Lab 7

Harry Potter Revenue

Assume alpha level is 0.05 for all tests

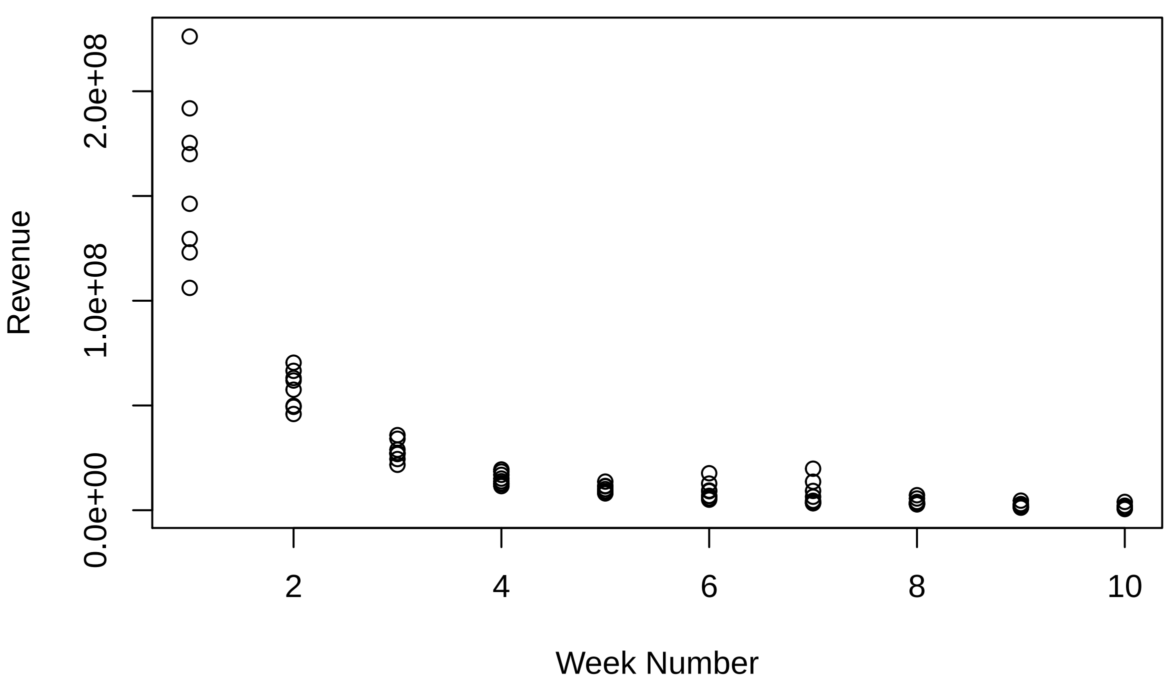
Models with polynomial terms

We will examine revenue from Harry Potter films over the first ten weeks in theaters. First, we'll read in the data from a csv file.

```
#Read in csv from current directory
harry <- read.csv('harrypotter.csv')
```

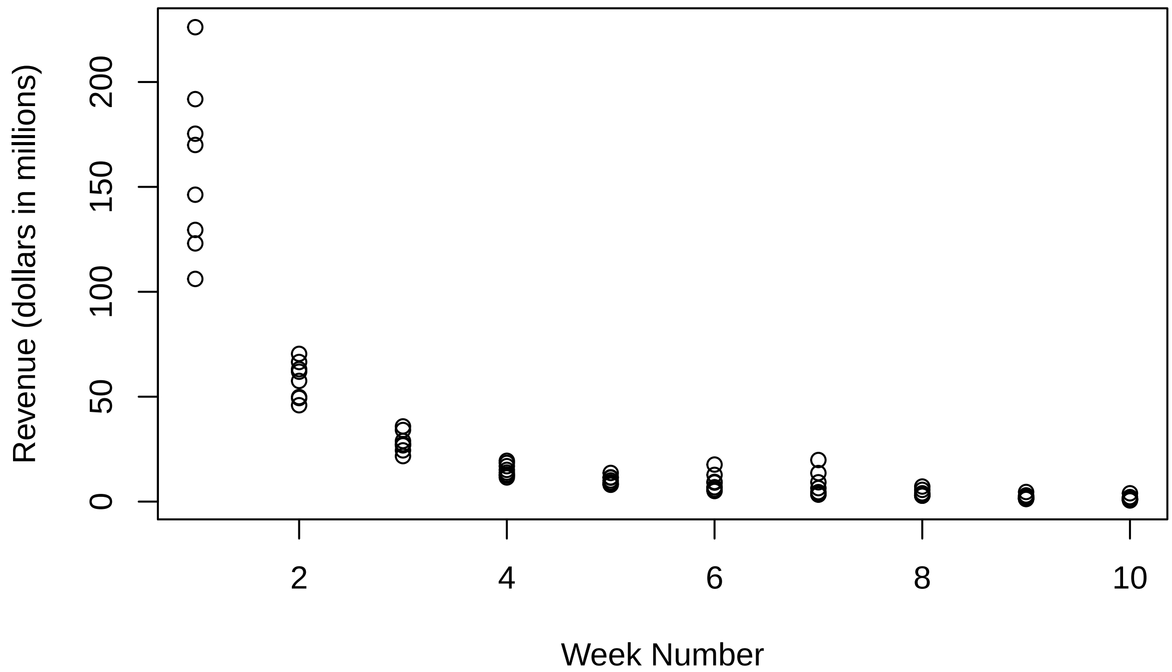
Take a look at the dataset use the View function or by clicking the dataframe icon next to 'harry' in your environment window. Now, let's plot revenue against week number to look at the shape of the relationship.

```
# Initial plot
plot(harry$weeknum, harry$revenue, xlab = "Week Number", ylab = "Revenue")
```




```
# Improve interpretability of data by changing the units on revenue to dollars in millions
# 1e6 means "1 * 10^6", or 1 million
harry$revenue <- harry$revenue / 1e6

# Plot again
plot(harry$weeknum, harry$revenue, xlab = "Week Number", ylab = "Revenue (dollars in millions)")
```

