Unit 9 Homework: Large-Sample Regression Theory

w203: Statistics for Data Science

What Makes a Successful Video Game?

The file `video_games.csv` contains data on 1212 video games that were on sold in 2011. It was compiled by Joe Cox, an economist at the University of Portsmouth.

Three key variables are as follows:

Variable	Meaning
Metrics.Sales	The total sales, measured in millions of dollars.
Metrics.Review.Score	Metacritic review score, an indicator of quality, out of 100.
Length.Completionists.Average	The mean time that players reported completing everything in the game, in hours.

You can find an explanation of other variables at https://think.cs.vt.edu/corgis/csv/video_games/.

You want to fit a regression predicting `Metrics.Sales`, with `Metrics.Review.Score` and `Length.Completionists.Average` as predictors.

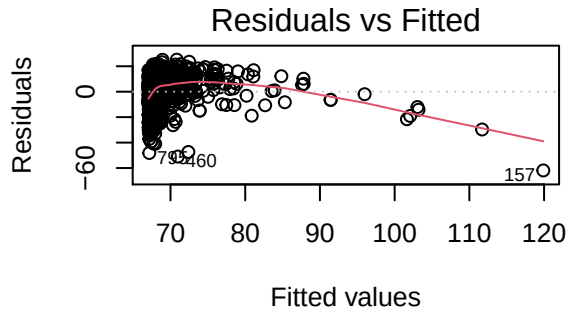
0. Rename the variables that you are going to use to something sensible – variable names that have both periods and capital letters are not sensible. :fire: Better would be, for example changing `Metrics.Sales` to just `sales`.

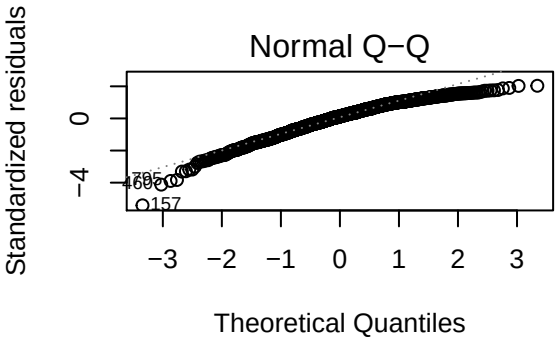
1. Examining the data, and using your background knowledge, evaluate the assumptions of the large-sample linear model.

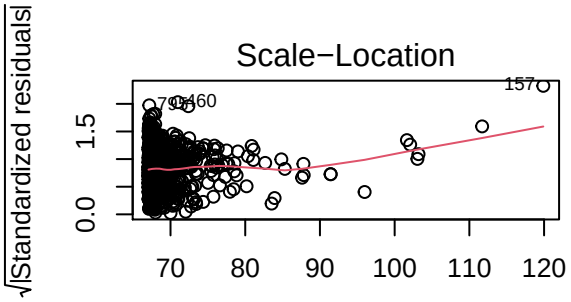
```
summary(df)
```

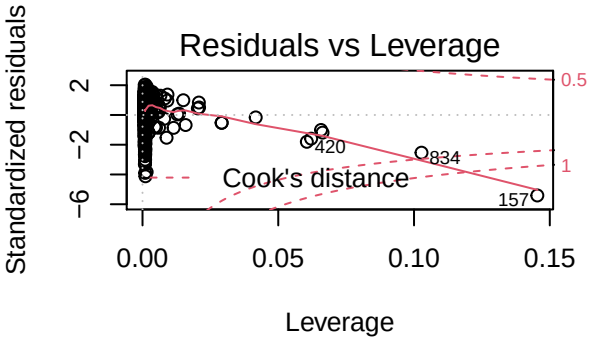
```
##      sales             score           length
##  Min.   : 0.0100   Min.   :19.00   Min.   :  0.00
##  1st Qu.: 0.0900   1st Qu.:60.00   1st Qu.:  0.00
##  Median : 0.2100   Median :70.00   Median :  6.00
##  Mean   : 0.5032   Mean   :68.83   Mean   : 19.81
##  3rd Qu.: 0.4600   3rd Qu.:79.00   3rd Qu.: 21.55
##  Max.   :14.6600   Max.   :98.00   Max.   :683.13
```

```
fit <- lm(df$score ~ df$sales)
par(mfrow=c(2,2))
plot (fit)
```

