CSI_Assignment1_2024_Rcode.R

xiaoyuwan

2025-03-25

```
##########################################################
## CSI 2024 Major Assignment 1 R code
##########################################################

##############
# Question 1 #
##############
library(table1)
```

```
## Warning: package 'table1' was built under R version 4.4.3

##
## Attaching package: 'table1'

## The following objects are masked from 'package:base':
##
##     units, units<-
```

```
library(EValue)
```

```
## Warning: package 'EValue' was built under R version 4.4.3
```

```
setwd("C:/Users/xiaoyuwan/OneDrive/unimelb/causal inference/csi assignment1")
rm(list=ls())

#Read in the datasets:
data01 <- read.csv( file="CSI2024_data01.csv")
data02 <- read.csv( file="CSI2024_data02.csv")
data03 <- read.csv( file="CSI2024_data03.csv")
data04 <- read.csv( file="CSI2024_data04.csv")

#a) ACE of A on Y
#without adjusting for Z:
reg01_data01 <- lm(Y~A, data=data01)
summary(reg01_data01)
```

```
##
## Call:
## lm(formula = Y ~ A, data = data01)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6923 -0.9905  0.0144  0.9937  4.1497
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.05862    0.06378   0.919    0.358
## A            1.14105    0.06700  17.030   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.426 on 498 degrees of freedom
## Multiple R-squared:  0.368,  Adjusted R-squared:  0.3668
## F-statistic:   290 on 1 and 498 DF,  p-value: < 2.2e-16
```

```r
confint(reg01_data01, "A")
```

```
##      2.5 %   97.5 %
## A 1.009405 1.272693
```

```r
reg01_data02 <- lm(Y~A, data=data02)
summary(reg01_data02)
```

```
##
## Call:
## lm(formula = Y ~ A, data = data02)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2828 -0.5859 -0.0078  0.6466  2.6152
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0002959  0.0461140   0.006    0.995
## A           0.9840431  0.0459576  21.412   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.03 on 498 degrees of freedom
## Multiple R-squared:  0.4793, Adjusted R-squared:  0.4783
## F-statistic: 458.5 on 1 and 498 DF,  p-value: < 2.2e-16
```

```r
confint(reg01_data02, "A")
```

```
##       2.5 %   97.5 %
## A 0.8937484 1.074338
```

```r
reg01_data03 <- lm(Y~A, data=data03)
summary(reg01_data03)
```

```
##
## Call:
## lm(formula = Y ~ A, data = data03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9497 -0.7890  0.0211  0.7361  4.3441
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03053    0.05511   0.554     0.58
## A            1.07494    0.03982  26.997   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.232 on 498 degrees of freedom
## Multiple R-squared:  0.5941, Adjusted R-squared:  0.5933
## F-statistic: 728.8 on 1 and 498 DF,  p-value: < 2.2e-16
```

```r
confint(reg01_data03, "A")
```

```
##       2.5 %   97.5 %
## A 0.9967099 1.153171
```

```r
reg01_data04 <- lm(Y~A, data=data04)
summary(reg01_data04)
```

```
##
## Call:
## lm(formula = Y ~ A, data = data04)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5013 -0.8628  0.0217  0.9060  4.0483
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.04796    0.06173  -0.777    0.438
## A            0.97860    0.04471  21.886   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.379 on 498 degrees of freedom
## Multiple R-squared:  0.4903, Adjusted R-squared:  0.4893
## F-statistic:   479 on 1 and 498 DF,  p-value: < 2.2e-16
```

```r
confint(reg01_data04, "A")
```

```
##       2.5 %   97.5 %
## A 0.8907549 1.066454
```

```r
#adjusting for Z:
reg02_data01 <- lm(Y~A + Z, data=data01)
summary(reg02_data01)
```

```
##
## Call:
## lm(formula = Y ~ A + Z, data = data01)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.69825 -0.66583 -0.00351  0.70731  2.94297
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.056623   0.047640   1.189    0.235