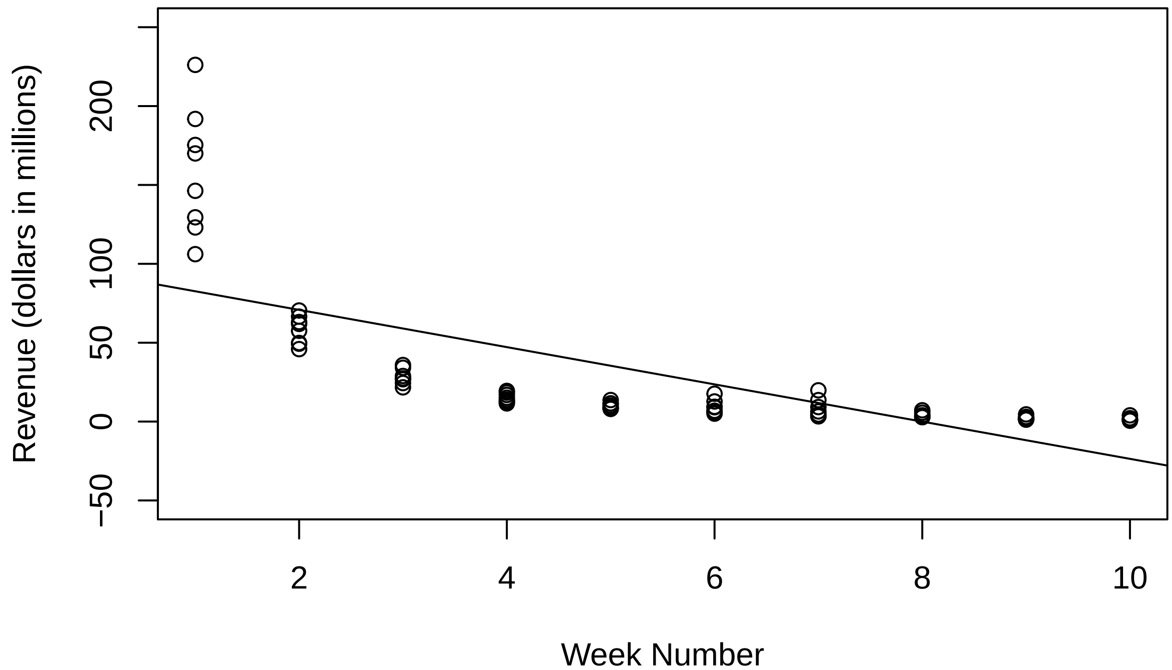


This relationship appears curved. Let's try a simple linear model first. Then we can compare it to a linear model with a polynomial term.

```r
model_simple <- lm(revenue ~ weeknum, data = harry)
summary(model_simple)
```

```
##
## Call:
## lm(formula = revenue ~ weeknum, data = harry)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.313 -24.881  -7.488  13.599 143.507
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   94.415      8.167  11.561  < 2e-16 ***
## weeknum      -11.805      1.316  -8.969 1.24e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.81 on 78 degrees of freedom
## Multiple R-squared:  0.5077, Adjusted R-squared:  0.5014
## F-statistic: 80.44 on 1 and 78 DF,  p-value: 1.241e-13
```

```r
# Take a look at a plot of the data with the fitted line
plot(harry$weeknum, harry$revenue,
     xlab = "Week Number", ylab = "Revenue (dollars in millions)",
     xlim = c(1, 10), ylim = c(-50, 250))
abline(model_simple)
```



According to this model, we expect revenue from Harry Potter movies to decrease by $_______ from week 4 to week 5 on average.

