Stat 416 Statistical Analysis of Time Series

Julia Schedler

2024-09-23

Table of contents

Welcome to Time Series!

The lecture slides and code here are heavily drawn from the book I have selected for this course.

Also, big thanks to our Lab Assistant, Bena Smith, for serving as editor for these notes!

Part I

Week 1

Welcome to week 1! We will cover Chapter 1 and part of Chapter 2.

1 Lecture 1

2 Welcome!

2.1 Agenda

10:10 Introductions
10:30 Course Rhythm/Roadmap
10:45 Syllabus
11:00 R Setup
11:15 Activity
11:45 Introduction to Time Series Models

Extra time: Get started on Exercises

2.2 Introductions

Arrange yourselves into groups of 3-4 and share the following:

- Name
- Major
- Whether you spend more time thinking about the past or the future
- Whether you are more certain when you think about the past or the future

Please be prepared to share summary data with the class!

2.3 About Me (Professional)

- Cal Poly SLO B.S. in Statistics and Pure Math
- PhD (and MA) in Statistics from Rice University
  - Expertise: Spatial/Spatiotemporal Statistics (specifically spatial weight matrices)
  - Also did graduate certificate in teaching and learning

- 1.5 years authoring online interactive course material for zyBooks/Wiley (EdTech portion Higher Education industry)
- 1.5 years managing a team of Statistics authors at zyBooks/Wiley
- 1.5 years as Research Scientist (wastewater epidemiology) at Rice University

2.4 Teaching Philosophy

- Minimize yapping
- Promote collaboration
- Provide varied opportunities for feedback

2.5 Course Rhythm

- Assignments due Monday nights at Midnight (11:59:59 PM)
- New assignments posted Mondays at 5pm
- Quizzes due Thursday nights at midnight
- One Midterm on Wednesday October 23, in class
- One cumulative final exam
  - Section 1 (10am): Wednesday, December 11 from 10am-1pm
  - Section 2 (2pm): Friday, December 13 from 1pm-4pm

2.6 Syllabus

Syllabus on Canvas (section 1)

Syllabus on Canvas (section 2)

3 Software Setup

3.1 Installing R

- Go to https://cran.r-project.org/
- Select your operating system
- Follow the download instructions

3.2 Installing RStudio

- Go to https://posit.co/download/rstudio-desktop/
- Step 2 should have a clickable link with your operating system (auto-detected)
- Follow the download instructions

3.3 Getting Started

- In R Studio, Click File –> New –> Quarto Document
- Title it Lecture 1 Notes
- Delete the template material
- Add setup chunk

```
library(astsa)
```

4 Activity

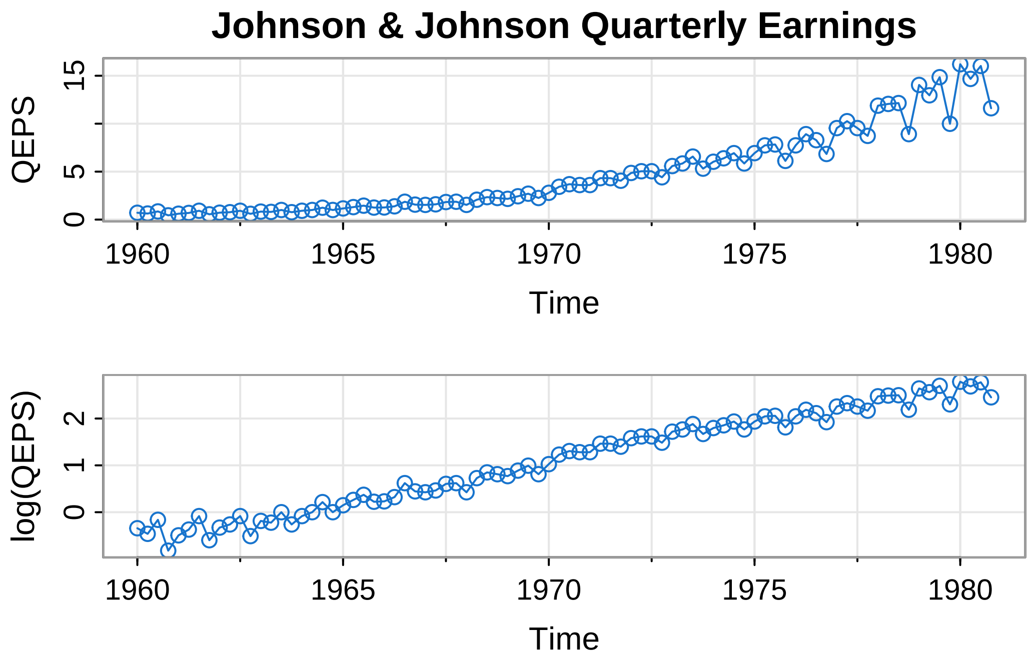
4.1 Group time!

Split into groups of 4, I will come around and assign an example to you

- In your quarto document, create a heading with a title of your example and "Equations" and "Visualizatons"
- Read over the description of the example (access the book through Canvas) install and load the `astsa` package.
- Copy the R code from https://github.com/nickpoison/tsda/blob/main/Rcode.md#chapter-1
- Run the R code and verify whether you can reproduce the figure from the text
- Write down a research question that could be answered using the time series data for your example

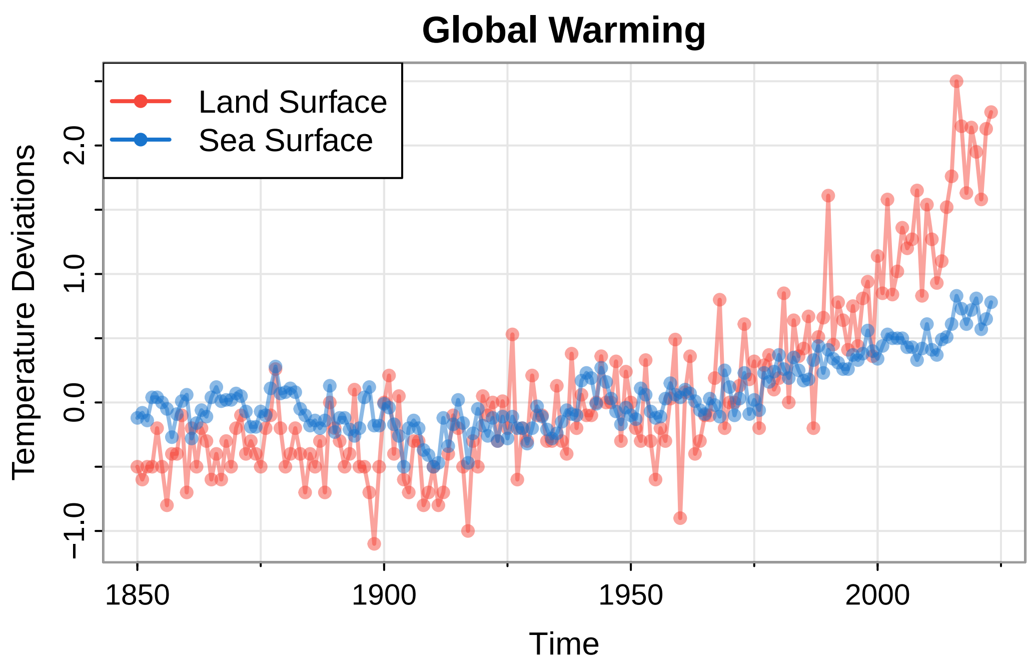
4.2 Example 1.1 (Quarterly Earnings)

```
par(mfrow=2:1)
tsplot(jj, ylab="QEPS", type="o", col=4, main="Johnson & Johnson Quarterly Earnings")
tsplot(log(jj), ylab="log(QEPS)", type="o", col=4)
```



4.3 Example 1.2 (Climate Change)

```
tsplot(cbind(gtemp_land,gtemp_ocean), spaghetti=TRUE, col = astsa.col(c(2,4), .5),
        lwd=2, type="o", pch=20, ylab="Temperature Deviations", main="Global Warming")
legend("topleft", col=c(2,4), lty=1, lwd=2, pch=20,  bg="white",
        legend=c("Land Surface", "Sea Surface"))
```



