Homework 3: Time Series Analysis Homework 3

Time Series Analysis (STAT 6391)

Willliam Ofosu Agyapong*

September 21, 2023

*woagyapong@miners.utep.edu, PhD Data Science, University of Texas at El Paso (UTEP).

William Ofori Agyapong
STAT 6391
Homework 3

Problem 2.10

Suppose

$$x_t = \mu + w_t + \theta w_{t-1}$$

where $w_t \sim wn(0, \sigma_w^2)$.

Part (a)

Want to show that mean function is $E(x_t) = \mu$

Proof:

The mean function of x_t is computed as

$$E(x_t) = E[\mu + w_t + \theta w]$$

$$= \mu + E(w_t) + \theta E(w_{t-1})$$

$$= \mu + 0 + 0$$

$$= \mu,$$

as required.

▨

Part (b)

Want to show that the autocovariance function of x_t is given by

$$\gamma_x(h) = \begin{cases} \sigma_w^2(1+\theta^2) & h=0 \\ \sigma_w^2\theta & h=\pm \\ 0 & \text{otherwise} \end{cases}$$

Proof:

Given $x_t = \mu + w_t + \theta w_{t-1}$.

By definition, the autocovariance function $\gamma_x(h)$ is given by

$$\gamma_x(h) = \operatorname{cov}(x_{t+h}, x_t)$$
$$= \operatorname{cov}(w_{t+h} + \theta w_{t+h-1}, w_t + \theta w_{t-1}),$$

which implies that

$$\gamma_x(0) = \operatorname{cov}(w_t + \theta w_{t-1}, w_t + \theta w_{t-1})$$
$$= \operatorname{cov}(w_t, w_t) + \theta^2 \operatorname{cov}(w_{t-1}, w_{t-1})$$
$$= \sigma_w^2 + \theta^2\sigma_w^2 = \sigma_w^2(1+\theta^2)$$

$$\gamma_x(1) = \operatorname{cov}(w_{t+1} + \theta w_t, w_t + \theta w_{t-1})$$
$$= \theta \operatorname{cov}(w_t, w_t) = \theta\sigma_w^2$$
$$= \gamma_x(-1)$$ (By symmetry of the covariance function)

$$\gamma_x(2) = \operatorname{cov}(w_{t+2} + \theta w_{t+1}, w_t + \theta w_{t-1})$$
$$= 0 = \gamma_x(-2),$$ since w_t and w_k are uncorrelated for $t \neq k$.

and $\gamma_x(h) = 0 \ \forall |h|>1$

Thus,

$$\gamma_x(h) = \begin{cases} \sigma_w^2(1+\theta^2) & h=0 \\ \sigma_w^2\theta & h=\pm 1 \\ 0 & |h|>1, \end{cases}$$ as required. ▨

Part c

We wish to show that x_t is stationary for all values of $\theta \in \mathbb{R}$.

Proof:

From Part (b), we know that the mean function of x_t is

$$E(x_t) = \mu,$$

and the autocovariance function of x_t is

$$\gamma_x(h) = \begin{cases} \sigma_w^2(1+\theta^2) & h = 0 \\ \sigma_w^2 \theta & h = \pm 1 \\ 0 & |h| > 1 \end{cases},$$

which are independent of time t $\forall \theta \in \mathbb{R}$.

Therefore it follows from the stationary series definition that x_t is stationary.

$\blacksquare$

Part (d)

According to 2.20,

$$\operatorname{Var}(\bar{x}) = \frac{1}{n}\sum_{h=-n}^{n}\left(1-\frac{|h|}{n}\right)\gamma_x(h)$$

Noting that all terms associated with $|h|>1$ in the summation go to zero since $\gamma_x(h)=0$ for all $|h|>1$, we have

$$\operatorname{Var}(\bar{x}) = \frac{1}{n}\left[(1)\sigma_w^2(1+\theta^2) + 2\left(1-\frac{1}{n}\right)\sigma_w^2\theta\right] \quad \forall \theta \in \mathbb{R}$$

— (*)

(i) when $\theta = 1$, (*) becomes

$$\operatorname{Var}(\bar{x}) = \frac{1}{n}\left[\sigma_w^2(1+1) + 2\left(1-\frac{1}{n}\right)\sigma_w^2(1)\right]$$

$$= \frac{1}{n}\left[2\sigma_w^2 + 2\sigma_w^2\left(\frac{n-1}{n}\right)\right]$$

$$= \frac{2\sigma_w^2}{n}\left[1 + \frac{n-1}{n}\right]$$

(ii) when $\theta = 0$, (*) becomes

$$\operatorname{Var}(\bar{x}) = \frac{1}{n}\left[\sigma_w^2(1+0) + 2\left(1-\frac{1}{n}\right)\sigma_w^2(0)\right]$$

$$= \frac{1}{n}\left(\sigma_w^2\right) = \frac{\sigma_w^2}{n}$$

(iii) when $\theta = -1$, we have

$$\operatorname{Var}(\bar{x}) = \frac{1}{n}\left[2\sigma_w^2 - 2\left(1-\frac{1}{n}\right)\sigma_w^2\right]$$

$$= \frac{2\sigma_w^2}{n}\left[1 - \left(\frac{n-1}{n}\right)\right]$$

▨

Part (e)

