```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

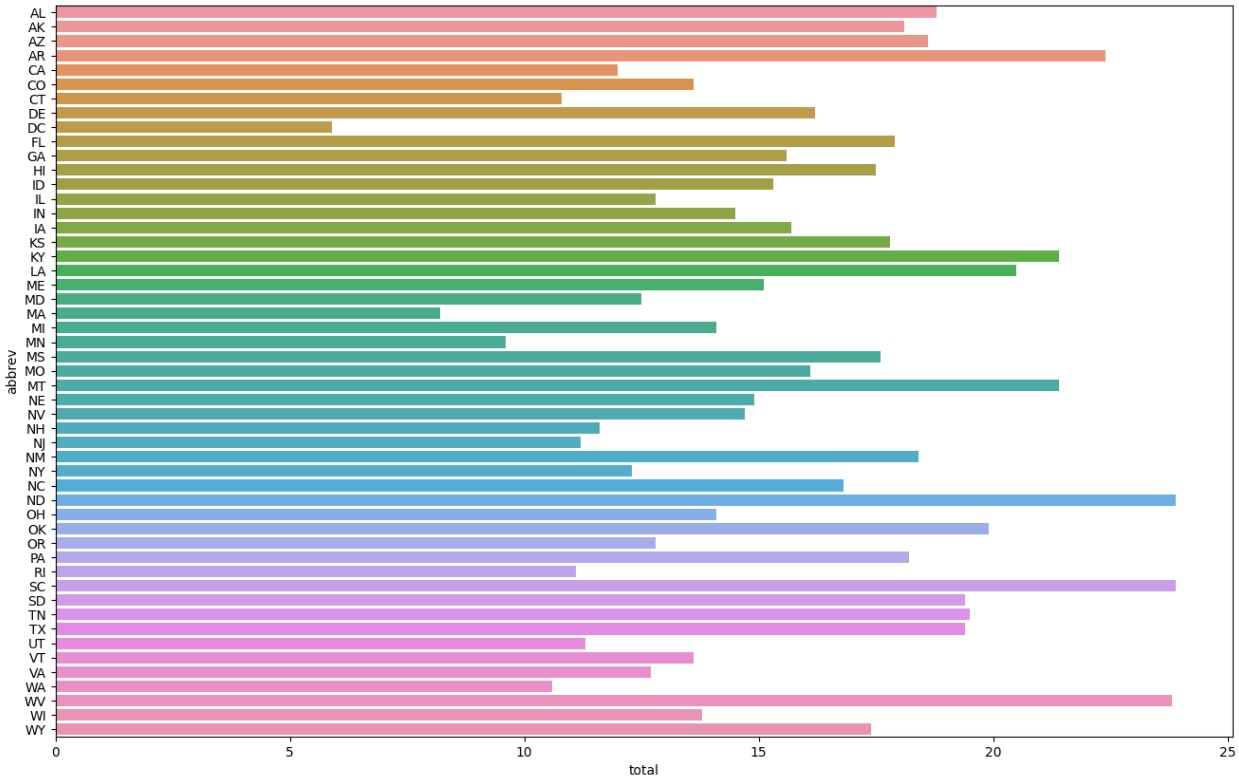
```
dataset=sns.load_dataset('car_crashes')

dataset.head()
```

```
   total  speeding  alcohol  not_distracted  no_previous  ins_premium
\
0   18.8     7.332    5.640          18.048       15.040       784.55

1   18.1     7.421    4.525          16.290       17.014      1053.48

2   18.6     6.510    5.208          15.624       17.856       899.47

3   22.4     4.032    5.824          21.056       21.280       827.34

4   12.0     4.200    3.360          10.920       10.680       878.41


   ins_losses abbrev
0      145.08     AL
1      133.93     AK
2      110.35     AZ
```

