STATS 326/786

Code

Assignment 03

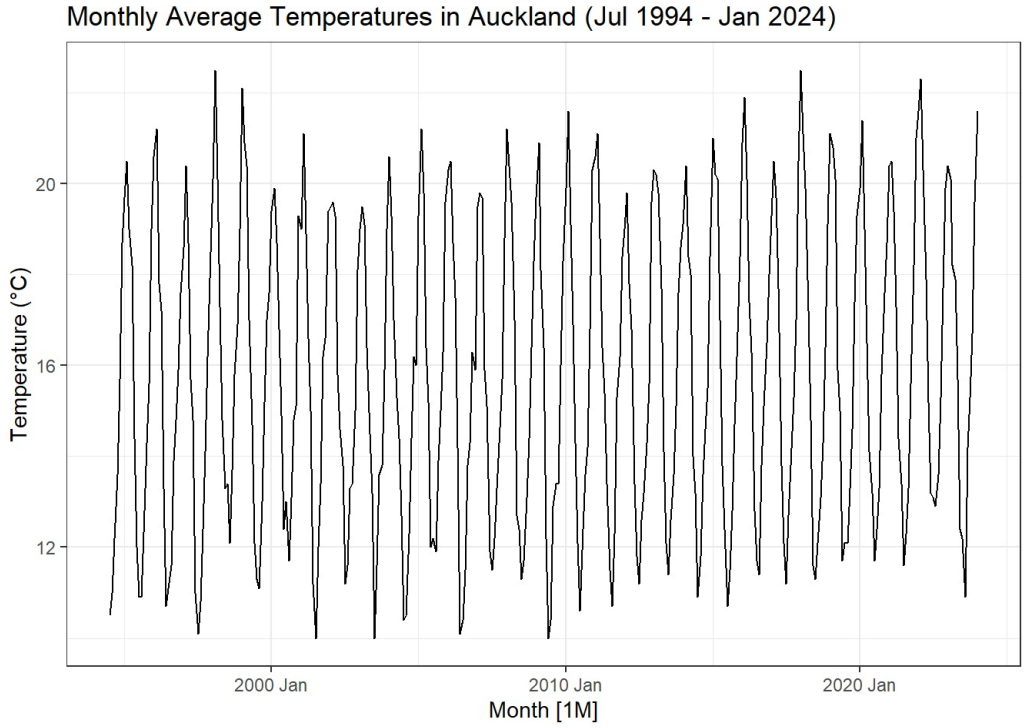
General comments:

- All the plots should be labelled appropriately (axes, legends, titles).
- Please submit both your `.Rmd` , and the generated output file `.html` or `.pdf` on Canvas before the due date/time.
- Please make sure that the `.Rmd` file compiles without any errors. The marker will not spend time fixing the bugs in your code.
- Please avoid specifying absolute paths.
- Your submission must be original, and if we recognize that you have copied answers from another student in the course, we will deduct your marks.
- You will need to use the `tidyverse` and `fpp3` libraries for this assignment.
- IMPORTANT NOTE: There are some questions that are for STATS 786 only. Students taking STATS 326, while you are welcome to attempt these questions, please do not submit answers to them.

Due: Friday 3 May 2024 at 16:00 PM (NZ time)

Problem 1: Monthly Average Auckland Temperatures

In Assignments 1 and 2, you investigated monthly average temperatures in Auckland. In this problem, you will do some further analysis. The data set `auckland_temps.csv` contains the monthly average temperatures in Auckland from July 1994 until January 2024. The time series plot is given below. For this question do not Box-Cox transform the data.

