=== Week1\_class.Rmd Requires: datasets, fma, forecast packages ===

### let’s play with time series data in R using TS and Forecast

First let’s get some data

We’ll start with the classic “airline passenger example” - the data are in the datasets package

library(datasets)   
  
data(AirPassengers)   
AP <- AirPassengers  
AP

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1949 112 118 132 129 121 135 148 148 136 119 104 118  
## 1950 115 126 141 135 125 149 170 170 158 133 114 140  
## 1951 145 150 178 163 172 178 199 199 184 162 146 166  
## 1952 171 180 193 181 183 218 230 242 209 191 172 194  
## 1953 196 196 236 235 229 243 264 272 237 211 180 201  
## 1954 204 188 235 227 234 264 302 293 259 229 203 229  
## 1955 242 233 267 269 270 315 364 347 312 274 237 278  
## 1956 284 277 317 313 318 374 413 405 355 306 271 306  
## 1957 315 301 356 348 355 422 465 467 404 347 305 336  
## 1958 340 318 362 348 363 435 491 505 404 359 310 337  
## 1959 360 342 406 396 420 472 548 559 463 407 362 405  
## 1960 417 391 419 461 472 535 622 606 508 461 390 432

#Attributes  
class(AP)

## [1] "ts"

length(AP)

## [1] 144

start(AP)

## [1] 1949 1

end(AP)

## [1] 1960 12

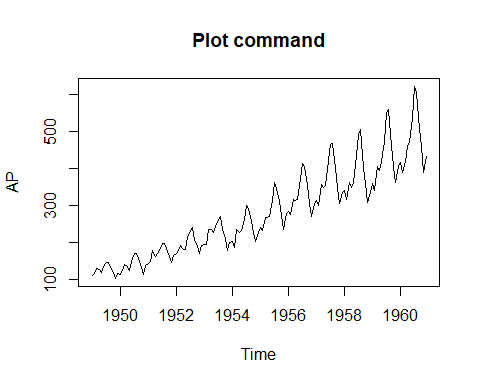
summary(AP)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 104.0 180.0 265.5 280.3 360.5 622.0

How about some basic exploratory data analysis (EDA)? First some pictures.

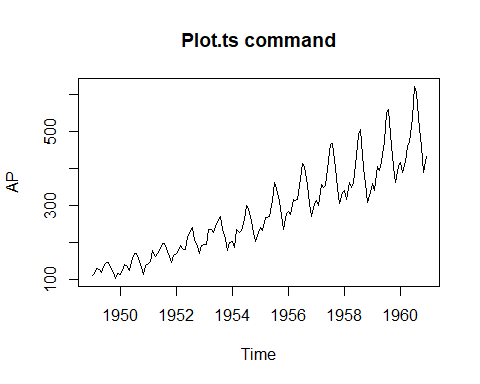
There are lots of ways to plot time series data. Here are some of the most basic commands.

# The base "plot" command  
plot(AP, main = "Plot command")



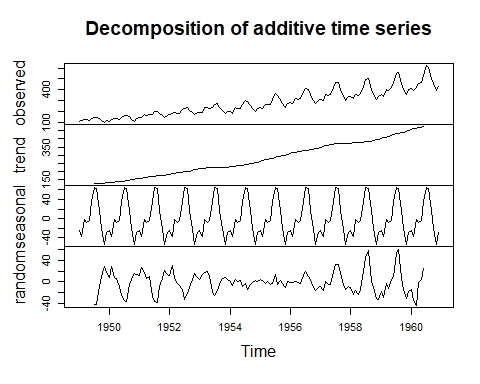
The Plot.ts command requires the same time frame

plot.ts(AP, main = "Plot.ts command")



The qualities you notice will affect the way we model the data. Trend, cycles, and seasonality are all on the agenda for next week.

plot(decompose(AP))



Now we can explore some of the simple forecasting methods we talked about earlier.

These commands are part of the “forecast” package so you’ll need to load that.

### Some simple forecasts - let’s look 4 years out

library(forecast)

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

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

## Registered S3 methods overwritten by 'forecast':  
## method from   
## fitted.fracdiff fracdiff  
## residuals.fracdiff fracdiff

Mean <- meanf(AP, h=48)   
class(Mean)

## [1] "forecast"

head(Mean)

## $method  
## [1] "Mean"  
##   
## $level  
## [1] 80 95  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1949 112 118 132 129 121 135 148 148 136 119 104 118  
## 1950 115 126 141 135 125 149 170 170 158 133 114 140  
## 1951 145 150 178 163 172 178 199 199 184 162 146 166  
## 1952 171 180 193 181 183 218 230 242 209 191 172 194  
## 1953 196 196 236 235 229 243 264 272 237 211 180 201  
## 1954 204 188 235 227 234 264 302 293 259 229 203 229  
## 1955 242 233 267 269 270 315 364 347 312 274 237 278  
## 1956 284 277 317 313 318 374 413 405 355 306 271 306  
## 1957 315 301 356 348 355 422 465 467 404 347 305 336  
## 1958 340 318 362 348 363 435 491 505 404 359 310 337  
## 1959 360 342 406 396 420 472 548 559 463 407 362 405  
## 1960 417 391 419 461 472 535 622 606 508 461 390 432  
##   
## $series  
## [1] "AP"  
##   
## $mean  
## Jan Feb Mar Apr May Jun Jul  
## 1961 280.2986 280.2986 280.2986 280.2986 280.2986 280.2986 280.2986  
## 1962 280.2986 280.2986 280.2986 280.2986 280.2986 280.2986 280.2986  
## 1963 280.2986 280.2986 280.2986 280.2986 280.2986 280.2986 280.2986  
## 1964 280.2986 280.2986 280.2986 280.2986 280.2986 280.2986 280.2986  
## Aug Sep Oct Nov Dec  
## 1961 280.2986 280.2986 280.2986 280.2986 280.2986  
## 1962 280.2986 280.2986 280.2986 280.2986 280.2986  
## 1963 280.2986 280.2986 280.2986 280.2986 280.2986  
## 1964 280.2986 280.2986 280.2986 280.2986 280.2986  
##   
## $lower  
## 80% 95%  
## Jan 1961 125.3066 42.34016  
## Feb 1961 125.3066 42.34016  
## Mar 1961 125.3066 42.34016  
## Apr 1961 125.3066 42.34016  
## May 1961 125.3066 42.34016  
## Jun 1961 125.3066 42.34016  
## Jul 1961 125.3066 42.34016  
## Aug 1961 125.3066 42.34016  
## Sep 1961 125.3066 42.34016  
## Oct 1961 125.3066 42.34016  
## Nov 1961 125.3066 42.34016  
## Dec 1961 125.3066 42.34016  
## Jan 1962 125.3066 42.34016  
## Feb 1962 125.3066 42.34016  
## Mar 1962 125.3066 42.34016  
## Apr 1962 125.3066 42.34016  
## May 1962 125.3066 42.34016  
## Jun 1962 125.3066 42.34016  
## Jul 1962 125.3066 42.34016  
## Aug 1962 125.3066 42.34016  
## Sep 1962 125.3066 42.34016  
## Oct 1962 125.3066 42.34016  
## Nov 1962 125.3066 42.34016  
## Dec 1962 125.3066 42.34016  
## Jan 1963 125.3066 42.34016  
## Feb 1963 125.3066 42.34016  
## Mar 1963 125.3066 42.34016  
## Apr 1963 125.3066 42.34016  
## May 1963 125.3066 42.34016  
## Jun 1963 125.3066 42.34016  
## Jul 1963 125.3066 42.34016  
## Aug 1963 125.3066 42.34016  
## Sep 1963 125.3066 42.34016  
## Oct 1963 125.3066 42.34016  
## Nov 1963 125.3066 42.34016  
## Dec 1963 125.3066 42.34016  
## Jan 1964 125.3066 42.34016  
## Feb 1964 125.3066 42.34016  
## Mar 1964 125.3066 42.34016  
## Apr 1964 125.3066 42.34016  
## May 1964 125.3066 42.34016  
## Jun 1964 125.3066 42.34016  
## Jul 1964 125.3066 42.34016  
## Aug 1964 125.3066 42.34016  
## Sep 1964 125.3066 42.34016  
## Oct 1964 125.3066 42.34016  
## Nov 1964 125.3066 42.34016  
## Dec 1964 125.3066 42.34016

