TASK 1 - DATA ANALYSIS

1 - Introduction and importing dataset

This dataset was downloaded through data.gov.ie. The data was collected by Google and Dublin City Council as part of a project named Air View Dublin. A car equiped with multiple sensors to detect concentration of pollutants drove through the roads of Dublin measuring street by street air quality. The measures were made at 1-second intervalls and the data collected was split apart in 2 different datasets, one of them organized by measurings on time, which depicts the car in every measuring grouped by time, and a second one data points were aggregated in aproximately 50m road segments. The latter is object of this study. This study aims to analyse the variables of the dataset and its chracteristics. The direct link for the dataset is: https://data.gov.ie/dataset/google-airview-data-dublin-city/resource/f3b5c4bf-5646-4f0b-b4f6-8e8beebcff3b?inner_span=True

In [17]:

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

airquality_df=pd.read_csv("https://data.smartdublin.ie/dataset/4976e11e-a015-4ef9-9179-dc
```

In [18]:

```
airquality_df.head()
```

Out[18]:

	road_id	the_geom	osm_id	osm_code	osm_fclass	osm_name	osm_ref	osm_onewa
0	3633278	LINESTRING(-6.156470225 53.394400525, -6.15665...	497788125	5141	service	NaN	NaN	
1	3639035	LINESTRING(-6.3266322 53.3421535, -6.3266241 5...	500417276	5141	service	NaN	NaN	
2	2099409	LINESTRING(-6.1891464 53.3795598, -6.1895315 5...	236680313	5141	service	NaN	NaN	
3	3636088	LINESTRING(-6.2796231 53.3262885, -6.2796544 5...	498987932	5141	service	NaN	NaN	
4	3962473	LINESTRING(-6.264441 53.3131986, -6.2644378 53...	684445633	5141	service	NaN	NaN	

5 rows × 30 columns

2 - Reading the data and creating a dictionary

The first step is just visualizing the name of the columns.

In [19]:

```
#getting basic information of the dataset
airquality_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24694 entries, 0 to 24693
Data columns (total 30 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   road_id        24694 non-null  int64  
 1   the_geom       24694 non-null  object 
 2   osm_id         24694 non-null  int64  
 3   osm_code       24694 non-null  int64  
 4   osm_fclass     24694 non-null  object 
 5   osm_name       20209 non-null  object 
 6   osm_ref        6415 non-null   object 
 7   osm_oneway     24694 non-null  object 
 8   osm_maxspeed   24694 non-null  int64  
 9   osm_layer      24694 non-null  int64  
 10  osm_bridge     24694 non-null  bool   
 11  osm_tunnel     24694 non-null  bool   
 12  NO2points      24694 non-null  int64  
 13  NO2drives      24495 non-null  float64
 14  NO2_ugm3       24495 non-null  float64
 15  NOpoints       24694 non-null  int64  
 16  NOdrives       24610 non-null  float64
 17  NO_ugm3        24610 non-null  float64
 18  CO2points      24694 non-null  int64  
 19  CO2drives      24489 non-null  float64
 20  CO2_mgm3       24489 non-null  float64
 21  COpoints       24694 non-null  int64  
 22  COdrives       24648 non-null  float64
 23  CO_mgm3        24648 non-null  float64
 24  O3points       24694 non-null  int64  
 25  O3drives       23446 non-null  float64
 26  O3_ugm3        23446 non-null  float64
 27  PM25points     24694 non-null  int64  
 28  PM25drives     24676 non-null  float64
 29  PM25_ugm3      24676 non-null  float64
dtypes: bool(2), float64(12), int64(11), object(5)
memory usage: 5.3+ MB
```

The dataset is composed by 30 columns and 24693 observations. Among this variables, summarily, 7 are categorical and 23 are numerical. The dataset also does not contain any null value.

In the next step it is print the columns, with their unique values and the number of unique values in each column, to easen the visualization of the dataset and make possible to understand summarily shape of data in every column and certify its classification.

In [20]:
```
#Checking unique values for every column

for column in airquality_df:
    print('Column: {} - Unique Values: {} - Number of unique: {}'.format(column, airqualit
```

```
Column: road_id - Unique Values: [3633278 3639035 2099409 ...   21092   42617  343547] - N
umber of unique: 24694
Column: the_geom - Unique Values: ['LINESTRING(-6.156470225 53.394400525, -6.1566529 53.39
40763)'
 'LINESTRING(-6.3266322 53.3421535, -6.3266241 53.3422119, -6.3266168 53.3422973, -6.32659
35 53.3425677)'
 'LINESTRING(-6.1891464 53.3795598, -6.1895315 53.3799436)' ...
 'LINESTRING(-6.24108335 53.343096975, -6.2411545 53.3427458)'
 'LINESTRING(-6.236942 53.3380911, -6.2368972 53.3381538, -6.2368778 53.3382424, -6.236842
53397864 53.3384035439329)'
 'LINESTRING(-6.24168914461433 53.342807773594, -6.2411545 53.3427458, -6.24096915300442 5
3.3427228196043)'] - Number of unique: 24694
Column: osm_id - Unique Values: [497788125 500417276 236680313 ...  80390834  49057073  51
```

```
269108] - Number of unique: 10192
Column: osm_code - Unique Values: [5141 5122 5121 5134 5114 5111 5115 5123 5131 5135 5112
5132] - Number of unique: 12
Column: osm_fclass - Unique Values: ['service' 'residential' 'unclassified' 'secondary_lin
k' 'secondary'
 'motorway' 'tertiary' 'living_street' 'motorway_link' 'tertiary_link'
 'trunk' 'trunk_link'] - Number of unique: 12
Column: osm_name - Unique Values: [nan 'Cooley Road' 'Quarry Road' ... 'Mespil Road' 'MacM
ahon Bridge'
 'Haddington Road'] - Number of unique: 3238
Column: osm_ref - Unique Values: [nan 'R812' 'L8431' 'L8145' 'R138' 'R816' 'R839' 'R825'
'R818' 'M50'
 'R137' 'L4005' 'R810' 'R131' 'L8041' 'R105' 'R112' 'R132' 'R139' 'R809'
 'L3101' 'L1084' 'R108' 'L8422' 'L8178' 'L2190' 'R102' 'R147' 'R110'
 'R117' 'L1006' 'L8107' 'R801' 'L4022' 'L3031' 'N2' 'R807' 'L8111' 'R104'
 'R803' 'L4021' 'R135' 'R107' 'R805' 'R101' 'R118' 'R820' 'R148' 'R833'
 'R819' 'R806' 'R111' 'R802' 'R804' 'R103' 'N50' 'R824' 'R811' 'L2145'
 'L3080' 'R114' 'R109' 'R808' 'L5704' 'R815' 'R817' 'L1014' 'R840' 'R834'
 'R813' 'R814'] - Number of unique: 70
Column: osm_oneway - Unique Values: ['B' 'F'] - Number of unique: 2
Column: osm_maxspeed - Unique Values: [30  0 50 20 10 16 25 60 15 80 40  8  5] - Number of
unique: 13
Column: osm_layer - Unique Values: [ 0 -1  1 -3  2] - Number of unique: 5
Column: osm_bridge - Unique Values: [False  True] - Number of unique: 2
Column: osm_tunnel - Unique Values: [False  True] - Number of unique: 2
Column: NO2points - Unique Values: [   0    1    2 ...  907 1750 2774] - Number of unique:
1221
Column: NO2drives - Unique Values: [ nan   1.   2. 165.   3.   4.   5.   6.   7.   8.   9.
  10.  12.  11.
 178.  13.  14.  15.  16.  17.  18.  20.  19.  21.  22.  23.  24.  25.
  26.  27.  28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.
  40.  41.  43.  42.  44.  47.  45.  46.  48.  50.  49.  51.  52.  53.
  54.  55.  56.  57.  58.  59.  60.  61.  62.  63.  67.  64.  68.  65.
  66.  69.  71.  70.  72.  74.  76.  78.  73.  79.  77.  81.  80.  82.
  85.  75.  83.  84.  86.  88.  89.  87.  90.  92.  93.  99.  97.  91.
 100. 101.  94. 105. 103. 104.  96.  95.  98. 106. 111. 108. 114. 102.
 109. 115. 116. 110. 112. 107. 122. 126. 131. 128. 130. 113. 129. 137.
 138. 136. 143. 120. 149. 158.] - Number of unique: 131
Column: NO2_ugm3 - Unique Values: [    nan -34.13    3.659 ...   6.016  17.511  30.58 ] -
Number of unique: 19660
Column: NOpoints - Unique Values: [   0    1    2 ... 1230 1916 3382] - Number of unique:
1365
Column: NOdrives - Unique Values: [ nan   1.   2. 269.   3.   4.   6.   5.   7.   8.   9.
10.  11.  12.
 261.  13.  14.  15.  17.  16.  18.  19.  20.  21.  22.  23.  25.  26.
  24.  28.  27.  29.  30.  31.  32.  34.  33.  36.  35.  38.  37.  39.
  41.  42.  43.  40.  44.  46.  45.  47.  50.  49.  48.  51.  52.  53.
  54.  55.  56.  58.  57.  59.  60.  61.  63.  62.  64.  65.  67.  66.
  68.  69.  70.  71.  75.  72.  74.  73.  76.  77.  78.  79.  80.  83.
  82.  85.  84.  86.  91.  87.  89.  95.  94.  96.  90.  88.  93. 101.
 103. 104. 107. 105. 106. 102.  97. 110.  81. 116. 114. 118. 119. 115.
 122. 121. 125. 113.  92. 120. 117. 126. 124.  98. 127. 128. 129. 130.
 131. 134. 135. 136. 142. 163. 156. 150. 158. 151. 157. 152. 162. 166.
 164. 165. 180. 169. 185. 176.] - Number of unique: 145
Column: NO_ugm3 - Unique Values: [    nan -31.723   8.453 ...   4.041  25.822  27.364] - N
umber of unique: 19181
Column: CO2points - Unique Values: [   0    1 7161 ... 1182 1828 2968] - Number of unique:
1274
Column: CO2drives - Unique Values: [ nan   1. 254.   2.   3.   4.   5.   6.   7.   8.   9.
  10.  11.  12.
 249.  13.  14.  15.  16.  17.  18.  20.  19.  21.  23.  22.  24.  25.
  26.  27.  28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.
  40.  41.  42.  43.  44.  46.  45.  48.  47.  49.  50.  51.  52.  53.
  54.  55.  56.  57.  58.  59.  60.  61.  62.  63.  64.  65.  66.  67.
  68.  70.  71.  72.  69.  73.  75.  74.  76.  77.  78.  79.  80.  84.
  83.  81.  82.  85.  86.  89.  88.  91.  92.  87.  90.  95.  94.  93.
```

