R Notebook

The following is your first chunk to start with. Remember, you can add chunks using the menu above (Insert -> R) or using the keyboard shortcut Ctrl+Alt+I. A good practice is to use different code chunks to answer different questions. You can delete this comment if you like.

Other useful keyboard shortcuts include Alt- for the assignment operator, and Ctrl+Shift+M for the pipe operator. You can delete these reminders if you don't want them in your report.

```
#setwd("C:/") #Don't forget to set your working directory before you start!

library("tidyverse")

## -- Attaching packages ----------------------------------------------------
------------------------------------------------------------------- tidyverse
1.3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts -------------------------------------------------------------
-------------------------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library("tidymodels")

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## -- Attaching packages ----------------------------------------------------
------------------------------------------------------------------ tidymodels
0.0.3 --

## v broom     0.5.3     v recipes   0.1.9
## v dials     0.0.4     v rsample   0.0.5
## v infer     0.5.1     v yardstick 0.0.4
## v parsnip   0.0.5

## -- Conflicts -------------------------------------------------------------
------------------------------------------------------------
tidymodels_conflicts() --
## x scales::discard()   masks purrr::discard()
## x dplyr::filter()     masks stats::filter()
## x recipes::fixed()    masks stringr::fixed()
## x dplyr::lag()        masks stats::lag()
## x dials::margin()     masks ggplot2::margin()
## x yardstick::spec()   masks readr::spec()
## x recipes::step()     masks stats::step()
## x recipes::yj_trans() masks scales::yj_trans()
```

```
library("plotly")
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library("skimr")
```

```
dfTit <-
  read_csv("walmartSales.csv")
```

```
## Parsed with column specification:
## cols(
##   Store = col_double(),
##   Date = col_date(format = ""),
##   IsHoliday = col_logical(),
##   Temperature = col_double(),
##   Fuel_Price = col_double(),
##   CPI = col_double(),
##   Unemployment = col_double(),
##   Size = col_double(),
##   Weekly_Sales = col_double()
## )
```

```
dfTit
```

```
## # A tibble: 6,435 x 9
##    Store Date       IsHoliday Temperature Fuel_Price   CPI Unemployment
Size
##    <dbl> <date>     <lgl>           <dbl>      <dbl> <dbl>        <dbl>
<dbl>
##  1    26 2011-08-26 FALSE            61.1       3.80  136.         7.77
152513
```

```
##  2    34 2011-03-25 FALSE            53.1       3.48  129.        10.4 
158114
##  3    21 2010-12-03 FALSE            50.4       2.71  211.         8.16
140167
##  4     8 2010-09-17 FALSE            75.3       2.58  215.         6.32
155078
##  5    19 2012-05-18 FALSE            58.8       4.03  138.         8.15
203819
##  6    13 2012-03-16 FALSE            52.5       3.53  131.         6.10
219622
##  7    19 2010-08-06 FALSE            74.2       2.94  133.         8.10
203819
##  8     2 2010-12-24 FALSE            50.0       2.89  211.         8.16
202307
##  9    32 2010-10-08 FALSE            61.8       2.74  191.         9.14
203007
## 10    45 2012-03-02 FALSE            41.6       3.82  190.         8.42
118221
## # ... with 6,425 more rows, and 1 more variable: Weekly_Sales <dbl>
```

Create a regression model using Weekly_Sales as the DV (Dependent Variable, outcome variable), and CPI as the IV (Independent Variable, feature, predictor, explanatory variable). [If you don’t remember how to run and interpret a linear model in R, see the appendix]

```
head(dfTit)

## # A tibble: 6 x 9
##   Store Date       IsHoliday Temperature Fuel_Price   CPI Unemployment
Size
##   <dbl> <date>     <lgl>           <dbl>      <dbl> <dbl>        <dbl>
<dbl>
## 1    26 2011-08-26 FALSE            61.1       3.80  136.         7.77
152513
## 2    34 2011-03-25 FALSE            53.1       3.48  129.        10.4 
158114
## 3    21 2010-12-03 FALSE            50.4       2.71  211.         8.16
140167
## 4     8 2010-09-17 FALSE            75.3       2.58  215.         6.32
155078
## 5    19 2012-05-18 FALSE            58.8       4.03  138.         8.15
203819
## 6    13 2012-03-16 FALSE            52.5       3.53  131.         6.10
219622
## # ... with 1 more variable: Weekly_Sales <dbl>

nrow(dfTit)

## [1] 6435

skim(dfTit)
```

Data summary

Name	dfTit
Number of rows	6435
Number of columns	9

Column type frequency:

Date	1
logical	1
numeric	7

Group variables	None

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
Date	0	1	2010-02-05	2012-10-26	2011-06-17	143

Variable type: logical

skim_variable	n_missing	complete_rate	mean	count
IsHoliday	0	1	0.07	FAL: 5985, TRU: 450

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Store	0	1	23.00	12.99	1.00	12.00	23.00	34.00	45.00	▇▇▇▇▇
Temperature	0	1	60.66	18.44	-2.06	47.46	62.67	74.94	100.14	▁▃▆▇▃
Fuel_Price	0	1	3.36	0.46	2.47	2.93	3.44	3.73	4.47	▆▆▇▇▁
CPI	0	1	171.58	39.36	126.06	131.74	182.62	212.74	227.23	▇▁▁▂▇
Unemployment	0	1	8.00	1.88	3.88	6.89	7.87	8.62	14.31	▂▇▆▁▁
Size	0	1	1302	6311	3487	7071	1265	2023	21962	▇▂▆

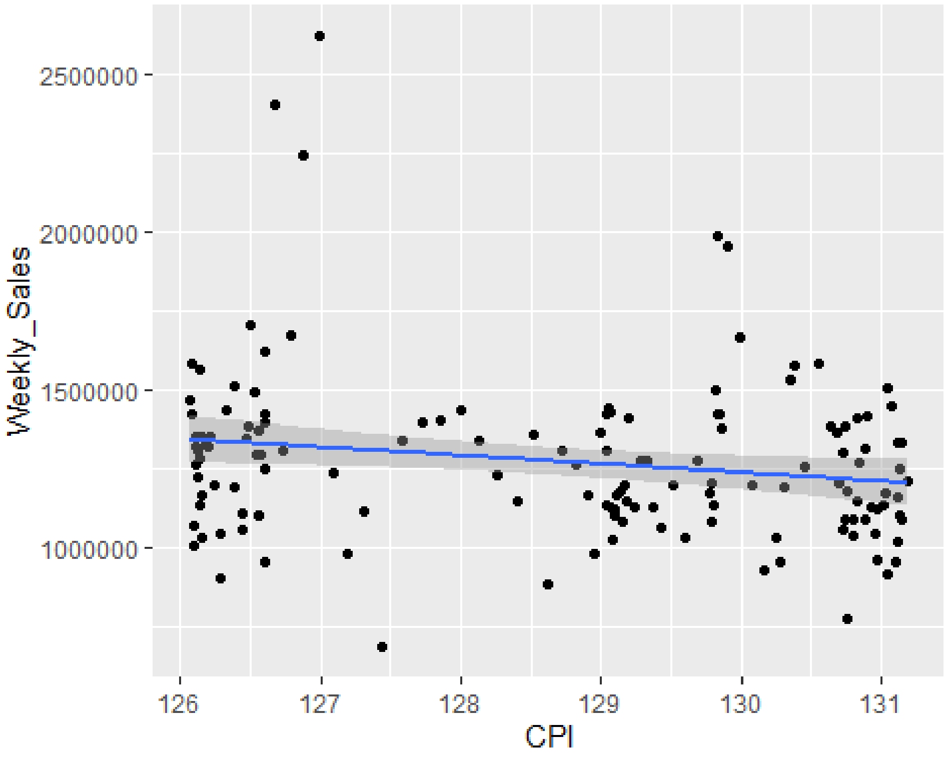
			87.60	7.02	5.00	3.00	12.00	07.00	2.00	▁▇
Weekly_ Sales	0	1	701559.55	391594.18	68982.11	375613.92	639652.39	958807.42	2773216.28	▇▅▁▁▁

