



VIT[®]
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

CSE3046-Programming for Data Science

Process: Obtain Data from Various Resources

Lab Assignment-4

Statistical Methods and Hypothesis

NAME :D VASANTH KUMAR

REG NO: 19BDS0083

Submitted Date: 15/10/2021

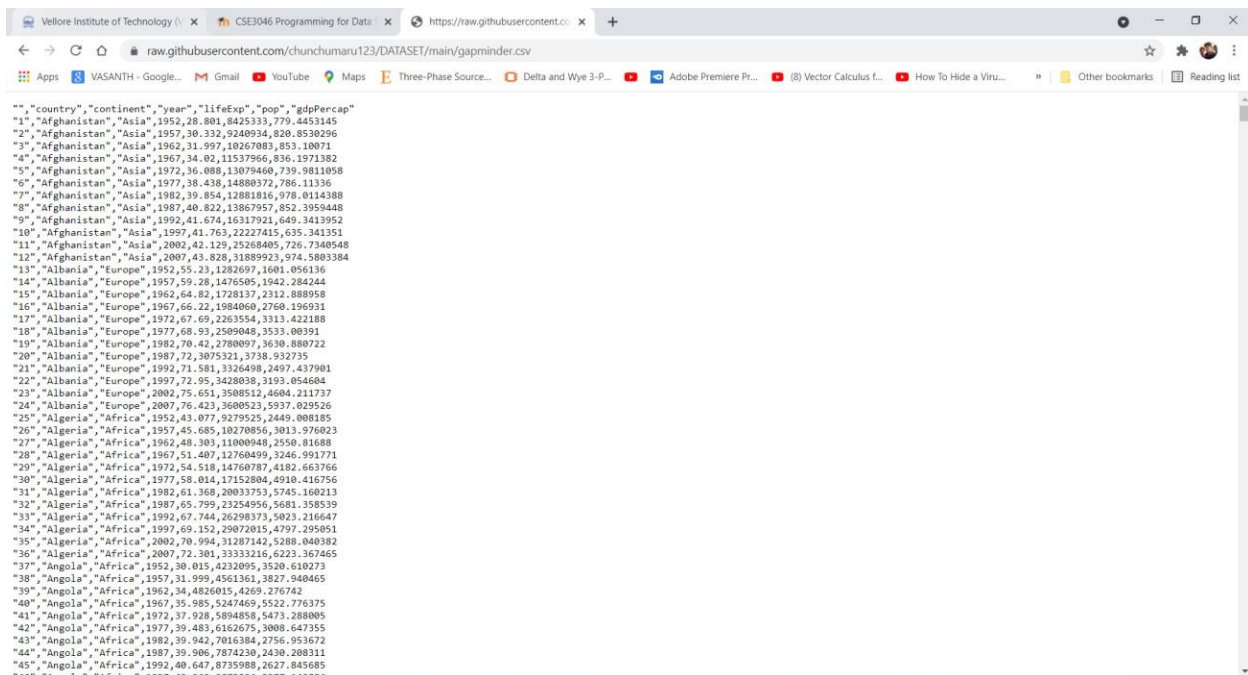
Course Instructor: Dr. Anthoniraj A

Retrieve the data set from the following URL,

<https://raw.githubusercontent.com/chunchumaru123/DATASET/main/gapminder.csv>

STEP 1:

THE DATA SET:



The screenshot shows a web browser window with the address bar displaying `https://raw.githubusercontent.com/chunchumaru123/DATASET/main/gapminder.csv`. The browser's tab bar shows several open tabs, including "Vellore Institute of Technology", "CSE3046 Programming for Data", and the current tab. The main content area displays a list of data rows from the gapminder.csv dataset, starting with a header row: `country,continent,year,lifeExp,pop,gdpPercap`. The data rows are numbered 1 through 46, representing various countries and their corresponding continent, year, life expectancy, population, and GDP per capita. The countries listed include Afghanistan, Albania, Algeria, Angola, and others, with data points for years 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, 2002, 2007, 2012, 2017, 2022, 2027, 2032, 2037, 2042, 2047, 2052, 2057, 2062, 2067, 2072, 2077, 2082, 2087, 2092, 2097, 2102, 2107, 2112, 2117, 2122, 2127, 2132, 2137, 2142, 2147, 2152, 2157, 2162, 2167, 2172, 2177, 2182, 2187, 2192, 2197, 2202, 2207, 2212, 2217, 2222, 2227, 2232, 2237, 2242, 2247, 2252, 2257, 2262, 2267, 2272, 2277, 2282, 2287, 2292, 2297, 2302, 2307, 2312, 2317, 2322, 2327, 2332, 2337, 2342, 2347, 2352, 2357, 2362, 2367, 2372, 2377, 2382, 2387, 2392, 2397, 2402, 2407, 2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, 2462, 2467, 2472, 2477, 2482, 2487, 2492, 2497, 2502, 2507, 2512, 2517, 2522, 2527, 2532, 2537, 2542, 2547, 2552, 2557, 2562, 2567, 2572, 2577, 2582, 2587, 2592, 2597, 2602, 2607, 2612, 2617, 2622, 2627, 2632, 2637, 2642, 2647, 2652, 2657, 2662, 2667, 2672, 2677, 2682, 2687, 2692, 2697, 2702, 2707, 2712, 2717, 2722, 2727, 2732, 2737, 2742, 2747, 2752, 2757, 2762, 2767, 2772, 2777, 2782, 2787, 2792, 2797, 2802, 2807, 2812, 2817, 2822, 2827, 2832, 2837, 2842, 2847, 2852, 2857, 2862, 2867, 2872, 2877, 2882, 2887, 2892, 2897, 2902, 2907, 2912, 2917, 2922, 2927, 2932, 2937, 2942, 2947, 2952, 2957, 2962, 2967, 2972, 2977, 2982, 2987, 2992, 2997, 3002, 3007, 3012, 3017, 3022, 3027, 3032, 3037, 3042, 3047, 3052, 3057, 3062, 3067, 3072, 3077, 3082, 3087, 3092, 3097, 3102, 3107, 3112, 3117, 3122, 3127, 3132, 3137, 3142, 3147, 3152, 3157, 3162, 3167, 3172, 3177, 3182, 3187, 3192, 3197, 3202, 3207, 3212, 3217, 3222, 3227, 3232, 3237, 3242, 3247, 3252, 3257, 3262, 3267, 3272, 3277, 3282, 3287, 3292, 3297, 3302, 3307, 3312, 3317, 3322, 3327, 3332, 3337, 3342, 3347, 3352, 3357, 3362, 3367, 3372, 3377, 3382, 3387, 3392, 3397, 3402, 3407, 3412, 3417, 3422, 3427, 3432, 3437, 3442, 3447, 3452, 3457, 3462, 3467, 3472, 3477, 3482, 3487, 3492, 3497, 3502, 3507, 3512, 3517, 3522, 3527, 3532, 3537, 3542, 3547, 3552, 3557, 3562, 3567, 3572, 3577, 3582, 3587, 3592, 3597, 3602, 3607, 3612, 3617, 3622, 3627, 3632, 3637, 3642, 3647, 3652, 3657, 3662, 3667, 3672, 3677, 3682, 3687, 3692, 3697, 3702, 3707, 3712, 3717, 3722, 3727, 3732, 3737, 3742, 3747, 3752, 3757, 3762, 3767, 3772, 3777, 3782, 3787, 3792, 3797, 3802, 3807, 3812, 3817, 3822, 3827, 3832, 3837, 3842, 3847, 3852, 3857, 3862, 3867, 3872, 3877, 3882, 3887, 3892, 3897, 3902, 3907, 3912, 3917, 3922, 3927, 3932, 3937, 3942, 3947, 3952, 3957, 3962, 3967, 3972, 3977, 3982, 3987, 