Noting that $\frac{n-1}{n} \approx 1$ for large n, the results in part (d) becomes

$$\mathrm{Var}(\bar{x}) = \begin{cases} \frac{2\sigma_w^2}{n}(1+1) = \frac{4\sigma_w^2}{n} & \theta = 1 \\ \frac{\sigma_w^2}{n} & \theta = 0 \\ \frac{2\sigma_w^2}{n}(1-1) = 0 & \theta = -1 \end{cases}$$

$$0 < \frac{\sigma_w^2}{n} < \frac{4\sigma_w^2}{n}$$

We see from the above results that the accuracy of the estimate of the mean improves with decreasing values of θ. In fact, for large n, $\bar{x}$ becomes a much better estimator of the mean when $\theta = -1$ (variance goes to zero).

▨

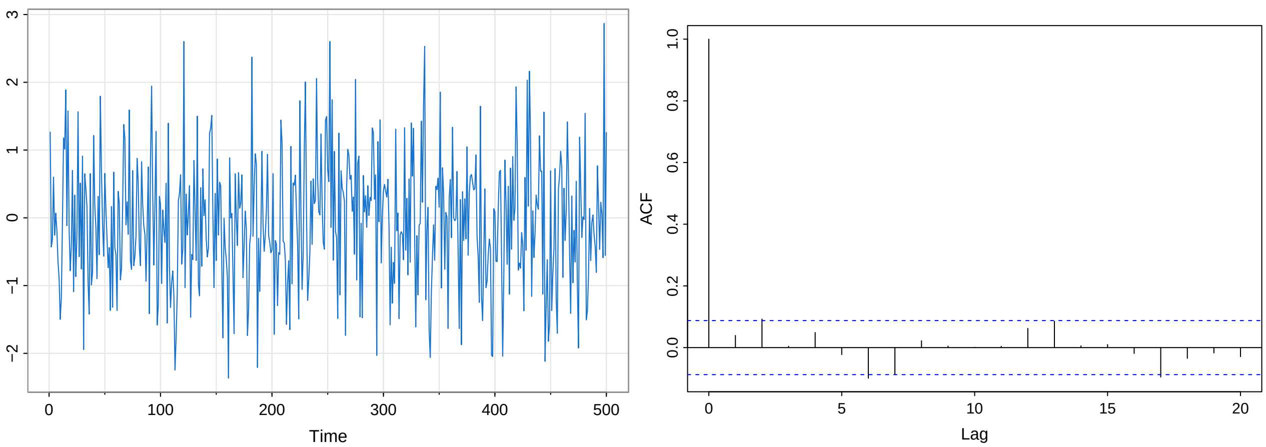
Problem 2.11

Part (a)

In this problem, we simulated 500 Gaussian white noise observations and computed the sample ACP to lag 20 which are shown in Figure 1. For a white noise process, we know that the theoretical ACF is given by

$$\rho_x(h) = \begin{cases} 1 & h = 0 \\ 0 & h \neq 0 \end{cases} .$$

From plot (b) in Figure 1, it is clear that the sample ACF from the simulated white noise process is approximately about the same as the theoretical ACF as most of the sample ACF values are approximately zero for $h \neq 0$ and of course 1 at lag 0.



(a) White noise process for n=500.

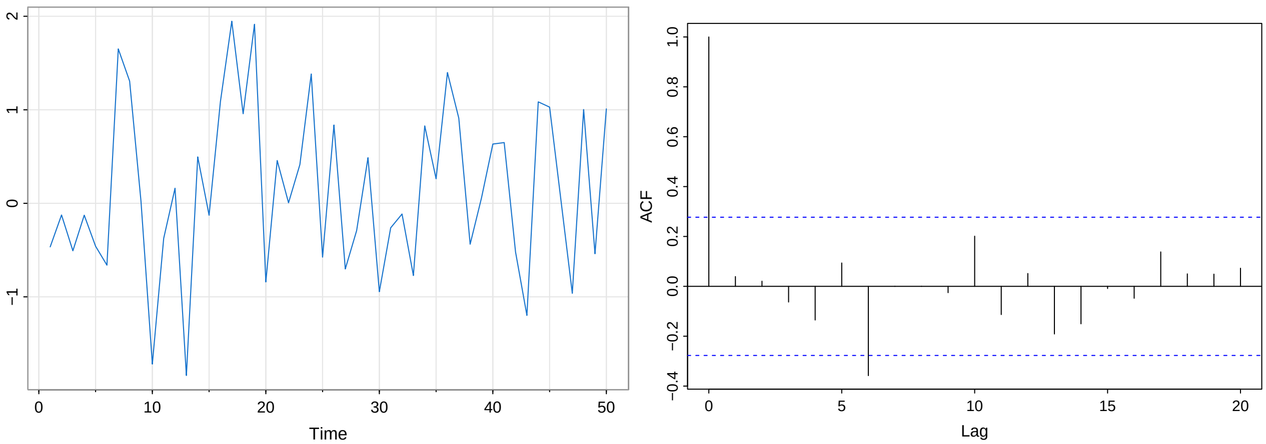
(b) Sample autocorrelation function up to lag 20.