## A           -0.003441   0.076259  -0.045    0.964
## Z            0.980849   0.049312  19.891   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.065 on 497 degrees of freedom
## Multiple R-squared:  0.6481, Adjusted R-squared:  0.6467
## F-statistic: 457.7 on 2 and 497 DF,  p-value: < 2.2e-16
```

```r
confint(reg02_data01, "A")
```

```
##        2.5 %    97.5 %
## A -0.1532712 0.1463884
```

```r
reg02_data02 <- lm(Y~A + Z, data=data02)
summary(reg02_data02)
```

```
##
## Call:
## lm(formula = Y ~ A + Z, data = data02)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.80769 -0.52599 -0.02555  0.53967  2.14424
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.01068    0.03590  -0.298    0.766
## A            0.41509    0.04771   8.701   <2e-16 ***
## Z            0.47476    0.02634  18.026   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.802 on 497 degrees of freedom
## Multiple R-squared:  0.6852, Adjusted R-squared:  0.6839
## F-statistic: 540.8 on 2 and 497 DF,  p-value: < 2.2e-16
```

```r
confint(reg02_data02, "A")
```

```
##       2.5 %    97.5 %
## A 0.3213629 0.5088218
```

```r
reg02_data03 <- lm(Y~A + Z, data=data03)
summary(reg02_data03)
```

```
##
## Call:
## lm(formula = Y ~ A + Z, data = data03)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.87815 -0.67835 -0.00354  0.64142  2.90666
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.02029    0.04455   0.455    0.649
## A            0.56193    0.04504  12.476   <2e-16 ***
## Z            0.98020    0.06020  16.283   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9961 on 497 degrees of freedom
## Multiple R-squared:  0.7353, Adjusted R-squared:  0.7342
## F-statistic: 690.3 on 2 and 497 DF,  p-value: < 2.2e-16
```

```r
confint(reg02_data03, "A")
```

```
##       2.5 %    97.5 %
## A 0.4734381 0.6504207
```

```r
reg02_data04 <- lm(Y~A + Z, data=data04)
summary(reg02_data04)
```

```
##
## Call:
## lm(formula = Y ~ A + Z, data = data04)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5645 -0.8523  0.0395  0.8976  3.9390
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03565    0.06173  -0.578   0.5638
## A            0.88372    0.06137  14.401   <2e-16 ***
## Z            0.02368    0.01053   2.247   0.0251 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.373 on 497 degrees of freedom
## Multiple R-squared:  0.4954, Adjusted R-squared:  0.4934
## F-statistic:   244 on 2 and 497 DF,  p-value: < 2.2e-16
```

```
confint(reg02_data04, "A")
```

```
##       2.5 %   97.5 %
## A 0.7631485 1.004284
```

```
####################################################################
# Question 2

require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(survey)
```

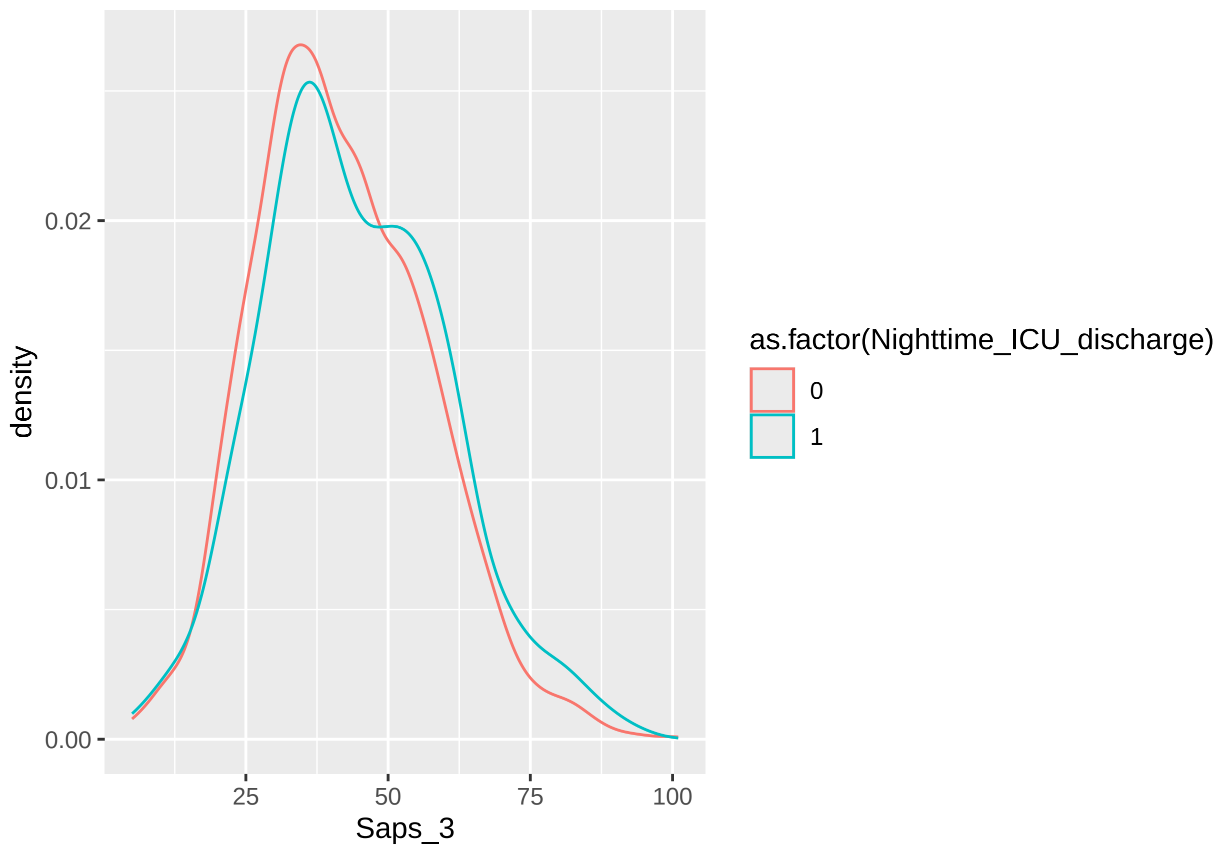
```
## Loading required package: survey

## Loading required package: grid

## Loading required package: Matrix

## Loading required package: survival

## Warning: package 'survival' was built under R version 4.4.1

##
## Attaching package: 'survey'

## The following object is masked from 'package:EValue':
##
##     HR

## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
require(EValue)
require(table1)

#Read in the data
nighticu <- read.csv("NighttimeICU.csv")

#Look at the means/proportions in each exposure group
table1(~ Saps_3 + factor(Reason_index_ICU_admisssion)  + factor(Admission_Source) +
         factor(Systemic_hypertension) + factor(Diabetes_mellitus) +
         factor(Cancer) + factor(Congestive_heart_failure) + factor(COPD) +
         factor(Chronic_Kidney_disease) + factor(Liver_cirrhosis) | Nighttime_ICU_discharge,
       data=nighticu)
```

