Homework 4

Joshua Ingram

Problem 1

1.

a.

```
fit_gaus <- glm(visits ~ chronic + age + gender + insurance, family = "gaussian",  data = nmes)
fit_pois <- glm(visits ~ chronic + age + gender + income + insurance, family = poisson, data = nmes)
fit_zip <- zeroinfl(visits ~ chronic + age + gender + income + insurance, data = nmes)
```

Training Data

```
mse_gaus <- round(mean((predict(fit_gaus) - nmes$visits)^2),2)
mse_pois <- round(mean((predict(fit_pois, type="response") -  nmes$visits)^2),2)
mse_zip <- round(mean((predict(fit_zip) -  nmes$visits)^2),2)
data.frame(mse_gaus, mse_pois, mse_zip)
```

```
##   mse_gaus mse_pois mse_zip
## 1    42.14    42.55   42.25
```

Cross-Validation

```
mse_gaus <- round(cv.glm(nmes, fit_gaus, K = 10)$delta[1],2)
mse_pois <- round(cv.glm(nmes, fit_pois, K = 10)$delta[1],2)
mse_zip <- round(cv.glm(nmes, fit_zip, K = 10)$delta[1],2)
data.frame(mse_gaus, mse_pois, mse_zip)
```

```
##   mse_gaus mse_pois mse_zip
## 1    42.26    42.66   42.44
```

Comments

For the training data performance, the Gaussian GLM fit has the lowest mse, though the fits are very similar in values (only differing by a several tenths). The ZIP model has the next lowest MSE, followed by the regular poisson for the performance of point predictions.

For 10-fold cross-validation, the gaussian GLM fit performs the best, yet again. It's a similar story, with the MSEs only differing by several tenths. The MSE is greater for the cross-validation and ZIP has the next best performance for point predictions.

b.

Gaussian GLM

1.

```
gaus_probs <- pnorm(0, mean = predict(fit_gaus), sd =sigma(fit_gaus), lower.tail = FALSE)
```

2.

```
summary(gaus_probs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.6185  0.7480  0.8046  0.8033  0.8567  0.9861
```

Poisson GLM

1.

```
pois_probs <- ppois(-1,predict(fit_pois, type="response"), lower.tail = FALSE)
```

2.

```
summary(pois_probs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1       1       1       1       1       1
```

ZIP

1.

```
zip_probs <- ppois(-1,predict(fit_zip, type="response"), lower.tail = FALSE)
```

2.

```
summary(zip_probs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1       1       1       1       1       1
```

Comments

For both the ZIP and regular poisson models, the probabilities of having at least 0 visits is 1 for all individuals. This makes sense because poisson distributions model count data, which must be a non-negative valued integer, so no values less than 0 can occur. The Gaussian model varies, as it allows for negative values in the distribution. The range of the probabilities is greater, going from .6 to .9, as the distribution will depend upon the “mean” and stanard deviation, affecting the probability of at least 0 visits.

c.

Gaussian GLM

1.

```
gaus_probs <- pnorm(0, mean = predict(fit_gaus), sd =sigma(fit_gaus), lower.tail = FALSE) - pnorm(3, mea
```

2.

```
summary(gaus_probs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02722 0.12970 0.15052 0.14488 0.16626 0.18218
```

Poisson GLM

1.

```
pois_probs <- ppois(-1,predict(fit_pois, type="response"), lower.tail = FALSE) - ppois(4,predict(fit_poi
```

2.

```
summary(pois_probs)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000149 0.2064120 0.3711770 0.3784046 0.5345659 0.8273599
```

ZIP

1.

```
zip_probs <- ppois(-1,predict(fit_zip, type="response"), lower.tail = FALSE) - ppois(4,predict(fit_zip,
```

2.

```
summary(zip_probs)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0001462 0.1925264 0.3547051 0.3787231 0.5554838 0.9083531
```

Comments

The Gaussian model has the smallest range of the probabilities (for no more than 3 visits), going anywhere from 0.03 to .18. This differs from the ZIP and regular poisson models, with the range of this probability going from less than 0.01 to .83 for the regular poisson model and to 0.91 for the ZIP model. The reasoning for this is quite obvious, as the entire poisson distribution covers a range of values of at least 0, whereas the Gaussian distribution wil cover a range that includes negative values.

Problem 2

Australian GDP

1.

```r
head(gdp)
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 1971           4612 4651
## 1972 4645 4615 4645 4722
```

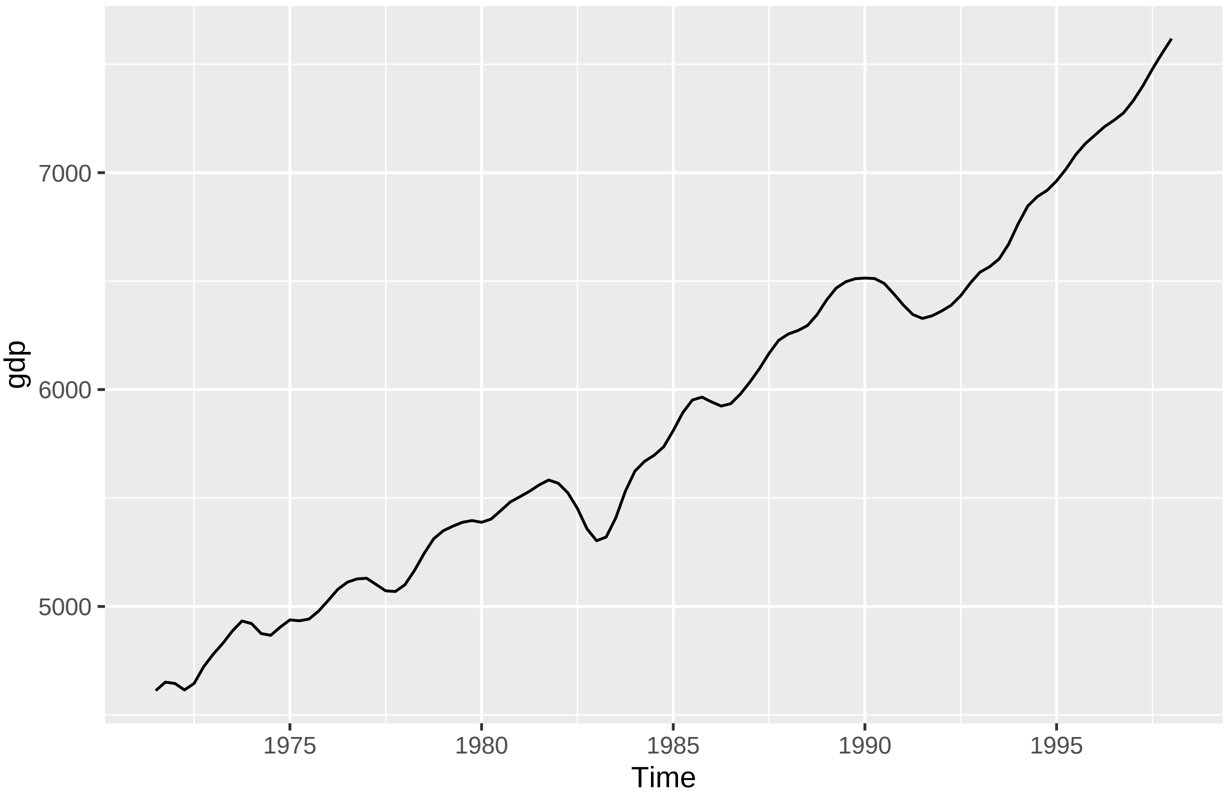
```r
frequency(gdp)
```

```
## [1] 4
```

This data set has quarterly data/frequency.

2.

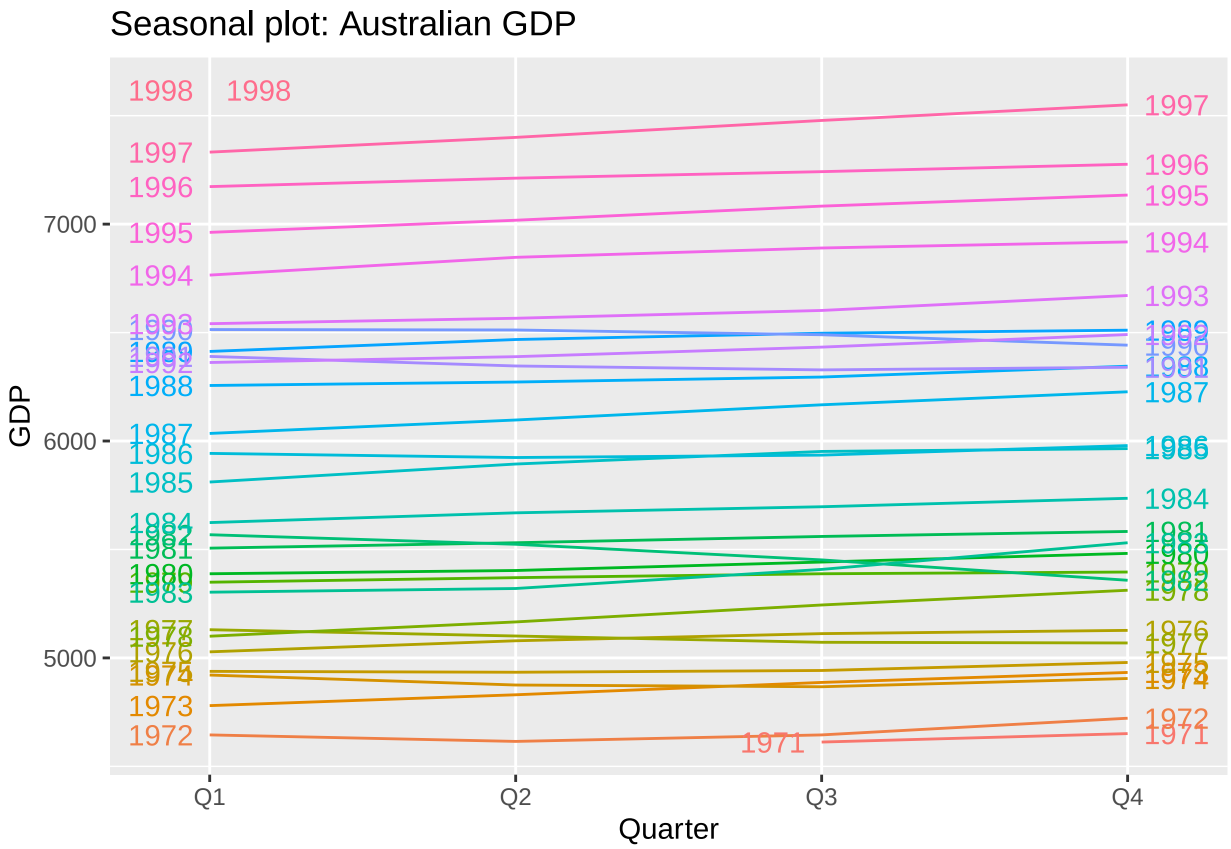
```r
autoplot(gdp)
```



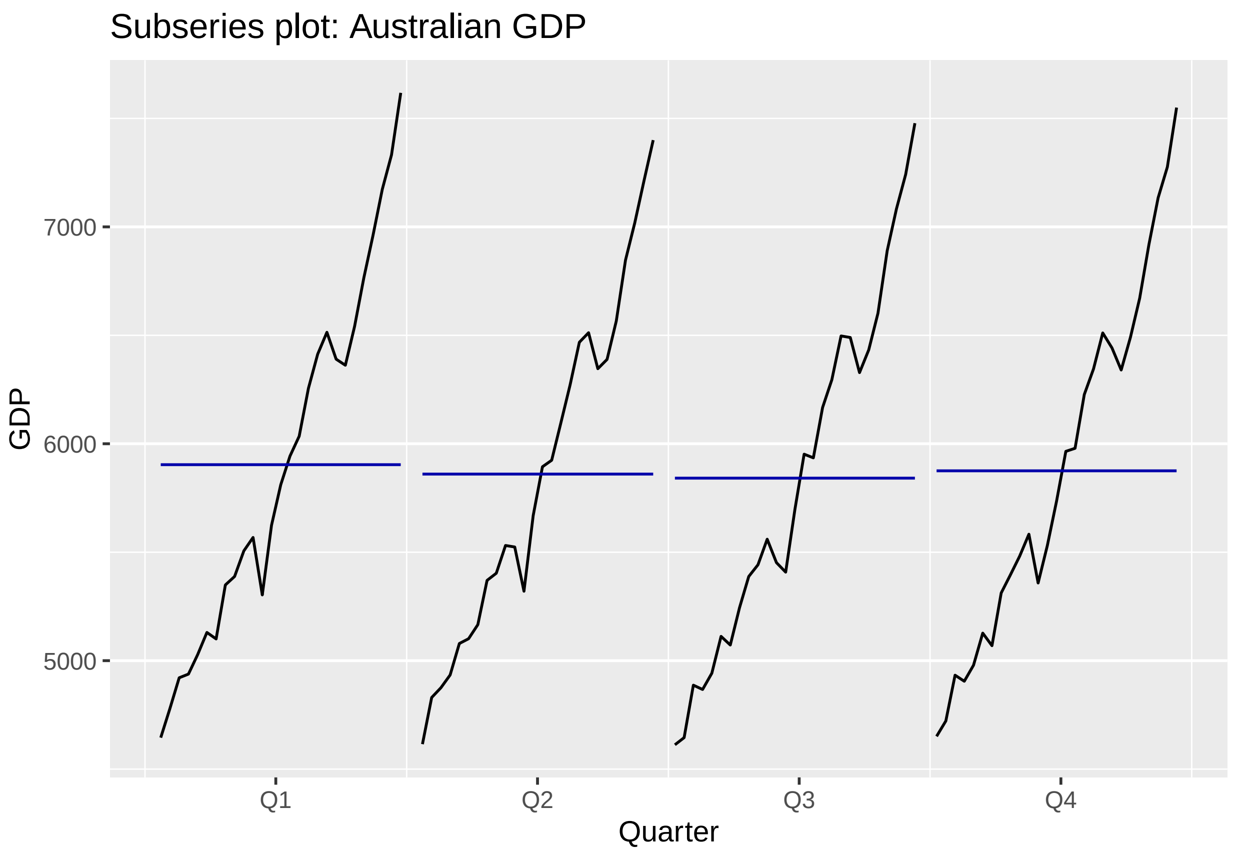
The GDP for Australia has an uptrend from 1971 and on, with some dips in the GDP (likely where recessions occurred). There does not appear to be seasonality within each year.

3.

```
ggseasonplot(gdp, year.labels=TRUE, year.labels.left=TRUE) +
  ylab("GDP") +
  ggtitle("Seasonal plot: Australian GDP")
```



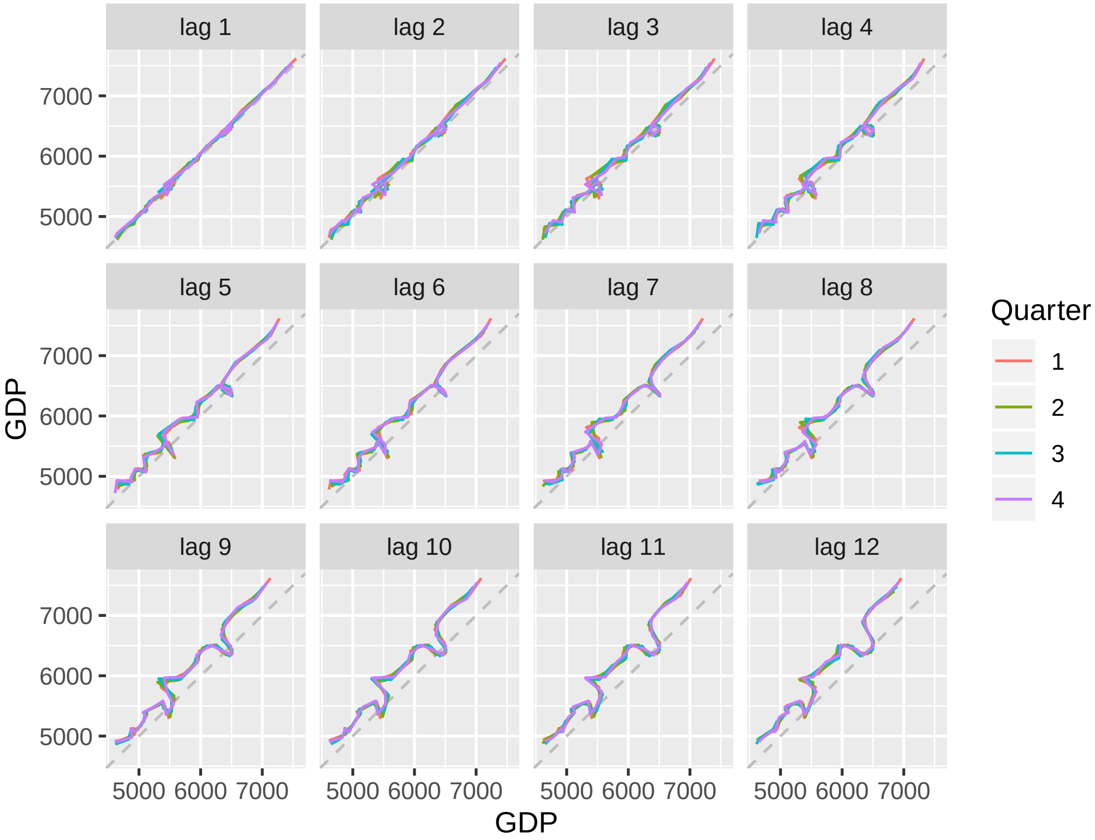
```
ggsubseriesplot(gdp) + ylab("GDP") +
  ggtitle("Subseries plot: Australian GDP")
```



The "quarter" constitutes the season in the data. There does not seem to be a pattern by season, though throughout all seasons we see an increase in GDP over time. e.g. looking at Q4 only, we see GDP increases at a similar rate as every other quarter throughout each year.

4.

```r
gglagplot(gdp, lags=12) +
  ylab("GDP") + xlab("GDP")
```



There does not appear to be any seasonality based on the lag-plots, either. Each quarter-line is basically the same as the others. As we increase the number of lags included, there is more of a pattern in the plots (e.g. more dips, different rates of increases depending on time, etc).

Australian Beer Production

1.

```
head(beer)
```

```
##      Qtr1 Qtr2 Qtr3 Qtr4
## 1956  284  213  227  308
## 1957  262  228
```

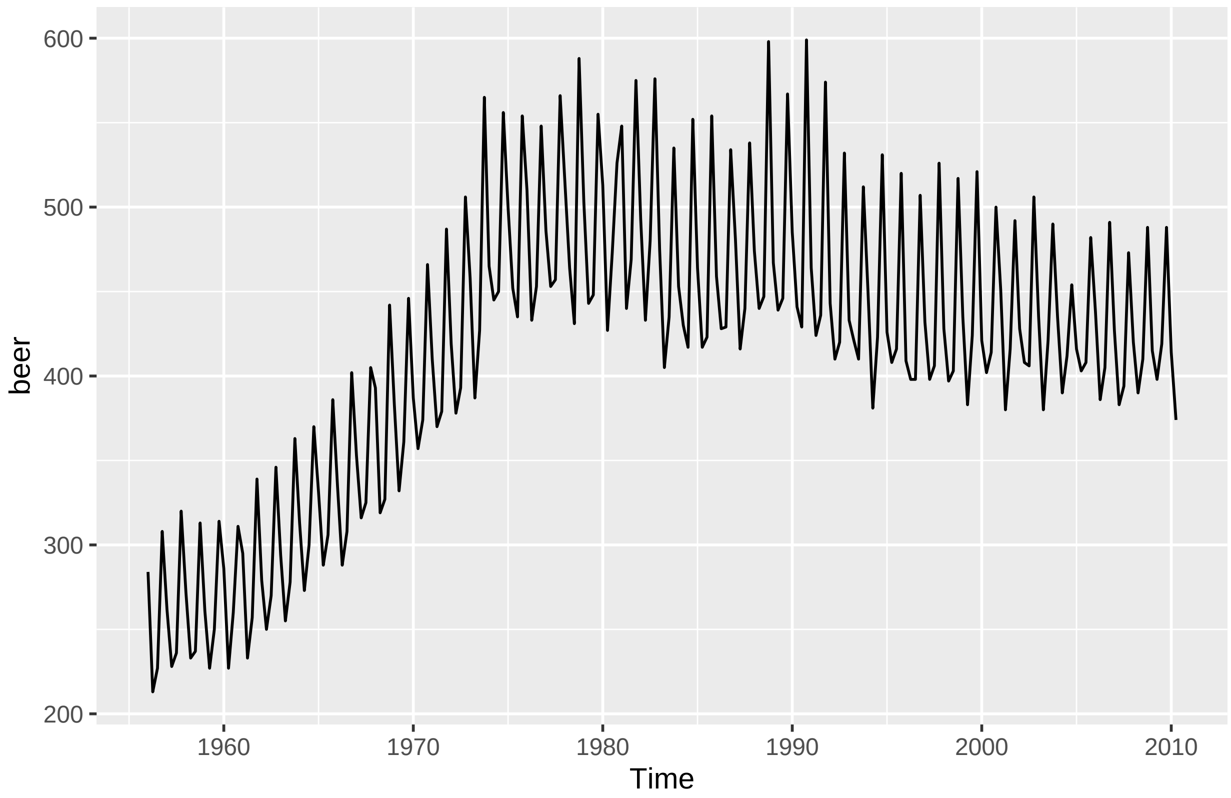
```
frequency(beer)
```

```
## [1] 4
```

This data set has quarterly data/frequency.

2.

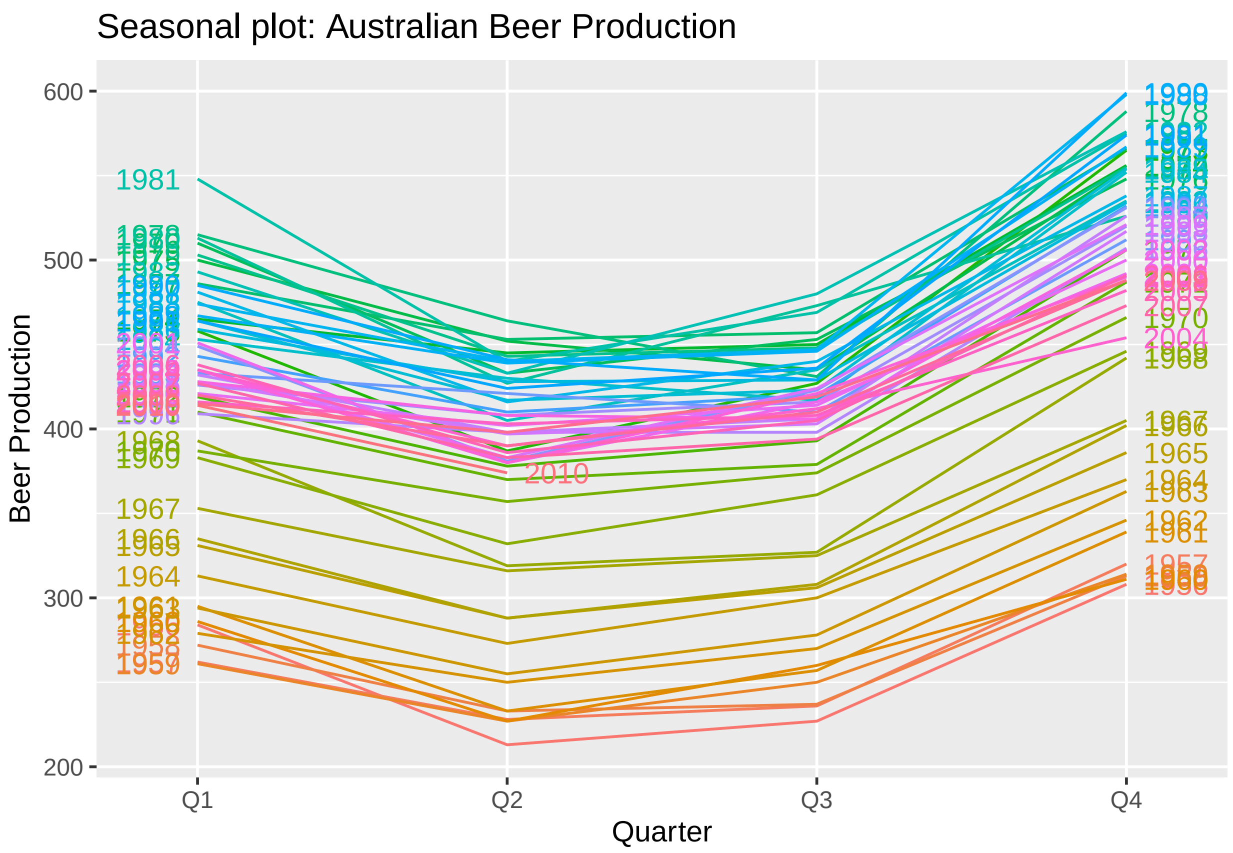
```r
autoplot(beer)
```



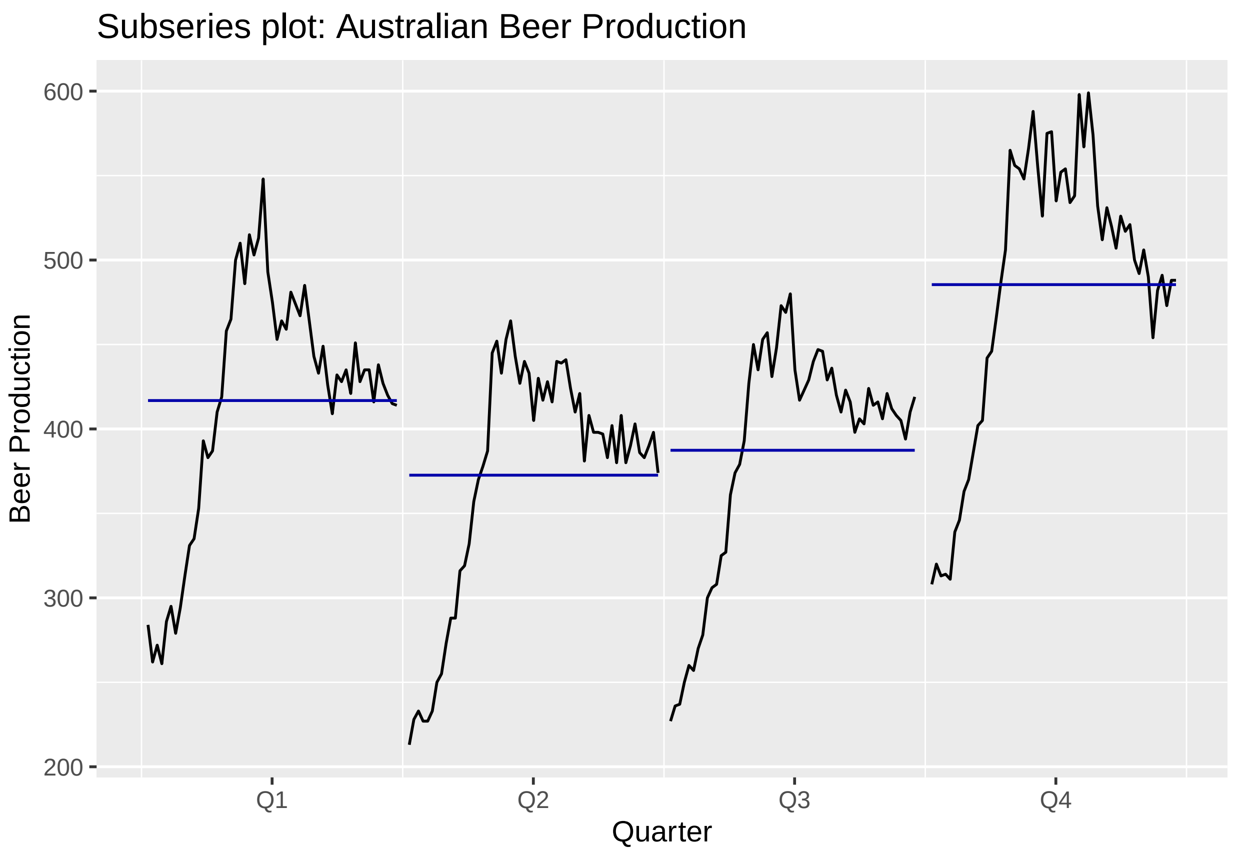
The production of beer in Australia has a clear uptrend until about 1975, then the production stagnates. In fact, it appears production might be decreasing after that point, though slighlty. There does appear to be seasonality based on the plot, which can be seen through the volatility within each year ("spikes").

3.

```r
ggseasonplot(beer, year.labels=TRUE, year.labels.left=TRUE) +
  ylab("Beer Production") +
  ggtitle("Seasonal plot: Australian Beer Production")
```



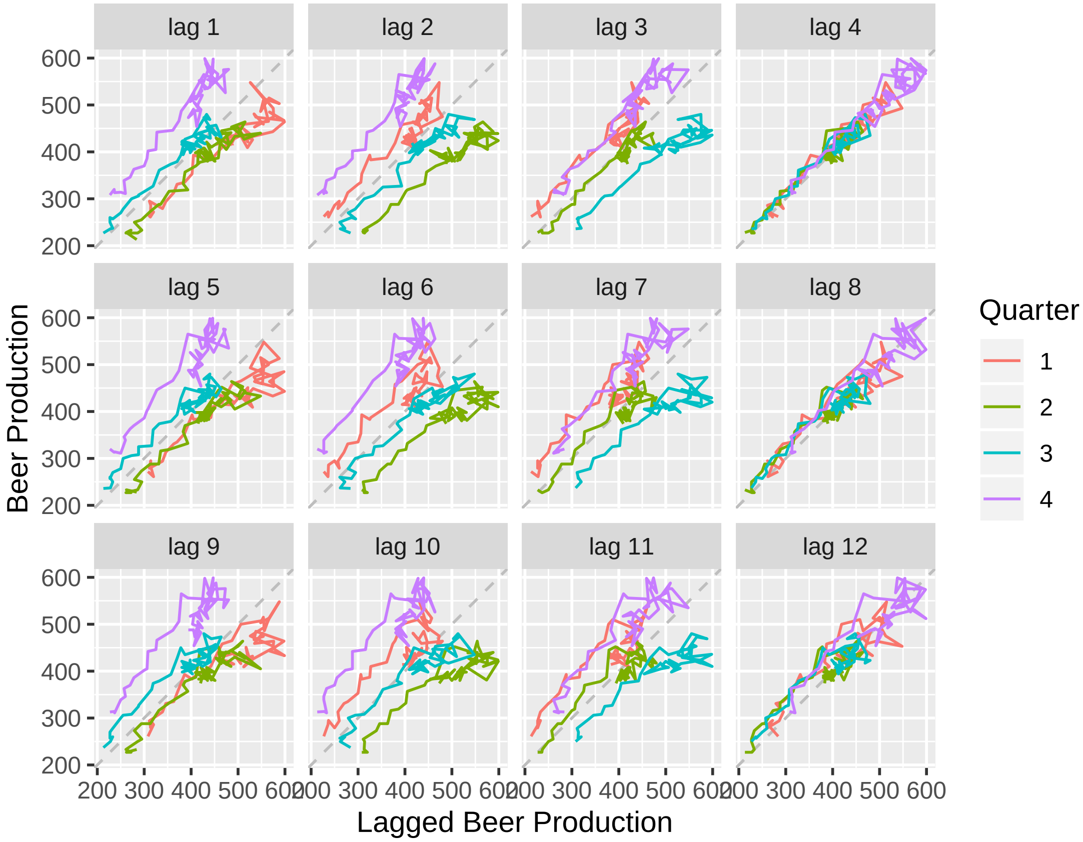
```r
ggsubseriesplot(beer) + ylab("Beer Production") +
  ggtitle("Subseries plot: Australian Beer Production")
```



The season within this data set is also the quarter. There is clearly seasonality occurring, with production decreasing each year in Q2 and then increasing again in Q4. For each season, there is a clear uptrend (likely till 1975-ish) and then there is a decrease in production throughout each season after that point. The data also becomes more volatile after the uptrend stops.

4.

```
gglagplot(beer, lags=12) +
  ylab("Beer Production") + xlab("Lagged Beer Production")
```



For lag 4, 8, and 12, there are similar trends in the quarter, with the plots being clustered closely together. The trends differ noticeably for the other lag plots. There does appear to be seasonality, which can be noticed in the lag 1-3, lag 5-7, and lag 9-11 plots.

Seasonality

Even if our data has underlying seasons, such as quarterly data, it does not mean that we observe seasonal patterns. This can be shown in the Australian GDP data, where there is an uptrend in GDP but there is not seasonal trend, even though we have quarterly data.