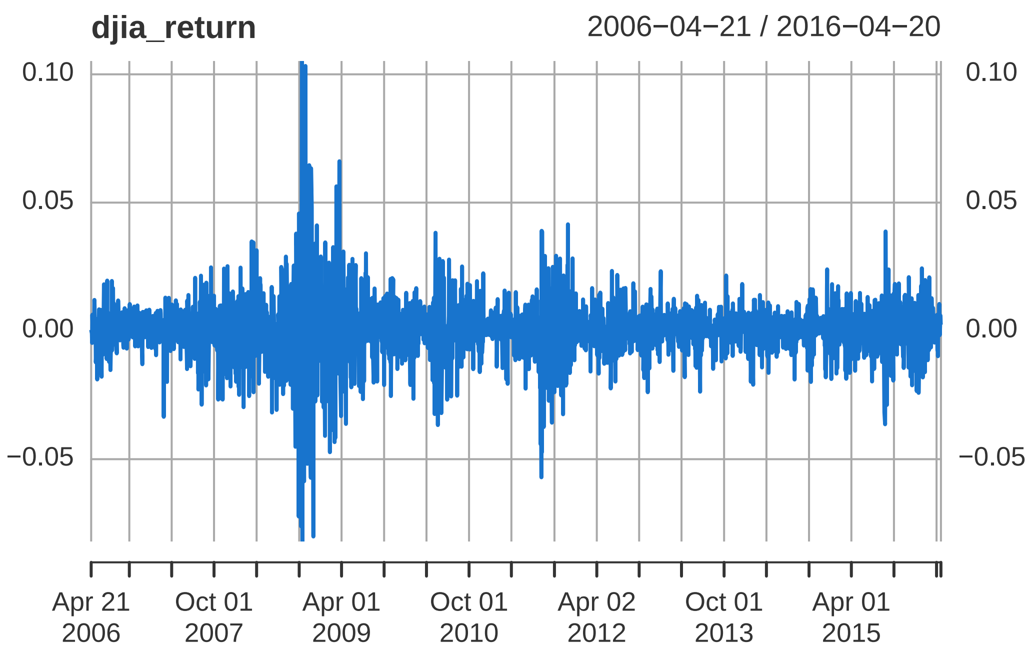
4.4 Example 1.3 (Dow Jones Industrial Average)

```
library(xts)     # install.packages("xts") if you don't have it already
```

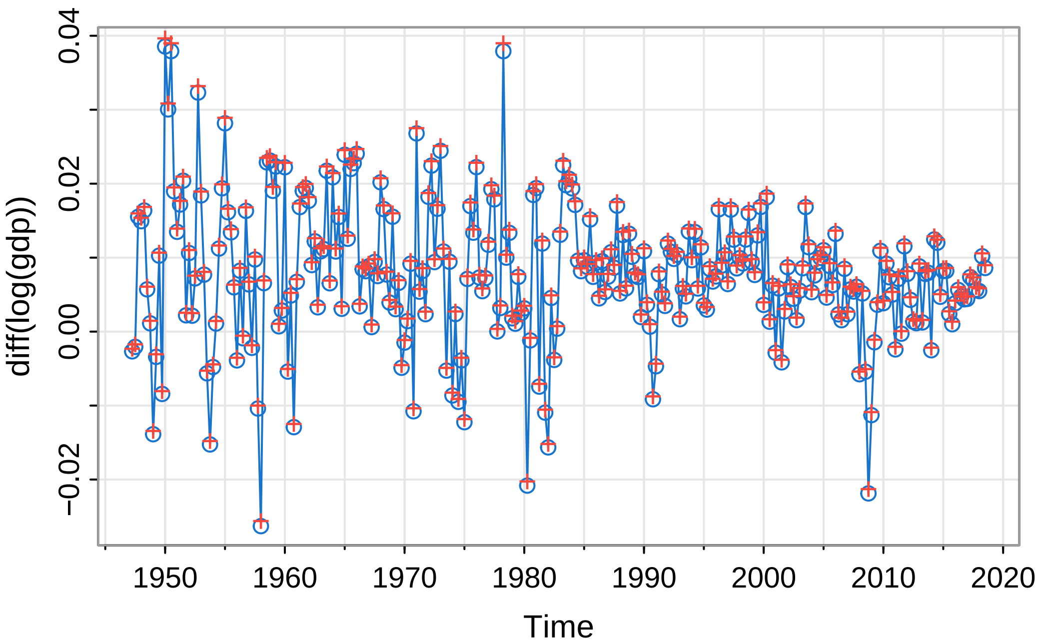
```
Loading required package: zoo


Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```
djia_return = diff(log(djia$Close))[-1]
#par(mfrow=2:1)
plot(djia$Close, col=4)
```


```
plot(djia_return, col=4)
```

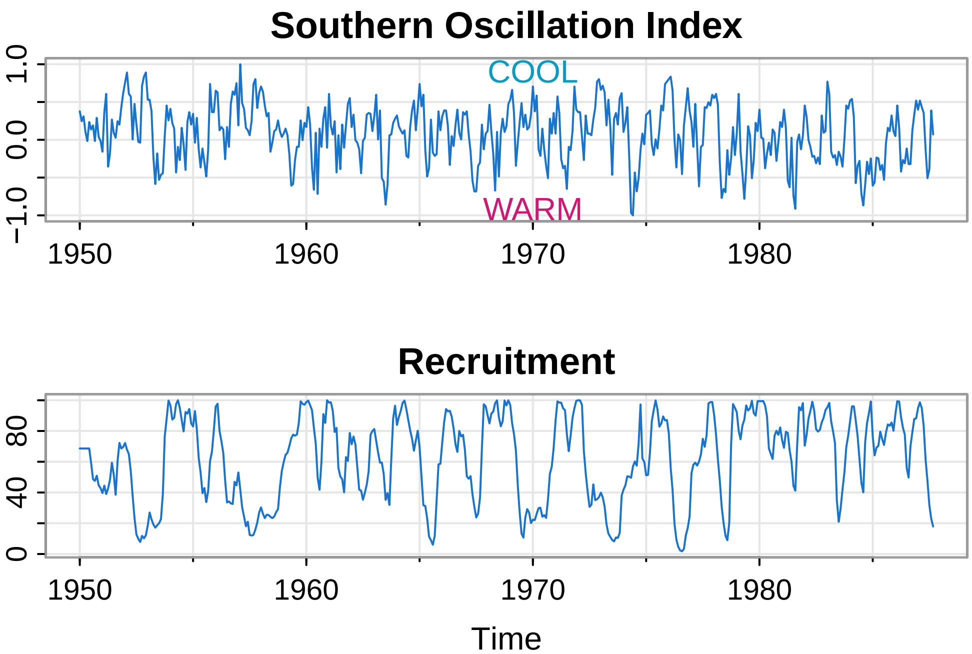


```
tsplot(diff(log(gdp)), type="o", col=4)       # using diff log
points(diff(gdp)/lag(gdp,-1), pch="+", col=2) # actual return
```