Figure 1: Sample autocorrelation function of a Gaussian white noise process up to lag 20. The white noise process is in the left panel.

Part (b)

Now, we repeat what we did in part (a) using only $n = 50$ this time.

From plot (a) in Figure **??**, we can observe that decreasing the sample size (n) to 50 resulted in increased variability in the sample ACF.



(a) White noise process for n=50.

(b) Sample autocorrelation function up to lag 20.

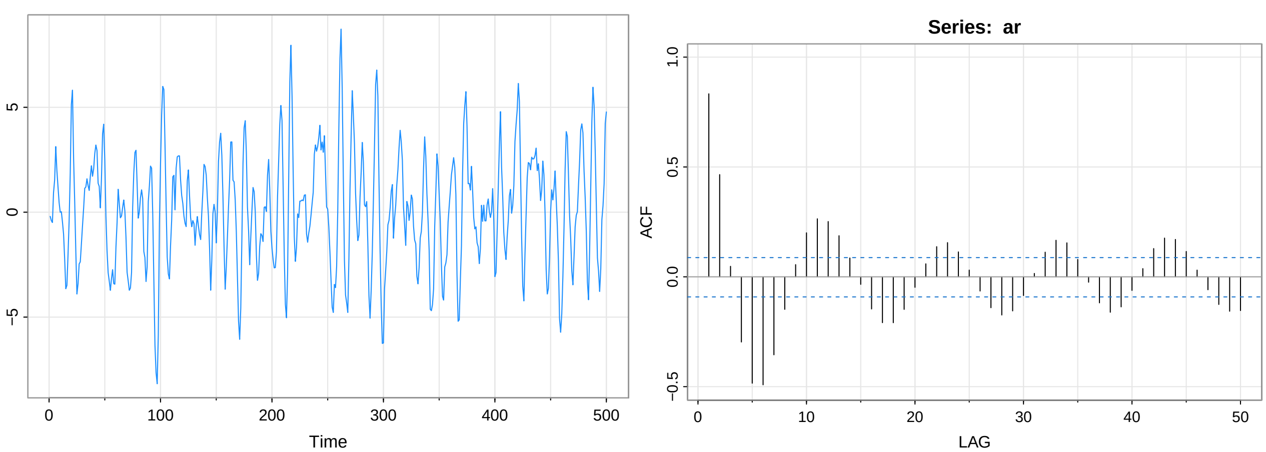
Figure 2: Sample autocorrelation function of a Gaussian white noise process up to lag 20. The white noise process is in the left panel.

Problem 2.13

We simulated a series of $n = 500$ moving average observations based on the AR model below:

$$x_t = 1.5x_{t-1} - .75x_{t-2} + w_t.$$

The simulated series and its corresponding ACF to lag 50 are shown in Figure 3. The sample ACF plot is somewhat periodic which reveals the approximate cyclical behavior of the data.



(a) 500 observations from AR(2) model.

(b) Sample autocorrelation function up to lag 50.

Figure 3: A simulated AR(2) process with autocorrelation function up to lag 50.

Problem 3.1

Part (a)

In this problem, we fitted the regression model

$$x_t = \beta t + \alpha_1 Q_1(t) + \alpha_2 Q_2(t) + \alpha_3 Q_3(t) + \alpha_4 Q_4(t) + w_t$$

where $Q_i(t) = 1$ if time t corresponds to quarter $i = 1, 2, 3, 4$, and zero otherwise. w_t is assumed to be a Gaussian white noise sequence.

Results from the fitted model are provided in Tables 1 and 2.

term	estimate	std.error	t statistic	95% CI	p.value
trend	0.167	0.002	73.999	[0.163, 0.172]	<0.001
Q1	1.053	0.027	38.480	[0.998, 1.107]	<0.001
Q2	1.081	0.027	39.500	[1.026, 1.135]	<0.001
Q3	1.151	0.027	42.035	[1.097, 1.206]	<0.001
Q4	0.882	0.027	32.186	[0.828, 0.937]	<0.001

Table 1: Structural Regression Model estimates for the logged Johnson and Johnson data.

R^2	Adjusted R^2	Residual std.error	F stat	df	df residual	p-value
0.993	0.993	0.125	2406.67	5	79	<0.001

Table 2: Overall model performance statistics.

Part (b)

According to Table 1 in part (a), if the model is correct, then the estimated average annual increase in the logged earnings per share is about 0.167 (as measured by the coefficient of the trend component).