```
  98.  96. 103. 101. 100. 102.  99. 105. 104. 107. 108. 109. 110. 106.
 114. 112. 111. 113. 115. 118. 116. 120. 117. 122. 125. 121. 126. 123.
 119. 127. 128. 129. 130. 124. 131. 132. 133. 134. 135. 138. 136. 139.
 142. 144. 141. 143. 147. 156. 152. 149. 145. 155. 148. 159. 157. 160.
 162. 164. 166. 188. 197. 200. 202.] - Number of unique: 160
Column: CO2_mgm3 - Unique Values: [    nan 793.601 791.418 ... 819.535 792.295 828.572] -
Number of unique: 20816
Column: COpoints - Unique Values: [   0 8770    1 ... 2090 2435 3465] - Number of unique:
1385
Column: COdrives - Unique Values: [ nan 256.   1.   2.   3.   4.   5.   6.   7.   8.   9.
10. 266.  11.
  12.  13.  14.  15.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.
  26.  27.  28.  29.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.
  40.  41.  42.  43.  44.  45.  46.  47.  48.  49.  50.  51.  52.  53.
  54.  55.  56.  57.  58.  59.  60.  61.  62.  63.  64.  65.  66.  67.
  68.  69.  70.  71.  72.  73.  74.  75.  76.  77.  78.  79.  80.  81.
  82.  83.  84.  85.  86.  87.  88.  89.  90.  91.  92.  93.  94.  95.
  96.  97.  98.  99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 110.
 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124.
 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139.
 140. 141. 142. 143. 144. 145. 146. 147. 148. 150. 152. 156. 159. 160.
 161. 162. 166. 168. 170. 171. 172. 176. 198. 205. 207. 213.] - Number of unique: 165
Column: CO_mgm3 - Unique Values: [  nan 0.341 0.469 0.297 0.442 0.384 0.336 0.303 0.416 0.
377 0.423 0.411
 0.327 0.388 0.293 0.219 0.525 0.376 0.355 0.371 0.896 0.431 0.886 0.365
 0.31  0.329 0.375 0.296 0.326 0.447 0.353 0.473 0.52  0.319 0.418 0.331
 0.505 0.298 0.312 0.424 0.311 0.292 0.345 0.33  0.318 0.432 0.446 0.412
 0.281 0.511 0.304 0.385 0.34  0.42  0.397 0.307 0.793 0.275 0.324 0.325
 0.306 0.764 0.289 0.369 0.46  0.47  0.316 0.288 0.414 0.359 0.267 0.782
 0.356 0.475 0.438 0.284 0.998 0.271 0.334 0.283 0.29  0.342 0.335 0.338
 0.391 0.357 0.413 0.459 0.407 0.754 0.503 0.348 0.277 0.256 0.363 0.364
 1.057 0.315 0.333 0.485 0.305 0.402 0.332 0.368 0.399 0.517 0.382 0.343
 0.339 0.531 0.269 0.302 0.361 0.664 0.347 0.433 0.451 0.346 0.452 0.76
 0.445 0.374 0.587 0.299 0.282 0.435 0.506 0.328 0.398 1.03  0.378 0.497
 0.301 0.362 0.4   0.38  0.48  0.314 0.3   0.417 0.426 0.322 0.366 0.252
 0.395 2.175 0.291 0.367 0.313 0.392 0.51  0.317 0.32  0.321 0.286 0.308
 0.37  0.276 0.449 0.441 0.393 0.43  0.592 0.309 0.527 0.263 0.351 0.28
 0.545 0.434 0.257 0.566 0.483 0.516 0.403 0.502 0.641 0.381 0.35  0.354
 0.405 0.455 0.665 0.358 0.565 0.472 0.584 0.323 0.749 0.295 0.274 0.285
 0.461 0.683 0.474 0.415 0.487 0.513 0.751 0.49  0.515 0.478 0.41  0.736
 0.425 0.36  0.408 0.27  0.858 0.994 0.254 0.396 1.154 0.985 1.04  0.519
 0.349 0.279 0.428 0.755 0.272 0.707 0.421 0.352 0.437 0.246 0.255 0.389
 0.373 0.55  0.596 0.625 0.372 0.268 0.404 0.504 0.863 0.477 0.695 0.489
 0.581 0.409 0.499 0.669 0.549 0.623 0.56  0.878 0.662 0.494 0.386 0.456
 0.427 1.981 0.264 1.439 0.518 0.273 0.248 0.537 0.529 0.835 0.498 0.651
 0.39  0.287 0.728 0.814 0.742 0.574 0.379 0.344 0.586 0.64  0.401 0.294
 0.419 0.688 0.436 0.278 0.383 0.44  0.721 0.731 0.577 0.53  1.211 0.476
 0.486 0.466 0.514 0.337 0.265 0.458 1.116 0.463 1.093 0.538 0.815 0.462
 0.642 0.495 0.509 0.582 0.501 0.539 0.614 0.394 0.557 0.454 0.562 0.482
 0.468 0.597 0.444 0.631 0.716 0.25  0.26  0.262 0.5   0.628 0.555 0.67
 0.251 0.259 0.266 0.258 0.253 0.261 0.387 0.523 0.756 0.406 0.429 0.443
 0.752 0.563 0.457 0.439 0.422 0.45  0.553 0.453 0.554 0.558 0.678 0.859
 0.956 0.613 0.491 0.536 0.507 0.512 0.535 0.575 0.552 0.588 0.247 0.916
 0.59  0.585 0.679 0.786 0.745 0.966 0.611 0.578 0.484 0.828 0.612 0.622
 0.568 0.57  0.542 0.488 0.647 0.618 0.471 0.963 0.448 0.479 0.719 0.589
 0.569 0.621 1.391 0.636 1.293 0.907 0.691 1.937 0.672 0.541 0.675 0.595
 0.551 0.571 0.627 1.995 1.175 0.561 0.62  0.544 0.738 0.819 0.687 0.992
 0.493 0.63  0.681 0.703 0.465 0.629 0.643 0.609 0.979 0.481 0.88  0.508
 0.594 0.533 0.567 0.464 0.546 0.639 0.619 0.524 0.654 0.714 0.779 0.467
 0.843 0.559 0.776 0.548 0.87  0.634 0.635 0.492 0.652 0.547 0.649 0.522
 0.532 0.564 0.723 0.573 0.496 0.65  0.556 0.534 0.684 0.724 0.71  0.583
 0.686 0.616 0.54  0.528 0.701 0.644 0.608 0.543 0.794 0.626 0.526 0.633
 0.846 0.854 1.286 1.257 1.317 1.303 1.268 0.84  0.757 0.888 1.064 1.132
 0.841 0.857 0.86  0.617 0.799 0.848 0.699 0.58  0.521 0.624 0.607 0.661
 0.602 0.615 0.6   0.778 0.61  0.593] - Number of unique: 497
Column: O3points - Unique Values: [    0     1  1701     2     4   102     3     7     9
```

```
     5    12    16
    11    13     8   160    10    21     6   171    18   172  2607    17
    37    25    27   536    33   199    43    14    48    31    15    53
    24    23   122   248    41    26    42    19    29    46   113    20
  2521  2545    30    28    34    56    50   267    52    32    35    96
    40    54  1258   217   294    22    63    36   101   166    38   111
  2522  3125   182   265   103    47   114  1123   138    62    92    90
   426    76    39    78    94    45   154   180    44   459  2709    51
   325    49    71    72    61    68  3201    57   164    88    66    98
   151    64   568   292    59   108   167   110   159   191   230    86
    74    70   162    60   121    91   100    65    75    58   139    84
   107   106    69   117    80   181    67   104    82   131    77    55
   128    93    85   718   337   137  6329   120   125    97    73    99
   109   134   227   202    79   211   146   189   523   118   178   123
    95   346   169    89   194    87   115   132   141   133    81   163
   193   153   170   136   145   161   188    83   157   231   269   119
   305   126   301   105   127   147   129   433   144   315   116   263
   222   421   249   135   250   253   190   124   213   306   241   228
   150   140   207   152   142 15638  1114   187   225   197   185   322
   278   112   143   165   215  2452   229   336   317   130   156   177
   271   340   261   370   195   179   149   210   243   356   304   242
   175   359   291   490   226   277  1986   198   184   320   401   350
   174   205   158   247   273   352   275   218   284   186   206   240
   232   375   196   380   353   168   282   148   381   313   239   256
   219   389   155   257   262   270  2659   296   404   212  1204   201
   251   214   216   233   339   192   363   258   515   209   357   221
   176  1032   387   489   272   204   341   235   431   416   411   333
   203   264   299   238   347   295   400   513   208   173   379   399
   374   220   280   268   297   831   183   223   324   576   334   405
   200   266   409   545   328   318   236   309   279   460   358   504
   415   424   237   505   245   331   287   321   550   255  1038   388
  2467   327   364   662   372   562   398   246   298   274   469  1252
   537   332   618   464   701   535  1862   435   286   311   303   516
   369   316   300   312   658   285   288   629   252  1646  1116   407
  2075   343   289   383  7653   799   567   413   589   461   283   319
   554   939   360   652   326   637   585   408   329   371   473   703
   361   378   344   276   749   281   254   259   234   414   519   365
   696   582   773   410   864   548   451   706  1690   641   470   427
   373   330   683   962   970   437   946   402   575   354   447   472
   290   581   307   457   391   783   314   386   509   865   390   630
   850   682   769   529   681  1031   419   293   367   512   453   308
   335   522   627   454   244   456   478   579   338   577   342   438
   465   499   444   425   440   998  2444  1447   556   558   578   310
   434   574  2295   559   436  4996   822   392   527   835   501   584
   791   467   849   931   366   351   479   869   302   423   393   420
   475   476   492   995   450   759   877   659   700   397   355   771
   871   905  1131  1072  1048   685   563   345   530   590   432   323
   580   987   813   598   670   825   468   442   471   586   396  1192
   495   376   224   534   487  1018   599   422   560  1007   898   583
   776   796   858   439   806   520  1486   736  1021   483   631  1046
   533   441   633   712   755   430   794   518   452   714   521   507
   349   570   648   772   644   607  1180   485   753   778  1319   474
  2031  1393  1296   740   821   872  4389   902   592   514   552   660
  1041] - Number of unique: 637
Column: O3drives - Unique Values: [ nan   1. 171.   2.   3.   4.   5.   6.   7.   8.   9.
10. 169.  11.
  12.  13.  14.  15.  16.  18.  17.  19.  20.  21.  22.  23.  24.  25.
  26.  27.  29.  28.  30.  31.  32.  33.  34.  35.  36.  37.  38.  39.
  40.  41.  42.  43.  44.  46.  45.  47.  48.  50.  49.  52.  53.  51.
  56.  55.  54.  57.  58.  59.  60.  62.  61.  63.  64.  66.  65.  67.
  68.  69.  71.  70.  74.  73.  78.  76.  72.  77.  79.  81.  75.  80.
  82.  83.  85.  86.  88.  87.  89.  84.  93.  91.  94.  90. 100.  98.
  92. 104. 101. 102. 103.  96.  99.  97. 106. 105. 108.  95. 109. 107.
 112. 116. 113. 110. 114. 133. 125. 111. 131. 121. 120. 137. 138. 136.
 140. 124. 126. 139. 128.] - Number of unique: 130
Column: O3_ugm3 - Unique Values: [   nan 54.447 43.125 ... 48.29  38.765 59.397] - Number
```

