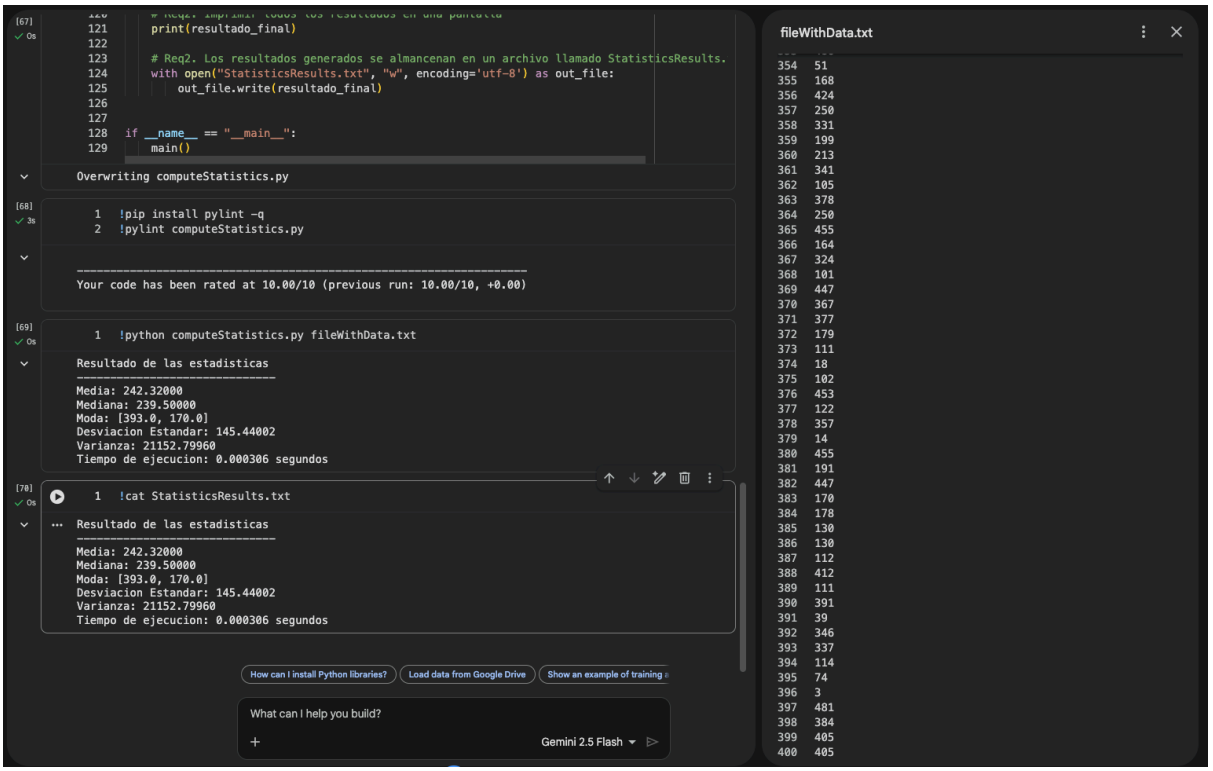
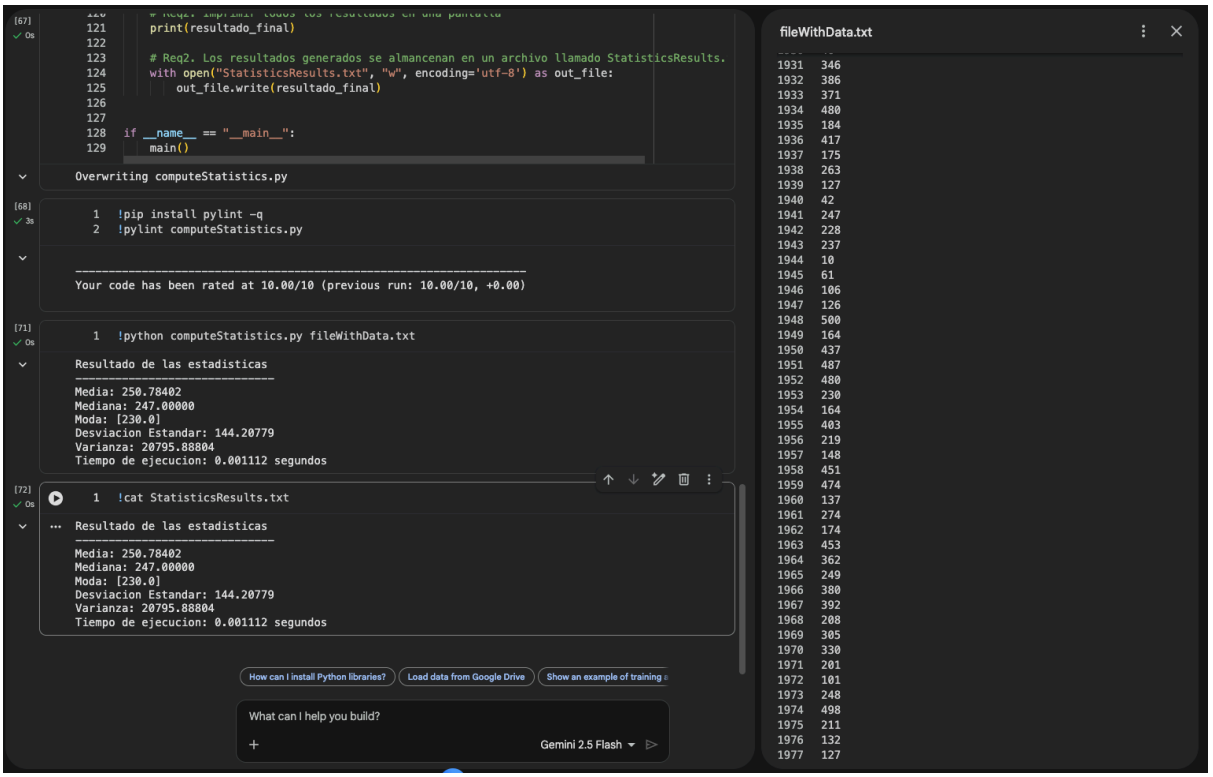