Part (c)

From Table 1 in part (a), the estimated coefficients associated with the third and fourth quarters are **1.151** and **0.882**, respectively, suggesting that the average logged earnings rate decreased from the third quarter to the fourth quarter by approximately 23.35%.

Part (d)

When we included intercept term in the model, the intercept term absorbed the first quarter term as we see that the estimated intercept coefficient in Table 3 is exactly the same as the estimated coefficient for the first quarter in Table 1 in part (a). We also note that the second quarter term was rendered insignificant (p-value > .05) which nullifies the effect of the second quarter and makes it impossible to predict movements over that period.

term	estimate	std.error	t statistic	95% CI	p.value
(Intercept)	1.053	0.027	38.480	[0.998, 1.107]	<0.001
trend	0.167	0.002	73.999	[0.163, 0.172]	<0.001
Q2	0.028	0.039	0.727	[-0.049, 0.105]	0.5
Q3	0.098	0.039	2.538	[0.021, 0.175]	0.013
Q4	-0.171	0.039	-4.403	[-0.248, -0.093]	<0.001

Table 3: Structural Regression Model estimates for the logged Johnson and Johnson data with an intercept term.

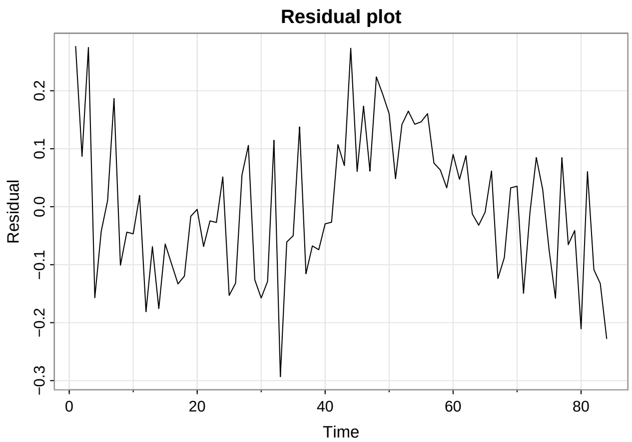
R^2	Adjusted R^2	Residual std.error	F stat	df	df residual	p-value
0.986	0.985	0.125	1378.755	4	79	<0.001

Table 4: Overall model performance statistics from the model with intercept term.

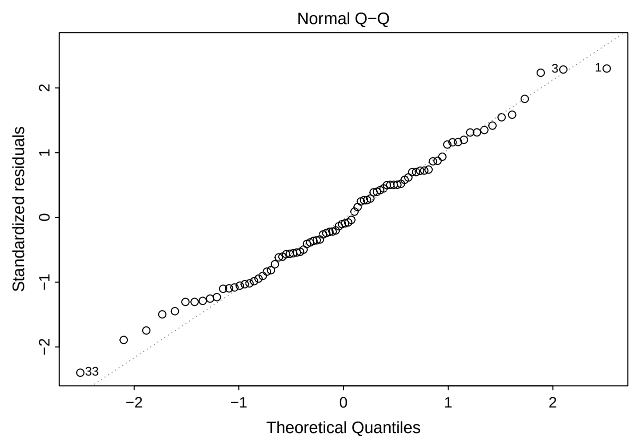
Part (e)

Figure 4 presents a plot of the logged Johnson & Johnson data (blue) with the regression fitted values (red) along with two other plots of the residuals.

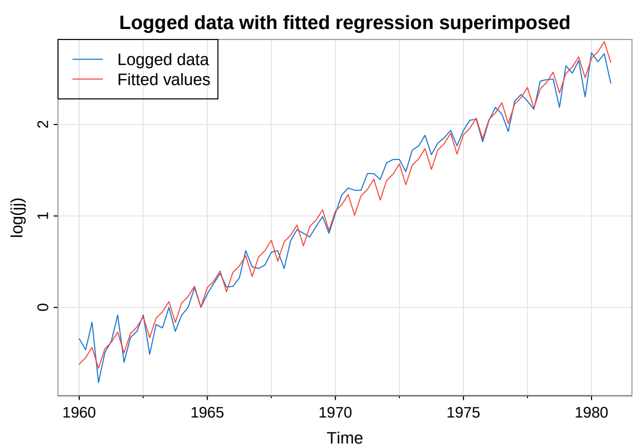
We used the normal QQ plot and the residuals plot to examine the residuals. According to plot (b), the distribution of the residuals appear Gaussian. However, the residuals plot shows clear increasing and decreasing pattern, a sign of non-constant residual variance which contradicts the assumption that w_t is white. Therefore, the residuals do not look white and thus the model appears does not fit the data well. We see from plot (c) that the model over fits the data.



(a)



(b)



(c)

Figure 4: Residual diagnostic plots and logged data with fitted regression line superimposed.

Problem 3.2

Part (a)

We added another component P_{t-4} to the regression in (3.17) of the text that accounts for the particulate count four weeks prior. The fitted regression results are presented in Tables 5 and 6. This model accounts for **60.8%** of the total variability in the weekly mortality.

