Nejprve je zde kód pro výpočty ohledně numerické studie na konci kapitoly 2.2. Začínám výpočtem hodnot $\beta_\tau = 1 + 2\,\varphi^{2\,\tau}$ pro AR(1) pro zvolené hodnoty φ a τ (v druhém výpočtu přidávám hodnotu pro $\tau \to \infty$)

```
In[113]:= iFiniteN01 =
  Table[N[1 + 2 ϕ^(2 τ)], {ϕ, {0.1, 0.5, 0.9}}, {τ, {1, 2, 5, 10, 20, 50}}] // Transpose

Out[113]= {{1.02, 1.5, 2.62}, {1.0002, 1.125, 2.3122}, {1., 1.00195, 1.69736},
  {1., 1., 1.24315}, {1., 1., 1.02956}, {1., 1., 1.00005}}

In[114]:= AppendTo[iFiniteN01, Table[Limit[1 + 2 ϕ^(2 τ), τ → Infinity], {ϕ, {0.1, 0.5, 0.9}}]]

Out[114]= {{1.02, 1.5, 2.62}, {1.0002, 1.125, 2.3122}, {1., 1.00195, 1.69736},
  {1., 1., 1.24315}, {1., 1., 1.02956}, {1., 1., 1.00005}, {1., 1., 1.}}
```

Dalé postupnými úpravami umožňuji vytvoření pomocné tabulky pro vyplňování tabulky v bakalářské práci.

```
In[115]:= iFiniteN01 = iFiniteN01 // Transpose;
PrependTo[iFiniteN01,
  {"τ = 1", "τ = 2", "τ = 5", "τ = 10", "τ = 20", "τ = 50", "τ -> ∞"}];
iFiniteN01 = iFiniteN01 // Transpose;
PrependTo[iFiniteN01, {"τ/φ", "φ = 0.1", "φ = 0.5", "φ = 0.9"}];
iTabel = Grid[Prepend[iFiniteN01,
   {"βτ = 1 + 2φ^(2 τ)", SpanFromLeft, SpanFromLeft, SpanFromLeft}], Frame → All]
```

Out[119]=

$\beta_\tau = 1 + 2\varphi^{2\,\tau}$			
τ/φ	φ = 0.1	φ = 0.5	φ = 0.9
τ = 1	1.02	1.5	2.62
τ = 2	1.0002	1.125	2.3122
τ = 5	1.	1.00195	1.69736
τ = 10	1.	1.	1.24315
τ = 20	1.	1.	1.02956
τ = 50	1.	1.	1.00005
τ -> ∞	1.	1.	1.

V dalším výpočtu počítám pro AR(1) hodnoty $\lambda_\tau = \frac{(1+\varphi^2)(1-\varphi^{2\,\tau})}{1-\varphi^2} - 2\,\tau\varphi^{2\,\tau}$ opět pro různé hodnoty φ a τ (v druhém výpočtu pro $\tau \to \infty$).

```
In[89]:= iFiniteN01A = Table[N[(((1 + ϕ^2) (1 - ϕ^(2 τ))) / (1 - ϕ^2)) - 2 τ * ϕ^(2 τ)],
   {ϕ, {0.1, 0.5, 0.9}}, {τ, {1, 2, 5, 10, 20, 50}}] // Transpose

Out[89]= {{0.99, 0.75, 0.19}, {1.0197, 1.3125, 0.6517}, {1.0202, 1.65527, 2.71791},
  {1.0202, 1.66665, 5.93661}, {1.0202, 1.66667, 8.79427}, {1.0202, 1.66667, 9.52341}}

In[90]:= AppendTo[iFiniteN01A,
  Table[Limit[(((1 + ϕ^2) (1 - ϕ^(2 τ))) / (1 - ϕ^2)) - 2 τ * ϕ^(2 τ), τ → Infinity],
   {ϕ, {0.1, 0.5, 0.9}}]]

Out[90]= {{0.99, 0.75, 0.19}, {1.0197, 1.3125, 0.6517}, {1.0202, 1.65527, 2.71791},
  {1.0202, 1.66665, 5.93661}, {1.0202, 1.66667, 8.79427},
  {1.0202, 1.66667, 9.52341}, {1.0202, 1.66667, 9.52632}}
```

Provádím úpravy pro vytvoření pomocné tabulky.

In[91]:=
```
iFiniteN01A = iFiniteN01A // Transpose;
PrependTo[iFiniteN01A,
  {"τ = 1", "τ = 2", "τ = 5", "τ = 10", "τ = 20", "τ = 50", "τ -> ∞"}];
iFiniteN01A = iFiniteN01A // Transpose;
PrependTo[iFiniteN01A, {"τ/φ", "φ = 0.1", "φ = 0.5", "φ = 0.9"}];
iTabelA = Grid[Prepend[iFiniteN01A, {"λτ = ((1 + φ^2) (1 - φ^(2 τ)))/(1 - φ^2) - 2τφ^(2 τ)",
   SpanFromLeft, SpanFromLeft, SpanFromLeft}], Frame → All]
```

Out[95]=

$\lambda_\tau = \frac{(1+\varphi^2)(1-\varphi^{2\tau})}{1-\varphi^2} - 2\tau\varphi^{2\tau}$			
τ/φ	φ = 0.1	φ = 0.5	φ = 0.9
τ = 1	0.99	0.75	0.19
τ = 2	1.0197	1.3125	0.6517
τ = 5	1.0202	1.65527	2.71791
τ = 10	1.0202	1.66665	5.93661
τ = 20	1.0202	1.66667	8.79427
τ = 50	1.0202	1.66667	9.52341
τ -> ∞	1.0202	1.66667	9.52632

Druhá část kódu je určena k simulacím a výpočtům v praktické části bakalářské práce v kapitole 4. Jako první generuji bílý šum z N(0,1) pomocí pseudogenerátoru náhodných procesů **RandomFunction** a funkce simulující bílý šum z N(0,1) **WhiteNoiseProcess[]**. Postupně generuji pro různé délky $n \in \{50, 100, 500, 100\}$ vždy 1000 realizací bílého šumu. Tato generace je společná pro kapitoly 4.1 a 4.2.

In[137]:=
```
SeedRandom[15];
dataWN =
 Table[RandomFunction[WhiteNoiseProcess[], {0, nn}, 1000], {nn, {50, 100, 500, 1000}}]
```

Rozděluji si generovaná data do proměnných pro jednotlivé délky bílého šumu.

In[139]:=
```
dataWN50 = Table[dataWN[[1]]["Values", i], {i, 1000}];
dataWN100 = Table[dataWN[[2]]["Values", i], {i, 1000}];
dataWN500 = Table[dataWN[[3]]["Values", i], {i, 1000}];
dataWN1000 = Table[dataWN[[4]]["Values", i], {i, 1000}];
```

Nyní se budou opakovat bloky kódu pro různé délky bílého šumu. Vždy nejprve podle vzorce (2.10) napočítám hodnoty

$$b_{\tau,n} = \frac{\frac{1}{n}\sum_{t=1}^{n-\tau} X_t^2 X_{t+\tau}^2}{\left(\frac{1}{n}\sum_{t=1}^{n} X_t^2\right)^2}$$

pro všech 1000 realizací bílého šumu dané délky postupně pro různá $\tau \in \{1, 2, 5, 10, 20\}$ a poté spočítám průměr $b_{\tau,n}$ přes 1000 napočítaných $b_{\tau,n}$ daného n a τ. Tedy pro danou délku n vždy 5 hodnot pro různá τ. Vše nejdříve pro $n = 50$.

In[ ]:=
```
iMeansWN50 = Table[
  Mean[Transpose[Table[50 * (Sum[dataWN50[[Nn]][[t]]^2 * dataWN50[[Nn]][[t + τ]]^2,
          {t, 50 - τ}] / (Sum[dataWN50[[Nn]][[t]]^2, {t, 50}])^2),
      {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]][[j]]], {j, 5}]
```

Out[ ]=
```
{0.945524, 0.917407, 0.868458, 0.773622, 0.55764}
```

Dále spočítám směrodatnou odchylku $b_{\tau,n}$ přes 1000 napočítaných $b_{\tau,n}$ daného n a τ. Opět tak dostáváme 5 hodnot.

In[•]:=
```
iSDsWN50 = Table[StandardDeviation[
   Transpose[Table[50 * (Sum[dataWN50[[Nn]][[t]]^2 * dataWN50[[Nn]][[t + τ]]^2,
         {t, 50 - τ}] / (Sum[dataWN50[[Nn]][[t]]^2, {t, 50}])^2),
      {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]][[j]]], {j, 5}]
```

Out[•]= {0.257274, 0.25206, 0.250074, 0.241499, 0.191565}

Rovněž v bílém šumu (za platnosti hypotézy testů o nekorelovanosti) určuji empirickou hladinu testu pro všechny 3 testy z kapitoly 3.1. Nejprve pro test s testovou statistikou (3.9) s výběrovou autokorelací prvního řádu r_1. Pro její výpočet používám zabudovaný empirický estimátor autokorelace z generovaných dat **CorrelationFunction**. Počítám hodnoty testové statistiky $\sqrt{n}\ |r_1|$ pro danou délku a všech 1000 realizací bílého šumu. Poté zjistím procházením 1000 hodnot testové statistiky funkcí **For**, kolik je jich větších než příslušný kvantil N(0,1) $u_{1-\frac{0.05}{2}} = 1.96$ při zvolené hladině testu 0,05. Toto číslo v proměnné *iCounts50* podělím 1000 a dostanu tak relativní četnost řad, které porušily test, tedy empirický odhad hladiny testu.

In[374]:=
```
i50r1 = Table[50^(1/2) Abs[CorrelationFunction[dataWN50[[Nn]], 1]], {Nn, 1000}];
iCounts50 = 0;
For[i = 1, i < 1001, i++, If[i50r1[[i]] > 1.96, iCounts50++]];
iCounts50
iα50 = N[iCounts50 / 1000]
```

Out[377]= 41

Out[378]= 0.041

To samé provedu i pro test s testovou statistikou Q_{28} z (3.11). Zde nejprve musím napočítat vektor výběrových autokorelací až do řádu 28, až pak určit samotných 1000 hodnot testové statistiky. Poté je ve funkci **For** porovnám s kvantilem $\chi^2_{28}(1 - 0.05) = 41.34$ rozdělení χ^2_{28}, zda jsou větší a porušují test.

In[390]:=
```
iR50 = Table[ CorrelationFunction[dataWN50[[Nn]], k], {Nn, 1000}, {k, 28}];

iQ50 = Table[50 * Sum[(iR50[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq50 = 0;
For[i = 1, i < 1001, i++, If[iQ50[[i]] > 41.34, iCountsq50++]];
iCountsq50
iαq50 = N[iCountsq50 / 1000]
```

Out[394]= 9

Out[395]= 0.009

Jako poslední určuji empirckou hladinu testu pro test s testovou statistikou (3.10) pro stejnou hodnotu $k = 28$ jako v předchozím testu, navíc dle textu bakalářské práce volíme $N_0 = 4$. Zde musím určit vnitřní funkcí **For** počet výběrových autokorelací řádů 1 až 28, které porušují test (3.9), až poté vnější funkcí **For** určuji počet řad z 1000 realizací, které nesplňují test (3.10), tedy že součet pravděpodobností $\alpha(k, N_0)$ (testová statistika) je větší než zvolená hladina testu 0,05. Pak už jen určím v posledním kroku relativní četnost takových řad.

In[396]:=
```
iCountsN50 = 0;
iCC50 = {};
For[i = 1, i < 1001, i++,
  For[j = 1, j < 29, j++, If[Abs[iR50[[i]][[j]]] > 1.96 / (50^(1/2)), iCountsN50++]];
  iCC50 = Join[iCC50, {iCountsN50}];
  iCountsN50 = 0];
iCountsN50 = 0;
For[i = 1, i < 1001, i++, If[iCC50[[i]] > 3, iCountsN50++]];
iCountsN50
iαN50 = N[iCountsN50 / 1000]
```

Out[401]= 12

Out[402]= 0.012

V následujících výpočtech se to samé opakuje pro $n = 100$.

In[∘]:=
```
iMeansWN100 = Table[
  Mean[Transpose[Table[100 * (Sum[dataWN100[[Nn]][[t]]^2 * dataWN100[[Nn]][[t + τ]]^2,
          {t, 100 - τ}] / (Sum[dataWN100[[Nn]][[t]]^2, {t, 100}])^2),
      {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]][[j]]], {j, 5}]
iSDsWN100 = Table[StandardDeviation[Transpose[Table[
      100 * (Sum[dataWN100[[Nn]][[t]]^2 * dataWN100[[Nn]][[t + τ]]^2, {t, 100 - τ}] /
          (Sum[dataWN100[[Nn]][[t]]^2, {t, 100}])^2),
      {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]][[j]]], {j, 5}]
```

Out[∘]= {0.973875, 0.961737, 0.92947, 0.888113, 0.773857}

Out[∘]= {0.178799, 0.183468, 0.189301, 0.18657, 0.175435}

In[143]:=
```
i100r1 = Table[100^(1/2) Abs[CorrelationFunction[dataWN100[[Nn]], 1]], {Nn, 1000}];
iCounts100 = 0;
For[i = 1, i < 1001, i++, If[i100r1[[i]] > 1.96, iCounts100++]];
iCounts100
iα100 = N[iCounts100 / 1000]
```

Out[146]= 44

Out[147]= 0.044

In[403]:=
```
iR100 = Table[ CorrelationFunction[dataWN100[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[404]:=
```
iQ100 = Table[100 * Sum[(iR100[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq100 = 0;
For[i = 1, i < 1001, i++, If[iQ100[[i]] > 41.34, iCountsq100++]];
iCountsq100
iαq100 = N[iCountsq100 / 1000]
```

Out[407]= 19

Out[408]= 0.019

In[409]:=
```
iCountsN100 = 0;
iCC100 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR100[[i]][[j]]] > 1.96 / (100^(1/2)), iCountsN100++]];
  iCC100 = Join[iCC100, {iCountsN100}];
  iCountsN100 = 0];
iCountsN100 = 0;
For[i = 1, i < 1001, i++, If[iCC100[[i]] > 3, iCountsN100++]];
iCountsN100
iαN100 = N[iCountsN100 / 1000]
```

Out[414]= 22

Out[415]= 0.022

V následujících výpočtech se to samé opakuje pro $n = 500$. Pro nižší časovou náročnost jsem zde opravoval kód v tom smyslu, že průměry a směrodatné odchylky $b_{\tau,n}$ počítám dohromady.

In[∘]:=
```
ibτnWN500 = Transpose[
  Table[500 * (Sum[dataWN500[[Nn]][[t]]^2 * dataWN500[[Nn]][[t + τ]]^2, {t, 500 - τ}] /
      (Sum[dataWN500[[Nn]][[t]]^2, {t, 500}])^2), {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]]
```

In[∘]:=
```
iMeansWN500 = Table[Mean[ibτnWN500[[i]]], {i, 5}]
iSDsWN500 = Table[StandardDeviation[ibτnWN500[[i]]], {i, 5}]
```

Out[∘]= {0.990269, 0.994688, 0.98438, 0.975275, 0.950881}

Out[∘]= {0.0898538, 0.0866493, 0.0904491, 0.0853613, 0.0855733}

In[148]:=
```
i500r1 = Table[500^(1/2) Abs[CorrelationFunction[dataWN500[[Nn]], 1]], {Nn, 1000}];
iCounts500 = 0;
For[i = 1, i < 1001, i++, If[i500r1[[i]] > 1.96, iCounts500++]];
iCounts500
iα500 = N[iCounts500 / 1000]
```

Out[151]= 57

Out[152]= 0.057

In[416]:=
```
iR500 = Table[ CorrelationFunction[dataWN500[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[417]:=
```
iQ500 = Table[500 * Sum[(iR500[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq500 = 0;
For[i = 1, i < 1001, i++, If[iQ500[[i]] > 41.34, iCountsq500++]];
iCountsq500
iαq500 = N[iCountsq500 / 1000]
```

Out[420]= 45

Out[421]= 0.045

In[422]:=
```
iCountsN500 = 0;
iCC500 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR500[[i]][[j]]] > 1.96 / (500^(1/2)), iCountsN500++]];
  iCC500 = Join[iCC500, {iCountsN500}];
  iCountsN500 = 0];
iCountsN500 = 0;
For[i = 1, i < 1001, i++, If[iCC500[[i]] > 3, iCountsN500++]];
iCountsN500
iαN500 = N[iCountsN500 / 1000]
```

Out[427]= 34

Out[428]= 0.034

V následujících výpočtech se to samé opakuje pro $n = 1000$. Zde jsem pro větší úsporu času rozdělil vypočet $b_{\tau,n}$ postupně pro jednotlivá τ, poté dohromady průměry a směrodatné odchylky.

In[∘]:=
```
iMeansWN1000Tst = Table[
  1000 * (Sum[dataWN1000[[Nn]][[t]]^2 * dataWN1000[[Nn]][[t + 1]]^2, {t, 1000 - 1}] /
     (Sum[dataWN1000[[Nn]][[t]]^2, {t, 1000}]) ^2), {Nn, 1000}]
```

In[∘]:=
```
iMeansWN1000Tst2 = Table[
  1000 * (Sum[dataWN1000[[Nn]][[t]]^2 * dataWN1000[[Nn]][[t + 2]]^2, {t, 1000 - 2}] /
     (Sum[dataWN1000[[Nn]][[t]]^2, {t, 1000}]) ^2), {Nn, 1000}]
```

In[∘]:=
```
iMeansWN1000Tst3 = Table[
  1000 * (Sum[dataWN1000[[Nn]][[t]]^2 * dataWN1000[[Nn]][[t + 5]]^2, {t, 1000 - 5}] /
     (Sum[dataWN1000[[Nn]][[t]]^2, {t, 1000}]) ^2), {Nn, 1000}]
```

In[∘]:=
```
iMeansWN1000Tst4 = Table[
  1000 * (Sum[dataWN1000[[Nn]][[t]]^2 * dataWN1000[[Nn]][[t + 10]]^2, {t, 1000 - 10}] /
     (Sum[dataWN1000[[Nn]][[t]]^2, {t, 1000}]) ^2), {Nn, 1000}]
```

In[∘]:=
```
iMeansWN1000Tst5 = Table[
  1000 * (Sum[dataWN1000[[Nn]][[t]]^2 * dataWN1000[[Nn]][[t + 20]]^2, {t, 1000 - 20}] /
     (Sum[dataWN1000[[Nn]][[t]]^2, {t, 1000}]) ^2), {Nn, 1000}]
```

In[∘]:=
```
iMeansWN1000 =
 Table[Mean[Join[{iMeansWN1000Tst}, {iMeansWN1000Tst2}, {iMeansWN1000Tst3},
     {iMeansWN1000Tst4}, {iMeansWN1000Tst5}][[i]]], {i, 5}]
```

Out[∘]= {0.994669, 0.995501, 0.990804, 0.990233, 0.977867}

In[∘]:=
```
iSDsWN1000 = Table[
  StandardDeviation[Join[{iMeansWN1000Tst}, {iMeansWN1000Tst2}, {iMeansWN1000Tst3},
     {iMeansWN1000Tst4}, {iMeansWN1000Tst5}][[i]]], {i, 5}]
```

Out[∘]= {0.0617332, 0.0624127, 0.062461, 0.0623193, 0.0638947}

In[153]:=
```
i1000r1 =
  Table[1000^(1/2) Abs[CorrelationFunction[dataWN1000[[Nn]], 1]], {Nn, 1000}];
iCounts1000 = 0;
For[i = 1, i < 1001, i++, If[i1000r1[[i]] > 1.96, iCounts1000++]];
iCounts1000
iα1000 = N[iCounts1000/1000]
```

Out[156]= 47

Out[157]= 0.047

In[429]:=
```
iR1000 = Table[ CorrelationFunction[dataWN1000[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[430]:=
```
iQ1000 = Table[1000 * Sum[(iR1000[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq1000 = 0;
For[i = 1, i < 1001, i++, If[iQ1000[[i]] > 41.34, iCountsq1000++]];
iCountsq1000
iαq1000 = N[iCountsq1000/1000]
iCountsN1000 = 0;
iCC1000 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR1000[[i]][[j]]] > 1.96/(1000^(1/2)), iCountsN1000++]];
  iCC1000 = Join[iCC1000, {iCountsN1000}];
  iCountsN1000 = 0];
iCountsN1000 = 0;
For[i = 1, i < 1001, i++, If[iCC1000[[i]] > 3, iCountsN1000++]];
iCountsN1000
iαN1000 = N[iCountsN1000/1000]
```

Out[433]= 49

Out[434]= 0.049

Out[440]= 44

Out[441]= 0.044

Vytvořil jsem si zde pomocné tabulky pro průměry a směrodatné odchylky $b_{\tau,n}$.

In[∘]:=
```
Grid[Prepend[Prepend[{iMeansWN50, iMeansWN100, iMeansWN500, iMeansWN1000} // Transpose,
    {"n = 50", "n = 100", "n = 500", "n = 1000"}] // Transpose,
  {"", "τ = 1", "τ = 2", "τ = 5", "τ = 10", "τ = 20"}], Frame → All]
```

Out[∘]=

	τ = 1	τ = 2	τ = 5	τ = 10	τ = 20
n = 50	0.945524	0.917407	0.868458	0.773622	0.55764
n = 100	0.973875	0.961737	0.92947	0.888113	0.773857
n = 500	0.990269	0.994688	0.98438	0.975275	0.950881
n = 1000	0.994669	0.995501	0.990804	0.990233	0.977867

In[∘]:=
```
Grid[Prepend[Prepend[{iSDsWN50, iSDsWN100, iSDsWN500, iSDsWN1000} // Transpose,
    {"n = 50", "n = 100", "n = 500", "n = 1000"}] // Transpose,
  {"", "τ = 1", "τ = 2", "τ = 5", "τ = 10", "τ = 20"}], Frame → All]
```

Out[∘]=

	τ = 1	τ = 2	τ = 5	τ = 10	τ = 20
n = 50	0.257274	0.25206	0.250074	0.241499	0.191565
n = 100	0.178799	0.183468	0.189301	0.18657	0.175435
n = 500	0.0898538	0.0866493	0.0904491	0.0853613	0.0855733
n = 1000	0.0617332	0.0624127	0.062461	0.0623193	0.0638947

Dále generuji AR(1) s bílým šumem z N(0,1) opět pomocí **RandomFunction** a funkce simulující AR(1) **ARProcess** pro různé volby jediného parametru $\varphi \in \{0.1, 0.5, 0.9\}$ a s definovaným bílým šumem z

N(0,1). Postupně generuji pro různé délky $n \in \{50,\ 100,\ 500,\ 100\}$ vždy 1000 realizací AR(1). Tato generace je společná pro kapitoly 4.1 a 4.2.

In[120]:=
```
SeedRandom[15];
dataAR = Table[RandomFunction[ARProcess[{ϕ}, 1], {0, nn}, 1000],
  {ϕ, {.1, .5, .9}}, {nn, {50, 100, 500, 1000}}]
```

Generovaná data si rozdělím do proměnných dle dané délky časové řady a dané hodnoty parametru φ.

In[122]:=
```
dataAR501 = Table[dataAR[[1]][[1]]["Values", i], {i, 1000}];
dataAR1001 = Table[dataAR[[1]][[2]]["Values", i], {i, 1000}];
dataAR5001 = Table[dataAR[[1]][[3]]["Values", i], {i, 1000}];
dataAR10001 = Table[dataAR[[1]][[4]]["Values", i], {i, 1000}];
dataAR505 = Table[dataAR[[2]][[1]]["Values", i], {i, 1000}];
dataAR1005 = Table[dataAR[[2]][[2]]["Values", i], {i, 1000}];
dataAR5005 = Table[dataAR[[2]][[3]]["Values", i], {i, 1000}];
dataAR10005 = Table[dataAR[[2]][[4]]["Values", i], {i, 1000}];
dataAR509 = Table[dataAR[[3]][[1]]["Values", i], {i, 1000}];
dataAR1009 = Table[dataAR[[3]][[2]]["Values", i], {i, 1000}];
dataAR5009 = Table[dataAR[[3]][[3]]["Values", i], {i, 1000}];
dataAR10009 = Table[dataAR[[3]][[4]]["Values", i], {i, 1000}];
```

Nyní se budou opakovat bloky stejné kódu jako u kódu pro bílý šum. Výpočet průměrů a směrodatných odchylek $b_{\tau,n}$ probíhá stejně jako u bílého šumu. Průměry a směrodatné odchylky počítám již vždy dohromady.

```
ibτnAR501 = Transpose[
   Table[50 * (Sum[dataAR501[[Nn]][[t]]^2 * dataAR501[[Nn]][[t + τ]]^2, {t, 50 - τ}] /
       (Sum[dataAR501[[Nn]][[t]]^2, {t, 50}])^2), {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];
```

In[ ]:=
```
iMeansAR501 = Table[Mean[ibτnAR501[[j]]], {j, 5}]
iSDsAR501 = Table[StandardDeviation[ibτnAR501[[j]]], {j, 5}]
```

Out[ ]= {0.954502, 0.918503, 0.866808, 0.760904, 0.56395}

Out[ ]= {0.247665, 0.251016, 0.247956, 0.232761, 0.200212}

Průběh výpočtu empirické síly testů z kapitoly 3.1 probíhá stejným principem jako výpočet empirické hladiny testů, akorát v AR(1), tedy za platnosti alternativy testů o nekorelovanosti. Začínám pro $n = 50$ a $\varphi = 0.1$.

In[158]:=
```
i501r1 = Table[50^(1/2) Abs[CorrelationFunction[dataAR501[[Nn]], 1]], {Nn, 1000}];
iCounts501 = 0;
For[i = 1, i < 1001, i++, If[i501r1[[i]] > 1.96, iCounts501++]];
iCounts501
iβ501 = N[iCounts501/1000]
```

Out[161]= 52

Out[162]= 0.052

In[173]:=
```
iR501 = Table[ CorrelationFunction[dataAR501[[Nn]], k], {Nn, 1000}, {k, 28}];
```

```
In[174]:= iQ501 = Table[50 * Sum[(iR501[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
          iCountsq501 = 0;
          For[i = 1, i < 1001, i++, If[iQ501[[i]] > 41.34, iCountsq501++]];
          iCountsq501
          iβq501 = N[iCountsq501 / 1000]

Out[177]= 11

Out[178]= 0.011

In[179]:= iCountsN501 = 0;
          iCC501 = {};
          For[i = 1, i < 1001, i++,
            For[j = 1, j < 29, j++, If[Abs[iR501[[i]][[j]]] > 1.96 / (50^(1/2)), iCountsN501++]];
            iCC501 = Join[iCC501, {iCountsN501}];
            iCountsN501 = 0];
          iCountsN501 = 0;
          For[i = 1, i < 1001, i++, If[iCC501[[i]] > 3, iCountsN501++]];
          iCountsN501
          iβN501 = N[iCountsN501 / 1000]

Out[184]= 6

Out[185]= 0.006
```

Další část kódu je pro $n = 50$ a $\varphi = 0.5$.

```
ibτnAR505 = Transpose[
   Table[50 * (Sum[dataAR505[[Nn]][[t]]^2 * dataAR505[[Nn]][[t + τ]]^2, {t, 50 - τ}] /
       (Sum[dataAR505[[Nn]][[t]]^2, {t, 50}])^2), {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];

In[∘]:= iMeansAR505 = Table[Mean[ibτnAR505[[j]]], {j, 5}]
        iSDsAR505 = Table[StandardDeviation[ibτnAR505[[j]]], {j, 5}]

Out[∘]= {1.30101, 0.991703, 0.840785, 0.748611, 0.573736}

Out[∘]= {0.373592, 0.269254, 0.241185, 0.236358, 0.199649}

In[163]:= i505r1 = Table[50^(1/2) Abs[CorrelationFunction[dataAR505[[Nn]], 1]], {Nn, 1000}];
          iCounts505 = 0;
          For[i = 1, i < 1001, i++, If[i505r1[[i]] > 1.96, iCounts505++]];
          iCounts505
          iβ505 = N[iCounts505 / 1000]

Out[166]= 890

Out[167]= 0.89

In[186]:= iR505 = Table[ CorrelationFunction[dataAR505[[Nn]], k], {Nn, 1000}, {k, 28}];

In[187]:= iQ505 = Table[50 * Sum[(iR505[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
          iCountsq505 = 0;
          For[i = 1, i < 1001, i++, If[iQ505[[i]] > 41.34, iCountsq505++]];
          iCountsq505
          iβq505 = N[iCountsq505 / 1000]

Out[190]= 357

Out[191]= 0.357
```

In[192]:=
```
iCountsN505 = 0;
iCC505 = {};
For[i = 1, i < 1001, i++,
  For[j = 1, j < 29, j++, If[Abs[iR505[[i]][[j]]] > 1.96 / (50^(1/2)), iCountsN505++]];
  iCC505 = Join[iCC505, {iCountsN505}];
  iCountsN505 = 0];
iCountsN505 = 0;
For[i = 1, i < 1001, i++, If[iCC505[[i]] > 3, iCountsN505++]];
iCountsN505
iβN505 = N[iCountsN505 / 1000]
```

Out[197]= 195

Out[198]= 0.195

Další část kódu je pro $n = 50$ a $\varphi = 0.9$.

```
ibτnAR509 = Transpose[
   Table[50 * (Sum[dataAR509[[Nn]][[t]]^2 * dataAR509[[Nn]][[t + τ]]^2, {t, 50 - τ}] /
       (Sum[dataAR509[[Nn]][[t]]^2, {t, 50}])^2), {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];
```

In[∘]:=
```
iMeansAR509 = Table[Mean[ibτnAR509[[j]]], {j, 5}]
iSDsAR509 = Table[StandardDeviation[ibτnAR509[[j]]], {j, 5}]
```

Out[∘]= {1.90885, 1.55139, 0.997371, 0.682355, 0.476329}

Out[∘]= {0.477747, 0.428572, 0.30402, 0.247029, 0.213945}

In[168]:=
```
i509r1 = Table[50^(1/2) Abs[CorrelationFunction[dataAR509[[Nn]], 1]], {Nn, 1000}];
iCounts509 = 0;
For[i = 1, i < 1001, i++, If[i509r1[[i]] > 1.96, iCounts509++]];
iCounts509
iβ509 = N[iCounts509 / 1000]
```

Out[171]= 1000

Out[172]= 1.

In[199]:=
```
iR509 = Table[ CorrelationFunction[dataAR509[[Nn]], k], {Nn, 1000}, {k, 28}];
```

```
In[200]:= iQ509 = Table[50 * Sum[(iR509[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq509 = 0;
For[i = 1, i < 1001, i++, If[iQ509[[i]] > 41.34, iCountsq509++]];
iCountsq509
iβq509 = N[iCountsq509 / 1000]
iCountsN509 = 0;
iCC509 = {};
For[i = 1, i < 1001, i++,
  For[j = 1, j < 29, j++, If[Abs[iR509[[i]][[j]]] > 1.96 / (50^(1 / 2)), iCountsN509++]];
  iCC509 = Join[iCC509, {iCountsN509}];
  iCountsN509 = 0];
iCountsN509 = 0;
For[i = 1, i < 1001, i++, If[iCC509[[i]] > 3, iCountsN509++]];
iCountsN509
iβN509 = N[iCountsN509 / 1000]

Out[203]= 974

Out[204]= 0.974

Out[210]= 888

Out[211]= 0.888
```

Další část kódu je pro $n = 100$ a $\varphi = 0.1$.

```
ibτnAR1001 =
  Transpose[Table[100 * (Sum[dataAR1001[[Nn]][[t]]^2 * dataAR1001[[Nn]][[t + τ]]^2,
        {t, 100 - τ}] / (Sum[dataAR1001[[Nn]][[t]]^2, {t, 100}])^2),
    {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];

In[∘]:= iMeansAR1001 = Table[Mean[ibτnAR1001[[j]]], {j, 5}]
iSDsAR1001 = Table[StandardDeviation[ibτnAR1001[[j]]], {j, 5}]

Out[∘]= {0.988962, 0.960856, 0.938603, 0.878908, 0.775357}

Out[∘]= {0.179956, 0.185966, 0.187082, 0.187379, 0.184241}

In[212]:= i1001r1 = Table[100^(1 / 2) Abs[CorrelationFunction[dataAR1001[[Nn]], 1]], {Nn, 1000}];
iCounts1001 = 0;
For[i = 1, i < 1001, i++, If[i1001r1[[i]] > 1.96, iCounts1001++]];
iCounts1001
iβ1001 = N[iCounts1001 / 1000]

Out[215]= 131

Out[216]= 0.131

In[217]:= iR1001 = Table[ CorrelationFunction[dataAR1001[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[218]:=
```
iQ1001 = Table[100 * Sum[(iR1001[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq1001 = 0;
For[i = 1, i < 1001, i++, If[iQ1001[[i]] > 41.34, iCountsq1001++]];
iCountsq1001
iβq1001 = N[iCountsq1001 / 1000]
iCountsN1001 = 0;
iCC1001 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR1001[[i]][[j]]] > 1.96 / (100^(1/2)), iCountsN1001++]];
  iCC1001 = Join[iCC1001, {iCountsN1001}];
  iCountsN1001 = 0];
iCountsN1001 = 0;
For[i = 1, i < 1001, i++, If[iCC1001[[i]] > 3, iCountsN1001++]];
iCountsN1001
iβN1001 = N[iCountsN1001 / 1000]
```

Out[221]= 32

Out[222]= 0.032

Out[228]= 33

Out[229]= 0.033

Další část kódu je pro $n = 100$ a $\varphi = 0.5$.

```
ibτnAR1005 =
  Transpose[Table[100 * (Sum[dataAR1005[[Nn]][[t]]^2 * dataAR1005[[Nn]][[t + τ]]^2,
        {t, 100 - τ}] / (Sum[dataAR1005[[Nn]][[t]]^2, {t, 100}])^2),
    {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];
```

In[ ]:=
```
iMeansAR1005 = Table[Mean[ibτnAR1005[[j]]], {j, 5}]
iSDsAR1005 = Table[StandardDeviation[ibτnAR1005[[j]]], {j, 5}]
```

Out[ ]= {1.43658, 1.06138, 0.923646, 0.87031, 0.774129}

Out[ ]= {0.321874, 0.229341, 0.190829, 0.192483, 0.182181}

In[230]:=
```
i1005r1 = Table[100^(1/2) Abs[CorrelationFunction[dataAR1005[[Nn]], 1]], {Nn, 1000}];
iCounts1005 = 0;
For[i = 1, i < 1001, i++, If[i1005r1[[i]] > 1.96, iCounts1005++]];
iCounts1005
iβ1005 = N[iCounts1005 / 1000]
```

Out[233]= 998

Out[234]= 0.998

In[235]:=
```
iR1005 = Table[ CorrelationFunction[dataAR1005[[Nn]], k], {Nn, 1000}, {k, 28}];
```

```
In[236]:= iQ1005 = Table[100 * Sum[(iR1005[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq1005 = 0;
For[i = 1, i < 1001, i++, If[iQ1005[[i]] > 41.34, iCountsq1005++]];
iCountsq1005
iβq1005 = N[iCountsq1005 / 1000]
iCountsN1005 = 0;
iCC1005 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
    If[Abs[iR1005[[i]][[j]]] > 1.96 / (100^(1/2)), iCountsN1005++]];
  iCC1005 = Join[iCC1005, {iCountsN1005}];
  iCountsN1005 = 0];
iCountsN1005 = 0;
For[i = 1, i < 1001, i++, If[iCC1005[[i]] > 3, iCountsN1005++]];
iCountsN1005
iβN1005 = N[iCountsN1005 / 1000]

Out[239]= 839

Out[240]= 0.839

Out[246]= 445

Out[247]= 0.445
```

Další část kódu je pro $n = 100$ a $\varphi = 0.9$.

```
ibτnAR1009 =
  Transpose[Table[100 * (Sum[dataAR1009[[Nn]][[t]]^2 * dataAR1009[[Nn]][[t + τ]]^2,
        {t, 100 - τ}] / (Sum[dataAR1009[[Nn]][[t]]^2, {t, 100}])^2),
    {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];

In[ ]:= iMeansAR1009 = Table[Mean[ibτnAR1009[[j]]], {j, 5}]
iSDsAR1009 = Table[StandardDeviation[ibτnAR1009[[j]]], {j, 5}]

Out[ ]= {2.17709, 1.83531, 1.24135, 0.889489, 0.707241}

Out[ ]= {0.487272, 0.465297, 0.365304, 0.261976, 0.240379}

In[248]:= i1009r1 = Table[100^(1/2) Abs[CorrelationFunction[dataAR1009[[Nn]], 1]], {Nn, 1000}];
iCounts1009 = 0;
For[i = 1, i < 1001, i++, If[i1009r1[[i]] > 1.96, iCounts1009++]];
iCounts1009
iβ1009 = N[iCounts1009 / 1000]

Out[251]= 1000

Out[252]= 1.

In[253]:= iR1009 = Table[ CorrelationFunction[dataAR1009[[Nn]], k], {Nn, 1000}, {k, 28}];
```

```
In[254]:= iQ1009 = Table[100 * Sum[(iR1009[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq1009 = 0;
For[i = 1, i < 1001, i++, If[iQ1009[[i]] > 41.34, iCountsq1009++]];
iCountsq1009
iβq1009 = N[iCountsq1009 / 1000]
iCountsN1009 = 0;
iCC1009 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR1009[[i]][[j]]] > 1.96 / (100^(1 / 2)), iCountsN1009++]];
  iCC1009 = Join[iCC1009, {iCountsN1009}];
  iCountsN1009 = 0];
iCountsN1009 = 0;
For[i = 1, i < 1001, i++, If[iCC1009[[i]] > 3, iCountsN1009++]];
iCountsN1009
iβN1009 = N[iCountsN1009 / 1000]

Out[257]= 1000

Out[258]= 1.

Out[264]= 992

Out[265]= 0.992
```

Další část kódu je pro $n = 500$ a $\varphi = 0.1$.

```
ibτnAR5001 =
  Transpose[Table[500 * (Sum[dataAR5001[[Nn]][[t]]^2 * dataAR5001[[Nn]][[t + τ]]^2,
        {t, 500 - τ}] / (Sum[dataAR5001[[Nn]][[t]]^2, {t, 500}])^2),
     {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];
iMeansAR5001 = Table[Mean[ibτnAR5001[[j]]], {j, 5}]
iSDsAR5001 = Table[StandardDeviation[ibτnAR5001[[j]]], {j, 5}]

Out[∘]= {1.01019, 0.994367, 0.984222, 0.975879, 0.951141}

Out[∘]= {0.0903503, 0.0875923, 0.0899861, 0.0847785, 0.0850082}

In[266]:= i5001r1 = Table[500^(1 / 2) Abs[CorrelationFunction[dataAR5001[[Nn]], 1]], {Nn, 1000}];
iCounts5001 = 0;
For[i = 1, i < 1001, i++, If[i5001r1[[i]] > 1.96, iCounts5001++]];
iCounts5001
iβ5001 = N[iCounts5001 / 1000]

Out[269]= 563

Out[270]= 0.563

In[271]:= iR5001 = Table[ CorrelationFunction[dataAR5001[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[272]:=
```
iQ5001 = Table[500 * Sum[(iR5001[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq5001 = 0;
For[i = 1, i < 1001, i++, If[iQ5001[[i]] > 41.34, iCountsq5001++]];
iCountsq5001
iβq5001 = N[iCountsq5001 / 1000]
iCountsN5001 = 0;
iCC5001 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR5001[[i]][[j]]] > 1.96 / (500^(1/2)), iCountsN5001++]];
  iCC5001 = Join[iCC5001, {iCountsN5001}];
  iCountsN5001 = 0];
iCountsN5001 = 0;
For[i = 1, i < 1001, i++, If[iCC5001[[i]] > 3, iCountsN5001++]];
iCountsN5001
iβN5001 = N[iCountsN5001 / 1000]
```

Out[275]= 140

Out[276]= 0.14

Out[282]= 104

Out[283]= 0.104

Další část kódu je pro $n = 500$ a $\varphi = 0.5$.

In[134]:=
```
ibτnAR5005 =
  Transpose[Table[500 * (Sum[dataAR5005[[Nn]][[t]]^2 * dataAR5005[[Nn]][[t + τ]]^2,
        {t, 500 - τ}] / (Sum[dataAR5005[[Nn]][[t]]^2, {t, 500}])^2),
    {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];
```

In[135]:=
```
iMeansAR5005 = Table[Mean[ibτnAR5005[[j]]], {j, 5}]
iSDsAR5005 = Table[StandardDeviation[ibτnAR5005[[j]]], {j, 5}]
```

Out[135]= {1.47994, 1.10682, 0.985072, 0.974801, 0.951211}

Out[136]= {0.166447, 0.12171, 0.0926483, 0.0938595, 0.0917047}

In[284]:=
```
i5005r1 = Table[500^(1/2) Abs[CorrelationFunction[dataAR5005[[Nn]], 1]], {Nn, 1000}];
iCounts5005 = 0;
For[i = 1, i < 1001, i++, If[i5005r1[[i]] > 1.96, iCounts5005++]];
iCounts5005
iβ5005 = N[iCounts5005 / 1000]
```

Out[287]= 1000

Out[288]= 1.

In[289]:=
```
iR5005 = Table[ CorrelationFunction[dataAR5005[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[290]:=
```
iQ5005 = Table[500 * Sum[(iR5005[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq5005 = 0;
For[i = 1, i < 1001, i++, If[iQ5005[[i]] > 41.34, iCountsq5005++]];
iCountsq5005
iβq5005 = N[iCountsq5005 / 1000]
iCountsN5005 = 0;
iCC5005 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR5005[[i]][[j]]] > 1.96 / (500^(1/2)), iCountsN5005++]];
  iCC5005 = Join[iCC5005, {iCountsN5005}];
  iCountsN5005 = 0];
iCountsN5005 = 0;
For[i = 1, i < 1001, i++, If[iCC5005[[i]] > 3, iCountsN5005++]];
iCountsN5005
iβN5005 = N[iCountsN5005 / 1000]
```

Out[293]= 1000

Out[294]= 1.

Out[300]= 802

Out[301]= 0.802

Další část kódu je pro $n = 500$ a $\varphi = 0.9$.

```
ibτnAR5009 =
  Transpose[Table[500 * (Sum[dataAR5009[[Nn]][[t]]^2 * dataAR5009[[Nn]][[t + τ]]^2,
        {t, 500 - τ}] / (Sum[dataAR5009[[Nn]][[t]]^2, {t, 500}])^2),
    {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];
```

In[ ]:=
```
iMeansAR5009 = Table[Mean[ibτnAR5009[[j]]], {j, 5}]
iSDsAR5009 = Table[StandardDeviation[ibτnAR5009[[j]]], {j, 5}]
```

Out[ ]= {2.50066, 2.18413, 1.57562, 1.15078, 0.945561}

Out[ ]= {0.399788, 0.384767, 0.326107, 0.226685, 0.168622}

In[302]:=
```
i5009r1 = Table[500^(1/2) Abs[CorrelationFunction[dataAR5009[[Nn]], 1]], {Nn, 1000}];
iCounts5009 = 0;
For[i = 1, i < 1001, i++, If[i5009r1[[i]] > 1.96, iCounts5009++]];
iCounts5009
iβ5009 = N[iCounts5009 / 1000]
```

Out[305]= 1000

Out[306]= 1.

In[307]:=
```
iR5009 = Table[ CorrelationFunction[dataAR5009[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[308]:=
```
iQ5009 = Table[500 * Sum[(iR5009[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq5009 = 0;
For[i = 1, i < 1001, i++, If[iQ5009[[i]] > 41.34, iCountsq5009++]];
iCountsq5009
iβq5009 = N[iCountsq5009 / 1000]
iCountsN5009 = 0;
iCC5009 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR5009[[i]][[j]]] > 1.96 / (500^(1/2)), iCountsN5009++]];
  iCC5009 = Join[iCC5009, {iCountsN5009}];
  iCountsN5009 = 0];
iCountsN5009 = 0;
For[i = 1, i < 1001, i++, If[iCC5009[[i]] > 3, iCountsN5009++]];
iCountsN5009
iβN5009 = N[iCountsN5009 / 1000]
```

Out[311]= 1000

Out[312]= 1.

Out[318]= 1000

Out[319]= 1.

Další část kódu je pro $n = 1000$ a $\varphi = 0.1$.

```
ibτnAR10001 =
  Transpose[Table[1000 * (Sum[dataAR10001[[Nn]][[t]]^2 * dataAR10001[[Nn]][[t + τ]]^2,
        {t, 1000 - τ}] / (Sum[dataAR10001[[Nn]][[t]]^2, {t, 1000}])^2),
    {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];
```

In[∘]:=
```
iMeansAR10001 = Table[Mean[ibτnAR10001[[j]]], {j, 5}]
iSDsAR10001 = Table[StandardDeviation[ibτnAR10001[[j]]], {j, 5}]
```

Out[∘]= {1.01399, 0.995317, 0.990783, 0.990978, 0.977359}

Out[∘]= {0.0643933, 0.0625169, 0.0625071, 0.0618711, 0.0637237}

In[320]:=
```
i10001r1 =
  Table[1000^(1/2) Abs[CorrelationFunction[dataAR10001[[Nn]], 1]], {Nn, 1000}];
iCounts10001 = 0;
For[i = 1, i < 1001, i++, If[i10001r1[[i]] > 1.96, iCounts10001++]];
iCounts10001
iα10001 = N[iCounts10001 / 1000]
```

Out[323]= 874

Out[324]= 0.874

In[325]:=
```
iR10001 = Table[ CorrelationFunction[dataAR10001[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[326]:=
```
iQ10001 = Table[1000 * Sum[(iR10001[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq10001 = 0;
For[i = 1, i < 1001, i++, If[iQ10001[[i]] > 41.34, iCountsq10001++]];
iCountsq10001
iαq10001 = N[iCountsq10001 / 1000]
iCountsN10001 = 0;
iCC10001 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR10001[[i]][[j]]] > 1.96 / (1000^(1/2)), iCountsN10001++]];
  iCC10001 = Join[iCC10001, {iCountsN10001}];
  iCountsN10001 = 0];
iCountsN10001 = 0;
For[i = 1, i < 1001, i++, If[iCC10001[[i]] > 3, iCountsN10001++]];
iCountsN10001
iαN10001 = N[iCountsN10001 / 1000]
```

Out[329]= 314

Out[330]= 0.314

Out[336]= 131

Out[337]= 0.131

Další část kódu je pro $n = 1000$ a $\varphi = 0.5$.

```
ibτnAR10005 =
  Transpose[Table[1000 * (Sum[dataAR10005[[Nn]][[t]]^2 * dataAR10005[[Nn]][[t + τ]]^2,
        {t, 1000 - τ}] / (Sum[dataAR10005[[Nn]][[t]]^2, {t, 1000}])^2),
    {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];
```

In[ ]:=
```
iMeansAR10005 = Table[Mean[ibτnAR10005[[j]]], {j, 5}]
iSDsAR10005 = Table[StandardDeviation[ibτnAR10005[[j]]], {j, 5}]
```

Out[ ]= {1.49377, 1.11569, 0.990841, 0.986501, 0.976337}

Out[ ]= {0.116541, 0.0843373, 0.0693124, 0.0674994, 0.069452}

In[338]:=
```
i10005r1 =
  Table[1000^(1/2) Abs[CorrelationFunction[dataAR10005[[Nn]], 1]], {Nn, 1000}];
iCounts10005 = 0;
For[i = 1, i < 1001, i++, If[i10005r1[[i]] > 1.96, iCounts10005++]];
iCounts10005
iα10005 = N[iCounts10005 / 1000]
```

Out[341]= 1000

Out[342]= 1.

In[343]:=
```
iR10005 = Table[ CorrelationFunction[dataAR10005[[Nn]], k], {Nn, 1000}, {k, 28}];
```

```
In[344]:= iQ10005 = Table[1000 * Sum[(iR10005[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq10005 = 0;
For[i = 1, i < 1001, i++, If[iQ10005[[i]] > 41.34, iCountsq10005++]];
iCountsq10005
iαq10005 = N[iCountsq10005 / 1000]
iCountsN10005 = 0;
iCC10005 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR10005[[i]][[j]]] > 1.96 / (1000^(1 / 2)), iCountsN10005++]];
  iCC10005 = Join[iCC10005, {iCountsN10005}];
  iCountsN10005 = 0];
iCountsN10005 = 0;
For[i = 1, i < 1001, i++, If[iCC10005[[i]] > 3, iCountsN10005++]];
iCountsN10005
iαN10005 = N[iCountsN10005 / 1000]

Out[347]= 1000

Out[348]= 1.

Out[354]= 888

Out[355]= 0.888
```

Další část kódu je pro $n = 1000$ a $\varphi = 0.9$.

```
ibτnAR10009 =
  Transpose[Table[1000 * (Sum[dataAR10009[[Nn]][[t]]^2 * dataAR10009[[Nn]][[t + τ]]^2,
        {t, 1000 - τ}] / (Sum[dataAR10009[[Nn]][[t]]^2, {t, 1000}])^2),
     {Nn, 1000}, {τ, {1, 2, 5, 10, 20}}]];

In[ ]:= iMeansAR10009 = Table[Mean[ibτnAR10009[[j]]], {j, 5}]
iSDsAR10009 = Table[StandardDeviation[ibτnAR10009[[j]]], {j, 5}]

Out[ ]= {2.55613, 2.24464, 1.62841, 1.18211, 0.98177}

Out[ ]= {0.297956, 0.287397, 0.245318, 0.177636, 0.133141}

In[356]:= i10009r1 =
  Table[1000^(1 / 2) Abs[CorrelationFunction[dataAR10009[[Nn]], 1]], {Nn, 1000}];
iCounts10009 = 0;
For[i = 1, i < 1001, i++, If[i10009r1[[i]] > 1.96, iCounts10009++]];
iCounts10009
iα10009 = N[iCounts10009 / 1000]

Out[359]= 1000

Out[360]= 1.

In[361]:= iR10009 = Table[ CorrelationFunction[dataAR10009[[Nn]], k], {Nn, 1000}, {k, 28}];
```

In[362]:=
```
iQ10009 = Table[1000 * Sum[(iR10009[[Nn]][[k]])^2, {k, 28}], {Nn, 1000}];
iCountsq10009 = 0;
For[i = 1, i < 1001, i++, If[iQ10009[[i]] > 41.34, iCountsq10009++]];
iCountsq10009
iαq10009 = N[iCountsq10009 / 1000]
iCountsN10009 = 0;
iCC10009 = {};
For[i = 1, i < 1001, i++, For[j = 1, j < 29, j++,
   If[Abs[iR10009[[i]][[j]]] > 1.96 / (1000^(1/2)), iCountsN10009++]];
  iCC10009 = Join[iCC10009, {iCountsN10009}];
  iCountsN10009 = 0];
iCountsN10009 = 0;
For[i = 1, i < 1001, i++, If[iCC10009[[i]] > 3, iCountsN10009++]];
iCountsN10009
iαN10009 = N[iCountsN10009 / 1000]
```

Out[365]= 1000

Out[366]= 1.

Out[372]= 1000

Out[373]= 1.

Vytvořil jsem pomocné tabulky pro směrodatné odchylky a průměry $b_{\tau,n}$ v AR(1).