There are two ways we can answer this:

```
# Option 1: Use coefficient from the model
coef(model_simple)[2]
```

```
##   weeknum
## -11.80474
```

```
# Option 2: Use the predict function
rev_wk4and5 <- predict(model_simple, data.frame(weeknum = 4:5))
rev_wk4and5
```

```
##        1        2
## 47.19623 35.39149
```

```
diff(rev_wk4and5)
```

```
##         2
## -11.80474
```

Let's see if we can improve our model by adding a quadratic term. There are two ways to specify the squared term.

```
# Option 1: Use the I() function. This inhibits the interpretation of the "^" symbol as a formula opera
model_quad <- lm(revenue ~ weeknum + I(weeknum^2), data = harry)
summary(model_quad)
```

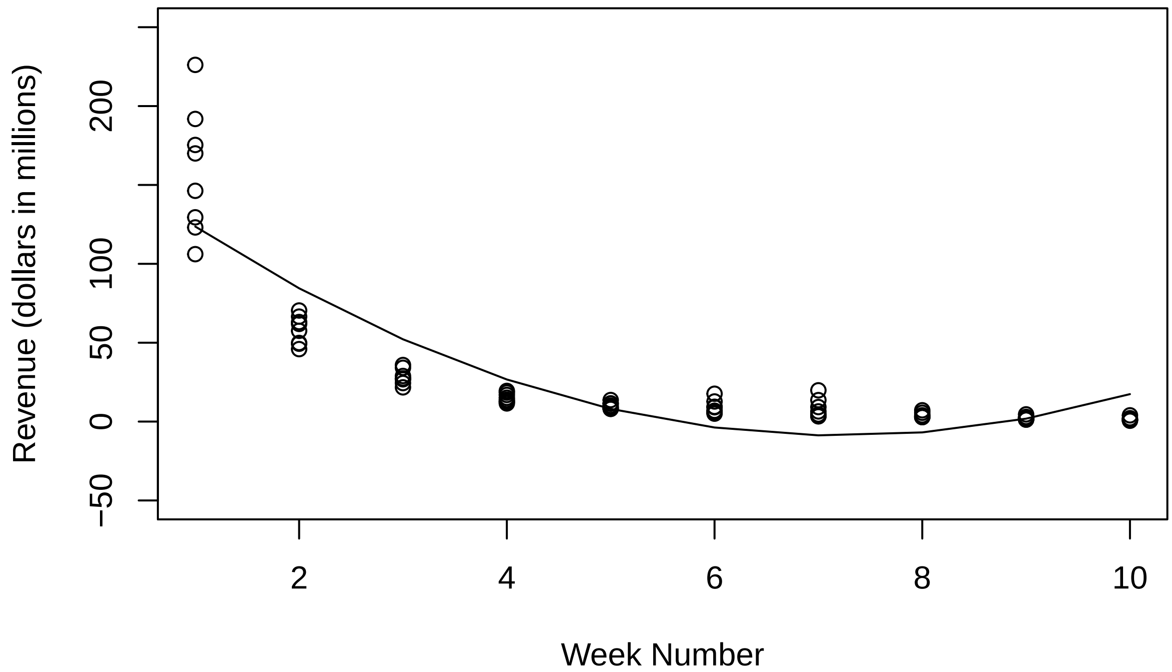
```
##
## Call:
## lm(formula = revenue ~ weeknum + I(weeknum^2), data = harry)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -38.550 -16.224   0.151  10.655 102.487
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  169.6182     9.4573  17.935   <2e-16 ***
## weeknum      -49.4063     3.9497 -12.509   <2e-16 ***
## I(weeknum^2)   3.4183     0.3499   9.769    4e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.74 on 77 degrees of freedom
## Multiple R-squared:  0.7801, Adjusted R-squared:  0.7744
## F-statistic: 136.6 on 2 and 77 DF,  p-value: < 2.2e-16
```

```r
# Option 2: Create a new variable for the squared term first. Then enter into the model as usual.
harry$weeknum2 <- (harry$weeknum)^2
model_q2 <- lm(revenue ~ weeknum + weeknum2, data = harry)
summary(model_q2)
```

```
##
## Call:
## lm(formula = revenue ~ weeknum + weeknum2, data = harry)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.550 -16.224   0.151  10.655 102.487
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 169.6182     9.4573  17.935   <2e-16 ***
## weeknum     -49.4063     3.9497 -12.509   <2e-16 ***
## weeknum2      3.4183     0.3499   9.769    4e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.74 on 77 degrees of freedom
## Multiple R-squared:  0.7801, Adjusted R-squared:  0.7744
## F-statistic: 136.6 on 2 and 77 DF,  p-value: < 2.2e-16
```

```r
# Take a look at a plot of the data with the fitted line
plot(harry$weeknum, harry$revenue,
     xlab = "Week Number", ylab = "Revenue (dollars in millions)",
     xlim = c(1, 10), ylim = c(-50, 250))
weeknums = c(1:10)
revs <- predict(model_quad, data.frame(weeknum = 1:10))
lines(weeknums, revs)
```



