Startup investment in the Nordic countries

by Arda Acikalin, Yemeskabeba Gessesse, Omar Ismail Mfarij and Dmitry Tolonen

Based on the **StartUp Investments (Crunchbase) dataset** - Information about startup companies and investment via Crunchbase https://www.kaggle.com/arindam235/startup-investments-crunchbase (https://www.kaggle.com/arindam235/startup-investments-crunchbase)

Table of contents:

Interesting general themes and the group's methodology

- Our group discussed various mutually interesting subject areas for the final project such as startups and innovation, the sustainability of 'crunch culture' and the mental/health situation within the tech industry, the biotech/health sector and the entertainment, news and media sectors as well as distribution (Netflix and Spotify type operations).
- In trying to find a unique combination of these subjects, we found many interesting datasets such as the OSMI mental health in tech survey data. Ultimately, however, we were drawn to a Crunchbase dataset for startup investment, which could offer some other indicators of the tech industry's health through data relating to the type, duration and sources of investment and transfer ownership of new startups, particularly in the context of IP remaining within the originating countries. These were lofty goals, especially as the data relating to the entities acquiring startups is concerned, but we hoped that we could get at least an initial picture to help us further.
- We would therefore have to limit our scope for the dataset at some point in order to gain a core understanding of our subject and for this we decided to focus mostly on the type of financing among Nordic countries.

Introduction to our dataset

StartUp Investments (Crunchbase): https://www.kaggle.com/arindam235/startup-investments-crunchbase (https://www.kaggle.com/arindam235/startup-investments-crunchbase)

- The dataset we used was downloaded from Kaggle but the source of the data is from Crunchbase, which is described as "a platform for finding business information about private and public companies."

The dataset consists mostly of financial data relating to startup investment, such as:

- startup name
- url
- market segment/category (e.g. software, biotech, health and fitness, real estate, search, mobile, education, transportation, finance etc.)
- status (operating, acquired, closed)
- location: country, city, region
- funding: total funding in usd, type of funding (public grant, seed, angel, debt taken, venture capital & so-called funding rounds)

- The Kaggle page description indicates an interest in seeing whether subsequent rounds of investment help a company to move to a company status of 'operating/closed/aquired', presumably from a status of starting up or seed/grant/angel funding. At face level, this sounds like a obvious description, but we were interested in digging deeper into the data relating to the industries that could possibly employ us and funders that could possibly finance any startup ideas of ours.

In [2]:

```python
# Importing necessary Python packages
import numpy as np
import pandas as pd
import re # regex module - did we need to use this?
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

Challenges of the dataset and Jupyter collaboration

- The dataset is fairly large at 39 columns and 50k rows, so we knew a lot of the data would not readily serve our purposes. Also there was some data that would be redundant for us, duplicating date data etc. This we would tackle in our choices tof focus on a subset of countries and columns as well as in the clean up stage.
- In researching collaboration methods for our group, we landed on using Google's Colab tools, which is an online Jupyter environment linked to your Google account. We used this, Hangouts and Whatsapp for communication and had set up a Google Slides doc in case we had to present in Powerpoint/Slides format.
- Although Colab has proven to be reliable, there have been some hiccups and therefore, we mainly used our Colab Jupyter document as a central repository of our current/overnight work as well as a sketchbook while meeting in Google Hangouts. We did our main data analysis on our own Jupyter environments on our computers.
- Another thing to note about Colab: we stored our Colab work in Dmitry's Google Drive, which required frequent authentication for every user, so to solve this issue, the CSV file was moved to a website (dmitrytolonen.com in the read_csv cell), as this avoided the authentication issue and allowed everyone better access to the work file.
- Finally, the dataset lacked some data to which we would have liked to have access, such as more data about the entities making acquisitions, whether they were large, rich countries such as the US, China etc. or whether the Nordic IP remained in the region (with its potential employment and taxation ramifications).

In [3]:

```
# Trying to read CSV file from Google Drive into Google Colab - constant access/authorisation problems

#from google.colab import drive
#drive.mount('/content/drive')

# trick to force remount Google Drive, try: drive.mount("/content/drive", force_remount=True)
```

In [4]:

```
# Reading the csv files from personal website (dmitrytolonen.com) instead of Google Drive/Colab due to access/auth problems
investments = pd.read_csv('http://www.dmitrytolonen.com/assets/colab/investments_VC.csv', encoding = "ISO-8859-1")

# CSV file had an encoding read error, so we added an encoding option from Stack Overflow:
# https://stackoverflow.com/questions/18171739/unicodedecodeerror-when-reading-csv-file-in-pandas-with-python
```

Focusing our study: Startup investment in the Nordic countries

Ok, let's get to work! Well, we have too much data at our hands, so first we had a look at the structure and brief contents of the data, such as the columns. We soon chose to select a subset of columns along our focus on the Nordics, avoiding some overlap especially in the date-related columns.

In [5]:

```
investments.head()

# for a quick glance at the material's first five rows.
```

Out[5]:

	permalink	name	homepage_url	
0	/organization/waywire	#waywire	http://www.waywire.com	|Entertainment|P
1	/organization/tv-communications	&TV Communications	http://enjoyandtv.com	
2	/organization/rock-your-paper	'Rock' Your Paper	http://www.rockyourpaper.org	
3	/organization/in-touch-network	(In)Touch Network	http://www.InTouchNetwork.com	|Electronics|Guides|Cc
4	/organization/r-ranch-and-mine	-R- Ranch and Mine	NaN	|Tour

5 rows × 39 columns

In [6]:

```
investments.info()

# using the Pandas info function, we get a summary of the dataframe's contents
,
# eg. dtypes, columns and their names, entries and non-null values.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54294 entries, 0 to 54293
Data columns (total 39 columns):
permalink               49438 non-null object
name                    49437 non-null object
homepage_url            45989 non-null object
category_list           45477 non-null object
 market                 45470 non-null object
 funding_total_usd      49438 non-null object
status                  48124 non-null object
country_code            44165 non-null object
state_code              30161 non-null object
region                  44165 non-null object
city                    43322 non-null object
funding_rounds          49438 non-null float64
founded_at              38554 non-null object
founded_month           38482 non-null object
founded_quarter         38482 non-null object
founded_year            38482 non-null float64
first_funding_at        49438 non-null object
last_funding_at         49438 non-null object
seed                    49438 non-null float64
venture                 49438 non-null float64
equity_crowdfunding     49438 non-null float64
undisclosed             49438 non-null float64
convertible_note        49438 non-null float64
debt_financing          49438 non-null float64
angel                   49438 non-null float64
grant                   49438 non-null float64
private_equity          49438 non-null float64
post_ipo_equity         49438 non-null float64
post_ipo_debt           49438 non-null float64
secondary_market        49438 non-null float64
product_crowdfunding    49438 non-null float64
round_A                 49438 non-null float64
round_B                 49438 non-null float64
round_C                 49438 non-null float64
round_D                 49438 non-null float64
round_E                 49438 non-null float64
round_F                 49438 non-null float64
round_G                 49438 non-null float64
round_H                 49438 non-null float64
dtypes: float64(23), object(16)
memory usage: 16.2+ MB
```

In [7]:

```
investments.describe()
# for some statistics on the dataframe's non-null entries (49k+) as well as it
s the mean, min, max values etc. - how the data is distributed.
```

Out[7]:

	funding_rounds	founded_year	seed	venture	equity_crowdfunding	u
count	49438.000000	38482.000000	4.943800e+04	4.943800e+04	4.943800e+04	4.9
mean	1.696205	2007.359129	2.173215e+05	7.501051e+06	6.163322e+03	1.3
std	1.294213	7.579203	1.056985e+06	2.847112e+07	1.999048e+05	2.9
min	1.000000	1902.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.0
25%	1.000000	2006.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.0
50%	1.000000	2010.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.0
75%	2.000000	2012.000000	2.500000e+04	5.000000e+06	0.000000e+00	0.0
max	18.000000	2014.000000	1.300000e+08	2.351000e+09	2.500000e+07	2.9

8 rows × 23 columns

In [8]:

```
# Choosing the most suitable columns for our study
investments = investments[['name', ' market ',
       ' funding_total_usd ', 'status', 'country_code', 'region',
       'city', 'funding_rounds', 'founded_at', 'founded_year','first_funding_a
t',
       'last_funding_at', 'seed', 'venture', 'equity_crowdfunding',
       'undisclosed', 'convertible_note', 'debt_financing', 'angel', 'grant',
       'private_equity', 'post_ipo_equity', 'post_ipo_debt',
       'secondary_market', 'product_crowdfunding', 'round_A', 'round_B',
       'round_C', 'round_D', 'round_E', 'round_F', 'round_G', 'round_H']]
```

In [9]:

```
# Further limiting our dataset to only nordic countries and setting the index
from the default to 'country_code'
investments = investments.set_index('country_code')
```

In [10]:

```
# Using the Pandas loc function to access specific labelled rows and columns
investments = investments.loc[['FIN', 'SWE', 'NOR', 'DNK', 'ISL'], :]
```

In [11]:

```
# For a random sample from the selected subset

investments.sample(10)
```

Out[11]:

	name	market	funding_total_usd	status	region	
country_code						
SWE	Smart Eye	Hardware + Software	38,95,881	operating	Gothenburg	G
SWE	Imsys	Hardware + Software	17,94,902	operating	SWE - Other	Up
DNK	Conferize	Curated Web	18,38,000	operating	Copenhagen	Cope
DNK	HeatGear	NaN	5,70,000	operating	DNK - Other	Skan
SWE	Ascade	Software	25,90,000	acquired	Stockholm	Stc
SWE	PocketMobile	Mobile	63,50,000	operating	Stockholm	Stc
SWE	Incentive	Enterprise Software	22,55,925	operating	Malmo	
FIN	Sensinode	Software	13,68,100	acquired	Oulu	
SWE	The African Management Initiative (AMI)	Education	7,50,000	operating	SWE - Other	Johan
SWE	Denator	Biotechnology	83,24,204	operating	Gothenburg	Got

10 rows × 32 columns

In [12]:

```
investments['round_F'].unique()
# We were interested in seeing whether there is much - or any - useful data in
round_F - or, actually, after round_C?
```

Out[12]:

```
array([0.000000e+00, 3.800204e+06, 2.500000e+08, 1.354000e+08])
```

In [13]:

```
investments.head()
```

Out[13]:

	name	market	funding_total_usd	status	region	city	fundi
country_code							
FIN	720°	Predictive Analytics	1,90,035	operating	Helsinki	Espoo	
FIN	Aava Mobile	Mobile	42,60,000	operating	Helsinki	Helsinki	
FIN	Academica	Consulting	-	operating	Helsinki	Helsinki	
FIN	AddSearch	Enterprise Search	6,50,000	operating	Helsinki	Helsinki	
FIN	Aito Technologies	Software	34,03,750	operating	Helsinki	Espoo	

5 rows × 32 columns

In [14]:

```
investments.reset_index()
```

Out[14]:

	country_code	name	market	funding_total_usd	status	region	
0	FIN	720°	Predictive Analytics	1,90,035	operating	Helsinki	
1	FIN	Aava Mobile	Mobile	42,60,000	operating	Helsinki	
2	FIN	Academica	Consulting	-	operating	Helsinki	
3	FIN	AddSearch	Enterprise Search	6,50,000	operating	Helsinki	
4	FIN	Aito Technologies	Software	34,03,750	operating	Helsinki	
...	...	...	...	...	...	...	
828	ISL	Snjohus Software	Apps	16,000	operating	Reyjavik	R
829	ISL	SuitMe	Software	16,000	operating	Reyjavik	R
830	ISL	Tour Desk	Tourism	4,000	operating	Reyjavik	R
831	ISL	Transmit	Sales and Marketing	4,50,000	operating	Reyjavik	R
832	ISL	vivio	Reviews and Recommendations	2,50,000	operating	Reyjavik	R

833 rows × 33 columns

Data clean up stage

- At this stage, we wanted to make sure some of the spaces, commas, dtypes, date types and decimal places were correct and usable for our purposes. We also dropped columns with zero values (in particular, funding rounds) and grouped countries by country_code.
- We used the replace function for commas and spaces, pd.to.datetime function for date changes and groupby for grouping by country_code label and, finally, the drop function for dropping columns with zero values.

In [15]:

```
#cleaning column names and reformatting the data
#Column names had some empty spaces
investments.columns = investments.columns.str.replace(' ','')
```

In [16]:

```
#removed the commas from the total funding
investments.funding_total_usd = investments.funding_total_usd.str.replace(',','')
```

In [17]:

```
#it was string, now it is a float.
investments.funding_total_usd = pd.to_numeric(investments.funding_total_usd, errors='coerce')
```

In [18]:

```
# changing the data type for column founded_at to datetime type
investments['founded_at'] = pd.to_datetime(investments['founded_at'], errors = 'coerce' )
```

In [19]:

```
# changing the data type for column first_funding_at to datetime type
investments.first_funding_at = pd.to_datetime(investments.first_funding_at, format='%Y/%m/%d', errors='coerce')
```

In [20]:

```
# changing the data type for column first_funding_at to datetime type
investments.first_funding_at = pd.to_datetime(investments.first_funding_at, format='%Y/%m/%d', errors='coerce')
```

In [21]:

```
# Now we can groupby country code.
investments.groupby('country_code').mean()
```

Out[21]:

	funding_total_usd	funding_rounds	founded_year	seed	ventu
country_code					
DNK	8.165931e+06	1.366667	2008.070588	258798.557143	3.985626e+0
FIN	6.251534e+06	1.422680	2008.166667	380874.051546	4.315700e+0
ISL	3.281002e+06	1.312500	2009.812500	65474.000000	2.953125e+0
NOR	9.156303e+06	1.479592	2006.779412	149348.979592	4.560979e+0
SWE	1.072787e+07	1.460317	2006.884120	190803.193651	7.015214e+0

5 rows × 24 columns

In [22]:

```
# deleting the columns round_G and round_H because all the countries have zero
values
investments = investments.drop(["round_G", "round_H"], axis=1)
```

In [23]:

```
# rounding the mean values to 4 decimal places
investments.groupby('country_code').mean().round(4)
```

Out[23]:

	funding_total_usd	funding_rounds	founded_year	seed	venture
country_code					
DNK	8.165931e+06	1.3667	2008.0706	258798.5571	3.985626e+06
FIN	6.251534e+06	1.4227	2008.1667	380874.0515	4.315700e+06
ISL	3.281002e+06	1.3125	2009.8125	65474.0000	2.953125e+06
NOR	9.156303e+06	1.4796	2006.7794	149348.9796	4.560979e+06
SWE	1.072787e+07	1.4603	2006.8841	190803.1937	7.015214e+06

5 rows × 22 columns

In [24]:

```
# rounding the values of the data frame to 4 decimal places
investments= investments.round(4)
investments
```

Out[24]:

	name	market	funding_total_usd	status	region	ci
country_code						
FIN	720°	Predictive Analytics	190035.0	operating	Helsinki	Esp
FIN	Aava Mobile	Mobile	4260000.0	operating	Helsinki	Helsin
FIN	Academica	Consulting	NaN	operating	Helsinki	Helsin
FIN	AddSearch	Enterprise Search	650000.0	operating	Helsinki	Helsin
FIN	Aito Technologies	Software	3403750.0	operating	Helsinki	Esp
...	...	...	...	...	...	
ISL	Snjohus Software	Apps	16000.0	operating	Reyjavik	Reykjav
ISL	SuitMe	Software	16000.0	operating	Reyjavik	Reykjav
ISL	Tour Desk	Tourism	4000.0	operating	Reyjavik	Reykjav
ISL	Transmit	Sales and Marketing	450000.0	operating	Reyjavik	Reykjav
ISL	vivio	Reviews and Recommendations	250000.0	operating	Reyjavik	Reykjav

833 rows × 30 columns

In [25]:

```
investments.columns
```

Out[25]:

```
Index(['name', 'market', 'funding_total_usd', 'status', 'region',
'city',
       'funding_rounds', 'founded_at', 'founded_year', 'first_fund
ing_at',
       'last_funding_at', 'seed', 'venture', 'equity_crowdfunding'
,
       'undisclosed', 'convertible_note', 'debt_financing', 'angel
', 'grant',
       'private_equity', 'post_ipo_equity', 'post_ipo_debt',
       'secondary_market', 'product_crowdfunding', 'round_A', 'rou
nd_B',
       'round_C', 'round_D', 'round_E', 'round_F'],
      dtype='object')
```

In [26]:

```
# We discuss whether to keep the region column instead of the city column,
# as 'region' is clearer and less complex when we run data analysis

investments['region'].unique()
```

Out[26]:

```
array(['Helsinki', 'Turku', 'Oulu', 'Vantaa', 'Tampere', 'FIN - Ot
her',
       'Kuopio', 'Lappeenranta', 'Jyväskylä', 'Stockholm', 'SWE -
Other',
       'Linkoping', 'Malmo', 'Gothenburg', 'Jonkoping', 'Lulea',
       'Sundsvall', 'Jamtlands Lan', 'Umea', 'Norrkoping', 'Karlsk
rona',
       'Halmstad', 'Danderyd', 'Aust-Agder', 'Trondheim', 'NOR - O
ther',
       'Oslo', 'Sandnes', 'Lysaker', 'Tromso', 'Fornebu', 'Skien',
       'Kristiansand', 'Stavanger', 'Copenhagen', 'DNK - Other',
       'Ballerup', 'Lyngby', 'Aarhus', 'Frederiksberg', 'Valby', '
Vejle',
       'Aalborg', 'Odense', 'Hørsholm', 'Allerød', 'Herlev', 'Kold
ing',
       'Hellerup', 'Taastrup', 'Farum', 'Reyjavik', 'ISL - Other']
,
      dtype=object)
```

In [27]:

```
# So, our option would be to drop the city column as it DOES ADD lots of compl
ex data when we run data analysis

investments['city'].unique()
```

Out[27]:

```
array(['Espoo', 'Helsinki', 'Turku', 'Oulu', 'Vantaa', 'Tampere',
'Sipoo',
       'Kuopio', 'Nokia', 'Kaarina', 'Lappeenranta', 'Jyväskylä',
       'Kajaani', 'Itäharju', 'Mikkeli', 'Kempele', 'Uusikaupunki'
,
       'Stockholm', 'Kista', 'Härnösand', 'Linköping', 'Helsingbor
g',
       'Göteborg', 'Malmö', 'Gothenburg', 'Jönköping', nan, 'Skarp
näck',
       'Limhamn', 'Lund', 'Västra Frölunda', 'Luleå', 'Malma',
       'Örnsköldsvik', 'Sundsvall', 'Falkenberg', 'Partille', 'Upp
sala',
       'Umeå', 'Bodafors', 'Solna', 'Mölndal', 'Upplands-väsby', '
Gävle',
       'Skövde', 'Norrköping', 'Åkarp', 'Karlskrona', 'Jämtland',
       'Hyssna', 'Täby', 'Halmstad', 'Karlstad', 'Huddinge', 'Falu
n',
       'Västerås', 'Haninge', 'Sundbyberg', 'Hammenhög', 'Norrtälj
e',
       'Johannesberg', 'Sölvesborg', 'Danderyd', 'Värmdö', 'Trondh
eim',
       'Fredrikstad', 'Sarpsborg', 'Tiller', 'Mjømna', 'Porsgrunn'
,
       'Oslo', 'Asker', 'Sandnes', 'Hvalstad', 'Lysaker', 'Ballsta
d',
       'Tromsø', 'Fornebu', 'Skien', 'Måløy', 'Kristiansand', 'Jes
sheim',
       'Mysen', 'Lillestrøm', 'Horten', 'Stavanger', 'Tofte', 'Hal
den',
       'Sogndal', 'Copenhagen', 'Kongens Lyngby', 'Sønderborg',
       'Østerby Havn', 'Ballerup', 'Helsingør', 'Hvidovre', 'Nordb
org',
       'Lyngby', 'Fredensborg', 'Aarhus', 'Løgstør', 'Frederiksber
g',
       'Gentofte', 'Roskilde', 'Valby', 'Jyderup', 'Vejle', 'Lejre
',
       'Holstebro', 'Aalborg', 'Hammel', 'Gerning', 'Odense', 'Vær
løse',
       'Fredericia', 'Haderslev', 'Risskov', 'Stenløse', 'Hjalleru
p',
       'Hørsholm', 'Skanderborg', 'Korsør', 'Allerød', 'Københoved
',
       'Herlev', 'Nærum', 'Kolding', 'Glostrup', 'Hellerup', 'Hill
erød',
       'Taastrup', 'Esbjerg', 'Hedehusene', 'Birkerød', 'Vipperød'
,
       'Farum', 'Skovlunde', 'Hornsyld', 'Kjoge', 'Smørumnedre',
       'Reykjavík', 'Ísafjörður', 'Hafnarfjörður'], dtype=object)
```

In [28]:

```
# We decide to go with dropping the city column for clarity's sake.
investments = investments.drop(["city"], axis=1)
```

Visualisation: Status, scale and popular market segments for investors

- Here's where we get into the fun part. In both tabular and visual form, we can inspect the data we have and make some observations and connections from our selected statistics.
- We move closer into our central focus and compare the Nordic countries in a number of areas.

In [29]:

```
investments.head()
```

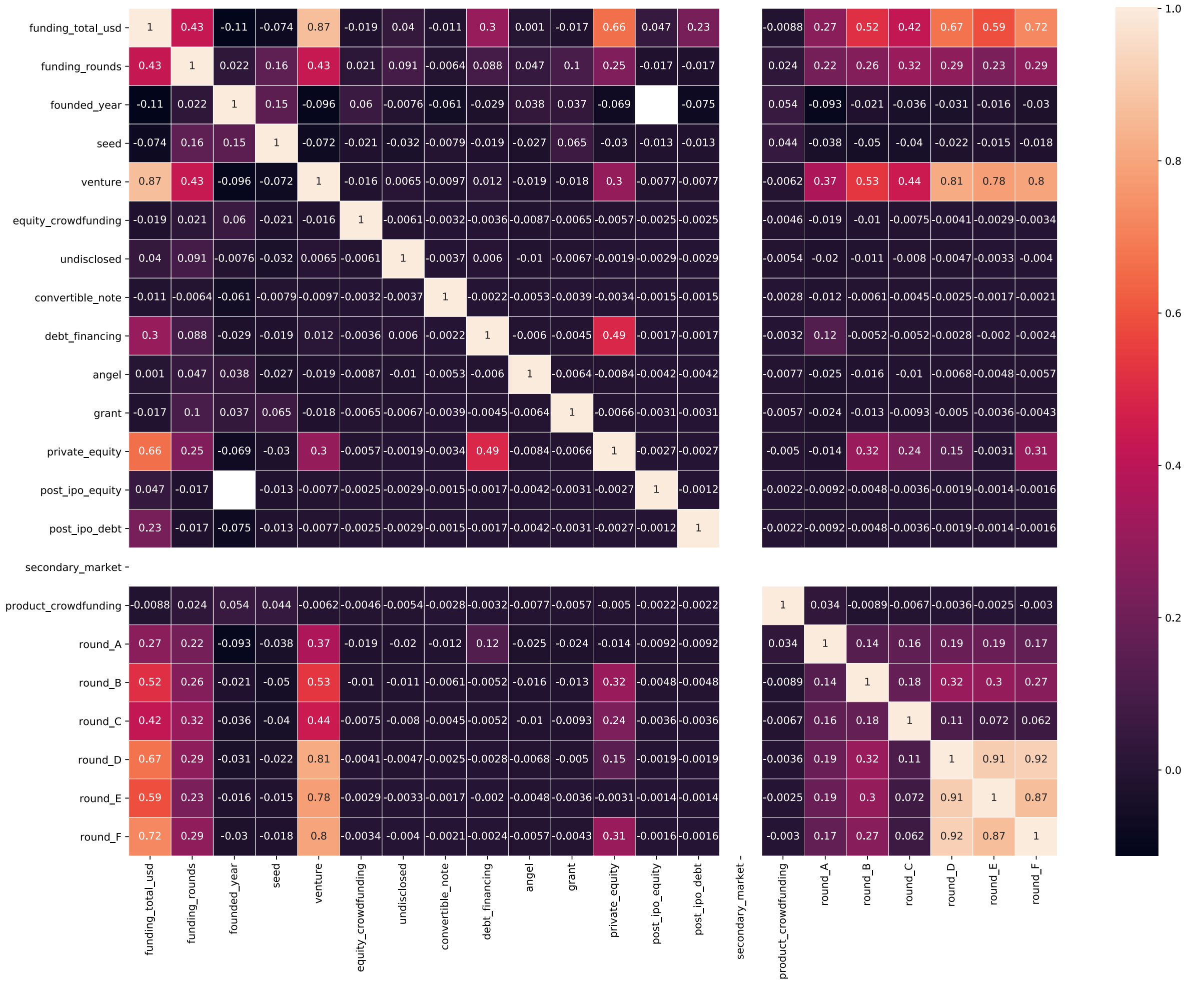
Out[29]:

	name	market	funding_total_usd	status	region	funding_round
country_code						
FIN	720°	Predictive Analytics	190035.0	operating	Helsinki	1.
FIN	Aava Mobile	Mobile	4260000.0	operating	Helsinki	1.
FIN	Academica	Consulting	NaN	operating	Helsinki	1.
FIN	AddSearch	Enterprise Search	650000.0	operating	Helsinki	1.
FIN	Aito Technologies	Software	3403750.0	operating	Helsinki	1.

5 rows × 29 columns

In [30]:

```
plt.subplots(figsize=(20,15))

sns.heatmap(investments.corr(), annot=True, linewidth=0.5);

# This heat map serves to give us a bigger picture of our investments data.
```



In [31]:

```
investments['founded_year'].unique()
```

Out[31]:

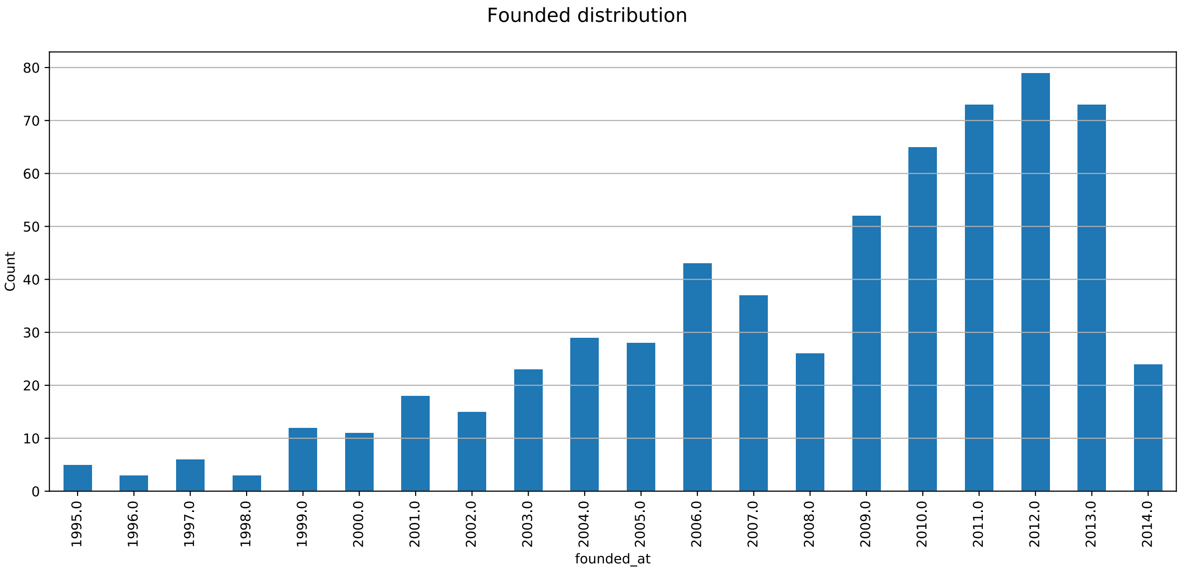
```
array([2012., 2009.,   nan, 2013., 2006., 2001., 2008., 2002., 200
3.,
       2011., 1995., 1991., 2010., 2004., 2014., 2005., 2000., 199
9.,
       2007., 1992., 1968., 1972., 1990., 1997., 1998., 1989., 199
6.,
       1994., 1961., 1983., 1926.])
```

In [32]:

```
# Ungroup dataset

investments = investments.reset_index(level='country_code')
```

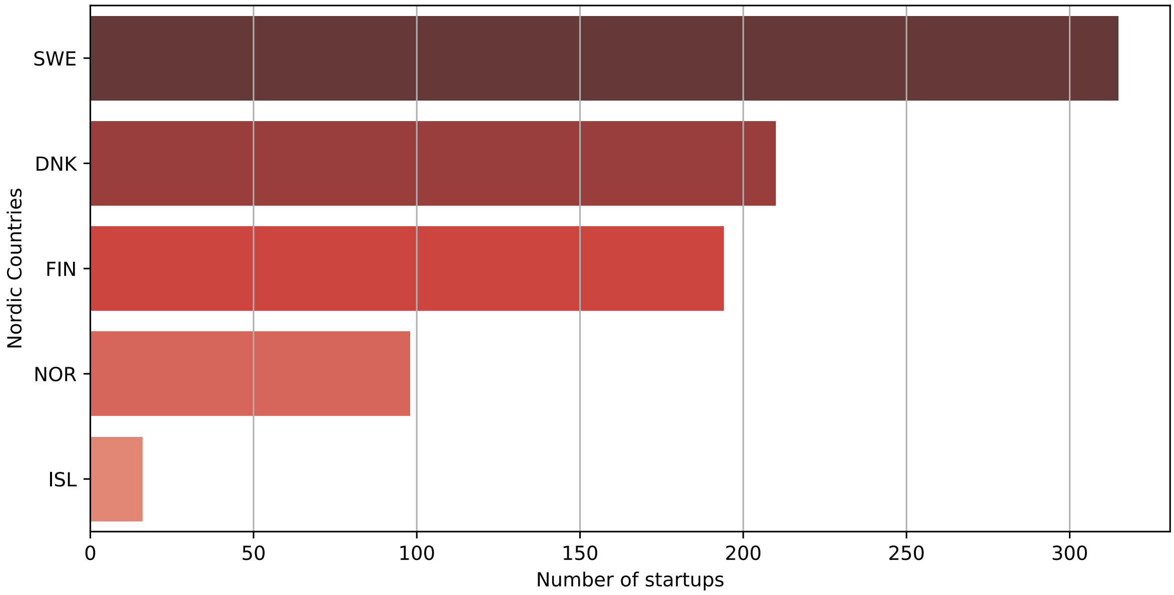
In [33]:

```
# Startup etablishment for the past 20 years in Nordic countries (NC)

plt.rcParams['figure.figsize'] = 15,6
investments['name'].groupby(investments["founded_at"].dt.year).count().tail(20
).plot(kind="bar")

ax = plt.axes()
ax.yaxis.grid()
plt.ylabel('Count')
plt.title("Founded distribution ", fontdict=None, position= [0.48,1.05], size
= 'x-large')
plt.show()
```



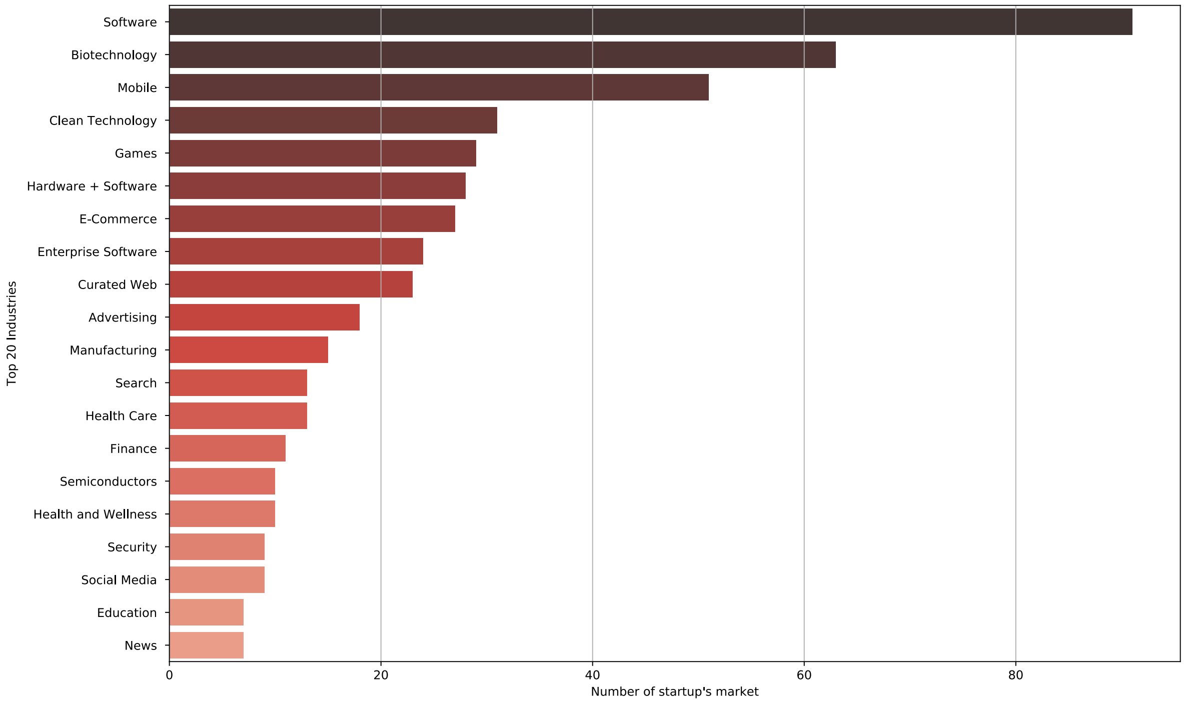
In [34]:

```
# The total number of startups in each Nordic country

plt.figure(figsize=(10,5))

sns.barplot(x=investments['country_code'].value_counts(), y=investments['country_code'].value_counts().index, palette='Reds_d')

ax = plt.axes()
ax.xaxis.grid()
plt.xlabel('Number of startups')
plt.ylabel('Nordic Countries')
plt.show()
```



In [35]:

```
# Top 20 popular startup markets in Nordic countries.

plt.figure(figsize=(15,10))

sns.barplot(x=investments['market'].value_counts().head(20), y=investments['market'].value_counts().head(20).index, palette='Reds_d')

ax = plt.axes()
ax.xaxis.grid()
plt.xlabel("Number of startup's market")
plt.ylabel('Top 20 Industries')
plt.show()
```



In [36]:

```
# Top 10 startup status based on market sector

operating = investments[investments.status == 'operating']
acquired = investments[investments.status == 'acquired']
closed = investments[investments.status == 'closed']

operating_count  = operating['market'].value_counts()
operating_count = operating_count[:10,]

acquired_count  = acquired['market'].value_counts()
acquired_count = acquired_count[:10,]

closed_count  = closed['market'].value_counts()
closed_count = closed_count[:10,]

print('Operating')
print(operating_count)
print('')
print('-------------------------------------------')
print('Acquired')
print(acquired_count)
print('')
print('-------------------------------------------')
print('Closed')
print(closed_count)
```

```
Operating
 Software                71
 Biotechnology           55
 Mobile                  47
 Clean Technology        29
 Games                   26
 E-Commerce              25
 Hardware + Software     25
 Enterprise Software     22
 Curated Web             20
 Manufacturing           15
Name: market, dtype: int64

-------------------------------------------
Acquired
 Software                 10
 Mobile                    3
 Biotechnology             3
 Semiconductors            2
 Games                     2
 Hardware + Software       2
 Web Hosting               1
 Meeting Software          1
 Social Network Media      1
 Social Media              1
Name: market, dtype: int64