```
## Warning in table1.formula(~Saps_3 + factor(Reason_index_ICU_admisssion) + :
## Terms to the right of '|' in formula 'x' define table columns and are expected
## to be factors with meaningful labels.

## Get nicer 'table1' LaTeX output by simply installing the 'kableExtra' package
```

	0	1	Overall
	(N=3663)	(N=650)	(N=4313)
Saps_3			
Mean (SD)	41.9 (15.0)	44.2 (16.0)	42.2 (15.2)
Median [Min, Max]	40.0 [5.00, 101]	42.0 [5.00, 94.0]	41.0 [5.00, 101]
factor(Reason_index_ICU_admisssion)			
0	1744 (47.6%)	251 (38.6%)	1995 (46.3%)
1	1919 (52.4%)	399 (61.4%)	2318 (53.7%)
factor(Admission_Source)			
0	1278 (34.9%)	265 (40.8%)	1543 (35.8%)
1	308 (8.4%)	71 (10.9%)	379 (8.8%)
2	202 (5.5%)	34 (5.2%)	236 (5.5%)
3	1717 (46.9%)	245 (37.7%)	1962 (45.5%)
4	158 (4.3%)	35 (5.4%)	193 (4.5%)
factor(Systemic_hypertension)			
0	1660 (45.3%)	312 (48.0%)	1972 (45.7%)
1	2003 (54.7%)	338 (52.0%)	2341 (54.3%)
factor(Diabetes_mellitus)			
0	2570 (70.2%)	456 (70.2%)	3026 (70.2%)
1	1093 (29.8%)	194 (29.8%)	1287 (29.8%)
factor(Cancer)			
0	2851 (77.8%)	506 (77.8%)	3357 (77.8%)
1	812 (22.2%)	144 (22.2%)	956 (22.2%)
factor(Congestive_heart_failure)			
0	3292 (89.9%)	571 (87.8%)	3863 (89.6%)
1	371 (10.1%)	79 (12.2%)	450 (10.4%)
factor(COPD)			
0	3363 (91.8%)	592 (91.1%)	3955 (91.7%)
1	300 (8.2%)	58 (8.9%)	358 (8.3%)
factor(Chronic_Kidney_disease)			
0	3431 (93.7%)	615 (94.6%)	4046 (93.8%)
1	232 (6.3%)	35 (5.4%)	267 (6.2%)
factor(Liver_cirrhosis)			
0	3527 (96.3%)	614 (94.5%)	4141 (96.0%)
1	136 (3.7%)	36 (5.5%)	172 (4.0%)

```r
#Look at density of SAPS 3 across exposure groups
ggplot(nighticu, aes(Saps_3, colour=as.factor(Nighttime_ICU_discharge))) +
  geom_density()
```




```
#Calculate the standardised differences:
#Standardised differences
#Binary confounders:
#unweighted counts, percentages:
p0 = 100*sum(nighticu$Reason_index_ICU_admisssion[nighticu$Nighttime_ICU_discharge == 0]==1)/sum(!is.na
p1 = 100*sum(nighticu$Reason_index_ICU_admisssion[nighticu$Nighttime_ICU_discharge == 1]==1)/sum(!is.na
#standardised differences:
stdiff1 = ((p1 - p0)/sqrt(((p0/100)*(1-p0/100) + (p1/100)*(1-(p1/100)))/2))
print(paste("For admission reason standardised difference =", round(stdiff1, 2)))
```

```
## [1] "For admission reason standardised difference = 18.24"
```

```
#replace Reason_index_ICU_admisssion with other binary confounders in the above to get standardised dif
#for other binary confounders

#Categorical confounder: here we need to consider each level of the
#confounder: we'll consider Admission_Source
table(nighticu$Admission_Source, nighticu$Nighttime_ICU_discharge)
```

```
##
##         0    1
##   0  1278  265
##   1   308   71
##   2   202   34
##   3  1717  245
```

```
##    4  158   35
```