```
of unique: 6470
Column: PM25points - Unique Values: [   2    1    5 ... 2256 2488 3723] - Number of uniqu
e: 1419
Column: PM25drives - Unique Values: [  1.   2. 251.  nan   3.   4.   5.   6.   7.   8.
9.  10.  12.  11.
  13. 266.  15.  14.  16.  17.  18.  19.  20.  21.  22.  23.  24.  25.
  26.  27.  31.  30.  28.  29.  33.  32.  34.  35.  36.  39.  37.  38.
  40.  45.  41.  42.  43.  44.  46.  47.  48.  50.  51.  49.  52.  55.
  53.  54.  58.  56.  57.  61.  59.  60.  63.  67.  62.  64.  65.  66.
  68.  69.  70.  71.  72.  74.  75.  73.  76.  77.  79.  83.  78.  82.
  81.  80.  84.  87.  85.  86.  89.  88.  90.  93.  92.  91.  96.  99.
  94. 101.  98. 100.  95. 102.  97. 103. 106. 108. 107. 104. 109. 112.
 114. 110. 111. 113. 115. 117. 120. 116. 119. 121. 122. 123. 124. 129.
 126. 127. 125. 128. 130. 131. 133. 134. 135. 132. 137. 136. 149. 138.
 139. 140. 141. 142. 143. 144. 146. 147. 148. 151. 157. 156. 158. 161.
 165. 163. 162. 167. 168. 170. 171. 172. 176. 191. 211. 202. 215.] - Number of unique: 166
Column: PM25_ugm3 - Unique Values: [ 3.03    5.042 32.5    ...   8.699   7.878   8.28 ] - Numbe
r of unique: 8218
```

After visualizing the matrix above, we can verify that in fact the dataset presents variables according to the function info() used before, and it does not show any inconsistency at first sight.

From the matrix printed above, we can split the dataset in 4 different categories:

Identification of line segments: This information are useful if the intention is creating a map. The dataset was created to be used as a tool of analyses in ArcGis, therefore these data are important to create a physical visualisation of the data over a global map application, such as google maps.

1 - road_id: unique road segment ID - Numerical

2 - the_geom: road segment linestring, it provides coordinates to identify the linestring in a map - Categorical

3 - osm_id: OSM road ID - Numerical

4 - osm_code: OSM code - Numerical

5 - osm_name OSM road name - Categorical

Characteristics of line segments: This information will classify the line segments regarding its specific characteristics.

6 - osm_fclass OSM fclass / road-type - Categorical

7 - osm_ref OSM road ref - Categorical

8 - osm_oneway OSM description of traffic way on the via - B indicates the line segment can be drove only to the direction the car is moving, and F indicates the line segment can be drove in the direction the car is moving and in the opposite direction - Categorical

9 - osm_maxspeed OSM description of maximum speed permitted in the via - Numerical

10 - osm_layer OSM description of level of the via (it is usefull to represent overlapping among vias and relative level) - Categorical (it was converted in numbers but its representation is categorical).

11 - osm_bridge OSM description whether via is over a bridge or not - Categorical

12 - osm_tunnel OSM description whether via is through a tunnel or not - Categorical

Characteristics of the measurement: This information will state characteristics of the measurement itself, and it is usefull to analyse reliability of the data row by row.

13 - NO2points number of measurements on this road segment - Numerical

14 - NO2drives number of drive passes on this road segment - Numerical

15 - NOpoints number of measurements on this road segment - Numerical

16 - NOdrives number of drive passes on this road segment - Numerical

17 - CO2points number of measurements on this road segment - Numerical

18 - CO2drives number of drive passes on this road segment - Numerical

19 - COpoints number of measurements on this road segment - Numerical

20 - COdrives number of drive passes on this road segment - Numerical

21 - O3points number of measurements on this road segment - Numerical

22 - O3drives number of drive passes on this road segment - Numerical

23 - PM25points number of measurements on this road segment - Numerical

24 - PM25drives number of drive passes on this road segment - Numerical

Concentration of pollutants: This is the concentration of pollutants object of this study.

25 - NO2_ugm3 NO2 concentration (median of drive pass mean) in µg/m3 - Numerical

26 - NO_ugm3 NO concentration (median of drive pass mean) in µg/m3 - Numerical

27 - CO2_mgm3 CO2 concentration (median of drive pass mean) in mg/m3 - Numerical

28 - CO_mgm3 CO concentration (median of drive pass mean) in mg/m3 - Numerical

29 - O3_ugm3 O3 concentration (median of drive pass mean) in µg/m3 - Numerical

30 - PM25_ugm3 PM2.5 concentration (median of drive pass mean) in µg/m3 - Fine particule metter that are 2.5 microns or less in diameter - Numerical

3 - Exploratory analysis and data cleaning

In this section it is sought to analyse if any information is useless for the scope of this project, or for some reason unrealiable, and after this process this information will be dropped or replaced properly.

All the data classified as Identification of Line Segments is considered useless to the analysis that will be presented in this report, because they serve to a purpose of map visualization, which is out of the scope of this project. Furthermore, the column 'osm_ref', is only a numerical classification for the column 'osm_fclass', what makes it an unnecessary doubled information. Therefore, all of this data will be dropped.

In [21]:

```
airquality_df.drop(['road_id', 'the_geom', 'osm_code', 'osm_id', 'osm_name','osm_ref'], ax
```

As discussed before osm_layer represents a categorical value converted to number, therefore it will be converted to string to easen preliminary analysis.

In [22]:

```
airquality_df['osm_layer'] = airquality_df['osm_layer'].astype(str)
```

Elimitating innacurate data according to methodology proposed

The link https://insights.sustainability.google/labs/airquality describes the methodology applied to create the dataset. The methodology applied suggested that data measured less than 10 times should be analysed very carefully, once its confiability is lower. For this reason, columns described by 'Feature'points will be used to drop line segments measured less than or equal to 10 times.

In [23]:

```
airquality_df2 = airquality_df[(airquality_df.NO2points >= 10)
                              & (airquality_df.NOpoints >= 10)
                              & (airquality_df.CO2points >= 10)
                              & (airquality_df.COpoints >= 10)
                              & (airquality_df.O3points >= 10)
                              & (airquality_df.PM25points >= 10)]
```

Once innacurate data was eliminated, we need to check the variables to indentify if they contain any null value.

In [24]:

```
airquality_df2.isnull().sum()
```

Out[24]:

```
osm_fclass       0
osm_oneway       0
osm_maxspeed     0
osm_layer        0
osm_bridge       0
osm_tunnel       0
NO2points        0
NO2drives        0
NO2_ugm3         0
NOpoints         0
NOdrives         0
NO_ugm3          0
CO2points        0
CO2drives        0
CO2_mgm3         0
COpoints         0
COdrives         0
CO_mgm3          0
O3points         0
O3drives         0
O3_ugm3          0
PM25points       0
PM25drives       0
PM25_ugm3        0
dtype: int64
```

In [25]:

```
airquality_df2.head()
```

Out[25]:

	osm_fclass	osm_oneway	osm_maxspeed	osm_layer	osm_bridge	osm_tunnel	NO2points	NO2drives
46	service	B	20	0	False	False	5008	165.0
76	residential	B	30	0	False	False	267	1.0
137	residential	B	30	0	False	False	31	1.0
144	service	F	30	0	False	False	43	1.0
145	service	B	0	0	False	False	24	1.0

5 rows × 24 columns

As explained before, columns classificed as Characteristics of measurements are useful to analyse the confiability of that measurement itself. The methodology do not suggest further steps regarding confiability of data using Characteristics of measurements, and it considers the data collected generally robust. Thus, this data will no longer be used to any further analysis and will be dropped.

In [26]:
```
#After innacurate data being eliminated, columns 'Feature'points and 'Feature'drives will
#therefore, they will be dropped to easen visualization of the dataset.

airquality_df3 = airquality_df2.drop(['NO2points', 'NO2drives',
                    'NOpoints', 'NOdrives',
                    'CO2points', 'CO2drives',
                    'COpoints', 'COdrives',
                    'O3points','O3drives',
                    'PM25points','PM25drives'], axis = 1)
```

In [27]:
```
airquality_df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 17788 entries, 46 to 24693
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   osm_fclass     17788 non-null  object
 1   osm_oneway     17788 non-null  object
 2   osm_maxspeed   17788 non-null  int64
 3   osm_layer      17788 non-null  object
 4   osm_bridge     17788 non-null  bool
 5   osm_tunnel     17788 non-null  bool
 6   NO2_ugm3       17788 non-null  float64
 7   NO_ugm3        17788 non-null  float64
 8   CO2_mgm3       17788 non-null  float64
 9   CO_mgm3        17788 non-null  float64
 10  O3_ugm3        17788 non-null  float64
 11  PM25_ugm3      17788 non-null  float64
dtypes: bool(2), float64(6), int64(1), object(3)
memory usage: 1.5+ MB
```

Finally, the dataset object if defined as airquality_df3. It is composed by 4 categorical variables, and 8 numerical variables as it is possible to verify in the step above. Variables indexed with 0 to 5, which are all characteristics of line segments, will be considered independent variables, while variables indexed from 6 to 11, which are all Concentration of pollutants, will be considerend dependent variables.

3 - Descriptive analysis and data presentation

This section aimns to give a brief introduction of the dataset, start drawing summary conclusions, understand somewhat the shape of the dataset and its general behavior.

In [14]:
```
airquality_df3.head()
```

Out[14]:

	osm_fclass	osm_oneway	osm_maxspeed	osm_layer	osm_bridge	osm_tunnel	NO2_ugm3	NO_ugm3
46	service	B	20	0	False	False	7.269	-7.661
76	residential	B	30	0	False	False	-32.181	-5.565
137	residential	B	30	0	False	False	-7.844	-19.309
144	service	F	30	0	False	False	14.474	-9.241

	osm_fclass	osm_oneway	osm_maxspeed	osm_layer	osm_bridge	osm_tunnel	NO2_ugm3	NO_ugm3
145	service	B	0	0	False	False	-19.801	1.681

After the step of cleaning and selection of the dataset according to the scope of the repport as well as the methodology of the study which has generated this dataset, it was reduced from 24694 rows and 30 columns to 17788 rows and 12 columns. Rows were reduced by 27.97%. However it is important to make clear that all data eliminated was considered with low reliability by the methodology studied, hence nevertheless less data will be used, a higher accuracy is expected to conclusions reached by the end of the study.

In [15]:

```
numerical = ['osm_maxspeed','NO2_ugm3','NO_ugm3','CO2_mgm3','CO_mgm3','O3_ugm3','PM25_ugm
categorical = ['osm_fclass','osm_oneway','osm_layer', 'osm_bridge', 'osm_tunnel']
```

In [16]:

```
airquality_df3[numerical].describe()
```

Out[16]:

	osm_maxspeed	NO2_ugm3	NO_ugm3	CO2_mgm3	CO_mgm3	O3_ugm3	PM25_ugm
count	17788.000000	17788.000000	17788.000000	17788.000000	17788.000000	17788.000000	17788.00000
mean	41.321003	12.444767	7.477287	811.376198	0.374988	47.717049	6.73258
std	12.980152	19.819688	61.883027	35.440102	0.051501	8.990745	3.70082
min	0.000000	-32.651000	-105.857000	740.153000	0.248000	-3.717000	1.47400
25%	30.000000	0.394250	-7.676500	793.548750	0.339000	42.569000	5.26675
50%	50.000000	8.983000	-2.456000	805.827500	0.365000	48.230000	6.24500
75%	50.000000	20.820750	9.202750	822.874250	0.400000	53.502000	7.42525
max	80.000000	503.936000	1785.447000	2103.918000	1.303000	104.759000	111.69100

In [17]:

```
((airquality_df3[numerical].std(ddof=1)/airquality_df3[numerical].mean())*100).reset_inde
```

Out[17]:

	index	CoefficientOfVariation
0	osm_maxspeed	31.412965
1	NO2_ugm3	159.261218
2	NO_ugm3	827.613377
3	CO2_mgm3	4.367900
4	CO_mgm3	13.733901
5	O3_ugm3	18.841787
6	PM25_ugm3	54.968849

Some points can be highlighted as important information:

1 - Among the gases CO2 has the highest concentration, specially when it is verified that its unity is miligram per cubic meter, while most of the others are microgram per cubic meter. However, it is the gas with the smaller coefficient of variation. This suggests that the variables studied on this report do not affect its concentration strongly.

2 - The concetration of NO is the one with a widest range, and also the more disperse, this might indicates strong influence of other variables on the dataset over this variable.

In [18]:
```
airquality_df3[categorical].describe()
```

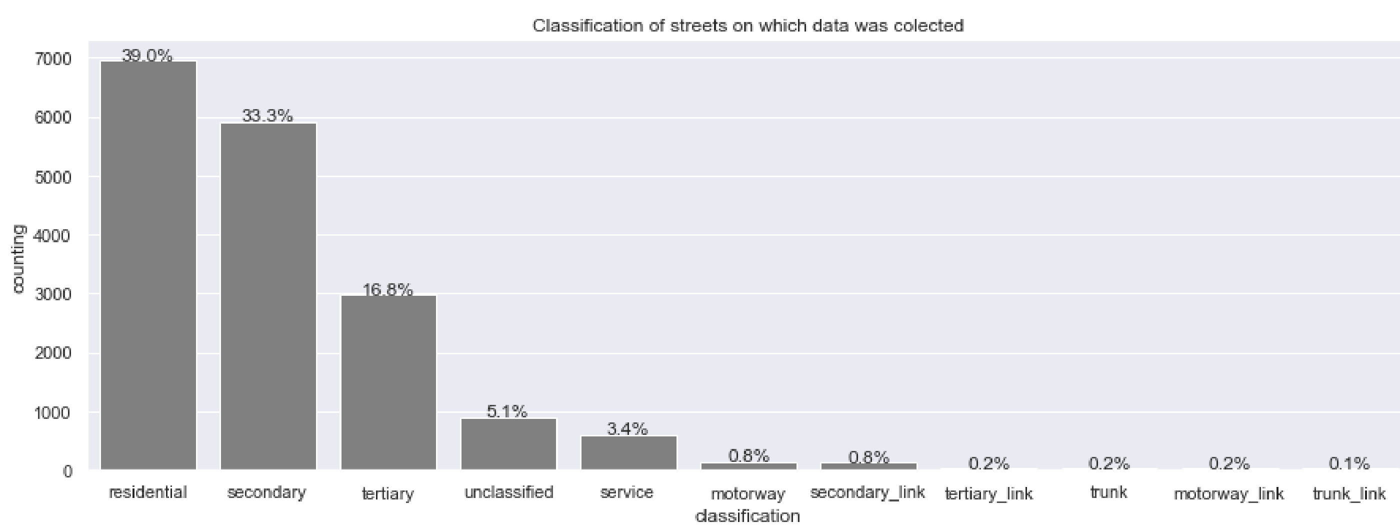
Out[18]:

	osm_fclass	osm_oneway	osm_layer	osm_bridge	osm_tunnel
count	17788	17788	17788	17788	17788
unique	11	2	5	2	2
top	residential	B	0	False	False
freq	6945	13704	17483	17613	17658

We can see in this brief description that most of data is residential, in streets with only one way driving, in the level of main streets, out of bridges and tunnels, and specially to layers, bridges and tunnel the data is extremelly unbalanced. In fact, less than 2% of the of the data was collected out of the top value for the 3 mentioned features. This is a normal situation though, once that most of streets in a city are usually only of one level.

In [31]:
```
fclass = airquality_df3.osm_fclass.value_counts().rename_axis('classification').reset_inde
fclass['percent']=(fclass['counting']/fclass['counting'].sum())*100
```

In [32]:
```
fig1 = sns.barplot(data = fclass, x = 'classification', y = 'counting', color = 'grey')
sns.set(rc = {'figure.figsize':(15,5)})
patches = fig1.patches
for i in range(len(patches)):
   x = patches[i].get_x() + patches[i].get_width()/2
   y = patches[i].get_height()+.05
   fig1.annotate('{:.1f}%'.format(fclass['percent'][i]), (x, y), ha='center')
plt.title('Classification of streets on which data was colected')
plt.show()
```

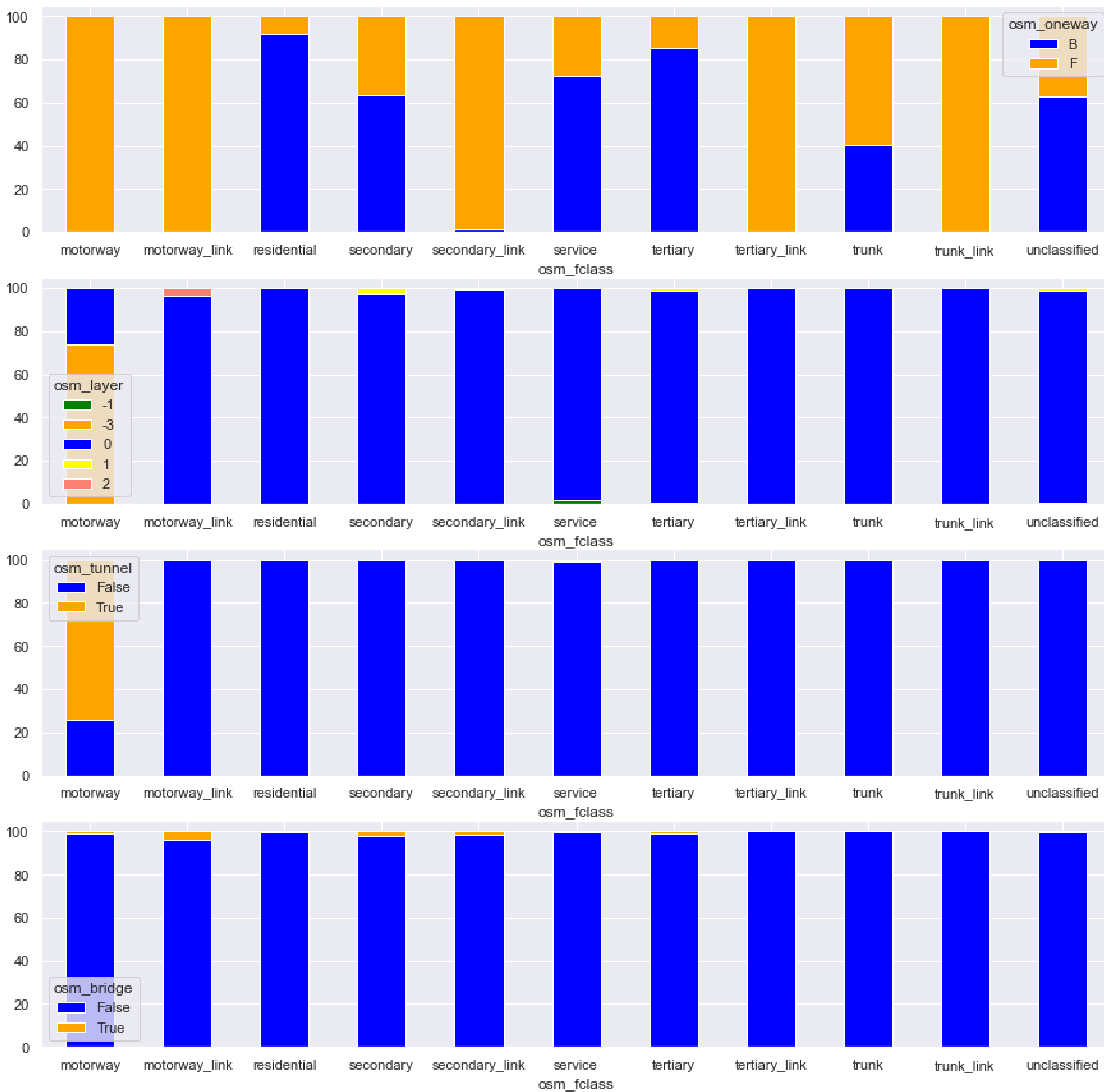


In [71]:
```
sns.set(rc = {'figure.figsize':(15,15)})
fig, fig1 = plt.subplots(4,1)
oneway_by_classification = pd.crosstab(airquality_df3['osm_fclass'], airquality_df3['osm_c
oneway_by_classification.plot(kind='bar',  stacked=True,
                                 rot=0, color=['blue','orange'], ax = fig1[0])
layer_by_classification = pd.crosstab(airquality_df3['osm_fclass'], airquality_df3['osm_la
layer_by_classification.plot(kind='bar',  stacked=True,
                                 rot=0, color=['green','orange','blue', 'yellow','salmon']
```

```python
tunnel_by_classification = pd.crosstab(airquality_df3['osm_fclass'], airquality_df3['osm_t
tunnel_by_classification.plot(kind='bar',  stacked=True,
                              rot=0, color=['blue','orange'], ax = fig1[2])
bridge_by_classification = pd.crosstab(airquality_df3['osm_fclass'], airquality_df3['osm_b
bridge_by_classification.plot(kind='bar',  stacked=True,
                              rot=0, color=['blue','orange'], ax = fig1[3])
```

Out[71]:

```
<AxesSubplot:xlabel='osm_fclass'>
```



The graph above is not useful to identify absolute values, however it is useful to identify proportions between variables on the dataset.

1 - Vias marked with F are mainly found in residential and in tertiary streets, and they are not found at all in motorway or in motorway links.

2 - Most of motorways are in level -3, in addition it is the only classification of street where -3 layers are found. Most of the remaining classification are in level 0.

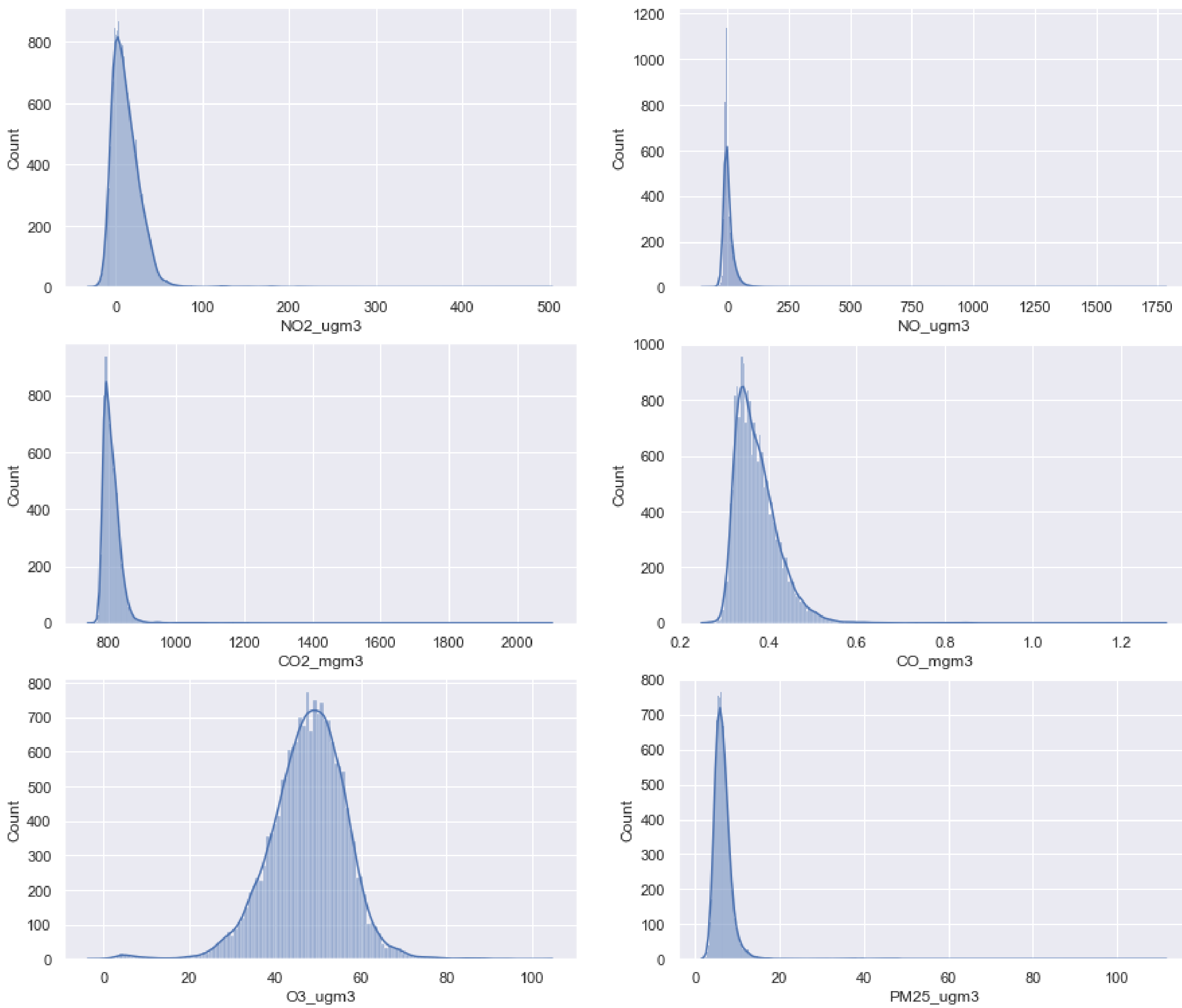
3 - Most of motorways consist of tunnels, in addition it is the only classification of streets where tunnels are found.

4 - Bridges are seldom found, but its majority is found in motorway_link.

Analysing behavior and shape of the indicators of air quality

In [74]:

```
sns.set(rc = {'figure.figsize':(15,13)})
fig, fig2 = plt.subplots(3,2)
sns.histplot(airquality_df3.NO2_ugm3,kde = True, ax = fig2[0,0])
sns.histplot(airquality_df3.NO_ugm3, kde = True, ax = fig2[0,1])
sns.histplot(airquality_df3.CO2_mgm3, kde = True, ax = fig2[1,0])
sns.histplot(airquality_df3.CO_mgm3, kde = True, ax = fig2[1,1])
sns.histplot(airquality_df3.O3_ugm3, kde = True, ax = fig2[2,0])
sns.histplot(airquality_df3.PM25_ugm3, kde = True, ax = fig2[2,1])

plt.show()
```



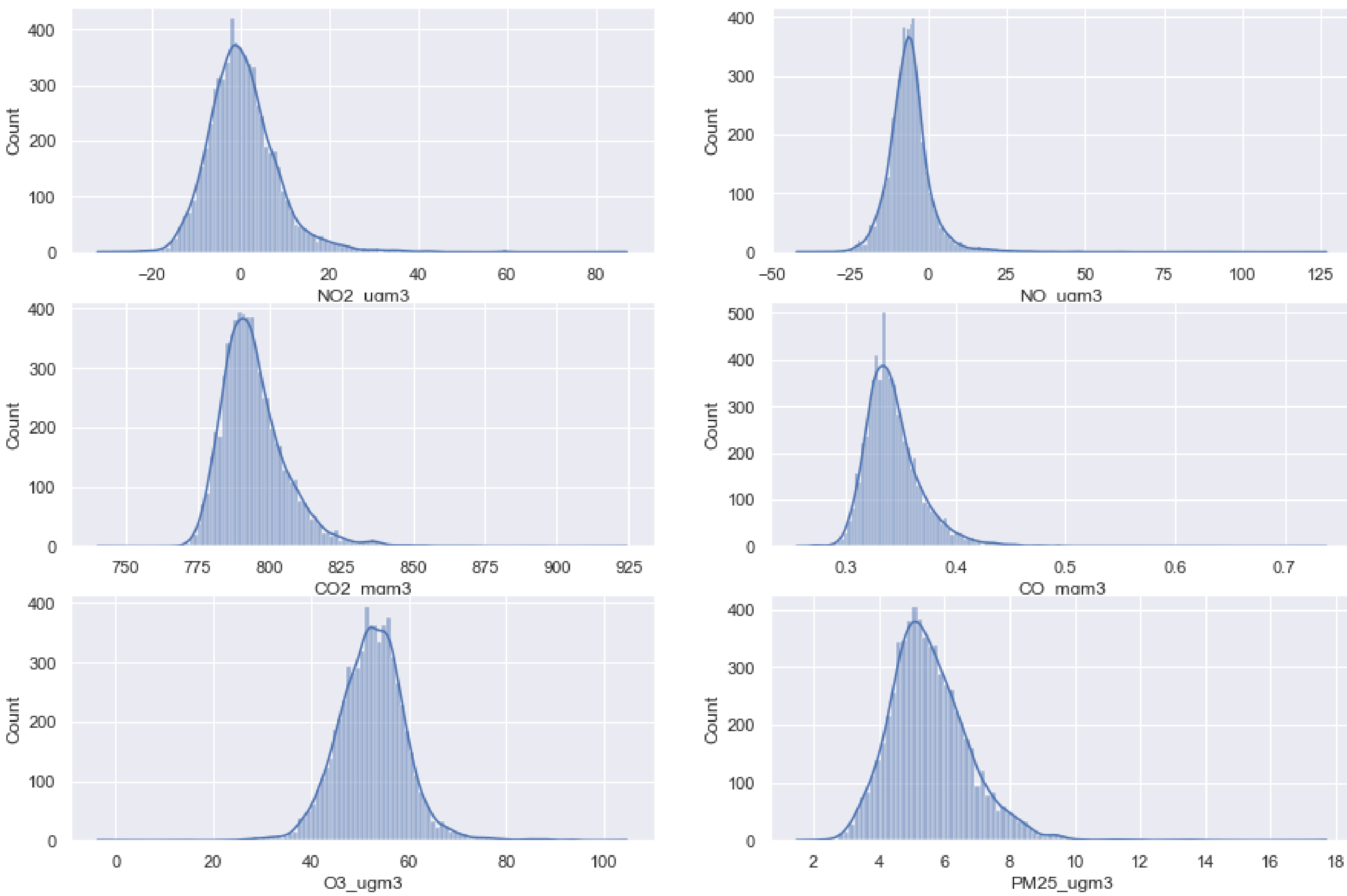
Graphs above show us the shape of concentration of pollutants analysed, it is possible verify that with exception of O3, all the rest are posively skewed, whereas O3 is nearly symmetric.

In [22]:

```
#Filtering all the information in the Dataset airquality_df3 which represent the most comm
#dataset according to its description
airquality_df4 = airquality_df3[(airquality_df3['osm_layer']== '0')
```

```
                        & (airquality_df3['osm_bridge'] == 0)
                        & (airquality_df3['osm_tunnel'] == 0)
                        & (airquality_df3['osm_oneway'] == 'B')
                        & (airquality_df3['osm_fclass'] == 'residential')]
```

In [23]:
```
sns.set(rc = {'figure.figsize':(15,10)})
fig, fig2 = plt.subplots(3,2)
sns.histplot(airquality_df4.NO2_ugm3,kde = True, ax = fig2[0,0])
sns.histplot(airquality_df4.NO_ugm3, kde = True, ax = fig2[0,1])
sns.histplot(airquality_df4.CO2_mgm3, kde = True, ax = fig2[1,0])
sns.histplot(airquality_df4.CO_mgm3, kde = True, ax = fig2[1,1])
sns.histplot(airquality_df4.O3_ugm3, kde = True, ax = fig2[2,0])
sns.histplot(airquality_df4.PM25_ugm3, kde = True, ax = fig2[2,1])

plt.show()
```



If all the features considered independent are settled to its top value, it is possible to see that the dataset, although is still posively skewed for the same variables, behaves in a much more more way, what indicates that independent variable influence on the concentration of pollutants. Although the influence might be big, it is not possible to observe in this graphs because the dataset does not provide data enough out of the top values. It would be a good aproach plotting overlapping histograms to analyse how much difference each variable make with the pollutants, however there is not data enough to this, once the data is very unbalanced. The correlation between variables can be seen in the following correlation matrix, plotted as a heatmap.

In [76]:
```
airquality_df5 = airquality_df3
```

In [77]:
```
#To plot a correlation matrix all the categorical variables on the dataset need to be tra
#In this code categorical values are being replace.
```

```
airquality_df5['osm_fclass'].replace({'service': 0, 'residential': 1, 'unclassified': 2,
                                      'secondary': 3, 'secondary_link': 4, 'tertiary': 5,
                                      'tertiary_link': 6, 'motorway_link': 7, 'motorway':
                                      'trunk_link': 9, 'trunk': 10}, inplace = True)
airquality_df5['osm_oneway'].replace({'B': 0, 'F': 1}, inplace = True)
airquality_df5['osm_bridge'].replace({True: 1, False: 0}, inplace = True)
airquality_df5['osm_tunnel'].replace({True: 1, False: 0}, inplace = True)
airquality_df5['osm_layer'] = airquality_df5['osm_layer'].astype(int)
```

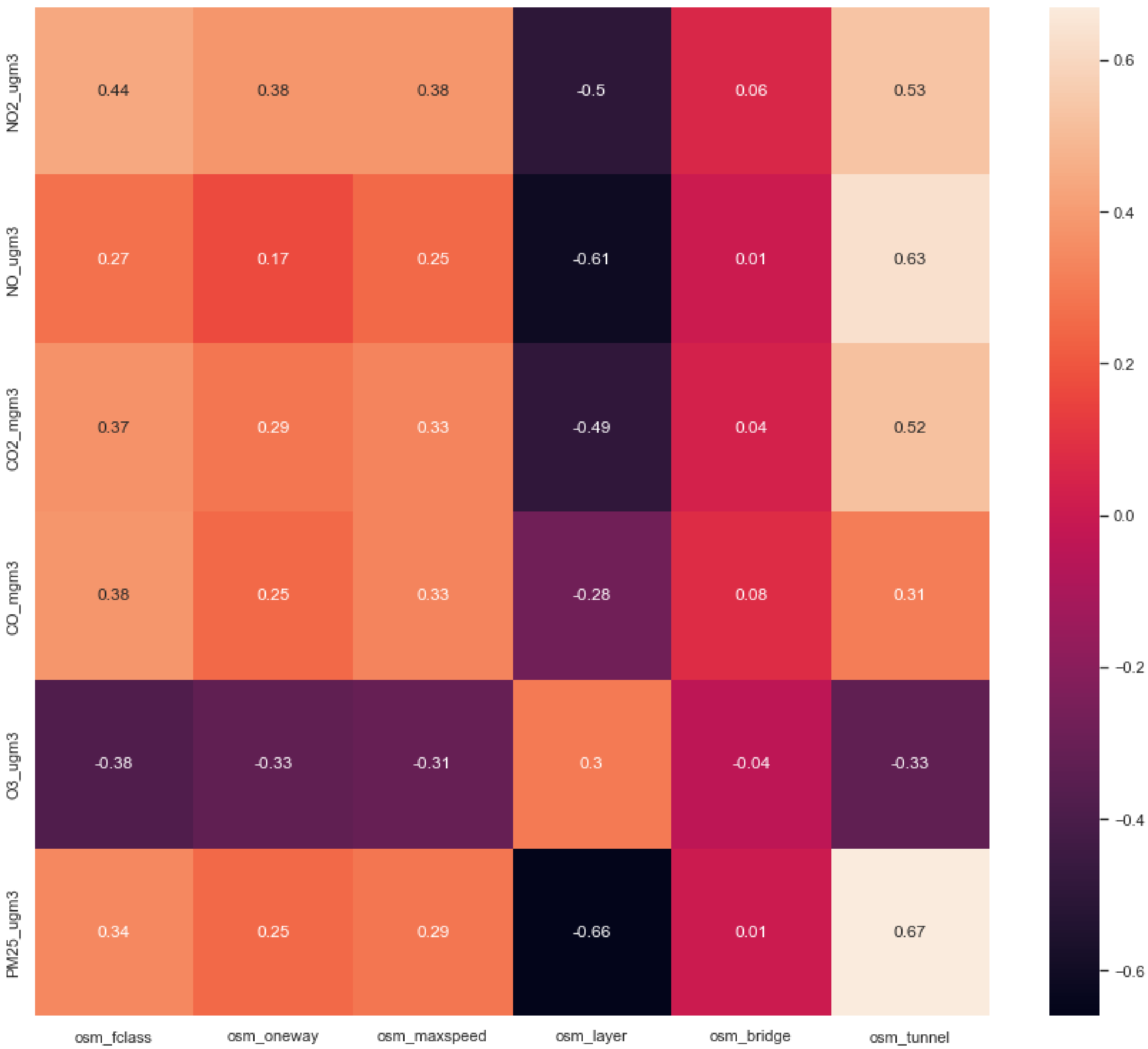
In [78]:

```
#Recovering the original dataset for airquality_df3, that will be used above to better unc
airquality_df3 = airquality_df2.drop(['NO2points', 'NO2drives',
                     'NOpoints', 'NOdrives',
                     'CO2points', 'CO2drives',
                     'COpoints', 'COdrives',
                     'O3points','O3drives',
                     'PM25points','PM25drives'], axis = 1)
```

In [79]:

```
correlation = airquality_df5.corr().round(2).iloc[6: , :6]
sns.heatmap(correlation, annot=True)
```

Out[79]: <AxesSubplot:>



The first conclusion that can be drawn from the heatmap and correlation matrix is that O3_ugm3 is influenced in the opposite way of all the other polutants for every feature. osm_fclass, osm_oneway, osm_maxsppeed, osm_bridge and osm_tunnel, are positively correlated to all polutants except O3_ugm3, whereas osm_layer is correlated negatively to all polutants except O3_ugm3. We can also understand the osm_bridge are very weakly correlated to all variables, and therefore will not be further analysed in this report. The two variables more influent to the conentratiom are osm_tunnel, which indicates whether measure site is through a tunnel or not, and osm_layer, which indicates overlapping among vias.

Analysing correlations with classification of streets and max speed

In [29]:

```
airquality_df3.head()
```

Out[29]:

	osm_fclass	osm_oneway	osm_maxspeed	osm_layer	osm_bridge	osm_tunnel	NO2_ugm3	NO_ugm3
46	service	B	20	0	False	False	7.269	-7.661
76	residential	B	30	0	False	False	-32.181	-5.565
137	residential	B	30	0	False	False	-7.844	-19.309
144	service	F	30	0	False	False	14.474	-9.241
145	service	B	0	0	False	False	-19.801	1.681

To analyse correlation between the concentration of polutants, classification of streets and max speed of vias, the dataset will be grouped by osm_fclass, and all features grouped by its mean for each classification of street. Than the new dataset generate will be sorted according to max speed in descending order, and concentration will be plotted against osm_fclass organized by mean of max speed.

In [30]:

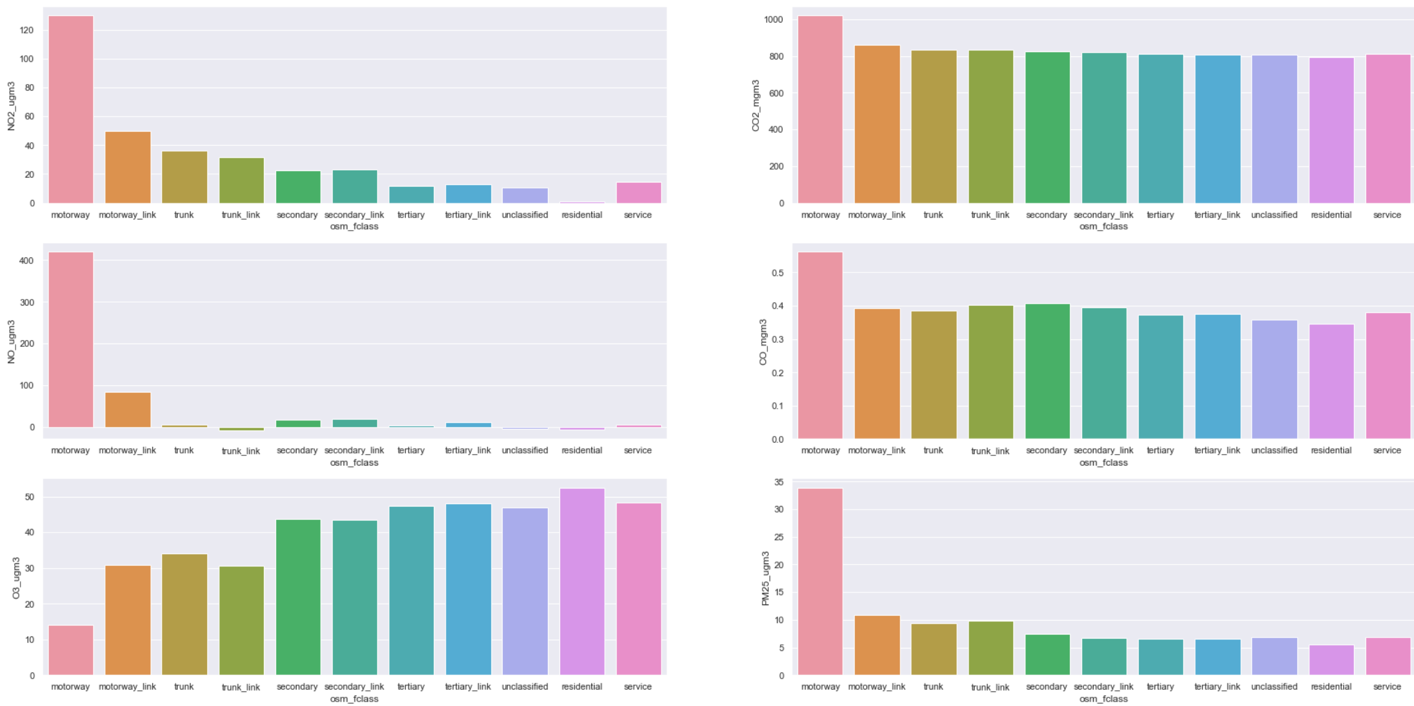
```
#Grouping street classifications to find patterns within it
#Data will be sorted in order of osm_maxspeed, which is a unique caracteristic of every c

street_class = airquality_df3.groupby('osm_fclass', axis=0).mean()
street_class.sort_values(by = 'osm_maxspeed', ascending = False, inplace= True)
street_class = street_class.reset_index()
street_class
```

Out[30]:

	osm_fclass	osm_maxspeed	osm_bridge	osm_tunnel	NO2_ugm3	NO_ugm3	CO2_mgm3	CO_mgm
0	motorway	77.200000	0.006667	0.740000	129.843000	421.467527	1019.809427	0.56226
1	motorway_link	69.259259	0.037037	0.000000	50.032000	83.540741	861.308000	0.39240
2	trunk	57.297297	0.000000	0.000000	36.380919	5.688216	834.479730	0.38532
3	trunk_link	51.250000	0.000000	0.000000	31.872292	-7.898375	834.777125	0.40325
4	secondary	49.956081	0.022973	0.000338	22.666553	16.646617	825.879387	0.40765
5	secondary_link	48.027211	0.013605	0.000000	23.291299	18.252054	820.364497	0.39449
6	tertiary	42.589495	0.009033	0.002007	11.827326	2.595429	809.552783	0.37367
7	tertiary_link	42.564103	0.000000	0.000000	12.713051	10.449077	807.646077	0.37515
8	unclassified	39.712389	0.004425	0.001106	10.705945	-4.391814	807.953445	0.35892
9	residential	34.964723	0.000432	0.000720	0.948358	-5.929260	795.178395	0.34469
10	service	12.752475	0.001650	0.008251	14.368246	5.382812	811.332985	0.37985

In [31]:

```
sns.set(rc = {'figure.figsize':(30,15)})
fig, fig3 = plt.subplots(3,2)
sns.barplot(data = street_class, x = 'osm_fclass', y = 'NO2_ugm3', ax = fig3[0,0])
sns.barplot(data = street_class, x = 'osm_fclass', y = 'NO_ugm3', ax = fig3[1,0])
sns.barplot(data = street_class, x = 'osm_fclass', y = 'CO2_mgm3', ax = fig3[0,1])
sns.barplot(data = street_class, x = 'osm_fclass', y = 'CO_mgm3', ax = fig3[1,1])
sns.barplot(data = street_class, x = 'osm_fclass', y = 'O3_ugm3', ax = fig3[2,0])
sns.barplot(data = street_class, x = 'osm_fclass', y = 'PM25_ugm3', ax = fig3[2,1])
plt.show()
```



It seems to exist correlation between classification of street and concentration of polutants. Once the dataframe used to produce graphs were sorted by mean maximun speed of each classification, it may exist a correlation between speed of trafic and concentration of polutants.

Analysing correlation with direction of via

In [32]:

```
#Calculating the mean of different features according to osm_oneway.

