Lecture 11 Code Examples

1 Unit root testing

```r
set.seed(5209)
phi_vals <- c(0.5, 0.7, 0.9)
n <- 100
ar1_proc <- map(phi_vals, ~ arima.sim(model = list(ar = .), n = n))
phi_vals[[4]] <- 1
ar1_proc[[4]] <- arima.sim(model = list(order = c(0, 1, 0)), n = n)

adf_vals <- map(ar1_proc, ~ adf.test(.))
```

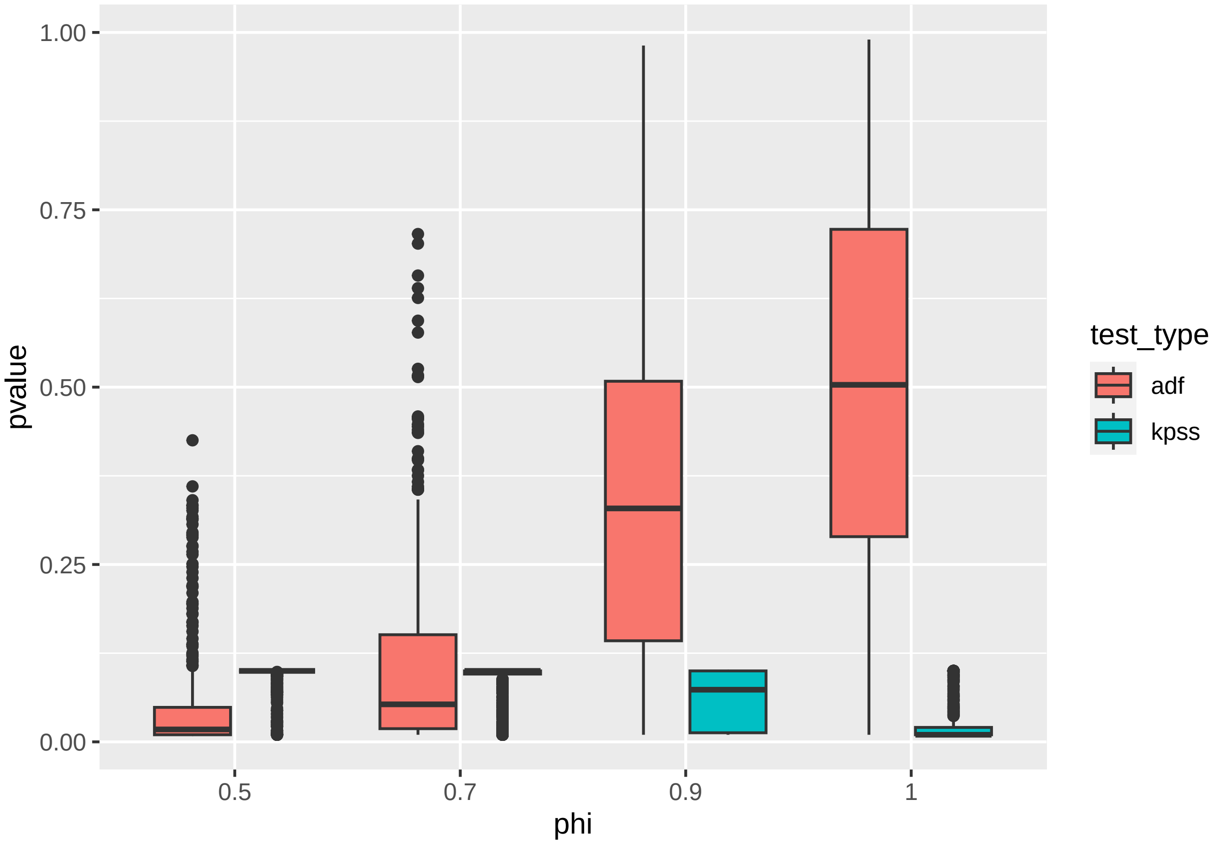
```
## Warning in adf.test(.): p-value smaller than printed p-value
```

```r
kpss_vals <- map(ar1_proc, ~ kpss.test(.))
```

```
## Warning in kpss.test(.): p-value greater than printed p-value
## Warning in kpss.test(.): p-value smaller than printed p-value

## Warning in kpss.test(.): p-value smaller than printed p-value
```