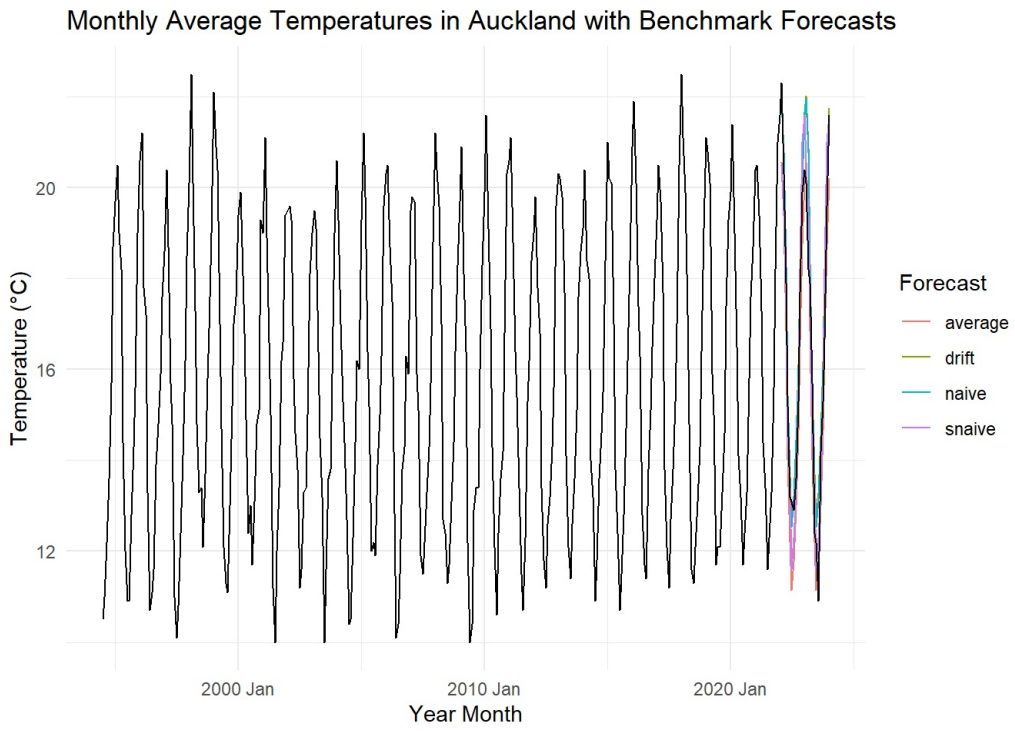


1. 12 Marks

- Recall from class that you can forecast with time series decomposition by forecasting the seasonal component and seasonally-adjusted series separately. The seasonal component will automatically be forecasted using the default `SNAIVE` method, but you will need to specify which benchmark methods you are using on the seasonally-adjusted series.
- Create a training set by filtering out the most recent 2 years from the temperature data. The training set should contain dates `1994 Jul` until `2022 Jan`.
- Fit an STL model (with `robust = TRUE`) to the training set, and using the seasonally-adjusted series from this STL decomposition, fit the naive (`NAIVE`), the average (`MEAN`) and the random walk with drift (`RW`) benchmark forecast methods. The `decomposition_model` function will be helpful. Also fit a seasonal naive (`SNAIVE`) to the training set (without doing an STL decomposition).
- Forecast 2 years up until `2024 Jan` so that you can compare what actually happened in the real data to what you predict would have happened with your forecast models.
- Plot the point-forecasts with the original data with `autoplot`. Do not plot the prediction intervals (this can be achieved with the argument `level = NULL`).
- Evaluate the point-forecasts by computing and comparing the forecast accuracy measures discussed in class.
- Discuss which forecast method is better for monthly average Auckland temperatures, and why.

