Oman Road Traffic Accidents (RTA) Analysis in R

Using Time Series

1. Data Extraction

In [1]:
```
# @author: Aamir M. Khan
# Created First: April 2 2019
# Updated Last:
```

Import Libraries

In [2]:
```
library(reshape2)
library(ggplot2)
library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.2.1 --
v tibble  2.1.1     v purrr   0.3.2
v tidyr   0.8.3     v dplyr   0.8.1
v readr   1.3.1     v stringr 1.4.0
v tibble  2.1.1     v forcats 0.4.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

Load Data Files

In [3]:
```
data <- read.csv("../data/fatalities.csv")
head(data,3)
```

Year	January	February	March	April	May	June	July	August	September	Octobar	November	December
2000	56	38	49	38	31	43	61	24	23	52	32	45
2001	40	35	41	27	36	47	40	40	47	60	28	58
2002	46	56	30	86	38	59	42	55	35	45	45	43

In [4]:
```
dim(data)
```

19 13

In [5]:
```
class(data)
```

'data.frame'

Formatting the Data

In [6]:
```
fatalities = melt(data, id='Year', variable.name='Month', value.name='Value')
head(fatalities,5)
```

Year	Month	Value
2000	January	56
2001	January	40
2002	January	46
2003	January	70
2004	January	45

In [7]:
```
fatalities$Month <- as.integer(factor(fatalities$Month, levels = unique(fatalities$Month)))
fatalities = na.omit(fatalities)
fatalities$Time <- as.Date(sprintf("%d-%02d-%02d", fatalities$Year, fatalities$Month,1))
fatalities <- fatalities[c('Time','Value')]
fatalities <- fatalities[order(fatalities$Time),]
rownames(fatalities) <- NULL
head(fatalities,5)
```

Time	Value
2000-01-01	56
2000-02-01	38
2000-03-01	49
2000-04-01	38
2000-05-01	31

In [8]:
```
fatalities$Year <- format(fatalities$Time, "%Y")
```

In [9]:
```
str(fatalities)
```

```
'data.frame':   228 obs. of  3 variables:
 $ Time : Date, format: "2000-01-01" "2000-02-01" ...
 $ Value: int  56 38 49 38 31 43 61 24 23 52 ...
 $ Year : chr  "2000" "2000" "2000" "2000" ...
```

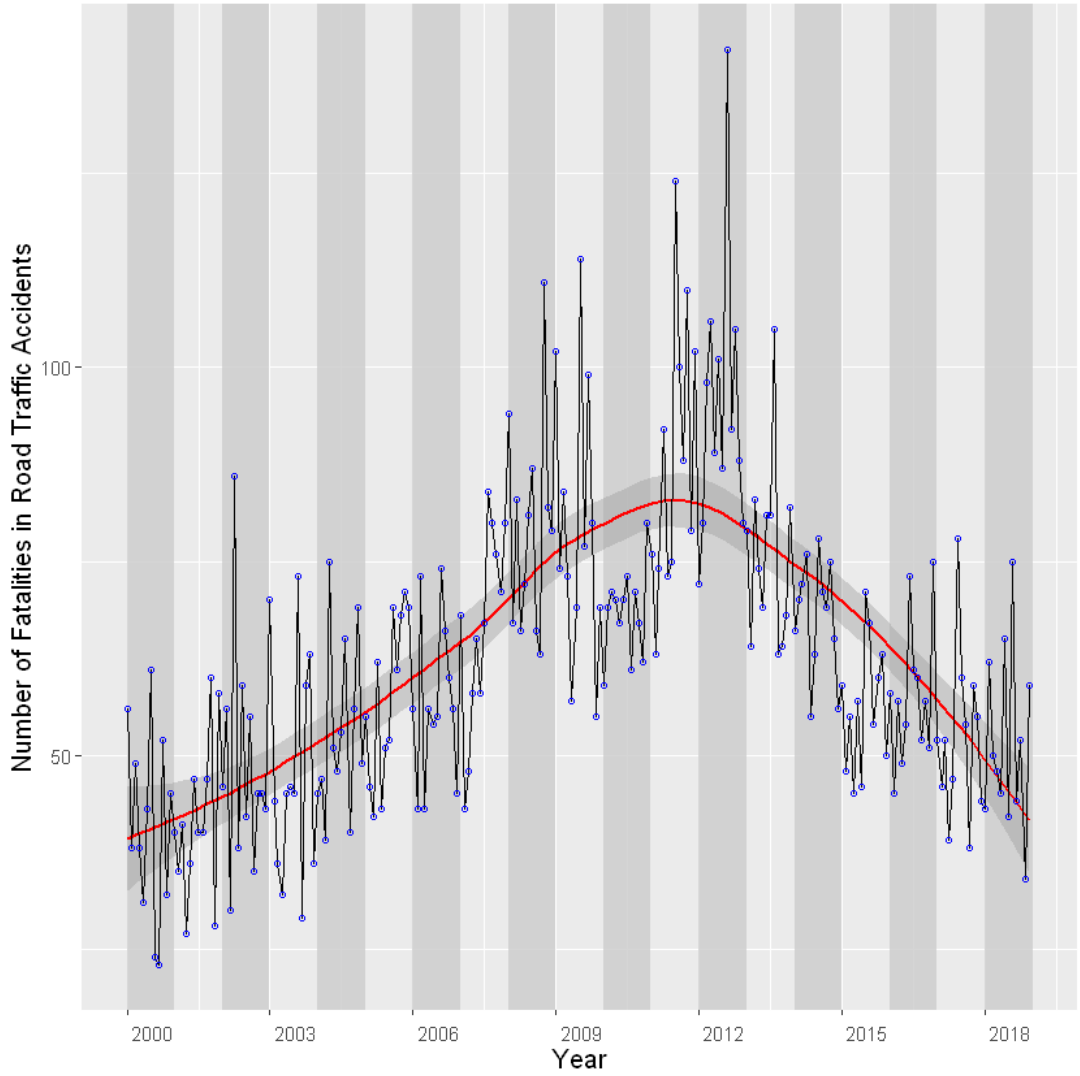
In [10]:
```
seq(2000, 2018, by=3)
```

