# S520 Instructor's Solutions

Spring 2023 STAT-S 520

```
library(infer)
```

## 1.

### 1a.

```
obs = c(121,84,118,226,226,123)
n = sum(obs)
p = c(0.13,0.14,0.13,0.24,0.20,0.16)
exp = n * p
exp
```

```
## [1] 116.74 125.72 116.74 215.52 179.60 143.68
```

```
G2 = sum(2*obs*log(obs/exp))
G2
```

```
## [1] 30.56574
```

We find the degrees of freedom now. The unrestricted set has $6 - 1 = 5$ probabilities that are free to vary. The null hypothesis (restricted case) specifies a single point for each probability as created with the variable `p` so there is no freedom under the restricted case. The degrees of freedom are then $(6 - 1) - 0 = 5$

```
df = (6 - 1) - 0
1 - pchisq(G2, df)
```

```
## [1] 1.141029e-05
```

The p value is very small (close to zero). We can conclude that the claimed proportions are not credible in light of these data.

### 1b.

We first create a data frame to work with that contains the observed data and then hypothesize our probabilities.

```
X2 = sum((obs - exp)^2/exp) # The test statistic needed from the original sample

# Here is the code for the data frame
choc = c("Brown","Yellow","Red","Blue","Orange","Green")
choc.vec = rep(choc,obs)
df1 = data.frame(choc.vec)
null_dist <- df1 |>
  specify(response = choc.vec) |>
  hypothesize(null = "point", p = c("Brown" = 0.13,
                                    "Yellow" = 0.14,
                                    "Red" = 0.13,
                                    "Blue"=0.24,
                                    "Orange" = 0.20,
                                    "Green" = 0.16)) |>
  generate(reps = 5000, type = "draw") |>
  calculate(stat = "Chisq")

null_dist |>
  get_p_value(obs_stat = X2, direction = "greater")
```

```
## Warning: Please be cautious in reporting a p-value of 0. This result is an
## approximation based on the number of `reps` chosen in the `generate()` step.
## See `?get_p_value()` for more information.

## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

The p value is zero, so none of test statistics obtained from the simulated samples was as extreme as the one obtained from the original sample. This result matches our conclusion using the theory-based approach.

### 1c.

```
obs
```

```
## [1] 121  84 118 226 226 123
```

```
exp
```

```
## [1] 116.74 125.72 116.74 215.52 179.60 143.68
```

By comparing expected counts with observed counts, it seems that the expected probability of Yellow should be smaller and the expected probability of Orange should be greater. So let's make the required changes to Yellow (2nd value) and Orange (5th values) accordingly:

```
n = sum(obs)
p # former vector of probabilities
```

```
## [1] 0.13 0.14 0.13 0.24 0.20 0.16
```

```
p[2] = p[2] - 0.03 #Yellow
p[5] = p[5] + 0.03 #Orange
p # adjusted vector of probabilities
```

```
## [1] 0.13 0.11 0.13 0.24 0.23 0.16
```

```
exp = n * p # new expected counts
cbind(exp,obs,p)
```

```
##         exp obs    p
## [1,] 116.74 121 0.13
## [2,]  98.78  84 0.11
## [3,] 116.74 118 0.13
## [4,] 215.52 226 0.24
## [5,] 206.54 226 0.23
## [6,] 143.68 123 0.16
```

```
G2 = sum(2*obs*log(obs/exp))
G2
```

```
## [1] 7.908566
```

```
df = (6 - 1) - 0
1 - pchisq(G2, df)
```

```
## [1] 0.1613472
```

The p-value is no longer small. We fail to reject the null hypothesis and conclude that the adjusted probabilities are plausible and potentially an appropriate representation of the M&M color proportions.

## 2.

### 2a.

If $X$ is the random variable that counts the number of black pixels, under the null hypothesis, we assume that $X \sim binomial(n = 16, p = 0.29)$, the expected values for the 9 cells in the problem will be $P(E_1) = P(X = 0) + P(X = 1)$, $P(E_2) = P(X = 2), \ldots, P(E_8) = P(X = 8)$ and $P(E_9) = P(X \geq 9)$. We can use the R function `dbinom()` to calculate the expected counts:

```
obs=c(30, 93, 159, 184, 195, 171, 92, 45, 31)
n = sum(obs)

p = dbinom(0:16, 16, 0.29)
p.exp = c(p[1]+p[2], p[3:9], sum(p[10: 17]))
p.exp
```

```
## [1] 0.03142165 0.08348225 0.15912578 0.21123387 0.20706870 0.15505849 0.09047678
## [8] 0.04157472 0.02055776
```

```r
exp = n * p.exp
cbind(obs,exp)
```

```
##      obs       exp
##  [1,]  30  31.42165
##  [2,]  93  83.48225
##  [3,] 159 159.12578
##  [4,] 184 211.23387
##  [5,] 195 207.06870
##  [6,] 171 155.05849
##  [7,]  92  90.47678
##  [8,]  45  41.57472
##  [9,]  31  20.55776
```

We find the degrees of freedom now. The unrestricted set has 9 cells, so $9 - 1 = 8$ probabilities that are free to vary. The null hypothesis (restricted case) specifies a single point for each probability as created with the variable `p` so there is no freedom under the restricted case. The degrees of freedom for the test are $(9 - 1) - 0 = 8$

```r
G2 = sum(2*obs*log(obs/exp))
df = (9-1) - 0
1 - pchisq(G2, df)
```

```
## [1] 0.1525781
```

The p value is fairly large. We cannot reject the null hypothesis so it seems that the coloring algorithm is performing as intended.

## 2b.

We first create a dataframe to work with that contains the observed data and then hypothesize our probabilities.