From Table 5, the lagged term (`part4`) for the particulate count four weeks prior is significant ($p<.001$), confirming the observation that mortality peaks a few weeks after pollution peaks.

term	estimate	std.error	t statistic	95% CI	p.value
(Intercept)	2808.331	198.852	14.123	[2417.639, 3199.023]	<0.001
trend	-1.385	0.101	-13.765	[-1.583, -1.188]	<0.001
temp	-0.406	0.035	-11.503	[-0.475, -0.336]	<0.001
temp2	0.022	0.003	7.688	[0.016, 0.027]	<0.001
part	0.203	0.023	8.954	[0.158, 0.247]	<0.001
part4	0.103	0.025	4.147	[0.054, 0.152]	<0.001

Table 5: Regression estimates for the Polulution, Temperature and Mortality data with a lagged variable.

R^2	Adjusted R^2	Residual std.error	F stat	df	df residual	p-value
0.608	0.604	6.287	154.503	5	498	<0.001

Table 6: Overall performance statistics for the model.

Part (b)

Table 7 presents AIC and BIC values computed from both models for comparison. We notice that both AIC and BIC decreased for the new model with the lagged variable, thus showing an improvement over the final model in Example 3.5 of Shumway et al. (2019).

	AIC	BIC
Model (3.7)	4.72	4.77
Model (3.7) with lagged variable	4.69	4.75

Table 7: AIC and BIC for the original and modified model.

Problem 3.3

In this problem, we explore the difference between a random walk and a trend stationary process.

Part (a)

Figure 5 presents four generated series that are random walk with drift of length $n = 500$, with $\delta = 0.01$ and $\sigma_w = 1$. The true mean function ($\mu_t = 0.01t$) and fitted regression values ($\hat{x}_t = \hat{\beta}t$) are also shown in each plot.