Hide

```
#create training set
temp.train = data %>%
  filter(Month %in% yearmonth('1994 Jul'):yearmonth('2022 Jan'))

#Fit benchmark forecast methods to seasonally adjusted temperatures, also fit SNAIVE to training set
temp.fc <- temp.train %>%
  model(
    naive = decomposition_model(STL(Temperature, robust = TRUE), NAIVE(season_adjust)),
    average = decomposition_model(STL(Temperature, robust = TRUE), MEAN(season_adjust)),
    drift = decomposition_model(STL(Temperature, robust = TRUE), RW(season_adjust ~ drift())),
    snaive = SNAIVE(Temperature)) %>% forecast(h = "2 years")

#fit forecasts to original data
temp.fc %>%
  autoplot(data, level = NULL) +
  xlab("Year Month")+ ylab("Temperature (\u00B0C)") +
  ggtitle("Monthly Average Temperatures in Auckland with Benchmark Forecasts") +
  guides(colour=guide_legend(title = "Forecast")) +
  theme_minimal()
```



Hide

```
#retrieve forecast accuracy measures
fc.measures <- temp.fc %>%
  accuracy(data) %>%
  select(.model, MAE, MASE, RMSE, MAPE)

fc.measures
```

```
## # A tibble: 4 × 5
##   .model    MAE  MASE  RMSE  MAPE
##   <chr>   <dbl> <dbl> <dbl> <dbl>
## 1 average 0.963 1.07  1.07   6.05
## 2 drift   0.798 0.887 1.05   5.14
## 3 naive   0.735 0.817 0.990  4.74
## 4 snaive  0.85  0.944 0.976  5.32
```

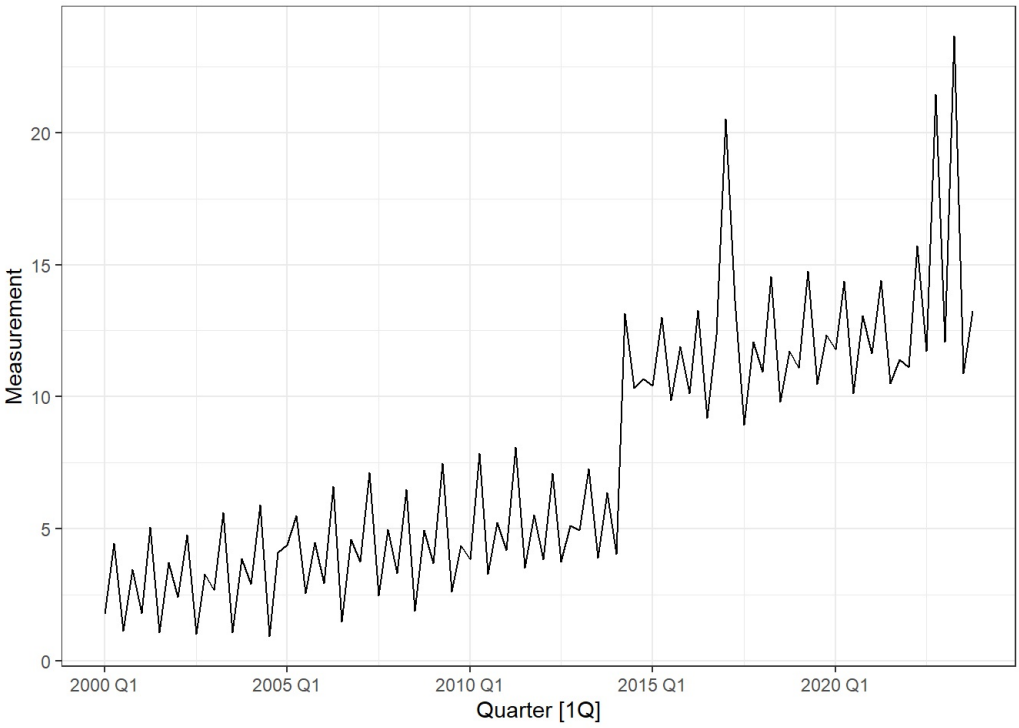
We calculated that the naive method has a MAE value of 0.735, MASE of 0.817, RMSE of 0.990 and MAPE of 4.74 which are the lowest forecast errors across three out of four metrics. This suggests that it provides the most accurate fit to the actual data from the two most recent years. Hence, naive is the best forecast method for monthly average Auckland temperatures.

