Solutions:

1. (5 points)

```
> # Setting directory and loading data
> setwd("~/Senior Year")
> load("flights.RData")
> library(vcd)
> # Finding the completed flights for each month
> jan_completed <- flights[which(flights$MONTH==1&flights$CANCELLED==0), ]
> jul_completed <- flights[which(flights$MONTH==7&flights$CANCELLED==0), ]
> # This is the day with most completed flights in January
> names(which.max(table(jan_completed$DAY_OF_MONTH)))
```

```
[1] "7"
```

```
> # This is the day with most completed flights in July
> names(which.max(table(jul_completed$DAY_OF_MONTH)))
```

```
[1] "26"
```

2. (5 points)

```
> # Finding total movement for january
> jan_total_mov <- table(jan_completed$ORIGIN) + table(jan_completed$DEST)
> # These are the three busiest airports in January
> head(sort(jan_total_mov, decreasing=TRUE), 3)
```

```
  LAX   ORD   ATL
17840 17461 17001
```

```
> # For july
> jul_total_mov <- table(jul_completed$ORIGIN) + table(jul_completed$DEST)
> # These are the three busiest airports in July
> head(sort(jul_total_mov, decreasing=TRUE), 3)
```

```
  ORD   LAX   ATL
21868 20890 20308
```

3. (5 points)

```
> # for January
> jan_flights <- flights[which(flights$MONTH == 1), ]
> jan_can <- jan_flights[which(jan_flights$CANCELLED == 1), ]
> #percentage of cancelled flights in january is
> length(jan_can$CANCELLED) / length(jan_flights$CANCELLED)
```

```
[1] 0.02597247
```

```
> # for July
> jul_flights <- flights[which(flights$MONTH == 7), ]
> jul_can <- jul_flights[which(jul_flights$CANCELLED == 1), ]
> #percentage of cancelled flights in january is
> length(jul_can$CANCELLED) / length(jul_flights$CANCELLED)
```

```
[1] 0.02129267
```

```r
> # so January has a higher percentage of cancelled flights
```

4. (5 points)

```r
> # 3 origin airports with the highest percentage of cancelled flights
> flights_can <- flights[which(flights$CANCELLED == 1), ]
> cancel_perc <- table(flights_can$ORIGIN) / table(flights$ORIGIN)
> head(sort(cancel_perc, decreasing=TRUE), 3)
```

```

       ORD        LGA        DCA 
0.05146881 0.04704842 0.04470331 
```

5. (5 points)

```r
> # Finding airports with a delayed flight >= 15 minutes
> flights_comp <- flights[which(flights$CANCELLED == 0), ]
> del_flights <- flights_comp[which(flights_comp$DEP_DELAY >= 15  | flights_comp$ARR_DELAY >= 15), ]
> del_airport <- table(del_flights$ORIGIN) + table(del_flights$DEST)
> del_perc <- del_airport / (table(flights$ORIGIN) + table(flights$DEST))
> # The three airports with the highest percentage of delayed flights
> head(sort(del_perc, decreasing=TRUE), 3)
```

```

      LGA       EWR       ORD 
0.3279990 0.3222707 0.3155395 
```

6. (5 points)

```r
> # seperating delayed flights by months
> del_flights_jan <- del_flights[which(del_flights$MONTH == 1), ]
> del_flights_jul <- del_flights[which(del_flights$MONTH == 7), ]
> # perc of delayed flights for January
> length(del_flights_jan$CANCELLED) / length(jan_flights$CANCELLED)
```

```
[1] 0.2433404
```

```r
> # perc of delayed flights for July
> length(del_flights_jul$CANCELLED) / length(jul_flights$CANCELLED)
```

```
[1] 0.2800162
```

```r
> # so July has a higher percentage of delayed flights
```

7. (5 points)

```r
> # finding the total delay time greater than 0 and 15
> total_del <- flights_comp$DEP_DELAY + flights_comp$ARR_DELAY
> total_del_0 <- total_del[which(total_del > 0)]
> total_del_15 <- total_del[which(total_del > 15)]
> # percentage for 0 minutes
> length(total_del_0) / length(total_del)
```

```
[1] 0.3889622
```

```
> # for 15 minutes
> length(total_del_15) / length(total_del)
```

```
[1] 0.2711599
```