```r
X2 = sum((obs - exp)^2/exp)

pix = as.character(1:9)
pix.vec = rep(pix,obs)
df1 = data.frame(pix.vec)


names(p.exp) = pix
p.exp # creating the vector of probabilities with names
```

```
##          1          2          3          4          5          6          7
## 0.03142165 0.08348225 0.15912578 0.21123387 0.20706870 0.15505849 0.09047678
##          8          9
## 0.04157472 0.02055776
```

```r
null_dist <- df1 |>
  specify(response = pix.vec) |>
  hypothesize(null = "point",
  p = p.exp) |>
  generate(reps = 5000, type = "draw") |>
  calculate(stat = "Chisq")

null_dist |>
  get_p_value(obs_stat = X2, direction = "greater")
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1   0.123
```

The p-value is large enough, and it leads to the same conclusion obtained using the theory-based approach.

## 2c.

We can observe from the question that number of black pixel will follow a binomial distribution with any p. We can estimate the value of p from the data by calculating the proportion of the pixels that are painted by black out of 1000 non overlapping squares.

```r
black.prop = sum(obs*(1:9))/16
phat = black.prop/1000
phat
```

```
## [1] 0.2945625
```

This again turns out to be a value which is close to the mentioned probability. Using this to perform the test.

```r
obs=c(30, 93, 159, 184, 195, 171, 92, 45, 31)
n = 1000
vec = n*dbinom(0:16, 16, phat)
exp = c(vec[1]+vec[2], vec[3:9], sum(vec[10:17]))
exp
```

```
## [1]  28.89031  78.69656 153.34918 208.10558 208.55177 159.65195  95.23468
## [8]  44.73697  22.78301
```

The degrees of freedom do change. The unrestricted dimensions remain the same but now the restricted case has 1 degree of freedom, because the probability of success was not given (so it was free to vary). The degrees of freedom are now $df = (9 - 1) - 1 = 7$.

```r
G2 = sum(2*obs*log(obs/exp))
1 - pchisq(G2, 7)
```

```
## [1] 0.1845779
```

The p-value is still large enough and the conclusion remains the same as before (fail to reject $H_0$.

## 3.

### 3a.

We need to perfom a test for independence. We can create the contingency table using matrix().

```
Positive = c(74, 68, 154, 18)
Partial = c(18, 16, 54, 10)
None = c(12, 12, 58, 44)
obs = data.frame(Positive, Partial, None)
rownames(obs) = c("LP", "NS", "MC", "LD")
obs
```

```
##    Positive Partial None
## LP       74      18   12
## NS       68      16   12
## MC      154      54   58
## LD       18      10   44
```

We can now use the outer product to calculate the expected values under the assumption of independence.

```
exp = (rowSums(obs) %o% colSums(obs))/sum(obs)
exp
```

```
##     Positive  Partial     None
## LP  60.69888 18.94424 24.35688
## NS  56.02974 17.48699 22.48327
## MC 155.24907 48.45353 62.29740
## LD  42.02230 13.11524 16.86245
```

The degrees of freedom will be (r-1) * (c-1)

```
G2 = 2*sum(obs*log(obs/exp))
1 - pchisq(G2, (4-1)*(3-1))
```

```
## [1] 9.139356e-13
```

We reject the null hypothesis of independence: The response to treatment is related to the histological type.

### b.

```
X2 = sum((obs - exp)^2/exp)
X2
```

```
## [1] 75.89015
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
colnames(obs) # We'll use these names in our code below
```

```
## [1] "Positive" "Partial"  "None"
```

```r
df.obs = as.data.frame(obs)
# The long data will have 'Type' (Rows) and 'Response' (Columns) as the new variables names

data2 <- df.obs |>
  rownames_to_column('Type') |>
  pivot_longer(cols=colnames(obs),
               names_to='Response',
               values_to='count') |>
  rowwise() |>
  mutate(count = list(1:count)) |>
  unnest(count) |>
  select(-count)
```

```r
null_dist <- data2 |>
  specify(Response ~ Type) |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>
  calculate(stat = "Chisq")

null_dist |>
  get_p_value(obs_stat = X2, direction = "greater")
```

```
## Warning: Please be cautious in reporting a p-value of 0. This result is an
## approximation based on the number of `reps` chosen in the `generate()` step.
## See `?get_p_value()` for more information.
```

```
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

We reject the null hypothesis of independence which matches the theory based approach: The response to treatment is related to the histological type.