_age['job_work_province_str'] = job_df_orig_age['job_work_province'].map
job_df_repaired_age['job_work_province_str'] = job_df_repaired_age['job_work_provin

```

```

job_df_orig_age['cand_domicile_region_str'] = job_df_orig_age['cand_domicile_region
job_df_repaired_age['cand_domicile_region_str'] = job_df_repaired_age['cand_domicil

```

```

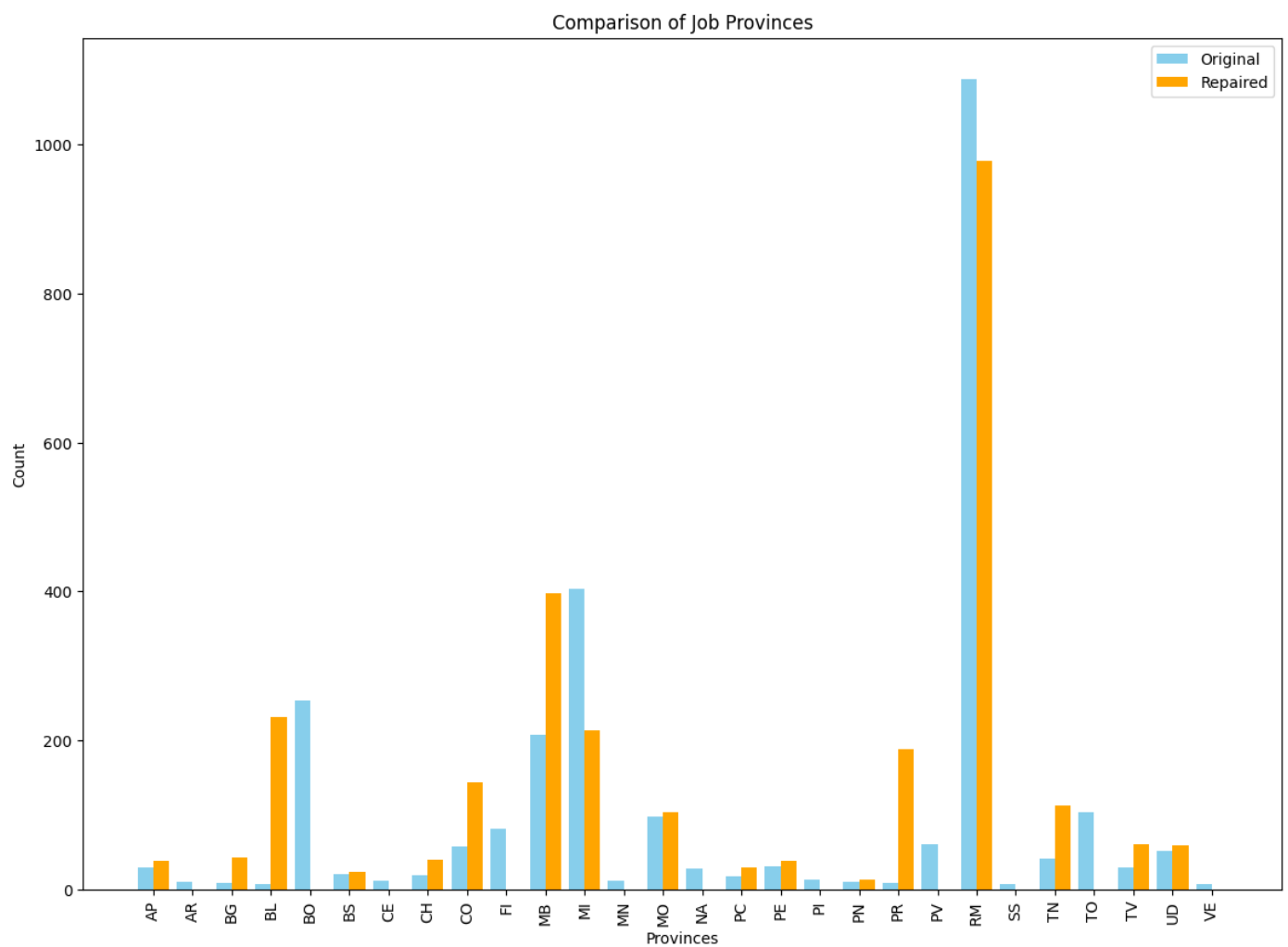
orig_counts = job_df_orig_age['job_work_province_str'].value_counts()
repaired_counts = job_df_repaired_age['job_work_province_str'].value_counts()

```

