HW3, SOLUTIONS

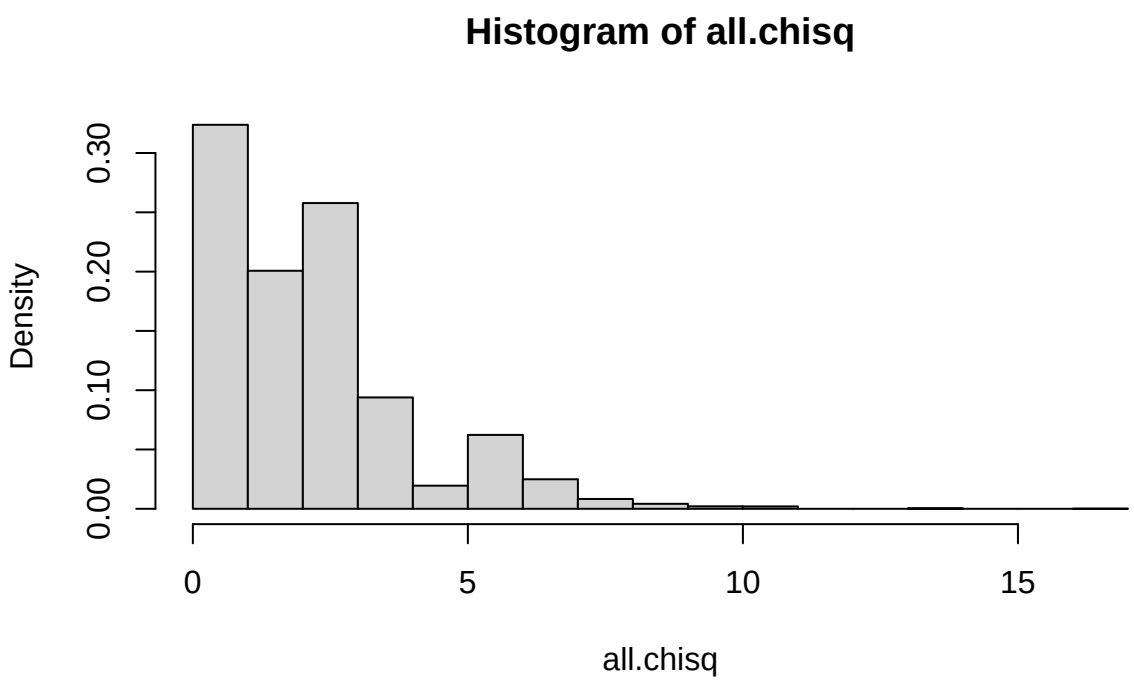
Problem #1

1. Code below:

```
my.permutation.test <- function(df, n.perm){
  # Calculating the full (non-permuted) X^2 test statistic value
  # (to be later used for p-value calculation).
  full.chisq.val <- chisq.test(table(df))$statistic
  all.chisq <- numeric(n.perm)
  for (j in 1:n.perm){
    permute.x <- sample(df[,1])
    permute.df <- data.frame(x = permute.x,
                             y = df[,2])
    all.chisq[j] <- chisq.test(table(permute.df))$statistic
  }

  hist(all.chisq, freq=F)
  perm.pval <- mean(all.chisq >= full.chisq.val)
  return(list(tab = table(df),
              pval = perm.pval))
}
```

2. a. For Snowden data:

```
student.status <- c(rep("US", 12),
                    rep("Intl", 8))
opinion <- c("Hero", rep("Criminal", 9), rep("Neither", 2),
             rep("Hero", 5), rep("Criminal", 2), rep("Neither", 1))
Snowden <- data.frame(student.status, opinion)

set.seed(1)
my.permutation.test(Snowden, n.perm=10000)
```




```
## $tab
##                opinion
## student.status Criminal Hero Neither
##           Intl        2    5       1
##           US          9    1       2
##
## $pval
## [1] 0.0345
```

Conclusion: we reject the H_0 of independence at $\alpha = 0.05$ level, as p-value is $0.0345 < 0.05$. The histogram looks very similar to the one in the slides.