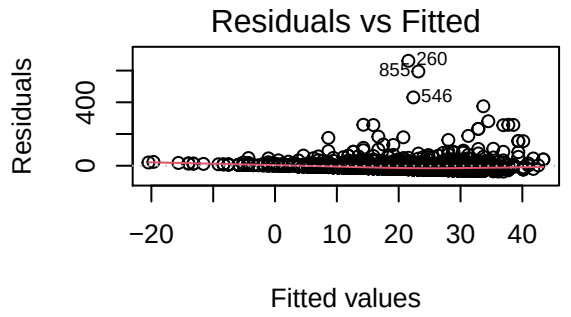


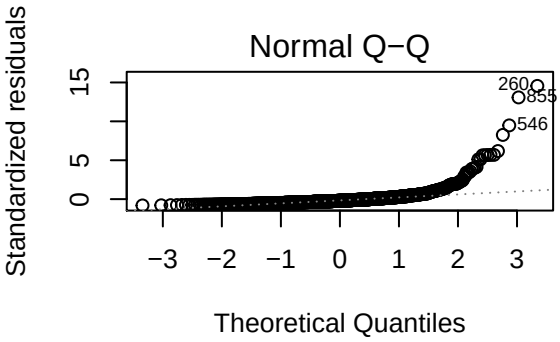


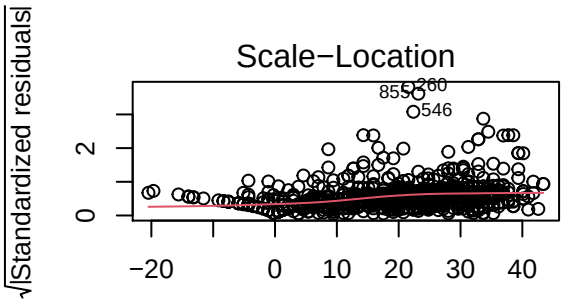


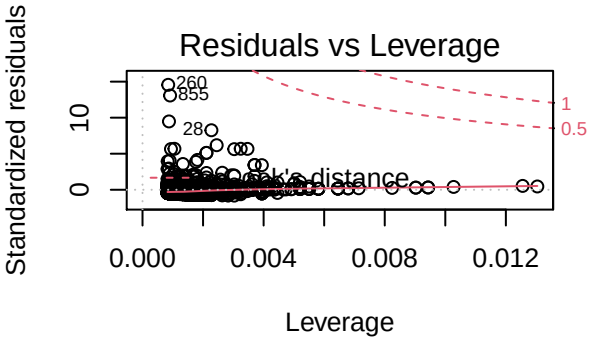



```
fit <- lm(df$length ~ df$score)
par(mfrow=c(2,2))
plot (fit)
```

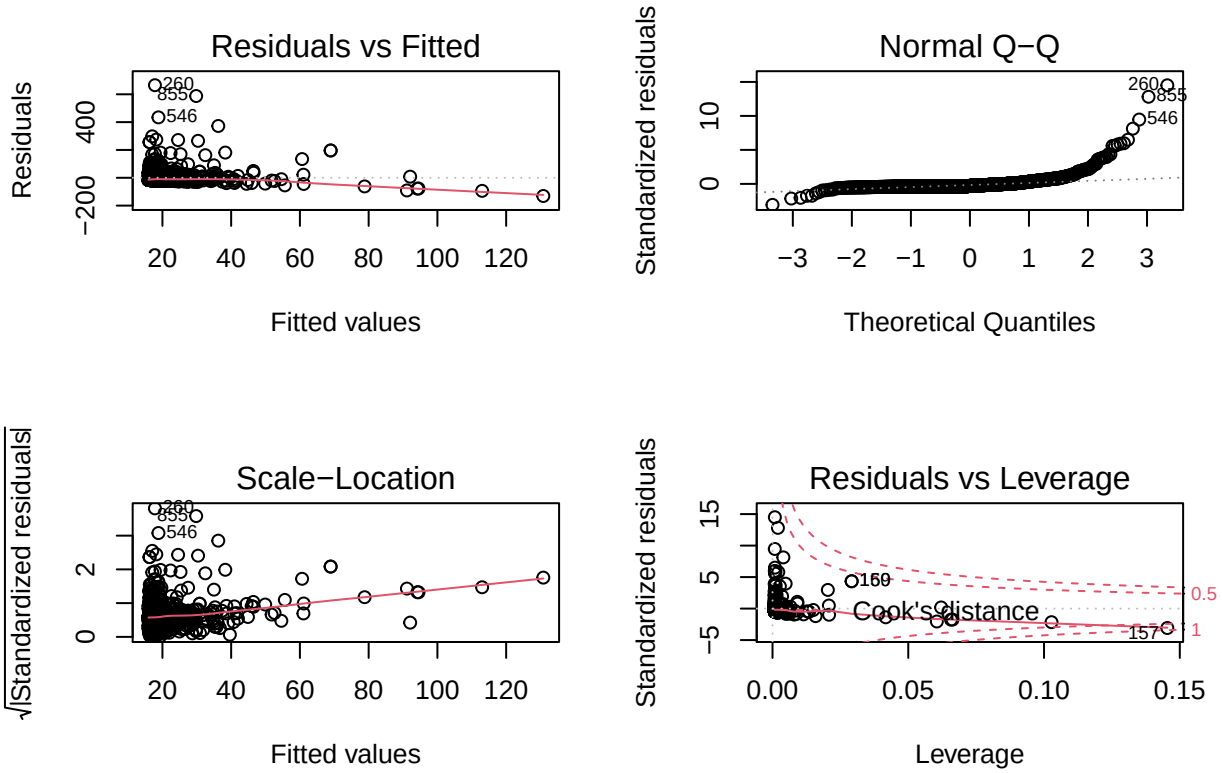








```
fit <- lm(df$length ~ df$sales)
par(mfrow=c(2,2))
plot (fit)
```



