Devoir 1

Méthodes d'analyse des données (STT-7330)

Nicolas Corneau-Tremblay et Ève Paquette-Bigras

1/30/2018

Question 1

(a)

La collecte de données est effectuée par sondage sur un échantillon aléatoire à plusieurs degrés (trad. de *multi-stage random sampling*). C'est une méthode probabiliste. La sélection aléatoire est effectuée sur les unités, soit les écoles, et non sur les individus.

L'échantillon et une partie des données sont tirés de l'étude longitudinale *Monitoring the Future* (consulter le site http://www.monitoringthefuture.org/ pour plus de détails sur l'étude MTF). Les données concernant les étudiants ont été collectées par sondage autoadministré complété par les étudiants lors de l'étude MTF. Les données concernant les caractéristiques et les politiques de chaque école ont été collectées auprès des gestionnaires en parallèle de l'étude MTF.

(b)

Les données ne permettent pas de conclure que tester la drogue augmente la consommation de drogue, entre autres parce que la méthodologie de l'étude ne permet pas de démontrer une relation de causalité. Notamment, il est possible que les écoles ayant des problématiques de consommation de drogues parmi ses élèves aient instauré des tests afin de détecter la consommation. Dans ce cas, les tests n'augmentent pas la consommation de drogues, mais ont été instaurés dans les lieux où la consommation est la plus importante. Une façon d'identifier correctement l'effet causal serait de procéder à une expérience.

Question 2

(a)

L'étude 2 permet ce genre de conclusion. Puisqu'elle se base sur une expérience randomisée, on peut de conclure que la possession d'un chien est la cause de la différence de pression artérielle observée *pour les personnes qui ont participé à l'étude*. En effet, si la possession d'un chien est attribuée entre des propriétaires potentiels de façon aléatoire, la randomisation de permet de s'assurer que les propriétaires de chien et les non-propriétaires sont en moyenne similaires dans leurs caractéristiques observables et inobservables. Étant en moyenne similaire en tous points, outre le fait de posséder ou non un chien, la différence de pression artérielle observée entre les propriétaires et les non-propriétaires ne peut être attribuée qu'à une chose : la possession d'un chien.

L'étude 1 ne permet pas de conclure que la possession d'un chien est la cause de la différence de pression artérielle observée. Elle permet plutôt de conclure, avec un degré d'incertitude plus ou moins important selon la taille de la population, que la pression artérielle des résidents qui possèdent un chien est plus basse que la pression artérielle des résidents qui ne possèdent pas un chien.

(b)

Les résultats obtenus lors de l'étude 2 sont de nature exploratoire et ne peuvent être généralisés. Les participants de l'étude étaient des volontaires. Or, il est probable que les volontaires pour une étude portant sur les effets bénéfiques pour la santé de la possession d'un chien soient enclins à apprécier la compagnie des animaux. Cette appréciation n'est pas forcément répandue dans toute la population des habitants de la région, il est donc possible de s'attendre à des résultats différents pour les habitants non volontaires.

Les résultats obtenus lors de de l'étude 1 peuvent être généralisés, mais ils ne prouvent aucune causalité. Toutefois, le croisement des résultats des deux études peut justifier que soit réalisée une étude plus exhaustive sur une relation de causalité possible entre la pression artérielle et la possession d'un animal de compagnie dans toute la population.

Question 3

```r
# Importation des packages nécessaires
library(knitr)
library(MASS)

# Nombre d'itérations
iter = 500
obs = 250

# Dataframes de stockage
res <- data.frame("mean_y1" = rep(0, iter),
                  "mean_y2" = rep(0, iter),
                  "var_y1" = rep(0, iter),
                  "var_y2" = rep(0, iter),
                  "cor_y1y2" = rep(0, iter))

# Listes de stockage
mcar <- list(res, res, res, res)
mar <- list(res, res, res, res)
nmar <- list(res, res, res, res)

# Définition d'une fonction calculant les 2 moyennes, les 2 variances et la corrélation
mat_stats <- function(mat_input){

  # Vecteur de stockage
  output <- rep(0, 5)

  # Moyenne y1
  output[1] <- mean(mat_input[, 1])
  # Moyenne y2
  output[2] <- mean(mat_input[, 2])
  # Variance y1
  output[3] <- var(mat_input[, 1])
  # Variance y2
  output[4] <- var(mat_input[, 2])
  # CorrÃ©lation y1, y2
  output[5] <-cor(mat_input[, 1], mat_input[, 2])

  return(output)
```

```r
}

# Définition d'une fonction effectuant les imputations
imp_stats <- function(mat_input){

  # Liste de stockage
  output <- list(0)

  # CCA
  temp <- mat_input[complete.cases(mat_input), ]
  output[[1]] <- mat_stats(temp)

  # Imputation par la moyenne
  temp <- mat_input
  temp[, 2] <- ifelse(is.na(temp[, 2]), mean(temp[, 2], na.rm = TRUE), temp[, 2])
  output[[2]] <- mat_stats(temp)

  # Imputation simple par régression linéaire
  temp <- mat_input
  lm_res <- lm(temp[, 2] ~ temp[, 1], na.action = na.omit)
  predictions <- predict(lm_res, as.data.frame(temp[, 1]))
  temp[, 2] <- ifelse(is.na(temp[, 2]), predictions, temp[, 2])
  output[[3]]  <- mat_stats(temp)

  # Imputation simple par régression linéaire, stochastique
  temp <- mat_input
  lm_res <- lm(temp[, 2] ~ temp[, 1], na.action = na.omit)
  predictions <- predict(lm_res, as.data.frame(temp[, 1]))
  resid_sd <- (summary(lm_res)$sigma)
  temp[, 2] <- ifelse(is.na(temp[, 2]), predictions + rnorm(sum(is.na(temp[, 2])), 0, resid_sd), temp[,
  output[[4]] <- mat_stats(temp)

  return(output)
}

# Construction de la matrice de covariance
cov_mat <- matrix(c(169, 19.5, 19.5, 9), nrow = 2, ncol = 2)

# Initialisation du seed
set.seed(222)

# Boucle permettant d'effectuer les simulations
for(i in 1:iter){
  mat <- mvrnorm(n = obs, mu = c(100, 12), Sigma = cov_mat, empirical = TRUE)

  # MCAR
  mat_mcar <- mat
  select <- runif(obs)
  mat_mcar[, 2] <- ifelse(select > median(select), NA, mat_mcar[, 2])

  mcar_output <- imp_stats(mat_mcar)
  mcar[[1]][i, ] <- mcar_output[[1]]
  mcar[[2]][i, ] <- mcar_output[[2]]
```

```
  mcar[[3]][i, ] <- mcar_output[[3]]
  mcar[[4]][i, ] <- mcar_output[[4]]

  # MAR
  mat_mar <- mat
  mat_mar[, 2] <- ifelse(mat_mar[, 1] < median(mat_mar[, 1]), NA, mat_mar[, 2])

  mar_output <- imp_stats(mat_mar)
  mar[[1]][i, ] <- mar_output[[1]]
  mar[[2]][i, ] <- mar_output[[2]]
  mar[[3]][i, ] <- mar_output[[3]]
  mar[[4]][i, ] <- mar_output[[4]]

  # NMAR
  mat_nmar <- mat
  mat_nmar[, 2] <- ifelse(mat_nmar[, 2] < median(mat_nmar[, 2], na.rm = TRUE), NA, mat_nmar[, 2])

  nmar_output <- imp_stats(mat_nmar)
  nmar[[1]][i, ] <- nmar_output[[1]]
  nmar[[2]][i, ] <- nmar_output[[2]]
  nmar[[3]][i, ] <- nmar_output[[3]]
  nmar[[4]][i, ] <- nmar_output[[4]]

}

# Présentation des résultats
names <- c("CCA", "Imp_moy", "Imp_reg", "Imp_reg_stoc")

mcar_df <- cbind(colMeans(mcar[[1]]),
                 colMeans(mcar[[2]]),
                 colMeans(mcar[[3]]),
                 colMeans(mcar[[4]]))
colnames(mcar_df) <- names
kable(round(mcar_df, 3), caption = "Résultats pour simulation avec MCAR (itér. = 500)")
```

Table 1: Résultats pour simulation avec MCAR (itér. = 500)

	CCA	Imp_moy	Imp_reg	Imp_reg_stoc
mean_y1	100.005	100.000	100.000	100.000
mean_y2	12.006	12.006	12.005	12.008
var_y1	168.853	169.000	169.000	169.000
var_y2	8.994	4.479	5.638	8.989
cor_y1y2	0.502	0.354	0.634	0.502

```
mar_df <- cbind(colMeans(mar[[1]]),
                 colMeans(mar[[2]]),
                 colMeans(mar[[3]]),
                 colMeans(mar[[4]]))
colnames(mar_df) <- names
kable(round(mar_df, 3), caption = "Résultats pour simulation avec MAR (itér. = 500)")
```

Table 2: Résultats pour simulation avec MAR (itér. = 500)

	CCA	Imp_moy	Imp_reg	Imp_reg_stoc
mean_y1	110.344	100.000	100.000	100.000
mean_y2	13.197	13.197	12.004	12.006
var_y1	61.159	169.000	169.000	169.000
var_y2	7.602	3.786	5.730	9.096
cor_y1y2	0.326	0.138	0.621	0.495

```
nmar_df <- cbind(colMeans(nmar[[1]]),
                 colMeans(nmar[[2]]),
                 colMeans(nmar[[3]]),
                 colMeans(nmar[[4]]))
colnames(nmar_df) <- names
kable(round(nmar_df, 3), caption = "Résultats pour simulation avec NMAR (itér. = 500)")
```

Table 3: Résultats pour simulation avec NMAR (itér. = 500)

	CCA	Imp_moy	Imp_reg	Imp_reg_stoc
mean_y1	105.165	100.000	100.000	100.000
mean_y2	14.389	14.389	14.135	14.134
var_y1	141.927	169.000	169.000	169.000
var_y2	3.292	1.639	1.894	3.365
cor_y1y2	0.327	0.212	0.468	0.352

Les tables ci-haut présentent les résultats provenant de 500 simulations pour quatre méthodes d'imputation dans trois contextes de valeurs manquantes différents. Les quatre méthodes d'imputation sont l'analyse de cas complets (*CCA*), l'imputation par la moyenne (*Imp_moy*), l'imputation par régression (*Imp_reg*) et l'imputation par régression stochastique (*Imp_reg_stoc*). Les trois contextes de valeurs manquantes sont les valeurs manquantes complètement aléatoires (*MCAR*), les valeurs manquantes aléatoires (*MAR*) et les valeurs manquantes non aléatoires (*NMCAR*). Ces simulations ont été réalisées sur un jeu de données contenant deux variables de moyenne 100 et 12 respectivement, de variance 169 et 9 respectivement, et de covariance 19,5.

Comme les tables en témoignent, toutes les méthodes d'imputation permettent une estimation sans biais de la moyenne dans le cas des MCAR. Dans les cas MAR et NMAR, la méthode de l'analyse des cas complets biaise l'estimation de la moyenne et de la variance. Notons de plus que la variance est sous-estimée avec la méthode d'imputation par la moyenne et celle d'imputation par régression simple, et la covariance est également biaisée. On constate également que la méthode d'imputation par régression non stochastique surestime la corrélation entre les variables même en contexte MCAR.

En résumé, en contexte MCAR, l'estimation la moins biaisée est produite par la méthode d'imputation par analyse des cas complets, suivie de près par la méthode par régression stochastique. En contexte MAR et NMAR, l'estimation la moins biaisée est produite par la méthode d'imputation par régression stochastique.

Question 4

```
library(ggplot2)

iter2 = 500
obs2 = 250
```

```r
# Définition d'un seed
set.seed(222)

# Vecteur de stockage
moy_y2 <- rep(0, iter2)
var_moy_y2 <- rep(0, iter2)

# Boucle permettant d'effectuer les simulations
for(i in 1:iter2){
  mat <- mvrnorm(n = obs2, mu = c(100, 12), Sigma = cov_mat, empirical = TRUE)

  # MCAR
  mat_mcar <- mat
  select <- runif(obs2)
  mat_mcar[, 2] <- ifelse(select > median(select), NA, mat_mcar[, 2])

  # Imputation simple par régression linéaire stochastique
  lm_res <- lm(mat_mcar[, 2] ~ mat_mcar[, 1], na.action = na.omit)
  predictions <- predict(lm_res, as.data.frame(mat_mcar[, 1]))
  resid_sd <- (summary(lm_res)$sigma)
  mat_mcar[, 2] <- ifelse(is.na(mat_mcar[, 2]),
                          predictions + rnorm(sum(is.na(mat_mcar[, 2])), 0, resid_sd),
                          mat_mcar[, 2])

  # Nombre d'observation
  n <- obs2
  # Moyenne y2
   moy_y2[i]<- mean(mat_mcar[, 2])
  # Variance y2
  var_temp <- var(mat_mcar[, 2])
  # Variance de la moyenne de y2
  var_moy_y2[i] <- (var_temp / n)

}

# Moyenne des variances de l'estimateur de la moyenne
print(mean(var_moy_y2))
```

```
## [1] 0.03563837
```

```r
# Variance des estimateurs de la moyenne
print(var(moy_y2))
```

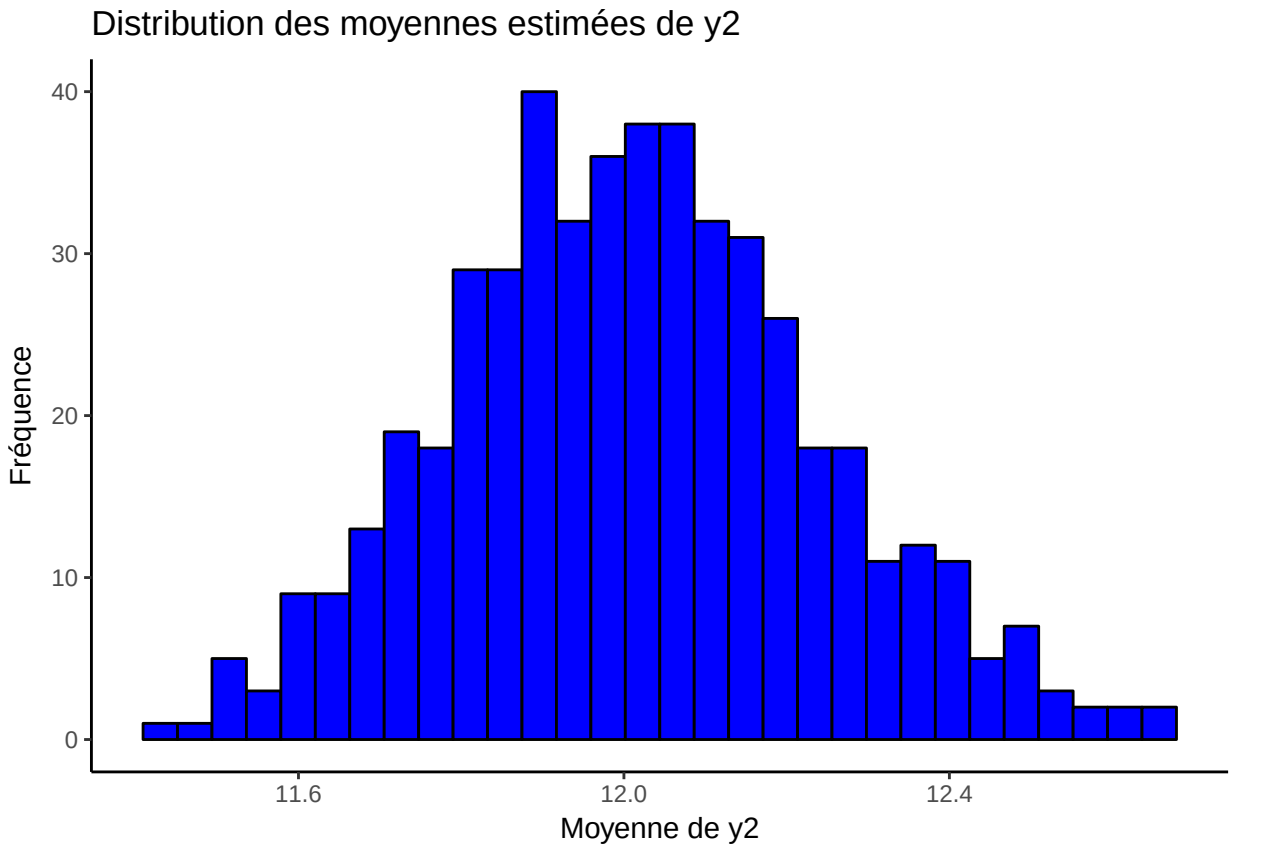
```
## [1] 0.05109115
```

```r
# Biais relatif
print((mean(var_moy_y2) - var(moy_y2)) / var(moy_y2))
```

```
## [1] -0.3024552
```

```r
# Distribution
ggplot() +
  geom_histogram(aes(moy_y2), color = "black", fill = "blue") +
  xlab("Moyenne de y2") +
  ylab("Fréquence") +
```

```
  labs(title = "Distribution des moyennes estimées de y2") +
  theme_classic()
```



Comme il est possible de le constater, suite à l'imputation à l'aide de la régression stochastique, la variance des 500 moyennes estimées est supérieure à la moyenne des 500 variances des moyennes estimées de y2. Le biais relatif est d'environ 30%. Ceci suggère que l'estimation de la variance de la moyenne suite à la régression stochastique sous-estime celle-ci.