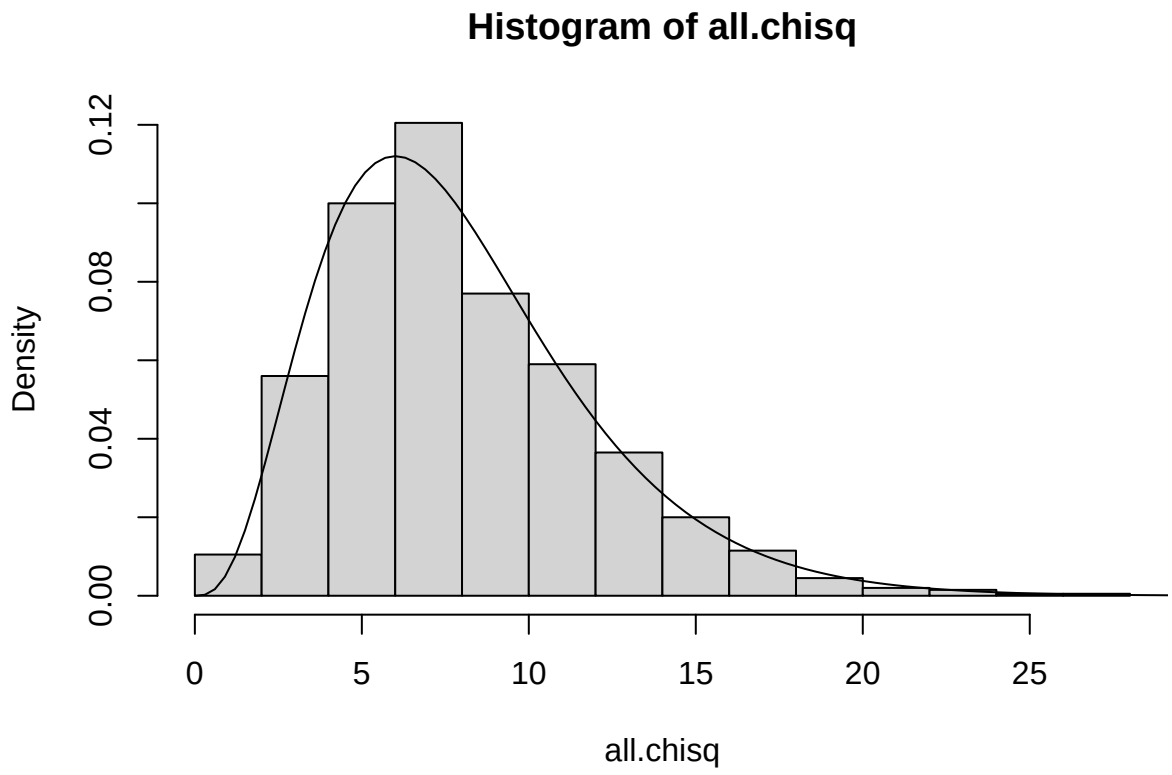
b. For Airbnb data:

```
listings <- read.csv("~/Downloads/listings.csv")

set.seed(1)
my.permutation.test(listings[,c("neighbourhood_group","room_type")], n.perm=1000)
```

```
## $tab
##                      room_type
## neighbourhood_group Entire home/apt Private room Shared room
##       Bronx                    378          659          68
##       Brooklyn                9565        10131         418
##       Manhattan              13054         7931         471
##       Queens                  2118         3489         204
##       Staten Island            181          187          10
```

```
##
## $pval
## [1] 0
```

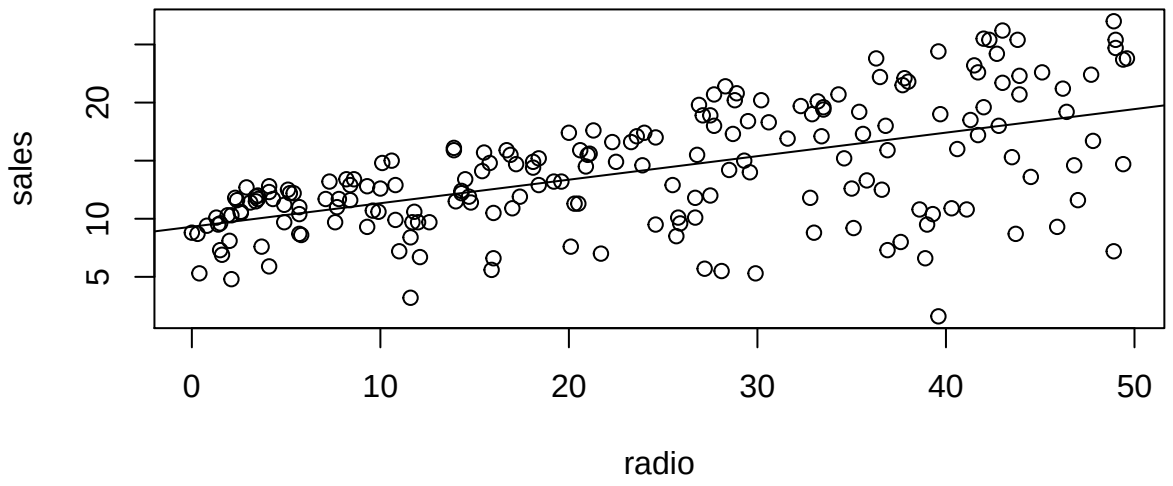
```r
# df = (r-1)x(c-1) = (5-1)*(3-1) = 8;
# =>  Chi^2_8,  E[Chi^2_8] = 8

# Overlaying the density of Chi^2_8.
g <- function(x){dchisq(x,df=8)}
curve(g, from=0, to=30, add=T)
```



