

Estatística Aplicada

Laboratório 05 - Manhã

Aluno: Marcus Vinicius Leite Costa

Matrícula: 116110728

1)

a) Para a letra A, calcular μ e p da população, foi-se utilizado o seguinte código:

```
##### QUESTAO 01 #####

domicilios =
c(4,5,2,9,1,4,4,6,7,2,2,4,4,7,4,5,6,8,1,2,6,4,2,3,2,3,2,4,5,6,8,5,
2,3,4,1,6,3,2,3,5,4,8,5,4,2,4,3,2,4,5,9,5,6,4,3,4,5,4,2,9,8,18,8,7
,9,6,14,8,9,22,8,9,14,9,9,8,8,15,7,7,9,9,8,7,12,8,9,8,8)
qtdDomicilios = length(domicilios)

##### letra a #####

proporcaoMaiorQueCinco = function(populacao, qtdPopulacao) {
  contador = 0
  for(i in 1:qtdPopulacao) {
    if(populacao[i] > 5){
      contador = contador + 1
    }
  }

  return (contador/qtdPopulacao)
}

# Variavel que armazena a proporcao referente aos
# domicilios com mais de 5 comodos (populacional)
parametroPopP = proporcaoMaiorQueCinco(domicilios, qtdDomicilios)
# Variavel que armazena a media de comodos dos domicilios
(populacional)
parametroPopMi = mean(domicilios)
```

Os valores obtidos para parametroPopP e parametroPopMi foram:

parametroPop...	5.98888888888889
parametroPopP	0.477777777777778

b) Para gerar 10000 amostras de 20 domicílios, e calcular os limites inferiores/superiores para as estatísticas μ e p , foi-se utilizado o seguinte código:

```
##### letra B #####

qtdAmostra = 20

zy = 1.96
t_alfa = 2.262
limitesSuperioresP = c()
limitesInferioresP = c()
limitesSuperioresMi = c()
limitesInferioresMi = c()

for(j in 1:10000) {
  amostraDomicilios = sample(domicilios, qtdAmostra, replace =
FALSE)

  parametroP = proporcaoMaiorQueCinco(amostraDomicilios,
qtdAmostra)
  parametroMi = mean(amostraDomicilios)

  intervaloP = zy * (sqrt((parametroP*(1-parametroP)) /
qtdAmostra))
  intervaloMi = t_alfa * (sd(amostraDomicilios) /
sqrt(qtdAmostra))

  limiteInferiorP = parametroP - intervaloP
  limiteSuperiorP = parametroP + intervaloP

  limiteInferiorMi = parametroMi - intervaloMi
  limiteSuperiorMi = parametroMi + intervaloMi

  limitesSuperioresP = append(limitesSuperioresP, limiteSuperiorP)
  limitesInferioresP = append(limitesInferioresP, limiteInferiorP)
  limitesSuperioresMi = append(limitesSuperioresMi,
limiteSuperiorMi)
```

```

    limitesInferioresMi = append(limitesInferioresMi,
    limiteInferiorMi)

}

# Plotando em um dataframe os 10.000 valores obtidos dos
# limites superiores e inferiores para as estatísticas p e mi
frame = data.frame(limitesSuperioresP, limitesInferioresP,
limitesSuperioresMi, limitesInferioresMi)

```

c) Para obter-se a quantidade de parâmetros verdadeiros, e logo em seguida, obter-se a proporção do mesmo, para fins de comparação com o gama fornecido (95%), foi-se utilizado o seguinte código:

```

##### letra C #####

# Funcao que retorna a quantidade de parametros verdadeiros para
# os limites superiores e inferiores passados em relacao ao
parametro populacional

contaParametrosVerdadeiros = function(limitesSuperiores,
limitesInferiores, parametro) {
    paramVerdadeiro = 0
    tamanhoLimites = length(limitesSuperiores)

    for(k in 1:tamanhoLimites) {
        if(limitesSuperiores[k] >= parametro && limitesInferiores[k]
<= parametro) {
            paramVerdadeiro = paramVerdadeiro + 1
        }
    }

    return(paramVerdadeiro)
}

proporcaoP = contaParametrosVerdadeiros(limitesSuperioresP,
limitesInferioresP, parametroPopP) / 10000
proporcaoMi = contaParametrosVerdadeiros(limitesSuperioresMi,
limitesInferioresMi, parametroPopMi) / 10000

```

```
print(proporcaoP)
print(proporcaoMi)
```

Os valores obtidos para `proporcaoP` e `proporcaoMi` foram:

```
> print(proporcaoP)
[1] 0.9565
> print(proporcaoMi)
[1] 0.9654
>
```

Logo, fica nítido que esses valores se aproximam bastante do gama que foi previamente dado (95%)