Q1>

```
#Q1
dfTit

## # A tibble: 6,435 x 9
##    Store Date       IsHoliday Temperature Fuel_Price   CPI Unemployment
Size
##    <dbl> <date>     <lgl>           <dbl>      <dbl> <dbl>        <dbl>
<dbl>
##  1    26 2011-08-26 FALSE            61.1       3.80  136.         7.77
152513
##  2    34 2011-03-25 FALSE            53.1       3.48  129.        10.4
158114
##  3    21 2010-12-03 FALSE            50.4       2.71  211.         8.16
140167
##  4     8 2010-09-17 FALSE            75.3       2.58  215.         6.32
155078
##  5    19 2012-05-18 FALSE            58.8       4.03  138.         8.15
203819
##  6    13 2012-03-16 FALSE            52.5       3.53  131.         6.10
219622
##  7    19 2010-08-06 FALSE            74.2       2.94  133.         8.10
203819
##  8     2 2010-12-24 FALSE            50.0       2.89  211.         8.16
202307
##  9    32 2010-10-08 FALSE            61.8       2.74  191.         9.14
203007
## 10    45 2012-03-02 FALSE            41.6       3.82  190.         8.42
118221
## # ... with 6,425 more rows, and 1 more variable: Weekly_Sales <dbl>

fitCPI<-lm(formula=Weekly_Sales~CPI, data=dfTit)

summary(fitCPI)

##
## Call:
## lm(formula = Weekly_Sales ~ CPI, data = dfTit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -662386 -318443  -73868  258442 2095880
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 827280.5    21778.4  37.986  < 2e-16 ***
## CPI           -732.7      123.7  -5.923 3.33e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 390600 on 6433 degrees of freedom
## Multiple R-squared:  0.005423,   Adjusted R-squared:  0.005269
## F-statistic: 35.08 on 1 and 6433 DF,  p-value: 3.332e-09

?lm

## starting httpd help server ... done
```

Q2>

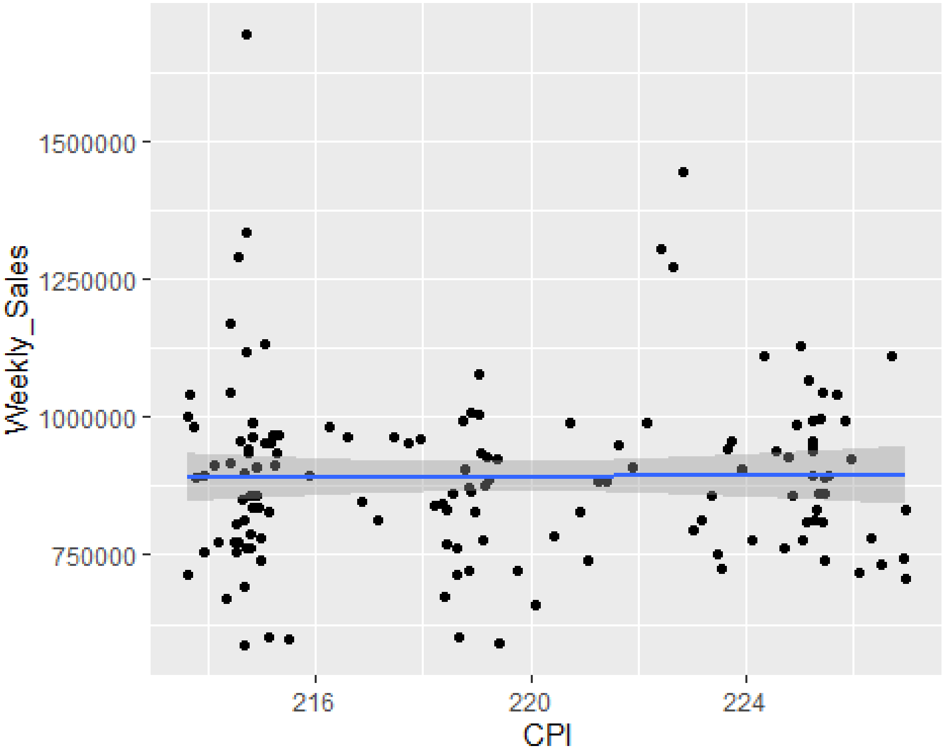
```
#Q2
plot <- dfTit %>%
  filter(Store==10)%>%
  ggplot(aes(x=CPI, y=Weekly_Sales))+
  geom_point()+
  geom_smooth(method = 'lm')

plot
```



```
ggplotly(plot)
```

Q2>

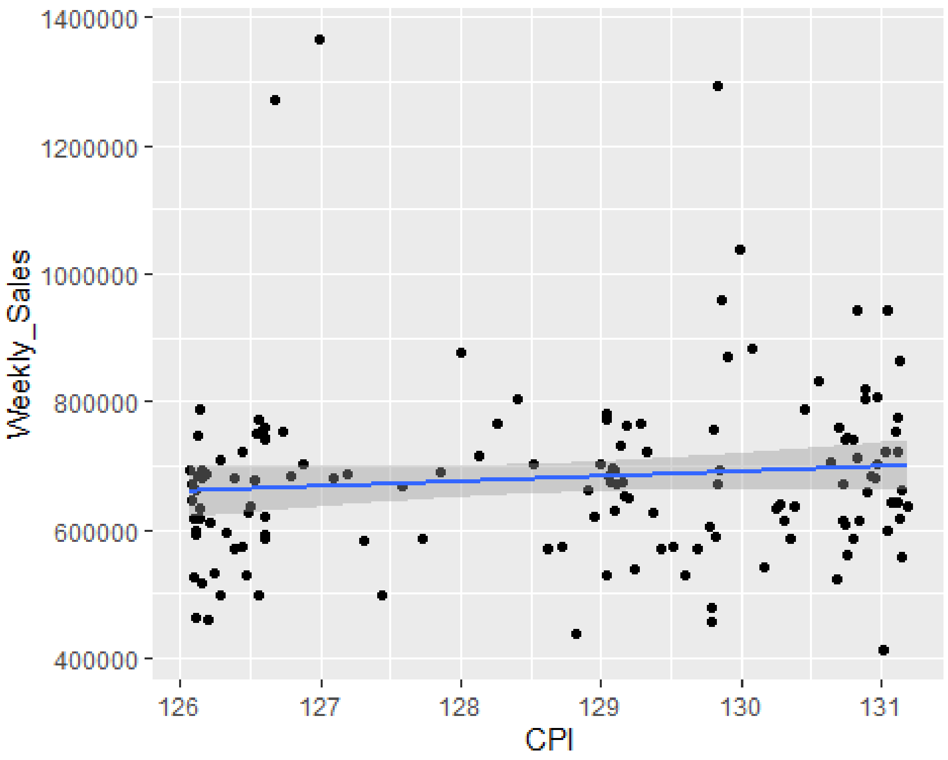
```
#Q2
plot <- dfTit %>%
  filter(Store==11)%>%
  ggplot(aes(x=CPI, y=Weekly_Sales))+
  geom_point()+
  geom_smooth(method = 'lm')

plot
```



```
ggplotly(plot)
```