3992, 3997, 4002, 4007, 4012, 4017, 4022, 4027, 4032, 4037, 4042, 4047, 4052, 4057, 4062, 4067, 4072, 4077, 4082, 4087, 4092, 4097, 4102, 4107, 4112, 4117, 4122, 4127, 4132, 4137, 4142, 4147, 4152, 4157, 4162, 4167, 4172, 4177, 4182, 4187, 4192, 4197, 4202, 4207, 4212, 4217, 4222, 4227, 4232, 4237, 4242, 4247, 4252, 4257, 4262, 4267, 4272, 4277, 4282, 4287, 4292, 4297, 4302, 4307, 4312, 4317, 4322, 4327, 4332, 4337, 4342, 4347, 4352, 4357, 4362, 4367, 4372, 4377, 4382, 4387, 4392, 4397, 4402, 4407, 4412, 4417, 4422, 4427, 4432, 4437, 4442, 4447, 4452, 4457, 4462, 4467, 4472, 4477, 4482, 4487, 4492, 4497, 4502, 4507, 4512, 4517, 4522, 4527, 4532, 4537, 4542, 4547, 4552, 4557, 4562, 4567, 4572, 4577, 4582, 4587, 4592, 4597, 4602, 4607, 4612, 4617, 4622, 4627, 4632, 4637, 4642, 4647, 4652, 4657, 4662, 4667, 4672, 4677, 4682, 4687, 4692, 4697, 4702, 4707, 4712, 4717, 4722, 4727, 4732, 4737, 4742, 4747, 4752, 4757, 4762, 4767, 4772, 4777, 4782, 4787, 4792, 4797, 4802, 4807, 4812, 4817, 4822, 4827, 4832, 4837, 4842, 4847, 4852, 4857, 4862, 4867, 4872, 4877, 4882, 4887, 4892, 4897, 4902, 4907, 4912, 4917, 4922, 4927, 4932, 4937, 4942, 4947, 4952, 4957, 4962, 4967, 4972, 4977, 4982, 4987, 4992, 4997, 5002, 5007, 5012, 5017, 5022, 5027, 5032, 5037, 5042, 5047, 5052, 5057, 5062, 5067, 5072, 5077, 5082, 5087, 5092, 5097, 5102, 5107, 5112, 5117, 5122, 5127, 5132, 5137, 5142, 5147, 5152, 5157, 5162, 5167, 5172, 5177, 5182, 5187, 5192, 5197, 5202, 5207, 5212, 5217, 5222, 5227, 5232, 5237, 5242, 5247, 5252, 5257, 5262, 5267, 5272, 5277, 5282, 5287, 5292, 5297, 5302, 5307, 5312, 5317, 5322, 5327, 5332, 5337, 5342, 5347, 5352, 5357, 5362, 5367, 5372, 5377, 5382, 5387, 5392, 5397, 5402, 5407, 5412, 5417, 5422, 5427, 5432, 5437, 5442, 5447, 5452, 5457, 5462, 5467, 5472, 5477, 5482, 5487, 5492, 5497, 5502, 5507, 5512, 5517, 5522, 5527, 5532, 5537, 5542, 5547, 5552, 5557, 5562, 5567, 5572, 5577, 5582, 5587, 5592, 5597, 5602, 5607, 5612, 5617, 5622, 5627, 5632, 5637, 5642, 5647, 5652, 5657, 5662, 5667, 5672, 5677, 5682, 5687, 5692, 5697, 5702, 5707, 5712, 5717, 5722, 5727, 5732, 5737, 5742, 5747, 5752, 5757, 5762, 5767, 5772, 5777, 5782, 5787, 5792, 5797, 5802, 5807, 5812, 5817, 5822, 5827, 5832, 5837, 5842, 5847, 5852, 5857, 5862, 5867, 5872, 5877, 5882, 5887, 5892, 5897, 5902, 5907, 5912, 5917, 5922, 5927, 5932, 5937, 5942, 5947, 5952, 5957, 5962, 5967, 5972, 5977, 5982, 5987, 5992, 5997, 6002, 6007, 6012, 6017, 6022, 6027, 6032, 6037, 6042, 6047, 6052, 6057, 6062, 6067, 6072, 6077, 6082, 6087, 6092, 6097, 6102, 6107, 6112, 6117, 6122, 6127, 6132, 6137, 6142, 6147, 6152, 6157, 6162, 6167, 6172, 6177, 6182, 6187, 6192, 6197, 6202, 6207, 6212, 6217, 6222, 6227, 6232, 6237, 6242, 6247, 6252, 6257, 6262, 6267, 6272, 6277, 6282, 6287, 6292, 6297, 6302, 6307, 6312, 6317, 6322, 6327, 6332, 6337, 6342, 6347, 6352, 6357, 6362, 6367, 6372, 6377, 6382, 6387, 6392, 6397, 6402, 6407, 6412, 6417, 6422, 6427, 6432, 6437, 6442, 6447, 6452, 6457, 6462, 6467, 6472, 6477, 6482, 6487, 6492, 6497, 6502, 6507, 6512, 6517, 6522, 6527, 6532, 6537, 6542, 6547, 6552, 6557, 6562, 6567, 6572, 6577, 6582, 6587, 6592, 6597, 6602, 6607, 6612, 6617, 6622, 