```
3       142.39      AR
4       165.63      CA
```

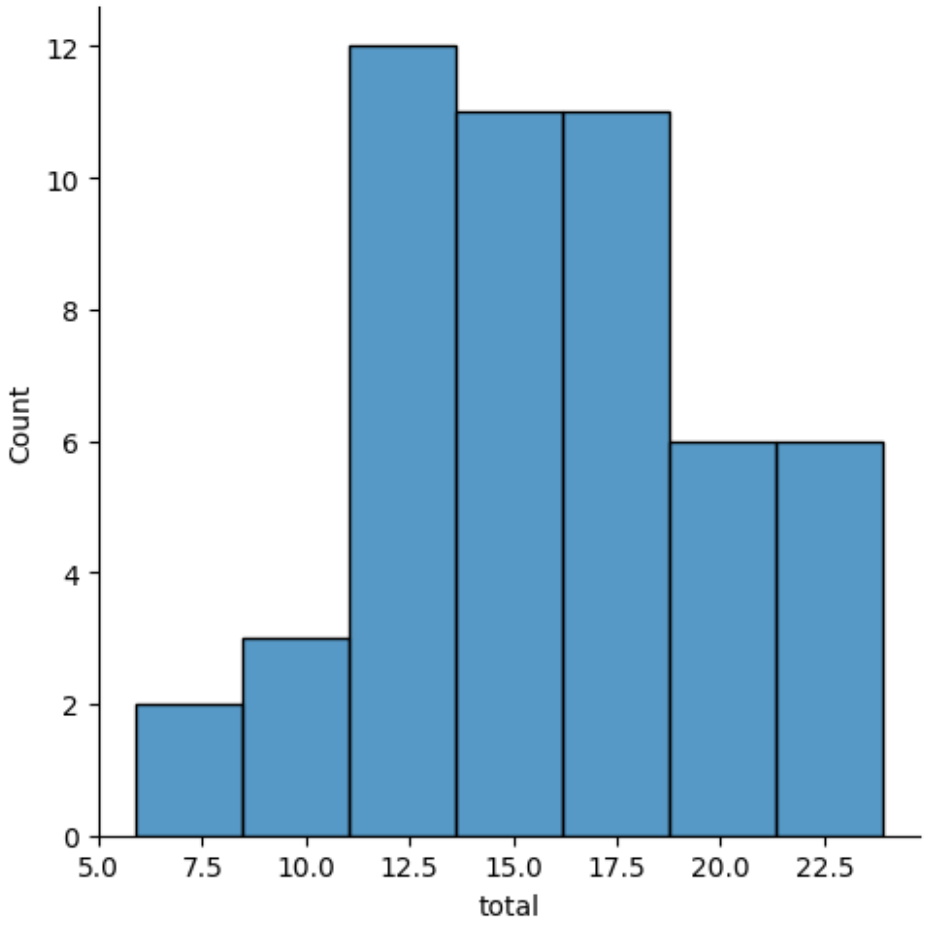
```
plt.subplots(figsize=(16,10))
sns.barplot(x='total',y= 'abbrev',data=dataset,orient='h')
#infernce: For ND,SC,WV abbrevations  total is very high and equal
```

```
<Axes: xlabel='total', ylabel='abbrev'>
```



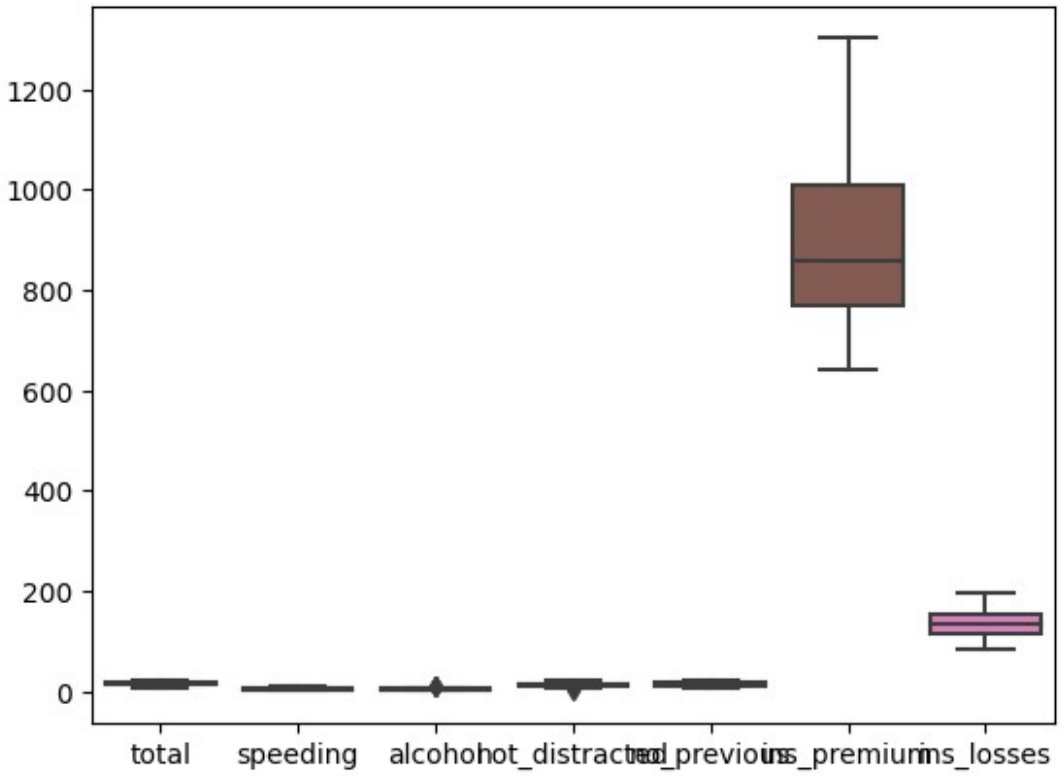

