Campione : Armenia

Preparazione del piano di lavoro e importazione dati

In [1]:
```
# importazione dei moduli necessari
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# importazione del file (scaricabile da qui: https://www.worldvaluessurvey.org/w
data = pd.read_csv("/content/WVS_Cross-National_Wave_7_csv_v6_0.csv", low_memory
data.shape
```

Out[2]: (97220, 613)

Pulizia del dataset

In [3]:
```
# variabili da selezionare
generali = ['A_WAVE', 'A_YEAR','Q_MODE', 'H_URBRURAL','B_COUNTRY']
var_dipendenti = ['Q235', 'Q192', 'Q194']
cont_conformismo_dipendenza = ['Q27','Q37','Q38','Q108']
cat_conformismo_dipendenza = ['Q7','Q8','Q17']
cont_anger_hate_moral_aut = ['Q46', 'Q176']
cont_self_def_paranoia = ['Q62','Q63']
cat_self_def_paranoia = ['Q42','Q57', 'Q124', 'Q150']
cat_identity = ['Q97R','Q98R','Q99R','Q101R', 'Q105R', 'Q173']
cat_values = ['Q9','Q10', 'Q12','Q13','Q14', 'Q16','Q17', 'Q29']
cat_etnia = ['Q290']
cat_equality = ['Q149', 'Q150']
cont_uso_media_per_info = ['Q201','Q202','Q203','Q204', 'Q205', 'Q206','Q207']
cont_network = ['Q2','Q58', 'Q60', 'Q94', 'Q200', 'Q208', 'Q238', 'Q239']
cont_narrative = ['Q2', 'Q4', 'Q6','Q42', 'Q50','Q121', 'Q156']
cont_needs = ['Q48', 'Q49','Q254']
cont_social_injustice = ['Q48','Q112','Q113','Q120','Q224','Q225', 'Q227', 'Q232
cont_inequalit_power = ['Q251']
cont_credenza_non_immorale = ['Q191']
cont_extremism_ideology_secular = ['Q158','Q246','Q250']
cont_extremism_ideology_religion = ['Q6','Q160','Q161', 'Q164', 'Q169', 'Q170']
cont_values = ['Q20', 'Q30', 'Q32', 'Q33', 'Q37', 'Q38', 'Q108','Q109', 'Q110','
               'Q188','Q189','Q195', 'Q240']
cont_threat = ['Q131', 'Q152']

lista_gen = (generali + var_dipendenti + cont_conformismo_dipendenza + cat_confo
             cont_self_def_paranoia + cat_self_def_paranoia + cat_identity + cat
          cont_network + cont_narrative + cont_needs + cont_social_injustice + c
          cont_extremism_ideology_secular + cont_extremism_ideology_religion + c
```

In [4]:
```
# verifico esistenza di duplicati tra le liste
from collections import Counter
label_counts = Counter(lista_gen)
# Trovare le etichette duplicate
duplicates = [label for label, count in label_counts.items() if count > 1]
```

```
# Risultato
print("Item duplicati:", duplicates)
```

```
Item duplicati: ['Q37', 'Q38', 'Q108', 'Q17', 'Q176', 'Q42', 'Q150', 'Q2', 'Q6',
'Q48']
```

In [5]:
```
# elimino i duplicati, una sola occorrenza per ciascun duplicato
for item in duplicates:
    if item in lista_gen:
        lista_gen.remove(item)
```

In [6]:
```
# Verifica della presenza residua degli elementi duplicati
presenza_residua = {item: lista_gen.count(item) == 1 for item in duplicates}
print("Elementi duplicati ora sono unici?\n", presenza_residua)
```

```
Elementi duplicati ora sono unici?
 {'Q37': True, 'Q38': True, 'Q108': True, 'Q17': True, 'Q176': True, 'Q42': True,
'Q150': True, 'Q2': True, 'Q6': True, 'Q48': True}
```

In [7]:
```
# estraggo le variabili di interesse riducendo dataset
lista_var = []
for i in lista_gen:
        lista_var.append(i)
data = data[lista_var]    # riduco le dimensioni del dataset
data.shape #verifico riduzione dimensioni
```

Out[7]: (97220, 98)

In [8]:
```
# seleziono solo dati dal campione B_COUNTRY: ARMENIA (51).
df = data[data['B_COUNTRY'] == 51]
# elimino le colonne che non mi servono
df = df.drop(labels =['A_WAVE','A_YEAR', 'Q_MODE', 'H_URBRURAL','B_COUNTRY'], ax
# verifico riduzione dimensionalità
df.shape
```

Out[8]: (1223, 93)

Pulizia dei dati: Missing values : -1 , -2, -4, -5

Colonne

In [9]:
```
# Identifico le colonne che presentano valori missing, o di domanda non fatta, o
cercati1, cercati2, cercati3, missing    = [-1], [-2], [-4], [-5]
# Crea una maschera per individuare le colonne che contengono almeno uno dei val
maschera1 , maschera2, maschera3, maschera4 = df.isin(cercati1).any(axis=0), df.

# Ne ottengo i nomi
valori1 = df.columns[maschera1]
valori2 = df.columns[maschera2]
valori3 = df.columns[maschera3]
colmissing = df.columns[missing]
print(
    'N. colonne con valori:\n -1: ', len(valori1),
     '\n -2: ', len(valori2),
```

```
     '\n -4: ', len(valori3),
     '\n -5: ', len(colmissing))
```

```
N. colonne con valori:
 -1:  88
 -2:  85
 -4:  0
 -5:  1
```

In [10]:
```
# Esistono colonne costituite interamente da un solo valore missing?
valori_cercati = {-1, -2, -4, -5}
# crea dizionario che associa ogni colonna del df a un insieme dei valori unici
# ma solo se i valori unici della colonna sono tra uno dei valori cercati
colonne_relative = {
    colonna: set(df[colonna].unique()) # ottengo valori unici presenti nella col
    for colonna in df.columns# itero tra colonne del df
    if set(df[colonna].unique()).issubset(valori_cercati)  # considera solo colo
}
# ciclo per identificare colonne
if colonne_relative:
    for colonna, valori in colonne_relative.items():
        print("Colonne con valori che sono solo {-1, -2, -4, -5}:")
        print(f"Colonna: {colonna}, Valori unici: {valori}")
else:
    print("Non ci sono colonne che presentano esclusivamente valori -1, -2, -4,
```

```
Non ci sono colonne che presentano esclusivamente valori -1, -2, -4, -5.
```

Righe

In [11]:
```
# faccio la stessa cosa, ma con le righe
# Crea una maschera per individuare le righe che contengono almeno uno dei valor
maschera1 , maschera2, maschera3, maschera4 = df.isin(cercati1).any(axis=1), df.
# le conto, sommando i valori True
print(
    'N. righe con valori:\n -1: ', maschera1.sum(),
     '\n -2: ', maschera2.sum(),
     '\n -4: ', maschera3.sum(),
     '\n -5: ', maschera4.sum())
```

```
N. righe con valori:
 -1:  861
 -2:  177
 -4:  0
 -5:  0
```

In [12]:
```
# elimino le righe che hanno valori -1, -2 e -4 e -5 verifico dimensioni dataset
df = df[~df.isin([-1, -2, -4, -5]).any(axis=1)]
df.shape
```

Out[12]:
```
(332, 93)
```

In [13]:
```
# verifico eventuale presenza di costanti implicite: colonne con un valore unico
colonne_costanti = [colonna for colonna in df.columns if df[colonna].nunique() =
print("Colonne costanti:", colonne_costanti)
```

```
Colonne costanti: []
```

Rinomino le variabili

In [14]:
```
# e rinomino tutte le variabili
dizionario_vars = {
 'Q235': 'strong_leader',
 'Q192': 'justifiable_terrorism',
 'Q194': 'justifiable_pol_viol',
 'Q27': 'goal_life_parent_proud',
 'Q108': 'gov_vs_ind_responsbility',
 'Q7': 'important_child_good_mann',
 'Q8': 'important_child_independence',
 'Q17': 'important_child_obedience',
 'Q46': 'happiness',
 'Q62': 'trust_ppl_other_rel',
 'Q63': 'trust_ppl_other_nat',
 'Q57': 'trust_most_ppl',
 'Q124': 'immigration_increase_crime',
 'Q150': 'import_freedom_or_security',
 'Q97R': 'membership_labour_union',
 'Q98R': 'membership_pol_party',
 'Q99R': 'membership_environ_org',
 'Q101R':'membership_chari_huma',
 'Q105R': 'membership_other_org',
 'Q173': 'religious_person',
 'Q9': 'important_child_hard_work',
 'Q10': 'important_child_responsibility',
 'Q12': 'important_child_tolerance_respect',
 'Q13': 'important_child_savings',
 'Q14': 'important_child_determination_perseverance',
 'Q16': 'important_unselfishness',
 'Q149': 'import_freedom_equality',
 'Q201': 'info_source_newspaper',
 'Q202': 'info_source_tv',
 'Q203': 'info_source_radio',
 'Q204': 'info_source_mobile',
 'Q205': 'info_source_email',
 'Q206': 'info_source_internet',
 'Q207': 'info_source_socia_media',
 'Q58': 'trust_family',
 'Q60': 'trust_ ppl_know_personally',
 'Q94': 'membership_church_religious_org',
 'Q200': 'often_discuss_pol_friends',
 'Q208': 'info_source_friends_collegues',
 'Q238': 'valut_dem_pol_system',
 'Q239': 'valut_rel_pol_system',
 'Q2':'importance_friends',
 'Q4': 'importance_politics',
 'Q42': 'Basic_att_concerning_society',
 'Q50': 'satisfaction_financial',
 'Q121': 'impact_immigration_country',
 'Q156': 'most_impo_for_country',
 'Q48': 'freedom_of_choice_control',
 'Q49': 'life_satisfactio',
 'Q254': 'nation_pride',
 'Q290': 'racial_belonging',
 'Q112': 'perception_corruption_country',
 'Q113': 'corruption_stat_auth',
 'Q120': 'risk_accountable_bribe',
 'Q224': 'votes_count_fairly',
 'Q225': 'opposition_imped_running',
 'Q227': 'voter_bribed',
```

```
 'Q232': 'voter_choice',
 'Q234A': 'pol_system_allows_say',
 'Q252': 'satisfactio_with_pol_syst_perform',
 'Q251': 'democratic_country',
 'Q191': 'justifiable_violence_other_ppl',
 'Q158': 'eval_science_tech',
 'Q246': 'demo_must_protect_rights',
 'Q250': 'importance_democracy',
 'Q6': 'importance_religion',
 'Q160': 'dependence_science_faith',
 'Q161': 'science_ppl_no_moral',
 'Q164': 'importance_god',
 'Q169': 'religion_always_right',
 'Q170': 'only_acc_my_rel',
 'Q20': 'neigh_ppl_aids',
 'Q30': 'univ_more_boy',
 'Q29': 'men_better_pol',
 'Q32': 'housewife_fulfilling',
 'Q33': 'job_men_more_rights',
 'Q37': 'children_duty_society',
 'Q38': 'children_parent_duty' ,
 'Q109': 'competition_good',
 'Q110': 'success_hardwork_luck',
 'Q176': 'trouble_moral_values',
 'Q179': 'justifiable_stealing',
 'Q180': 'justifiable_cheating_tax',
 'Q182': 'justifiable_homosex',
 'Q183': 'justifiable_prostitution',
 'Q184': 'justifiable_abortion',
 'Q185':'justifiable_divorce',
 'Q188': 'justifiable_euthanasia',
 'Q189': 'justifiable_beat_wife',
 'Q195': 'justifiable_death_penal',
 'Q240': 'Or_pol',
 'Q131': 'security_neigh',
 'Q152': 'aims_of_country'}

# riassegno i nomi nel dataset (anche per df da usare per ordinal regression)
df = df.rename(columns= dizionario_vars)
```

In [15]:

```
# creo Markdown con range per ogni variabile
# Inizializza una lista per contenere le righe di Markdown
markdown_lines = []
for column in df.columns:
    markdown_lines.append(f"**{column}**: left - right")
markdown_text = "\n".join(markdown_lines)
print(markdown_text)
```

```
**strong_leader**: left - right
**justifiable_terrorism**: left - right
**justifiable_pol_viol**: left - right
**goal_life_parent_proud**: left - right
**important_child_good_mann**: left - right
**important_child_independence**: left - right
**happiness**: left - right
**trust_ppl_other_rel**: left - right
**trust_ppl_other_nat**: left - right
**trust_most_ppl**: left - right
**immigration_increase_crime**: left - right
**membership_labour_union**: left - right
**membership_pol_party**: left - right
**membership_environ_org**: left - right
**membership_chari_huma**: left - right
**membership_other_org**: left - right
**religious_person**: left - right
**racial_belonging**: left - right
**important_child_hard_work**: left - right
**important_child_responsibility**: left - right
**important_child_tolerance_respect**: left - right
**important_child_savings**: left - right
**important_child_determination_perseverance**: left - right
**important_unselfishness**: left - right
**important_child_obedience**: left - right
**men_better_pol**: left - right
**import_freedom_equality**: left - right
**import_freedom_or_security**: left - right
**info_source_newspaper**: left - right
**info_source_tv**: left - right
**info_source_radio**: left - right
**info_source_mobile**: left - right
**info_source_email**: left - right
**info_source_internet**: left - right
**info_source_socia_media**: left - right
**trust_family**: left - right
**trust_ ppl_know_personally**: left - right
**membership_church_religious_org**: left - right
**often_discuss_pol_friends**: left - right
**info_source_friends_collegues**: left - right
**valut_dem_pol_system**: left - right
**valut_rel_pol_system**: left - right
**importance_friends**: left - right
**importance_politics**: left - right
**Basic_att_concerning_society**: left - right
**satisfaction_financial**: left - right
**impact_immigration_country**: left - right
**most_impo_for_country**: left - right
**life_satisfactio**: left - right
**nation_pride**: left - right
**freedom_of_choice_control**: left - right
**perception_corruption_country**: left - right
**corruption_stat_auth**: left - right
**risk_accountable_bribe**: left - right
**votes_count_fairly**: left - right
**opposition_imped_running**: left - right
**voter_bribed**: left - right
**voter_choice**: left - right
**pol_system_allows_say**: left - right
**satisfactio_with_pol_syst_perform**: left - right
```

```
**democratic_country**: left - right
**justifiable_violence_other_ppl**: left - right
**eval_science_tech**: left - right
**demo_must_protect_rights**: left - right
**importance_democracy**: left - right
**importance_religion**: left - right
**dependence_science_faith**: left - right
**science_ppl_no_moral**: left - right
**importance_god**: left - right
**religion_always_right**: left - right
**only_acc_my_rel**: left - right
**neigh_ppl_aids**: left - right
**univ_more_boy**: left - right
**housewife_fulfilling**: left - right
**job_men_more_rights**: left - right
**children_duty_society**: left - right
**children_parent_duty**: left - right
**gov_vs_ind_responsbility**: left - right
**competition_good**: left - right
**success_hardwork_luck**: left - right
**trouble_moral_values**: left - right
**justifiable_stealing**: left - right
**justifiable_cheating_tax**: left - right
**justifiable_homosex**: left - right
**justifiable_prostitution**: left - right
**justifiable_abortion**: left - right
**justifiable_divorce**: left - right
**justifiable_euthanasia**: left - right
**justifiable_beat_wife**: left - right
**justifiable_death_penal**: left - right
**Or_pol**: left - right
**security_neigh**: left - right
**aims_of_country**: left - right
```

Variabili considerate - Item e livelli di risposta

Di seguito i predittori selezionati.

Quando la variabile è su scala ordinale verranno presentati solo il minimo e il massimo; quando su scala ordinale, verrà indicato ogni valore presente

Variabile responso: **justifiable_terrorism**: Justifiable - Terrorism as a political, ideological or religious mean: 1.Never justifiable - 10.Always justifiable

Predittori

- **strong_leader**: Having a strong leader who doeas not have to bother with parliament and elections: 1.Very good - 4.Very bad;
- **justifiable_pol_viol**: Justifiable - Political violence: 1.Never justifiable - 10.Always justifiable;

- **goal_life_parent_proud**: One of main goal in life has been to make my parents proud: 1.Agree strongly - 4.Strongly disagree;
- **important_child_good_mann**: Important child qualities - Good manners: 2.Not mentioned - 1.Important;
- **important_child_independence**: Important child qualities - Independence: 2.Not mentioned - 1.Important;
- **happiness**: Feeling of happiness: 1.Very happy - 4.Not at all happy;
- **trust_ppl_other_rel**: Trust - People of other religion: 1.Trust completely - 4.Do not trust at all;
- **trust_ppl_other_nat**: Trust - People of other nationality: 1.Trust completely - 4.Do not trust at all;
- **trust_most_ppl**: Most people can be trusted: 1.Most people can be trusted - 2.Need to be very careful;
- **immigration_increases_crime**: Immigration in your country - Increases the crime rate: 0.Disagree - 2.Agree;
- **membership_labour_union**: Membership - Labour union: 0.Not mentioned (do not belong) - 1.Mentioned (member);
- **membership_pol_party**: Membership - Political party: 0.Not mentioned (do not belong) - 1.Mentioned (member);
- **membership_environ_org**: Membership - Environmental organization union: 0.Not mentioned (do not belong) - 1.Mentioned (member);
- **membership_chari_huma**: Membership - charitable/humanitarian organization: 0.Not mentioned (do not belong) - 1.Mentioned (member);
- **membership_other_org**: Membership - Other organization: 0.Not mentioned (do not belong) - 1.Mentioned (member);
- **religious_person**: Religious person: 1 - A religious person, 3 - An atheist;
- **racial_belonging**: 51001.Armenian, 51005.Russian, 51006.Yazidis, 51007.Other;
- **important_child_hard_work**: Important child qualities - Hard work: 2.Not mentioned - 1.Important;
- **important_child_responsibility**: Important child qualities - Feeling of responsibility: 2.Not mentioned - 1.Important;
- **important_child_tolerance_respect**: Important child qualities - Tolerance and respect for other people: 2.Not mentioned - 1.Important;
- **important_child_savings**: Important child qualities - Thrift saving money and things: 2.Not mentioned - 1.Important;
- **important_child_determination_perseverance**: Important child qualities - Determination perseverance: 2.Not mentioned - 1.Important;
- **important_unselfishness**: Important child qualities - Unselfishness: 2.Not mentioned - 1.Important;
- **important_child_obedience**: Important child qualities - Obedience: 2.Not mentioned - 1.Important;
- **import_freedom_equality**: Freedom and equality - Which more important: 1.Freedom - 2.Equality;
- **import_freedom_or_security**: Freedom and security - Which more important: 1.Freedom - 2.Security;

