

# Winery Sales

---

Ruby on Rails Application for Wineries to Improve Sales by co-creating regional Sales Figures and Graphs

- The [todo list](#) and roadmap is in the design folder

## Requirements

---

- Ubuntu 14+
- Passenger v5.0.28 and Passenger Nginx module
- Ruby version 2.3.1
- Rails version 5.0.0
- [PostgreSQL](#) v9.4.7
- [ImageMagick](#)

## Install

---

1. Clone Winery Sales into App directory
2. Copy env.yml into [config/](#) inside of App directory
3. Install PostgreSQL `sudo apt install postgresql` and ImageMagick `sudo apt-get install imagemagick`
4. Create production psql database with the user *winery* and set password
5. run whenever -w (to write the chrontabs for the email reminders & ...) (not yet implemented)

# Database

---

- Database settings located in [config/database.yml](#)

## Backup the Database / DB DUMP

---

Generate a database backup (data.dump) by running the following command in the project root: `sudo -u postgres pg_dump --oids --no-owner -Fc --disable-triggers --clean -f data.dump winery`

Transfer the `data.dump` file to a secure location using your preferred tool (rsync, cp, ftp). Use the `data.dump` file for database backup or importing into development environment.

## Restore the database from a `data.dump` file

---

Run the following command in your project root:

```
pg_restore --clean --no-owner -d winery data.dump
```

## The Data Models

---

Models associated with storing the sales data.

The heavy lifting for crunching that data is all in [sales\\_summary.rb](#) and [region.rb](#) model files.

- **Wineries**

- Describes a winery
- Has many TastingRooms

- **TastingRooms**

- Has many SalesSummaries
- Belongs to a Region

- **SalesSummaries**

- sales summaries are the main benchmark data.
- The Winery's Monthly Sales Data is stored here.
- Entry per month
- The Model File has the functions for calculating the fields of the benchmark not stored in the DB.
  - i.e. "percent\_tasters\_purchahased"

- **Regions**

- Belongs to a County
- The model file has the methods for calculating the region averages

## Other Models

---

- Counties
- Users
- WineryUsers Describes which user belongs to which winery

# Session

---

The session controller creates login and logout. The Session Helper has functions for keeping the user active in the session after login.

## Mailer

---

- All mail is sent using the MailGun API
- The mailer view is located at [views/user\\_email](#)
- The view is rendered via `render_to_string` inside of the function *send\_user\_activation\_mail* located at [controllers/concerns/user\\_mailgun.rb](#)