Problem set 7

2025-03-15

1. Load the **HistData** package. Create a `galton_height` data with the father's height and one randomly selected daughter from each family. Exclude families with no female children. Set the seed at 2007 and use the function `sample_n` to select the random child. You should end up with a `heights` dataset with two columns: `father` and `daughter`.

```r
library(HistData)
```

```
Warning: package 'HistData' was built under R version 4.4.3
```

```r
library(dplyr)
```

```

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
names(GaltonFamilies)
```

```
[1] "family"          "father"          "mother"          "midparentHeight"
[5] "children"        "childNum"        "gender"          "childHeight"
```

```
set.seed(2007)
heights <- GaltonFamilies %>%
  filter(gender == "female") %>%
  group_by(family) %>%
  sample_n(1) %>%
  ungroup() %>%
  select(father, daughter = childHeight)

head(heights)
```

```
# A tibble: 6 x 2
  father daughter
   <dbl>    <dbl>
1   78.5     69.2
2   75.5     65.5
3   75       68
4   75       64.5
5   75       62.5
6   74       69.5
```

2. Estimate the intercept and slope of the regression line for predicting daughter height Y using father height X. Use the following regression line formula:

$$\frac{\hat{Y} - \mu_Y}{\sigma_Y} = \rho \frac{x - \mu_x}{\sigma_x}$$

```
mu_x <- mean(heights$father)
mu_y <- mean(heights$daughter)
sigma_x <- sd(heights$father)
sigma_y <- sd(heights$daughter)

# Calculate the correlation coefficient
rho <- cor(heights$father, heights$daughter)

# Calculate regression coefficients
beta_1 <- rho * (sigma_y / sigma_x)
beta_0 <- mu_y - beta_1 * mu_x

# Parameters of the output regression equation
cat("Intercept ( 0):", beta_0, "\n")
```

```
Intercept ( 0): 36.56251
```

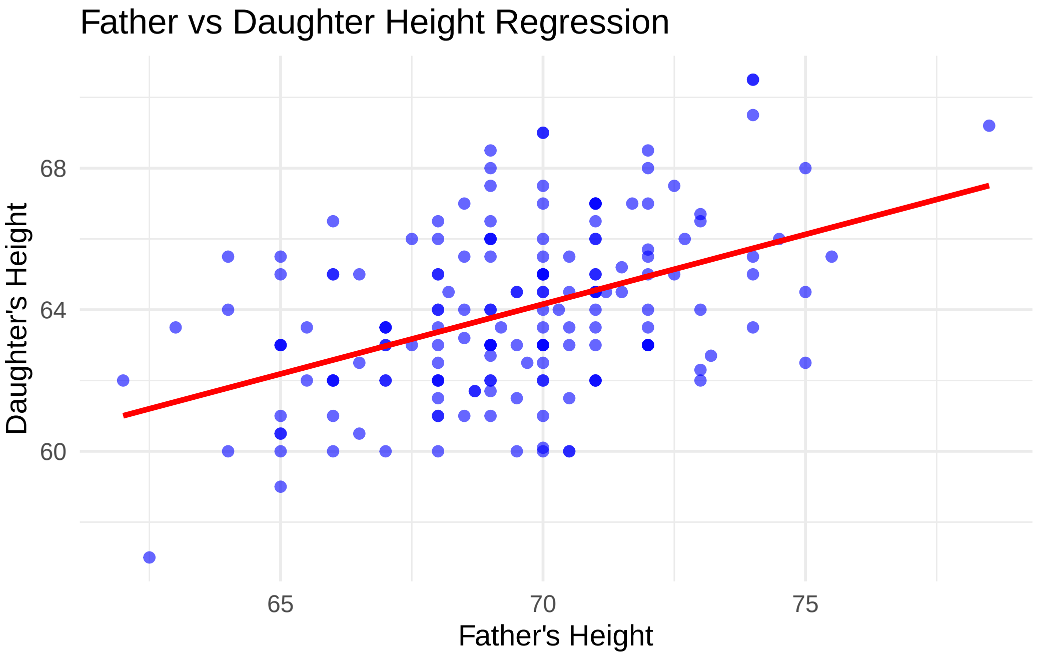
```
cat("Slope ( 1):", beta_1, "\n")
```

```
Slope ( 1): 0.394218
```

3. Make a plot to confirm the regression line goes through the data.

```
library(ggplot2)

ggplot(heights, aes(x = father, y = daughter)) +
  geom_point(color = "blue", alpha = 0.6) +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Father vs Daughter Height Regression",
       x = "Father's Height",
       y = "Daughter's Height") +
  theme_minimal()
```

```
`geom_smooth()` using formula = 'y ~ x'
```