-**info_source_newspaper**: Information source: Daily newspaper: 1.Daily - 5.Never;

- **info_source_tv**: Information source: Tv news: 1.Daily - 5.Never;
- **info_source_radio**: Information source: Radio News: 1.Daily - 5.Never;
- **info_source_mobile**: Information source: Mobile phone: 1.Daily - 5.Never;
- **info_source_email**: Information source: Email: 1.Daily - 5.Never;
- **info_source_internet**: Information source: Internet: 1.Daily - 5.Never;
- **info_source_socia_media**: Information source: Social media (Facebook, Twitter etc.): 1.Daily - 5.Never;
- **trust_family**: :How much you trust - Your family: 1.Trust completely - 4.Do not trust at all;
- **trust_ ppl_know_personally**: How much you trust - People you know personally: 1.Trust completely - 4.Do not trust at all;
- **membership_church_religious_org**: Active/inactive membership - Church or religious organization: 0.Don't belong - 2.Active member;
- **often_discuss_pol_friends**: How often discusses political matters with friends: 1.Frequently - 3.Never;
- **info_source_friends_collegues**: Information source: Talk with friends or colleagues: 1.Daily - 5.Never;
- **valut_dem_pol_system**: Political system - Having a democratic political system: 1.Very good - 4.Very bad;
- **valut_rel_pol_system**: Political system - Having a system governed by religious law in which there are no political parties or elections: 1.Very good - 4.Very bad;
- **importance_friends**: Important in life - Friends: 1.Very important - 4.Not at all important;
- **importance_democracy**: Importance of democracy: 1.Not at all important - 10.Absolutely important;
- **importance_religion**: Important in life - Religion: 1.Very important - 4.Not at all;
- **importance_politics**: Important in life - Politics: 1.Very important - 4.Not at all important;
- **Basic_att_concerning_society**: Basic kinds of attitude concerning society: 1.The entire way our society is organized must be radically changed by revolutionary action, 2.Our society must be gradually improved by reforms, 3.Our present society must be valiantly defended against all subversive forces;
- **satisfaction_financial**: Satisfaction with financial situation of household: 1.Dissatisfied - 10.Satisfied;
- **impact_immigration_country**: Impact of immigrants on the development of the country: 1.Rather bad - 5.Very good;
- **most_impo_for_country**: Most important (for country): 1. A stable economy, 2.Progress toward a less impersonal and motre humane society, 3.Progress toward society in which ideas count more than money, 4.The fight against crime;
- **life_satisfactio**: Satisfaction with your life: 1.Completely dissatisfied - 10.Completely satisfied;
- **nation_pride**: National pride: 1.Very proud - 5.I am not ...(nationality);
- **freedom_of_choice_control**: How much freedom of choice and control: 1.None at all - 10.A great deal;

- **perception_corruption_country**: Perceptions of corruption in the country: 1 - 10;
- **corruption_stat_auth**: Involved in corruption - State authorities: 1.None of them - 4.All of them;
- **risk_accountable_bribe**: Risk to be held accountable for giving or receiving bribe: 1.No risk at all - 10.Very high risk;
- **votes_count_fairly**: how often in country's elections - Votes are counted fairly: 1.Very often - 4.Not at all;
- **opposition_imped_running**: How often in country's elections - Opposition candidates are prevented from running: 1.Very often - 4.Not at all;
- **voter_bribed**: How often in country's elections - Voters are bribed: 1.Very often - 4.Not at all often;
- **voter_choice**: How often in country's elections - Voters are offered a genuine choice in elections: 1.Very often - 4.Not at all;
- **pol_system_allows_say**: How much the political system allows people to have a say: 1.A great deal - 5.Not at all;
- **satisfactio_with_pol_syst_perform**: Satisfaction with the political system performance: 1.Not satisfied at all - 10.Completely satisfied;
- **democratic_country**: How democratically is this country being governed today?: 1.Not at all democratic - 10.Completely democratic;
- **justifiable_violence_other_ppl**: Justifiable - Violence against other people: 1.Never justifiable - 10.Always justifiable;
- **eval_science_tech**: Science and technology are making our lives healthier, easier, and more comfortable: 1.Completely disagree - 10.Completely agree
- **demo_must_protect_rights**: Civil rights protect people's liberty against oppression: 0.it is against democracy (spontaneous) - 10.An essential characteristic of democracy;
- **dependence_science_faith**: We depend too much on science and not enough on faith: 1.Completely Disagree - 10.Completely agree;
- **science_ppl_no_moral**: One of the effects of science is that it breaks down people's idea of right and wrong: 1.Completely disagree - 10.Completely agree_faith;
- **importance_god**: Importance of God: 1.Not at all important - 10.Very important;
- **religion_always_right**: Whenever science and religion conflict, religion is always right: 1.Strongly agree - 4.Strongly disagree;
- **only_acc_my_rel**: The only acceptable religion is my religion: 1.Strongly agree - 4.Strongly disagree;
- **man_better_pol**: Neighbours - People who have AIDS: 1.Mentioned - 2.Not mentioned;
- **univ_more_boy**: University is more important for a boy than for a girl: 1.Agree strongly - 4.Strongly disagree;
- **men_better_pol**: Men make better political leaders than women do: 1.Agree strongly - 4.Strongly disagree;
- **housewife_fulfilling**: Being a housewife just as fulfilling: 1.Agree strongly - 4.Strongly disagree;
- **job_men_more_rights**: Job scarce - Men should have more right to a job than women: 1.Agree - 3.Neither agree nor disagree;

- **children_duty_society**: Duty toward society to have children: 1.Agree strongly - 5.Disagree strongly;
- **children_parent_duty**: It is children duty to take care of ill parents: 1.Agree strongly - 5.Disagree strongly;
- **gov_vs_ind_responsbility**: Government's vs individual's responsibility: 1.The government should take more responsibility to ensure that everyone is provided for - 10.People should take more responsibility to provide for themselves;
- **competition_good**: Competition good or harmful: 1.Competition is good - 10.Competition is harmful;
- **success_hardwork_luck**: Success - Hard work vs luck: 1.In the long run, hard work usually brings a better life - 10.Hard work doesn't generally bring success. It is more a matter of luck and connections;
- **justifiable_iolence_other_ppl**: Justifiable - Violence against other people: 1.Never justifiable - 10.Always justifiable;
- **trouble_moral_values**: Degree of agreement - Nowadays one often has trouble deciding which moral rules are the right ones to follow: 1.Competely agree - 10.Completely disagree;
- **justifiable_stealing**: Justifiable - Stealing property: 1.Never justifiable - 10.Always justifiable;
- **justifiable_cheating_tax**: Justifiable - Cheating on taxes: 1.Never justifiable - 10.Always justifiable;
- **justifiable_homosex**: Justifiable - Homosexuality: 1.Never justifiable - 10.Always justifiable;
- **justifiable_prostitution**: Justifiable - Prostitution: 1.Never justifiable - 10.Always justifiable;
- **justifiable_abortion**: Justifiable - Abortion: 1.Never justifiable - 10.Always justifiable;
- **justifiable_divorce**: Justifiable - Divorce: 1.Never justifiable - 10.Always justifiable;
- **justifiable_euthanasia**: Justifiable - Euthanasia: 1.Never justifiable - 10.Always justifiable;
- **justifiable_pol_viol**: Justifiable - Political violence: 1.Never justifiable - 10.Always justifiable;
- **justifiable_beat_wife**:Justifiable - For a man to beat his wife: 1.Never justifiable - 10.Always justifiable;
- **justifiable_death_penal**: Justifiable - Death penalty: 1.Never justifiable - 10.Always justifiable;
- **Or_pol**: Left-righ political scale: 1.Left - 10.Right;
- **security_neigh**: Secure in neighborhood: 1.Very secure - 4.Not at all secure;
- **aims_of_country**: Aims of country: 1.A high level of economic growth, 2.Strong defence forces, 3.People have more say about how things are done, 4.Trying to make our cities and countryside more beautiful.

Strategia analitica: considerare variabili con valori unici ordinati >=5 come variabili continue

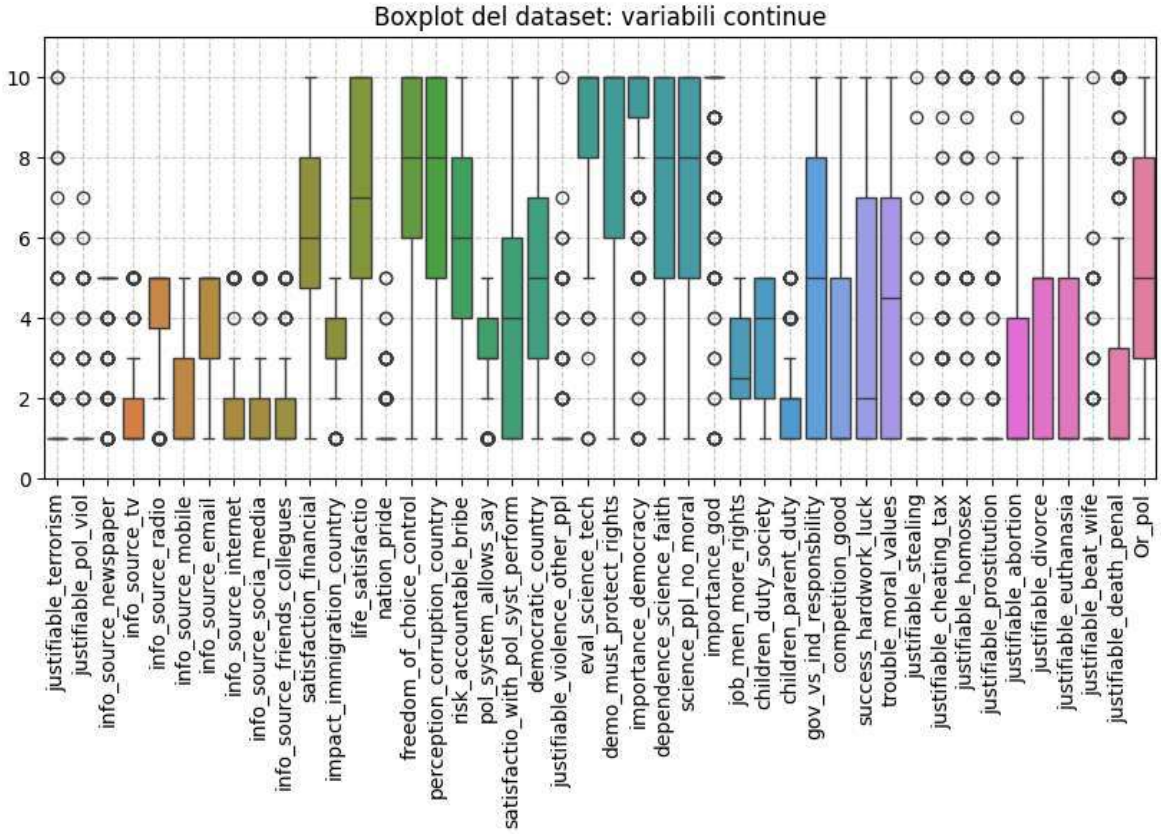
Le altre saranno considerate categoriali

In [16]:
```
# Lista delle colonne con 5 o più valori unici
colonne_lineari = [colonna for colonna in df.columns if df[colonna].nunique() >=
print("Colonne con 5 o più valori unici:", colonne_lineari)
```

```
Colonne con 5 o più valori unici: ['justifiable_terrorism', 'justifiable_pol_vio
l', 'info_source_newspaper', 'info_source_tv', 'info_source_radio', 'info_source_
mobile', 'info_source_email', 'info_source_internet', 'info_source_socia_media',
'info_source_friends_collegues', 'satisfaction_financial', 'impact_immigration_co
untry', 'life_satisfactio', 'nation_pride', 'freedom_of_choice_control', 'percept
ion_corruption_country', 'risk_accountable_bribe', 'pol_system_allows_say', 'sati
sfactio_with_pol_syst_perform', 'democratic_country', 'justifiable_violence_other
_ppl', 'eval_science_tech', 'demo_must_protect_rights', 'importance_democracy',
'dependence_science_faith', 'science_ppl_no_moral', 'importance_god', 'job_men_mo
re_rights', 'children_duty_society', 'children_parent_duty', 'gov_vs_ind_responsb
ility', 'competition_good', 'success_hardwork_luck', 'trouble_moral_values', 'jus
tifiable_stealing', 'justifiable_cheating_tax', 'justifiable_homosex', 'justifiab
le_prostitution', 'justifiable_abortion', 'justifiable_divorce', 'justifiable_eut
hanasia', 'justifiable_beat_wife', 'justifiable_death_penal', 'Or_pol']
```

Distribuzione delle variabili con valori unici >= 5

In [17]:
```
# boxplot
plt.figure(figsize=(10, 4))
sns.boxplot(data=df[colonne_lineari])
plt.xticks(rotation=90)
plt.grid(True, linestyle='--', alpha=0.6)
plt.ylim(df[colonne_lineari].min().min() - 1, df[colonne_lineari].max().max() +
plt.title("Boxplot del dataset: variabili continue")
plt.show()
```



In [18]:
```
# valori di skew delle variabili continue
skew_values = df[colonne_lineari].skew()
# seleziono valori di skew >1 e <- 1
skew = skew_values[(skew_values > 1) | (skew_values < -1)]
skew
```

Out[18]:

	0
justifiable_terrorism	5.133510
justifiable_pol_viol	3.904743
info_source_newspaper	-1.660822
info_source_tv	1.577456
info_source_radio	-1.346341
info_source_mobile	1.132248
info_source_email	-1.071732
info_source_internet	1.457734
info_source_socia_media	1.494345
info_source_friends_collegues	1.631590
nation_pride	2.941156
justifiable_violence_other_ppl	3.520756
eval_science_tech	-1.654599
demo_must_protect_rights	-1.214696
importance_democracy	-2.247657
importance_god	-2.675865
children_parent_duty	1.218140
competition_good	1.468945
justifiable_stealing	7.489589
justifiable_cheating_tax	3.033294
justifiable_homosex	3.608596
justifiable_prostitution	3.274179
justifiable_abortion	1.465725
justifiable_euthanasia	1.417452
justifiable_beat_wife	4.959937
justifiable_death_penal	1.569918

dtype: float64

In [19]:
```
# riduco i valori di skew utilizzando modulo PowerTransformer di sklearn e appli
# sia avalori di skew positivi che a valori negativi
```

```
from sklearn.preprocessing import PowerTransformer
transformer = PowerTransformer(method='yeo-johnson', standardize=True)
df[skew.index] = transformer.fit_transform(df[skew.index])
df[skew.index].skew().sort_values()
```

Out[19]:

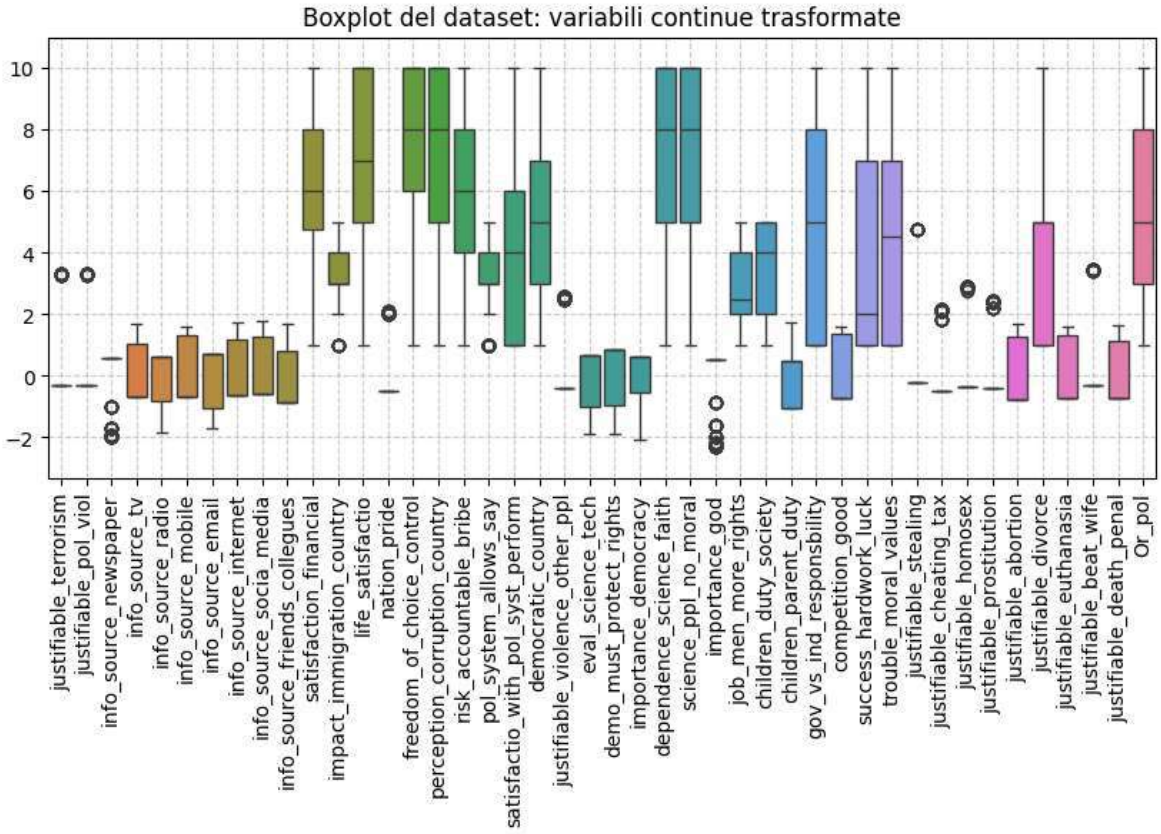
	0
importance_god	-1.560229
info_source_newspaper	-1.303329
importance_democracy	-1.167254
info_source_radio	-1.111538
eval_science_tech	-0.913059
info_source_email	-0.853958
demo_must_protect_rights	-0.689032
children_parent_duty	0.226938
info_source_friends_collegues	0.461178
justifiable_abortion	0.549690
justifiable_euthanasia	0.651520
justifiable_death_penal	0.723394
competition_good	0.733492
info_source_tv	0.786222
info_source_mobile	0.804835
info_source_internet	1.022675
info_source_socia_media	1.082138
nation_pride	1.565060
justifiable_cheating_tax	1.622181
justifiable_prostitution	1.966281
justifiable_violence_other_ppl	2.178201
justifiable_homosex	2.530652
justifiable_pol_viol	3.005137
justifiable_terrorism	3.005150
justifiable_beat_wife	3.153413
justifiable_stealing	4.576833

dtype: float64

In [20]:

```
# boxplot con variabili trasformate
plt.figure(figsize=(10, 4))
sns.boxplot(data=df[colonne_lineari])
```

```
plt.xticks(rotation=90)
plt.grid(True, linestyle='--', alpha=0.6)
plt.ylim(df[colonne_lineari].min().min() - 1, df[colonne_lineari].max().max() +
plt.title("Boxplot del dataset: variabili continue trasformate")
plt.show()
```