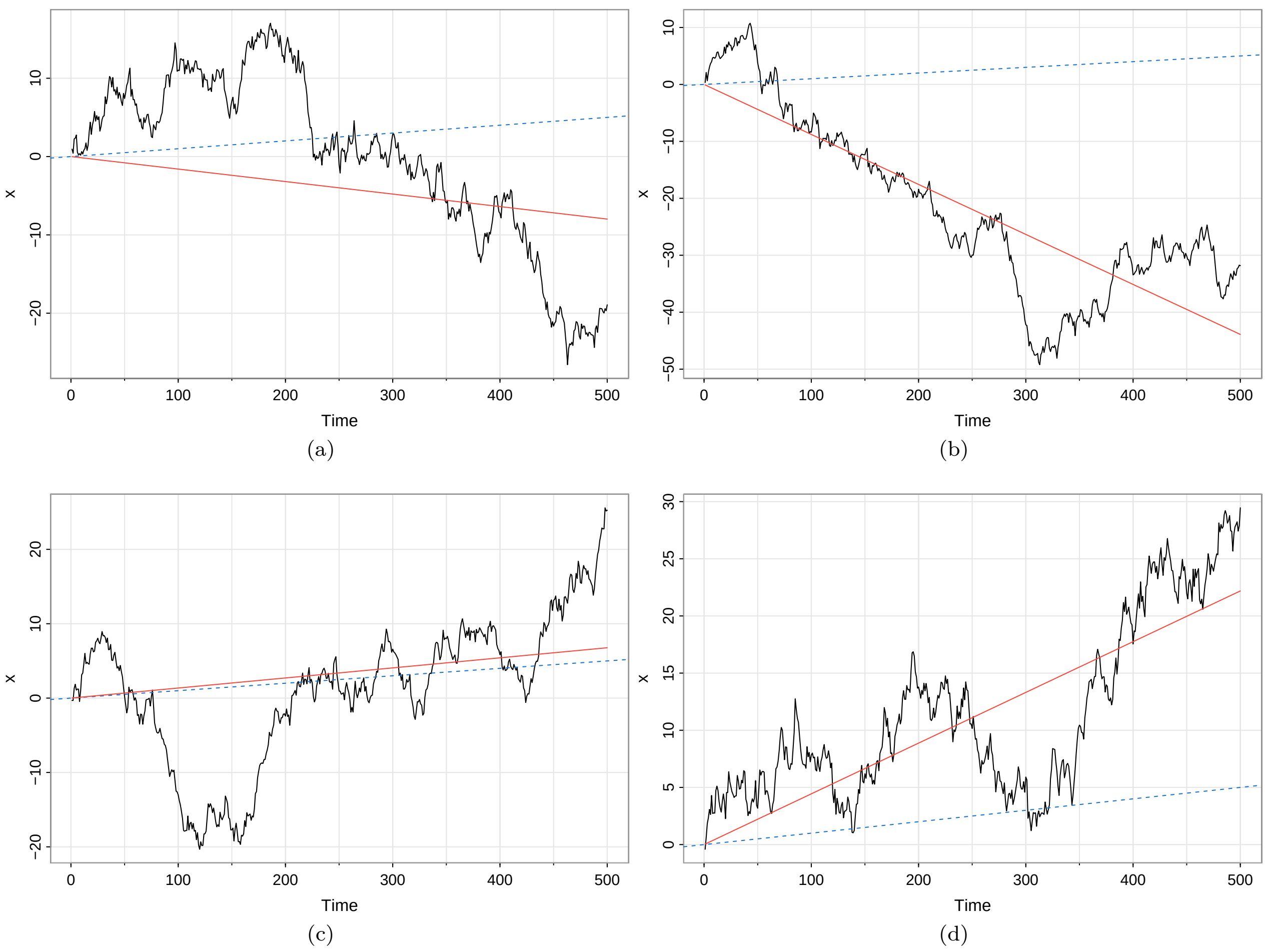


Figure 5: Four random walk models with drift ($\delta = 0.01$) and $\sigma_w = 1$, with fitted regression line (solid red) and true mean function (dotted blue) superimposed.

Part (b)

Similar to part(a), we generated another four series of length (n=500) that are linear trend plus noise, $y_t = 0.01t + w_t$ and fitted a regression model of the form $y_t = \beta + w_t$ to the resulting data. The plotted data, the true mean function, and the fitted regression line are shown in Figure 6.

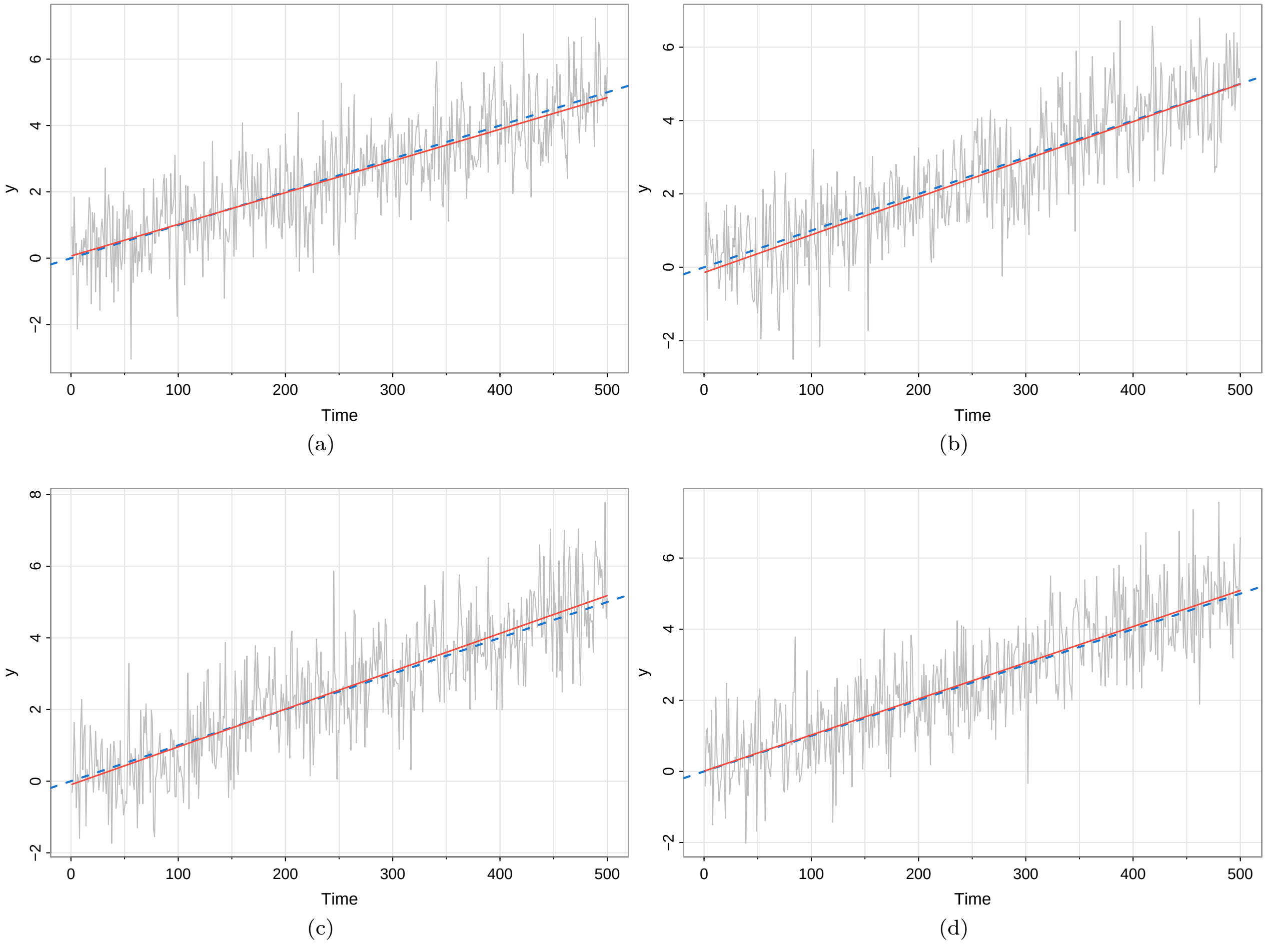


Figure 6: Four linear trend plus noise series with $\sigma_w = 1$, with regression line (solid red) and true mean function (dotted blue) superimposed.