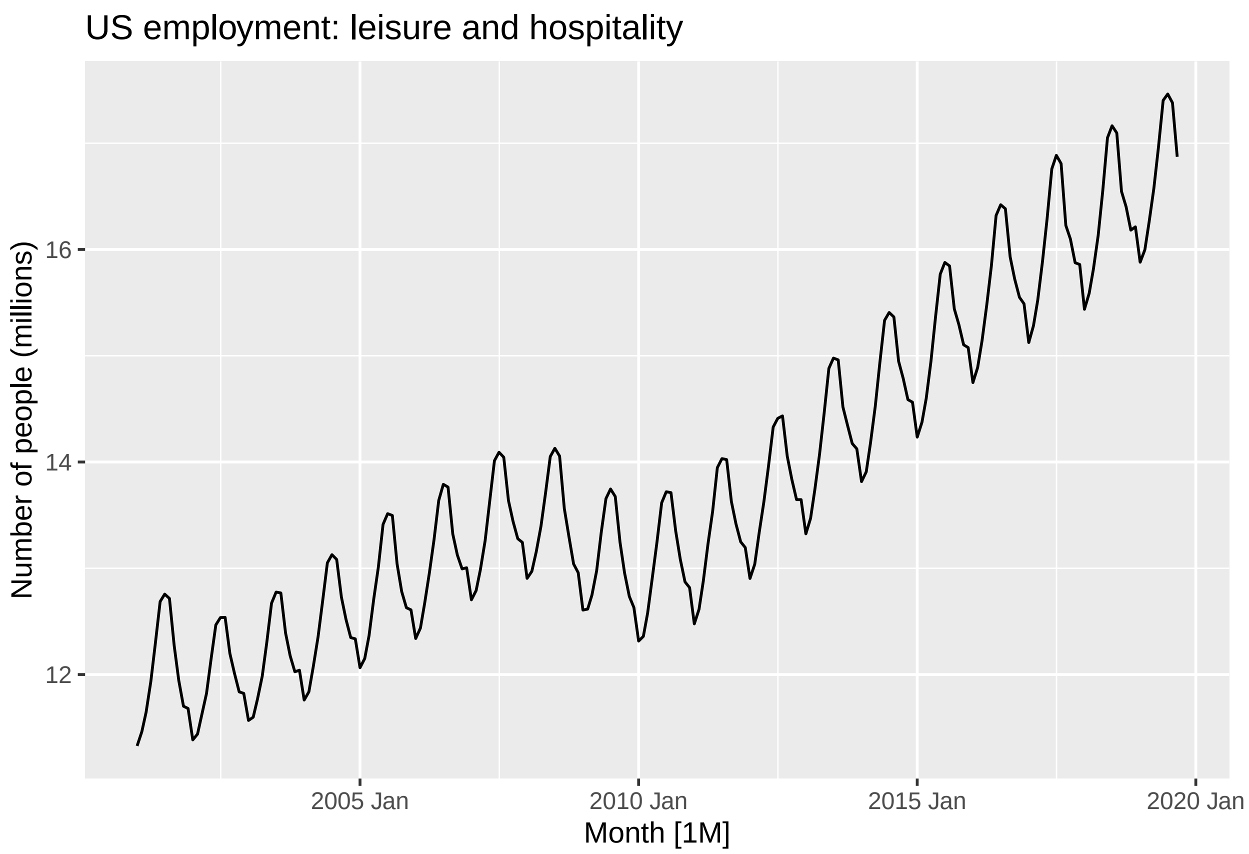
```r
test_results <- tibble(phi = phi_vals,
                       adf = map_dbl(adf_vals, ~ .$p.value),
                       kpss = map_dbl(kpss_vals, ~ .$p.value))
```

```r
B <- 500
run_tests <- function() {
  phi_vals <- c(0.5, 0.7, 0.9)
  n <- 100
  ar1_proc <- map(phi_vals, ~ arima.sim(model = list(ar = .), n = n))
  phi_vals[[4]] <- 1
  ar1_proc[[4]] <- arima.sim(model = list(order = c(0, 1, 0)), n = n)

  adf_vals <- map(ar1_proc, ~ adf.test(.))
  kpss_vals <- map(ar1_proc, ~ kpss.test(.))
  test_results <- tibble(phi = phi_vals,
                         adf = map_dbl(adf_vals, ~ .$p.value),
                         kpss = map_dbl(kpss_vals, ~ .$p.value)) |>
    pivot_longer(cols = c("adf", "kpss"),
                 names_to = "test_type",
                 values_to = "pvalue")
  test_results
}
all_results <- map(1:B, ~ run_tests()) |>
  bind_rows() |>
  mutate(phi = as.factor(phi))
ggplot(all_results) + geom_boxplot(aes(x = phi, y = pvalue, fill = test_type))
```