6627, 6632, 6637, 6642, 6647, 6652, 6657, 6662, 6667, 6672, 6677, 6682, 6687, 6692, 6697, 6702, 6707, 6712, 6717, 6722, 6727, 6732, 6737, 6742, 6747, 6752, 6757, 6762, 6767, 6772, 6777, 6782, 6787, 6792, 6797, 6802, 6807, 6812, 6817, 6822, 6827, 6832, 6837, 6842, 6847, 6852, 6857, 6862, 6867, 6872, 6877, 6882, 6887, 6892, 6897, 6902, 6907, 6912, 6917, 6922, 6927, 6932, 6937, 6942, 6947, 6952, 6957, 6962, 6967, 6972, 6977, 6982, 6987, 6992, 6997, 7002, 7007, 7012, 7017, 7022, 7027, 7032, 7037, 7042, 7047, 7052, 7057, 7062, 7067, 7072, 7077, 7082, 7087, 7092, 7097, 7102, 7107, 7112, 7117, 7122, 7127, 7132, 7137, 7142, 7147, 7152, 7157, 7162, 7167, 7172, 7177, 7182, 7187, 7192, 7197, 7202, 7207, 7212, 7217, 7222, 7227, 7232, 7237, 7242, 7247, 7252, 7257, 7262, 7267, 7272, 7277, 7282, 7287, 7292, 7297, 7302, 7307, 7312, 7317, 7322, 7327, 7332, 7337, 7342, 7347, 7352, 7357, 7362, 7367, 7372, 7377, 7382, 7387, 7392, 7397, 7402, 7407, 7412, 7417, 7422, 7427, 7432, 7437, 7442, 7447, 7452, 7457, 7462, 7467, 7472, 7477, 7482, 7487, 7492, 7497, 7502, 7507, 7512, 7517, 7522, 7527, 7532, 7537, 7542, 7547, 7552, 7557, 7562, 7567, 7572, 7577, 7582, 7587, 7592, 7597, 7602, 7607, 7612, 7617, 7622, 7627, 7632, 7637, 7642, 7647, 7652, 7657, 7662, 7667, 7672, 7677, 7682, 7687, 7692, 7697, 7702, 7707, 7712, 7717, 7722, 7727, 7732, 7737, 7742, 7747, 7752, 7757, 7762, 7767, 7772, 7777, 7782, 7787, 7792, 7797, 7802, 7807, 7812, 7817, 7822, 7827, 7832, 7837, 7842, 7847, 7852, 7857, 7862, 7867, 7872, 7877, 7882, 7887, 7892, 7897, 7902, 7907, 7912, 7917, 7922, 7927, 7932, 7937, 7942, 7947, 7952, 7957, 7962, 7967, 7972, 7977, 7982, 7987, 7992, 7997, 8002, 8007, 8012, 8017, 8022, 8027, 8032, 8037, 8042, 8047, 8052, 8057, 8062, 8067, 8072, 8077, 8082, 8087, 8092, 8097, 8102, 8107, 8112, 8117, 8122, 8127, 8132, 8137, 8142, 8147, 8152, 8157, 8162, 8167, 8172, 8177, 8182, 8187, 8192, 8197, 8202, 8207, 8212, 8217, 8222, 8227, 8232, 8237, 8242, 8247, 8252, 8257, 8262, 8267, 8272, 8277, 8282, 8287, 8292, 8297, 8302, 8307, 8312, 8317, 8322, 8327, 8332, 8337, 8342, 8347, 8352, 8357, 8362, 8367, 8372, 8377, 8382, 8387, 8392, 8397, 8402, 8407, 8412, 8417, 8422, 8427, 8432, 8437, 8442, 8447, 8452, 8457, 8462, 8467, 8472, 8477, 8482, 8487, 8492, 8497, 8502, 8507, 8512, 8517, 8522, 8527, 8532, 8537, 8542, 8547, 8552, 8557, 8562, 8567, 8572, 8577, 8582, 8587, 8592, 8597, 8602, 8607, 8612, 8617, 8622, 8627, 8632, 8637, 8642, 8647, 8652, 8657, 8662, 8667, 8672, 8677, 8682, 8687, 8692, 8697, 8702, 8707, 8712, 8717, 8722, 8727, 8732, 8737, 8742, 8747, 8752, 8757, 8762, 8767, 8772, 8777, 8782, 8787, 8792, 8797, 8802, 8807, 8812, 8817, 8822, 8827, 8832, 8837, 8842, 8847, 8852, 8857, 8862, 8867, 8872, 8877, 8882, 8887, 8892, 8897, 8902, 8907, 8912, 8917, 8922, 8927, 8932, 8937, 8942, 8947, 8952, 8957, 8962, 8967, 8972, 8977, 8982, 8987, 8992, 8997, 9002, 9007, 9012, 9017, 9022, 9027, 9032, 9037, 9042, 9047, 9052, 9057, 9062, 9067, 9072, 9077, 9082, 9087, 9092, 9097, 9102, 9107, 9112, 9117, 9122, 9127, 9132, 9137, 9142, 9147, 9152, 9157, 9162, 9167, 9172, 9177, 9182, 9187, 9192, 9197, 9202, 9207, 9212, 9217, 9222, 9227, 9232, 9237, 9242, 9247, 9252, 9257, 