8.
a.(5 points)

```
> # I used a factor to assign all days with the july weekdays, and then assigned the january ones
> jul_days <- c(rep(c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
+                    "Saturday", "Sunday"), times = 4), "Monday", "Tuesday", "Wednesday")
> jan_days <- c(rep(c("Tuesday", "Wednesday", "Thursday", "Friday", "Saturday",
+                     "Sunday", "Monday"), times = 4), "Tuesday", "Wednesday", "Thursday")
>
> flights$DAY_OF_WEEK <- factor(flights$DAY_OF_MONTH, levels = c(1:31), labels = jul_days)
>
> flights[which(flights$MONTH==1), ]$DAY_OF_WEEK<-  factor(jan_flights$DAY_OF_MONTH,
+                                                          levels = c(1:31), labels = jan_days)
>
> # this is how the new variable looks like
> head(flights$DAY_OF_WEEK)
```

```
[1] Sunday    Tuesday   Wednesday Wednesday Thursday  Friday
Levels: Monday Tuesday Wednesday Thursday Friday Saturday Sunday
```

```
> table(flights$DAY_OF_WEEK)
```

```

   Monday   Tuesday Wednesday  Thursday    Friday  Saturday    Sunday
    41067     43465     43996     39104     35310     28757     33645
```

b.(5 points)

```
> # resetting variables
> flights_can <- flights[which(flights$CANCELLED == 1), ]
>
> # day with most cancelled flights
> which.max(table(flights_can$DAY_OF_WEEK) / table(flights$DAY_OF_WEEK))
```

```
Thursday
       4
```

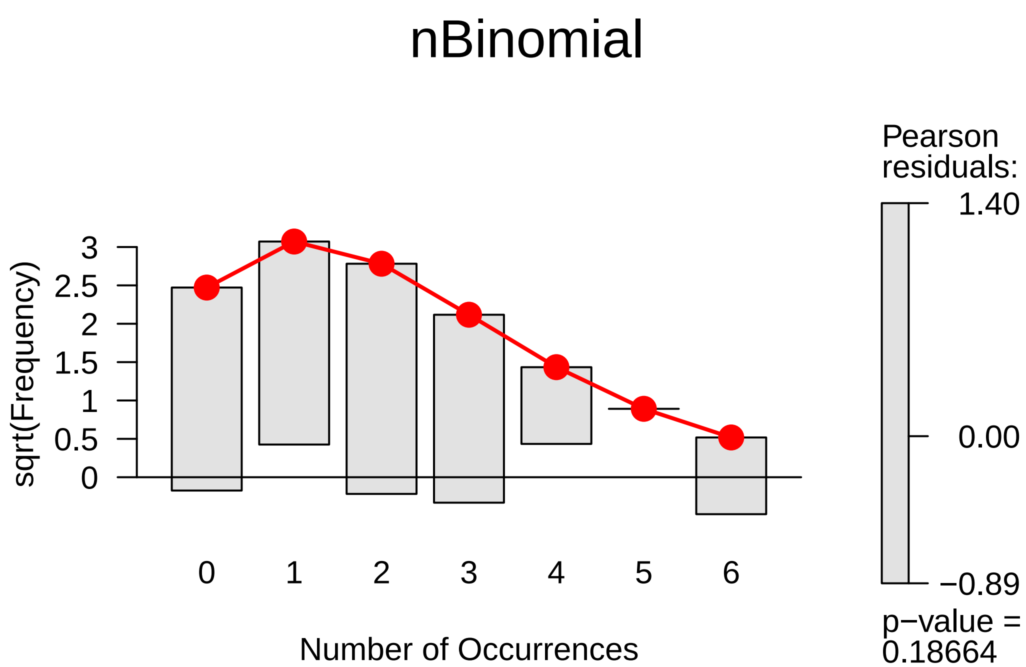
c.(5 points)

```
> # resetting variables
>
> flights_comp <- flights[which(flights$CANCELLED == 0), ]
> total_del <- flights_comp$DEP_DELAY + flights_comp$ARR_DELAY
> total_del_15 <- flights_comp[which(total_del > 15), ]
>
> # day with highest percentage of flights delayed over 15 minutes
> which.max(table(total_del_15$DAY_OF_WEEK) / table(flights$DAY_OF_WEEK))
```