```
2000 2003 2006 2009 2012 2015 2018
```

Time Series of Road Traffic Accidents

In [15]:
```
xmin <- seq(2000,2018,2)
xlbl1 <- seq(as.Date("2000-01-01"), as.Date("2018-12-01"), by="3 years")
xlbl2 <- seq(2000, 2018, by=3)
p <- ggplot() +
    geom_rect(data=data.frame(xmin), aes(xmin=as.Date(paste0(xmin,"-01-01")),
                                xmax=as.Date(paste0(xmin,"-12-31")),
                                ymin=-Inf, ymax=+Inf),
            fill='gray80', alpha=0.8) +
    geom_smooth(data=fatalities, aes(x=Time, y=Value), col='red', size=0.7) +
    geom_line(data=fatalities, aes(x=Time, y=Value), size=0.5) +
    geom_point(data=fatalities, aes(x=Time, y=Value), shape=1, col='blue', size=1) +
    #ggtitle("Road Traffic Deaths in Oman from 2000 to 2018") +
    labs(x='Year', y='Number of Fatalities in Road Traffic Accidents') +
    scale_x_date(breaks = as.Date(xlbl1),labels = xlbl2) +
    theme_grey() +
    theme(
        #plot.title = element_text(color="blue", size=14, face="bold.italic",hjust = 0.5),
        axis.title.x = element_text(size=12),
        axis.text.x = element_text(lineheight = 0, angle = 0, vjust=0, hjust=-0.1),
        axis.title.y = element_text(size=12)
    )
p
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



In [16]:
```
ggsave('Fatalities_Ts.png', p, device='png', dpi=1200, limitsize=FALSE)
```

```
Saving 6.67 x 6.67 in image
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

Decomposition of Time Series Data

In [17]:
```
# transpose and convert to vector form
datav <- as.vector(t(data[,-1]))
datav
```