```

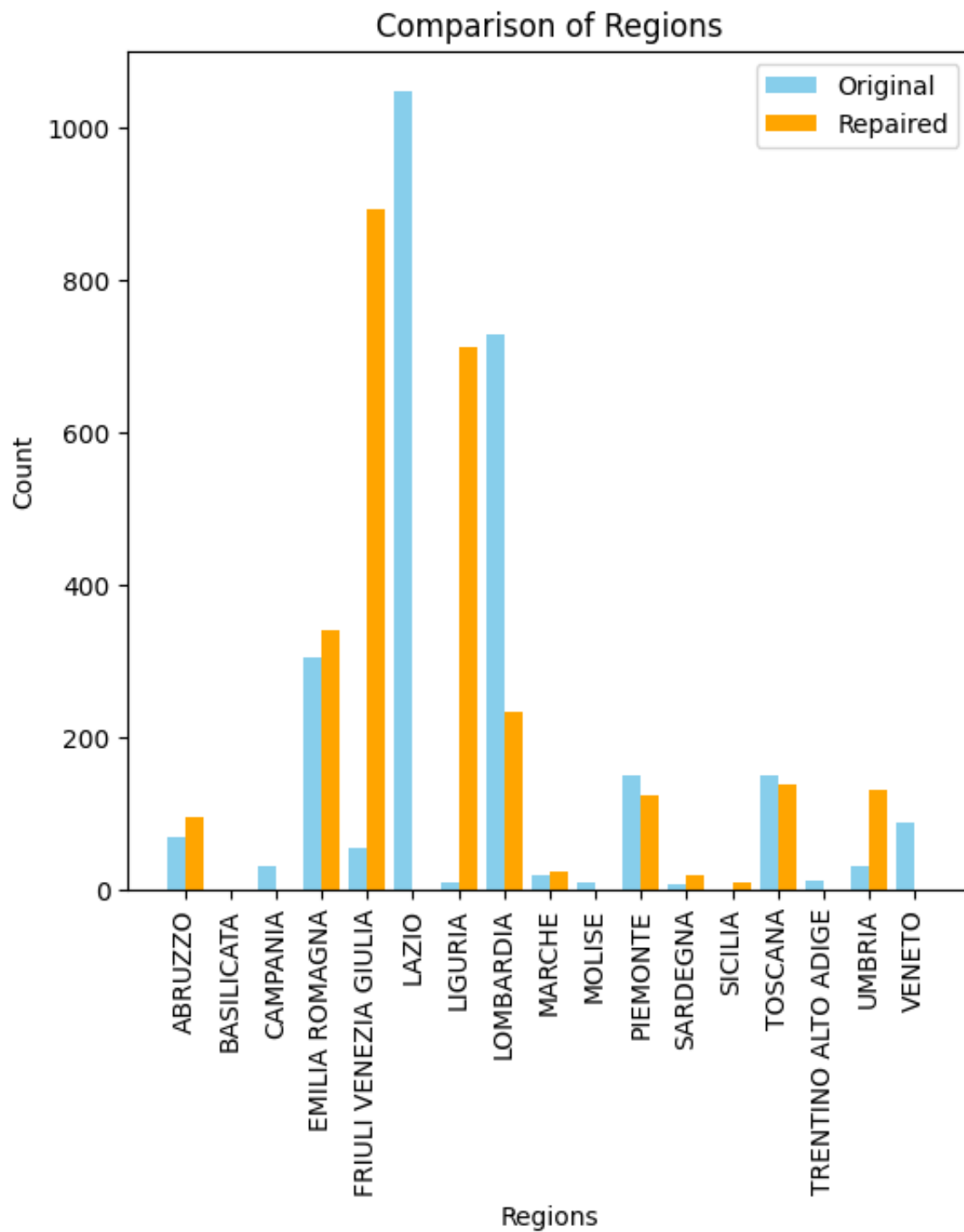
orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repa
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Job Provinces',

```



```
orig_counts = job_df_orig_age['cand_domicile_region_str'].value_counts()
repaired_counts = job_df_repaired_age['cand_domicile_region_str'].value_counts()

orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repaired_counts, all_index)
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Regions', 'Region')
```



Location

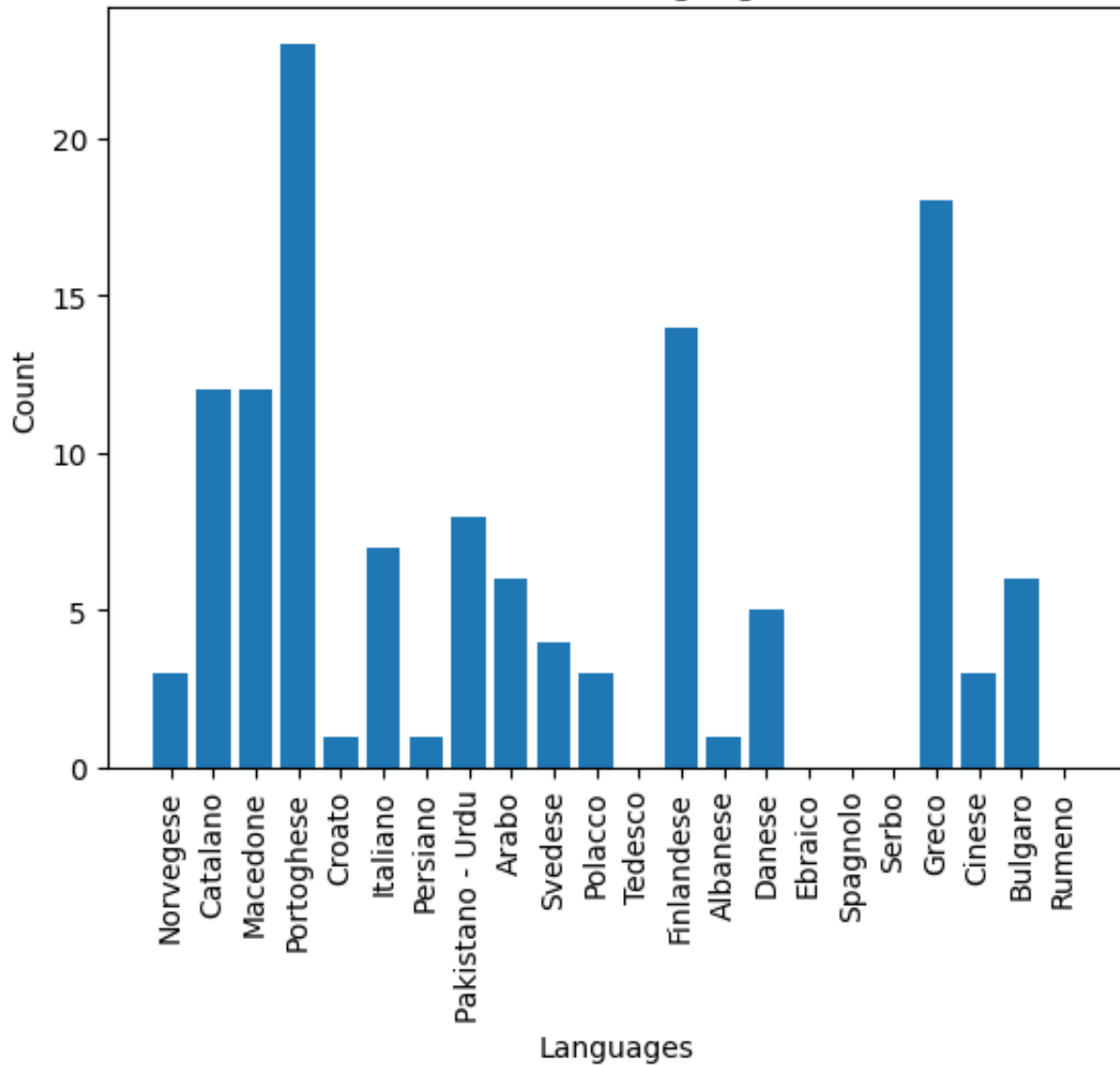
```
protected_attribute_location = "same_location"  
results_df_location = f.compute_bias_differences_percentage(df_mitigation,sectors,p  
results_df_location.to_excel('Results/Mitigation_reverse/results_df_location_revers  
results_df_location
```

	cand_gender	cand_age_bucket	cand_domicile_province	cand_domicile_region	job_contr
0	0.0	0.0	46.03	22.64	
1	0.0	0.0	76.50	74.67	
2	0.0	0.0	52.42	53.30	
3	0.0	0.0	66.73	0.89	
4	0.0	0.0	47.70	12.04	
5	0.0	0.0	77.11	22.74	
6	0.0	0.0	79.35	13.53	
7	0.0	0.0	47.44	15.58	
8	0.0	0.0	57.14	16.43	
9	0.0	0.0	53.76	16.22	
10	0.0	0.0	83.72	56.98	
11	0.0	0.0	53.50	15.32	
12	0.0	0.0	41.72	12.56	
13	0.0	0.0	72.39	16.08	
14	0.0	0.0	52.65	12.90	
15	0.0	0.0	86.32	23.55	

16 rows × 29 columns

```
results_df_location_total = f.compute_bias_differences_percentage(df_mitigation, sec
sums_location = results_df_location_total.iloc[:, first_language_index:-1].sum()
f.plot_series(sums_location, 'Removed Languages', 'Languages')
```

Removed Languages



```

job_df_orig_location, job_df_repaired_location = f.compute_repaired_df(df_mitigatio
job_df_orig_location['job_work_province_str'] = job_df_orig_location['job_work_prov
job_df_repaired_location['job_work_province_str'] = job_df_repaired_location['job_w

job_df_orig_location['cand_domicile_province_str'] = job_df_orig_location['cand_dom
job_df_repaired_location['cand_domicile_province_str'] = job_df_repaired_location['

job_df_orig_location['cand_domicile_region_str'] = job_df_orig_location['cand_domic
job_df_repaired_location['cand_domicile_region_str'] = job_df_repaired_location['ca

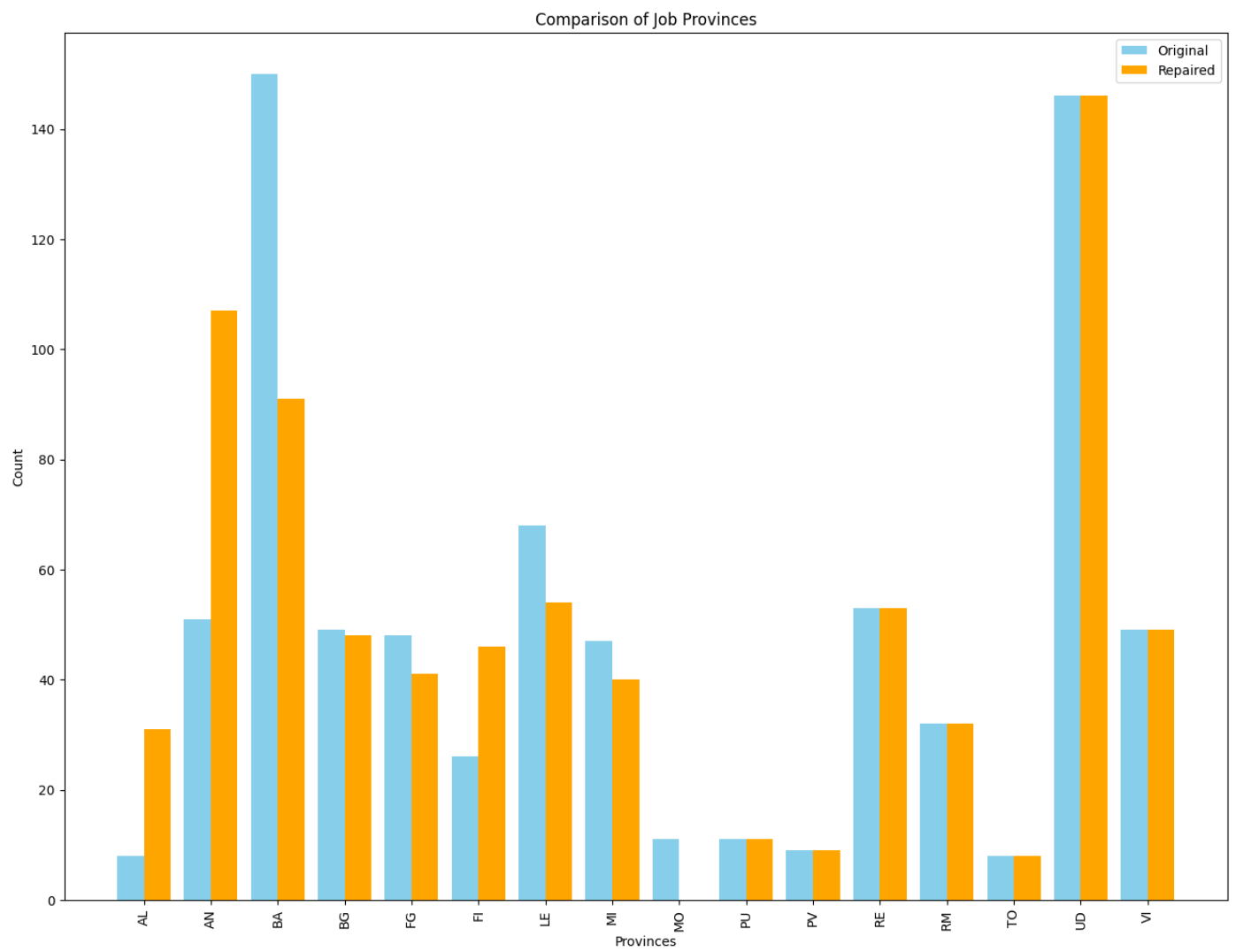
```

```

orig_counts = job_df_orig_location['job_work_province_str'].value_counts()
repaired_counts = job_df_repaired_location['job_work_province_str'].value_counts()

orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repa
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Job Provinces',

```



```
orig_counts = job_df_orig_location['cand_domicile_region_str'].value_counts()
repaired_counts = job_df_repaired_location['cand_domicile_region_str'].value_counts()

orig_counts, repaired_counts, all_index = f.prepare_different_series(orig_counts, repaired_counts, all_index)
f.compare_plot(orig_counts, repaired_counts, all_index, 'Comparison of Regions', 'Region')
```

Comparison of Regions