2 Seasonal ARIMA

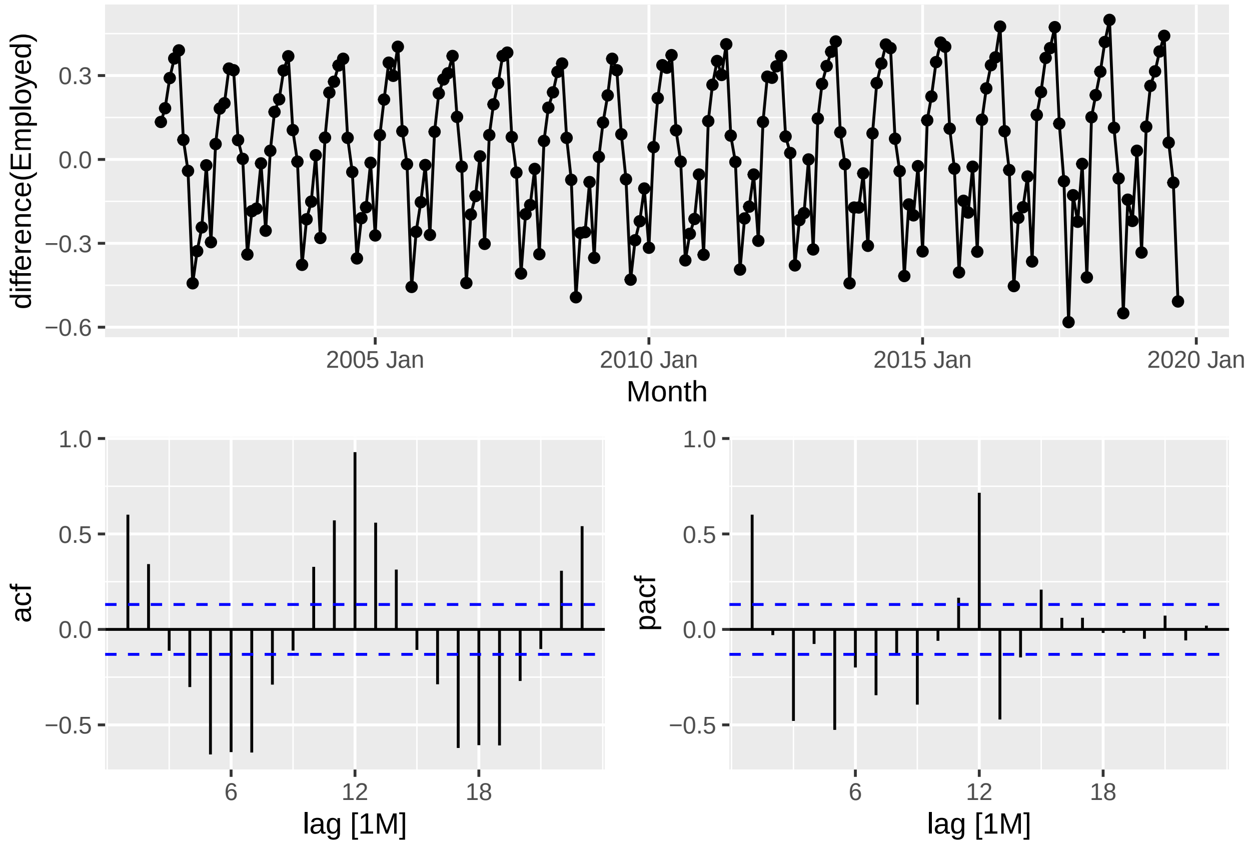
```
leisure <- us_employment |>
  filter(Title == "Leisure and Hospitality",
         year(Month) > 2000) |>
  mutate(Employed = Employed/1000) |>
  select(Month, Employed)

autoplot(leisure, Employed) +
  labs(title = "US employment: leisure and hospitality",
       y="Number of people (millions)")
```



```
gg_tsdisplay(leisure, difference(Employed), plot_type = "partial")
```