```
sns.displot(dataset['total'])
#inference: here for total 15.0 the count is maximum
```

```
<seaborn.axisgrid.FacetGrid at 0x7ccff1727b20>
```



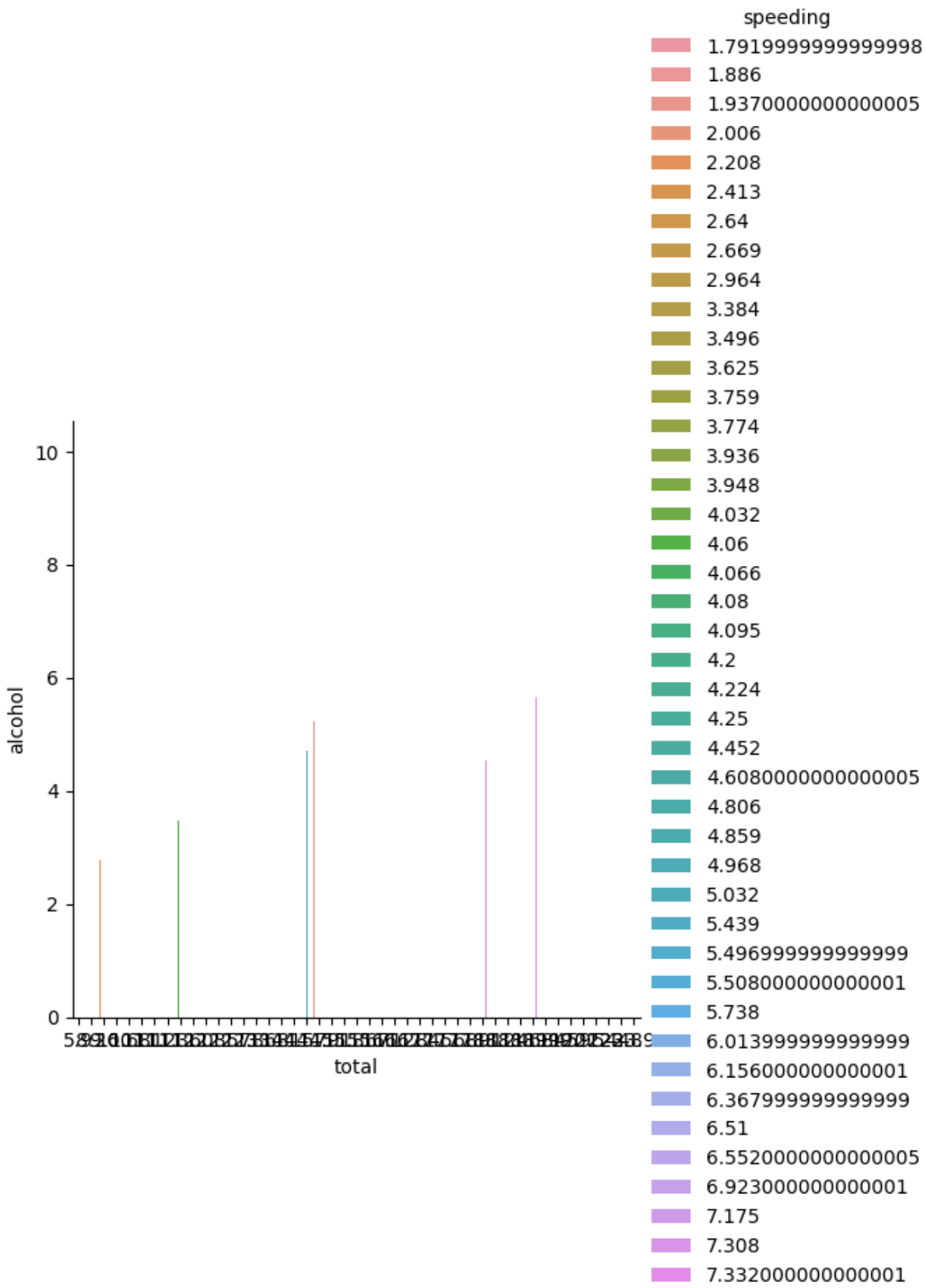