One-hot encoding di ogni altra variabile ad esclusione delle variabili continue

In [21]:
```
# seleziono tutte le colonne del df ad esclusione di quelle continue e applico o
colonne_non_lineari = df.columns.difference(colonne_lineari)
df = pd.get_dummies(df, prefix_sep='_', columns=colonne_non_lineari,
                    drop_first=True,  # Droppa la prima categoria per evitare mu
                    dtype=float)      # Tipo di dato, per regressione lineare
```

In [22]:
```
# elenco nuove dummies + loro conteggio
print('Variabili categoriali: ')
print(list(df.columns.difference(colonne_lineari)))
print('Variabili categoriali attualmente nel dataset: ', len(df.columns.differen
```

```
Variabili categoriali:
['Basic_att_concerning_society_2', 'Basic_att_concerning_society_3', 'aims_of_cou
ntry_2', 'aims_of_country_3', 'aims_of_country_4', 'corruption_stat_auth_2', 'cor
ruption_stat_auth_3', 'corruption_stat_auth_4', 'goal_life_parent_proud_2', 'goal
_life_parent_proud_3', 'goal_life_parent_proud_4', 'happiness_2', 'happiness_3',
'happiness_4', 'housewife_fulfilling_2', 'housewife_fulfilling_3', 'housewife_ful
filling_4', 'immigration_increase_crime_1', 'immigration_increase_crime_2', 'impo
rt_freedom_equality_2', 'import_freedom_or_security_2', 'importance_friends_2',
'importance_friends_3', 'importance_friends_4', 'importance_politics_2', 'importa
nce_politics_3', 'importance_politics_4', 'importance_religion_2', 'importance_re
ligion_3', 'importance_religion_4', 'important_child_determination_perseverance_
2', 'important_child_good_mann_2', 'important_child_hard_work_2', 'important_chil
d_independence_2', 'important_child_obedience_2', 'important_child_responsibility
_2', 'important_child_savings_2', 'important_child_tolerance_respect_2', 'importa
nt_unselfishness_2', 'membership_chari_huma_1', 'membership_church_religious_org_
1', 'membership_church_religious_org_2', 'membership_environ_org_1', 'membership_
labour_union_1', 'membership_other_org_1', 'membership_pol_party_1', 'men_better_
pol_2', 'men_better_pol_3', 'men_better_pol_4', 'most_impo_for_country_2', 'most_
impo_for_country_3', 'most_impo_for_country_4', 'neigh_ppl_aids_2', 'often_discus
s_pol_friends_2', 'often_discuss_pol_friends_3', 'only_acc_my_rel_2', 'only_acc_m
y_rel_3', 'only_acc_my_rel_4', 'opposition_imped_running_2', 'opposition_imped_ru
nning_3', 'opposition_imped_running_4', 'racial_belonging_51005', 'racial_belongi
ng_51006', 'religion_always_right_2', 'religion_always_right_3', 'religion_always
_right_4', 'religious_person_2', 'religious_person_3', 'security_neigh_2', 'secur
ity_neigh_3', 'security_neigh_4', 'strong_leader_2', 'strong_leader_3', 'strong_l
eader_4', 'trust_ ppl_know_personally_2', 'trust_ ppl_know_personally_3', 'trust_
ppl_know_personally_4', 'trust_family_2', 'trust_family_3', 'trust_most_ppl_2',
'trust_ppl_other_nat_2', 'trust_ppl_other_nat_3', 'trust_ppl_other_nat_4', 'trust
_ppl_other_rel_2', 'trust_ppl_other_rel_3', 'trust_ppl_other_rel_4', 'univ_more_b
oy_2', 'univ_more_boy_3', 'univ_more_boy_4', 'valut_dem_pol_system_2', 'valut_dem
_pol_system_3', 'valut_dem_pol_system_4', 'valut_rel_pol_system_2', 'valut_rel_po
l_system_3', 'valut_rel_pol_system_4', 'voter_bribed_2', 'voter_bribed_3', 'voter
_bribed_4', 'voter_choice_2', 'voter_choice_3', 'voter_choice_4', 'votes_count_fa
irly_2', 'votes_count_fairly_3', 'votes_count_fairly_4']
Variabili categoriali attualmente nel dataset:  104
```

In [23]:
```
# dimensioni attuali del dataset
df.shape
```

Out[23]:
```
(332, 148)
```

In [24]:
```
# costruisco matrice dei predittori, seleziono vd e la elimino dal df
Y = df['justifiable_terrorism']
df = df.drop(columns= ['justifiable_terrorism'])
X = df  # matrice dei predittori
```

Modello OLS

In [25]:
```
# Importo moduli necessari
import statsmodels.api as sm
from sklearn.metrics import mean_squared_error
```

In [26]:
```
# modello e fitting
X = sm.add_constant(X) # aggiungo intercetta
model =sm.OLS(Y, X)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     justifiable_terrorism   R-squared:                       0.658
Model:                               OLS   Adj. R-squared:                  0.384
Method:                    Least Squares   F-statistic:                     2.403
Date:                   Mon, 16 Dec 2024   Prob (F-statistic):           1.04e-08
Time:                           23:05:54   Log-Likelihood:                -293.21
No. Observations:                    332   AIC:                             882.4
Df Residuals:                        184   BIC:                             1446.
Df Model:                            147
Covariance Type:               nonrobust
==============================================================================
================================
                                                   coef    std err          t
P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
--------------------------------
const                                           -0.6401      0.915     -0.700
0.485      -2.445       1.165
justifiable_pol_viol                             0.2060      0.070      2.963
0.003       0.069       0.343
info_source_newspaper                            0.0751      0.060      1.255
0.211      -0.043       0.193
info_source_tv                                   0.0018      0.067      0.026
0.979      -0.131       0.134
info_source_radio                                0.0071      0.059      0.120
0.904      -0.108       0.123
info_source_mobile                              -0.0392      0.072     -0.547
0.585      -0.181       0.102
info_source_email                               -0.0584      0.066     -0.882
0.379      -0.189       0.072
info_source_internet                             0.2234      0.081      2.767
0.006       0.064       0.383
info_source_socia_media                         -0.1207      0.077     -1.561
0.120      -0.273       0.032
info_source_friends_collegues                    0.0249      0.071      0.350
0.727      -0.115       0.165
satisfaction_financial                          -0.0307      0.027     -1.154
0.250      -0.083       0.022
impact_immigration_country                       0.0397      0.068      0.584
0.560      -0.094       0.174
life_satisfactio                                 0.0082      0.029      0.286
0.775      -0.048       0.065
nation_pride                                     0.1274      0.068      1.865
0.064      -0.007       0.262
freedom_of_choice_control                       -0.0076      0.027     -0.280
0.780      -0.061       0.046
perception_corruption_country                    0.0121      0.027      0.454
0.650      -0.040       0.065
risk_accountable_bribe                          -0.0172      0.022     -0.799
0.425      -0.060       0.025
pol_system_allows_say                            0.1002      0.054      1.869
0.063      -0.006       0.206
satisfactio_with_pol_syst_perform               -0.0016      0.026     -0.062
0.951      -0.054       0.051
democratic_country                               0.0210      0.027      0.769
0.443      -0.033       0.075
justifiable_violence_other_ppl                   0.1370      0.074      1.857
0.065      -0.009       0.282
eval_science_tech                                0.0143      0.055      0.258
```

```
0.797      -0.095       0.123
demo_must_protect_rights                          0.0319      0.058      0.551
0.583      -0.082       0.146
importance_democracy                              0.0422      0.064      0.655
0.514      -0.085       0.169
dependence_science_faith                         -0.0049      0.019     -0.253
0.800      -0.043       0.033
science_ppl_no_moral                              0.0049      0.019      0.255
0.799      -0.033       0.043
importance_god                                   -0.0157      0.072     -0.219
0.827      -0.158       0.126
job_men_more_rights                               0.0091      0.042      0.219
0.827      -0.073       0.091
children_duty_society                             0.0601      0.042      1.422
0.157      -0.023       0.143
children_parent_duty                              0.0263      0.062      0.422
0.673      -0.097       0.149
gov_vs_ind_responsbility                         -0.0183      0.017     -1.101
0.272      -0.051       0.014
competition_good                                  0.0462      0.062      0.743
0.459      -0.077       0.169
success_hardwork_luck                            -0.0087      0.018     -0.495
0.621      -0.043       0.026
trouble_moral_values                              0.0078      0.018      0.429
0.669      -0.028       0.044
justifiable_stealing                             -0.0189      0.072     -0.263
0.793      -0.160       0.123
justifiable_cheating_tax                          0.1115      0.068      1.642
0.102      -0.023       0.245
justifiable_homosex                              -0.0661      0.073     -0.900
0.369      -0.211       0.079
justifiable_prostitution                         -0.0898      0.075     -1.198
0.233      -0.238       0.058
justifiable_abortion                              0.0359      0.070      0.510
0.611      -0.103       0.175
justifiable_divorce                               0.0084      0.025      0.329
0.742      -0.042       0.059
justifiable_euthanasia                           -0.0085      0.068     -0.126
0.900      -0.143       0.125
justifiable_beat_wife                             0.2904      0.063      4.597
0.000       0.166       0.415
justifiable_death_penal                          -0.0282      0.069     -0.407
0.684      -0.165       0.108
Or_pol                                            0.0204      0.020      1.005
0.316      -0.020       0.060
Basic_att_concerning_society_2                    0.0740      0.148      0.499
0.618      -0.219       0.367
Basic_att_concerning_society_3                   -0.4392      0.336     -1.308
0.193      -1.102       0.223
aims_of_country_2                                 0.0634      0.137      0.462
0.644      -0.207       0.334
aims_of_country_3                                -0.3482      0.416     -0.836
0.404      -1.170       0.473
aims_of_country_4                                 0.1309      0.190      0.689
0.491      -0.244       0.506
corruption_stat_auth_2                            0.0061      0.358      0.017
0.986      -0.701       0.713
corruption_stat_auth_3                           -0.2735      0.366     -0.746
0.456      -0.996       0.449
corruption_stat_auth_4                           -0.1387      0.394     -0.352
```

```
0.725      -0.916       0.639
goal_life_parent_proud_2                        0.0256      0.136      0.188
0.851      -0.243       0.294
goal_life_parent_proud_3                        0.2654      0.257      1.034
0.303      -0.241       0.772
goal_life_parent_proud_4                       -0.3823      0.554     -0.690
0.491      -1.475       0.711
happiness_2                                     0.0974      0.123      0.791
0.430      -0.145       0.340
happiness_3                                     0.1330      0.287      0.463
0.644      -0.434       0.700
happiness_4                                    -0.2133      0.457     -0.466
0.642      -1.116       0.689
housewife_fulfilling_2                          0.1852      0.172      1.078
0.282      -0.154       0.524
housewife_fulfilling_3                          0.2824      0.179      1.577
0.116      -0.071       0.636
housewife_fulfilling_4                         -0.0720      0.201     -0.359
0.720      -0.468       0.324
immigration_increase_crime_1                   -0.2242      0.278     -0.807
0.421      -0.772       0.324
immigration_increase_crime_2                   -0.1679      0.122     -1.380
0.169      -0.408       0.072
import_freedom_equality_2                       0.0149      0.130      0.115
0.909      -0.241       0.271
import_freedom_or_security_2                   -0.0722      0.157     -0.459
0.647      -0.383       0.238
importance_friends_2                            0.1129      0.127      0.887
0.376      -0.138       0.364
importance_friends_3                            0.0655      0.185      0.354
0.724      -0.300       0.431
importance_friends_4                            0.8511      0.396      2.148
0.033       0.069       1.633
importance_politics_2                          -0.0642      0.188     -0.341
0.733      -0.436       0.307
importance_politics_3                           0.0407      0.176      0.231
0.818      -0.307       0.389
importance_politics_4                          -0.2466      0.175     -1.405
0.162      -0.593       0.100
importance_religion_2                          -0.2603      0.159     -1.632
0.104      -0.575       0.054
importance_religion_3                          -0.1908      0.245     -0.778
0.438      -0.675       0.293
importance_religion_4                          -0.9406      0.425     -2.213
0.028      -1.779      -0.102
important_child_determination_perseverance_2   -0.2039      0.133     -1.533
0.127      -0.466       0.059
important_child_good_mann_2                    -0.2062      0.141     -1.467
0.144      -0.483       0.071
important_child_hard_work_2                    -0.0784      0.139     -0.566
0.572      -0.352       0.195
important_child_independence_2                 -0.4060      0.153     -2.659
0.009      -0.707      -0.105
important_child_obedience_2                    -0.0707      0.140     -0.506
0.613      -0.346       0.205
important_child_responsibility_2                0.1818      0.148      1.232
0.220      -0.109       0.473
important_child_savings_2                      -0.1694      0.161     -1.051
0.295      -0.488       0.149
important_child_tolerance_respect_2             0.0823      0.124      0.662
```

```
0.509      -0.163       0.328
important_unselfishness_2                      -0.2939      0.147     -1.999
0.047      -0.584      -0.004
membership_chari_huma_1                        -0.4591      0.255     -1.803
0.073      -0.961       0.043
membership_church_religious_org_1               0.0442      0.207      0.214
0.831      -0.364       0.452
membership_church_religious_org_2               0.2117      0.197      1.074
0.284      -0.177       0.600
membership_environ_org_1                       -0.3164      0.266     -1.189
0.236      -0.841       0.209
membership_labour_union_1                       0.3205      0.281      1.142
0.255      -0.233       0.874
membership_other_org_1                         -0.0520      0.317     -0.164
0.870      -0.677       0.573
membership_pol_party_1                         -0.4685      0.384     -1.221
0.224      -1.226       0.288
men_better_pol_2                               -0.3907      0.167     -2.333
0.021      -0.721      -0.060
men_better_pol_3                               -0.1360      0.170     -0.802
0.424      -0.471       0.199
men_better_pol_4                               -0.0647      0.209     -0.310
0.757      -0.476       0.347
most_impo_for_country_2                         0.5292      0.275      1.923
0.056      -0.014       1.072
most_impo_for_country_3                         0.1343      0.157      0.855
0.394      -0.176       0.444
most_impo_for_country_4                        -0.0100      0.154     -0.065
0.948      -0.315       0.295
neigh_ppl_aids_2                               -0.1461      0.130     -1.127
0.261      -0.402       0.110
often_discuss_pol_friends_2                     0.0615      0.138      0.444
0.658      -0.212       0.335
often_discuss_pol_friends_3                    -0.0484      0.197     -0.246
0.806      -0.437       0.340
only_acc_my_rel_2                               0.1545      0.155      0.996
0.320      -0.151       0.460
only_acc_my_rel_3                               0.5735      0.196      2.926
0.004       0.187       0.960
only_acc_my_rel_4                               0.5175      0.331      1.566
0.119      -0.135       1.170
opposition_imped_running_2                      0.2017      0.195      1.034
0.303      -0.183       0.587
opposition_imped_running_3                      0.2067      0.201      1.029
0.305      -0.190       0.603
opposition_imped_running_4                      0.0411      0.185      0.222
0.824      -0.323       0.406
racial_belonging_51005                         -0.4065      0.726     -0.560
0.576      -1.838       1.025
racial_belonging_51006                          0.3779      1.081      0.349
0.727      -1.756       2.511
religion_always_right_2                        -0.1882      0.175     -1.076
0.283      -0.533       0.157
religion_always_right_3                         0.2526      0.177      1.430
0.154      -0.096       0.601
religion_always_right_4                         0.4988      0.227      2.195
0.029       0.050       0.947
religious_person_2                             -0.3562      0.199     -1.788
0.075      -0.749       0.037
religious_person_3                             -0.4733      0.582     -0.814
```

```
0.417      -1.621       0.674
security_neigh_2                                -0.0919      0.164     -0.560
0.576      -0.415       0.231
security_neigh_3                                -0.1424      0.183     -0.776
0.439      -0.504       0.220
security_neigh_4                                -0.0242      0.215     -0.113
0.911      -0.449       0.400
strong_leader_2                                 -0.3136      0.146     -2.148
0.033      -0.602      -0.026
strong_leader_3                                 -0.1506      0.183     -0.824
0.411      -0.511       0.210
strong_leader_4                                  0.1226      0.198      0.620
0.536      -0.268       0.513
trust_ ppl_know_personally_2                     0.3912      0.154      2.537
0.012       0.087       0.695
trust_ ppl_know_personally_3                     0.0675      0.201      0.335
0.738      -0.330       0.465
trust_ ppl_know_personally_4                     0.2425      0.325      0.746
0.457      -0.399       0.884
trust_family_2                                  -0.2892      0.286     -1.010
0.314      -0.854       0.276
trust_family_3                                   0.2332      0.713      0.327
0.744      -1.174       1.640
trust_most_ppl_2                                -0.0126      0.202     -0.062
0.950      -0.412       0.387
trust_ppl_other_nat_2                            0.3709      0.408      0.909
0.365      -0.434       1.176
trust_ppl_other_nat_3                            0.4087      0.425      0.962
0.337      -0.429       1.247
trust_ppl_other_nat_4                            0.5848      0.437      1.338
0.182      -0.277       1.447
trust_ppl_other_rel_2                            0.1369      0.487      0.281
0.779      -0.824       1.098
trust_ppl_other_rel_3                            0.4161      0.501      0.831
0.407      -0.572       1.404
trust_ppl_other_rel_4                            0.1407      0.499      0.282
0.778      -0.843       1.124
univ_more_boy_2                                  0.2520      0.288      0.874
0.383      -0.317       0.821
univ_more_boy_3                                 -0.1311      0.241     -0.543
0.588      -0.607       0.345
univ_more_boy_4                                 -0.0426      0.235     -0.181
0.856      -0.507       0.422
valut_dem_pol_system_2                           0.1498      0.135      1.112
0.268      -0.116       0.416
valut_dem_pol_system_3                           0.3610      0.237      1.523
0.129      -0.107       0.829
valut_dem_pol_system_4                           1.1525      0.296      3.897
0.000       0.569       1.736
valut_rel_pol_system_2                          -0.2988      0.248     -1.206
0.229      -0.788       0.190
valut_rel_pol_system_3                           0.0169      0.269      0.063
0.950      -0.515       0.548
valut_rel_pol_system_4                          -0.2596      0.242     -1.075
0.284      -0.736       0.217
voter_bribed_2                                   0.0657      0.142      0.464
0.643      -0.214       0.345
voter_bribed_3                                   0.3459      0.196      1.761
0.080      -0.042       0.733
voter_bribed_4                                   0.1322      0.222      0.595
```

```
0.552      -0.306       0.570
voter_choice_2                                  0.2372      0.131      1.812
0.072      -0.021       0.496
voter_choice_3                                  0.2543      0.190      1.336
0.183      -0.121       0.630
voter_choice_4                                  0.4831      0.211      2.291
0.023       0.067       0.899
votes_count_fairly_2                            0.1093      0.200      0.548
0.585      -0.285       0.503
votes_count_fairly_3                            0.0537      0.194      0.277
0.782      -0.328       0.435
votes_count_fairly_4                           -0.1141      0.204     -0.559
0.577      -0.517       0.288
==============================================================================
Omnibus:                       37.981   Durbin-Watson:                   2.044
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               61.555
Skew:                           0.703   Prob(JB):                     4.30e-14
Kurtosis:                       4.572   Cond. No.                         605.
==============================================================================

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

In [27]:

```python
# seleziono dati che mi interessano per fare confronti
Y_pred = results.predict(X)
MSE_OLS = mean_squared_error(Y, Y_pred)
R_quadro_Adj_OLS = results.rsquared_adj
F_OLS = results.fvalue
Likelihood_OLS = results.llf
Predittori_OLS = len(results.params) - 1
BIC_OLS = results.bic
AIC_OLS = results.aic
print('MSE: ', MSE_OLS)
```

```
MSE:  0.34248025161343115
```

In [28]:

```python
# test per eteroschedasticità
import statsmodels.stats.diagnostic as smd
bp_test = smd.het_breuschpagan(results.resid, results.model.exog)
print(f"P-value del test di Breusch-Pagan: {bp_test[1]}")
```

```
P-value del test di Breusch-Pagan: 0.08995178373176912
```

Lettura rapida dei risultati

- Numero di osservazioni : 332
- Numero dei predittori: 147
- Calcolo della covarianza non robusto all'eteroschedasticità
- Variabilità dei dati spiegata dal modello (R_square): 0.658
- Variabilità dei dati penalizzando per la numerosità dei predittori(Adj_R_squared): 0.384
- Valore della statistica F e probabilità associata: il modello presenta un migliore adattamento ai dati rispetto al modello nullo
- Assunti sulla normalità dei residui (Omnibus e Jarque-Bera) : non soddisfatta
- Skewness - 0 < x < 1: Simmetria dei dati : accettabile
- Curtosi: elevata, presenza frequente di valori estremi (code grasse)

- Numero di condizionamento: elevato, in parte dovuto alla multicollinearità dei predittori categoriali, in parte dovuto ai dati
- Durbin Watson : circa 2, i residui non autocorrelano a un lag

In primo luogo, la numerosità dei predittori in rapporto alle osservazioni e il mancato rispetto degli assunti di normalità sui residui suggeriscono l'imprecisione delle stime.

In secondo luogo, la numerosità dei predittori rende difficile l'interpretabilità del modello, elemento supportato anche dal valore dell'Adjusted R_square rispetto a R_square

Sembra pertanto conveniente operare per ridurre la dimensionalità della matrice dei predittori.

A questo scopo verrà applicato il metodo Sure Indipendence Screening (SIR; Fan & Lv, 2008), che seleziona i predittori ordinando i coefficientiti univariati dei predittori standardizzati:

- **Primo step**: seleziono i predittori dalla matrice originaria e elimino la costante
- **Secondo step**: Standardizzo i predittori
- **Terzo step**: Effettuo regressioni lineari univariate della variabile responso, senza la costante
- **Quarto step**: Ordino i beta dei predittori in base al valore assoluto
- **Quinto step**: calcolo n/log(n), trovando il numero di predittori da selezionare
- **Sesto**: analisi di regressione con il subset di variabili selezionate

In [29]:
```
X.columns
```

Out[29]:
```
Index(['const', 'justifiable_pol_viol', 'info_source_newspaper',
       'info_source_tv', 'info_source_radio', 'info_source_mobile',
       'info_source_email', 'info_source_internet', 'info_source_socia_media',
       'info_source_friends_collegues',
       ...
       'valut_rel_pol_system_4', 'voter_bribed_2', 'voter_bribed_3',
       'voter_bribed_4', 'voter_choice_2', 'voter_choice_3', 'voter_choice_4',
       'votes_count_fairly_2', 'votes_count_fairly_3', 'votes_count_fairly_4'],
      dtype='object', length=148)
```

In [30]:
```
# avere una colonna con solo valori = 0 potrebbe darmi problemi
# verifico se ho colonne con valori pari a zero
has_all_zero = (X == 0).all(axis=0).any()
print('C\'è una colonna che ha tutti i valori = 0?: ', has_all_zero)
```

```
C'è una colonna che ha tutti i valori = 0?:  False
```

In [31]:
```
# droppo intercetta da X
X = X.drop(['const'], axis = 1)
X.columns
```

Out[31]:
```
Index(['justifiable_pol_viol', 'info_source_newspaper', 'info_source_tv',
       'info_source_radio', 'info_source_mobile', 'info_source_email',
       'info_source_internet', 'info_source_socia_media',
       'info_source_friends_collegues', 'satisfaction_financial',
       ...
       'valut_rel_pol_system_4', 'voter_bribed_2', 'voter_bribed_3',
       'voter_bribed_4', 'voter_choice_2', 'voter_choice_3', 'voter_choice_4',
       'votes_count_fairly_2', 'votes_count_fairly_3', 'votes_count_fairly_4'],
      dtype='object', length=147)
```

In [32]:
```
# considero ciascuna colonna e la standardizzo senza centrarla
# Calcola deviazione standard solo per le colonne numeriche
std_devs = X.std(axis=0)
print('Standard deviations: ')
print(std_devs)
X_standard= X / std_devs  # Standardizzazione
print('I predittori sono stati standardizzati?')
print('------------')
print('Prima riga predittori standardizzati:\n', X_standard.head(1))
```

```
Standard deviations:
justifiable_pol_viol     1.001509
info_source_newspaper    1.001509
info_source_tv           1.001509
info_source_radio        1.001509
info_source_mobile       1.001509
                           ...
voter_choice_3           0.326016
voter_choice_4           0.299647
votes_count_fairly_2     0.430127
votes_count_fairly_3     0.487458
votes_count_fairly_4     0.420769
Length: 147, dtype: float64
I predittori sono stati standardizzati?
------------
Prima riga predittori standardizzati:
       justifiable_pol_viol  info_source_newspaper  info_source_tv  \
2009             -0.303031              -1.981439       -0.701387

      info_source_radio  info_source_mobile  info_source_email  \
2009           0.600197            -0.68927            0.69082

      info_source_internet  info_source_socia_media  \
2009              1.746271                 1.779718

      info_source_friends_collegues  satisfaction_financial  ...  \
2009                      -0.846859                2.608738  ...

      valut_rel_pol_system_4  voter_bribed_2  voter_bribed_3  voter_bribed_4  \
2009                1.997022             0.0             0.0        3.101108

      voter_choice_2  voter_choice_3  voter_choice_4  votes_count_fairly_2  \
2009        2.136485             0.0             0.0                   0.0

      votes_count_fairly_3  votes_count_fairly_4
2009               2.05146                   0.0

[1 rows x 147 columns]
```

In [33]:
```
# ciclo su ogni colonna una regressione univariata senza intercetta ed estraggo
coefficienti = []
for i in X_standard.columns:
    # Definisci il modello
    model = sm.OLS(Y, X_standard[[i]])
    results = model.fit()
    coefficienti.append(results.params[i])
```

In [34]:
```
# Crea un DataFrame con i coefficienti e li ordino dal più grande al più piccolo
coeff_df = pd.DataFrame({
    'Colonna': X_standard.columns,
    'Coefficiente': coefficienti
})
coeff_df['Assoluto'] = coeff_df['Coefficiente'].abs()
coeff_df_sorted = coeff_df.sort_values(by='Assoluto', ascending=False).drop('Ass
print('Primi coefficienti, esempio: ')
coeff_df_sorted.head()
```

```
Primi coefficienti, esempio:
```

Out[34]:

	Colonna	Coefficiente
19	justifiable_violence_other_ppl	0.426137
0	justifiable_pol_viol	0.415616
40	justifiable_beat_wife	0.396053
34	justifiable_cheating_tax	0.273117
53	goal_life_parent_proud_4	0.226314

In [35]:
```
# calcolo selettore in base a formula n/log(n)
selettore = X_standard.shape[0]/np.log(X_standard.shape[0])
selettore
```

Out[35]: 57.19074608561028

In [36]:
```
# Estraggo le prime 57 righe di coeff_df_sorted
first_57_df = coeff_df_sorted.head(57)
# Ottieni la lista dei nomi delle colonne da selezionare
colonne_predittori = first_57_df['Colonna'].tolist()
```

In [37]:
```
# isolo il sottoinsieme dei predittori selezionati sulla matrice dei predittori
X = X[colonne_predittori]
```

In [38]:
```
print('Predittori estratti: ', X.columns)
print('Dimensione matrice dei predittori', X.shape)
```

```
Predittori estratti:  Index(['justifiable_violence_other_ppl', 'justifiable_pol_v
iol',
       'justifiable_beat_wife', 'justifiable_cheating_tax',
       'goal_life_parent_proud_4', 'only_acc_my_rel_3',
       'justifiable_euthanasia', 'nation_pride', 'justifiable_stealing',
       'justifiable_prostitution', 'justifiable_abortion', 'importance_god',
       'info_source_tv', 'only_acc_my_rel_4', 'voter_choice_3',
       'children_parent_duty', 'valut_dem_pol_system_4',
       'religion_always_right_4', 'justifiable_death_penal',
       'importance_democracy', 'justifiable_homosex',
       'info_source_socia_media', 'importance_religion_4',
       'importance_friends_4', 'trust_family_3', 'often_discuss_pol_friends_3',
       'trust_ppl_other_rel_3', 'trust_ ppl_know_personally_3',
       'goal_life_parent_proud_3', 'housewife_fulfilling_4',
       'importance_politics_4', 'trust_ppl_other_rel_4', 'happiness_2',
       'importance_politics_2', 'most_impo_for_country_2', 'voter_bribed_4',
       'justifiable_divorce', 'membership_other_org_1',
       'membership_pol_party_1', 'info_source_newspaper',
       'important_child_responsibility_2',
       'important_child_tolerance_respect_2', 'strong_leader_3',
       'aims_of_country_4', 'valut_dem_pol_system_3', 'only_acc_my_rel_2',
       'valut_rel_pol_system_2', 'trust_ppl_other_nat_4',
       'often_discuss_pol_friends_2', 'demo_must_protect_rights',
       'religion_always_right_2', 'men_better_pol_2', 'aims_of_country_2',
       'votes_count_fairly_4', 'importance_friends_2', 'importance_religion_3',
       'security_neigh_3'],
      dtype='object')
Dimensione matrice dei predittori (332, 57)
```

Rifitto OLS con i predittori selezionati

In [39]:

```python
# rifitto il modello solo con i predittori selezionati
X = sm.add_constant(X)
# Rifitta il modello con i predittori significativi
model = sm.OLS(Y, X)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     justifiable_terrorism   R-squared:                       0.474
Model:                               OLS   Adj. R-squared:                  0.365
Method:                    Least Squares   F-statistic:                     4.334
Date:                   Mon, 16 Dec 2024   Prob (F-statistic):           1.32e-16
Time:                           23:05:55   Log-Likelihood:                -364.40
No. Observations:                    332   AIC:                             844.8
Df Residuals:                        274   BIC:                             1065.
Df Model:                             57
Covariance Type:               nonrobust
==============================================================================
=======================
                                        coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
----------------------
const                                 -0.3772      0.199     -1.895      0.059
-0.769       0.015
justifiable_violence_other_ppl         0.1946      0.062      3.155      0.002
0.073       0.316
justifiable_pol_viol                   0.1760      0.055      3.208      0.001
0.068       0.284
justifiable_beat_wife                  0.2254      0.055      4.100      0.000
0.117       0.334
justifiable_cheating_tax               0.0623      0.056      1.107      0.269
-0.048       0.173
goal_life_parent_proud_4               0.2677      0.460      0.581      0.561
-0.639       1.174
only_acc_my_rel_3                      0.4499      0.158      2.854      0.005
0.140       0.760
justifiable_euthanasia                -0.0079      0.058     -0.137      0.891
-0.121       0.105
nation_pride                           0.0836      0.055      1.507      0.133
-0.026       0.193
justifiable_stealing                   0.0047      0.057      0.082      0.935
-0.107       0.117
justifiable_prostitution              -0.0492      0.060     -0.818      0.414
-0.168       0.069
justifiable_abortion                   0.0199      0.056      0.359      0.720
-0.089       0.129
importance_god                         0.0181      0.057      0.319      0.750
-0.094       0.130
info_source_tv                         0.0217      0.056      0.386      0.700
-0.089       0.133
only_acc_my_rel_4                      0.2992      0.275      1.087      0.278
-0.243       0.841
voter_choice_3                         0.2517      0.155      1.621      0.106
-0.054       0.557
children_parent_duty                   0.0747      0.052      1.443      0.150
-0.027       0.177
valut_dem_pol_system_4                 0.6025      0.226      2.669      0.008
0.158       1.047
religion_always_right_4                0.1667      0.155      1.077      0.282
-0.138       0.471
justifiable_death_penal               -0.0190      0.055     -0.342      0.732
-0.128       0.090
importance_democracy                   0.0249      0.052      0.476      0.635
-0.078       0.128
justifiable_homosex                   -0.0724      0.059     -1.218      0.224
```

```
-0.189       0.045
info_source_socia_media                -0.0253      0.049     -0.514      0.608
-0.122       0.072
importance_religion_4                  -0.5657      0.358     -1.580      0.115
-1.271       0.139
importance_friends_4                    0.5989      0.313      1.915      0.057
-0.017       1.215
trust_family_3                          0.0129      0.536      0.024      0.981
-1.041       1.067
often_discuss_pol_friends_3            -0.0149      0.150     -0.099      0.921
-0.310       0.281
trust_ppl_other_rel_3                   0.3534      0.133      2.663      0.008
0.092       0.615
trust_ ppl_know_personally_3           -0.2429      0.135     -1.794      0.074
-0.510       0.024
goal_life_parent_proud_3                0.0921      0.199      0.464      0.643
-0.299       0.483
housewife_fulfilling_4                 -0.1064      0.142     -0.751      0.453
-0.385       0.173
importance_politics_4                  -0.2601      0.114     -2.289      0.023
-0.484      -0.036
trust_ppl_other_rel_4                   0.0391      0.144      0.271      0.786
-0.244       0.323
happiness_2                            -0.0033      0.101     -0.033      0.974
-0.203       0.196
importance_politics_2                   0.0022      0.134      0.017      0.987
-0.261       0.265
most_impo_for_country_2                 0.3127      0.220      1.424      0.155
-0.119       0.745
voter_bribed_4                          0.1322      0.153      0.863      0.389
-0.169       0.434
justifiable_divorce                     0.0157      0.021      0.736      0.462
-0.026       0.058
membership_other_org_1                 -0.4153      0.269     -1.545      0.124
-0.945       0.114
membership_pol_party_1                 -0.2630      0.273     -0.964      0.336
-0.800       0.274
info_source_newspaper                   0.0458      0.050      0.924      0.356
-0.052       0.143
important_child_responsibility_2        0.1444      0.121      1.195      0.233
-0.093       0.382
important_child_tolerance_respect_2     0.0403      0.099      0.406      0.685
-0.155       0.235
strong_leader_3                         0.0900      0.132      0.682      0.496
-0.170       0.350
aims_of_country_4                       0.1455      0.153      0.950      0.343
-0.156       0.447
valut_dem_pol_system_3                  0.1075      0.183      0.588      0.557
-0.252       0.467
only_acc_my_rel_2                       0.1136      0.120      0.947      0.345
-0.123       0.350
valut_rel_pol_system_2                 -0.1095      0.124     -0.881      0.379
-0.354       0.135
trust_ppl_other_nat_4                   0.1380      0.135      1.023      0.307
-0.128       0.404
often_discuss_pol_friends_2             0.1826      0.107      1.714      0.088
-0.027       0.392
demo_must_protect_rights                0.0690      0.049      1.409      0.160
-0.027       0.165
religion_always_right_2                -0.1585      0.116     -1.361      0.175
```

```
-0.388       0.071
men_better_pol_2                      -0.1806      0.107     -1.684      0.093
-0.392       0.031
aims_of_country_2                      0.0999      0.113      0.884      0.377
-0.123       0.322
votes_count_fairly_4                  -0.2306      0.118     -1.950      0.052
-0.463       0.002
importance_friends_2                   0.1835      0.098      1.872      0.062
-0.009       0.377
importance_religion_3                 -0.1917      0.186     -1.029      0.304
-0.559       0.175
security_neigh_3                      -0.0509      0.111     -0.458      0.648
-0.270       0.168
==============================================================================
Omnibus:                      101.736   Durbin-Watson:                   2.028
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              321.616
Skew:                           1.363   Prob(JB):                     1.45e-70
Kurtosis:                       6.978   Cond. No.                         68.5
==============================================================================

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

In [40]:

```
# seleziono dati che mi interessano per fare confronti
Y_pred = results.predict(X)
MSE_OLS_rid = mean_squared_error(Y, Y_pred)
F_OLS_rid = results.fvalue
R_quadro_Adj_OLS_rid = results.rsquared_adj
Likelihood_OLS_rid = results.llf
Predittori_OLS_rid = len(results.params) # la costante non è sign
BIC_OLS_rid = results.bic
AIC_OLS_rid = results.aic
print('MSE: ', MSE_OLS_rid)
```

```
MSE:  0.5258615325854477
```

In [41]:

```
from statsmodels.stats.diagnostic import linear_reset

# Test per verificare la specificazione errata del modello: Test di Ramsey RESET
reset_test = linear_reset(results, power=2, use_f=True)

# Stampa anche il valore F-statistic e p-value
print("F-statistic:", reset_test.fvalue)
print("p-value:", reset_test.pvalue)
```

```
F-statistic: 103.18545795714448
p-value: 8.996998913966108e-21
```