Q2>

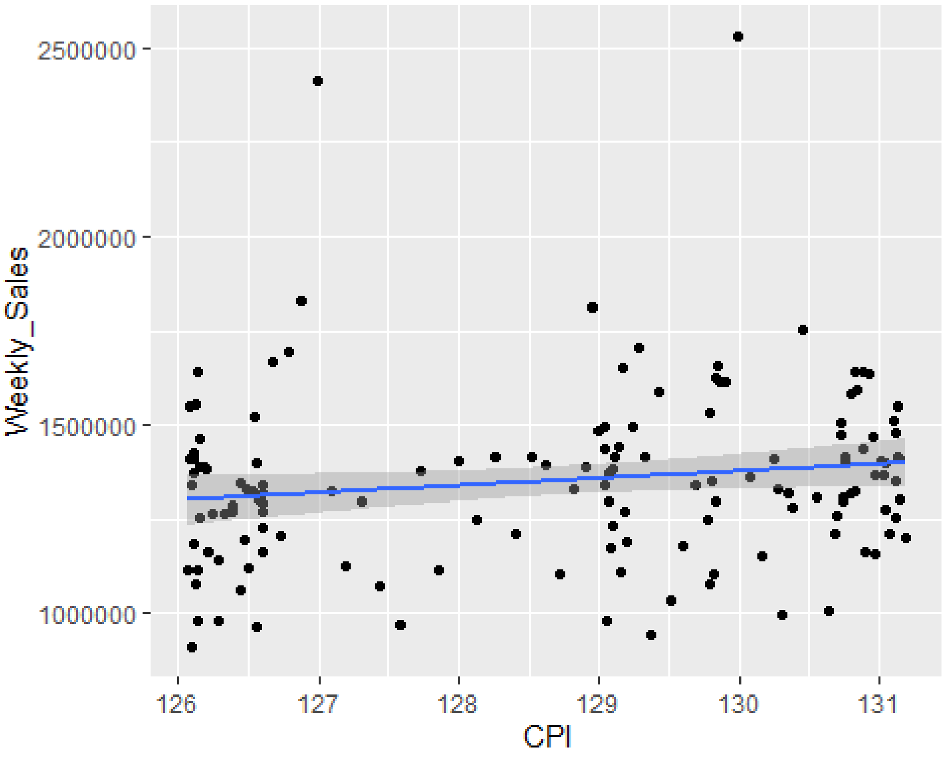
```
#Q2
plot <- dfTit %>%
  filter(Store==12)%>%
  ggplot(aes(x=CPI, y=Weekly_Sales))+
  geom_point()+
  geom_smooth(method = 'lm')

plot
```



```
ggplotly(plot)
```

Q2>

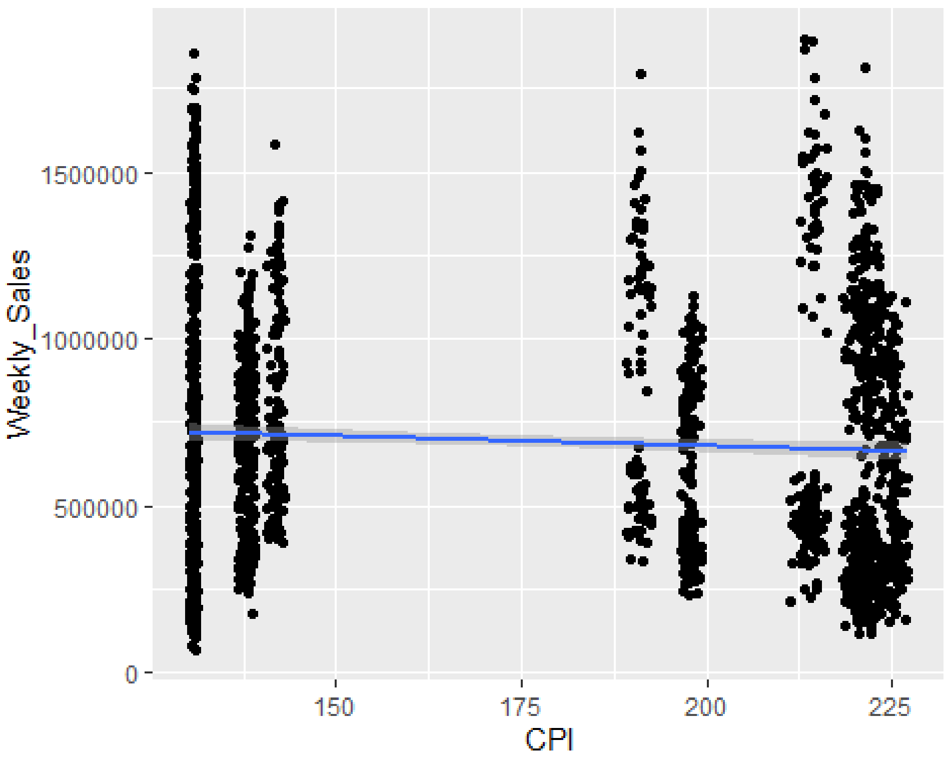
```
#Q2
plot <- dfTit %>%
  filter(Store==13)%>%
  ggplot(aes(x=CPI, y=Weekly_Sales))+
  geom_point()+
  geom_smooth(method = 'lm')

plot
```



```
ggplotly(plot)
```

Q3>

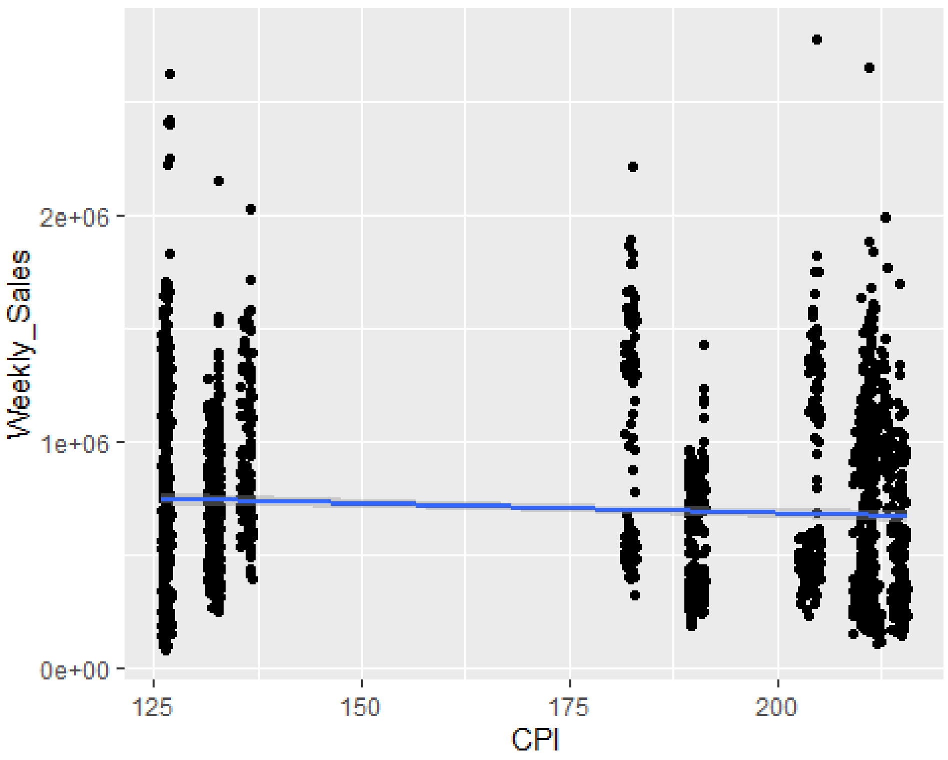
```
#Q3
library(lubridate)

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

plot <- dfTit %>%
  filter(year(Date)==2012)%>%
  #group_by(Store)%>%
  ggplot(aes(x=CPI, y=Weekly_Sales))+
  geom_point()+
  geom_smooth(method = 'lm')


plot
```



```
ggplotly(plot)
```

Q4>

```
#Q4
library(lubridate)
plot <- dfTit %>%
  filter(year(Date)==2010)%>%
  ggplot(aes(x=CPI, y=Weekly_Sales))+
  geom_point()+
  geom_smooth(method = 'lm')

#year(Date)==2010,,Store==1

plot
```



```
ggplotly(plot)
```

Build another regression model but this time include both CPI and Size as independent variables and call it fitCPISize. Compare this model with the model you built in Q1. Which model is better at explaining Weekly Sales? Why? Hint: Use anova() as well.