```
sns.boxplot(dataset)
#inference: there is a big outlier for ins_permium
```

```
<Axes: >
```



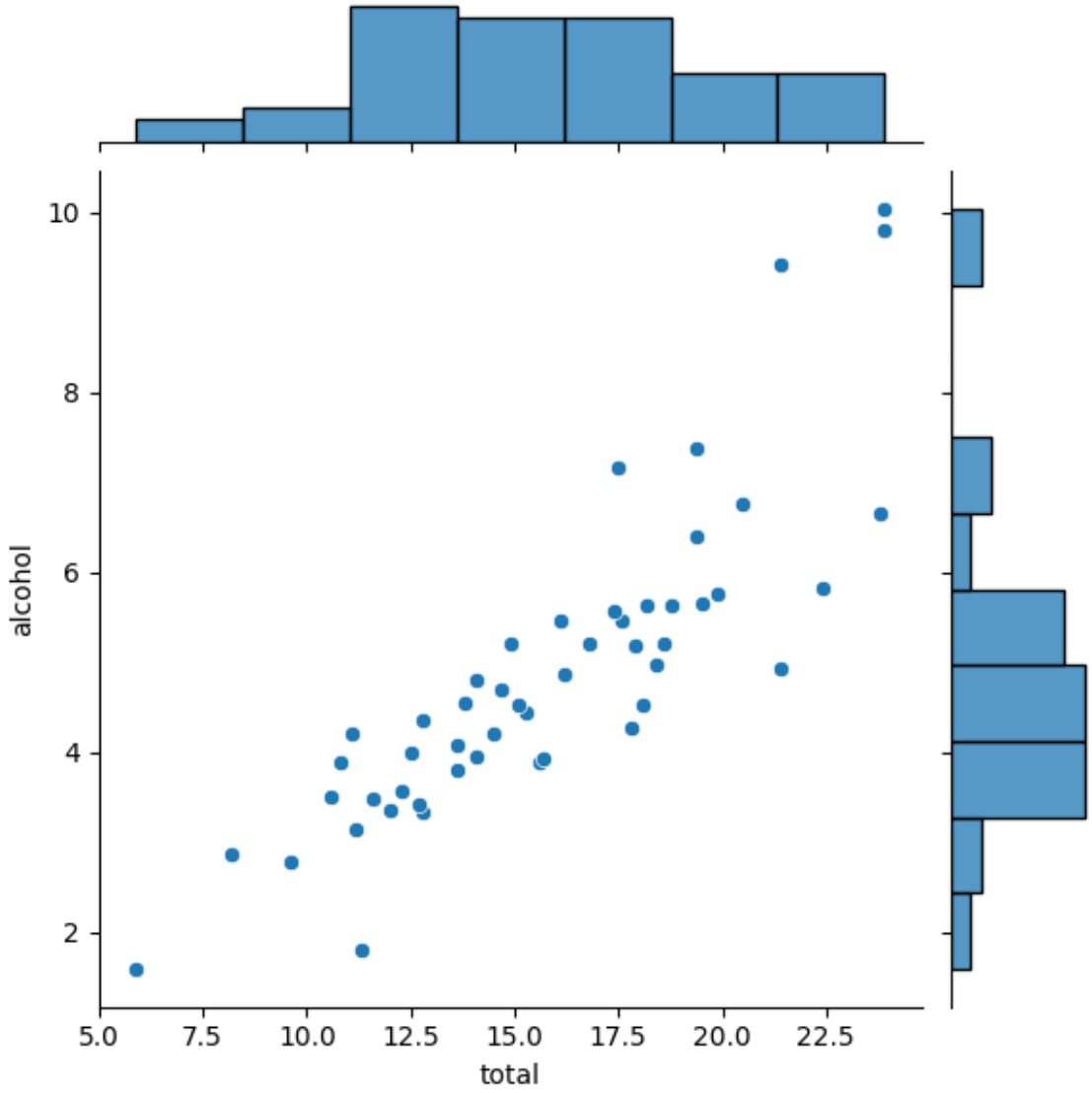

```
sns.catplot(data=dataset, kind="bar", x="total", y="alcohol",
hue="speeding")
#inference: the plot went from 1 to 9 but alcohol consumption became
low to up and up to down
```