4. Recompute the slope and intercept coefficients, this time using `lm` and confirm you get the same answer as with the formula used in problem 2.

```
# Calculate the regression model using lm()
model <- lm(daughter ~ father, data = heights)

# Output the regression coefficients
summary(model)
```

```

Call:
lm(formula = daughter ~ father, data = heights)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.3549 -1.5929 -0.1371  1.4937  4.8422 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 36.56251    4.09418   8.930 5.91e-16 ***
father       0.39422    0.05893   6.689 2.95e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.13 on 174 degrees of freedom
Multiple R-squared:  0.2046,    Adjusted R-squared:    0.2 
F-statistic: 44.75 on 1 and 174 DF,  p-value: 2.948e-10
```

```
beta_0_lm <- coef(model)[1]
beta_1_lm <- coef(model)[2]

cat("Intercept ( 0) from lm():", beta_0_lm, "\n")
```

```
Intercept ( 0) from lm(): 36.56251
```

```
cat("Slope ( 1) from lm():", beta_1_lm, "\n")
```

```
Slope ( 1) from lm(): 0.394218
```

5. Note that the interpretation of the intercept is: the height prediction for the daughter whose father is 0 inches tall. This is not a very useful interpretation. Re-run the regression but instead of father height use inches above average for each father: instead of using the x_is use $x_i - \bar{x}$. What is the interpretation of the intercept now? Does the slope estimate change?

```r
heights <- heights %>%
  mutate(father_centered = father - mean(father))

model_centered <- lm(daughter ~ father_centered, data = heights)

summary(model_centered)
```

```

Call:
lm(formula = daughter ~ father_centered, data = heights)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.3549 -1.5929 -0.1371  1.4937  4.8422 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     63.92841    0.16052 398.258  < 2e-16 ***
father_centered  0.39422    0.05893   6.689 2.95e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.13 on 174 degrees of freedom
Multiple R-squared:  0.2046,    Adjusted R-squared:    0.2 
F-statistic: 44.75 on 1 and 174 DF,  p-value: 2.948e-10
```

```r
beta_0_centered <- coef(model_centered)[1]
beta_1_centered <- coef(model_centered)[2]

cat("Intercept ( 0) with centered father heights:", beta_0_centered, "\n")
```

```
Intercept ( 0) with centered father heights: 63.92841 
```

```r
cat("Slope ( 1) with centered father heights:", beta_1_centered, "\n")
```

```
Slope ( 1) with centered father heights: 0.394218 
```

6. When using the centered father heights as a predictor, is the intercept the same as the average daughter height? Check if this is the case with the values you computed and then show that mathematically this has to be the case.

```r
mu_y <- mean(heights$daughter)

cat("Mean daughter height:", mu_y, "\n")
```

```
Mean daughter height: 63.92841
```

```r
cat("Intercept from centered regression:", beta_0_centered, "\n")
```

```
Intercept from centered regression: 63.92841
```

```r
all.equal(beta_0_centered, mu_y)
```

```
[1] "names for target but not for current"
```

For the next exercises install the **excessmort** package. For the latest version use

```r
library(devtools)
install_github("rafalab/excessmort")
```

7. Define an object `counts` by wrangling `puerto_rico_counts` to 1) include data only from 2002-2017 and counts for people 60 or over. We will focus in this older subset throughout the rest of the problem set.

```r
library(excessmort)
```

```
Warning: package 'excessmort' was built under R version 4.4.3
```

```r
library(dplyr)
library(lubridate)
```

```


Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```r
head(puerto_rico_counts)
```

```
  agegroup       date    sex population outcome
1      0-4 1985-01-01 female   158843.0       2
2      0-4 1985-01-01   male   164476.6       0
3      0-4 1985-01-02 female   158837.8       0
4      0-4 1985-01-02   male   164471.2       0
5      0-4 1985-01-03 female   158832.6       1
6      0-4 1985-01-03   male   164465.9       0
```

```r
puerto_rico_counts <- puerto_rico_counts %>%
  mutate(year = year(date))

unique(puerto_rico_counts$agegroup)
```

```
 [1] 0-4    5-9    10-14  15-19  20-24  25-29  30-34  35-39  40-44  45-49 
[11] 50-54  55-59  60-64  65-69  70-74  75-79  80-84  85-Inf
18 Levels: 0-4 5-9 10-14 15-19 20-24 25-29 30-34 35-39 40-44 45-49 ... 85-Inf
```

```r
filtered_counts <- puerto_rico_counts %>%
  filter(year >= 2002 & year <= 2017,
         agegroup %in% c("60-64", "65-69", "70-74", "75-79", "80-84", "85-Inf"))

head(filtered_counts)
```

```
  agegroup       date    sex population outcome year
1    60-64 2002-01-01 female   89850.74       3 2002
2    60-64 2002-01-01   male   76586.25       4 2002
3    60-64 2002-01-02 female   89858.23       3 2002
4    60-64 2002-01-02   male   76591.41       7 2002
5    60-64 2002-01-03 female   89865.73       1 2002
6    60-64 2002-01-03   male   76596.58       2 2002
```

8. Use R to determine what day of the week María made landfall in PR (September 20, 2017).

```r
landfall_date <- as.Date("2017-09-20")

day_of_week <- weekdays(landfall_date)
cat("Hurricane Maria made landfall in Puerto Rico on a", day_of_week, "\n")
```

```
Hurricane Maria made landfall in Puerto Rico on a Wednesday
```