____

Has the estimated coefficient for CPI changed? If so, why do you think it has changed?

____

Q5>

```
#Q5
dfTit

## # A tibble: 6,435 x 9
##    Store Date       IsHoliday Temperature Fuel_Price   CPI Unemployment
Size
##    <dbl> <date>     <lgl>           <dbl>      <dbl> <dbl>        <dbl>
<dbl>
##  1    26 2011-08-26 FALSE            61.1       3.80  136.         7.77
152513
##  2    34 2011-03-25 FALSE            53.1       3.48  129.        10.4
158114
##  3    21 2010-12-03 FALSE            50.4       2.71  211.         8.16
140167
##  4     8 2010-09-17 FALSE            75.3       2.58  215.         6.32
```

```
155078
##  5    19 2012-05-18 FALSE            58.8       4.03  138.         8.15
203819
##  6    13 2012-03-16 FALSE            52.5       3.53  131.         6.10
219622
##  7    19 2010-08-06 FALSE            74.2       2.94  133.         8.10
203819
##  8     2 2010-12-24 FALSE            50.0       2.89  211.         8.16
202307
##  9    32 2010-10-08 FALSE            61.8       2.74  191.         9.14
203007
## 10    45 2012-03-02 FALSE            41.6       3.82  190.         8.42
118221
## # ... with 6,425 more rows, and 1 more variable: Weekly_Sales <dbl>
```

```
fitCPISize<-lm(formula=Weekly_Sales~CPI+Size, data=dfTit)
summary(fitCPISize)
```

```
##
## Call:
## lm(formula = Weekly_Sales ~ CPI + Size, data = dfTit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -563750 -167145  -29612  112172 1912650
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.828e+05  1.497e+04  12.216   <2e-16 ***
## CPI         -6.570e+02  7.692e+01  -8.542   <2e-16 ***
## Size         4.847e+00  4.796e-02 101.048   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 242800 on 6432 degrees of freedom
## Multiple R-squared:  0.6156, Adjusted R-squared:  0.6155
## F-statistic:  5151 on 2 and 6432 DF,  p-value: < 2.2e-16
```

```
summary(fitCPI)
```

```
##
## Call:
## lm(formula = Weekly_Sales ~ CPI, data = dfTit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -662386 -318443  -73868  258442 2095880
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 827280.5    21778.4  37.986  < 2e-16 ***
## CPI           -732.7      123.7  -5.923 3.33e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 390600 on 6433 degrees of freedom
## Multiple R-squared:  0.005423,   Adjusted R-squared:  0.005269
## F-statistic: 35.08 on 1 and 6433 DF,  p-value: 3.332e-09
```

```
#anova(fitCPISize)
#anova(fitCPI)

anova(fitCPISize,fitCPI)
```

```
## Analysis of Variance Table
##
## Model 1: Weekly_Sales ~ CPI + Size
## Model 2: Weekly_Sales ~ CPI
##   Res.Df        RSS Df   Sum of Sq     F    Pr(>F)
## 1   6432 3.7924e+14
## 2   6433 9.8128e+14 -1 -6.0204e+14 10211 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
?anova
?aov
?lm
```

?anova Q7>

```
#Q7
dfTit
```

```
## # A tibble: 6,435 x 9
##    Store Date       IsHoliday Temperature Fuel_Price   CPI Unemployment
Size
##    <dbl> <date>     <lgl>           <dbl>      <dbl> <dbl>        <dbl>
<dbl>
##  1    26 2011-08-26 FALSE            61.1       3.80  136.         7.77
152513
##  2    34 2011-03-25 FALSE            53.1       3.48  129.        10.4
158114
##  3    21 2010-12-03 FALSE            50.4       2.71  211.         8.16
140167
##  4     8 2010-09-17 FALSE            75.3       2.58  215.         6.32
155078
##  5    19 2012-05-18 FALSE            58.8       4.03  138.         8.15
203819
##  6    13 2012-03-16 FALSE            52.5       3.53  131.         6.10
219622
##  7    19 2010-08-06 FALSE            74.2       2.94  133.         8.10
203819
```

```
##  8     2 2010-12-24 FALSE            50.0       2.89  211.          8.16
202307
##  9    32 2010-10-08 FALSE            61.8       2.74  191.          9.14
203007
## 10    45 2012-03-02 FALSE            41.6       3.82  190.          8.42
118221
## # ... with 6,425 more rows, and 1 more variable: Weekly_Sales <dbl>
```

```
fitFull<-
lm(formula=Weekly_Sales~IsHoliday+Temperature+Fuel_Price+CPI+Unemployment+Siz
e, data=dfTit)

summary(fitFull)
```

```
##
## Call:
## lm(formula = Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price +
##     CPI + Unemployment + Size, data = dfTit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -557148 -165608  -24125  112851 1918479
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.133e+05  3.546e+04   8.834  < 2e-16 ***
## IsHolidayTRUE  6.012e+04  1.196e+04   5.026 5.14e-07 ***
## Temperature    1.002e+03  1.739e+02   5.761 8.72e-09 ***
## Fuel_Price    -1.333e+04  6.822e+03  -1.954   0.0507 .
## CPI           -9.461e+02  8.445e+01 -11.203  < 2e-16 ***
## Unemployment  -1.252e+04  1.725e+03  -7.258 4.40e-13 ***
## Size           4.840e+00  4.802e-02 100.786  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 241200 on 6428 degrees of freedom
## Multiple R-squared:  0.621,  Adjusted R-squared:  0.6206
## F-statistic:  1755 on 6 and 6428 DF,  p-value: < 2.2e-16
```

```
summary(fitCPISize)
```

```
##
## Call:
## lm(formula = Weekly_Sales ~ CPI + Size, data = dfTit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -563750 -167145  -29612  112172 1912650
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.828e+05  1.497e+04  12.216   <2e-16 ***
## CPI         -6.570e+02  7.692e+01  -8.542   <2e-16 ***
## Size         4.847e+00  4.796e-02 101.048   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 242800 on 6432 degrees of freedom
## Multiple R-squared:  0.6156, Adjusted R-squared:  0.6155
## F-statistic:  5151 on 2 and 6432 DF,  p-value: < 2.2e-16
```

```
anova(fitCPISize,fitFull)
```

```
## Analysis of Variance Table
##
## Model 1: Weekly_Sales ~ CPI + Size
## Model 2: Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price + CPI +
Unemployment +
##     Size
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1   6432 3.7924e+14
## 2   6428 3.7394e+14  4 5.3028e+12 22.789 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Q8>

```
#Q8
fitFullTemp<-
lm(formula=Weekly_Sales~IsHoliday+Temperature+Fuel_Price+CPI+Unemployment+Siz
e+I(Temperature^2), data=dfTit)