```
<seaborn.axisgrid.FacetGrid at 0x7ccff1535240>
```


speeding
1.7919999999999998
1.886
1.9370000000000005
2.006
2.208
2.413
2.64
2.669
2.964
3.384
3.496
3.625
3.759
3.774
3.936
3.948
4.032
4.06
4.066
4.08
4.095
4.2
4.224
4.25
4.452
4.6080000000000005
4.806
4.859
4.968
5.032
5.439
5.496999999999999
5.508000000000001
5.738
6.013999999999999
6.156000000000001
6.367999999999999
6.51
6.5520000000000005
6.923000000000001
7.175
7.308
7.332000000000001
alcohol
0
2
4
6
8
10
total

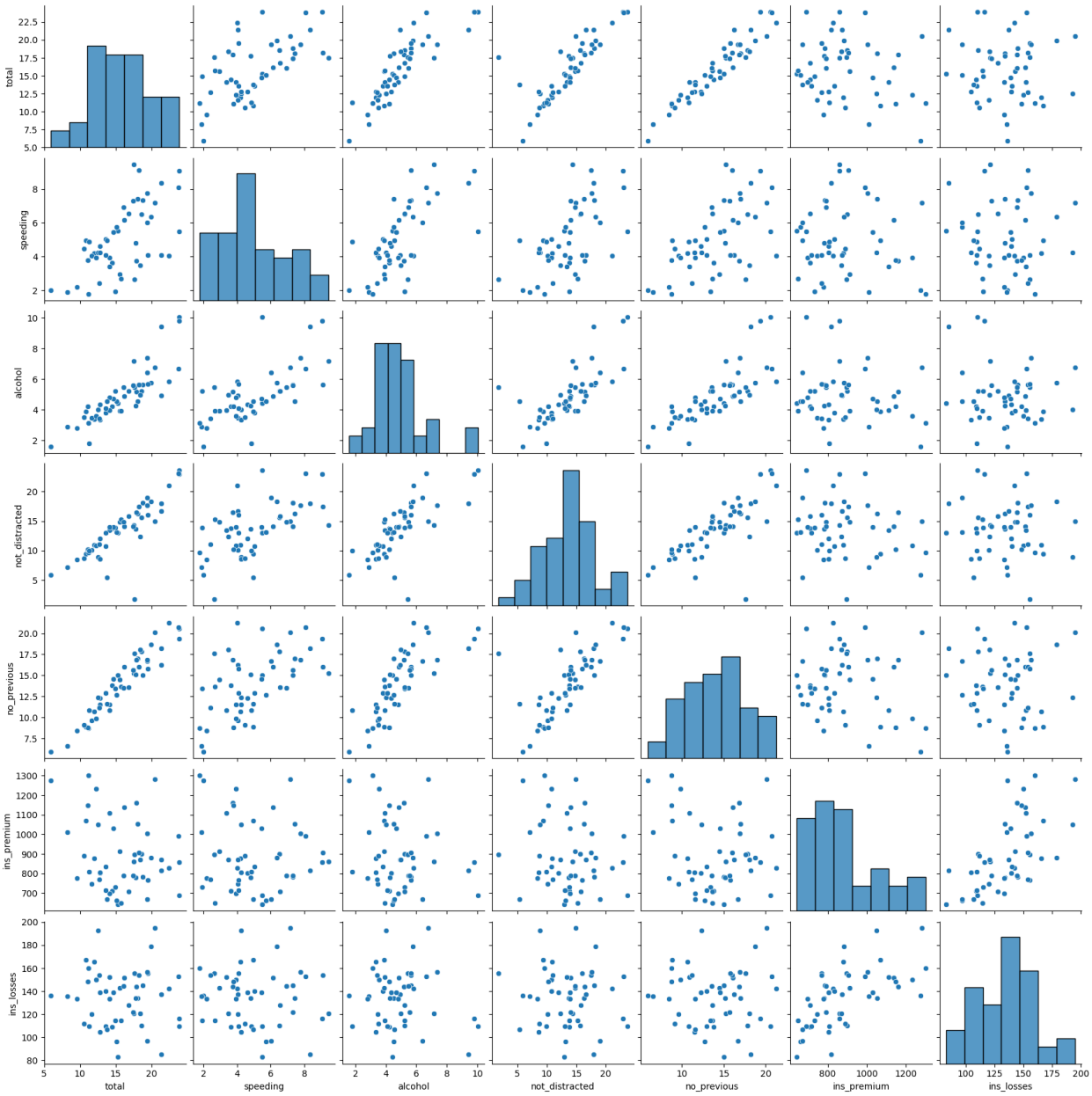
```
sns.jointplot(x='total',y='alcohol',data=dataset)
#inference: here we observe that alcohol consumption is raising as
total increasing
```