```
56 38 49 38 31 43 61 24 23 52 32 45 40 35 41 27 36 47 40 40 47 60 28 58 46 56 30 86 38 59 42 55
35 45 45 43 70 44 36 32 45 46 45 73 29 59 63 36 45 47 39 75 51 48 53 65 40 56 69 49 55 46 42 62
43 51 52 69 61 68 71 69 56 43 73 43 56 54 55 74 66 60 56 45 68 43 48 58 65 58 67 84 80 76 71 80
94 67 83 66 72 81 87 66 63 111 82 79 102 74 84 73 57 69 114 77 99 80 55 69 59 69 71 70 67 70 73 61
71 67 62 80 76 63 74 92 73 75 124 100 88 110 79 102 72 80 98 106 89 101 87 141 92 105 88 80 79 64
83 74 69 81 81 105 63 64 68 82 66 70 72 76 55 63 78 71 69 75 65 56 59 48 55 45 57 46 71 67 54 60
63 50 58 45 57 49 54 73 61 60 52 57 51 75 52 46 52 39 47 78 60 54 38 59 55 44 43 62 50 48 45 65
42 75 44 52 34 59
```

In [18]:
```
# create time series from data
data_ts <- ts(datav,frequency=12, start=c(2000,1))
```

In [19]:
```
summary(data_ts)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  23.00   47.00   60.50   62.33   73.00  141.00 
```

In [20]:
```
print(data_ts)
```

```
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2000  56  38  49  38  31  43  61  24  23  52  32  45
2001  40  35  41  27  36  47  40  40  47  60  28  58
2002  46  56  30  86  38  59  42  55  35  45  45  43
2003  70  44  36  32  45  46  45  73  29  59  63  36
2004  45  47  39  75  51  48  53  65  40  56  69  49
2005  55  46  42  62  43  51  52  69  61  68  71  69
2006  56  43  73  43  56  54  55  74  66  60  56  45
2007  68  43  48  58  65  58  67  84  80  76  71  80
2008  94  67  83  66  72  81  87  66  63 111  82  79
2009 102  74  84  73  57  69 114  77  99  80  55  69
2010  59  69  71  70  67  70  73  61  71  67  62  80
2011  76  63  74  92  73  75 124 100  88 110  79 102
2012  72  80  98 106  89 101  87 141  92 105  88  80
2013  79  64  83  74  69  81  81 105  63  64  68  82
2014  66  70  72  76  55  63  78  71  69  75  65  56
2015  59  48  55  45  57  46  71  67  54  60  63  50
2016  58  45  57  49  54  73  61  60  52  57  51  75
2017  52  46  52  39  47  78  60  54  38  59  55  44
2018  43  62  50  48  45  65  42  75  44  52  34  59
```

In [21]:
```
# decompose seasonal time series data: original, seasonal, trend, random
fitF <- decompose(data_ts)
fitF
```

```
$x
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2000  56  38  49  38  31  43  61  24  23  52  32  45
2001  40  35  41  27  36  47  40  40  47  60  28  58
2002  46  56  30  86  38  59  42  55  35  45  45  43
2003  70  44  36  32  45  46  45  73  29  59  63  36
2004  45  47  39  75  51  48  53  65  40  56  69  49
2005  55  46  42  62  43  51  52  69  61  68  71  69
2006  56  43  73  43  56  54  55  74  66  60  56  45
2007  68  43  48  58  65  58  67  84  80  76  71  80
2008  94  67  83  66  72  81  87  66  63 111  82  79
2009 102  74  84  73  57  69 114  77  99  80  55  69
2010  59  69  71  70  67  70  73  61  71  67  62  80
2011  76  63  74  92  73  75 124 100  88 110  79 102
2012  72  80  98 106  89 101  87 141  92 105  88  80
2013  79  64  83  74  69  81  81 105  63  64  68  82
2014  66  70  72  76  55  63  78  71  69  75  65  56
2015  59  48  55  45  57  46  71  67  54  60  63  50
2016  58  45  57  49  54  73  61  60  52  57  51  75
2017  52  46  52  39  47  78  60  54  38  59  55  44
2018  43  62  50  48  45  65  42  75  44  52  34  59

$seasonal
            Jan        Feb        Mar        Apr        May        Jun
2000  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2001  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2002  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2003  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2004  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2005  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2006  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2007  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2008  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2009  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2010  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2011  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2012  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2013  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2014  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2015  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2016  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2017  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
2018  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
           Jul        Aug        Sep        Oct        Nov        Dec
2000  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2001  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2002  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2003  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2004  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2005  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2006  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2007  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2008  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2009  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2010  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2011  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2012  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2013  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2014  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2015  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2016  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2017  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997
2018  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997

$trend
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
2000       NA       NA       NA       NA       NA       NA 40.33333 39.54167
2001 37.70833 37.50000 39.16667 40.50000 40.66667 41.04167 41.83333 42.95833
2002 49.08333 49.79167 49.91667 48.79167 48.87500 48.95833 49.33333 49.83333
2003 44.95833 45.83333 46.33333 46.66667 48.00000 48.45833 47.12500 46.20833
2004 51.16667 51.16667 51.29167 51.62500 51.75000 52.54167 53.50000 53.87500
2005 52.54167 52.66667 53.70833 55.08333 55.66667 56.58333 57.45833 57.37500
2006 59.70833 60.04167 60.45833 60.33333 59.37500 57.75000 57.25000 57.75000
2007 58.50000 59.41667 60.41667 61.66667 62.95833 65.04167 67.58333 69.66667
2008 77.58333 77.66667 76.20833 76.95833 78.87500 79.29167 79.58333 80.20833
2009 80.04167 81.62500 83.58333 83.79167 81.37500 79.83333 77.62500 75.62500
2010 73.29167 70.91667 69.08333 67.37500 67.12500 67.87500 69.04167 69.50000
2011 74.37500 78.12500 80.45833 82.95833 85.45833 87.08333 87.83333 88.37500
2012 94.20833 94.37500 96.25000 96.20833 96.37500 95.83333 95.20833 94.83333
2013 86.66667 84.91667 82.20833 79.29167 76.75000 76.00000 75.54167 75.25000
2014 71.95833 70.41667 69.25000 69.95833 70.29167 69.08333 67.70833 66.50000
2015 60.04167 59.58333 58.79167 57.54167 56.83333 56.50000 56.20833 56.04167
2016 58.00000 57.29167 56.91667 56.70833 56.08333 56.62500 57.41667 57.20833
2017 55.79167 55.50000 54.66667 54.16667 54.41667 53.29167 51.62500 51.91667
2018 51.16667 51.29167 52.41667 52.37500 51.20833 50.95833       NA       NA
          Sep      Oct      Nov      Dec
2000 39.08333 38.29167 38.04167 38.41667
2001 43.37500 45.37500 47.91667 48.50000
2002 49.58333 47.58333 45.62500 45.37500
2003 46.45833 48.37500 50.41667 50.75000
2004 53.95833 53.54167 52.66667 52.45833
2005 58.54167 59.04167 58.79167 59.45833
2006 56.70833 56.29167 57.29167 57.83333
2007 72.12500 73.91667 74.54167 75.79167
```

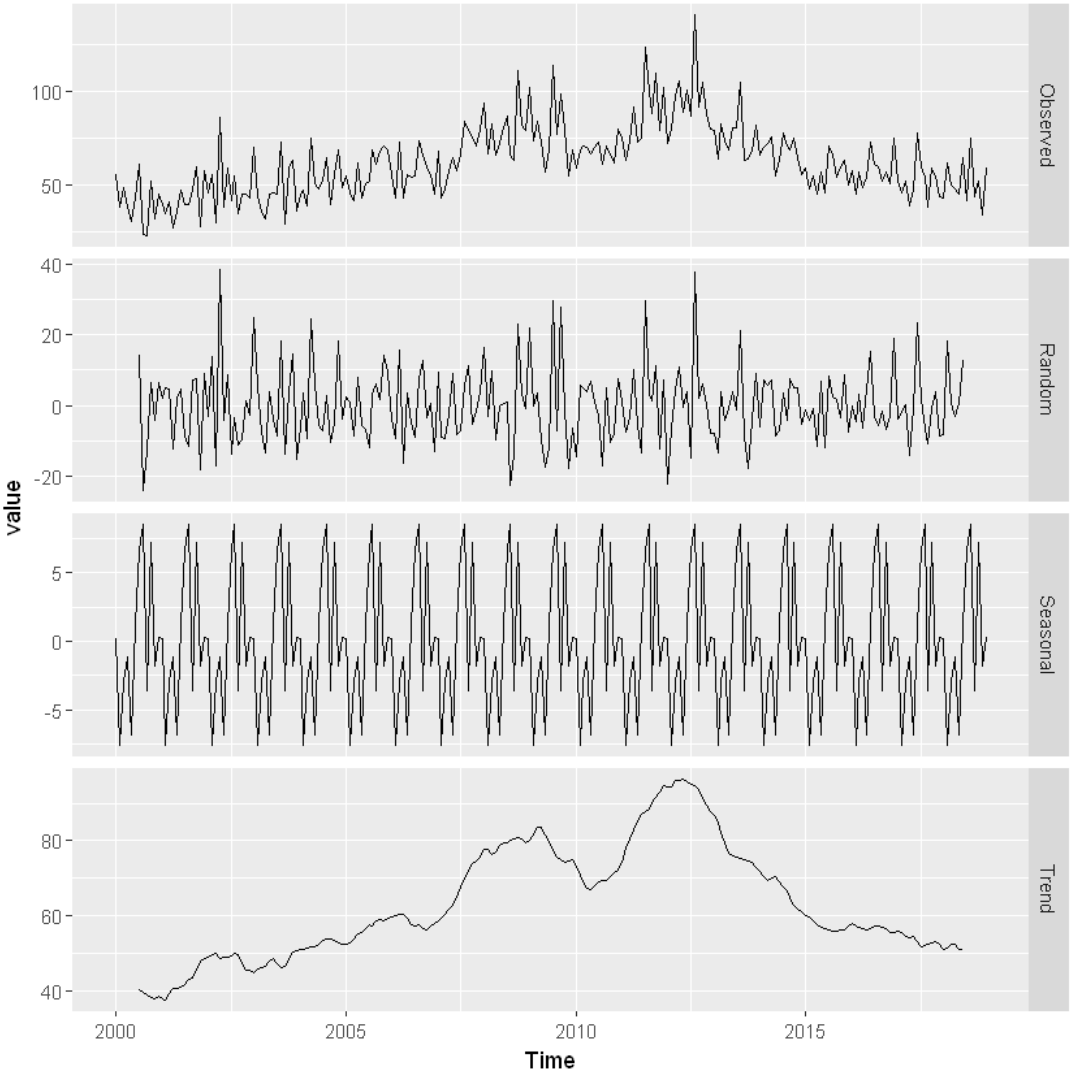
```
2008 80.54167 80.87500 80.54167 79.41667
2009 74.87500 74.20833 74.50000 74.95833
2010 69.37500 70.41667 71.58333 72.04167
2011 90.08333 91.66667 92.91667 94.66667
2012 93.54167 91.58333 89.41667 87.75000
2013 75.04167 74.66667 74.16667 72.83333
2014 64.87500 62.87500 61.66667 61.04167
2015 56.00000 56.25000 56.29167 57.29167
2016 57.04167 56.41667 55.70833 55.62500
2017 52.50000 52.79167 53.08333 52.45833
2018       NA       NA       NA       NA

$random
              Jan          Feb          Mar          Apr          May
2000           NA           NA           NA           NA           NA
2001   2.12982253   5.07889660   4.80111883 -12.31693673   2.18769290
2002  -3.24517747  13.78722994 -16.94888117  38.39139660  -4.02064043
2003  24.87982253   5.74556327  -7.36554784 -13.48360340   3.85435957
2004  -6.32851080   3.41222994  -9.32388117  24.55806327   6.10435957
2005   2.29648920   0.91222994  -8.74054784   8.09972994  -5.81230710
2006  -3.87017747  -9.46277006  15.50945216 -16.15027006   3.47935957
2007   9.33815586  -8.83777006  -9.44888117  -2.48360340   8.89602623
2008  16.25482253  -3.08777006   9.75945216  -9.77527006  -0.02064043
2009  21.79648920  -0.04610340   3.38445216  -9.60860340 -17.52064043
2010 -14.45351080   5.66222994   4.88445216   3.80806327   6.72935957
2011   1.46315586  -7.54610340  -3.49054784  10.22472994  -5.60397377
2012 -22.37017747  -6.79610340   4.71778549  10.97472994  -0.52064043
2013  -7.82851080 -13.33777006   3.75945216  -4.10860340  -0.89564043
2014  -6.12017747   7.16222994   5.71778549   7.22472994  -8.43730710
2015  -1.20351080  -4.00443673  -0.82388117 -11.35860340   7.02102623
2016  -0.16184414  -4.71277006   3.05111883  -6.52527006   4.77102623
2017  -3.95351080  -1.92110340   0.30111883 -13.98360340  -0.56230710
2018  -8.32851080  18.28722994   0.55111883  -3.19193673   0.64602623
              Jun          Jul          Aug          Sep          Oct
2000           NA  14.08352623 -24.04378858 -12.52758488   6.51176698
2001   4.73861883  -8.41647377 -11.46045525   7.18074846   7.42843364
2002   8.82195216 -13.91647377  -3.33545525 -11.02758488  -9.77989969
2003  -3.67804784  -8.70814043  18.28954475 -13.90258488   3.42843364
2004  -5.76138117  -7.08314043   2.62287809 -10.40258488  -4.73823302
2005  -6.80304784 -12.04147377   3.12287809   6.01408179   1.76176698
2006  -4.96971451  -8.83314043   7.74787809  12.84741512  -3.48823302
2007  -8.26138117  -7.16647377   5.83121142  11.43074846  -5.11323302
2008   0.48861883   0.83352623 -22.71045525 -13.98591821  22.92843364
2009 -12.05304784  29.79185957  -7.12712191  27.68074846  -1.40489969
2010   0.90528549  -2.62480710 -17.00212191   5.18074846 -10.61323302
2011 -13.30304784  29.58352623   3.12287809   1.47241512  11.13676698
2012   3.94695216 -14.79147377  37.66454475   2.01408179   6.22010031
2013   3.78028549  -1.12480710  21.24787809  -8.48591821 -17.86323302
2014  -7.30304784   3.70852623  -4.00212191   7.68074846   4.92843364
2015 -11.71971451   8.20852623   2.45621142   1.55574846  -3.44656636
2016  15.15528549  -2.99980710  -5.71045525  -1.48591821  -6.61323302
2017  23.48861883   1.79185957  -6.41878858 -10.94425154  -0.98823302
2018  12.82195216           NA           NA           NA           NA
              Nov          Dec
2000  -4.23823302   6.30343364
2001 -18.11323302   9.22010031
2002   1.17843364  -2.65489969
2003  14.38676698 -15.02989969
2004  18.13676698  -3.73823302
2005  14.01176698   9.26176698
2006   0.51176698 -13.11323302
2007  -1.73823302   3.92843364
2008   3.26176698  -0.69656636
2009 -17.69656636  -6.23823302
2010  -7.77989969   7.67843364
2011 -12.11323302   7.05343364
2012   0.38676698  -8.02989969
2013  -4.36323302   8.88676698
2014   5.13676698  -5.32156636
2015   8.51176698  -7.57156636
2016  -2.90489969  19.09510031
2017   3.72010031  -8.73823302
2018           NA           NA

$figure
 [1]  0.1618441 -7.5788966 -2.9677855 -1.1830633 -6.8543596  1.2197145
 [7]  6.5831404  8.5021219 -3.5557485  7.1965664 -1.8034336  0.2798997

$type
[1] "additive"

attr(,"class")
[1] "decomposed.ts"
```

In [22]:
```
Time = attributes(data_ts)[[1]]
Time = seq(Time[1],Time[2], length.out=(Time[2]-Time[1])*Time[3])
```

In [25]:
```
# convert to data frame
dat1 = cbind(Time, with(fitF, data.frame(Observed=x, Trend=trend, Seasonal=seasonal, Random=random)))
dat2 = gather(dat1, component, value, -Time)
```

In [26]:
```
# plot decomposed time series frame
q <- ggplot(dat2, aes(Time, value)) +
    facet_grid(component ~ ., scales="free_y") +
    geom_line(size=0.5) #+
#     #Labs(y=expression(Fatalities~of~Accidents), x="Time (Years)") +
#     labs(x='Year', y='Number of Fatalities in Road Traffic Accidents') +
# #     ggtitle(expression(Decomposed~RTA~time~series)) +
# #     theme_bw() +
# #     theme(plot.title=element_text(hjust=0.5))
#     theme_grey() +
#     theme(
#         #plot.title = element_text(color="blue", size=14, face="bold.italic",hjust = 0.5),
#         axis.title.x = element_text(size=12),
#         axis.text.x = element_text(lineheight = 0, angle = 0, vjust=0, hjust=-0.1),
#         axis.title.y = element_text(size=12)
#     )
q
```

```
Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
Warning message:
"Removed 6 rows containing missing values (geom_path)."
```



In [27]:
```
ggsave('Decomposed.png', q, device='png', dpi=1200, limitsize=FALSE)
```