summary(fitFullTemp)
```

```
##
## Call:
## lm(formula = Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price +
##     CPI + Unemployment + Size + I(Temperature^2), data = dfTit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -561455 -165260  -24674  112058 1911166
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.610e+05  4.111e+04   6.350 2.30e-10 ***
## IsHolidayTRUE     6.230e+04  1.199e+04   5.197 2.09e-07 ***
## Temperature       3.294e+03  9.301e+02   3.542   0.0004 ***
## Fuel_Price       -1.471e+04  6.841e+03  -2.151   0.0315 *
## CPI              -9.547e+02  8.449e+01 -11.300  < 2e-16 ***
## Unemployment     -1.253e+04  1.724e+03  -7.268 4.09e-13 ***
## Size              4.831e+00  4.811e-02 100.420  < 2e-16 ***
## I(Temperature^2) -1.982e+01  7.901e+00  -2.509   0.0121 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 241100 on 6427 degrees of freedom
## Multiple R-squared:  0.6214, Adjusted R-squared:  0.621
## F-statistic:  1507 on 7 and 6427 DF,  p-value: < 2.2e-16
```

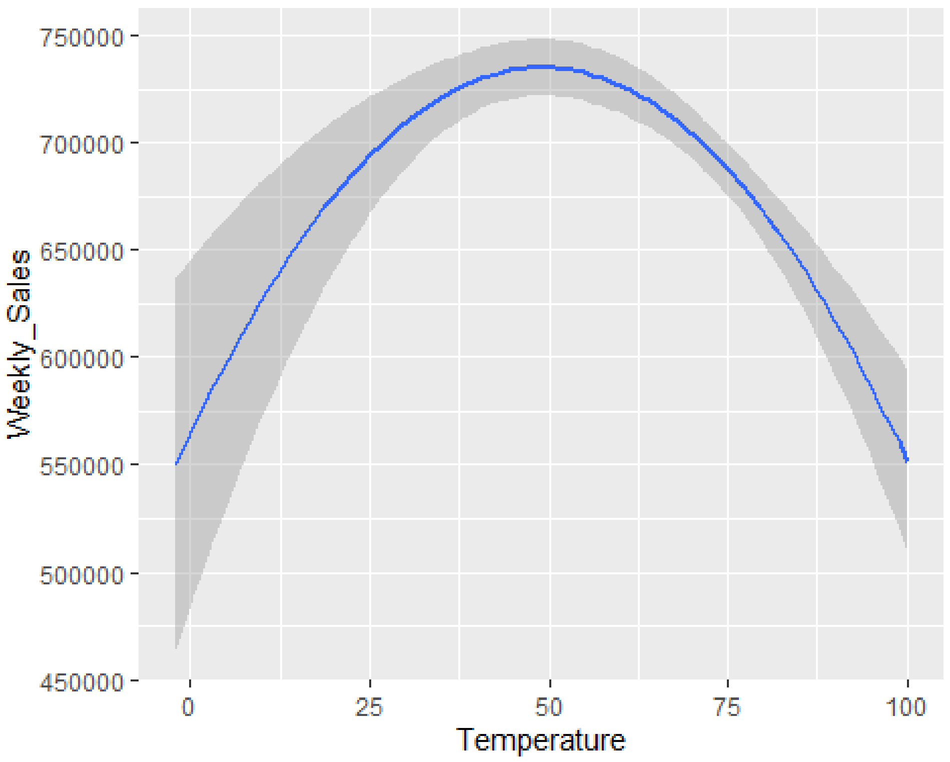
```
anova(fitFullTemp,fitFull)
```

```
## Analysis of Variance Table
##
## Model 1: Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price + CPI +
Unemployment +
##     Size + I(Temperature^2)
## Model 2: Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price + CPI +
Unemployment +
##     Size
##   Res.Df        RSS Df   Sum of Sq      F  Pr(>F)
## 1   6427 3.7357e+14
## 2   6428 3.7394e+14 -1 -3.6586e+11 6.2943 0.01214 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
?anova
```

Q8>

```
#Q8
dfTit %>% ggplot(aes(x=Temperature,y=Weekly_Sales)) +
geom_smooth(method = 'lm', formula = y ~ x + I(x^2))
```



```
#":>?:<
#"
```

Q9)a)b)

```
#Q9a)b)
set.seed(333)

dfwTrain <- dfTit %>% sample_frac(0.8)
dfwTest <- dplyr::setdiff(dfTit, dfwTrain)
```

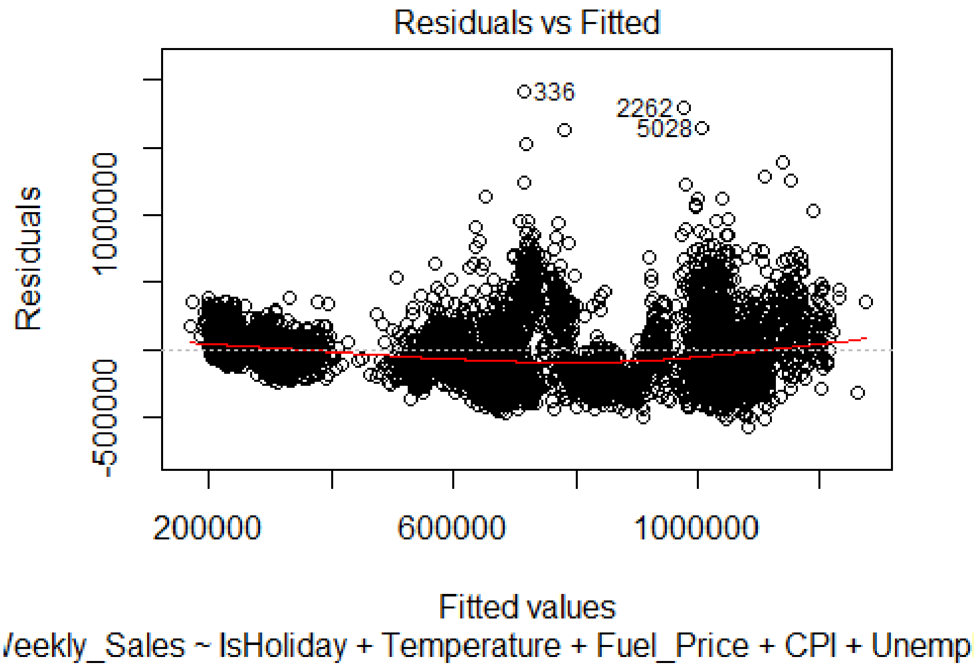
Q9)c)

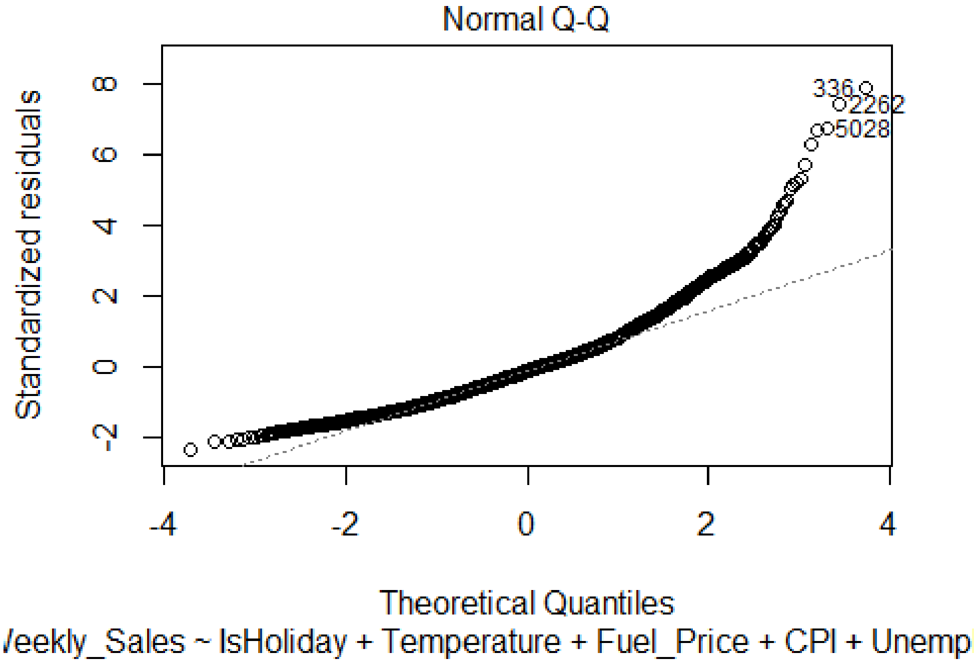
```
#Q9)c)
fitOrg<-
lm(formula=Weekly_Sales~IsHoliday+Temperature+Fuel_Price+CPI+Unemployment+Siz
e+I(Temperature^2), data=dfwTrain)