Let's try to answer the same question as before, but with our new model:
According to this model, we expect revenue from Harry Potter movies to decrease by $________ from week 4 to week 5 on average.

This time we can't simply use the coefficient; its meaning is different because the squared term will also change as we increase week number. We must use the predict function here.

```
rev_wk4and5_quad <- predict(model_quad, data.frame(weeknum = 4:5))
rev_wk4and5_quad
```

```
##         1         2
## 26.686296  8.044916
```

```
diff(rev_wk4and5_quad)
```

```
##         2
## -18.64138
```

According to the quadratic model, at about what week do we expect the change in revenue to begin increasing from week to week?

```
### Using model_quad
# Use the model to predict each week's revenue
predict(model_quad, data.frame(weeknum = 1:10))
```

```
##          1          2          3          4          5          6
## 123.630294  84.478985  52.164318  26.686296   8.044916  -3.759820
##          7          8          9         10
##  -8.727914  -6.859364   1.845830  17.387666
```

```
# Check difference week-to-week
diff(predict(model_quad, data.frame(weeknum = 1:10)))
```

```
##          2          3          4          5          6          7
## -39.151309 -32.314666 -25.478023 -18.641380 -11.804736  -4.968093
##          8          9         10
```

```
##   1.868550   8.705193  15.541837
```

```r
### Using model_q2 and squared term variable
# Create dataframe of weeks 1-10 and their square
newdata <- data.frame(weeknum = 1:10, weeknum2 = (1:10)^2)

# Use the model to predict each week's revenue
predict(model_q2, newdata)
```

```
##          1          2          3          4          5          6 
## 123.630294  84.478985  52.164318  26.686296   8.044916  -3.759820 
##          7          8          9         10 
##  -8.727914  -6.859364   1.845830  17.387666
```

```r
# Check difference week-to-week
diff(predict(model_q2, newdata))
```

```
##          2          3          4          5          6          7 
## -39.151309 -32.314666 -25.478023 -18.641380 -11.804736  -4.968093 
##          8          9         10 
##   1.868550   8.705193  15.541837
```