9262, 9267, 9272, 9277, 9282, 9287, 9292, 9297, 9302, 9307, 9312, 9317, 9322, 9327, 9332, 9337, 9342, 9347, 9352, 9357, 9362, 9367, 9372, 9377, 9382, 9387, 9392, 9397, 9402, 9407, 9412, 9417, 9422, 9427, 9432, 9437, 9442, 9447, 9452, 9457, 9462, 9467, 9472, 9477, 9482, 9487, 9492, 9497, 9502, 9507, 9512, 9517, 9522, 9527, 9532, 9537, 9542, 9547, 9552, 9557, 9562, 9567, 9572, 9577, 9582, 9587, 9592, 9597, 9602, 9607, 9612, 9617, 9622, 9627, 9632, 9637, 9642, 9647, 9652, 9657, 9662, 9667, 9672, 9677, 9682, 9687, 9692, 9697, 9702, 9707, 9712, 9717, 9722, 9727, 9732, 9737, 9742, 9747, 9752, 9757, 9762, 9767, 9772, 9777, 9782, 9787, 9792, 9797, 9802, 9807, 9812, 9817, 9822, 9827, 9832, 9837, 9842, 9847, 9852, 9857, 9862, 9867, 9872, 9877, 9882, 9887, 9892, 9897, 9902, 9907, 9912, 9917, 9922, 9927, 9932, 9937, 9942, 9947, 9952, 9957, 9962, 9967, 9972, 9977, 9982, 9987, 9992, 9997, 10002, 10007, 10012, 10017, 10022, 10027, 10032, 10037, 10042, 10047, 10052, 10057, 10062, 10067, 10072, 10077, 10082, 10087, 10092, 10097, 10102, 10107, 10112, 10117, 10122, 10127, 10132, 10137, 10142, 10147, 10152, 10157, 10162, 10167, 10172, 10177, 10182, 10187, 10192, 10197, 10202, 10207, 10212, 10217, 10222, 10227, 10232, 10237, 10242, 10247, 10252, 10257, 10262, 10267, 10272, 10277, 10282, 10287, 10292, 10297, 10302, 10307, 10312, 10317, 10322, 10327, 10332, 10337, 10342, 10347, 10352, 10357, 10362, 10367, 10372, 10377, 10382, 10387, 10392, 10397, 10402, 10407, 10412, 10417, 10422, 10427, 10432, 10437, 10442, 10447, 10452, 10457, 10462, 10467, 10472, 10477, 10482, 10487, 10492, 10497, 10502, 10507, 10512, 10517, 10522, 10527, 10532, 10537, 10542, 10547, 10552, 10557, 10562, 10567, 10572, 10577, 10582, 10587, 10592, 10597, 10602, 10607, 10612, 10617, 10622, 10627, 10632, 10637, 10642, 10647, 10652, 10657, 10662, 10667, 10672, 10677, 10682, 10687, 10692, 10697, 10702, 10707, 10712, 10717, 10722, 10727, 10732, 10737, 10742, 10747, 10752, 10757, 10762, 10767, 10772, 10777, 10782, 10787, 10792, 10797, 10802, 10807, 10812, 10817, 10822, 10827, 10832, 10837, 10842, 10847, 10852, 10857, 10862, 10867, 10872, 10877, 10882, 10887, 10892, 10897, 10902, 10907, 10912, 10917, 10922, 10927, 10932, 10937, 10942, 10947, 10952, 10957, 10962, 10967, 10972, 10977, 10982, 10987, 10992, 10997, 11002, 11007, 11012, 11017, 11022, 11027, 11032, 11037, 11042, 11047, 11052, 11057, 11062, 11067, 11072, 11077, 11082, 11087, 11092, 11097, 11102, 11107, 11112, 11117, 11122, 11127, 11132, 11137, 11142, 11147, 11152, 11157, 11162, 11167, 11172, 11177, 11182, 11187, 11192, 11197, 11202, 11207, 11212, 11217, 11222, 11227, 11232, 11237, 11242, 11247, 11252, 11257, 11262, 11267, 11272, 11277, 11282, 11287, 11292, 11297, 11302, 11307, 11312, 11317, 11322, 11327, 11332, 11337, 11342, 11347, 11352, 11357, 11362, 11367, 11372, 11377, 11382, 11387, 11392, 11397, 11402, 11407, 11412, 11417, 11422, 11427, 11432, 11437, 11442, 11447, 11452, 11457, 11462, 11467, 11472, 11477, 11482, 11487, 11492, 11497, 11502, 11507, 11512, 11517, 11522, 11527, 11532, 11537, 11542, 11547, 11552, 11557, 11562, 11567, 11572, 11577, 11582, 11587, 11592, 11597, 11602, 11607, 11612, 11617, 116