> Total marks for **Problem 1**: 12 Marks

Problem 2: Time series regression

The data set `tsr.csv` is a data set containing quarterly measurements of an anonymised variable from 2000 Q1 until 2023 Q4. The time series is plotted below. You will notice the following features in this time series:

- Seasonality.
- Linear trend with level shift (steps).
- Outliers (spikes).



1. 4 Marks Using any approach you want, find the three quarters that the outliers occur at. Create a dummy variable that is 1 at these three quarters and 0 otherwise. Note: Usually when dealing with outliers, you would create a dummy variable for each individual outlier. However, here you may assume the outliers are due to the same phenomenon, so only create one dummy variable for all three outliers.

Hide

```
#create dummy variable that is 1 if measurements are greater than 20, 0 otherwise.
data <- data %>%
  mutate(Outlier = ifelse(Measurement > 20, 1, 0))

data %>% filter(Outlier == 1)
```

```
## # A tsibble: 3 x 3 [1Q]
##   Quarter Measurement Outlier
##     <qtr>       <dbl>   <dbl>
## 1 2017 Q1        20.5       1
## 2 2022 Q4        21.4       1
## 3 2023 Q2        23.7       1
```

2. 4 Marks Using any approach you want, find the quarter at which the level-shift occurs (i.e., where the time series level moves up). Create a dummy variable for the level-shift that takes the value 1 from that quarter onwards and 0 otherwise.

Hide

```
#for all quarters from 2014 Q2 onwards, set dummy variable to 1, 0 otherwise.
data <- data %>%
  mutate(Level_Shift = ifelse(Quarter >= yearquarter('2014 Q2'), 1, 0))
```

3. 6 Marks Fit the following three different time series regression models.

- Model 1: Linear trend with level shift and seasonal factors.
- Model 2: Linear trend with level shift, seasonal factors, and outliers.
- Model 3: Linear trend with level shift, $K = 1$ Fourier terms, and outliers.

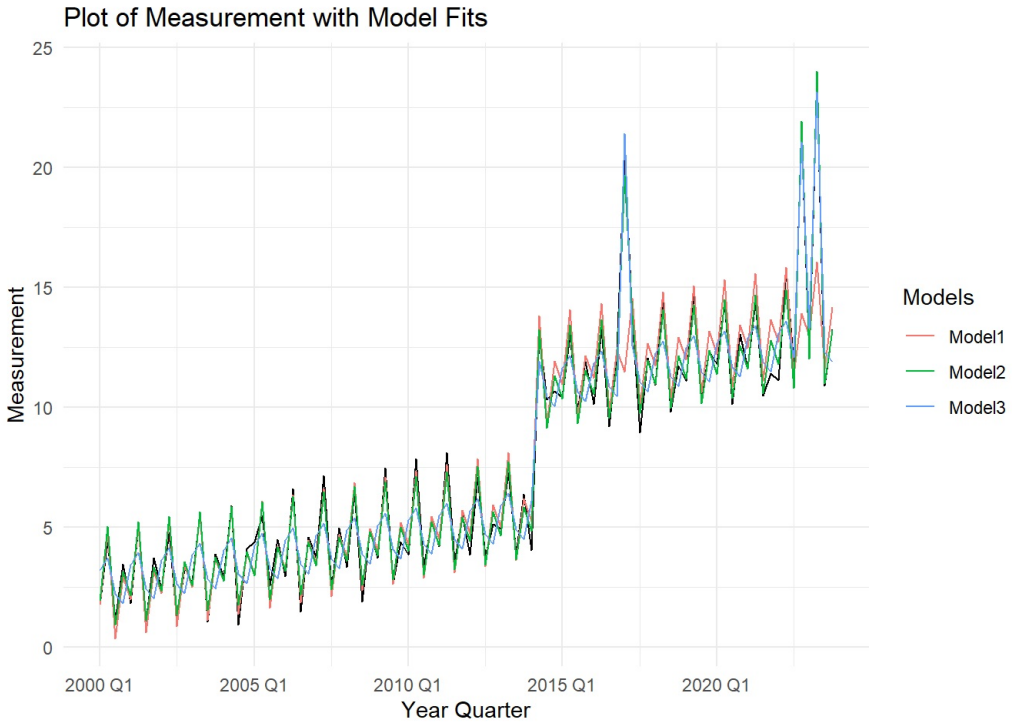
Hide

```
#fit three different models
Models.fit <- data %>%
  model(Model1 = TSLM(Measurement ~ trend() + season() + Level_Shift),
        Model2 = TSLM(Measurement ~ trend() + season() + Level_Shift + Outlier),
        Model3 = TSLM(Measurement ~ trend() + fourier(K = 1) + Level_Shift + Outlier))

#extract fitted models in a variable
Models <- Models.fit %>% augment()
```

4. 6 Marks Plot the original time series with the three sets of fitted values and comment on how well each model fits the data.

Hide

```
#fit models to original data
data %>%
  autoplot(Measurement) +
  geom_line(data = Models, aes(y = .fitted, colour = .model)) +
  xlab("Year Quarter")+ ylab("Measurement") +
  ggtitle("Plot of Measurement with Model Fits") +
  guides(colour=guide_legend(title = "Models")) +
  theme_minimal()
```



In observing our regression models, it appears that Model 1 generates a more consistent variability in the fitted values as we have excluded the Outlier term in our model. This model seems to be more reliable as we usually exclude outliers, however, excluding the Outlier term makes it the least accurate fit to our data. Model 2 and 3 have included the Outlier term with the only differences being the Seasonal and Fourier terms, they both appear to reflect the actual data very well and we would require more analysis to determine which provides the most accurate fit.

5. 6 Marks Compare the three models using AICc and the CV-statistic and conclude which model has the best predictive ability. This will be your selected model for the questions to follow. Report the estimates for your selected model.

Hide

```
#compare AICc and Cv-statistic values
glance(Models.fit) %>%
  select(.model, r_squared, adj_r_squared, df, AICc, CV)
```

```
## # A tibble: 3 × 6
##   .model r_squared adj_r_squared    df   AICc    CV
##   <chr>      <dbl>         <dbl> <int>  <dbl> <dbl>
## 1 Model1     0.894         0.889     6  102.  2.83
## 2 Model2     0.990         0.990     7 -124.  0.275
## 3 Model3     0.918         0.914     6   77.2 2.14
```

Hide

```
#report model 2 estimates
Models.fit %>% select(Model2) %>% report()
```

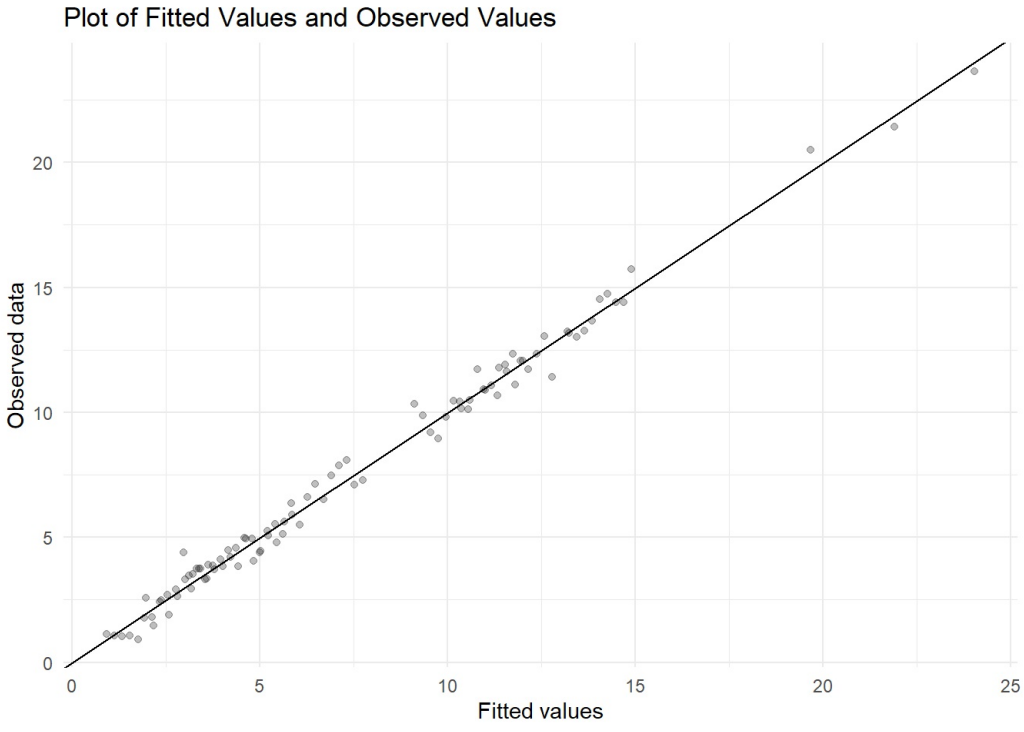
```
## Series: Measurement
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36831 -0.33043 -0.02308  0.31677  1.41538
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.877660   0.148145  12.675  < 2e-16 ***
## trend()        0.052125   0.003496  14.911  < 2e-16 ***
## season()year2  3.047059   0.143211  21.277  < 2e-16 ***
## season()year3 -1.105161   0.143805  -7.685 1.91e-11 ***
## season()year4  1.040672   0.143216   7.266 1.35e-10 ***
## Level_Shift    5.280669   0.196289  26.902  < 2e-16 ***
## Outlier        8.914841   0.301250  29.593  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4957 on 89 degrees of freedom
## Multiple R-squared: 0.9903,  Adjusted R-squared: 0.9896
## F-statistic:  1507 on 6 and 89 DF, p-value: < 2.22e-16
```

Model 1 has an AICc value of 102.06 and CV-statistic value of 2.83, which are the highest amongst the fitted models. This means Model 1 provides the least accurate fit compared to other models. Model 3 has an AICc value of 77.24 and a CV-statistic value of 2.14, which are lower than Model 1, hence it provides a better fit than Model 1. Model 2 has an AICc value -124.35 and CV-statistic value of 0.275 which are significantly lower than Model 1 and Model 3, hence Model 2 provides the most accurate fit to our measurements.

6. 10 Marks Check the model assumptions for your chosen model. You will need to assess linearity, independence, normality, zero mean, and equality of variance. Note 1: Linearity can be checked by comparing the observed versus fitted values.
   Note 2: For the Ljung-Box test, you can find the number of estimated parameters in the report you produced in (5), and because this is a seasonal time series, use $2m$ lags.

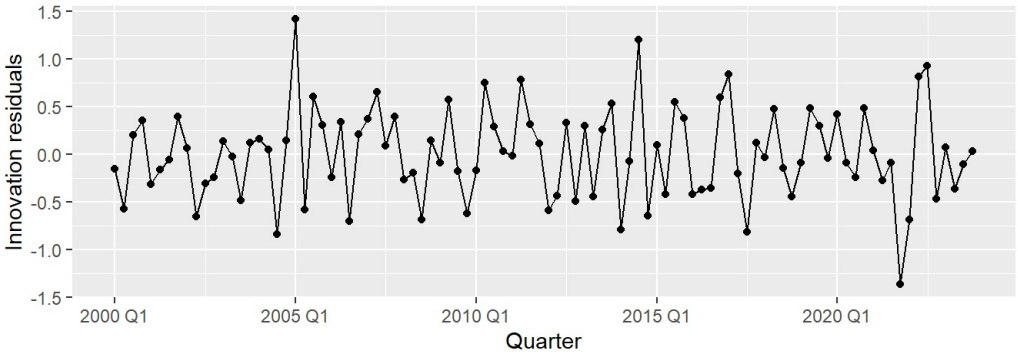
Hide

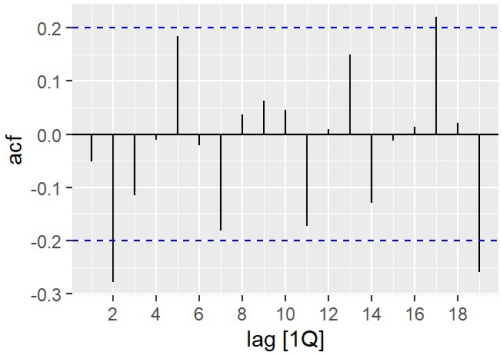
```
#store model 2 into a variable
Model2 <- data %>% model(Model2 = TSLM(Measurement ~ trend() + season() + Level_Shift + Outlier))

#plot observed vs fitted values
augment(Model2) %>%
  ggplot(aes(x = .fitted, y = Measurement)) +
  geom_point(alpha = 0.25) +
  labs(x = "Fitted values",
       y = "Observed data",
       title = "Plot of Fitted Values and Observed Values") +
  geom_abline(intercept = 0, slope = 1) +
  theme_minimal()
```

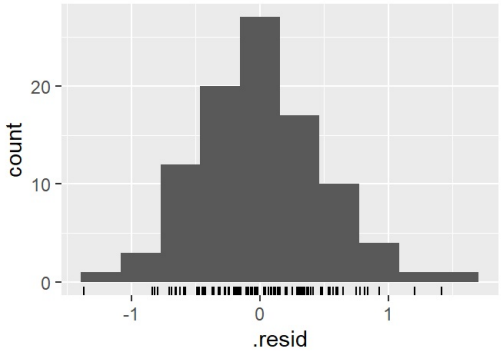

Plot of Fitted Values and Observed Values
Observed data
Fitted values
0
5
10
15
20
25

Hide

```
#residual and ACF plot
Model2 %>% gg_tsresiduals()
```







Hide

```
#Ljung_Box test
augment(Model2) %>% features(.resid, features = ljung_box, lag = 8, dof = 7)
```

```
## # A tibble: 1 × 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 Model2    16.4 0.0000501
```

The relationship between the observed values and fitted values appear to be approximately linear with the gradient being close to 1, which suggests a highly positive correlation between the observed and fitted values. The residuals appear to have approximately zero mean, and the variance of the residuals also seems roughly constant as most values fall within the range of -0.5 and 0.5, with only a few values exceeding this range. The residuals appears to be normally distributed which satisfies normality.

In our ACF plot, it appears that there are a few ACF values exceeding the threshold 0.2 and -0.2 for specific lags of 2,17 and 19. This means there is a 15.8% chance (3/19 = 0.158) that the lagged values exceed the 95% confidence interval, which is larger than the 5% significance level. This suggests that our residuals are not consistent with the assumptions of independence as there is

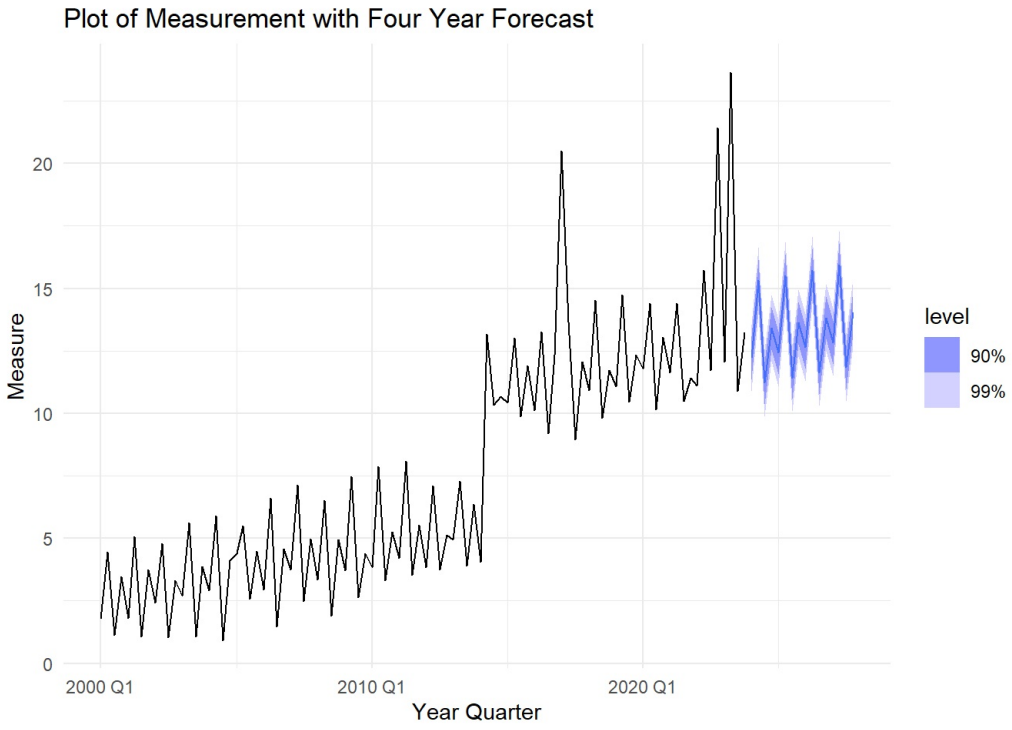
autocorrelation, non-zero mean and non-constant variance. Furthermore, we obtained an extremely small p-value of 5e-05 which suggests extremely strong evidence against the null hypothesis that the residuals are independent, in favor of the alternative hypothesis that there is autocorrelation, non-zero mean and non-constant variance in our residuals.

Since the ACF plot and Ljung-Box test indicates that the residuals are not independent, this means the assumptions of the time series linear regression model are not all satisfied.

7. 8 Marks Regardless of your assumption check, forecast four years into the future for your chosen model. Plot your forecasts with the 90% and 99% prediction intervals. Hint 1: The `new_data` function may be helpful. Hint 2: You will need to provide values for the two dummy predictors you created. Assume that no outliers will occur in these four years.

Hide

```
# Create tsibble with dummy variables
future_data <- new_data(data, n = 16)

# Creates new tsibble
future_data <- future_data %>%
  mutate(Outlier = rep(c(0,0,0,0), 4),
         Level_Shift = rep(c(1,1,1,1), 4))

# Forecast 4 years ahead
Model2.fc <- forecast(Model2, new_data = future_data)

# Plot forecast
Model2.fc %>%
  autoplot(data, level=c(90,99)) +
  labs(x = "Year Quarter",
       y = "Measure",
       title = "Plot of Measurement with Four Year Forecast") +
  theme_minimal()
```



8. 4 Marks Output and interpret the 90% prediction interval for the 1-step forecast of your chosen model (i.e., for `2024 Q1`). Hint: The `hilo` function may be helpful. Note: You may need to knit your document to see the output.

Hide

```
Model2.fc %>% hilo(level = 90) %>% filter(Quarter == yearquarter("2024 Q1"))
```

```
## # A tsibble: 1 x 7 [1Q]
## # Key:       .model [1]
##   .model Quarter Measurement .mean Outlier Level_Shift                `90%`
##   <chr>    <qtr>      <dist> <dbl>   <dbl>       <dbl>               <hilo>
## 1 Model2 2024 Q1 N(12, 0.27)  12.2       0           1 [11.3659, 13.06303]90
```

With 90% confidence, we estimate that the average measurement for 2024 Q1 is between 11.37 and 13.06.

> Total marks for **Problem 2**: 48 Marks for 326 58 Marks for 786

> Total possible marks for **Assignment 3**: 60 Marks for 326 70 Marks for 786