F Tests

Which model fits the data better? We can assess model fit using an F test. The F test is a way to compare nested models using the Residual Sum of Squares (RSS) of each model to compute an F ratio. R computes an F test every time you run the lm command. In this case, R compares your current model to a model with no independent variables (often called a null model). The null model just gives you the average outcome, so you're essentially comparing your fitted line to a horizontal line at the average outcome. This F test is sometimes called the "global F test" or "test of model significance." The null hypothesis of this test is that all the betas are zero. The alternative is that at least one is not zero. That is, we are testing all the beta coefficients jointly rather than individually.

```r
# Create null model
# Note that there's no F test at the bottom!
model_null <- lm(revenue ~ 1, data = harry)
summary(model_null)
```

```
## 
## Call:
## lm(formula = revenue ~ 1, data = harry)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -28.942 -25.956 -19.763  -2.273 196.628 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   29.489      5.354   5.508 4.38e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 47.89 on 79 degrees of freedom
```

```r
# Compare coefficient to mean revenue
mean(harry$revenue)
```

```
## [1] 29.48912
```

```r
# Compare F test from output to F test of simple vs null
summary(model_simple)
```

```
##
## Call:
## lm(formula = revenue ~ weeknum, data = harry)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.313 -24.881  -7.488  13.599 143.507
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   94.415      8.167  11.561  < 2e-16 ***
## weeknum      -11.805      1.316  -8.969 1.24e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 33.81 on 78 degrees of freedom
## Multiple R-squared:  0.5077, Adjusted R-squared:  0.5014
## F-statistic: 80.44 on 1 and 78 DF,  p-value: 1.241e-13
```

```r
anova(model_null, model_simple)
```

```
## Analysis of Variance Table
##
## Model 1: revenue ~ 1
## Model 2: revenue ~ weeknum
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     79 181157
## 2     78  89185  1     91972 80.438 1.241e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With the F test, model fit is a measured using RSS. This is simply a transformation of the residuals, or how much the predicted outcome differs from the true outcome at each point. The smaller the difference, the better the fit of the line to the data. Within the F statistic, the RSS of each model is compared. If the new model reduces the RSS without adding too many degrees of freedom, the F statistic may be large enough to reject the null that the models equally fit the data. Note that if your models differ by more than one term, you are testing if at least one of them improves the model (but not necessarily all of them are significant).

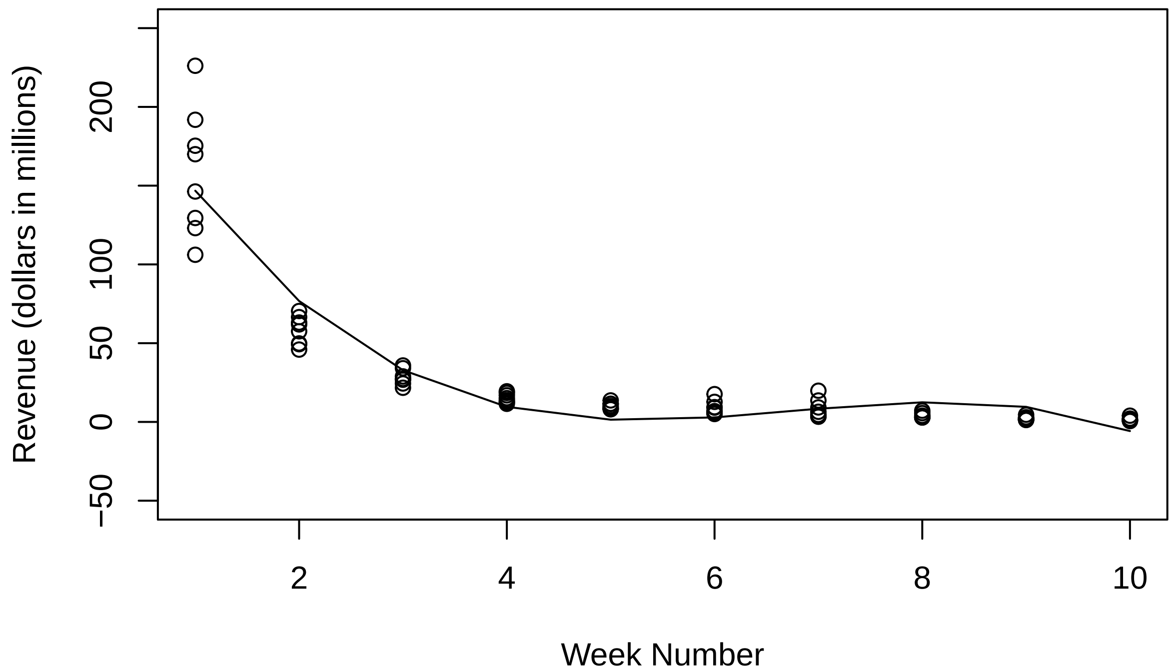
```r
# Compare simple with added quadratic term
anova(model_simple, model_quad)
```

```
## Analysis of Variance Table
##
## Model 1: revenue ~ weeknum
## Model 2: revenue ~ weeknum + I(weeknum^2)
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1     78 89185
## 2     77 39828  1     49357 95.424 4.002e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fitting a cubic term (just for fun!)
model_cube <- lm(revenue ~ weeknum + I(weeknum^2) + I(weeknum^3), data = harry)
summary(model_cube)
```

```
##
## Call:
## lm(formula = revenue ~ weeknum + I(weeknum^2) + I(weeknum^3),
##     data = harry)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.670  -8.198   1.044   6.913  79.316
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   248.5096    10.8311  22.944  < 2e-16 ***
## weeknum      -119.3787     8.1185 -14.705  < 2e-16 ***
## I(weeknum^2)   18.5897     1.6746  11.101  < 2e-16 ***
## I(weeknum^3)   -0.9195     0.1004  -9.157 6.69e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.78 on 76 degrees of freedom
## Multiple R-squared:  0.8955, Adjusted R-squared:  0.8913
## F-statistic:   217 on 3 and 76 DF,  p-value: < 2.2e-16
```

```r
plot(harry$weeknum, harry$revenue,
     xlab = "Week Number", ylab = "Revenue (dollars in millions)",
     xlim = c(1, 10), ylim = c(-50, 250))
weeknums = c(1:10)
revs <- predict(model_cube, data.frame(weeknum = 1:10))
lines(weeknums, revs)
```



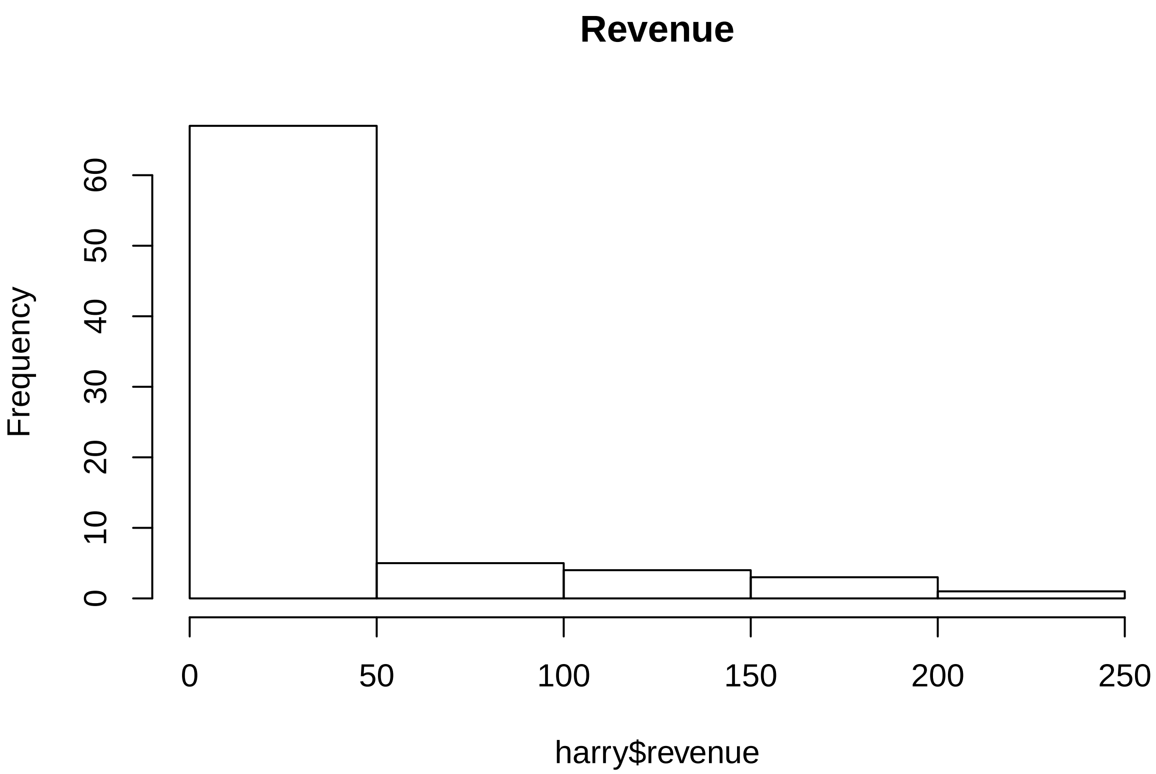
```r
anova(model_simple, model_quad, model_cube)
```

```
## Analysis of Variance Table
##
## Model 1: revenue ~ weeknum
## Model 2: revenue ~ weeknum + I(weeknum^2)
## Model 3: revenue ~ weeknum + I(weeknum^2) + I(weeknum^3)
##   Res.Df   RSS Df Sum of Sq       F    Pr(>F)
## 1     78 89185
## 2     77 39828  1     49357 198.092 < 2.2e-16 ***
## 3     76 18936  1     20891  83.846 6.692e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Another option: log transform