STEP 2:

The aim of this activity is to understand the various statistical methods and empirical rule. You can use the same data set which was used for Missing Data Imputation. You are expected to analyze the data without using any predefined R functions (You have to use user-defined functions for ALL methods expect plots and graphs - E.g. Do not use predefined function `mean()` instead of writing your own function with simple loop `class_mean()`)

1. Find Mean (0.5)
2. Find Median (0.5)
3. Find Mode (1)
4. Find IQR (1)
5. Find Standard Deviation (1)
6. Find Probability values on Empirical Rule (1)
7. Plot the Graph/Histogram/Normal Distribution and Compare your functions return value with predefined functions in R for mean, median, IQR, and sd. (1)
8. Formulate the Null Hypothesis and Alternative Hypothesis for your data set and prove it based on the p-value. (3 Marks)

SAMPLE CODE:

19BDS0083.R

```
1 |
2 #-----
3 #WRITING META DATA
4
5 #USER INFORMATION : 19BDS0083 D VASANTH KUMAR
6
7 #DATA SOURCE : chunchumaru123/DATASET/gapminder.csv
8
9 #Description: Per-capita GDP (Gross domestic product) is given in units of international dollars, "a hypot
10
11 #Data shape: 1704 rows and 7 columns
12 #TAGS FOR THE DATA SET : MULTIVARIATE,TIME-SERIES,STATISTICAL DATA
13
14 #-----
15 #LIBRARIES USED
16 library(rvest)
17 library(dplyr)
18 library(tidyr)
19 library(utils)
20 library(ggplot2)
21
22 #READING THE DATA SET
23 gap <- read.csv("https://raw.githubusercontent.com/chunchumaru123/DATASET/main/gapminder.csv")
24 gap <- gap[2:7]
25
26 #STORING THE DATA SET IN NEW VARIABLE KEEPS THE ORIGINALITY OF THE DATA SET BEFORE CLEANING
27 df1 <- data.frame(gap)
28
29 View(df1)
30
31 str(df1)
32 summary(df1)
33 attach(df1)
34 #-----
35 #CALCULATING MEAN OF A COLUMN
36 #FUNCTION TO CALCULATE MEAN
37 get_mean <- function(df1)
38 {
39   b=0
40   n=0
41   for(i in df1$pop)
42   {
43     b <- b+i
44     n=n+1
45   }
46   mean_1=b/n
47   return(mean_1)
48 }
49 #Main Function Calling- User defined
50 get_mean1 <- get_mean(df1)
51 print("Mean:")
52 print(get_mean1)
53 #checking mean with inbuilt Function
```

```

56 #CALCULATING MEDIAN OF A COLUMN
57
58 n=dim(df1)[1]
59 #THE COLUMN IS SORTED USING INBUILT FUNCTION
60 s_df1=sort(df1$pop)
61 #FUNCTION TO CALCULATE MEDIAN
62 get_median <- function(s_df1)
63 {
64     if(n%%2==0)
65     {
66         median1=s_df1[n/2]
67         median2=s_df1[(n-1)/2]
68         medians=(median1+median2)/2
69     }else{
70         medians=s_df1[n/2]
71     }
72     return(medians)
73 }
74 #Main Function Calling- User defined
75 get_median <- get_median(s_df1)
76 print(paste("Median:",get_median))
77 #checking median inbuilt Function
78 median(s_df1)
79 #-----
80 #CALCULATING MODE OF A COLUMN
81 #FUNCTION TO CALCULATE MODE
82 Mode <- function(x) {
83     ux <- unique(x)
84     mode1 <- ux[which.max(tabulate(match(x, ux)))]
85     return(mode1)
86 }
87
88 x=df1$pop
89 #Main Function Calling- User defined
90 get_mode <- Mode(x)
91 print(paste("Mode",get_mode))
92
93 #Testing with inbuilt Function
94 #NO BUILT-IN FUNCTION FOR MODE
95 #-----
96 #CALCULATING INTER QUARTILE RANGE OF A COLUMN
97 #INITIALIZING THE GLOBAL VARIABLES
98 x=df1$pop
99 n=dim(df1)[1]
100 z=0
101 r=0
102 #THE COLUMN POP AFTER SORTING
103 x1=sort(x)

```

SAMPLE DATA SET:

country	continent	year	lifeExp	pop	gdpPerCap
1 Afghanistan	Asia	1952	28.801	8425333	779.4453
2 Afghanistan	Asia	1957	30.332	9240934	820.8530
3 Afghanistan	Asia	1962	31.997	10267083	853.1007
4 Afghanistan	Asia	1967	34.020	11537966	836.1971
5 Afghanistan	Asia	1972	36.088	13079460	739.9811
6 Afghanistan	Asia	1977	38.438	14880372	786.1134
7 Afghanistan	Asia	1982	39.654	12861616	978.0114
8 Afghanistan	Asia	1987	40.822	13867997	852.3959
9 Afghanistan	Asia	1992	41.674	16317821	649.3414
10 Afghanistan	Asia	1997	41.763	22227415	635.3414
11 Afghanistan	Asia	2002	42.129	25266405	726.7341
12 Afghanistan	Asia	2007	43.828	31869923	974.5803
13 Albania	Europe	1952	55.230	1282897	1601.0561
14 Albania	Europe	1957	59.280	1476305	1942.2842
15 Albania	Europe	1962	64.820	1728137	2312.8890
16 Albania	Europe	1967	66.220	1984060	2760.1969
17 Albania	Europe	1972	67.690	2263354	3313.4222
18 Albania	Europe	1977	68.930	2509048	3533.0039
19 Albania	Europe	1982	70.420	2780097	3630.8807
20 Albania	Europe	1987	72.000	3075321	3738.9327
21 Albania	Europe	1992	71.581	3326498	2497.4379
22 Albania	Europe	1997	72.950	3428038	3193.0546
23 Albania	Europe	2002	75.651	3508512	4604.2117
24 Albania	Europe	2007	76.423	3600923	5937.0295
25 Algeria	Africa	1952	43.077	9279525	2449.0062
26 Algeria	Africa	1957	45.685	10270856	3013.9760
27 Algeria	Africa	1962	48.303	11000948	2550.8169
28 Algeria	Africa	1967	51.407	12760499	3246.9918
29 Algeria	Africa	1972	54.518	14760787	4182.6638
30 Algeria	Africa	1977	58.014	17152804	4910.4168
31 Algeria	Africa	1982	61.568	20033753	6745.1602