Part (c)

Below is a comment on the differences between the results of part (a) and part (b).

There is much variation in the results of part (a), with fitted regression line deviating mostly from the true mean function. For some of the series, as the true mean line exhibited an increasing trend, the fitted line behaved in the opposite direction. In contrast, there is much stability in the results of part (b) and the behavior appears the same for all four series. Interestingly, the fitted regression lines mirror the true mean function.

References

- Robert H. Shumway, & David S. Stoffer. (2019). Time Series: A Data Analysis Approach Using R.

Appendix: R codes

```
# Set global options for output rendering
knitr::opts_chunk$set(eval = T, echo = F, warning = F, message = F,
                      fig.pos = "H", out.extra = "", fig.align = "center",
                      cache = F)

#----------------- Load required packages
# library(dplyr)
library(knitr)
library(kableExtra)
library(broom)
library(stats)
library(astsa)


#----------------- set the current working directory to the file path
setwd(dirname(rstudioapi::getSourceEditorContext()$path))



#---- Problem 2.11, part (a)

n <- 500
wn <- rnorm(n)  # generate the white noise process
tsplot(wn, col = 4, ylab = "")
sample_acf <- acf(wn, lag.max = 20, plot = T)



#---- Problem 2.11, part (b)
n <- 50
wn <- rnorm(n)  # generate the white noise process
tsplot(wn, col = 4, ylab = "")
sample_acf <- acf(wn, lag.max = 20, plot = T)
# sample_acf



#-------- Problem 2.13
set.seed(123)
wn <- rnorm(500 + 50)
ar <- filter(wn, filter = c(1.5, -.75), method = "recursive")[-(1:50)]

# par(mfrow=2:1)
tsplot(ar , col="dodgerblue", ylab = "")
sample_acf <- acf1(ar, 50)
# sample_acf
#--- Problem 3.1 (a): fitted model and model statistics
```

```r
trend <- time(jj) - 1970    # helps "center" the time
Q <- factor(cycle(jj))      # make quarterly factors
reg_mod <- lm(log(jj) ~ 0 + trend + Q, na.action = NULL) # no intercept model
# head(model.matrix(reg_mod))

# summary(reg_mod)
tidy(reg_mod, conf.int = T) |>
    dplyr::mutate(
        dplyr::across(-c(term,p.value), round, 3),
        `95% CI` = stringr::str_glue("[{conf.low}, {conf.high}]"),
        .before = "p.value",
        p.value = gtsummary::style_pvalue(p.value)
        ) |>
    dplyr::rename(`t statistic` = statistic) |>
    dplyr::select(-c(conf.low, conf.high)) |>
    kable(booktabs=T, linesep="", align = "lccccc",
          caption = "Structural Regression Model estimates for the logged Johnson and Johnson data.") |>
    kable_styling(latex_options = c("HOLD_position")) |>
    kable_classic()



#--- Problem 3.1 (a): overall model performance statistics
 glance(reg_mod) |>
        dplyr::select(r.squared, adj.r.squared, sigma, statistic, df,df.residual,  p.value) |>
        dplyr::mutate(dplyr::across(-p.value, round,3),
            p.value = gtsummary::style_pvalue(p.value)) |>
    kable(booktabs=T, linesep = "",
          caption = "Overall model performance statistics.", escape = F, align = "ccccc",
          col.names = c("$R^2$", "Adjusted $R^2$", "Residual std.error", "F stat", "df", "df residual",
        kable_styling(latex_options = c("HOLD_position", "repeat_header")) |>
    kable_classic()



# part (c): percent change
b <- coef(reg_mod)
percent_change <- abs(b[5]-b[4])*100/b[4]


# attempt to include intercept term in the model
reg_mod2 <- lm(log(jj) ~ trend + Q, na.action = NULL)
# summary(reg_mod2)

tidy(reg_mod2, conf.int = T) |>
    dplyr::mutate(
        dplyr::across(-c(term,p.value), round, 3),
        `95% CI` = stringr::str_glue("[{conf.low}, {conf.high}]"),
        .before = "p.value",
        p.value = gtsummary::style_pvalue(p.value)
        ) |>
    dplyr::rename(`t statistic` = statistic) |>
    dplyr::select(-c(conf.low, conf.high)) |>
```

```
    kable(booktabs=T, linesep="", align = "lccccc",
          caption = "Structural Regression Model estimates for the logged Johnson and Johnson data with
    kable_styling(latex_options = c("HOLD_position")) |>
    kable_classic()


#--- Problem 3.1 (a): overall model performance statistics
 glance(reg_mod2) |>