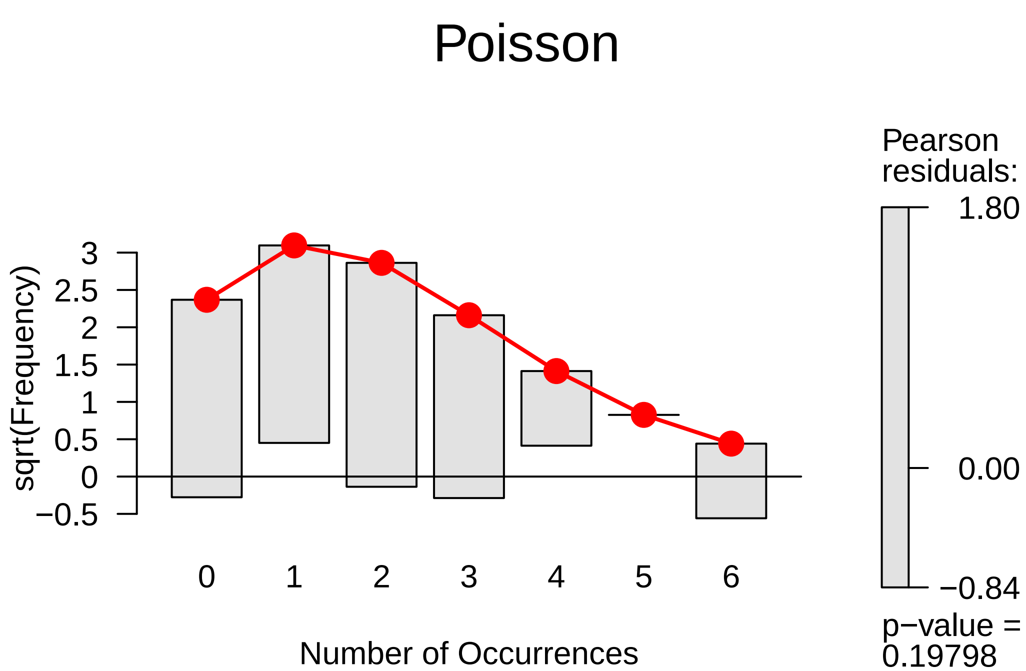
```
Thursday
       4
```

9. (5 points)

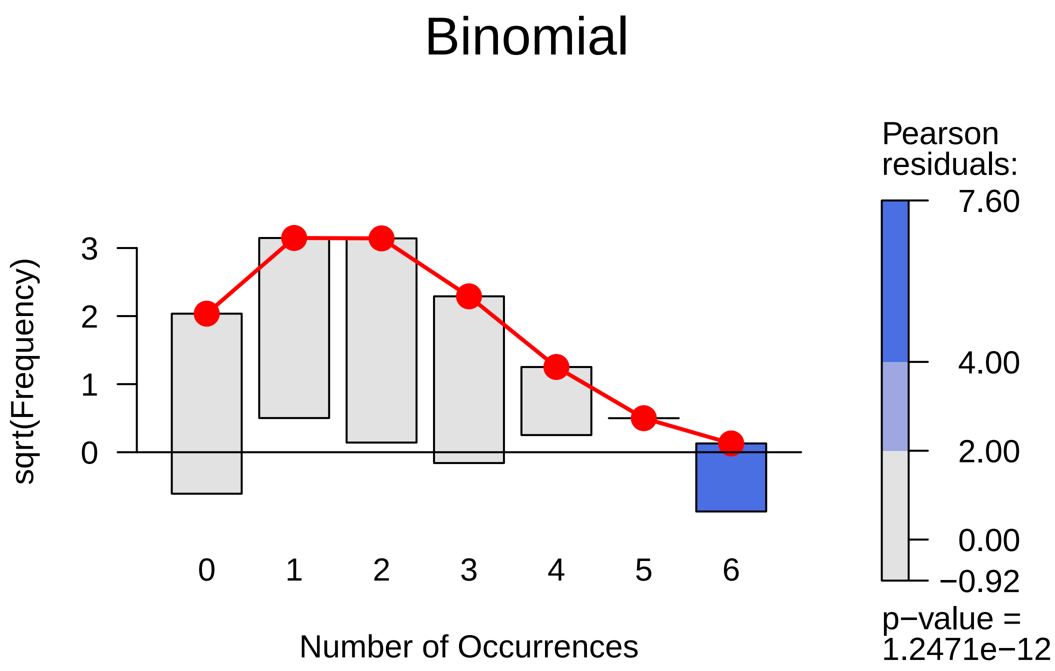
```
> # finding the cancelled flights scheduled to leave seattle
> X <- jan_can[which(jan_can$ORIGIN == "SEA"), ]
> # finding the number of cancelled flights for the unique days
> C <- as.vector(table(X$DAY_OF_MONTH))
> # does not count the days with non cancelled flights, so I will add them individually
> C <-c(C, 0, 0, 0, 0, 0, 0, 0)
>
> plot(goodfit(C, "nbinomial"), main = "nBinomial", shade = TRUE)
```




```
> plot(goodfit(C, "poisson"), main = "Poisson",shade = TRUE)
```



```
> plot(goodfit(C, "binomial"), main = "Binomial", shade = TRUE)
```




```
> git <- goodfit(C, "poisson")
```

The hypotheses

The Null Hypothesis : there is no significant difference between X and the poisson distribution. The Poisson distribution fits X the best.

