



Transcript for Session 022

Listen to the podcast session, see resources & links:

<http://chandoo.org/session22/>

Transcript:

Hey podcast listeners. Welcome to session 22 of chandoo.org podcast. This podcast is aimed to make you awesome in data analysis, charting, dashboards and VBA using Microsoft Excel.

Thank you so much for joining me in this episode. I am really glad to meet you again after a very successful, fun and awesome trip to USA. As you might remember, I had been talking about my trip to Houston and Dallas to conduct a couple of master classes and attending an Excel conference. Let me tell you this - my trip has been incredibly fun. I met a lot of interesting people and I had so much fun doing my first ever conference speaking in USA. I got a lot of positive feedback from the master class attendees. A couple of them gave me really fun, glowing and honest testimonials on videos. We had about 25 people attend the classes in Houston. Most of them really enjoyed the class. There were a couple of people who had a lot of prior experience in Excel and were expecting a little more in-depth material. But when you conduct a class where people have different skill levels, you have to strike a good medium balance. That's why a couple of people were left high and dry! But, overall, it was really good. I made lots of new friends and I learnt a couple of really interesting techniques of using Excel well.

After that I went to Dallas. It was really fun. It was for about 3 nights. I stayed in a very beautiful and super expensive place called the Four Seasons resort in Dallas. The whole conference was happening there. It's a picture perfect place. When I would look out of my window, I would see a huge golf course there with perfect green lawns and people playing golf all day. The conference itself was really so much fun. The BizNet people who were running the conference are so passionate about Excel and every day I would meet people who are as stoked as I am about Excel. We had lots of fun. Even when we were having beer, wine or playing some silly games, Excel was always at the back of our minds. So we could instantly connect with each other and share experiences and stories. As far as my sessions go, I had about 80 people attending them. Even though I didn't read the feedback because it was directly given to BizNet, I could collect from talking to the attendees as well as the BizNet employees that everybody loved my classes, sessions and presentations. They're certainly looking forward to seeing me again next year. Hopefully I'll get to go back next year too. Maybe I'll take my wife, Jyotsana, and my kids so that they can also experience the beauty and meet all these nice, awesome people. That's a little bit about my conference experience. Maybe I will share the longer version of this conference story along with some pictures in a blog post sometime soon.

For now, let us talk a little bit about macros. **What is a macro?** Quite often you hear people saying, "Hey, why don't you use a macro for this?" Or, "Why don't you write a VBA program for this?" This would leave me scratching my head early on in my career. In fact, back when I was doing my MBA from 2004-2006, around the end of the first year we had a guest lecture by one of our seniors. Apart from the regular classes, the seniors or people working in an industry would sometimes visit our campus and



conduct a guest lecture. One afternoon, I was done with my classes, I had finished my lunch and I was about to go and crash in my room for some well-deserved sleep. You know how these MBA classes are - really hectic, lots of workload and you have to keep up with a whole lot of things in order to stay at a bare minimum level of academic performance. I was really tired that day and was about to crash, but my friend told me, "Hey, we have this guest lecture on VBA. It helps you do a lot of powerful things with Excel. I think you should attend it." At that time my Excel skills were really mediocre. I could open up Excel, punch in some numbers and add them up. But, I wasn't using it for anything as serious as I do today. I would say that I was less than a beginner at that time. So, I was really keen to figure out what VBA was. The other motivation for me was that I was a computer science programmer (I'm a computer engineer) and I had learnt in my school and college days how to program using Basic and Visual Basic. So, I knew those two things. And, VBA seemed like a natural extension to it. So I went to the session even though I was incredibly sleepy and least interested in what the session was going to be. At the back of my head I was thinking that pretty soon I would be done with my MBA and would go and join a company as a Manager or Assistant Manager. I didn't feel like I would need to write programs anymore because I would be managing rather than doing things! I know it sounds silly but that's the kind of thing you tend to think about when you're in B-school and with all your heightened imaginations and feelings about the real world. Little did I know that as soon as I pass out, I would be pretty much doing Excel and PowerPoint all day!