Naive <- naive(AP, h=48)  
class(Naive)

## [1] "forecast"

head(Naive)

## $method  
## [1] "Naive method"  
##   
## $model  
## Call: naive(y = AP, h = 48)   
##   
## Residual sd: 33.7543   
##   
## $lambda  
## NULL  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1949 112 118 132 129 121 135 148 148 136 119 104 118  
## 1950 115 126 141 135 125 149 170 170 158 133 114 140  
## 1951 145 150 178 163 172 178 199 199 184 162 146 166  
## 1952 171 180 193 181 183 218 230 242 209 191 172 194  
## 1953 196 196 236 235 229 243 264 272 237 211 180 201  
## 1954 204 188 235 227 234 264 302 293 259 229 203 229  
## 1955 242 233 267 269 270 315 364 347 312 274 237 278  
## 1956 284 277 317 313 318 374 413 405 355 306 271 306  
## 1957 315 301 356 348 355 422 465 467 404 347 305 336  
## 1958 340 318 362 348 363 435 491 505 404 359 310 337  
## 1959 360 342 406 396 420 472 548 559 463 407 362 405  
## 1960 417 391 419 461 472 535 622 606 508 461 390 432  
##   
## $fitted  
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1949 NA 112 118 132 129 121 135 148 148 136 119 104  
## 1950 118 115 126 141 135 125 149 170 170 158 133 114  
## 1951 140 145 150 178 163 172 178 199 199 184 162 146  
## 1952 166 171 180 193 181 183 218 230 242 209 191 172  
## 1953 194 196 196 236 235 229 243 264 272 237 211 180  
## 1954 201 204 188 235 227 234 264 302 293 259 229 203  
## 1955 229 242 233 267 269 270 315 364 347 312 274 237  
## 1956 278 284 277 317 313 318 374 413 405 355 306 271  
## 1957 306 315 301 356 348 355 422 465 467 404 347 305  
## 1958 336 340 318 362 348 363 435 491 505 404 359 310  
## 1959 337 360 342 406 396 420 472 548 559 463 407 362  
## 1960 405 417 391 419 461 472 535 622 606 508 461 390  
##   
## $residuals  
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1949 NA 6 14 -3 -8 14 13 0 -12 -17 -15 14  
## 1950 -3 11 15 -6 -10 24 21 0 -12 -25 -19 26  
## 1951 5 5 28 -15 9 6 21 0 -15 -22 -16 20  
## 1952 5 9 13 -12 2 35 12 12 -33 -18 -19 22  
## 1953 2 0 40 -1 -6 14 21 8 -35 -26 -31 21  
## 1954 3 -16 47 -8 7 30 38 -9 -34 -30 -26 26  
## 1955 13 -9 34 2 1 45 49 -17 -35 -38 -37 41  
## 1956 6 -7 40 -4 5 56 39 -8 -50 -49 -35 35  
## 1957 9 -14 55 -8 7 67 43 2 -63 -57 -42 31  
## 1958 4 -22 44 -14 15 72 56 14 -101 -45 -49 27  
## 1959 23 -18 64 -10 24 52 76 11 -96 -56 -45 43  
## 1960 12 -26 28 42 11 63 87 -16 -98 -47 -71 42

Seasonal <- snaive(AP, h=48)  
class(Seasonal)

## [1] "forecast"

Seasonal

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## Jan 1961 417 370.4595 463.5405 345.8224 488.1776  
## Feb 1961 391 344.4595 437.5405 319.8224 462.1776  
## Mar 1961 419 372.4595 465.5405 347.8224 490.1776  
## Apr 1961 461 414.4595 507.5405 389.8224 532.1776  
## May 1961 472 425.4595 518.5405 400.8224 543.1776  
## Jun 1961 535 488.4595 581.5405 463.8224 606.1776  
## Jul 1961 622 575.4595 668.5405 550.8224 693.1776  
## Aug 1961 606 559.4595 652.5405 534.8224 677.1776  
## Sep 1961 508 461.4595 554.5405 436.8224 579.1776  
## Oct 1961 461 414.4595 507.5405 389.8224 532.1776  
## Nov 1961 390 343.4595 436.5405 318.8224 461.1776  
## Dec 1961 432 385.4595 478.5405 360.8224 503.1776  
## Jan 1962 417 351.1818 482.8182 316.3397 517.6603  
## Feb 1962 391 325.1818 456.8182 290.3397 491.6603  
## Mar 1962 419 353.1818 484.8182 318.3397 519.6603  
## Apr 1962 461 395.1818 526.8182 360.3397 561.6603  
## May 1962 472 406.1818 537.8182 371.3397 572.6603  
## Jun 1962 535 469.1818 600.8182 434.3397 635.6603  
## Jul 1962 622 556.1818 687.8182 521.3397 722.6603  
## Aug 1962 606 540.1818 671.8182 505.3397 706.6603  
## Sep 1962 508 442.1818 573.8182 407.3397 608.6603  
## Oct 1962 461 395.1818 526.8182 360.3397 561.6603  
## Nov 1962 390 324.1818 455.8182 289.3397 490.6603  
## Dec 1962 432 366.1818 497.8182 331.3397 532.6603  
## Jan 1963 417 336.3895 497.6105 293.7169 540.2831  
## Feb 1963 391 310.3895 471.6105 267.7169 514.2831  
## Mar 1963 419 338.3895 499.6105 295.7169 542.2831  
## Apr 1963 461 380.3895 541.6105 337.7169 584.2831  
## May 1963 472 391.3895 552.6105 348.7169 595.2831  
## Jun 1963 535 454.3895 615.6105 411.7169 658.2831  
## Jul 1963 622 541.3895 702.6105 498.7169 745.2831  
## Aug 1963 606 525.3895 686.6105 482.7169 729.2831  
## Sep 1963 508 427.3895 588.6105 384.7169 631.2831  
## Oct 1963 461 380.3895 541.6105 337.7169 584.2831  
## Nov 1963 390 309.3895 470.6105 266.7169 513.2831  
## Dec 1963 432 351.3895 512.6105 308.7169 555.2831  
## Jan 1964 417 323.9190 510.0810 274.6449 559.3551  
## Feb 1964 391 297.9190 484.0810 248.6449 533.3551  
## Mar 1964 419 325.9190 512.0810 276.6449 561.3551  
## Apr 1964 461 367.9190 554.0810 318.6449 603.3551  
## May 1964 472 378.9190 565.0810 329.6449 614.3551  
## Jun 1964 535 441.9190 628.0810 392.6449 677.3551  
## Jul 1964 622 528.9190 715.0810 479.6449 764.3551  
## Aug 1964 606 512.9190 699.0810 463.6449 748.3551  
## Sep 1964 508 414.9190 601.0810 365.6449 650.3551  
## Oct 1964 461 367.9190 554.0810 318.6449 603.3551  
## Nov 1964 390 296.9190 483.0810 247.6449 532.3551  
## Dec 1964 432 338.9190 525.0810 289.6449 574.3551

