

Tips & Tricks

Julia

Začít s

```
using Statistics, Distributions, Random, LinearAlgebra, Clustering, StatsBase, Plots, StatsPlots, HypothesisTests
```

```
dist(X; itr = eachrow) = [norm(vec1 - vec2) for vec1 in itr(X), vec2 in itr(X)]
```

```
function confint_var(s, n; level=0.95, tail=:both)
```

```
    α = 1 - level
```

```
    if tail == :both
```

```
        return ( (n-1)*s^2 / quantile(Chisq(n-1), 1 - α/2), (n-1)*s^2 / quantile(Chisq(n-1), α/2) )
```

```
    elseif tail == :left
```

```
        return ( -Inf, (n-1)*s^2 / quantile(Chisq(n-1), α) )
```

```
    elseif tail == :right
```

```
        return ( (n-1)*s^2 / quantile(Chisq(n-1), 1 - α), Inf )
```

```
    end
```

```
end
```

```
confint_std(s, n; kwargs...) = sign.(confint_var(s,n; kwargs...)) .* sqrt.(abs.( confint_var(s,n; kwargs...) ))
```

“ Potřebné balíčky:

```
] add Distributions, Clustering, StatsBase, Plots, StatsPlots, HypothesisTests
```

Statistika

- vektor průměrů `mean(X; dims = 1)`
- kovarianční matice `cov(X)`
- výběrová korelační matice `cor(X)`

- vlastní čísla a vlastní vektory výběrové kovarianční matice
`vals, vecs = eigen(X)`
 - procenta variability `cumsum(vals) / sum(vals) * 100`
 - hodí se na PCA
- matice vzdáleností `dist(X)` přičemž potřebujeme první definovat
`dist(X; itr = eachrow) = [norm(vec1 - vec2) for vec1 in itr(X), vec2 in itr(X)]`
- shlukování `hcl = hclust(dist(X); linkage = :single)` (více [zde](#))
 - dobré potom ještě použít `cuttree(hcl; k = <chtěný počet clusterů>)`
- výběrová distribuční funkce (vektorová data `X` - pouze jednorozměrná)
`plot((minimum(X) - 5):0.01:(maximum(X) + 5), y -> ecdf(X)(y))`

Pravděpodobnost

- kvantil na hladině α `quantile(<distribution>, α)` např.
`quantile(Normal(0,1), 0.95)`
- hodnota distribuční funkce v bodě `cdf(<distribution>, <where>)` např.
`cdf(Normal(0,1), 1.6)`

Testování hypotéz

- test střední hodnoty normálního rozdělení - **z-test** `OneSampleZTest(<výběrový průměr>, <směrodatná odchylka onoho normálního rozdělení>, <počet pozorování>, <testovaná hodnota>)` např.
`OneSampleZTest(mean(X), 0.15, 9, 10)` pro příklad 4/2
 - v případě, že testujeme střední hodnotu normálního rozdělení se směrodatnou odchylkou určenou z dat, můžeme použít `OneSampleZTest(X, <testovaná střední hodnota>)` např. `OneSampleZTest(X, 10)`
 - p-hodnotu můžeme získat také pomocí `pvalue(<test>)` např.
`pvalue(OneSampleZTest(mean(X), 0.15, 9, 10))`, přičemž ještě můžeme specifikovat "jednostrannost" tohoto testu
`pvalue(OneSampleZTest(mean(X), 0.15, 9, 10); tail=:left)`
 - obdobně pro interval spolehlivosti pomocí `confint`
- testy jsou dostupné [zde](#)
- **nepodařilo se mi najít test o rozptylu**
 - **konfidenční interval** testu o rozptylu: `confint_var(<směrodatná odchylka>, <počet pozorování>; level=<hladina významnosti>, tail=:both)`

```
function confint_var(s, n; level=0.95, tail=:both)
   $\alpha$  = 1 - level
  if tail == :both
    return ( (n-1)*s^2 / quantile(Chisq(n-1), 1 -  $\alpha$ /2), (n-1)*s^2 / quantile(Chisq(n-1),  $\alpha$ /2) )
  elseif tail == :left
    return ( -Inf, (n-1)*s^2 / quantile(Chisq(n-1),  $\alpha$ ) )
```

```

elseif tail == :right
    return ( (n-1)*s^2 / quantile(Chisq(n-1), 1 -  $\alpha$ ), Inf )
end
end

```

- obdobně lze použít i `confint_std` na IS pro směrodatnou odchylku
- párový t-test: `OneSampleTTest([1.8, 1.0, 2.2, 0.9, 1.5, 1.6], [1.5, 1.1, 2.0, 1.1, 1.4, 1.4])`
- asymptotické testy alternativního rozdělení můžeme řešit pomocí z-testu se směrodatnou odchylkou $\sqrt{m(1-m)}$, např. `OneSampleZTest(0.05, sqrt(0.05*(1-0.05)), 500, 0)`
- asymptotický test o shodě parametřů 2 alternativních rozdělení můžeme opět provést pomocí z-testu, kde $m^* = \frac{m_1 + m_2}{2}$ a tedy `OneSampleZTest((16-12)/150, sqrt(m*(1-m)*(1/150 + 1/150)), 1)`
 - používat radši místo společného odhadu `m` vzorec `m1*(1 - m1)/n1 + m2*(1 - m2)/n2`
- ANOVA test podle skupin: `OneWayANOVATest([12, 10, 9, 10, 11, 9], [10, 12, 11, 12, 14, 13], [19, 18, 16, 16, 17, 15])`

Revision #18

Created 24 May 2023 13:19:41 by Sceptri

Updated 24 May 2023 21:30:48 by Sceptri