10. Redefine the date column to be the start of the week that date is part of: in other words, round the date down to the nearest week. Use the day of the week María made landfall as the first day. So, for example, 2017-09-20, 2017-09-21, 2017-09-22 should all be rounded down to 2017-09-20, while 2017-09-19 should be rounded down to 2017-09-13. Save the resulting table in `weekly_counts`.

```r
library(lubridate)
library(dplyr)

weekly_counts <- puerto_rico_counts %>%
  mutate(week_start = date - (wday(date) - 4) %% 7)

head(weekly_counts)
```

```
  agegroup       date    sex population outcome year week_start
1      0-4 1985-01-01 female   158843.0       2 1985 1984-12-26
2      0-4 1985-01-01   male   164476.6       0 1985 1984-12-26
3      0-4 1985-01-02 female   158837.8       0 1985 1985-01-02
4      0-4 1985-01-02   male   164471.2       0 1985 1985-01-02
5      0-4 1985-01-03 female   158832.6       1 1985 1985-01-02
6      0-4 1985-01-03   male   164465.9       0 1985 1985-01-02
```

11. Now collapse the `weekly_count` data frame to store only one mortality value for each week, for each `sex` and `agegroup`. To this by by redefining `outcome` to have the total deaths that week for each `sex` and `agegroup`. Remove weeks that have less the 7 days of data. Finally, add a column with the MMWR week. Name the resulting data frame `weekly_counts`.

```r
library(MMWRweek)
```

```
Warning: package 'MMWRweek' was built under R version 4.4.3
```

```r
colnames(weekly_counts)
```

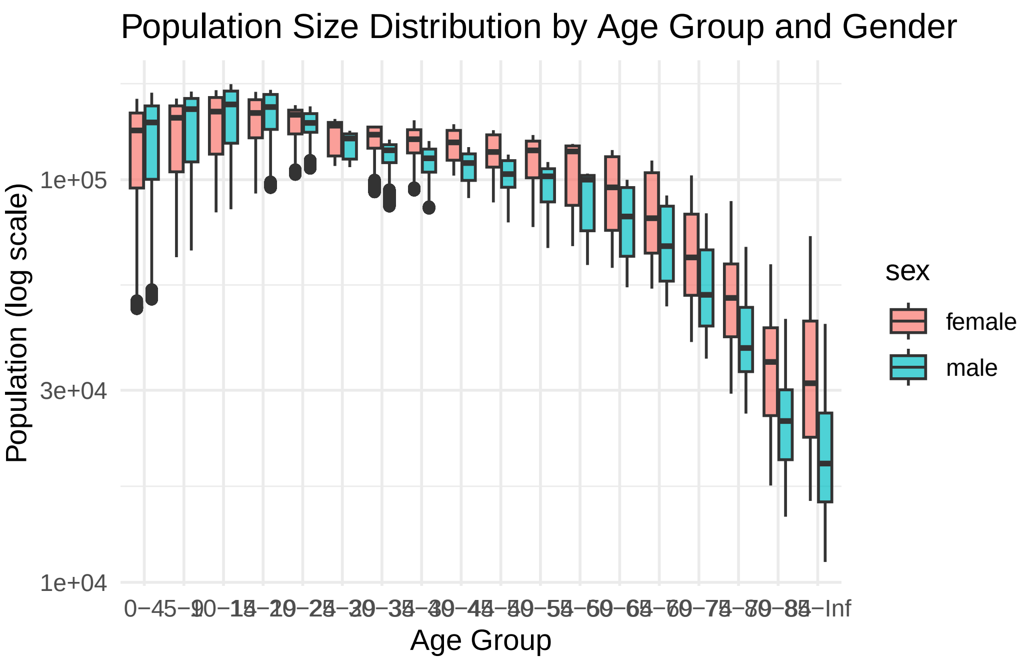
```
[1] "agegroup"   "date"       "sex"        "population" "outcome"
[6] "year"       "week_start"
```

```r
weekly_counts <- weekly_counts %>%
  group_by(week_start, sex, agegroup) %>%
  summarise(weekly_outcome = sum(outcome),
            days_counted = n(),
            .groups = "drop") %>%
  filter(days_counted == 7)

weekly_counts <- weekly_counts %>%
  mutate(MMWR_year = year(week_start),
         MMWR_week = MMWRweek(week_start)$MMWRweek)

head(weekly_counts)
```

```
# A tibble: 6 x 7
  week_start sex    agegroup weekly_outcome days_counted MMWR_year MMWR_week
  <date>     <chr>  <fct>             <dbl>        <int>     <dbl>     <dbl>
1 1985-01-02 female 0-4                   6            7      1985         1
2 1985-01-02 female 5-9                   0            7      1985         1
3 1985-01-02 female 10-14                 0            7      1985         1
4 1985-01-02 female 15-19                 2            7      1985         1
5 1985-01-02 female 20-24                 2            7      1985         1
6 1985-01-02 female 25-29                 3            7      1985         1
```

12. Comparing mortality totals is often unfair because the two groups begin compared have different population sizes. It is particularly important we consider rates rather than totals in this dataset because the demographics in Puerto Rico changed dramatically in the last 20 years. To see this use `puerto_rico_counts` to plot the population sizes by age group and gender. Provide a two sentence description of what you see.

```r
library(ggplot2)
library(dplyr)

puerto_rico_counts %>%
  ggplot(aes(x = agegroup, y = population, fill = sex)) +
  geom_boxplot(alpha = 0.7) +
  scale_y_log10() +
  labs(title = "Population Size Distribution by Age Group and Gender",
       x = "Age Group",
       y = "Population (log scale)") +
  theme_minimal()
```



13. Make a boxplot for each MMWR week's mortality rate based on the 2002-2016 data. Each week has 15 data points, one for each year. Then add the 2017 data as red points.