airquality_df3.groupby('osm_oneway').mean()
```

Out[32]:

osm_oneway	osm_maxspeed	osm_bridge	osm_tunnel	NO2_ugm3	NO_ugm3	CO2_mgm3	CO_mgm3	O3
B	40.07049	0.009267	0.000876	8.355229	1.762674	805.802159	0.367958	49.
F	45.51714	0.011753	0.028893	26.167351	26.652864	830.080071	0.398577	42.

It is possible to verify that the mean for the concentrations are consistently high to every pollutants, except O3, which is always inversely proportional, therefore, vias of two aways tend to have more polution.

Analysing correlation with the presence of a tunnel

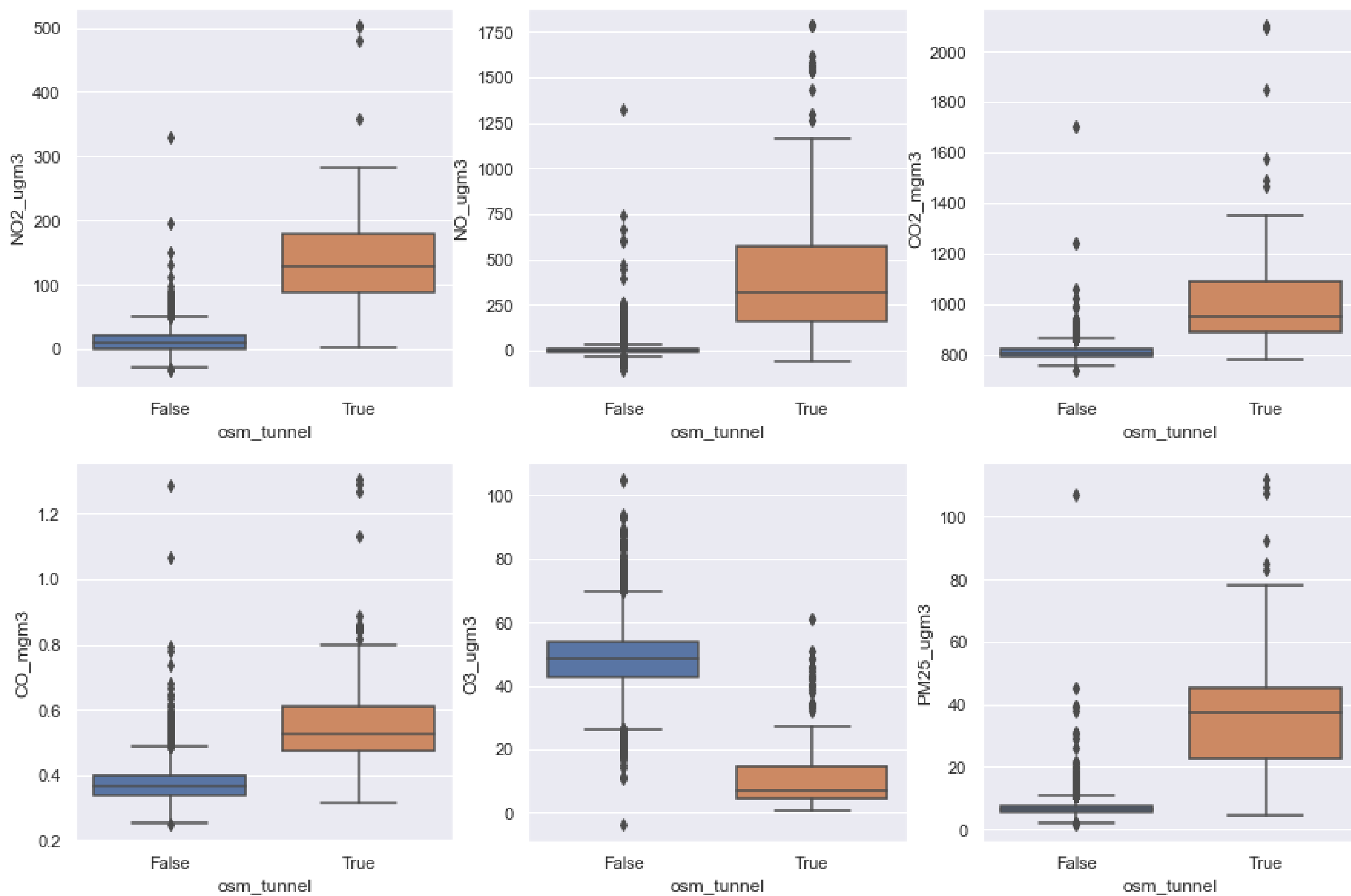
In [81]:

```
sns.set(rc = {'figure.figsize':(15,10)})
sns.set()

fig, axes=plt.subplots(2,3)
sns.boxplot(x = 'osm_tunnel', y = 'NO2_ugm3', data = airquality_df3, ax=axes[0,0])
```

```
sns.boxplot(x = 'osm_tunnel', y = 'NO_ugm3', data = airquality_df3, ax=axes[0,1])
sns.boxplot(x = 'osm_tunnel', y = 'CO2_mgm3', data = airquality_df3, ax=axes[0,2])
sns.boxplot(x = 'osm_tunnel', y = 'CO_mgm3', data = airquality_df3, ax=axes[1,0])
sns.boxplot(x = 'osm_tunnel', y = 'O3_ugm3', data = airquality_df3, ax=axes[1,1])
sns.boxplot(x = 'osm_tunnel', y = 'PM25_ugm3', data = airquality_df3, ax=axes[1,2])
```

Out[81]:

```
<AxesSubplot:xlabel='osm_tunnel', ylabel='PM25_ugm3'>
```



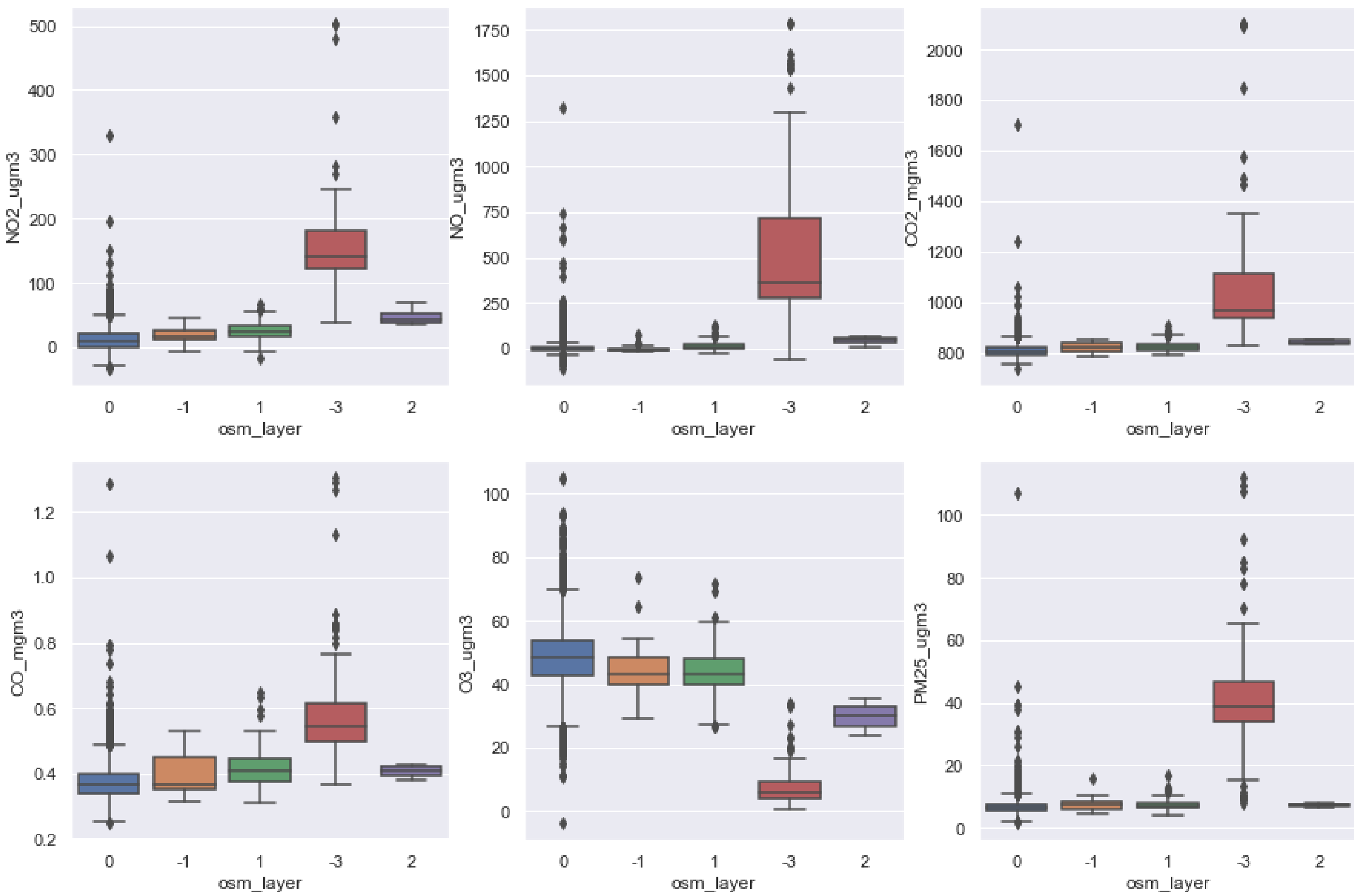
It is possible to verify that all parameters are strongly influenced by the presence of a tunnel. Concetrations of NO2, NO, CO2, CO and PM25 are significantly higher under tunnels and concetration of O3 is significantly lower. It is also possible to verify that generally under tunnels the concentration has a winder range.

In [83]:

```
sns.set(rc = {'figure.figsize':(15,10)})
sns.set()

fig, axes=plt.subplots(2,3)
sns.boxplot(x = 'osm_layer', y = 'NO2_ugm3', data = airquality_df3, ax=axes[0,0])
sns.boxplot(x = 'osm_layer', y = 'NO_ugm3', data = airquality_df3, ax=axes[0,1])
sns.boxplot(x = 'osm_layer', y = 'CO2_mgm3', data = airquality_df3, ax=axes[0,2])
sns.boxplot(x = 'osm_layer', y = 'CO_mgm3', data = airquality_df3, ax=axes[1,0])
sns.boxplot(x = 'osm_layer', y = 'O3_ugm3', data = airquality_df3, ax=axes[1,1])
sns.boxplot(x = 'osm_layer', y = 'PM25_ugm3', data = airquality_df3, ax=axes[1,2])
```

Out[83]:

```
<AxesSubplot:xlabel='osm_layer', ylabel='PM25_ugm3'>
```



Such as happened to the variable of tunnels, presence of overlap of vias also influence the concentration of pollutants.

4 - Conclusion

The dataset was analysed according to its summary features, to understand shape and basic correlations among data. First, data useful only for identification of the line segments in a map, classified as Identification of Line Segments had been discarded, once this was not object of this study.

The following step was cleaning the data according to the methodology purposed by the supplier of the data, in which every line segment measure 10 times or less, had been dropped for its expected low reliability. Afterwards all columns used to clean the data were also dropped as they would not be useful in the scope of this report.

The descriptive analysis were presented to all numerical data, and preliminary statements were made about the data. Together with the descriptive analysis, the coefficient of variance was calculated for the numerical variables. It was useful to compare the variability of the data which has different unities, therefore using an § parameter was important.

The categorical variables were also described and it was noticed that the dataset is extremely unbalanced in its observations. The large majority of the data is concentrated in the more common value for each categorical variable.

Histoplots were ploted to analyse concentration of pollutants, and it was verified that 5 out 6 variables were positively skewed, representing the assymetry of the data. However the concentration of O3 showed a behavior very near to normal. After filtering the data only to the most common values for dependent variables, the histoplots changed significantly, becoming more centered, but still positively skewed.

Bar plots were plotted to compare the concentration of polutants in differents classifications of streets. The classification of streets were sorted by its mean max speed in each classification. Motorways, which has the higher mean for maximum speed, had also the highest concentration of NO2, NO, CO2, CO and PM25, and lowest concentrstion for O3.

Box plots were plotted and it became clear that being under a tunnel or in vias where overlapping is found, changes the mean of concentration of pollutants.

5 - References

1. data.gov.ie. (n.d.). Google Project Air View Data - Dublin City (May 2021 - August 2022) - AirView_DublinCity_RoadData_CSV - data.gov.ie. [online] Available at: https://data.gov.ie/dataset/google-airview-data-dublin-city/resource/f3b5c4bf-5646-4f0b-b4f6-8e8beebcff3b?inner_span=True [Accessed 02 Apr. 2023].

2. insights.sustainability.google. (n.d.). Google Environmental Insights Explorer - Make Informed Decisions. [online] Available at: https://insights.sustainability.google/labs/airquality [Accessed 04 Apr. 2023].

3. data.smartdublin.ie. (n.d.). Google Project Air View Data - Dublin City (May 2021 - August 2022) - AirView_DublinCity_RoadData_CSV - data.smartdublin.ie. [online] Available at: https://data.smartdublin.ie/dataset/google-airview-data-dublin-city/resource/f3b5c4bf-5646-4f0b-b4f6-8e8beebcff3b [Accessed 02Apr. 2023].

TASK 2 - PROBABILITY (DISCRETE)

2.1 What is the probabilty of rolling exactly two 6s in five rolls of a fair dice?

In [85]:

```
import numpy as np
from numpy import random
from scipy.stats import binom
from scipy.stats import poisson
from scipy.stats import norm
```

The distribution can be noted as:

X~Bin(n=5, p=1/6)

The problem consists in finding:

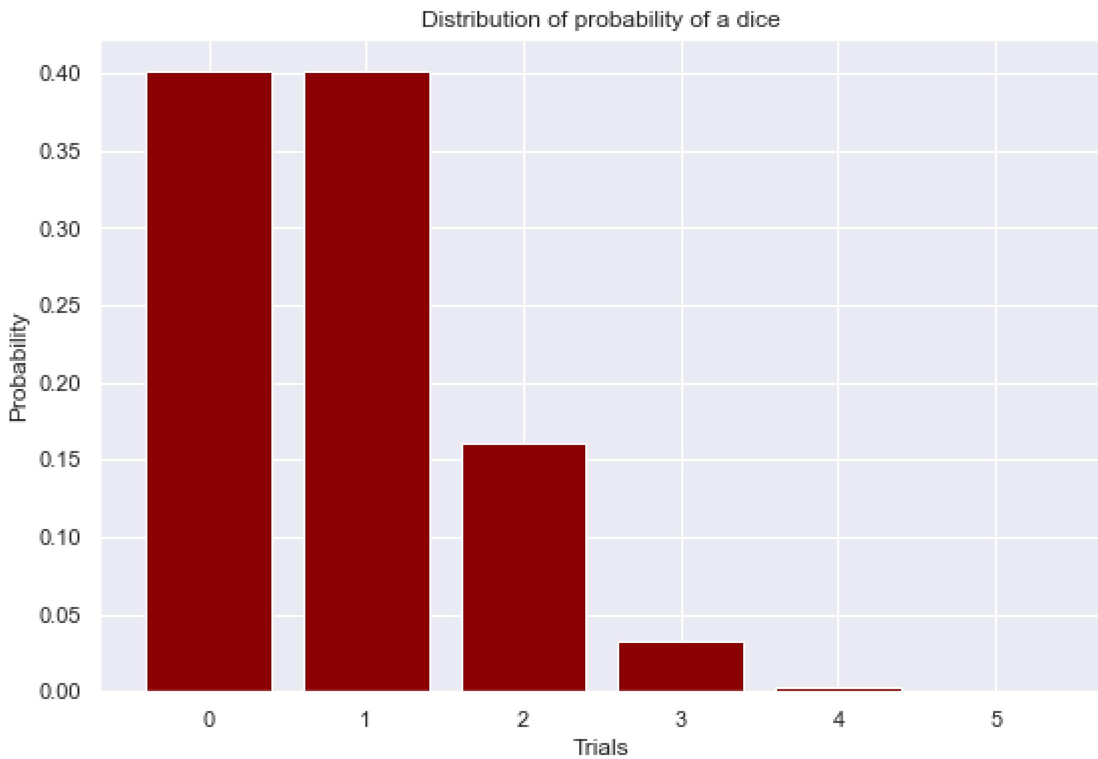
P(X=2)?

The distributtion will be printed below

In [86]:

```
x = np.linspace(0, 5, 6)
y = binom.pmf(k = x, n=5, p=1/6)
fig, ax = plt.subplots(figsize=(9,6))
ax.bar(x,y, color='darkred')
ax.set_xlabel('Trials')
ax.set_ylabel('Probability')
ax.set_title('Distribution of probability of a dice')
```

Out[86]: Text(0.5, 1.0, 'Distribution of probability of a dice')



In [106…

```
a = binom.pmf(k=2,n=5,p=1/6)
print('The probability of rolling exactly two 6s in five rolls of a fair dice is:', "{:.2%
```

The probability of rolling exactly two 6s in five rolls of a fair dice is: 16.08%

2.2 What is the probability of happening a week in which no more than 2 accidents occurs?

The distribution can be noted as:

X~ Pois(λ=0.75)

The problem consists in finding:

P(X<=2)?

In [105…

```
b = poisson.cdf(k=2, mu=0.75)
print('The probability of occuring less than 2 accidents in a week is: ', "{:.2%}".format
```

The probability of occuring less than 2 accidents in a week is: 95.95%

TASK 3 - PROBABILITY (CONTINUOUS)

In [89]:

```
import scipy.stats as ss
from numpy.random import seed
from numpy.random import normal
```

The distribution can be noted as:

Z~ N(μ=90, σ=10)

mu=90 sigma=10

The distribution is represented by the graph that follows.

In [90]:
```
mu=90
sigma=10
x = np.linspace(mu-4*sigma, mu+4*sigma, 100)
y = ss.norm.pdf(x, 90, 10)
```

In [91]:
```
fig, ax = plt.subplots(figsize=(9,6))
ax.plot(x,y, color='darkblue')
ax.set_xlabel('Time')
ax.set_ylabel('Probability')
ax.set_title('Distribution of probability of a customer visiting a Zoo')
```

Out[91]:
```
Text(0.5, 1.0, 'Distribution of probability of a customer visiting a Zoo')
```

3.1 What is the probability of a visitor selected at random spend at most 85 minutes?

If a visitor is selected at random, the probability of it spending at most 85 minutes is equivalent to:

P(Z<=85)?

The following graph brings a visual representation of the problem defined, in which calculating this probability is equivalent to calculate the area of the blue shadow under the bell curve.

The probability will be calculated in this case using the function in python norm.cdf(85, loc=90, scale=10)

In [92]:
```
fig, ax = plt.subplots(figsize=(9,6))
ax.plot(x,y, color='darkblue')
x_fill = np.linspace(mu - 4*sigma, 85, 100)
y_fill= ss.norm.pdf(x_fill, 90, 10)
ax.fill_between(x_fill, y_fill, alpha=0.3, color='blue')
```

```
ax.set_xlabel('Time')
ax.set_ylabel('Probability')
ax.set_title('Distribution of probability of a customer visiting a Zoo')
```

Out[92]: Text(0.5, 1.0, 'Distribution of probability of a customer visiting a Zoo')

In [103…

```
c = norm.cdf(85, loc=90, scale=10)
print("The probability of a random visitor spend at most 85 minutes is:", "{:.2%}".format
```

```
The probability of a random visitor spend at most 85 minutes is: 30.85%
```

3.2 What is the probability of a visitor selected at random spend at least 100 minutes?

If a visitor is selected at random, the probability of it spending at least 100 minutes is equivalent to a probability of 100% reduced the probability of spending at most 100 minutes, or in standard notation:

P(Z>=100) = 1 - P(Z<=100)

The following graph brings a visual representation of the problem defined, in which calculating this probability is equivalent to calculate the area of the blue shadow under the bell curve.

The probability will be calculated in this case using the function in python 1- norm.cdf(100, loc=90, scale=10)

In [94]:

```
fig, ax = plt.subplots(figsize=(9,6))
ax.plot(x,y, color='darkblue')
x_fill = np.linspace(100, mu+4*sigma, 100)
y_fill= ss.norm.pdf(x_fill, 90, 10)
ax.fill_between(x_fill, y_fill, alpha=0.3, color='blue')
ax.set_xlabel('Time')
ax.set_ylabel('Probability')
ax.set_title('Distribution of probability of a customer visiting a Zoo')
```

Text(0.5, 1.0, 'Distribution of probability of a customer visiting a Zoo')

Out[94]:

In [102…

```
d = 1-norm.cdf(100, loc=90, scale=10)
print("The probability of a random visitor spend at least 100 minutes is:", "{:.2%}".forma
```

```
The probability of a random visitor spend at least 100 minutes is: 15.87%
```

3.3 What is the probability of a visitor that is known to have spent over the average, to spend over 100 minutes?

If is known that a certain guest spent longer than the average in the zoo, it means that it has 100% of chances of having spent there over than 90 minutes. In this case, all the probability concentrates itself on the right side of the bell curve, in other words, instead of having 100% under the whole bell curve, now we have it under the right side only. Therefore, all the distribution it the right side is multiplied by 2. Hence the problem consists in calculating the area under the bell curve, after 100 minutes, what is made with 100% less the area before 100, multiplied by 2.

P(Z>=100) = 1 - 2*(P(Z<=100) - P(Z<=90))

In other notation, we can use the Baye's theorem, that stands for: P(A|B) = P(A ∩ B)/P(B) , The probability of A given B is equal to the probability of both A and B occuring divided by the probability of B.

In our example: P(Z>=100|Z>=90) = P(Z>=100 and Z>=90)/P(Z>=90) P(Z>=100 and Z>=90) = P(Z>=100) P(Z>=100|Z>=90) = P(Z>=100)/P(Z>=90) = (1-P(Z<=100))/(1-P(Z<=90))

The graph below brings a visual representation, all the red shadow area should be desconsidered, and all the concentration of probability will concentrate in the right side of the bell curve. Therefore we want to calculate the area of the blue shadow, knowing that the right side of the bell curve concentrate 100% of the probability.

The probability will be calculated in this case using in two methods:

Method 1: Function in python 1- 2*(norm.cdf(100, loc=90, scale=10)-norm.cdf(90, loc=90, scale=10)

Method 2: Function in python (1- (norm.cdf(100, loc=90, scale=10))/(1-norm.cdf(90, loc=90, scale=10))

In [96]:

```
fig, ax = plt.subplots(figsize=(9,6))
ax.plot(x,y, color='darkblue')
x_fill = np.linspace(100, mu+4*sigma, 100)
x_fill2 = np.linspace(mu-4*sigma,mu, 100)
y_fill= ss.norm.pdf(x_fill, 90, 10)
y_fill2 = ss.norm.pdf(x_fill2, 90, 10)
ax.fill_between(x_fill, y_fill, alpha=0.3, color='blue')
ax.fill_between(x_fill2, y_fill2, color='red')
ax.set_xlabel('Time')
ax.set_ylabel('Probability')
ax.set_title('Distribution of probability of a customer visiting a Zoo')
```

Out[96]:

```
Text(0.5, 1.0, 'Distribution of probability of a customer visiting a Zoo')
```

In [100…

```
#Method 1 - Using description of the problem
e = 1- 2*(norm.cdf(100, loc=90, scale=10) - norm.cdf(90, loc = 90, scale = 10))
print('The probability of certain guest spend longer than 100 minutes on the zoo, given it
```

```
The probability of certain guest spend longer than 100 minutes on the zoo, given it has sp
ent longer than average is: 31.73%
```

In [101…

```
#Method 2: Using Baye's Theorem.
f = (1-norm.cdf(100, loc=90, scale=10))/(1-norm.cdf(90, loc=90, scale=10))
print('The probability of certain guest spend longer than 100 minutes on the zoo, given it
```

```
The probability of certain guest spend longer than 100 minutes on the zoo, given it has sp
ent longer than average is: 31.73%
```

In [ ]: