GSCM 451: R Summary

Monday December 21, 2020 at 02:52

Contents

```
library(lessR)
```

```
##
## lessR 3.9.8  feedback: gerbing@pdx.edu  web: lessRstats.com/new
## ---------------------------------------------------------------
## > d <- Read("")   Read text, Excel, SPSS, SAS, or R data file
##   d is default data frame, data= in analysis routines optional
##
## Access many vignettes to show by example how to use lessR.
## To learn about reading, writing, & manipulating data, graphics,
##   means & models, factor analysis, & customization:
## enter,  browseVignettes("lessR")  or
## visit,  https://CRAN.R-project.org/package=lessR
```

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(knitr)
```

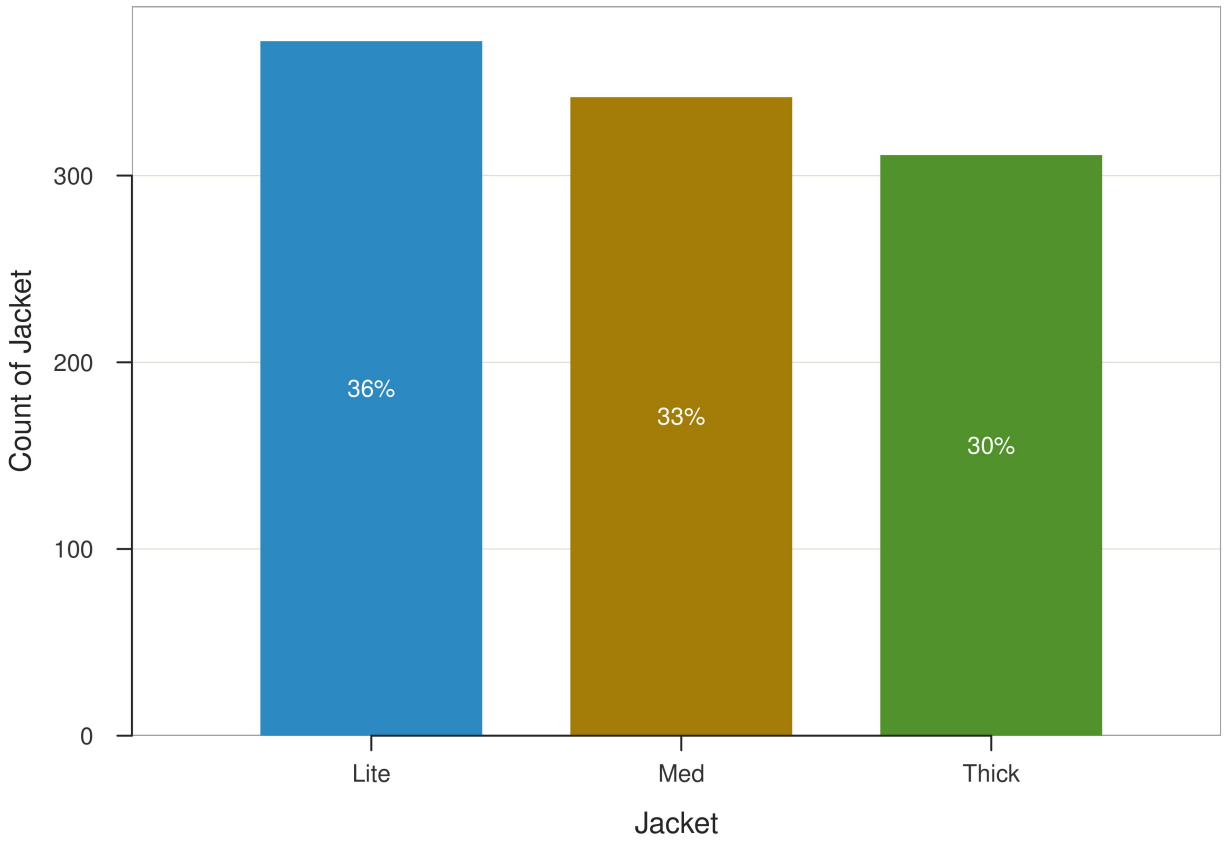
HW #1

```
d <- Read("http://web.pdx.edu/~gerbing/data/Jackets.csv")
```

2e: To show a distribution of a categorical variable, use a bar chart or a pie chart.

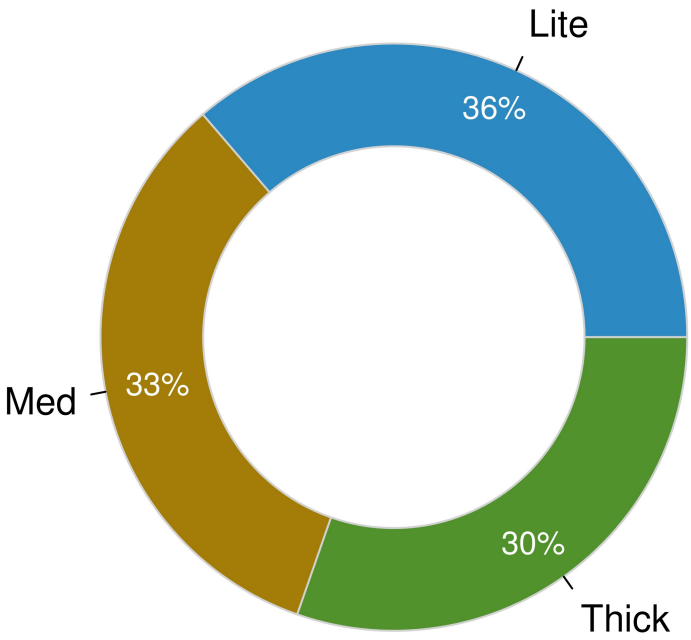
```
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## character: Non-numeric data values
## ------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1      Bike character   1025       0       2   BMW  Honda  Honda ... Honda  Honda  BMW
##  2    Jacket character   1025       0       3   Lite  Lite  Lite ... Lite  Med  Lite
## ------------------------------------------------------------------------------------------
```

```
BarChart(Jacket)
```




```
## >>> Suggestions
## BarChart(Jacket, horiz=TRUE)  # horizontal bar chart
## BarChart(Jacket, fill="greens")  # sequential green bars
## PieChart(Jacket)  # doughnut (ring) chart
## Plot(Jacket)  # bubble plot
## Plot(Jacket, stat="count")  # lollipop plot
##
##
## --- Jacket ---
##
##
## Missing Values of Jacket: 0
##
##
##                 Lite    Med  Thick      Total
## Frequencies:     372    342    311       1025
## Proportions:   0.363  0.334  0.303      1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 5.446, df = 2, p-value = 0.066
```

```
PieChart(Jacket)
```



2f

```
## >>> Suggestions
## PieChart(Jacket, hole=0)  # traditional pie chart
## PieChart(Jacket, values="%")  # display %'s on the chart
## BarChart(Jacket)  # bar chart
## Plot(Jacket)  # bubble plot
## Plot(Jacket, values="count")  # lollipop plot
##
##
## --- Jacket ---
##
##
##                 Lite    Med  Thick      Total
## Frequencies:     372    342    311       1025
## Proportions:   0.363  0.334  0.303      1.000
##
##
## Chi-squared test of null hypothesis of equal probabilities
##   Chisq = 5.446, df = 2, p-value = 0.066
```

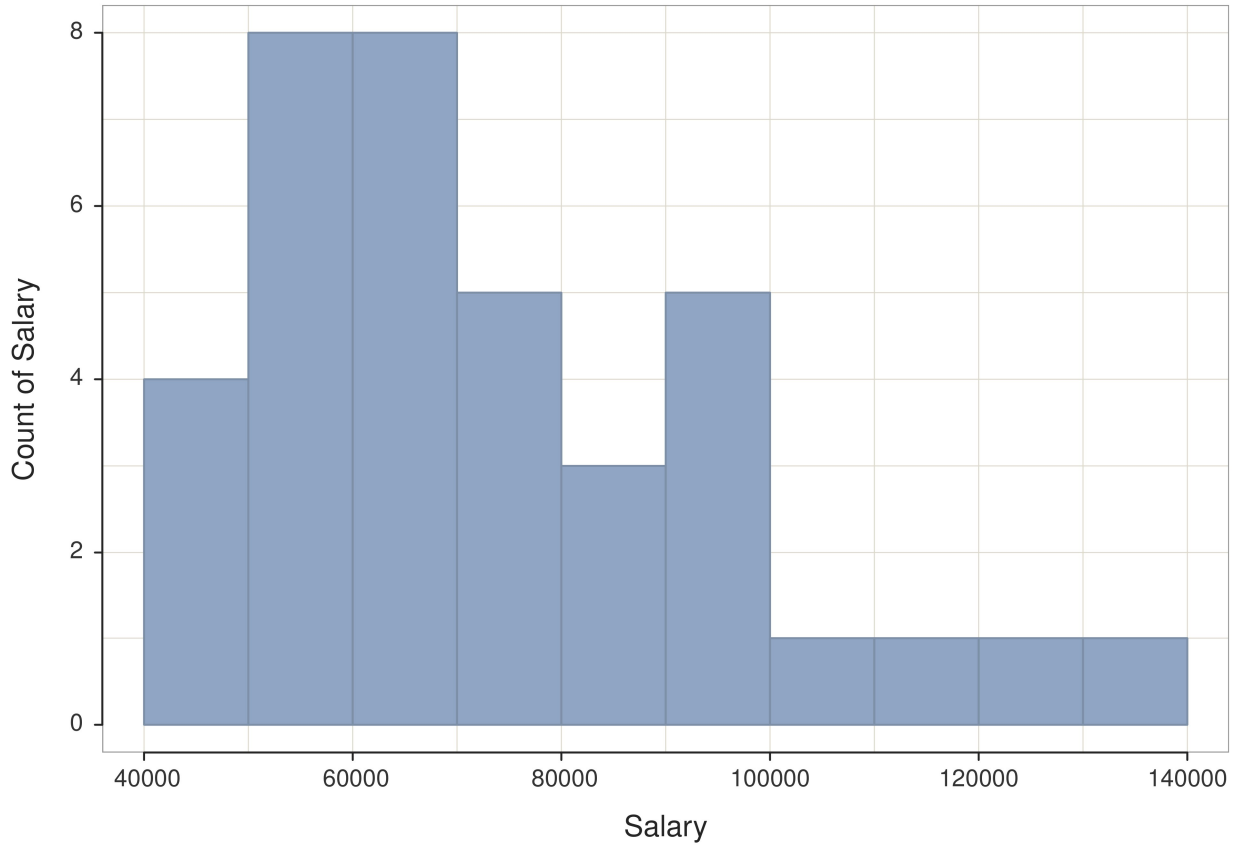
```
d <- Read("http://lessRstats.com/data/employee.xlsx")
```

3a

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
```

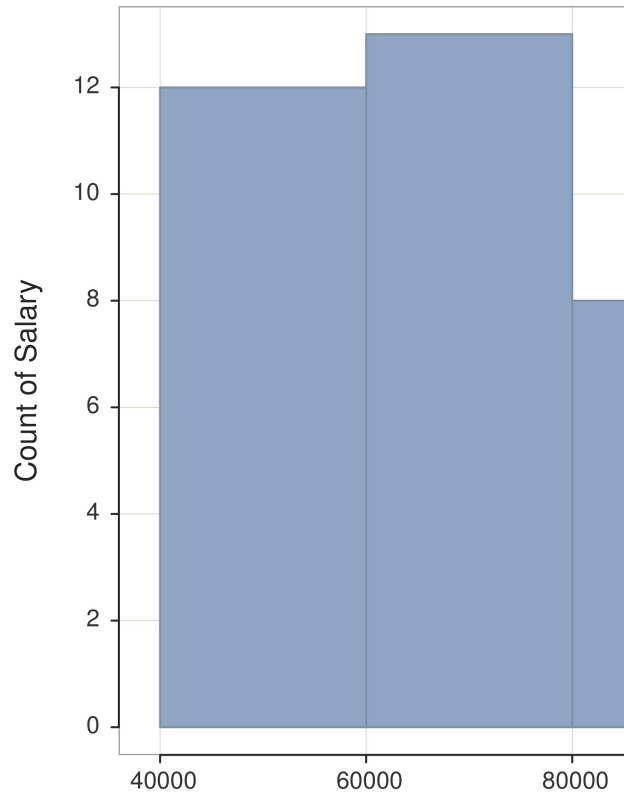
```
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## -------------------------------------------------------------
## character: Non-numeric data values
## integer: Numeric data values, integers only
## double: Numeric data values with decimal digits
## -------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name      Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------------
##  1      Name character      37       0      37   Ritchie, Darnell ... Cassinelli, Anastis
##  2     Years   integer      36       1      16   7  NA  15 ... 1  2  10
##  3    Gender character      37       0       2   M  M  M ... F  F  M
##  4      Dept character      36       1       5   ADMN  SALE  SALE ... MKTG  SALE  FINC
##  5    Salary    double      37       0      37   53788.26  94494.58 ... 56508.32  57562.36
##  6    JobSat character      35       2       3   med  low  low ... high  low  high
##  7      Plan   integer      37       0       3   1  1  3 ... 2  2  1
##  8       Pre   integer      37       0      27   82  62  96 ... 83  59  80
##  9      Post   integer      37       0      22   92  74  97 ... 90  71  87
## ------------------------------------------------------------------------------------------------
```

```
Histogram(Salary)
```



```
## >>> Suggestions
## bin_width: set the width of each bin
## bin_start: set the start of the first bin
## bin_end: set the end of the last bin
## Density(Salary)  # smoothed density curves plus histogram
## Plot(Salary)  # Violin/Box/Scatterplot (VBS) plot
##
##
## --- Salary ---
##
##      n   miss         mean           sd          min          mdn          max
##      37      0    73795.557    21799.533    46124.970    69547.600   134419.230
##
##
## (Box plot) Outliers: 1
##
## Small      Large
## -----      -----
##             134419.2
##
##
## Bin Width: 10000
## Number of Bins: 10
##
##              Bin  Midpnt  Count    Prop  Cumul.c  Cumul.p
## ------------------------------------------------------------
##   40000 >  50000   45000      4    0.11        4     0.11
##   50000 >  60000   55000      8    0.22       12     0.32
##   60000 >  70000   65000      8    0.22       20     0.54
##   70000 >  80000   75000      5    0.14       25     0.68
##   80000 >  90000   85000      3    0.08       28     0.76
##   90000 > 100000   95000      5    0.14       33     0.89
##  100000 > 110000  105000      1    0.03       34     0.92
##  110000 > 120000  115000      1    0.03       35     0.95
##  120000 > 130000  125000      1    0.03       36     0.97
##  130000 > 140000  135000      1    0.03       37     1.00
```

```
Histogram(Salary, bin.width = 20000)
```



3d (do not need to over-smooth, just get a different bin width)

```
## >>> Suggestions
## bin_width: set the width of each bin
## bin_start: set the start of the first bin
## bin_end: set the end of the last bin
## Density(Salary)  # smoothed density curves plus histogram
## Plot(Salary)  # Violin/Box/Scatterplot (VBS) plot
##
##
## --- Salary ---
##
##      n   miss         mean           sd          min          mdn          max
##      37      0    73795.557    21799.533    46124.970    69547.600   134419.230
##
##
## (Box plot) Outliers: 1
##
## Small      Large
## -----      -----
##             134419.2
##
##
## Bin Width: 20000
## Number of Bins: 5
##
##              Bin  Midpnt  Count    Prop  Cumul.c  Cumul.p
## ------------------------------------------------------------
```

```
##   40000 >  60000   50000     12    0.32       12     0.32
##   60000 >  80000   70000     13    0.35       25     0.68
##   80000 > 100000   90000      8    0.22       33     0.89
##  100000 > 120000  110000      2    0.05       35     0.95
##  120000 > 140000  130000      2    0.05       37     1.00
```

```
d <- Read("/Users/risto/Google Drive/1 Academic Programs/Undergrad/A School Year 2018-19/1 Spring 2019/
```