The Alternative Hypothesis: there is significant difference between X and the poisson distribution. The Binomial or nBinomial distributions fit X the best.

We reject the Alternative Hypothesis since the p for poisson is 0.19798 which is greater than .05 and because this p value is greater than the ones for binomial and nbinomial. So poisson fits X best along with there being no significant difference between the two.

10. (5 points)

a.

```
> dpois(0, git$par$lambda)
```

```
[1] 0.1809241
```

b.

```
> 1-dpois(0, git$par$lambda)
```

```
[1] 0.8190759
```

c.

```
> sum(dpois(0:5, git$par$lambda))
```

```
[1] 0.9917894
```

d.

```
> sum(dpois(0:3, git$par$lambda))
```

```
[1] 0.9053576
```

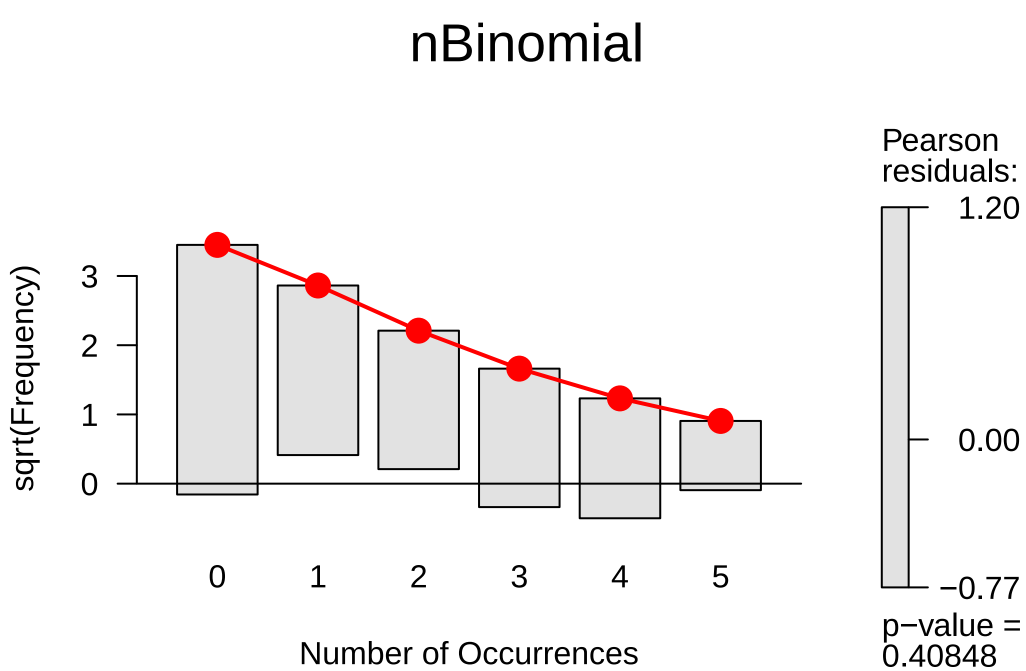
e.

```
> sum(dpois(0:2, git$par$lambda))
```