2. Whether you consider the large-sample linear model sufficiently valid or not, proceed to fit the linear model using `lm()`.

```
library(gvlma)

fit <- lm(df$sales ~ df$score + df$length)
gvmodel <- gvlma(fit)
summary(gvmodel)
```

```
##
## Call:
## lm(formula = df$sales ~ df$score + df$length)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1125 -0.4223 -0.1852  0.0918 14.4534
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.0919732  0.1592648  -6.856 1.12e-11 ***
## df$score     0.0223899  0.0023092   9.696  < 2e-16 ***
## df$length    0.0027297  0.0006416   4.255 2.25e-05 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.015 on 1209 degrees of freedom
## Multiple R-squared:  0.1022, Adjusted R-squared:  0.1007
## F-statistic: 68.79 on 2 and 1209 DF,  p-value: < 2.2e-16
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = fit)
##
##                       Value   p-value                   Decision
## Global Stat        262748.77 0.000e+00 Assumptions NOT satisfied!
## Skewness            10030.69 0.000e+00 Assumptions NOT satisfied!
## Kurtosis           252418.54 0.000e+00 Assumptions NOT satisfied!
## Link Function          49.76 1.738e-12 Assumptions NOT satisfied!
## Heteroscedasticity    249.78 0.000e+00 Assumptions NOT satisfied!
```

```
library(car)
```