```
<seaborn.axisgrid.JointGrid at 0x7cd02a24ece0>
```



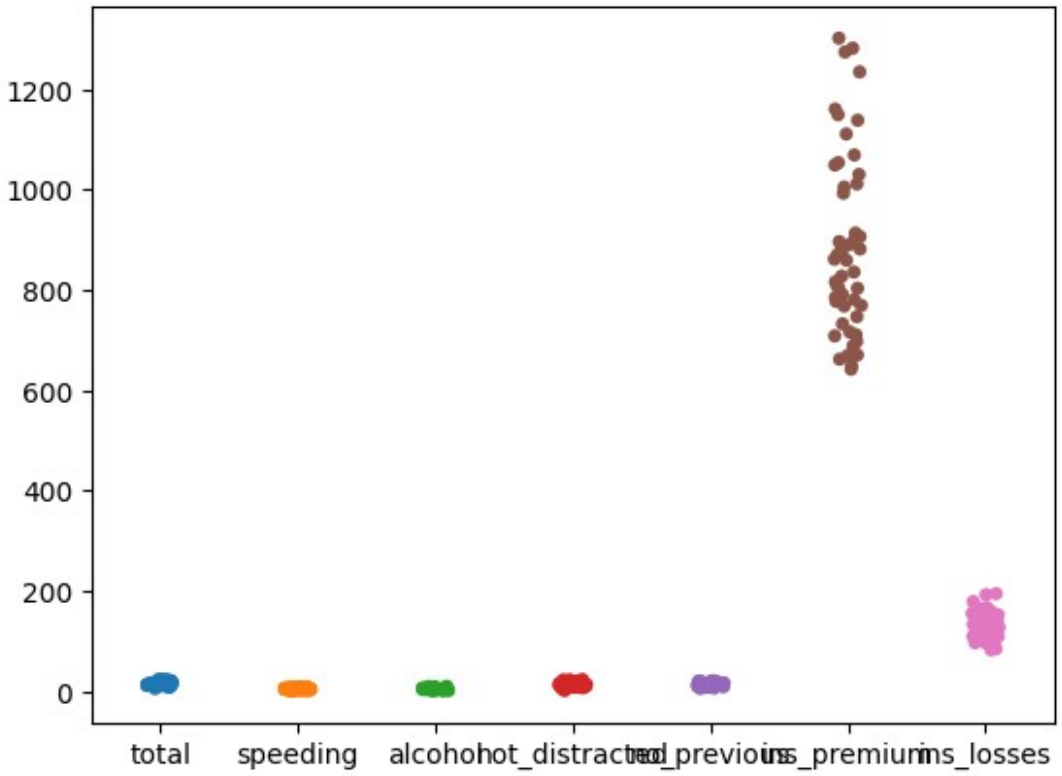

```
sns.pairplot(dataset)
#inference: here we can observe different parameters having up down
increase of total,speeding,alcohol,not distracted,no
previous,ins_premium,ins_losses
```

```
<seaborn.axisgrid.PairGrid at 0x7ccfee48e920>
```

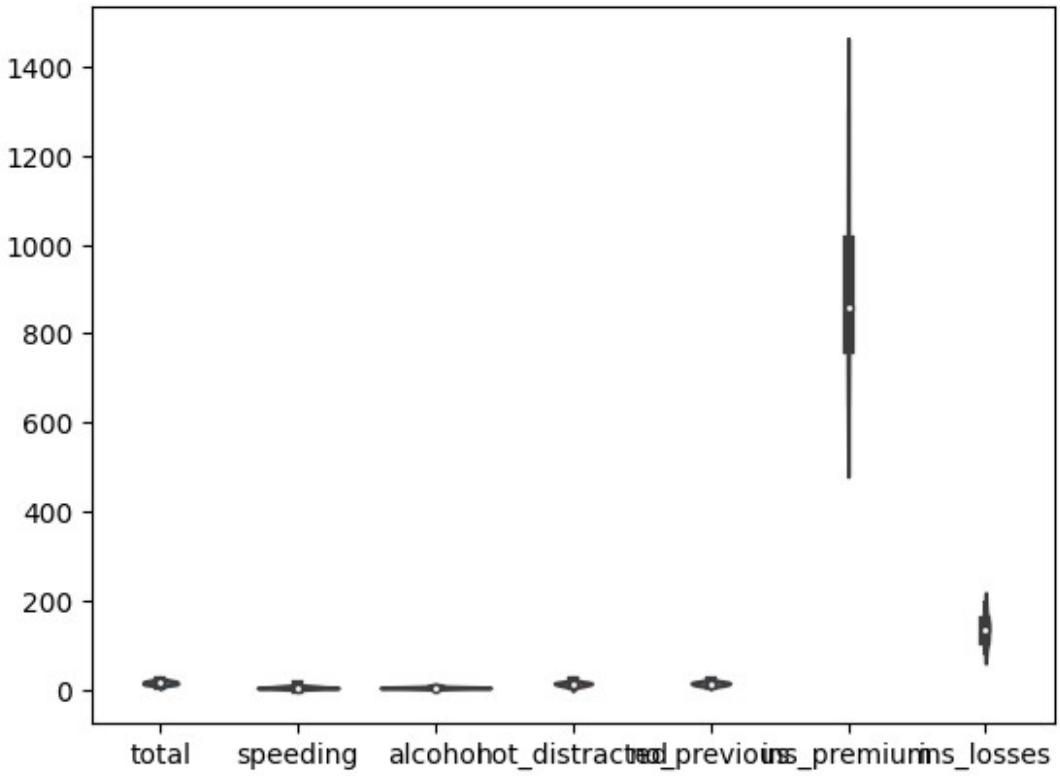