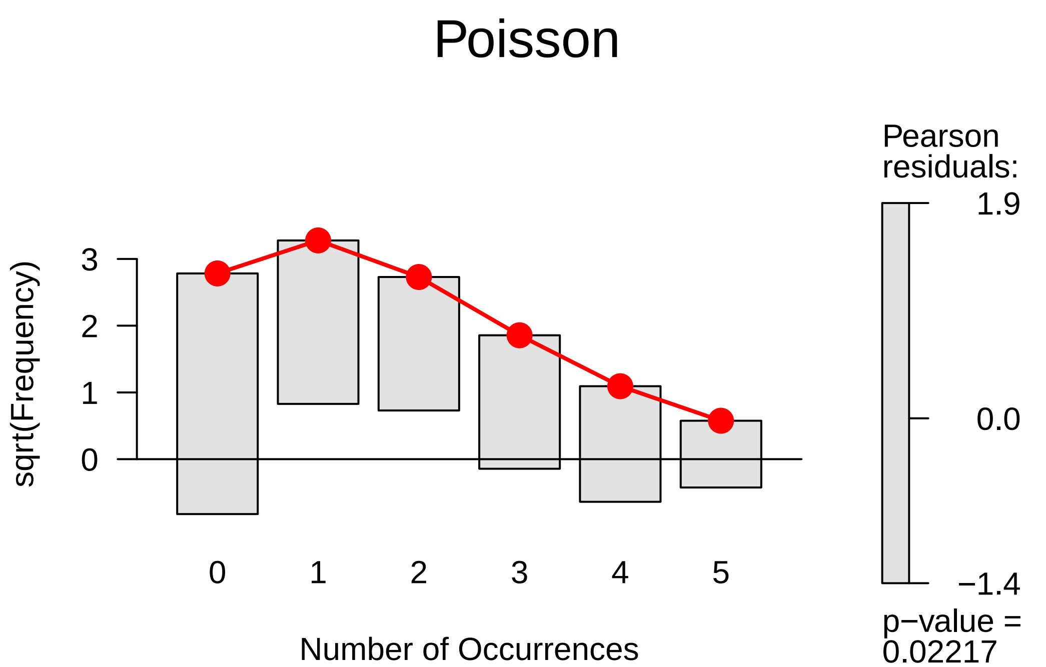
```
[1] 0.7546664
```

11. (5 points)

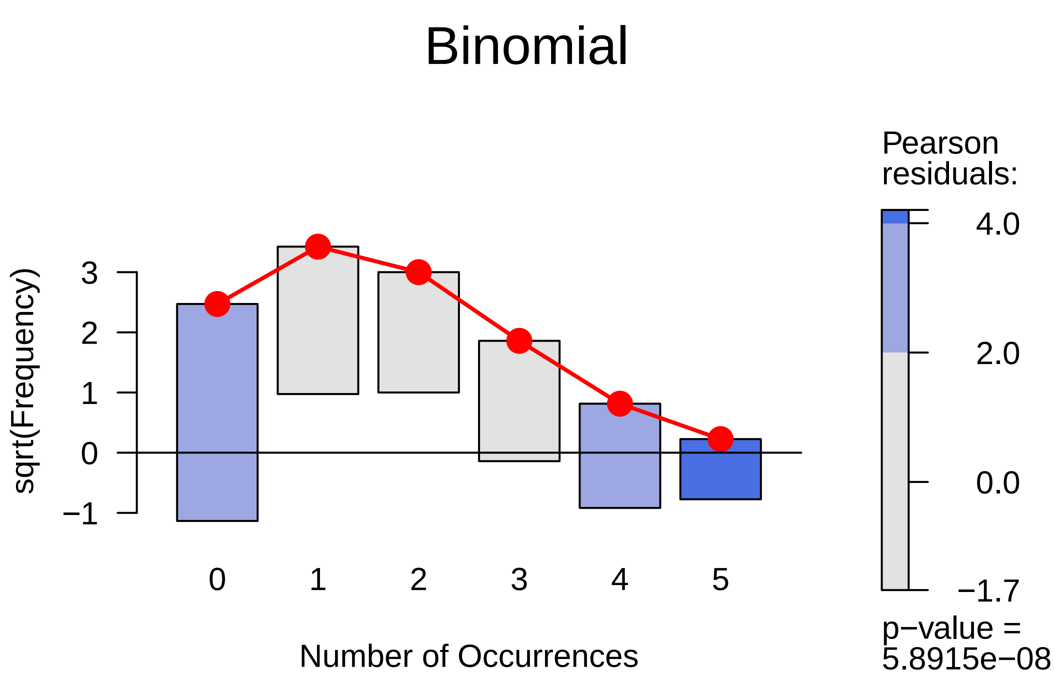
```
> Y <- jul_can[which(jul_can$ORIGIN == "SEA"), ]
> D <- as.vector(table(Y$DAY_OF_MONTH))
> D <- c(D, rep(0, times = 13))
> plot(goodfit(D, "nbinomial"), main = "nBinomial", shade = TRUE)
```



```
> plot(goodfit(D, "poisson"), main = "Poisson",shade = TRUE)
```




```
> plot(goodfit(D, "binomial"), main = "Binomial",shade = TRUE)
```




```
> # as we see, nbinomial has the highest p value
> gd <- goodfit(D, "nbinomial")$par
> Ex <- git$par$lambda
> Ey <- (gd$size / gd$prob)-gd$size
> Ex

[1] 1.709677
```

```
> Ey
```

```
[1] 1.387087
```

```
> # we see Ex>Ey, so january is expected to have more cancellation
```

12. (5 points)

```
> set.seed(1)
> tf <- rpois(10000, git$par$lambda) < rnbinom(10000, gd$size, gd$prob)
> true_tf <- tf[which(tf == TRUE)]
> length(true_tf) / 10000
```

```
[1] 0.2888
```