In [42]:

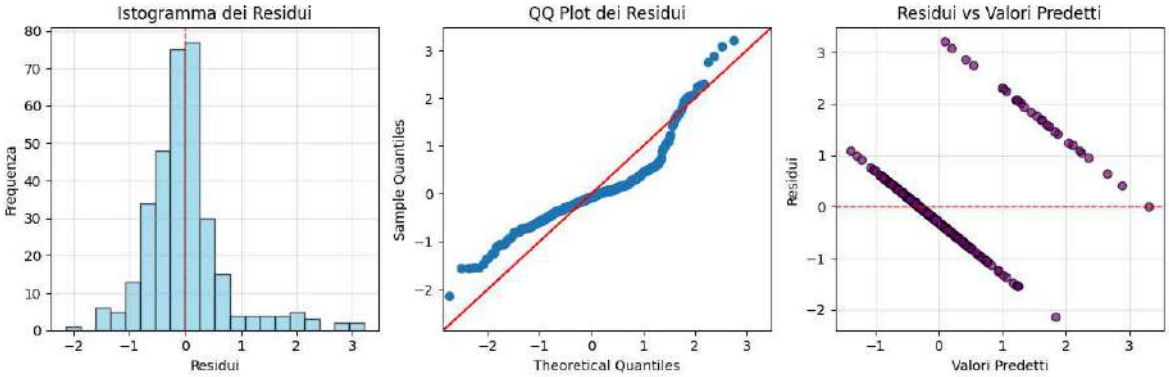
```
# test per eteroschedasticità
import statsmodels.stats.diagnostic as smd
bp_test = smd.het_breuschpagan(results.resid, results.model.exog)
print(f"P-value del test di Breusch-Pagan: {bp_test[1]}")
```

```
P-value del test di Breusch-Pagan: 0.00016619016395944856
```

In [43]:

```
# Residui e valori predetti
residuals = results.resid
fitted_values = results.fittedvalues

# Crea la figura con tre sottotrame affiancate
```

```
fig, axes = plt.subplots(1, 3, figsize=(12,4))  # 1 riga, 3 colonne

# Istogramma dei residui
axes[0].hist(residuals, bins=20, color='skyblue', edgecolor='black', alpha=0.7)
axes[0].set_title("Istogramma dei Residui")
axes[0].set_xlabel("Residui")
axes[0].set_ylabel("Frequenza")
axes[0].axvline(0, color='red', linestyle='dashed', linewidth=1)  # Linea vertic
axes[0].grid(alpha=0.3)

# QQ plot dei residui
sm.qqplot(residuals, line='45', ax=axes[1])
axes[1].set_title("QQ Plot dei Residui")

# Scatter plot residui vs fitted values
axes[2].scatter(fitted_values, residuals, alpha=0.7, color='purple', edgecolor='
axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)  # Linea orizzo
axes[2].set_title("Residui vs Valori Predetti")
axes[2].set_xlabel("Valori Predetti")
axes[2].set_ylabel("Residui")
axes[2].grid(alpha=0.3)

# Ottimizza gli spazi e mostra i grafici
plt.tight_layout()
plt.show()
```



Lettura rapida dei risultati

- Numero di osservazioni : 332
- Numero dei predittori: 57
- Calcolo della covarianza non robusto all'eteroschedasticità
- Variabilità dei dati spiegata dal modello (R_square): 0.474
- Variabilità dei dati penalizzando per la numerosità dei predittori(Adj_R_squared): 0.365
- Valore della statistica F e probabilità associata: il modello presenta un migliore adattamento ai dati rispetto al modello nullo
- Assunti sulla normalità dei residui (Omnibus e Jarque-Bera) : non soddisfatta
- Skewness - 0 < x < 1: Simmetria dei dati : accettabile
- Curtosi: levata, presenza frequente di valori estremi (code grasse)
- Numero di condizionamento: elevato, in parte dovuto alla multicollinearità dei predittori categoriali, in parte dovuto ai dati, comunque inferiore al modello precedente
- Durbin Watson : circa 2, i residui non autocorrelano a un lag

Grafici dei residui

Distribuzione dei residui non normale. Presenza di code grasse dimostra sia a livello dello scatterplot che a livello dell'istogramma. Il grafico dei residui vs valori predetti indica pattern nella distribuzione dei residui e in associazione con il RESET test fornisce prove che il modello è probabilmente mal specificato.

Elimino outliers, rifitto il modello e poi lo rifitto solo con predittori significativi

Distanze di Cook

In [44]:
```
# Calcola la distanza di Cook
influence = results.get_influence()
cooks_d = influence.cooks_distance[0]

# Calcola la soglia 4/n
n = len(X)  # Numero di osservazioni
threshold = 4 / n
# Calcola la soglia: 3 volte la media delle distanze di Cook
threshold3 = 3 * cooks_d.mean()
# Crea un nuovo DataFrame con le distanze di Cook
cooks_df = pd.DataFrame({
    'cooks_distance': cooks_d,
    'cooks_distance_gt_1': cooks_d > 1,    # True se la distanza di Cook > 1
    'cooks_distance_gt_4n': cooks_d > threshold,  # True se la distanza di Cook
    'cooks_distance_gt_3mean': cooks_d > threshold3  # # True se la distanza di
})
```

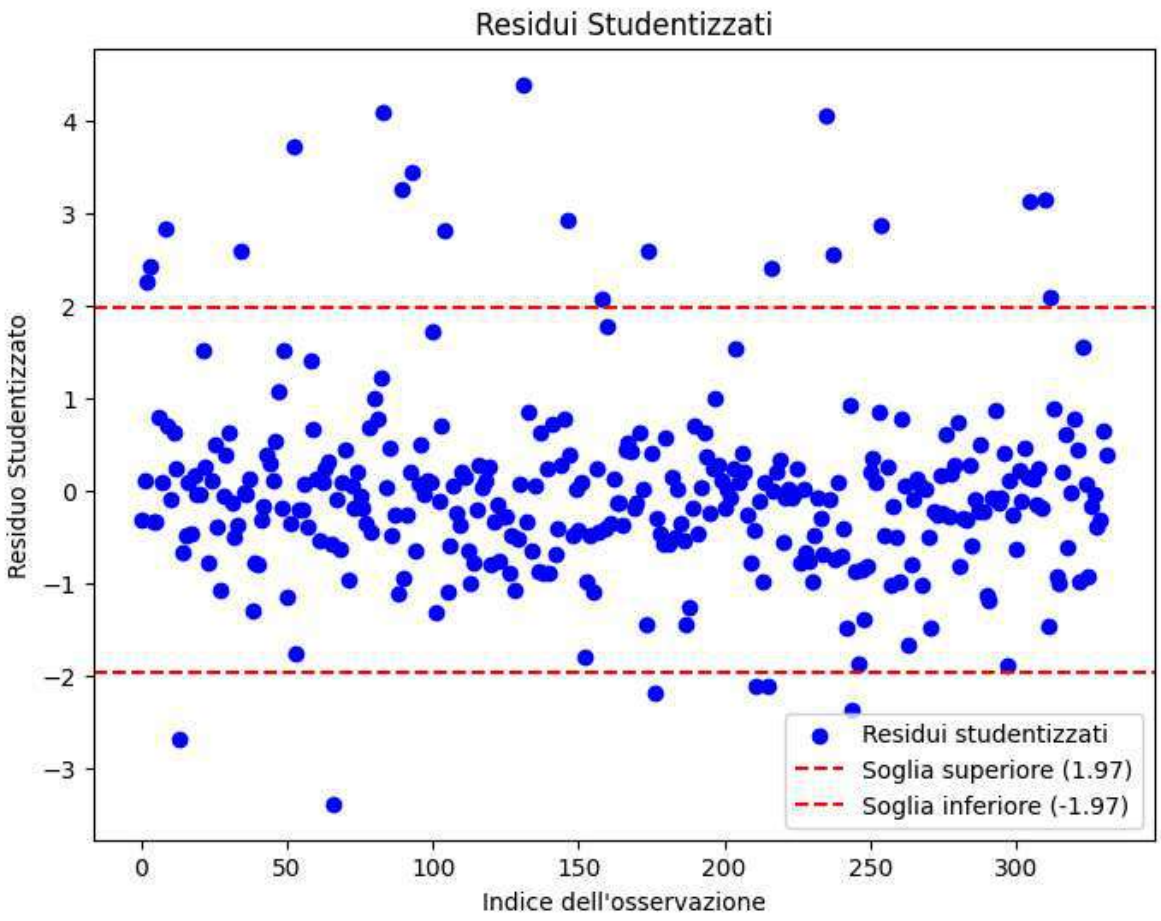
In [45]:
```
# Conta quante osservazioni soddisfano i criteri
count_gt_1 = cooks_df['cooks_distance_gt_1'].sum()
count_gt_4n = cooks_df['cooks_distance_gt_4n'].sum()
count_gt_3mean = cooks_df['cooks_distance_gt_3mean'].sum()

# Stampa i risultati
print(f"Numero di osservazioni con distanza di Cook > 1: {count_gt_1}")
print(f"Numero di osservazioni con distanza di Cook > 4/n: {count_gt_4n}")
print(f"Numero di osservazioni con distanza di Cook > 3mean: {count_gt_3mean}")
print('---------')
# Indici per distanza di Cook
outliers1 = np.where(cooks_d > 1)[0]
outliers_4n = np.where(cooks_d > threshold)[0]
outliers_3mean = np.where(cooks_d > threshold3)[0]
print(f"Indici per Cook's distance 4/n: {outliers1}")
print(f"Indici per Cook's distance 4/n: {outliers_4n}")
print(f"Indici per Cook's distance 3mean: {outliers_3mean}")
```

```
Numero di osservazioni con distanza di Cook > 1: 0
Numero di osservazioni con distanza di Cook > 4/n: 36
Numero di osservazioni con distanza di Cook > 3mean: 30
---------
Indici per Cook's distance 4/n: []
Indici per Cook's distance 4/n: [  2   3   8  13  34  49  52  53  66  83  89  93
100 104 131 146 152 158
 160 174 176 204 211 215 216 235 237 242 244 246 254 263 297 305 310 312]
Indici per Cook's distance 3mean: [  2   3   8  13  34  52  66  83  89  93 104 13
1 146 152 158 174 176 204
 211 215 216 235 237 244 246 254 263 305 310 312]
```

In [46]:

```
# calcolo i residui studentizzati per rilevare se ce ne sono altri non rilevati
from statsmodels.stats.outliers_influence import OLSInfluence as OI
from scipy import stats
# Calcolare i residui studentizzati
influence = OI(results)

studentized_residuals = influence.resid_studentized


# Calcolare i gradi di libertà
n = len(X)  # numero di osservazioni
p = X.shape[1]  # numero di predittori (incluso l'intercetta)
df = n - p - 2  # gradi di libertà

# Calcolare il valore critico della t di Student per un livello di significativi
alpha = 0.05
t_critical = stats.t.ppf(1 - alpha / 2, df)  # 1 - alpha/2 per il test a due coa

print(f"Valore critico della t di Student: {t_critical}")

# Identificare gli outliers (residui studentizzati > t_critical o < -t_critical)
outliers = np.where(np.abs(studentized_residuals) > t_critical)[0]
print("Outliers identificati (indici):", outliers)

# Visualizzare i residui studentizzati
plt.figure(figsize=(8, 6))
plt.scatter(range(len(studentized_residuals)), studentized_residuals, color='blu
plt.axhline(y=t_critical, color='red', linestyle='--', label=f'Soglia superiore
plt.axhline(y=-t_critical, color='red', linestyle='--', label=f'Soglia inferiore
plt.title('Residui Studentizzati')
plt.xlabel('Indice dell\'osservazione')
plt.ylabel('Residuo Studentizzato')
plt.legend()
plt.show()
```

```
Valore critico della t di Student: 1.9687238473650965
Outliers identificati (indici): [  2   3   8  13  34  52  66  83  89  93 104 131
146 158 174 176 211 215
 216 235 237 244 254 305 310 312]
```



In [47]:
```
# Verifica se i valori in 'outliers' (residui studentizzati) sono inclusi in 'ou
outliers_in_4n = np.isin(outliers, outliers_4n)  # Verifica per 'outliers_4n'
outliers_in_3mean = np.isin(outliers, outliers_3mean)  # Verifica per 'outliers_
outliers_in_1 = np.isin(outliers, outliers1)  # Verifica per 'outliers1'

# Stampa i risultati
print("Valori in 'outlier' che sono in 'outliers_4n':", outliers[outliers_in_4n]
print("Valori in 'outlier' che sono in 'outliers_3mean':", outliers[outliers_in_
print("Valori in 'outlier' che sono in 'outliers1':", outliers[outliers_in_1])
```

```
Valori in 'outlier' che sono in 'outliers_4n': [  2   3   8  13  34  52  66  83
89  93 104 131 146 158 174 176 211 215
 216 235 237 244 254 305 310 312]
Valori in 'outlier' che sono in 'outliers_3mean': [  2   3   8  13  34  52  66  8
3  89  93 104 131 146 158 174 176 211 215
 216 235 237 244 254 305 310 312]
Valori in 'outlier' che sono in 'outliers1': []
```

In [48]:
```
# elimino dal df e dalla y gli outliers comuni a distanze di Cook e residui stud
cooks_distance_gt_4n = cooks_d > threshold
cooks_distance_gt_3mean = cooks_d > threshold3
mask = ~((cooks_d > threshold) | (cooks_d > threshold3))
X = X.loc[mask]
Y = Y.loc[mask]
print('Verifico riduzione delle osservazioni per X e per Y: ', X.shape, Y.shape)
```

```
Verifico riduzione delle osservazioni per X e per Y:  (296, 58) (296,)
```

Rifitto il modello OLS ridotto senza outliers

In [49]:
```
# rifitto il modello solo con i predittori selezionati
X = sm.add_constant(X)
# Rifitta il modello con i predittori significativi
model = sm.OLS(Y, X)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     justifiable_terrorism   R-squared:                       0.686
Model:                               OLS   Adj. R-squared:                  0.611
Method:                    Least Squares   F-statistic:                     9.133
Date:                   Mon, 16 Dec 2024   Prob (F-statistic):           1.96e-35
Time:                           23:05:56   Log-Likelihood:                 11.442
No. Observations:                    296   AIC:                             93.12
Df Residuals:                        238   BIC:                             307.2
Df Model:                             57
Covariance Type:               nonrobust
==============================================================================
======================
                                        coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
----------------------
const                                 -0.3029      0.068     -4.423      0.000
-0.438      -0.168
justifiable_violence_other_ppl         0.0807      0.025      3.276      0.001
0.032       0.129
justifiable_pol_viol                   0.1777      0.025      7.095      0.000
0.128       0.227
justifiable_beat_wife                  0.1806      0.029      6.191      0.000
0.123       0.238
justifiable_cheating_tax               0.0366      0.021      1.730      0.085
-0.005       0.078
goal_life_parent_proud_4               1.9197      0.394      4.868      0.000
1.143       2.697
only_acc_my_rel_3                      0.1346      0.058      2.328      0.021
0.021       0.248
justifiable_euthanasia                -0.0310      0.020     -1.574      0.117
-0.070       0.008
nation_pride                           0.0451      0.020      2.253      0.025
0.006       0.084
justifiable_stealing                  -0.0132      0.023     -0.572      0.568
-0.059       0.032
justifiable_prostitution              -0.0037      0.022     -0.169      0.866
-0.047       0.040
justifiable_abortion                   0.0115      0.020      0.587      0.558
-0.027       0.050
importance_god                        -0.0110      0.020     -0.543      0.587
-0.051       0.029
info_source_tv                        -0.0105      0.020     -0.537      0.592
-0.049       0.028
only_acc_my_rel_4                     -0.0070      0.119     -0.059      0.953
-0.241       0.227
voter_choice_3                        -0.0031      0.058     -0.053      0.958
-0.118       0.112
children_parent_duty                   0.0126      0.018      0.689      0.492
-0.023       0.049
valut_dem_pol_system_4                 0.1749      0.082      2.136      0.034
0.014       0.336
religion_always_right_4               -0.0380      0.058     -0.652      0.515
-0.153       0.077
justifiable_death_penal               -0.0302      0.019     -1.549      0.123
-0.068       0.008
importance_democracy                   0.0062      0.018      0.336      0.737
-0.030       0.042
justifiable_homosex                   -0.0052      0.022     -0.237      0.813
```

```
-0.048       0.038
info_source_socia_media                -0.0266      0.017     -1.599      0.111
-0.059       0.006
importance_religion_4                   0.2387      0.134      1.780      0.076
-0.025       0.503
importance_friends_4                    0.0424      0.118      0.360      0.719
-0.190       0.275
trust_family_3                          0.0297      0.202      0.147      0.883
-0.368       0.427
often_discuss_pol_friends_3             0.0083      0.051      0.164      0.870
-0.092       0.108
trust_ppl_other_rel_3                   0.0777      0.047      1.669      0.096
-0.014       0.169
trust_ ppl_know_personally_3           -0.0457      0.048     -0.947      0.344
-0.141       0.049
goal_life_parent_proud_3                0.0093      0.076      0.122      0.903
-0.141       0.160
housewife_fulfilling_4                 -0.0093      0.048     -0.192      0.848
-0.104       0.086
importance_politics_4                  -0.0414      0.039     -1.051      0.294
-0.119       0.036
trust_ppl_other_rel_4                   0.0230      0.049      0.470      0.639
-0.073       0.119
happiness_2                             0.0102      0.035      0.288      0.774
-0.060       0.080
importance_politics_2                   0.0173      0.048      0.358      0.721
-0.078       0.113
most_impo_for_country_2                -0.0400      0.080     -0.503      0.616
-0.197       0.117
voter_bribed_4                         -0.0069      0.051     -0.136      0.892
-0.107       0.093
justifiable_divorce                    -0.0026      0.008     -0.345      0.730
-0.017       0.012
membership_other_org_1                 -0.0425      0.096     -0.445      0.657
-0.231       0.146
membership_pol_party_1                  0.0319      0.096      0.334      0.739
-0.156       0.220
info_source_newspaper                  -0.0032      0.017     -0.189      0.850
-0.037       0.030
important_child_responsibility_2        0.0539      0.043      1.265      0.207
-0.030       0.138
important_child_tolerance_respect_2     0.0348      0.034      1.013      0.312
-0.033       0.102
strong_leader_3                         0.1360      0.047      2.884      0.004
0.043       0.229
aims_of_country_4                       0.0540      0.052      1.038      0.300
-0.048       0.156
valut_dem_pol_system_3                  0.0479      0.063      0.757      0.450
-0.077       0.173
only_acc_my_rel_2                       0.0017      0.041      0.040      0.968
-0.080       0.083
valut_rel_pol_system_2                 -0.0056      0.043     -0.131      0.896
-0.090       0.079
trust_ppl_other_nat_4                   0.0278      0.047      0.594      0.553
-0.064       0.120
often_discuss_pol_friends_2            -0.0196      0.037     -0.523      0.601
-0.093       0.054
demo_must_protect_rights                0.0060      0.017      0.355      0.723
-0.027       0.039
religion_always_right_2                -0.0394      0.040     -0.986      0.325
```

```
-0.118       0.039
men_better_pol_2                      -0.0075      0.037     -0.202      0.840
-0.081       0.066
aims_of_country_2                      0.0037      0.039      0.095      0.924
-0.073       0.080
votes_count_fairly_4                  -0.0367      0.041     -0.900      0.369
-0.117       0.044
importance_friends_2                   0.0403      0.035      1.140      0.255
-0.029       0.110
importance_religion_3                  0.0469      0.068      0.690      0.491
-0.087       0.181
security_neigh_3                       0.0910      0.038      2.406      0.017
0.016       0.166
==============================================================================
Omnibus:                      175.932   Durbin-Watson:                   2.257
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4018.820
Skew:                           1.932   Prob(JB):                         0.00
Kurtosis:                      20.633   Cond. No.                         138.
==============================================================================

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

In [50]:
```
# seleziono dati che mi interessano per fare confronti
Y_pred = results.predict(X)
MSE_OLS_rid_no_out = mean_squared_error(Y, Y_pred)
F_OLS_rid_no_out = results.fvalue
R_quadro_Adj_OLS_rid_no_out = results.rsquared_adj
Likelihood_OLS_rid_no_out = results.llf
Predittori_OLS_rid_no_out = len(results.params) -1 # la costante è significativa
BIC_OLS_rid_no_out = results.bic
AIC_OLS_rid_no_out = results.aic
print('MSE: ', MSE_OLS_rid_no_out)
```