4a

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## character: Non-numeric data values
## integer: Numeric data values, integers only
## ------------------------------------------------------------
##
##     Variable                  Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1    Gender character      8       0       2   F  F  M ... M  F  M
##  2    Weight   integer      8       0       8   150  138  240 ... 200  140  220
##  3    Height   integer      7       1       6   66  66  NA ... 74  70  77
## ------------------------------------------------------------------------------------------
```

```
Ht_cm <- d$Height * 2.54
Ht_cm
```

```
## [1] 167.64 167.64     NA 180.34 162.56 187.96 177.80 195.58
```

```
Wt_kg <- d$Weight * 0.45359237
Wt_kg
```

```
## [1]  68.03886  62.59575 108.86217  80.73944  58.96701  90.71847  63.50293
## [8]  99.79032
```

```
round(Ht_cm, 2)
```

4b

```
## [1] 167.64 167.64     NA 180.34 162.56 187.96 177.80 195.58
```

```
round(Wt_kg, 2)
```

```
## [1]  68.04  62.60 108.86  80.74  58.97  90.72  63.50  99.79
```

```
d$Height <- Ht_cm
d$Weight <- Wt_kg
```

```
Write(to="HW01_451_revised", data=d, format="Excel", ExcelTable = TRUE)
```

4d

```
## [with the writeDataTable function from Alexander Walker's openxlsx package]
##
## The d data values written at the current working directory
##         HW01_451_revised.xlsx  in:   C:/Users/risto/Google Drive/1 Academic Programs/Undergrad/A Sch
```

```
d
```

```
##   Gender    Weight Height
## 1      F  68.03886 167.64
## 2      F  62.59575 167.64
## 3      M 108.86217     NA
## 4      M  80.73944 180.34
## 5      F  58.96701 162.56
## 6      M  90.71847 187.96
## 7      F  63.50293 177.80
## 8      M  99.79032 195.58
```

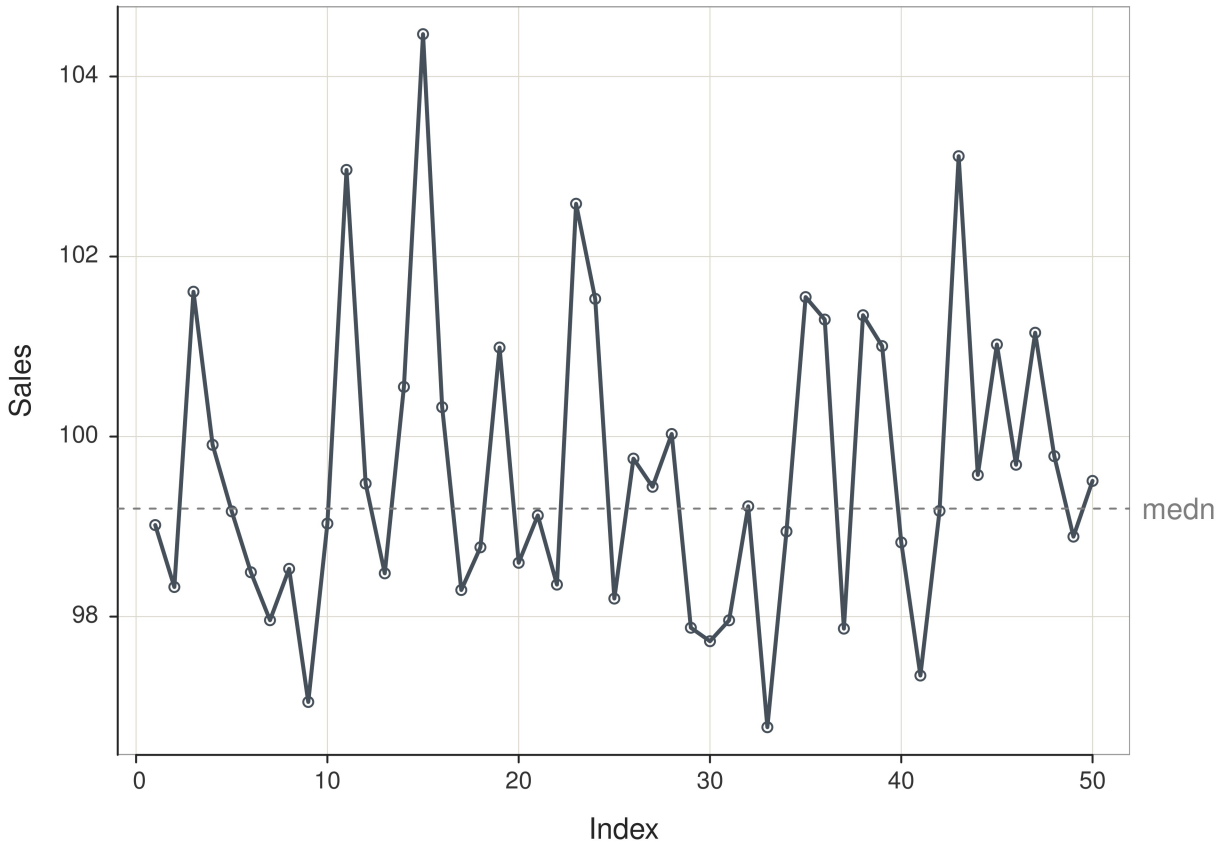
HW #2

```
d <- rd("http://web.pdx.edu/~gerbing/451/Data/HW2_2.xlsx")
```

2a

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1     Sales    double     50       0      50   99.0172  98.329 ... 98.8887  99.5078
## ------------------------------------------------------------------------------------------
```

```
Plot(Sales, run=TRUE)
```




```
## >>> Suggestions
## Plot(Sales, run=TRUE, size=0)  # just line segments, no points
## Plot(Sales, run=TRUE, lwd=0)  # just points, no line segments
## Plot(Sales, run=TRUE, fill="on")  # default color fill
##
##      n   miss          mean            sd           min           mdn           max
##     50      0      99.61408       1.65053      96.77150      99.19990     104.47060
##
## ------------
## Run Analysis
## ------------
##
## Total number of runs: 22
## Total number of values that do not equal the median: 50
## Total number of values ignored that equal the median: 0
```

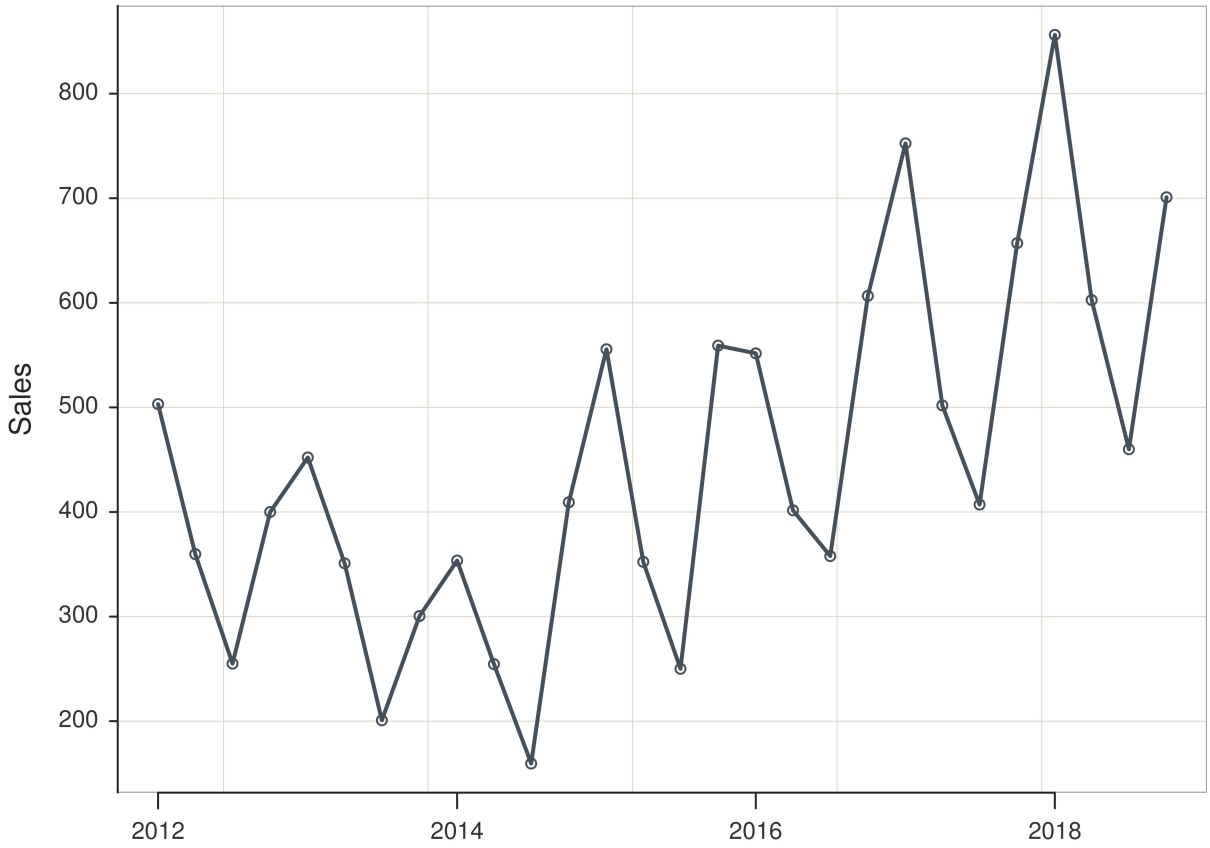
```
rm(list=ls())
d <- Read("http://web.pdx.edu/~gerbing/451/Data/HW2_3.xlsx")
```

3a

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
```

```
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## -------------------------------------------------------------
## character: Non-numeric data values
## Date: Date with year, month and day
## double: Numeric data values with decimal digits
## -------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1     Month      Date     56       0      28   2012-01-01 ... 2018-10-01
##  2     Sales    double     56       0      55   503.2  359.74 ... 429.63  400.07
##  3   Product character     56       0       2   Widget  Widget ... Opener  Opener
## ------------------------------------------------------------------------------------------
```

```
Plot(Month, Sales, rows=(Product=="Widget"))
```




```
## >>> Suggestions
## Plot(Month, Sales, fit="lm", fit_se=c(.90,.99))  # fit line, standard errors
## Plot(Month, Sales, out_cut=.10)  # label top 10% potential outliers
## Plot(Month, Sales, enhance=TRUE)  # many options
##
##
## >>> Pearson's product-moment correlation
##
```

```
## Number of paired values with neither missing, n = 28
##
##
## Sample Correlation of Month and Sales: r = 0.616
##
##
## Hypothesis Test of 0 Correlation:  t = 3.985,  df = 26,  p-value = 0.000
## 95% Confidence Interval for Correlation:  0.315 to 0.804
```

```
Plot(Month, Sales, rows=(Product=="Widget"), fill="on")
```

3b

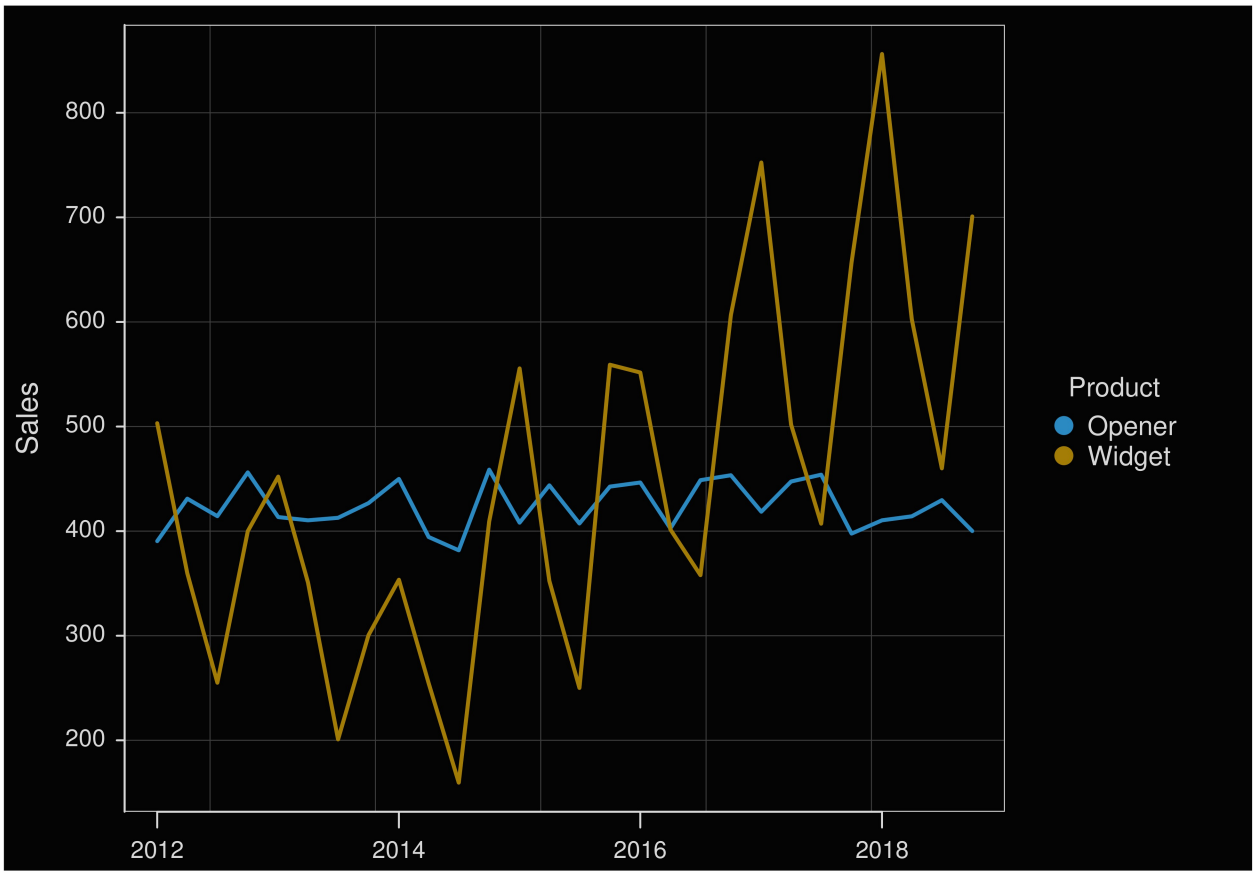
```
## >>> Suggestions
## Plot(Month, Sales, fit="lm", fit_se=c(.90,.99))  # fit line, standard errors
## Plot(Month, Sales, out_cut=.10)  # label top 10% potential outliers
## Plot(Month, Sales, enhance=TRUE)  # many options
##
##
## >>> Pearson's product-moment correlation
##
## Number of paired values with neither missing, n = 28
##
##
## Sample Correlation of Month and Sales: r = 0.616
##
##
```

```
## Hypothesis Test of 0 Correlation:  t = 3.985,  df = 26,  p-value = 0.000
## 95% Confidence Interval for Correlation:  0.315 to 0.804
```

```
style(sub.theme="black", grid.color="gray25", color="steelblue2", fill="steelblue3", trans=.55)
```

```
Plot(Month, Sales, by=Product)
```

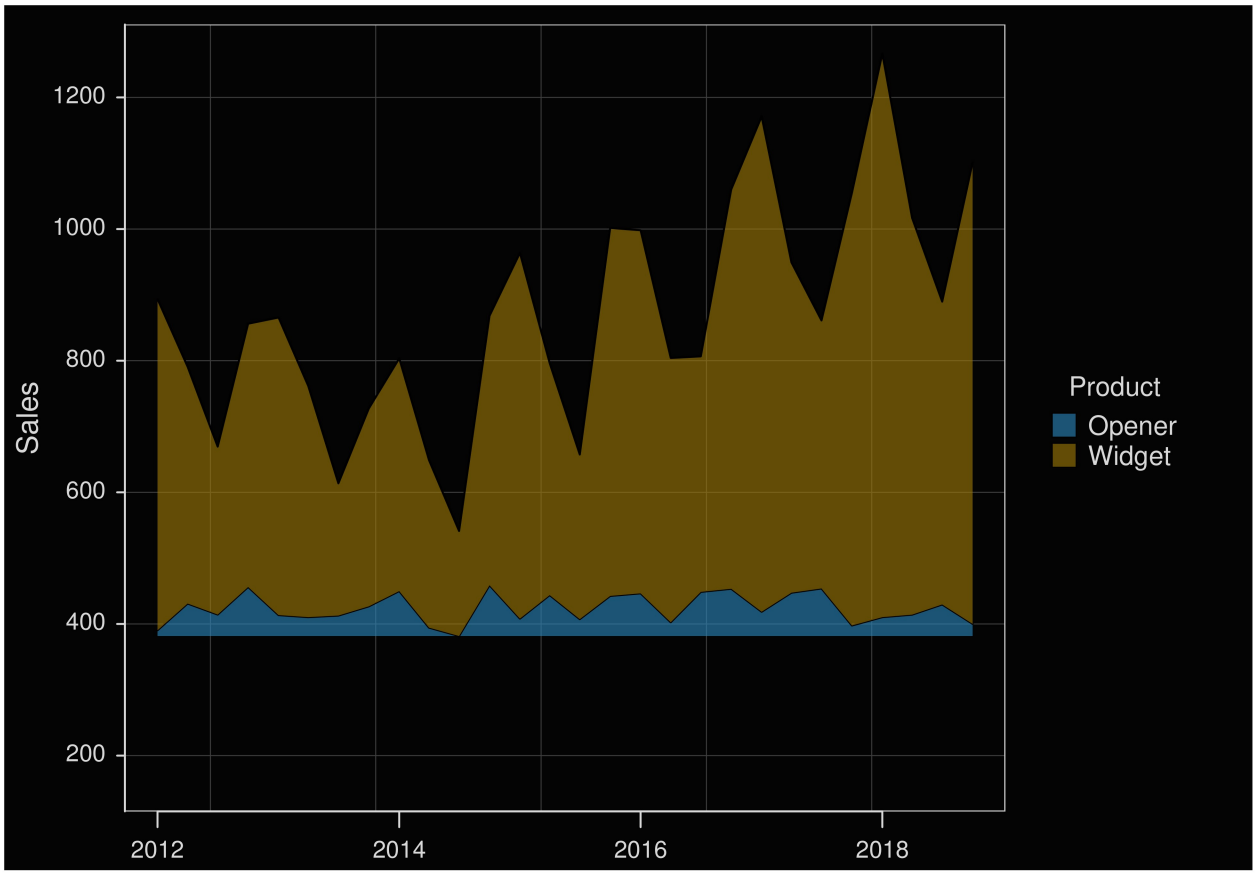
3c




```
## >>> Suggestions
## Plot(Month, Sales, fit="lm", fit_se=c(.90,.99))  # fit line, standard errors
## Plot(Month, Sales, out_cut=.10)  # label top 10% potential outliers
## Plot(Month, Sales, enhance=TRUE)  # many options
##
##
## >>> Pearson's product-moment correlation
##
## Number of paired values with neither missing, n = 56
##
##
## Sample Correlation of Month and Sales: r = 0.437
##
##
## Hypothesis Test of 0 Correlation:  t = 3.565,  df = 54,  p-value = 0.001
## 95% Confidence Interval for Correlation:  0.196 to 0.627
```

```
Plot(Month, Sales, by=Product, stack=TRUE, trans=.4)
```

3d

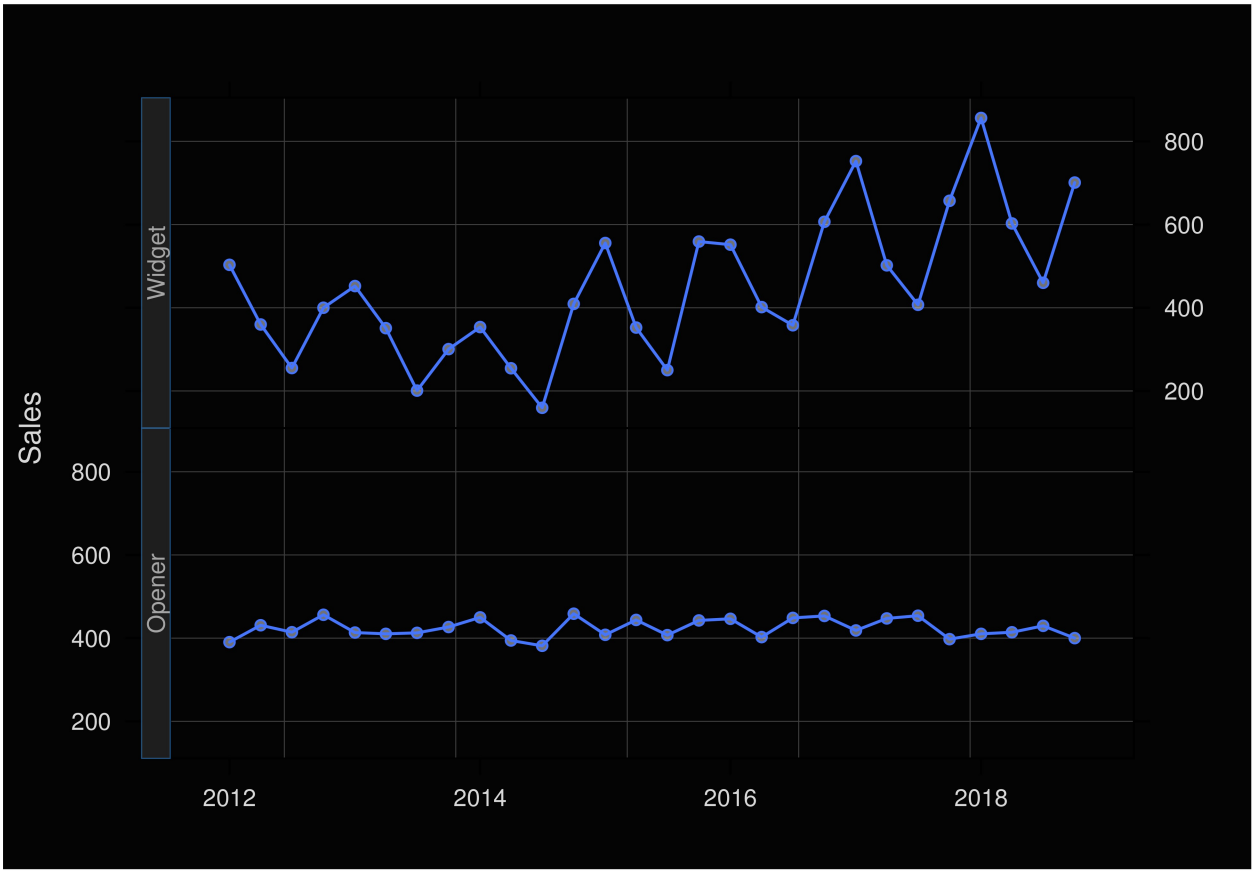



```
## >>> Suggestions
## Plot(Month, Sales, fit="lm", fit_se=c(.90,.99))  # fit line, standard errors
## Plot(Month, Sales, out_cut=.10)  # label top 10% potential outliers
## Plot(Month, Sales, enhance=TRUE)  # many options
##
##
## >>> Pearson's product-moment correlation
##
## Number of paired values with neither missing, n = 56
##
##
## Sample Correlation of Month and Sales: r = 0.437
##
##
## Hypothesis Test of 0 Correlation:  t = 3.565,  df = 54,  p-value = 0.001
## 95% Confidence Interval for Correlation:  0.196 to 0.627
```

```
Plot(Month, Sales, by1=Product)
```

3e

```
## [Trellis graphics from Deepayan Sarkar's lattice package]
```




```
d <- Read("http://web.pdx.edu/~gerbing/451/Data/HW2_3.csv")
```

4a

```
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## character: Non-numeric data values
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1     Month character     56       0      28   1/1/12  4/1/12 ... 7/1/18  10/1/18
##  2     Sales    double     56       0      55   503.2  359.74 ... 429.63  400.07
##  3   Product character     56       0       2   Widget  Widget ... Opener  Opener
## ------------------------------------------------------------------------------------------
```

```
d$Month <- as.Date(d$Month, format="%m/%d/%y")
```

```
str(d$Month)
```

4b

```
##  Date[1:56], format: "2012-01-01" "2012-04-01" "2012-07-01" "2012-10-01" "2013-01-01" "2013-04-01" "
```

HW #3

```
# prob.znorm(mu=50, sigma=10)
# prob.norm(mu=50, sigma=10, lo=50, hi=60)
```

3b

HW #4

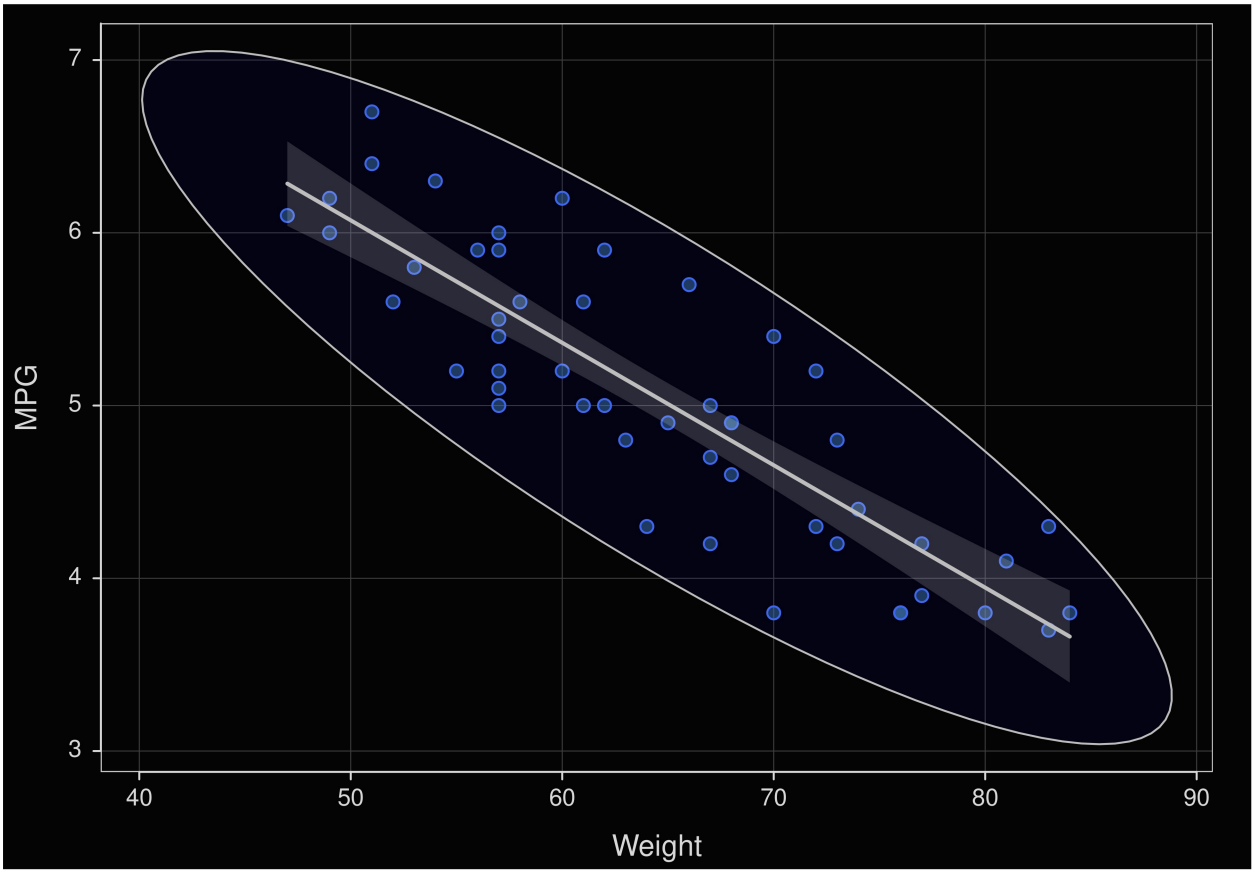
```
d <- Read("http://web.pdx.edu/~gerbing/451/Data/HW8_1.xlsx")
```

1 (plot the time series, run the regression analysis, forecast from X=59,60)

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## integer: Numeric data values, integers only
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1    Weight   integer     50       0      29   73  54  49 ... 63  62  84
##  2       MPG    double     50       0      26   4.2  6.3  6.2 ... 4.8  5.9  3.8
##  3    Cetane   integer     50       0       3   45  55  55 ... 50  55  50
## ------------------------------------------------------------------------------------------
```

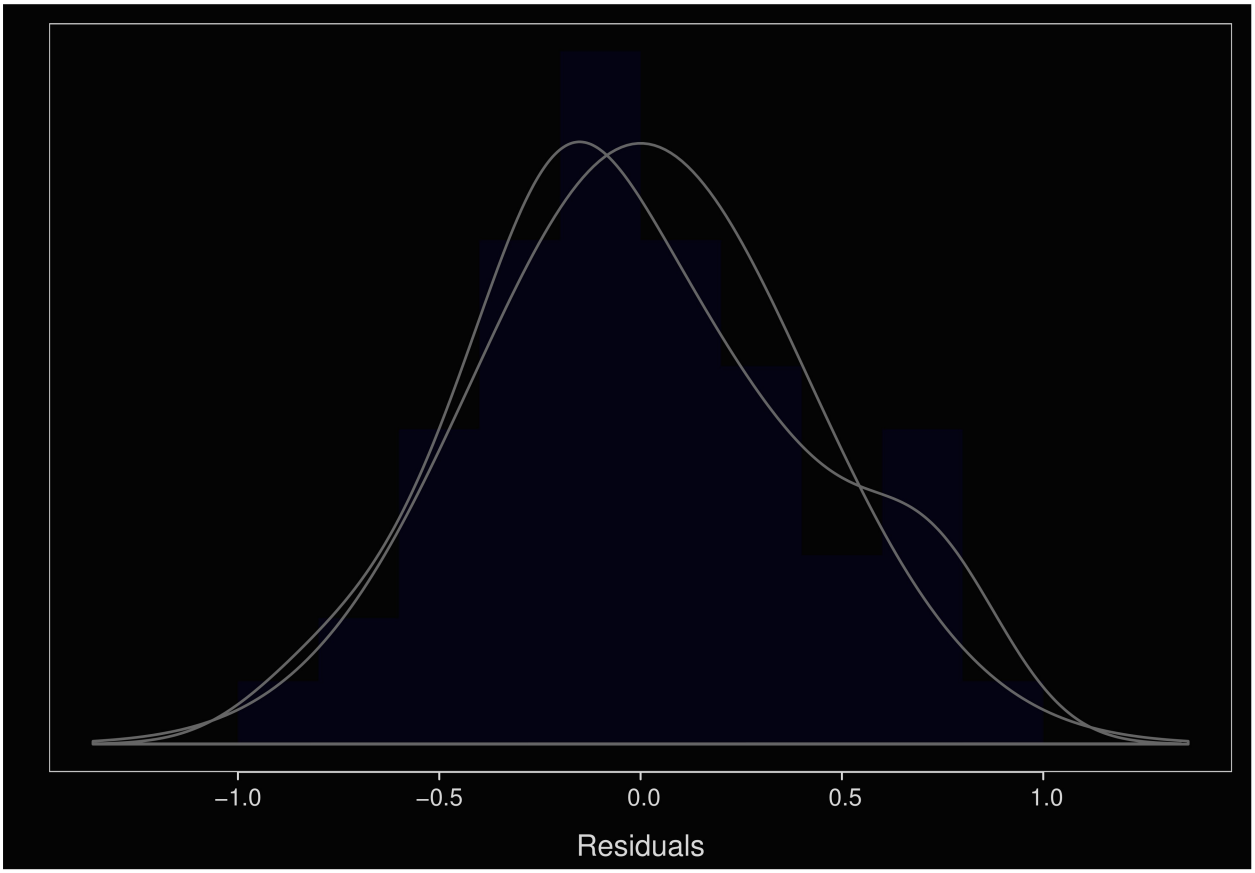
```
sp(Weight, MPG, ellipse=.95, fit="lm")
```