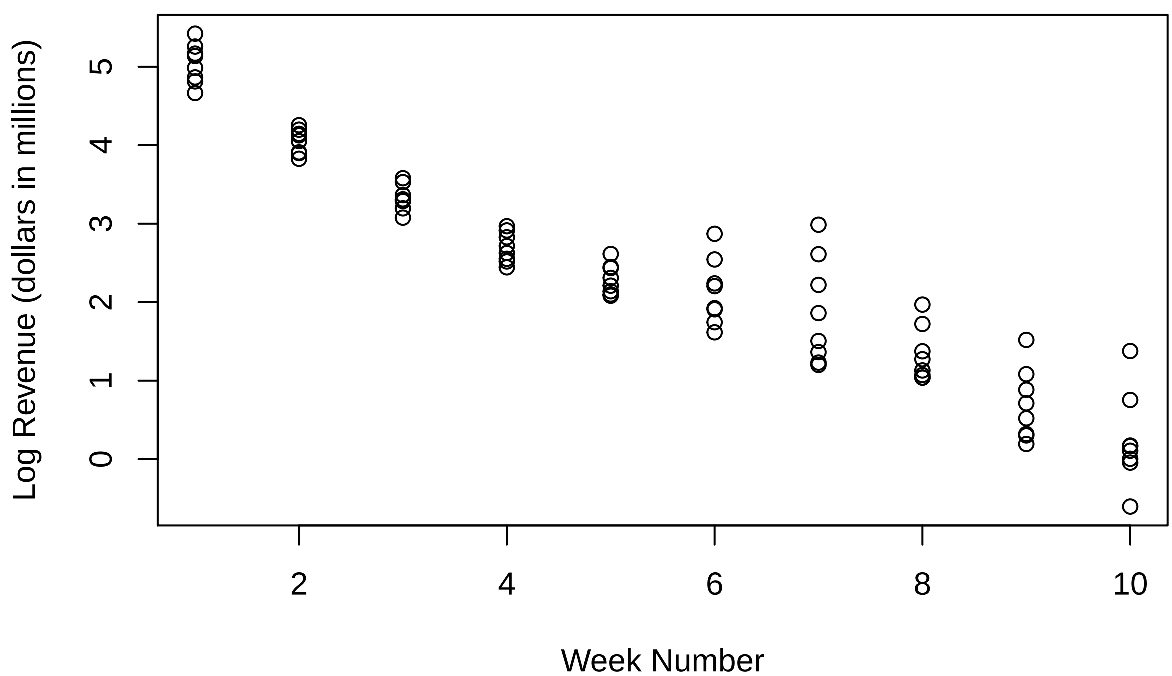
Note that revenue is mostly very small, with a few large outliers. By taking log(revenue), the distribution looks more normal and the relationship between revenue and week number looks more linear.

```r
hist(harry$revenue, main="Revenue")
```




```r
harry$logrev = log(harry$revenue)
hist(harry$logrev, main="Log Revenue")
```


```
plot(harry$weeknum, harry$logrev,
     xlab = "Week Number", ylab = "Log Revenue (dollars in millions)",
     xlim = c(1, 10))
```



We can now fit the model predicting log revenue. Remember that the model is now:

$$log(revenue) = \beta_0 + \beta_1 \cdot weeknum$$

We can exponentiate both sides of this equation to get the following equation where the left side is then simply the revenue.

$$e^{\log(revenue)} = e^{\beta_0 + \beta_1 \cdot weeknum}$$

$$revenue = e^{(\beta_0)} \cdot e^{(\beta_1 \cdot weeknum)}$$

So, if we compare revenue at weeknum=x and weeknum=x+1, we get:

$$e^{\beta_0} \cdot e^{\beta_1 \cdot (x+1)} / e^{\beta_0} \cdot e^{\beta_1 \cdot x} = e^{\beta_1}$$

Therefore, we expect the revenue in a particular week to be e^{β_1} times higher than in the previous week. And e^{β_0} is now the expected revenue at week 0.

```
lm_logtrans = lm(logrev ~ weeknum, data = harry)
summary(lm_logtrans)
```

```
##
## Call:
## lm(formula = logrev ~ weeknum, data = harry)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8074 -0.2863 -0.1232  0.2056  1.3406
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.01161    0.10812   46.35   <2e-16 ***
## weeknum     -0.48081    0.01743  -27.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4477 on 78 degrees of freedom
## Multiple R-squared:  0.9071, Adjusted R-squared:  0.9059
## F-statistic: 761.3 on 1 and 78 DF,  p-value: < 2.2e-16
```

```
exp(lm_logtrans$coefficients[1]) #e^b0
```

```
## (Intercept)
##    150.1464
```

```
exp(lm_logtrans$coefficients[2]) #e^b1
```

```
##   weeknum
## 0.6182853
```

Wald test: NOTE: THIS IS NOT COVERED IN CLASS/HW, and is therefore optional

There are many types of Wald tests. Technically, all of the t tests on the coefficients are Wald tests. Also, the F test is directly related to the Wald test, as W = F*q, where q is the number of restrictions. As mentioned in the lecture slides, the Wald test using the W statistic and chi-square distribution is best with larger sample sizes. The Wald test using W has the advantage of only needing one model rather than two to compare.

```
#install.packages("lmtest")
require(lmtest)
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
waldtest(model_quad,model_cube)
```

```
## Wald test
##
## Model 1: revenue ~ weeknum + I(weeknum^2)
## Model 2: revenue ~ weeknum + I(weeknum^2) + I(weeknum^3)
##   Res.Df Df      F    Pr(>F)
## 1     77
## 2     76  1 83.846 6.692e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This wald test function tells us which of our models fit the data better and in this case, the model with the cubed term is the better model.

Wald test matrix algebra

```r
# Subset to just the first three films
harry_four <- harry[harry$film < 5, ]

# Create dummy variables for film number
harry_four$sorcerer <- as.numeric(harry_four$film == 1)
harry_four$chamber <- as.numeric(harry_four$film == 2)
harry_four$azkaban <- as.numeric(harry_four$film == 3)
harry_four$goblet <- as.numeric(harry_four$film == 4)

# Run linear model predicting revenue from film name dummy variables with sorcerer's stone as the refer
model_film <- lm(revenue ~ chamber + azkaban + goblet, data = harry_four)
summary(model_film)
```

```
##
## Call:
## lm(formula = revenue ~ chamber + azkaban + goblet, data = harry_four)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.463 -22.028 -15.789  -0.889 117.707
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   31.017     12.379   2.506   0.0169 *
## chamber       -5.205     17.506  -0.297   0.7679
## azkaban       -6.438     17.506  -0.368   0.7152
## goblet        -2.441     17.506  -0.139   0.8899
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 39.15 on 36 degrees of freedom
## Multiple R-squared:  0.004495,   Adjusted R-squared:  -0.07846
## F-statistic: 0.05418 on 3 and 36 DF,  p-value: 0.9831
```

Construct a Wald test to test whether there is a statistically significant difference in revenue between azkaban and goblet films. Report the p-value for this test.

This is the same as asking, "does beta_azkaban = beta_goblet?" Further, we can subtract beta_goblet from both sides to get, "does beta_azkaban - beta_goblet = 0?"

Since exactly what we test with the Wald test changes in different contexts, we use the null hypothesis theta = 0, and we change what theta is (e.g. all the betas = 0, beta_1 - beta2 = 0, etc.). In this case, theta is beta_azkaban - beta_goblet. Now we use a matrix D multiplied by the betahats to express this. Then we compute W using the formulas from class.

```r
# Assign the coefficients from the model to betahat
betahat <- coef(model_film)
betahat
```

```
## (Intercept)     chamber     azkaban      goblet
##   31.017160   -5.205196   -6.437882   -2.440869
```

```r
# Create the D matrix as an empty matrix
Dmat <- matrix(0, 1, ncol=4)

# Fill D matrix so that when we multiply D by betahat we get azkaban - goblet, which is the third coeff
Dmat[1,] <- c(0, 0, 1, -1)

# Create thetahat from D times betahat
thetahat <- Dmat %*% betahat
thetahat
```

```
##           [,1]
## [1,] -3.997012
```

```r
# We need to find the covariance of thetahat
theta.cov <- Dmat %*% vcov(model_film) %*% t(Dmat)
theta.cov
```

```
##        [,1]
## [1,] 306.47
```

```r
# Now we can get the W statistic
W <- t(thetahat) %*% solve(theta.cov) %*% thetahat
W
```

```
##            [,1]
## [1,] 0.05212943
```

```r
# Now we compare our W to a chi-square distribution to see if it is unlikely to get a W of this value g
pchisq(W, nrow(Dmat), lower.tail=FALSE)
```

```
##           [,1]
## [1,] 0.8193985
```

We get a p-value of 0.819. This means that if the null is true (beta_azkaban = beta_goblet), the probability of obtaining a W at least as extreme as our W of 0.052 is about 81.9%. This means our W seems very likely given the null is true, so we fail to reject the null. We do not have evidence that there is a statistically significant difference in revenue between the azkaban and goblet films at the 5% significance level.

```r
# Compare F to W testing if all betas equal 0 for film name
Dmat <- matrix(0, 3, 4)

# Fill D matrix so that when we multiply D by betahat we get azkaban - goblet, which is the third coeff
Dmat[1,] <- c(0, 1, 0, 0)
Dmat[2,] <- c(0, 0, 1, 0)
Dmat[3,] <- c(0, 0, 0, 1)

# Create thetahat from D times betahat
thetahat <- Dmat %*% betahat
thetahat
```

```
##           [,1]
## [1,] -5.205196
## [2,] -6.437882
## [3,] -2.440869
```

```r
# We need to find the covariance of thetahat
theta.cov <- Dmat %*% vcov(model_film) %*% t(Dmat)
theta.cov
```

```
##         [,1]    [,2]    [,3]
## [1,] 306.470 153.235 153.235
## [2,] 153.235 306.470 153.235
## [3,] 153.235 153.235 306.470
```

```r
# Now we can get the W statistic
W <- t(thetahat) %*% solve(theta.cov) %*% thetahat
W
```

```
##           [,1]
## [1,] 0.1625532
```

```r
# Now we compare our W to a chi-square distribution to see if it is unlikely to get a W of this value g
pchisq(W, nrow(Dmat), lower.tail=FALSE)
```

```
##           [,1]
## [1,] 0.9833953
```

```r
# The F statistic from earlier
Fstat <- summary(model_film)$fstatistic[1]

# Check if F*3 equals W. By adding 3 coefficients to the model, we added 3 restrictions (q = 3).
Fstat
```

```
##     value
## 0.0541844
```

```r
Fstat*3
```

```
##     value
## 0.1625532
```

```r
W
```

```
##           [,1]
## [1,] 0.1625532
```