Conclusion: we reject the H_0 hypothesis of independence. The resulting permutation distribution of X^2 is very close to χ^2_8, which means that our χ^2-test results from HW3 were likely appropriate for the Airbnb data.

Problem #2

```r
Advertising <- read.csv("~/Downloads/Advertising.csv")

lm.obj <- lm(sales ~ radio,
             data=Advertising)
plot(sales ~ radio,
     data=Advertising)
abline(lm.obj)
```




```
lm.obj
```

```
##
## Call:
## lm(formula = sales ~ radio, data = Advertising)
##
## Coefficients:
## (Intercept)        radio
##      9.3116       0.2025
```

```
predict(lm.obj, data.frame(radio=50))
```

```
##        1
## 19.43643
```

1. Yes, the relationship looks roughly positive linear, hence linear regression is not the worst tool imaginable here.
2. Fitted equation is:
$$\widehat{Sales} = 9.3116 + 0.2025\ radio$$
3. Interpretations:
   - Intercept: Markets that invest 0$ into radio advertisement will sell 9,311 items, **on average**.
   - Slope: Per $1,000$ increase in radio advertisement budget, we will sell $0.202 \times 1,000 = 202$ more items, **on average**.
4. Prediction for $50k$ invested into radio advertisement: $\approx 19.5k$ items sold. Interpretation: Markets that invest $50k$ into radio advertisement will sell $\approx 19.5k$ items, **on average**.
5. $RSE = 4.275$: Our model's predicted sales are off by $4,275$ items compared to the observed sales, on average.

6. $R^2 = 33.2\%$: Our model (linear regression with radio budget as predictor) explains 33.2% of uncertainty/variation in sales.

```
lm.obj
```

```
##
## Call:
## lm(formula = sales ~ newspaper, data = Advertising)
##
## Coefficients:
## (Intercept)    newspaper
##    12.35141      0.05469
```

```
predict(lm.obj, data.frame(newspaper=50))
```

```
##        1
## 15.08606
```

–>

Problem #3

1. a. Code below:

```
Advertising <- read.csv("~/Downloads/Advertising.csv")
attach(Advertising)

## Function definition
beta.hat.fun <- function(X,Y){
  beta1.hat <- sum((TV - mean(TV)) * (sales - mean(sales)))/sum((TV-mean(TV))^2)
  beta0.hat <- mean(sales) - beta1.hat*mean(TV)
  return(c(beta0.hat,
           beta1.hat))
}
```

b. Code below

```
## "Sanity check"
beta.hat.fun(TV, sales)
```

```
## [1] 7.03259355 0.04753664
```

```
lm.obj <- lm(sales ~ TV,
             data=Advertising)
lm.obj
```

```
##
## Call:
## lm(formula = sales ~ TV, data = Advertising)
```

```
##
## Coefficients:
## (Intercept)           TV
##     7.03259      0.04754
```

2. a. Code below:

```
## Function definition
RSE.R2.fun <- function(X,Y){
  n <- length(Y)
  Y.hat <- fitted(lm(Y~X))

  # RSE calculation
  RSE <- sqrt(sum((Y - Y.hat)^2)/(n-2))

  # R2 calculation
  TSS <- sum((Y-mean(Y))^2)
  RSS <- sum((Y-Y.hat)^2)
  R2 <- (TSS - RSS)/TSS

  print(c(RSE=RSE,
          R2=R2))
}
```

b. Code below:

```
Advertising <- read.csv("~/Downloads/Advertising.csv")
attach(Advertising)
```

```
## The following objects are masked from Advertising (pos = 3):
##
##     newspaper, radio, sales, TV, X
```

```
## "Sanity check"
RSE.R2.fun(TV, sales)
```

```
##       RSE        R2
## 3.2586564 0.6118751
```

```
summary(lm(sales ~ TV))[c("sigma", "r.squared")]
```

```
## $sigma
## [1] 3.258656
##
## $r.squared
## [1] 0.6118751
```