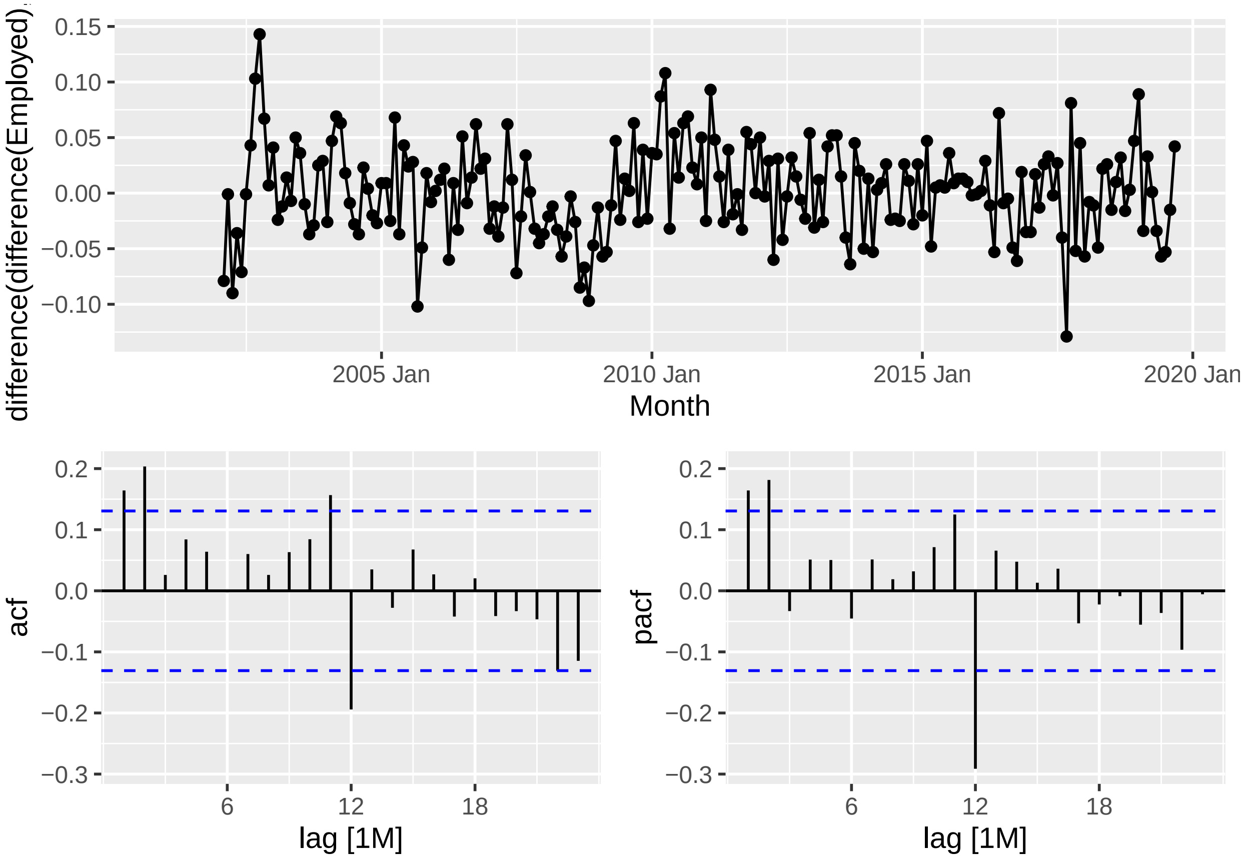
```
## Warning: Removed 1 row containing missing values (`geom_line()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



```
gg_tsdisplay(leisure, difference(Employed) |> difference(12),
             plot_type = "partial")
```

```
## Warning: Removed 13 rows containing missing values (`geom_line()`).