Drift <- rwf(AP, drift=TRUE, h=48)  
class(Drift)

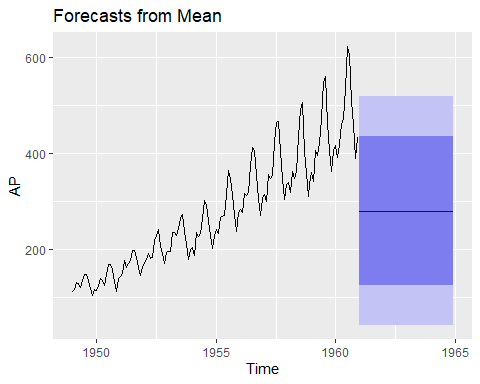
## [1] "forecast"

head(Drift)

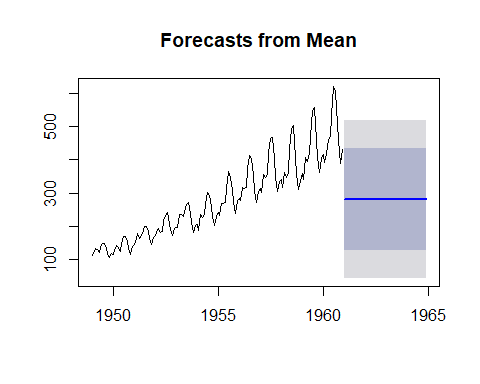
## $method  
## [1] "Random walk with drift"  
##   
## $model  
## Call: rwf(y = AP, h = 48, drift = TRUE)   
##   
## Drift: 2.2378 (se 2.8227)  
## Residual sd: 33.7543   
##   
## $lambda  
## NULL  
##   
## $x  
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1949 112 118 132 129 121 135 148 148 136 119 104 118  
## 1950 115 126 141 135 125 149 170 170 158 133 114 140  
## 1951 145 150 178 163 172 178 199 199 184 162 146 166  
## 1952 171 180 193 181 183 218 230 242 209 191 172 194  
## 1953 196 196 236 235 229 243 264 272 237 211 180 201  
## 1954 204 188 235 227 234 264 302 293 259 229 203 229  
## 1955 242 233 267 269 270 315 364 347 312 274 237 278  
## 1956 284 277 317 313 318 374 413 405 355 306 271 306  
## 1957 315 301 356 348 355 422 465 467 404 347 305 336  
## 1958 340 318 362 348 363 435 491 505 404 359 310 337  
## 1959 360 342 406 396 420 472 548 559 463 407 362 405  
## 1960 417 391 419 461 472 535 622 606 508 461 390 432  
##   
## $fitted  
## Jan Feb Mar Apr May Jun Jul  
## 1949 NA 114.2378 120.2378 134.2378 131.2378 123.2378 137.2378  
## 1950 120.2378 117.2378 128.2378 143.2378 137.2378 127.2378 151.2378  
## 1951 142.2378 147.2378 152.2378 180.2378 165.2378 174.2378 180.2378  
## 1952 168.2378 173.2378 182.2378 195.2378 183.2378 185.2378 220.2378  
## 1953 196.2378 198.2378 198.2378 238.2378 237.2378 231.2378 245.2378  
## 1954 203.2378 206.2378 190.2378 237.2378 229.2378 236.2378 266.2378  
## 1955 231.2378 244.2378 235.2378 269.2378 271.2378 272.2378 317.2378  
## 1956 280.2378 286.2378 279.2378 319.2378 315.2378 320.2378 376.2378  
## 1957 308.2378 317.2378 303.2378 358.2378 350.2378 357.2378 424.2378  
## 1958 338.2378 342.2378 320.2378 364.2378 350.2378 365.2378 437.2378  
## 1959 339.2378 362.2378 344.2378 408.2378 398.2378 422.2378 474.2378  
## 1960 407.2378 419.2378 393.2378 421.2378 463.2378 474.2378 537.2378  
## Aug Sep Oct Nov Dec  
## 1949 150.2378 150.2378 138.2378 121.2378 106.2378  
## 1950 172.2378 172.2378 160.2378 135.2378 116.2378  
## 1951 201.2378 201.2378 186.2378 164.2378 148.2378  
## 1952 232.2378 244.2378 211.2378 193.2378 174.2378  
## 1953 266.2378 274.2378 239.2378 213.2378 182.2378  
## 1954 304.2378 295.2378 261.2378 231.2378 205.2378  
## 1955 366.2378 349.2378 314.2378 276.2378 239.2378  
## 1956 415.2378 407.2378 357.2378 308.2378 273.2378  
## 1957 467.2378 469.2378 406.2378 349.2378 307.2378  
## 1958 493.2378 507.2378 406.2378 361.2378 312.2378  
## 1959 550.2378 561.2378 465.2378 409.2378 364.2378  
## 1960 624.2378 608.2378 510.2378 463.2378 392.2378  
##   
## $residuals  
## Jan Feb Mar Apr May  
## 1949 NA 3.7622378 11.7622378 -5.2377622 -10.2377622  
## 1950 -5.2377622 8.7622378 12.7622378 -8.2377622 -12.2377622  
## 1951 2.7622378 2.7622378 25.7622378 -17.2377622 6.7622378  
## 1952 2.7622378 6.7622378 10.7622378 -14.2377622 -0.2377622  
## 1953 -0.2377622 -2.2377622 37.7622378 -3.2377622 -8.2377622  
## 1954 0.7622378 -18.2377622 44.7622378 -10.2377622 4.7622378  
## 1955 10.7622378 -11.2377622 31.7622378 -0.2377622 -1.2377622  
## 1956 3.7622378 -9.2377622 37.7622378 -6.2377622 2.7622378  
## 1957 6.7622378 -16.2377622 52.7622378 -10.2377622 4.7622378  
## 1958 1.7622378 -24.2377622 41.7622378 -16.2377622 12.7622378  
## 1959 20.7622378 -20.2377622 61.7622378 -12.2377622 21.7622378  
## 1960 9.7622378 -28.2377622 25.7622378 39.7622378 8.7622378  
## Jun Jul Aug Sep Oct  
## 1949 11.7622378 10.7622378 -2.2377622 -14.2377622 -19.2377622  
## 1950 21.7622378 18.7622378 -2.2377622 -14.2377622 -27.2377622  
## 1951 3.7622378 18.7622378 -2.2377622 -17.2377622 -24.2377622  
## 1952 32.7622378 9.7622378 9.7622378 -35.2377622 -20.2377622  
## 1953 11.7622378 18.7622378 5.7622378 -37.2377622 -28.2377622  
## 1954 27.7622378 35.7622378 -11.2377622 -36.2377622 -32.2377622  
## 1955 42.7622378 46.7622378 -19.2377622 -37.2377622 -40.2377622  
## 1956 53.7622378 36.7622378 -10.2377622 -52.2377622 -51.2377622  
## 1957 64.7622378 40.7622378 -0.2377622 -65.2377622 -59.2377622  
## 1958 69.7622378 53.7622378 11.7622378 -103.2377622 -47.2377622  
## 1959 49.7622378 73.7622378 8.7622378 -98.2377622 -58.2377622  
## 1960 60.7622378 84.7622378 -18.2377622 -100.2377622 -49.2377622  
## Nov Dec  
## 1949 -17.2377622 11.7622378  
## 1950 -21.2377622 23.7622378  
## 1951 -18.2377622 17.7622378  
## 1952 -21.2377622 19.7622378  
## 1953 -33.2377622 18.7622378  
## 1954 -28.2377622 23.7622378  
## 1955 -39.2377622 38.7622378  
## 1956 -37.2377622 32.7622378  
## 1957 -44.2377622 28.7622378  
## 1958 -51.2377622 24.7622378  
## 1959 -47.2377622 40.7622378  
## 1960 -73.2377622 39.7622378

These simple forecasts can be charted with the “plot command”

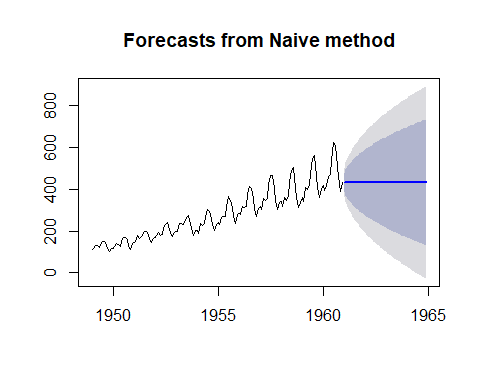
library(ggplot2)  
autoplot(Mean)



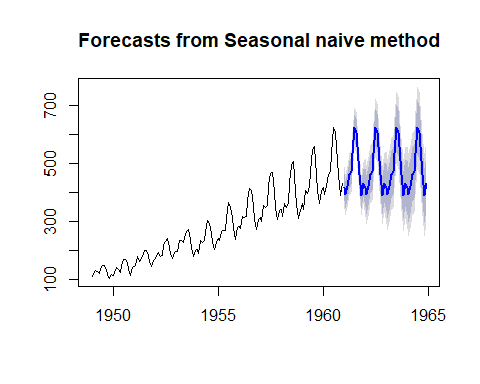
plot(Mean)



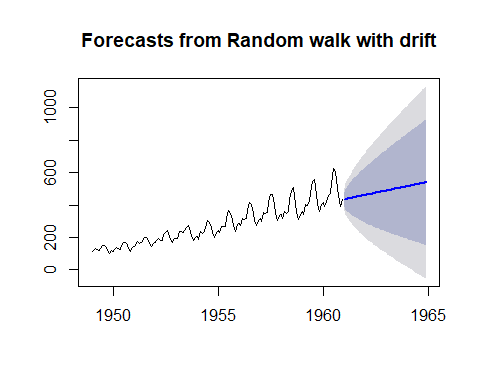
plot(Naive)



plot(Seasonal)

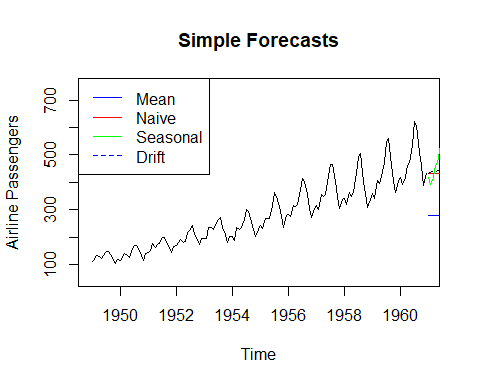


plot(Drift)



Let’s look at them on the same chart.

plot(AP, plot.type="single", main="Simple Forecasts", ylab="Airline Passengers",ylim = c(50,750))  
lines(Mean$mean, col = "blue")  
lines(Naive$mean, col = "red")  
lines(Seasonal$mean, col = "green")  
lines(Drift$mean, lty = 2)  
legend("topleft", legend=c("Mean","Naive","Seasonal","Drift"), col=c("blue", "red", "green"), lty = c(1,1,1,2))



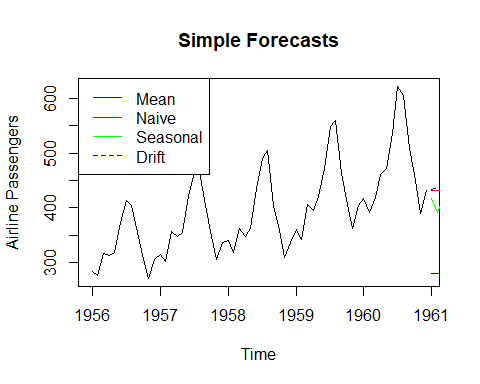
What if we change the time frame for the chart? Let’s just look at 1956 on? Use the “window” command.

AP.short = window(AP, start = c(1956,1))  
AP.short

## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec  
## 1956 284 277 317 313 318 374 413 405 355 306 271 306  
## 1957 315 301 356 348 355 422 465 467 404 347 305 336  
## 1958 340 318 362 348 363 435 491 505 404 359 310 337  
## 1959 360 342 406 396 420 472 548 559 463 407 362 405  
## 1960 417 391 419 461 472 535 622 606 508 461 390 432

Now lets see that chart again:

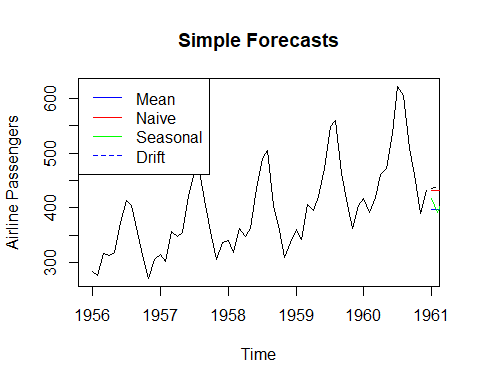
plot(AP.short, plot.type="single",main="Simple Forecasts", ylab="Airline Passengers")  
lines(Mean$mean, col = "blue")  
lines(Naive$mean, col = "red")  
lines(Seasonal$mean, col = "green")  
lines(Drift$mean, lty = 2)  
legend("topleft", legend=c("Mean","Naive","Seasonal","Drift"), col=c("blue", "red", "green"), lty = c(1,1,1,2))

 In this case, the window just changed the appearance of the chart. But you need to pay attention to the date range you are using for your calculations because they can change your results.

Mean.short <- meanf(AP.short, h=48)  
Naive.short <- naive(AP.short, h=48)  
Seasonal.short <- snaive(AP.short, h=48)  
Drift.short <- rwf(AP.short, drift=TRUE, h=48)

Where would you expect to see the differences?

plot(AP.short, plot.type="single",main="Simple Forecasts", ylab="Airline Passengers")  
lines(Mean.short$mean, col = "blue")  
lines(Naive.short$mean, col = "red")  
lines(Seasonal.short$mean, col = "green")  
lines(Drift.short$mean, lty = 2)  
legend("topleft", legend=c("Mean","Naive","Seasonal","Drift"), col=c("blue", "red", "green"), lty = c(1,1,1,2))



Let’s compare the numbers

all\_mean <- cbind(Mean$mean, Mean.short$mean)  
all\_naive <- cbind(Naive$mean, Naive.short$mean)  
all\_seasonal <- cbind(Seasonal$mean, Seasonal.short$mean)  
all\_drift <- cbind(Drift$mean, Drift.short$mean)  
  
  
all\_mean

## Mean$mean Mean.short$mean  
## Jan 1961 280.2986 396.4333  
## Feb 1961 280.2986 396.4333  
## Mar 1961 280.2986 396.4333  
## Apr 1961 280.2986 396.4333  
## May 1961 280.2986 396.4333  
## Jun 1961 280.2986 396.4333  
## Jul 1961 280.2986 396.4333  
## Aug 1961 280.2986 396.4333  
## Sep 1961 280.2986 396.4333  
## Oct 1961 280.2986 396.4333  
## Nov 1961 280.2986 396.4333  
## Dec 1961 280.2986 396.4333  
## Jan 1962 280.2986 396.4333  
## Feb 1962 280.2986 396.4333  
## Mar 1962 280.2986 396.4333  
## Apr 1962 280.2986 396.4333  
## May 1962 280.2986 396.4333  
## Jun 1962 280.2986 396.4333  
## Jul 1962 280.2986 396.4333  
## Aug 1962 280.2986 396.4333  
## Sep 1962 280.2986 396.4333  
## Oct 1962 280.2986 396.4333  
## Nov 1962 280.2986 396.4333  
## Dec 1962 280.2986 396.4333  
## Jan 1963 280.2986 396.4333  
## Feb 1963 280.2986 396.4333  
## Mar 1963 280.2986 396.4333  
## Apr 1963 280.2986 396.4333  
## May 1963 280.2986 396.4333  
## Jun 1963 280.2986 396.4333  
## Jul 1963 280.2986 396.4333  
## Aug 1963 280.2986 396.4333  
## Sep 1963 280.2986 396.4333  
## Oct 1963 280.2986 396.4333  
## Nov 1963 280.2986 396.4333  
## Dec 1963 280.2986 396.4333  
## Jan 1964 280.2986 396.4333  
## Feb 1964 280.2986 396.4333  
## Mar 1964 280.2986 396.4333  
## Apr 1964 280.2986 396.4333  
## May 1964 280.2986 396.4333  
## Jun 1964 280.2986 396.4333  
## Jul 1964 280.2986 396.4333  
## Aug 1964 280.2986 396.4333  
## Sep 1964 280.2986 396.4333  
## Oct 1964 280.2986 396.4333  
## Nov 1964 280.2986 396.4333  
## Dec 1964 280.2986 396.4333

all\_naive

## Naive$mean Naive.short$mean  
## Jan 1961 432 432  
## Feb 1961 432 432  
## Mar 1961 432 432  
## Apr 1961 432 432  
## May 1961 432 432  
## Jun 1961 432 432  
## Jul 1961 432 432  
## Aug 1961 432 432  
## Sep 1961 432 432  
## Oct 1961 432 432  
## Nov 1961 432 432  
## Dec 1961 432 432  
## Jan 1962 432 432  
## Feb 1962 432 432  
## Mar 1962 432 432  
## Apr 1962 432 432  
## May 1962 432 432  
## Jun 1962 432 432  
## Jul 1962 432 432  
## Aug 1962 432 432  
## Sep 1962 432 432  
## Oct 1962 432 432  
## Nov 1962 432 432  
## Dec 1962 432 432  
## Jan 1963 432 432  
## Feb 1963 432 432  
## Mar 1963 432 432  
## Apr 1963 432 432  
## May 1963 432 432  
## Jun 1963 432 432  
## Jul 1963 432 432  
## Aug 1963 432 432  
## Sep 1963 432 432  
## Oct 1963 432 432  
## Nov 1963 432 432  
## Dec 1963 432 432  
## Jan 1964 432 432  
## Feb 1964 432 432  
## Mar 1964 432 432  
## Apr 1964 432 432  
## May 1964 432 432  
## Jun 1964 432 432  
## Jul 1964 432 432  
## Aug 1964 432 432  
## Sep 1964 432 432  
## Oct 1964 432 432  
## Nov 1964 432 432  
## Dec 1964 432 432