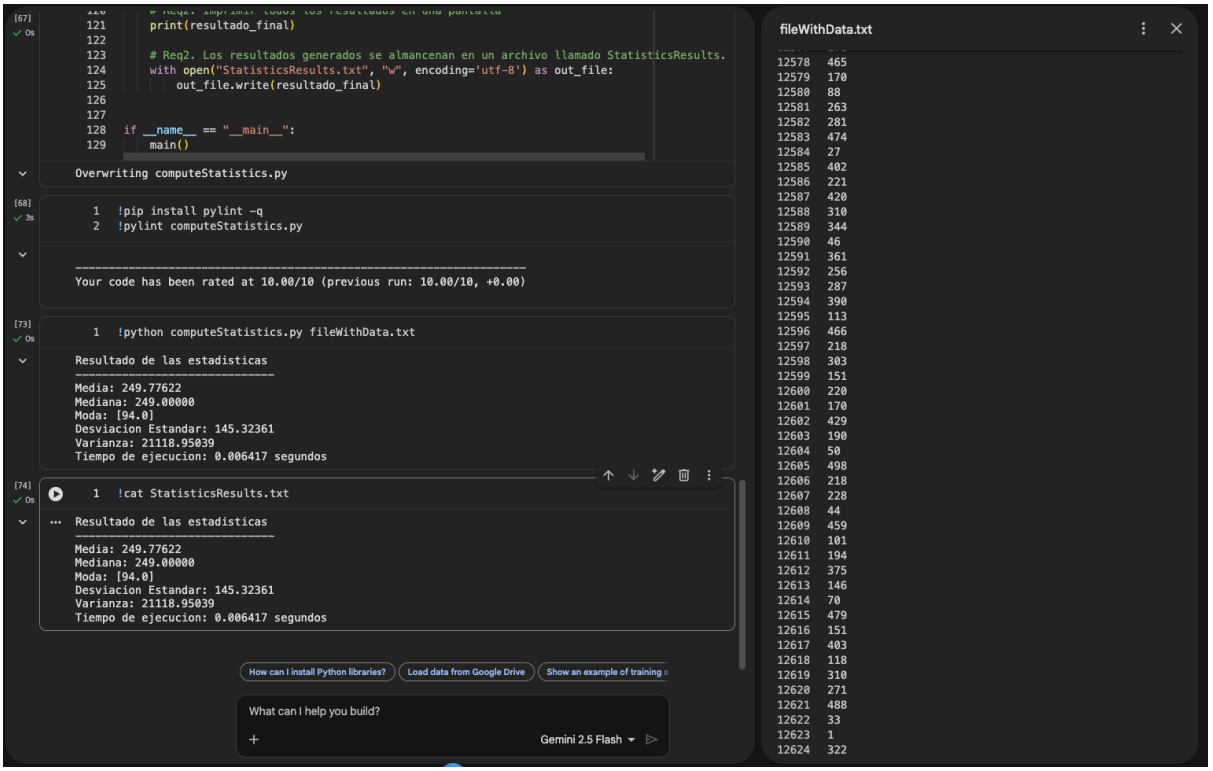
TC1


print(resultado_final)