So, anyway, I was in that room where our senior was explaining how he had used macros so that he could finish his summer internship work much faster and really impress the bosses and colleagues. I heard the preamble to the story where he told us about a scenario where he had a very difficult problem - lots of data and lots of reports to be generated - and how he used VBA to automate it. He went on to explain the techniques of automation. That's where I pretty much zoned out! Even though I wasn't sleeping, I was thinking that this did not apply to me. If I had to program, I could always do so, because I have been a programmer and it seemed unlikely that I would do any programming after passing out of B-school. I kind of zoned out and did not even listen to the VBA lecture.

After a couple of years, I started working as an Analyst and I was dealing with data most of the time. You could say that I was neck deep in data analysis! I would often find myself doing the same type of job every Monday, Tuesday or once every few weeks. I kept thinking that it would be better if I could automate this whole workflow. Instead of doing it again and again, if I did it once and recorded the steps in Excel, then the next time I had to do it, I would just need to press a button and Excel would do the work for me! This was the kind of thinking I had. Being a programmer, I knew that I was really looking for a way to program Excel so that it would do the work for me once I gave it the instructions of how to do it. I connected the dots and realised that I was really looking for a VBA solution. I went to the bookstore in the office campus and bought a VBA book. It was pretty big. I still have it right on my computer desk actually! The funny story is that I am using that book as a prop to adjust the height of my monitor. I have more books now and so I don't need the older version VBA books. Hence it became a prop! This VBA book helped me understand a lot of the fundamentals of the VBA ecosystem and eventually develop programs that I could automate my work with. To cut a long story short, VBA or a macro is really your way to automate a bunch of steps or program Excel so it can do something for you. Excel by itself is programmatic in nature. You're writing formulas so that you can get a result. Think of VBA as formulas++ (it's a very loose interpretation). That's a macro in my mind.

A macro or a VBA program is a set of steps that you tell Excel so that it can repeat them when you click a



button. I'll give you an example. Let's say that you're generating a particular type of report. The process of report generation involves collecting data from three different places (maybe 3 .csv files) and making four charts and saving the report as a PDF. These are the steps for the report generation. You do this every month. When you do it the first time, you're pretty excited. You're interested to know how the data works and how to connect it and make the charts. By the second month you still might be interested. Maybe your interest levels are at 60% of what they were earlier. But you still plough through the job. By the third or fourth month, you're least excited. Since you're not paying any attention to the work, there is a chance that you might even make a manual error. This is where VBA really shines. You do it the first time and record the steps. When you need to do it the second time, you press a button so that the VBA macro can do the steps for you. That's really what a macro does.

The terms macros and VBA programs are really interchangeable. People often use them in the same sense. When you hear somebody say VBA program or macro or automation, they all really mean in the same thing in the context of Excel. You're connecting a couple of steps into a computer program so that when you repeat the program the steps will be done for you. If you are thinking what the plausible reasons for using VBA are, the first one is to automate. Whenever you are doing the same set of steps every now and then, you could write an automation script or automation VBA macro so that those steps will be repeated for you when you click a button. A good example is the report generation example that we discussed earlier.

There are also a couple of other reasons for VBA. Another reason is to extend Excel's capacity. Excel has a lot of features but sometimes you might find that the feature you want is not available in Excel. In such cases, you could write your own program into Excel in order to extend Excel's capabilities. This is another classic and very much used way when it comes to VBA.

The other reason for using VBA is efficiency. Sometimes you might feel that using Excel's features - formulas, charts, conditional formatting - is not the most efficient way to do it. But the way that Excel has implemented them is the only way that you can use it. In such cases, you could write a VBA script or macro so that you can do the things in a more efficient fashion.

Another reason for using VBA is to create simple applications inside Excel. For example, Excel is pretty good for collecting data, analyzing and presenting results. But, at the same time, Excel is really poor at certain aspects like providing interactions with users or creating more of a program-like feel. VBA can help in such cases. For example, instead of asking users to enter data into cells, you could create a nice looking user form so that data can be entered there and results can be processed in the background. If you might remember, we were talking about this with Debra Dalgeish in a podcast some time ago. She talked about how user forms can be used with form controls to handle these kinds of things. These are the reasons why you can use Excel VBA.