4.5 Example 1.4 El Niño

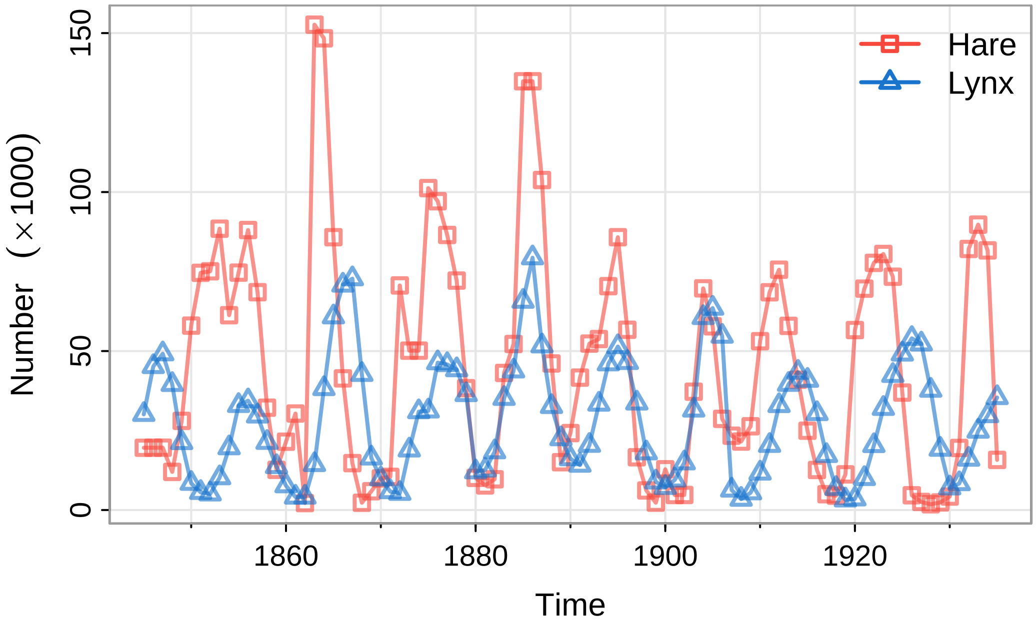
```
par(mfrow = c(2,1))
tsplot(soi, ylab="", xlab="", main="Southern Oscillation Index", col=4)
text(1970, .91, "COOL", col=5)
text(1970,-.91, "WARM", col=6)
tsplot(rec, ylab="", main="Recruitment", col=4)
```



4.6 Example 1.5 (Predator-Prey Interactions)

Link to more info!

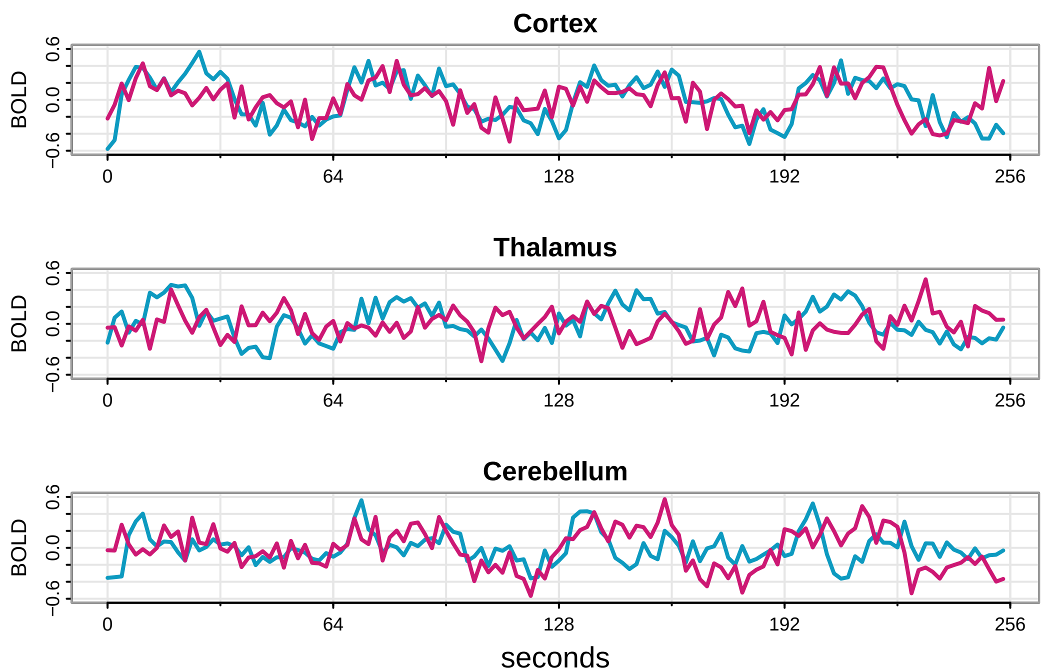
```
tsplot(cbind(Hare,Lynx), col = astsa.col(c(2,4), .6), lwd=2, type="o", pch=c(0,2),
         spaghetti=TRUE, ylab=expression(Number~~~(""%*% 1000)))
legend("topright", col=c(2,4), lty=1, lwd=2, pch=c(0,2), legend=c("Hare", "Lynx"), bty="n")
```



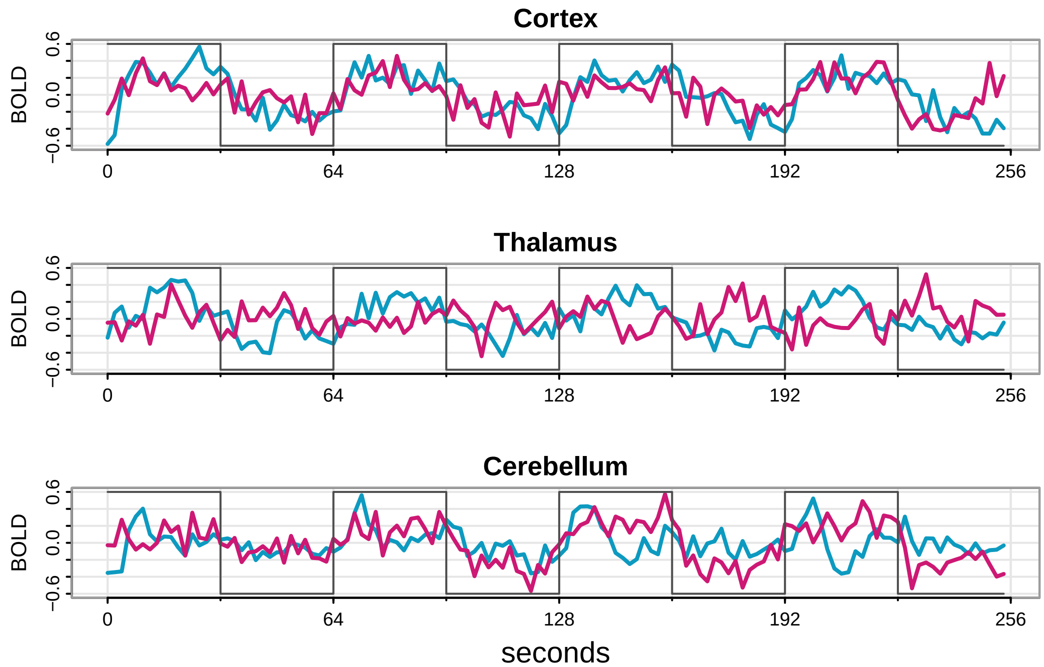
4.6.1 Cute animal pictures

4.7 Example 1.6 fMRI Imaging

```
par(mfrow=c(3,1))
x = ts(fmri1[,4:9], start=0, freq=32)         # data
names = c("Cortex","Thalamus","Cerebellum")
u = ts(rep(c(rep(.6,16), rep(-.6,16)), 4), start=0, freq=32) # stimulus signal

for (i in 1:3){
 j = 2*i-1
 tsplot(x[,j:(j+1)], ylab="BOLD", xlab="", main=names[i], col=5:6, ylim=c(-.6,.6),
        lwd=2, xaxt="n", spaghetti=TRUE)
 axis(seq(0,256,64), side=1, at=0:4)
 #lines(u, type="s", col=gray(.3))
}
mtext("seconds", side=1, line=1.75, cex=.9)
```




```
par(mfrow=c(3,1))
x = ts(fmri1[,4:9], start=0, freq=32)        # data
names = c("Cortex","Thalamus","Cerebellum")
u = ts(rep(c(rep(.6,16), rep(-.6,16)), 4), start=0, freq=32) # stimulus signal

for (i in 1:3){
 j = 2*i-1
 tsplot(x[,j:(j+1)], ylab="BOLD", xlab="", main=names[i], col=5:6, ylim=c(-.6,.6),
        lwd=2, xaxt="n", spaghetti=TRUE)
 axis(seq(0,256,64), side=1, at=0:4)
 lines(u, type="s", col=gray(.3))
}
mtext("seconds", side=1, line=1.75, cex=.9)
```