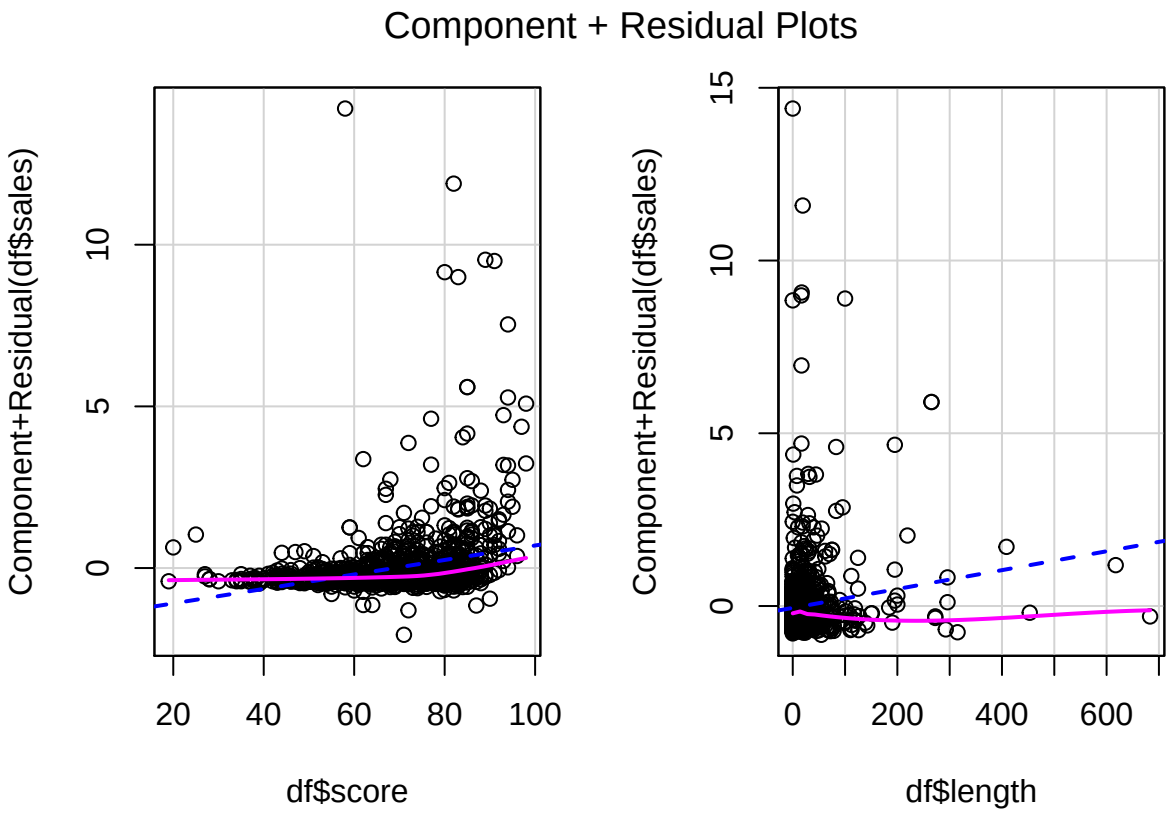
```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

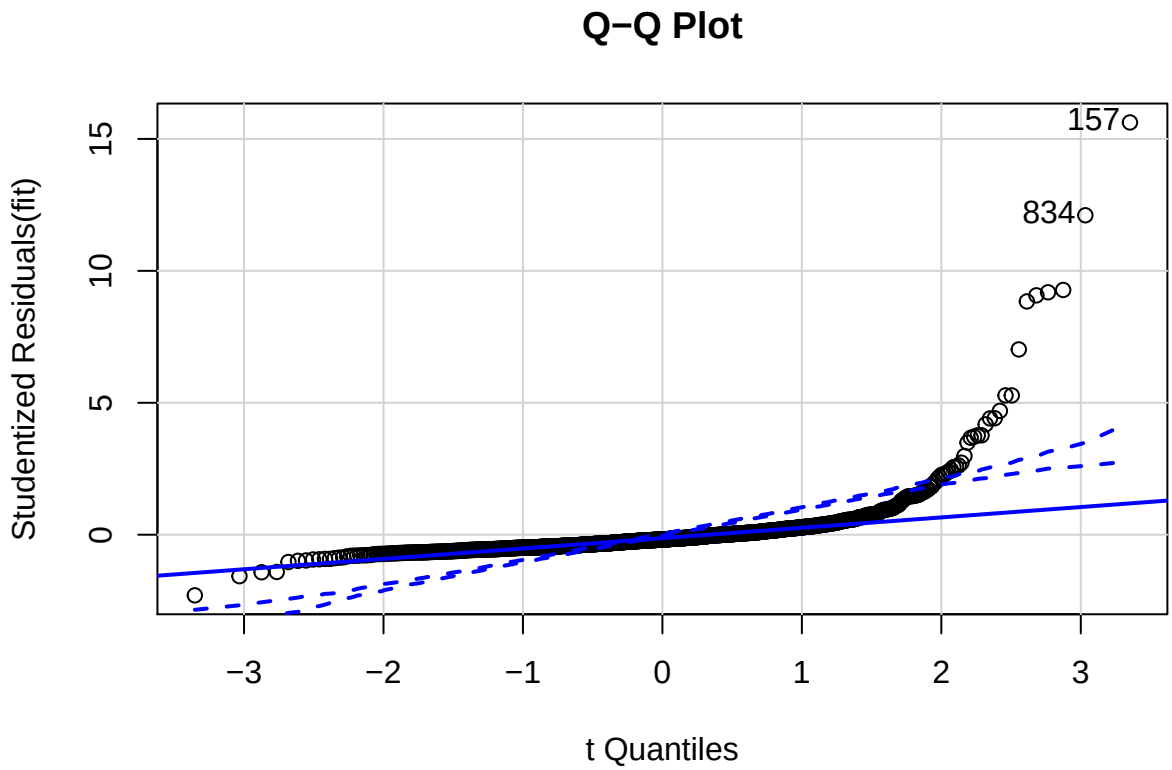
```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
crPlots(fit)
```




