Forecasting Task - Problem Set 4

Bernardo Calvente e Ricardo Castro

2024-08-25

1 Setup

The following packages were used:

```
library(vars)
library(varr) #devtools::install_github("ricardo-semiao/varr")
library(urca)

library(tidyverse)
library(glue)
library(broom)
library(patchwork)

library(stargazer)
library(knitr)
library(kableExtra)

theme_set(theme_bw())
```

Custom function for the ADF test:

```
output_dftest <- function(data,
  nlag = NULL, pval = TRUE, index = TRUE, ..., types = 1:3) {
  aTSA::adf.test(data, nlag = nlag, output = FALSE) %>%
    imap_dfr(~ tibble(type = .y, as_tibble(.x))) %>%
    mutate(
      p.value = if (pval) {glue("({round(p.value, 2)})")} else {""},
      ADF = round(ADF, 2)
    ) %>%
    unite(Statistic, ADF, p.value, sep = " ") %>%
    pivot_wider(names_from = type, values_from = "Statistic") %>%
    set_names("Lag", glue("Type {1:3}")[index]) %>%
    select(c(1, types + 1)) %>%
    stargazer(summary = FALSE, header = FALSE, table.placement = "H")
}
```

2 Part I

3 Part II

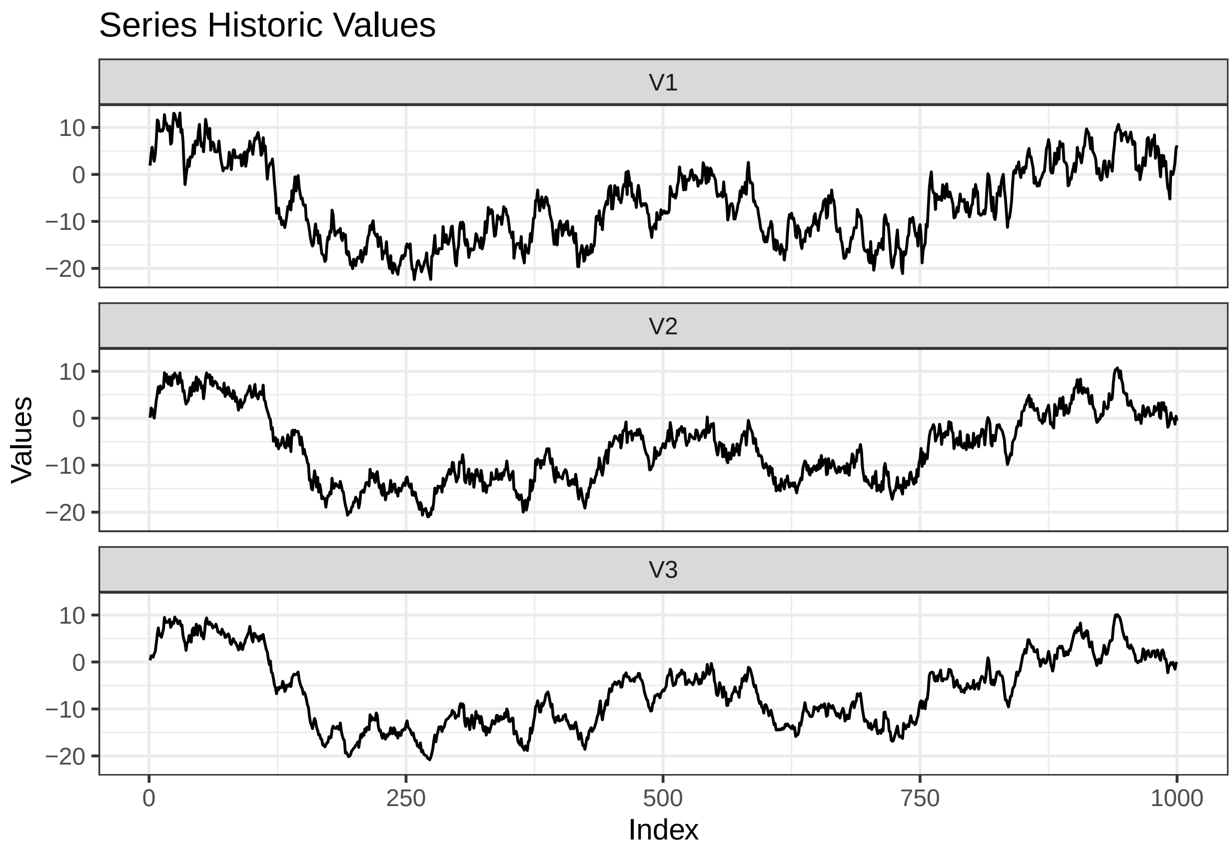
Lets load the data. For the estimation, lets discard the 100 first observations, to remove pseudo-randomness effect. Lets leave 100 observations for forecast comparison.

```r
data_vec <- read_csv("y_sys5.csv", col_select = -1)
data_vec_est <- slice(data_vec, 101:900)
```

3.1 Stationarity Analisys

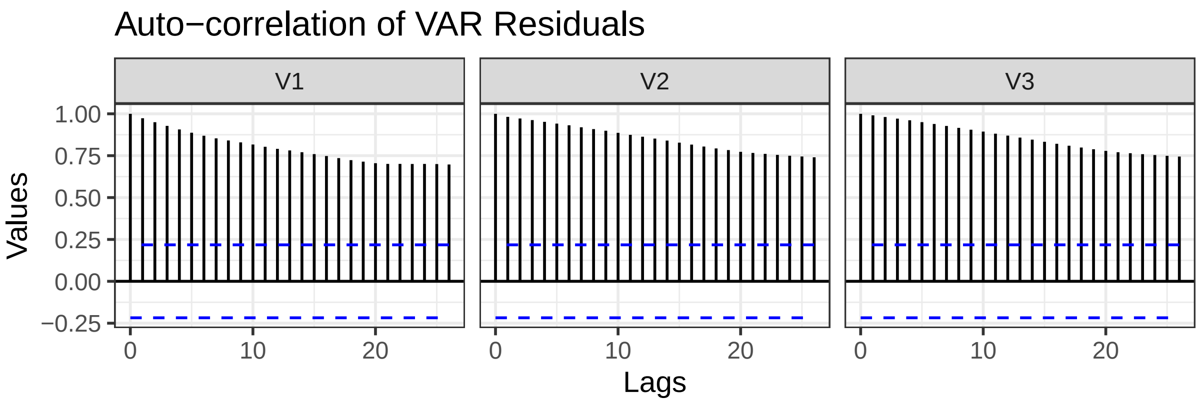
First, lets plot the data.

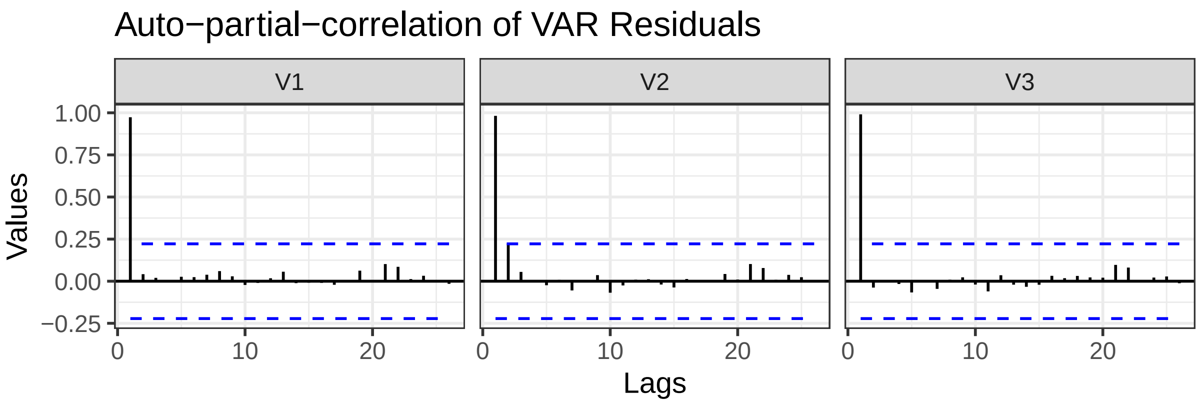
```r
ggvar_history(data_vec, args_facet = list(ncol = 1))
```



The series are similar, and seem to present a stochastic trend. Lets find more evidence in the ACFs/PACFs:

```r
ggvar_acf(data_vec) + ggvar_acf(data_vec, type = "partial") +
  plot_layout(ncol = 1)
```





We can see that there is a lot of autocorrelation, which does not quckly decreases. This seems like a non-stationary series. Lets use a ADF test to consolidate our hypothesis.

```
walk(data_vec, ~ output_dftest(.x, nlag = 3, types = 2:3))
```

Table 1:

	Lag	Type 2	Type 3
1	0	-3.46 (0.01)	-3.56 (0.04)
2	1	-3.3 (0.02)	-3.43 (0.05)
3	2	-3.26 (0.02)	-3.39 (0.05)

Table 2:

	Lag	Type 2	Type 3
1	0	-2.98 (0.04)	-3.08 (0.12)
2	1	-2.36 (0.18)	-2.49 (0.37)
3	2	-2.23 (0.24)	-2.36 (0.42)

Table 3:

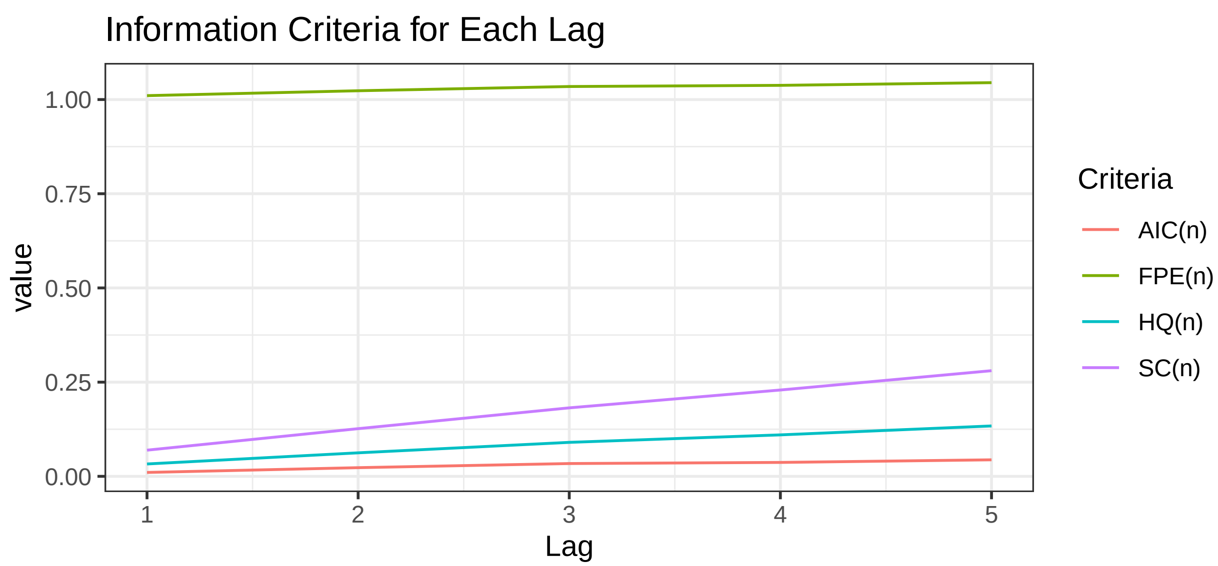
	Lag	Type 2	Type 3
1	0	-2.07 (0.3)	-2.19 (0.5)
2	1	-2.17 (0.26)	-2.29 (0.45)
3	2	-2.17 (0.26)	-2.3 (0.45)

The series clearly have a drift, such that only the test types 2 and 3 were considered. While the first one presents mixed evidence from the type 2 test, the majority of the test points to a non-stationary set of variables. Thus, we will be interested in diferentiating the series.

3.2 VAR Selection

The lag selection is done below. It was done before differentiation, as the prompt asked.

```
ggvar_select(VARselect(data_vec, lag.max = 5))
```




```
VARselect(data_vec, lag.max = 5)$criteria %>%
  round(4) %>%
  kable()
```

	1	2	3	4	5
AIC(n)	0.0103	0.0230	0.0339	0.0370	0.0438
HQ(n)	0.0328	0.0624	0.0901	0.1100	0.1338
SC(n)	0.0694	0.1265	0.1818	0.2291	0.2804
FPE(n)	1.0104	1.0233	1.0345	1.0377	1.0448

All of the criterias have their minimum at $p = 1$.

3.3 Model Creation

We create the $VAR(2)$ at level, the $VAR(1)$ in differences, and the VECM at last.

```
var_l2 <- VAR(data_vec, p = 2)

var_d1 <- VAR(map_dfc(data_vec, diff), p = 1)
```

We can see the result of the Johansen Procedure for the number of cointegrating relations:

```
vec_test <- ca.jo(data_vec,
  type = 'eigen',
  ecdet = 'const',
  K = 2,
  spec = 'longrun'
)

cbind(Test = vec_test@teststat, vec_test@cval) %>%
  kable()
```

	Test	10pct	5pct	1pct
r <= 2 |	4.742047	7.52	9.24	12.97
r <= 1 |	73.221806	13.75	15.67	20.20
r = 0 |	385.195852	19.77	22.00	26.81

We find that there is only one relation, and now we can define our VECM:

```
vec <- cajools(vec_test, reg.number = 1)
```

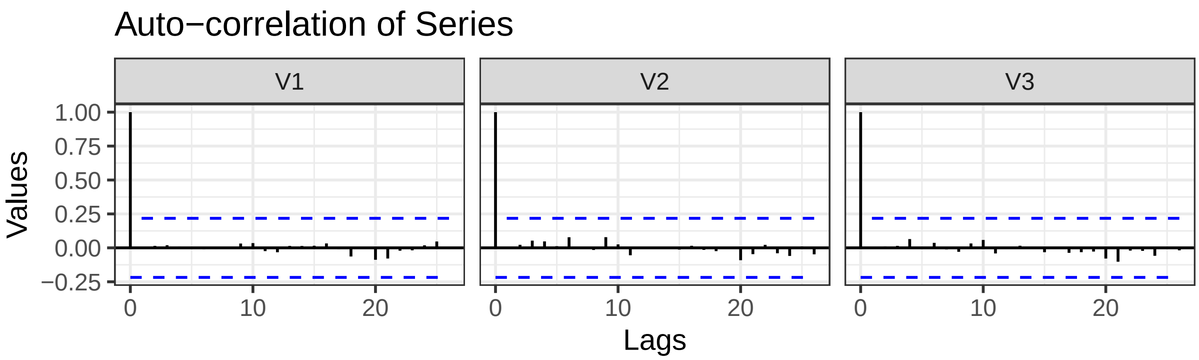
3.4 Model Diagnostics

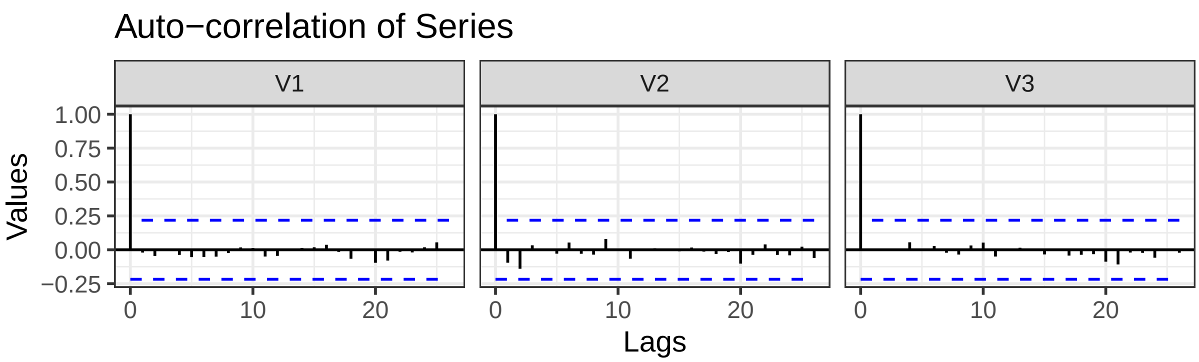
Below I present several graphs of the diagnostics of the models:

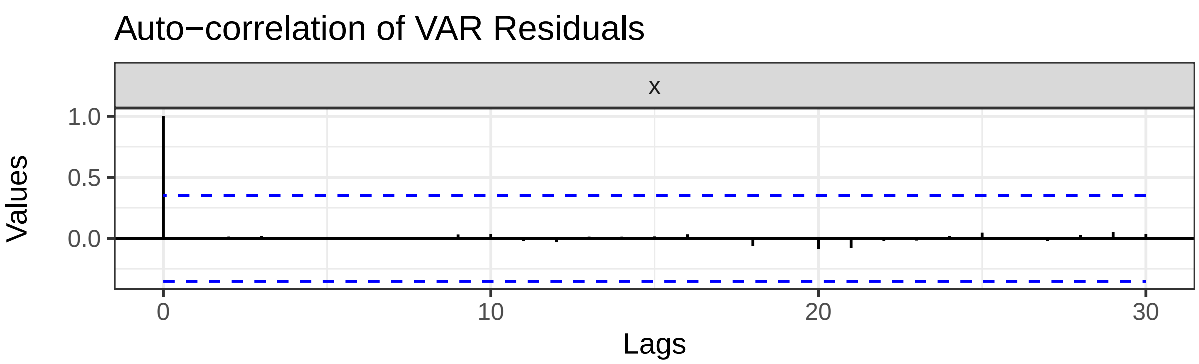
- The ACF of and CCF between residuals. Even in the VAR in levels, we do not end up with residual autocorrelation. Still, it can be a problem for forecasting.
- The dispersion of residuals, which does not seem to present major heteroskedasticity issues. But, this plot does not remove the possibility of ARCH errors.
- The distribution of the residuals, which seem fairly normal-like.
- The chow-test stability of the residuals, which seem to show now structural breaks.

All of those are presented in the order VAR(2) L., VAR(1) D., and VECM.

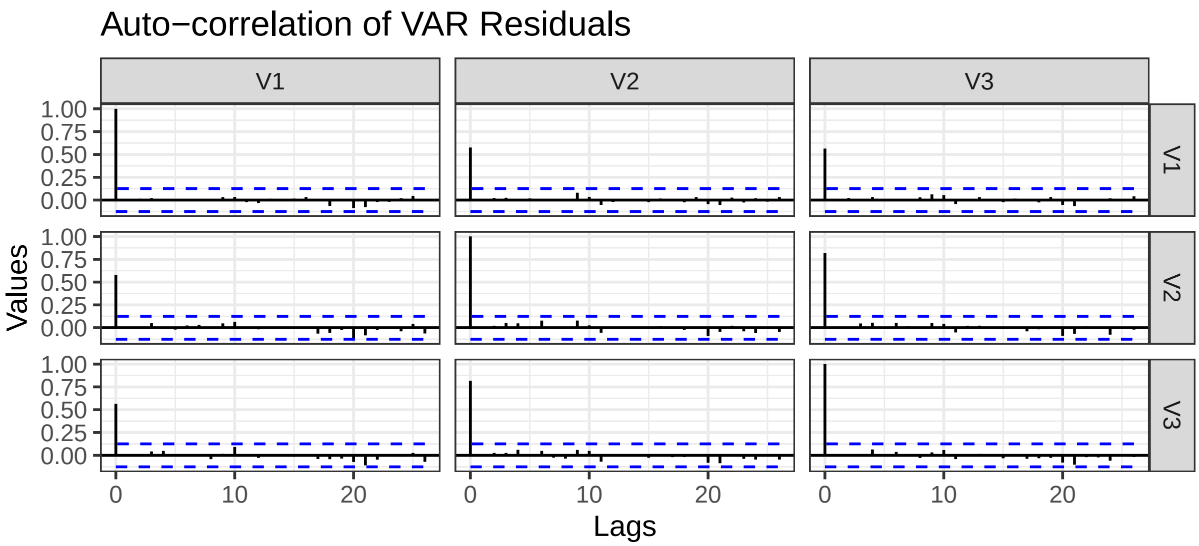
```
ggvar_acf(var_l2) + ggvar_acf(var_d1) + ggvar_acf(residuals(vec)) +
  plot_layout(ncol = 1)
```

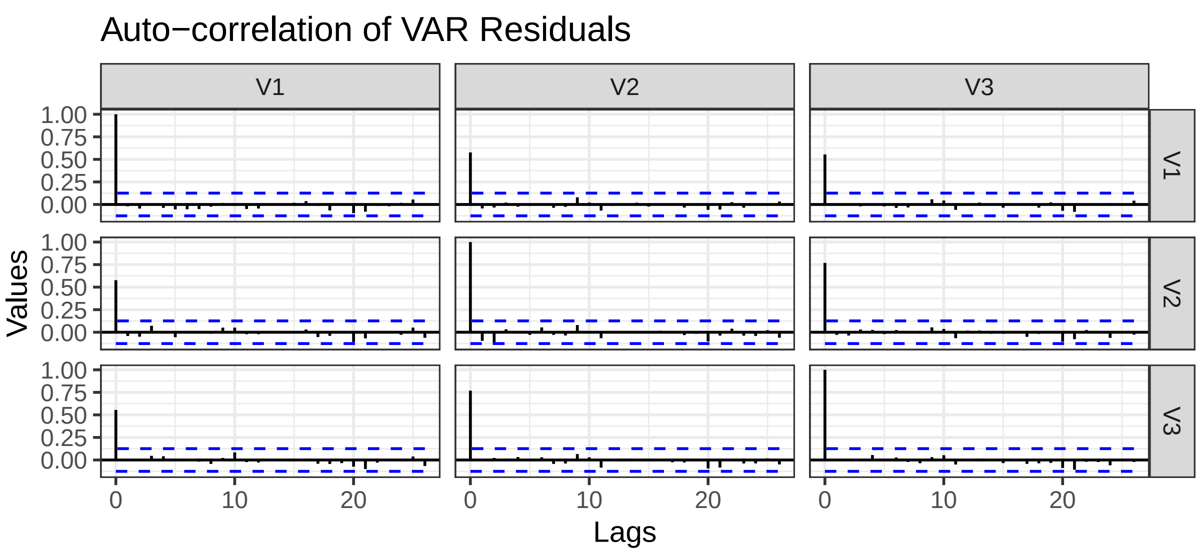




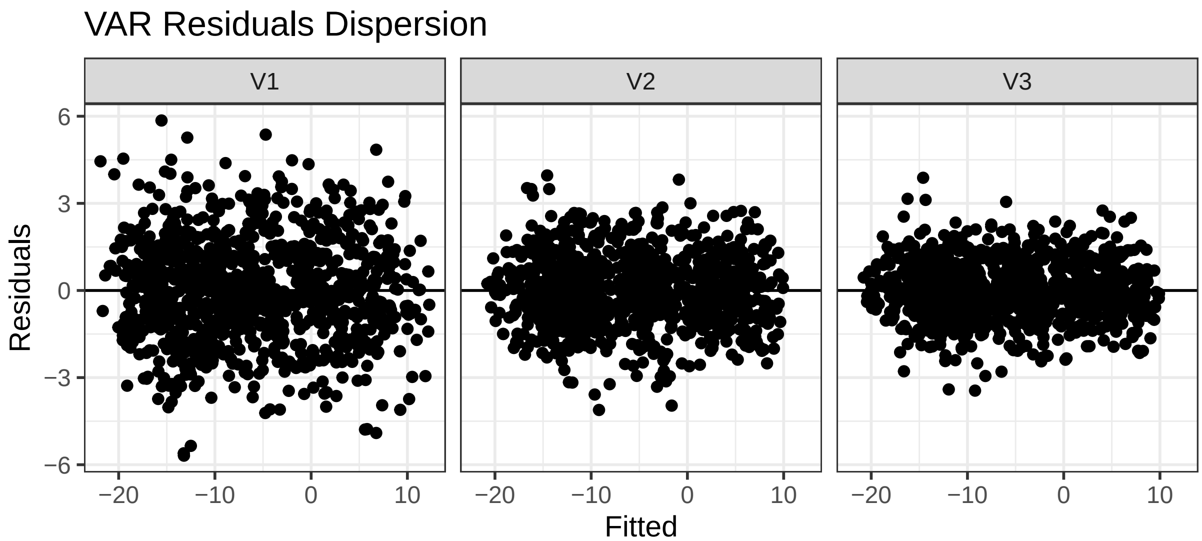


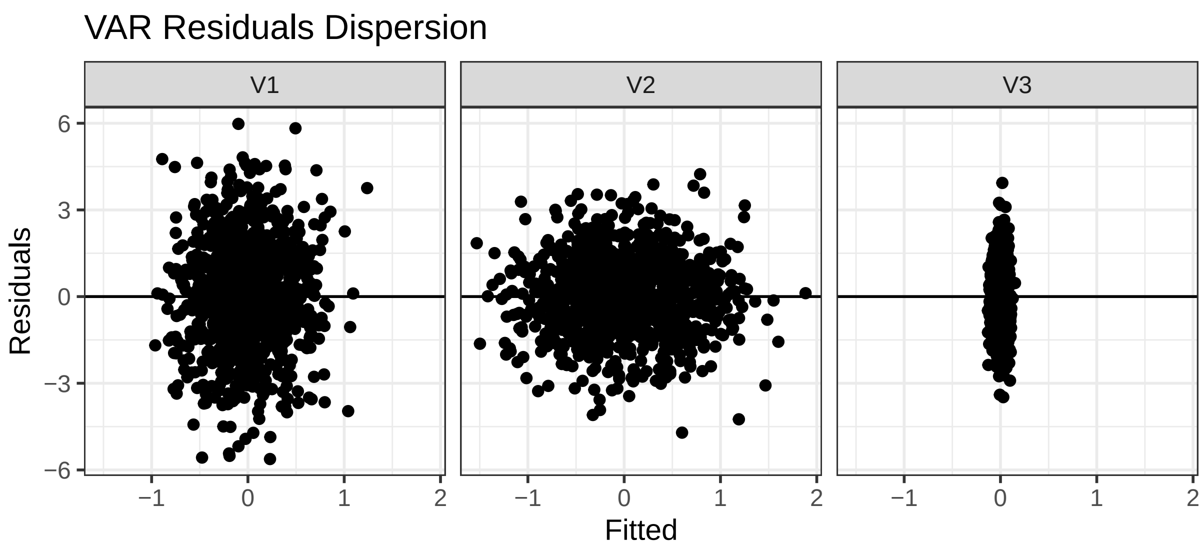

```
ggvar_ccf(var_l2) + ggvar_ccf(var_d1) +
  plot_layout(ncol = 1)
```



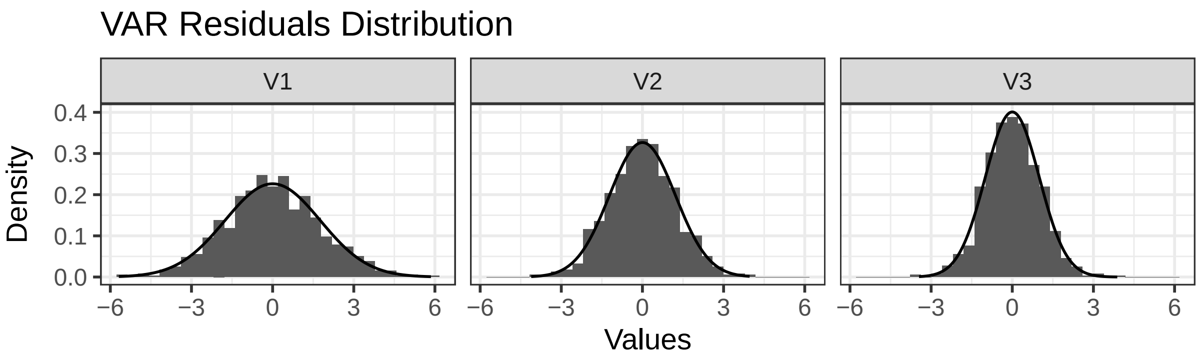


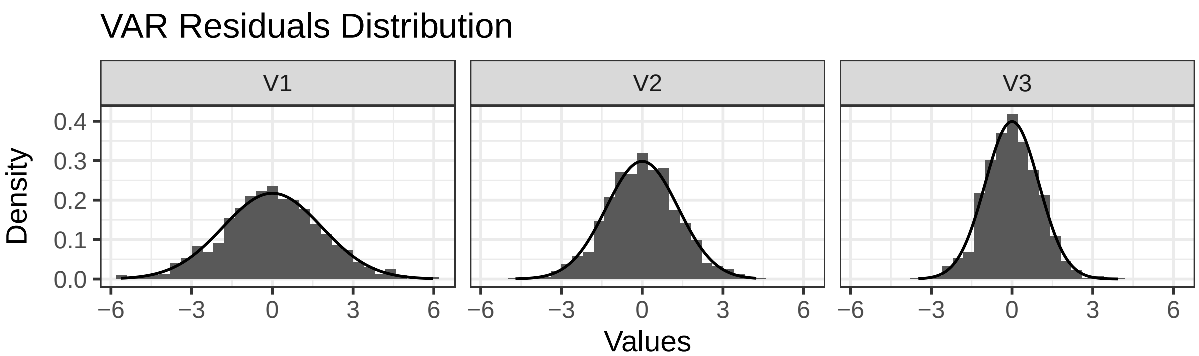

```
ggvar_dispersion(var_l2) + ggvar_dispersion(var_d1) +
  plot_layout(ncol = 1)
```

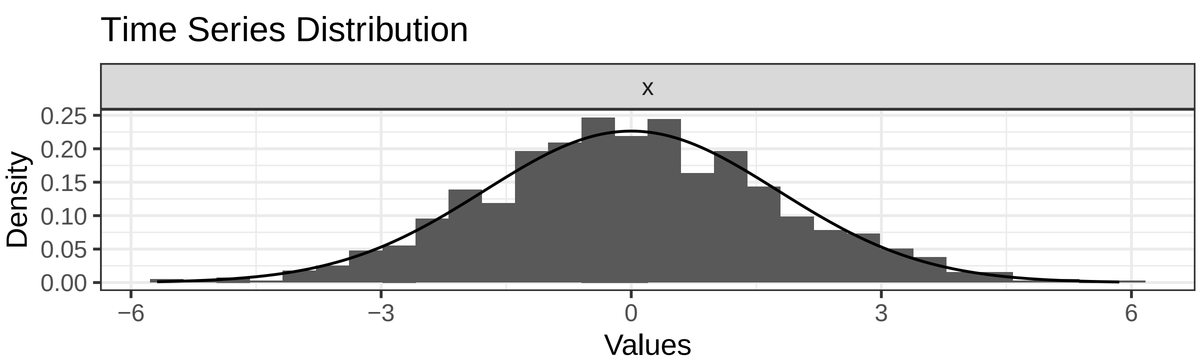




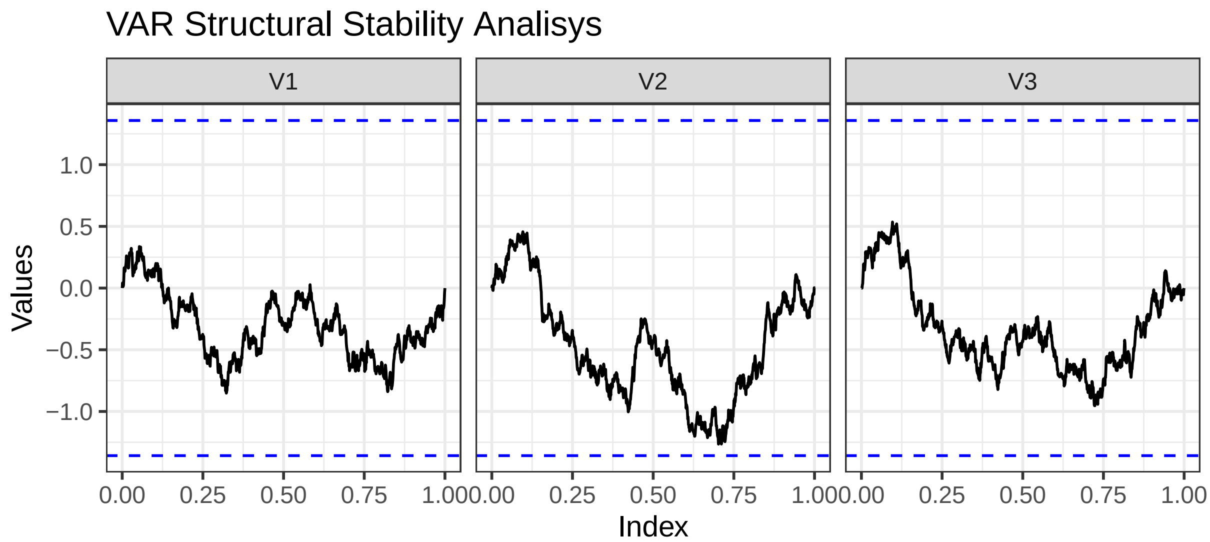

```
ggvar_distribution(var_l2) + ggvar_distribution(var_d1) + ggvar_distribution(residuals(vec)) +
  plot_layout(ncol = 1)
```

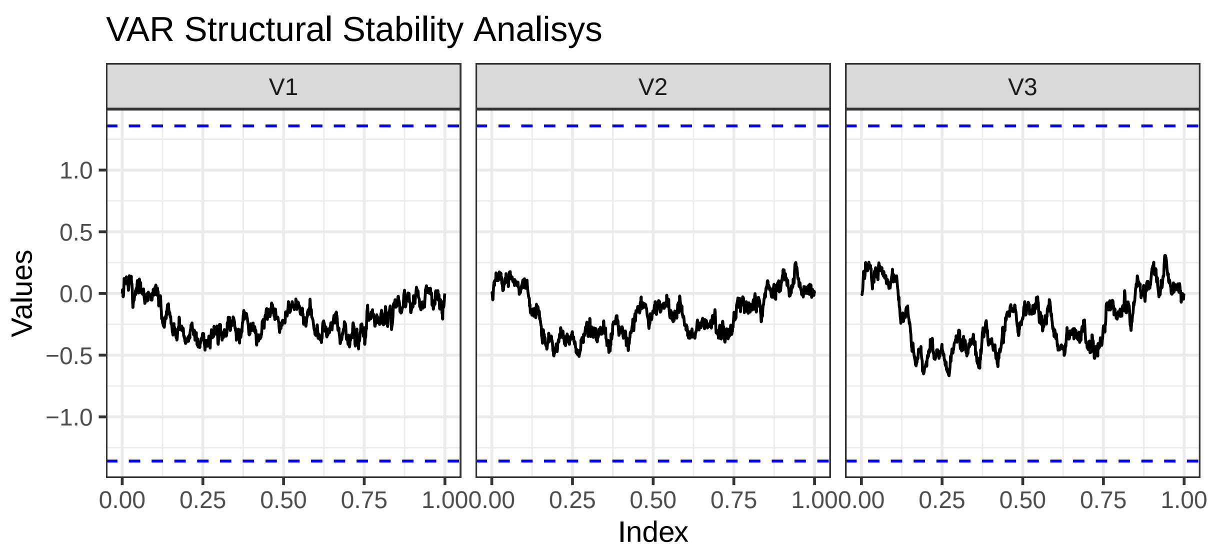







```
ggvar_stability(var_l2) + ggvar_stability(var_d1) +
  plot_layout(ncol = 1)
```





3.5 Forecasting

Now, we can generate predictions from our models.