```
MSE:  0.05419372701250925
```

In [51]:
```
# Test per verificare la specificazione errata del modello: Test di Ramsey RESET
reset_test = linear_reset(results, power=2, use_f=True)
# Stampa anche il valore F-statistic e p-value
print("F-statistic:", reset_test.fvalue)
print("p-value:", reset_test.pvalue)
```

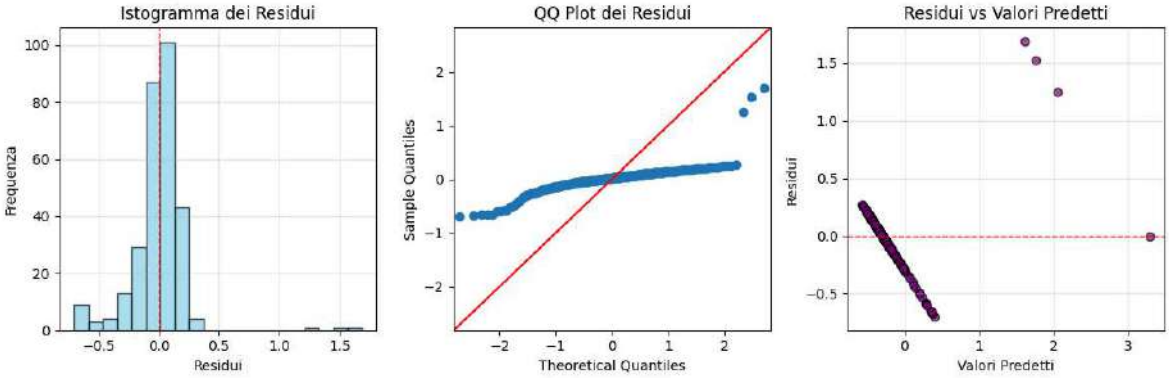
```
F-statistic: 3028.966448314146
p-value: 5.340902537897964e-137
```

In [52]:
```
# test per eteroschedasticità
bp_test = smd.het_breuschpagan(results.resid, results.model.exog)
print(f"P-value del test di Breusch-Pagan: {bp_test[1]}")
```

```
P-value del test di Breusch-Pagan: 4.849001394416066e-22
```

In [53]:
```
# Residui e valori predetti
residuals = results.resid
fitted_values = results.fittedvalues

# Crea la figura con tre sottotrame affiancate
fig, axes = plt.subplots(1, 3, figsize=(12,4))  # 1 riga, 3 colonne

# Istogramma dei residui
axes[0].hist(residuals, bins=20, color='skyblue', edgecolor='black', alpha=0.7)
```

```
axes[0].set_title("Istogramma dei Residui")
axes[0].set_xlabel("Residui")
axes[0].set_ylabel("Frequenza")
axes[0].axvline(0, color='red', linestyle='dashed', linewidth=1)  # Linea vertic
axes[0].grid(alpha=0.3)

# QQ plot dei residui
sm.qqplot(residuals, line='45', ax=axes[1])
axes[1].set_title("QQ Plot dei Residui")

# Scatter plot residui vs fitted values
axes[2].scatter(fitted_values, residuals, alpha=0.7, color='purple', edgecolor='
axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)  # Linea orizzo
axes[2].set_title("Residui vs Valori Predetti")
axes[2].set_xlabel("Valori Predetti")
axes[2].set_ylabel("Residui")
axes[2].grid(alpha=0.3)

# Ottimizza gli spazi e mostra i grafici
plt.tight_layout()
plt.show()
```



Lettura rapida dei risultati

- Numero di osservazioni : 296
- Numero dei predittori: 57
- Calcolo della covarianza non robusto all'eteroschedasticità
- Variabilità dei dati spiegata dal modello (R_square): 0.686
- Variabilità dei dati penalizzando per la numerosità dei predittori(Adj_R_squared): 0.611
- Valore della statistica F e probabilità associata: il modello presenta un migliore adattamento ai dati rispetto al modello nullo
- Assunti sulla normalità dei residui (Omnibus e Jarque-Bera) : non soddisfatta
- Skewness - 1 < x: aumento dell'asimmetria dei residui
- Curtosi: elevata, indica presenza frequente di valori estremi (code grasse)
- Numero di condizionamento: molto elevato, in parte dovuto alla multicollinearità dei predittori categoriali, in parte dovuto ai dati, comunque inferiore al modello precedente
- Durbin Watson : < 2.5, valori accettabili, i residui non autocorrelano a un lag

Grafici dei residui

Distribuzione dei residui non normale. Aumenta l'asimmetria nei dati. Presenza di code grasse dimostrata sia a livello dello scatterplot che a livello dell'istogramma. Il grafico dei residui vs valori predetti e indica pattern nella distribuzione dei residui e in associazione con il RESET test e con il QQ plot forniscono prove che il modello è mal specificato e che la varianza dei residui non è distribuita normalmente. Il test di Breusch-Pagan fornisce ulteriori evidenze ($p < 0.05$) in merito.

In [54]:
```
# coefficienti significativi
coefficients = results.params
p_values = results.pvalues
# Filtra i coefficienti con p inferiore a 0.05
significant_coefficients = coefficients[p_values < 0.05]
significant_columns = significant_coefficients.index
significant_coefficients = significant_coefficients.sort_values(ascending=True)
print('Coefficienti significativi:\n', significant_coefficients)
```

```
Coefficienti significativi:
 const                           -0.302874
nation_pride                     0.045055
justifiable_violence_other_ppl   0.080662
security_neigh_3                 0.091019
only_acc_my_rel_3                0.134577
strong_leader_3                  0.135989
valut_dem_pol_system_4           0.174930
justifiable_pol_viol             0.177747
justifiable_beat_wife            0.180588
goal_life_parent_proud_4         1.919733
dtype: float64
```

In [55]:
```
X_sign = X[significant_columns]
X_sign.columns
```

Out[55]:
```
Index(['const', 'justifiable_violence_other_ppl', 'justifiable_pol_viol',
       'justifiable_beat_wife', 'goal_life_parent_proud_4',
       'only_acc_my_rel_3', 'nation_pride', 'valut_dem_pol_system_4',
       'strong_leader_3', 'security_neigh_3'],
      dtype='object')
```

In [56]:
```
# Rifitta il modello con i predittori significativi
X_sign = sm.add_constant(X_sign)
model = sm.OLS(Y, X_sign)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     justifiable_terrorism   R-squared:                       0.640
Model:                               OLS   Adj. R-squared:                  0.629
Method:                    Least Squares   F-statistic:                     56.61
Date:                   Mon, 16 Dec 2024   Prob (F-statistic):           2.00e-58
Time:                           23:05:57   Log-Likelihood:                -8.7233
No. Observations:                    296   AIC:                             37.45
Df Residuals:                        286   BIC:                             74.35
Df Model:                              9
Covariance Type:               nonrobust
==============================================================================
=================
                                    coef    std err          t      P>|t|
[0.025      0.975]
------------------------------------------------------------------------------
-----------------
const                             -0.2630      0.021    -12.619      0.000      -
0.304      -0.222
justifiable_violence_other_ppl     0.0618      0.021      2.985      0.003
0.021       0.103
justifiable_pol_viol               0.1746      0.022      8.031      0.000
0.132       0.217
justifiable_beat_wife              0.1998      0.024      8.252      0.000
0.152       0.247
goal_life_parent_proud_4           1.9220      0.276      6.976      0.000
1.380       2.464
only_acc_my_rel_3                  0.1058      0.044      2.383      0.018
0.018       0.193
nation_pride                       0.0560      0.016      3.547      0.000
0.025       0.087
valut_dem_pol_system_4             0.1787      0.072      2.473      0.014
0.037       0.321
strong_leader_3                    0.1281      0.041      3.104      0.002
0.047       0.209
security_neigh_3                   0.1048      0.033      3.171      0.002
0.040       0.170
==============================================================================
Omnibus:                      206.402   Durbin-Watson:                   2.319
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6995.210
Skew:                           2.304   Prob(JB):                         0.00
Kurtosis:                      26.366   Cond. No.                         22.0
==============================================================================

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

In [57]:

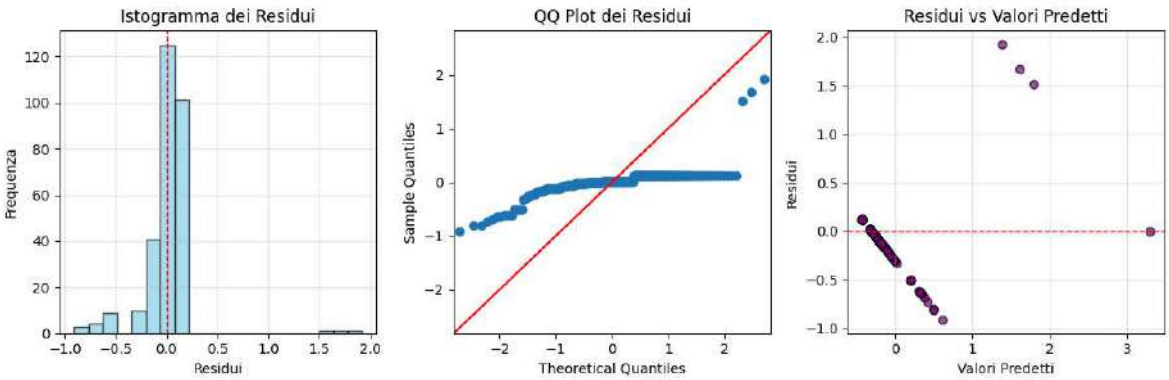
```python
# seleziono dati che mi interessano per fare confronti
Y_pred = results.predict(X_sign)
MSE_OLS_rid_sign = mean_squared_error(Y, Y_pred)
F_OLS_rid_sign = results.fvalue
R_quadro_Adj_OLS_rid_sign = results.rsquared_adj
Likelihood_OLS_rid_sign = results.llf
Predittori_OLS_rid_sign = len(results.params) - 1
BIC_OLS_rid_sign = results.bic
AIC_OLS_rid_sign = results.aic
print('MSE: ', MSE_OLS_rid_sign)
```

```
MSE:  0.06210454037022208
```

In [58]:
```
# Test per verificare la specificazione errata del modello: Test di Ramsey RESET
reset_test = linear_reset(results, power=2, use_f=True)

# Stampa anche il valore F-statistic e p-value
print("F-statistic:", reset_test.fvalue)
print("p-value:", reset_test.pvalue)
```

```
F-statistic: 2133.57017883838
p-value: 2.2879121899281624e-134
```

In [59]:
```
# Residui e valori predetti
residuals = results.resid
fitted_values = results.fittedvalues

# Crea la figura con tre sottotrame affiancate
fig, axes = plt.subplots(1, 3, figsize=(12,4))  # 1 riga, 3 colonne

# Istogramma dei residui
axes[0].hist(residuals, bins=20, color='skyblue', edgecolor='black', alpha=0.7)
axes[0].set_title("Istogramma dei Residui")
axes[0].set_xlabel("Residui")
axes[0].set_ylabel("Frequenza")
axes[0].axvline(0, color='red', linestyle='dashed', linewidth=1)  # Linea vertic
axes[0].grid(alpha=0.3)

# QQ plot dei residui
sm.qqplot(residuals, line='45', ax=axes[1])
axes[1].set_title("QQ Plot dei Residui")

# Scatter plot residui vs fitted values
axes[2].scatter(fitted_values, residuals, alpha=0.7, color='purple', edgecolor='
axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)  # Linea orizzo
axes[2].set_title("Residui vs Valori Predetti")
axes[2].set_xlabel("Valori Predetti")
axes[2].set_ylabel("Residui")
axes[2].grid(alpha=0.3)

# Ottimizza gli spazi e mostra i grafici
plt.tight_layout()
plt.show()
```



Lettura rapida dei risultati

- Numero di osservazioni : 332
- Numero dei predittori: 9

Vale quanto detto in precedenza. Si osserva una riduzione importante nel valore di condizionamento, e un miglioramento significativo di log-likelihood BIC e AIC, tuttavia senza un sostanziale miglioramento nella varianza spiegata dal modello (Adj_R_square = 0.629)

Sulla distribuzione dei residui e la mal specificazione del modello vale quanto detto in precedenza

OLS ridotto , modello robusto

In [59]:

Funzione di perdita di Huber

La funzione di perdita di Huber è una combinazione di una funzione quadratica per errori piccoli e lineare per errori grandi (outlier). La formula è la seguente:

$$L_\delta(r) = \begin{cases} \frac{1}{2}r^2 & \text{se } |r| \leq \delta, \\ \delta(|r| - \frac{1}{2}\delta) & \text{se } |r| > \delta. \end{cases}$$

Dove:

- r è il residuo,
- δ è un parametro che controlla il punto in cui la funzione passa dalla forma quadratica a quella lineare.
- Quando $|r| \leq \delta$, la funzione di perdita assume una forma quadratica.
- Quando $|r| > \delta$, la funzione di perdita assume una forma lineare, riducendo l'influenza degli outlier.

Questo approccio riduce l'impatto degli errori grandi (outliers) nella stima dei parametri, rendendo il modello più robusto.

MAD (Median Absolute Deviation)

Il **MAD** (Median Absolute Deviation) è un metodo robusto per stimare la **deviazione standard** che riduce l'influenza degli outliers:

$$MAD = \text{mediana}\left(|r_i - \text{mediana}(r)|\right)$$

Dove:

- r_i sono i residui, cioè le differenze tra i valori osservati e quelli predetti.
- **Mediana** rappresenta la mediana dei residui.
- $|r_i - \text{mediana}(r)|$ è la deviazione assoluta dei residui dalla mediana, che misura quanto ciascun residuo si discosta dal valore centrale.

Il MAD è una misura robusta di dispersione, poiché la mediana non è influenzata dagli outliers, a differenza della media. Questa proprietà lo rende utile per stimare la variabilità

in presenza di dati anomali.