all\_seasonal

## Seasonal$mean Seasonal.short$mean  
## Jan 1961 417 417  
## Feb 1961 391 391  
## Mar 1961 419 419  
## Apr 1961 461 461  
## May 1961 472 472  
## Jun 1961 535 535  
## Jul 1961 622 622  
## Aug 1961 606 606  
## Sep 1961 508 508  
## Oct 1961 461 461  
## Nov 1961 390 390  
## Dec 1961 432 432  
## Jan 1962 417 417  
## Feb 1962 391 391  
## Mar 1962 419 419  
## Apr 1962 461 461  
## May 1962 472 472  
## Jun 1962 535 535  
## Jul 1962 622 622  
## Aug 1962 606 606  
## Sep 1962 508 508  
## Oct 1962 461 461  
## Nov 1962 390 390  
## Dec 1962 432 432  
## Jan 1963 417 417  
## Feb 1963 391 391  
## Mar 1963 419 419  
## Apr 1963 461 461  
## May 1963 472 472  
## Jun 1963 535 535  
## Jul 1963 622 622  
## Aug 1963 606 606  
## Sep 1963 508 508  
## Oct 1963 461 461  
## Nov 1963 390 390  
## Dec 1963 432 432  
## Jan 1964 417 417  
## Feb 1964 391 391  
## Mar 1964 419 419  
## Apr 1964 461 461  
## May 1964 472 472  
## Jun 1964 535 535  
## Jul 1964 622 622  
## Aug 1964 606 606  
## Sep 1964 508 508  
## Oct 1964 461 461  
## Nov 1964 390 390  
## Dec 1964 432 432

all\_drift

## Drift$mean Drift.short$mean  
## Jan 1961 434.2378 434.5085  
## Feb 1961 436.4755 437.0169  
## Mar 1961 438.7133 439.5254  
## Apr 1961 440.9510 442.0339  
## May 1961 443.1888 444.5424  
## Jun 1961 445.4266 447.0508  
## Jul 1961 447.6643 449.5593  
## Aug 1961 449.9021 452.0678  
## Sep 1961 452.1399 454.5763  
## Oct 1961 454.3776 457.0847  
## Nov 1961 456.6154 459.5932  
## Dec 1961 458.8531 462.1017  
## Jan 1962 461.0909 464.6102  
## Feb 1962 463.3287 467.1186  
## Mar 1962 465.5664 469.6271  
## Apr 1962 467.8042 472.1356  
## May 1962 470.0420 474.6441  
## Jun 1962 472.2797 477.1525  
## Jul 1962 474.5175 479.6610  
## Aug 1962 476.7552 482.1695  
## Sep 1962 478.9930 484.6780  
## Oct 1962 481.2308 487.1864  
## Nov 1962 483.4685 489.6949  
## Dec 1962 485.7063 492.2034  
## Jan 1963 487.9441 494.7119  
## Feb 1963 490.1818 497.2203  
## Mar 1963 492.4196 499.7288  
## Apr 1963 494.6573 502.2373  
## May 1963 496.8951 504.7458  
## Jun 1963 499.1329 507.2542  
## Jul 1963 501.3706 509.7627  
## Aug 1963 503.6084 512.2712  
## Sep 1963 505.8462 514.7797  
## Oct 1963 508.0839 517.2881  
## Nov 1963 510.3217 519.7966  
## Dec 1963 512.5594 522.3051  
## Jan 1964 514.7972 524.8136  
## Feb 1964 517.0350 527.3220  
## Mar 1964 519.2727 529.8305  
## Apr 1964 521.5105 532.3390  
## May 1964 523.7483 534.8475  
## Jun 1964 525.9860 537.3559  
## Jul 1964 528.2238 539.8644  
## Aug 1964 530.4615 542.3729  
## Sep 1964 532.6993 544.8814  
## Oct 1964 534.9371 547.3898  
## Nov 1964 537.1748 549.8983  
## Dec 1964 539.4126 552.4068

So PAY ATTENTION to your date range

Next up: forecast accuracy

How good are our simple forecasts? How do we know? We need to compare the forecast with the actual. The charts we did before showed forecasts outside of our sample range - in this dataset, we don’t have actual numbers to compare with.

Remember what you learned in ADM and PM: training and test datasets. This is why the windowing is important.

Let’s look at models using the entire date range and set up the appropriate data structures.

#what if we use the 80/20 split from ADM?  
length(AP)

## [1] 144

trainObs = round(length(AP) \* .8)  
trainObs

## [1] 115

testObs = length(AP) - trainObs  
testObs

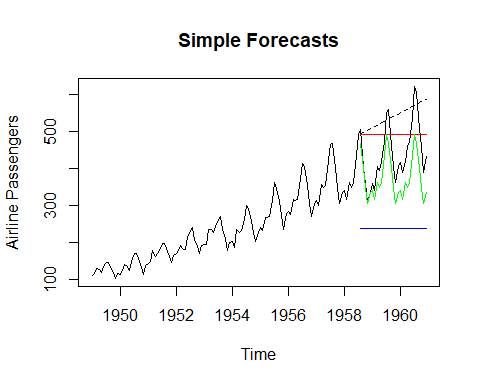
## [1] 29

train.AP <- window(AP, start = c(1949,1), end = c(1949,trainObs))  
  
test.AP <- window(AP, start = c(1949,trainObs+1))

Do we think this makes sense? Remember, our test/training split before didn’t have the concept of time. Here we are splitting things in the middle of a year.

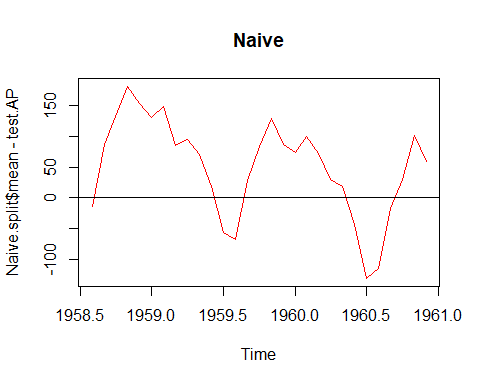
Let’s proceed to see where it gets us.

Mean.split <- meanf(train.AP, h=testObs)  
Naive.split <- naive(train.AP, h=testObs)  
Seasonal.split <- snaive(train.AP, h=testObs)  
Drift.split <- rwf(train.AP, drift=TRUE, h=testObs)  
  
plot(AP, plot.type="single",main="Simple Forecasts", ylab="Airline Passengers")  
lines(Mean.split$mean, col = "blue")  
lines(Naive.split$mean, col = "red")  
lines(Seasonal.split$mean, col = "green")  
lines(Drift.split$mean, lty = 2)

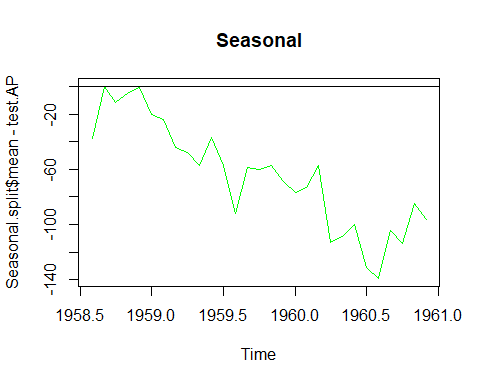


What do we think? How do we evaluate things? Let’s start by looking at the residuals. How far off is our simple prediction?