Showing 1 to 31 of 1,704 entries, 6 total columns

CODE IN R STUDIO:

```
# _____  
-----
```

```
#WRITING META DATA
```

```
#USER INFORMATION : 19BDS0083 D VASANTH KUMAR
```

```
#DATA SOURCE : chunchumaru123/DATASET/gapminder.csv
```

```
#Description: Per-capita GDP (Gross domestic product) is given in units  
of international dollars, “a hypothetical unit of currency that has the  
same purchasing power parity that the U.S. dollar had in the United  
States at a given point in time” – 2005, in this case.
```

```
#Data shape: 1704 rows and 7 columns
```

```
#TAGS FOR THE DATA SET : MULTIVARIATE,TIME-SERIES,STATISTICAL  
DATA
```

```
# _____  
-----
```

```
#LIBRARIES USED
```

```
library(rvest)
```

```
library(dplyr)
```

```
library(tidyr)
```

```
library(utils)
```

```
library(ggplot2)
```

```
#READING THE DATA SET
```

```
gap <-
```

```
read.csv("https://raw.githubusercontent.com/chunchumaru123/DATASETS/main/gapminder.csv")
```

```
gap <- gap[2:7]
```

```
#STORING THE DATA SET IN NEW VARIABLE KEEPS THE ORIGINALITY OF  
THE DATA SET BEFORE CLEANING
```

```
df1 <- data.frame(gap)
```

```
View(df1)
```

```
str(df1)
```

```
summary(df1)
```

```
attach(df1)
```

```
# _____
```

```
#CALCULATING MEAN OF A COLUMN
```

```
#FUNCTION TO CALCULATE MEAN
```



```
get_mean <- function(df1)
{
  b=0
  n=0
  for(i in df1$pop)
  {
    b <- b+i
    n=n+1
  }
  mean_1=b/n
  return(mean_1)
}

#Main Function Calling- User defined
get_mean1 <-get_mean(df1)
print("Mean:")
print(get_mean1)

#checking mean with inbuilt Function
print(paste("Mean(inbuilt-function):",round(mean(df1$pop),0)))

# _____
```

```
#CALCULATING MEDIAN OF A COLUMN
```

```
n=dim(df1)[1]
```

```
#THE COLUMN IS SORTED USING INBUILT FUNCTION
```

```
s_df1=sort(df1$pop)
```

```
#FUNCTION TO CALCULATE MEDIAN
```

```
get_median <- function(s_df1)
```

```
{
```

```
  if(n%%2==0)
```

```
  {
```

```
    median1=s_df1[n/2]
```

```
    median2=s_df1[(n-1)/2]
```

```
    medians=(median1+median2)/2
```

```
  }else{
```

```
    medians=s_df1[n/2]
```

```
  }
```

```
  return(medians)
```

```
}
```

```
#Main Function Calling- User defined
```

```
get_median <- get_median(s_df1)
```

```
print(paste("Median:",get_median))
```

```
#checking median inbuilt Function
```

```
median(s_df1)
```

```
#
```

```
#CALCULATING MODE OF A COLUMN
```

```
#FUNCTION TO CALCULATE MODE
```

```
Mode <- function(x) {
```

```
  ux <- unique(x)
```

```
  mode1 <- ux[which.max(tabulate(match(x, ux)))]
```

```
  return(mode1)
```

```
}
```

```
x=df1$pop
```

```
#Main Function Calling- User defined
```

```
get_mode <- Mode(x)
```

```
print(paste("Mode",get_mode))
```

```
#Testing with inbuilt Function
```

```
#NO BUILT-IN FUNCTION FOR MODE
```

```
#
```

```
#CALCULATING INTER QUARTILE RANGE OF A COLUMN
```

```
#INITIALIZING THE GLOBAL VARIABLES
```

```
x=df1$pop
```

```
n=dim(df1)[1]
```

```
z=0
```

```
r=0
```

```
#THE COLUMN POP AFTER SORTING
```

```
x1=sort(x)
```

```
#Function to give index of the median
```

```
median1 <- function(x1,a,n){
```

```
  z = n - a + 1
```

```
  z = (z + 1) / 2 - 1
```

```
  return(z + a)
```

```
}
```

```
#Function to give iqr
```

```
iqr <-function(x,n){
```

```
  #Index of median of entire data
```

```
  mid_index=median1(x1,0,n)
```

```
  #Median of first half
```

```

Q1 <- x1[median1(x1,0,mid_index)]

#Median of second half

Q3 <-x1[median1(x1, mid_index, n)]

#IQR calculation

return(Q3-Q1)

}

#Main Function Calling- User defined

print(paste("IQR(user-defined func):",iqr(x,n)))

#Testing with inbuilt Function

IQR(x1)

#_____

#CALCULATING STANDARD DEVIATION OF A COLUMN

n=dim(df1)[1]

#FUNCITON TO CALCULATE SD

calculatesd <- function(x)

{

  sum=0.0

  SD=0.0

  mean_new=get_mean1

```

```

for(i in x)
{
  SD=SD+(i-mean_new)^2
}
return(sqrt(SD/n))
}

#Main Function Calling- User defined
get_sd <-calculatesd(df1$pop)
print(paste("Standard Deviation:",round(calculatesd(df1$pop),0)))

#Testing with inbuilt Function
sd(df1$pop)

# _____

#Empirical Rule Check

#68%, 95%, and 99.7% Sigma rule

the.mean=mean(df1$pop)
the.sd=get_sd

#calculate the lower and upper bounds:
lower.bounds = the.mean - 1:3*the.sd