-------------------------------------------
Closed
 Software             8
 Health Care          3
 Biotechnology        3
 Curated Web          2
 Clean Technology     2
 Advertising          2
 E-Commerce           2
 SaaS                 1
 Mobile Games         1
 Entertainment        1
Name: market, dtype: int64
```

In [37]:

```
# Startup status for each Nordic country

startup = investments.groupby('country_code').status.value_counts()

startup
```

Out[37]:

```
country_code  status   
DNK           operating    186
              acquired      13
              closed        10
FIN           operating    171
              acquired      11
              closed         8
ISL           operating     14
              closed         2
NOR           operating     85
              acquired       6
              closed         6
SWE           operating    277
              closed        18
              acquired      12
Name: status, dtype: int64
```

In [38]:

```
# Startup status for each Nordic country for bar plot
startup1 = investments.groupby('country_code').status.value_counts().reset_ind
ex(name='counts')

startup1
```

Out[38]:

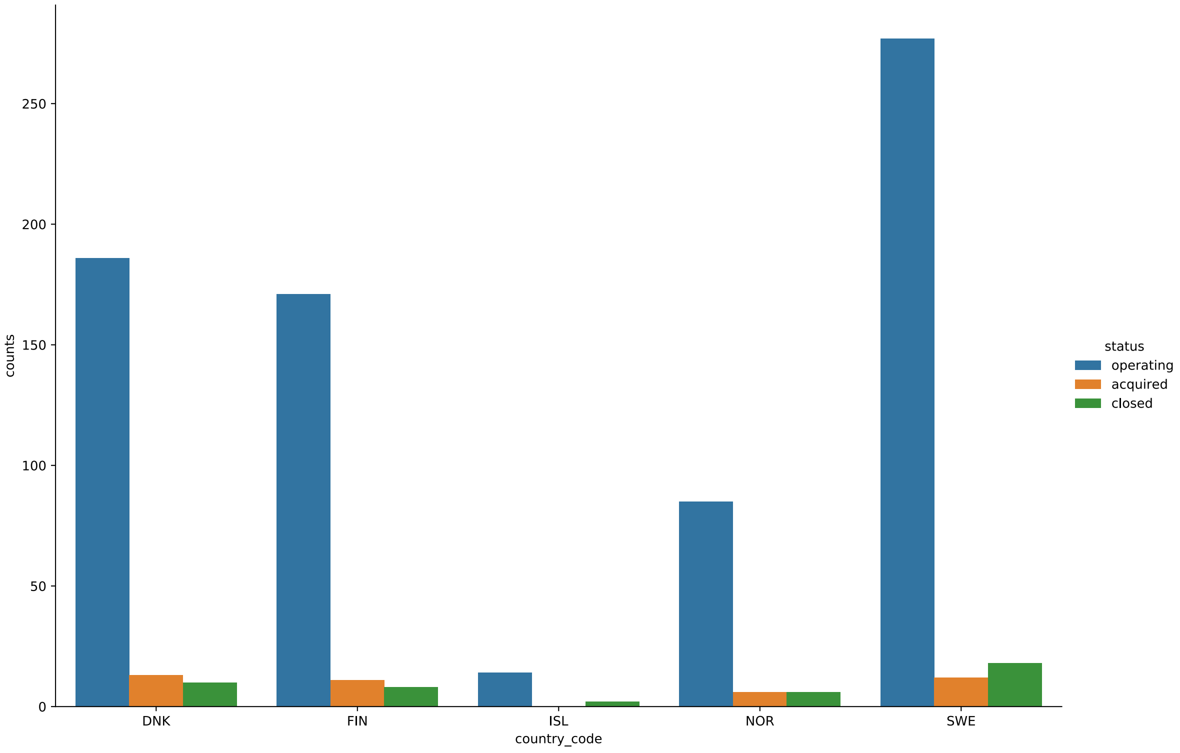
	country_code	status	counts
0	DNK	operating	186
1	DNK	acquired	13
2	DNK	closed	10
3	FIN	operating	171
4	FIN	acquired	11
5	FIN	closed	8
6	ISL	operating	14
7	ISL	closed	2
8	NOR	operating	85
9	NOR	acquired	6
10	NOR	closed	6
11	SWE	operating	277
12	SWE	closed	18
13	SWE	acquired	12

In [39]:

```
sns.catplot(x="country_code", y="counts", hue="status", kind="bar", data=startup1, height=8.27, aspect=11.7/8.27)
```

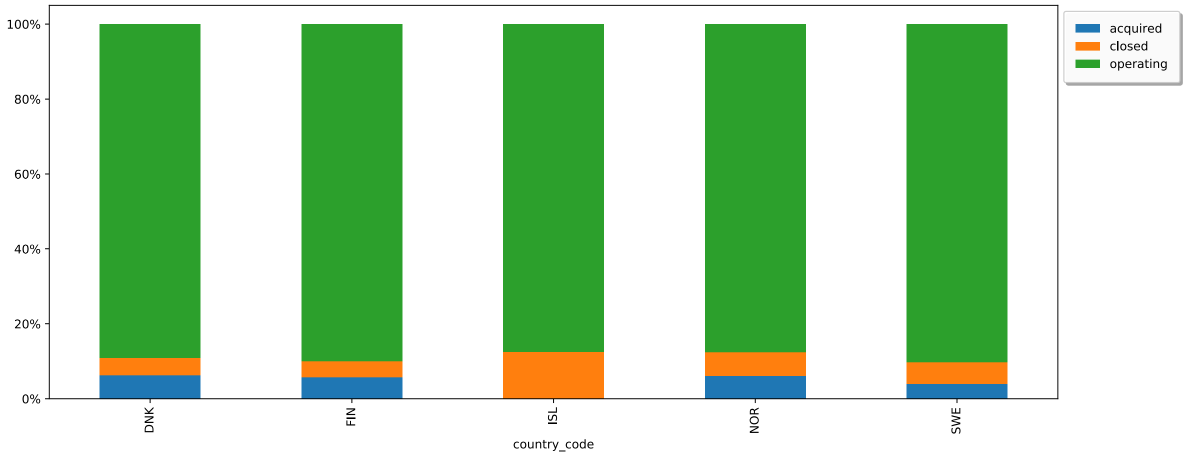
Out[39]:

```
<seaborn.axisgrid.FacetGrid at 0x122f75e80>
```



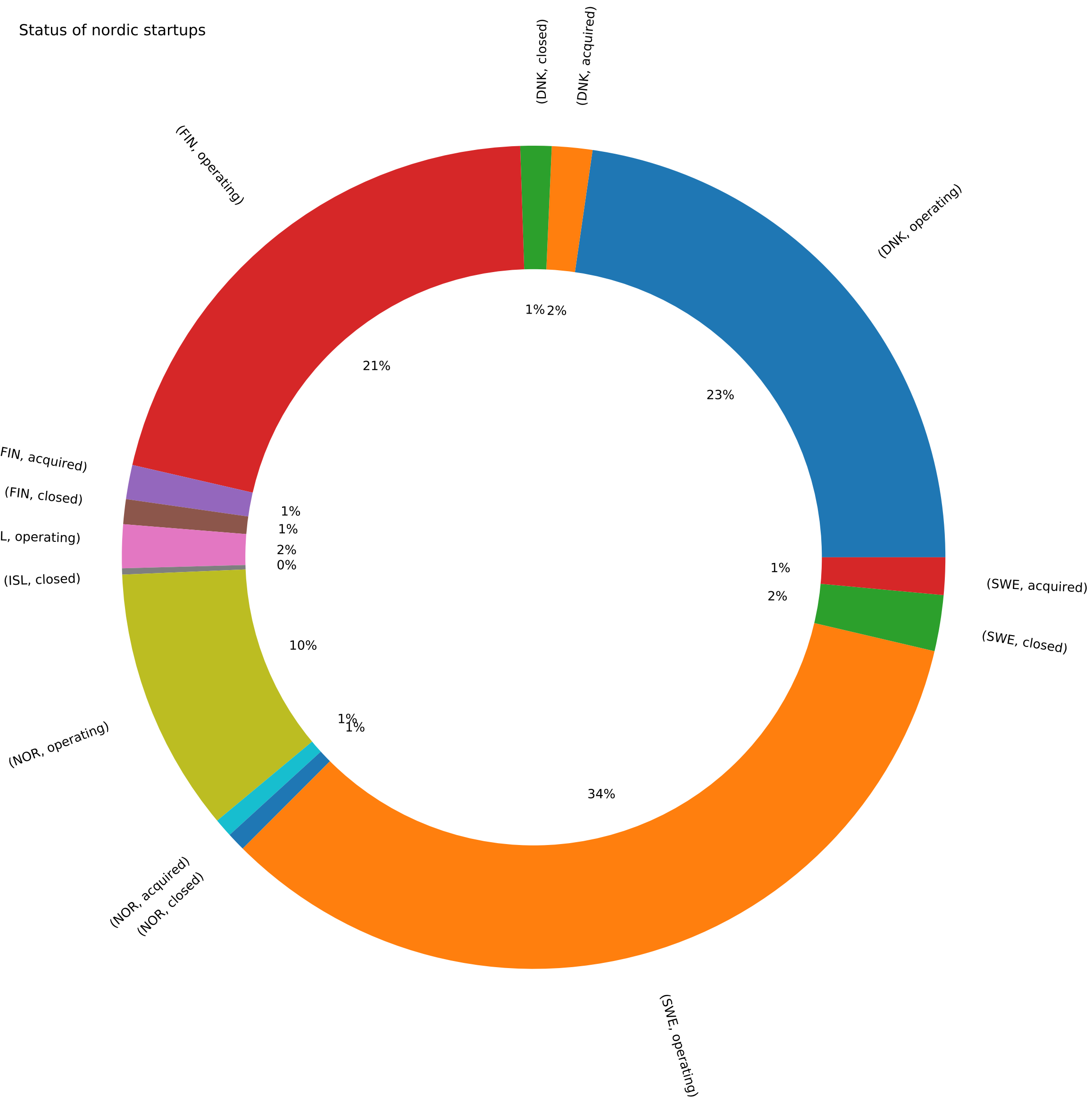
In [40]:

```
# Relationship between operating, acquired and closed startups in each Nordic
country - normalised to a 100 percent bar chart

import matplotlib.ticker as mtick

investments.groupby(['country_code','status']).size().groupby(level=0).apply(
    lambda x: 100 * x / x.sum()
).unstack().plot(kind='bar',stacked=True)

plt.gca().yaxis.set_major_formatter(mtick.PercentFormatter())
plt.legend(bbox_to_anchor=(1,1), frameon = True, fancybox = True, framealpha =
0.95, shadow = True,
           borderpad = 1)
plt.show()
```



In [41]:

```
# Status of Nordic startups in donut form

chart = startup.plot(kind="pie", figsize=(20,15), autopct="%1.0f%%", rotatelabels=True )
cen_cir = plt.Circle((0,0),0.70, fc='w')
plt.gcf().gca().add_artist(cen_cir)

chart.set_ylabel('')
plt.title("Status of nordic startups", loc='left')
plt.show()
```



In [42]:

```
# Most funded startups/companies in Nordic countries

most_funded = investments.nlargest(20, ['funding_total_usd'])
most_funded
```

Out[42]:

	country_code	name	market	funding_total_usd	status	regi
447	SWE	Spotify	Entertainment	537779080.0	operating	Stockhol
779	DNK	Symphogen	Biotechnology	387185400.0	operating	Lyngl
334	SWE	Klarna	Payments	282074000.0	operating	Stockhol
563	NOR	NorSun	Clean Technology	272120000.0	operating	Os
153	FIN	Supercell	Games	272000000.0	acquired	Helsir
414	SWE	Recipharm	Biotechnology	210384730.0	operating	SWE - Oth
688	DNK	Genmab	Biotechnology	157000000.0	operating	Copenhag
325	SWE	iZettle	Mobile Payments	108520072.0	operating	Stockhol
24	FIN	Blyk	App Marketing	107764700.0	operating	Helsir
568	NOR	OptiNose	Biotechnology	83500000.0	operating	Os
480	SWE	Truecaller	Curated Web	80100000.0	operating	Stockhol
361	SWE	Nanoradio	Semiconductors	79960000.0	closed	Stockhol
127	FIN	Rovio Entertainment	Mobile Games	76075497.0	operating	Helsir
471	SWE	Tobii Technology	Hardware + Software	71779200.0	operating	Dander
293	SWE	GLO	Nanotechnology	66612432.0	operating	Maln
285	SWE	FlatFrog Laboratories	Hardware + Software	53408000.0	operating	Maln
383	SWE	Orexo	Biotechnology	50600000.0	operating	SWE - Oth
240	SWE	BONESUPPORT	Health Care	50074933.0	operating	Maln
593	NOR	Thin Film Electronics ASA	NFC	47000000.0	operating	Os
793	DNK	Trustpilot	Curated Web	43400000.0	operating	DNK - Oth

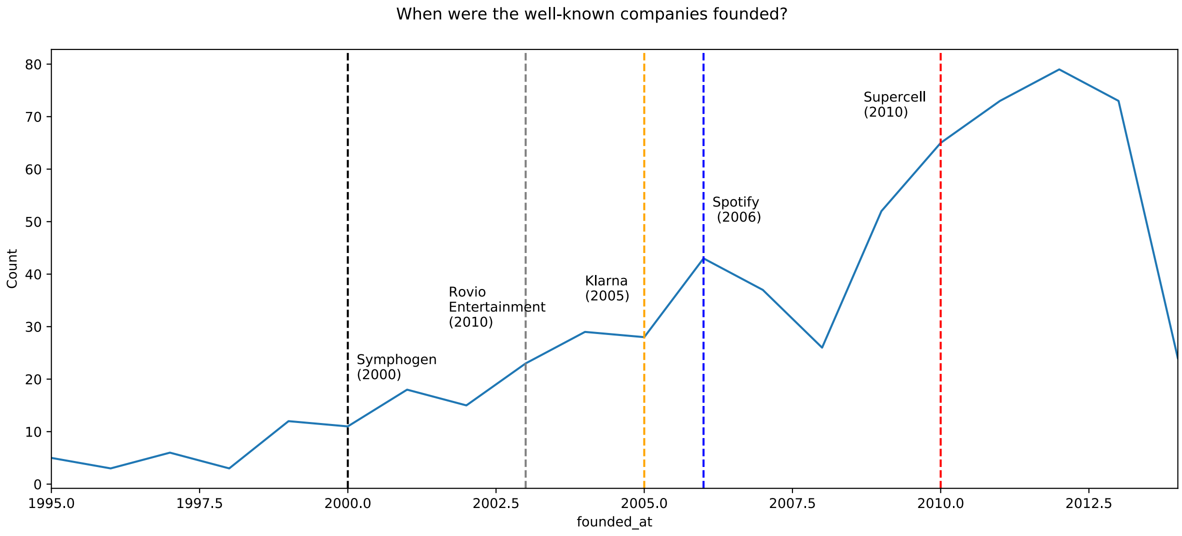
20 rows × 30 columns

In [43]:

```
Spotify_founded_year = investments['founded_at'][investments['name']=="Spotify"].dt.year.values[0]
Symphogen_founded_year  = investments['founded_at'][investments['name']=="Symphogen"].dt.year.values[0]
Klarna_founded_year = investments['founded_at'][investments['name']=="Klarna"].dt.year.values[0]
Supercell_founded_year  = investments['founded_at'][investments['name']=="Supercell"].dt.year.values[0]
Rovio_founded_year  = investments['founded_at'][investments['name']=="Rovio Entertainment"].dt.year.values[0]
```

In [44]:

```
# Comparison of founding dates selected, successful Nordic startups

plt.rcParams['figure.figsize'] = 15,6
investments['name'][investments["founded_at"].dt.year >= 1995].groupby(investments["founded_at"].dt.year).count().plot(kind="line")
plt.ylabel('Count')

plt.axvline(Spotify_founded_year,color='blue',linestyle ="--")
plt.text(Spotify_founded_year+0.15, 50,"Spotify \n (2006)")

plt.axvline(Symphogen_founded_year,color='black',linestyle ="--")
plt.text(Symphogen_founded_year+0.15, 20,"Symphogen \n(2000)")

plt.axvline(Klarna_founded_year,color='orange',linestyle ="--")
plt.text(Klarna_founded_year-1.00, 35,"Klarna \n(2005)")

plt.axvline(Supercell_founded_year,color='red',linestyle ="--")
plt.text(Supercell_founded_year-1.30, 70,"Supercell \n(2010)")

plt.axvline(Rovio_founded_year,color='grey',linestyle ="--")
plt.text(Rovio_founded_year-1.30, 30,"Rovio \nEntertainment \n(2010)")

plt.title("When were the well-known companies founded?", fontdict=None, position= [0.48,1.05])
plt.show()
```



Visualisation: Public grants, debt and a word about finland

In [45]:

```
# Total number of Nordic startups per market/industry with more than 1 million USD investment

most_high= investments[['market', 'name']][investments['funding_total_usd'] > 1000000].groupby(['market'],
                                        as_index=False).count().sort_values('name', ascending=False)
most_high.head(20)
top20 = most_high.head(20)
top20
```

Out[45]:

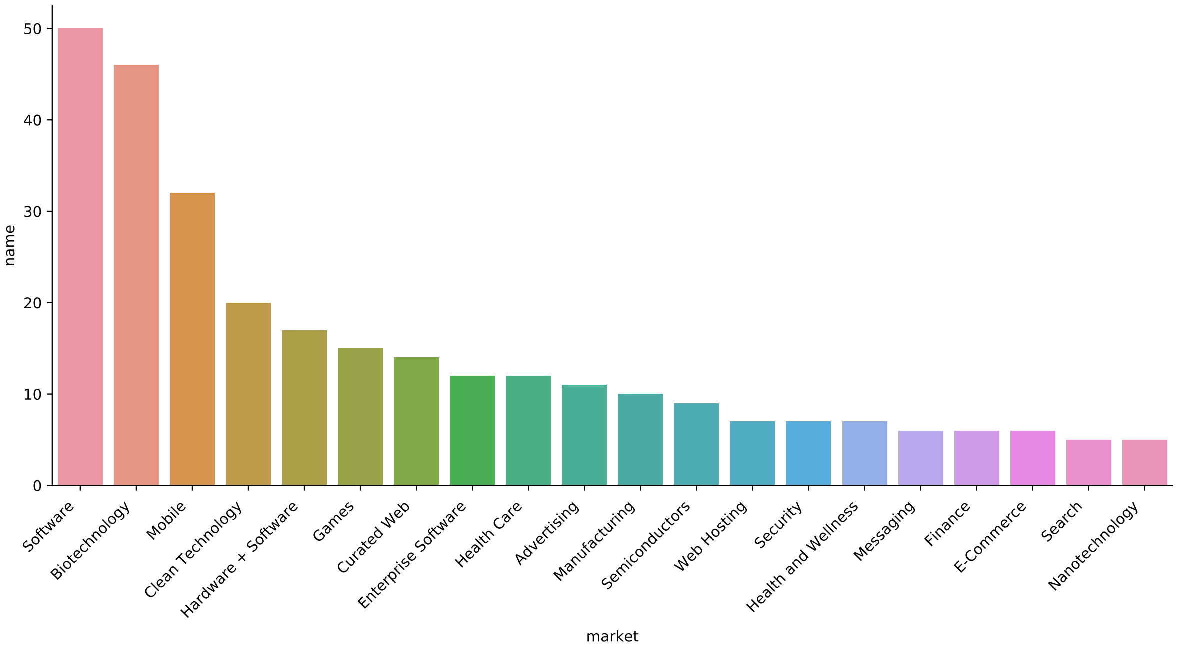
	market	name
87	Software	50
8	Biotechnology	46
57	Mobile	32
11	Clean Technology	20
41	Hardware + Software	17
38	Games	15
18	Curated Web	14
28	Enterprise Software	12
42	Health Care	12
0	Advertising	11
50	Manufacturing	10
82	Semiconductors	9
95	Web Hosting	7
80	Security	7
43	Health and Wellness	7
56	Messaging	6
34	Finance	6
25	E-Commerce	6
79	Search	5
62	Nanotechnology	5

In [46]:

```
Nordm = sns.catplot(x="market", y="name",  kind="bar", data=top20, height=5.27
, aspect=11.7/5.27)
Nordm.set_xticklabels(rotation=45, horizontalalignment='right')
```

Out[46]:

```
<seaborn.axisgrid.FacetGrid at 0x10f6e30b8>
```



In [47]:

```
# Total number of startups per Finnish market with more than 1 million USD investment

fin_high= investments[investments['country_code'] == 'FIN']

finh = fin_high[['market', 'name']][fin_high['funding_total_usd'] > 1000000].groupby(['market'],
                                        as_index=False).count().sort_values('name', ascending=False)

Finl10=finh.head(10)
Finl10
```

Out[47]:

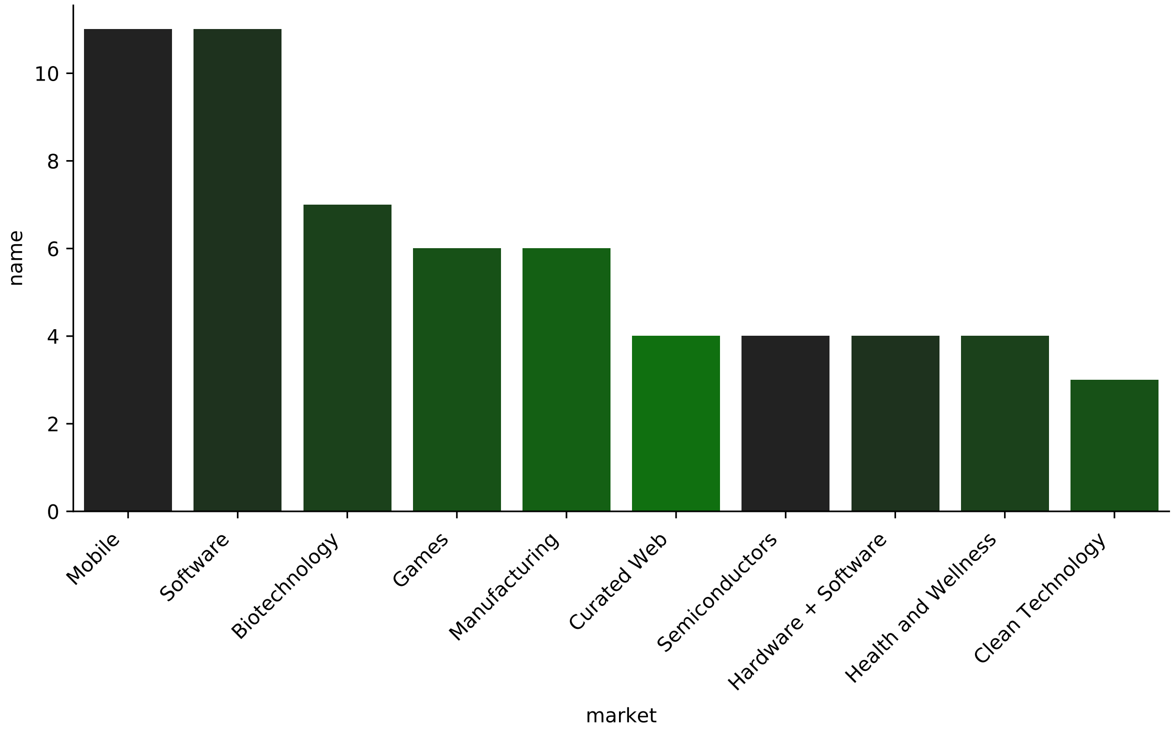
	market	name
34	Mobile	11
45	Software	11
7	Biotechnology	7
24	Games	6
31	Manufacturing	6
12	Curated Web	4
42	Semiconductors	4
25	Hardware + Software	4
27	Health and Wellness	4
9	Clean Technology	3

In [48]:

```
Finm = sns.catplot(x="market", y="name",  kind="bar", data=Finl10, height=4.27
, aspect=8.7/4.27,palette=sns.dark_palette("green"))
Finm.set_xticklabels(rotation=45, horizontalalignment='right')
```

Out[48]:

```
<seaborn.axisgrid.FacetGrid at 0x122fa4be0>
```



In [49]:

```
# Total number of startups per Swedish market with more than 1 million USD investment

swe_high= investments[investments['country_code'] == 'SWE']

sweh = swe_high[['market', 'name']][swe_high['funding_total_usd'] > 1000000].groupby(['market'],
                                        as_index=False).count().sort_values('name', ascending=False)

SWL10=sweh.head(10)
SWL10
```

Out[49]:

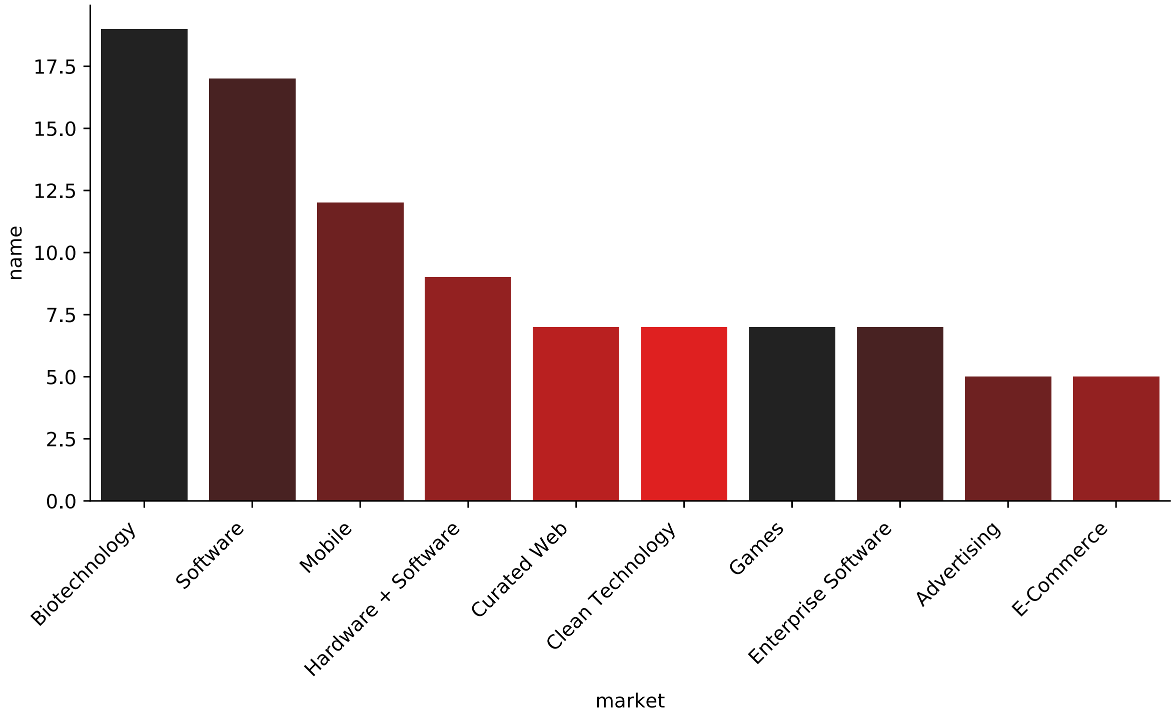
	market	name
3	Biotechnology	19
47	Software	17
32	Mobile	12
21	Hardware + Software	9
10	Curated Web	7
5	Clean Technology	7
19	Games	7
12	Enterprise Software	7
0	Advertising	5
11	E-Commerce	5

In [50]:

```
SWm = sns.catplot(x="market", y="name",  kind="bar", data=SWL10, height=4.27,
aspect=8.7/4.27,palette=sns.dark_palette("red"))

SWm.set_xticklabels(rotation=45, horizontalalignment='right')
```

Out[50]:

```
<seaborn.axisgrid.FacetGrid at 0x122c58e10>
```



In [51]:

```
# Total number of startups per Norwegian market with more than 1 million USD investment

nor_high= investments[investments['country_code'] == 'NOR']

norh = nor_high[['market', 'name']][nor_high['funding_total_usd'] > 1000000].groupby(['market'],
                                        as_index=False).count().sort_values('name', ascending=False)

Nor10 =norh.head(10)
Nor10
```

Out[51]:

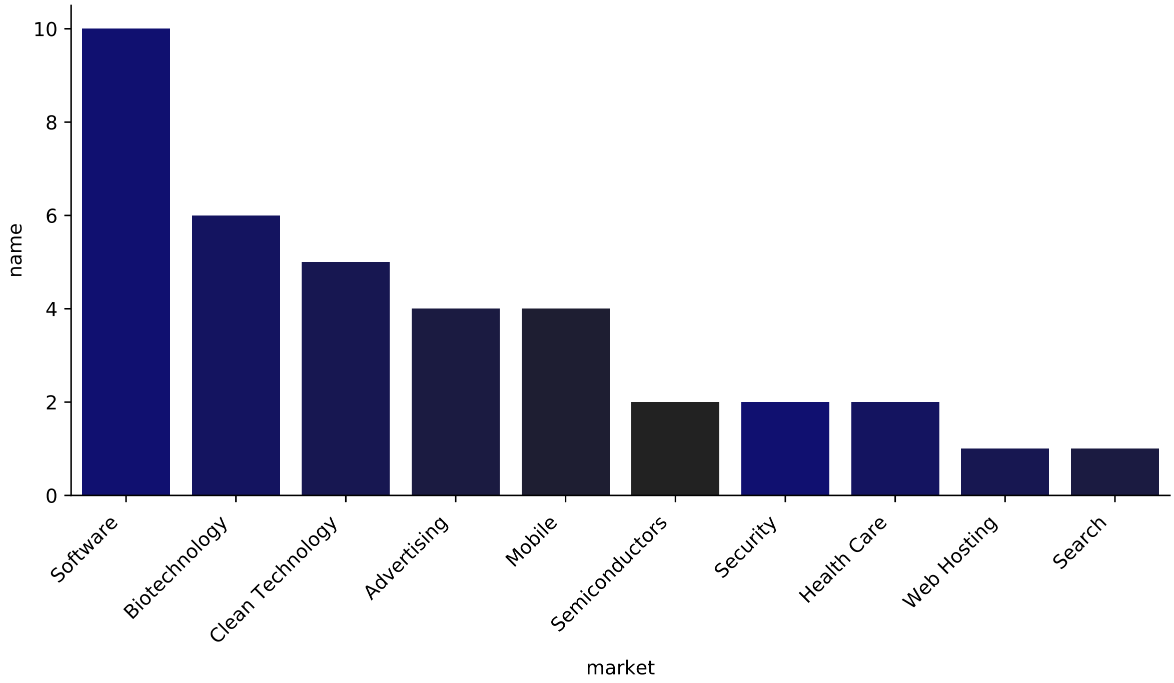
	market	name
23	Software	10
2	Biotechnology	6
3	Clean Technology	5
0	Advertising	4
15	Mobile	4
22	Semiconductors	2
21	Security	2
12	Health Care	2
24	Web Hosting	1
20	Search	1

In [52]:

```
NRm = sns.catplot(x="market", y="name",  kind="bar", data=Nor10, height=4.27,
aspect=8.7/4.27,palette=sns.dark_palette("navy", reverse=True))

NRm.set_xticklabels(rotation=45, horizontalalignment='right')
```

Out[52]:

```
<seaborn.axisgrid.FacetGrid at 0x122dd0e80>
```



In [53]:

```
# Total number of startups per Danish market with more than 1 million USD investment

dnk_high= investments[investments['country_code'] == 'DNK']

dnkh = dnk_high[['market', 'name']][dnk_high['funding_total_usd'] > 1000000].groupby(['market'],
                                                   as_index=False).count().sort_values('name', ascending=False)

DN10=dnkh.head(10)
DN10
```

Out[53]:

	market	name
1	Biotechnology	14
37	Software	11
2	Clean Technology	5
21	Mobile	5
12	Finance	3
5	Curated Web	3
16	Health Care	3
15	Hardware + Software	3
13	Games	2
31	SaaS	2

In [ ]:

```
DNm = sns.catplot(x="market", y="name",  kind="bar", data=DN10, height=4.27, aspect=8.7/4.27,palette=sns.diverging_palette(255, 133, l=60, n=7, center="dark"))

DNm.set_xticklabels(rotation=45, horizontalalignment='right')
```

In [55]:

```
# Total number of startups per Icelandic market with more than 1 million USD i
nvestment

isl_high= investments[investments['country_code'] == 'ISL']

islh = isl_high[['market', 'name']][isl_high['funding_total_usd'] > 1000000].g
roupby(['market'],
                                                      as_index=False).count().sort_values('n
ame', ascending=False)

IS10=islh.head(10)
IS10
```

Out[55]:

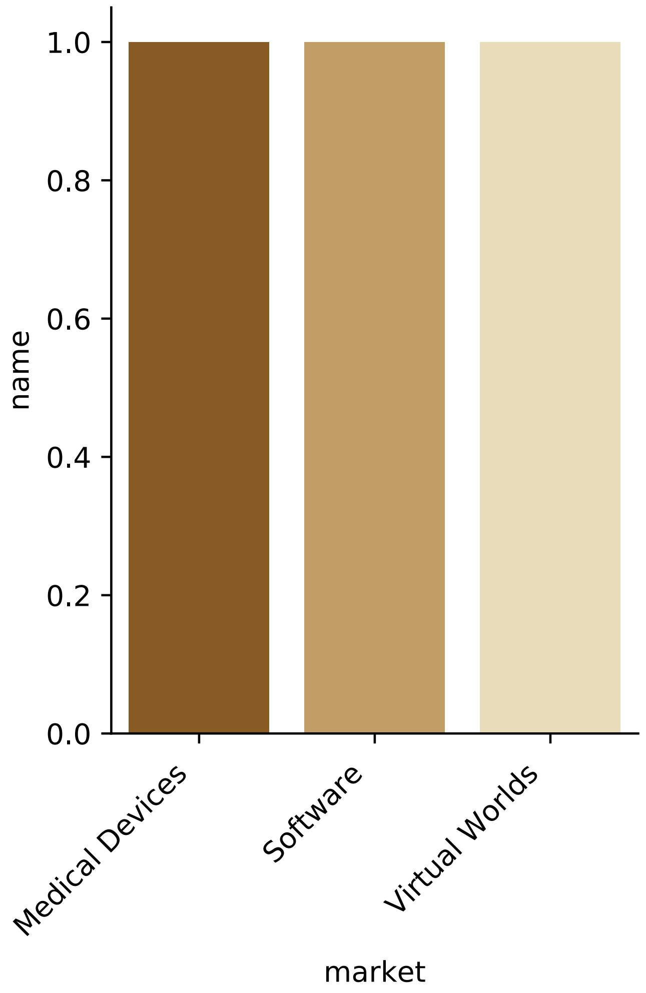
	market	name
0	Medical Devices	1
1	Software	1
2	Virtual Worlds	1

In [56]:

```
ISm = sns.catplot(x="market", y="name",  kind="bar", data=IS10, height=4.27, a
spect=6.7/8.27,palette=sns.color_palette("BrBG", 7))