# Req2. Los resultados generados se almancenan en un archivo llamado StatisticsResults.
with open("StatisticsResults.txt", "w", encoding='utf-8') as out_file:
out_file.write(resultado_final)
if __name__ == "__main__":
main()
Overwriting computeStatistics.py
!pip install pylint -q
!pylint computeStatistics.py
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
!python computeStatistics.py fileWithData.txt
Resultado de las estadisticas
Media: 242.32000
Mediana: 239.50000
Moda: [393.0, 170.0]
Desviacion Estandar: 145.44002
Varianza: 21152.79960
Tiempo de ejecucion: 0.000306 segundos
!cat StatisticsResults.txt
Resultado de las estadisticas
Media: 242.32000
Mediana: 239.50000
Moda: [393.0, 170.0]
Desviacion Estandar: 145.44002
Varianza: 21152.79960
Tiempo de ejecucion: 0.000306 segundos
How can I install Python libraries?
Load data from Google Drive
Show an example of training
What can I help you build?
Gemini 2.5 Flash
fileWithData.txt
354 51
355 168
356 424
357 250
358 331
359 199
360 213
361 341
362 105
363 378
364 250
365 455
366 164
367 324
368 101
369 447
370 367
371 377
372 179
373 111
374 18
375 102
376 453
377 122
378 357
379 14
380 455
381 191
382 447
383 170
384 178
385 130
386 130
387 112
388 412
389 111