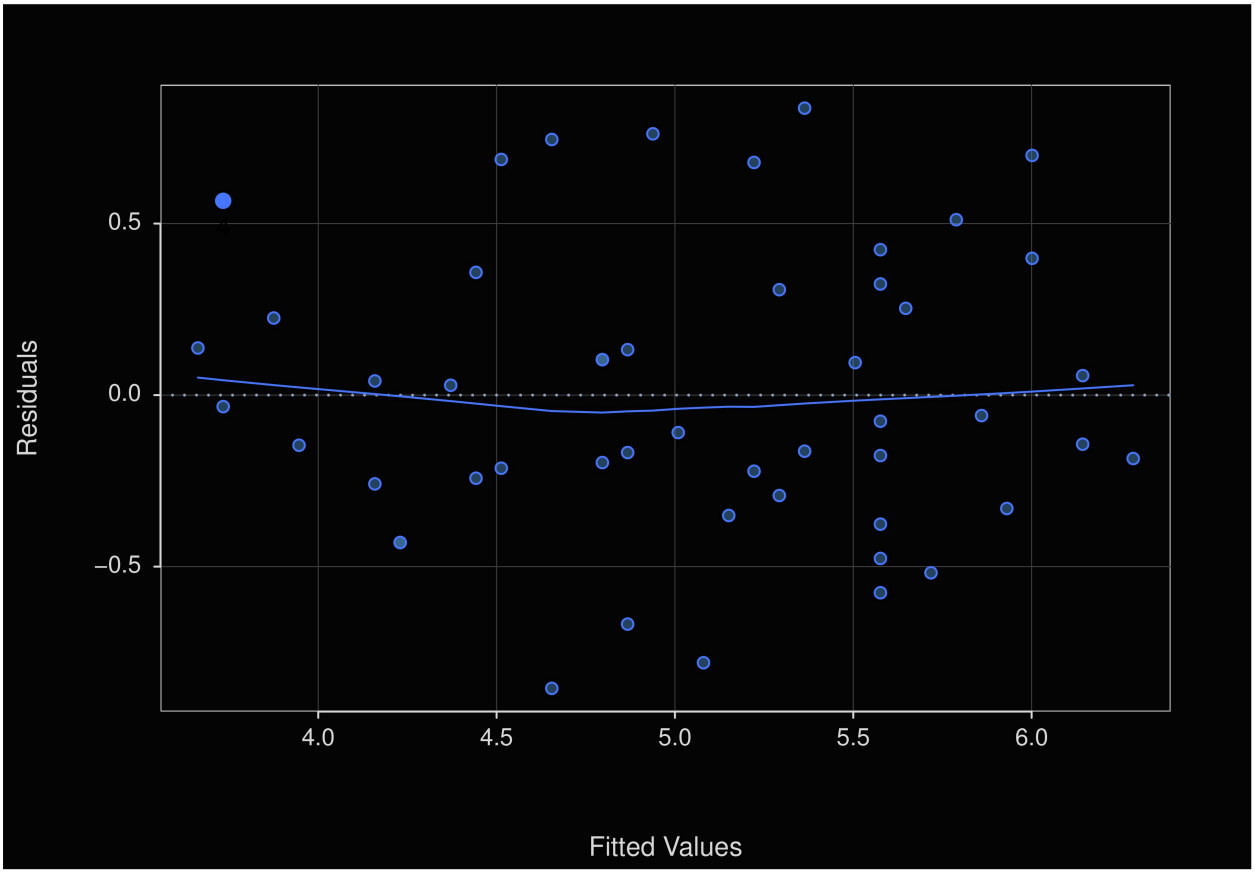
```
## [Ellipse with Murdoch and Chow's function ellipse from the ellipse package]
```

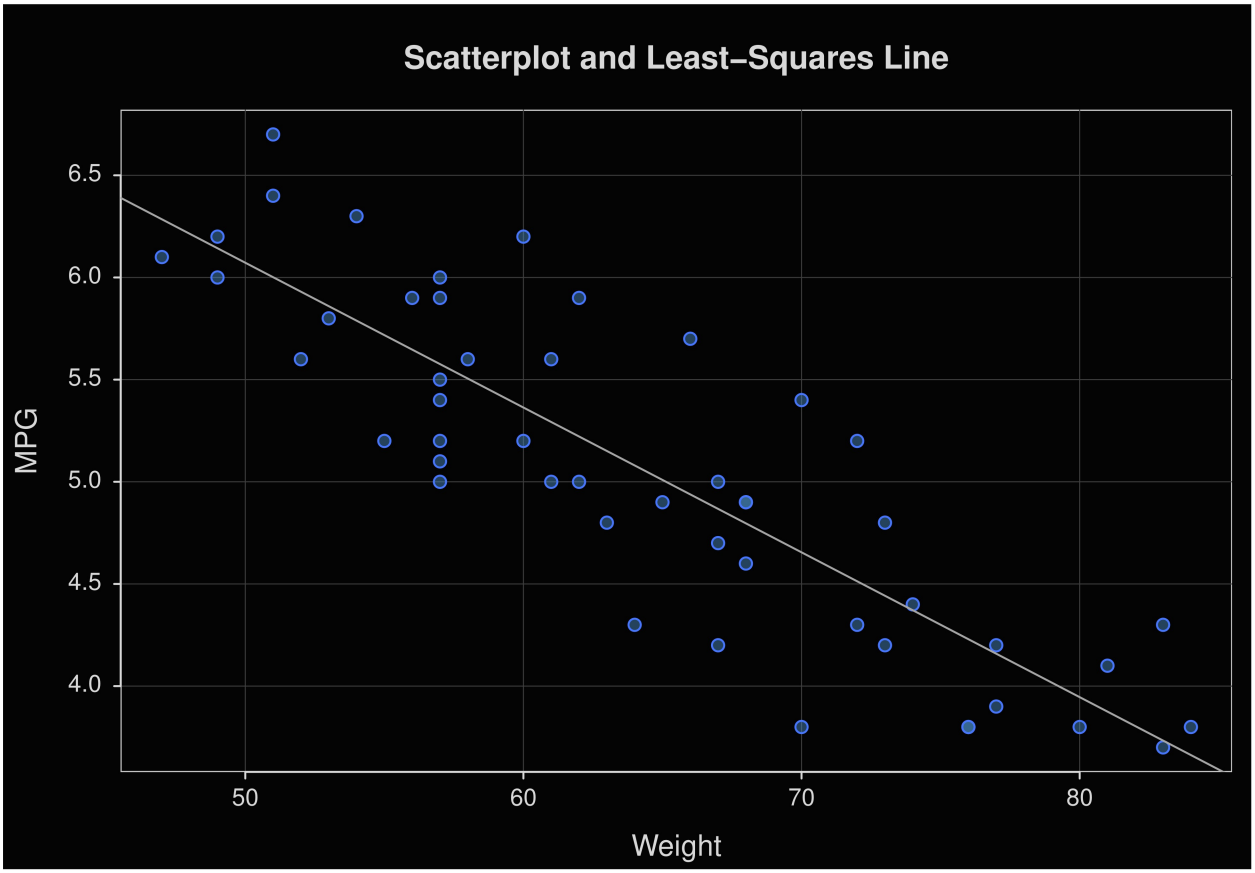



```
## >>> Suggestions
## Plot(Weight, MPG, out_cut=.10)  # label top 10% potential outliers
## Plot(Weight, MPG, enhance=TRUE)  # many options
##
##
## >>> Pearson's product-moment correlation
##
## Number of paired values with neither missing, n = 50
##
##
## Sample Correlation of Weight and MPG: r = -0.860
##
##
## Hypothesis Test of 0 Correlation:  t = -11.676,  df = 48,  p-value = 0.000
## 95% Confidence Interval for Correlation:  -0.92 to -0.76
```

```
Regression(MPG ~ Weight, X1.new=c(59, 60))
```


−1.0
−0.5
0.0
0.5
1.0
Residuals


Residuals
0.5
0.0
−0.5
4.0
4.5
5.0
5.5
6.0
Fitted Values




```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=MPG ~ Weight, X1.new=c(59, 60), Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  d
##
## Response Variable: MPG
## Predictor Variable: Weight
##
## Number of cases (rows) of data:  50
## Number of cases retained for analysis:  50
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)     9.616      0.396   24.288    0.000       8.820      10.412
##      Weight    -0.071      0.006  -11.676    0.000      -0.083      -0.059
##
##
## Standard deviation of residuals:  0.423 for 48 degrees of freedom
##
## R-squared:  0.740    Adjusted R-squared:  0.734    PRESS R-squared:  0.721
```

```
##
## Null hypothesis that all population slope coefficients are 0:
##   F-statistic: 136.330     df: 1 and 48     p-value:  0.000
##
##
##             df    Sum Sq   Mean Sq   F-value   p-value
## Model        1    24.351    24.351   136.330     0.000
## Residuals   48     8.574     0.179
## MPG         49    32.924     0.672
##
##
##   K-FOLD CROSS-VALIDATION
##
##   RELATIONS AMONG THE VARIABLES
##
##            MPG Weight
##      MPG  1.00  -0.86
##   Weight -0.86   1.00
##
##
##   RESIDUALS AND INFLUENCE
##
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 50 rows of data, or do res_rows="all"]
## -------------------------------------------------
##      Weight   MPG fitted  resid rstdnt dffits cooks
##    4     83 4.300  3.734  0.566  1.420  0.449 0.099
##   37     51 6.700  6.001  0.699  1.738  0.429 0.088
##   13     70 3.800  4.655 -0.855 -2.123 -0.349 0.057
##   22     60 6.200  5.363  0.837  2.071  0.326 0.050
##   25     72 5.200  4.513  0.687  1.683  0.304 0.045
##   11     70 5.400  4.655  0.745  1.830  0.301 0.043
##   10     64 4.300  5.080 -0.780 -1.916 -0.274 0.036
##   47     66 5.700  4.938  0.762  1.868  0.270 0.035
##    2     54 6.300  5.789  0.511  1.243  0.262 0.034
##   27     55 5.200  5.718 -0.518 -1.257 -0.252 0.031
##   30     57 5.000  5.576 -0.576 -1.399 -0.252 0.031
##   23     51 6.400  6.001  0.399  0.971  0.240 0.029
##   49     62 5.900  5.222  0.678  1.651  0.243 0.029
##   17     67 4.200  4.867 -0.667 -1.623 -0.240 0.028
##   35     76 3.800  4.230 -0.430 -1.042 -0.232 0.027
##   38     76 3.800  4.230 -0.430 -1.042 -0.232 0.027
##   12     57 5.100  5.576 -0.476 -1.149 -0.207 0.021
##   42     52 5.600  5.930 -0.330 -0.800 -0.188 0.018
##    5     57 6.000  5.576  0.424  1.020  0.184 0.017
##   32     73 4.800  4.442  0.358  0.859  0.164 0.013
##
##
##   FORECASTING ERROR
##
## Data, Predicted, Standard Error of Forecast, 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
## -----------------------------------------------
```

```
##     Weight MPG  pred    sf pi:lwr pi:upr width
##   2 60.000     5.363 0.428  4.504  6.223 1.720
##   1 59.000     5.434 0.428  4.574  6.295 1.722
##
##
## ----------------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: Scatterplot and Least-Squares Line
## ----------------------------------------
```

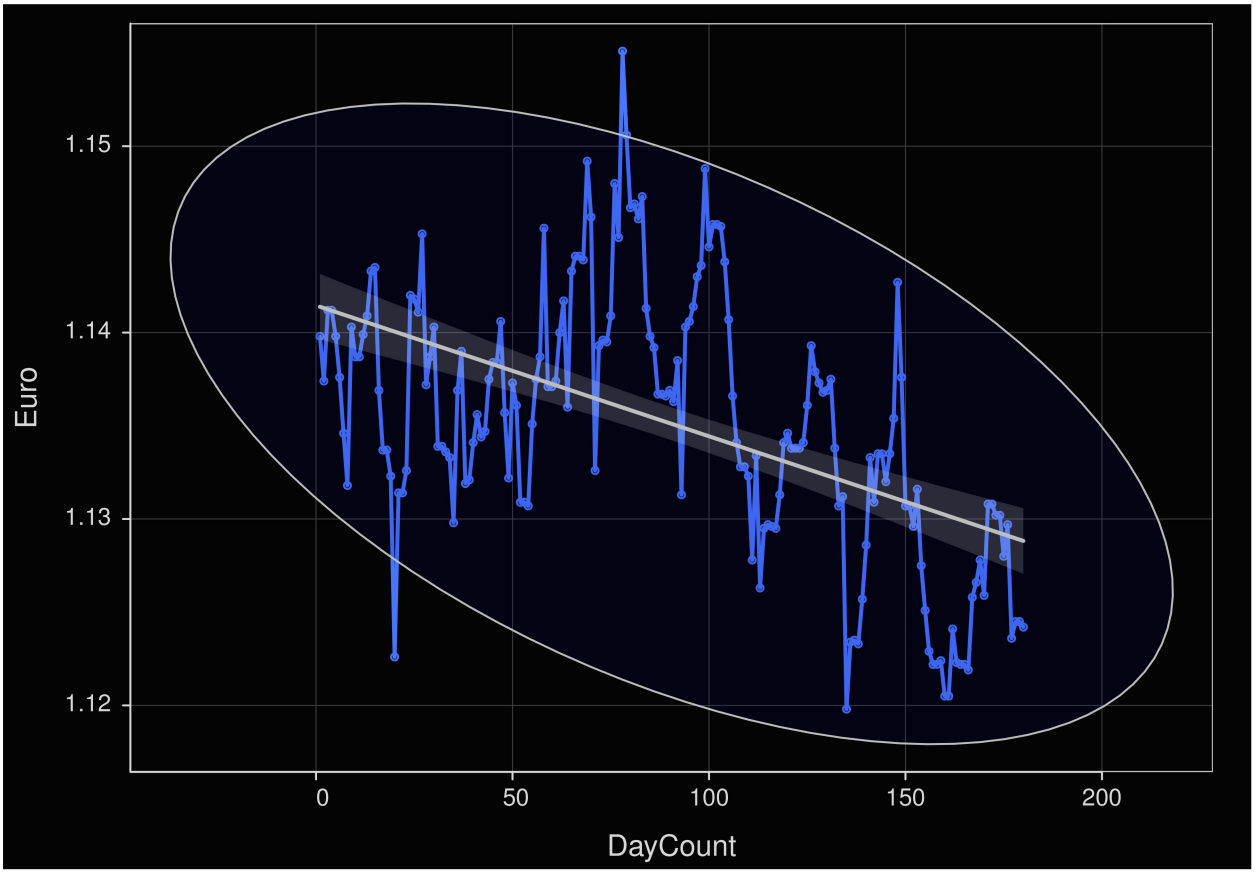
```
d <- Read("http://web.pdx.edu/~gerbing/451/Data/HW4_2.xlsx")
```

2 (plot the time sereis, run the regression analysis, no forecast needed)

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## Date: Date with year, month and day
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                  Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ---------------------------------------------------------------------------------------------
##  1       Day      Date    180       0     180   2018-10-24 ... 2019-04-21
##  2      Euro    double    180       0     120   1.1398  1.1374 ... 1.1245  1.1242
##  3     Pound    double    180       0     150   1.2883  1.2822 ... 1.2996  1.2995
## ---------------------------------------------------------------------------------------------
```

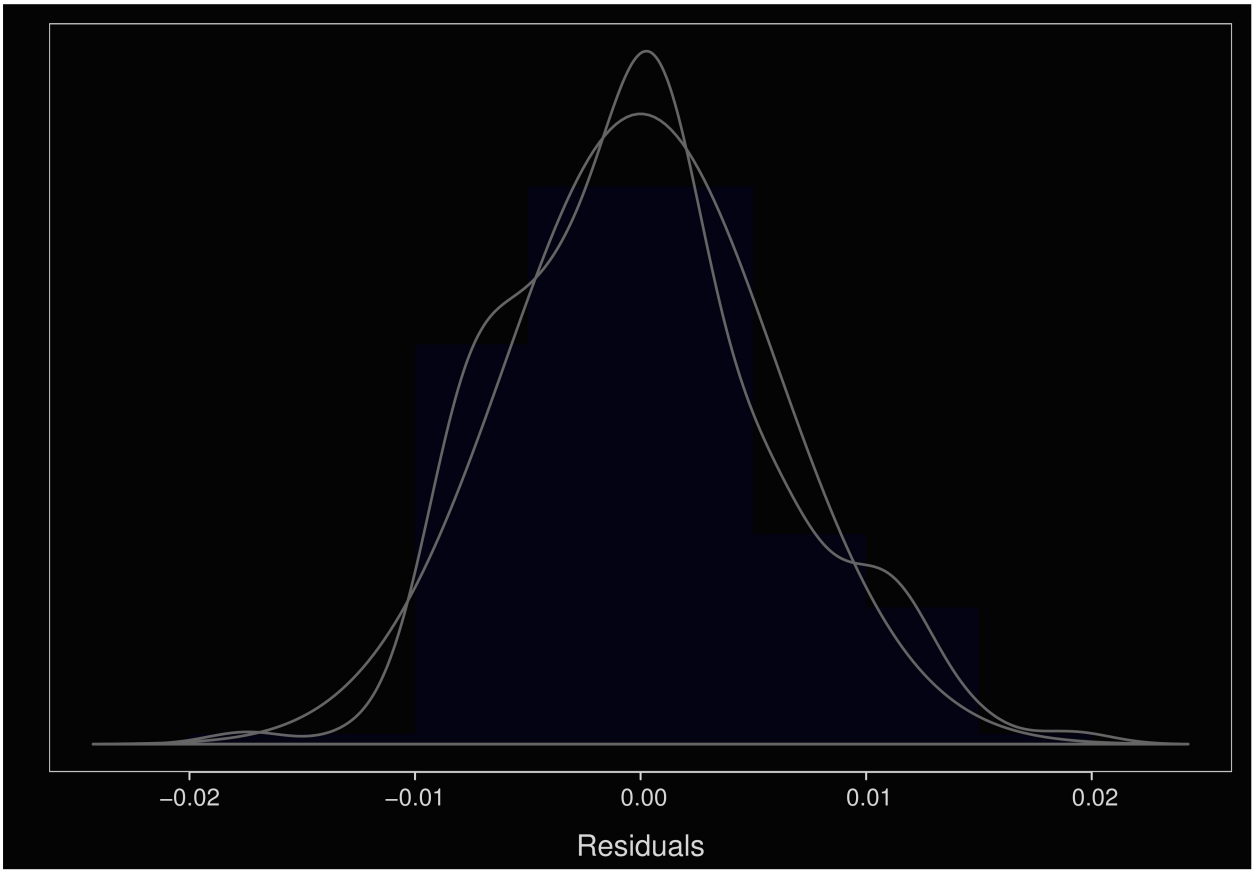
```
d$DayCount <- 1:nrow(d)
sp(DayCount, Euro, ellipse=.95, fit="lm")
```

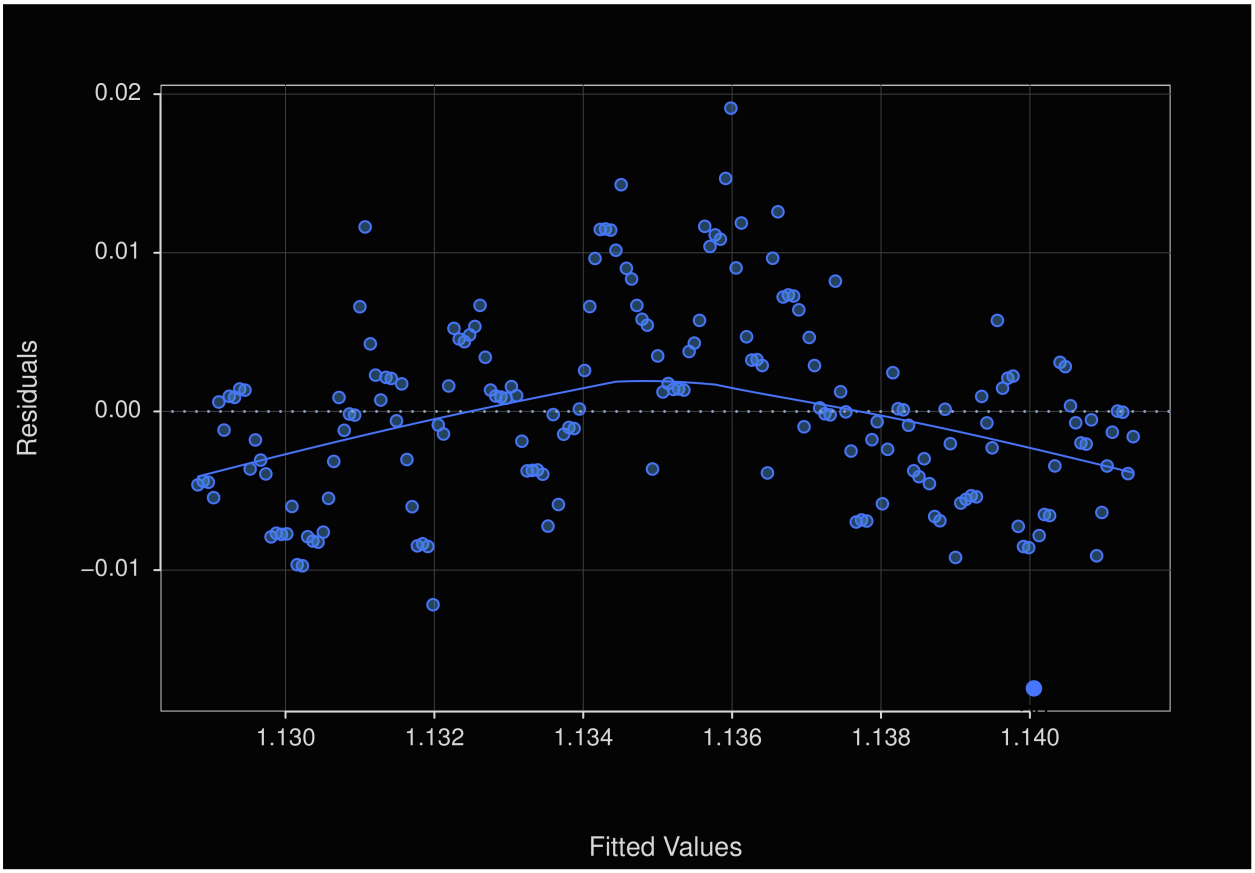
```
## [Ellipse with Murdoch and Chow's function ellipse from the ellipse package]
```

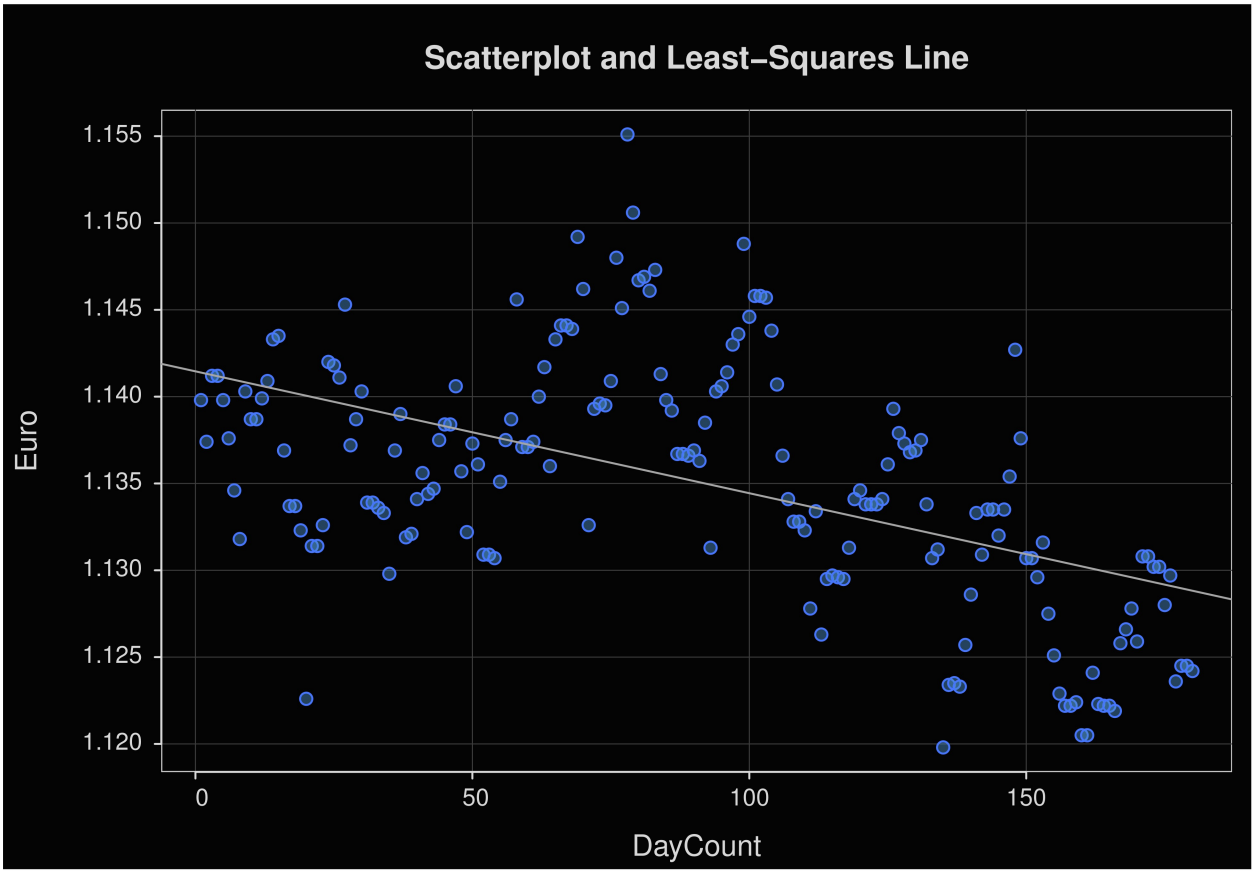



```
## >>> Suggestions
## Plot(DayCount, Euro, out_cut=.10)  # label top 10% potential outliers
## Plot(DayCount, Euro, enhance=TRUE)  # many options
##
##
## >>> Pearson's product-moment correlation
##
## Number of paired values with neither missing, n = 180
##
##
## Sample Correlation of DayCount and Euro: r = -0.521
##
##
## Hypothesis Test of 0 Correlation:  t = -8.140,  df = 178,  p-value = 0.000
## 95% Confidence Interval for Correlation:  -0.62000 to -0.40500
```

```
Regression(Euro ~ DayCount, X1.new=240)
```


−0.02
−0.01
0.00
0.01
0.02
Residuals


0.02
0.01
0.00
−0.01
Residuals
1.130
1.132
1.134
1.136
1.138
1.140
Fitted Values




```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=Euro ~ DayCount, X1.new=240, Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  d
##
## Response Variable: Euro
## Predictor Variable: DayCount
##
## Number of cases (rows) of data:  180
## Number of cases retained for analysis:  180
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)   1.14146    0.00090 1268.419    0.000     1.13968     1.14323
##    DayCount  -0.00007    0.00001   -8.140    0.000    -0.00009    -0.00005
##
##
## Standard deviation of residuals:  0.00601 for 178 degrees of freedom
##
## R-squared:  0.271    Adjusted R-squared:  0.267    PRESS R-squared:  0.257
```

```
##
## Null hypothesis that all population slope coefficients are 0:
##   F-statistic: 66.255     df: 1 and 178     p-value:  0.000
##
##
##              df    Sum Sq   Mean Sq   F-value   p-value
## Model         1   0.00239   0.00239  66.25464     0.000
## Residuals   178   0.00643   0.00004
## Euro        179   0.00883   0.00005
##
##
##   K-FOLD CROSS-VALIDATION
##
##   RELATIONS AMONG THE VARIABLES
##
##             Euro DayCount
##       Euro  1.00    -0.52
##   DayCount -0.52     1.00
##
##
##   RESIDUALS AND INFLUENCE
##
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 180 rows of data, or do res_rows="all"]
## ------------------------------------------------------------------
##        DayCount    Euro  fitted    resid   rstdnt   dffits   cooks
##    20        20 1.12260 1.14005 -0.01745 -2.99120 -0.37878 0.06867
##    78        78 1.15510 1.13598  0.01912  3.27540  0.25184 0.03007
##   148       148 1.14270 1.13107  0.01163  1.96226  0.21951 0.02371
##     8         8 1.13180 1.14090 -0.00910 -1.53395 -0.21667 0.02330
##   161       161 1.12050 1.13016 -0.00966 -1.62669 -0.20599 0.02102
##   160       160 1.12050 1.13023 -0.00973 -1.63845 -0.20555 0.02093
##   135       135 1.11980 1.13198 -0.01218 -2.05458 -0.20260 0.02016
##    79        79 1.15060 1.13591  0.01469  2.48568  0.19031 0.01760
##    21        21 1.13140 1.13998 -0.00858 -1.44345 -0.18109 0.01630
##    99        99 1.14880 1.13451  0.01429  2.41624  0.18301 0.01630
##    22        22 1.13140 1.13991 -0.00851 -1.43130 -0.17788 0.01573
##   166       166 1.12190 1.12981 -0.00791 -1.32950 -0.17632 0.01548
##    69        69 1.14920 1.13661  0.01259  2.12088  0.17164 0.01445
##   165       165 1.12220 1.12988 -0.00768 -1.29028 -0.16956 0.01432
##   164       164 1.12220 1.12995 -0.00775 -1.30199 -0.16953 0.01431
##    35        35 1.12980 1.13900 -0.00920 -1.54575 -0.16959 0.01427
##   158       158 1.12220 1.13037 -0.00817 -1.37228 -0.16895 0.01420
##   157       157 1.12220 1.13044 -0.00824 -1.38401 -0.16879 0.01417
##    19        19 1.13230 1.14012 -0.00782 -1.31480 -0.16806 0.01406
##   163       163 1.12230 1.13002 -0.00772 -1.29673 -0.16729 0.01394
##
##
##   FORECASTING ERROR
##
## Data, Predicted, Standard Error of Forecast, 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
## ------------------------------------------------------------
```

```
##      DayCount Euro    pred      sf  pi:lwr  pi:upr   width
##   1 240.00000      1.12461 0.00616 1.11245 1.13678 0.02433
##
##
## ----------------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: Scatterplot and Least-Squares Line
## ----------------------------------------
```

HW #6

```
d <- rd("http://web.pdx.edu/~gerbing/451/Data/Sales07.xlsx")
```

1a

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## character: Non-numeric data values
## integer: Numeric data values, integers only
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ---------------------------------------------------------------------------------------
##  1      Year   integer     16       0       4   2016  2016  2016 ... 2019  2019  2019
##  2       Qtr character     16       0       4   Winter  Spring ... Summer  Fall
##  3     Sales    double     16       0      15   0.41  0.65  0.96 ... 2.11  2.25  1.74
## ---------------------------------------------------------------------------------------
```

```
d
```

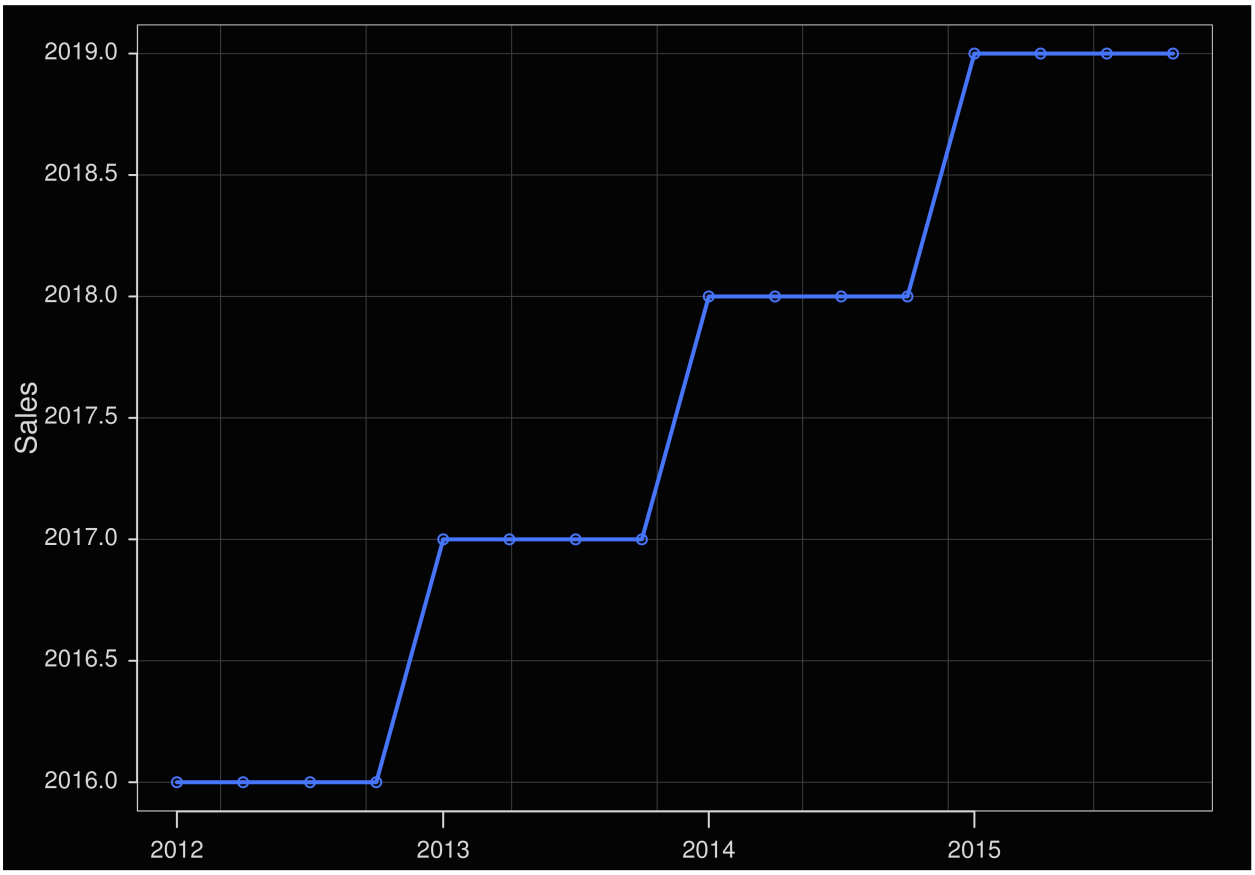
```
##    Year    Qtr Sales
## 1  2016 Winter  0.41
## 2  2016 Spring  0.65
## 3  2016 Summer  0.96
## 4  2016   Fall  0.57
## 5  2017 Winter  0.59
## 6  2017 Spring  1.20
## 7  2017 Summer  1.53
## 8  2017   Fall  0.97
## 9  2018 Winter  0.93
## 10 2018 Spring  1.71
## 11 2018 Summer  1.74
## 12 2018   Fall  1.42
## 13 2019 Winter  1.36
## 14 2019 Spring  2.11
```

```
## 15 2019 Summer  2.25
## 16 2019   Fall  1.74
```

```
Sales <- ts(d$Y, frequency=4, start=c(2012,1))
Plot(Sales)
```

1b

```
## >>> Note: Sales is from the workspace, not in a data frame (table)
```




```
## >>> Suggestions
## Plot(Sales, Sales, fit="lm", fit_se=c(.90,.99))  # fit line, standard errors
## Plot(Sales, Sales, out_cut=.10)  # label top 10% potential outliers
## Plot(Sales, Sales, enhance=TRUE)  # many options
##
##
## >>> Pearson's product-moment correlation
##
## Number of paired values with neither missing, n = 16
##
##
## Sample Correlation of Sales and Sales: r = 0.970
##
##
## Hypothesis Test of 0 Correlation:  t = 14.997,  df = 14,  p-value = 0.000
## 95% Confidence Interval for Correlation:  0.9 to 1.0
```

```
fit_Sales <- tslm(Sales ~ trend + season)
f <- forecast(fit_Sales, h=8)
f
```

1g

```
##         Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## 2016 Q1           2020  2020  2020  2020  2020
## 2016 Q2           2020  2020  2020  2020  2020
## 2016 Q3           2020  2020  2020  2020  2020
## 2016 Q4           2020  2020  2020  2020  2020
## 2017 Q1           2021  2021  2021  2021  2021
## 2017 Q2           2021  2021  2021  2021  2021
## 2017 Q3           2021  2021  2021  2021  2021
## 2017 Q4           2021  2021  2021  2021  2021
```

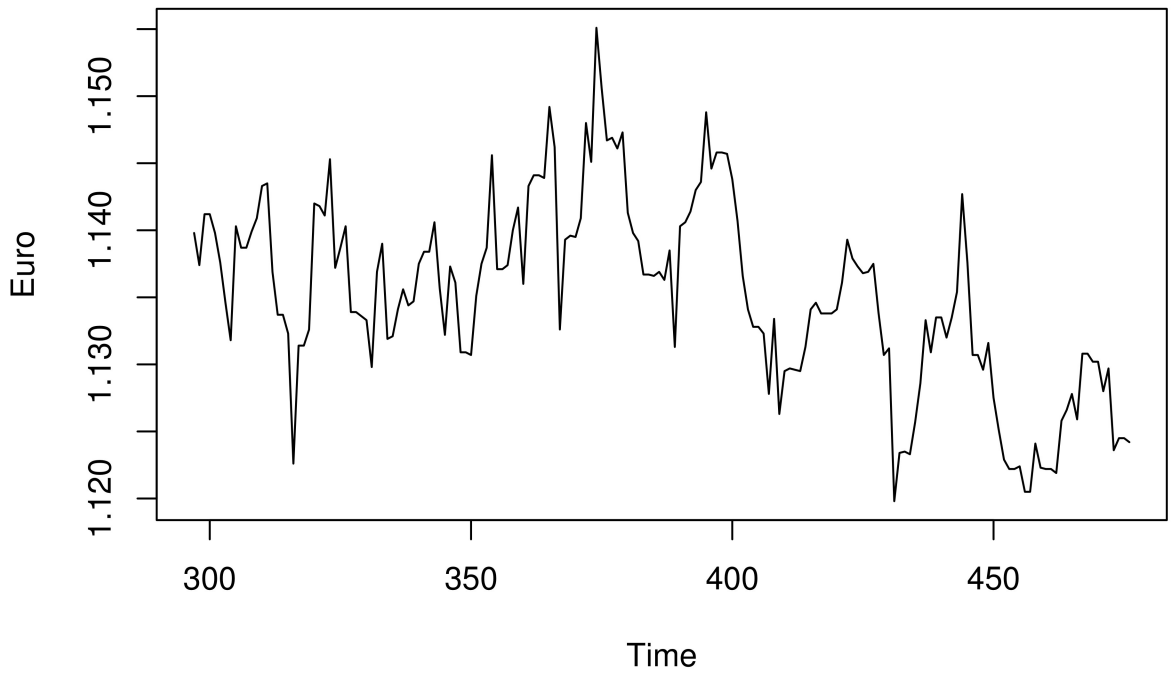
```
d <- rd("http://web.pdx.edu/~gerbing/451/Data/HW4_2.xlsx")
```

3a

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## Date: Date with year, month and day
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                  Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1       Day      Date    180       0     180   2018-10-24 ... 2019-04-21
##  2      Euro    double    180       0     120   1.1398  1.1374 ... 1.1245  1.1242
##  3     Pound    double    180       0     150   1.2883  1.2822 ... 1.2996  1.2995
## ------------------------------------------------------------------------------------------
```

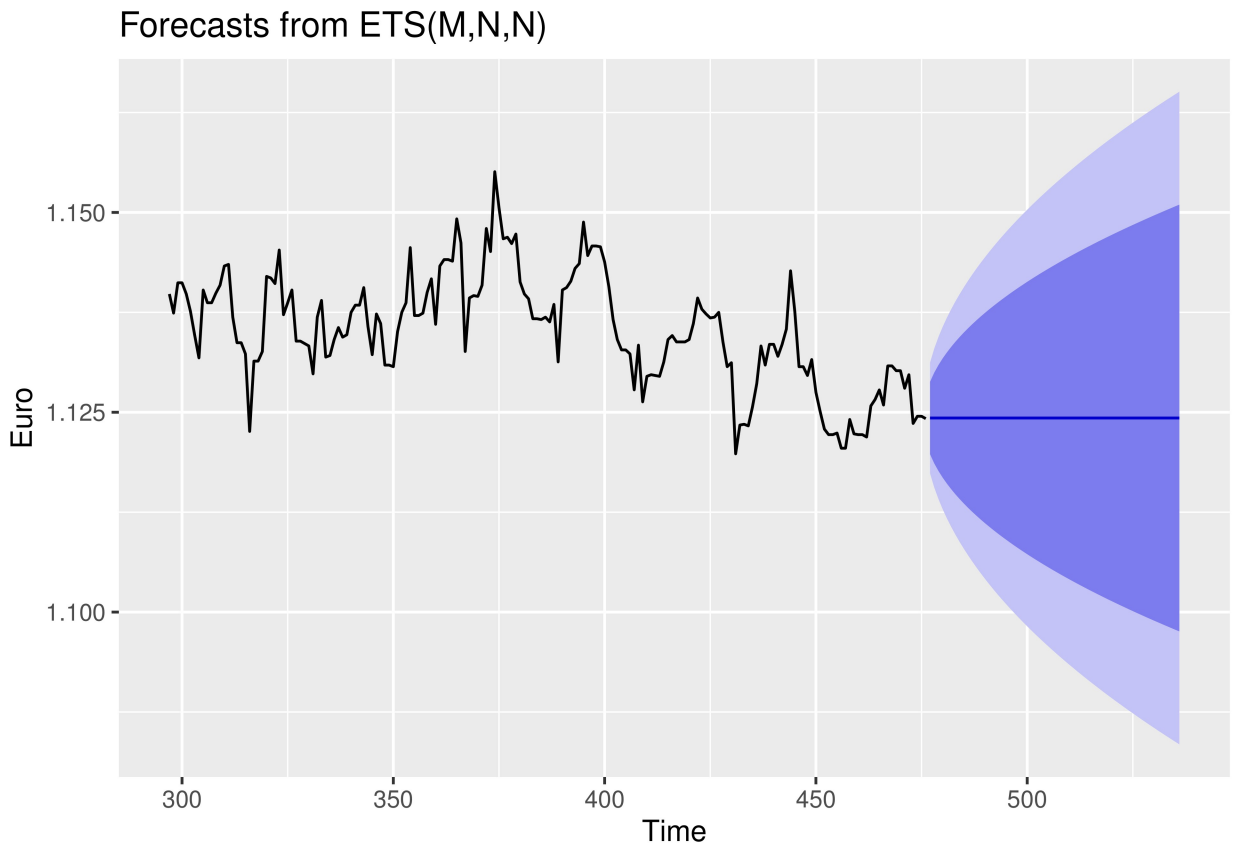
```
Euro <- ts(d$Euro, start=c(297,1))
```