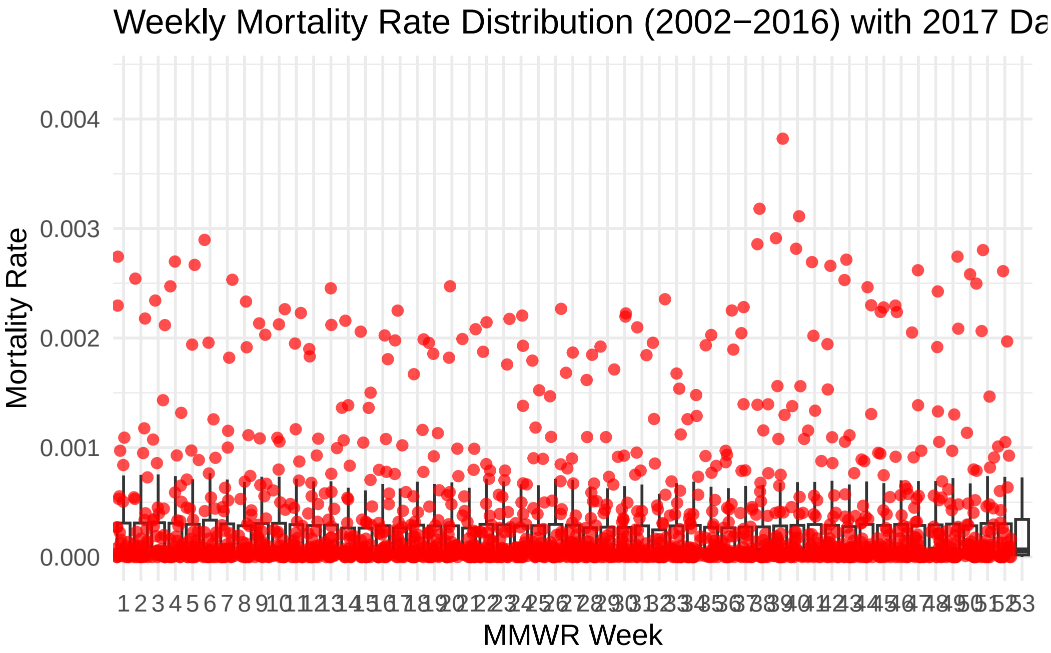
```
weekly_counts <- puerto_rico_counts %>%
  mutate(week_start = date - (wday(date) - 4) %% 7) %>%
  group_by(week_start, sex, agegroup) %>%
  summarise(
    weekly_outcome = sum(outcome, na.rm = TRUE),
    population = mean(population, na.rm = TRUE),
    days_counted = n(),
    .groups = "drop"
  ) %>%
  filter(days_counted == 7)

weekly_counts <- weekly_counts %>%
  mutate(MMWR_year = MMWRweek(week_start)$MMWRyear,
         MMWR_week = MMWRweek(week_start)$MMWRweek)

colnames(weekly_counts)
```

```
[1] "week_start"     "sex"            "agegroup"       "weekly_outcome"
[5] "population"     "days_counted"   "MMWR_year"      "MMWR_week"
```

```r
head(weekly_counts)
```