```
sns.stripplot(dataset)
#inference: the ins_premium total is high of car crashes
```

```
<Axes: >
```

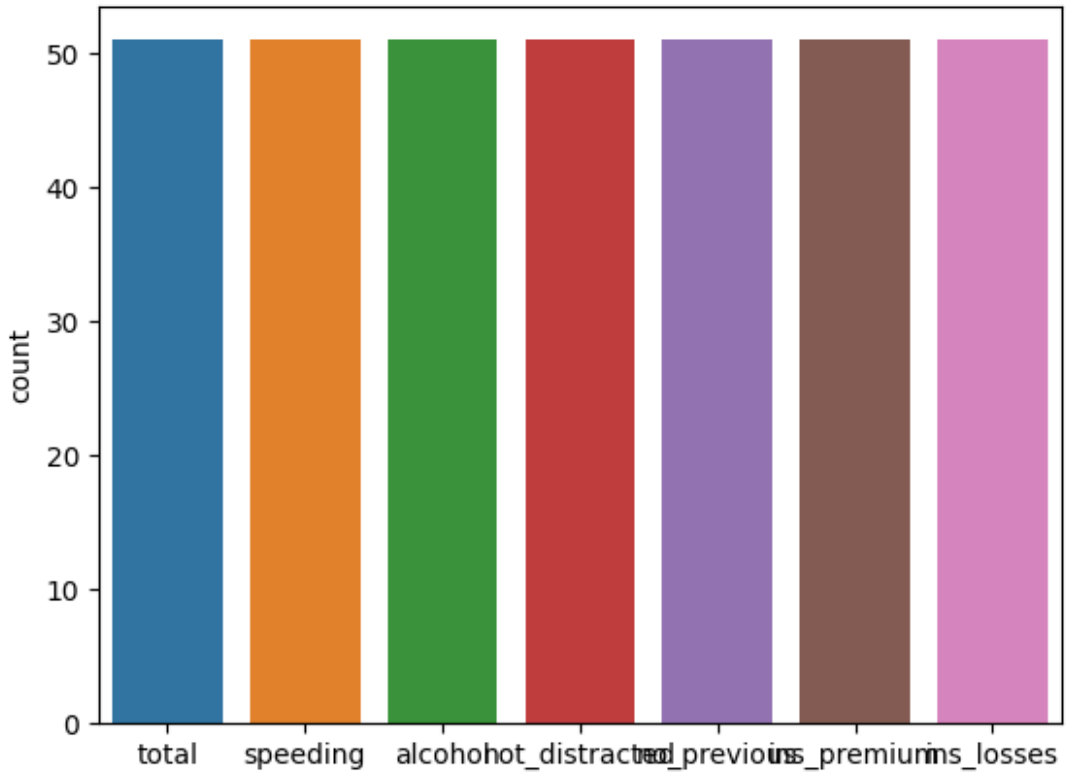



```
sns.violinplot(dataset)
#inference: the data gets shrinked here in this plot-reference and for
ins_premius it went high