ISm.set_xticklabels(rotation=45, horizontalalignment='right')
```

Out[56]:

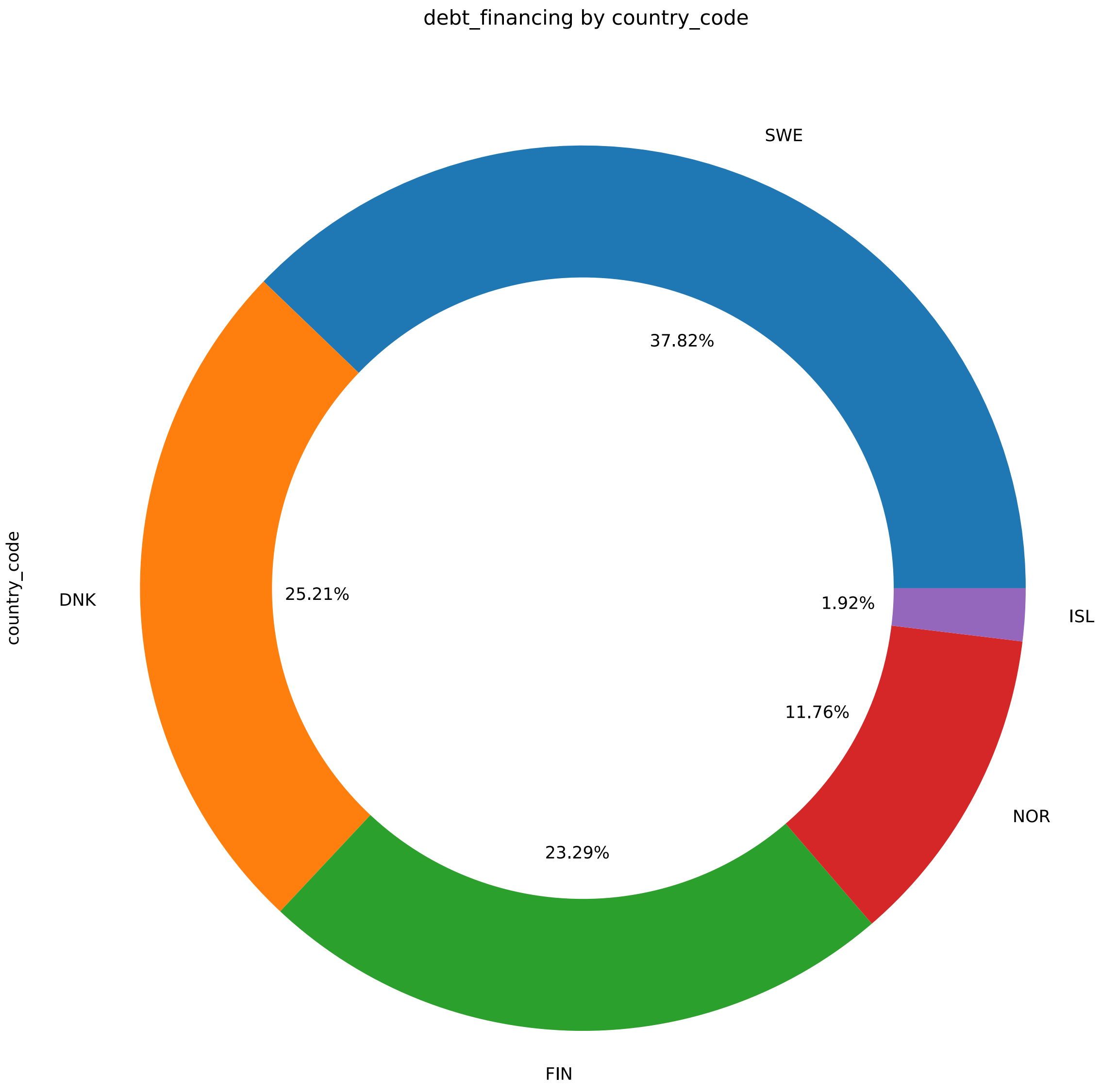
```
<seaborn.axisgrid.FacetGrid at 0x122f125c0>
```



In [57]:

```
# 833 largest  debt_financing in the Nordic startups


LDF = investments.nlargest(833,'debt_financing')
ax = LDF.country_code.value_counts().plot(kind='pie',autopct='%.2f%%',figsize=
(12,12))
add_circle = plt.Circle((0,0),0.7,color='white')
fig=plt.gcf()
fig.gca().add_artist(add_circle)
ax.set_title(' debt_financing by country_code')
```

```
Out[57]:

Text(0.5, 1.0, ' debt_financing by country_code')
```

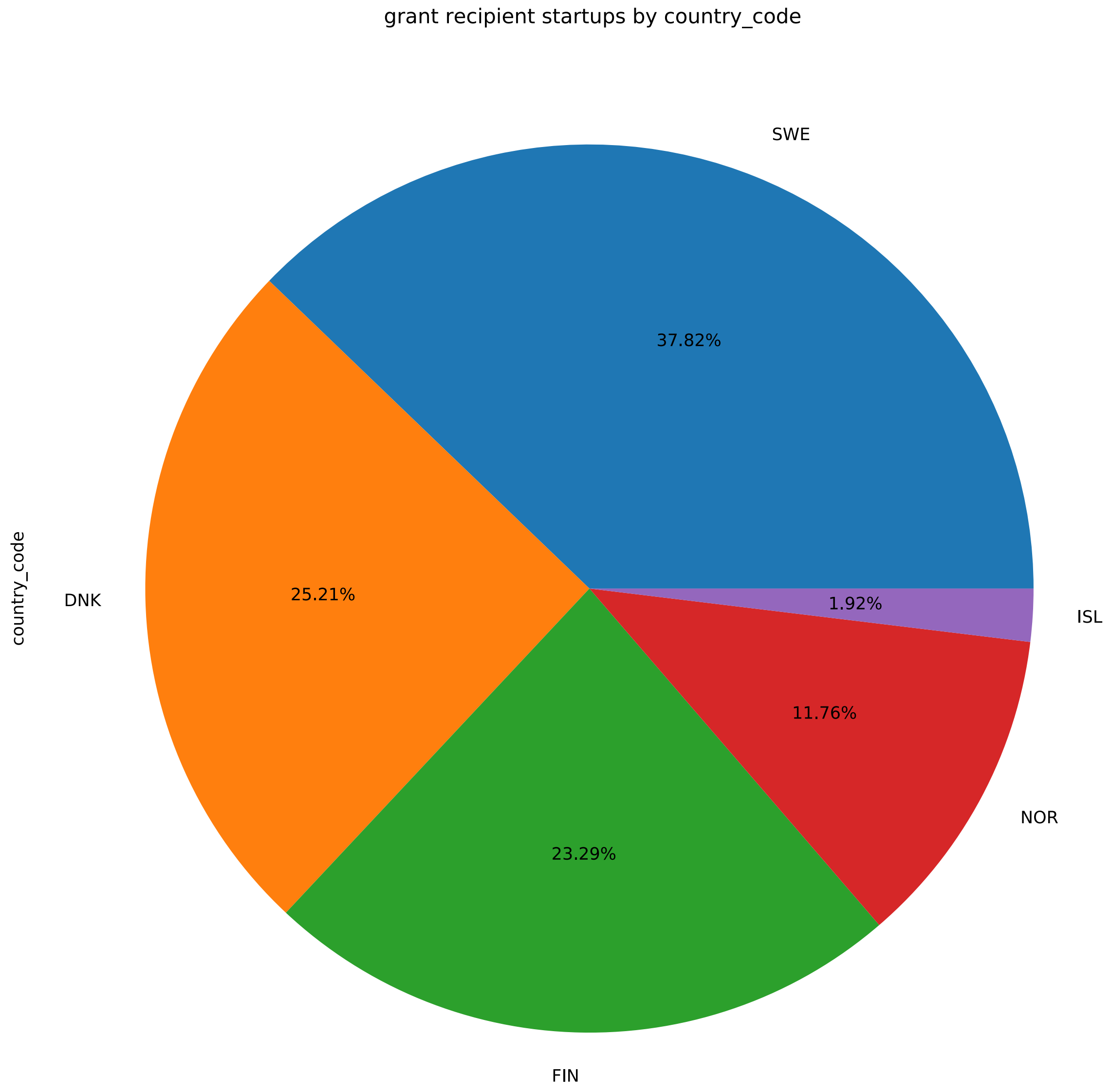


In [58]:

```
# grant recipients by country_code

LG = investments.nlargest(833,'grant')
ax = LG.country_code.value_counts().plot(kind='pie',autopct='%.2f%%',figsize=(
12,12))
figG=plt.gcf()
figG.gca()
ax.set_title(' grant recipient startups by country_code')
```

Out[58]:

```
Text(0.5, 1.0, ' grant recipient startups by country_code')
```

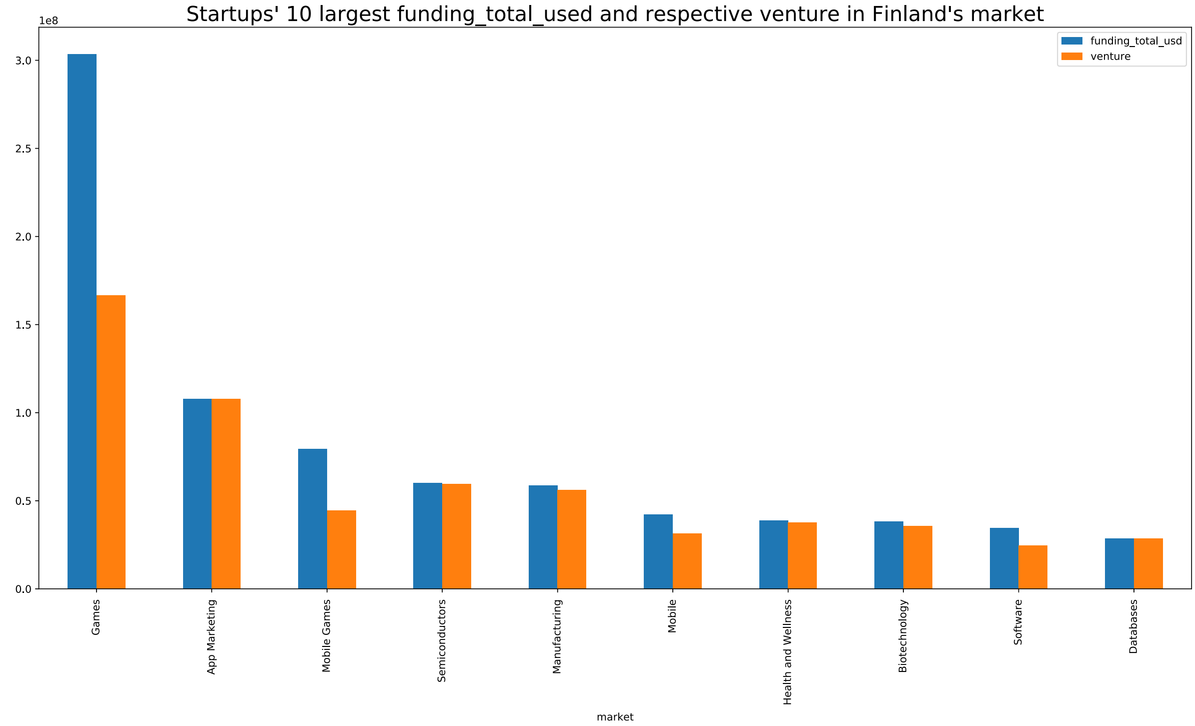


In [59]:

```
# 10 largest funding_total_used and respective venture in Finland market
gbf=investments[(investments['country_code'] == 'FIN')]

gg= gbf.groupby('market').sum()
LF = gg.nlargest(10,'funding_total_usd')
fg=LF.plot(kind ='bar', y=['funding_total_usd','venture'], figsize=(20,10))

fg.set_title('Startups\' 10 largest funding_total_used and respective venture
in Finland\'s market',fontsize=(20))
```

Out[59]:

```
Text(0.5, 1.0, "Startups' 10 largest funding_total_used and respec
tive venture in Finland's market")
```

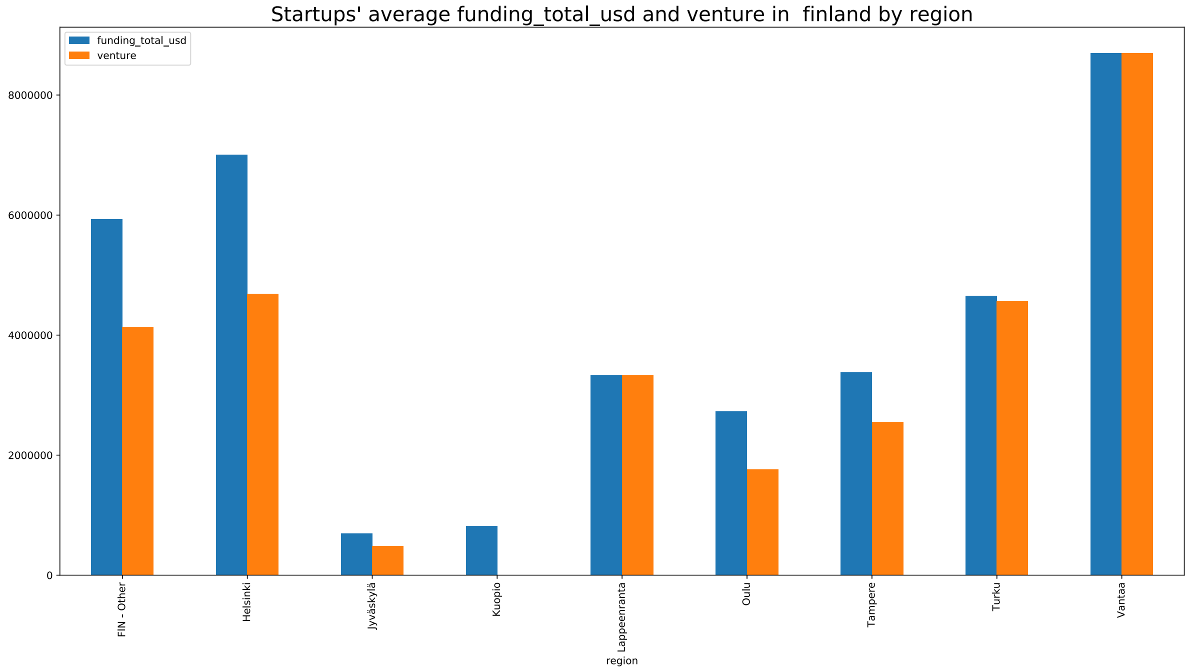


In [60]:

```
# The average fundding_total_used and respective venture by region in Finland
gbf=investments[(investments['country_code'] == 'FIN')]
gg= gbf.groupby('region').mean()
fg=gg.plot(kind ='bar', y=['funding_total_usd','venture'], figsize=(20,10))

fg.set_title('Startups\' average funding_total_usd and venture in  finland by
region',fontsize=(20))
```

Out[60]:

```
Text(0.5, 1.0, "Startups' average funding_total_usd and venture in
finland by region")
```

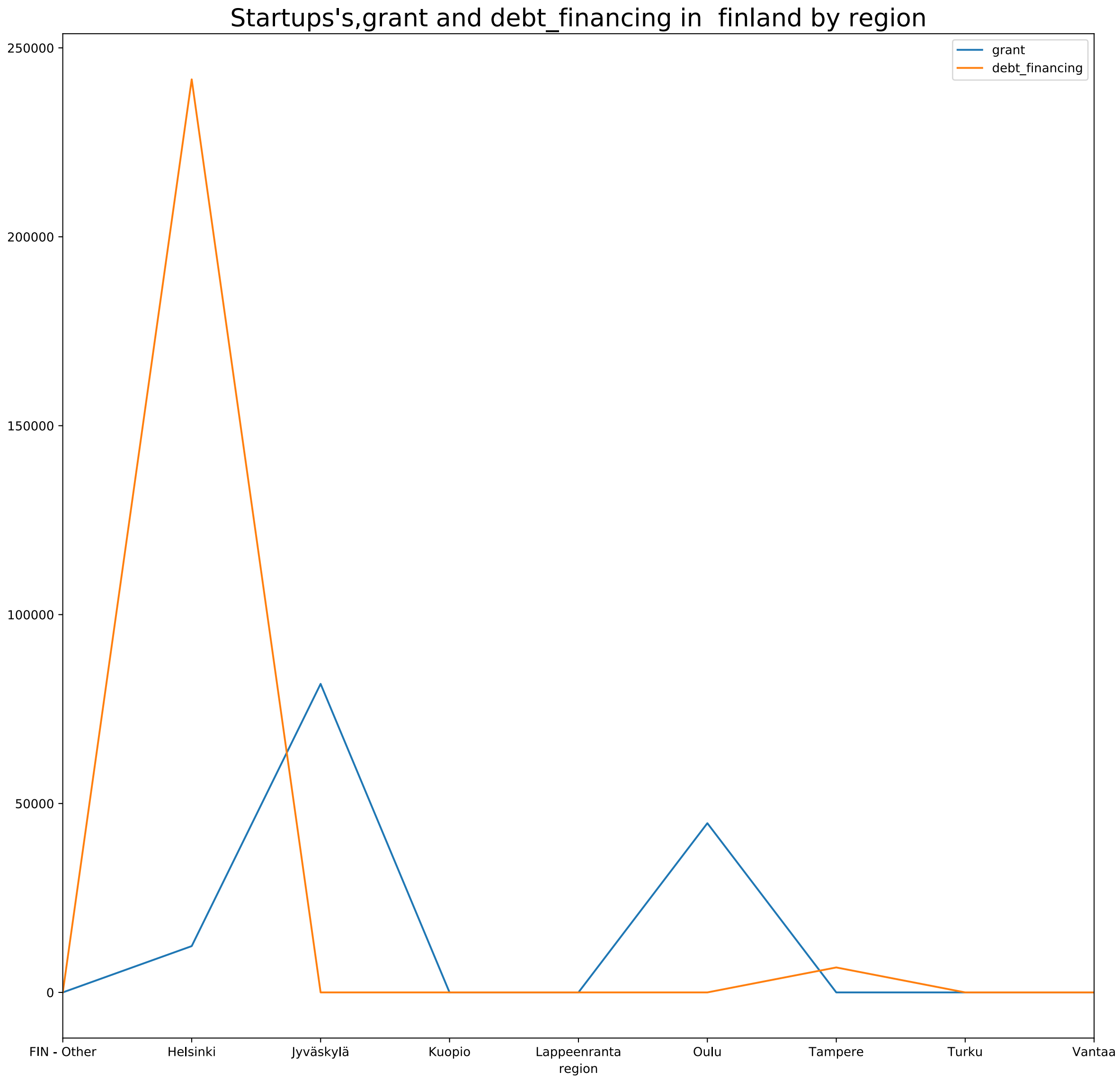


In [61]:

```
# The average funding from grant and debt_financing in Finland by region

gbf=investments[(investments['country_code'] == 'FIN')]
rg= gbf.groupby('region').mean()
fr=rg.plot(kind ='line', y=['grant','debt_financing'], figsize=(15,15))

fr.set_title('Startups\'s,grant and debt_financing in  finland by region',font
size=(20))
```

Out[61]:

```
Text(0.5, 1.0, "Startups's,grant and debt_financing in  finland by
region")
```

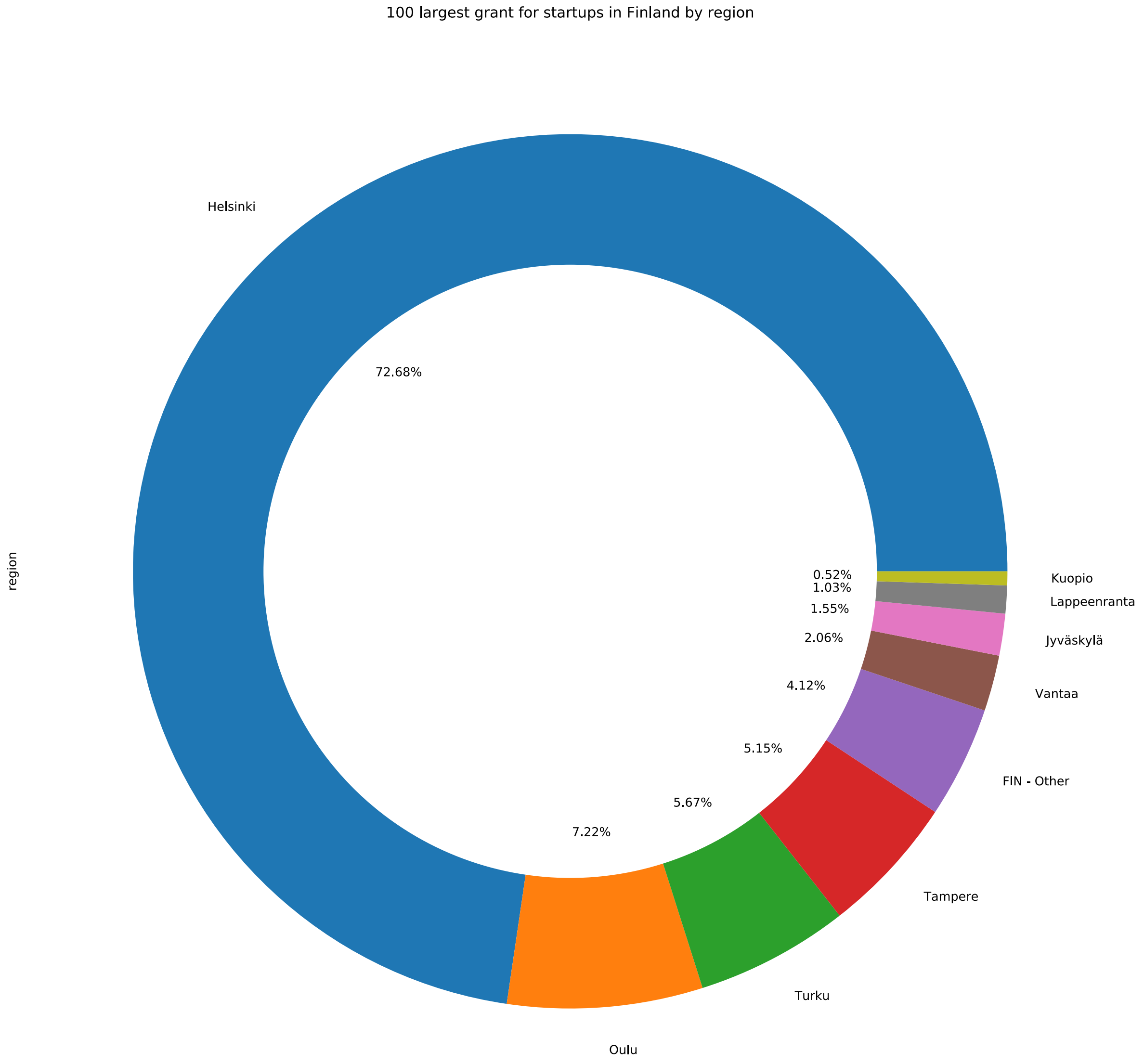


In [62]:

```
# 200 largest grant for startups in Finland by region'
fin =investments[(investments['country_code'] == 'FIN')]
Lfd = fin.nlargest(200,'grant')
ax = Lfd.region.value_counts().plot(kind='pie',autopct='%.2f%%',figsize=(16,20
))
add_circle = plt.Circle((0,0),0.7,color='white')
figd=plt.gcf()
figd.gca().add_artist(add_circle)
ax.set_title('100 largest grant for startups in Finland by region')
```

Out[62]:

```
Text(0.5, 1.0, '100 largest grant for startups in Finland by regio
n')
```



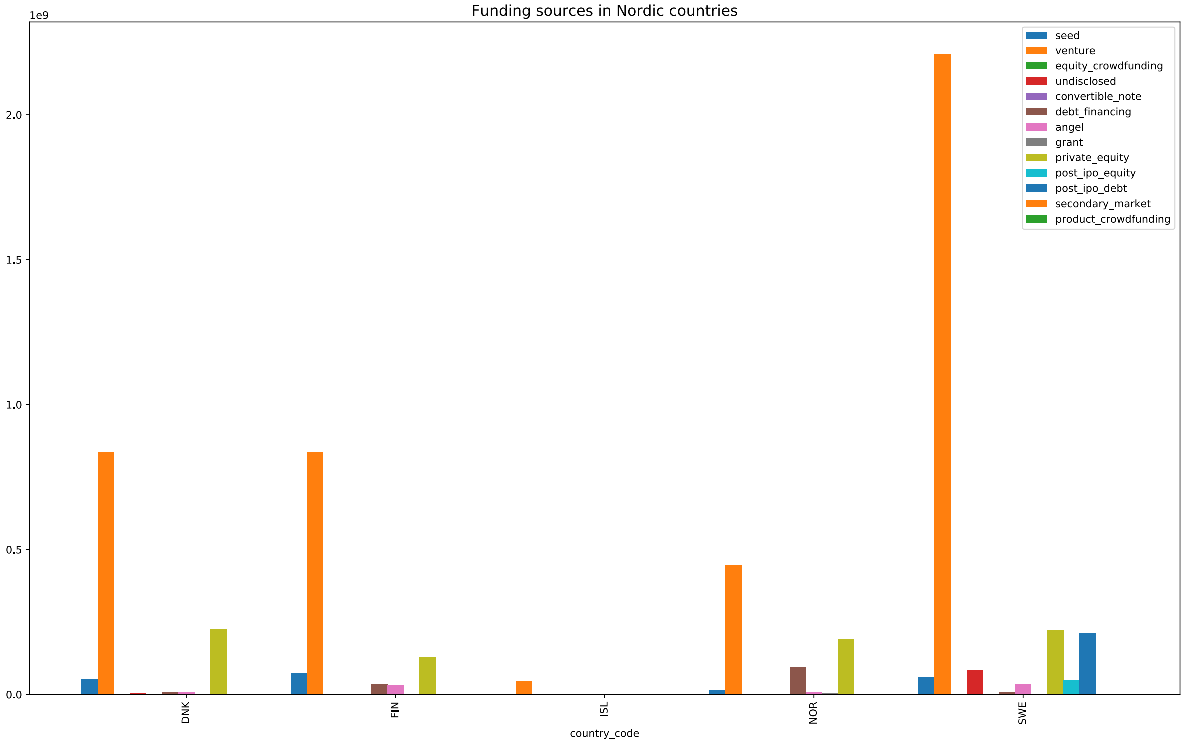
Visualisation: Success stories, funding rounds and private funding

In [63]:

```
#checking differences in funding sources among the countries.
investments.groupby('country_code').sum()[['seed',
                                           'venture',
                                           'equity_crowdfunding',
                                           'undisclosed',
                                           'convertible_note',
                                           'debt_financing',
                                           'angel',
                                           'grant',
                                           'private_equity',
                                           'post_ipo_equity',
                                           'post_ipo_debt',
                                           'secondary_market',
                                           'product_crowdfunding']].plot(kind
= 'bar', figsize = (20,12), width = 1)
plt.title('Funding sources in Nordic countries', size = 'x-large')
```

Out[63]:

```
Text(0.5, 1.0, 'Funding sources in Nordic countries')
```

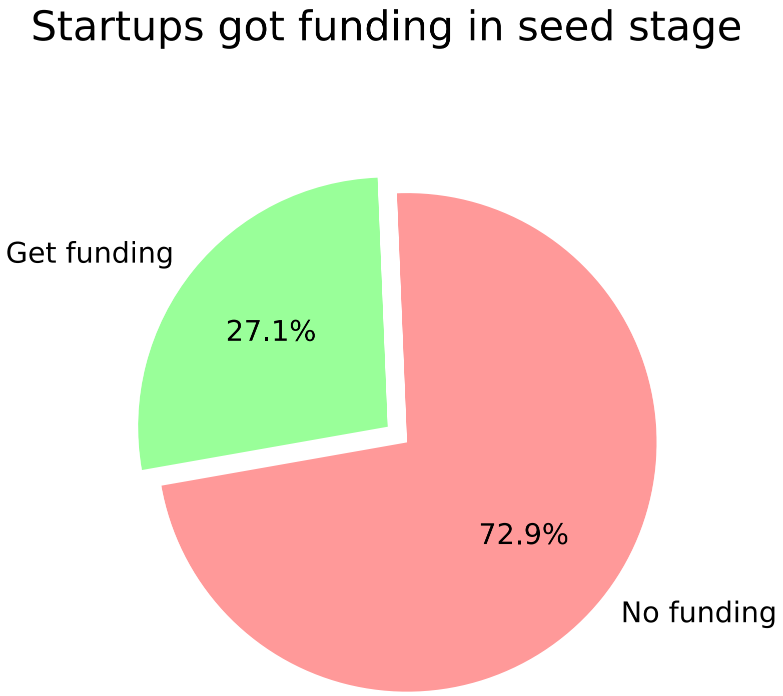


In [64]:

```
investments['funding_in_seed'] = investments['seed'].map(lambda x :1  if x > 0
else 0)
```

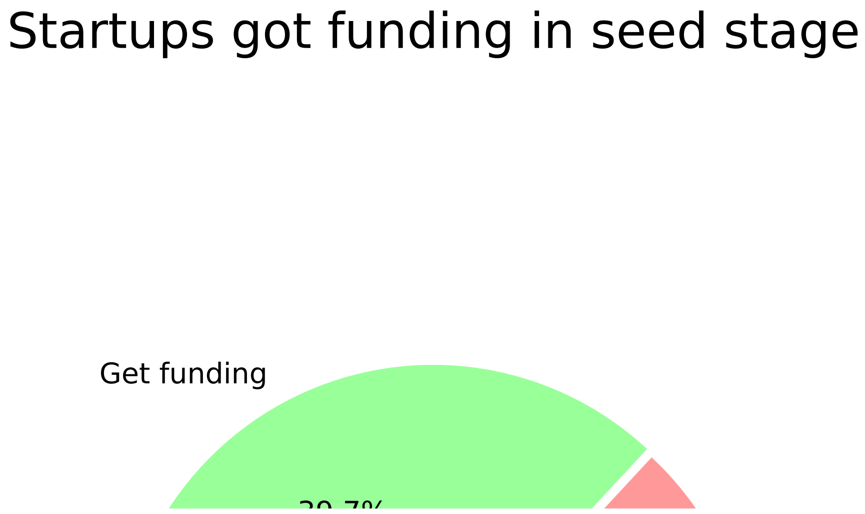
```
In [65]:
```

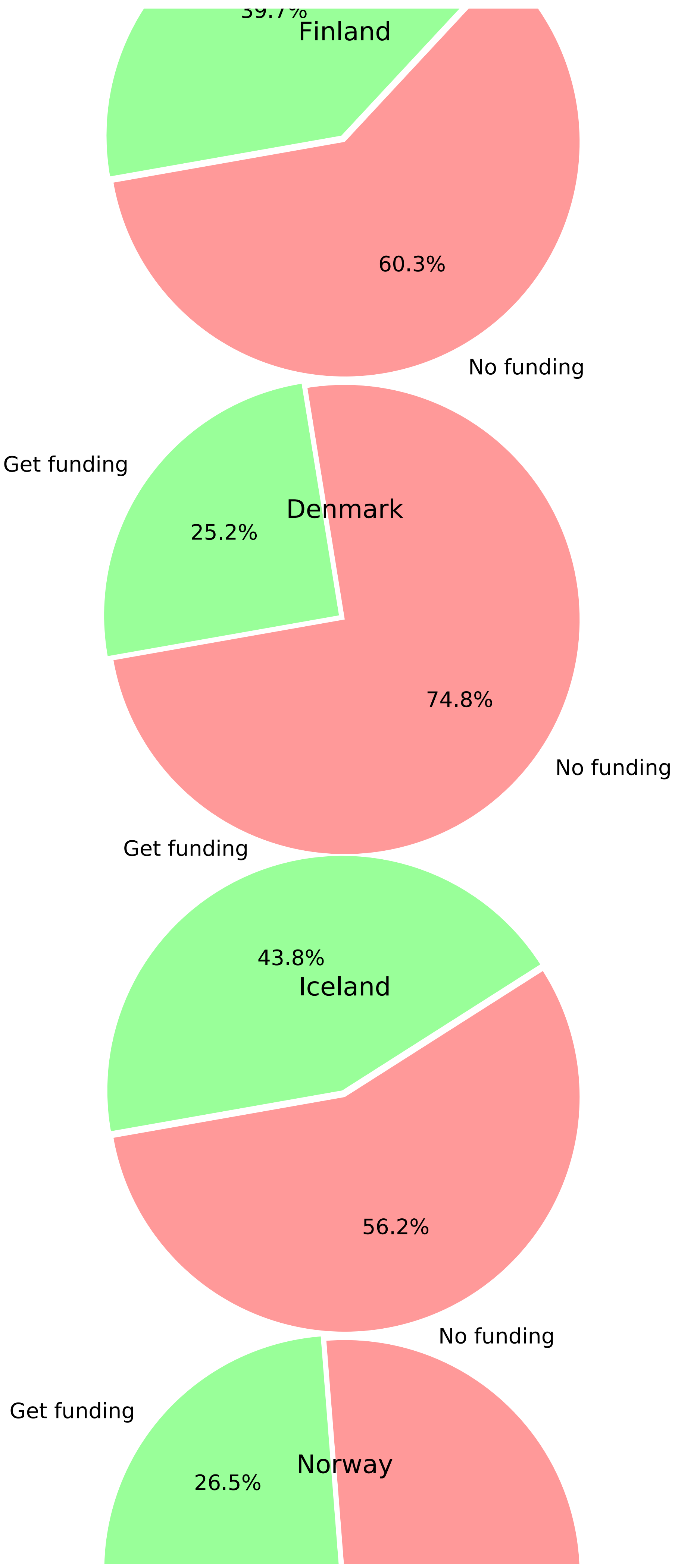
```
plt.rcParams['figure.figsize'] =4,4
labels = ['No funding','Get funding']
sizes = investments['funding_in_seed'].value_counts().tolist()
explode = (0, 0.1)
colors =  ['#ff9999','#99ff99']

plt.pie(sizes, explode = explode, colors = colors ,labels=labels, autopct='%1.
1f%%',
        shadow=False, startangle=190)
plt.axis('equal')
plt.tight_layout()
plt.title("Startups got funding in seed stage", fontdict=None, position= [0.48
,1.1], size = 'x-large')

plt.show()
```

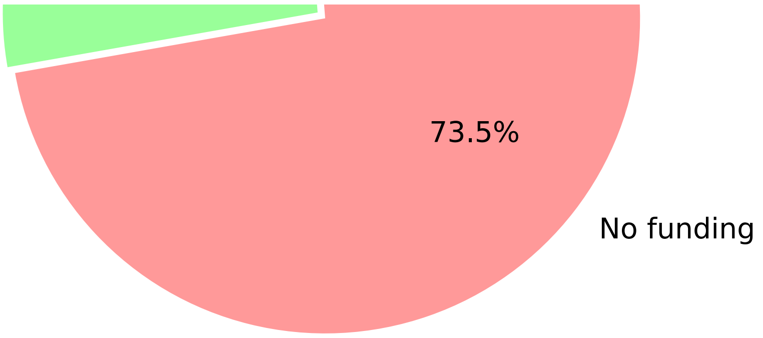


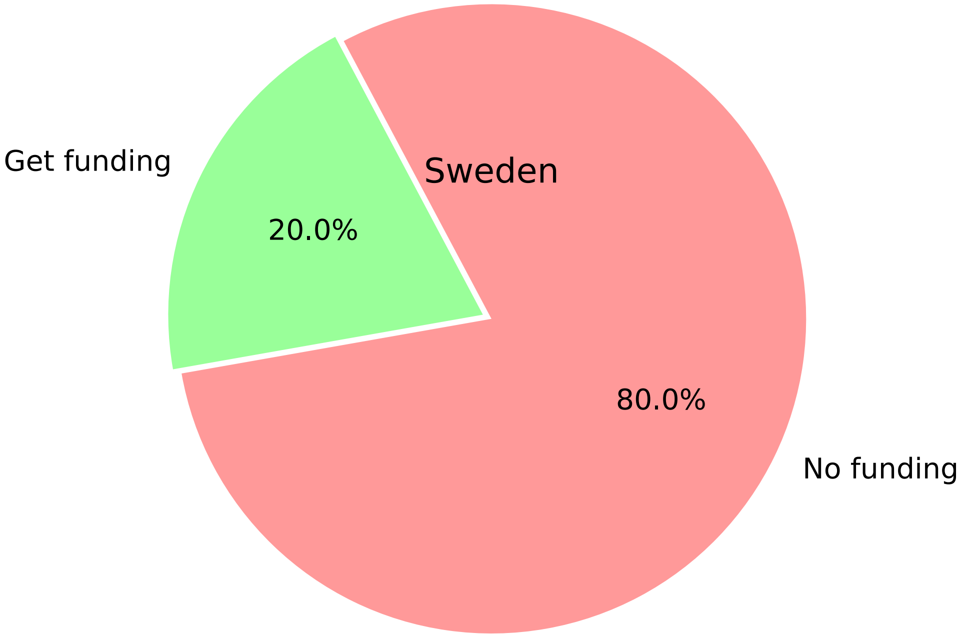
In [66]:

```
grouped_by_country = investments.groupby('country_code')
fin = grouped_by_country.get_group('FIN')
dnk = grouped_by_country.get_group('DNK')
isl = grouped_by_country.get_group('ISL')
nor = grouped_by_country.get_group('NOR')
swe = grouped_by_country.get_group('SWE')

fig, ax = plt.subplots(nrows=5, ncols=1, figsize = (1.5,20))

labels = ['No funding','Get funding']
sizes_fin = fin['funding_in_seed'].value_counts().tolist()
sizes_dnk = dnk['funding_in_seed'].value_counts().tolist()
sizes_isl = isl['funding_in_seed'].value_counts().tolist()
sizes_nor = nor['funding_in_seed'].value_counts().tolist()
sizes_swe = swe['funding_in_seed'].value_counts().tolist()
explode = (0, 0.1)
colors =  ['#ff9999','#99ff99']

ax[0].set_title("Finland")
ax[0].pie(sizes_fin, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[1].set_title("Denmark")
ax[1].pie(sizes_dnk, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[2].set_title("Iceland")
ax[2].pie(sizes_isl, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[3].set_title("Norway")
ax[3].pie(sizes_nor, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[4].set_title("Sweden")
ax[4].pie(sizes_swe, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
fig.suptitle('Startups got funding in seed stage' , size = 'xx-large')
plt.show()
```




39.7%
Finland
60.3%
No funding
Get funding
Denmark
25.2%
74.8%
No funding
Get funding
43.8%
Iceland
56.2%
No funding
Get funding
Norway
26.5%



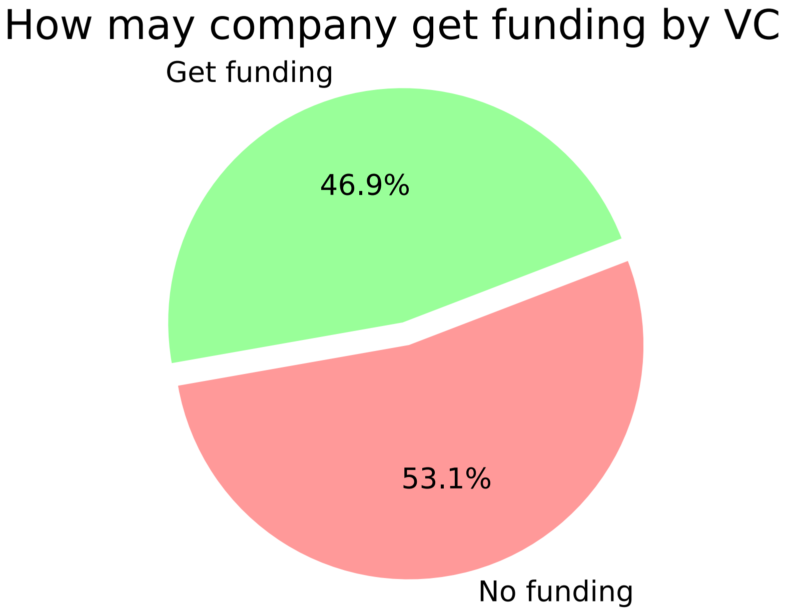


In [67]:

```
investments['funding_vc'] = investments['venture'].map(lambda v :1  if v > 0 else 0)
```

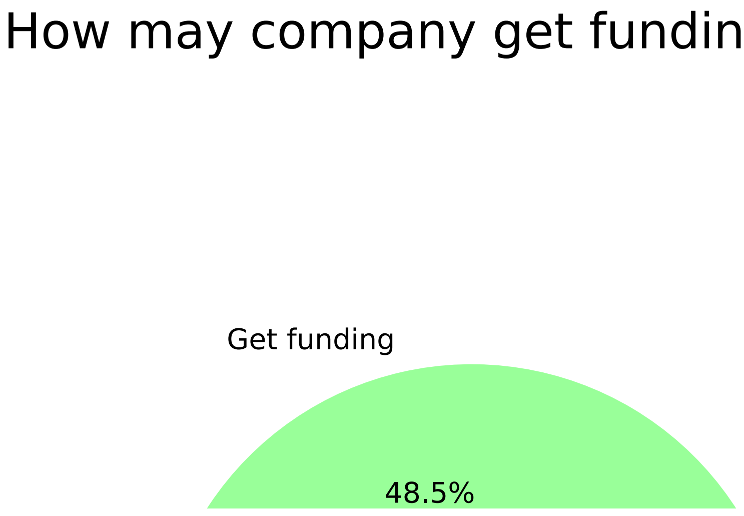
In [68]:

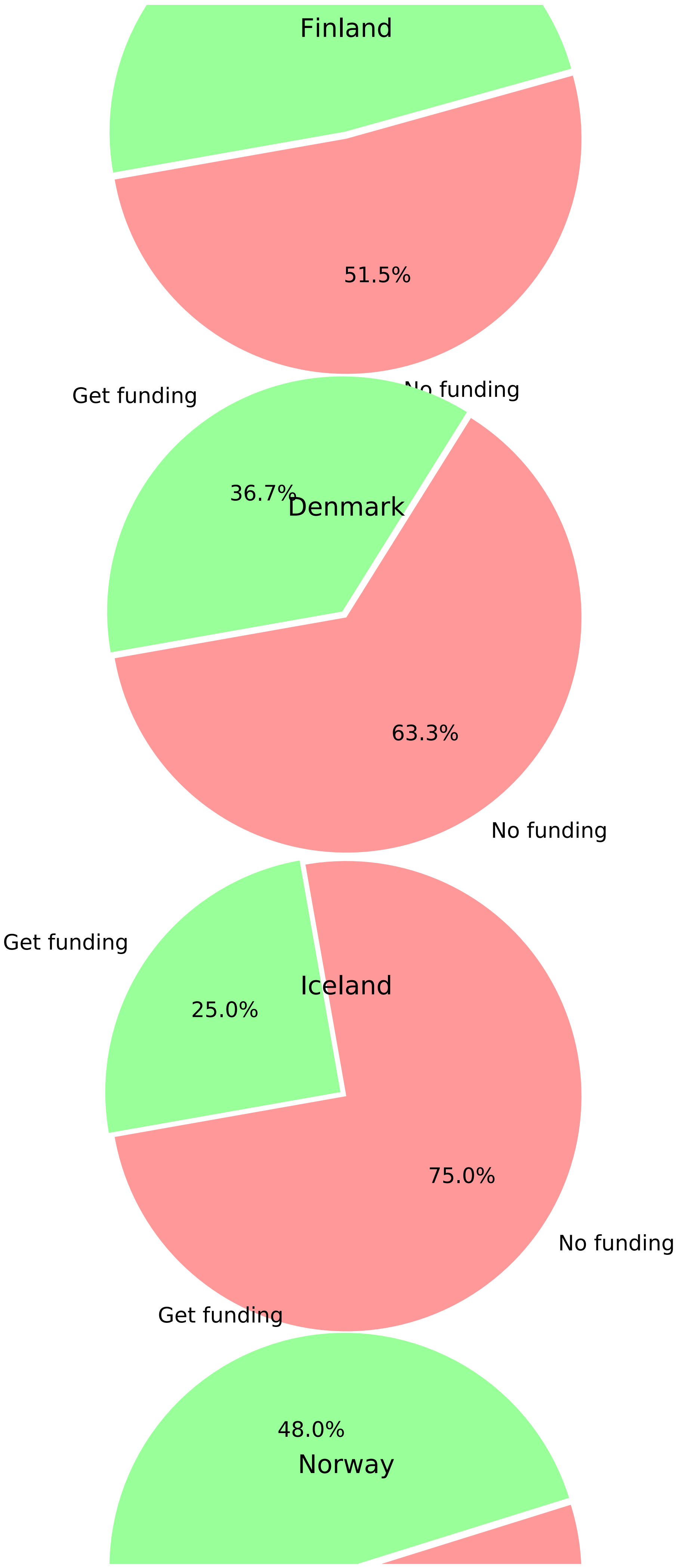
```
plt.rcParams['figure.figsize'] =3,3
labels = ['No funding','Get funding']
sizes = investments['funding_vc'].value_counts().tolist()
explode = (0, 0.1)
colors =  ['#ff9999','#99ff99']

plt.pie(sizes, explode = explode, colors = colors ,labels=labels, autopct='%1.
1f%%',
        shadow=False, startangle=190)
plt.axis('equal')
plt.tight_layout()
plt.title("How may company get funding by VC", fontdict=None, position= [0.48,
1.1], size = 'x-large')

plt.show()
```

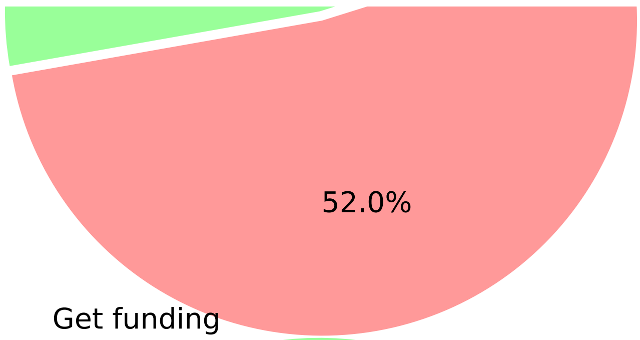


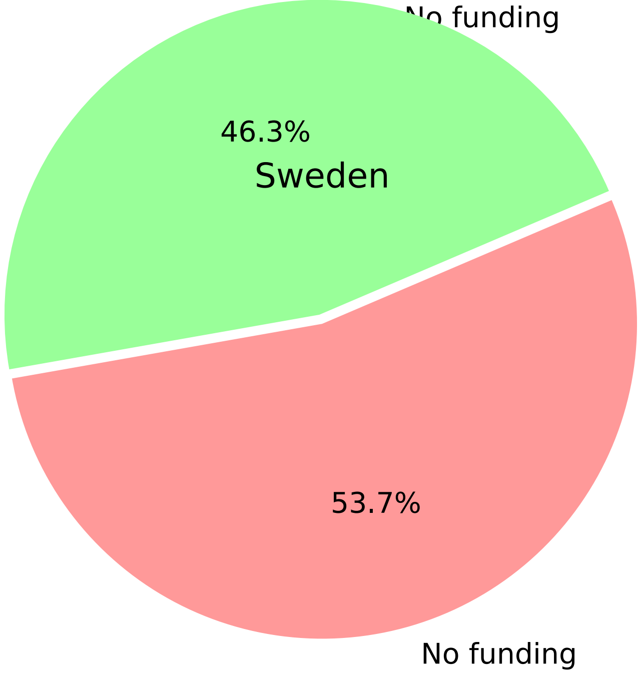
In [69]:

```
grouped_by_country = investments.groupby('country_code')
fin = grouped_by_country.get_group('FIN')
dnk = grouped_by_country.get_group('DNK')
isl = grouped_by_country.get_group('ISL')
nor = grouped_by_country.get_group('NOR')
swe = grouped_by_country.get_group('SWE')

fig, ax = plt.subplots(nrows=5, ncols=1, figsize = (1.5,20))

labels = ['No funding','Get funding']
sizes_fin = fin['funding_vc'].value_counts().tolist()
sizes_dnk = dnk['funding_vc'].value_counts().tolist()
sizes_isl = isl['funding_vc'].value_counts().tolist()
sizes_nor = nor['funding_vc'].value_counts().tolist()
sizes_swe = swe['funding_vc'].value_counts().tolist()
explode = (0, 0.1)
colors =  ['#ff9999','#99ff99']

ax[0].set_title("Finland")
ax[0].pie(sizes_fin, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[1].set_title("Denmark")
ax[1].pie(sizes_dnk, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[2].set_title("Iceland")
ax[2].pie(sizes_isl, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[3].set_title("Norway")
ax[3].pie(sizes_nor, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[4].set_title("Sweden")
ax[4].pie(sizes_swe, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
fig.suptitle("How may company get funding by VC" , size = 'xx-large')
plt.show()
```




Finland
51.5%
Get funding
No funding
36.7%
Denmark
63.3%
No funding
Get funding
Iceland
25.0%
75.0%
No funding
Get funding
48.0%
Norway



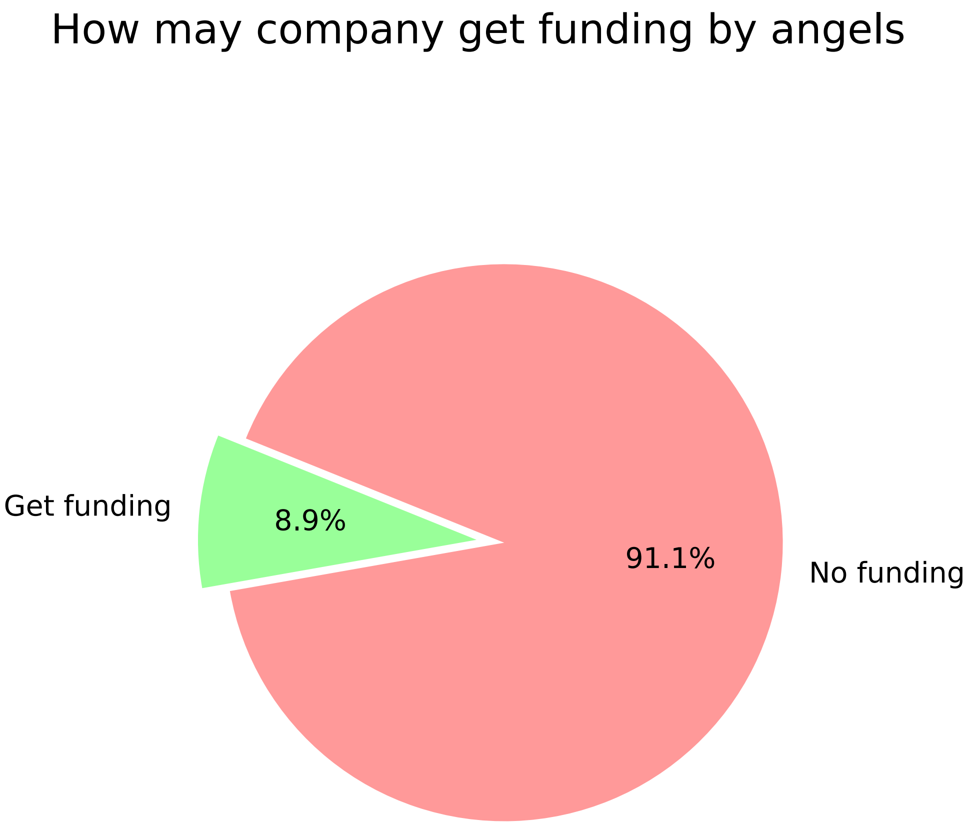


In [70]:

```
investments['funding_angel'] = investments['angel'].map(lambda a :1  if a > 0
else 0)
```

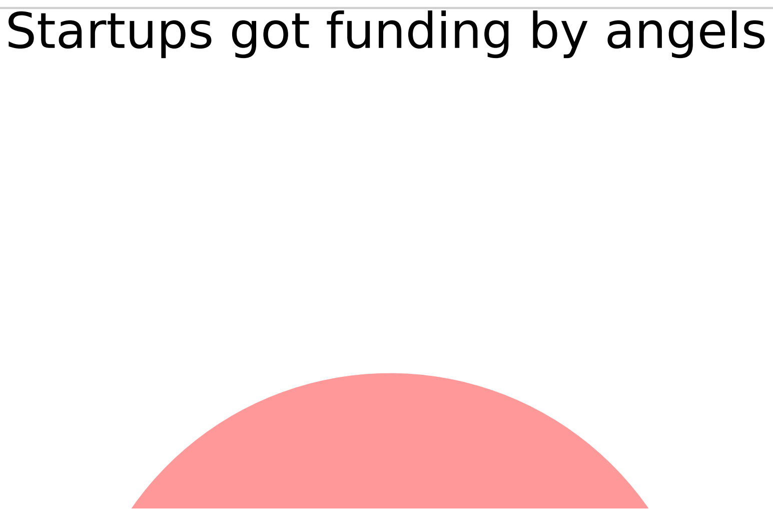
In [71]:

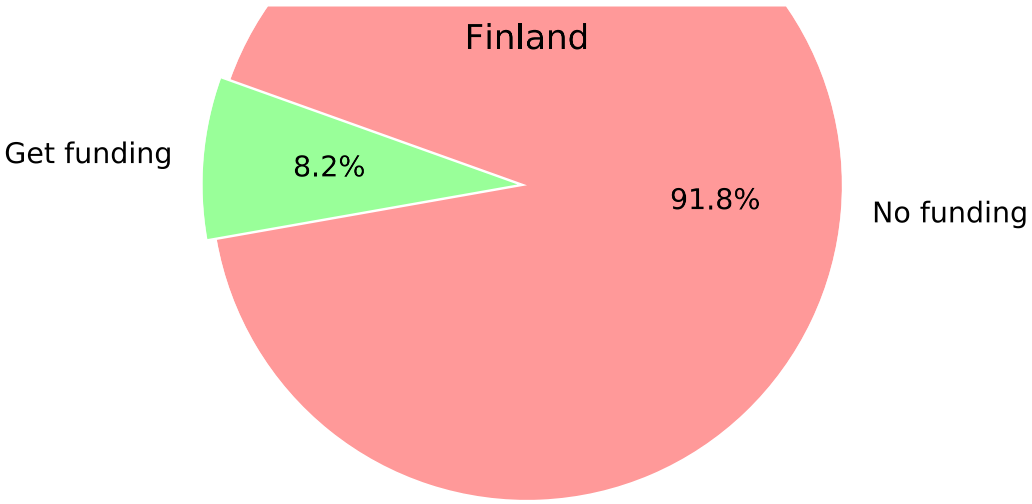
```
plt.rcParams['figure.figsize'] =5,5
labels = ['No funding','Get funding']
sizes = investments['funding_angel'].value_counts().tolist()
explode = (0, 0.1)
colors =  ['#ff9999','#99ff99']