In[ ]:=
```
Grid[Prepend[
  Prepend[Prepend[{iSDsAR501, iSDsAR505, iSDsAR509, iSDsAR1001, iSDsAR1005, iSDsAR1009,
       iSDsAR5001, iSDsAR5005, iSDsAR5009, iSDsAR10001, iSDsAR10005, iSDsAR10009} //
      Transpose, {"n = 50", "n = 100", "n = 500", "n = 1000", "n = 50",
      "n = 100", "n = 500", "n = 1000", "n = 50", "n = 100", "n = 500", "n = 1000"}],
    {"ϕ = 0,1", SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,5",
     SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,9",
     SpanFromAbove, SpanFromAbove, SpanFromAbove}] // Transpose,
  {"", "", "τ = 1", "τ = 2", "τ = 5", "τ = 10", "τ = 20"}], Frame → All]
```

Out[ ]=

		τ = 1	τ = 2	τ = 5	τ = 10	τ = 20
ϕ = 0,1	n = 50	0.247665	0.251016	0.247956	0.232761	0.200212
	n = 100	0.373592	0.269254	0.241185	0.236358	0.199649
	n = 500	0.477747	0.428572	0.30402	0.247029	0.213945
	n = 1000	0.179956	0.185966	0.187082	0.187379	0.184241
ϕ = 0,5	n = 50	0.321874	0.229341	0.190829	0.192483	0.182181
	n = 100	0.487272	0.465297	0.365304	0.261976	0.240379
	n = 500	0.0903503	0.0875923	0.0899861	0.0847785	0.0850082
	n = 1000	0.166447	0.12171	0.0926483	0.0938595	0.0917047
ϕ = 0,9	n = 50	0.399788	0.384767	0.326107	0.226685	0.168622
	n = 100	0.0643933	0.0625169	0.0625071	0.0618711	0.0637237
	n = 500	0.116541	0.0843373	0.0693124	0.0674994	0.069452
	n = 1000	0.297956	0.287397	0.245318	0.177636	0.133141

In[∘]:=
```
Grid[Prepend[
  Prepend[Prepend[{iMeansAR501, iMeansAR505, iMeansAR509, iMeansAR1001, iMeansAR1005,
       iMeansAR1009, iMeansAR5001, iMeansAR5005, iMeansAR5009,
       iMeansAR10001, iMeansAR10005, iMeansAR10009} // Transpose,
     {"n = 50", "n = 100", "n = 500", "n = 1000", "n = 50", "n = 100",
      "n = 500", "n = 1000", "n = 50", "n = 100", "n = 500", "n = 1000"}],
    {"ϕ = 0,1", SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,5",
     SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,9",
     SpanFromAbove, SpanFromAbove, SpanFromAbove}] // Transpose,
  {"", "", "τ = 1", "τ = 2", "τ = 5", "τ = 10", "τ = 20"}], Frame → All]
```

Out[∘]=

		τ = 1	τ = 2	τ = 5	τ = 10	τ = 20
ϕ = 0,1	n = 50	0.954502	0.918503	0.866808	0.760904	0.56395
	n = 100	1.30101	0.991703	0.840785	0.748611	0.573736
	n = 500	1.90885	1.55139	0.997371	0.682355	0.476329
	n = 1000	0.988962	0.960856	0.938603	0.878908	0.775357
ϕ = 0,5	n = 50	1.43658	1.06138	0.923646	0.87031	0.774129
	n = 100	2.17709	1.83531	1.24135	0.889489	0.707241
	n = 500	1.01019	0.994367	0.984222	0.975879	0.951141
	n = 1000	1.47994	1.10682	0.985072	0.974801	0.951211
ϕ = 0,9	n = 50	2.50066	2.18413	1.57562	1.15078	0.945561
	n = 100	1.01399	0.995317	0.990783	0.990978	0.977359
	n = 500	1.49377	1.11569	0.990841	0.986501	0.976337
	n = 1000	2.55613	2.24464	1.62841	1.18211	0.98177

Pomocná tabulka s empirickými hladinami testu s testovou statistikou (3.9) pro různé délky časové řady.

In[380]:=
```
Grid[Prepend[{{iα50}, {iα100}, {iα500}, {iα1000}} // Transpose,
  {"n = 50", "n = 100", "n = 500", "n = 1000"}], Frame → All]
```

Out[380]=

n = 50	n = 100	n = 500	n = 1000
0.041	0.044	0.057	0.047

Pomocná tabulka s empirickými silami testu s testovou statistikou (3.9) pro různé parametry φ a délky časové řady.

In[389]:=
```
Grid[Prepend[
  Transpose[Prepend[Prepend[{{iβ501, iβ1001, iβ5001, iα10001, iβ505, iβ1005, iβ5005,
       iα10005, iβ509, iβ1009, iβ5009, iα10009}},
     {"n = 50", "n = 100", "n = 500", "n = 1000", "n = 50", "n = 100",
      "n = 500", "n = 1000", "n = 50", "n = 100", "n = 500", "n = 1000"}],
    {"ϕ = 0,1", SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,5",
     SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,9", SpanFromAbove,
     SpanFromAbove, SpanFromAbove}]], {"", "", "β"}], Frame → All]
```

Out[389]=

		β
ϕ = 0,1	n = 50	0.052
	n = 100	0.131
	n = 500	0.563
	n = 1000	0.874
ϕ = 0,5	n = 50	0.89
	n = 100	0.998
	n = 500	1.
	n = 1000	1.
ϕ = 0,9	n = 50	1.
	n = 100	1.
	n = 500	1.
	n = 1000	1.

Pomocná tabulka s empirickými hladinami testu s testovou statistikou (3.11) pro různé délky časové řady.

In[442]:=
```
Grid[Prepend[{{iαq50}, {iαq100}, {iαq500}, {iαq1000}} // Transpose,
  {"n = 50", "n = 100", "n = 500", "n = 1000"}], Frame → All]
```

Out[442]=

n = 50	n = 100	n = 500	n = 1000
0.009	0.019	0.045	0.049

Pomocná tabulka s empirickými silami testu s testovou statistikou (3.11) pro různé parametry φ a délky časové řady.

In[443]:=
```
Grid[Prepend[
  Transpose[Prepend[Prepend[{{iβq501, iβq1001, iβq5001, iαq10001, iβq505, iβq1005,
       iβq5005, iαq10005, iβq509, iβq1009, iβq5009, iαq10009}},
     {"n = 50", "n = 100", "n = 500", "n = 1000", "n = 50", "n = 100",
      "n = 500", "n = 1000", "n = 50", "n = 100", "n = 500", "n = 1000"}],
    {"ϕ = 0,1", SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,5",
     SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,9", SpanFromAbove,
     SpanFromAbove, SpanFromAbove}]], {"", "", "β"}], Frame → All]
```

Out[443]=

		β
ϕ = 0,1	n = 50	0.011
	n = 100	0.032
	n = 500	0.14
	n = 1000	0.314
ϕ = 0,5	n = 50	0.357
	n = 100	0.839
	n = 500	1.
	n = 1000	1.
ϕ = 0,9	n = 50	0.974
	n = 100	1.
	n = 500	1.
	n = 1000	1.

Pomocná tabulka s empirickými hladinami testu s testovou statistikou (3.10) pro různé délky časové řady.

In[444]:=
```
Grid[Prepend[{{iαN50}, {iαN100}, {iαN500}, {iαN1000}} // Transpose,
  {"n = 50", "n = 100", "n = 500", "n = 1000"}], Frame → All]
```

Out[444]=

n = 50	n = 100	n = 500	n = 1000
0.012	0.022	0.034	0.044

Pomocná tabulka s empirickými silami testu (3.10) pro různé parametry φ a délky časové řady.

In[445]:=
```
Grid[Prepend[
  Transpose[Prepend[Prepend[{{iβN501, iβN1001, iβN5001, iαN10001, iβN505, iβN1005,
       iβN5005, iαN10005, iβN509, iβN1009, iβN5009, iαN10009}},
     {"n = 50", "n = 100", "n = 500", "n = 1000", "n = 50", "n = 100",
      "n = 500", "n = 1000", "n = 50", "n = 100", "n = 500", "n = 1000"}],
    {"ϕ = 0,1", SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,5",
     SpanFromAbove, SpanFromAbove, SpanFromAbove, "ϕ = 0,9", SpanFromAbove,
     SpanFromAbove, SpanFromAbove}]], {"", "", "β"}], Frame → All]
```

Out[445]=

		β
ϕ = 0,1	n = 50	0.006
	n = 100	0.033
	n = 500	0.104
	n = 1000	0.131
ϕ = 0,5	n = 50	0.195
	n = 100	0.445
	n = 500	0.802
	n = 1000	0.888
ϕ = 0,9	n = 50	0.888
	n = 100	0.992
	n = 500	1.
	n = 1000	1.