Cortex
Thalamus
Cerebellum
BOLD
0.6
0.0
−0.6
0
64
128
192
256
seconds

5 Introduction to Time Series Models

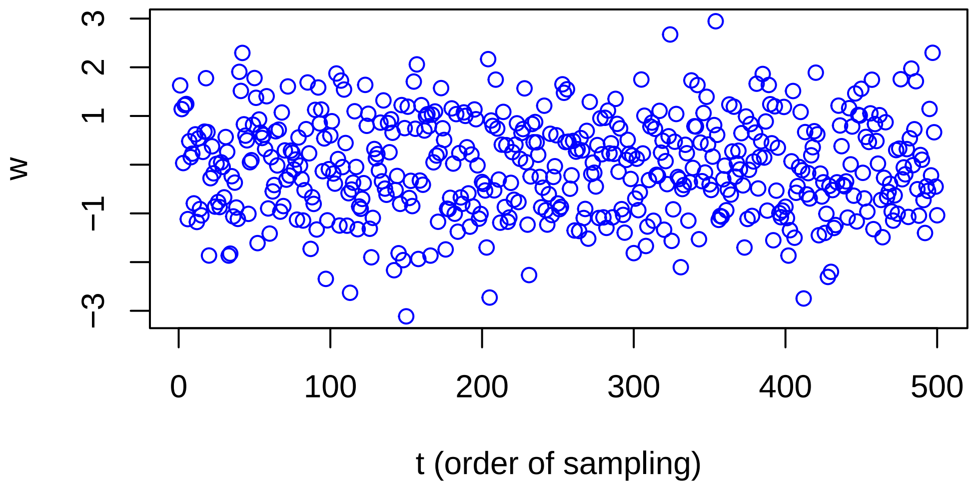
5.1 White Noise

- in general, a collection of random variables w_t
  - uncorrelated
  - mean 0, variance σ_w^2
  - denoted $w_t \sim wn(0, \sigma_w^2)$
- for us, usually independent and identically distributed (i.i.d.) normal
  - $w_t \sim \text{iid } N(0, \sigma_w^2)$

5.2 Plotting White Noise

Which example does this bear the most resemblance to?

```
w <- rnorm(500, 0, 1)
plot(w, type = "p", col = "blue", xlab = "t (order of sampling)")
```



5.3 What White Noise *isn't*

- serially correlated – no temporal structure
- smooth – "nice" trend/temporal structure

How can we build this "nice" structure into the model?

5.4 Moving Averages, Smoothing, and Filtering

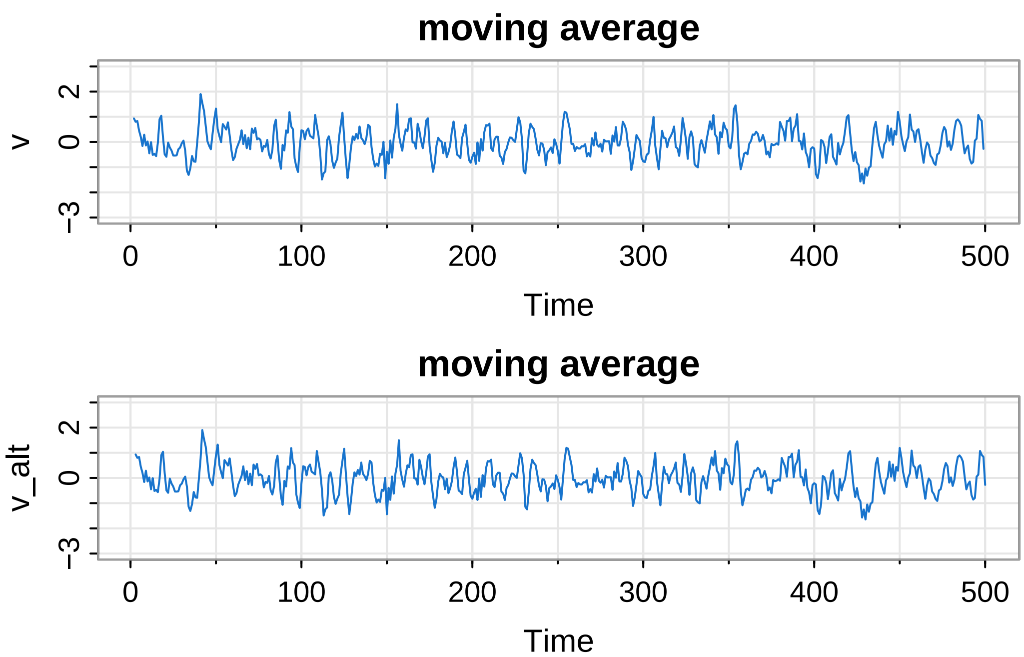
Replace w_t with an average of its current value and two previous values:

$$v_t = \frac{1}{3}(w_{t-2} + w_{t-1} + w_t)$$

- Why do we divide by 3?
- If $w_t \sim \text{iid } N(0, \sigma_w^2)$, what is the distribution of v_t?
- Why only the previous two values? Why not one in the past and one in the future?

5.5 Plotting a Moving Average

```
v = stats::filter(w, sides = 2, filter = rep(1/3, 3))
v_alt = stats::filter(w, sides = 1, filter = rep(1/3,3))
par(mfrow=2:1)
tsplot(v, ylim = c(-3, 3), col = 4, main="moving average")
tsplot(v_alt, ylim = c(-3, 3), col = 4, main="moving average")
```



Compare this moving average to the SOI and Recruitment series. How do they differ?

5.6 Autoregressions

Starting with white noise w_t, consider the equation:

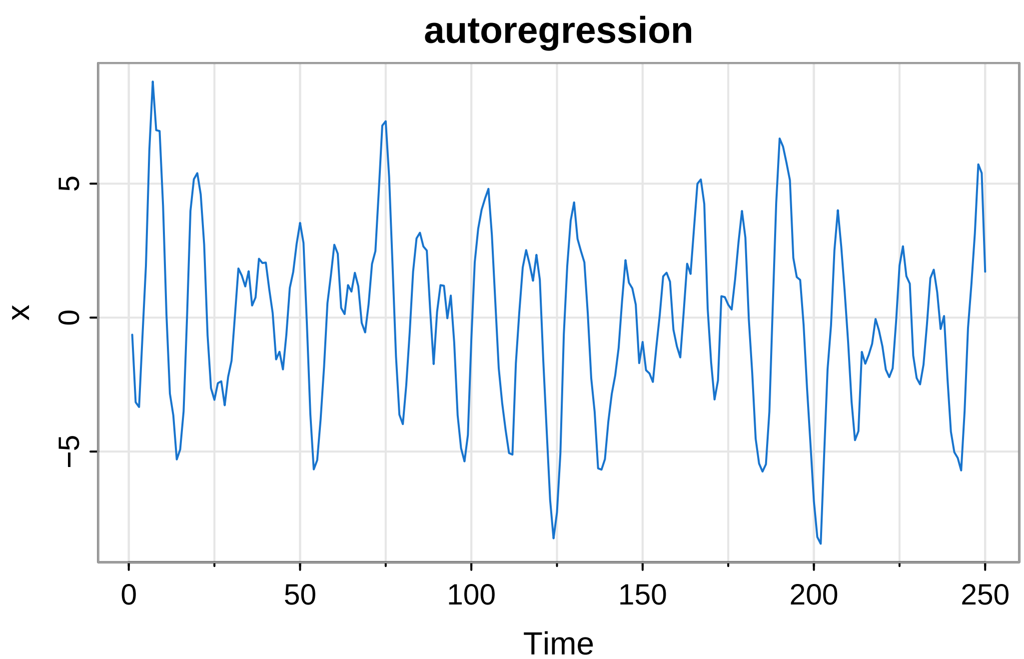
$$x_t = 1.5x_{t-1} - 0.75x_{t-2} + w_t$$

- a "second-order equation" (why?)
- A regression of the current value x_t of a time series as a function of the past two values of the series

- "auto" means self
- recall (multiple) regression of Y on $X = (X_1, X_2)$ is $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$ and compare to autoregression formula above
- See (or hear) details in textbook page 11

5.7 Plotting Autoregressions

```
set.seed(90210)
w = rnorm(250 + 50) # 50 extra to avoid startup problems
x = filter(w, filter=c(1.5,-.75), method="recursive")[-(1:50)]
tsplot(x, main="autoregression", col=4)
```



5.8 Random Walk with Drift

Again starting with white noise $w_t \sim wn(0, \sigma_2^2)$, consider the time series

$$x_t = \delta + x_{t-1} + w_t$$

This is called the "random walk with drift" model.

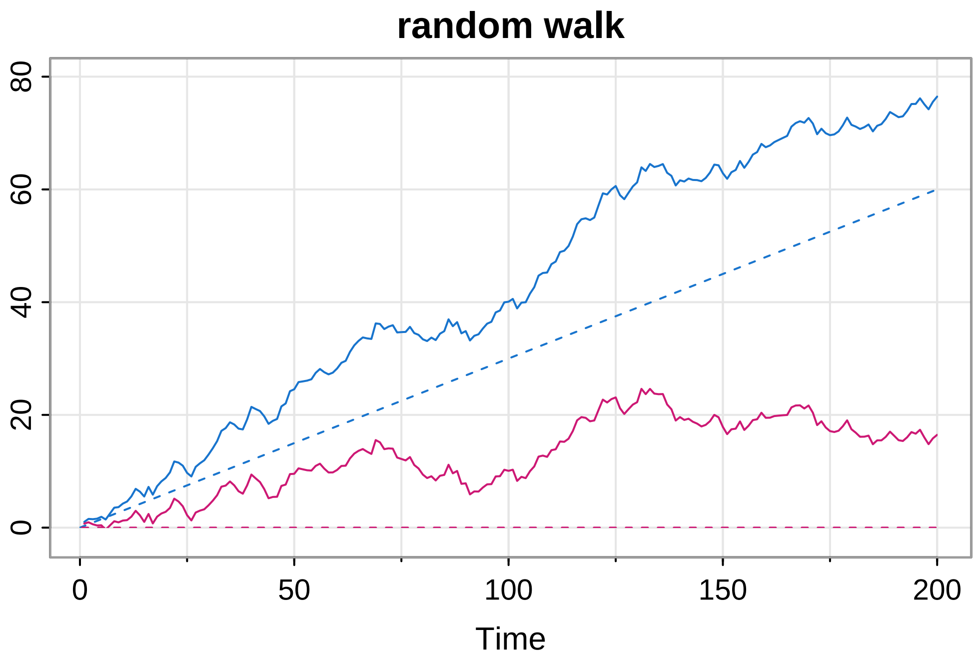
- δ is the drift term ($\delta = 0$ corresponds to "random walk"- no drift)
- initial condition $x_0 = 0$

Can be rewritten

$$x_t = \delta t + \sum_{j=1}^{t} w_j$$

5.9 Plotting a Random Walk with Drift

```
set.seed(314159265) # so you can reproduce the results
w  = rnorm(200)  ## Gaussian white noise
x  = cumsum(w)
wd = w +.3
xd = cumsum(wd)
tsplot(xd, ylim=c(-2,80), main="random walk", ylab="", col=4)
 clip(0, 200, 0, 80)
 abline(a=0, b=.3, lty=2, col=4) # drift
lines(x, col=6)
 clip(0, 200, 0, 80)
 abline(h=0, col=6, lty=2)
```



5.10 Signal Plus Noise

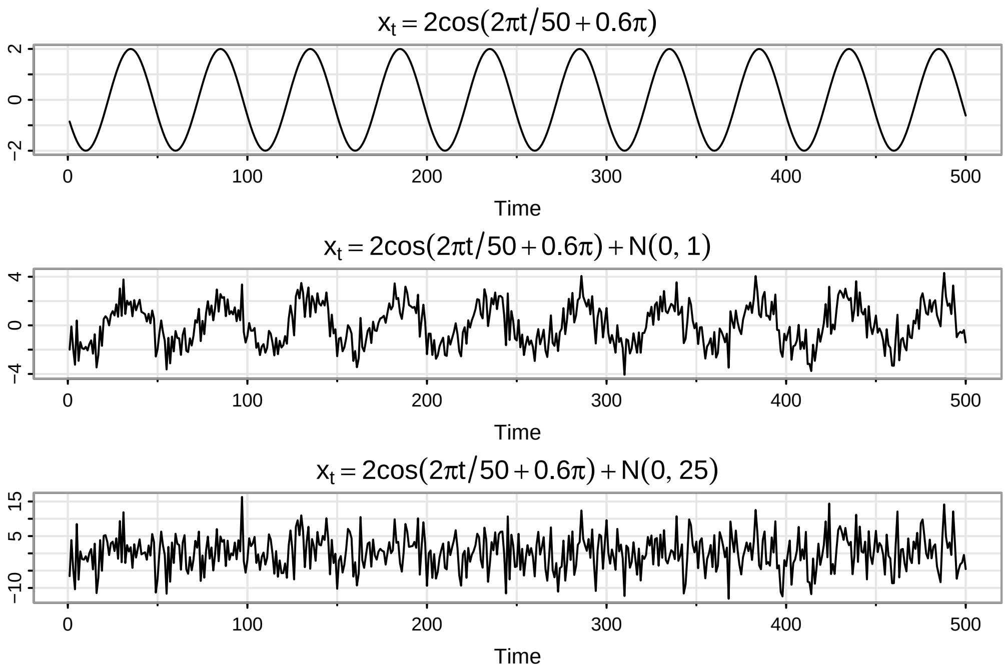
Consider the model:

$$x_t = 2\cos(2\pi\frac{t+15}{50}) + w_t$$

- $2\cos(2\pi\frac{t+15}{50})$ is the signal
- w_t is the noise

5.11 Plotting Signal Plus Noise (two scenarios)

```
# cs = 2*cos(2*pi*(1:500)/50 + .6*pi)    # as in the text
cs = 2*cos(2*pi*(1:500+15)/50)           # same thing
w  = rnorm(500,0,1)
par(mfrow=c(3,1))
tsplot(cs, ylab="", main = expression(x[t]==2*cos(2*pi*t/50+.6*pi)))
tsplot(cs + w, ylab="", main = expression(x[t]==2*cos(2*pi*t/50+.6*pi)+N(0,1)))
tsplot(cs + 5*w, ylab="", main = expression(x[t]==2*cos(2*pi*t/50+.6*pi)+N(0,25)))
```



5.12 Next Time

- Exercises at the end of chapter 1
- Start Chapter 2
    - Review definition of covariance, correlation, expected value, and variance (good use of AI– prompt then Wikipedia?)

6 Lecture 2

6.1 Recap from last time

- Several examples of time series data sets
- Experience plotting the time series
- Exposure to some common time series models

6.2 Today

- Notation review
- Mean and covariance function of a time series
- R code activity
- Stationarity (if time)

6.3 Coming up/notices

- I combined the Canvas sections (applies to section 2)
- Quiz 1 posted today, due tomorrow at midnight (20 minutes to do it)
- Assignment 1 will also be posted today, due Monday at midnight (boundary between Monday and Tuesday)
- Next week's office hours: M 4-5, T 12-2

7 Review of notation

7.1 Notation and Data- White noise

"Let w_t be a white noise series"

t	Random Variable	Example data
1	$w_1 \sim N(0, \sigma_w^2)$	-0.0777401
2	$w_2 \sim N(0, \sigma_w^2)$	-1.2742499
	$\vdots$	$\vdots$
t	$w_t \sim N(0, \sigma_w^2)$	-0.3434436
$\vdots$	$\vdots$	$\vdots$
n	$w_n \sim N(0, \sigma_w^2)$	0.068451

If we interpret the collection of w_t as a random vector, then $w_t \sim MVN(\vec{0}, I)$ (why I?)

Note: sometimes w_t could mean a (univariate) value of a white noise series for a particular time t (kind of like how you refer to an arbitrary x_i when you have a sample $x_1, \dots, x_n$).

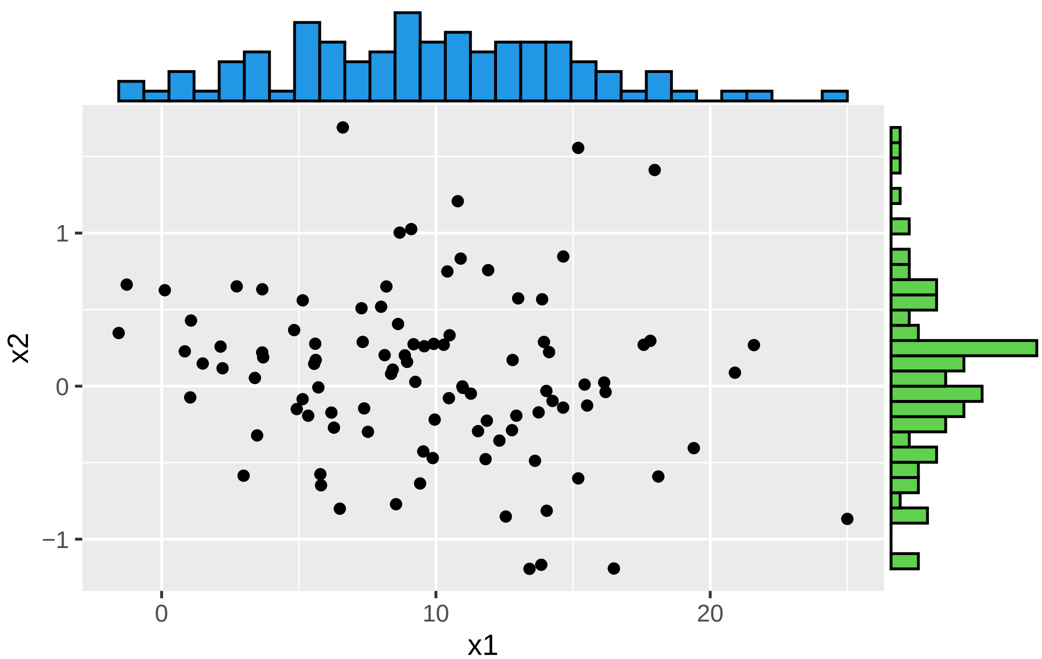
7.2 (Aside) The Multivariate normal distribution

Let's look on Wikipedia. What are the parameters?

- mean **vector**
- variance (covariance) **matrix**
  - If the covariance matrix is the identity matrix, the the covariances are 0

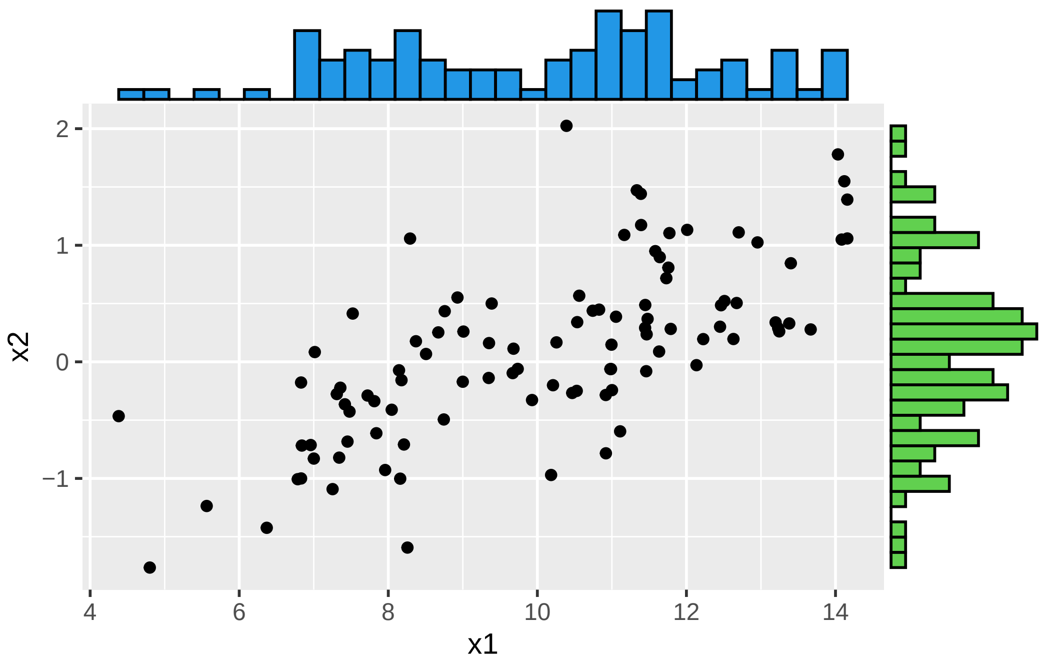
7.3 (Aside) Bivariate normal distribution for uncorrelated case

```r
# install.packages("ggplot2")
# install.packages("ggExtra")
library(ggplot2)
library(ggExtra)

x1 <- rnorm(100, 10, 5)
x2 <- rnorm(100, .1, .5)

x <- data.frame(x1, x2)
# Save the scatter plot in a variable
p <- ggplot(x, aes(x = x1, y = x2)) +
  geom_point()

# Arguments for each marginal histogram
ggMarginal(p, type = "histogram",
           xparams = list(fill = 4),
           yparams = list(fill = 3))
```



7.4 (Aside) Bivariate normal distribution for correlated case

```
#install.packages("MASS")
library(MASS)

mu <- c(10, .1)
varcov <- matrix(c(5, 1, 1, .5),
                 ncol = 2)
x<- mvrnorm(100, mu = mu, Sigma =varcov)
x <- data.frame(x1 = x[,1], x2 = x[,2])
# Save the scatter plot in a variable
p <- ggplot(x, aes(x = x1, y = x2)) +
  geom_point()

# Arguments for each marginal histogram
ggMarginal(p, type = "histogram",
           xparams = list(fill = 4),
           yparams = list(fill = 3))
```



7.5 Building time series models from White Noise

Model	Inputs	Output
White noise	probability distribution, independence assumption, σ_w^2	
Moving average with p points	$w_1, w_2, \ldots, w_n$	
Autoregression of order p	$w_1, w_2, \ldots, w_n$ and $\phi = (\phi_1, \ldots, \phi_p)$	
Random walk with drift	$w_1, w_2, \ldots, w_n$ and δ	
Signal plus noise	$w_1, w_2, \ldots, w_n$ and a function $f(t)$	

Identify which of the inputs are random variables, pre-specified constants, pre-specified functions, or parameters to be estimated.

7.6 Building time series models from White Noise

Model	Inputs	Output
White noise	probability distribution, independence assumption, σ_w^2	$w_1, w_2, \ldots, w_n$; for each $t = 1, \ldots, n$ we have $w_t \sim N(0, \sigma_w^2)$
Moving average with p points	$w_1, w_2, \ldots, w_n$	$v_t = \frac{1}{p} \sum_{i=1}^{p} w_{t-(p-i)}$
Autoregression of order p	$w_1, w_2, \ldots, w_n$ and $\phi = (\phi_1, \ldots, \phi_p)$	$x_t = \sum_{i=1}^{p} \phi_i x_{t-i} + w_t$
Random walk with drift	$w_1, w_2, \ldots, w_n$ and δ	$x_t = \delta + x_{t-1} + w_t$
Signal plus noise	$w_1, w_2, \ldots, w_n$ and a function $f(t)$	$x_t = f(t) + w_t$

Identify which of the inputs are random variables, pre-specified constants, pre-specified functions, or parameters to be estimated.

7.7 Notation and Data

Consider the general version of the autoregressive model of order 1:

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + w_t$$

If you had data representing this process, what would it look like in R?