<Axes: >
```



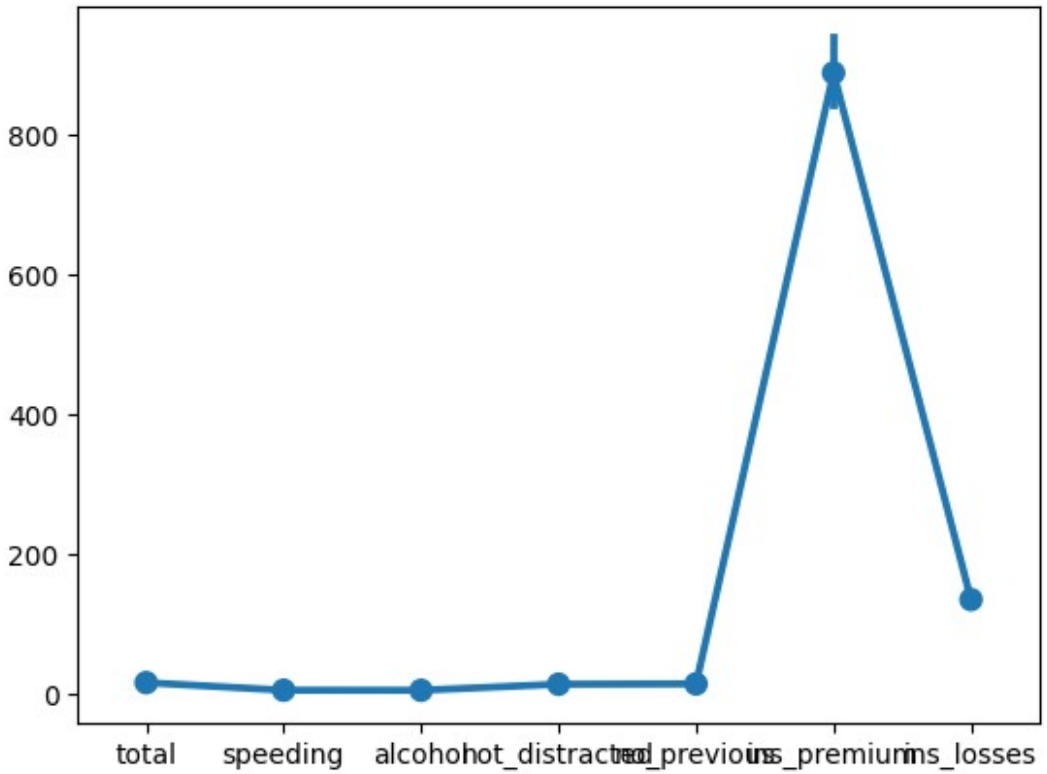

```
sns.countplot(dataset)
#inference: it depicts the count of all parameters of car crashes
```

```
<Axes: ylabel='count'>
```




```
sns.pointplot(dataset)
#inference: as depicted in the dataset the ins_premiums has more no of
carcrashes.
```

```
<Axes: >
```




```
!jupyter nbconvert --to html /Assignment 2.ipynb

[NbConvertApp] WARNING | pattern '/Assignment' matched no files
[NbConvertApp] WARNING | pattern '2.ipynb' matched no files
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE
RELEASES.

Options
=======
The options below are convenience aliases to configurable class-
options,
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
```

```
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
    Execute the notebook prior to export.
    Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an
error and include the error message in the cell output (the default
behaviour is to abort conversion). This flag is only relevant if '--
execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting
notebook with default basename 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False --
NbConvertApp.export_format=notebook --FilesWriter.build_directory=]
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False --
NbConvertApp.export_format=notebook --FilesWriter.build_directory= --
ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True --
TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True --
TemplateExporter.exclude_input=True --
TemplateExporter.exclude_input_prompt=True]
--allow-chromium-download
    Whether to allow downloading chromium if no suitable version is
found on the system.
    Equivalent to: [--WebPDFExporter.allow_chromium_download=True]
--disable-chromium-sandbox
```

```
    Disable chromium security sandbox when converting to PDF..
    Equivalent to: [--WebPDFExporter.disable_sandbox=True]
--show-input
    Shows code input. This flag is only useful for dejavu users.
    Equivalent to: [--TemplateExporter.exclude_input=False]
--embed-images
    Embed the images as base64 dataurls in the output. This flag is
only useful for the HTML/WebPDF/Slides exports.
    Equivalent to: [--HTMLExporter.embed_images=True]
--sanitize-html
    Whether the HTML in Markdown cells and cell outputs should be
sanitized..
    Equivalent to: [--HTMLExporter.sanitize_html=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN',
'ERROR', 'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown',
'notebook', 'pdf', 'python', 'rst', 'script', 'slides', 'webpdf']
            or a dotted object name that represents the import path
for an
            ``Exporter`` class
    Default: ''
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_name]
--template-file=<Unicode>
    Name of the template file to use
    Default: None
    Equivalent to: [--TemplateExporter.template_file]
--theme=<Unicode>
    Template specific theme(e.g. the name of a JupyterLab CSS theme
distributed
    as prebuilt extension for the lab template)
    Default: 'light'
    Equivalent to: [--HTMLExporter.theme]
--sanitize_html=<Bool>
    Whether the HTML in Markdown cells and cell outputs should be
sanitized.This
```

```
    should be set to True by nbviewer or similar tools.
    Default: False
    Equivalent to: [--HTMLExporter.sanitize_html]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                can only be used when converting one notebook at a
time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                    to output to the directory of each
notebook. To recover
                                    previous default behaviour
(outputting to the current
                                    working directory) use . as the flag
value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url
pointing to a copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to
a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of
the
            current directory (from which the server is run).
            See the usage documentation

(https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html-
slideshow)
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
    The nbformat version to write.
```

```
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb --to html

            Options include ['asciidoc', 'custom', 'html', 'latex',
'markdown', 'notebook', 'pdf', 'python', 'rst', 'script', 'slides',
'webpdf'].

            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates.
LaTeX includes
            'base', 'article' and 'report'.  HTML includes 'basic',
'lab' and
            'classic'. You can specify the flavor of the format used.

            > jupyter nbconvert --to html --template lab
mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post
serve

            Multiple notebooks can be given at the command line in a
couple of
            different ways:

            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb

            or you can specify the notebooks list in a config file,
containing::

                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]
```

```
            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```