```
Saving 6.67 x 6.67 in image
Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.
Warning message:
"Removed 6 rows containing missing values (geom_path)."
```

In [ ]:
```

```

In [469]:
```
xmin <- seq(2000,2018,2)
ggplot() +
    facet_wrap(gather(dat, component, value, -Time), aes(Time, value), component ~ ., scales="free_y") +
    geom_line(gather(dat, component, value, -Time), aes(Time, value), col='blue', size=0.5) +
    #labs(y=expression(Fatalities~of~Accidents), x="Time (Years)") +
    labs(x='Year', y='Number of Fatalities in Road Traffic Accidents') +
#     ggtitle(expression(Decomposed~RTA~time~series)) +
#     theme_bw() +
#     theme(plot.title=element_text(hjust=0.5))
    theme_grey() +
    theme(
        #plot.title = element_text(color="blue", size=14, face="bold.italic",hjust = 0.5),
        axis.title.x = element_text(size=12),
        axis.text.x = element_text(lineheight = 0, angle = 0, vjust=0, hjust=-0.1),
        axis.title.y = element_text(size=12)
    )
```

```
Warning message:
"Only the first value of `nrow` will be used."Warning message:
"Coercing `nrow` to be an integer."

Error in sanitise_dim(nrow): (list) object cannot be coerced to type 'integer'
Traceback:

1. facet_wrap(gather(dat, component, value, -Time), aes(Time, value),
 .     component ~ ., scales = "free_y")
2. sanitise_dim(nrow)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

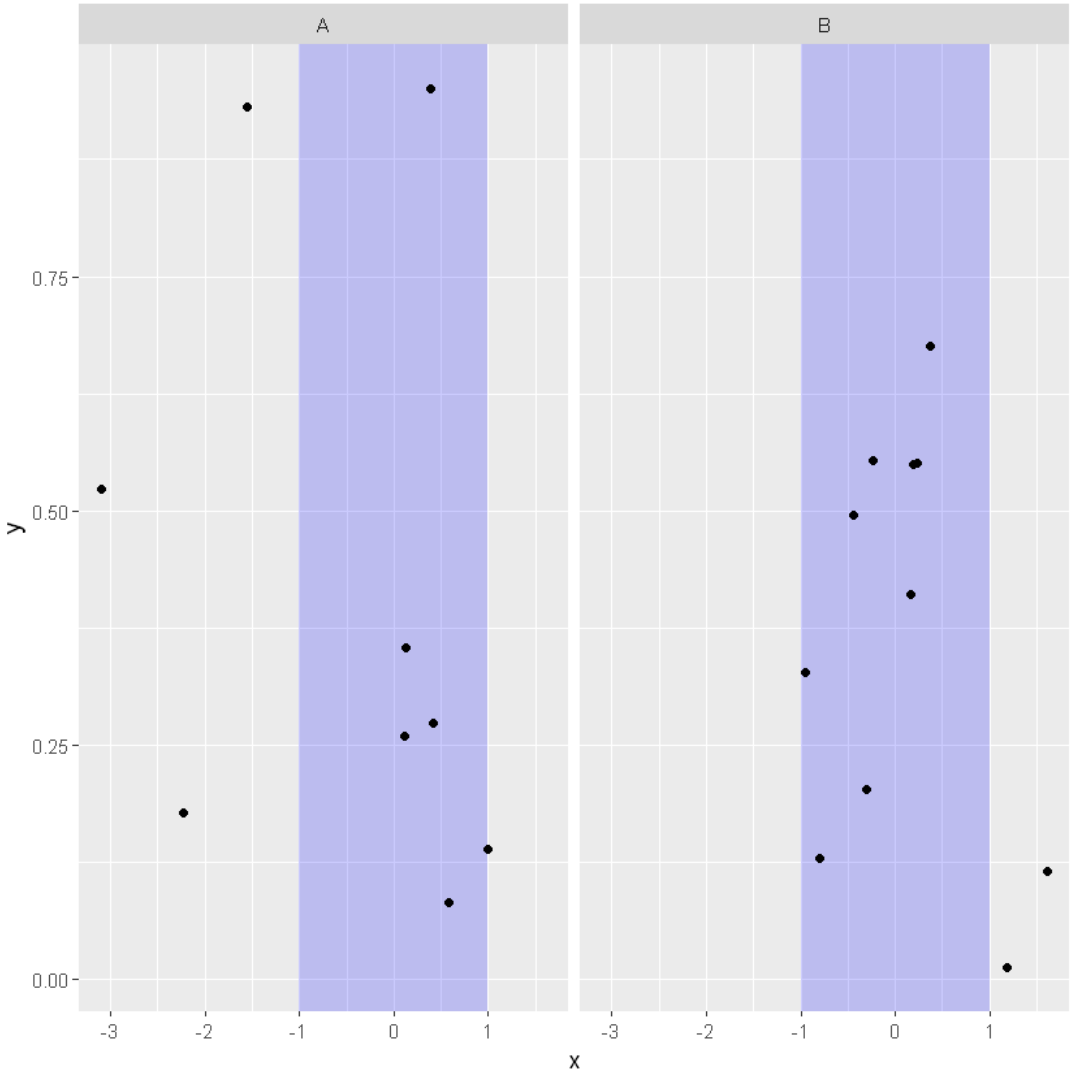
In [450]:
```
df <- data.frame(x = rnorm(20),
                 y = runif(20),
                 facet = sample(c("A", "B"),
                                20,
                                replace = TRUE))
df
```