## Warning: Removed 13 rows containing missing values (`geom_point()`).
```

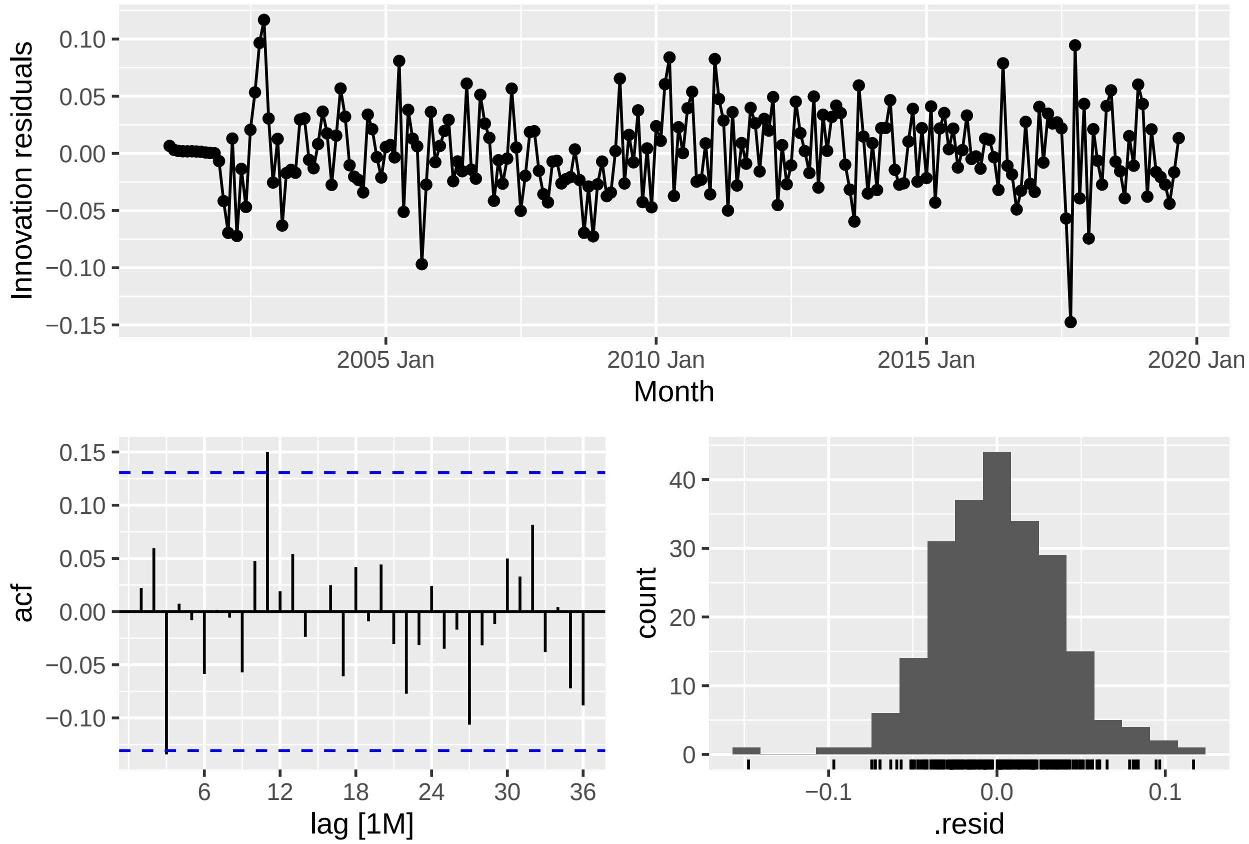


```
fit <- leisure |>
  model(
    arima012011 = ARIMA(Employed ~ pdq(0,1,2) + PDQ(0,1,1)),
    arima210011 = ARIMA(Employed ~ pdq(2,1,0) + PDQ(0,1,1)),
    arima210 = ARIMA(Employed ~ pdq(2,1,0) + PDQ(0,0,0)),
    semi_auto = ARIMA(Employed ~ pdq(0:2,1,0)),
    auto = ARIMA(Employed)
  )
```

