Conoscenze per passare la parte in R

1) Il primo punto consiste sempre nel prendere la popolazione e un campione casuale di essa.

Comandi:

`setwd()` ← per settare la cartella di lavoro; `read.table(Nome File.txt)` ← per caricare i dati della popolazione

`set.seed(Numero Matricola)` ← per generare i numeri random; `sample(Intervallo Ex: 100:110, Numero di estrazioni)`
↑ per prendere Numero di estrazioni numeri dall'intervallo

Infine per prendere il campione dalla popolazione si fa:

```
campione <- popolazione[sample(nrow(popolazione), n), ]
```

2) Ora ogni prova ha un ordine diverso ma tutto sommato ci sono sempre le stesse cose.

- Se viene richiesto di creare delle classi bisogna creare prima un array `c(c0, c1, c2, c3, ..., cn)` delle classi e poi usare `cut(campione$var, breaks= Array sopra, include.lowest = TRUE o FALSE, ordered= T o F)` oppure si può fare un hist con i breaks giusti.
- La **mediana** in un caso senza classi è semplice da trovare (basta usare `median()`), però se si ha una divisione in classi bisogna usare la Formula $\frac{\frac{1}{2} - F_{i^*-1}}{h_{i^*}} + c_{i^*-1}$ dove i^* rappresenta la classe a cui appartiene la mediana. Quindi da codice dobbiamo farte `table(campione diviso in classi)/n`; per frequenze relative fare $F =$ `cumsum(tabella di prima)` poi trovare $i^* =$ `min(which(F >0.5))` e infine tramite hist trovo $h_{i^*} =$ `hist$density[i*]`
- Come la mediana la **Media** nel caso senza classi basta fare `mean()` nel caso delle classi bisognerà prima fare un istogramma con `hist(campione, breaks=c(c0,c1,...,cn))` poi prendere i mids (`hist$mids`) e i count (`hist$count`) mids = `rep(hist$mids, hist$counts)` e poi fare la media quindi media = `mean(mids)`
- Anche la **Varianza** nel caso base ha la sua funzione `var()` mentre nel caso della divisione in classi riprendiamo la variabile mids come sopra e si fa `var(mids)` $\times ((n-1)/n)$ e troviamo la varianza distorta.
- Per i **Quantili** la stessa cosa, c'è la funzione `quantile()` ma nel caso della distribuzione in classi dobbiamo fare in un altro modo. Per iniziare diciamo che dobbiamo trovare il 60esimo quantile $(x=0.6)$ ⇒ riutilizzo cose usate sopra: $i^* =$ `min(which(F>= x))` allora $q = (x - F[i^*-1])/h_{i^*} +$ `ArrayClassi[i*]`
- **Valore della Funzione di ripartizione empirica** nel caso base basterà fare `ecdf(campione)(x)` mentre nel caso delle classi prendiamo $j^* =$ `max(which(ArrClassi <= x))` $c^* =$ `ArrClassi[j*]`
$h^* =$ `hist$density[j*]` ⇒ Trovo $F_a = F[j^*-1] + (x - c^*) \times h^*$

- Dipendenza in base al χ^2 ovviamente viene verificata dipendenza di 2 variabili quindi per prima cosa creo una tabella con i due dati: table(campione$lung, campione$colore) = t prendo il χ^2 con metodo summary(t)$statistic = χ^2 poi calcolo χ^2_{max} = min(dim(t)-1) × n e analizzo $\frac{\chi^2}{\chi^2_{max}}$ più è piccolo e più sono indipendenti, più è grande e più sono dipendenti.
- Regressione lineare: si usa per vedere se 2 variabili sono correlate, una delle 2 dipenderà dall'altra ora diciamo che V_3 dipenda da V_1 ⇒ dovremo fare lm(V_3 ~ V_1, campione) ora dobbiamo vedere se sono correlate tramite R^2 e la distr. dei residui summary(lm) e vediamo R^2 plot(campione$V1, lm$residuals) abline(h=0) e guardo come sono i residui, devono essere distribuiti casualmente. Possiamo anche fare delle predizioni: con predict(lm, data.frame(V1=10)) Faccio una predizione di V_3 quando V_1=10.
- Cambiamento di scala su Regr. lin. possiamo fare dei cambi di scala quando R^2 è basso e si fa: lm(log(V_3) ~ V_1, campione) in caso di esponenziale, e ricordarsi che poi la predizione va fatta: exp(predict(lm, data.frame(V1=10))). Altro cambio di scala è con la radice lm(V_3^(1/2) ~ V_1, campione) e la predizione diventa: (predict(lm, data.frame(V1=10)))^2

Oss: $R^2 > 0.7$ ⇒ Accettabile

- Intervallo di confidenza Intervallo di confidenza al 90% per la media di V_1 t.test(campione$V1, conf.level = 0.9)

Intervallo di confidenza 90% di $V_1 > 0.6$ ⇒ X = sum(V1>0.6)

binom.test(X, n, conf.level = 0.9)

↳ $conf.int |> as.numeric();

- Livello di Significatività:

Test conoscenze pre-esame 15/01/2025 ore 8:30

Mediana in classi cut(), hist() F <- table(cut)/n F <- cumsum(F)
i-star <- min(which(F > 0.5)) $m = \left(\frac{1}{2} - F[i\text{-star}-1]\right) / hist\$density[i\text{-star}] + c[i\text{-star}-1]$

Per quantile stessa cosa solo che X = quantile i-star = min(which(F > x))
$q = (X - F[i\text{-star}-1]) / hist\$density[i\text{-star}] + c[i\text{-star}]$

Media hist() mean(rep(hist$mids, hist$counts))

Varianza -> Var(rep*) x $\left(\frac{n-1}{n}\right)$ F.Emp. di X J* <- max(which(Classi > x)) c* <- Classi[J*] FE = F[J*-1] + (x - c*) x h[J*]

Indip table(var1, var2) summary(table)$statistic max(which(dim(table)-1) x n
poi divido

opp con test chisq.test(table)$ p.value > 0.05 ? => Indip

Regr. lin

lm(a ~ b, campione) summary(lm) vedo R^2 > 0.7 è OK

plot(campioneb, lmresiduals) guardo come sono distribuiti i residui
abline(h=0)

predict(lm, data.frame(b=150))

Intervallo conf t.test(campione, conf.level = %) $ conf.int

Intervallo di conf al 90% che $V_1 > 0.6$

X <- sum(V_1 > 0.6)

binom.test(X, n, conf.level=0.9) $ conf.int

liv. sign. X = sum(V_1 > 10) quando viene richiesta prob

binom.test(X, n, p=0.5, alternative = "less") $ p.value > liv.sign ⇒ accetto H_0

semonò accetto H_1

Se viene chiesto solo di verificare val. atteso ≥ 0.5 con liv.sig 10%

⇒ t.test(campione, mu = 0.5, alternative = "less") $ p.value < 0.1 ?

⇒ rifiuto H_0

semonò Accetto H_0

Ok queste sono le cose che ho imparato con questi app.