        dplyr::select(r.squared, adj.r.squared, sigma, statistic, df,df.residual,  p.value) |>
        dplyr::mutate(dplyr::across(-p.value, round,3),
            p.value = gtsummary::style_pvalue(p.value)) |>
    kable(booktabs=T, linesep = "",
          caption = "Overall model performance statistics from the model with intercept term.", escape =
          col.names = c("$R^2$", "Adjusted $R^2$", "Residual std.error", "F stat", "df", "df residual",
        kable_styling(latex_options = c("HOLD_position", "repeat_header")) |>
    kable_classic()




# examine the residuals
tsplot(ts(resid(reg_mod)), ylab = "Residual", main = "Residual plot")
plot(reg_mod, 2)    # normal qqplot

# data with fitted regeression line
tsplot(log(jj), col = 4, main = "Logged data with fitted regression superimposed")
lines(fitted(reg_mod), col=2)
legend("topleft", legend = c("Logged data", "Fitted values"), lty = 1, col = c(4,2))

#------ Problem 3.2

temp <- tempr - mean(tempr) # center temperature
temp2 <- temp^2
trend <- time(cmort) # time is trend
fit <- lm(cmort~ trend + temp + temp2 + part, na.action=NULL)


# add the lagged variable in terms of the particulate count four weeks prior
dat <- ts.intersect(cmort, trend, temp, temp2, part, part4=lag(part,-4))
new_fit <- lm(cmort~ trend + temp + temp2 + part + part4, data = dat, na.action=NULL)
# regression results
tidy(new_fit, conf.int = T) |>
    dplyr::mutate(
        dplyr::across(-c(term,p.value), round, 3),
        `95% CI` = stringr::str_glue("[{conf.low}, {conf.high}]"),
        .before = "p.value",
        p.value = gtsummary::style_pvalue(p.value)
        ) |>
    dplyr::rename(`t statistic` = statistic) |>
    dplyr::select(-c(conf.low, conf.high)) |>
    kable(booktabs=T, linesep="", align = "lccccc",
          caption = "Regression estimates for the Polulution, Temperature and Mortality data with a lag
    kable_styling(latex_options = c("HOLD_position")) |>
    kable_classic()
```

```r
#--- Problem 3.1 (a): overall model performance statistics
 glance(new_fit) |>
        dplyr::select(r.squared, adj.r.squared, sigma, statistic, df,df.residual,  p.value) |>
        dplyr::mutate(dplyr::across(-p.value, round,3),
            p.value = gtsummary::style_pvalue(p.value)) |>
    kable(booktabs=T, linesep = "",
          caption = "Overall performance statistics for the model.", escape = F, align = "ccccc",
          col.names = c("$R^2$", "Adjusted $R^2$", "Residual std.error", "F stat", "df", "df residual",
        kable_styling(latex_options = c("HOLD_position", "repeat_header")) |>
    kable_classic()



## compute the AIC and BIC

# for the original model referenced
num <- length(cmort)    # sample size
aic <- AIC(fit)/num - log(2*pi) # AIC
bic <- BIC(fit)/num - log(2*pi) # BIC

# for the new model
n <- nrow(dat)          # new sample size
new_aic <- AIC(new_fit)/n - log(2*pi)
new_bic <- BIC(new_fit)/n - log(2*pi)
result_df <- data.frame(rbind(round(c(aic, bic),2), round(c(new_aic, new_bic),2)))
rownames(result_df) <- c("Model (3.7)", "Model (3.7) with lagged variable")
 kable(result_df,booktabs=T, row.names = T, col.names = c("AIC","BIC"),
      caption = "AIC and BIC for the original and modified model.") |>
     kable_styling(latex_options = c("HOLD_position")) |>
     kable_classic()

#---- Problem 3.3 codes

set.seed(125) # seed for reproducibility of results

# repeat this process 4 times
# par(mfrow=c(2,2))
for(i in 1:4) {
    wd <- rnorm(500) + 0.01
    x <- ts(cumsum(wd))
    t <- time(x)
    # fit regression
    fit1 <- lm(x ~ 0 + t, na.action = NULL)     # with no intercept
    # plot the data
    tsplot(x)
    # add true mean function
    abline(b=0.01, a=0, lty=2, col=4)
    # add the fitted line
    lines(fitted(fit1), col=2)
}

set.seed(125) # seed for reproducibility of results
```

```r
# repeat this process 4 times
# par(mfrow=c(2,2))
for(i in 1:4) {
    w <- rnorm(500)
    y <- 0.01*t + w
    t <- seq_along(y)
    # fit regression
    fit2 <- lm(y ~ t)
    summary(fit2)
    # plot the data
    tsplot(y, col="gray")
    # add true mean function
    abline(b=0.01, a=0, lty=2, col=4, lwd=2)
    # add the fitted line
    lines(fitted(fit2), col=2, lwd=1.5)
}
```