```
glance(fit)
```

```
## # A tibble: 5 x 8
##   .model       sigma2 log_lik    AIC   AICc    BIC ar_roots    ma_roots
##   <chr>         <dbl>   <dbl>  <dbl>  <dbl>  <dbl> <list>      <list>
## 1 arima012011 0.00146    391.  -775.  -775.  -761. <cpl [0]>   <cpl [14]>
## 2 arima210011 0.00145    392.  -776.  -776.  -763. <cpl [2]>   <cpl [12]>
## 3 arima210    0.0442      32.2  -58.4  -58.3  -48.2 <cpl [2]>   <cpl [0]>
## 4 semi_auto   0.00142    395.  -778.  -778.  -758. <cpl [14]> <cpl [24]>
## 5 auto        0.00143    395.  -777.  -777.  -757. <cpl [25]> <cpl [13]>
```

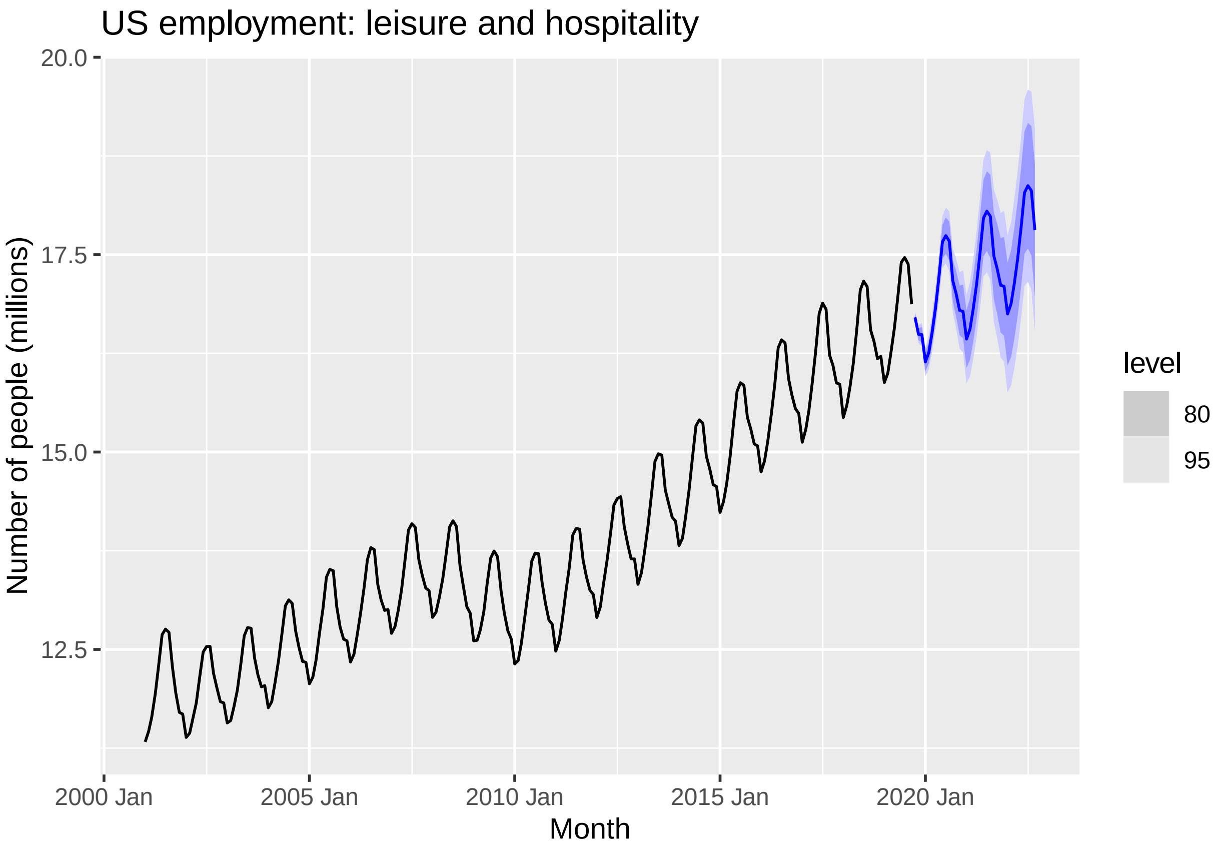
```
fit |> select(auto) |> gg_tsresiduals(lag=36)
```



```
augment(fit) |>
  filter(.model == "auto") |>
  features(.innov, ljung_box, lag=24, dof=5)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 auto      17.6     0.547
```

```
forecast(fit, h=36) |>
  filter(.model=='auto') |>
  autoplot(leisure) +
  labs(title = "US employment: leisure and hospitality",
       y="Number of people (millions)")
```


US employment: leisure and hospitality
20.0
17.5
15.0
12.5
Number of people (millions)
2000 Jan
2005 Jan
2010 Jan
2015 Jan
2020 Jan
Month
level
80
95