Homework 9 - Predictive Modeling in Finance and Insurance

Dennis Goldenberg

2024-04-05

```
suppressPackageStartupMessages(library(zoo, quietly = TRUE))
```

```
## Warning: package 'zoo' was built under R version 4.3.3
```

```
library(ggplot2)
```

1. Exploratory Time Series Analysis

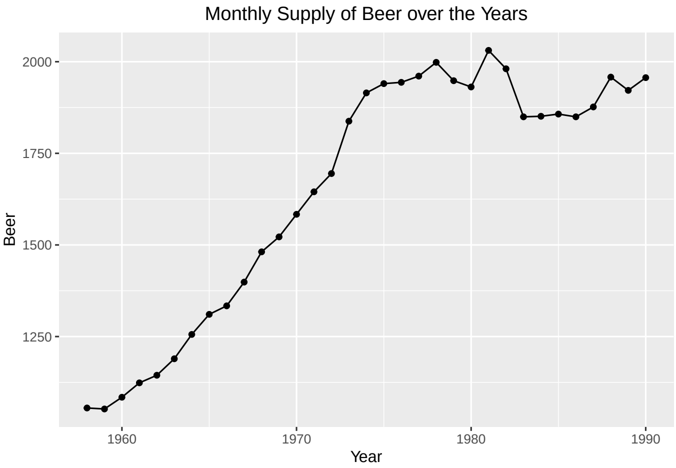
a. EDA - time plot of the data

I import the data and create a column to represent the month and year:

```
cbeDat <- read.table("cbe.dat", header = TRUE)
cbeDat$month <- as.yearmon(seq(as.Date("1958-01-01"), as.Date("1990-12-01"), by = "months"))
cbeDat <- cbeDat[,c(4,1,2,3)]
```

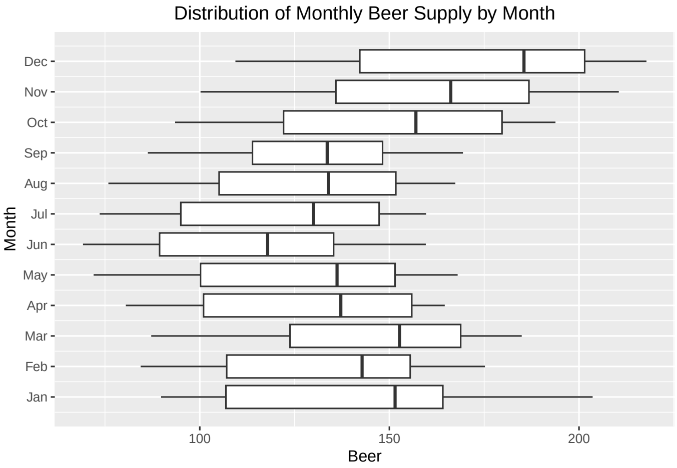
Then, I plot the aggregate annual series:

```
yearBeer <- c()
i <- 0
while(12 * i < dim(cbeDat)[1]){
  yearBeer <- append(yearBeer, sum(cbeDat$beer[(12*i + 1):((12*i)+12)]))
  i <- i + 1
}
sumDat <- as.data.frame(cbind(1958:1990, yearBeer))
colnames(sumDat) <- c("Year", "Beer")
ggplot(data = sumDat, aes(x = Year, y = Beer)) + geom_point() + geom_line() +
  ggtitle("Monthly Supply of Beer over the Years") +
  theme(plot.title = element_text(hjust = 0.5))
```



The graph shows a relatively linear increase in Beer supply between the years 1960 and 1980 and then a bit of a drop off in the 80s. I then plot the box plots by season:

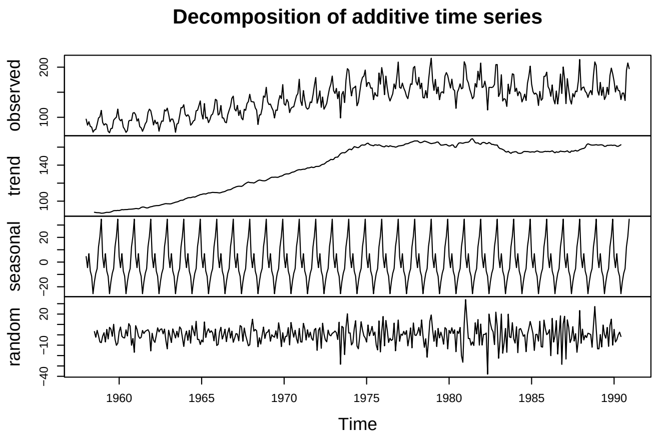
```
monthD <- as.numeric(substring(as.character(cbeDat$month), 1,3) == "Feb") +
  2 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Mar") +
  3 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Apr") +
  4 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "May") +
  5 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Jun") +
  6 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Jul") +
  7 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Aug") +
  8 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Sep") +
  9 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Oct") +
  10 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Nov") +
  11 * as.numeric(substring(as.character(cbeDat$month), 1,3) == "Dec")
monthD <- factor(monthD, levels = 0:11, labels =
  c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"))
seasonDat <- as.data.frame(cbind(monthD, cbeDat$beer))
colnames(seasonDat) <- c("Month", "Beer")
ggplot(data = seasonDat) +
  geom_boxplot(aes(x = Beer, y = Month, group = Month)) +
  scale_y_continuous(breaks = seq(1, 12), labels = levels(monthD)) +
  ggtitle("Distribution of Monthly Beer Supply by Month") +
  theme(plot.title = element_text(hjust = 0.5))
```



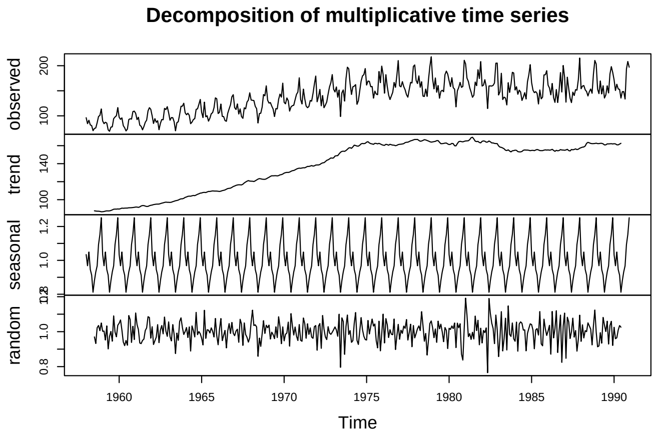
The supply of beer seems, on average, to be highest at the end of the year (October, November, December) and lowest in the summer Months, particularly in June. The Spring months have means in between the summer and the winter months, indicating some seasonality.

b. Decomposition Using two methods

```r
beer.month.ts <- ts(cbeDat$beer, start = 1958, freq = 12)
plot(decompose(beer.month.ts, type="add"))
```



```r
plot(decompose(beer.month.ts, type="mul"))
```



c. Comparison of Decomposition methods

Note that, via the observed graph, the seasonal variation tends to increase over time. This suggests that the multiplicative model is better; also, when examining the randomness, the multiplicative model seems to display more stationarity in variance than the additive model; thus the multiplicative model is likely a better fit in this case.

2. Sample Autocorrelation

The sample lag 2 autocorrelation formula is as follows:

$$\hat{\rho}_2 = \frac{\sum_{t=3}^{T}(y_t - \bar{y})(y_{t-2} - \bar{y})}{\sum_{t=1}^{T}(y_t - \bar{y})^2}$$

Note that $\bar{y} = \frac{1+1.5+1.6+1.4+1.5+1.7}{6} = 1.45$. So, I calculate, using the numbers given:

$$\hat{\rho}_2 = \frac{(1.6-1.45)(1-1.45)+(1.4-1.45)(1.5-1.45)+(1.5-1.45)(1.6-1.45)+(1.7-1.45)(1.4-1.45)}{(1-1.45)^2+(1.5-1.45)^2+(1.6-1.45)^2+(1.5-1.45)^2+(1.7-1.45)^2}$$

$$= \frac{-.075}{0.2925} = -\mathbf{0.2564}$$

3. Forecast error

Note that $y_{10} = y_0 + \sum_{i=1}^{10} c_i$. Since $\bar{c}_{10} = 2$, this means that $\sum_{i=1}^{10} c_i = 10 * \bar{c}_{10} = 10 * 2 = 20$. Therefore, $y_{10} = y_0 + 20$. From this, and the data given:

$$y_{19} = y_0 + \sum_{i=1}^{10} c_i + \sum_{i=11}^{19} c_i = y_0 + 20 + 26 = y_0 + 46$$

From the fact that all c_t values are positive, this seems to be a random walk model with drift. Therefore, to develop the estimate of $\hat{y}_{19}$ from $\{y_i : i \in [10]\}$, I estimate the drift parameter δ:

$$\hat{\delta} = \mathbb{E}\left[\nabla y\right] = \bar{c}_{10} = 2$$

Therefore, as Forecast steps $= \ell = 19 - 10 = 9$, I predict $\hat{y}_{19}$ from y_10:

$$\hat{y}_{19} = y_{10} + \ell * \hat{\delta} = y_0 + 20 + 9 * 2 = y_0 + 38$$

Thus, the forecast error is:

$$y_{19} - \hat{y}_{19} = (y_0 + 46) - (y_0 + 38) = \mathbf{8}$$

4. Forecast for AR(1) model

I am given that $\bar{y} = 8.01$. I am also given that $\hat{\alpha} = -0.79$ as the estimated parameter from the model. Note that the mean is subtracted before the model is fit; therefore:

$$\hat{y}_9 - \bar{y} = \hat{\alpha}^{9-7}\,(y_7 - \bar{y})$$

So, I solve for $\hat{y}_9$:

$$\hat{y}_9 = \bar{y} + \hat{\alpha}^{9-7}(y_7 - \bar{y}) = 8.01 + (-.79)^2(8.5 - 8.01) = \mathbf{8.315809}$$

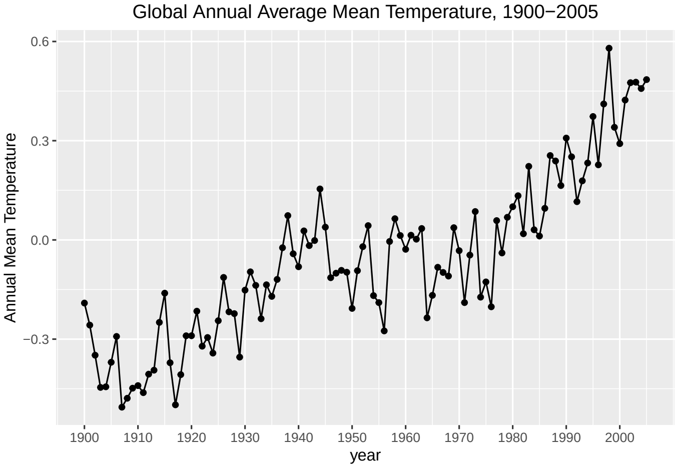
5. Time Series, AutoRegression, GLS

```
globDat <- read.table("Global.dat", header = FALSE)
colnames(globDat) <- c("Jan", "Feb", "Mar", "Apr", "Jun", "Jul",
                       "Aug", "Sep", "Oct", "Nov", "Dec")
means <- rowMeans(as.matrix(globDat))
year <- 1856:2005
dataGlob <- as.data.frame(cbind(year, means,globDat))
```

a. Annual Temperature Plot

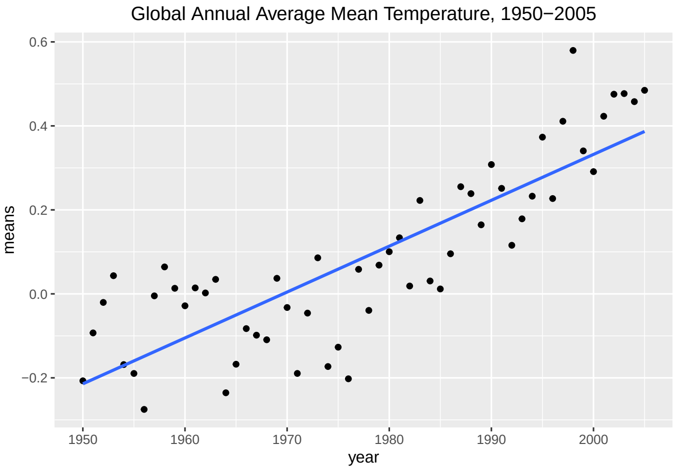
I plot the annual temperature from 1900 to 2005:

```
ggplot(data = dataGlob[(1900-1855):(2005-1855),],aes(x = year, y = means)) +
  geom_point() + geom_line() + ylab("Annual Mean Temperature") +
  ggtitle("Global Annual Average Mean Temperature, 1900-2005") +
  scale_x_continuous(breaks = seq(1900, 2005, by = 10)) +
  theme(plot.title = element_text(hjust = 0.5))
```



b. Plotting new series average annual temperature

```
ggplot(data = dataGlob[(1950-1855):(2005-1855),], aes(x = year, y = means)) +
  geom_point() + geom_smooth(method = 'lm', se = FALSE) +
  ggtitle("Global Annual Average Mean Temperature, 1950-2005") +
  scale_x_continuous(breaks = seq(1900, 2005, by = 10)) +
  theme(plot.title = element_text(hjust = 0.5))
```