If all of this sounds interesting then the next logical step would be - "How do I get started with VBA programming? How do I learn how to write macros?" Well, you don't have to learn much. Excel comes with an intuitive feature called macro recorder. The way this works is that Excel has an in-built recorder and anytime that you find yourself repeating a bunch of steps - for example download the .csv file data and format the first column as dates, the second column as currency, the third column as text and the fourth column as numbers etc. - i.e. some sort of repetitive steps that you are doing frequently, you could turn on this recorder, do all the steps and turn off the recorder. Excel has now recorded a short



tape repeating these actions for you. The next time that you want to do this on another set of data, you will play this tape and Excel will do the steps for you. That's what the Excel recorder feature is. It's called 'macro recorder' and you can use this from the Excel Developer ribbon. Once you use the recorder, Excel will record the macros for you and then you can play them again by using the 'view macros' button from either the Developer or the View ribbon. That's the easiest way to get started with macros.

In case you are curious to know how the macro is actually written in the background, you can also view the code. To do that, right click on any sheet, select 'view code' and it'll take you to the Visual Basic editor which is where you can explore the code. Using the Visual Basic editor, you can explore all the Visual Basic recordings that have been done. You can edit them, save them, modify them etc. In fact, we have been talking in plain English terms so far - recording, playing, tape etc. - but the real terms for this is 'SUB()'. When you record a small macro, it's called a 'SUB()'. It is the short form for sub-routine which is computer terminology and means that this is a routine or series of steps that we need to repeat. A sub-routine will be recorded every time that you press the recorder button and it will be available in what is called a module. In an Excel workbook we have worksheets and each sheet contains data or charts. In a sheet I can have five charts or two pivot tables or a bunch of formulas. A sheet can contain any number of things. Likewise, in the background, each module can contain any number of macros. To keep your modules nice and tidy, you can add additional modules so that one module can deal with automation aspects, another module can deal with chart generation aspects etc. By default when you hit the recorder button, Excel will insert Module 1 and it will record the first macro in it. These are sub-routines that will be grouped into a module. Again, from a beginner point of view, you don't need to worry about all these semantics. You need to focus on the steps you are repeating and how the macro is recording them. My suggestion to you is to open a blank workbook, paste some random data into it, press the recorder button and do some arbitrary steps like formatting data, filling in some colors, making a chart etc., stop the recorder button, go to 'view code' and see what code was recorded. Start with something really tiny like formatting a couple of columns. See the code and try to understand it.

Visual Basic is pretty much like plain English with some brackets and some special keywords. But if you read it, it looks almost like a plain English version of what the computer is doing. So try to read it and understand what's going on there. You will see a lot of similarities between what Excel would do and how VBA has recorded it. For example, if you are formatting by currency, the custom format code that we use in Excel will be same code that VBA will use. There is a lot of correlation between how you do things in Excel and how VBA records them. That said, there are also certain things that won't make immediate sense and that's where a little bit of investigation and playfulness will come in handy. The very first step is to start the recorder, do a couple of steps, stop the recorder and examine the code. After that, go ahead and record once again. Then maybe alter the steps or change the way you are doing things, stop the recorder and examine the code. This is the best way to get started and get your fingers dirty in VBA. This is the easiest and best way to learn VBA.

Now that you're feeling curious about VBA and you've started using it by playing with the recorder, if you're feeling curious and thinking how to really learn all this - a recorder is like a typist in the courtroom incessantly typing whatever conversation is taking place in court whether the judge says hmmm or the witness says thank you regardless of whether or not it's related to the case - VBA macros are also the same. If you are formatting a bunch of cells and you record a macro for it, but in between the recording you randomly click on a cell outside the range - we are moving the mouse and end up clicking on an un-related cell - even that step is going to be recorded. And, when you click back on the



original cell to format it again, that will be recorded as well. Everything you do, including dropping a pin, will be recorded! When you look at the recorded code you end up seeing a lot of garbage often. These include things that are not meant to be there but that you just did because you weren't thinking that somebody is watching you. You are just doing it as you do things every day.