fitOrg

## 
## Call:
## lm(formula = Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price + 
##     CPI + Unemployment + Size + I(Temperature^2), data = dfwTrain)
## 
## Coefficients:
##      (Intercept)      IsHolidayTRUE        Temperature          Fuel_Price
##       263485.260         65687.645          3635.909        -17481.200
##              CPI      Unemployment              Size  I(Temperature^2)
##         -988.269        -12805.089             4.851           -21.915
```

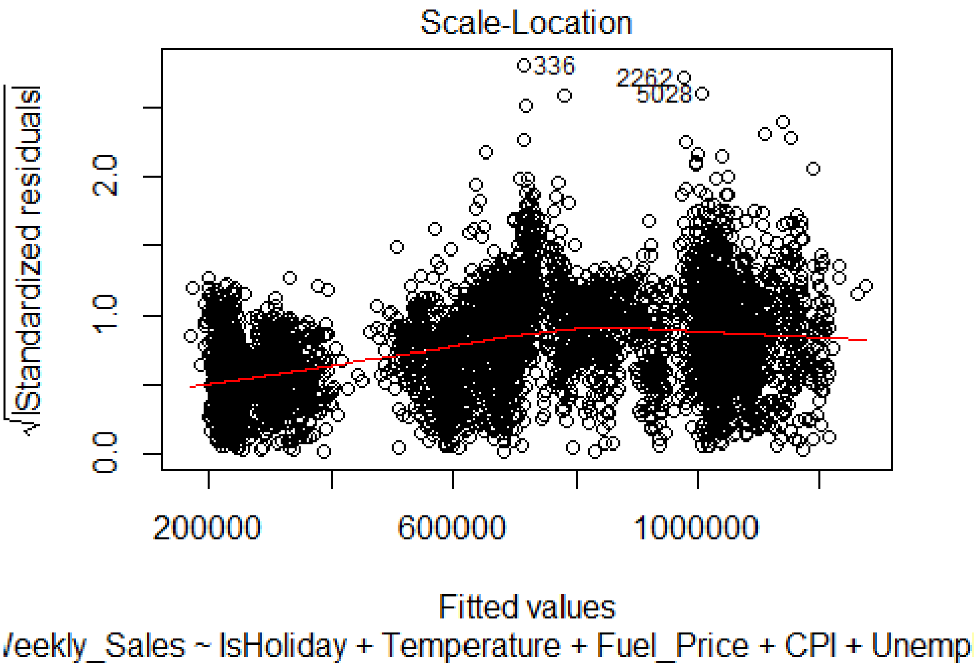
```r
#tidy?

summary(fitOrg)
```

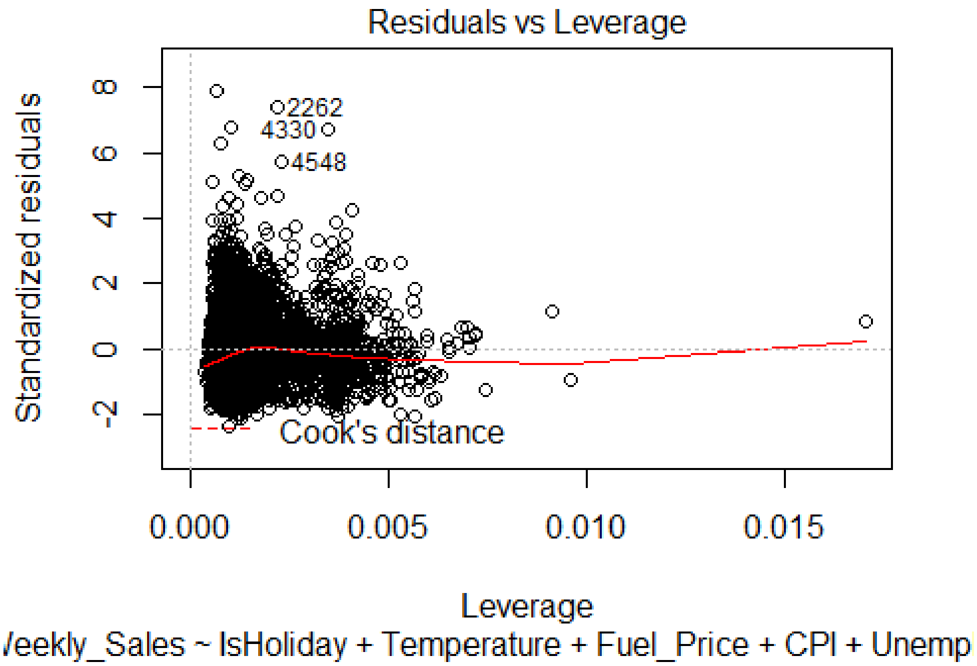
```
##
## Call:
## lm(formula = Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price +
##     CPI + Unemployment + Size + I(Temperature^2), data = dfwTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -564201 -166879  -25149  111412 1909304
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.635e+05  4.630e+04   5.691 1.34e-08 ***
## IsHolidayTRUE     6.569e+04  1.365e+04   4.811 1.55e-06 ***
## Temperature       3.636e+03  1.039e+03   3.498 0.000473 ***
## Fuel_Price       -1.748e+04  7.694e+03  -2.272 0.023130 *
## CPI              -9.883e+02  9.491e+01 -10.413  < 2e-16 ***
## Unemployment     -1.281e+04  1.939e+03  -6.603 4.43e-11 ***
## Size              4.851e+00  5.408e-02  89.686  < 2e-16 ***
## I(Temperature^2) -2.192e+01  8.832e+00  -2.481 0.013119 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 242200 on 5140 degrees of freedom
## Multiple R-squared:  0.6212, Adjusted R-squared:  0.6207
## F-statistic:  1204 on 7 and 5140 DF,  p-value: < 2.2e-16
```

```r
plot(fitOrg)
```


Residuals vs Fitted
Residuals
-500000
1000000
336
2262
5028
200000
600000
1000000
Fitted values
/eekly_Sales ~ IsHoliday + Temperature + Fuel_Price + CPI + Unemp


Normal Q-Q
Standardized residuals
-2
0
2
4
6
8
336
2262
5028
-4
-2
0
2
4
Theoretical Quantiles
/eekly_Sales ~ IsHoliday + Temperature + Fuel_Price + CPI + Unemp





```
tidy(fitOrg)

## # A tibble: 8 x 5
##   term                  estimate  std.error statistic   p.value
##   <chr>                 <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)      263485.   46302.          5.69 1.34e- 8
## 2 IsHolidayTRUE     65688.   13655.          4.81 1.55e- 6
## 3 Temperature        3636.    1039.          3.50 4.73e- 4
## 4 Fuel_Price       -17481.    7694.         -2.27 2.31e- 2
## 5 CPI                -988.      94.9       -10.4  3.86e-25
## 6 Unemployment     -12805.    1939.         -6.60 4.43e-11
## 7 Size                  4.85     0.0541     89.7  0.
## 8 I(Temperature^2)    -21.9      8.83       -2.48 1.31e- 2
```

Q9)d)

```
#Q9)d)
resultsOrg <-

  dfwTest %>%

  mutate(predictedSales = predict(fitOrg, dfwTest))



resultsOrg

## # A tibble: 1,287 x 10
##    Store Date       IsHoliday Temperature Fuel_Price   CPI Unemployment
Size
##    <dbl> <date>     <lgl>           <dbl>      <dbl> <dbl>        <dbl>
<dbl>
##  1    34 2011-03-25 FALSE            53.1       3.48  129.         10.4
158114
##  2     8 2010-09-17 FALSE            75.3       2.58  215.          6.32
155078
##  3    13 2012-03-16 FALSE            52.5       3.53  131.          6.10
219622
##  4    45 2011-02-18 FALSE            40.7       3.24  184.          8.55
118221
##  5    38 2011-08-26 FALSE            94.6       3.74  129.         13.5
39690
##  6     1 2010-04-16 FALSE            66.3       2.81  210.          7.81
151315
##  7    22 2010-10-01 FALSE            69.3       2.72  137.          8.57
119557
##  8    40 2010-04-02 FALSE            41.4       2.83  132.          5.44
155083
##  9    36 2010-11-26 TRUE             67.7       2.72  211.          8.48
39910
## 10    22 2010-08-20 FALSE            73.2       2.80  137.          8.43
119557
## # ... with 1,277 more rows, and 2 more variables: Weekly_Sales <dbl>,
## #   predictedSales <dbl>
```

```
?predict
```

Q9)e)

```
#Q9)e)
performance<-metric_set(rmse,mae)

Model<-performance(resultsOrg,truth=Weekly_Sales,estimate=predictedSales)

Model

## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard     236687.
## 2 mae     standard     177863.
```

Q9)f)

```
#Q9)f)

fitOrgDate<-
lm(formula=Weekly_Sales~IsHoliday+Temperature+Fuel_Price+CPI+Unemployment+Siz
e+Date+I(Temperature^2), data=dfwTrain)

resultsOrgDate <- dfwTest %>% mutate(predictedSales = predict(fitOrgDate,
dfwTest))


summary(fitOrgDate)

##
## Call:
## lm(formula = Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price +
##     CPI + Unemployment + Size + Date + I(Temperature^2), data = dfwTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -562281 -167059  -25354  111694 1909518
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.194e+05  2.803e+05   0.426 0.670102
## IsHolidayTRUE     6.505e+04  1.371e+04   4.745 2.14e-06 ***
## Temperature       3.660e+03  1.041e+03   3.517 0.000439 ***
## Fuel_Price       -2.278e+04  1.275e+04  -1.786 0.074114 .
## CPI              -1.001e+03  9.792e+01 -10.221  < 2e-16 ***
## Unemployment     -1.252e+04  2.017e+03  -6.207 5.83e-10 ***
## Size              4.851e+00  5.410e-02  89.669  < 2e-16 ***
## Date              1.065e+01  2.043e+01   0.521 0.602246
## I(Temperature^2) -2.217e+01  8.845e+00  -2.506 0.012247 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 242200 on 5139 degrees of freedom
## Multiple R-squared:  0.6212, Adjusted R-squared:  0.6206
## F-statistic:  1053 on 8 and 5139 DF,  p-value: < 2.2e-16

performance<-metric_set(rmse,mae)

ModelDate<-
performance(resultsOrgDate,truth=Weekly_Sales,estimate=predictedSales)

ModelDate

## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard     236595.
## 2 mae     standard     177765.
```

Q9)g)

```
#Q9)g)

fitOrgNoUn<-
lm(formula=Weekly_Sales~IsHoliday+Temperature+Fuel_Price+CPI+Size+I(Temperatu
re^2), data=dfwTrain)

summary(fitOrgNoUn)

##
## Call:
## lm(formula = Weekly_Sales ~ IsHoliday + Temperature + Fuel_Price +
##     CPI + Size + I(Temperature^2), data = dfwTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -571464 -169026  -27962  112635 1905709
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.125e+05  4.043e+04   2.783  0.00541 **
## IsHolidayTRUE     6.362e+04  1.371e+04   4.641 3.55e-06 ***
## Temperature       3.419e+03  1.043e+03   3.278  0.00105 **
## Fuel_Price       -1.087e+04  7.660e+03  -1.419  0.15605
## CPI              -7.762e+02  8.968e+01  -8.655  < 2e-16 ***
## Size              4.878e+00  5.414e-02  90.097  < 2e-16 ***
## I(Temperature^2) -2.197e+01  8.868e+00  -2.478  0.01325 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 243200 on 5141 degrees of freedom
## Multiple R-squared:  0.618,  Adjusted R-squared:  0.6175
## F-statistic:  1386 on 6 and 5141 DF,  p-value: < 2.2e-16

resultsOrgNoUn <-

  dfwTest %>%

  mutate(predictedSales = predict(fitOrgNoUn, dfwTest))

performance<-metric_set(rmse,mae)

ModelNoUn<-
performance(resultsOrgNoUn,truth=Weekly_Sales,estimate=predictedSales)

ModelNoUn

## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard     237532.
## 2 mae     standard     178680.
```

The finale has to be sweet, right? Instead of using sales, create a log-transformed version, set the seed, split the data, run the model fitLog, make predictions, calculate performance. Have the coefficient estimates and variance explained in DV improved? Compare the model output and performance of fitLog with that of fitOrg from Q9c, and discuss. Check and compare the diagnostics from fitLog with those from fitOrg, and discuss.

Q10>

```
#Q10

set.seed(333)

dfTit<-dfTit%>%
  mutate(logsale=log(Weekly_Sales))

dfwTrain2 <- dfTit %>% sample_frac(0.8)
dfwTest2 <- dplyr::setdiff(dfTit, dfwTrain2)



fitLog<-
lm(formula=logsale~IsHoliday+Temperature+Fuel_Price+CPI+Size+I(Temperature^2)
, data=dfwTrain2)


resultsLog <-
```

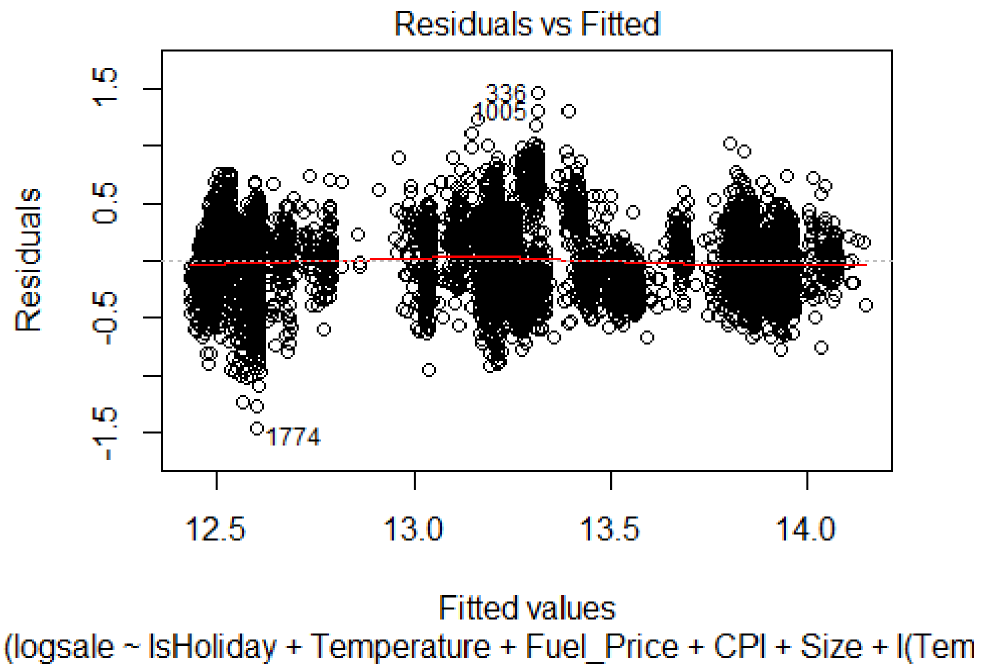
```
  dfwTest2 %>%

  mutate(predictedSales2 = predict(fitLog, dfwTest2))

performance<-metric_set(rmse,mae)

ModelLog<-performance(resultsLog,truth=logsale,estimate=predictedSales2)

ModelLog
```

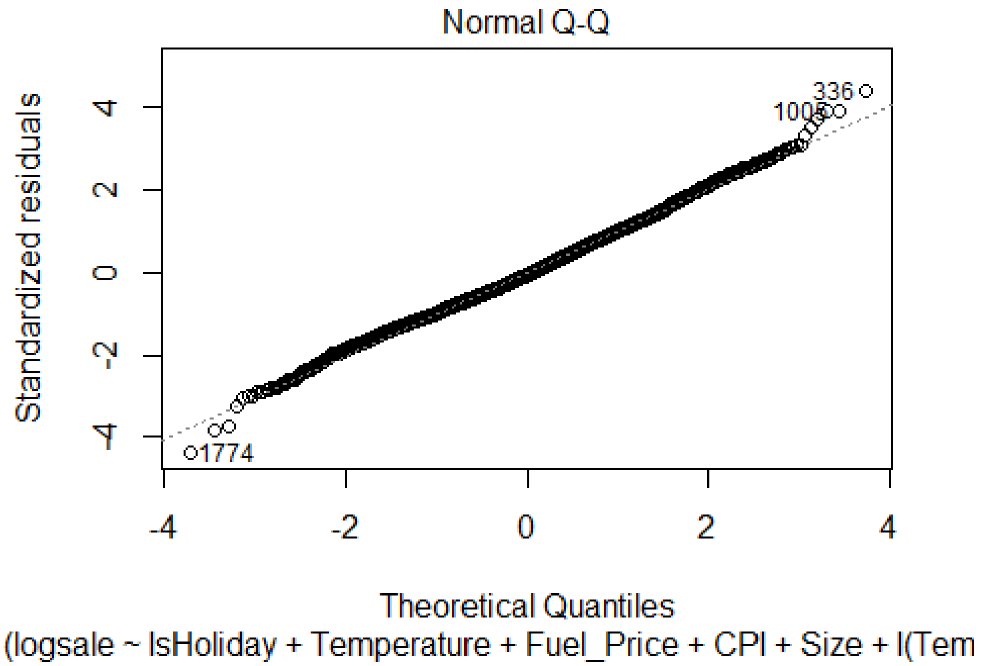
```
## # A tibble: 2 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       0.319
## 2 mae     standard       0.257
```

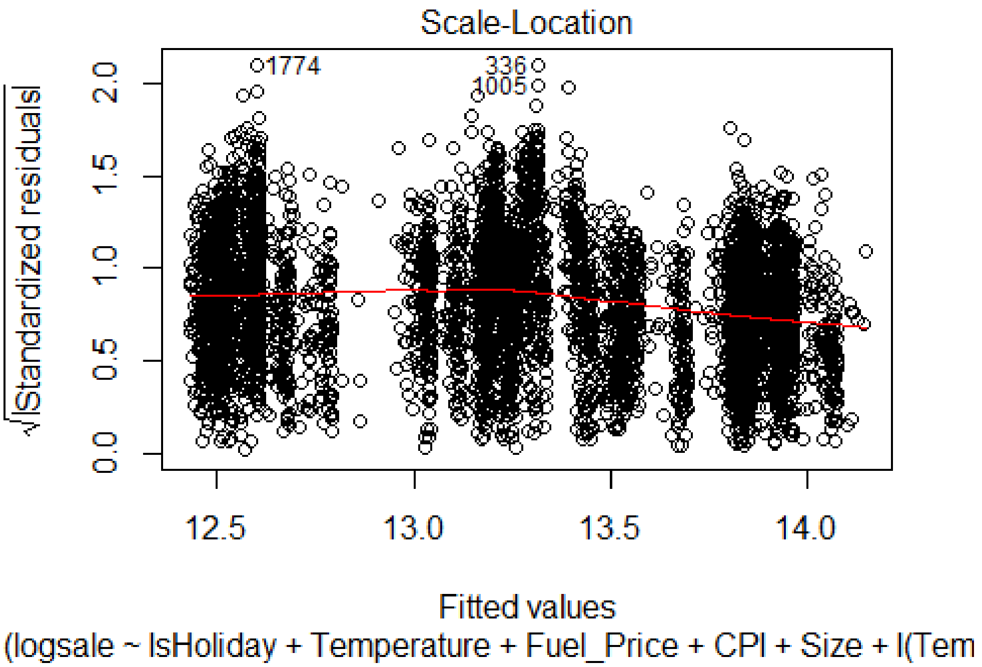
```
summary(fitLog)
```

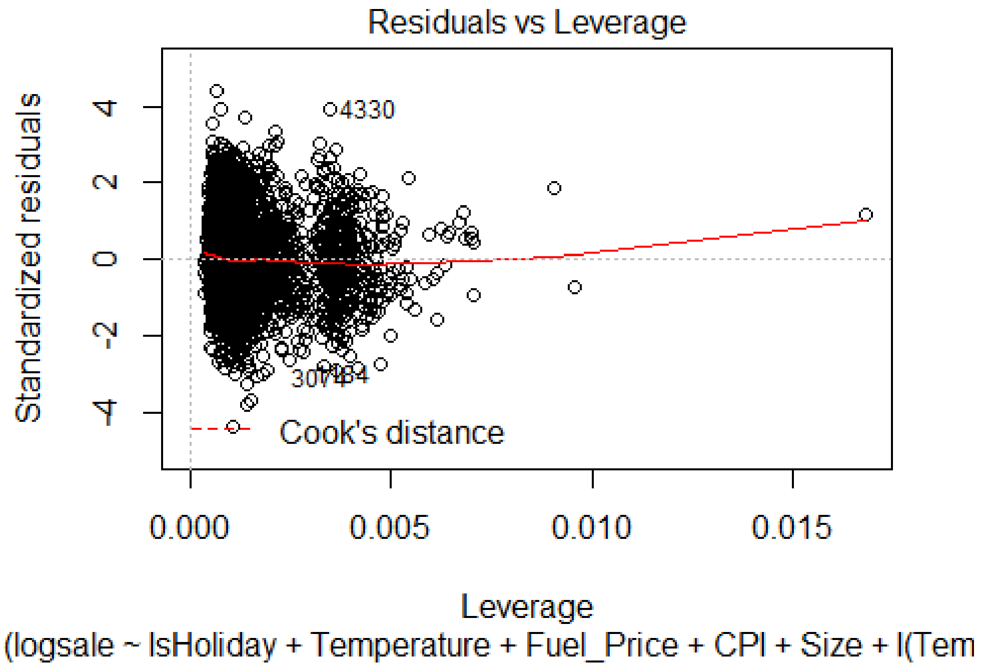
```
## 
## Call:
## lm(formula = logsale ~ IsHoliday + Temperature + Fuel_Price + 
##     CPI + Size + I(Temperature^2), data = dfwTrain2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1.45721 -0.22990 -0.01992  0.22395  1.46495 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       1.225e+01  5.542e-02 221.005  < 2e-16 ***
## IsHolidayTRUE     7.830e-02  1.879e-02   4.167 3.14e-05 ***
## Temperature       5.543e-03  1.430e-03   3.876 0.000107 ***
## Fuel_Price        1.636e-03  1.050e-02   0.156 0.876183    
## CPI              -1.083e-03  1.229e-04  -8.808  < 2e-16 ***
## Size              8.160e-06  7.422e-08 109.942  < 2e-16 ***
## I(Temperature^2) -4.595e-05  1.216e-05  -3.780 0.000159 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3334 on 5141 degrees of freedom
## Multiple R-squared:  0.7079, Adjusted R-squared:  0.7075 
## F-statistic:  2076 on 6 and 5141 DF,  p-value: < 2.2e-16
```

```
plot(fitLog)
```


Residuals vs Fitted
Residuals
1.5
0.5
-0.5
-1.5
12.5
13.0
13.5
14.0
336
1005
1774
Fitted values
(logsale ~ IsHoliday + Temperature + Fuel_Price + CPI + Size + I(Tem


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
-4
-2
0
2
4
336
1005
1774
Theoretical Quantiles
(logsale ~ IsHoliday + Temperature + Fuel_Price + CPI + Size + I(Tem


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
1774
336
1005
12.5
13.0
13.5
14.0
Fitted values
(logsale ~ IsHoliday + Temperature + Fuel_Price + CPI + Size + I(Tem


Residuals vs Leverage
Standardized residuals
-4
-2
0
2
4
4330
Cook's distance
0.000
0.005
0.010
0.015
Leverage
(logsale ~ IsHoliday + Temperature + Fuel_Price + CPI + Size + I(Tem