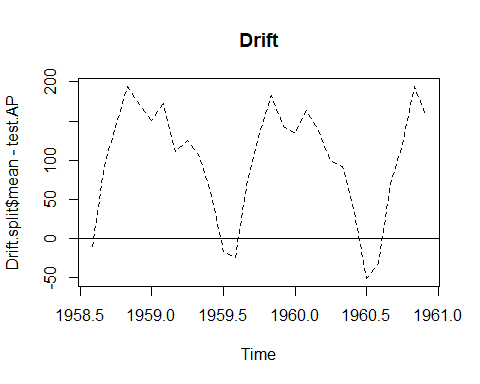
plot(Naive.split$mean - test.AP, col = "red", main = "Naive")  
abline(a = 0, b = 0)



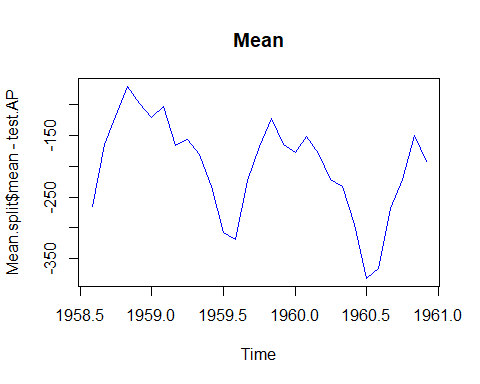
plot(Seasonal.split$mean - test.AP, col = "green",main = "Seasonal")  
abline(a = 0, b = 0)



plot(Drift.split$mean - test.AP, lty = 2,main = "Drift")  
abline(a = 0, b = 0)



plot(Mean.split$mean - test.AP, col = "blue", main = "Mean")  
abline(a = 0, b = 0)

 If a forecast is correct, the predicted value should equal the actual value. Given that we have many observations, we want our forecast to equal the actual ON AVERAGE - which is where the 0 mean for residuals comes from.

Remember we said they should be \* uncorrelated \* have mean zero \* have constant variance \* be normally distributed

So, how did this simple model do?

mean(test.AP - Mean.split$mean)

## [1] 200.3625

mean(test.AP - Naive.split$mean)

## [1] -50.68966

mean(test.AP - Seasonal.split$mean)

## [1] 64.75862

mean(test.AP - Drift.split$mean)

## [1] -100.5581

It doesn’t look like the mean of the residuals are zero. Are these models bad? In this simple case, probably.

How do we assess the accuracy of a forecast? Remember, we have several measures. Luckily, they are all packaged into one command.

accuracy(Mean.split, test.AP)

## ME RMSE MAE MPE MAPE MASE  
## Training set -3.002671e-15 90.94976 75.41837 -15.53870 36.51114 2.563727  
## Test set 2.003625e+02 215.05727 200.36252 43.85248 43.85248 6.811003  
## ACF1 Theil's U  
## Training set 0.9148518 NA  
## Test set 0.7628574 3.954492

accuracy(Naive.split, test.AP)

## ME RMSE MAE MPE MAPE MASE  
## Training set 3.324561 26.91752 21.06140 0.7590861 8.702959 0.7159487  
## Test set -50.689655 93.13394 81.44828 -14.8934393 20.200133 2.7687038  
## ACF1 Theil's U  
## Training set 0.2730432 NA  
## Test set 0.7628574 2.099392

accuracy(Seasonal.split, test.AP)

## ME RMSE MAE MPE MAPE MASE  
## Training set 29.08738 33.04954 29.41748 11.68056 11.83988 1.000000  
## Test set 64.75862 75.23389 64.75862 14.03790 14.03790 2.201366  
## ACF1 Theil's U  
## Training set 0.7918015 NA  
## Test set 0.8310727 1.409712

accuracy(Drift.split, test.AP)

## ME RMSE MAE MPE MAPE  
## Training set 4.986843e-15 26.71142 20.9092 -0.8297474 8.712336  
## Test set -1.005581e+02 122.63239 109.9138 -26.0282814 27.625750  
## MASE ACF1 Theil's U  
## Training set 0.7107749 0.2730432 NA  
## Test set 3.7363435 0.6644454 2.759247

Another example: Stock market data NOTE: working with daily data in R is painful. We will use a simple format here and get back to the issues of dealing with “raw” data later.

#Dow jones daily data  
library(fma)  
  
dj <- dowjones  
class(dj)

## [1] "ts"

length(dj)

## [1] 78

start(dj)

## [1] 1 1

end(dj)

## [1] 78 1

head(dj)

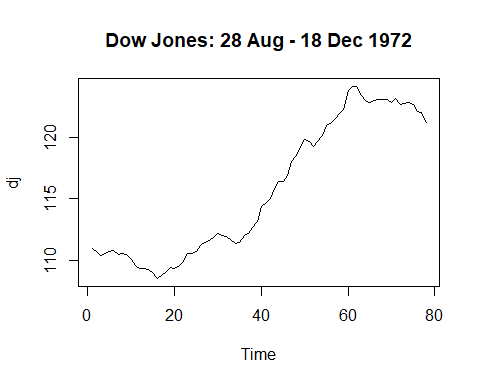
## Time Series:  
## Start = 1   
## End = 6   
## Frequency = 1   
## [1] 110.94 110.69 110.43 110.56 110.75 110.84

summary(dj)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 108.5 110.6 113.8 115.7 121.9 124.1

This is a time series that has been rescaled to take out the date part. It’s daily closing prices for the Dow Jones from 28 Aug - 18 Dec 1972

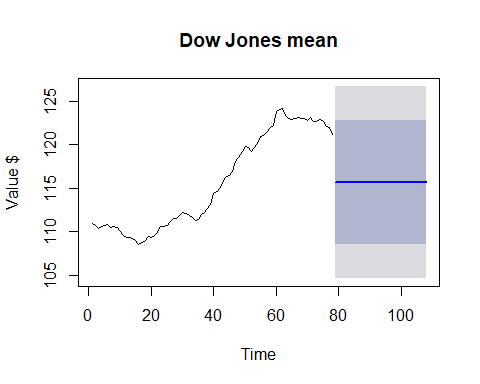
plot(dj, main = "Dow Jones: 28 Aug - 18 Dec 1972")



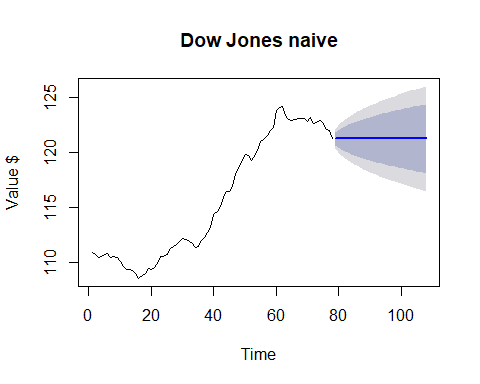
Let’s see how the simple forecasts do. What if we forecast out 30 days?