```
plot(Euro)
```



3b

```
f <- forecast(Euro, h=60)
autoplot(f, ylab="Euro")
```



3d

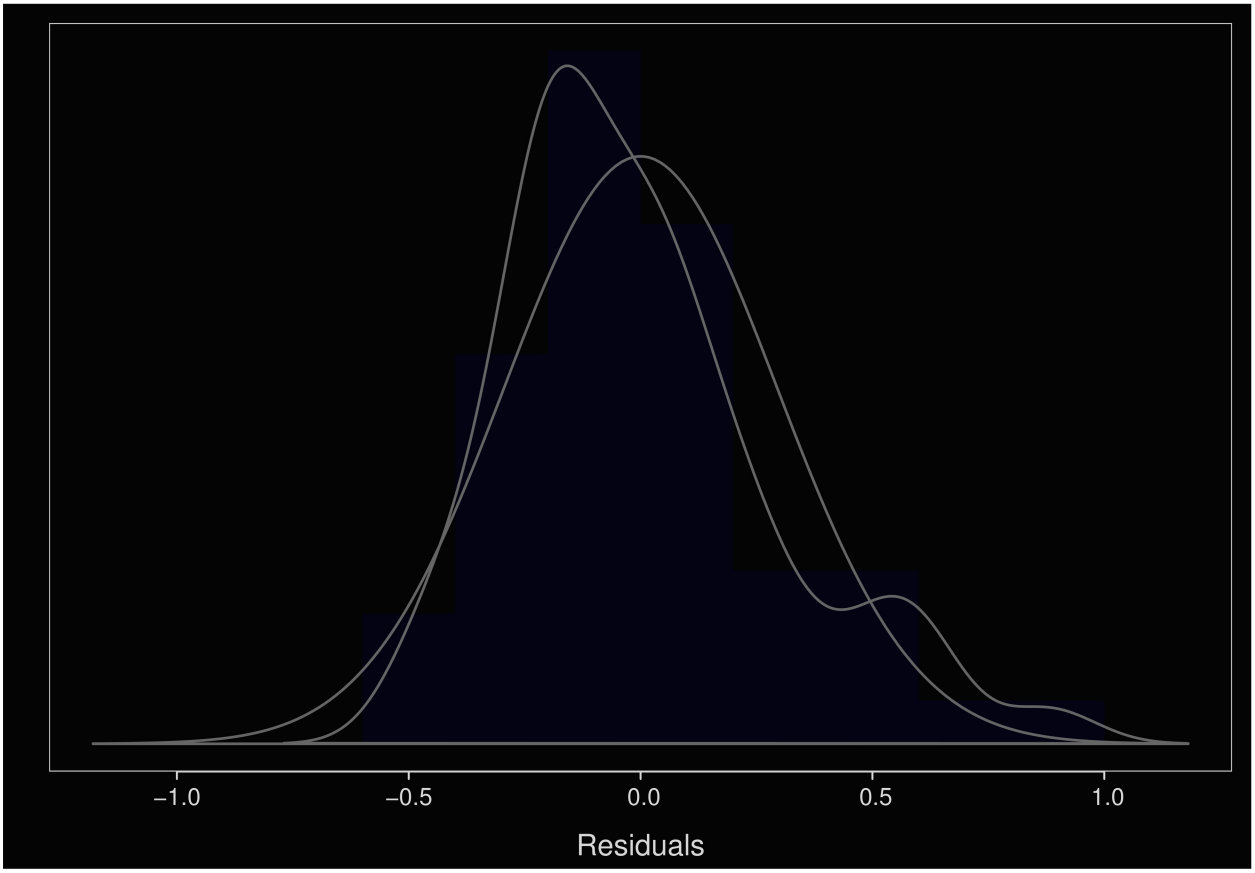
HW #8

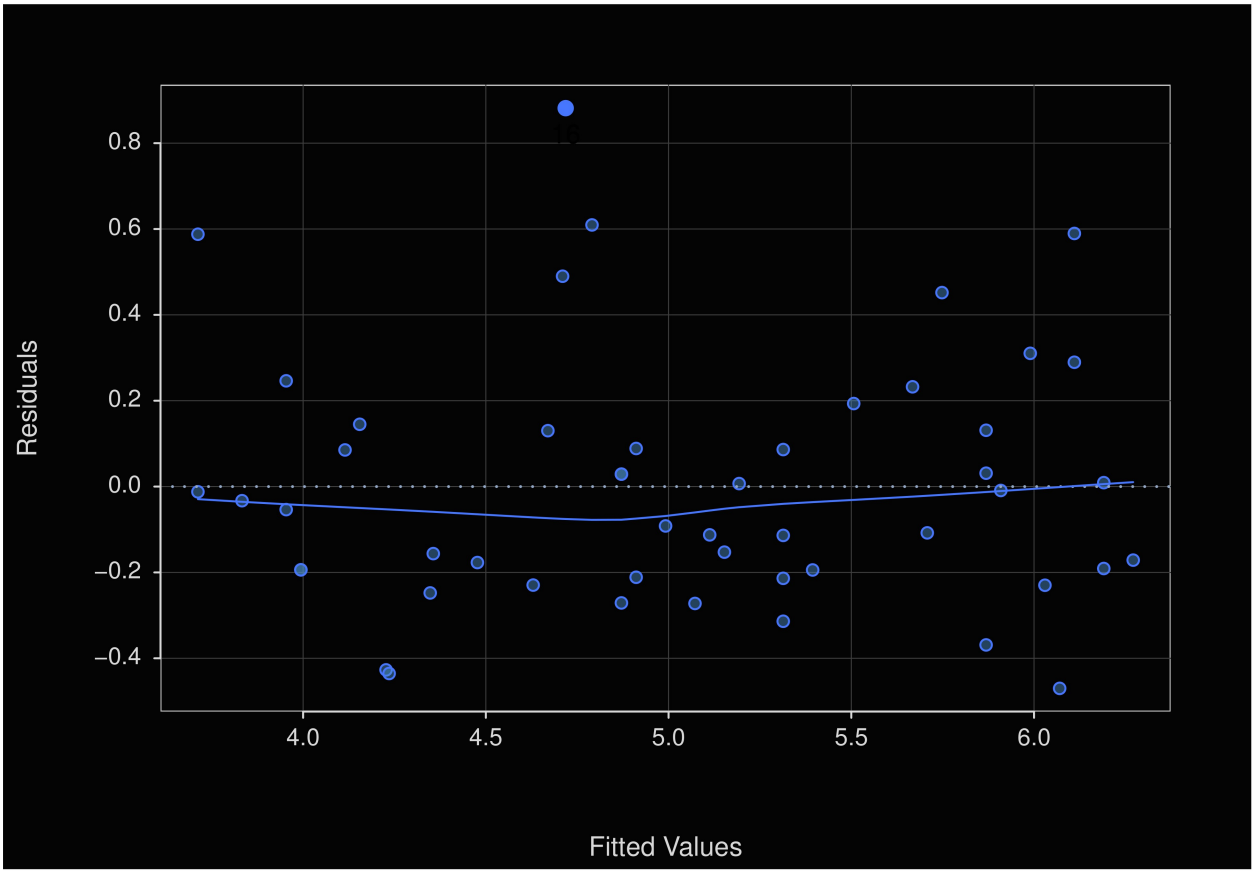
```
d <- rd("http://web.pdx.edu/~gerbing/451/Data/HW8_1.xlsx")
```

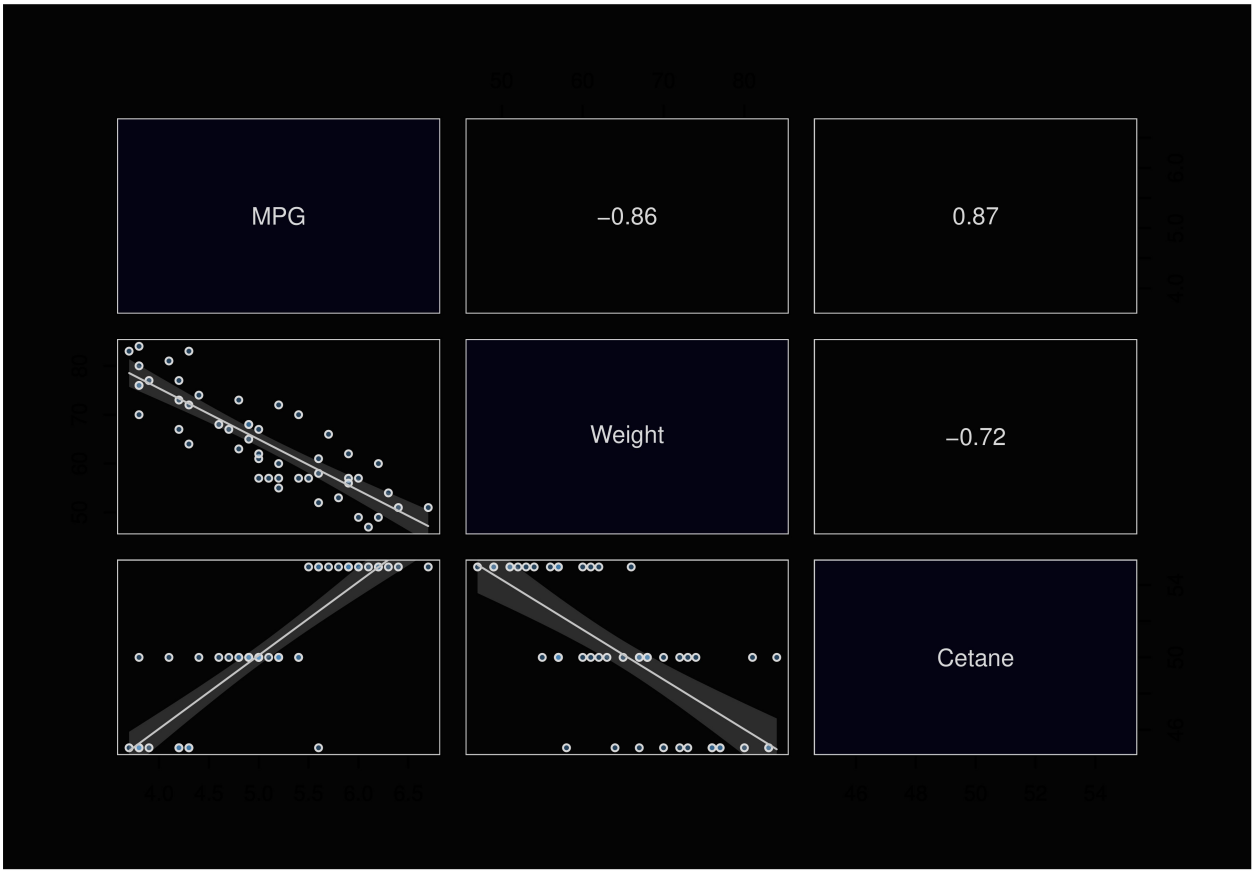
1a-ii (multiple regression with forecast)

```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## integer: Numeric data values, integers only
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ---------------------------------------------------------------------------------------
##  1    Weight   integer     50       0      29   73  54  49 ... 63  62  84
##  2       MPG    double     50       0      26   4.2  6.3  6.2 ... 4.8  5.9  3.8
##  3    Cetane   integer     50       0       3   45  55  55 ... 50  55  50
## ---------------------------------------------------------------------------------------
```

```
Regression(MPG ~ Weight + Cetane, X1.new = 59, X2.new = c(50, 55))
```




Residuals
Fitted Values
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
4.0
4.5
5.0
5.5
6.0




```
## >>> Suggestion
## # Create an R markdown file for interpretative output with  Rmd = "file_name"
## Regression(my_formula=MPG ~ Weight + Cetane, X1.new=59, X2.new=c(50, 55), Rmd="eg")
##
##
##   BACKGROUND
##
## Data Frame:  d
##
## Response Variable: MPG
## Predictor Variable 1: Weight
## Predictor Variable 2: Cetane
##
## Number of cases (rows) of data:  50
## Number of cases retained for analysis:  50
##
##
##   BASIC ANALYSIS
##
##              Estimate    Std Err  t-value  p-value   Lower 95%   Upper 95%
## (Intercept)     2.057      1.126    1.827    0.074      -0.208       4.322
##      Weight    -0.040      0.006   -6.520    0.000      -0.053      -0.028
##      Cetane     0.111      0.016    6.933    0.000       0.079       0.143
##
##
## Standard deviation of residuals:  0.300 for 47 degrees of freedom
```

```
##
## R-squared:  0.871    Adjusted R-squared:  0.866    PRESS R-squared:  0.845
##
## Null hypothesis that all population slope coefficients are 0:
##   F-statistic: 159.027     df: 2 and 47     p-value:  0.000
##
##
##             df    Sum Sq   Mean Sq   F-value   p-value
##    Weight    1    24.351    24.351   269.993     0.000
##    Cetane    1     4.335     4.335    48.061     0.000
##
## Model        2    28.685    14.343   159.027     0.000
## Residuals   47     4.239     0.090
## MPG         49    32.924     0.672
##
##
##   K-FOLD CROSS-VALIDATION
##
##   RELATIONS AMONG THE VARIABLES
##
##            MPG Weight Cetane
##      MPG  1.00  -0.86   0.87
##   Weight -0.86   1.00  -0.72
##   Cetane  0.87  -0.72   1.00
##
##
##          Tolerance       VIF
##   Weight     0.488     2.049
##   Cetane     0.488     2.049
##
##
##  Weight Cetane    R2adj    X's
##       1      1    0.866      2
##       0      1    0.750      1
##       1      0    0.734      1
##
## [based on Thomas Lumley's leaps function from the leaps package]
##
##
##
##   RESIDUALS AND INFLUENCE
##
## Data, Fitted, Residual, Studentized Residual, Dffits, Cook's Distance
##    [sorted by Cook's Distance]
##    [res_rows = 20, out of 50 rows of data, or do res_rows="all"]
## ------------------------------------------------------------
##      Weight Cetane   MPG fitted  resid rstdnt dffits cooks
##   16     58     45 5.600  4.718  0.882  3.615  1.645 0.717
##   50     84     50 3.800  4.227 -0.427 -1.588 -0.724 0.169
##    4     83     45 4.300  3.712  0.588  2.129  0.673 0.140
##   37     51     55 6.700  6.110  0.590  2.097  0.531 0.088
##   13     70     45 3.800  4.235 -0.435 -1.522 -0.407 0.054
##   42     52     55 5.600  6.070 -0.470 -1.640 -0.402 0.052
##   22     60     55 6.200  5.748  0.452  1.575  0.392 0.050
```

```
##   11     70     50 5.400  4.791  0.609  2.138  0.379 0.045
##   25     72     50 5.200  4.710  0.490  1.699  0.350 0.039
##   18     81     50 4.100  4.348 -0.248 -0.881 -0.337 0.038
##   24     57     55 5.500  5.869 -0.369 -1.269 -0.295 0.029
##   23     51     55 6.400  6.110  0.290  0.995  0.252 0.021
##    2     54     55 6.300  5.990  0.310  1.063  0.249 0.021
##   30     57     50 5.000  5.314 -0.314 -1.072 -0.239 0.019
##   47     66     55 5.700  5.507  0.193  0.673  0.218 0.016
##   40     77     45 4.200  3.954  0.246  0.844  0.217 0.016
##   49     62     55 5.900  5.668  0.232  0.798  0.214 0.015
##   10     64     45 4.300  4.477 -0.177 -0.618 -0.211 0.015
##   31     74     50 4.400  4.630 -0.230 -0.783 -0.187 0.012
##   20     53     55 5.800  6.030 -0.230 -0.784 -0.187 0.012
##
##
##   FORECASTING ERROR
##
## Data, Predicted, Standard Error of Forecast, 95% Prediction Intervals
##    [sorted by lower bound of prediction interval]
## -------------------------------------------------------
##     Weight Cetane MPG  pred    sf pi:lwr pi:upr width
##   1 59.000 50.000     5.233 0.306  4.618  5.848 1.230
##   2 59.000 55.000     5.788 0.308  5.168  6.409 1.241
##
##
## ---------------------------------
## Plot 1: Distribution of Residuals
## Plot 2: Residuals vs Fitted Values
## Plot 3: ScatterPlot Matrix
## ---------------------------------
```

HW #9

```
d <- Read("http://web.pdx.edu/~gerbing/451/Data/aus_tourists.xlsx")
```

1a

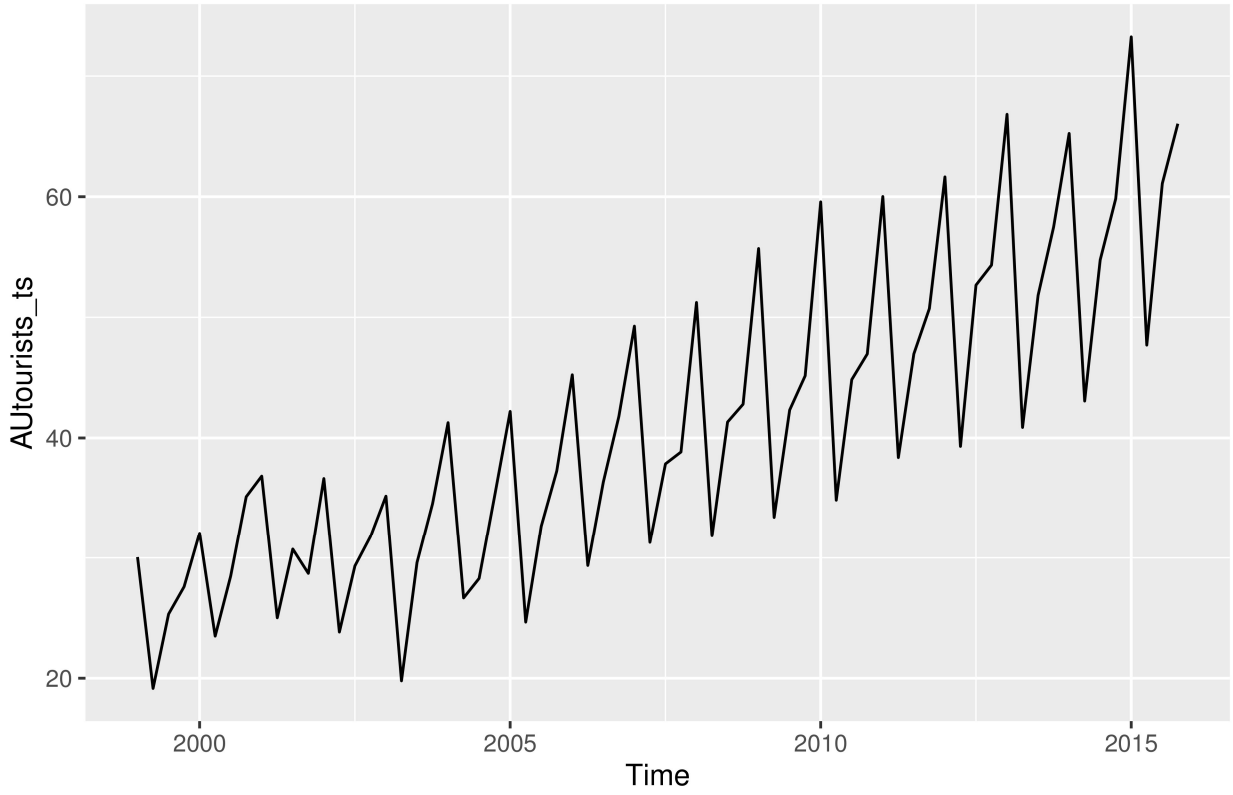
```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## Date: Date with year, month and day
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                  Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1      date      Date     68       0      68   1999-01-01 ... 2015-10-01
```

```
##  2  tourists    double     68       0      68   30.052513 ... 66.0557612187001
## ------------------------------------------------------------------------------------------
```

```
AUtourists_ts <- ts(d$tourists, frequency=4, start=c(1999,1))
AUtourists_ts
```

```
##          Qtr1     Qtr2     Qtr3     Qtr4
## 1999 30.05251 19.14850 25.31769 27.59144
## 2000 32.07646 23.48796 28.47594 35.12375
## 2001 36.83848 25.00702 30.72223 28.69376
## 2002 36.64099 23.82461 29.31168 31.77031
## 2003 35.17788 19.77524 29.60175 34.53884
## 2004 41.27360 26.65586 28.27986 35.19115
## 2005 42.20566 24.64917 32.66734 37.25735
## 2006 45.24246 29.35048 36.34421 41.78208
## 2007 49.27660 31.27540 37.85063 38.83704
## 2008 51.23690 31.83855 41.32342 42.79900
## 2009 55.70836 33.40714 42.31664 45.15712
## 2010 59.57608 34.83733 44.84168 46.97125
## 2011 60.01903 38.37118 46.97586 50.73380
## 2012 61.64687 39.29957 52.67121 54.33232
## 2013 66.83436 40.87119 51.82854 57.49191
## 2014 65.25147 43.06121 54.76076 59.83447
## 2015 73.25703 47.69662 61.09777 66.05576
```

```
autoplot(AUtourists_ts)
```



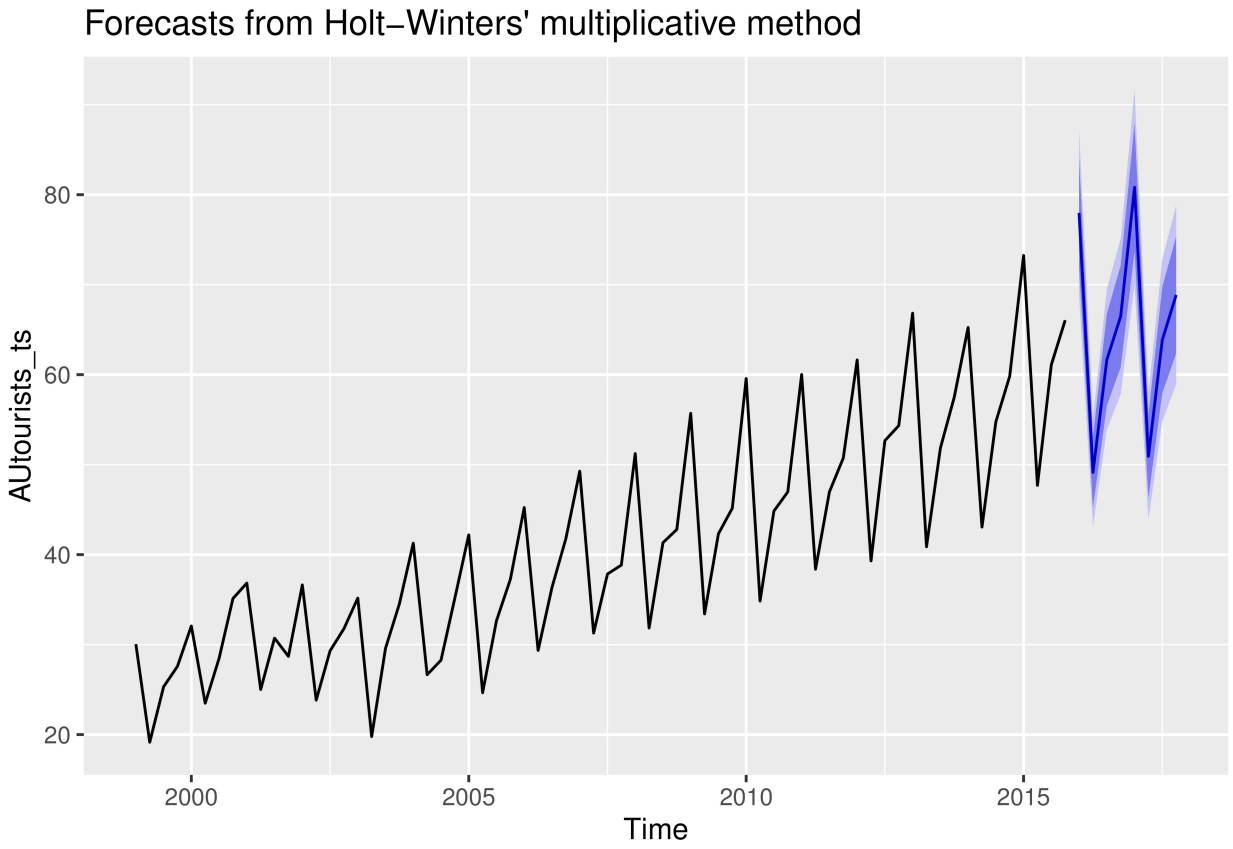
1b

```
fit1 <- hw(AUtourists_ts, seasonal = "multiplicative")
fit1
```

1f

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2016 Q1       77.97306 71.94180 84.00433 68.74903 87.19709
## 2016 Q2       49.13475 45.20183 53.06768 43.11986 55.14964
## 2016 Q3       61.64632 56.55395 66.73868 53.85822 69.43442
## 2016 Q4       66.49844 60.84265 72.15422 57.84866 75.14822
## 2017 Q1       80.82401 73.72354 87.92448 69.96478 91.68324
## 2017 Q2       50.91502 46.32846 55.50158 43.90049 57.92956
## 2017 Q3       63.85989 57.97028 69.74950 54.85250 72.86728
## 2017 Q4       68.86502 62.37164 75.35841 58.93425 78.79580
```

```
autoplot(fit1)
```



1g

```
d <- Read("http://web.pdx.edu/~gerbing/451/Data/HW9_2.xlsx")
```

2a

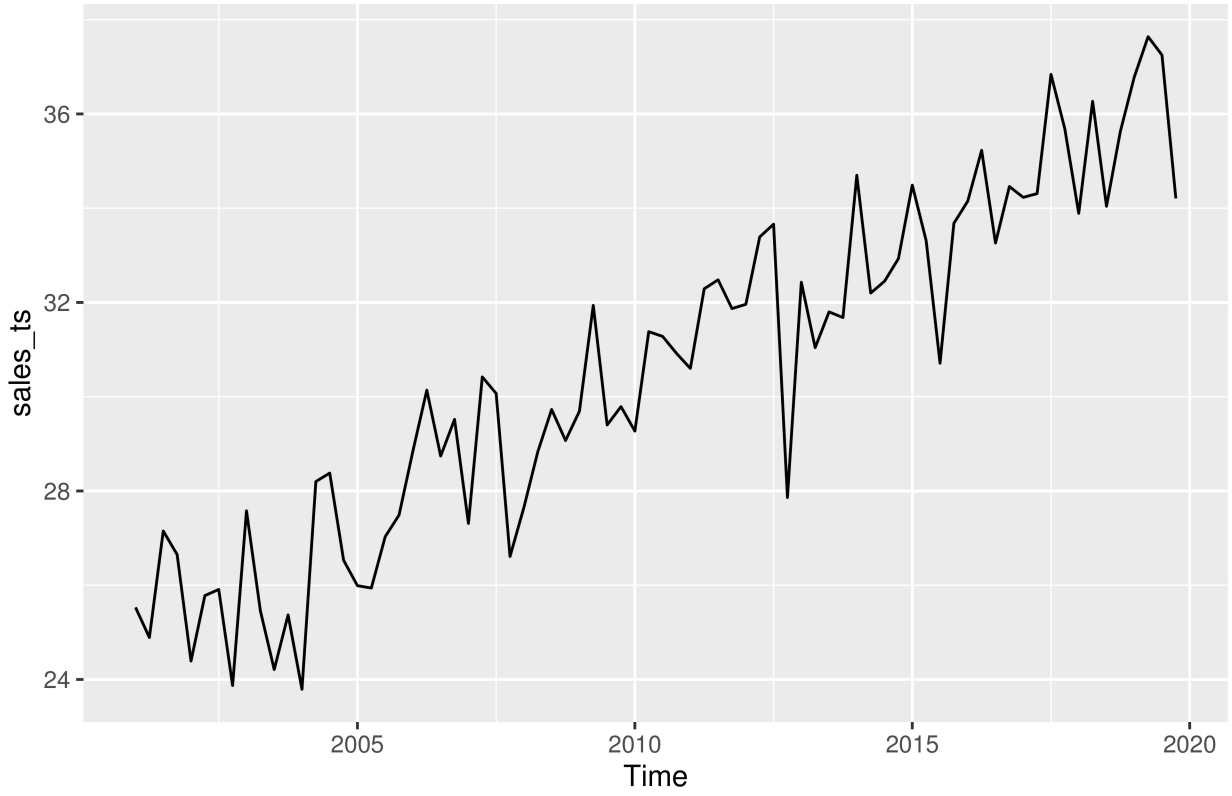
```
## [with the read.xlsx function from Alexander Walker's openxlsx package]
##
## >>> Suggestions
```

```
## To read a csv or Excel file of variable labels, var_labels=TRUE
##    Each row of the file:  Variable Name, Variable Label
## Details about your data, Enter:  details()  for d, or  details(name)
##
## Data Types
## ------------------------------------------------------------
## double: Numeric data values with decimal digits
## ------------------------------------------------------------
##
##     Variable                   Missing  Unique
##         Name     Type  Values  Values  Values   First and last values
## ------------------------------------------------------------------------------------------
##  1     Sales    double     76       0      76   25.53  24.89 ... 37.25  34.2075
## ------------------------------------------------------------------------------------------
```

```
sales_ts <- ts(d$Sales, frequency=4, start=c(2001,1))
sales_ts
```

```
##         Qtr1    Qtr2    Qtr3    Qtr4
## 2001 25.5300 24.8900 27.1500 26.6500
## 2002 24.3900 25.7800 25.9100 23.8700
## 2003 27.5800 25.4600 24.2100 25.3700
## 2004 23.7900 28.2000 28.3800 26.5300
## 2005 25.9900 25.9400 27.0300 27.4900
## 2006 28.8500 30.1400 28.7400 29.5200
## 2007 27.3100 30.4200 30.0700 26.6100
## 2008 27.6500 28.8300 29.7300 29.0700
## 2009 29.6900 31.9400 29.4000 29.7900
## 2010 29.2700 31.3800 31.2800 30.9200
## 2011 30.6000 32.2900 32.4800 31.8700
## 2012 31.9600 33.3900 33.6600 27.8600
## 2013 32.4300 31.0400 31.8000 31.6800
## 2014 34.7000 32.2000 32.4500 32.9300
## 2015 34.4900 33.3100 30.7100 33.6800
## 2016 34.1500 35.2300 33.2600 34.4600
## 2017 34.2300 34.3100 36.8400 35.6800
## 2018 33.8900 36.2700 34.0400 35.6300
## 2019 36.7800 37.6400 37.2500 34.2075
```

```
autoplot(sales_ts)
```



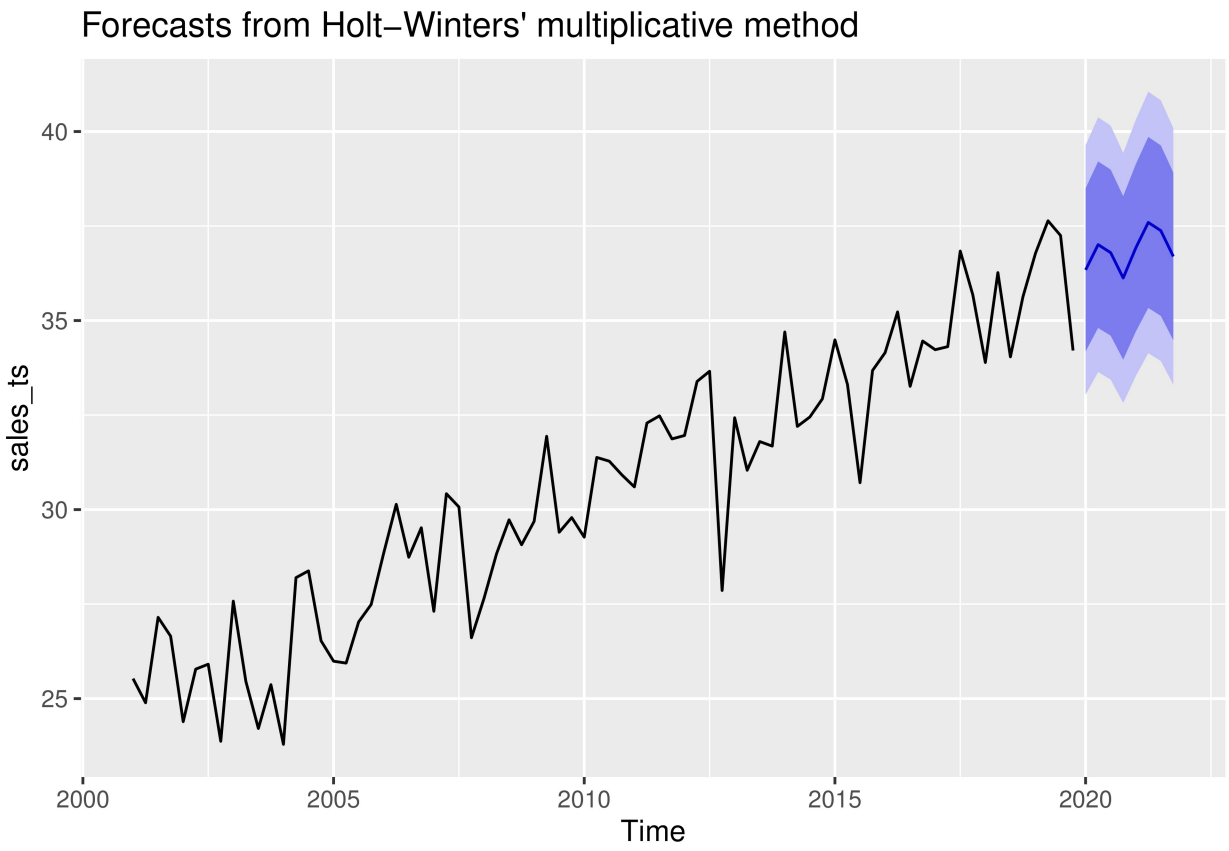
2b

```
fit1 <- hw(sales_ts, seasonal = "multiplicative") # holt-winters trend-seasonality exponential smoothin
fit1
```

2e

```
##         Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2020 Q1       36.33990 34.18280 38.49701 33.04090 39.63891
## 2020 Q2       37.00833 34.80557 39.21109 33.63950 40.37716
## 2020 Q3       36.79664 34.60058 38.99270 33.43806 40.15522
## 2020 Q4       36.12697 33.96513 38.28880 32.82072 39.43321
## 2021 Q1       36.91946 34.70436 39.13456 33.53175 40.30717
## 2021 Q2       37.59620 35.33461 39.85779 34.13739 41.05501
## 2021 Q3       37.37884 35.12451 39.63317 33.93114 40.82653
## 2021 Q4       36.69632 34.47749 38.91514 33.30292 40.08972
```

```
autoplot(fit1)
```


Forecasts from Holt−Winters' multiplicative method
sales_ts
40
35
30
25
2000
2005
2010
2015
2020
Time

2f

fin