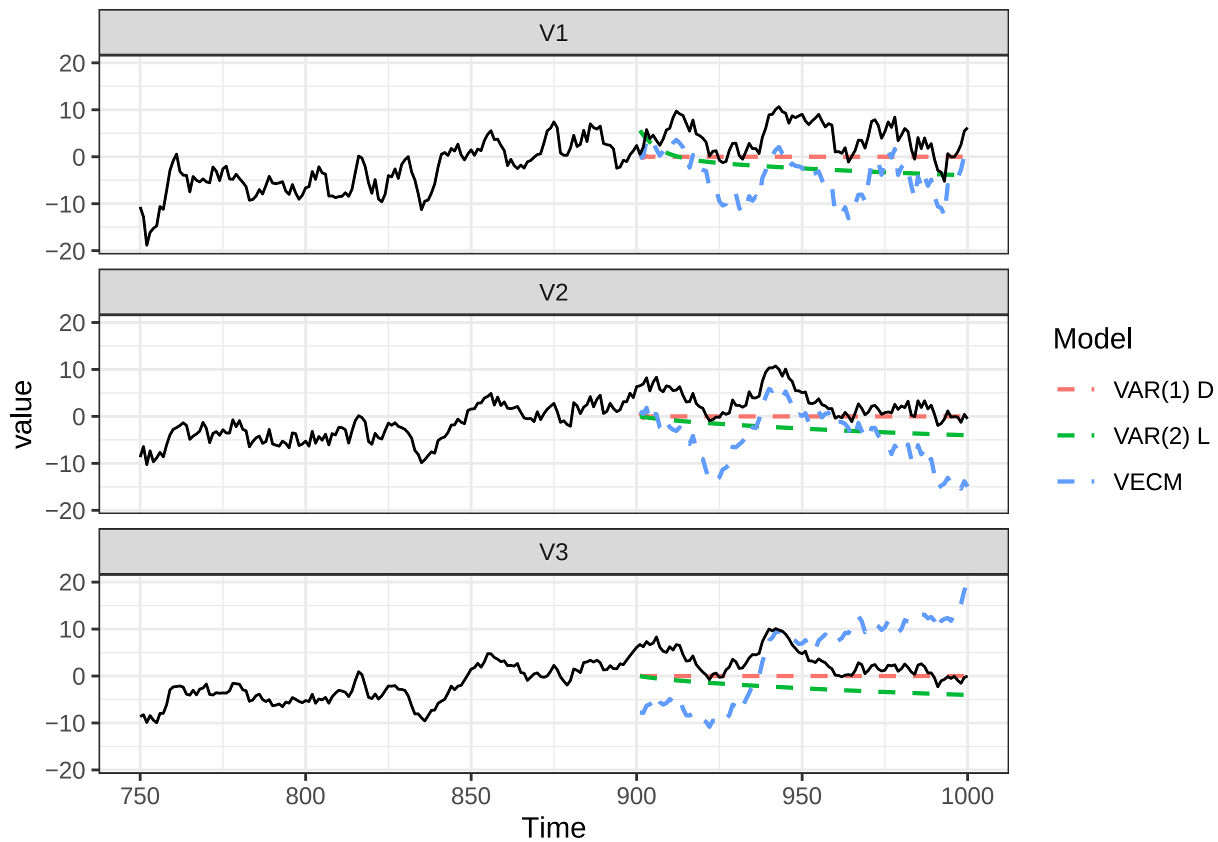
```
predictions <- list(
  `VAR(2) L` = map_dfc(predict(var_l2, n.ahead = 100)$fcst, ~.x[,"fcst"]),
  `VAR(1) D` = map_dfc(predict(var_d1, n.ahead = 100)$fcst, ~.x[,"fcst"]),
  VECM = data_vec[901:1000,] + cumsum(predict(vec, n.ahead = 100))
) %>%
  imap_dfr(~ cbind(Model = .y, Time = 901:1000, .x))
```

We can plot the results and get the prediction statistics:

```
predictions_g <- predictions %>%
  bind_rows(cbind(Time = 750:1000, Model = "True", data_vec[750:1000,])) %>%
  pivot_longer(-c(Model, Time))

ggplot(filter(predictions_g, Model != "True"), aes(Time, value)) +
```

```
  geom_line(aes(color = Model), linetype = 2, linewidth = 0.75) +
  geom_line(data = filter(predictions_g, Model == "True")) +
  facet_wrap(vars(name), ncol = 1)
```




```
predictions %>%
  select(-Time) %>%
  group_by(Model) %>%
  group_map(function(group, key) {
    imap_dfc(group, function(col, name) {
      e <- col - data_vec[[name]][901:1000]
      c(sqrt(mean(e^2)), mean(abs(e))) %>% round(3)
    }) %>%
      mutate(Model = key$Model, Statistic = c("RMSE", "MAE"), .before = 1)
  }) %>%
  bind_rows() %>%
  kable()
```

Model	Statistic	V1	V2	V3
VAR(1) D	RMSE	5.403	4.334	4.249
VAR(1) D	MAE	4.470	3.242	3.209
VAR(2) L	RMSE	7.214	6.124	6.050
VAR(2) L	MAE	6.290	5.499	5.482
VECM	RMSE	8.844	8.470	9.736
VECM	MAE	8.333	7.640	8.865

The VAR models do not carry enough information, and their prediction quickly become just an intercept. The VECM carries the cointegration relation, but that did not surpass the simpler prediction of the VARs models.