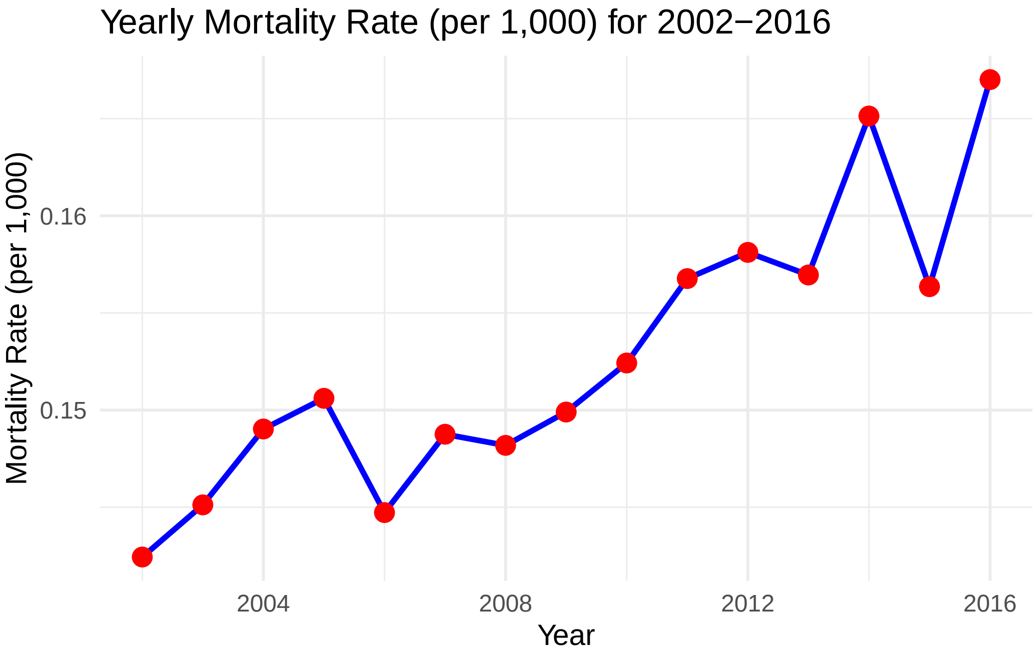
```
# A tibble: 6 x 8
  week_start sex    agegroup weekly_outcome population days_counted MMWR_year
  <date>     <chr>  <fct>             <dbl>      <dbl>        <int>     <dbl>
1 1985-01-02 female 0-4                   6    158822.            7      1985
2 1985-01-02 female 5-9                   0    159044.            7      1985
3 1985-01-02 female 10-14                 0    166233.            7      1985
4 1985-01-02 female 15-19                 2    165288.            7      1985
5 1985-01-02 female 20-24                 2    144553.            7      1985
6 1985-01-02 female 25-29                 3    132613.            7      1985
# i 1 more variable: MMWR_week <dbl>
```

```r
weekly_counts <- weekly_counts %>%
  mutate(mortality_rate = weekly_outcome / population)

weekly_counts_pre2017 <- weekly_counts %>%
  filter(MMWR_year >= 2002 & MMWR_year <= 2016)

weekly_counts_2017 <- weekly_counts %>%
  filter(MMWR_year == 2017)

ggplot(weekly_counts_pre2017, aes(x = as.factor(MMWR_week), y = mortality_rate)) +
  geom_boxplot(outlier.shape = NA, alpha = 0.7) +
  geom_jitter(data = weekly_counts_2017, aes(x = as.factor(MMWR_week), y = mortality_rate),
              color = "red", size = 1.5, alpha = 0.7) +
  labs(title = "Weekly Mortality Rate Distribution (2002-2016) with 2017 Data",
       x = "MMWR Week",
       y = "Mortality Rate") +
  theme_minimal()
```



14. Note two things: 1) there is a strong week effect and 2) 2017 is lower than expected. Plot the yearly rates (per 1,000) for 2002-2016:

```
yearly_counts <- weekly_counts %>%
  filter(MMWR_year >= 2002 & MMWR_year <= 2016) %>%
  group_by(MMWR_year) %>%
  summarise(
    total_deaths = sum(weekly_outcome, na.rm = TRUE),
    total_population = sum(population, na.rm = TRUE),
    mortality_rate = (total_deaths / total_population) * 1000
  )

ggplot(yearly_counts, aes(x = MMWR_year, y = mortality_rate)) +
  geom_line(color = "blue", linewidth = 1) +
  geom_point(size = 3, color = "red") +
  labs(title = "Yearly Mortality Rate (per 1,000) for 2002-2016",
       x = "Year",
       y = "Mortality Rate (per 1,000)") +
  theme_minimal()
```



15. The plot made in 14 explains why 2017 is below what is expected: there appears to be a general decrease in mortality with time. A possible explanation is that medical care is improving and people are living more healthy lives.

Fit a linear model to the weekly data for the 65 and older to the 2002-2016 data that accounts for:

- A changing population.
- The trend observed in 12.
- The week effect.
- Age effect.
- A sex effect.

Use rate as the outcome in the model.

```
weekly_counts_65plus <- weekly_counts %>%
  filter(MMWR_year >= 2002 & MMWR_year <= 2016, agegroup %in% c("65-69", "70-74", "75-79", "8
  mutate(
    mortality_rate = weekly_outcome / population,
    MMWR_week = as.factor(MMWR_week),
    sex = as.factor(sex),
    agegroup = as.factor(agegroup)
  )
```

```r
model <- lm(mortality_rate ~ MMWR_year + MMWR_week + agegroup + sex, data = weekly_counts_65p

summary(model)
```