When you are recording things, although it is a very good place to start, it is also a place where you have to remove a lot of garbage before you can understand it. In order to move beyond the recorder, I suggest these four resources:

1. **Get a book** - VBA is a programming language and there is a lot of terminology, concepts and idea that need to be learnt. My suggestion is to get an 'Excel Power Programming' book. This book is available on all major versions of Excel, i.e. there is Excel Power Programming for 2007, 2010 and 2013. I have the 2010 book. I don't plan to purchase the 2013 book because VBA is more or less similar between various versions of Excel. This book is written by John Walkenbach and it is a really good book to start learning VBA. That's my recommendation. You can buy this book or you can buy any other book that works for you. Learn with the help of a book. This is how I learnt programming way back in college. Being a technical person and a programmer at heart, I find VBA quite intuitive. But, that may not be the case for some of you. For example, if you are from a different background - I studied computer engineering and I was fascinated by computers since I was a child - but some of you may not have similar interests. You might be fascinated by some other things. In such cases, you might find programming a little counter intuitive. So, reading a book may not help you. If this is the case, then I suggest that you learn by playing.
2. The best way to learn, even better than through a book, is **by playing**. So, record or download some example macros or go to our website where we have tons of example workbooks on VBA. Download something, view the code, change things here and there and see how it works. Learn by playing. It's really learning through experimentation. This provides you with immediate feedback as to whether you are on the right track or not and what happens when a certain step changes. It's also incredibly fun. So, learn by playing.
3. The third recommendation that I have for you is to **learn by online reading**. We have a ton of very good articles on VBA on <http://chandoo.org> and there are lots of other websites where VBA is frequently discussed. You can go through these websites and learn. On the show notes page on <http://chandoo.org/session22/> I will provide some useful links for online learning of VBA.
4. The last recommendation that I have for you is to **join a class**. VBA is a very massive and in-depth technology. It is unlikely that you will understand what it is and how to apply it to your work in a 10-minute podcast, an hour long reading of a book or even in a day spent reading online. It is better to learn it in a structured way. You could go for a class. If there is any university near you where they are offering a VBA course in the evening, go for it. If you find that you don't have the time to dedicate for something like that you could also enrol in an online training program. At <http://chandoo.org> I run a course called 'VBA class'. It's a completely online class where we teach everything right from the basics of VBA to advanced VBA, where you can build user forms and simple applications using VBA. It is a very comprehensive course with 24 hours of video material. And, it's all on-demand. Anytime that you want to watch a video you can log in to the course website and watch it. There is a detailed lesson plan to help you through the basics all the way along to advanced VBA. Each lesson comes with an example workbook that you can download and play. There are also some homework and class projects to



check and monitor your progress. And, at any point if you have doubts, you can take our staff's help.

These are some recommendations. Again, take a pick on whatever works for you and go with it. Even if you choose an online class, it's not terribly expensive. It usually starts at about \$97 and there are some costlier options as well. Go with something that works best for you and learn VBA.

From my personal experience, even though I was really bored and zoned out at that afternoon guest lecture in 2005, once I started appreciating what VBA could do for me, I have seen tremendous productivity improvements in my own work. Even just yesterday I was writing a small VBA script to improve my work. So, VBA is something that can help you no matter what kind of work you are doing or how you use Excel unless you are doing something really rudimentary with Excel. If you are spending more than a couple of hours a week using Excel, I encourage you to learn VBA.

There you go. I hope this podcast has enlightened you on what a macro is and how to get into macros with some introduction. Please visit <http://chandoo.org/session22/> where I will provide all the show notes and resources for this podcast. I will mention the book recommendations, example workbooks as well as some resources for online reading along with a link to join VBA classes from chandoo.org.

Thank you so much. Stay awesome.