plt.pie(sizes, explode = explode, colors = colors ,labels=labels, autopct='%1.1f%%',
        shadow=False, startangle=190)
plt.axis('equal')
plt.tight_layout()
plt.title("How may company get funding by angels", fontdict=None, position= [0.48,1.1], size = 'x-large')

plt.show()
```

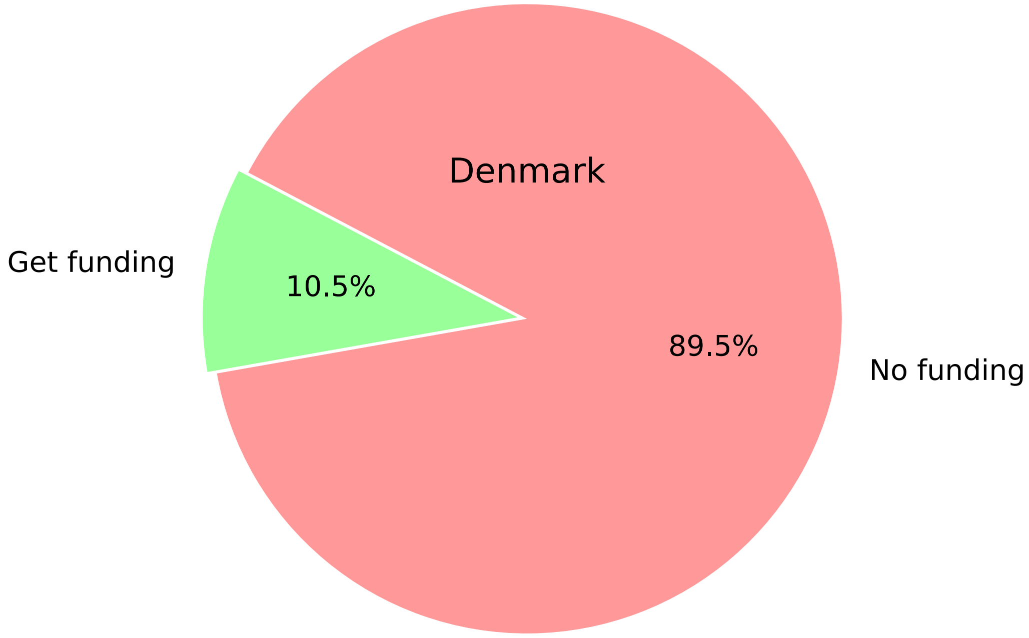


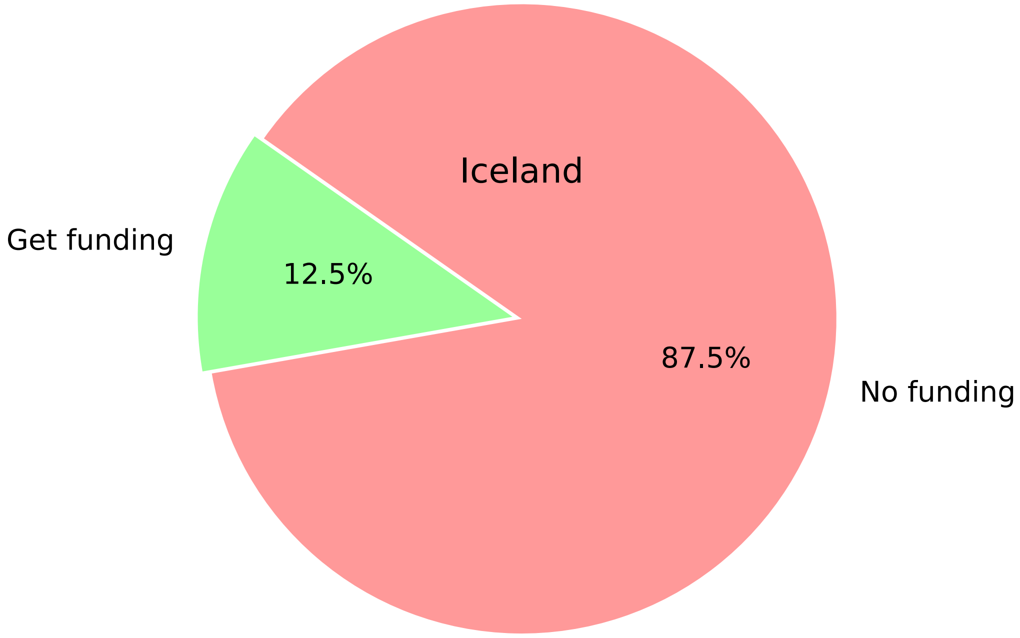
In [72]:

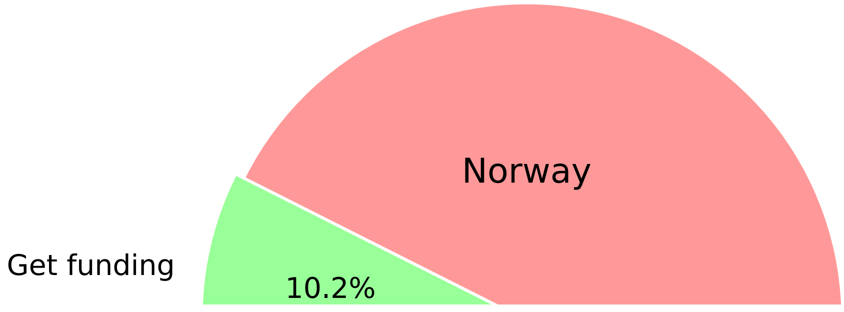
```
grouped_by_country = investments.groupby('country_code')
fin = grouped_by_country.get_group('FIN')
dnk = grouped_by_country.get_group('DNK')
isl = grouped_by_country.get_group('ISL')
nor = grouped_by_country.get_group('NOR')
swe = grouped_by_country.get_group('SWE')

fig, ax = plt.subplots(nrows=5, ncols=1, figsize = (1.5,20))

labels = ['No funding','Get funding']
sizes_fin = fin['funding_angel'].value_counts().tolist()
sizes_dnk = dnk['funding_angel'].value_counts().tolist()
sizes_isl = isl['funding_angel'].value_counts().tolist()
sizes_nor = nor['funding_angel'].value_counts().tolist()
sizes_swe = swe['funding_angel'].value_counts().tolist()
explode = (0, 0.1)
colors =  ['#ff9999','#99ff99']

ax[0].set_title("Finland")
ax[0].pie(sizes_fin, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[1].set_title("Denmark")
ax[1].pie(sizes_dnk, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[2].set_title("Iceland")
ax[2].pie(sizes_isl, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[3].set_title("Norway")
ax[3].pie(sizes_nor, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
ax[4].set_title("Sweden")
ax[4].pie(sizes_swe, explode = explode, colors = colors ,labels=labels, autopc
t='%1.1f%%',
        shadow=False, startangle=190, radius = 3.30)
fig.suptitle("Startups got funding by angels" , size = 'xx-large')
plt.show()
```

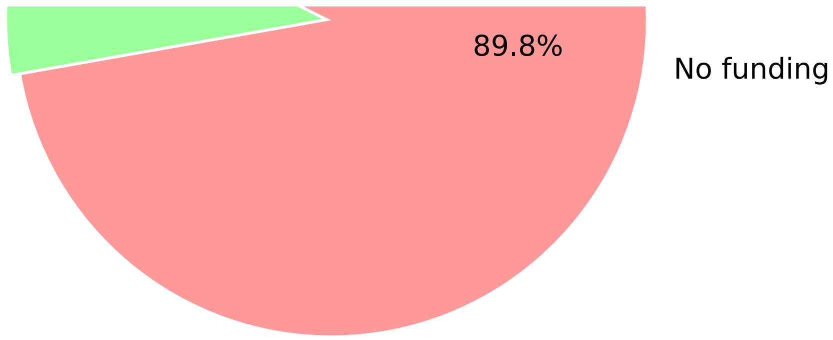


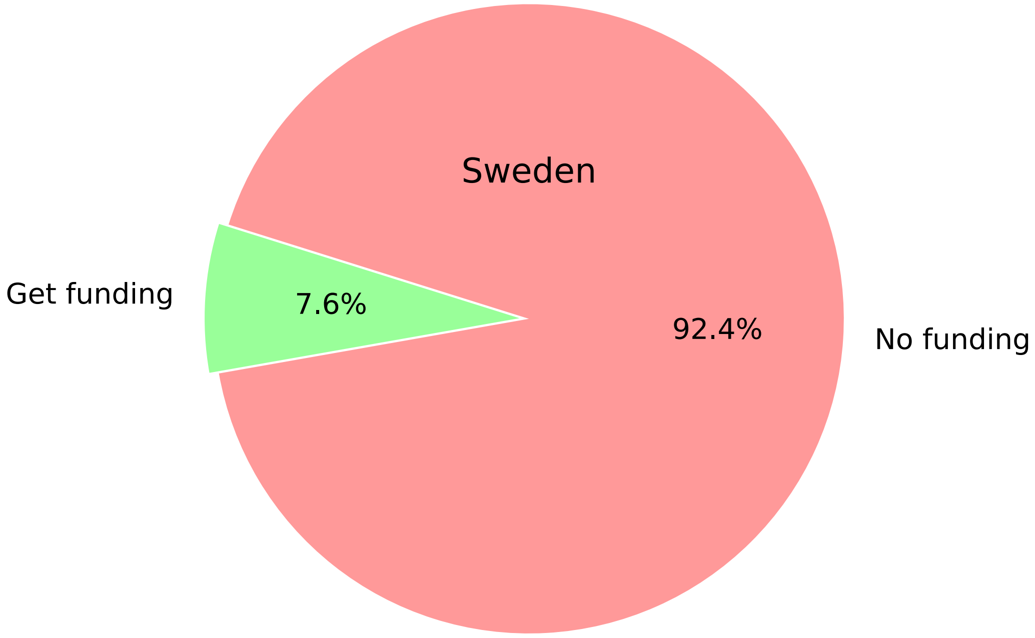

Finland
Get funding
8.2%
91.8%
No funding


Denmark
Get funding
10.5%
89.5%
No funding


Iceland
Get funding
12.5%
87.5%
No funding


Norway
Get funding
10.2%


89.8%
No funding


Sweden
Get funding
7.6%
92.4%
No funding

In [73]:

```
print('Total number of values in round_A: ', len(investments[investments['round_A'] != 0]))
print('Sum of round_A: $', investments['round_A'].sum())
print('')
print('Total number of values in round_B: ', len(investments[investments['round_B'] != 0]))
print('Sum of round_B: $', investments['round_B'].sum())
print('')
print('Total number of values in round_C: ', len(investments[investments['round_C'] != 0]))
print('Sum of round_C: $', investments['round_C'].sum())
print('')
print('Total number of values in round_D: ', len(investments[investments['round_D'] != 0]))
print('Sum of round_D: $', investments['round_D'].sum())
print('')
print('Total number of values in round_E: ', len(investments[investments['round_E'] != 0]))
print('Sum of round_E: $', investments['round_E'].sum())
print('')
print('Total number of values in round_F: ', len(investments[investments['round_F'] != 0]))
print('Sum of round_F: $', investments['round_F'].sum())
```

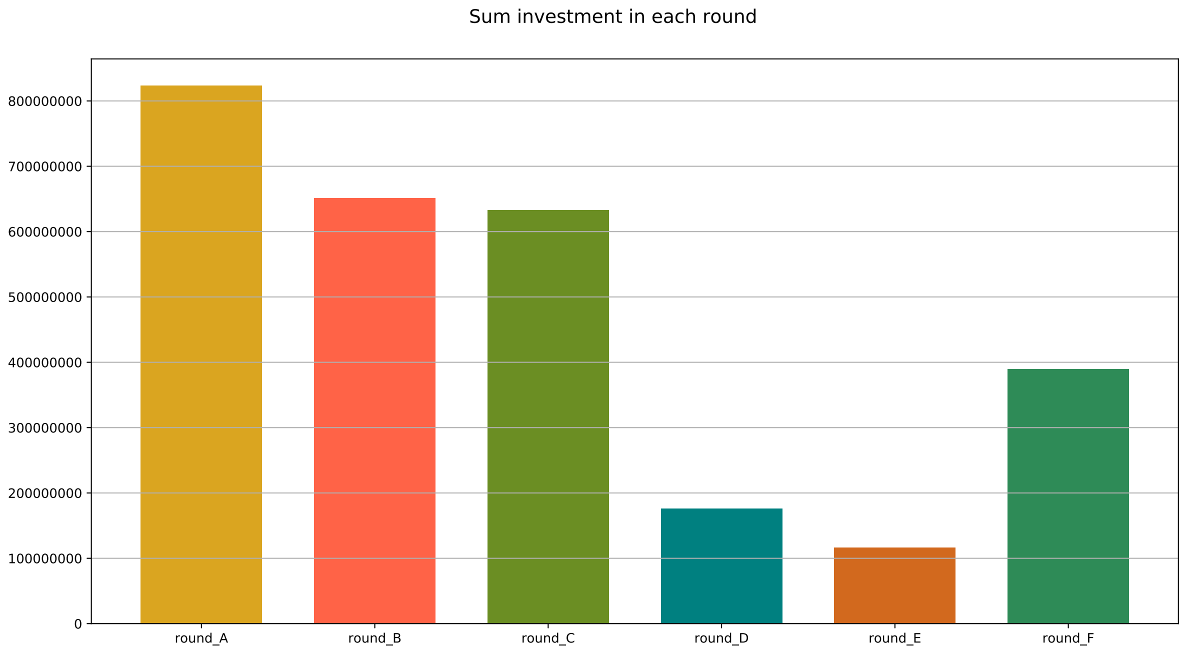
```
Total number of values in round_A:  136
Sum of round_A: $ 823159181.0

Total number of values in round_B:  59
Sum of round_B: $ 650998988.0

Total number of values in round_C:  24
Sum of round_C: $ 633185900.0

Total number of values in round_D:  6
Sum of round_D: $ 175710000.0

Total number of values in round_E:  2
Sum of round_E: $ 116000000.0

Total number of values in round_F:  3
Sum of round_F: $ 389200204.0
```

In [74]:

```
rounds = ['round_A','round_B','round_C','round_D','round_E','round_F']
amount = [investments['round_A'].sum(),
          investments['round_B'].sum(),
          investments['round_C'].sum(),
          investments['round_D'].sum(),
          investments['round_E'].sum(),
          investments['round_F'].sum()]
```

In [75]:

```
plt.rcParams['figure.figsize'] = 15,8
height = amount
bars =  rounds
y_pos = np.arange(len(bars))

plt.bar(y_pos, height , width=0.7, color= ['goldenrod','tomato','olivedrab','teal','chocolate','seagreen'] )
plt.ticklabel_format(style = 'plain')
plt.xticks(y_pos, bars)
ax = plt.axes()
ax.yaxis.grid()
plt.title("Sum investment in each round", fontdict=None, position= [0.48,1.05], size = 'x-large')
plt.show()
```



In closing: conclusions and future opportunities for research

1) **Cleanup and collaboration**

- Online collaboration tools are only as strong as their weakest link: in Google Colab’s case, if all participants have full editing access to a file, but Colab keeps asking for authorisation, you need to start hacking solutions.
- Try not to change software, tools, platforms or update py modules in the middle of a project, it may cause problems or change your output in unexpected ways.

2) **Status, scale and market segments**

- We can see that there is still a significantly larger volume of startups coming out of Sweden than

Finland, and that the startups are largely driven by the software, biotech and mobile market segments. Certain large companies such as Spotify and Supercell account for a large part of this.
- The ratio of acquired startups (in 'status') to the total seems to be lower than one would expect based on the public perception of fast-moving startup culture.
- The top 5 funded startups can be found in entertainment (Spotify leading the way), biotech (Symphogen), payments (Klarna), clean technology (NorSun), and games (Supercell).
- according to our data, startup formation has had two spikes in 2006 and 2012.

3) **Public grants, debt and Finland**

- It is likely that taxation and incentives would play a part in investment. Unfortunately, this didn't form a part of our dataset, but is a worthy area of research in the future. Here, a comparison of neighbouring countries could yield an understanding e.g. of Finland's place in the startup investment arena.
- There were 11 startups each in the categories of Mobile and Software, and 7 in Biotech, with a higher than 1 million USD investment in the Finnish market. By comparison, the Nordic figures put the number of startups with over 1 million USD investment at 50 startups (Software), 46 startups (Biotech) and 32 (Mobile). We can see that the Nordic figures reveal a higher focus on the Biotech segment than in Finland.
- Startups seem to take more debt than receive grants in Finland, although the main investment type is still venture capital, and those grants are focused mostly in the Helsinki region, with Oulu coming in second and Turku third. The further we move into the periphery, the fewer the grants in an industry that is built on decentralisation.

4) **Success stories, funding rounds and private funding**

- Data for interesting questions related to who is acquiring startups was thin. Yes, there was data related to how many startups had been acquired, but we would need further research to understand better the retention of IP in the Nordic countries or Finland.
- To this end, we would have liked to take advantage of such additional data to employ Machine Learning techniques for predictions of trends in startup investment.
- We can see that, overall, just over a quarter of startups got seed funding, which is slightly higher in Finland at 39.7 percent. Only Iceland has a higher share of startups receiving seed funding at 43.8 percent.
- When it comes to venture capital funding, however, Iceland is the lowest at 25 percent with Finland (48.5 percent) sitting roughly at the Nordic average of 46.9 percent.
- Angel funding accounts for 8.9 percent for the Nordic countries, with most companies in this area - Finland sits at 8.2 percent.

In [ ]:

```

```