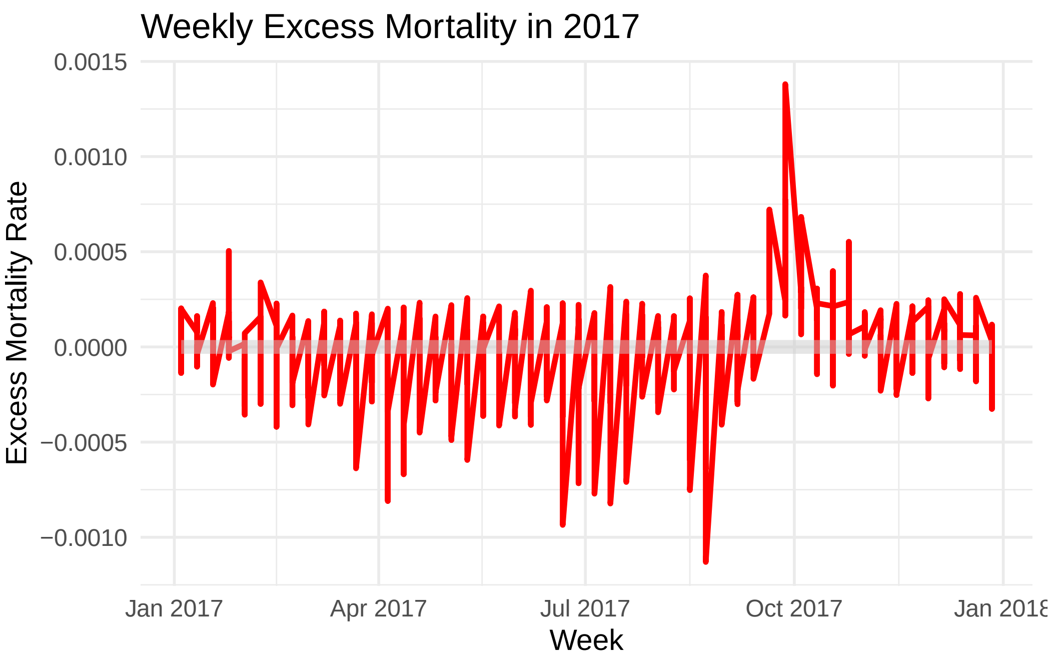
```
Call:
lm(formula = mortality_rate ~ MMWR_year + MMWR_week + agegroup +
    sex, data = weekly_counts_65plus)

Residuals:
       Min         1Q     Median         3Q        Max
-9.897e-04 -1.076e-04 -3.660e-06  9.799e-05  1.603e-03

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.626e-02  1.106e-03  32.780  < 2e-16 ***
MMWR_year      -1.795e-05  5.505e-07 -32.608  < 2e-16 ***
MMWR_week2      3.953e-05  2.431e-05   1.626 0.103888
MMWR_week3     -1.901e-06  2.431e-05  -0.078 0.937675
MMWR_week4     -4.742e-05  2.431e-05  -1.951 0.051117 .
MMWR_week5      5.360e-05  2.431e-05   2.205 0.027463 *
MMWR_week6      1.490e-05  2.431e-05   0.613 0.539876
MMWR_week7     -2.124e-06  2.431e-05  -0.087 0.930375
MMWR_week8     -2.004e-05  2.431e-05  -0.825 0.409581
MMWR_week9     -5.075e-07  2.431e-05  -0.021 0.983343
MMWR_week10    -2.358e-05  2.431e-05  -0.970 0.332107
MMWR_week11    -1.434e-05  2.431e-05  -0.590 0.555091
MMWR_week12    -4.792e-06  2.431e-05  -0.197 0.843715
MMWR_week13    -4.504e-05  2.431e-05  -1.853 0.063899 .
MMWR_week14    -4.629e-05  2.431e-05  -1.904 0.056902 .
MMWR_week15    -7.246e-05  2.431e-05  -2.981 0.002879 **
MMWR_week16    -6.808e-05  2.431e-05  -2.801 0.005106 **
MMWR_week17    -5.190e-05  2.431e-05  -2.135 0.032781 *
MMWR_week18    -8.338e-05  2.431e-05  -3.431 0.000605 ***
MMWR_week19    -9.207e-05  2.431e-05  -3.788 0.000153 ***
MMWR_week20    -5.910e-05  2.431e-05  -2.431 0.015065 *
MMWR_week21    -4.835e-05  2.431e-05  -1.989 0.046709 *
MMWR_week22    -7.125e-05  2.431e-05  -2.931 0.003386 **
MMWR_week23    -7.426e-05  2.431e-05  -3.055 0.002255 **
MMWR_week24    -5.534e-05  2.431e-05  -2.277 0.022826 *
MMWR_week25    -8.410e-05  2.431e-05  -3.460 0.000543 ***
MMWR_week26    -5.782e-05  2.431e-05  -2.379 0.017394 *
```

```
MMWR_week27    -8.853e-05  2.431e-05  -3.642 0.000272 ***
MMWR_week28    -1.026e-04  2.431e-05  -4.221 2.46e-05 ***