390 391
391 39
392 346
393 337
394 114
395 74
396 3
397 481
398 384
399 405
400 405

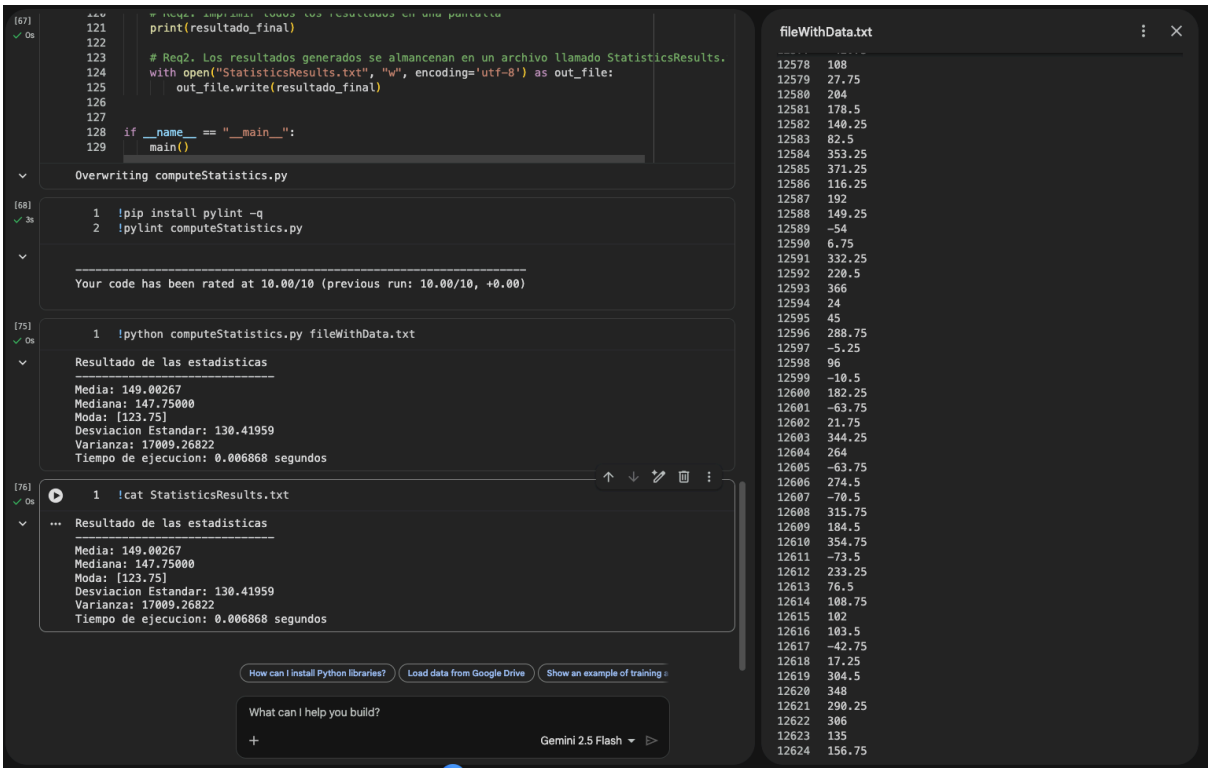
TC2


print(resultado_final)
# Req2. Los resultados generados se almancenan en un archivo llamado StatisticsResults.
with open("StatisticsResults.txt", "w", encoding='utf-8') as out_file:
out_file.write(resultado_final)
if __name__ == "__main__":
main()
Overwriting computeStatistics.py
!pip install pylint -q
!pylint computeStatistics.py
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
!python computeStatistics.py fileWithData.txt
Resultado de las estadisticas
Media: 250.78402
Mediana: 247.00000
Moda: [230.0]
Desviacion Estandar: 144.20779
Varianza: 20795.88804
Tiempo de ejecucion: 0.001112 segundos
!cat StatisticsResults.txt
Resultado de las estadisticas
Media: 250.78402
Mediana: 247.00000
Moda: [230.0]
Desviacion Estandar: 144.20779
Varianza: 20795.88804
Tiempo de ejecucion: 0.001112 segundos
How can I install Python libraries?
Load data from Google Drive
Show an example of training
What can I help you build?
Gemini 2.5 Flash
fileWithData.txt
1931 346
1932 386
1933 371
1934 480
1935 184
1936 417
1937 175
1938 263
1939 127
1940 42
1941 247
1942 228
1943 237
1944 10
1945 61
1946 106
1947 126