There seems to be a clear upward trajectory in mean average global temperature as the years pass.

c. Fitting Regression model with time component

I fit the desired linear regression, setting t as the number of years since 1950:

```
dataReg <- dataGlob[(1950-1855):(2005-1855),]
dataReg$year <- dataReg$year - 1950
regTime <- lm("means ~ year", data = dataReg)
regTime$coefficients
```

```
## (Intercept)        year
## -0.21429135  0.01092775
```

I get the 95% confidence interval using the coefficients:

```
stdErrors <- summary(regTime)$coefficients[,2]
CI <- cbind(regTime$coefficients - 1.96 * stdErrors,
      regTime$coefficients + 1.96 * stdErrors)
colnames(CI) <- c("Lower CI", "Upper CI")
CI
```

```
##                 Lower CI    Upper CI
## (Intercept) -0.275657255 -0.15292545
## year         0.009003948  0.01285155
```

d. Esimating autocorrelation

I use the formula for autocorrelation of the residuals:

$$\hat{\rho}_1 = \frac{\sum_{t=2}^{T}(z_t - \bar{z})(z_{t-1} - \bar{z})}{\sum_{t=1}^{T}(z_t - \bar{z})^2}$$

```
m <- regTime$residuals
lag <- m[1:length(m) - 1] - mean(m)
curr <- m[2:length(m)] - mean(m)
numer <- (lag %*% curr)[1,1]
denom <- (t(m - mean(m)) %*% (m - mean(m)))[1,1]
rho_1 <- numer/denom
sprintf("Prediction for correlation lag 1: %.3f", rho_1)
```

```
## [1] "Prediction for correlation lag 1: 0.432"
```

e. Fitting GLS model with AR(1) residual

I run the model and get the coefficients:

```
library(nlme)
gls1 <- gls(model = means ~ year, data = dataReg, correlation = corAR1())
gls1$coefficients
```

```
## (Intercept)        year
## -0.21684900  0.01108259
```

Then, using standard error, I get the confidence intervals:

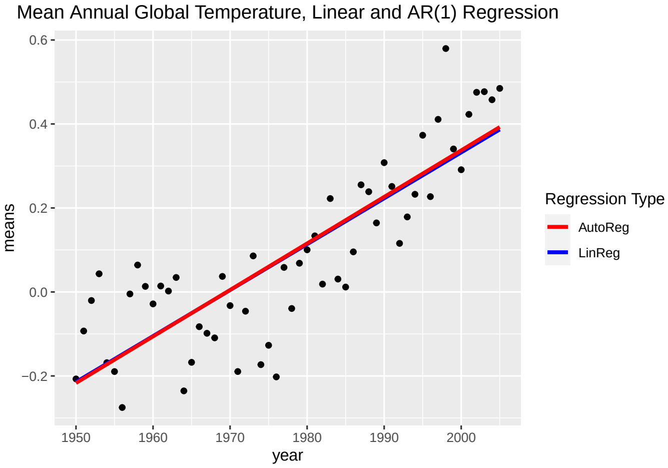
```
stdErrors2 <- summary(gls1)$tTable[,c("Std.Error")]
CI2 <- cbind(gls1$coefficients - 1.96 * stdErrors2,
      gls1$coefficients + 1.96 * stdErrors2)
colnames(CI2) <- c("Lower CI", "Upper CI")
CI2
```

```
##                 Lower CI    Upper CI
## (Intercept) -0.319665255 -0.11403274
## year         0.007887037  0.01427814
```

f. overlaying GLS fitted series to answer in b

I generate the GLS fitted series, and then overlay:

```
finData <- dataReg[,c("year", "means")]
finData$lmPred <- regTime$coefficients[1] +
  regTime$coefficients[2]*finData$year
finData$glsPred <- gls1$coefficients[1] + gls1$coefficients[2]*finData$year
ggplot(data = finData) + geom_point(aes(year, means)) +
  geom_line(aes(year, lmPred, color = 'LinReg'), linewidth = 1.2) +
  geom_line(aes(year, glsPred, color = 'AutoReg'), linewidth = 1.2) +
  scale_color_manual(name = "Regression Type", values = c("red", "blue")) +
  scale_x_continuous(breaks=seq(0,50,by = 10),labels = seq(1950,2000,by = 10))+
  ggtitle("Mean Annual Global Temperature, Linear and AR(1) Regression")+
  theme(plot.title = element_text(hjust = 0.5))
```



It is notable that the linear regression and autocorrelation produce very similar results, with the auto regression line being very slightly steeper.