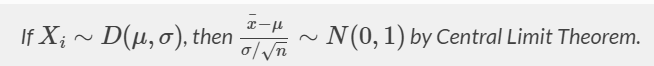
|  |  |
| --- | --- |
| 1a. Za 20 podatoci da se nacrta bpxplot |  |
| -1a. Calculate and display the summary statistics n, x¯, and s, the sample standard deviation.  -1b. Create a graph to display the distribution of this data. Choose which type of graph is effective for this purpose.  -1c. Describe the distribution of daily maximum temperatures as shown by the graph. Is the distribution strongly skewed? Are there unusual measurements? | |
| 2 Compare the standard normal distribution with the t distribution with 19 degrees of freedom.  -2a. Calculate the 0.975 quantiles from each of these two distribution.  -2b. On the same graph, display the density functions of these two distributions, using blue for normal and red for t. - Add colored (using the same color scheme) dashed vertical lines at the corresponding 0.975 quantiles. - Shade the area in tail areas below the 0.025 and above the 0.975 quantiles of each distribution, setting alpha = 0.5 for partial transparency. | ggplot() +  geom\_t\_density(df = 19, color = "red") +  geom\_norm\_density(mu = 0, sigma = 1, color = "black") +  xlim(c(-3, 3)) +  labs(title = "Standard Normal vs. Three t Distributions",  subtitle = "Z = black, t(2) = red, t(10) = blue, t(30) = green")  qnorm(0.975)  ## [1] 1.959964  qnorm(0.025)  qt(0.975, df = 19)  qt(0.025, df =19)  crtanje krivi  test\_stat = (xbar - 240) / (s / sqrt(n))  gt(df = n - 1) +  geom\_vline(xintercept = test\_stat, color = "red") |
| 3 Using the data from Problem 1: -3a. Construct a 95% confidence interval for μ using the theory of the t distribution by direct calculation using the summary statistics from the first part of the problem.  -3b. Then use the t.test() function to verify your calculation.  -3c. Interpret this interval in context. | C = 0.95  moe = qt(C + (1-C)/2, df = n - 1) \* se  left = xbar - moe  right = xbar + moe  c(left, right)  ## [1] 234.3172 236.6952  times = bm %>% filter(Sex == "female" & Age\_Range == "18-34" & Year == 2010) %>% pull(Time) # pull(x) works like dataframe$x, extracts the column x from the dataframe as a vector  times treba da e filtrirana kolona max  mozes I tukda definiras nacin na formiranjeto  **t.test(times)**  t.test(times, conf.level = 0.9) |
|  |  |
|  |  |
|  |  |

bm\_summary = bm %>%

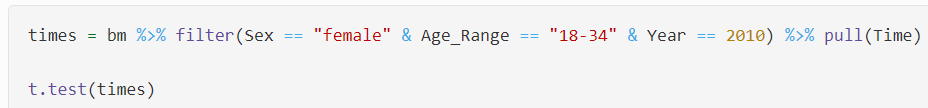
filter(Sex == "female", Age\_Range == "18-34", Year == 2010) %>%

summarize(averageTime = mean(Time), sdTime = sd(Time), n = n())

bm\_summary

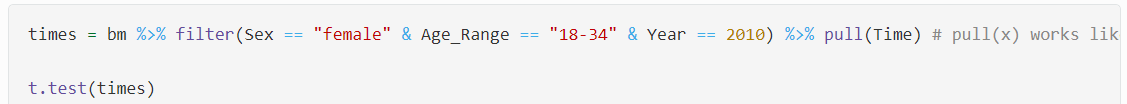


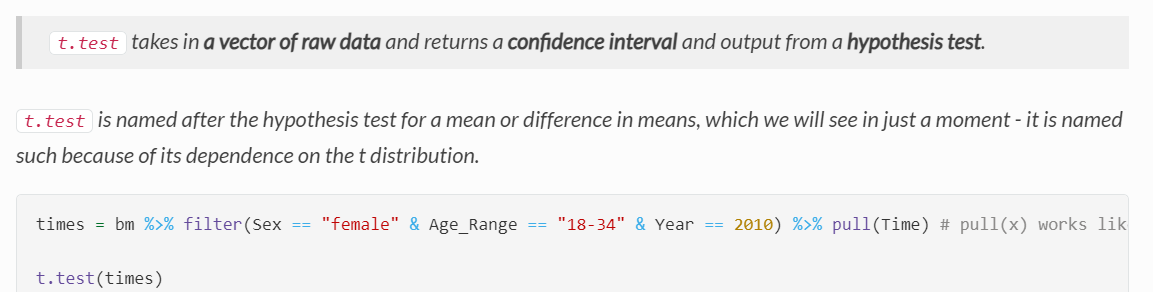




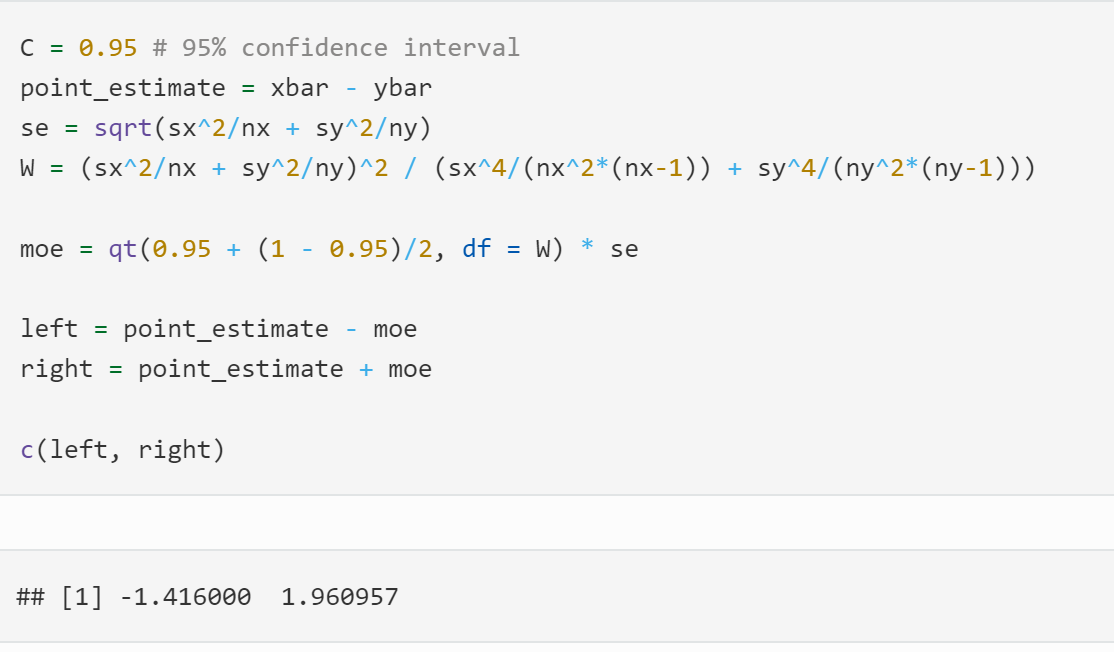
# pull(x) works like dataframe $x, extracts the column x from the dataframe as a vector

* Note that t.test **takes the full, n length vector of raw Xi** values, NOT a summarized version.









*We are 95% confident that the difference in true, underlying average times among 18-34 year old female runners between 2010 and 2011 is between -1.416 and 1.961.*