MMWR_week29    -1.200e-04  2.431e-05  -4.936 8.12e-07 ***
MMWR_week30    -8.438e-05  2.431e-05  -3.471 0.000520 ***
MMWR_week31    -1.010e-04  2.431e-05  -4.156 3.27e-05 ***
MMWR_week32    -6.238e-05  2.431e-05  -2.567 0.010289 *
MMWR_week33    -1.138e-04  2.431e-05  -4.680 2.92e-06 ***
MMWR_week34    -1.241e-04  2.431e-05  -5.105 3.38e-07 ***
MMWR_week35    -1.056e-04  2.431e-05  -4.344 1.42e-05 ***
MMWR_week36    -6.045e-05  2.431e-05  -2.487 0.012895 *
MMWR_week37    -9.220e-05  2.431e-05  -3.793 0.000150 ***
MMWR_week38    -8.363e-05  2.431e-05  -3.441 0.000583 ***
MMWR_week39    -1.023e-04  2.431e-05  -4.207 2.61e-05 ***
MMWR_week40    -1.093e-04  2.431e-05  -4.497 6.99e-06 ***
MMWR_week41    -7.848e-05  2.431e-05  -3.229 0.001248 **
MMWR_week42    -9.504e-05  2.431e-05  -3.910 9.30e-05 ***
MMWR_week43    -7.835e-05  2.431e-05  -3.223 0.001273 **
MMWR_week44    -6.464e-05  2.431e-05  -2.659 0.007847 **
MMWR_week45    -7.326e-05  2.431e-05  -3.014 0.002585 **
MMWR_week46    -5.911e-05  2.431e-05  -2.432 0.015043 *
MMWR_week47    -5.445e-05  2.431e-05  -2.240 0.025096 *
MMWR_week48    -5.402e-05  2.431e-05  -2.223 0.026269 *
MMWR_week49    -4.982e-05  2.431e-05  -2.050 0.040417 *
MMWR_week50    -2.266e-05  2.431e-05  -0.932 0.351145
MMWR_week51     2.479e-06  2.431e-05   0.102 0.918761
MMWR_week52     5.228e-05  2.431e-05   2.151 0.031528 *
MMWR_week53     7.910e-05  4.210e-05   1.879 0.060304 .
agegroup70-74   1.470e-04  7.523e-06  19.546  < 2e-16 ***
agegroup75-79   4.053e-04  7.523e-06  53.880  < 2e-16 ***
agegroup80-84   8.754e-04  7.523e-06 116.373  < 2e-16 ***
agegroup85-Inf  2.190e-03  7.523e-06 291.188  < 2e-16 ***
sexmale         3.003e-04  4.758e-06  63.110  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0002105 on 7771 degrees of freedom
Multiple R-squared:  0.9373,    Adjusted R-squared:  0.9368
F-statistic:  2001 on 58 and 7771 DF,  p-value: < 2.2e-16
```