1948 500
1949 164
1950 437
1951 487
1952 480
1953 230
1954 164
1955 403
1956 219
1957 148
1958 451
1959 474
1960 137
1961 274
1962 174
1963 453
1964 362
1965 249
1966 380
1967 392
1968 208
1969 305
1970 330
1971 201
1972 101
1973 248
1974 498
1975 211
1976 132
1977 127

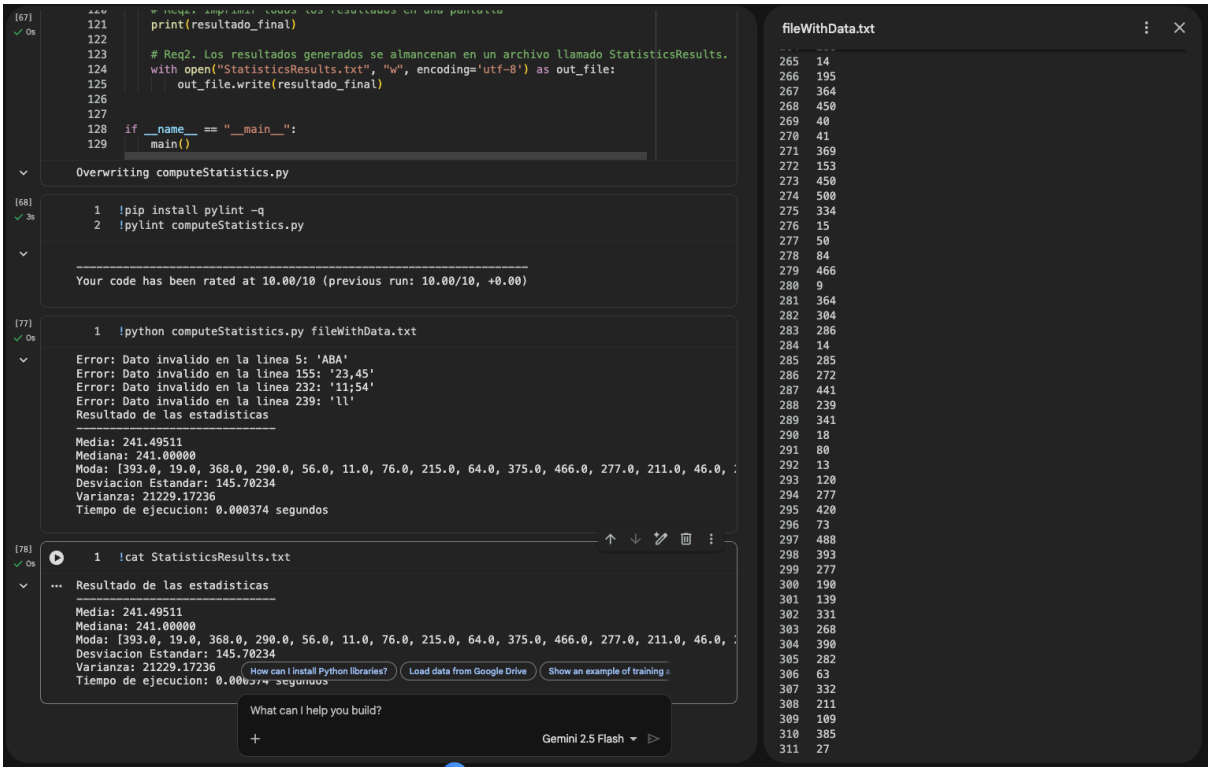
TC3


print(resultado_final)
# Req2. Los resultados generados se almancenan en un archivo llamado StatisticsResults.
with open("StatisticsResults.txt", "w", encoding='utf-8') as out_file:
out_file.write(resultado_final)
if __name__ == "__main__":
main()
Overwriting computeStatistics.py
!pip install pylint -q
!pylint computeStatistics.py
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
!python computeStatistics.py fileWithData.txt
Resultado de las estadisticas
Media: 249.77622
Mediana: 249.00000
Moda: [94.0]
Desviacion Estandar: 145.32361
Varianza: 21118.95039
Tiempo de ejecucion: 0.006417 segundos
!cat StatisticsResults.txt
Resultado de las estadisticas
Media: 249.77622
Mediana: 249.00000
Moda: [94.0]
Desviacion Estandar: 145.32361
Varianza: 21118.95039
Tiempo de ejecucion: 0.006417 segundos
How can I install Python libraries?
Load data from Google Drive
Show an example of training
What can I help you build?