### Some simple forecasts

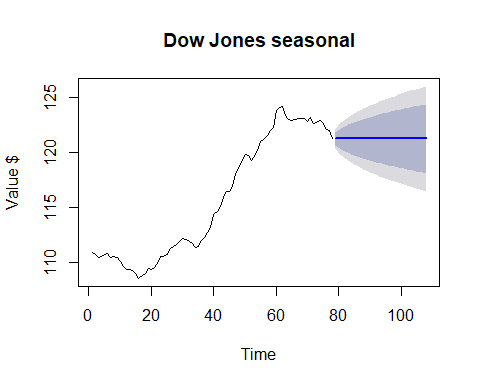
plot(meanf(dj, h = 30), xlab = "Time", ylab = "Value $", main = "Dow Jones mean")



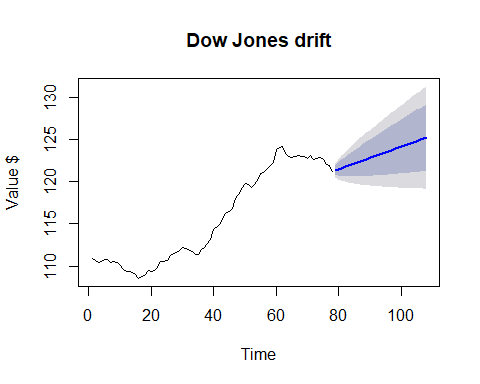
plot(naive(dj, h = 30), xlab = "Time", ylab = "Value $", main = "Dow Jones naive")



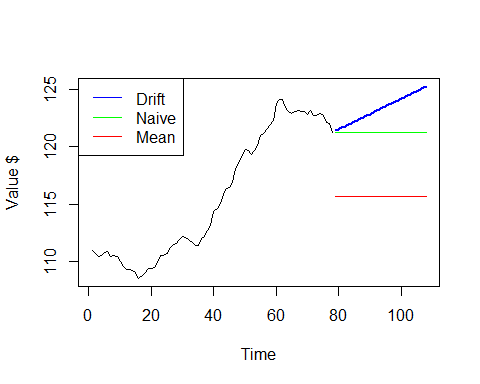
plot(snaive(dj, h = 30), xlab = "Time", ylab = "Value $", main = "Dow Jones seasonal")



plot(rwf(dj, drift = TRUE, h = 30), xlab = "Time", ylab = "Value $", main = "Dow Jones drift")



# All on one chart  
plot(rwf(dj, drift=TRUE, h=30, level=0), xlab="Time", ylab="Value $", main="")  
lines(naive(dj, h=30, level=0)$mean, xlab="", ylab="", main="", col="green")  
lines(meanf(dj, h=30, level=0)$mean, xlab="", ylab="", main="", col="red")  
  
legend("topleft",  
 legend = c("Drift", "Naive", "Mean"),  
 col = c("blue", "green", "red"), lty=1)



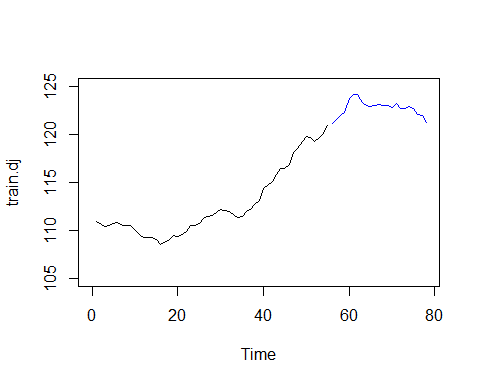
These are “out of sample forecasts” - so we can’t really gauge the accuracy. Let’s do our in sample test.

How about the test/train split? What makes sense?

#because this is less than one year's worth of daily data, we can still use a percentage approach.  
trainObs = round(length(dj) \* .7)  
trainObs

## [1] 55

train.dj <- window(dj, end = trainObs)  
  
test.dj <- window(dj, start = trainObs+1)  
  
  
plot(train.dj, ylim = c(105, 125), xlim = c(0,length(dj)))  
lines(test.dj, col = "blue")

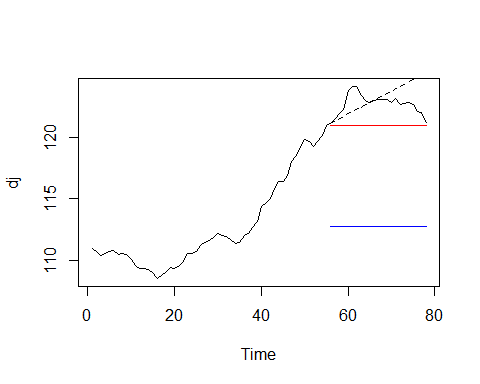


Okay, now for our simple forecasts

Mean.dj.split <- meanf(train.dj, h=length(test.dj))  
Naive.dj.split <- naive(train.dj, h=length(test.dj))  
Seasonal.dj.split <- snaive(train.dj, h=length(test.dj))  
Drift.dj.split <- rwf(train.dj, drift = TRUE, h=length(test.dj))

First, let’s look at the pictures.

plot(dj)  
lines(Mean.dj.split$mean, col = "blue")  
lines(Naive.dj.split$mean, col = "red")  
lines(Drift.dj.split$mean, lty = 2)

 Which model would you use?

Let’s look at the accuracy measures.

accuracy(Mean.dj.split, test.dj)

## ME RMSE MAE MPE MAPE  
## Training set -4.392446e-15 3.574552 2.968899 -0.09768495 2.600730  
## Test set 9.986672e+00 10.017437 9.986672 8.13366410 8.133664  
## MASE ACF1 Theil's U  
## Training set 8.596276 0.9422896 NA  
## Test set 28.915833 0.6516231 20.92774

accuracy(Naive.dj.split, test.dj)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.1857407 0.4329336 0.3453704 0.1595713 0.3031997 1.000000  
## Test set 1.7552174 1.9225538 1.7552174 1.4261661 1.4261661 5.082131  
## ACF1 Theil's U  
## Training set 0.4742188 NA  
## Test set 0.6516231 4.06902

accuracy(Drift.dj.split, test.dj)

## ME RMSE MAE MPE MAPE MASE  
## Training set 0.0000000 0.391065 0.3146296 -0.00529669 0.2776078 0.910992  
## Test set -0.4736715 1.602468 1.1960064 -0.39074811 0.9751396 3.462968  
## ACF1 Theil's U  
## Training set 0.4742188 NA  
## Test set 0.8281628 3.407816