In [60]:
```
# fitto  modello robusto con RLM
model_robust = sm.RLM(Y, X)
# Fitta il modello
results_robust = model_robust.fit()
# Riepilogo dei risultati
print(results_robust.summary())
```

```
                    Robust linear Model Regression Results
==============================================================================
Dep. Variable:     justifiable_terrorism   No. Observations:                  296
Model:                               RLM   Df Residuals:                      238
Method:                             IRLS   Df Model:                           57
Norm:                            HuberT
Scale Est.:                         mad
Cov Type:                            H1
Date:                   Mon, 16 Dec 2024
Time:                           23:05:58
No. Iterations:                      50
==============================================================================
=======================
                                        coef    std err          z      P>|z|
[0.025      0.975]
------------------------------------------------------------------------------
----------------------
const                                 -0.3035   8.77e-16  -3.46e+14      0.000
-0.303      -0.303
justifiable_violence_other_ppl       5.551e-17   3.15e-16      0.176      0.860
-5.63e-16    6.74e-16
justifiable_pol_viol                  7.72e-16   3.21e-16      2.405      0.016
1.43e-16     1.4e-15
justifiable_beat_wife                7.286e-17   3.74e-16      0.195      0.845
-6.6e-16    8.05e-16
justifiable_cheating_tax             8.708e-16   2.71e-16      3.212      0.001
3.39e-16     1.4e-15
goal_life_parent_proud_4                3.6035   5.05e-15   7.13e+14      0.000
3.603       3.603
only_acc_my_rel_3                    2.246e-16   7.41e-16      0.303      0.762
-1.23e-15    1.68e-15
justifiable_euthanasia              -6.245e-17   2.52e-16     -0.248      0.804
-5.57e-16    4.32e-16
nation_pride                        -6.293e-16   2.56e-16     -2.456      0.014
-1.13e-15   -1.27e-16
justifiable_stealing                -7.425e-16   2.95e-16     -2.515      0.012
-1.32e-15   -1.64e-16
justifiable_prostitution            -6.592e-17   2.83e-16     -0.233      0.816
-6.21e-16    4.89e-16
justifiable_abortion                   6.8e-16    2.5e-16      2.716      0.007
1.89e-16    1.17e-15
importance_god                      -7.407e-16   2.58e-16     -2.868      0.004
-1.25e-15   -2.34e-16
info_source_tv                      -2.255e-17    2.5e-16     -0.090      0.928
-5.13e-16    4.67e-16
only_acc_my_rel_4                   -7.772e-16   1.52e-15     -0.510      0.610
-3.77e-15    2.21e-15
voter_choice_3                      -1.388e-17   7.48e-16     -0.019      0.985
-1.48e-15    1.45e-15
children_parent_duty                -5.716e-16   2.35e-16     -2.436      0.015
-1.03e-15   -1.12e-16
valut_dem_pol_system_4               2.345e-15   1.05e-15      2.236      0.025
2.89e-16     4.4e-15
religion_always_right_4             -4.718e-16   7.46e-16     -0.632      0.527
-1.93e-15    9.91e-16
justifiable_death_penal              6.332e-17   2.49e-16      0.254      0.800
-4.25e-16    5.52e-16
importance_democracy                 3.057e-17   2.34e-16      0.130      0.896
-4.29e-16     4.9e-16
justifiable_homosex                  7.268e-16   2.81e-16      2.583      0.010
```

```
1.75e-16    1.28e-15
info_source_socia_media              1.006e-16   2.13e-16      0.472      0.637
-3.17e-16    5.19e-16
importance_religion_4                1.506e-15   1.72e-15      0.877      0.381
-1.86e-15    4.87e-15
importance_friends_4                 1.832e-15   1.51e-15      1.212      0.225
-1.13e-15    4.79e-15
trust_family_3                       1.277e-15   2.58e-15      0.494      0.621
-3.79e-15    6.34e-15
often_discuss_pol_friends_3          4.961e-16   6.51e-16      0.762      0.446
-7.8e-16    1.77e-15
trust_ppl_other_rel_3                6.661e-16   5.96e-16      1.117      0.264
-5.03e-16    1.83e-15
trust_ ppl_know_personally_3         8.674e-17   6.18e-16      0.140      0.888
-1.13e-15     1.3e-15
goal_life_parent_proud_3             7.529e-16   9.78e-16      0.770      0.441
-1.16e-15    2.67e-15
housewife_fulfilling_4               2.949e-16   6.17e-16      0.478      0.633
-9.14e-16     1.5e-15
importance_politics_4               -3.747e-16   5.04e-16     -0.743      0.458
-1.36e-15    6.14e-16
trust_ppl_other_rel_4                8.708e-16   6.26e-16      1.391      0.164
-3.56e-16     2.1e-15
happiness_2                         -9.368e-16   4.54e-16     -2.065      0.039
-1.83e-15   -4.79e-17
importance_politics_2               -6.245e-16    6.2e-16     -1.007      0.314
-1.84e-15    5.91e-16
most_impo_for_country_2              4.302e-16   1.02e-15      0.422      0.673
-1.57e-15    2.43e-15
voter_bribed_4                      -5.343e-16   6.53e-16     -0.819      0.413
-1.81e-15    7.45e-16
justifiable_divorce                   -5.1e-16   9.68e-17     -5.269      0.000
-7e-16    -3.2e-16
membership_other_org_1                -6.8e-16   1.23e-15     -0.554      0.579
-3.08e-15    1.72e-15
membership_pol_party_1               -1.18e-16   1.22e-15     -0.096      0.923
-2.52e-15    2.28e-15
info_source_newspaper                 -6.8e-16   2.19e-16     -3.104      0.002
-1.11e-15   -2.51e-16
important_child_responsibility_2     2.949e-16   5.45e-16      0.541      0.589
-7.74e-16    1.36e-15
important_child_tolerance_respect_2  4.476e-16   4.39e-16      1.018      0.309
-4.14e-16    1.31e-15
strong_leader_3                      9.437e-16   6.04e-16      1.562      0.118
-2.4e-16    2.13e-15
aims_of_country_4                    9.714e-17   6.66e-16      0.146      0.884
-1.21e-15     1.4e-15
valut_dem_pol_system_3              -1.336e-16   8.12e-16     -0.165      0.869
-1.72e-15    1.46e-15
only_acc_my_rel_2                    1.527e-16    5.3e-16      0.288      0.773
-8.87e-16    1.19e-15
valut_rel_pol_system_2               1.839e-16    5.5e-16      0.334      0.738
-8.94e-16    1.26e-15
trust_ppl_other_nat_4               -3.886e-16   5.99e-16     -0.648      0.517
-1.56e-15    7.86e-16
often_discuss_pol_friends_2         -4.233e-16    4.8e-16     -0.881      0.378
-1.36e-15    5.18e-16
demo_must_protect_rights              4.51e-17   2.16e-16      0.209      0.834
-3.77e-16    4.68e-16
religion_always_right_2             -3.513e-16   5.12e-16     -0.687      0.492
```

```
-1.35e-15    6.52e-16
men_better_pol_2                  -5.586e-16   4.77e-16     -1.171      0.242
-1.49e-15    3.76e-16
aims_of_country_2                    5.1e-16   4.99e-16      1.021      0.307
-4.69e-16    1.49e-15
votes_count_fairly_4               1.769e-16   5.22e-16      0.339      0.735
-8.46e-16     1.2e-15
importance_friends_2               2.914e-16   4.53e-16      0.643      0.520
-5.97e-16    1.18e-15
importance_religion_3              3.574e-16   8.72e-16      0.410      0.682
-1.35e-15    2.07e-15
security_neigh_3                   8.223e-16   4.85e-16      1.696      0.090
-1.28e-16    1.77e-15
==============================================================================
=======================

If the model instance has been used for another fit with different fit parameter
s, then the fit options might not be the correct ones anymore .
```

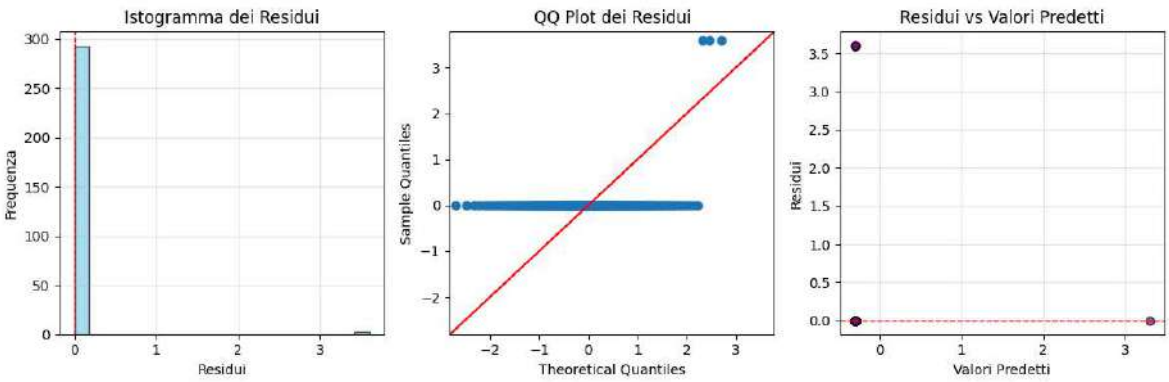
In [61]:

```
# seleziono dati che mi interessano per fare confronti
Y_pred = results_robust.predict(X)
MSE_OLS_rid_rob= mean_squared_error(Y, Y_pred)
Predittori_OLS_rid_rob = len(results_robust.params) - 1
print('MSE: ', MSE_OLS_rid_rob)
```

```
MSE:  0.13123288894218169
```

In [62]:

```
# Residui e valori predetti
residuals = results_robust.resid
fitted_values = results_robust.fittedvalues

# Crea la figura con tre sottotrame affiancate
fig, axes = plt.subplots(1, 3, figsize=(12,4))  # 1 riga, 3 colonne

# Istogramma dei residui
axes[0].hist(residuals, bins=20, color='skyblue', edgecolor='black', alpha=0.7)
axes[0].set_title("Istogramma dei Residui")
axes[0].set_xlabel("Residui")
axes[0].set_ylabel("Frequenza")
axes[0].axvline(0, color='red', linestyle='dashed', linewidth=1)  # Linea vertic
axes[0].grid(alpha=0.3)

# QQ plot dei residui
sm.qqplot(residuals, line='45', ax=axes[1])
axes[1].set_title("QQ Plot dei Residui")

# Scatter plot residui vs fitted values
axes[2].scatter(fitted_values, residuals, alpha=0.7, color='purple', edgecolor='
axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)  # Linea orizzo
axes[2].set_title("Residui vs Valori Predetti")
axes[2].set_xlabel("Valori Predetti")
axes[2].set_ylabel("Residui")
axes[2].grid(alpha=0.3)

# Ottimizza gli spazi e mostra i grafici
plt.tight_layout()
plt.show()
```



Lettura rapida dei risultati

- Numero di osservazioni : 296
- Numero dei predittori: 57
- Metodo di stima: Iteratively Reweighted Least Squares (IRLS)
- Norma per calcolare la funzione di perdita: HuberT (quadratica per gli errori piccoli e lineare per gli errori grandi)
- Metodo di stima della scala: MAD, stima della deviazione standard non influenzata dagli outliers
- Matrice di covarianza: H1, matrice di covarianza robusta utilizzata per correggere l'eteroschedasticità nei residui.

In [63]:
```
# coefficienti significativi
coefficients = results_robust.params
p_values = results_robust.pvalues

# Filtra i coefficienti con p inferiore a 0.05
significant_coefficients = coefficients[p_values < 0.05]
significant_columns = significant_coefficients.index
significant_coefficients = significant_coefficients.sort_values(ascending=True)
print('Coefficienti significativi:\n', significant_coefficients)
```

```
Coefficienti significativi:
 const                     -3.034878e-01
happiness_2               -9.367507e-16
justifiable_stealing      -7.424616e-16
importance_god            -7.407269e-16
info_source_newspaper     -6.800116e-16
nation_pride              -6.292709e-16
children_parent_duty      -5.715914e-16
justifiable_divorce       -5.100087e-16
justifiable_abortion       6.800116e-16
justifiable_homosex        7.268491e-16
justifiable_pol_viol       7.719519e-16
justifiable_cheating_tax   8.708312e-16
valut_dem_pol_system_4     2.345346e-15
goal_life_parent_proud_4   3.603463e+00
dtype: float64
```

In [64]:
```
# rifitto il modello solo con i coefficienti significativi e la costante
X_sign = X[significant_columns]
# Rifitta il modello con i predittori significativi
model = sm.RLM(Y, X_sign)
results_robust = model.fit()
```

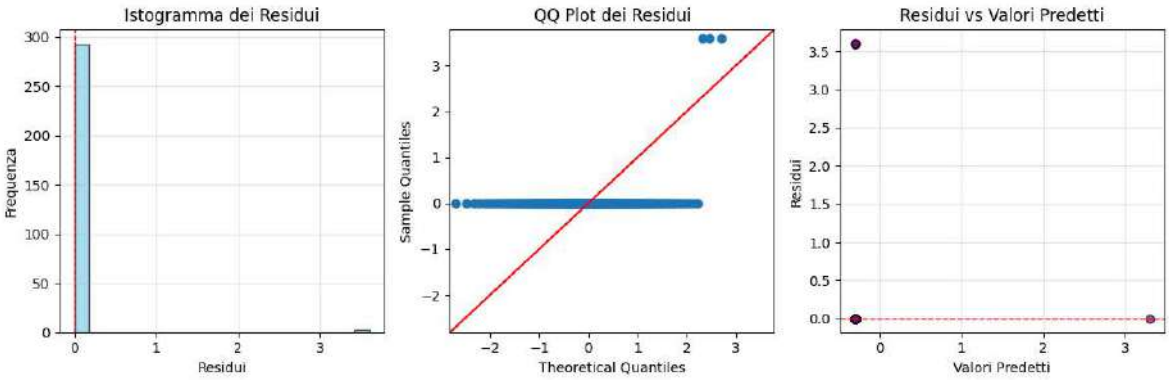
```
# Stampa il nuovo sommario
print(results_robust.summary())
```

```
                     Robust linear Model Regression Results
==============================================================================
Dep. Variable:     justifiable_terrorism   No. Observations:                  296
Model:                               RLM   Df Residuals:                      282
Method:                             IRLS   Df Model:                           13
Norm:                            HuberT
Scale Est.:                         mad
Cov Type:                            H1
Date:                   Mon, 16 Dec 2024
Time:                           23:05:59
No. Iterations:                      50
==============================================================================
===========
                              coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
-----------
const                      -0.3035   8.56e-16  -3.55e+14      0.000      -0.303
-0.303
justifiable_pol_viol      1.256e-15   5.58e-16      2.252      0.024    1.63e-16
2.35e-15
justifiable_cheating_tax  1.409e-15   4.68e-16      3.007      0.003    4.91e-16
2.33e-15
goal_life_parent_proud_4     3.6035   7.39e-15   4.87e+14      0.000       3.603
3.603
nation_pride             -1.651e-15   4.42e-16     -3.736      0.000   -2.52e-15
-7.85e-16
justifiable_stealing      6.731e-16   5.16e-16      1.304      0.192   -3.39e-16
1.68e-15
justifiable_abortion     -3.868e-16    4.5e-16     -0.860      0.390   -1.27e-15
4.95e-16
importance_god           -9.888e-17   4.47e-16     -0.221      0.825   -9.75e-16
7.77e-16
children_parent_duty       4.69e-16   4.05e-16      1.158      0.247   -3.25e-16
1.26e-15
valut_dem_pol_system_4    2.442e-15    1.9e-15      1.283      0.199   -1.29e-15
6.17e-15
justifiable_homosex      -3.556e-16   4.66e-16     -0.764      0.445   -1.27e-15
5.57e-16
happiness_2               1.546e-15   8.19e-16      1.888      0.059   -5.93e-17
3.15e-15
justifiable_divorce       1.207e-15   1.63e-16      7.397      0.000    8.87e-16
1.53e-15
info_source_newspaper      8.76e-17   3.92e-16      0.224      0.823    -6.8e-16
8.55e-16
==============================================================================
===========

If the model instance has been used for another fit with different fit parameter
s, then the fit options might not be the correct ones anymore .
```

In [65]:
```
# seleziono dati che mi interessano per fare confronti
Y_pred = results_robust.predict(X_sign)
MSE_OLS_rid_rob_sign = mean_squared_error(Y, Y_pred)
Predittori_OLS_rid_rob_sign = len(results_robust.params) - 1
print('MSE: ', MSE_OLS_rid_rob_sign)
```

```
MSE:  0.13123288894218105
```

In [66]:
```
# Residui e valori predetti
residuals = results_robust.resid
fitted_values = results_robust.fittedvalues

# Crea la figura con tre sottotrame affiancate
fig, axes = plt.subplots(1, 3, figsize=(12,4))  # 1 riga, 3 colonne

# Istogramma dei residui
axes[0].hist(residuals, bins=20, color='skyblue', edgecolor='black', alpha=0.7)
axes[0].set_title("Istogramma dei Residui")
axes[0].set_xlabel("Residui")
axes[0].set_ylabel("Frequenza")
axes[0].axvline(0, color='red', linestyle='dashed', linewidth=1)  # Linea vertic
axes[0].grid(alpha=0.3)

# QQ plot dei residui
sm.qqplot(residuals, line='45', ax=axes[1])
axes[1].set_title("QQ Plot dei Residui")

# Scatter plot residui vs fitted values
axes[2].scatter(fitted_values, residuals, alpha=0.7, color='purple', edgecolor='
axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)  # Linea orizzo
axes[2].set_title("Residui vs Valori Predetti")
axes[2].set_xlabel("Valori Predetti")
axes[2].set_ylabel("Residui")
axes[2].grid(alpha=0.3)

# Ottimizza gli spazi e mostra i grafici
plt.tight_layout()
plt.show()
```



Lettura rapida dei risultati

Nessun cambiamento sostanziale rispetto al modello con più predittori.

Riduco ulteriormente la dimensionalità con LASSO CV cercando automaticamente il miglior valore di alpha

Regressione LASSO (Least Absolute Shrinkage and Selection Operator)

La **regressione LASSO** è una tecnica di regressione che si distingue per l'uso di una penalizzazione basata sulla **norma L1** dei coefficienti, il che comporta una **shrinking** dei coefficienti e la **selezione** automatica delle variabili più importanti.

Formula della regressione LASSO

La formulazione della regressione LASSO è la seguente:

$$\hat{\beta}_{\text{LASSO}} = \arg\min_{\beta_0,\beta_1,\ldots,\beta_p} \left[\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{i,j}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

Dove:

- $\hat{\beta}_{\text{LASSO}}$ sono i **coefficienti stimati** del modello LASSO.
- y_i è il valore osservato della variabile dipendente per l'osservazione (i).
- $x_{i,j}$ è il valore della variabile indipendente (j) per l'osservazione (i).
- β_j sono i **coefficienti** che il modello deve stimare.
- β_0 è l'intercetta.
- λ è il **parametro di regolarizzazione** che controlla la forza della penalizzazione L1.
- $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$ è la **norma L1** dei coefficienti, la somma dei valori assoluti dei coefficienti di regressione, che serve a penalizzare i valori più grandi di β_j.

Penalizzazione e parametro di regolarizzazione

1. **Penalizzazione L1 (norma L1)**: $\lambda \sum_{j=1}^{p} |\beta_j|$, è la penalizzazione L1. La presenza del termine di penalizzazione ha due effetti principali:
   - I coefficienti β_j tendono a ridursi verso zero.
   - Con una penalizzazione L1 abbastanza forte, alcuni coefficienti β_j diventeranno esattamente zero, e verranno esclusi dal modello
2. **Parametro di regolarizzazione** λ: Il parametro λ controlla l'intensità della penalizzazione. Valori più grandi di λ aumentano la penalizzazione, riducendo maggiormente i coefficienti e facendo tendere più variabili a zero.

- per λ = 0 : il modello coincide con OLS
- per λ = ∞: il modello LASSO coicide con il modello nullo
- per λ > 0: Gli elementi di $X \in \mathbb{R}^{n \times p}$ derivano da una distribuzione di probabilità continua su $\mathbb{R}^{np} -> $ la soluzione del problema di

minimizzazione esiste ed è unica con probabilità 1, ed ha al massimo $\min(n, p)$ coefficienti non nulli.