Gemini 2.5 Flash
fileWithData.txt
12578 465
12579 170
12580 88
12581 263
12582 281
12583 474
12584 27
12585 402
12586 221
12587 420
12588 310
12589 344
12590 46
12591 361
12592 256
12593 287
12594 390
12595 113
12596 466
12597 218
12598 303
12599 151
12600 220
12601 170
12602 429
12603 190
12604 50
12605 498
12606 218
12607 228
12608 44
12609 459
12610 101
12611 194
12612 375
12613 146
12614 70
12615 479
12616 151
12617 403
12618 118
12619 310
12620 271
12621 488
12622 33
12623 1
12624 322

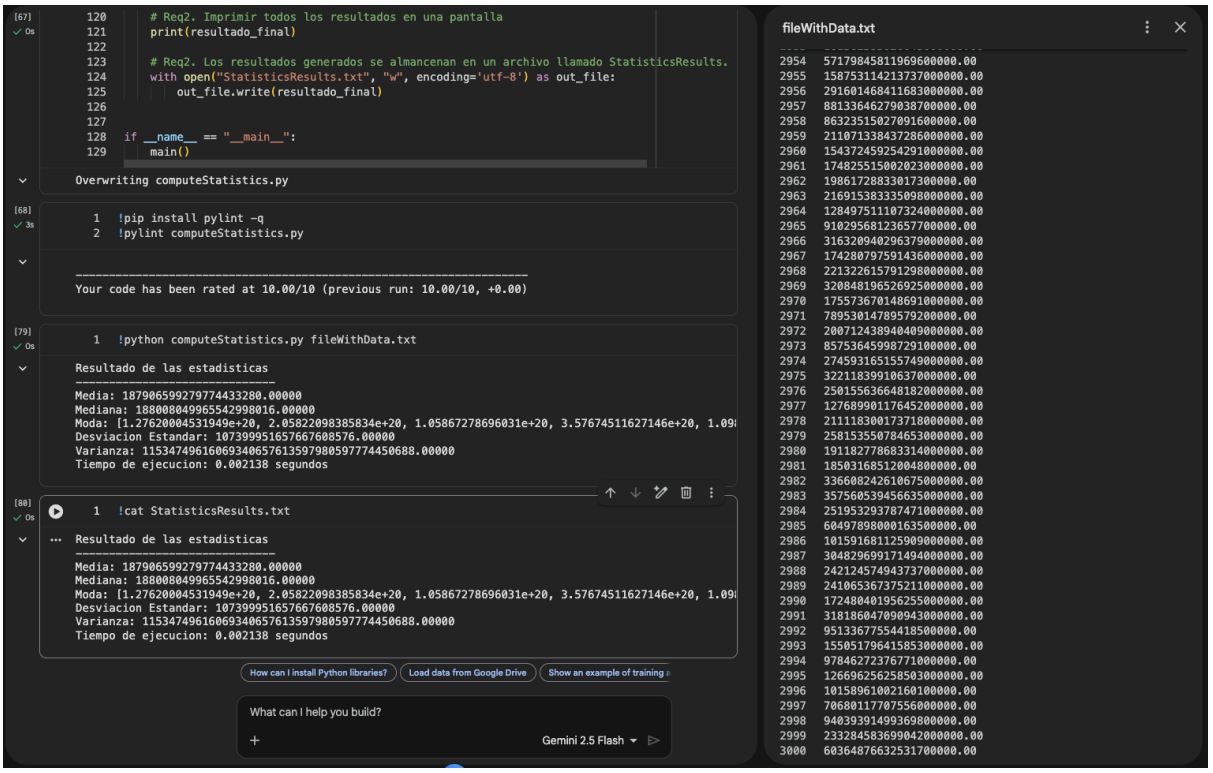
TC4


print(resultado_final)
# Req2. Los resultados generados se almancenan en un archivo llamado StatisticsResults.
with open("StatisticsResults.txt", "w", encoding='utf-8') as out_file:
out_file.write(resultado_final)
if __name__ == "__main__":
main()
Overwriting computeStatistics.py
!pip install pylint -q
!pylint computeStatistics.py
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
!python computeStatistics.py fileWithData.txt
Resultado de las estadisticas
Media: 149.00267
Mediana: 147.75000
Moda: [123.75]
Desviacion Estandar: 130.41959
Varianza: 17009.26822
Tiempo de ejecucion: 0.006868 segundos
!cat StatisticsResults.txt
Resultado de las estadisticas
Media: 149.00267
Mediana: 147.75000
Moda: [123.75]
Desviacion Estandar: 130.41959
Varianza: 17009.26822
Tiempo de ejecucion: 0.006868 segundos
How can I install Python libraries?
Load data from Google Drive
Show an example of training
What can I help you build?