```

```
upper.bounds = the.mean + 1:3*the.sd
```

```
#calculate the proportion of observations between each pair of the  
upper and lower bounds
```

```
one.sd = mean(df1$pop > lower.bounds[1] & df1$pop <  
upper.bounds[1]) #68%
```

```
two.sd = mean(df1$pop > lower.bounds[2] & df1$pop <  
upper.bounds[2]) #95%
```

```
three.sd = mean(df1$pop > lower.bounds[3] & df1$pop <  
upper.bounds[3]) #99.7%
```

```
# _____
```

```
#Histogram
```

```
hist(df1$pop)
```

```
# after seeing the histogram it is obvious that the data is positively  
skewed.Hence to reduce it log is used.
```

```
#A log transformation is a process of applying a logarithm to data to  
reduce its skew.usually done when the numbers are highly skewed to  
reduce the skew so the data can be understood easier
```

```
hist(log(df1$pop),main = "population plot ",col = "darkmagenta")
```

```
#Normal Probability Plot
```

```
qqnorm(log(df1$pop),main = "Normal Probability Plot population")
```

```
qqline(log(df1$pop))
```

```
# _____
```

```
#LINE GRAPH
```

```
plot(log(df1$pop),type = "o",main="population chart",col="RED",xlim =  
c(0,200))
```

```
# _____
```

```
#NORMAL DISTRIBUTION CURVE
```

```
x <- log(df1$pop)
```

```
y <- dnorm(x,mean=get_mean1,sd=get_sd)
```

```
plot(x,y,col="darkmagenta")
```

```
# _____
```

```
#HYPOTHESIS TESTING
```

```
#performing two-tailed-t-test
```


#Step 1: State the null and alternate hypothesis

(null hypothesis) H0: the difference in the lifeExp of countries Ireland and south Africa is zero

(alternate hypothesis) H1: there exists a difference between lifeExp of countries Ireland and south Africa

#Step 2: Collect data-creating new data frame

##Filtering the data by country of interest South africa and Ireland

```
df_dup <- df1 %>% select(country,lifeExp)%>%
```

```
filter(country=="South Africa" | country=="Ireland")%>%
```

```
group_by(country)
```

#Step 3: Perform a statistical test

##Using t.test

```
t.test(data=df_dup,lifeExp ~country)
```

#Step 4: Decide whether the null hypothesis is supported or refuted

#the Result tells us the average lifeExp in Ireland and South Africa is 73 years and 53 years respectively

with a difference of 20 years

Since the P- value is close to zero, it is unlikely that null hypothesis will happen

#So there exists a difference between lifeExp for the countries according to alternate hypothesis

#we can see that the difference in means for our sample data is 73.01725 and 53.99317 , and the confidence interval shows that the true difference

#in means is between 15.07022 and 22.97794. So, 95% of the time, the true difference in means will be different from 0. Our p-value of $4.466e-09$ is

#much smaller than 0.05, so we can reject the null hypothesis of no difference and say with a high degree of confidence that the true difference in means is not equal to zero.

#So null hypothesis is rejected

SAMPLE OUTPUT:

	country	lifeExp
1	Ireland	66.910
2	Ireland	68.900
3	Ireland	70.290
4	Ireland	71.080
5	Ireland	71.280
6	Ireland	72.030
7	Ireland	73.100
8	Ireland	74.360
9	Ireland	75.467
10	Ireland	76.122
11	Ireland	77.783
12	Ireland	78.885
13	South Africa	45.009
14	South Africa	47.985
15	South Africa	49.951
16	South Africa	51.927
17	South Africa	53.696
18	South Africa	55.527
19	South Africa	58.161
20	South Africa	60.834
21	South Africa	61.888
22	South Africa	60.236
23	South Africa	53.365
24	South Africa	49.339

Showing 1 to 24 of 24 entries, 2 total columns

```

>
> #HYPOTHESIS TESTING
>
> #performing two-tailed-t-test
> #Step 1: State the null and alternate hypothesis
>
> # (null hypothesis) H0: the difference in the lifeExp of countries Ireland and south Africa is zero
> # (alternate hypothesis) H1: there exists a difference between lifeExp of countries Ireland and south Africa
>
> #Step 2: Collect data-creating new data frame
>
> ##Filtering the data by country of interest South africa and Ireland
> df_dup <- df1 %>% select(country,lifeExp)%>%
+   filter(country=="South Africa" | country=="Ireland")%>%
+   group_by(country)
>
> #Step 3: Perform a statistical test
>
> ##Using t.test
> t.test(data=df_dup,lifeExp ~country)
      welch Two Sample t-test

data:  lifeExp by country
t = 10.067, df = 19.109, p-value = 4.466e-09
alternative hypothesis: true difference in means between group Ireland and group South Africa is not equal to 0
95 percent confidence interval:
 15.07022 22.97794
sample estimates:
 mean in group Ireland mean in group South Africa
      73.01725          53.99317

>
> #Step 4: Decide whether the null hypothesis is supported or refuted
>
> #the Result tells us the average lifeExp in Ireland and South Africa is 73 years and 53 years respectively
> # with a difference of 20 years
>
> # Since the P- value is close to zero,it is unlikely that null hypothesis will happen
> #So there is exists a difference between lifeExp for the countries according to alternate hypothesis
>
> #we can see that the difference in means for our sample data is 73.01725 and 53.99317 , and the confidence interval shows that the true difference
> #in means is between 15.07022 and 22.97794. So, 95% of the time, the true difference in means will be different from 0. Our p-value of 4.466e-09 is
> #much smaller than 0.05, so we can reject the null hypothesis of no difference and say with a high degree of confidence that the true difference in means is not equal to zero.
>
> #So null hypothesis is rejected
> |

```