In [67]:
```
# Importa le librerie necessarie
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.linear_model import LassoCV


# elimino la colonna 'costante' nella matrice dei predittori
X = X.drop(columns=['const'])

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.3, random_

# Scalo e standardizzo i dati
scaler = StandardScaler()    # centro i dati e divido per la deviazione standara
X_train_scaled = scaler.fit_transform(X_train)  # Fit e trasformazione sui dati
X_test_scaled = scaler.transform(X_test)        # Solo trasformazione sui dati a
X_train_scaled.shape, X_test_scaled.shape
```

Out[67]:
```
((207, 57), (89, 57))
```

In [68]:
```
# Fitta il modello LassoCV
lasso_cv = LassoCV(eps=0.001, n_alphas=1000, max_iter = 10000, cv=5, n_jobs=-1,
# considera 1000 valori di alpha con incremento di 0.001, 100000 iterazioni per
#  il dataset sarà diviso in 5 parti e il modello verrà addestrato 5 volte
lasso_cv.fit(X_train_scaled , y_train)

# Calcolo le predizioni sui dati
y_pred = lasso_cv.predict(X_test_scaled)

coefficients = lasso_cv.coef_
```

In [69]:
```
# Visualizza il miglior parametro alpha trovato
print(f"Miglior valore di alpha: {lasso_cv.alpha_}")
print('-----------------')
print("Coefficienti del modello Lasso:")
print(lasso_cv.coef_)
```

```
Miglior valore di alpha: 0.006013388303282942
-----------------
Coefficienti del modello Lasso:
[ 0.05484327  0.15076107  0.09556726  0.02883593  0.10266902  0.00798405
 -0.          0.02278421  0.03790968  0.00334278 -0.         -0.03513949
 -0.00431265 -0.00411964 -0.01653797  0.02407574  0.02962868 -0.
 -0.02570933 -0.0045882   0.         -0.02435059  0.04073622  0.
  0.01385452  0.          0.         -0.         -0.         -0.
 -0.01299575 -0.01730227  0.00463254 -0.00648768 -0.00816318  0.
  0.         -0.00939024  0.0019469  -0.00156856  0.02333308  0.
  0.02868528  0.02055089  0.03176909 -0.          0.00201012  0.01884535
  0.          0.00755139 -0.00372683 -0.         -0.         -0.00975405
  0.00435227 -0.02800003  0.03374729]
```

In [70]:
```
# Calcolo Adj_R_squared del modello
r_squared = lasso_cv.score(X_train, y_train)

# Numero di osservazioni e predittori
n = X_train.shape[0]  # Numero di osservazioni
p = X_train.shape[1]  # Numero di variabili predittive

# Calcola l'R^2 aggiustato
```

```
r_squared_adj = 1 - (1 - r_squared) * (n - 1) / (n - p - 1)
R_quadro_Adj_LASSO = r_squared_adj
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:486: UserWarning: X has f
eature names, but LassoCV was fitted without feature names
  warnings.warn(
```

In [71]:

```
column_names = X.columns

# Crea un DataFrame con i coefficienti associati ai nomi delle colonne
coefficients_df = pd.DataFrame(coefficients, index=column_names, columns=['Coeff

# Filtra i coefficienti diversi da zero
non_zero_coefficients_df = coefficients_df[coefficients_df['Coefficient'] != 0]
non_zero_coefficients_df.sort_values(by='Coefficient', ascending=True, inplace=T
non_zero_coefficients_df
```

```
<ipython-input-71-5a442a5c6ca3>:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  non_zero_coefficients_df.sort_values(by='Coefficient', ascending=True, inplace=
True)
```

Out[71]:

	Coefficient
importance_god	-0.035139
importance_religion_3	-0.028000
justifiable_death_penal	-0.025709
info_source_socia_media	-0.024351
trust_ppl_other_rel_4	-0.017302
voter_choice_3	-0.016538
importance_politics_4	-0.012996
votes_count_fairly_4	-0.009754
membership_other_org_1	-0.009390
most_impo_for_country_2	-0.008163
importance_politics_2	-0.006488
importance_democracy	-0.004588
info_source_tv	-0.004313
only_acc_my_rel_4	-0.004120
religion_always_right_2	-0.003727
info_source_newspaper	-0.001569
membership_pol_party_1	0.001947
valut_rel_pol_system_2	0.002010
justifiable_prostitution	0.003343
importance_friends_2	0.004352
happiness_2	0.004633
demo_must_protect_rights	0.007551
only_acc_my_rel_3	0.007984
trust_family_3	0.013855
trust_ppl_other_nat_4	0.018845
aims_of_country_4	0.020551
nation_pride	0.022784
important_child_responsibility_2	0.023333
children_parent_duty	0.024076
strong_leader_3	0.028685
justifiable_cheating_tax	0.028836
valut_dem_pol_system_4	0.029629
valut_dem_pol_system_3	0.031769

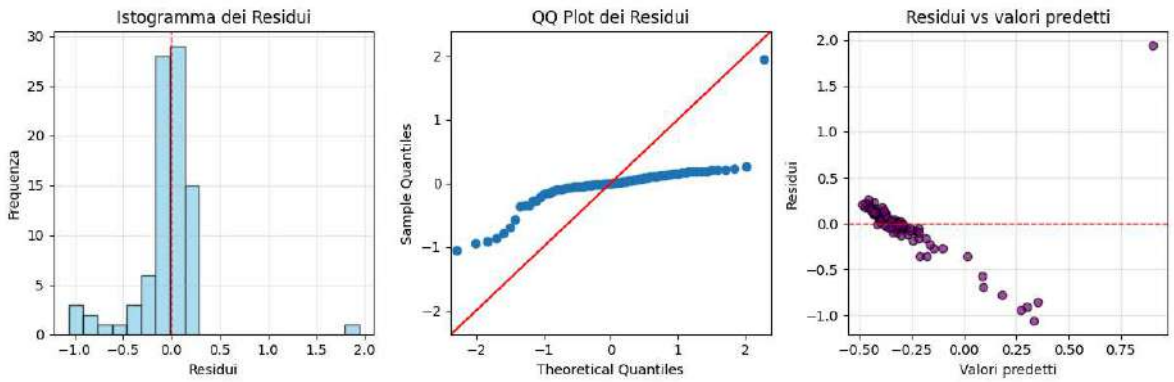
	Coefficient
security_neigh_3	0.033747
justifiable_stealing	0.037910
importance_religion_4	0.040736
justifiable_violence_other_ppl	0.054843
justifiable_beat_wife	0.095567
goal_life_parent_proud_4	0.102669
justifiable_pol_viol	0.150761

In [72]:
```
# seleziono dati che mi interessano per fare confronti
MSE_LASSO = mean_squared_error(y_test, y_pred)
Predittori_LASSO = len(non_zero_coefficients_df)
print('MSE: ', MSE_LASSO)
```

```
MSE:  0.11529565661492108
```

In [73]:
```
# Residui e valori predetti
residuals = y_test - y_pred
y_pred = lasso_cv.predict(X_test)
# Crea la figura con tre sottotrame affiancate
fig, axes = plt.subplots(1, 3, figsize=(12,4))  # 1 riga, 3 colonne

# Istogramma dei residui
axes[0].hist(residuals, bins=20, color='skyblue', edgecolor='black', alpha=0.7)
axes[0].set_title("Istogramma dei Residui")
axes[0].set_xlabel("Residui")
axes[0].set_ylabel("Frequenza")
axes[0].axvline(0, color='red', linestyle='dashed', linewidth=1)  # Linea vertic
axes[0].grid(alpha=0.3)

# QQ plot dei residui
sm.qqplot(residuals, line='45', ax=axes[1])
axes[1].set_title("QQ Plot dei Residui")

# Scatter plot residui vs fitted values
axes[2].scatter(y_pred, residuals, alpha=0.7, color='purple', edgecolor='black')
axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)  # Linea orizzo
axes[2].set_title("Residui vs valori predetti")
axes[2].set_xlabel("Valori predetti")
axes[2].set_ylabel("Residui")
axes[2].grid(alpha=0.3)

# Ottimizza gli spazi e mostra i grafici
plt.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:486: UserWarning: X has f
eature names, but LassoCV was fitted without feature names
  warnings.warn(
```



In [74]:
```
# Filtra le colonne di X_train_scaled e X_test_scaled per includere solo quelle
significant_columns = non_zero_coefficients_df.index  # Ottieni i nomi delle col
X_train_significant = X_train_scaled[:, X.columns.get_indexer(significant_column
X_test_significant = X_test_scaled[:, X.columns.get_indexer(significant_columns)
```

In [75]:
```
# Fitta il nuovo modello LassoCV sui dati ridotti
lasso_cv_significant = LassoCV(eps=0.001, n_alphas=1000, max_iter=10000, cv=5, n
lasso_cv_significant.fit(X_train_significant, y_train)
# Ottieni i nuovi coefficienti
coefficients_significant = lasso_cv_significant.coef_
```

In [76]:
```
# Visualizza il miglior parametro alpha trovato
print(f"Miglior valore di alpha: {lasso_cv_significant.alpha_}")
print('-----------------')
print("Coefficienti del modello Lasso:")
print(lasso_cv_significant.coef_)
```

```
Miglior valore di alpha: 0.00028105344834342914
-----------------
Coefficienti del modello Lasso:
[-0.03950947 -0.03555657 -0.03902508 -0.03670823 -0.03767802 -0.02413655
 -0.02632488 -0.01042017 -0.02831815 -0.02013325 -0.01468768  0.0006115
 -0.00656414 -0.01370386 -0.01813448 -0.00727671  0.0147382   0.00735059
  0.01053662  0.01249062  0.01414116  0.02176062  0.01409665  0.02012649
  0.04962008  0.03759877  0.03421131  0.02581487  0.03066852  0.0385906
  0.03806913  0.02808821  0.04707426  0.04607963  0.04245513  0.04072594
  0.06403855  0.09071473  0.09574494  0.15103241]
```

In [77]:
```
# Calcolo Adj_R_squared del modello
r_squared = lasso_cv.score(X_train, y_train)

# Numero di osservazioni e predittori
n = X_train.shape[0]  # Numero di osservazioni
p = X_train.shape[1]  # Numero di variabili predittive

# Calcola l'R^2 aggiustato
r_squared_adj_lasso_rid_sign = 1 - (1 - r_squared) * (n - 1) / (n - p - 1)
R_quadro_Adj_LASSO_rid_sign = r_squared_adj_lasso_rid_sign
R_quadro_Adj_LASSO_rid_sign
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:486: UserWarning: X has f
eature names, but LassoCV was fitted without feature names
  warnings.warn(
```

Out[77]: 0.4089698689860347

In [78]:
```
# Crea un nuovo DataFrame con i coefficienti
coefficients_significant_df = pd.DataFrame(coefficients_significant, index=signi
```

```
coefficients_significant_df.sort_values(by='Coefficient', ascending=True, inplac
# Mostra i coefficienti del nuovo modello
print(coefficients_significant_df)
```

```
                                Coefficient
importance_god                    -0.039509
justifiable_death_penal           -0.039025
trust_ppl_other_rel_4             -0.037678
info_source_socia_media           -0.036708
importance_religion_3             -0.035557
membership_other_org_1            -0.028318
importance_politics_4             -0.026325
voter_choice_3                    -0.024137
most_impo_for_country_2           -0.020133
religion_always_right_2           -0.018134
importance_politics_2             -0.014688
only_acc_my_rel_4                 -0.013704
votes_count_fairly_4              -0.010420
info_source_newspaper             -0.007277
info_source_tv                    -0.006564
importance_democracy               0.000612
valut_rel_pol_system_2             0.007351
justifiable_prostitution           0.010537
importance_friends_2               0.012491
only_acc_my_rel_3                  0.014097
happiness_2                        0.014141
membership_pol_party_1             0.014738
trust_family_3                     0.020126
demo_must_protect_rights           0.021761
important_child_responsibility_2   0.025815
valut_dem_pol_system_4             0.028088
children_parent_duty               0.030669
nation_pride                       0.034211
aims_of_country_4                  0.037599
justifiable_cheating_tax           0.038069
strong_leader_3                    0.038591
importance_religion_4              0.040726
justifiable_stealing               0.042455
security_neigh_3                   0.046080
valut_dem_pol_system_3             0.047074
trust_ppl_other_nat_4              0.049620
justifiable_violence_other_ppl     0.064039
justifiable_beat_wife              0.090715
goal_life_parent_proud_4           0.095745
justifiable_pol_viol               0.151032
```

In [79]:

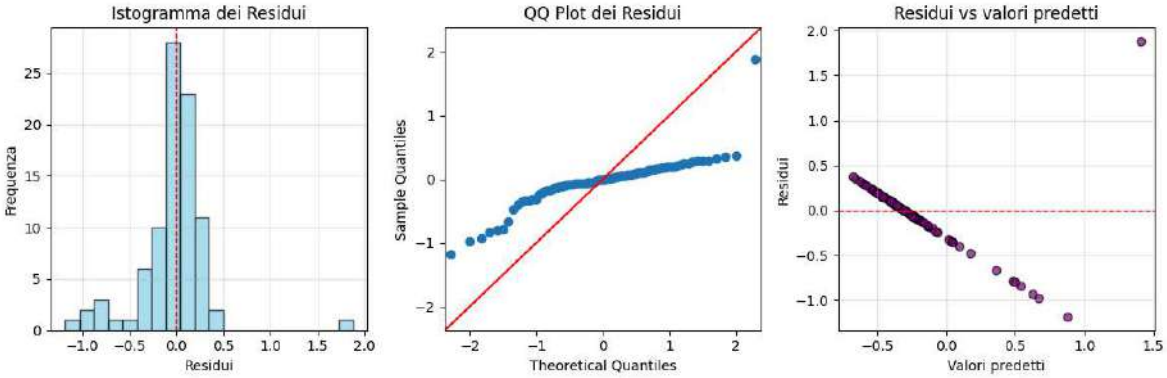
```
y_pred_significant = lasso_cv_significant.predict(X_test_significant)
MSE_LASSO_sign = mean_squared_error(y_test, y_pred_significant)
Predittori_LASSO_sign = len(coefficients_significant_df)
print('MSE: ', MSE_LASSO_sign)
```

```
MSE:  0.13250258984590021
```

In [80]:

```
# Residui e valori predetti
residuals = y_test - y_pred_significant

# Crea la figura con tre sottotrame affiancate
fig, axes = plt.subplots(1, 3, figsize=(12,4))  # 1 riga, 3 colonne

# Istogramma dei residui
axes[0].hist(residuals, bins=20, color='skyblue', edgecolor='black', alpha=0.7)
```

```
axes[0].set_title("Istogramma dei Residui")
axes[0].set_xlabel("Residui")
axes[0].set_ylabel("Frequenza")
axes[0].axvline(0, color='red', linestyle='dashed', linewidth=1)  # Linea vertic
axes[0].grid(alpha=0.3)

# QQ plot dei residui
sm.qqplot(residuals, line='45', ax=axes[1])
axes[1].set_title("QQ Plot dei Residui")

# Scatter plot residui vs fitted values
axes[2].scatter(y_pred_significant, residuals, alpha=0.7, color='purple', edgecc
axes[2].axhline(0, color='red', linestyle='dashed', linewidth=1)  # Linea orizzo
axes[2].set_title("Residui vs valori predetti")
axes[2].set_xlabel("Valori predetti")
axes[2].set_ylabel("Residui")
axes[2].grid(alpha=0.3)

# Ottimizza gli spazi e mostra i grafici
plt.tight_layout()
plt.show()
```



Confronto tra modelli

In [81]:

```
# creo df con confronto tra modelli

confronto_df = pd.DataFrame({
    'Modello': ['OLS', 'OLS ridotto','OLS rid_no_out', 'OLS rid_sign', 'OLS robu
                'OLS robusto_rid_sign', 'LASSO_rid', 'LASSO_rid_sign'],
    'N. Predittori': [Predittori_OLS, Predittori_OLS_rid, Predittori_OLS_rid_no_
                      Predittori_OLS_rid_rob, Predittori_OLS_rid_rob_sign,
                      Predittori_LASSO, Predittori_LASSO_sign],
    'F_value': [F_OLS, F_OLS_rid, F_OLS_rid_no_out, F_OLS_rid_sign, '--', '--','
    'R_quadro_adjusted': [R_quadro_Adj_OLS, R_quadro_Adj_OLS_rid, R_quadro_Adj_C
                          '--', '--', R_quadro_Adj_LASSO, '--'],
    'Log-likelihood': [Likelihood_OLS, Likelihood_OLS_rid,Likelihood_OLS_rid_no_
                       '--', '--', '--', '--'],
    'AIC': [AIC_OLS, AIC_OLS_rid,AIC_OLS_rid_no_out, AIC_OLS_rid_sign, '--', '--
    'BIC': [BIC_OLS, BIC_OLS_rid,BIC_OLS_rid_no_out, BIC_OLS_rid_sign, '--', '--
    'MSE': [MSE_OLS, MSE_OLS_rid,MSE_OLS_rid_no_out, MSE_OLS_rid_sign, MSE_OLS_r
            MSE_OLS_rid_rob_sign, MSE_LASSO, MSE_LASSO_sign]
})

pd.set_option('display.expand_frame_repr', False); pd.set_option('display.colhea
```

```
confronto_df[confronto_df.select_dtypes(include=['number']).columns] = confronto
print(confronto_df)
```

```
        Modello         N. Predittori  F_value   R_quadro_adjusted Log-likelihoo
d     AIC         BIC        MSE
0                  OLS       147        2.403111      0.383908       -293.21174
882.42348  1445.583456  0.342
1           OLS ridotto        58        4.334205      0.364744      -364.396513
844.793027  1065.490855  0.526
2        OLS rid_no_out        57        9.133174      0.611121        11.442331
93.115338   307.156186  0.054
3          OLS rid_sign         9       56.607486      0.629149        -8.723251
37.446503    74.350097  0.062
4       OLS robusto_rid        57              --            --               --
--           --  0.131
5  OLS robusto_rid_sign        13              --            --               --
--           --  0.131
6             LASSO_rid        40              --       0.40897               --
--           --  0.115
7        LASSO_rid_sign        40              --            --               --
--           --  0.133
```

Osservazioni

Per LASSO (regolarizzazione L1) è stata effettuata una divisione in training e test set e si è ottenuta una riduzione ulteriore della dimensionalità della matrice dei predittori rispetto a quella effettuata in precedenza. Comunque non sembra raggiungere valori di bontà di adattamento equivalenti a quelli dei modelli dove non è stata utilizzata la suddivisione in train e test set (tutti gli altri).

La modalità con la quale è stata ridotta la dimensionalità del dataset,e i valori di R_quadro aggiustato, MSE e numero dei predittori in OLS rid_sign_no_out, suggeriscono che il modello sia sovradattato ai dati.

Il valore di MSE nel LASSO risulta essere migliore del valore di MSE nel modello robusto, suggerendo nel complesso che la soluzione LASSO sia quella che raggiunge il compromesso migliore tra bias e varianza, rispetto a tutti gli altri.

Tuttavia è da notare che si è in presenza di elevata collinearità tra predittori e la matrice non è sparsa, (come si vede dai risultati del LASSO), e che pertanto i modelli come ridge regression (regolarizzazione L2) o elastic net (regolarizzazione L1 e L2) potrebbero fornire - usando OLS - stime migliori.