Gemini 2.5 Flash
fileWithData.txt
12578 108
12579 27.75
12580 204
12581 178.5
12582 140.25
12583 82.5
12584 353.25
12585 371.25
12586 116.25
12587 192
12588 149.25
12589 -54
12590 6.75
12591 332.25
12592 220.5
12593 366
12594 24
12595 45
12596 288.75
12597 -5.25
12598 96
12599 -10.5
12600 182.25
12601 -63.75
12602 21.75
12603 344.25
12604 264
12605 -63.75
12606 274.5
12607 -70.5
12608 315.75
12609 184.5
12610 354.75
12611 -73.5
12612 233.25
12613 76.5
12614 108.75
12615 102
12616 103.5
12617 -42.75
12618 17.25
12619 304.5
12620 348
12621 290.25
12622 306
12623 135
12624 156.75

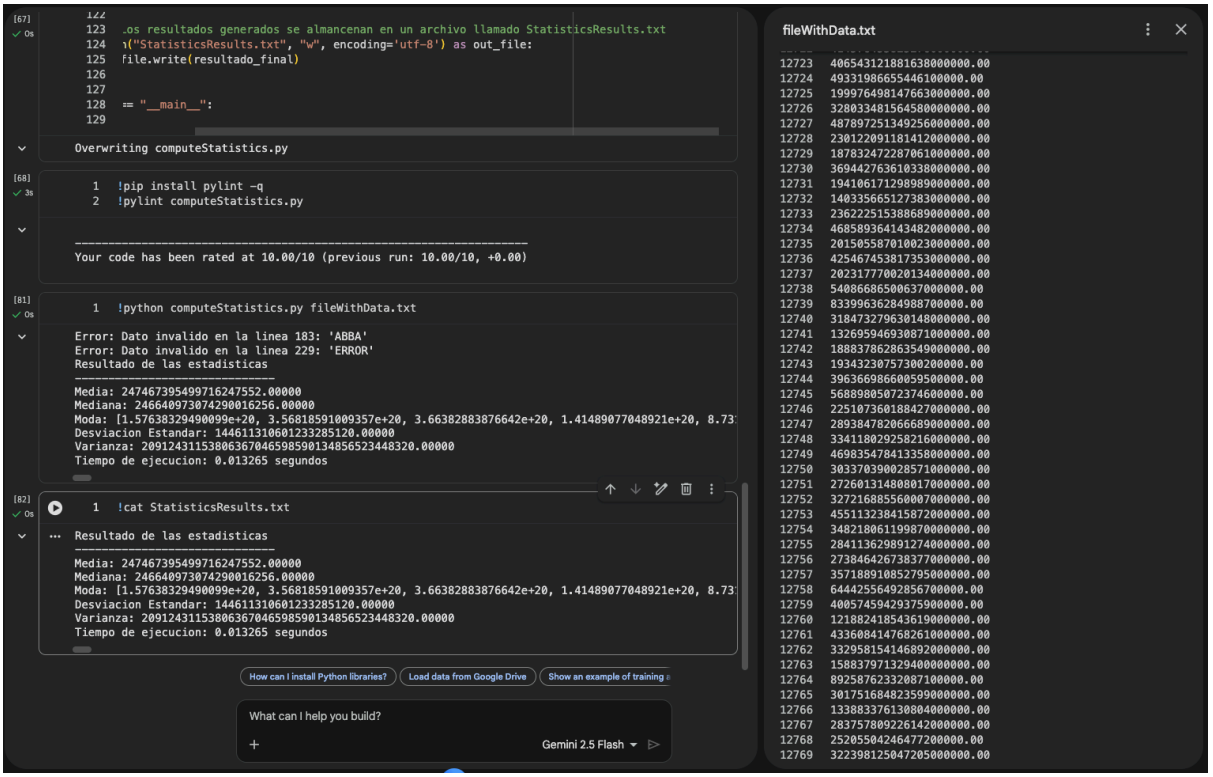
TC5


print(resultado_final)
# Req2. Los resultados generados se almancenan en un archivo llamado StatisticsResults.
with open("StatisticsResults.txt", "w", encoding='utf-8') as out_file:
out_file.write(resultado_final)
if __name__ == "__main__":
main()
Overwriting computeStatistics.py
!pip install pylint -q
!pylint computeStatistics.py
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
!python computeStatistics.py fileWithData.txt
Error: Dato invalido en la linea 5: 'ABA'
Error: Dato invalido en la linea 155: '23,45'
Error: Dato invalido en la linea 232: '11;54'
Error: Dato invalido en la linea 239: 'll'
Resultado de las estadisticas
Media: 241.49511
Mediana: 241.00000
Varianza: 21229.17236
Tiempo de ejecucion: 0.000374 segundos
!cat StatisticsResults.txt
How can I install Python libraries?