16. Now obtain expected counts for the entire dataset, including 2017. Compute the difference between the observed count and expected count and plot the total excess death for each week. Construct a confidence interval for the excess mortality estimate for each week. Hint: use the `predict` function.

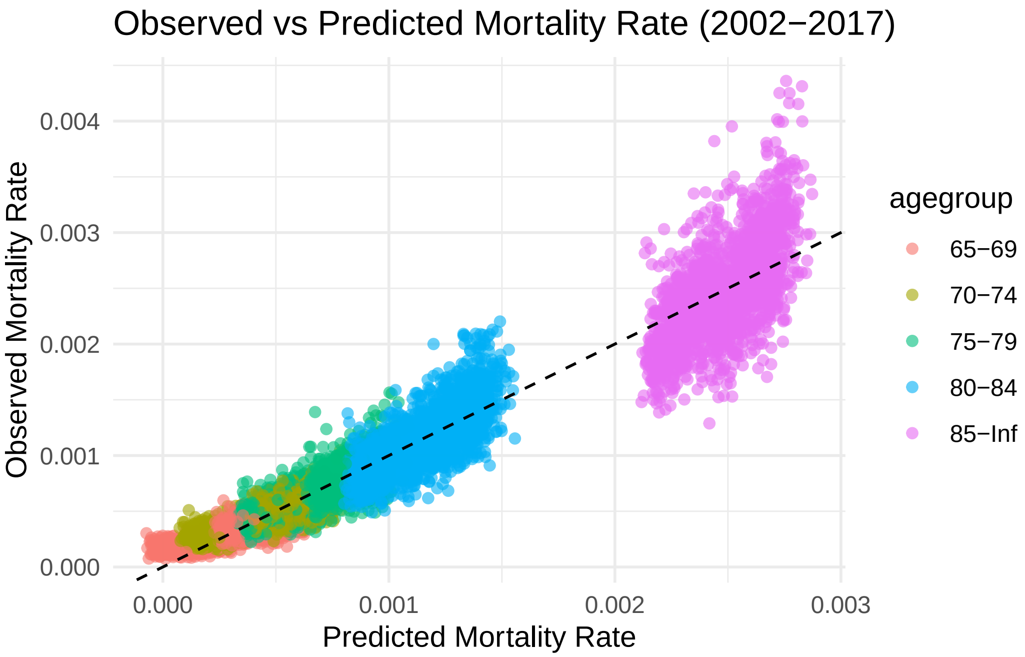
```
weekly_counts_2017 <- weekly_counts %>%
  filter(MMWR_year == 2017) %>%
  filter(agegroup %in% c("65-69", "70-74", "75-79", "80-84", "85-Inf")) %>%
  mutate(
    agegroup = factor(agegroup, levels = levels(weekly_counts$agegroup)),
    MMWR_week = factor(MMWR_week)
  )
predictions_2017 <- predict(model, newdata = weekly_counts_2017, interval = "confidence", lev

weekly_counts_2017 <- weekly_counts_2017 %>%
  mutate(
    predicted_rate = predictions_2017[, "fit"],
    lower_CI = predictions_2017[, "lwr"],
    upper_CI = predictions_2017[, "upr"],
    excess_mortality = mortality_rate - predicted_rate
  )

weekly_counts_2017 <- weekly_counts_2017 %>%
  filter(!is.na(excess_mortality))

ggplot(weekly_counts_2017, aes(x = week_start, y = excess_mortality)) +
  geom_line(color = "red", size = 1) +
  geom_ribbon(aes(ymin = lower_CI - predicted_rate, ymax = upper_CI - predicted_rate),
              fill = "gray80", alpha = 0.5) +
  labs(title = "Weekly Excess Mortality in 2017",
       x = "Week",
       y = "Excess Mortality Rate") +
  theme_minimal()
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

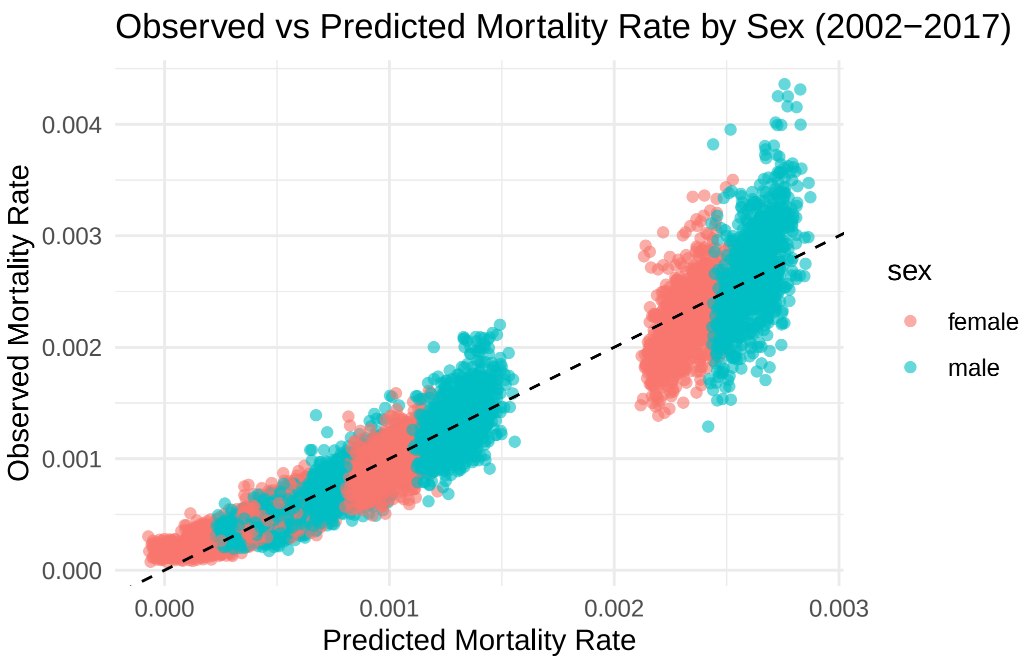


17. Finally, plot the observed rates and predicted rates from the model for each `agegroup` and `sex`. Comment on how well the model fits and what you might do differently.

```
weekly_counts_pre2017 <- weekly_counts %>%
  filter(MMWR_year >= 2002 & MMWR_year <= 2016) %>%
  filter(agegroup %in% c("65-69", "70-74", "75-79", "80-84", "85-Inf")) %>%
  mutate(
    agegroup = factor(agegroup, levels = levels(weekly_counts$agegroup)),
    MMWR_week = factor(MMWR_week)
  )
predictions_pre2017 <- predict(model, newdata = weekly_counts_pre2017, interval = "confidenc

weekly_counts_pre2017 <- weekly_counts_pre2017 %>%
  mutate(predicted_rate = predictions_pre2017[, "fit"])

all_data <- bind_rows(weekly_counts_pre2017, weekly_counts_2017)

all_data <- all_data %>%
  filter(!is.na(predicted_rate) & !is.na(mortality_rate))

ggplot(all_data, aes(x = predicted_rate, y = mortality_rate, color = agegroup)) +
  geom_point(alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  labs(title = "Observed vs Predicted Mortality Rate (2002-2017)",
       x = "Predicted Mortality Rate",
       y = "Observed Mortality Rate") +
  theme_minimal()
```



```
ggplot(all_data, aes(x = predicted_rate, y = mortality_rate, color = sex)) +
  geom_point(alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed") +
  labs(title = "Observed vs Predicted Mortality Rate by Sex (2002-2017)",
       x = "Predicted Mortality Rate",
       y = "Observed Mortality Rate") +
  theme_minimal()
```


Observed vs Predicted Mortality Rate by Sex (2002−2017)
Observed Mortality Rate
0.004
0.003
0.002
0.001
0.000
0.000
0.001
0.002
0.003
Predicted Mortality Rate
sex
female
male