

Statistiques Appliquées

TD 4

Calculs et simulations avec

4.1 Intelligence

Les tests d'intelligence (QI) sont conçus de telle façon que les résultats suivent une distribution gaussienne $N(100, 10)$.

Calculer en utilisant R (famille de fonctions `(d,p,q,r)norm`) :

- Quelle proportion de la population obtient un résultat inférieur à 105.
- Quelle proportion de la population obtient un résultat entre 95 et 105.
- Quelles valeurs de QI, symétriques autour de la moyenne, couvrent 90% de la population.
- Que représente un résultat égal à 120?

Vérifier en utilisant les tables de la normale standard.

4.2 Écart-type et déciles de la normale

Les figures produites par le script de la v.a. uniforme contiennent plus d'informations (écart-type autour de l'espérance et déciles) que celles produites par le script de la v.a. normale.

Modifier le deuxième script de telle façon qu'il produise des figures équivalentes à celles du premier.

Script loi uniforme

Script loi normale



4.3 Un grand nombre d'échantillons

L'application **CenLimit** (à télécharger et à jouer avec) prélève un très grand nombre d'échantillons de taille n à partir d'une population de distribution $p_X(x)$ donnée, calcule la valeur de la moyenne (v.a. \bar{X}) de chaque échantillon et trace un histogramme avec toutes les valeurs \bar{x} ainsi obtenues.

On propose ici de faire ces mêmes calculs avec R en allant un peu plus loin : calculer aussi la variance de l'échantillon.

- Générer une population de $N = 10^6$ individus suivant la loi normale (`rnorm`, paramètres de votre choix).
- Créer un vecteur vide pour stocker les moyennes et un autre pour les variances, les deux de longueur $k = 100$ (le nombre d'échantillons qu'on va prélever), avec les commandes :

```
x.bar.vec = array( 0, dim=k )  
s2.vec = array( 0, dim=k )
```

- c. Dans une boucle `for` de k itérations, prélever un échantillon de taille $n = 10$ de la population.
- d. Calculer la moyenne \bar{x} et la variance s^2 de l'échantillon et stocker les résultats dans les vecteurs `x.bar.vec` et `s2.vec` déjà définis.
- e. Après la fin des itérations, générer un histogramme avec les valeurs recueillies dans `x.bar.vec` et un autre avec celles du `s2.vec`. (On peut choisir de montrer les *fréquences* (effectifs de chaque classe) ou les *densités* (fréquences relatives de chaque classe divisées par la largeur de la classe, équivalentes à une densité de probabilité si le nombre d'observations est très élevé) avec l'option `freq=TRUE/FALSE` de la fonction `hist`.)
- f. Essayer d'autres valeurs pour N, n, k .
- g. Essayer avec une autre population, p.ex. uniforme.
- h. Lancer plusieurs simulations avec les mêmes paramètres et observer les résultats.
- i. Est-ce qu'on retrouve les mêmes phénomènes qu'avec `CenLimit` ?
- j. Remarques, conclusions ...

Remarque 1 On peut exporter en pdf en remplaçant `dev.copy2eps(...)` par `dev.print(pdf, ...)`.