Load data from Google Drive
Show an example of training
What can I help you build?
Gemini 2.5 Flash
fileWithData.txt

TC6


print(resultado_final)
# Req2. Los resultados generados se almancenan en un archivo llamado StatisticsResults.
with open("StatisticsResults.txt", "w", encoding='utf-8') as out_file:
out_file.write(resultado_final)
if __name__ == "__main__":
main()
Overwriting computeStatistics.py
!pip install pylint -q
!pylint computeStatistics.py
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
!python computeStatistics.py fileWithData.txt
Resultado de las estadisticas
Media: 187906599279774433280.00000
Mediana: 188008049965542998016.00000
Desviacion Estandar: 1073999516576676085 76.00000
Varianza: 1153474961606934065761359798059774450688.00000
Tiempo de ejecucion: 0.002138 segundos
!cat StatisticsResults.txt
How can I install Python libraries?
Load data from Google Drive
Show an example of training
What can I help you build?
Gemini 2.5 Flash
fileWithData.txt

TC7


[67]
.os resultados generados se almacenan en un archivo llamado StatisticsResults.txt
("StatisticsResults.txt", "w", encoding='utf-8') as out_file:
file.write(resultado_final)
== "__main__":
Overwriting computeStatistics.py
[68]
3s
!pip install pylint -q
!pylint computeStatistics.py
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
[81]
!python computeStatistics.py fileWithData.txt
Error: Dato invalido en la linea 183: 'ABBA'
Error: Dato invalido en la linea 229: 'ERROR'
Resultado de las estadisticas
Media: 247467395499716247552.00000
Mediana: 246640973074290016256.00000
Desviacion Estandar: 144611310601233285120.00000
Varianza: 20912431153806367046598590134856523448320.00000
Tiempo de ejecucion: 0.013265 segundos
[82]
!cat StatisticsResults.txt
Resultado de las estadisticas
Media: 247467395499716247552.00000
Mediana: 246640973074290016256.00000
Desviacion Estandar: 144611310601233285120.00000
Varianza: 20912431153806367046598590134856523448320.00000
Tiempo de ejecucion: 0.013265 segundos
How can I install Python libraries?
Load data from Google Drive
Show an example of training
What can I help you build?
Gemini 2.5 Flash
fileWithData.txt
12723 4065431218816380000000.00
12724 4933198665544610000.00
12725 1999764981476630000000.00
12726 3280334815645800000000.00
12727 4878972513492560000000.00
12728 2301220911814120000000.00
12729 1878324722287061000000.00
12730 3694427636103388000000.00
12731 1941061712989890000000.00
12732 1403356651273830000000.00
12733 2362225153886890000000.00
12734 4685893641434820000000.00
12735 2015055870100230000000.00
12736 4254674538173530000000.00
12737 2023177700201340000000.00
12738 5408668650063700000.00
12739 8339963628498870000.00
12740 3184732796301480000000.00
12741 1326959469308710000000.00
12742 1888378628635490000000.00
12743 1934323075730020000.00
12744 3963669866005950000.00
12745 5688980507237460000.00
12746 2251073601884270000000.00
12747 2893847820666890000000.00
12748 3341180292582160000000.00
12749 4698354784133580000000.00
12750 3033703900285710000000.00
12751 2726013148080170000000.00
12752 3272168855600070000000.00
12753 4551132384158720000000.00
12754 3482180611998700000000.00
12755 2841136298912740000000.00
12756 2738464267383770000000.00
12757 3571889108527950000000.00
12758 6444255649285670000.00
12759 4005745942937590000.00
12760 1218824185436190000000.00
12761 4336084147682610000000.00
12762 3329581541468920000000.00
12763 1588379713294000000000.00
12764 8925876233208710000.00
12765 3017516848235990000000.00
12766 1338833761308040000000.00
12767 2837578092261420000000.00
12768 2520550424647720000.00
12769 3223981250472050000000.00