7.8 Notation and Data

Suppose $\phi_1 = 1.5$ and $\phi_2 = -0.75$.

```
set.seed(90210)
w = rnorm(250 + 50) # 50 extra to avoid startup problems
x = filter(w, filter=c(1.5,-.75), method="recursive")[-(1:50)]
x
```

```
  [1] -0.635871231 -3.159366457 -3.336983558 -0.670017029  1.928041062
  [6]  6.262719337  8.811276769  6.994297589  6.964249838  4.172630149
 [11]  0.109387891 -2.838470465 -3.650839732 -5.293859716 -4.924149166
 [16] -3.496962661 -0.001206165  3.982335012  5.166059171  5.391303364
 [21]  4.598152813  2.726933281 -0.656314289 -2.634587218 -3.070392399
 [26] -2.447369835 -2.377035961 -3.272222376 -2.212579163 -1.609152064
 [31]  0.088151906  1.834292884  1.566977751  1.162326919  1.731270484
 [36]  0.452095019  0.751851590  2.197474589  2.037090911  2.053962776
 [41]  1.057859241  0.173798276 -1.559467228 -1.275347235 -1.934447794
 [46] -0.637721288  1.108281203  1.703590245  2.757116948  3.535041828
 [51]  2.785518718 -0.255025902 -3.601017151 -5.665073618 -5.320832378
 [56] -3.801870752 -1.843797185  0.540063136  1.577259338  2.719389114
 [61]  2.386440948  0.360417214  0.130240105  1.213682241  0.970840444
 [66]  1.672645132  1.169230978 -0.197824215 -0.552895930  0.483295378
 [71]  2.002207259  2.483139041  4.761206339  7.166338800  7.329547964
 [76]  5.238955522  1.955859515 -1.445155254 -3.624225029 -3.976740747
 [81] -2.522488940 -0.560280191  1.716462129  2.956985039  3.167747954
 [86]  2.655920142  2.503263867  0.243980727 -1.733850533  0.218414375
 [91]  1.212655465  1.188737220 -0.024525903  0.824315000 -0.929797989
 [96] -3.643408960 -4.872924684 -5.365789994 -4.379073769 -0.816292614
[101]  2.069716217  3.317790830  4.024356559  4.445225438  4.807260941
[106]  3.077726417  0.597309443 -1.889650709 -3.193428803 -4.189085934
[111] -5.056410971 -5.113692514 -1.701862879  0.197898712  1.872046685
[116]  2.519653174  1.995693810  1.375972346  2.342728546  1.412737664
[121] -1.604693814 -4.224595521 -6.808370201 -8.238970434 -7.267053979
[126] -5.073807901 -0.614874790  1.926334410  3.620792660  4.301297376
[131]  2.938794190  2.482699730  2.062144627  0.145550378 -2.263580334
[136] -3.515516041 -5.626740964 -5.675586843 -5.285219511 -3.877662550
[141] -2.843191932 -2.159754220 -1.134175851  0.621526810  2.144676177
[146]  1.301986893  1.090681772  0.483465932 -1.699760373 -0.907358670
[151] -1.964189610 -2.083464483 -2.401372850 -1.102177741  0.090984198
[156]  1.539763874  1.675986590  1.340872200 -0.451023892 -1.070116007
[161] -1.485934032  0.223487236  2.011408533  1.630095949  3.323091734
```

```
[166]  4.997983168  5.156394449  4.241727271  0.336603262 -1.668930450
[171] -3.056412007 -2.346607210  0.799083342  0.765047225  0.480923824
[176]  0.301524245  1.422952841  2.820001236  3.981388964  2.988835261
[181] -0.058956147 -2.066827932 -4.518505369 -5.447774381 -5.746818410
[186] -5.473607376 -3.515892394  0.432262861  4.283988479  6.685899229
[191]  6.379550991  5.781828167  5.127569880  2.228597185  1.512254758
[196]  1.407053783 -0.275040161 -2.623401872 -4.707758722 -6.845203817
[201] -8.189848947 -8.441072069 -5.100352049 -1.929194703 -0.289395357
[206]  2.511067946  4.007902754  2.638931037  0.953911823 -0.914044608
[211] -3.131803887 -4.574239309 -4.239263041 -1.278975512 -1.720543477
[216] -1.393708189 -0.978071153 -0.052109821 -0.479546542 -1.072444773
[221] -1.940146902 -2.221618511 -1.892476988 -0.145214604  1.941929437
[226]  2.662695670  1.548128421  1.266366609 -1.415637008 -2.255649300
[231] -2.492380384 -1.758574495 -0.272146596  1.472164787  1.788881267
[236]  0.946614002 -0.426152284  0.059796487 -2.263388225 -4.255693202
[241] -5.023127496 -5.240398677 -5.705131625 -3.494170488 -0.385992861
[246]  1.270003055  3.142585019  5.720389808  5.393790259  1.711581565
```

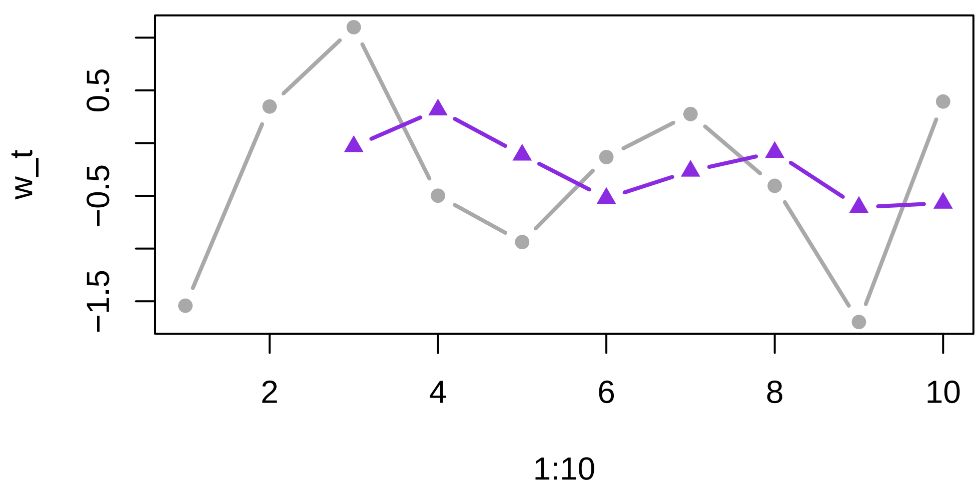
7.9 R example - Moving Average

```r
set.seed(70)

# generate white noise
w_t <- rnorm(10, 0, 1)

## manually lag terms
w_t1 <- c(NA, w_t[1:9])
w_t2 <- c(NA, NA, w_t[1:8])

## manually compute MA(3)
v_t <- (w_t + w_t1 + w_t2)/3

## compare the vectors
ma_3 <- cbind(v_t, w_t, w_t1, w_t2)
round(ma_3, 3)
```

```
         v_t    w_t   w_t1   w_t2
 [1,]     NA -1.542     NA     NA
 [2,]     NA  0.347 -1.542     NA
 [3,] -0.032  1.099  0.347 -1.542
```

```
 [4,]  0.316 -0.499  1.099  0.347
 [5,] -0.112 -0.938 -0.499  1.099
 [6,] -0.523 -0.132 -0.938 -0.499
 [7,] -0.265  0.276 -0.132 -0.938
 [8,] -0.087 -0.405  0.276 -0.132
 [9,] -0.609 -1.696 -0.405  0.276
[10,] -0.569  0.394 -1.696 -0.405
```

```r
## plot
#par(mfrow = 2:1)
plot(1:10, w_t, type = "b", lwd = 2, pch = 16, col = "darkgrey")
points(1:10, v_t, type = "b", lwd = 2, pch = 17, col = "blueviolet")
```