```
qqPlot(fit, labels = row.names(df), id.method = 'identify', simulate = TRUE, main = 'Q-Q Plot')
```




```
## [1] 157 834
```

```
ncvTest(fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 386.6485, Df = 1, p = < 2.22e-16
```

```
durbinWatsonTest(fit)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.6379745     0.7124182       0
##  Alternative hypothesis: rho != 0
```

```
vif(fit)
```

```
##  df$score df$length
##  1.053108  1.053108
```

3. Examine the coefficient for Metrics.Review.Score and give an interpretation of what it means.

```
scoresales.lm <- lm(score ~ sales, data=df)
scorelength.lm <- lm(score ~ length, data=df)
summary(scoresales.lm)$r.squared
```

```
## [1] 0.0887267
```

```
summary(scorelength.lm)$r.squared
```

```
## [1] 0.05042935
```

The most common interpretation of the coefficient of determination is how well the regression model fits the observed data. For example, a coefficient of determination of 60% shows that 60% of the data fit the regression model. Generally, a higher coefficient indicates a better fit for the model.

However, it is not always the case that a high r-squared is good for the regression model. The quality of the coefficient depends on several factors, including the units of measure of the variables, the nature of the variables employed in the model, and the applied data transformation. Thus, sometimes, a high coefficient can indicate issues with the regression model.

No universal rule governs how to incorporate the coefficient of determination in the assessment of a model. The context in which the forecast or the experiment is based is extremely important, and in different scenarios, the insights from the statistical metric can vary.

```
bptest(scoresales.lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  scoresales.lm
## BP = 41.037, df = 1, p-value = 1.494e-10
```

```
bptest(scorelength.lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  scorelength.lm
## BP = 14.348, df = 1, p-value = 0.0001519
```

4. Perform a hypothesis test to assess whether video game quality has a relationship with total sales. Please use `vcovHC` from the `sandwich` package with the default options ("HC3") to compute robust standard errors. To conduct the test, use `coeftest` from the `lmtest` package.

```
library(lmtest)
m1 <- lm(df$score ~ df$sales + df$length)
bptest(m1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  m1
## BP = 27.841, df = 2, p-value = 9.004e-07
```

```
#ptest(m1, studentsize=FALSE)


m2 <- lm(df$sales ~ df$score + df$length)
bptest(m2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  m2
## BP = 10.637, df = 2, p-value = 0.0049
```

```
#ptest(m1, studentsize=FALSE)


m3 <- lm(df$length ~ df$sales + df$score)
bptest(m3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  m3
## BP = 7.9776, df = 2, p-value = 0.01852
```

```
#ptest(m1, studentsize=FALSE)
```

```r
library(sandwich)
#summary(m1)
#NeweyWest(m1)

#result1<-coeftest(m1, vcov = NeweyWest(m1))
result2<-coeftest(m1, vcov. = vcovHC, type = "HC3")

print(result2)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) 66.234573   0.495075 133.7871 < 2.2e-16 ***
## df$sales     3.222391   0.800116   4.0274 5.991e-05 ***
## df$length    0.049092   0.014227   3.4506 0.0005786 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(m1)
```

```
##
## Call:
## lm(formula = df$score ~ df$sales + df$length)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.475  -7.762   1.877   9.363  24.990
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 66.234573   0.404867 163.596  < 2e-16 ***
## df$sales     3.222391   0.332345   9.696  < 2e-16 ***
## df$length    0.049092   0.007624   6.439 1.73e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.17 on 1209 degrees of freedom
## Multiple R-squared:  0.1189, Adjusted R-squared:  0.1175
## F-statistic: 81.61 on 2 and 1209 DF,  p-value: < 2.2e-16
```

```r
#m1 <- lm(df$score ~ df$sales)

#rhs<-c(0,1)
#vcovhc
```

5. **Optional:** Open the attached paper by Joe Cox, and read section 3. Which assumption did the author focus on, and why do you think that is?

Note: Maximum score on any homework is 100%