x	y	facet
-0.357615902	0.08661050	A
-2.249300745	0.90498928	A
0.002837861	0.19894589	B
1.283033216	0.97319802	A
-0.267299063	0.06459416	A
-0.717604576	0.45351959	B
-0.604157249	0.27298650	B
-1.394921621	0.22182555	B
-1.598624461	0.02343691	B
1.605976503	0.75214399	B
0.625106883	0.87449888	A
-0.501474286	0.02234960	B
-1.555226410	0.44241519	A
-0.157451651	0.46291342	A
-0.289198126	0.97845921	A
0.139514194	0.39979908	A
0.812509093	0.32213330	A
0.422366586	0.24405862	A
0.109740932	0.67612420	A
0.652817236	0.45258115	B

In [451]:
```
rect1 <- data.frame(xmin = -1,
                    xmax = 1,
                    ymin = -Inf,
                    ymax = Inf,
                    facet = c("A", "B"))
rect1
```

xmin	xmax	ymin	ymax	facet
-1	1	-Inf	Inf	A
-1	1	-Inf	Inf	B

In [449]:
```
ggplot() +
  geom_rect(data = rect1 , aes(xmin = xmin,
                               xmax = xmax,
                               ymin = ymin,
                               ymax = ymax),
            alpha = 0.2, fill = "blue") +
  geom_point(data = df, aes(x = x, y = y))+
  facet_wrap(~facet)
```



In [ ]:

In [ ]:

In [ ]:

In [458]:
```
gather(dat, component, value, -Time)
```

Time	component	value
2000.000	Observed	56
2000.083	Observed	38
2000.167	Observed	49
2000.250	Observed	38
2000.333	Observed	31
2000.417	Observed	43
2000.500	Observed	61
2000.583	Observed	24
2000.667	Observed	23
2000.750	Observed	52
2000.833	Observed	32
2000.917	Observed	45
2001.000	Observed	40
2001.083	Observed	35
2001.167	Observed	41
2001.250	Observed	27
2001.333	Observed	36
2001.417	Observed	47
2001.500	Observed	40
2001.583	Observed	40
2001.667	Observed	47
2001.750	Observed	60
2001.833	Observed	28
2001.917	Observed	58
2002.000	Observed	46
2002.083	Observed	56
2002.167	Observed	30
2002.250	Observed	86
2002.333	Observed	38
2002.417	Observed	59
...	...	...
2016.500	Random	-2.9998071
2016.583	Random	-5.7104552
2016.667	Random	-1.4859182
2016.750	Random	-6.6132330
2016.833	Random	-2.9048997
2016.917	Random	19.0951003
2017.000	Random	-3.9535108
2017.083	Random	-1.9211034
2017.167	Random	0.3011188
2017.250	Random	-13.9836034
2017.333	Random	-0.5623071
2017.417	Random	23.4886188
2017.500	Random	1.7918596
2017.583	Random	-6.4187886
2017.667	Random	-10.9442515
2017.750	Random	-0.9882330
2017.833	Random	3.7201003
2017.917	Random	-8.7382330
2018.000	Random	-8.3285108
2018.083	Random	18.2872299
2018.167	Random	0.5511188
2018.250	Random	-3.1919367
2018.333	Random	0.6460262
2018.417	Random	12.8219522
2018.500	Random	NA
2018.583	Random	NA
2018.667	Random	NA
2018.750	Random	NA
2018.833	Random	NA

Time	component	value
2018.917	Random	NA

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: