Assess 1

Jodie

2024-04-17

General question: In this randomized study, what is the effect of the drug on elevated potassium levels (i.e. hyperkalemia)?

Data sets: 1. t05xc: Adverse event data 2. t016: Lab test data 3. master_data: Master database for clinical trial

Notes: 1. Hyperkalemia is defined as any potassium value >=5.5 2. Potassium values are recorded in the column labeled "k" in both the adverse event data set and the lab test data set. 3. Each patient may have multiple potassium values recorded. We are interested in whether each patient EVER experienced hyperkalemia. Assume that if no potassium values are available, then hyperkalemia was not present. 4. Patients are identified by "master_id" in each of the data sets. 5. Each patient's treatment assignment is identified by "treat" (0 for placebo; 1 for drug) in the master data set. 6. Geographic region is identified by "region" in the master data set.

Questions: 1. Is there evidence that the drug is associated with hyperkalemia? 2. Does the drug effect depend on geographic region?

Further questions: after concluding there is an association, how to identify the relationship–try doing regression analysis? What test to answer–does the drug effect depend on geographic region?

Is there evidence that the drug is associated with hyperkalemia? Primary approach: Do hypothesis test to see if there significant difference between drug group and placebo group -> chi sq test of independence (essentially a log linear model) Do logistic regression on the drug, with a random intercept terms for the patient and for the geographic region. Regress against hyperkalemia as label In other words, response variable is hyperkalemia and feature

```
# join dataframe by master_id

# test that looks at
# assuming all other variables are held constant
# do hypothesis test to see if there significant difference between drug group and placebo group -> chi
## H0: no difference in proportion between groups; Ha: there is difference between groups; if p-val < 0
# there hyperkalmeia is associated with presence of drug

# create table
# 1. create two subsets of data--> 1 for drug and 0 for no drug
# 2. calculate num of patients who experienced hyperkalemia in each group

# manipulating dataframe
library(dplyr)
```

Create a new column–overall_hyperk and join dataframes to create a single dataframe that has all variables of interest "treat", "region", "master_id", "overall_hyperk"

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.2.3
```

```r
master <- read.csv("/Users/jojoc/Downloads/Data specialist assessment/Assignment No 1/master_data.csv")
adverse.df <- read.csv("/Users/jojoc/Downloads/Data specialist assessment/Assignment No 1/t016.csv")
lab.df <- read.csv("/Users/jojoc/Downloads/Data specialist assessment/Assignment No 1/t05xc.csv")

#head(master)
#head(adverse.df)
#tail(lab.df)

#table(lab.df$test_type)
lab.df$test_type <- tolower(lab.df$test_type)
#lab.df

# revise lab df

# select only rows with "pota" or "k" in test_type col for lab.df using regular expression --> to inclu
sel_lab_ <- lab.df[grepl("pota|k", lab.df$test_type, ignore.case = TRUE), ]
#sel_lab_# updated lab df to contain only potassium tests, need to aggregate by patient later to assign

# aggregate by master_id to check if at least one k is >= 5.5; if so assign 1 (true) to has_hyperk col
sel_lab_2 <- sel_lab_ %>%
  mutate(k = replace_na(k, 0)) %>%
  group_by(master_id) %>%
  summarize(has_hyperk = any(k >= 5.5)) %>%
  mutate(has_hyperk_num = as.numeric(has_hyperk))


#sel_lab_2 # most updated lab
#sel_lab_

# revise master to contain treat and drug
rev_master <- master[, c("master_id", "treat", "region")] # only one occurrence for each patient
                                                          # as determined doing table()
#rev_master # most updated master
unique(rev_master$region) # region 1 or 2
```

```
## [1] 1 2
```

```
# revise adverse events df to also have has_hyperk col
adverse.df2 <- adverse.df %>%
  select(master_id, k) %>%
  mutate(k = replace_na(k, 0)) %>% # replace na's with 0
  group_by(master_id) %>%
  summarize(has_hyperk = any(k >= 5.5)) %>%
  mutate(has_hyperk_num = as.numeric(has_hyperk))

# most updated dataframes are sel_lab_2, rev_master, and adverse.df2
head(rev_master)
```

```
##   master_id treat region
## 1  10010013     1      1
## 2  10010025     0      1
## 3  10010037     1      1
## 4  10010049     1      1
## 5  10010051     1      1
## 6  10010063     0      1
```

```
head(sel_lab_2)
```

```
## # A tibble: 6 x 3
##   master_id has_hyperk has_hyperk_num
##       <int> <lgl>               <dbl>
## 1  10010025 FALSE                   0
## 2  10010051 FALSE                   0
## 3  10010102 TRUE                    1
## 4  10010114 FALSE                   0
## 5  10010152 FALSE                   0
## 6  10010188 FALSE                   0
```

```
head(adverse.df2)
```

```
## # A tibble: 6 x 3
##   master_id has_hyperk has_hyperk_num
##       <int> <lgl>               <dbl>
## 1  10010013 FALSE                   0
## 2  10010025 FALSE                   0
## 3  10010037 FALSE                   0
## 4  10010049 TRUE                    1
## 5  10010051 TRUE                    1
## 6  10010063 FALSE                   0
```

```
# might be some overlaps between sel_lab_2 and adverse.df2--
# assuming that that there are patients in both dataframes,
# if has hyperk is true in adverse.df2 but false in sel_lab_2, mark as true overall for has_hyperk
sel_lab_adverse <- rbind(sel_lab_2, adverse.df2)
sel_lab_adverse
```

```
## # A tibble: 3,826 x 3
##    master_id has_hyperk has_hyperk_num
##        <int> <lgl>               <dbl>
##  1  10010025 FALSE                   0
##  2  10010051 FALSE                   0
##  3  10010102 TRUE                    1
##  4  10010114 FALSE                   0
##  5  10010152 FALSE                   0
##  6  10010188 FALSE                   0
##  7  10010190 FALSE                   0
##  8  10010215 FALSE                   0
##  9  10010227 FALSE                   0
## 10  10010241 FALSE                   0
## # i 3,816 more rows
```

```
sel_lab_adverse2 <- sel_lab_adverse %>%
  group_by(master_id) %>%
  summarize(overall_has_hyperk = any(has_hyperk_num==1)) %>%
   mutate(overall_hyperk = as.numeric(overall_has_hyperk))

dim(sel_lab_adverse2) # final joint data frame created by joining sel_lab_2 and adverse.df2
```

```
## [1] 3437    3
```

```
dim(rev_master)
```

```
## [1] 3445    3
```

```
# join sel_lab_adverse2 and rev_master by master_id

# include all=TRUE to include patients that only appear in one of the two dfs being merged
final_df <- merge(sel_lab_adverse2, rev_master, by = "master_id")
head(final_df) # 3446 entries (when all=TRUE), otherwise 3,436 when all is false
```

```
##   master_id overall_has_hyperk overall_hyperk treat region
## 1  10010013              FALSE              0     1      1
## 2  10010025              FALSE              0     0      1
## 3  10010037              FALSE              0     1      1
## 4  10010049               TRUE              1     1      1
## 5  10010051               TRUE              1     1      1
## 6  10010063              FALSE              0     0      1
```

```
is.factor(final_df$region) # check if this column has levels, currently not so convert
```

To answer Q1, logistic regression on the treat and region and regress against overall_hyperk as label.

```
## [1] FALSE
```

```r
final_df$overall_hyperk <- as.factor(final_df$overall_hyperk)
final_df$treat <- as.factor(final_df$treat)
final_df$region <- as.factor(final_df$region) # making them factors might not make diference since each

# m1 <- glm(overall_hyperk ~ treat, data = final_df, family = binomial)
# summary(m1)
```

```r
m1_2 <- glm(overall_hyperk ~ treat + region, data = final_df, family = binomial)
summary(m1_2)
```

```
##
## Call:
## glm(formula = overall_hyperk ~ treat + region, family = binomial,
##     data = final_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7171  -0.5590  -0.4886  -0.3756   2.3180
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.06542    0.09272 -22.276  < 2e-16 ***
## treat1       0.83868    0.10475   8.007 1.18e-15 ***
## region2     -0.55049    0.10253  -5.369 7.92e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2768.2  on 3435  degrees of freedom
## Residual deviance: 2671.1  on 3433  degrees of freedom
## AIC: 2677.1
##
## Number of Fisher Scoring iterations: 5
```

```r
levels(final_df$treat) # inrepretreting the multiplicative change of odds in favor of y=1 as patient go
```

```
## [1] "0" "1"
```

```r
chisq.test(table(final_df$overall_hyperk, final_df$treat))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  table(final_df$overall_hyperk, final_df$treat)
## X-squared = 65.475, df = 1, p-value = 5.885e-16
```

Drug effect is significantly associated with presence of hyperkalemia as indicated by the chi-squared test. The estimate for treat is 0.83868 (significant as it its p-value < 0.05). For patients on drugs, the odds of developing hyperkalemia is multiplied by a factor of $\exp(0.83868) = 2.31$ compared to patients not on drugs. In other words, the odds of developing hyperkalmeia when on drugs are 2.31 times the odds for when one is not on drugs. Another way of interpreting this is, the probability of developing hyperkalemia are (2.31 - 1) x 100 = 131% higher when on drugs comprared to not being on drugs.

```
m2 <- glm(overall_hyperk ~ treat*region + treat + region, data = final_df, family = binomial)
summary(m2)
```

To answer Q2, add an interaction term (treatment*region) to the model and compare with m1_2 (without the interaction term)

```
##
## Call:
## glm(formula = overall_hyperk ~ treat * region + treat + region,
##     family = binomial, data = final_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7610  -0.5024  -0.4445  -0.4284   2.2063
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.34206    0.11931 -19.630  < 2e-16 ***
## treat1          1.25098    0.14231   8.790  < 2e-16 ***
## region2         0.07687    0.16795   0.458    0.647
## treat1:region2 -0.99190    0.21374  -4.641 3.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2768.2  on 3435  degrees of freedom
## Residual deviance: 2649.5  on 3432  degrees of freedom
## AIC: 2657.5
##
## Number of Fisher Scoring iterations: 5
```

If drug effect or treatment depends on region, we expect the model with the interaction term to fit our data better and is therefore the better model. To compare the models, we can use AIC which is included in the summary output. AIC tells you which model has better out of sample prediction accuracy. The better model will have a lower AIC. In this case, the model with the interaction term has lower AIC. $2657.5 < 2677.1$. Therefore, our model with the drug and region interaction term does better and drug depends on region. I considered computing the leave one cross validation to compare the two models but AIC for regression models is a sufficient approximation. It would be computationally expensive since I would need to fit the model once for each data point.

```
#coef(m2) # table that shows the slopes essitimates and intercept
coef(m2)
```

```
##    (Intercept)         treat1        region2 treat1:region2
##    -2.34205553     1.25097907     0.07687002    -0.99190383
```

```
table_matrix <- matrix(NA, nrow = 2, ncol = 2)

# Fill in the table with the calculated values
table_matrix[1, 1] <- exp(coef(m2)[1]) # -2.34  and raising that to e
```

```r
table_matrix[1, 2] <- exp(coef(m2)[1] + coef(m2)[3]) # -2.34 + 0.08 and raising that to e
table_matrix[2, 1] <- exp(coef(m2)[1] + coef(m2)[2]) #  -2.34 + 1.25 and raising that to e
table_matrix[2, 2] <- exp(coef(m2)[1] + coef(m2)[3] + coef(m2)[2] + coef(m2)[4]) # -2.34 + 0.08 - 0.99

# Convert the matrix to a data frame for better visualization
table_df <- as.data.frame(table_matrix)
rownames(table_df) <- c("treat0", "treat1")
colnames(table_df) <- c("region1", "region2")

table_df
```

```
##           region1   region2
## treat0 0.09612984 0.1038108
## treat1 0.33585477 0.1345109
```

This table represents the odds of getting hyperkalemia based on whether the patients are on drugs and what region they come from. From the table, you can see the odds are highest for patient on drugs and from region 1. For region 1, the treatment effect is 1.25. For region 2 the treatment effect is 1.25 - 0.99 = 0.26. Therefore there is a stronger association between drugs and getting hyperkalemia for patients from region 1.

```r
# table_matrix <- matrix(NA, nrow = 2, ncol = 2)
#
# # Fill in the table with the calculated values
# table_matrix[1, 1] <- exp(coef(m2)[1]) # -2.34  and raising that to e
# table_matrix[1, 2] <- exp(coef(m2)[1] + coef(m2)[2]) # -2.34 + 0.08 and raising that to e
# table_matrix[2, 1] <- exp(coef(m2)[1] + coef(m2)[3]) #  -2.34 + 1.25 and raising that to e
# table_matrix[2, 2] <- exp(coef(m2)[1] + coef(m2)[2] + coef(m2)[3] + coef(m2)[4]) # -2.34 + 0.08 - 0.9
#
# # Convert the matrix to a data frame for better visualization
# table_df <- as.data.frame(table_matrix)
# rownames(table_df) <- c("treat0", "treat1")
# colnames(table_df) <- c("region1", "region2")
```

Another way to answer Q2, compute the CI for the interaction term