7.10 R example - Moving Average

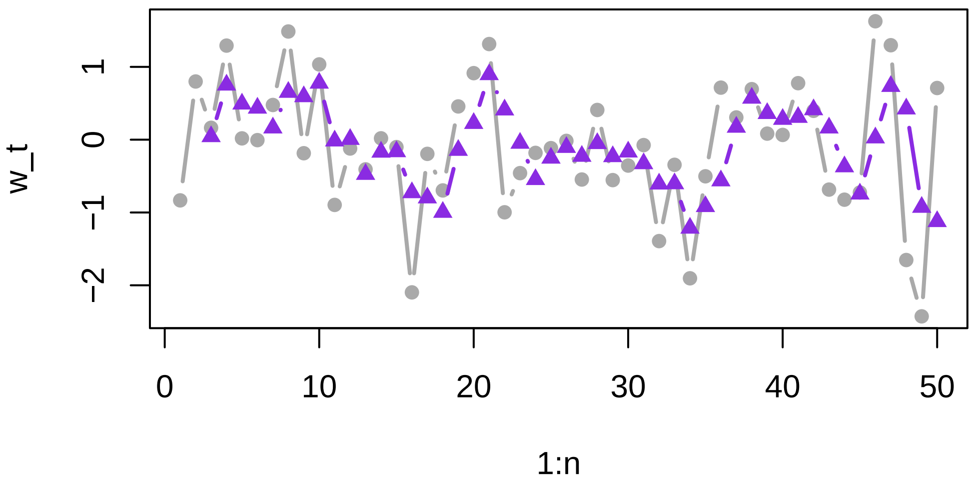
```r
# generate white noise
n = 50
w_t <- rnorm(n, 0, 1)

## manually lag terms
```

```r
w_t1 <- c(NA, w_t[1:(n-1)])
w_t2 <- c(NA, NA, w_t[1:(n-2)])

## manually compute MA(3)
v_t <- (w_t + w_t1 + w_t2)/3

## compare the vectors
ma_3 <- cbind(v_t, w_t, w_t1, w_t2)
round(ma_3, 3)
```

```
         v_t    w_t   w_t1   w_t2
 [1,]     NA -0.834     NA     NA
 [2,]     NA  0.799 -0.834     NA
 [3,]  0.043  0.163  0.799 -0.834
 [4,]  0.752  1.292  0.163  0.799
 [5,]  0.491  0.018  1.292  0.163
 [6,]  0.435 -0.006  0.018  1.292
 [7,]  0.163  0.476 -0.006  0.018
 [8,]  0.652  1.486  0.476 -0.006
 [9,]  0.592 -0.186  1.486  0.476
[10,]  0.778  1.034 -0.186  1.486
[11,] -0.016 -0.896  1.034 -0.186
[12,]  0.006 -0.121 -0.896  1.034
[13,] -0.475 -0.408 -0.121 -0.896
[14,] -0.170  0.019 -0.408 -0.121
[15,] -0.164 -0.102  0.019 -0.408
[16,] -0.727 -2.098 -0.102  0.019
[17,] -0.798 -0.195 -2.098 -0.102
[18,] -0.996 -0.697 -0.195 -2.098
[19,] -0.145  0.457 -0.697 -0.195
[20,]  0.225  0.914  0.457 -0.697
[21,]  0.895  1.314  0.914  0.457
[22,]  0.410 -0.998  1.314  0.914
[23,] -0.048 -0.459 -0.998  1.314
[24,] -0.546 -0.181 -0.459 -0.998
[25,] -0.252 -0.116 -0.181 -0.459
[26,] -0.105 -0.017 -0.116 -0.181
[27,] -0.227 -0.547 -0.017 -0.116
[28,] -0.052  0.408 -0.547 -0.017
[29,] -0.231 -0.555  0.408 -0.547
[30,] -0.168 -0.356 -0.555  0.408
[31,] -0.328 -0.074 -0.356 -0.555
```

```
[32,] -0.608 -1.393 -0.074 -0.356
[33,] -0.604 -0.345 -1.393 -0.074
[34,] -1.214 -1.904 -0.345 -1.393
[35,] -0.917 -0.503 -1.904 -0.345
[36,] -0.564  0.715 -0.503 -1.904
[37,]  0.173  0.306  0.715 -0.503
[38,]  0.572  0.694  0.306  0.715
[39,]  0.361  0.083  0.694  0.306
[40,]  0.281  0.065  0.083  0.694
[41,]  0.308  0.776  0.065  0.083
[42,]  0.413  0.397  0.776  0.065
[43,]  0.162 -0.686  0.397  0.776
[44,] -0.371 -0.824 -0.686  0.397
[45,] -0.745 -0.725 -0.824 -0.686
[46,]  0.026  1.627 -0.725 -0.824
[47,]  0.733  1.298  1.627 -0.725
[48,]  0.424 -1.653  1.298  1.627
[49,] -0.927 -2.427 -1.653  1.298
[50,] -1.123  0.710 -2.427 -1.653
```

```
## plot
#par(mfrow = 2:1)
plot(1:n, w_t, type = "b", lwd = 2, pch = 16, col = "darkgrey")
points(1:n, v_t, type = "b", lwd = 2, pch = 17, col = "blueviolet")
```

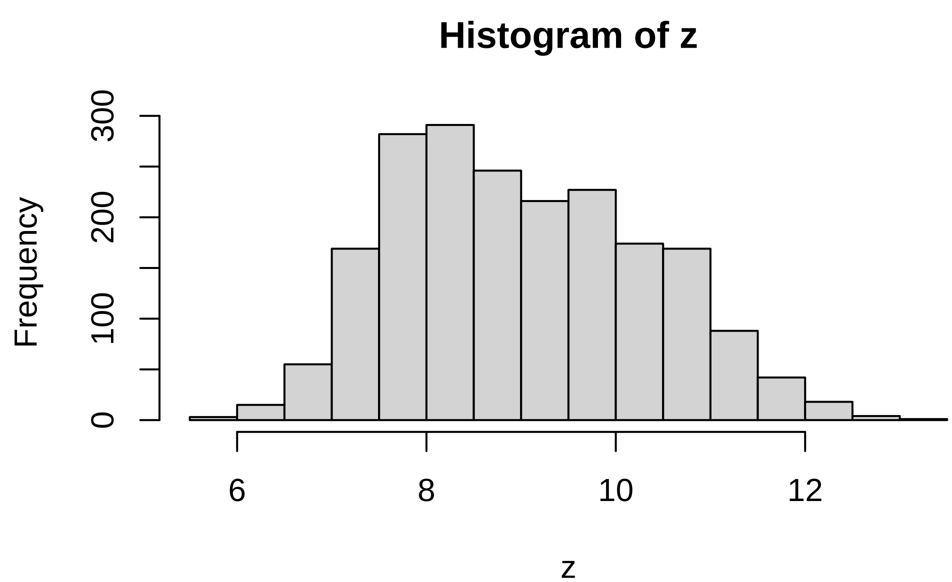

w_t
1
0
−1
−2
0
10
20
30
40
50
1:n

8 Chapter 2: Correlation and Stationary Time Series

8.1 Motivation

How do we summarize characteristics of a distribution?

```
set.seed(2024)
x <- rnorm(1000, 10, 1)
y <- rnorm(1000, 8, .75)
z <- c(x,y)
hist(z)
```

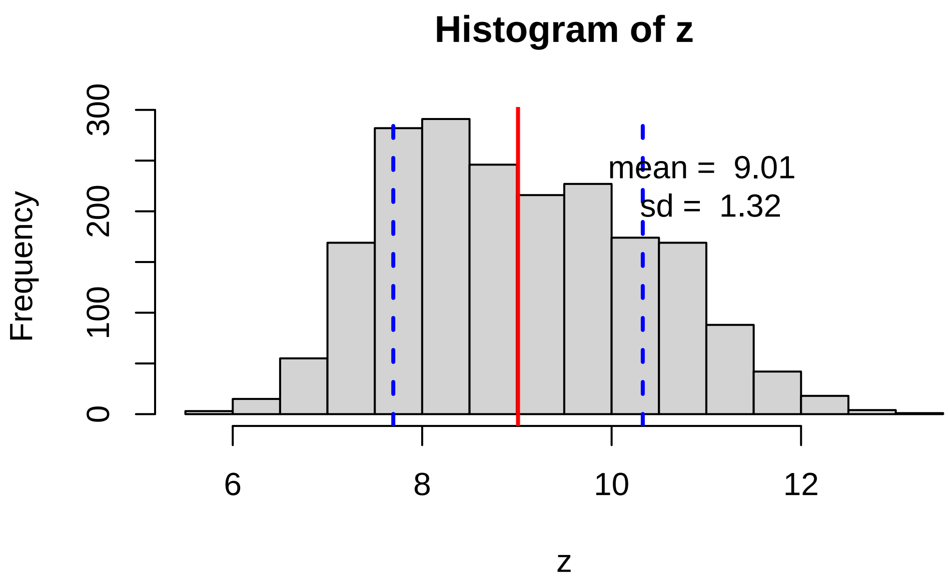


8.2 Motivation

How do we summarize characteristics of a distribution?

- mean
- variance(standard deviation)

```
hist(z)
abline(v = mean(z), col = "red", lwd = 2)
abline(v = mean(z) + sd(z), col = "blue", lty = 2, lwd = 2)
abline(v = mean(z) - sd(z), col = "blue", lty = 2, lwd = 2)
text(x = 11, y = 225,
     labels = paste("mean = ", round(mean(z),2),
                    "\n sd = ", round(sd(z),2)))
```



8.3 How do we summarize the characteristics of a distribution that changes over time?

- mean function (of time)
- (auto)(co)variance function (of time)