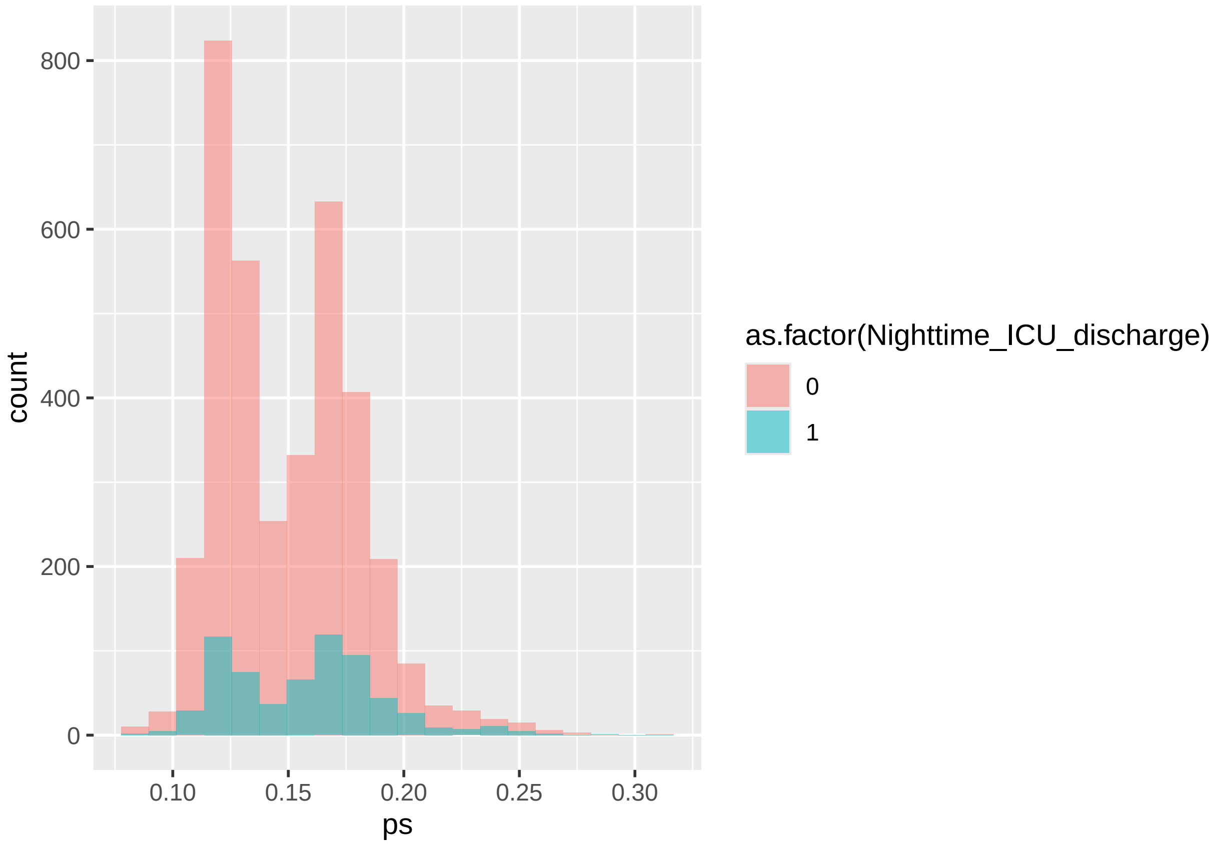
```
for(j in 0:4){
  p0 = 100*sum(nighticu$Admission_Source[nighticu$Nighttime_ICU_discharge == 0]==j)/sum(!is.na(nighticu$
  p1 = 100*sum(nighticu$Admission_Source[nighticu$Nighttime_ICU_discharge == 1]==j)/sum(!is.na(nighticu$
  #standardised differences:
  stdiff1 = ((p1 - p0)/sqrt(((p0/100)*(1-p0/100) + (p1/100)*(1-(p1/100)))/2))
  print(paste("For source=",j," standardised difference =", round(stdiff1, 2)))

}
```

```
## [1] "For source= 0  standardised difference = 12.15"
## [1] "For source= 1  standardised difference = 8.52"
## [1] "For source= 2  standardised difference = -1.26"
## [1] "For source= 3  standardised difference = -18.67"
## [1] "For source= 4  standardised difference = 4.99"
```

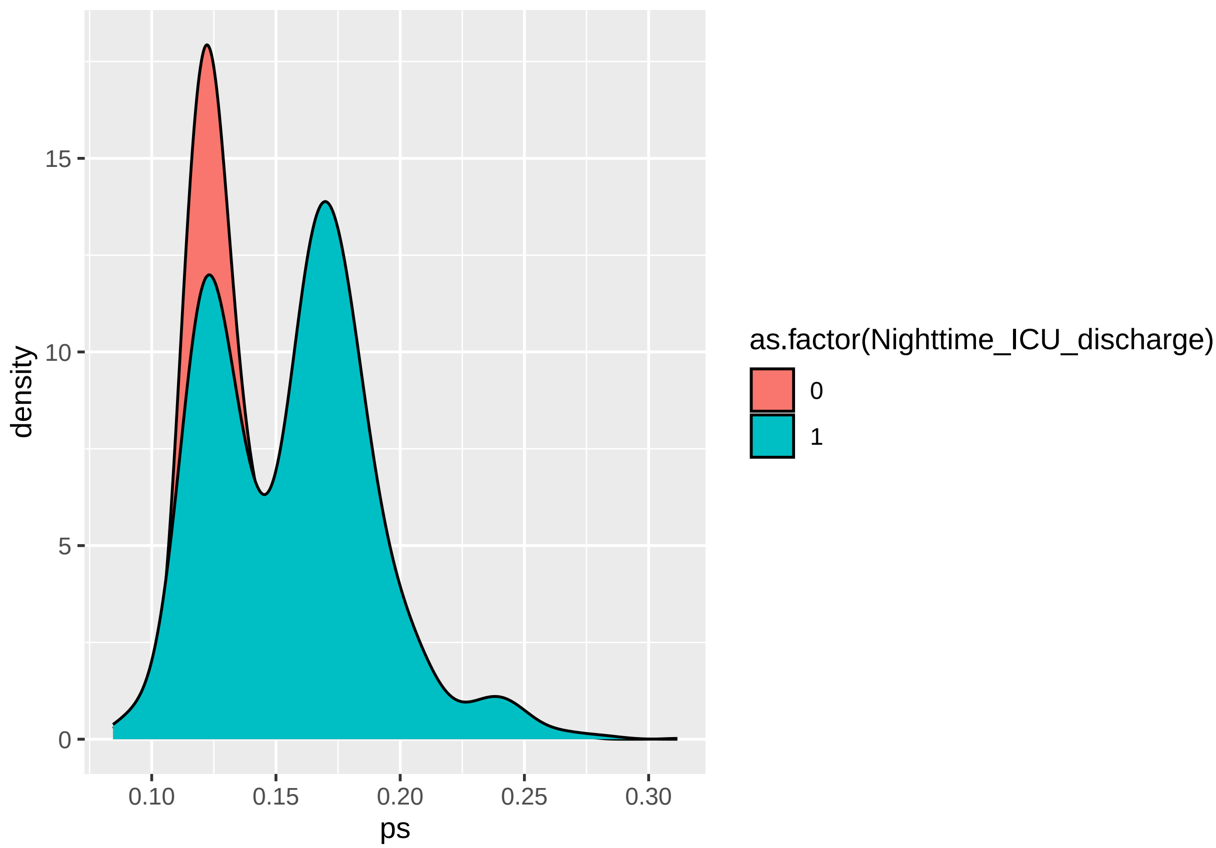
```
#Continuous confounder: Saps_3
#unweighted means, sds:
mean0 = mean(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge == 0])
var0 = var(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge == 0])
mean1 = mean(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge == 1])
var1 = var(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge == 1])
#standardised differences:
stdiff1= 100*((mean1 - mean0)/sqrt((var1 +var0)/2))
print(paste("For Saps 3 standardised difference =", round(stdiff1, 2)))
```

```
## [1] "For Saps 3 standardised difference = 15.25"
```

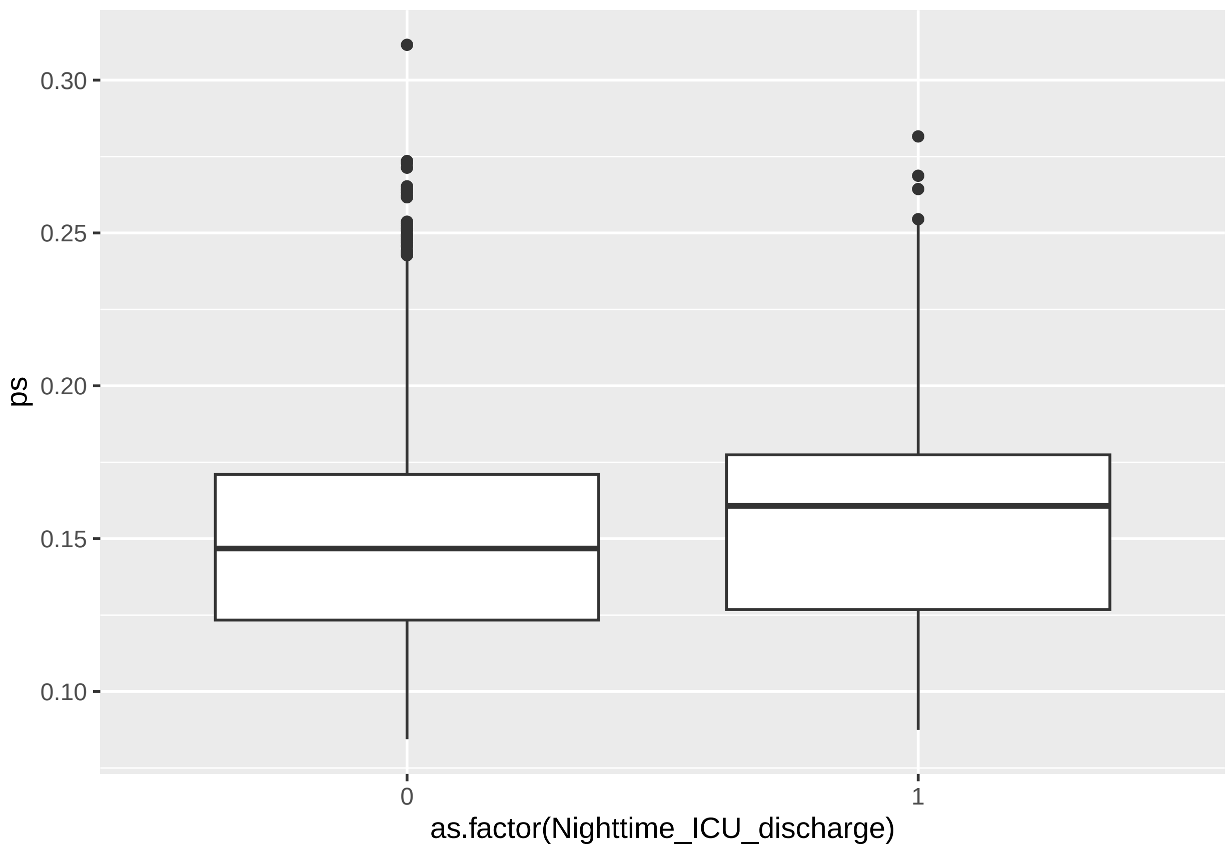
```
#Fit the propensity score model
propmodel <- glm(Nighttime_ICU_discharge ~ Saps_3 + as.factor(Reason_index_ICU_admisssion) +
                   as.factor(Admission_Source) + Systemic_hypertension+ Diabetes_mellitus +
                   Cancer + Congestive_heart_failure + COPD + Chronic_Kidney_disease +
                   Liver_cirrhosis,
                 family=binomial(logit), data=nighticu)
nighticu$ps <-  propmodel$fitted.values

#Take a look at the distribution of propensity scores

ggplot(nighticu, aes(x=ps, fill=as.factor(Nighttime_ICU_discharge))) +
  geom_histogram(bins=20, alpha=.5, position="identity")
```



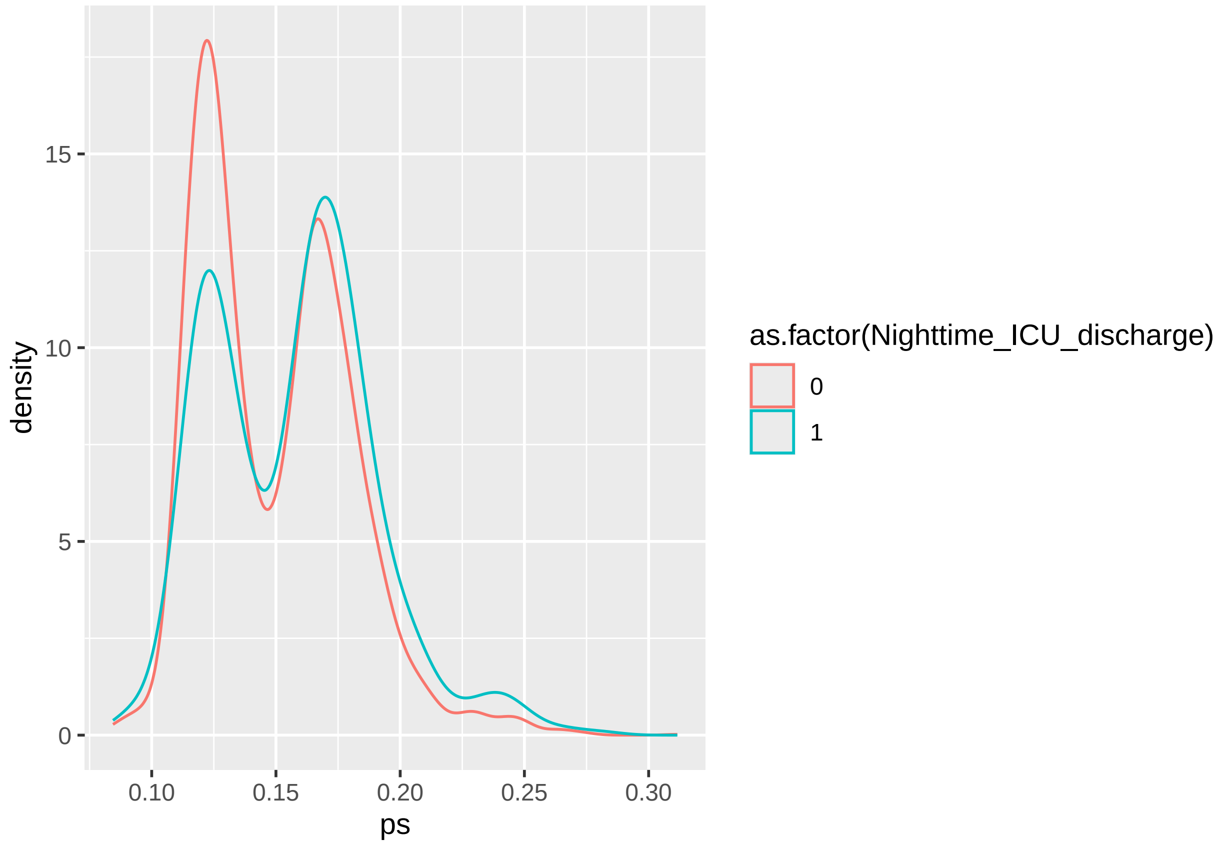

```r
ggplot(nighticu, aes(x=ps, fill=as.factor(Nighttime_ICU_discharge))) +
  geom_density()
```



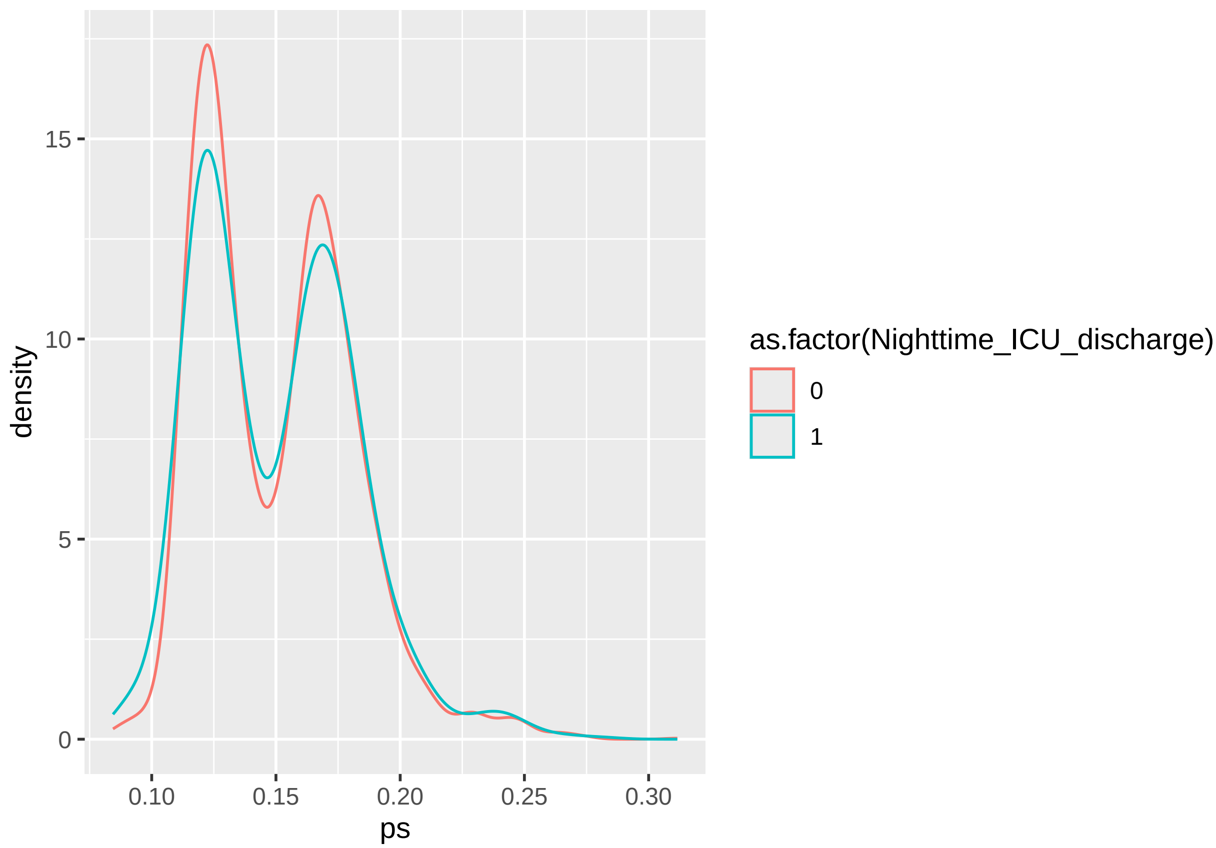

```r
ggplot(nighticu, aes(as.factor(Nighttime_ICU_discharge), ps))  +
  geom_boxplot()
```



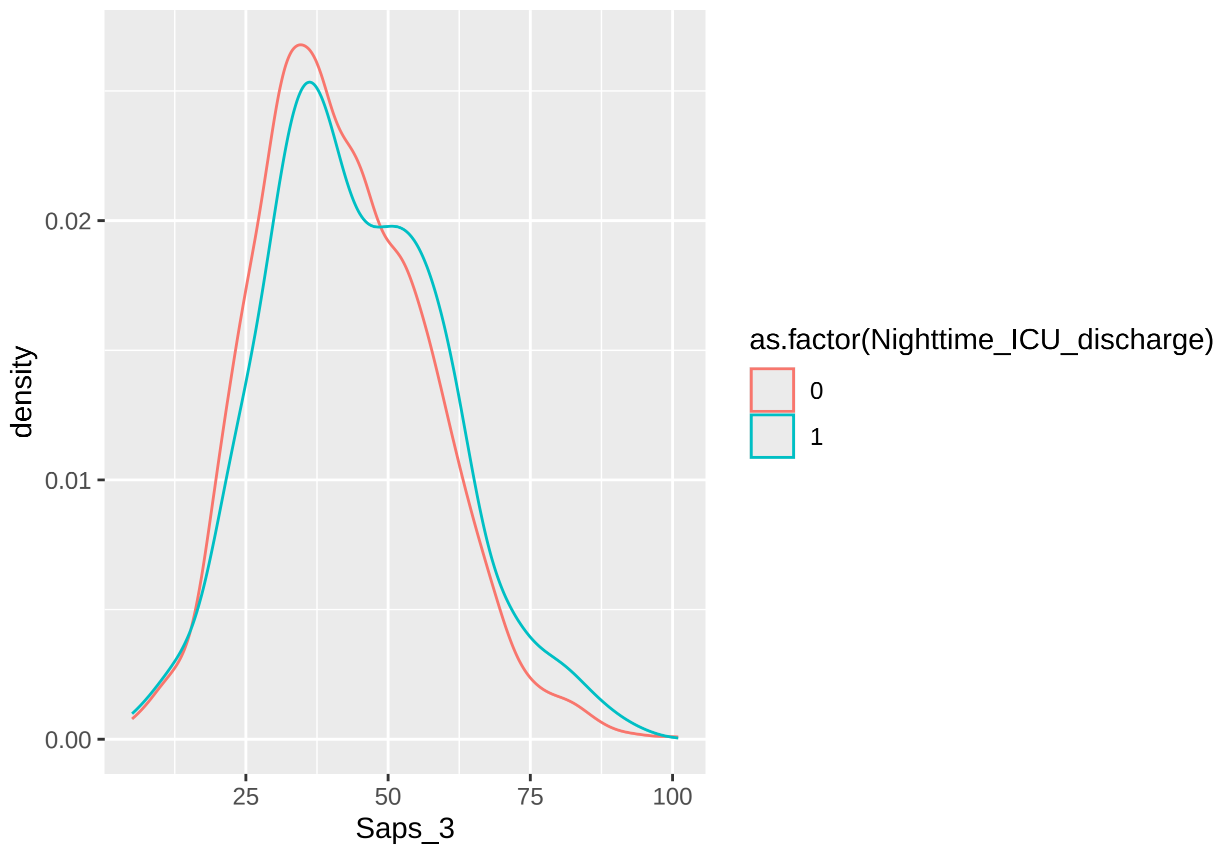

```r
#Look at the maximums and minimums in each exposure group
minps0 <- min(nighticu$ps[nighticu$Nighttime_ICU_discharge==0])
maxps0 <- max(nighticu$ps[nighticu$Nighttime_ICU_discharge==0])
minps1 <- min(nighticu$ps[nighticu$Nighttime_ICU_discharge==1])
maxps1 <- max(nighticu$ps[nighticu$Nighttime_ICU_discharge==1])

#Generate the IPWs:
nighticu$ipw[nighticu$Nighttime_ICU_discharge == 1] <-   1/(nighticu$ps[nighticu$Nighttime_ICU_discharg
nighticu$ipw[nighticu$Nighttime_ICU_discharge == 0] <-   1/(1-nighticu$ps[nighticu$Nighttime_ICU_discha

#Take a look at the plots
ggplot(nighticu,
       aes(ps, colour=as.factor(Nighttime_ICU_discharge))) + geom_density()
```



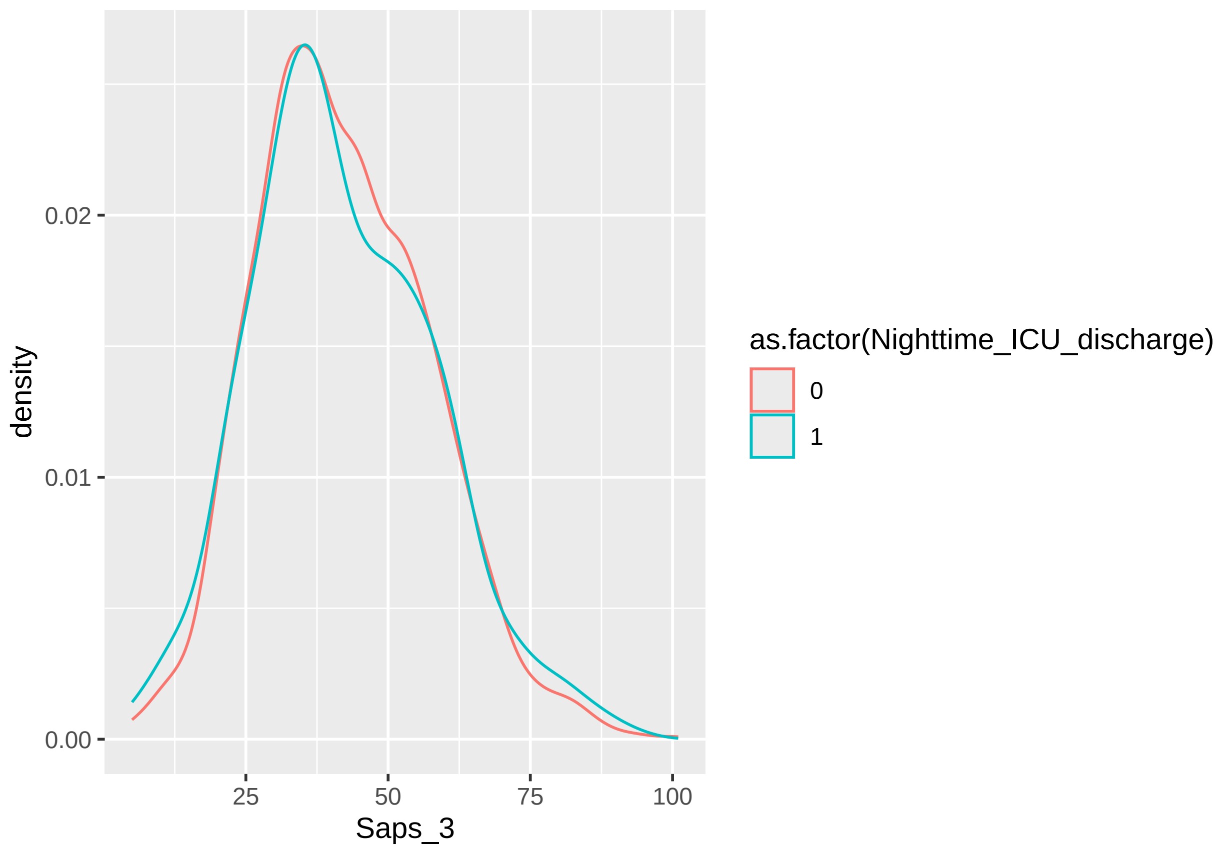

```
ggplot(nighticu,
       aes(ps, colour=as.factor(Nighttime_ICU_discharge),weight=ipw)) +
  geom_density()
```




```r
#Take a look at the plots: considering the distribution of Saps_3
#in unweighted and weighted spaces
ggplot(nighticu,
       aes(Saps_3, colour=as.factor(Nighttime_ICU_discharge))) +
  geom_density()
```




```
ggplot(nighticu,
       aes(Saps_3, colour=as.factor(Nighttime_ICU_discharge),weight=ipw)) +
  geom_density()
```




```
#Weighted std diffs
#Binary confounders:
#unweighted counts, percentages:
p0 = 100*sum(nighticu$Reason_index_ICU_admisssion[nighticu$Nighttime_ICU_discharge == 0]==1)/sum(!is.na
p1 = 100*sum(nighticu$Reason_index_ICU_admisssion[nighticu$Nighttime_ICU_discharge == 1]==1)/sum(!is.na
#weighted counts, percentages:
p0w = 100*sum(nighticu$ipw[nighticu$Reason_index_ICU_admisssion ==1 & nighticu$Nighttime_ICU_discharge =
p1w = 100*sum(nighticu$ipw[nighticu$Reason_index_ICU_admisssion ==1 &nighticu$Nighttime_ICU_discharge ==
#standardised differences: Note the denominator of the weighted stddiff!
stdiff1 = ((p1 - p0)/sqrt(((p0/100)*(1-p0/100) + (p1/100)*(1-(p1/100)))/2))
stdiffW = ((p1w - p0w)/sqrt(((p0/100)*(1-p0/100) + (p1/100)*(1-(p1/100)))/2))
print(paste("For Reason_index_ICU_admisssion standardised difference =", round(stdiff1, 2), "and the wei
```

```
## [1] "For Reason_index_ICU_admisssion standardised difference = 18.24 and the weighted standardised di
```

```
#Categorical confounder: here we need to consider each level of the
#confounder: we'll consider Admission_Source
table(nighticu$Admission_Source, nighticu$Nighttime_ICU_discharge)
```

```
##
##          0    1
##   0   1278  265
##   1    308   71
##   2    202   34
##   3   1717  245
```

```
##   4  158   35
```

```
for(j in 1:4){
  p0 = 100*sum(nighticu$Admission_Source[nighticu$Nighttime_ICU_discharge == 0]==j)/sum(!is.na(nighticu$
  p1 = 100*sum(nighticu$Admission_Source[nighticu$Nighttime_ICU_discharge == 1]==j)/sum(!is.na(nighticu$
  #standardised differences:
  p0w = 100*sum(nighticu$ipw[nighticu$Admission_Source ==j & nighticu$Nighttime_ICU_discharge == 0])/sum
  p1w = 100*sum(nighticu$ipw[nighticu$Admission_Source ==j &nighticu$Nighttime_ICU_discharge == 1])/sum
  #standardised differences: Note the denominator of the weighted stddiff!
  stdiff1 = ((p1 - p0)/sqrt(((p0/100)*(1-p0/100) + (p1/100)*(1-(p1/100)))/2))
  stdiffW = ((p1w - p0w)/sqrt(((p0/100)*(1-p0/100) + (p1/100)*(1-(p1/100)))/2))
  print(paste("For Admission_Source=",j," std diff =", round(stdiff1, 2), "and the weighted std diff is
  stdiff1 = ((p1 - p0)/sqrt(((p0/100)*(1-p0/100) + (p1/100)*(1-(p1/100)))/2))
}
```

```
## [1] "For Admission_Source= 1  std diff = 8.52 and the weighted std diff is  0.07"
## [1] "For Admission_Source= 2  std diff = -1.26 and the weighted std diff is  -0.13"
## [1] "For Admission_Source= 3  std diff = -18.67 and the weighted std diff is  0.64"
## [1] "For Admission_Source= 4  std diff = 4.99 and the weighted std diff is  -0.69"
```

```
#Continuous confounder:
#unweighted means, sds:
mean0 = mean(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge == 0])
var0 = var(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge == 0])
mean1 = mean(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge == 1])
var1 = var(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge == 1])
#weighted means:
mean0w = weighted.mean(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge==0], w=nighticu$ipw[nighticu$Ni
mean1w = weighted.mean(nighticu$Saps_3[nighticu$Nighttime_ICU_discharge==1], w=nighticu$ipw[nighticu$Ni
#standardised differences:
stdiff1= 100*((mean1 - mean0)/sqrt((var1 +var0)/2))
stdiffW= 100*((mean1w - mean0w)/sqrt((var1 +var0)/2))
print(paste("For Saps_3 std diff =", round(stdiff1, 2),"and weighted std. diff =", round(stdiffW, 2)))
```

```
## [1] "For Saps_3 std diff = 15.25 and weighted std. diff = -0.73"
```

```
#Fit the regression models
#Unadjusted
unadj <- glm(Status_Hospital_discharge ~ Nighttime_ICU_discharge,
    family= binomial(link="logit"),data=nighticu)
exp(confint(unadj))
```

```
## Waiting for profiling to be done...
```

```
##                              2.5 %     97.5 %
## (Intercept)             0.04628255 0.06212799
## Nighttime_ICU_discharge 0.89807745 1.79569381
```

```
#Adjusted
adj <- glm(Status_Hospital_discharge ~ Nighttime_ICU_discharge +
             Saps_3 + as.factor(Reason_index_ICU_admisssion) +
```

```r
             as.factor(Admission_Source) + Systemic_hypertension+ Diabetes_mellitus +
             Cancer + Congestive_heart_failure + COPD + Chronic_Kidney_disease +
             Liver_cirrhosis,
             family= binomial(link="logit"),data=nighticu)
exp(confint(adj))
```

```
## Waiting for profiling to be done...
```

```
##                                                2.5 %      97.5 %
## (Intercept)                             0.0006764246 0.004315402
## Nighttime_ICU_discharge                 0.6903703535 1.455960132
## Saps_3                                  1.0536594387 1.076073152
## as.factor(Reason_index_ICU_admisssion)1 0.4944017803 2.243140530
## as.factor(Admission_Source)1            1.3028408502 2.788021283
## as.factor(Admission_Source)2            1.4123165417 3.482828621
## as.factor(Admission_Source)3            0.3514959528 1.679223666
## as.factor(Admission_Source)4            0.4736177226 1.827789925
## Systemic_hypertension                   0.7463478225 1.368677525
## Diabetes_mellitus                       0.6765093411 1.270054305
## Cancer                                  0.9481371404 1.809326096
## Congestive_heart_failure                0.9355944123 1.954600353
## COPD                                    1.1800700381 2.540390347
## Chronic_Kidney_disease                  0.9935605426 2.425614702
## Liver_cirrhosis                         0.5084039591 1.981666460
```

```r
#IPW
ipwadj <- (svyglm(Status_Hospital_discharge ~ as.factor(Nighttime_ICU_discharge),
                  family=binomial(link="logit"),
                  design = svydesign(~ 1, weights = ~ ipw,
                                     data = nighticu)))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```r
exp(confint(ipwadj))
```

```
##                                          2.5 %     97.5 %
## (Intercept)                         0.04762135 0.06393619
## as.factor(Nighttime_ICU_discharge)1 0.74429571 1.49779877
```

```r
#E-value
evalue(OR(exp(ipwadj$coefficients[2]), rare=TRUE), lo=exp(confint(ipwadj)[2,])[1])
```

```
## Confidence interval crosses the true value, so its E-value is 1.
```

```
##             point     lower upper
## RR       1.055843 0.7442957    NA
## E-values 1.298664 1.0000000    NA
```

```r
evalue(OR(exp(confint(ipwadj)[2,])[1], rare=TRUE))
```

```
##              point lower upper
## RR       0.7442957    NA    NA
## E-values 2.0229488    NA    NA
```